



Northern Telecom

OSSAIN
Open Automated Protocol
Specification

Version 7.0 - Final

Revision 0

NIS: Q235-1

August 20, 1999

Table of Contents

Chapter 1: Introduction	27
1.1 Introduction	28
1.2 Document Structure	29
1.3 Terminology	30
1.4 Reason for Reissue	35
1.4.1 Version 1.0 Revision 0 (05/96)	36
1.4.2 Version 2.0 Revision 0 (10/96)	38
1.4.2.1 Call Processing Operations	38
1.4.2.2 Data Blocks	39
1.4.2.3 Fields	40
1.4.2.4 Error IDs	41
1.4.3 Version 3.0 Revision 0 (07/97)	42
1.4.3.1 Class Headers	42
1.4.3.2 Call Processing Operations	42
1.4.3.3 Non-Call Processing Operations	44
1.4.3.4 Data Blocks	45
1.4.3.5 Fields	46
1.4.3.6 Error IDs	48
1.4.3.7 Functionality Overview	49
1.4.4 Version 4.0 Revision 0 (3/98)	51
1.4.4.1 Call Processing Operations	51
1.4.4.2 Data Blocks	53
1.4.4.3 Fields	54
1.4.4.4 Error IDs	54
1.4.4.5 Functionality Overview	56
1.4.5 Version 5.0 Revision 0 (8/98)	57
1.4.5.1 Class Header	57
1.4.5.2 Call Processing Operations	57
1.4.5.3 Non-Call Processing Operations	57
1.4.5.4 Data Blocks	57
1.4.5.5 Fields	57
1.4.5.6 Error IDs	59
1.4.5.7 Functionality Overview	59
1.4.6 Version 6.0 Revision 0 (8/98)	60
1.4.6.1 Call Processing Operations	60
1.4.6.2 Data Blocks	61
1.4.6.3 Fields	62

1.4.6.4 Error IDs	63
1.4.6.5 Functionality Overview	63
1.4.7 Version 7.0 Revision 0 (8/99)	64
1.4.7.1 Call Processing Operations	64
1.4.7.2 Data Blocks	64
1.4.7.3 Fields	64
Chapter 2: OSSAIN Overview	67
2.1 General description of OSSAIN	68
2.2 Responsibilities of the Switch	69
2.2.1 Call Processing Responsibilities	69
2.2.2 Non-Call Processing Responsibilities	69
2.3 Responsibilities of Service Nodes	71
2.3.1 Call Processing Responsibilities	71
2.3.2 Non-Call Processing Responsibilities	71
2.4 System Configuration.	73
2.4.1 Transport	74
2.4.2 Communication with Service Nodes	74
2.4.3 Voice Connections	74
Chapter 3: Protocol Description	77
3.1 Protocol Objectives	78
3.2 High Level Description of OAP	79
3.2.1 Message Format	79
3.2.1.1 OAP Class Header.	80
3.2.1.2 Operation Header	80
3.2.1.2.1 OAP Operations	82
3.2.1.3 Data Blocks	83
3.3 Deterministic Nature	85
3.4 Protocol Versioning	86
3.4.1 Protocol Version for Node Class Messages	87
3.4.2 Protocol Versioning for Session Pool Class Messages	90
3.4.3 Protocol Version for Call Processing Class Messages	92
3.4.4 Protocol Increment	93
3.4.5 Protocol Version with OSAC	95
3.4.6 Protocol Evolution Strategy	98
3.5 Sequence Numbers	100
3.5.1 Sequence Number for Node, Node Alarm, Node Log Class Headers	100
3.5.2 Sequence Number for Session Pool, Session Pool Alarm, Session Pool Log Class Headers	100
3.5.3 Sequence Number for Call Processing Class	100
3.5.4 Sequence Number for MIS Class Headers	101
3.6 Error Handling For Lost Messages	104

3.6.1 General Strategies	104
3.6.2 Switch Timers	105
3.7 General Message Flow	107
3.7.1 Call Processing Message Flow	107
3.7.2 Non-Call Processing Message Flow	110
3.7.2.1 Node Level	110
3.7.2.2 Session Pool Level	111
3.8 Service Node Considerations	114
Chapter 4: Message Format.	117
4.1 Overview Of OAP Messaging	118
4.1.1 OAP Remote Operations Interface	118
4.1.1.1 ROSE Application Framework	119
4.1.1.2 ROSE Formatting	123
4.1.2 The OAP Message Layer	127
4.1.3 OAP Class Headers	128
4.1.3.1 Message Classes	128
4.1.3.2 Protocol Version	128
4.1.3.3 Lost Message Handling	128
4.1.3.4 The OAP Class Header Fields	129
4.2 OAP Message Layouts	131
4.2.1 OAP Rules.	132
4.2.2 OAP Class Headers	132
4.2.2.1 Node Class Header	135
4.2.2.2 Node Alarm Class Header	136
4.2.2.3 Node Log Class Header	137
4.2.2.4 Session Pool Class Header	138
4.2.2.5 Session Pool Alarm Class Header	139
4.2.2.6 Session Pool Log Class Header	140
4.2.2.7 Call Processing Class Header	141
4.2.2.8 MIS Class Header	142
4.2.3 OAP Operation Headers	143
4.2.3.1 Invoke Operation Header	144
4.2.3.2 Return Result Operation Header	145
4.2.3.3 Return Error Operation Header	146
4.2.3.4 Reject Operation Header	147
4.2.4 Data Block	148
4.3 Transport Protocol Switch Information	149
4.3.1 UDP.	149
4.3.2 TCP	149
Chapter 5: Data Format	151

5.1 Data Format	152
5.1.1 Binary Fields	152
5.1.2 BCD Fields	153
5.1.3 Hexadecimal Digit Fields	154
5.1.4 Set Fields	155
5.1.5 ASCII Fields	155
5.2 Data Block Word Alignment	158

Chapter 6: OAP Error Handling 161

6.1 OAP General Error and Exception Handling	162

Chapter 7: Call Processing Operations 173

7.0 Operation Grouping Name (Template)	174
7.0.1 Request Operation	175
7.0.1.1 Success Response	177
7.0.1.2 Error Response	178
7.0.2 Inform Operation	179
7.1 Operation Groups	180
7.1.1 AMA	181
7.1.2 Billing	183
7.1.3 Call Details	185
7.1.4 Call End From the Switch	186
7.1.5 Call Float	187
7.1.6 Call Resume	188
7.1.7 Carrier	189
7.1.8 Connecting To Treatment	190
7.1.9 Conferencing	191
7.1.10 Connection Status	193
7.1.11 Directory Number	194
7.1.12 Directory Number Connection	196
7.1.13 DTMF Receiver	197
7.1.14 End Call By Active Service Node	198
7.1.15 Estimate of Charges	199
7.1.16 Service Change	200
7.1.17 Service Provider Identifier	201
7.1.18 Session Begin from the Switch to the Service Node	202
7.1.19 Session Initiation by the Service Node	203
7.1.20 Simultaneous Interactions	205
7.1.21 Speech Path	206
7.1.22 Text To Operator	207
7.1.23 Timers	208
7.1.24 Transfer	209

7.1.25 Trigger Event Inform	210
7.1.26 Voice Link	211
7.2 Accept Control InformSV3	212
7.3 Append AMA Module Request.	216
7.3.1 Append AMA Module Success Response	218
7.3.2 Append AMA Module Error Response	219
7.4 Billing Number Request	220
7.4.1 Billing Number Success Response	222
7.4.2 Billing Number Error Response	223
7.5 Call Details Request	224
7.5.1 Call Details Success Response	225
7.5.2 Call Details Error Response	229
7.6 Call End Inform	230
7.7 Call Float Request.	231
7.7.1 Call Float Success Response	235
7.7.2 Call Float Error Response	236
7.8 Call Merge InformSV4	239
7.9 Call Merge RequestSV4	241
7.9.1 Call Merge Success Response.	243
7.9.2 Call Merge Error Response	244
7.10 Call Resume RequestSV3	245
7.10.1 Call Resume Success Response.	247
7.10.2 Call Resume Error Response	248
7.11 Calling Number BNS Query RequestSV7	249
7.11.1 Calling Number BNS Query Success Response.	250
7.11.2 Calling Number BNS Query Error Response	251
7.12 Carrier Assignment Request	252
7.12.1 Carrier Assignment Success Response	253
7.12.2 Carrier Assignment Error Response	254
7.13 Class Charge Request	255
7.13.1 Class Charge Success Response	257
7.13.2 Class Charge Error Response	258
7.14 Conference Add RequestSV4	259
7.14.1 Conference Add Success Response	261
7.14.2 Conference Add Error Response	262
7.15 Conference Create Request ^{SV4}	263
7.15.1 Conference Create Success Response	265
7.15.2 Conference Create Error Response.	266
7.16 Conference Details RequestSV4	267
7.16.1 Conference Details Success Response.	268
7.16.2 Conference Details Error Response	269
7.17 Conference Release RequestSV4.	270
7.17.1 Conference Release Success Response	271

7.17.2 Conference Release Error Response	272
7.18 Conference Remove RequestSV4	273
7.18.1 Conference Remove Success Response	274
7.18.2 Conference Remove Error Response	275
7.19 Connect DN Request	276
7.19.1 Connect DN Success Response	278
7.19.2 Connect DN Error Response.	279
7.20 Connect To Treatment RequestSV6	281
7.20.1 Connect To Treatment Success Response.	282
7.20.2 Connect To Treatment Error Response	283
7.21 Connection Status Inform	284
7.22 Conversation Timing Request	285
7.22.1 Conversation Timing Success Response	286
7.22.2 Conversation Timing Error Response	287
7.23 DTMF Digit Detected Inform	288
7.24 DTMF Receiver Release Request	289
7.24.1 DTMF Receiver Release Success Response	290
7.24.2 DTMF Receiver Release Error Response.	291
7.25 Directory Number Request.	292
7.25.1 Directory Number Success Response	295
7.25.2 Directory Number Error Response	297
7.26 End Call Request.	299
7.26.1 End Call Success Response	301
7.26.2 End Call Error Response	302
7.27 Estimate of Charges RequestSV6.	303
7.27.1 Estimate of Charges Success Response	304
7.27.2 Estimate of Charges Error Response	305
7.28 Generate AMA Record Request	306
7.28.1 Generate AMA Record Success Response	307
7.28.2 Generate AMA Record Error Response	308
7.29 ISUP Calling Party Number Update RequestSV4	309
7.29.1 ISUP Calling Party Number Update Success Response	311
7.29.2 ISUP Calling Party Number Update Error Response	312
7.30 LNP RequestSV3	313
7.30.1 LNP Success Response	314
7.30.2 LNP Error Response	315
7.31 LRN Update RequestSV7	316
7.31.1 LRN Update Success Response.	317
7.31.2 LRN Update Error Response	318
7.32 Node Release InformSV3	319
7.33 Notification Timer Inform	320
7.34 Notification Timer Request	321
7.34.1 Notification Timer Success Response	323

7.34.2 Notification Timer Error Response	324
7.35 Pass Control RequestSV3	325
7.35.1 Pass Control Success Response	326
7.35.2 Pass Control Error Response.	327
7.36 Pass Through InformSV3	328
7.37 Pass Through RequestSV3.	329
7.37.1 Pass Through Success Response	331
7.37.2 Pass Through Error Response	332
7.38 Passive Function Provider RequestSV3.	333
7.38.1 Passive Function Provider Success Response	335
7.38.2 Passive Function Provider Error Response	336
7.39 Passive Node Status InformSV3	337
7.40 Release DN Request.	338
7.40.1 Release DN Success Response	339
7.40.2 Release DN Error Response	340
7.41 Release Node RequestSV3.	341
7.41.1 Release Node Success Response	342
7.41.2 Release Node Error Response	343
7.42 Return Answer Message RequestSV7	344
7.42.1 Return Answer Message Success Response	345
7.42.2 Return Answer Message Error Response	346
7.43 Route To Treatment Request	347
7.43.1 Route To Treatment Success Response	348
7.43.2 Route to Treatment Error Response	349
7.44 Service Change RequestSV4	350
7.44.1 Service Change Success Response	352
7.44.2 Service Change Error Response.	353
7.45 Service Provider ID Assignment RequestSV3.	354
7.45.1 Service Provider ID Assignment Success Response	355
7.45.2 Service Provider ID Assignment Error Response	356
7.46 Session Begin Inform	357
7.47 Session Initiation Request	362
7.47.1 Session Initiation Success Response	364
7.47.2 Session Initiation Error Response	366
7.48 Session Recall RequestSV4	367
7.48.1 Session Recall Success Response	368
7.48.2 Session Recall Error Response	372
7.49 Speech Path Request	373
7.49.1 Speech Path Success Response	374
7.49.2 Speech Path Error Response.	375
7.50 Text To Operator Request	376
7.50.1 Text To Operator Success Response	377
7.50.2 Text To Operator Error Response	378

7.51 Transfer to Carrier Request	379
7.51.1 Transfer to Carrier Success Response	381
7.51.2 Transfer to Carrier Error Response	382
7.52 Transfer to Control List Request	384
7.52.1 Transfer to Control List Success Response	386
7.52.2 Transfer to Control List Error Response	387
7.53 Trigger Event Inform	388
7.54 Voice Connect Request	392
7.54.1 Voice Connect Success Response	395
7.54.2 Voice Connect Error Response	396
7.55 Voice Release Inform	398
7.56 Voice Release Request	399
7.56.1 Voice Release Success Response	400
7.56.2 Voice Release Error Response	401

Chapter 8: Non-Call Processing Operations 403

8.0 Operation Grouping Name (Template)	404
8.0.1 Request Operation	405
8.0.1.1 Success Response	407
8.0.1.2 Error Response	408
8.0.2 Inform Operation	409
8.1 Node Operations	410
8.1.1 MIS OSSAIN Inform	411
8.1.2 Node Alarm Inform	412
8.1.3 Node Audit Request	413
8.1.3.1 Node Audit Success Response	414
8.1.3.2 Node Audit Error Response	415
8.1.4 Node Busy Request	416
8.1.4.1 Node Busy Success Response	417
8.1.4.2 Node Busy Error Response	418
8.1.5 Node Log Inform	419
8.1.6 Node Return To Service Request	420
8.1.6.1 Node Return To Service Success Response	421
8.1.6.2 Node Return To Service Error Response	422
8.1.7 Node Test Request	423
8.1.7.1 Node Test Success Response	424
8.1.7.2 Node Test Error Response	425
8.2 Session Pool Maintenance Operations	426
8.2.1 Session Pool Alarm Inform	427
8.2.2 Session Pool Audit Request	428
8.2.2.1 Session Pool Audit Success Response	429
8.2.2.2 Session Pool Audit Error Response	430
8.2.3 Session Pool Busy Request	431

8.2.3.1 Session Pool Busy Success Response	432
8.2.3.2 Session Pool Busy Error Response.	433
8.2.4 Session Pool Drain Inform.	434
8.2.5 Session Pool Log Inform	435
8.2.6 Session Pool Return To Service Request	436
8.2.6.1 Session Pool Return To Service Success Response.	437
8.2.6.2 Session Pool Return To Service Error Response	438
8.2.7 Session Pool State InformSV3	439
8.2.8 Session Pool Test Request	440
8.2.8.1 Session Pool Test Success Response	441
8.2.8.2 Session Pool Test Error Response	442
8.2.9 Session Pool Throttle Request	443
8.2.9.1 Session Pool Throttle Success Response	444
8.2.9.2 Session Pool Throttle Error Response.	445
 Chapter 9: Data Block Descriptions	 447
9.0 Data Block Template	458
9.1 Alarm Information DB	459
9.2 Alternate Billing Directory Number DB	460
9.3 AMA Module DB	463
9.4 ASCII Card Number DB.	465
9.5 BCD Card Information DB	467
9.6 BCD Card Number DB	468
9.7 BNS Query Result DB	470
9.8 Billing Acceptance DB	471
9.9 Billing Information DB	472
9.10 Billing Number Input DB	474
9.11 Billing Number Validation DB	475
9.12 Branding Indicator DB	476
9.13 Broadcast DB	477
9.14 Broadcast Count DB.	479
9.15 CCV Query Result DB	480
9.16 Call Float Indicator DB.	481
9.17 Call In Queue DB	482
9.18 Call Information DB.	483
9.19 Call Queue Information DB	487
9.20 Call Timing DB	489
9.21 Call Update DB	491
9.22 Callid DB	492
9.23 Carrier DB.	493
9.24 Carrier Number DB	495
9.25 Charge Status DB	496
9.26 Class Charge DB.	497

9.27 Conference ID DB	500
9.28 Conference Party Connection DB	501
9.29 Conference Party Connection Information DB	502
9.30 Conference SN Voice Connection DB	504
9.31 Conference SN Voice Connection Information DB	505
9.32 Connection Event DB	506
9.33 Connection ID DB	507
9.34 Connection Progress DB	508
9.35 Context Block DB	509
9.36 Conversation Timing Indicator DB	511
9.37 Country Direct Information DB	512
9.38 DTMF Digit DB	514
9.39 Directory Number DB	515
9.40 Directory Number Information DB	519
9.41 Directory Number Update DB	521
9.42 End Call Handling DB	524
9.43 Estimate of Charges Alternate Date DB.	525
9.44 Estimate of Charges Alternate Time DB	526
9.45 Estimate of Charges Duration DB	527
9.46 Estimate of Charges Duration Information DB	528
9.47 Estimate of Charges General Information DB.	529
9.48 Estimate of Charges No Duration Information DB	530
9.49 Function DB	532
9.50 Generate AMA DB	533
9.51 ISUP Calling Party Number DB	534
9.52 Information ID DB	536
9.53 Invalid Billing Details DB	538
9.54 LATA Information DB	539
9.55 LNP Information DB	540
9.56 LNP Enhancements DB.	542
9.57 LRN Update DB	543
9.58 Language DB	544
9.59 Maintenance State DB	545
9.60 Maintenance Text DB	546
9.61 NBEC DB	547
9.62 Network Service DB	548
9.63 Node Identification DB	549
9.64 Notification Timer DB	550
9.65 OLNS Information DB	552
9.66 OSSAIN Call Queue Event DB	561
9.67 OSSAIN Non-Call Session Event DB	564
9.68 OSSAIN Session Event DB	566
9.69 Operator Identification DB.	573

9.70 Operator Services System Action DB	574
9.71 Operator System DB	575
9.72 Originating Station Information DB	576
9.73 Originating Trunk Information DB	578
9.74 Pass Through DB.	579
9.75 Passive Node Status DB	581
9.76 Preprocessing Information DB	583
9.77 RLT Request DB	585
9.78 Release Cause DB	586
9.79 Release Node DB	587
9.80 Report Text DB	588
9.81 Requested Service DB	589
9.82 Routing Index DB	590
9.83 Routing Number DB	591
9.84 SN Voice Link DB	592
9.85 SPID DB	593
9.86 Sequence Number Check DB	594
9.87 Session Information DB	595
9.88 Speech Path DB	596
9.89 Terminating Trunk Information DB	597
9.90 Text DB	598
9.91 Throttle DB	599
9.92 Time DB	600
9.93 Transfer Control List DB	602
9.94 Treatment DB.	603
9.95 Trigger DTMF DB	604
9.96 Trigger Information DB	605
9.97 Trigger Profile Index DB	607
9.98 Trigger Release Cause DB.	609
9.99 Trigger Ringing DB	610
9.100 Trigger Treatment DB.	611
9.101 Voice Channel DB	612
 Chapter 10: Field Definitions	 613
10.1 ADACC Transition Indicator	625
10.2 Access Indicator	625
10.3 Alarm Severity	626
10.4 Alarm Status	626
10.5 Allowed Billing Set	627
10.6 AMA Data.	628
10.7 AMA Handling	629
10.8 AMA Hex Digit Count	630
10.9 ANI Status.	630

10.10 Application ID	631
10.11 Append AMA Sequence Number	632
10.12 ASCII Card Number	632
10.13 ASCII Card Number Length	633
10.14 ASCII Card Number Type	634
10.15 ASCII PIN Length	634
10.16 Authorization Number.	635
10.17 Authorization Number Length	636
10.18 Available Sessions	636
10.19 BCD Card Number.	637
10.20 BCD Card Number Digit Count.	637
10.21 BCD Card Number Type	638
10.22 BCD PIN Digit Count	639
10.23 BNS Query Response	639
10.24 Billable Status	640
10.25 Billing Acceptance Response Method	641
10.26 Billing Data	642
10.27 Billing Number Input Method	642
10.28 Billing Number Type	643
10.29 Branding Status	644
10.30 Broadcast Cycle Count	644
10.31 CCV Query Response	645
10.32 Call Handling	646
10.33 Call ID	647
10.34 Call In Queue Take Down	647
10.35 Call Origination Type	648
10.36 Call Queue	651
10.37 Call Type for Queuing.	651
10.38 Calls In Queue	652
10.39 Calling Line Identifier Digit Count.	653
10.40 Calling Line Identifier Number	653
10.41 Carrier Assignment Source Indicator	654
10.42 Carrier Identification Code	655
10.43 Carrier Identification Digit Count	656
10.44 Carrier Name	656
10.45 Carrier Name Length	657
10.46 Carrier Number Status.	658
10.47 Carrier Service Status	658
10.48 Class Charge Selector	659
10.49 Coin Key Pad State.	659
10.50 Coin Overtime Type	660
10.51 Conference ID	661
10.52 Connection Event	661

10.53 Connection ID	662
10.54 Connection Status	663
10.55 Context Block Data	663
10.56 Context Block Length	664
10.57 Control List ID	665
10.58 Conversation Timing Status	665
10.59 Country Direct Access Code Count	666
10.60 Country Direct Access Code Number	666
10.61 Country Direct Carrier Code Count	667
10.62 Country Direct Carrier Code Number	667
10.63 Country Direct Country Code Count	668
10.64 Country Direct Country Code Number	668
10.65 Current Broadcast Connections	669
10.66 DB Byte Count	670
10.67 DTMF Digit	670
10.68 DTMF Receiver.	671
10.69 Data Block ID	671
10.70 Day	672
10.71 Directory Number	672
10.72 Directory Number Digit Count	673
10.73 Directory Number Type	673
10.74 Elapsed Minutes	674
10.75 Error ID	675
10.76 Estimated Charges	675
10.77 Estimate Duration	676
10.78 Float Connection Indicator	677
10.79 Function ID	677
10.80 Function Provider Request Status	678
10.81 Generate AMA Sequence Number	679
10.82 Handling Data	679
10.83 Hotel Room Number	680
10.84 Hotel Room Number Length.	681
10.85 Hour	681
10.86 Invalid Billing Set	682
10.87 Invoke ID.	683
10.88 LATA Restrictions	683
10.89 LATA Status.	684
10.90 LNP Query	684
10.91 LRN	685
10.92 LRN Digit Count	686
10.93 Language.	686
10.94 Logical Voice Channel ID	686
10.95 Maintenance State	687

10.96 Maintenance Text	688
10.97 Maintenance Text Length.	688
10.98 Minutes	689
10.99 MIS Sequence Number ID	689
10.100 Month	690
10.101 NBEC Code	690
10.102 NBEC Code Digit Count	691
10.103 Network ID.	691
10.104 Network Service ID	692
10.105 No Answer Time	693
10.106 Node ID.	693
10.107 Notification Time Type	694
10.108 Notification Timer Action	694
10.109 Notification Timer Mode	695
10.110 Notification Timer Status	696
10.111 Number of Operator to SN Handoffs	696
10.112 OLNS Account Owner	697
10.113 OLNS Account Owner Length.	697
10.114 OLNS Alphanumeric String	698
10.115 OLNS Alphanumeric String Length	698
10.116 OLNS Billing Service Provider	699
10.117 OLNS Billing Service Provider Length	699
10.118 OLNS DA Credit Card Indicator	700
10.119 OLNS DA Sent Paid Indicator.	700
10.120 OLNS DA Special BNS Indicator.	701
10.121 OLNS DA Third Number Indicator	701
10.122 OLNS DACC Indicator	702
10.123 OLNS Foreign Language	703
10.124 OLNS InterLATA Carrier Indicator	705
10.125 OLNS InterLATA Carrier Identification Digit Count	706
10.126 OLNS InterLATA Carrier Identification Code	706
10.127 OLNS International Carrier Indicator	706
10.128 OLNS International Carrier Identification Digit Count	707
10.129 OLNS International Carrier Identification Code	707
10.130 OLNS IntraLATA Toll Carrier Indicator.	708
10.131 OLNS IntraLATA Toll Carrier Identification Digit Count	709
10.132 OLNS IntraLATA Toll Carrier Identification Code	709
10.133 OLNS Service/Equipment Indicator	710
10.134 OLNS TA Collect Indicator.	711
10.135 OLNS TA Credit Card Indicator	712
10.136 OLNS TA Sent Paid Indicator	713
10.137 OLNS TA Special BNS Indicator	713
10.138 OLNS TA Third Number Indicator	714

10.139 OLNS Treatment Indicator	714
10.140 Operation ID	716
10.141 Operator ID.	717
10.142 Operator Services System Action	717
10.143 Operator System Selector	718
10.144 Operator System Type	719
10.145 Party	720
10.146 Pass Through Data	720
10.147 Pass Through Data Length	721
10.148 Passive Node Status	721
10.149 Phone Class Type	722
10.150 Portability Status	724
10.151 Predicted Wait Time	725
10.152 Presentation Indicator	726
10.153 Protocol Version	726
10.154 Queue Peg Indicators.	727
10.155 Queuing Event.	728
10.156 Queuing System	728
10.157 RAO Code	729
10.158 RAO Digit Count	730
10.159 Release Node Indicator	730
10.160 Report Text.	731
10.161 Report Text Length	732
10.162 Requested RLT Action	732
10.163 Reset Time Flag	733
10.164 Restart Severity	733
10.165 Restricted Billing Index	734
10.166 Routing Index	735
10.167 SN Voice Conferencing Status.	735
10.168 SN Voice Link Status	736
10.169 SPID Processing	736
10.170 SS7 Backward Call Indicators	737
10.171 SS7 Release Cause	740
10.172 Seconds	740
10.173 Sequence Number.	740
10.174 Service Node Mode	741
10.175 Service Peg Indicators	742
10.176 Service Provider ID	742
10.177 Service Provider ID Length.	743
10.178 Service Provider Party Identifier	743
10.179 Service Type	744
10.180 Service Type Provided by Switch	745
10.181 Serving Agent Status.	745

10.182 Session ID	746
10.183 Session Origination Type	747
10.184 Session Pool ID	747
10.185 Session State	748
10.186 Session Type	749
10.187 Solicitor Number	749
10.188 Special Number Format	750
10.189 Special Number Type	750
10.190 Special Number Validity	751
10.191 Speech Path Status	752
10.192 Tenths of Seconds Timestamp	752
10.193 Tenths Seconds	753
10.194 Terminal ID	753
10.195 Terminating Code Allowed Set	754
10.196 Text Block	755
10.197 Text Block Length	756
10.198 Time in Queue	756
10.199 Treatment	757
10.200 Trigger Detection Point	757
10.201 Trigger Event	758
10.202 Trigger Profile Index	758
10.203 Trunk Group ID	759
10.204 Trunk Member ID.	760
10.205 Type of Service Provider	760
10.206 Unenforced Restrictions.	761
10.207 Validation Request	761
10.208 Validation Result	762
10.209 Validation Result Determined by DMS Datafill	763
10.210 Validation Sequence Number	764
10.211 Voice Channel ID	764
10.212 Year	765

Chapter 11: Error ID Values 767

Chapter 12: Message Flows 777

12.1 AMA Message Flows	778
12.1.1 Append Billing Data (Success)	778
12.1.2 Append Billing Data (Failure)	779
12.1.3 Gen AMA (Success)	780
12.1.4 Gen AMA (Failure)	781
12.1.5 Billing Number and Class Charge Request Call Flow (Success)	782
12.1.6 Billing Number and Class Charge Request Call Flow (Failure)	783

12.1.7 Bill to Alternate Directory Number, Verification Required	784
12.2 Call Float Call Flows	786
12.2.1 Call Float (Success)	786
12.2.2 Call Float (Failure).	787
12.3 Class Charge Call Flows	788
12.3.1 Class Charge (Success)	788
12.3.2 Class Charge (Failure).	789
12.4 Conferencing	790
12.4.1 Consult Phase	791
12.4.2 Conference Phase	793
12.4.3 Party A disconnects during consult phase.	794
12.4.4 Party B disconnects during consult phase.	795
12.4.5 Party C disconnects during consult phase.	797
12.4.6 Party A releases party C consult phase	798
12.4.7 Party A disconnects during conference phase	799
12.4.8 Party B disconnects during conference phase	801
12.4.9 Party C disconnects during conference phase	803
12.5 Connection Status Call Flows	805
12.5.1 Connection Status Inform (Release after ACM).	805
12.6 DN Connection Call Flows	807
12.6.1 Connect DN Request (Non-ISUP Trunk, Success)	807
12.6.2 Connect DN Request (ISUP Trunk Outpulse Forward, Success)	809
12.6.3 Connect DN Request (ISUP Trunk Outpulse Backward, Success)	811
12.6.4 Connect DN Request (Failure on Outpulsing)	813
12.6.5 Connect DN Request (ISUP Trunk, Failure, Release with Cause)	815
12.6.6 Release DN Request (Success, non-ISUP Trunk)	817
12.6.7 Release DN Request 'equivalent' (Success, ISUP Trunk).	819
12.6.8 Release DN Request (Failure)	820
12.7 LNP Request	821
12.7.1 LNP Request (Success)	821
12.7.2 LNP Request (Failure)	822
12.8 Session Origination From Switch to Service Node Call Flow	823
12.9 Session Origination From Service Node to Switch Call Flow	824
12.10 Simultaneous Interaction Call Flows	825
12.10.1 Connecting a Passive Function Provider.	825
12.10.1.1 Connecting a Passive Service Node - Session Obtained Immediately	826
12.10.1.2 Connecting a Passive Service Node - Call Queues for Session	827
12.10.1.3 Connecting a Passive Service Node - Deflection or Overflow	828
12.10.1.4 Connecting a Passive Operator - Operator Obtained Immediately	829
12.10.1.5 Connecting a Passive Operator - Call Queues for Operator	830
12.10.1.6 Connecting a Passive Operator - Deflection or Overflow	831
12.10.2 Releasing a Passive Function Provider	832
12.10.2.1 Releasing a Passive Node at the Passive Node's Request	833

12.10.2.2 Releasing a Passive Node at the Active Node's Request	834
12.10.3 Releasing a Passive Node Due to a Call Disposition	835
12.10.4 Switch Releasing a Passive Node Due to an Error Condition	836
12.10.5 Passing Call Control	837
12.10.6 Pass Control at Active Service Node's Request	838
12.10.7 Pass Control Due to Active Service Node Release	839
12.10.8 Pass Control at Passive Operator's Request.	840
12.10.9 Pass Through Messaging	841
12.11 Speech Path Call Flows	843
12.11.1 Speech Path Request (Success)	844
12.11.2 Speech Path Request (Failure).	845
12.12 Text To Operator Message Flows	846
12.12.1 Text To Operator (Success).	846
12.12.2 Text To Operator (Failure)	847
12.13 Transfer to Carrier Message Flows	848
12.13.1 Transfer to Carrier (Success)	848
12.14 Transfer to Function Provider Message Flows	849
12.14.1 Transfer to Control List (Success)	849
12.14.2 Transfer to Control List (Failure)	850
12.15 Voice Link Message Flows	851
12.15.1 Connect Voice Link (Success).	851
12.15.2 Connect Voice Link (Failure)	852
12.15.3 Release Voice Link (Success)	853
12.15.4 Release Voice Link (Failure)	854
12.16 Non-Callp Node Based Message Flows	855
12.16.1 Node Audit Request - (Success)	855
12.16.2 Node Audit Request - (Failure)	856
12.16.3 Node Busy Request by Switch - (Success)	857
12.16.4 Node Busy Request by Switch - (Failure)	858
12.16.5 Node Busy Request by SN - (Success)	859
12.16.6 Node Test Request - (Success).	860
12.16.7 Node Test Request - (Failure)	861
12.16.8 Node Return To Service Request - Automated(Success).	862
12.16.9 Node Return To Service Request - Manual (Success)	863
12.16.10 Node Return To Service Request - (Failure)	864
12.16.11 Node Log/Alarm Inform	865
12.17 Non-Callp Session Pool Based Message Flows	866
12.17.1 Session Pool Audit Request (Success)	866
12.17.2 Session Pool Audit Request - (Failure)	867
12.17.3 Session Pool Busy Request by Switch - (Success)	868
12.17.4 Session Pool Busy Request - (Failure)	869
12.17.5 Session Pool Busy Request by SN - (Success).	870
12.17.6 Session Pool Test Request - (Success)	871

12.17.7 Session Pool Test Request - (Failure)	872
12.17.8 Session Pool Return To Service Request - Automated (Success)	873
12.17.9 Session Pool Return To Service Request - Automated (Failure)	874
12.17.10 Session Pool Throttle Request	875
12.17.11 Session Pool Log/Alarm Inform.	876
12.18 Example Call Flows	877
12.18.1 Call Billed to a Calling Card	878
12.18.2 Sequence Call	912

Chapter 13: Functionality Overview	953
13.1 Automated Alternate Billing Service (AABS) - Transition from an SN	954
13.2 Directory Assistance	955
13.3 Equal Access	956
13.3.1 Service Node Initiated Session: Connecting to the A Party	957
13.3.2 Service Node Initiated Session: Not Connecting the A Party	958
13.4 Estimated Charges	959
13.4.1 Estimate Time	959
13.4.2 Estimate Date	960
13.4.2.1 Estimate Time and Date Interactions	960
13.4.3 Estimates with/without a specified duration	961
13.4.4 AMA considerations	962
13.5 Local Number Portability	963
13.5.1 Background	963
13.5.2 LNP Information	964
13.5.3 LNP SOC	964
13.6 Maintenance	966
13.7 OSSAIN Centralization	967
13.8 OSAC Call Scenario	968
13.8.1 Parallel Datafill	969
13.9 OSSAIN Preprocessing	970
13.10 Queue Management System Management Information System	972
13.10.1 General Message Description	972
13.10.2 Pegging Strategy	972
13.10.2.1 Work Volume and Idle Time Pegging	973
13.10.2.2 Time of Day and Tenths of Seconds Pegging	974
13.10.2.3 Suggested Algorithm for MIS Pegging	974
13.10.3 MIS and the OSAC Environment	976
13.10.4 MIS message buffer	977
13.10.5 MIS data block generation	978
13.11 Release Link Trunking	983
13.11.1 RLT SOC	985
13.12 Simultaneous Interactions	986
13.12.1 Active Function Provider	986

13.12.2 Passive Function Provider	986
13.12.3 Non-Switch Connection (NSC)	987
13.12.4 Simultaneous Interaction Between TOPS TA Operator and A Service Node	987
13.13 Voice Connections	990
13.13.1 Switch/Service Node parallel functionality/datafill	991
13.13.2 Standard voice link operation	992
13.13.2.1 Signalling on standard voice links	992
13.13.3 Broadcast voice link operation.	992
13.13.3.1 Signalling on broadcast voice links	995
13.13.3.2 Broadcast voice links in simultaneous interactions	996
13.13.3.3 Voice Link Broadcasting in OSAC	996
13.13.4 Network delay allowance for all voice links	997
13.13.5 Trunk guard timing interactions for all voice links	997
13.13.6 Increased OAP messaging for broadcast voice link failure	998
13.13.7 Recommendations for cycling broadcast recordings	998
 Appendix A: ASN.1 Encoding of the OAP Operation Header	 1003
A.1 ASN.1 Syntax & Encoding	1004
A.1.1 ASN.1 Definitions of ROSE Operations	1004
A.1.1.1 ASN.1 Definition of Invoke Operation	1004
A.1.1.2 ASN.1 Definition of ReturnResult Operation	1004
A.1.1.3 ASN.1 Definition of ReturnError Operation	1005
A.1.1.4 ASN.1 Definition of Reject Operation	1005
A.1.2 General ASN.1 Encoding (Basic Encoding Rules)	1006
A.1.2.1 Encoding of TYPE	1006
A.1.2.2 Encoding of LENGTH	1008
 Appendix B: Message Set	 1009
B.1 Summary of Operations	1010
B.1.1 Call Processing Request Operations	1011
B.1.2 Call Processing Inform Operations	1013
B.1.3 Non-Call Processing Request Operations	1014
B.1.4 Restricted Operations	1015
B.2 Identifier Cross Referencing	1018
B.2.1 Operation Identifiers	1019
B.2.2 Data Block Identifiers	1022
B.3 Treatments	1025

List of Figures

Figure 1.	OSSAIN Topology	. 73
Figure 2.	Node Level Protocol Versioning	. 89
Figure 3.	Session Pool Level Protocol Versioning	. 91
Figure 4.	Protocol Version Error During Call Processing	. 93
Figure 5.	Message Protocol Version with Increment	. 94
Figure 6.	Service Node Initiated Call to an OSAC Remote	. 97
Figure 7.	Sequence Number Example	102
Figure 8.	Basic Subscriber Originated Call Processing Message Flow	109
Figure 9.	Busy Node Message Flow	112
Figure 10.	Return To Service Message Flow	113
Figure 11.	Invocation of a Remote Operation	119
Figure 12.	RO-Reject: Problem Types and Reject Reasons	122
Figure 13.	OAP Operation Format	125
Figure 14.	General Correlation of OAP to OSI Model	127
Figure 15.	OAP Header/Data Format	131
Figure 16.	Data Block Word Alignment with Fixed Length Fields	158
Figure 17.	Data Block Word Alignment with Variable Field Length Fields	159
Figure 18.	OSAC Topology	969
Figure 19.	OSSAIN Call gets session no queuing	978
Figure 20.	OSSAIN Call gets session with queuing	979
Figure 21.	Example MIS data block flows.	980
Figure 22.	Example MIS message flows continued.	981
Figure 23.	Example MIS message flows continued.	982
Figure 24.	Example call flow for simultaneous interaction	989
Figure 25.	Voice Link Broadcasting to 5 separate calling parties	993
Figure 26.	VL broadcasting with CF3P in 1 call, along with 2 other calls	993
Figure 27.	Voice Link Broadcasting in an OSAC environment.	997
Figure 28.	TYPE Format	1006
Figure 29.	LENGTH Format	1008

List of Tables

Table 1.	OAP Version to TOPS release	35
Table 1.	Message Type and Associated Operation Header	81
Table 2.	Message Type and Potential Responses	82
Table 3.	Operation Group to Class Type	82
Table 4.	Common Class Header Fields.	86
Table 5:	Invokes/Responses	120
Table 6:	Class Header Sections	132
Table 7.	ASCII Character Set	156
Table 8.	DMS OAP Incoming Message Error Handling	163
Table 9:	Data Blocks to Operations Mapping Table.	448
Table 10:	Valid Class Charges.	498
Table 11:	Trigger Events for each Trigger Detection Point	606
Table 12.	Fields To Data Blocks Mapping Table	614
Table 13:	Non-Call Processing Class Error IDs	768
Table 14:	Call Processing Error ID Values Which Do Not Result in Call Updates	769
Table 15:	Call Processing Error ID Values Which May Result in Call Updates .	770
Table 17.	OSAC Components	967
Table 18.	Example MIS Buffer with TCP Header.	977
Table 19.	Signalling used on broadcast voice links	995
Table 20.	Example order of broadcast voice link selection	1000
Table 21:	ASN.1 General Format	1006
Table 22:	Some Universal-Class Types (Defined in ASN.1)	1007
Table 23.	Call Processing Request Operations.	1011
Table 24.	Call Processing Inform Operations	1013
Table 25.	Non-Call Processing Request Operations	1014
Table 26.	Restricted Operations	1015
Table 27.	Identifier to Operation Cross Reference	1019
Table 28.	Identifier to Data Block Cross Reference	1022

Chapter 1: Introduction

1.1 Introduction

This document provides the functional description and interface specification for the interworking between a DMS TOPS switch and an external computer, referred to as a Service Node (SN) in this document. The Open Automated Protocol (OAP) is a component of the Operator Services Systems Advanced Intelligent Network (OSSAIN) product.

OAP is used as a medium for OSSAIN to provide service nodes with powerful capabilities of operator services, permitting services to be implemented and deployed independent of switch software development cycles. OAP is used to identify calls to service nodes, to allow service nodes to request specific actions, and to inform service nodes of changing call characteristics. The protocol is also used to inform service nodes of certain non-call related events that occur in the switch, and to perform maintenance on service nodes.

The following documents are intended to provide sufficient information to allow any operating company or third-party vendor to develop new services on external OSSAIN service nodes:

- Northern Telecom Translations Guide - Provides guidelines for the preparation, datafill, and validation activities associated with a DMS TOPS switch.
- OAP Security Document (NIS Q235-1-S) - Provides details on how vital data is secured within OAP.
- Operator Services Systems Advanced Intelligent Network User's Guide (NTP 297-8403-901) - Describes OSSAIN functionalities and how they work together to deliver services. It provides an overview of OSSAIN, a detailed description of the software, and supplementary information on datafill and maintenance activities.
- OSSAIN API User's Guide (DOSIS 100-1) - This document accompanies the API software for OSSAIN. It describes how to build and configure an application using the API.
- OSSAIN OAP Specification Document (NIS Q235-1) - Provides the description and specifications for the Open Automated Protocol.
- TOPS Context Block Document (NIS Q258-1) - This document provides information on the context block data format.
- TOPS Custom AMA Document (NIS Q259-1) - This document provides information on the custom AMA data format.

1.2 Document Structure

The Open Automated Protocol (OAP) Specification document is divided into 14 chapters and 2 appendices. A brief description of each chapter and appendix follows.

- Introduction - Provides an introduction to the protocol document and includes a list of definitions and acronyms. It also describes the revision history for each version of this document.
- OSSAIN Overview - Describes Operator Services AIN, listing responsibilities of the switch and service nodes. It also contains the OSSAIN system configuration.
- Protocol Description - Describes the Open Automated Protocol.
- Message Format - Provides a detailed description of the message format, describing the OAP headers and the format of the message body.
- Data Format - Provides a description of the formats used for fields and data blocks. It also explains some characteristics of OAP such as data block word alignment and the byte ordering used for individual fields.
- Error Handling - Describes the valid ranges for fields in all the Class Headers and Operation headers. It also states how the switch responds to an OAP message with an error in a specific field.
- Call Processing Operations - Describes in detail all of the call processing operations provided by OAP (i.e. operations that are call related).
- Non-Call Processing Operations - Describes in detail all of the non-call related operations provided by OAP (i.e. session pool level maintenance, node level maintenance).
- Data Block Descriptions - Describes in detail all of the data blocks provided by OAP. It also provides a mapping table from data blocks to operations.
- Field Definitions - Describes in detail each of the fields used in the data blocks. It also provides a mapping table from fields to data blocks.
- Error Values - Describes the possible errors that can be received for OAP operations.
- Message Flows - Provides sample message flows, demonstrating how some operations can be used and in what context they can be used.
- Functionality Overview - Describes in detail some OSSAIN functionality.
- Abstract Syntax Notation (Appendix) - Describes ASN encoding.
- Message Set (Appendix) - Tables of operations in specific groupings.

1.3 Terminology

<u>Term</u>	<u>Definition</u>
AABS	Automated Alternate Billing Service
ADACC	Automated Directory Assistance Call Completion
ADAS	Automated Directory Assistance Service
AE	Application Entity
AIN	Advanced Intelligent Network
AMA	Automatic Message Accounting
APDU	Application Protocol Data Unit
API	Application Programmer's Interface
ASCII	American Standard Code for Information Interchange
ASN	Abstract Syntax Notation
Active Function Provider	In a simultaneous interaction, the Active Function Provider is the entity that is directly in control of the TOPS OSSAIN call.
Agents	Abstractions which represent external entities that can connect with the switch (e.g. subscribers, carriers, conference port circuits, operator voice and data, service node voice and data).
BER	Basic Encoding Rules
BCD	Binary Coded Decimal
CBWV	Call Busy Work Volume
CCITT	International Telegraph and Telephone Consultative Committee
CIID	Card Issuer Identifier
CT4Q	Call Type for Queuing
Call Processing Switch	An Operator Services switch that communicates with a service node for call related activities. The switch is either a Standalone switch or Remote switch for the service node.
Control List	A list of functions applied sequentially to the call. All subscriber-originated calls are assigned a Control List. A new control list can be invoked through a request from either a service node or a live operator.
DA	Directory Assistance
DMS	Digital Multiplexing Switch

<u>Term</u>	<u>Definition</u>
DTMF	Dual Tone Multifrequency
EIU	Ethernet Interface Unit
Equal Access	Equal Access provides the ability to have a presubscribed carrier. It also allows a subscriber to override their presubscribed carrier by dialling 10XXX. An end office that offers equal access is known as a conforming or equal access end office. An end office that does not offer equal access is known as a non-conforming end office.
Event	A detectable condition that occurs during a call. An event is a potential trigger.
Float	The process of a service node, TOPS operator, or an existing TOPS automated system releasing from the call — prior to establishing a connection between the parties, — after establishing a connection to the forward party but before the forward party answers, or — while the parties are connected and talking.
Function	A specific, high level task, such as branding or alternate billing, performed by either a service node, a TOPS operator, or an existing TOPS automated system. A Function is an entry in a control list.
Function Provider	An entity in the call that is providing some function. A Function Provider can be either a service node, a TOPS operator, or an existing TOPS automated system.
IAM	Initial Address Message
IC	InterLATA Carrier
ICMP	Internet Control Message Protocol
IDLT	Idle Time
IP	Internet Protocol
ISO	International Standards Organization
ISS	Initial Session Seizure
ISUP	ISDN User Part. A type of network control and signalling using separate voice and data links. Also a trunk type using this type of signalling.
LAN	Local Area Network
LATA	Local Access and Transport Area
LEC	Local Exchange Carrier

<u>Term</u>	<u>Definition</u>
LIDB	Line Information Data Base
LNP	Local Number Portability. Refer to the Functionality Overview section on LNP for more information.
LRN	Location Routing Number. Refer to the Functionality Overview section on LNP for more information.
MIS	Management Information System
Maintenance Switch	The switch that communicates with a service node for audit and non-call processing activities.
NBEC	Non-Bell Exchange Carrier
NCWV	Non Call Work Volume
Node Identifier	A value that uniquely identifies an OSSAIN node (switch or service node) within an OSSAIN network.
Non-Switch Connection	A connection between two function providers that does not involve messaging through the TOPS switch. (NSC)
OAP	Open Automated Protocol
OLNS	Originating Line Number Screening
OM	Operational Measurements
OPDU	Operation Protocol Data Units
OPP	Open Position Protocol
OSI	Open Systems Interconnection
OSSAIN	Operator Services Systems Advanced Intelligent Network
Operation	An OAP action request (e.g. connect voice links) or informational message (e.g. subscriber party hook status).
OSAC	OSSAIN Centralization (OSAC) provides the ability to centralize service nodes at an OSAC Host switch. Call distribution, maintenance, and voice link connections to a service node are centralized at a Host switch. Call processing with a service node is done at both the Remote switch and Standalone switch. Refer to the Functionality Overview section on OSAC for more information.
OSAC Host Switch	The switch in an OSAC environment that supports a centralized service node. Call distribution, maintenance, and voice link connections for the service node are centralized at a Host switch. A Remote switch uses the Host switch for call distribution and voice connections to the centralized service node.

<u>Term</u>	<u>Definition</u>
OSAC Remote Switch	A call processing switch in an OSAC environment that requires the services of a centralized service node. The Remote switch obtains a session to a service node from the Host switch. The Remote switch and service node interact only for call processing. Maintenance of the service node is performed with the Host switch.
OSS	Operator Services System
OSSAIN Node	A switch or service node in the OSSAIN network.
PVE	Protocol Version Error
Parties	Abstractions which represent the subscribers involved in a call (e.g. the calling party, called party, etc.).
Passive Function Provider	In a simultaneous interaction, the Passive Function Provider is the entity that is participating in, but is not in control of, a TOPS OSSAIN call. The Passive Provider cannot directly affect call processing through interactions with the TOPS switch.
QMS	Queue Management System
RLT	Release Link Trunking. Refer to the Functionality Overview section on RLT.
RMB	Remote Make Busy
ROSE	Remote Operations Service Element
RTS	Return To Service
Restricted Input Mode	The mode the switch is in for a specific call when it receives a restricted operation. No other requests, except the End Call Request, from the service node for the call is accepted while the switch is in this mode. The service node can send the End Call Request during this time in order to end the call. Any other requests will be processed as an error. This is on a per call basis.
Restricted Operation	An operation that causes the switch to send a message to another process and wait for a response. The switch will not accept any messages from the service node for the call while it is processing the restricted operation. The service node should wait until it receives the response message before sending additional messages to the switch for this call.
SNOS	Service Node Origination Seizure
SOC	Software Optionality Control
SS7	Signaling System #7. A messaging protocol for controlling voice connections. Used on the data link of ISUP trunks.
STP	Signalling Transfer Point

<u>Term</u>	<u>Definition</u>
Service Node	An external computer that interacts with the switch to provide OSSAIN functions.
Session	A logical agent on a service node capable of controlling the call.
Session Identifier	A numeric representation of a session. A session identifier is in use starting with the initial message sent from either the switch or a service node and ending with the last message of the session.
Session Pool	A logical grouping of sessions (i.e. agents) established to serve one or more functions. Session pool IDs are unique within a given OSSAIN network.
Simultaneous Interaction	An interaction involving at most two function providers (i.e. a TOPS operator and a service node, or two service nodes) simultaneously connected to a call to provide a service. However, only one function provider is the active function provider (i.e. in control of the call). Refer to the Functionality Overview section on Simultaneous Interactions for more information.
Standalone Switch	A call processing switch which performs call distribution, voice link connections, and maintenance for a given service node. The switch also has call processing with the service node. It does not need to communicate with another switch regarding the service node.
TA	Toll and Assist
TGSS	Trigger Session Seizure
TOPS	Traffic Operator Position System
TXSS	Transfer Session Seizure
Throttle	To change the number of sessions available within a session pool. This sets the number of sessions a session pool has available to use.
Transition	The current service node, operator, or automated system is released and the TOPS switch is temporarily in control of the call until the connection to the subsequent service node, operator, or automated system is established.
Trigger	An event that occurs and causes the call to be redirected to a service node or a TOPS operator.
UDP	User Datagram Protocol
WAN	Wide Area Network

1.4 Reason for Reissue

The following rules apply for the version numbering of this document: Version x.y Revision z

- Version x.y: x is the release and y is the increment
- The release and increment of this document is the same as the release and increment of the OAP software released with this document. The version of this document is incremented each time the OAP version is incremented. For example, if this document is released with OAP version 2.3, the version of this document will be 2.3.
- The revision number is zero when the document is released with OAP version X.0 software. If there are any document only changes (i.e. changes to this document but no software changes to OAP or OSSAIN), then the revision number is incremented.

The following table specifies what OAP versions are supported for a given TOPS release. Note, that all increments of an OAP version will be supported in the specified TOPS release.

Table 1. OAP Version to TOPS release

	OAP 2.0	OAP 3.0	OAP 4.0	OAP 5.0	OAP 6.0	OAP 7.0
TOP 06	X					
TOP 07	X	X				
TOP 08	X	X				
TOP 8.1	X	X				
TOP 09	X	X	X			
TOP 10	X	X	X	X		
TOP 11		X	X	X	X	
TOP 12			X	X	X	X

The following sections indicate what has been changed for a specific OAP Version. The sections are divided by the OSAN SOC option. The OSSAIN SOC options currently supported are listed below:

- OSAN0002 (Operator Services AIN): Introduced with OAP Version 1.0
- OSAN0003 (Operator Services AIN Version 3.0): Introduced with OAP Version 3.0

In the Call Processing chapter:

- Each new operation included with this SOC option is identified by the superscript “SV3”.

- The data blocks within existing operations that are controlled by this SOC option are identified with the superscript “SV3”.

In the Data Blocks chapter:

- Fields within existing data blocks that are controlled by this SOC option are identified with the superscript “SV3”.

- OSAN0004 (Operator Services AIN): Introduced with OAP Version 4.0

In the Call Processing chapter:

- Each new operation included with this SOC option is identified by the superscript “SV4”.
- The data blocks within existing operations that are controlled by this SOC option are identified with the superscript “SV4”.

In the Data Blocks chapter:

- Fields within existing data blocks that are controlled by this SOC option are identified with the superscript “SV4”.

- OSAN0005 (Operator Services AIN): Introduced with OAP Version 5.0

In the Call Processing and Non-Call Processing chapters:

- Each new operation included with this SOC option is identified by the superscript “SV5”.
- The data blocks within existing operations that are controlled by this SOC option are identified with the superscript “SV5”.

In the Data Blocks chapter:

- Fields within existing data blocks that are controlled by this SOC option are identified with the superscript “SV5”.

- OSAN0006 (Operator Services AIN): Introduced with OAP Version 6.0

In the Call Processing and Non-Call Processing chapters:

- Each new operation included with this SOC option is identified by the superscript “SV6”.
- The data blocks within existing operations that are controlled by this SOC option are identified with the superscript “SV6”.

In the Data Blocks chapter:

- Fields within existing data blocks that are controlled by this SOC option are identified with the superscript “SV6”.

1.4.1 Version 1.0 Revision 0 (05/96)

Initial release of OAP as an interface between the DMS TOPS switch and service nodes for OSSAIN.

All functionality is included as part of the OSSAIN SOC option OSAN0002.

1.4.2 Version 2.0 Revision 0 (10/96)

All changes are included as part of the OSSAIN SOC option OSAN0002.

1.4.2.1 Call Processing Operations

1. Call Details Success Response: A treatment may be applied to either the A party or B party port when a call is transferred to a service node. An optional treatment data block is added to the Call Details Success Response. This data block indicates the treatment applied and will be sent if the call is connected to a treatment. An optional OLNS Information DB is added.
2. Call Float Request: Added two optional data blocks, Billing Acceptance DB and Operator Services System Action DB. These data blocks contain billing information required for calls billed to a calling card, third number, or collect calls.
3. Carrier Assignment Request: This existing operation request is changed. The mandatory Carrier DB for the request is replaced with the new Carrier Number DB. The service node now just needs to specify the carrier number only and not all the other information that is in the Carrier DB. The Carrier Assignment Success Response has also been changed. The Carrier DB is returned with the response. This existing data block returns information on the assigned carrier. Three new possible error ids were added to the Carrier Assignment Error Response (Translations failed, treatment applied; Carrier not allowed for this call).
4. Directory Number Request: An error id is added to the error response: #0724, Call is setup to be transferred to a carrier. The directory number cannot be changed since the call is ready to be transferred to a carrier.
5. End Call Request: Replaced the Abort Call DB with the new End Call Handling DB. The End Call Handling DB is similar to the Abort Call DB, with one field expanded (Call Handling). Two new optional data blocks are also added: Billing Acceptance DB and Operator Services System Action DB. These data blocks contain required information for calls billed to a calling card number, a third number, or billed collect.
6. Session Begin Inform: A treatment may be applied to either the A party or B party port when a call is transferred to a service node. An optional treatment data block is added to the Session Begin Inform message. An optional OLNS Information DB is added.
7. Session Initiation Request: This operation is changed. The optional Carrier DB is replaced with the new optional Carrier Number DB. The service node now just needs to specify the carrier number only and not all the other information that is in the Carrier DB.
8. Transfer to Carrier Request: This new operation allows the service node to transfer the call to a carrier. This request will release the service node from the call. Switch based trigger will not be in effect.
9. Transfer to Control List Request: Three optional data blocks are added to this operation: Billing Acceptance DB, Operator Services System Action DB, and Text DB.

10. Trigger Event Inform: A treatment may be applied to either the A party or B party port when a call is transferred to a service node. An optional treatment data block is added to the Trigger Event Inform message. An optional OLNS Information DB is added.

1.4.2.2 Data Blocks

1. Abort Call DB: This data block is replaced with the new End Call Handling DB. The new data block has the same information as the Abort Call DB. This data block has been removed from this document.
2. Alternate Billing Directory Number DB: This data block is changed. The Billing Number Input Method field is removed. It is now included in the new Billing Number Input DB. Also, the Directory Number Type field has been expanded. A comment is added to indicate if the service node sends this data block with a Directory Number Digit Count of 0, the switch will nil out the alternate billing number stored in the switch and set the class charge to unspecified. With the removal of the Billing Number Input Method field, a padding field was also removed. This changed the data block size from 20 bytes to 18 bytes.
3. ASCII Card Number DB: A comment is added. If the service node sends this data block with an ASCII Card Number Length of 0, the switch will nil out the card number stored in the switch and set the class charge to unspecified.
4. BCD Card Number DB: A comment is added. If the service node sends this data block with a BCD Card Number Count of 0, the switch will nil out the card number stored in the switch and set the class charge to unspecified.
5. BNS Query Result DB: This data block is changed. The old PIN Status field is replaced with the new CCV Query Response field.
6. Billing Acceptance DB: This new data block indicates the method of billing acceptance.
7. Billing Number Input DB: This new data block indicates the billing number input method.
8. CCV Query Result DB: This data block is changed. The old PIN Status field is replaced with the new CCV Query Response field.
9. Carrier DB: This existing data block is changed. Three new fields are added to the data block. These new fields provide additional information about the carrier (Service Type Provided by the Switch, Carrier Number Status, Carrier Service Status).
10. Carrier Number DB: This new data block has the carrier number that the service node wants to assign to the call.
11. Call Information DB: Added new Service Type field, which indicates the base service applied to the call.
12. End Call Handling DB: This new data block indicates why the service node is ending the call.

13. LATA Information DB: This new data block specifies LATA information for the call. The switch sends this data block at the beginning of a call and whenever the B party directory number is updated.
14. OLNS Information DB: This new data block contains calling number information which was returned from an OLNS query.
15. Operator Services System Action DB: This is a new data block which indicates what action is taken by the Operator Services System used for billing purposes.

1.4.2.3 Fields

1. Billing Acceptance Response Method: This is a new field which specifies the method by which billing acceptance was obtained from the billed party.
2. Billing Number Equipment: This field has been removed. It is replaced with the new BNS Query Response field. The new field is expanded to include additional BNS query values.
3. BNS Query Response: This new field replaces the Billing Number Equipment field. It contains values returned from a BNS query.
4. CCV Query Response: This new field replaces the PIN Status field. It specifies the result received from a CCV billing validation query.
5. Call Handling: New values are added to indicate if the call is being billed for ending a call due to normal call handling or if billing is cancelled.
6. Call Origination Type: New values are added which are supported for preprocessing calls.
7. Carrier Number Status: This new field specifies if the carrier number is valid for the call.
8. Carrier Service Status: This new field specifies if the operator action is valid with regard to the type of service provided for a carrier.
9. Connection Status: This field is expanded to indicate if the port is connected to a treatment. This field is set to the new value if a treatment is applied to either the A party or B party port.
10. Directory Number Type: New values are added to this field to indicate if a directory number is domestic local, domestic non-local, outgoing trunk, or a service number.
11. LATA Status: This new field indicates if the call is an interlata call or an intralata call.
12. LATA Restrictions: This new field specifies if there are any restrictions on the A party and equal access.
13. OLNS fields: New OLNS fields sent in the OLNS Information DB.
14. Operator Services System Action: This is a new field which specifies the action taken by the Operator Services System (OSS) for AMA recording purposes.

- 15. PIN Status: This field has been removed. It is replaced with the new CCV Query Response field.
- 16. Service Type: This new field indicates the base service applied to the call (i.e. TA or DA).
- 17. Service Type Provided by Switch: This new field specifies the type of service the switch provides on behalf of the carrier.
- 18. Validation Result: This field is expanded to include a new value. This value is returned to the service node if the billing number is disallowed due to no mutual honoring agreement between the Local Exchange Carrier and InterLATA carrier.

1.4.2.4 Error IDs

- 1. #071E, Operation not valid for DA
- 2. #071F, Routing to carrier failed
- 3. #0722, Carrier served by switch
- 4. #072A, Carrier not served by switch
- 5. #072B, Carrier not allowed for call
- 6. #073D, Carrier ID is restricted
- 7. #073E, Carrier not specified for call
- 8. #073F, Timing has started
- 9. #0740, Incoming trunk group not supported for transfer

1.4.3 Version 3.0 Revision 0 (07/97)

1.4.3.1 Class Headers

All new and changed class headers are included with the OSSAIN SOC option OSAN0002.

1. Node Class Header: A new Solicitor Number field is added to the class header.
2. Node Alarm Class Header: This is a new class header used for sending alarm information on a node basis.
3. Node Log Class Header: This is a new class header used for sending logs on a node basis.
4. Session Pool Class Header: A new Solicitor Number field is added to the class header.
5. Session Pool Alarm Class Header: This is a new class header used for sending alarm information on a session pool basis.
6. Session Pool Log Class Header: This is a new class header used for sending logs on a session pool basis.

1.4.3.2 Call Processing Operations

SOC OSAN0002

1. Call Details Success Response: The NBEC DB has been added.
2. Call Float Request: The Branding Indicator DB has been added to this operation.
3. Connect DN Request: The Connect DN Information DB is removed from the request since it was not needed.
4. Conversation Timing Request: This is a new operation that allows the service node to request that the conversation timing be started or stopped.
5. Directory Number Success Response: The NBEC DB has been added.
6. Session Begin Inform: Comments have been updated to specify how an active node denies a Session Begin and how a passive node denies a Session Begin. Comments have been updated to specify how another node connected in a simultaneous interaction is indicated to the service node receiving the Session Begin Inform. The Time DB and NBEC DB have been added.
7. Session Initiation Success Response: The Time DB, Carrier DB, and NBEC DB are added to the success response.
8. Transfer to Control List Request: The Branding Indicator DB has been added to this operation.
9. Trigger Event Inform: The Time DB and NBEC DB are added.

10. Voice Connect Request: This operation is now a restricted operation. If voice connections are being made for a call involving a centralized service node and an OSAC Remote, messaging will occur between the OSAC Remote and OSAC Host. The service node should not send another request to the Remote until the voice connections are complete.

SOC OSAN0003

1. Accept Control Inform: This new operation is used to indicate to the passive service node that it now has control of the call.
2. Billing Number Success Response: The LNP Information DB is an optional data block in the response.
3. Call Details Success Response: The LNP Information DB, RLT Eligible DB, SPID DB, and Preprocessing Information DB are added in the response.
4. Call Float Request: The Charge Status DB and RLT Request DB have been added to this operation.
5. Call Resume Request: This is a new operation that is used when the session type is OSSAIN Preprocessing. This operation releases the service node and the call resumes processing in the switch.
6. Class Charge Success Response: The LNP Information DB is an optional data block in the response.
7. Connect DN Request: The LNP Information DB and the RLT Eligible DB are optional data blocks in the success response.
8. Connection Status Inform: The new Call Timing data block has been added to this operation.
9. Directory Number Success Response: The LNP Information DB and SPID DB are optional data blocks in the response.
10. End Call Request: The new Call Timing DB is added to the End Call Success Response.
11. Generate AMA Record Request: The Charge Status DB has been added to this operation. Regardless of the value of the Charge Status DB, the charge status for the next call segment following a Gen AMA is automatically marked "Billable" by the switch.
12. LNP Request: This new operation is used to request the LNP information for the specified party. Refer to the Functionality Overview section on LNP for more information.
13. Node Release Inform: This new operation is used to inform a service node involved in a simultaneous interaction that it has been released.
14. Pass Control Request: This new operation is used by an active service node to pass control of the call to passive function provider.

15. Pass Through Inform: This new operation is used during a simultaneous interaction to pass information between function providers.
16. Passive Function Provider Request: This new operation is used by the active service node to request a passive function provider be connected to the call for a simultaneous interaction. Refer to the Functionality Overview section on Simultaneous Interactions for more information.
17. Passive Node Status Inform: This new operation is used to inform the active service node about changes in the status of the passive function provider during a simultaneous interaction.
18. Release Node Request: This new operation is used to request the release of a function provider during a simultaneous interaction.
19. Route To Treatment Request: The Charge Status DB has been added to this operation.
20. Service Provider ID Assignment Request: This new operation is used to assign a SPID for a specified party and type of service provider.
21. Session Begin Inform: The new Call Timing data block, the LNP Information DB, SPID DB, RLT Eligible DB, and Preprocessing Information DB are added.
22. Session Initiation Request: The SPID DB is added to the request operation as an optional data block. The LNP Information DB and SPID DB are added to the success response.
23. Transfer to Carrier Request: The Charge Status DB and RLT Request DB have been added to this operation.
24. Transfer to Control List Request: The Charge Status DB has been added to this operation.
25. Trigger Event Inform: The new Call Timing data block, SPID data block, and the LNP Information DB are added.

1.4.3.3 Non-Call Processing Operations

SOC OSAN0002

1. Node Alarm Inform: This is a new operation that allows the service node to generate a switch alarm for the service node.
2. Node Busy Request: This operation can now be sent by the service node.
3. Node Log Inform: This is a new operation that allows the service node to generate a switch log for the service node.
4. Session Pool Alarm Inform: This is a new operation that allows the service node to generate a switch alarm for a session pool.
5. Session Pool Busy Request: This operation can now be sent by the service node.
6. Session Pool Drain Inform: This is a new operation that informs the service node when a session pool has been drained at the switch.

7. Session Pool Log Inform: This is a new operation that allows the service node to generate a switch log for a session pool.
8. Session Pool RTS Success Response: The service node can change the number of available sessions for the session pool if the default should not be used.
9. Session Pool Throttle Request: This is a new operation that allows the service node to change the number of available sessions for a session pool.

SOC OSAN0003

1. Session Pool State Inform: This is a new operation that informs the node about the state of a session pool at the OSAC Remote switch.

1.4.3.4 Data Blocks

SOC OSAN0002

1. Alarm Information DB: This new data block contains the information on an alarm.
2. Branding Indicator DB: This new data block indicates the front-end branding status of the call.
3. Call Information DB: Three new fields are added to this data block (Session Type, Conversation Timing Status, Branding Status).
4. Connect DN Information DB: This data block is no longer used and is removed from the document.
5. Conversation Timing Indicator DB: This new data block specifies if timing should be started or stopped for the call.
6. NBEC DB: This new data block specifies the NBEC associated with the call.
7. Node Identification DB: The Service Node Mode field is added to this data block, which specifies the current status of the service node identified in this data block.
8. OLNS Information DB: This data block has been updated to return the account owner and billing service provider returned from the OLNS LIDB.
9. Preprocessing Information DB: This new data block contains on a preprocessing call.
10. Report Text DB: This new data block contains information for a log report.
11. Throttle DB: This new data block specifies the number of available sessions for a session pool.
12. Trigger Profile Index DB: Two new fields are added to this data block (Notification Time in seconds, Notification Time Type).

SOC OSAN0003

1. Call Queue Information DB: This new data block contains information on the status of a call queue in the switch for a simultaneous interaction.

2. Call Timing DB: This new data block indicates the answer time and the conversation time for the call.
3. LNP Information DB: This new data block identifies the LRN and portability status of a directory number. The LRN for the CLI is also returned if applicable.
4. Maintenance State DB: This new data block specifies the state of the node or session pool specified in the message.
5. Operator System DB: This new data block indicates if the call should bypass any TOPS automated system and go directly to an operator from the service node.
6. Pass Through DB: This new data block is used to pass information between two function providers during a simultaneous interaction.
7. Passive Node Status DB: This new data block informs the active service node about the status of a passive function provider during a simultaneous interaction.
8. RLT Eligible DB: This new data block informs the service node if the call is eligible for RLT. Refer to the Functionality Overview section on RLT for more information.
9. RLT Request DB: This new data block is used by the service node to indicate if RLT should be attempted for the call.
10. Release Node DB: This new data block indicates which function provider is to be released during simultaneous interaction.
11. SPID DB: This new data block contains SPID information for the specified party and type of service provider.

1.4.3.5 Fields

1. The format for BCD fields has changed. The nibbles are swapped so that the digits are sent left to right (first digit in nibble 1, second digit in nibble 0). The following existing fields are affected:
 - BCD Card Number
 - Calling Line Identifier Number
 - Carrier ID Code
 - Directory Number
 - OLNS InterLATA Carrier Code
 - OLNS International Carrier Code
 - OLNS IntraLATA Carrier Code
 - RAO Code
2. ADACC Transition Indicator: This new field indicates if a DA call has transitioned to a toll call for ADACC.
3. Alarm Severity: This new field specifies if an alarm is critical, major, or minor.
4. Alarm Status: This new field specifies if an alarm is on or off.
5. Available Sessions: This new field specifies the absolute number of available sessions for a session pool.

6. Branding Status: This new field specifies if the call has received front-end branding.
7. Call In Queue Take Down: Previously, for facilities that require atleast one party to be present (e.g ISUP trunks), the call would be taken down if all parties went on hook, regardless of the value for this field. Now, the call will remain in queue based on the value of this field.
8. Calls In Queue: This new field specifies the number of calls currently in queue for a particular call queue.
9. Call Queue: This new field specifies the call queue number used for a function provider request.
10. Conversation Timing Status: This new field specifies the conversation time recording for the call be started or stopped.
11. Elapsed Minutes: This new field specifies the elapsed conversation time in minutes.
12. Function Provider Request Status: This new field specifies the status of a request by a service node to connect another function provider to the call.
13. LRN: This new field specifies the location routing number for a directory number.
14. LRN Digit Count: This new field specifies the number of digits in the LRN.
15. Maintenance State: The new field specifies the state of the node or session pool.
16. NBEC Code: This new field specifies the NBEC code associated with the call.
17. NBEC Code Digit Count: This new field specifies the number of digits in the NBEC Code field.
18. Notification Time: This field has been removed. It has been replaced with the new Elapsed Minutes field in the Trigger Profile Index DB.
19. Notification Time Type: This new field identifies the basis for starting a trigger notification time.
20. Operator System Selector: This new field indicates if the call is eligible for a TOPS automated system or if it is only allowed to be handled by an operator.
21. Operator System Type: This new field identifies the operator system used for a TOPS call.
22. Pass Through Data: This new field is used to pass data between the two function providers connected to the call.
23. Pass Through Data Length: This new field specifies the length of the pass through data.
24. Passive Node Status: This new field indicates the status of a passive function provider during a simultaneous interaction.
25. Portability Status: This new field specifies the portability status of a directory number.

26. Predicted Wait Time: This new field specifies the predicted wait time for a given function provider request.
27. Queuing System: This new field specifies which queuing system was used in the switch for a given function provider request.
28. RLT Eligibility: This new field indicates if a call is eligible for RLT. Refer to the Functionality Overview section on RLT for more information.
29. Release Node Indicator: This new field specifies which function provider to release.
30. Report Text: This new field specifies the printable text for a log report.
31. Report Text Length: This new field specifies the number of characters in the Report Text.
32. Requested RLT Action: This new field specifies if the call should attempt RLT.
33. Serving Function Provider Status: This new field specifies whether or not there are any function providers defined in the switch to serve the given call queue.
34. Service Node Mode: This new field specifies the connection mode of the service node.
35. Service Provider ID: This new field specifies the SPID for the service provider party and type of service provider.
36. Service Provider ID Length: This new field specifies the length of the SPID.
37. Service Provider Party Identifier: This new field specifies the associated party for the SPID ID.
38. Session Type: This new field indicates the type of OSSAIN processing the call should receive by the service node.
39. Solicitor Number: This new field identifies the maintenance task on the switch or service node.
40. Tenths Seconds: This new field specifies the time in tenths of seconds.
41. Type of Service Provider: This new field specifies the type of SPID.

SOC OSAN0003

These new fields are added to existing data blocks and are controlled by SOC option OSAN0003.

1. OLNS Account Owner: This new field specifies the account owner returned from an OLNS LIDB. A length field is associated with this field.
2. OLNS Billing Service Provider: This new field specifies the billing service provider returned from an OLNS LIDB. A length field is associated with this field.

1.4.3.6 Error IDs

1. #10A, Operation invalid for session pool type
2. #010B, Invalid unsolicited operation

3. #010C, Invalid state
4. #010D, Throttle request is greater than MAXSESN
5. #010E, Maintenance in progress
6. #010F, Drain command entered
7. #604, Operation not allowed from a passive node.
8. #605, Voice link request in progress for passive node.
9. #606, Passive function provider request in progress.
10. #607, Passive operator connection in progress.
11. #608, Operation is restricted for preprocessing session.
12. #609, Operation is restricted for standard session.
13. #60A, OSAC voice connection in progress.
14. #60B, SOC option not turned on.
15. #741, A passive function provider must be connected or the call must be queued for a passive node
16. #742, Unknown or invalid function
17. #743, Queue overflow
18. #744, Queue deflection
19. #745, Other failure in routing and queuing
20. #746, Passive function provider already connected or call is already queued for a passive function provider.
21. #747, Conversation time exceeds notification time
22. #748, A passive SN must be connected
23. #749, LNP information not available
24. #74A, Host-Remote voice channel not available.
25. #74B, No response from OSAC Host.
26. #74C, No host-remote session available.
27. #74D, Session pool datafill mismatch.
28. #74E, Function datafill mismatch.
29. #074F, Invalid service provider id.
30. #750, Conversation timing cannot be performed for 'No AMA' calls.

1.4.3.7 Functionality Overview

1. A section has been added describing DA processing with OSSAIN.
2. A section has been added describing LNP processing with OSSAIN.

3. A section has been added describing OSSAIN Centralization.
4. A section has been added describing OSSAIN Preprocessing.
5. A section has been added describing RLT processing with OSSAIN.
6. A section has been added describing OSSAIN Simultaneous Interactions.

1.4.4 Version 4.0 Revision 0 (3/98)

1.4.4.1 Call Processing Operations

SOC OSAN0002

1. Call Merge Inform: This new operation is used to inform the service node when an A party has been merged into the call.
2. Call Merge Request: This new operation is used to move the A party from one call to another call without an A party.
1. Class Charge Request: The precondition for requiring the forward party DN has changed.
2. Connect DN Request: Added error id #72A.
3. Conversation Timing Request: Added two error ids to the error response: #73F and #750.
4. Directory Number Request: The Originating Trunk Information DB is added as an optional data block. The new Directory Number Information DB is added to the success response.
5. Notification Timer Inform: This new operation informs the service node that the notification timer has expired.
6. Notification Timer Request: This new operation is used to start/stop/reset the persistent-style notification timer.
7. Session Initiation Request: The Originating Trunk Information DB is added as an optional data block.

SOC OSAN0003

1. Passive Function Provider Success Response: An optional choice data block is added. If a passive function provider is immediately obtained, the Node Identification DB or the Operator Identification DB is sent to the active service node.
2. Passive Node Status Inform: An optional choice data block is added. If the call was queued for a passive function provider and is now obtained and connected, the Node Identification DB or the Operator Identification DB is sent to the active service node.

SOC OSAN0004

1. Accept Control Inform: The optionality for the Class Charge DB is changed. The RLT Eligible DB is removed. Terminating Trunk Information DB and ISUP Calling Party Number DB are added.
2. Call Details Success Response: New conferencing data blocks have been added: Conference ID DB, Conference Party Connection DB, and Conference SN Voice Connection DB. The optionality for the Class Charge DB is changed. The RLT Eligible DB is removed. Terminating Trunk Information DB and ISUP Calling Party Number DB are added.

3. Call Float Request: This operation is changed to specify if a party should be connected to a conference upon float.
4. Conference Add Request: This new operation is used to move a voice entity from a call to a conference.
5. Conference Create Request: This new operation is used to create a conference.
6. Conference Details Request: This new operation is used to obtain information about all voice entities associated with a conference.
7. Conference Release Request: This new operation is used to end an association between a call and a conference.
8. Conference Remove Request: This new operation is used to move a voice entity from a conference to a call.
9. Connect DN Request: This operation is changed to specify if the party should be connected to a conference. The RLT Eligible DB is removed from the success response. Terminating Trunk Information DB is added to the success response.
10. Conversation Timing Error Response. New error id #075F is added.
11. ISUP Calling Party Number Update Request: This new operation allows the service node to update the Calling Party Number parameter in the outgoing ISUP IAM message.
12. Service Change Request: This new operation allows the service node to change the service type of the call from TA to DA or from DA to TA.
13. Session Begin Inform: New conferencing data blocks have been added: Conference ID DB, Conference Party Connection DB, and Conference SN Voice Connection DB. The optionality for the Class Charge DB is changed. The RLT Eligible DB is removed. Terminating Trunk Information DB and ISUP Calling Party Number DB are added.
14. Session Recall Request: This new operation is used initiate a session with a floated call.
15. Trigger Event Inform: New conferencing data blocks have been added: Conference ID DB, Conference Party Connection DB, and Conference SN Voice Connection DB. Terminating Trunk Information DB and ISUP Calling Party Number DB are added. The optionality for the Class Charge DB is changed. The RLT Eligible DB is removed.
16. Voice Connect Request: This operation is changed to specify if the SN voice link should be connected to a conference. It also allows the service node to specify if the voice connection should be to a broadcast voice links. The new Broadcast Count DB is sent with the Voice Connect Success Response.
17. Voice Release Inform: This new operation is used to inform the service node when the switch has removed a broadcast voice link from a call.

1.4.4.2 Data Blocks

SOC OSAN0002

1. Call Information DB: This existing data block is changed to provide information on the persistent style notification timer.
2. Carrier DB: The Access Indicator field is replaced with the new field Carrier Assignment Source Indicator.
3. Carrier Number DB: The Carrier Assignment Source Indicator field is added to the data block.
4. Directory Number Information DB: This new data block is used to provide the service node additional information on the directory number that was just updated.
5. Notification Timer DB: This new data block is used to start/stop/reset a persistent style notification timer.
6. Trigger Profile Index DB: This existing data block is changed to specify the timer mode for the notification timer.

SOC OSAN0004

1. Broadcast DB: This new data block provides broadcast information for the voice connection.
2. Broadcast Count DB: This new data block indicates the number of current connections on a broadcast voice link.
3. Callid DB: This new data block identifies a call at the switch.
4. Call Information DB: A new field (SPID Processing) is added to this data block.
5. Class Charge DB: A new field (class charge selector) is added to this data block.
6. Conference ID DB: This new data block identifies a conference.
7. Conference Party Connection DB: This new data block identifies a particular conference party connection for a party in a call.
8. Conference Party Connection Information DB: This new data block provides information about a particular conference party
9. Conference SN Voice Connection DB: This new data block specifies whether or not the conference operation applies to the SN voice link in a call.
10. Conference SN Voice Connection Information DB: This new data block provides information about the SN voice link connected to a conference.
11. ISUP Calling Party Number DB: This new data block specifies the number used in the Calling Party Number parameter in the outgoing ISUP IAM.
12. RLT Eligible DB: This data block is removed from the protocol.
13. Requested Service DB: This new data block identifies the new service type requested by the service node.

14. Session Information DB: This new data block identifies a particular service node session.
15. Terminating Trunk Information DB: This new data block identifies the trunk the call connected to for the B party or alternate DN.

1.4.4.3 Fields

SOC OSAN0002

1. Access Indicator: This field is removed from the protocol. It provides redundant information.
2. Carrier Assignment Source Indicator: This new field identifies the source of the carrier assignment for a call.
3. Notification Timer Action: This new field specifies what should be done with the notification timer.
4. Notification Timer Mode: This new field specifies whether the notification timer is a float style or a persistent style timer.
5. Notification Timer Status: This new field specifies whether or not the notification timer is currently running.

SOC OSAN0004

1. Broadcast Cycle Count: This new field specifies if the switch should count cycles.
2. Class Charge Selector: This new field indicates if the class charge information is current or previous.
3. Conference ID: This new field identifies a particular conference.
4. Current Broadcast Connections: This new field specifies the number of connections currently on a broadcast voice link.
5. RLT Eligibility: This field is no longer needed and is removed.
6. SN Voice Conferencing Status: This new field specifies whether or not the SN voice link is connected to a conference.
7. SPID Processing: This new field indicates if SPID processing is enabled for a DN.
8. Session Origination Type: A new value of Reorigination is added to this field.

1.4.4.4 Error IDs

1. #60C, Invalid operation for calls involved in a conference. This error id is added to the Transfer to Control List Error Response, Transfer to Carrier Error Response, Route to Treatment Error Response, and Passive Function Provider Error Response.
2. #60D, Invalid operation for calls involved in a simultaneous interaction.
3. #721, Carrier not served in this office. This error id is removed. The error id is removed from the Carrier Assignment Request and is not replaced. The error id is replaced with #72A in the Session Initiation Request.

4. #752, Unknown or invalid conference id. This error id is added to the Call Float Error Response, Connect DN Error Response, and Voice Connect Error Response.
5. #753, Voice entity already connected to a conference. This error id is added to the Call Float Error Response.
6. #754, Voice entity not connected to a conference.
7. #755, Conference already exists.
8. #756, Too many voice entities in conference. This error id is added to the Call Float Error Response, Connect DN Error Response, and Voice Connect Error Response.
9. #757, The B party of the specified call is not connected and offhook.
10. #758, The specified call already has an A party connection.
11. #759, Unknown or invalid callid.
12. #75A, Notification timer not running.
13. #75B, Session is not idle.
14. #75C, Session is not used for SN originations.
15. #75D, Target call already has a session with an SN.
16. #75E, Target B party is connected to a different conference than the A party conference.
17. #75F, Conversation timing not started. This error is added to the Conversation Timing Error Response.
18. #0760, Previous class charge information cannot be set by the service node. This error is added to the Class Charge Error Response.
19. #761, SPID processing not enabled. This error is added to the Service Provider ID Assignment Error Response and the Session Initiation Error Response.
20. #762, Attempt to exceed max connections for broadcast voice link. This error is added to the Voice Connect Error Response. The new Broadcast Count DB (#155) is included with the error id.
21. #763, Voice link cutthru method invalid for cycle counting. This error is added to the Voice Connect Error Response.
22. #764, Invalid broadcast configuration. This error is added to the Voice Connect Error Response.
23. #766, Not a broadcast voice link. This error is added to the Voice Connect Error Response.
24. #767, OSAC Host SOC not on. This error is added to the Voice Connect Error Response.
25. #768, Voice link for broadcast only. This error is added to the Voice Connect Error Response.
26. #769, New service is the same as the old service.

27. #76A, Invalid trunk group and member.

1.4.4.5 Functionality Overview

1. The section on voice connections is updated.
2. The section on RLT is updated.
3. A section on equal access is added.

1.4.5 Version 5.0 Revision 0 (8/98)

1.4.5.1 Class Header

SOC OSAN0005

1. MIS: This new class header is used to send MIS data to the MIS node.

1.4.5.2 Call Processing Operations

SOC OSAN0004

1. Voice Release Inform: Remove 'broadcast' from Description and Comment #1 is changed. This message can be applicable to both non-broadcast and broadcast voice links.

SOC OSAN0005

1. Accept Control Inform: The new Country Direct Information DB is added as an optional data block.
2. Call Details Success Response: The new Country Direct Information DB is added as an optional data block.
3. Session Begin Inform: The new Country Direct Information DB is added as an optional data block.
4. Session Recall Success Response: The new Country Direct Information DB is added as an optional data block.
5. Trigger Event Inform: The new Country Direct Information DB is added as an optional data block.

1.4.5.3 Non-Call Processing Operations

1. MIS OSSAIN Inform: This new operation is used to send OSSAIN MIS data.

1.4.5.4 Data Blocks

SOC OSAN0005

1. Country Direct Information DB: This new data block contains country direct information for the call.
2. Information ID DB: This new data block provides initial MIS information.
3. OSSAIN Call Queue Event DB: This new data block provides MIS information on a queue event.
4. OSSAIN Call Session Event DB: This new data block provides MIS information on a call related event that affects a session.
5. OSSAIN Non-Call Session Event DB: This new data block provides MIS information on an event that affects a session.

1.4.5.5 Fields

SOC OSAN0005

1. Access Indicator: This new field is added to indicate if the call is being completed by the operating company or by a carrier.
2. Application ID: This new field identifies the QMSMIS application.
3. Call Type for Queuing: This new field identifies the final CT4Q used for the call.
4. Call Origination Type: This existing field is expanded to add a new value of country direct.
5. Country Direct Access Code Count: This new field indicates the number of digits in the Country Direct Access Code Number field.
6. Country Direct Access Code Number: This new field has the access code used for the country direct call.
7. Country Direct Carrier Code Count: This new field indicates the number of digits in the Country Direct Carrier Code Number field.
8. Country Direct Carrier Code Number: This new field has the carrier code used for the country direct call.
9. Country Direct Country Code Count: This new field indicates the number of digits in the Country Direct Country Code Number field.
10. Country Direct Country Code Number: This new field has the country code used for the country direct call.
11. MIS Sequence Number ID: This new field provides a sequence number for session and call queue event DBs used in the MIS data stream.
12. Network ID: This new field identifies the MIS network.
13. Queue Peg Indicators: This new field specifies what MIS statistics should be pegged.
14. Queuing Event: This new field specifies how the call was handled when queued.
15. Reset Time Flag: This new field indicates if the time associated with a session id should be reset for MIS data.
16. Restart Severity: This new field identifies the type of restart that occurred, if any.
17. Service Peg Indicators: This new field indicates what service measurements to peg.
18. Session State: This new field indicates the current state of the specified service node session.
19. Special Number Format: This new field specifies the format of the billing number.
20. Special Number Type: This new field specifies the type of billing number.
21. Special Number Validity: This new field specifies if the billing number is valid.
22. Tenths of Seconds Timestamp: This new field is the time specified in tenths of seconds.
23. Time in Queue: This new field specifies how long a call was queued.
24. Trunk Member ID: This existing field has its range extended to 10000.

1.4.5.6 Error IDs

1. The definition for Error ID #76B has changed. It is now used for Invalid for Country Direct Call. This error is added to the Service Change Request Error Response. The previous definition for #76B was No Origination Trunk Information. It was not used in any error responses prior to this release.
2. Remove unused Error ID #728, Insufficient Parties in Call.

1.4.5.7 Functionality Overview

1. The section on Equal Access is updated.
2. New section is added on QMS MIS.
3. The section on Voice Links is updated.

1.4.6 Version 6.0 Revision 0 (8/98)

1.4.6.1 Call Processing Operations

SOC OSAN0003

1. Pass Through Inform: this operation is now only sent by the switch and is only received by either an active or passive SN in a simultaneous interaction.
2. Pass Through Request: this new operation is used by the active or passive SN (in place of the Pass Through Inform) to **send** pass through data to the switch for a simultaneous interaction.

SOC OSAN0006

1. Accept Control Inform: the new SN Voice Link DB, Routing Number DB, Routing Index DB, Estimate of Charges General Information DB, Estimate of Charges Duration Information DB and Estimate of Charges No Duration Information DB are optional.
2. Billing Number Request: the new Routing Number DB and Routing Index DB are optional.
3. Billing Number Error Response: the new Invalid Routing Index, North American-only functionality and Connection ID - Party mismatch error IDs are added.
4. Call Details Success Resonse: the new SN Voice Link DB, Routing Number DB, Routing Index DB, Estimate of Charges General Information DB, Estimate of Charges Duration Information DB and Estimate of Charges No Duration Information DB are optional.
5. Call Float Request: the new Routing Number DB and Routing Index DB are optional.
6. Call Float Error Response: the Invalid Number Format and Translations failed error IDs are added. Also, the new Invalid Routing Index, new North American-only functionality, new Connection ID - Party mismatch and new Class Charge Missing But Required error IDs are added.
7. Conference Details Success Response: this operation may now have from 0 to 4 Conference Party Connection Information DBs.
8. Connect DN Request: the new Routing Number DB and Routing Index DB are optional.
9. Connect DN Error Response: the new Invalid Routing Index, new North American-only Functionality, new Connection ID - Party Mismatch, existing Invalid Number Format and new Class Charge Missing But Required error IDs are added.
10. Connect To Treatment Request: this new operation allows the SN to connect a party to treatment while still keeping control of the call.
11. Directory Number Request: the new Routing Number DB and Routing Index DB are optional.

12. Directory Number Error Response: the new Invalid Routing Index, North American-only functionality and Connection ID - Party mismatch error IDs are added.
13. Estimate of Charges Request: this new operation allows the SN to request the switch calculate charges for a call.
14. Passive Function Provider Error Response: added the Voice channel seize failure and Invalid broadcast configuration error ids.
15. Session Begin Inform: the new SN Voice Link DB, Routing Number DB, Routing Index DB, Estimate of Charges General Information DB, Estimate of Charges Duration Information DB and Estimate of Charges No Duration Information DB are optional.
16. Session Initiation Request: the new Routing Number DB and Routing Index DB are optional.
17. Session Initiation Success Response: the new SN Voice Link DB and the existing Directory Number Information DB are optional.
18. Session Initiation Error Response: the new Invalid Routing Index and North American-only functionality error IDs are added.
19. Session Recall Success Response: the new SN Voice Link DB, Routing Number DB, Routing Index DB, Estimate of Charges General Information DB, Estimate of Charges Duration Information DB and Estimate of Charges No Duration Information DB are optional.
20. Transfer to Carrier Request: the new Routing Index DB is optional.
21. Transfer to Carrier Error Response: the new Invalid Routing Index and Connection ID - Party mismatch error IDs are added.
22. Trigger Event Inform: the new Routing Number DB, Routing Index DB, Estimate of Charges General Information DB, Estimate of Charges Duration Information DB and Estimate of Charges No Duration Information DB are optional.
23. Voice Connect Request: the Voice Channel DB is now optional instead of mandatory.
24. Voice Connect Success Response: the new SN Voice Link DB is optional.
25. Voice Connect Error Response: the new SN voice link datafill missing and OSAC release too low error IDs are added. The new SN Voice Link DB is now optional.

1.4.6.2 Data Blocks

SOC OSAN0002

1. Directory Number DB: added new Presentation Indicator field.
2. Directory Number Update DB: added new Presentation Indicator field.
3. ISUP Calling Party Number DB: added new Presentation Indicator field.

SOC OSAN0003

1. Pass Through DB: the size of this DB is changed to accomodate the increased size of the Pass Through Data field.

SOC OSAN0006

1. Directory Number DB: added new Unenforced Restrictions field.
2. Directory Number Information DB: added new Unenforced Restrictions field.
3. Estimate of Charges Alternate Date DB: this new DB contains an alternate date for the switch to use for an estimate of charges calculation.
4. Estimate of Charges Alternate Time DB: this new DB contains an alternate time for the switch to use for an estimate of charges calculation.
5. Estimate of Charges Duration DB: this new DB specifies a duration for the switch to use for an estimate of charges calculation.
6. Estimate of Charges Duration Information DB: this new DB contains the calculated results of an estimated charge request when a duration was specified by the SN.
7. Estimate of Charges General Information DB: this new DB contains date and time specific results of an estimated charge calculation.
8. Estimate of Charges No Duration Information DB: this new DB contains the calculated results of an estimated charge request when no duration was specified by the SN.
9. Routing Index DB: this new DB contains an alternate routing index to be used when routing to the specified A, B or Alt party.
10. Routing Number DB: this new DB contains an alternate routing number to be used in place of the DN when translating a call to the specified A , B or Alt party.
11. SN Voice Link DB: This new DB contains voice link information such as trunk group, trunk member and voice channel ID.
12. Trigger DTMF DB: remove comment #3 as DTMF digits can be reported from either the A or the B party.

1.4.6.3 Fields

SOC OSAN0002

1. Presentation Indicator: this new field specifies whether the associated directory number has caller ID blocking.

SOC OSAN0003

1. Pass Through Data: the length of this field is increased to 100 bytes.
2. Pass Through Data Length: the range of this field is now 0 to 100 bytes.
3. Calls in Queue: the range of this field is now 0 to 32767.

SOC OSAN0006

1. Estimated Charges: this new field specifies the charges for a call.

2. Estimate Duration: this new field specifies the call duration to be used when estimating charges for a call.
3. Logical Voice Channel ID: this new field specifies the logical voice channel number for the voice connection between the switch and the service node.
4. Routing Index: this new field specifies the alternate routing index to be used when routing to a party.
5. SN Voice Link Status: this new field specifies whether a voice link has been connected or released.
6. Unenforced Restrictions: this new field specifies whether the associated party has any unenforced restrictions assigned.

1.4.6.4 Error IDs

1. #76C, Invalid routing index, is added to the Billing Number, Connect DN, Call Float, Directory Number, Transfer to Carrier and Session Initiation error responses.
2. #76D, North American-only functionality, is added to the Billing Number, Connect DN, Call Float, Directory Number, and Session Initiation error responses.
3. #76E, Connection ID - party mismatch, is added to the Billing Number, Connect DN, Call Float, Directory Number, Transfer to Carrier and Session Initiation error responses.
4. #76F, SN voice link datafill missing, is added to the Voice Connect Error Response.
5. #770, Estimate of charges failure, is added to the Estimate of Charges Error Response.
6. #771, Invalid estimate duration for numbering plan, is added to the Estimate of Charges Error Response.
7. #772, Invalid estimate date, is added to the Estimate of Charges Error Response.
8. #773, OSAC release too low, is added to the Voice Connect Error Response.
9. #774, Class charge missing but required, is added to the Call Float and Connect DN error responses.

1.4.6.5 Functionality Overview

1. A new section on Estimated Charges is added.
2. The section on Voice Links has been updated to include information on switch-selected voice link functionality.

1.4.7 Version 7.0 Revision 0 (8/99)

1.4.7.1 Call Processing Operations

SOC OSAN0002

1. Call Float Request: the new Call Update DB is optional.
2. Carrier Assignment Success Response: the Billing Information DB and Class Charge DB are now optional.
3. Generate AMA Record Request: remove the 'B party cannot be connected' precondition. Gen AMA is now allowed with B party connected.
4. Service Provider ID Assignment Success Response: the Billing Information DB and Class Charge DB are now optional.
5. Transfer to Control List Request: the new Call Update DB is optional.

SOC OSAN0007

1. Billing Number Request: the new LNP Enhancements DB is optional.
2. Calling Number BNS Query Request: this new operation is used when the SN wants the switch to perform a BNS query on the calling number.
3. Connect DN Request: the new LNP Enhancements DB is optional.
4. Directory Number Request: the new LNP Enhancements DB is optional.
5. LRN Update Request: this new operation is used when the SN wants to update the Location Routing Number for a particular party.
6. Return Answer Message Request: this new operation is used when the SN wants answer to be sent back to the calling party.
7. Session Initiation Request: the new LNP Enhancements DB is optional.

1.4.7.2 Data Blocks

SOC OSAN0002

1. Call Information DB: new field Number of Operator to SN Handoffs is added.
2. Call Update DB: this new DB contains information regarding the number of operator-to-SN handoffs allowed.

SOC OSAN0007

1. LNP Enhancements DB: this new DB is used to tell the switch whether to perform an LNP query for the associated operation.
2. LRN Update DB: this new DB is used to update the Location Routing Number for a particular party.

1.4.7.3 Fields

SOC OSAN0002

1. Number of Operator to SN Handoffs: this new field indicates the number of operator-to-SN handoffs allowed for the call.

SOC OSAN0003

1. Predicted Wait Time: the range of this field is now 0 to 32767.

SOC OSAN0007

1. Call Origination Type: the range is extended to include the SLRN call origination type.
2. LNP Query: this new field tells whether the switch should perform an LNP query or not on a particular DN.

Chapter 2: OSSAIN Overview

2.1 General description of OSSAIN

Operator Services Systems Advanced Intelligent Network (OSSAIN) defines a generic interface between the DMS TOPS switch and OSSAIN service nodes. It allows service nodes to control switch functionality typically associated with operator services in order to provide automated services. OSSAIN avoids making assumptions on the types of services that may be developed using this interface.

The primary goal of OSSAIN is to offer an infrastructure for the development of new services independent of switch software development cycles for faster deployment to market. Service nodes are given superior call control capabilities similar to those of an operator.

The OSSAIN infrastructure is generic enough to implement a wide variety of automated services. OSSAIN will also assist in migrating existing TOPS automated systems to a common platform. These systems can be upgraded and enhanced without switch development.

A secondary goal of OSSAIN is to provide live operator support to traditional Advanced Intelligent Network (AIN) services. This goal will require future enhancements to OSSAIN and thus to OAP.

The Open Automated Protocol (OAP), which constitutes the communication protocol between the TOPS switch and service nodes, is one component of the OSSAIN product.

2.2 Responsibilities of the Switch

2.2.1 Call Processing Responsibilities

For an OSSAIN call, service nodes are given control of the call. The switch is responsible for presenting calls to service nodes for new calls or as a result of a transitions, handling requests from service nodes while a session is active, and informing service nodes of call characteristics.

The list of actions the switch performs includes:

- Selecting a function provider for OSSAIN calls coming into a switch
- Initiating an OSSAIN call at the request of a service node
- Presenting calls to service nodes
- Providing a centralized location for billing operations
- Connecting and releasing subscriber parties
- Connecting and releasing specific network resources (e.g. voice links)
- Detecting unsolicited events and informing service nodes about them (e.g. subscriber going on-hook)
- Storing context information for service nodes at their request. On transitions, passing this context data between service nodes or between a service node and a TOPS operator
- Returning a call to a specific service node or a TOPS operator at some point after the call has been floated and a trigger event has occurred. Floating a call is the process of a service node, a TOPS operator, or an existing TOPS automated system releasing from the call prior to establishing, or having established, a connection between the parties.

The service node has been released from the call (i.e. the session ends between the switch and the service node) and the call has been floated. A call floated trigger event occurs that requires the call to be sent to a service node (i.e. a new session is started between the switch and the service node for the call) or an operator. The switch will send the call context to the service node or operator.

- Coordinating the transitioning of calls between service nodes, between service nodes and TOPS operators, and between service nodes and existing TOPS automated systems

2.2.2 Non-Call Processing Responsibilities

Maintenance can be performed on a service node or on the session pools for a service node. The switch and service nodes share the maintenance responsibilities, and coordinate with each other when maintenance is required on the service node.

The list of capabilities provided by the switch follows:

- Manual maintenance (e.g. take a service node out of service, return a service node back to in service)

- Automated maintenance
- Audits to ensure communications exist with a service node

2.3 Responsibilities of Service Nodes

2.3.1 Call Processing Responsibilities

Service nodes are given control for OSSAIN calls. This is true regardless of whether the switch originated a session to a service node or if a service node initiated a session with the switch. Once a session is established with a service node, it can begin providing a function (e.g., branding or alternate billing) for the call.

The following are actions service nodes can request from the switch:

- Establish a session with the switch for a call
- Perform specific actions relating to the call such as:
 - Append AMA modules and generate an AMA record
 - Set the class charge for the call and validate the billing number
 - Alter the speech path
 - Set up and release voice connections
 - Enter a directory number for a subscriber party
 - Connect or release subscriber parties

See Chapter 7: “Call Processing Operations” for specific operation requests.

- Release itself from the call, using one of the following requests:
 - Call Resume Request (Preprocessing Session)
 - End the call
 - Float the call
 - Release Node Request (Simultaneous Interactions)
 - Route the call to treatment
 - Transfer the call to a carrier
 - Transfer the call to a control list

2.3.2 Non-Call Processing Responsibilities

Service nodes respond to the switch requests for maintenance activities. They are responsible for the following:

- Responding to switch originated maintenance requests regarding the service node (e.g. a return to service request for the service node)
- Responding to switch originated maintenance requests regarding a session pool on the service node (e.g. taking the session pool out of service)
- Responding to audits to ensure that communications exist with the switch

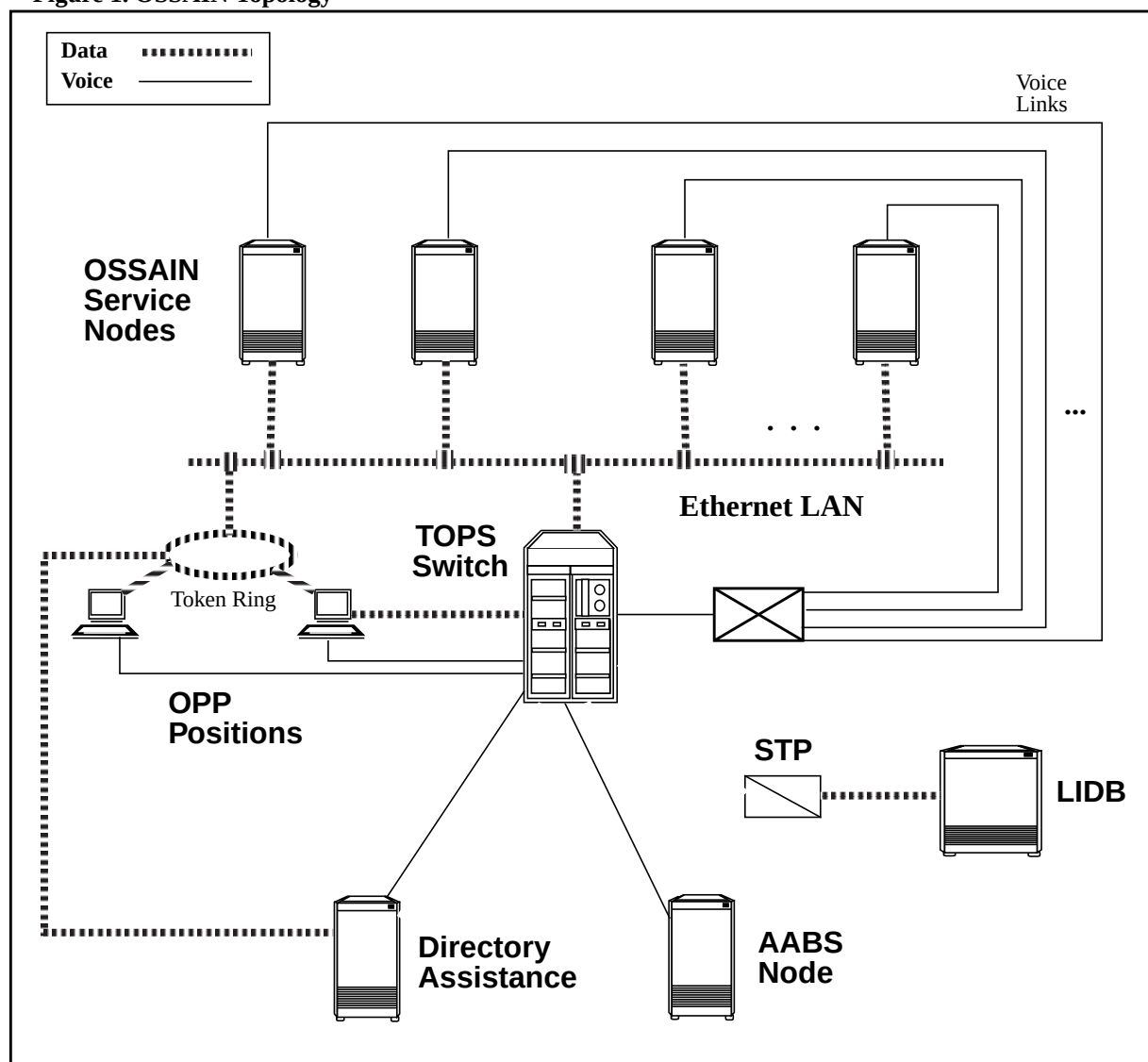
Service nodes are able to request the following from the switch:

- Generate logs at the switch on either a node basis or a session pool basis
- Generate an alarm at the switch on either a node basis or a session pool basis
- Change the number of sessions available for call processing.

2.4 System Configuration

This document does not attempt to fully describe the system configuration for OSSAIN. Refer to the *OSSAIN User's Guide* for detailed information. In the current release of OSSAIN, only data connectivity via the Local Area Network (LAN) is supported. Figure 1: "OSSAIN Topology" shows a high level view of an OSSAIN network connectivity via the LAN. Note that all call sessions start out as "data only." Voice connections between a service node and switch may be requested by the service node at any time during the call, or automatically selected by the switch when the session begins.

Figure 1. OSSAIN Topology



2.4.1 Transport

OSSAIN currently utilizes the Ethernet 10BaseT technology. The use of Ethernet provides a high bandwidth capability and connectivity to commercial computing environments. A network is typically comprised of a series of Ethernet LANs. LAN connectivity for the DMS switch is via the Ethernet Interface Unit (EIU).

The User Datagram Protocol (UDP) is used for most data messaging at the transport layer. UDP is a member of the Internet Protocol (IP) suite of protocols and provides a datagram oriented means for messaging between OSSAIN nodes (switch and service nodes).

OSSAIN nodes providing MIS functionality use the Transport Control Protocol (TCP) for MIS data messaging at the transport layer. TCP belongs to the IP suite of protocols and provides a reliable stream oriented data connection between OSSAIN nodes (switch and service nodes).

Each OSSAIN node in the network is known by its IP address, which uniquely identifies it from all other nodes. All messages sent through this network will be encapsulated within an IP message having an originating IP address and port number and a destination IP address and port number.

Refer to the *OSSAIN User's Guide* document for detailed information on the network configuration.

2.4.2 Communication with Service Nodes

The TOPS switch and a service node communicate with each other using a session for call processing. A session is a logical agent on the service node that is capable of controlling the call. Each session has a session identifier. The session ID is used for the duration of the interaction between the switch and service node for the current call.

For non-call processing, the switch and service node communicate at either the node level (for node level maintenance) or session pool level (for session pool level maintenance).

2.4.3 Voice Connections

Voice connections between a service node and switch may be requested by the service node or automatically selected by the switch when the session begins (the latter only available in OAP version 6.0). When the service node requests a voice link it may explicitly request a particular logical voice channel, or it may let the switch select the voice channel, depending on the format of the request.

Each service node that is selecting voice links is associated with its own set of voice channels. When a service node requests a voice connection for a particular session, it must specify a logical voice channel number. Logical voice channel numbers are mapped to actual voice circuits at the switch.

When the switch is selecting the voice link, the selection is from a group of trunks associated with a particular node *and* function. The switch selects an available member from this trunk group and informs the SN of its selection.

The DMS switch makes no assumptions about how voice circuits are terminated in the network. Voice circuits can be connected to the service node itself, to another service node, or to any other external system.

Refer to Section 13.13 Voice Connections for more information on voice connections.

Chapter 3: Protocol Description

3.1 Protocol Objectives

The goals of OSSAIN are:

- facilitate the deployment of new services independent of switch software development cycles by giving call control to service nodes
- offer operator backup for automated services
- facilitate the enhancements of current applications
- provide a switch software architecture which is both general and extensible

As a result, OAP has been designed to be an open and service-independent protocol. It is straightforward and deterministic to simplify development of OSSAIN services. Strategies are defined for version control, protocol upgrades, and compatibility across three older protocol versions. In addition, the OAP operation format has been developed using Abstract Syntax Notation (ASN) and Remote Operations Service Element (ROSE) concepts, which are discussed later in this document.

OAP is designed to be transport independent. Currently, OSSAIN is developed using the Ethernet network. However, this is not a limitation imposed by OAP.

3.2 High Level Description of OAP

OAP is an open, service-independent, switch-to-service node interface. It gives service nodes the ability to communicate with a DMS TOPS switch to control OSSAIN calls.

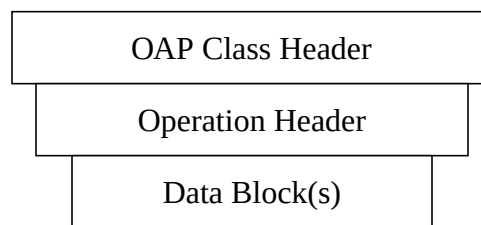
OAP provides a medium for information that pertains to OSSAIN which includes:

- Presenting calls to service nodes
- Accepting service node originated calls
- Providing centralized AMA operations (e.g. a service node can send an AMA module to the switch and have it appended to the AMA record for the call, service nodes can request the switch to generate the AMA record)
- Allowing service nodes to request specific billing operations be performed on the call (e.g. class charge, validate billing number)
- Connecting and releasing subscriber parties for the call
- Connecting and releasing specific network resources (e.g. voice links) at a service node's request
- Informing service nodes of call events and changes to resources associated with a session (e.g. subscriber party going on hook)
- Transitioning of calls between function providers
- Providing service node level maintenance (e.g. taking a service node out of service, returning a service node to service)
- Providing session pool level maintenance (e.g. taking a session pool out of service, returning a session pool to service)
- Generating logs and alarms at the switch on either a node basis or a session pool basis
- Ensuring communications exist between the switch and service nodes and between the switch and session pools via audits

3.2.1 Message Format

This section provides an overview of the OAP message format. For detailed information, refer to Chapter 4: "Message Format" .

The OAP message is formatted as shown below.



Each OAP message has one Class Header and one Operation header. Each operation may have zero or more data blocks.

3.2.1.1 OAP Class Header

The Class Header contains basic information that all messages require, such as routing information, message length, protocol version, and transport specific information. There is a header for each process that may act on a specific message, which allows messages to be routed to more specialized handlers.

The following are the predefined classes for OSSAIN:

- Call Processing: Used for messages that are call related. The operations within the message only make sense in the context of a specific call session.
- MIS: Used for sending event driven data about OSSAIN agents and call queues.
- Node: Used for messages that are node related. The operation in the message only makes sense in the context of an entire node (e.g node maintenance and audits).
- Node Alarm: Used for sending alarm information on a node basis.
- Node Log: Used for sending log information on a node basis.
- Session Pool: Used for messages that are related to session pools. The operation in the message only makes sense in the context of a specific session pool (e.g. session pool maintenance and audits).
- Session Pool Alarm: Used for sending alarm information on a session pool basis.
- Session Pool Log: Used for sending log information on a session pool basis.

In addition to the basic information, each class header has information specific to that class. For example, call processing requires a function identifier and a call identifier. These fields are included in the Call Processing Class Header. However, these fields are not needed for node or session pool maintenance processing and, therefore, are not included in the Node or Session Pool Class Header.

3.2.1.2 Operation Header

The Operation Header is a specific ROSE header and is dependent on the type of message. There are five basic types of OAP messages. Each type is independent of direction and message sender (i.e. switch and service node).

1. Inform: Used to send information updates resulting from unsolicited events (e.g. switch informs the service node of a change in a subscriber's hook status). It is also used to request an operation which does not require an acknowledgment. An inform message will not be responded to unless there is a protocol violation.
2. Request: Used to request a specific operation be performed. The receiver will respond to the request with either a success response message or an error response message.

3. **Success Response:** Used to indicate that the requested operation was successfully performed by the receiver of the request operation. A success response message will not be responded to unless there is a protocol violation.
4. **Error Response:** Used to indicate that the operation request was attempted, but failed. There was an error with the operation request. A set of possible errors is defined for each operation. The first error found when processing the operation request is returned. However, that error may not be the only error in the operation request. An error response message will not be responded to unless there is a protocol violation.
5. **Reject:** Used to indicate that there was a protocol violation or the message could not be parsed properly. If the message is an operation request or inform, an attempt is not made to perform the operation. A set of protocol violations is defined for OAP. A Reject is never sent as a response to a Reject message.

Table 1: “Message Type and Associated Operation Header” lists the message type to the associated operation ROSE header. Refer to Chapter 4: “Message Format” for detailed information on ROSE.

Table 1. Message Type and Associated Operation Header

Message Type	ROSE Operation Header
Inform	Invoke
Request	Invoke
Success Response	ReturnResult
Error Response	ReturnError
Reject	Reject

The Invoke Operation Header contains specific information about the operation, such as the operation identifier. Operations are requests (e.g. requesting a specific operation be performed, such as connect voice links) and informs (e.g. inform the message receiver of some event, such as a subscriber going on-hook). The Invoke header also has an invoke identifier which is returned in the response message. This identifier can be used by the sender of the operation to correlate a specific response to the corresponding invoke operation.

The ROSE headers used for responses indicate whether the operation was a success or not. The ReturnResult, ReturnError, and Reject headers contain the invoke identifier. The ReturnError header contains the error value and the Reject header contains the problem type and reject reason.

Table 2: “Message Type and Potential Responses” lists the message type and the expected response.

Table 2. Message Type and Potential Responses

Message Type	Success Response	Error Response	Reject
Inform	Not Applicable	Not Applicable	Applicable
Request	Applicable	Applicable	Applicable
Success Response	Not Applicable	Not Applicable	Applicable
Error Response	Not Applicable	Not Applicable	Applicable
Reject	Not Applicable	Not Applicable	Not Applicable

3.2.1.2.1 OAP Operations

Operations are requests and informs. Each operation and response has zero or more logical groupings of data fields, known as a data blocks. The data blocks are specified to be either required or optional. Refer to the definition of each inform operation, request operation, success response, and error response for the list of associated data blocks.

For the purpose of this document, the operations in this document have been divided between call processing operations and non-call processing operations. However, this division has no affect on the protocol itself.

Call processing operations are of class type Call Processing class. All other classes are non-call processing operations.

Table 3. Operation Group to Class Type

Operation Group	Class
Call Processing Operations	Call Processing
Non-Call Processing Operations	MIS
	Node
	Node Alarm
	Node Log
	Session Pool
	Session Pool Alarm
	Session Pool Log

The definition of each operation includes a list of preconditions, postconditions, side effects, possible responses, and a list of valid senders. The following is a list of all potential message senders:

- Call Processing Switch: The switch that communicates with a service node for call related activities.
- Maintenance Switch: The switch that communicates with a service node for non-call related activities.
- OSAC Remote: The switch that interacts with the service node for call related activities but does not maintain the service node. However, it does inform the service node of maintenance activities associated with the service node at the Remote.
- Active Service Node: The service node that has control of the call in a simultaneous interaction.
- Passive Service Node: The service node that is participating, but is not in control of the call in a simultaneous interaction.

The list of senders (i.e. a subset of the above list) stated with each operation can be used to determine whether the operation is unidirectional (sent by the switch only or sent by the service node only) or bidirectional (sent by both the switch and service node). For example, the valid senders for the Class Charge Request operation is Active Service Node. This operation is unidirectional and can only be sent by the service node that has control of the call.

Refer to Chapter 7: “Call Processing Operations” and Chapter 8: “Non-Call Processing Operations” for the definition of each operation in OAP. Refer to Chapter 9: “Data Block Descriptions” for a complete list of data blocks in OAP.

3.2.1.3 Data Blocks

Each operation and response has a predefined set of zero or more data blocks. Each data block is comprised of a data block Identifier, a data block byte count, and one or more data fields.

Each data block in an operation and response is specified as being mandatory or optional. If a data block is mandatory, the data block must be sent with the operation or response. If the data block is optional, it is sent only if it is applicable at that time.

Allowing for optional data blocks simplifies the protocol and reduces the message size (data blocks that are optional for a message are not always sent). The types of optionality supported are:

- An optional data block is sent only if it is applicable at the time (e.g. the Context Block data block is sent in a session begin message only if there is context block data).

- Multiples of the same data block may be sent for one operation (e.g. the Directory Number data block may be sent twice in a session begin message, one for the calling party and one for a forward party).
- From a set of data blocks, only one is sent. There may be several data blocks that contain similar information and only the one data block that is applicable is sent. For example, there are five types of trigger data blocks:
 - Trigger Information DB
 - Trigger Treatment DB
 - Trigger Release Cause DB
 - Trigger Ringing DB
 - Trigger DTMF DB

If the session begin is due to a trigger event, then only one out of the five trigger data blocks is sent.

The data blocks are listed in an arbitrary order for each operation and response. The data blocks can be sent in any order and nothing is inferred from the ordering.

Refer to Chapter 9: “Data Block Descriptions” for a complete list and definition of all OAP data blocks. Refer to Chapter 10: “Field Definitions” for a complete list and definition of all OAP fields.

3.3 Deterministic Nature

The OAP protocol is deterministic. For each operation sent in a request message, this document specifies the expected response message for a successful operation. Each operation is defined with a specific set of data blocks that are specified as required or optional. For errors or protocol violations, separate, equally deterministic response messages, with their own defined set of data blocks, are specified.

This document states the conditions under which request operations may be sent in order to receive a successful response message. Documentation for each operation and response indicates the reason for its invocation (i.e., successfully responding to an earlier request, indicating an error or protocol violation resulting from an earlier request, or announcing an unsolicited event).

In order to maintain the deterministic nature of OAP, certain preconditions must be imposed on some operations. Some incoming request operations may not make sense to the message receiver unless a particular set of request operations has been previously received or the call has specific characteristics. For example, service nodes may not class charge a call unless all directory numbers and the billing number has been entered. These preconditions will be clearly stated with each request operation in this document. If these preconditions are not met, an error response message will be returned. In many cases, possible failure reasons for an operation will align with the stated preconditions for that operation.

For each operation, this documentation states the resulting postconditions. These postconditions indicate that, once the requested action has been successfully performed, the call state or maintenance state aligns with the documented postconditions.

The error response for each operation request states whether there were any changes as a result of the failure.

3.4 Protocol Versioning

The protocol version number consists of the OAP release plus increment (X.Y). Each node in the OSSAIN network should be able to support releases that are 1, 2, and 3 releases older than the node's release. If the node's protocol release is X, then it must support X, X-1, X-2, and X-3 releases. All increments within any supported release that are less than what the node is at should also be supported (i.e. X-1 to X-3.0). The switch will support all increments of any supported releases from X-1 to X-3.0

As OSSAIN or OAP capabilities are enhanced, this protocol may be updated to support changed and/or new messages. The following describes how protocol compatibility is achieved between two nodes (switch and service node) running two different versions of OAP. The strategy described below is employed due to the complexity of determining protocol version that is added by OSSAIN Centralization. Refer to Section 3.4.3 "Protocol Version for Call Processing Class Messages" on page 92.

There are two protocol version numbers in each OAP Class Header. The first protocol version is the version of the message. The second protocol version is the maximum protocol version that the sender of the message is capable of communicating. Some common fields for each class header are shown below. The first seven fields in each class header and the format of these fields are fixed (i.e. the format is the same regardless of the protocol version).

Table 4. Common Class Header Fields

Class Header ID
Options
Length of following Message
Offset to Operation Header
Message Protocol Version
Sender's Maximum Protocol Version
Source Node ID
Other Common Class Header Fields
Class Dependent Fields

The Options field is used to set specific flags. Bit 0 in the Options field is set by a node to indicate to another node that the other node has sent a message at a protocol version that is not optimal. The Message Protocol Version is considered not optimal if one of the following occurs:

- Message Protocol Version is higher than what the receiving node can process (e.g. the message protocol version is 4.0 and the receiving node's maximum protocol version capability is 3.0)

- Message Protocol Version is lower than what both the sending node and receiving node can process (e.g. the sender's maximum protocol version capability is 4.0 and the receiving node's maximum protocol version capability is 4.0, but the message protocol version is 3.0)
- Sender's Maximum Protocol Version Capability is more than three releases lower than what the receiving node can process (e.g. the sender's maximum protocol version is 2.0 and the receiving node's maximum protocol version capability is 6.0)

When a node receiving a message determines that the Message Protocol Version for the message is not optimal and version negotiation should be performed, it sends a Protocol Version Error (PVE). An OAP message is a PVE if bit 0 in the options field is set.

The same class header received in the original message is used for the PVE. The node sending the PVE sets bit 0 in the Options field, updates the Sender's Maximum Protocol Version to reflect its protocol version, and updates the Source Node ID in the class header. The Length of Following Message field is also updated to reflect the length of the PVE message (i.e. class header minus the first six bytes plus ROSE Reject header). The rest of the information in the class header of the original message is returned in the PVE. This allows the receiver of the PVE to route the message to the process that should handle the PVE.

Note that bit 0 in the Options field is only set in a PVE message. In all other messages, bit 0 should not be set.

The PVE also includes a ROSE Reject Operation Header. The problem type in the operation header is General Problem and the reject reason is BadlyStructured APDU. The invoke ID in the Operation Header of the PVE is not expected to match the invoke id of the message being rejected. However, the value used for the PVE message should be within the valid range of invoke ids (i.e. #0100 to #FFFF).

A node may receive a message in which the Message Protocol Version is the same as the Sender's Maximum Protocol Version and both are lower than what the receiving node can support. If the protocol version is within three versions of what the receiving node can support, instead of returning a PVE, the receiving node can drop to the lower protocol version and continue to process the message.

If the switch receives a PVE for a node or session pool that is in service, a log will be generated and the PVE message will be ignored.

The following sections describe how the Message Protocol Version is determined.

3.4.1 Protocol Version for Node Class Messages

The protocol version for non-call processing messages to a service node is determined when the service node is returned to service (RTS). The switch sends an RTS message to the service node. The Message Protocol Version for the first RTS message is the maximum protocol version that the switch is capable of sending.

If the service node determines that the Message Protocol Version (MPV) is not optimal (e.g. higher than what the service node can process), it responds with a PVE. The PVE has the same Node Class Header that was used for the RTS message, with bit 0 of the Options field set. However, the Sender's Maximum Protocol Version is updated to reflect the maximum protocol version capability that the service node is capable of sending and the Source Node ID is updated with the service node ID. The length field is also updated to indicate the length of the PVE message. The rest of the header is not changed.

When the switch receives the PVE, it sets the node level protocol version to the lower of the switch's protocol version and service node's protocol version. The switch will use this protocol version to communicate with the service node for node level messages until it receives another PVE from the service node.

All node level messages (except for the Test message) from the service node must be at the protocol version that was negotiated during the Node RTS. If the switch receives a node level message at a different version, then it will return a protocol version reject. This reject message will be similar to a PVE message except the options bit will not be set. The problem type in the Reject operation header will be General Problem and the reject reason will be BadlyStructured APDU. When the service node receives this reject, it should resend the original message using the MPV determined during the Node RTS.

The message protocol version can be tested with the Node Test Request message exchange. Upon receiving a Node Test response message, the switch will renegotiate the protocol version if the version is not optimal.

The service node's protocol for node level messages can not be upgraded while it is in service with the switch. Doing so may cause the pre-determined MPV for node level messages to no longer be optimal. In order to change the service node's protocol version (e.g. upgrading the service node) the service node must be taken out of service and then returned to service again. During the next RTS, the protocol version can be re-negotiated.

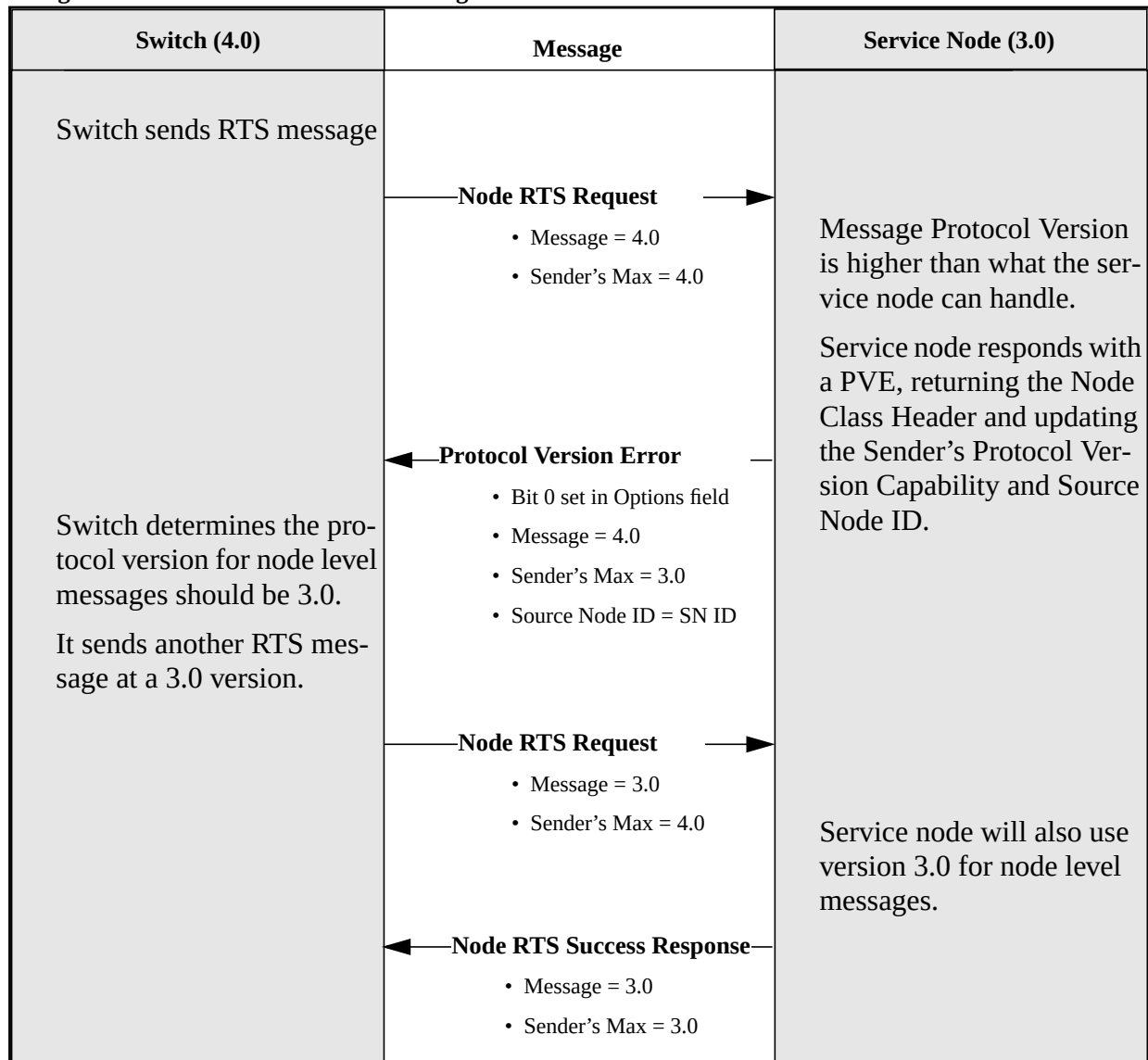
If the switch is upgraded, the service node's state will go out of service. The service node must go through an RTS sequence before it can service any calls. Therefore, the protocol version for node level messages will be re-negotiated using the switch's new maximum protocol version.

The protocol version determined for Node Class messages should be used for messages using the following class headers:

- MIS
- Node
- Node Alarm
- Node Log

Figure 2. “Node Level Protocol Versioning” shows an example of when a service node responds with a PVE. In this example, the switch’s maximum protocol version capability is 4.0 and the service node’s maximum protocol version capability is 3.0.

Figure 2. Node Level Protocol Versioning



3.4.2 Protocol Versioning for Session Pool Class Messages

The protocol version for session pool level messages is determined the same way as node level messages. The switch sends an RTS message to a session pool, using the switch's maximum protocol version capability for the Message Protocol Version. The service node responds with a PVE if necessary. In this case, the PVE has the same Session Pool Class Header that was used in the RTS message. However, the Sender's Maximum Protocol Version is that of the session pool and the Source Node ID is the service node's identifier. The length fields are also updated.

When the switch receives the PVE, it sets the session pool level protocol version to the lower of the switch's protocol version and session pool's protocol version. The switch will use this protocol version to communicate with the service node for session pool level messages until it receives another PVE from the service node.

All session pool level messages from that session pool must be at the protocol version that was negotiated during the Session Pool RTS. If the switch receives a session pool level message at a different version, then it will send a protocol version reject. This reject message will be similar to a PVE message except the options bit will not be set. The problem type in the Reject operation header will be General Problem and the reject reason will be BadlyStructured APDU. When the service node receives this reject, it should resend the original message using the MPV determined during the Session Pool RTS.

The message protocol version can be tested with the Session Pool Test Request message exchange. Upon receiving a Session Pool Test response message, the switch will renegotiate the protocol version if the version is not optimal.

The session pool's protocol for session pool level messages can not be upgraded while the session pool is in service with the switch. Doing so may cause the pre-determined MPV for session pool level messages with the session pool to no longer be optimal. In order to change the session pool's protocol version (e.g. upgrading the session pool) the session pool must be taken out of service and then returned to service again. During the next RTS, the protocol version can be re-negotiated.

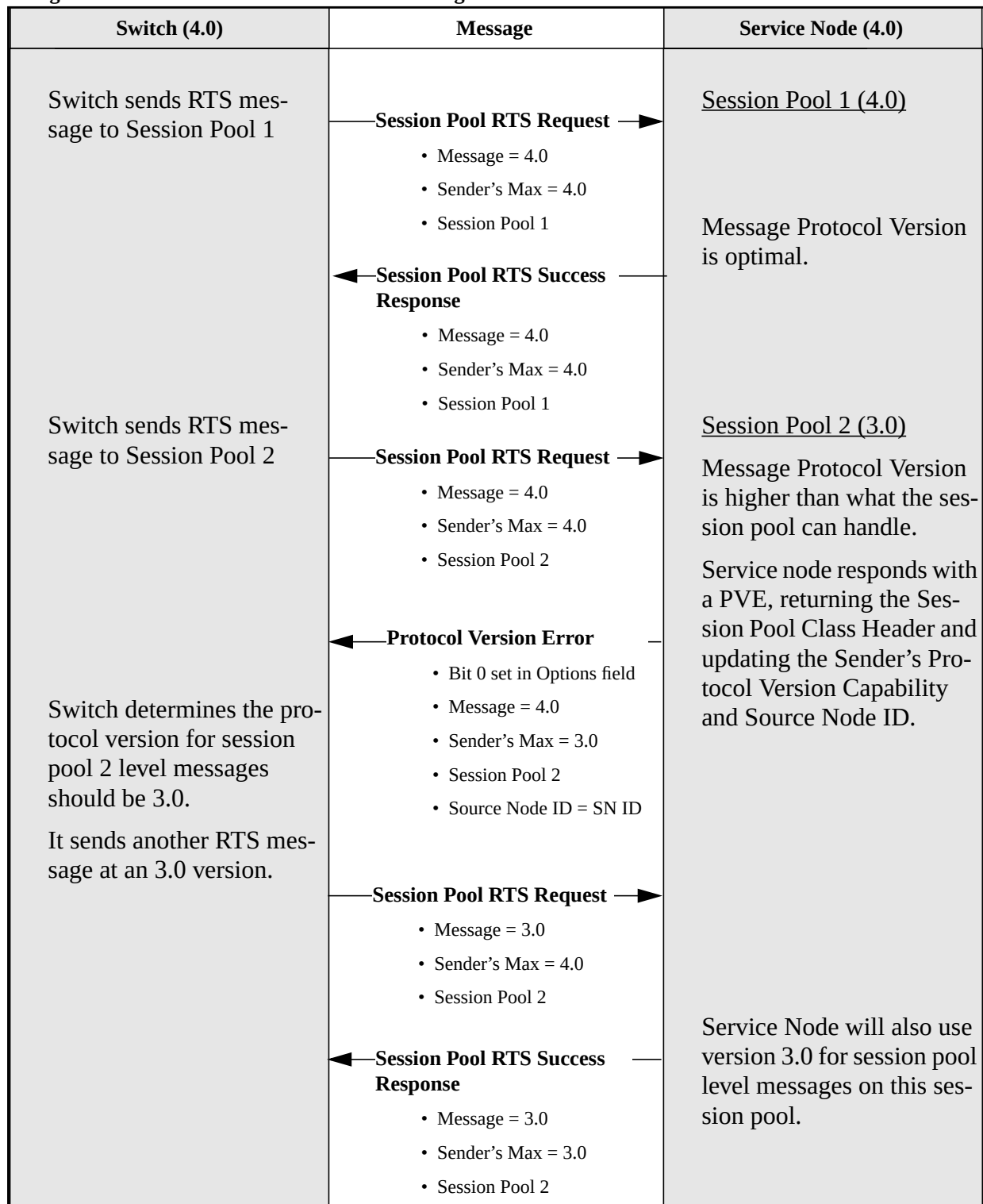
If the switch is upgraded, the service node's state and the states for all the session pools on the service node will go out of service. The session pools must go through an RTS sequence before it can service any calls. Therefore, the protocol version for session pool level messages will be re-negotiated using the switch's new maximum protocol version.

The protocol version determined for Session Pool Class messages should be used for messages using the following class headers:

- Session Pool
- Session Pool Alarm
- Session Pool Log

Figure 3. “Session Pool Level Protocol Versioning” shows an example for determining the protocol version for two session pools on a service node. Session pool 1 is at version 4.0 and session pool 2 is at version 3.0. The session pool ID is a class specific field in the Session Pool Class Header.

Figure 3. Session Pool Level Protocol Versioning



3.4.3 Protocol Version for Call Processing Class Messages

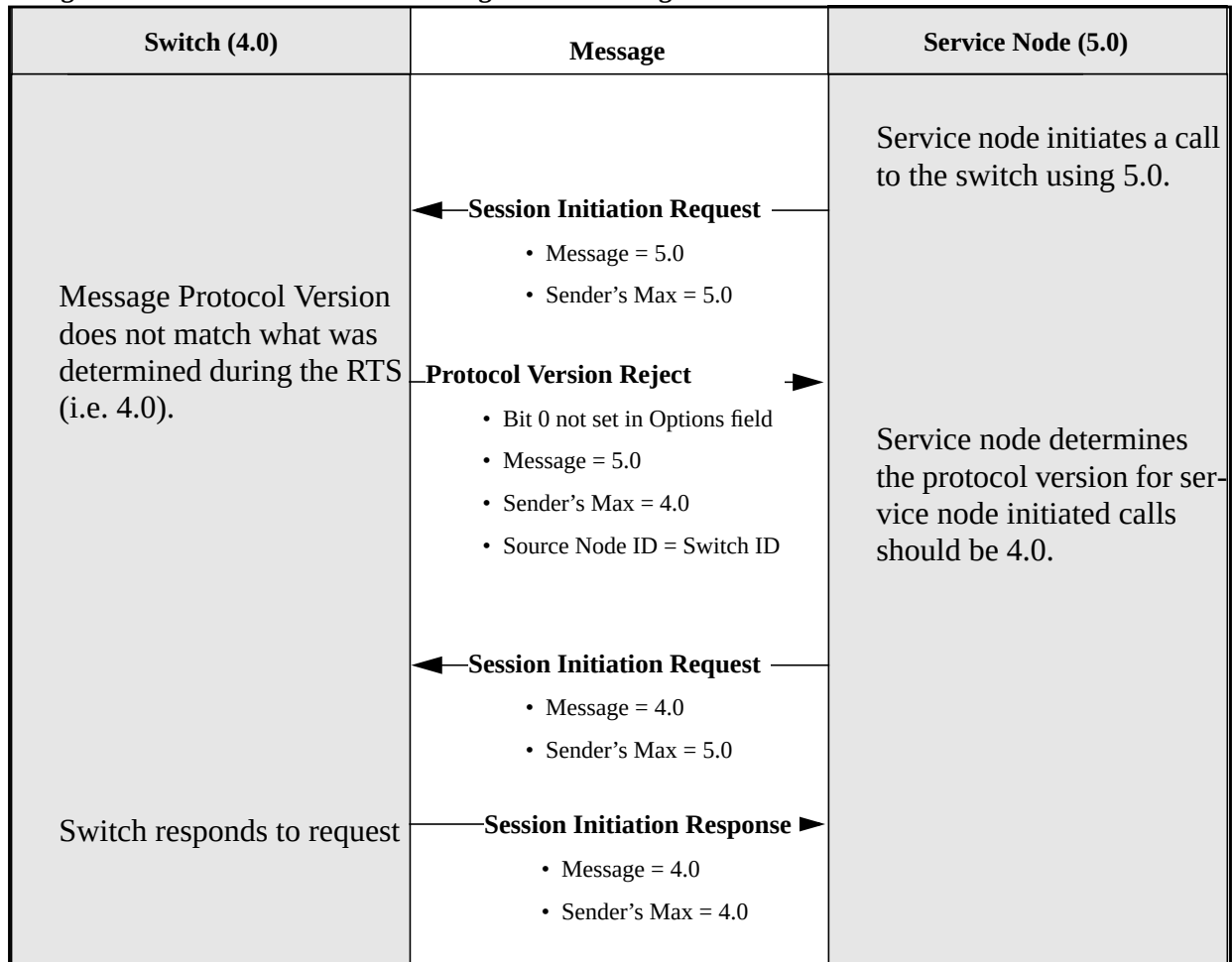
The Standalone switch uses the MPV that is determined during the RTS of a session pool for all Call Processing Class messages on that session pool. Refer to section 3.4.2 “Protocol Versioning for Session Pool Class Messages” for details on how the MPV is determined. The service node must also use the MPV for all Call Processing messages on that session pool. This is the case for session pools used for subscriber originated calls and for session pools used for service node initiated calls.

If the switch receives a message from the session pool at a different protocol version than the MPV, then the switch will send a protocol version reject back to the service node. This reject message will be similar to a PVE message except the options bit will not be set. The problem type in the Reject operation header will be General Problem and the reject reason will be BadlyStructured APDU. When the service node receives this reject, it may resend the original message using the MPV determined during the Session Pool RTS.

A PVE response by the service node to a Session Begin Inform message is not supported. Therefore, a session pool must be taken out of service and then returned to service before the protocol version can change for a session pool.

Figure 4: “Protocol Version Error During Call Processing” shows an example when a protocol version error occurs during call processing. In this example, the switch is at version 4.0 and the service node is at version 5.0. The MPV is 4.0, which was determined during the Session Pool RTS.

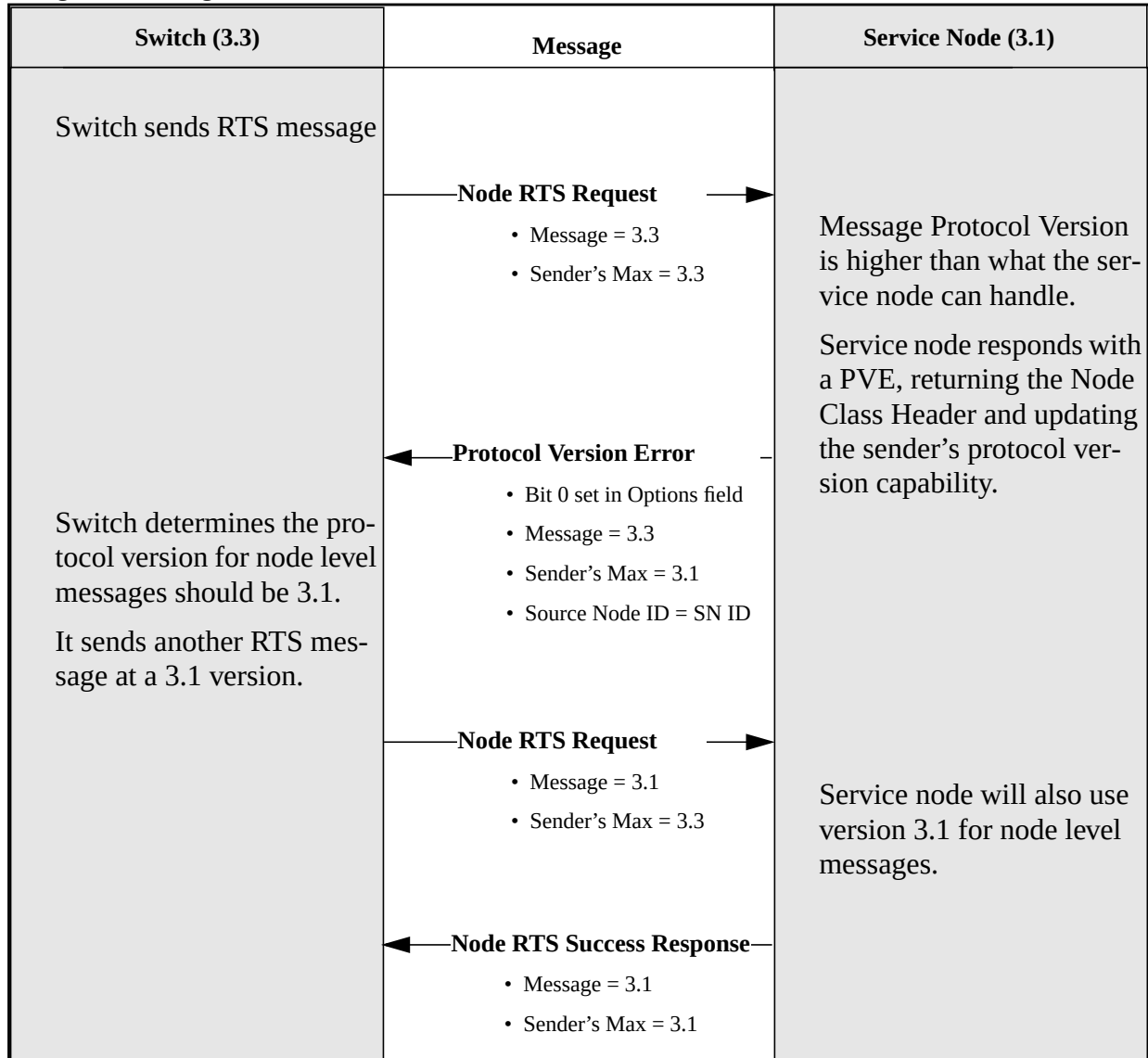
Figure 4. Protocol Version Error During Call Processing



3.4.4 Protocol Increment

For each protocol release, there may be zero or more increments for that release (due to patching new or changed functionality). For example, there may be 3 increments of OAP release 2.0 in TOPS 06 (i.e. 2.1, 2.2, and 2.3). Each node should support all increments of any supported OAP releases that are less than the one the node's at (i.e. release X-1 down to X-3.0). The switch will support all increments of each supported release.

The increment is also used for determining the Message Protocol Version. For example, if the maximum protocol version the switch can communicate at is 3.3 and the maximum protocol version a service node can communicate at is 3.1, the MPV for node level messages should be set to 3.1. This example is illustrated in Figure 5: “Message Protocol Version with Increment”.

Figure 5. Message Protocol Version with Increment

3.4.5 Protocol Version with OSAC

OSSAIN Centralization (OSAC) allows resources for a service node to be centralized, allowing multiple switches to share the services provided by a centralized service node. Maintenance is performed between service node and OSAC Host switches only. However, call processing can occur between service nodes and OSAC Remote switches. Refer to 13.7 “OSSAIN Centralization” on page 967 for more information on OSAC.

Since there are no node level and session pool level maintenance messages between an OSAC Remote and a centralized service node, the message protocol version is determined differently for Call Processing class messages between the service node and the OSAC Remote switch.

When a service node goes through a maintenance action with the OSAC Host, the Host informs the OSAC Remotes in the OSAC network. After a Session Pool RTS sequence between the Host and a centralized service node, the Host informs all OSAC Remotes of the session pool’s maximum protocol version. The Remotes use this to determine what the message protocol version should be for call processing messages with the session pool.

There is a small window where new subscriber originated calls are being processed at the Remote before the Host has a chance to update the Remote with the new session pool’s protocol version. In this case, the Remote may not have the latest protocol version stored for the session pool and the message protocol version may not be optimal. To eliminate this problem, when the Host allocates a session to the Remote for a new call, the Host will also return the session pool’s maximum protocol version. Therefore, the Remote can also determine the optimal message protocol version at the beginning of each subscriber originated call.

For subscriber originated calls, the OSAC Remote receives the maximum supported protocol version for a session pool on the service node. This is sent by the OSAC Host during maintenance and at the beginning of each call when the Host returns the service node session to the Remote. The OSAC Remote uses the maximum protocol version to determine the message protocol version for call processing messages on that session pool. The service node must also use this message protocol version when messaging with the Remote. If the Remote receives a message from the service node that is not at the message protocol version that the Remote determined, the Remote will return a protocol version reject. Options bit zero will not be set.

For service node initiated calls, the service node must track the maximum protocol version of the Remote, as follows:

- If the service node does not have the Remote’s protocol version stored, then the service node should use its maximum protocol version to initiate a call to an OSAC Remote. The service node may receive a protocol version reject message from the OSAC Remote (options bit 0 will not be set) for the Session Initiation Request. The Remote sends the reject when the protocol version used to initiate the call is not optimal (e.g. the Remote’s protocol version is less than the message protocol version of the message or does not

match the message protocol version that the Remote determined during its maintenance sequence between the Host and Remote). The protocol version reject will have the Remote's maximum protocol version in the Call Processing Class Header. The service node should store this version.

- If the maximum protocol version for the Remote is stored, the service node should use it to determine the optimal protocol version to use for Session Initiation Requests. If the Remote has been upgraded before the service node is updated with the Remote's protocol version, the Remote may return a protocol version reject (options bit zero is not set) for the Session Initiation Request. The service node should renegotiate the optimal protocol version and store the Remote's new protocol version.

Figure 6: "Service Node Initiated Call to an OSAC Remote" shows an example where the message protocol version is higher than what the Remote can process.

3.4.6 Protocol Evolution Strategy

An OAP protocol version is delivered with each DMS TOPS software release. However, there may not be a new protocol version for each TOPS release (i.e. the OAP version number is independent of the TOPS release number). The protocol version is updated only if there are changes to the protocol. The following example may help to clarify this point:

- a. TOPS 05 is released with OAP 1.0.
- b. There are protocol changes in the next TOPS release so the protocol version is increased. TOPS 06 is released with OAP 2.0.
- c. In the next TOPS release, there are no changes to the protocol. The protocol version is not increased. TOPS 07 is released with OAP 2.0.

The switch's maximum protocol version is the same for Node Class, Session Pool Class, and Call Processing Class messages.

The following are general strategies employed for evolution and version control of OAP:

1. The protocol version is increased only if there are changes to the protocol.
2. The protocol increment is changed only if there are changes to the protocol after it has been released.
3. The protocol version of this document is the same as the protocol version of the OAP software that it is released with. The Revision number of this document is incremented when there are document only changes.
4. The switch generates a Protocol Violation log whenever it returns a PVE to a service node. The log prints the Message Protocol Version, the Sender's Maximum Protocol Version, and the switch's protocol version. This will help to assure that compatibility between service nodes and the switch is not compromised and serve as a reminder that the switch is interacting with a service node with a different protocol version.
5. Each class header should have a unique class header ID.
6. For all class headers, the format and fields up to and including the Source Node ID will never change.
7. Obsolete operations, data blocks, and fields are supported for up to three releases.
8. An operation request must have a success response and an error response defined with it.
9. Each operation and success response is defined with a set of zero or more mandatory and/or optional data blocks.
10. New data blocks can be added to any operation or response in a future OAP release.
11. The order of the data blocks sent in an operation or response is not important and will not imply anything.
12. A data block can be associated with multiple operations and responses.
13. New fields can be added to any data block in a future OAP release.

14. A new field in a data block must be added at the end of the data block or must replace a padding field.
15. A field can be associated with multiple data blocks.
16. An error id can have a corresponding data block.
17. An error id can be associated with multiple operation requests.

There may be times when changes occur to the protocol that make it non-backwards compatible. This would require all nodes to communicate at some new minimum protocol version. Upgrades are necessary to support a specific functionality or to support evolution of OAP. When upgrades are needed, OAP and this protocol document will be updated and service node vendors will be notified when such changes are required.

The following are strategies employed when upgrades are required due to a change in OAP that makes it non-backwards compatible:

1. Service nodes should be upgraded before the switch is upgraded. This is consistent with upgrade strategies for DMS switch peripherals.
2. Protocol version upgrades should be done when traffic level is low.

The following are strategies employed when the protocol is incremented:

1. Fields will not be deleted in an increment.
2. New fields are added to the end of a data block.

3.5 Sequence Numbers

Sequence numbers are used to keep track of the order of messages in the message flow, to aid in the tracking of lost messages. The sequence number is contained in all OAP class headers and is included in every message between the switch and service node.

Each message contains only one sequence number. The number indicates the number of messages that the message sender has sent for this session, including the current one.

3.5.1 Sequence Number for Node, Node Alarm, Node Log Class Headers

Currently, the sequence number for Node based class messages is not used or processed by the switch. This field is set to a nil value (i.e. zero) for all messages from the switch to service nodes.

3.5.2 Sequence Number for Session Pool, Session Pool Alarm, Session Pool Log Class Headers

Currently, the sequence number for Session Pool based class messages is not used or processed by the switch. This field is set to a nil value (i.e. zero) for all messages from the switch to service nodes.

3.5.3 Sequence Number for Call Processing Class

For each call session, there are two sequence numbers: the switch's sequence number and the service node's sequence number. The switch's sequence number is independent of the service node's sequence number.

Each node (switch and service node) in the call keeps track of its own sequence number and stores the sequence number for the other node. The sequence number in the message is the sender's sequence number.

Both sequence numbers are initialized to zero at the start of each new session. The first message from the switch to the service node for a new call session will have a sequence number of one. The switch will increment the sequence number each time it sends a message to the service node. The first message from the service node to the switch must have a sequence number of one. The service node must increment the sequence number each time it sends a message to the switch.

The switch keeps track of the last received sequence number for each session involved in a call. When a message arrives from a service node, the sequence number is processed as follows:

- If the message sequence number is less than the sequence number expected by the switch, the switch will not process the message. The switch will keep a count of the number of times the switch receives an out of sequence message (via an operational measurement count). The service node message sequence number stored in the switch will not be updated and the service node will not receive a response if one was required. This allows

the service node to time out and take an appropriate action. The switch does not do anything to recover from the out of sequence message since the service node has control of the call.

- If the message sequence number is equal to the sequence number expected by the switch, the switch will update the service node message sequence number stored in the switch to match the sequence number received in the current message. The switch will then process the message.
- If the message sequence number is greater than the sequence number expected by the switch, the switch will update the service node message sequence number stored in the switch to match the sequence number received in the current message. It will also peg the out of sequence operational measurement register. The switch will then process the message. Refer to Section 3.6 “Error Handling For Lost Messages” on page 104 on what the switch does when it detects a lost message.

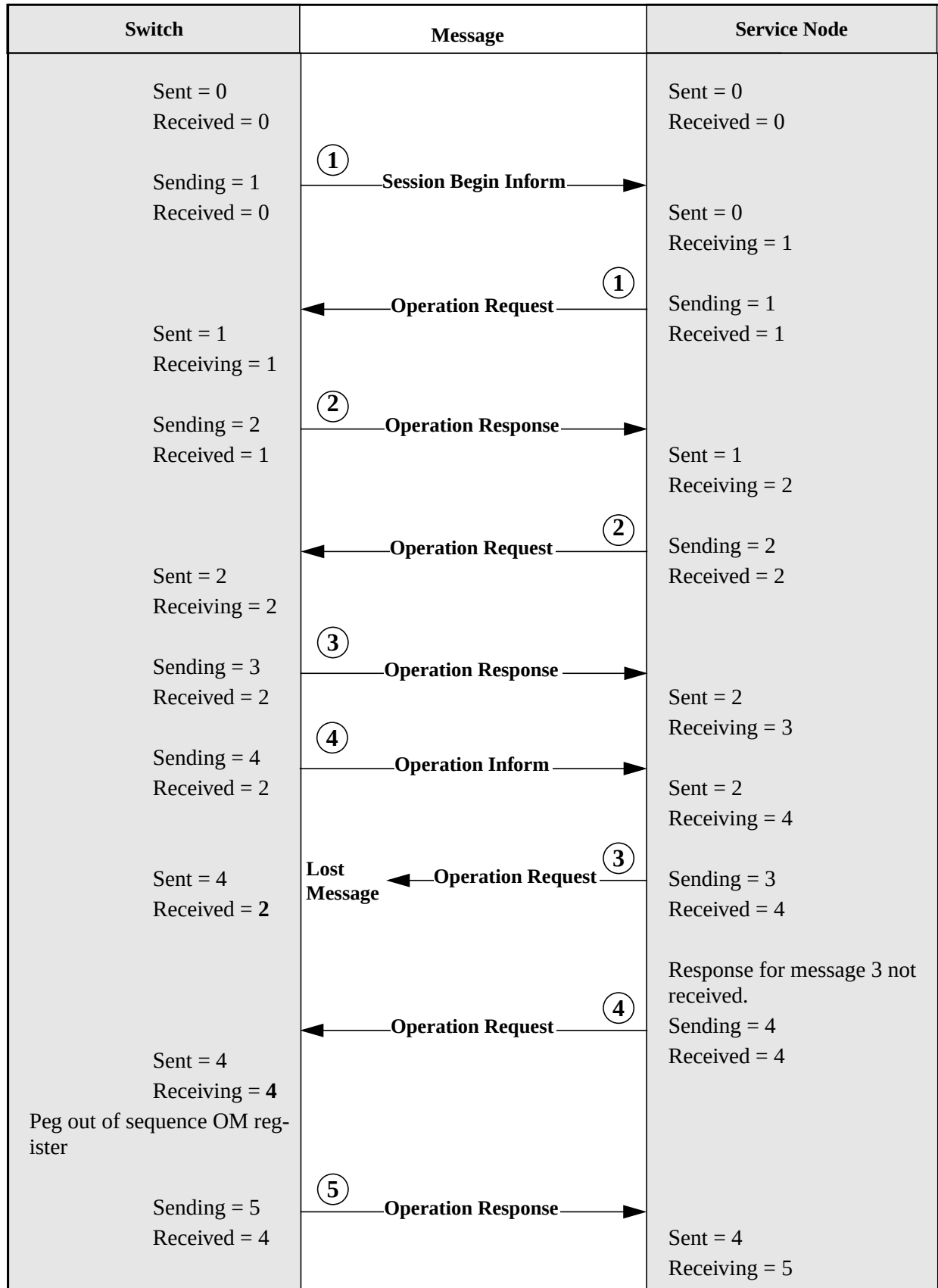
An example of the use of sequence numbers is shown in Figure 7: “Sequence Number Example”.

3.5.4 Sequence Number for MIS Class Headers

The sequence number stored in the MIS class header keeps track of the number of MIS operation messages sent by the switch. It gets incremented every time an MIS OSSAIN Inform message is sent. The MIS vendor can follow the sequence numbers to determine whether or not any MIS buffers were lost from one MIS transmission to the next. The number starts at zero when an MIS node is datafilled in switch table OANODINV. It gets reset on all switch restarts or if the node is deleted from the table.

Note that this sequence number is independent of the MIS Sequence Number ID field in the OSSAIN Call Queue Event, OSSAIN Call Session Event and OSSAIN Non-call Session Event Data Blocks. The MIS Sequence Number ID field keeps track of the number of queue events for a particular queue or the number of session events for a particular session.

Figure 7. Sequence Number Example



3.6 Error Handling For Lost Messages

Because OSSAIN is developed to be service independent, it is impossible for the switch to know what message flow to expect from a service node. Therefore, the switch does not have elaborate schemes for handling lost messages. However, it will be able to detect when a message has been lost when it receives a message with an out of order sequence number.

3.6.1 General Strategies

When the switch detects a message was lost from a service node, it will **not** send any type of message indicating that there was a lost message. The switch will increment an operational measurement count used to track the number of out of sequence message and may or may not process the message. Refer to Section 3.5 “Sequence Numbers” on page 100 for details on what the switch does for an out of sequence message.

Service nodes will also be able to detect when there was a lost message, either by receiving an out of sequence message or by not receiving a response message for a request operation (i.e. the service node times out waiting for a response). If a service node does not receive a response, then the lost message could have been the operation message (i.e. the operation was not received by the switch and, therefore, not processed) or the response message (i.e. the operation was received by the switch and processed, but the response to the service node was lost).

Subsequent request messages may fail due to the switch not processing data from a lost operation message. Because service nodes have control of the call, it is their responsibility to correct any actions that result due to a lost message. Service nodes can resend the operation that was previously lost to prevent or correct failures. The new message must have a new sequence number.

If a service node resends an operation due to not receiving a response (i.e. the operation was received by the switch but the response to the service node was lost), the switch may attempt to perform the operation again. For some operations, the second invocation may fail or produce undesirable results (e.g. appending the same AMA module twice). Specific failure strategies have been documented with those operations that may cause a problem if the switch performs them twice. The definition of inform operations indicate what happens if the message is lost.

There may be times when the switch has to terminate a call (for severe error cases). It may not be possible to send a call end message to the service node. When this occurs, all outstanding requests from the service node will be aborted and the switch will end the call. The service node will not receive anything from the switch regarding the aborted call.

3.6.2 Switch Timers

Sanity timers are set by the switch to ensure resources are not left connected to dead calls. When the timer expires, the switch automatically frees up any resources associated with the timer. The resources freed may include subscriber facilities, connections to external entities (e.g. service nodes), and switch call resources.

Typically, if a message is lost, the call will not progress normally. Subscribers connected to the call can be expected to go on-hook. The switch will eventually terminate the call. There is a Session Begin timer, an On-Hook sanity timer, and a Call sanity timer to prevent service nodes or any switch resources from being connected indefinitely to a dead call. These timers are set through switch datafill.

The Session Begin timer is used if the SBTIMER field in table OAFUNDEF is set to Y. If this field is set to Y, this timer is initiated when a Session Begin Inform is sent to the service node. If the switch receives an OAP operation from the service node before the timer expires, the timer is cleared. If the timer expires, the call is routed based on origination failure datafill in table OAFNDISP.

There is an On-Hook timer for each subscriber party connected to the call. The On-Hook timer for a particular subscriber is initiated when the switch detects that the subscriber has gone on-hook. The service node is informed when the subscriber goes on-hook. If the timer expires, resources associated with the subscriber connection are released, the timer is cleared (i.e. cancelled), and a log is generated. The subscriber going back off-hook has no effect on the timer.

The call sanity timer is initiated under the following conditions:

- the originator's On-Hook timer has expired and there is no terminating party connection
- the originator's and terminator's On-Hook timers have expired
- the terminator's On-Hook timer has expired and there is no originating party connection
- there are no originating or terminating party connections.
- the Session Begin timer is not set

The Call sanity timer is reset (i.e. cleared and reinitiated) if a message is received from the service node associated with the call. The timer is cleared if a subscriber is subsequently connected to the call. If the Call sanity timer expires, the call is taken down, resources associated with the call are released, and a log is generated. A message is sent to the service node indicating that the call has ended. Any subsequent messages sent by the service node to the switch pertaining to the aborted call will not be processed. The service node will not receive any responses for those messages.

The On-Hook timer and Call timer never run simultaneously. The Call sanity timer is initiated when all On-Hook timers expire for all connected parties or when there no parties remain connected.

3.7 General Message Flow

3.7.1 Call Processing Message Flow

The basic message flow between a switch and a service node controlling the call can be divided into three areas: beginning a session, servicing a call, and ending a session.

- **Beginning a Session**

A session between the switch and a service node is either originated by the switch or initiated by the service node.

When a call arrives at the switch requiring an OSSAIN function, the switch selects a session to a service node providing the function and sends a Session Begin Inform operation message. The service node can choose to either accept the call or not accept the call. If the service node accepts the call, it can begin providing a function for the call, which may result in sending requests to the switch to perform required operations. If the service node cannot service the call, it can send a message to end the session with the switch and release itself from the call.

A service node may initiate a session to the switch by sending a Session Initiation Request operation message to the switch. The switch will return a response message indicating whether or not the switch was able to begin the session.

After a call has been floated, the call may recall to a service node due to a trigger event. In this case, either a Session Begin Inform will be sent to the switch (if the service node is given control of the call) or a Trigger Event Inform will be sent (if the service node does not need control but just needs to be informed that the trigger event occurred). The inform message sent is based on switch trigger datafill.

- **Servicing a Call**

When a call is at a service node, the service node is in control of the call. The service node may send request and informational messages, via OAP, to the switch to perform specific operations (e.g. connect voice link, validate billing number). Once the switch receives a request, it attempts to perform the requested operation. If the switch is able to successfully perform the operation, it returns a success response message. If the switch attempts to perform the operation but fails, it returns an error response message with the appropriate error reason. The success and error response messages may or may not include the call attributes that have changed. The definition of each operation request in this protocol includes the postconditions that result if the operation execution was successful.

If the switch is unable to parse the message or detects a protocol violation, it may send a Reject message to the service node.

The switch sends an inform message to the service node when an unsolicited event occurs during the call (e.g. subscriber going on hook).

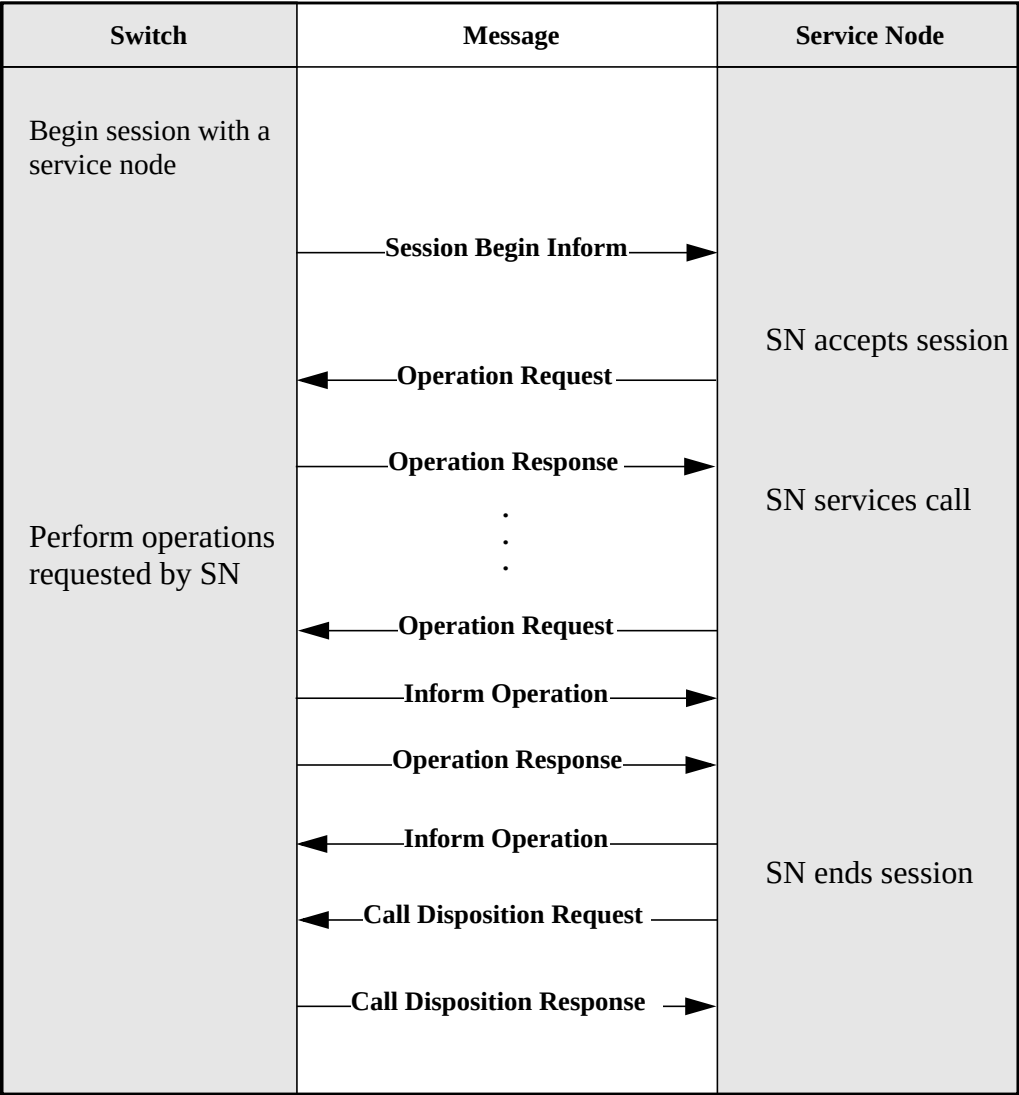
- **Ending a Session**

The session between the switch and a service node ends at a request by the service node or when the switch ends the session due to aborting the call. Service nodes have several ways of ending a session with the switch. The most basic way is to simply end the call by sending the End Call Request. This request essentially terminates the call and switch resources are released. Any operations the switch may actually be performing for the call on behalf of the service node are aborted.

Other service node requests that cause the session between the switch and a service node to end include:

- Call Merge Request
- Call Resume (Preprocessing Session)
- Float the call
- Release Node Request (Simultaneous Interactions)
- Route the call to treatment
- Transfer the call to a carrier
- Transfer the call to a control list

Figure 8. Basic Subscriber Originated Call Processing Message Flow



3.7.2 Non-Call Processing Message Flow

OSSAIN provides maintenance functionality for OSSAIN service nodes. There are two levels of maintenance: Node and Session Pool.

3.7.2.1 Node Level

- Alarms

When a service node goes out of service, an alarm is generated at the switch that maintains the service node. If the service node is returned to service, the alarm is cleared.

Alarms are also generated when the service node sends in a Node Alarm Inform message to the switch. An associated log may also be generated and included in the inform message.

- Audits

A passive auditing strategy is used to audit service nodes. A receipt of a successful message from a service node by the switch is considered a passive audit. By successful, the message should pass verification on the fixed part of the class header plus the session pool and session id fields of the call processing header. If successful OAP messages are regularly received from the service node, the switch does not need to send an explicit audit; the service node is considered to be in service. However, if no messages have been received within some amount of time (i.e. a predefined number of audit periods), a Node Audit Request message is sent to the service node to verify the node's sanity.

Once the number of consecutive failures exceeds field AUDTRIES in table OANOD-INV, the service node is taken out of service.

Service nodes should not set any timers for expected audits from the switch.

- Manual Service Node Maintenance

Service nodes can be maintained at the switch with the following basic maintenance capabilities:

- Busy a service node (i.e. take the service node out of service)
- Return a service node to service
- Test the connectivity between the switch and service node. This is done by sending one IP ping, which is a network level message. OSSAIN uses an ICMP (Internet Control Message Protocol) echo message to verify that remote systems IP software is up and operational. If the IP ping is successful, a test message is sent to the service node.

- Automated Service Node Maintenance: The following service node maintenance functionality is supported by OAP:

- Service node level audits, which originate only from the switch
- Switch will automatically begin the RTS sequence for service nodes that are system busy. The RTS is periodically sent until a response is received.

3.7.2.2 Session Pool Level

- Alarms

When a session pool goes out of service, an alarm is generated at the switch that maintains the service node associated with the session pool.

Alarms are also generated when the service node sends in a Session Pool Alarm Inform message to the switch. An associated log may also be generated if included in the inform message.

- Audits

A passive auditing strategy is used to audit session pools. A receipt of a successful message from a service node by the switch is considered a passive audit. By successful, the message should pass verification on the fixed part of the class header plus the session pool and session id fields of the call processing header. If successful OAP messages are regularly received from the session pool, the switch does not need to send an explicit audit; the session pool is considered to be in service. However, if no call processing messages have been received within some amount of time, a Session Pool Audit Request message is sent to the session pool to determine the state of the session pool.

Once the number of consecutive failures exceeds field AUDTRIES in table OANOD-INV, the session pool is taken out of service.

- Manual Session Pool Maintenance

- Busy a session pool (i.e. take the session pool out of service)
- Return a session pool to service
- Test the connectivity between the switch and service node using an IP ping.
- Make all idle sessions in a session pool unavailable for call processing. Active calls are allowed to progress normally. Once a call has ended, the session the call was using is made unavailable.

- Automated Session Pool Maintenance: The following session pool maintenance functionality is supported by OAP:

- Session pool level audits, which originate only from the switch.
- Switch will automatically begin the RTS sequence for session pools that are system busy.

Figure 9: “Busy Node Message Flow” shows an example message flow when a service node is manually busy’d at the switch.

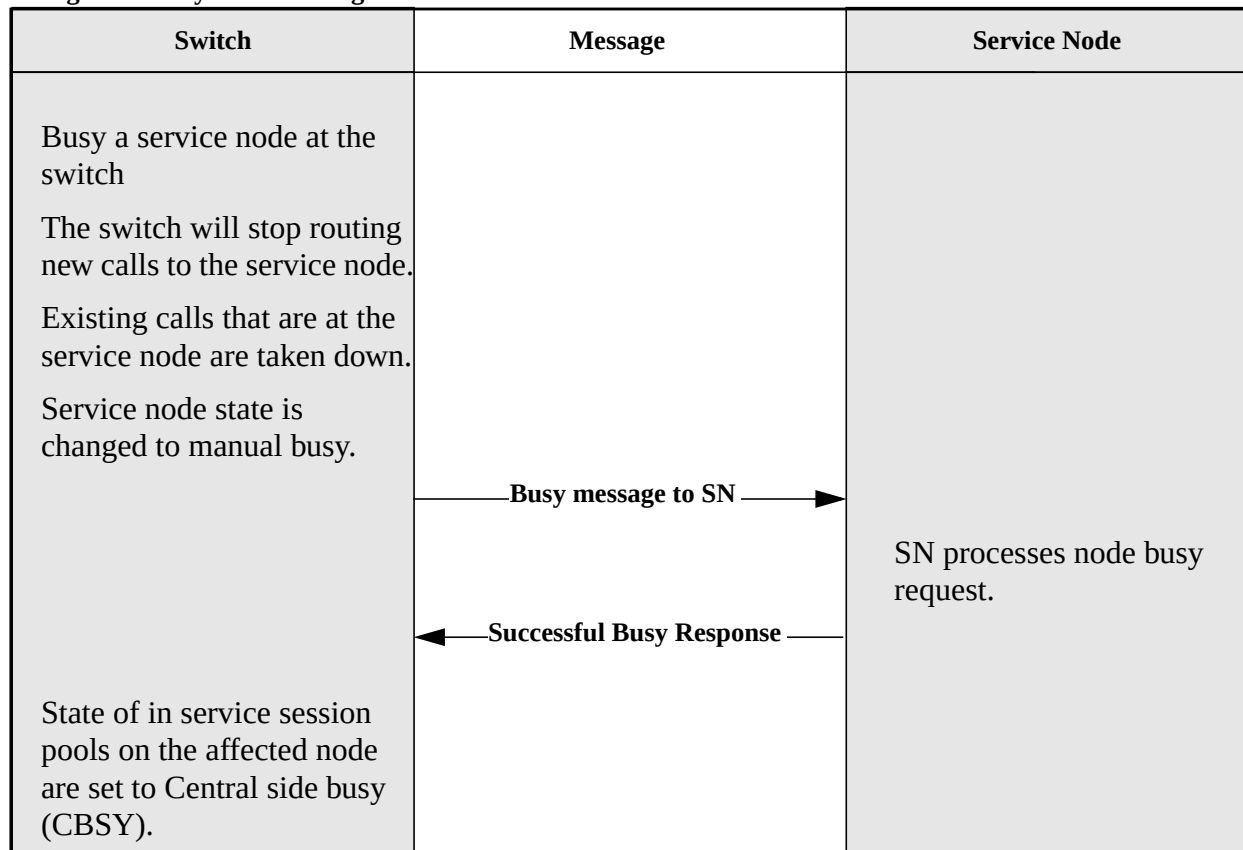
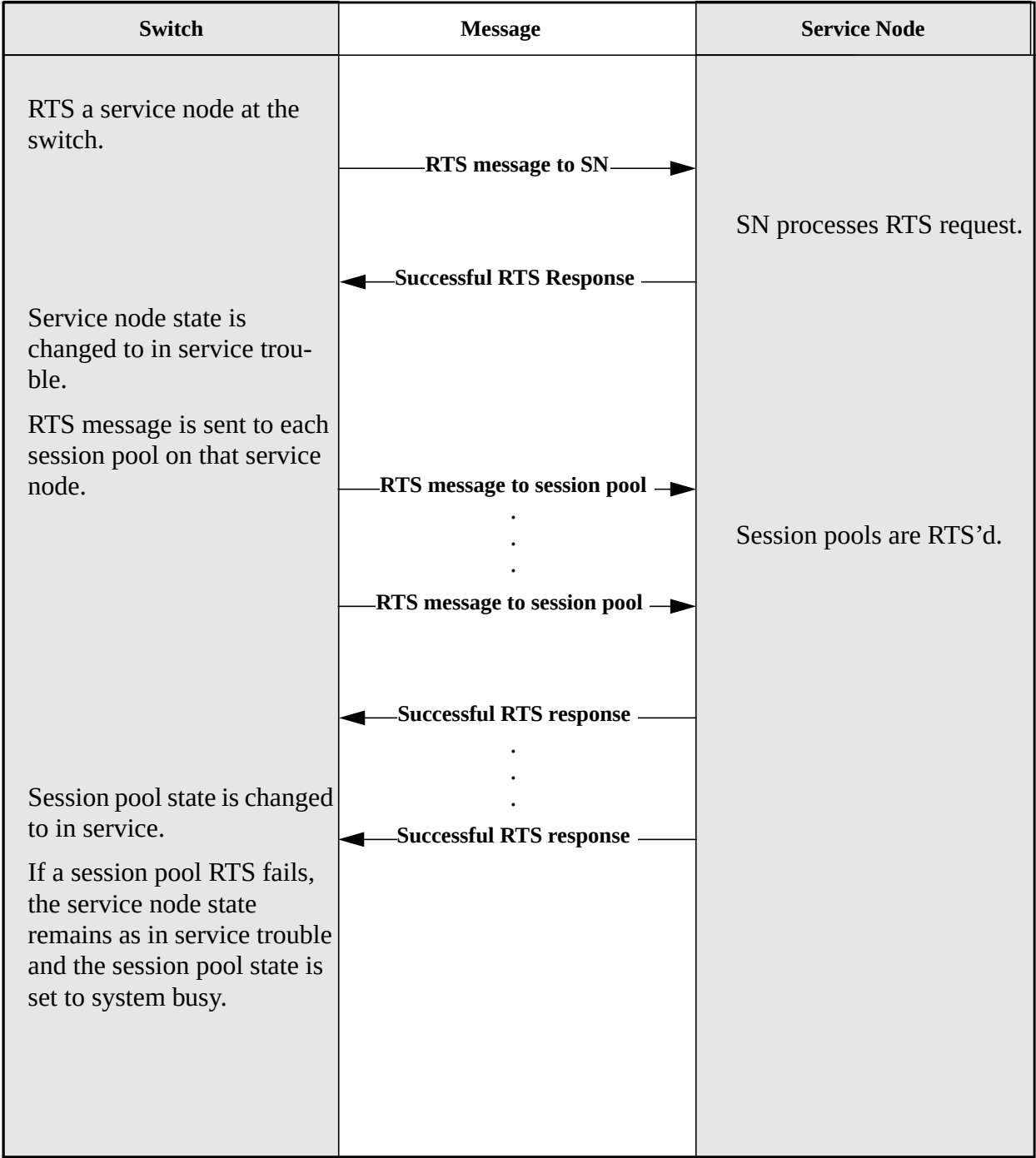
Figure 9. Busy Node Message Flow

Figure 10: “Return To Service Message Flow” shows an example message when a service node is manually returned to service at the switch.

Figure 10. Return To Service Message Flow



3.8 Service Node Considerations

Service node developers need to be aware of the following when developing services that will interact with OSSAIN.

- Operational Measurements (OMs)

Basic OSSAIN OMs are collected in the switch. However, the switch does not collect any OMs from the service node. The service node is responsible for collecting and reporting any OMs it requires.

- Node Level Audits

The switch sends an audit message to a service node if the switch has not received any OAP messages within some amount of time. There is no periodic audit message from the switch.

- Session Pool Level Audits

The switch sends an audit message to a session pool if the switch has not received any OAP messages with some amount of time on that session pool. There is no periodic audit message from the switch.

- Parallel Datafill

There is some OSSAIN datafill that is required to be identical in both the switch and service node. Refer to the field definitions for information on each item.

- Control List IDs
- Function IDs
- Languages
- Max number of Sessions per Session Pool ID
- Network Service IDs
- Node IDs
- Operator IDs
- Restricted Billing Indices
- Session Pool IDs
- Terminal IDs
- Treatments
- Trigger Profile Indices
- Trunk Group IDs
- Trunk Member IDs
- Voice Channel IDs

- Security

Encryption and decryption algorithms are in place to secure vital data (e.g. calling card numbers). Details are available in the *OAP Security Document*.

- Maintenance States

Service nodes should maintain logical node and session pool maintenance states with the switch. For example, if a service node is considered to be out of service by the switch, it should not send any calls to the switch. The service node must be returned to service at the switch before it starts initiating calls to the switch. Service node states and session pool states should attempt to follow what the switch considers the state to be.

Chapter 4: Message Format

4.1 Overview Of OAP Messaging

OAP provides a means for nodes (switch and service node) to share processing of operator service calls. This is accomplished, in part, by allowing capabilities to be distributed among a set of nodes according to their unique capabilities. OAP messages provide a means for one node to utilize the capabilities of other (i.e. remote) nodes, by requesting operations to be performed at that remote node. The OAP message set allows a node to offer its specific set of capabilities as operations to be requested in a predefined manner and with a well defined interface.

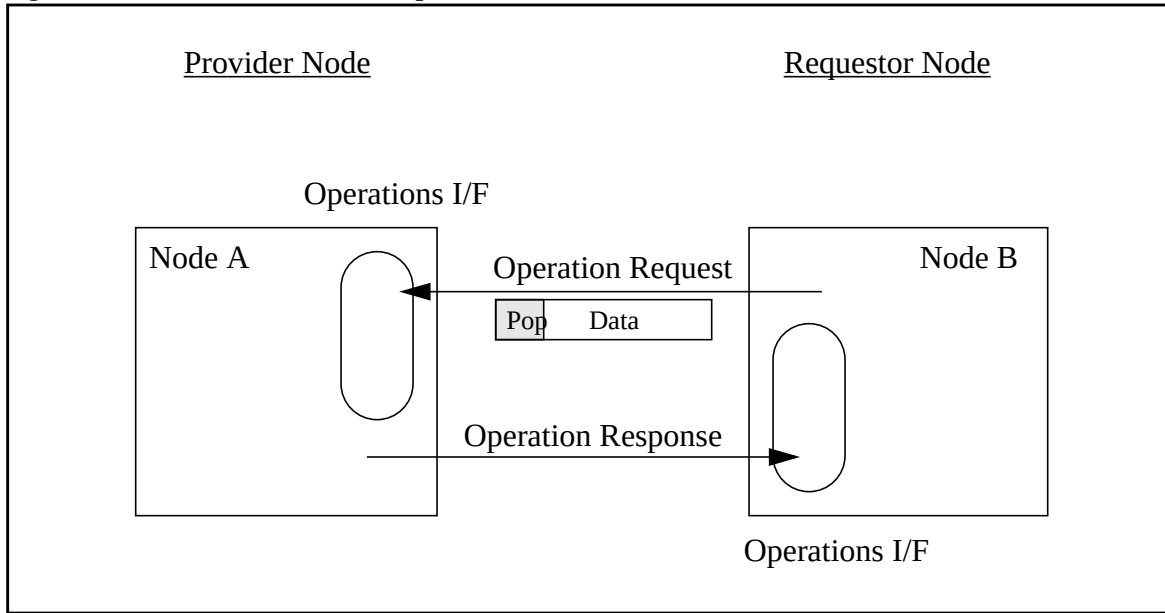
Note: "Remote", in the context of remote operations (or ROSE - Remote Operations Service Element), simply refers to communications with an external node, and in no way presumes the presence or absence of operator centralization or OSSAIN OSAC configurations. "Remote operation" is generally synonymous with "remote procedure", although, for our purposes, the procedure/operation is always one performed at an external node.

This chapter describes the basic format for OAP messages. The focus of OAP, and thus of the message format, is the exchange of application-level interactions (remote operations and their results) between OSSAIN nodes. This chapter discusses that framework for OAP operation exchange, as well as some of the support mechanisms. These discussions lay the groundwork for detailed layouts of the OAP class and operation headers.

4.1.1 OAP Remote Operations Interface

OAP provides remote access by external nodes to operations through a remote operations interface based on the International Standards Organization (ISO) Open Systems Interconnection (OSI) ROSE model for remote operations. ROSE defines a format and interaction framework for invoking and responding to remote operations. Through this model, a remote node may formulate a request for a specific operation and transmit it to a 'provider' node. The provider node then interprets the requested operation and its associated data, invokes the operation locally, gets a result (which may require interactions with other external entities before success or failure of the operation can be determined), and returns the result (or error) in a format understandable by the requestor node.

Figure 11. Invocation of a Remote Operation



4.1.1.1 ROSE Application Framework

The ROSE model defines a set of message types that provide this interaction capability:

- RO-Invoke (operation requests or informs)

Note: The abbreviation 'operation' is used where the context is sufficiently clear not to need qualifying as 'remote' or 'local'.

- RO-Result (success responses to operation requests)
- RO-Error (failure responses to operation requests)
- RO-Reject (message rejection due to malformed or invalid messages)

RO-Invokes, RO-Results, and RO-Errors provide the basic request/response functionality. RO-Rejects handle the issues of malformed messages and other protocol violations.

Whether a particular operation request (RO-Invoke) requires a response message (RO-Result or RO-Error) from the provider node is determined by the ROSE operation class of that particular operation:

- 1 - Synchronous operation: Reports success (result) or failure (error) response.
- 2 - Asynchronous operation: Reports success (result) or failure (error) response.
- 3- Asynchronous operation: Reports failure (error) response only.

- 4- Asynchronous operation: Reports success (result) response only.
- 5- Asynchronous operation: No success (result) or failure (error) responses reported.

Synchronous class here indicates that a requestor must await a response for a remote operation prior to invoking a subsequent operation. The asynchronous class removes this restriction. OAP utilizes operations classes 1 (primarily for restricted Request operations) and 2 (non-restricted operations), both of which require a response, and class 5, which requires no response (although rejects are valid). Rather than refer to ROSE operations classes by number, OAP uses a somewhat more intuitive labeling of class 1 and 2 operations as Requests and class 5 operations as Informs. OAP restrictions for particular operations are described with that operation. The combinations of invokes/responses can be shown as:

Table 5: Invokes/Responses

Message Type	Expected Response
Inform	none
	Reject
Request	Success Response
	Error Response
	Reject
Success Response	none
	Reject
Error Response	none
	Reject
Reject	none

In addition to agreeing to message types, ROSE requires that nodes participating in an interaction share some common protocol definitions (a purely hypothetical OAP example accompanies):

Note: For OAP, this document acts as the common repository for these definitions.

- A common set of operations, identified by their numeric Operation IDs.
(An example of a defined operation for OAP might be ‘Request Operator’. OAP operations are listed in “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations”.)

Note: For the illustrative purposes of this example, the operations, parameters, errors, etc. are fictitious, and thus do not intentionally represent any actual OAP operations, parameters, errors, etc., nor does their use infer any intent to implement similar functionality in a similar manner. Any similarities to actual OAP operations, parameters, errors, etc. (or their names) is coincidental and unintentional.

- The data that is to accompany each request for a given operation.

(A max queue time might be typical data to accompany an operation such as ‘Request Operator’. The data blocks associated with particular OAP operations are identified with the operation in “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations”, and described in detail in “Chapter 9: Data Block Descriptions” and “Chapter 10: Field Definitions”).

- Whether a particular operation should (or should not) return a success or error response to the requestor (i.e. assignment of an operation class to each operation).

(‘Request Operator’ might be expected to be a ROSE class 2 operation, i.e. an OAP Request, requiring a response from the operation provider. OAP operations are identified as Requests or Informs, indicating their requirements for responses. See “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations”).

- The data that is to accompany a success response to a given operation.

(Data that might accompany a success response to a ‘Request Operator’ operation might include the operator’s position number. The data accompanying a success response to an OAP operations is identified in “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations”, and described in detail in Chapters “Chapter 9: Data Block Descriptions” and “Chapter 10: Field Definitions”).

- A common set of error types, identified by their Error IDs.

(OAP error types defined might include an error such as ‘Estimate of queuing time exceeds maximum wait’. Error IDs are identified and described in “Chapter 11: Error ID Values”).

- The list of error types (i.e. Error IDs) valid in an error response to a given operation.

(The list of Error IDs for ‘Request Operator’ should include the identifier for ‘Estimate of queuing time exceeds maximum wait’. Associated with each operation in “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations” is the valid Error ID list, a subset of those listed in “Chapter 11: Error ID Values”, for the given operation.)

- The data that is to accompany a given error response.

(The actual wait time might accompany the ‘Estimate of queuing time exceeds maximum wait’ response. OAP data blocks are identified in “Chapter 11: Error ID Values”, on a per-Error ID basis, and described in detail in “Chapter 9: Data Block Descriptions” and “Chapter 10: Field Definitions”).

- Problem Types and Reject Reasons for RO-Reject messages. These are ROSE defined. A subset are listed below.

(The Operation ID for 'Request Operator' might have been improperly defined in either the switch or the SN. The receiving node should reply with a Problem Type of 'Invoke Problem' and a Reject Reason of 'Unrecognized Operation'.)

Figure 12. RO-Reject: Problem Types and Reject Reasons

Problem Type	
0 - General Problem	
1 - Invoke Problem	
2 - ReturnResult Problem	
3 - ReturnError Problem	
Reject Reasons	
<u>General Problem</u>	
0 - Unrecognized Application Protocol Data Unit (APDU)	
1 - Mistyped APDU	
2 - BadlyStructured APDU	
<u>Invoke Problem</u>	
0 - Duplicate Invocation	
1 - Unrecognized Operation	
2 - Mistyped Argument	
3 - Resource Limitation	
4 - Initiator Releasing	
5 - Unrecognized LinkID	
6 - Linked Response Unexpected	
7 - Unexpected Child Operation	
<u>ReturnResult Problem</u>	
0 - Unrecognized Invocation	
1 - ResultResponse Expected	
2 - Mistyped Result	
<u>ReturnError Problem</u>	
0 - Unrecognized Invocation	
1 - ErrorResponse Expected	
2 - Unrecognized Error	
3 - Unexpected Error	
4 - Mistyped Parameter	

Note: There are various conditions (such as a restricted operation in progress) that may result in the switch automatically returning an error to a new operation request. These conditions are operation specific and ReturnError rather than Reject is returned.

4.1.1.2 ROSE Formatting

ROSE defines a basic data formatting for operation requests and responses. This agreed upon formatting facilitates communication between greatly differing systems. This ROSE format identifies the data/parameters that accompany operations and responses, as well as the basic encoding of these parameters.

Note: Data parameter, as used here, denotes both simple data types (such as the Operation ID) and complex data structures (such as the Invoke Argument data).

The following list indicates the basic parameters (which include the operation identifier) for the various message types:

- RO-Invokes (OAP Requests and Informs)

Invoke ID (a ‘solicitor’ identification number, generated by an Invoke sender, and reflected in response messages (i.e. ReturnResult, ReturnError, and Reject), a primary use of which is to map particular response messages to specific operation request messages). The valid range for OAP Invoke ID values is #0100 - #FFFF.

Operation ID (the numerical Operation ID, as described above.)

Argument (the data associated with the Operation ID, as described above)

- RO-Results (OAP ReturnResult Responses)

Invoke ID (the solicitor identification number of the associated operation request message)

Result (the data associated with the Operation’s result, as described above)

- RO-Errors (OAP ReturnError Responses)

Invoke ID (the solicitor identification number of the associated operation request message)

Error ID (the numerical Error ID described above.)

Parameter (the data associated with the Error ID, as described above)

- RO-Rejects (OAP Rejects)

Invoke ID (the solicitor identification number of the associated operation request, returnresult, or returnerror message)

Problem Type (a ROSE predefined numerical value for error categories, such as InvokeProblem)

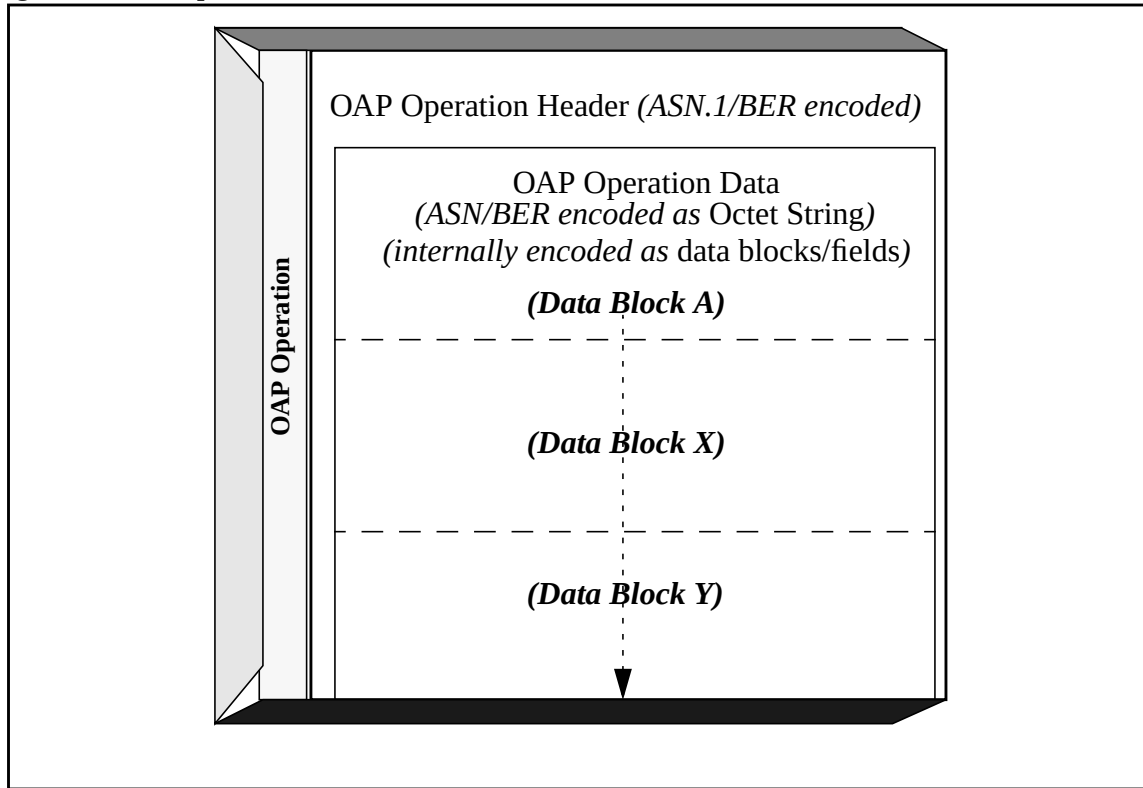
Reject Reason (a ROSE predefined numerical value for certain errors, such as UnrecognisedOperation)

Each data parameter listed is mandatory for a given message type, with the exception of the associated data parameters (i.e. Argument, Result, and Parameter fields), which are

‘optional’ (e.g. particular operations/results/errors may not require any associated data). The data parameters listed above must follow in the order mentioned for the particular message type. Additionally, ROSE specifies these parameters via ‘Abstract Syntax Notation 1’ (ASN.1). The only current standard for ASN.1 encoding is the Basic Encoding Rules (BER). With the exception of the selection of an internal encoding scheme for the associated data parameters, this defines a rather specific formatting for ROSE operation types (as is detailed in the following sections, with ASN/BER particulars covered in detail in Appendix A).

It is at this point that some OAP philosophy must enter. ASN/BER encoding is not particularly space-efficient, and although X.409/X.410 reflects a bias towards full ASN/BER encoding of all application data down to the primitive data type level, not all ROSE/ASN-based protocols actually take encoding to that level. Fortunately, the standards do indicate that proprietary encoding of the associated data parameters is acceptable, allowing protocol developers to encode portions of their data in this way. One such approach is mentioned in X.409, suggesting that OctetStrings might be used for this purpose. OAP takes full advantage of this, encoding the Argument, Result, and Parameter data as OctetStrings per ASN/BER, with a much more efficient data block and field level encoding of the actual internals.

OAP logically views the ROSE ASN.1/BER encoded fields as a somewhat distinct Operation Header, with the block/field encoded Argument/Result/Parameter data described separately as Operation Data. Thus, the general format of an OAP operation might best be depicted as:

Figure 13. OAP Operation Format

The Operation Header follows the ASN/BER encoding, but expands the Integer fields (Invoke ID, Operation ID, Error ID, and Reject Reason) to word length (use of “word” in OAP refers to a 16-bit value). This is accomplished for Invoke ID, Operation ID, and Error ID fields through defining the Integers as ASN Short Format Integers with a length of 2 Octets. This restricts values in these fields to #0100 hex to #FFFF hex, but has the value of eliminating variable width fields and ensuring that these word-length fields are word aligned. This allows compilers to overlay data structures over these fields, and thus provides a level of execution efficiency. Reject Reason is lengthened to word length as ASN Long Format Integers with a length of 2 Octets. Long Format allows a value range of #0000 hex to #FFFF hex.

Operation data blocks, when defined for specific operations, use a simple data block/field encoding, and are internally encoded as:

- Data Block ID (Unique ID identifying data block type)
- Data Block Byte Count (Length of subsequent data in data block)
- Field A
- Field B
- ...

Data block and field formats are detailed in “Chapter 9: Data Block Descriptions” and “Chapter 10: Field Definitions”.

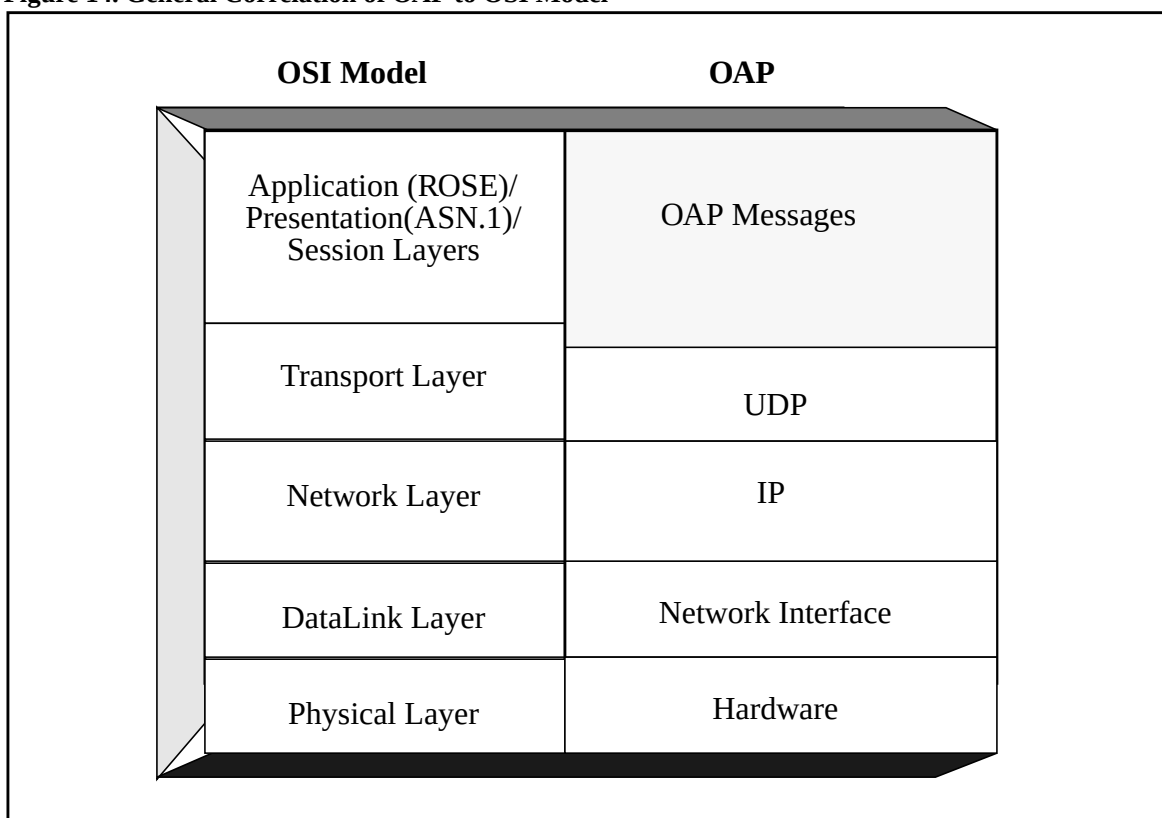
Data blocks are extended with a filler byte, if necessary, to word-align them. Single byte data fields are paired with each other (or a filler byte) similarly. Multiple byte fields begin on a word boundary. The data blocks are formatted in this manner for the same efficiency reasons as per the word alignment for Operation Header fields. Note that data blocks are sent in an arbitrary order in the message.

4.1.2 The OAP Message Layer

In the hierarchy of protocol layerings, the prior overview sections discussed OAP's operations, which sit at the application (OSI layer 7) layer of OAP, as well as the encoding, or presentation (OSI layer 6) layer. As primarily an application layer protocol, it is not the intent of OAP to dictate or describe a particular transport mechanism (OSI layers 1 - 4), nor is it desirable that OAP be tied to a particular mechanism, however, selection of a particular transport can impact the higher protocol layers in minor ways.

OAP, for availability and efficiency reasons, has chosen UDP/IP as its initial basic transport mechanism. A very general correlation of OAP with the ISO Open Systems Interconnection (OSI) model follows (using UDP/IP as a presumed Transport base):

Figure 14. General Correlation of OAP to OSI Model



UDP/IP is an industry standard transport layer protocol, and while this document makes no attempt to describe UDP/IP, it is important to note that UDP/IP does not guarantee reliable delivery, therefore the OAP messaging must incorporate methods for handling lost and out of sequence messages. (See Chapter 3 for details)

4.1.3 OAP Class Headers

The OAP Class Header is prepended to OAP Operations to provide OAP with several valuable functions:

- It allows messages to be routed to specialized handlers.
- It allows for protocol evolution through versioning.
- It allows some basic lost message handling to be incorporated.
- It allows application specific fields to be added.

4.1.3.1 Message Classes

Each OAP message type belongs to an OAP message class. The class of an OAP message determines where the OAP operations interface should route the message. This allows message routing to a specialized process (e.g. allows separation of maintenance, and other specialized messaging, to be processed apart from the more application-specific call processing). Message classes define (see “Chapter 7: Call Processing Operations” and “Chapter 8: Non-Call Processing Operations”) which operations they can process. (**As a result of this class-dependent routing, only operations defined for a given message class should be routed there.**) The first field in the OAP Class Header identifies its message class. The defined classes are:

- Call Processing Class
- MIS Class
- Node Class
- Node Alarm Class
- Node Log Class
- Session Pool Class
- Session Pool Alarm Class
- Session Pool Log Class

Each class header has a unique identifier and a specific header format. All class headers have a common set of fields followed by some class-specific fields.

4.1.3.2 Protocol Version

The Protocol Version, in addition to allowing transport mechanisms to evolve, allows both operations and message class headers to evolve. General evolution shall be through the addition and changes of fields, data blocks, or operations. Protocol Versioning is described in detail in “3.4 Protocol Versioning”. The Protocol Version fields are common across the various Class Headers.

4.1.3.3 Lost Message Handling

As previously noted, UDP/IP does not guarantee reliable message delivery. This requires some lost message handling capability within the upper layers. For the Call Processing class of messages, OAP implements a very basic approach utilizing sequence numbers.

Sequence number, and the associated issue of lost message handling, is described in detail in Section 3.5 “Sequence Numbers” and Section Figure 7. “Sequence Number Example”. Sequence numbers are common across the various Class Headers.

4.1.3.4 The OAP Class Header Fields

As noted, most of the OAP Class Header fields (for a UDP/IP transport mechanism) are common to all OAP classes. These common fields are included regardless of the protocol version. The common fields are:

- OAP Class Header ID - This field indicates the class of the message (e.g. Node, Session Pool, Call Processing), and thus the handler to which the message should be passed.
- Options - This field allows special aspects of the message to be specified. Currently, only bit 0 has been defined. Bit 0 of this special-purpose bit-field supports protocol version negotiation. The remaining bits in this field are reserved and should be set to 0. Bits 13 to 15 are reserved for service node use. See Section 3.4 “Protocol Versioning” for details on the use of bit 0 of this field.
- Length of Following Message - This field is here as both a sanity check and as an indicator of multiple operations. Processing should never exceed this length. This length, like the others used in OAP, is stated as the length of data following (e.g. the value 2 indicates 2 more bytes follow this length field). The value for this field should be the length of the OAP message minus the first six bytes of the class header.
- Offset to Operation Data - This is the offset from the OAP Class Header ID to the operation data (i.e. the Operation/Response Type). This value is equal to the length of the class header.
- Message Protocol Version - Identifies the protocol release/increment of this message. This indicates to the receiver the protocol release/increment interpretation that is required to decode this message. See Section 3.4 “Protocol Versioning” for details.
- Sender’s Max Protocol Version - Identifies the sender’s maximum protocol release/increment. This may be higher than the message’s version if the sender had to drop back to an older version to communicate with the receiver. See Section 3.4 “Protocol Versioning” for details on version handling.
- Source Node ID - A network wide OSSAIN node identifier. This requires parallel datafill between switch and SNs. This field ensures that communications are valid and proper. The switch will not acknowledge SNs with invalid SN IDs.

The following are other common fields and may change between protocol versions.

- Sequence Number - This field assists in identifying and handling lost messages. See Section 3.5 “Sequence Numbers” on page 100 for details.

The Node Class, Node Alarm Class, Node Log Class, Session Pool Class, Session Pool Alarm Class, and Session Pool Log Class headers have the following additional field:

- Solicitor Number - This field identifies the task associated with the operation.

The Session Pool Class, Session Pool Alarm Class, and Session Pool Log Class headers have the following additional field:

- Session Pool ID - The Session Pool ID field indicates the particular SN session pool this message regards. This is used for session pool maintenance.

The Call Processing Class Header has the following additional fields. These fields require parallel datafill between switch and SN to ensure reliable communications.

- Function - The Function ID identifies the function being provided by the service node. The function ID for subscriber originated calls is determined by the switch. The service node should echo the function id sent by the switch. For service node initiated calls, the function id is determined by the service node. The function id should be within the supported range for function ids (including the nil function id).
- Session Pool ID - The Session Pool ID is the pool of SN resources that support one or more functions. This field indicates which session pool is being selected.
- Session ID - This field is a specific session from the session pool.
- Call ID - This field identifies the specific call this message is intended for.

All class headers use word size fields for efficiency.

Note: The Class Headers are **not** ASN/BER encoded, as their functionality is predominantly routing/session-oriented.

4.2 OAP Message Layouts

The following sections detail the field layout of OAP messages. Following a brief description of the general rules of usage for certain OAP numbering schemes, layouts for the major subsections of the OAP message format are shown. They are listed in message order by:

The OAP Class Header variants:

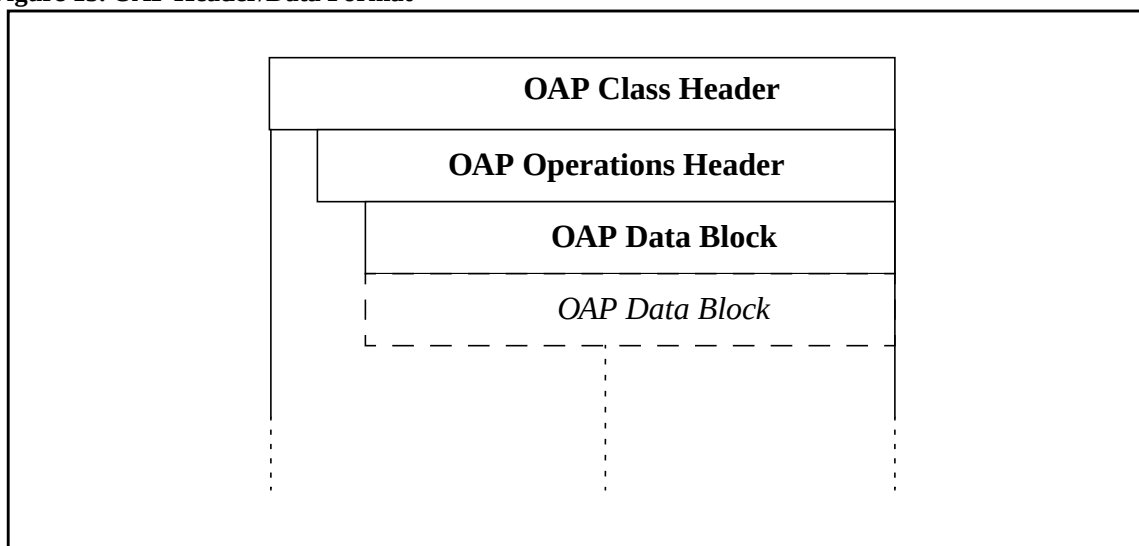
- Call Processing Class Header
- MIS Class Header
- Node Class Header
- Node Alarm Class Header
- Node Log Class Header
- Session Pool Class Header
- Session Pool Alarm Class Header
- Session Pool Log Class Header

The OAP Operation Header variants:

- Invoke Operation Header
- ReturnResult Operation Header
- ReturnError Operation Header
- Reject Operation Header

And the general format for OAP data blocks.

Figure 15. OAP Header/Data Format



4.2.1 OAP Rules

- All IDs (Operation, Data Block, Field, and Error) from #E000 through #FFFF, inclusive are reserved for external use (e.g. SN to SN messaging). These are not processed by the DMS.
- All IDs not specifically defined are reserved.
- Data Block IDs are sequentially allocated from #0000 through #DFFF, inclusive. These are unique system wide values.
- Operation IDs and Error IDs are allocated from #0100 through #DFFF, inclusive. These values are unique system wide values.

4.2.2 OAP Class Headers

Class headers are defined for each of the predefined classes for OAP.

- Call Processing
- MIS
- Node
- Node Alarm
- Node Log
- Session Pool
- Session Pool Alarm
- Session Pool Log

The class header contains general information regarding the overall length of the message, the OAP class protocol version, transport specific information, and class specific information. The detail of this addressing varies depending on the needs of the class.

The OAP Class Headers consist of three fairly distinct sections:

Table 6: Class Header Sections

Class Header Section		Class Header Fields
Common	Version Independent	Class Header ID
		Options
		Length of following message
		Offset (from Class Header ID to Operation Data)
		Message Protocol Version
		Sender's Max Protocol Version
		Source Node ID

Table 6: Class Header Sections

Class Header Section		Class Header Fields
Common	Version Dependent	Sequence Number
Class Specific	Version Dependent	<Class Specific>
		...
		<Class Specific>

As previously described, some fields are common to all of the OAP Class headers, while some fields are class-specific. In the layout of OAP Class headers, the common fields precede the class-specific fields. This is shown above in Table 6, “Class Header Sections,” on page 132.

Message Protocol and Sender’s Max Protocol Version fields are arranged with protocol release in the Most Significant Byte (first byte), and protocol increment in the Least Significant Byte (last byte).

Additionally, for the purpose of handling protocol version upgrades, it appears necessary that the format of the first several fields of the OAP message need to be fixed (version independent). This allows nodes running different protocol versions to negotiate to a common version even if some of the header fields have changed in the interim. This is discussed in detail in Section 3.5 “Protocol Versioning”.

4.2.2.1 Node Class Header

This header will appear as the Node Class Header, regardless of direction (SN->switch or switch->SN).

1	#0001	OAP Node Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Solicitor Number
18		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).

4.2.2.2 Node Alarm Class Header

This header will appear as the Node Class Header, regardless of direction (SN->switch or switch->SN).

1	#0006	OAP Node Alarm Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Solicitor Number
18		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).

4.2.2.3 Node Log Class Header

This header will appear as the Node Class Header, regardless of direction (SN->switch or switch->SN).

1	#0002	OAP Node Log Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Solicitor Number
18		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).

4.2.2.4 Session Pool Class Header

This header will appear as the Session Pool Class Header, regardless of the direction (SN->switch or switch->SN).

1	#0004	OAP Session Pool Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Session Pool ID
18		
19	<#nnnn>	Solicitor Number
20		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).
- The range of Solicitor Numbers for service node originated requests/informs is limited to 256-511.

4.2.2.5 Session Pool Alarm Class Header

This header will appear as the Session Pool Class Header, regardless of the direction (SN->switch or switch->SN).

1	#0007	OAP Session Pool Alarm Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Session Pool ID
18		
19	<#nnnn>	Solicitor Number
20		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).
- The range of Solicitor Numbers for service node originated requests/informs is limited to 256-511.

4.2.2.6 Session Pool Log Class Header

This header will appear as the Session Pool Class Header, regardless of the direction (SN->switch or switch->SN).

1	#0003	OAP Session Pool Log Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Session Pool ID
18		
19	<#nnnn>	Solicitor Number
20		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).
- The range of Solicitor Numbers for service node originated requests/informs is limited to 256-511.

4.2.2.7 Call Processing Class Header

This header will appear as the Call Processing Class Header, regardless of the direction (SN->switch or switch->SN).

1	#0005	OAP Call Processing Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Function ID
18		
19	<#nnnn>	Session Pool ID
20		
21	<#nnnn>	Session ID
22		
23	<#nnnnnnnn>	Call ID
24		
25		
26		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).
- For switch to SN directed messages, the Sequence Number reflects the switch's message ordering.
- For SN to switch directed messages, the Sequence Number reflects the SN message ordering.
- Sequence Numbers are computed in a modulo(#10000) fashion, #0000-#FFFF.
- Sequence numbers are used by the receiving node to determine out-of-sequence or lost messages.

4.2.2.8 MIS Class Header

This header will appear as the MIS Class Header, regardless of the direction (SN->switch or switch->SN).

1	#0008	OAP MIS Class Header ID
2		
3	<#nnnn>	Options
4		
5	<#nnnn>	Length of following Message
6		
7	<#nnnn>	Offset (from Class Header ID to Operation Data)
8		
9	<#nnnn>	Message Protocol Version
10		
11	<#nnnn>	Sender's Max Protocol Version
12		
13	<#nnnn>	Source Node ID
14		
15	<#nnnn>	Sequence Number
16		
17	<#nnnn>	Network ID
18		

- The value for the Length of following Message field is the length of the OAP message minus the first six bytes of the class header.
- Message and Sender's Max Protocol Version fields both consist of two one-byte subfields (Release and Increment).
- Message Protocol and Version and Sender's Max Protocol Version are used to perform OAP version negotiation following the same method used by other OSSAIN nodes.

4.2.3 OAP Operation Headers

Operation headers are ROSE/ASN.1 encoded and include information regarding the operation (or response) being requested (or responded to). Additional information in the Operation header include an Invoke ID to match operation responses to requests, an Operation ID for Invoke-type operations (Requests and Informs), and an Error ID for ReturnError responses. The Operation IDs and Error IDs are key for indicating what data follows. Rejects do not include additional data.

Notes:

- Operation Headers are ASN/ROSE encoded. (See “Appendix A: ASN.1 Encoding of the OAP Operation Header” for additional information.)
- These headers are always word-aligned.
- Operation Header IDs are defined by ASN/ROSE.
- Invoke ID values range from #0100 through #FFFF. Use of the range #0000 through #00FF is reserved/excluded to remain within ASN.1 formatting. (See “Appendix A: ASN.1 Encoding of the OAP Operation Header”).
- Invoke IDs are chosen by an Invoke (Request or Inform) sender so that responses (Return-Results, ReturnErrors, and Rejects) can be matched to the original Invoke. The sender may choose any value. Note that the switch sends #FFFF for the Invoke ID when it needs to send a Reject and it was unable to determine the Invoke ID of the incoming Request or Inform.
- Operation ID values are unique system wide. They are divided between Call Processing and non-Call Processing classes. This implies Operation IDs defined as Node Class or Session Pool Class should not be sent to a Call Processing Class handler (i.e. sent with a Call Processing Class Header).
- Operation ID values from #0000 through #00FF are reserved and not used.
- All Operation ID values from #0100 through #DFFF inclusive are reserved by NORTEL.
- Operation ID values from #E000 through #FFFF, inclusive are reserved for external use (e.g. SN to SN communications). These Operation IDs are NOT processed by the switch, and should not be sent to the switch.

Total length of the operation data, not including the Class Header: summation (operation header + associated data blocks)

- OAP messages Incoming to the switch should not exceed 288 bytes: ≤ 288 bytes
- OAP messages Outgoing from switch should not exceed 1500: ≤ 1500 bytes for UDP or 4000: ≤ 4000 bytes for TCP.

4.2.3.1 Invoke Operation Header

This header will appear as the Invoke Operation Header, regardless of the direction (SN->switch or switch->SN).

1	#A1	Operation/Response Type (Invoke)
2	#82	Operation/Response Length Format (Long, 2 byte Length)
3	<#nnnn>	Operation/Response Length (remaining header plus argument)
4		
5	#02	Invoke ID Type (Integer)
6	#02	Invoke ID Length Format (Short, Length = 2)
7	<#nnnn>	Invoke ID Value (#0100 - #FFFF)
8		
9	#02	Operation ID Type (Integer)
10	#02	Operation ID Length Format (Short, Length = 2)
11	<#nnnn>	Operation ID Value (#0100 - #DFFF)
12		
13	#04	Argument Type (OctetString)
14	#82	Argument Length Format (Long, 2 byte Length)
15	<#nnnn>	Argument Length
16		

- Operation IDs are unique.
- The Argument type is of Octet String. (i.e. The Data Blocks and Fields making up the Argument are not ASN.1 encoded.)
- This header is always word-aligned.
- Session Pool Class, Session Pool Alarm Class, Session Pool Log Class: the range of invoke ids for service node originated requests/informs is #0100 to #01FF.

4.2.3.2 Return Result Operation Header

This header will appear as the Return Result Operation Header, regardless of the direction (SN->switch or switch->SN).

1	#A2	Operation/Response Type (Return Result)
2	#82	Operation/Response Length Format (Long, 2 byte Length)
3	<#nnnn>	Operation/Response Length (remaining header plus argument)
4		
5	#02	Invoke ID Type (Integer)
6	#02	Invoke ID Length Format (Short, Length = 2)
7	<#nnnn>	Invoke ID Value (reflected from Invoke)
8		
9	#04	Result Type (OctetString)
10	#82	Result Length Format (Long, 2 byte Length)
11	<#nnnn>	Result Length
12		

- The Result type is of Octet String. (i.e. The Data Blocks and Fields making up the Result Data are not themselves ASN.1 encoded.)
- This header is always word-aligned.

4.2.3.3 Return Error Operation Header

This header will appear as the Return Error Operation Header, regardless of the direction (SN->switch or switch->SN).

1	#A3	Operation/Response Type (Return Error)
2	#82	Operation/Response Length Format (Long, 2 byte length)
3	<#nnnn>	Operation/Response Length (remaining header plus parameter)
4		
5	#02	Invoke ID Type (Integer)
6	#02	Invoke ID Length Format (Short, Length = 2)
7	<#nnnn>	Invoke ID Value (reflected from Invoke)
8		
9	#02	Error ID Type (Integer)
10	#02	Error ID Length Format (Short, Length = 2)
11	<#nnnn>	Error ID Value (#0100-#FFFF)
12		
13	#04	Parameter Type (OctetString)
14	#82	Parameter Length Format (Long, 2 byte Length)
15	<#nnnn>	Parameter Length
16		

- The Parameter type is of Octet String. (i.e. The Data Blocks and Fields making up the Parameter are not themselves ASN.1 encoded.)
- This header is always word-aligned.
- The Error IDs are sequentially numbered from #0100 and are system wide.

4.2.3.4 Reject Operation Header

This header will appear as the Reject Operation Header, regardless of the direction (SN->switch or switch->SN).

The format of the Reject Operation Header is to be Version Independent.

1	#A4	Operation/Response Type (Reject)
2	#0C	Operation/Response Length Format (Short, Length = 12)
3	#30	Sequence (structured) Type
4	#0A	Sequence Length Format (Short, Length = 10)
5	#02	Invoke ID Type (Integer)
6	#02	Invoke ID Length Format (Short, Length = 2)
7	<#nnnn>	Invoke ID Value (reflected from Operation/Result)
8		
9	#8n	Problem Type (IMPLICIT Integer, n = value)
10	#82	Reject Reason Length Format (Long, 2 byte Length)
11	#0002	Reject Reason Length (Length = 2)
12		
13	<#nnnn>	Reject Reason Value
14		

- The 'n' in '#8n' is the Problem Type. The '8' in '#8n' identifies the Type as Context-Specific.
- The 'nn' in '<#nnnn>' contains the RejectReason value.
- This header is always word-aligned.
- Problem Types and Reject Reasons are ROSE defined. See Section 4.1.1.1 "ROSE Application Framework" for defined values.
- The switch sends #FFFF for the Invoke ID if it was unable to determine the Invoke ID of the incoming Request or Inform.

4.2.4 Data Block

1	<#nnnn>	Data Block ID
2		
3	<#nnnn>	Data Block Byte Count
4		
5		Field 1
...		...
N		

Notes:

- Data Block IDs are defined from #0000 through #DFFF inclusive and are unique across the system.
- Data Block IDs from #E000 through #FFFF are reserved for external use. These are not processed by the DMS.
- The Data Block contents are encoded within the Octet String denoted in the Operation Header (i.e. the Data Blocks themselves are not ASN.1 encoded)
- The data block header (Data block ID and Data Block Byte Count) is always word-aligned.
- Data blocks are always word aligned.
- Data blocks are sent in an arbitrary order.

4.3 Transport Protocol Switch Information

The switch's IP address must be datafilled in Table IPNETWRK before any IP based transport protocols (e.g. UDP, TCP) can be used.

4.3.1 UDP

OSSAIN uses UDP as its primary transport protocol.

The port numbers used are based on the message Class Header:

- Call Processing Class: 5289
- Node Class: 5290
- Node Log Class: 5292
- Node Alarm Class: 5293
- Session Pool Class: 5291
- Session Pool Log Class: 5292
- Session Pool Alarm Class: 5293

4.3.2 TCP

Operations using the MIS class header are sent using TCP as their transport protocol. See section "4.2.2.8 MIS Class Header" for information about the class header.

The port number used on the switch's side of the TCP connection is determined dynamically, while the node's port number is datafilled in the OSSAIN tuple in Table QMSMIS.

Node maintenance messaging for a MIS node still uses UDP as the transport protocol, using the switch's Node Class UDP port defined above.

Currently, the only OSSAIN nodes to use TCP are the MIS nodes. MIS nodes do not support session pools, but a future TCP based application is expected to use UDP for session pool maintenance, if applicable.

Chapter 5: Data Format

5.1 Data Format

Each operation is composed of zero or more data blocks. The data blocks are comprised of data fields. Each field is defined to be of certain format and length. The fields are specified to be one of the following types:

- ASCII
- BCD (Binary Coded Decimal)
- Binary
- Hexadecimal
- Set

Numeric values will be specified in a decimal value and/or a hexadecimal value. Hexadecimal values will be preceded by an octothorpe (#).

The valid range of values will be specified for each field. A protocol violation occurs if a field value is out of range.

5.1.1 Binary Fields

A binary field allows a single value out of a range of values to be specified for a particular parameter. It may be interpreted as a numeric value or mapped into an enumerated type where each value corresponds to a particular condition or action.

For each byte value, bit 0 is the least significant bit and bit 7 is the most significant bit.

If a binary field spans more than one byte, the first byte is the most significant byte (i.e. the byte earlier in the message, closer to the header) and the last byte is the least significant byte (i.e. later in the message, further from the header).

In the following example, Field A is an enumerated type, one byte long. The value for the field is four (0000 0100).

Field A:

0 = blue

1 = green

2 = red

3 = yellow

4 = purple

5 = pink

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

In the following example, Field B is a numeric type, two bytes long. The actual value is Field B = 3250 (#0C B2).

Field B: 0 to 65534	Hex #	Binary								Byte #
		7	6	5	4	3	2	1	0	
	0C	0	0	0	0	1	1	0	0	0
	B2	1	0	1	1	0	0	1	0	1

5.1.2 BCD Fields

A BCD digit field is a variable sized field allowing data comprised of the digits 0 to 9 (e.g. directory number, carrier number) to be transmitted without conversion. Each BCD digit requires a nibble (i.e. 4 bits) of store. Therefore, each byte in a BCD digit field can hold up to two digits.

A BCD digit stream is stored from most significant byte (i.e. byte 0) to least significant byte (i.e. byte n). Within each byte, the first digit is stored in nibble 1 and the second digit is stored in nibble 0. For example, the digit string '123456' would be stored in the following order:

Nibble		Byte #
1	0	
1	2	0
3	4	1
5	6	2

If a BCD data field does not require as many BCD digits as the field provides for, then the unused nibbles of the field are considered undefined. This is accomplished through preceding the BCD data field with a count field to indicate how many BCD digits are in the following BCD field. A nil value of #F should be placed in the unused nibbles.

All variable sized fields have a count field, which contains the number of digits in the BCD data field. The nil values should not be included in the count value for the data field. The digit string '98761' (count value = 5) would be stored in the following order.

<u>Nibble</u>		Byte #
1	0	
9	8	0
7	6	1
1	F	2

5.1.3 Hexadecimal Digit Fields

A 'hex' digit field is a variable sized field allowing data comprised of the digits 0 to F(e.g. directory number, carrier number, DTMF digits) to be transmitted without conversion. Each hex digit requires a nibble (i.e. 4 bits) of store. Therefore, each byte in a hex digit field can hold up to two digits.

A hex digit stream is stored from most significant byte (i.e. byte 0) to least significant byte (i.e. byte n). Within each byte, the first digit is stored in nibble 1 and the second digit is stored in nibble 0. For example, the digit string '123DEF' would be stored in the following order:

<u>Nibble</u>		Byte #
1	0	
1	2	0
3	D	1
E	F	2

If a hex data field does not require as many hex digits as the field provides for, then the unused nibbles of the field are considered undefined. This is accomplished through preceding the hex data field with a count field to indicate how many hex digits are in the following hex field. For the above example, the preceding count field should equal 6. Although it is also within the valid digit range for this field type, #F should be placed in unused nibbles. If the count had been 5, then this example would indicate the digit string '123DE', and the 'F' would just be a filler digit. Filler digits should not be included in the count.

5.1.4 Set Fields

A set field allows multiple values to be simultaneously specified for a particular parameter. Each bit in the field corresponds to an element in a set of conditions or actions. If a bit is set to 1, then the set element associated with that bit is present in the field.

In the following example, Field C is a set field, one byte long. Bits 0, 2, 3, & 6 are true.

Field C:

Bit 0 = apple

Bit 1 = orange

Bit 2 = banana

Bit 3 = pear

Bit 4 = grapes

Bit 5 = tomato

Bit 6 = pineapple

7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	1

5.1.5 ASCII Fields

An ASCII field allows textual information to be passed between the switch and a service node. Each ASCII character requires one byte in a message.

A character count field is associated with each ASCII field. Unused bytes of the field are considered undefined. A nil value of #FF should be placed in the unused bytes. However, the nil values should not be included in the count value.

The first character (i.e. the left most character) is stored in the first byte (i.e. the most significant byte) and the last character (i.e. the right most character) is stored in the last byte (i.e. the least significant byte).

When an ASCII field is described as 'alphanumeric', only ASCII characters in the ranges {A-Z}, {a-z}, and {0-9} may be placed in the field. The definition of each ASCII field indicates if the field has any restrictions on what ASCII characters it supports.

The following table illustrates the ASCII character set. Bits 4-7 make up the most significant nibble and bits 0-3 make up the least significant nibble.

Table 7. ASCII Character Set

Bits 4 - 7								Bits 0 - 3
#0	#1	#2	#3	#4	#5	#6	#7	
null	datalink escape	space	0	@	P	' back quote	p	#0
start of heading	device control 1	!	1	A	Q	a	q	#1
start of text	device control 2	“	2	B	R	b	r	#2
end of text	device control 3	#	3	C	S	c	s	#3
end of transmission	device control 4	\$	4	D	T	d	t	#4
enquiry	negative ack	%	5	E	U	e	u	#5
acknowledgment	synchronous/idle	&	6	F	V	f	v	#6
bell or attention signal	end of transmitted block	' (apos- tro- phe)	7	G	W	g	w	#7
back space	cancel (error in data)	(	8	H	X	h	x	#8
horizontal tabulation	end of medium	)	9	I	Y	i	y	#9
line feed	start of special sequence	*	:	J	Z	j	z	#A
vertical tabulation	escape	+	;	K	[	k	{	#B
form feed	file separator	, (com ma)	<	L	\	l		#C
carriage return	group separator	-	=	M	]	m	}	#D
shift out	record separator	.	>	N	up arrow	n	~	#E
shift in	unit separator	/	?	O	_	o	delete	#F

The following example illustrates how the table is used to obtain the binary value of ASCII character 'G'. Bits 4-7 has the hexadecimal value 4 (i.e. 0100). Bits 0-3 has the hexadecimal value 7 (i.e. 0111). Therefore, the ASCII character G is #47.

7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	1

The value 'OPR 520' (count = 7) is stored as follows:

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
O	4F	0	1	0	0	1	1	1	1	0
P	50	0	1	0	1	0	0	0	0	1
R	52	0	1	0	1	0	0	1	0	2
space	20	0	0	1	0	0	0	0	0	3
5	35	0	0	1	1	0	1	0	1	4
2	32	0	0	1	1	0	0	1	0	5
0	30	0	0	1	1	0	0	0	0	6

5.2 Data Block Word Alignment

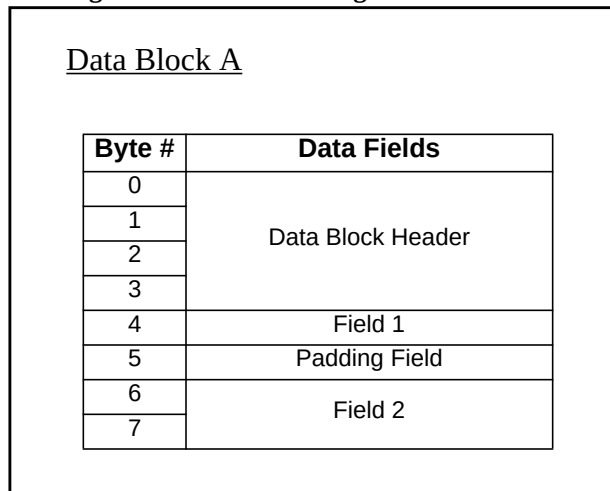
All data blocks must be word aligned according to the following requirements:

1. Even number of bytes in length, so that each data block starts on a word boundary
2. Each multiple byte field within a data block must begin and end on a word boundary

Padding fields (i.e. a filler) may be required to ensure that data blocks meet these requirements.

The following figure illustrates when padding is required to word align a data block which has two fields: Field 1 is one byte and Field 2 is 2 bytes.

Figure 16. Data Block Word Alignment with Fixed Length Fields



A padding field is defined after Field 1 so that Field 2 starts on a word boundary (requirement 2). The padding field is filled in with a nil value (i.e. #FF). The padding field also causes the data block to have an even number of bytes (requirement 1). Since the data block meets both requirements, it is word aligned.

The padding field is not part of Field 1. Field 1 is still defined to be one byte long. If a new one byte field needs to be added to data block A in a future OAP version, it can replace the padding field.

Variable length fields are handled a little differently. Fields in which the size of the value varies have an associated count field. The count field indicates the number of digits or characters in the variable field. Although the valid data in the field is variable in length, the field size is fixed. Unused bytes should contain nil values. There is a maximum length defined for each variable field.

In the following figure, the data block contains a count field and an ASCII variable field. In this case, a padding field is necessary so that the ASCII field begins on a word boundary (requirement 2). The maximum length of the ASCII field is declared to be four characters. The total length of the data block is ten bytes (requirement 1). Since the data block meets both requirements, it is word aligned.

Figure 17. Data Block Word Alignment with Variable Field Length Fields

<u>Data Block B</u>	
Byte #	Data Fields
0	Data Block Header
1	
2	
3	
4	Count Field
5	Padding Field
6	Variable Field
7	
8	
9	

In the above example, the count field is defined to precede the padding field. It would have been equally valid to have defined the data block with the padding field preceding the count field.

Chapter 6: OAP Error Handling

6.1 OAP General Error and Exception Handling

The following table, “DMS OAP Incoming Message Error Handling” on page 163, indicates how the DMS handles different types of OAP protocol errors and exceptions (e.g. protocol mismatches), associated with the general parsing of incoming OAP messages (Errors associated with the outcomes of specific OAP operations are covered under those specific operations. See the individual operation within the chapter “Call Processing Operations” on page 173.). This table covers errors and exceptions in processing the various fields of the OAP Class and Operation Headers, as well as generic errors in the processing of data blocks. As the first error encountered usually determines the resulting error response, the fields are listed within the table in general order of occurrence within OAP messages. For each OAP field, the table indicates:

- **Format:** The size and format of the field.
- **Valid Range:** The range of valid values for the field.
- **Logs for Invalid:** Whether a DMS log will be generated for a given error/exception.
- **OMs for Invalid:** Whether a DMS OM will be pegged for a given error/exception.
- **Failed Message Peg:** Whether a protocol error will be pegged by the DMS. The DMS periodically calculates the percentage of messages from a given service node that fail due to a protocol violation. If a certain percentage is exceeded, the service node will be taken out of service. The percentage is datafilled against each node in switch Table OANODINV.
- **Process Operations:** Whether the operations within the message will be processed (if not, they will, be discarded).
- **DMS Response:** The type of protocol response, if any, that will be returned by the DMS in response to the error/exception. A response or reject is not returned unless the message can be parsed far enough to determine the Invoke ID of the message.

Some OAP fields, such as Operation/Response Type, can have more than one error or exception type. Bullets are used within the field’s table row to differentiate the results for individual errors or exceptions.

Footnotes (which follow the table) are referenced when extended descriptions are needed.

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Class Header ID	16 bit	#0001 - #0007	Yes	Yes	Yes	No	None
Options (See footnote ^a)	16 x 1 bit	nnnn nnnn nnnn nnn0 nnnn nnnn nnnn nnn1	No	No	N/A	N/A	N/A
Length of Following Message	16 bit	Match to actual Range approx.: 16 - 294	Yes	Yes	Yes	No	None
Offset to Operations Data	16 bit	#0012: Node, Node Alarm, Node Log #0014: SP, SP Alarm, SP Log #001A: CallP	Yes	Yes	Yes	No	None

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Message Protocol Version	8 bits - Rel 8 bits - Inc	(See footnote ^b)	Yes	Yes	No	No	PVE Reject for non-optimum version or invalid. General-Problem, Badly-Structured APDU
Sender's Max Protocol Version	8 bit 8 bits	(See footnote ^c)	Yes	Yes	No	No	None
Source Node ID	16 bits	Table OAN-ODINV	Yes	Yes	Yes	No	None
Sequence Number	16 bits	(See footnote ^d)	Yes	Yes	Yes	For lower: No. For higher: Yes.	None.
Function ID	16 bits	Table OAF-UNDEF	Yes	Yes	Yes	No.	None.
Session Pool ID	16 bits	Table OASESNPL	Yes	Yes	Yes	No	None

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Session ID	16 bits	0 to 1023	Yes	Yes	Yes	No	None
Call ID	32 bits	Match to actual	Yes	Yes	Yes	No	None
Solicitor Number	16 bits	Node, Node Alarm, Node Log: 0 - 250 for switch to SN request; 256-65534 for sn to switch request. SP, SP Alarm, SP Log: 0 - 250 for switch to SN request; 256-511 for sn to switch request	Yes	No	Yes	No	None

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Operation/Response Type (See footnote ^e)	8 bits	#A1 #A2 #A3 #A4	Yes	Invalid: Yes Unex- pected: Yes Reject: Yes	Invalid: Yes Unex- pected: Yes Reject: No	Invalid: No Unex- pected: No Reject: Yes	None (See foot- note ^f)
Operation/Response Length Format	8 bits	#82	Yes	Yes	Yes	No	None
Operation Response Length	16 bits	(See foot- note ^g)	Yes	Yes	Yes	No	None
Invoke ID Type	8 bits	#02	Yes	Yes	Yes	No	None
Invoke ID Length Format	8 bits	#02	Yes	Yes	Yes	No	None
Invoke ID Value	16 bits	(See foot- note ^h)	Yes	Yes	Yes	No	None
Operation ID Type	8 bits	#02	Yes	Yes	Yes	No	Reject - InvokeP- roblem, Mistyped Argument

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Operation ID Length Format	8 bits	#02	Yes	Yes	Yes	No	Reject - InvokeP- roblem, Mistyped Argument
Operation ID Value	16 bits	See list of operation ids in appendix B	Invalid: Yes Restrict ed Mode: None	Invalid: Yes Restrict ed Mode: None	Invalid: Yes	No	Invalid: Reject - InvokeP- roblem, Unrecog- nizedOp- eration Restricted Mode: Return Error
Argument Type	8 bits	#04	Yes	Yes	Yes	No	Reject - InvokeP- roblem, Mistyped Argument

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Argument Length Format	8 bits	#82	Yes	Yes	Yes	No	Reject - InvokeP- roblem, Mistyped Argument
Argument Length	16 bits	See footnote ⁱ	Yes	Yes	Yes	No	Reject - InvokeP- roblem, Mistyped Argument
Result Type	8 bits	#04	Yes	Yes	Yes	No	Reject - ReturnRe- sultProb- lem, Mistyped Result
Result Length Format	8 bits	#82	Yes	Yes	Yes	No	Reject - ReturnRe- sultProb- lem, Mistyped Result

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Result Length	16 bits	See footnote ⁱ	Yes	Yes	Yes	No	Reject - ReturnResultProblem, Mistyped Result
Parameter Type	8 bits	#04	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, Mistyped Parameter
Parameter Length Format	8 bits	#82	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, Mistyped Parameter
Parameter Length	16 bits	See footnote ⁱ	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, Mistyped Parameter

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Error ID Type	8 bits	#02	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, Mistyped Parameter
Error ID Length Format	8 bits	#02	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, Mistyped Parameter
Error ID Value	16 bits	See Error ID Values chapter.	Yes	Yes	Yes	No	Reject - ReturnErrorProblem, UnrecognizedError
Problem Type	8 bits	See Message Format chapter. See footnote ^j	Yes	Yes	Yes	No	None
Reject Reason Length Format	8 bits	#82	Yes	Yes	Yes	No	None
Reject Reason Length	16 bits	#0002	Yes	Yes	Yes	No	None

Table 8. DMS OAP Incoming Message Error Handling

Field	Format	Valid Range	Logs for Invalid	OMs for Invalid	Failed Msg Peg	Process Operations	DMS Response
Reject Reason Value	16 bits	See Message Format chapter. See footnote ^j	Yes	Yes	Yes	No	None
Data Block ID	16 bits	See Appendix B.	Bad Format, Out of Range, Missing, Unexpected: Yes	Bad Format, Out of Range, Missing, Unexpected: Yes (See footnote ^k)	All: Yes	No	Reject. - (See footnote ^l)
Data Block Length	16 bits	See Data Block Descriptions chapter. See footnote ^m	Yes	Yes	Yes	No	Reject. - (See footnote ^l)
Field Value	See OAP Spec.	See Field Definitions chapter.	Yes	Yes	Yes	No	Reject. - (See footnote ^l)

- a. The high 15 bits of the Options field are not checked (i.e. no error is generated for any value). Bit 0, when set to 1 indicates a PVE Reject from the sender. The DMS will log the fact that the switch received a reject message.
- b. The valid Message Protocol Version range depends on the current DMS OAP version as well as a list of the prior versions supported. The general valid range is current to 3 releases back (dependent on forced upgrades and existence of prior releases) with all released increments supported (Note: Invalid increments within valid releases are considered invalid versions.). If the message version is within the valid range, but not optimum, a PVE Reject (with reflected header, etc.) is returned to the sender.
- c. The Sender's Max Protocol Version is only checked when negotiating the protocol level. If the Sender's Max Protocol Version is not within +/- 3 releases of the DMS OAP protocol version, a valid Message Protocol Version cannot be determined and the associated logs and OMs occur.
- d. The expected Sequence Number for the first incoming message in a session is 1. Subsequent incoming messages should increment that value. OMs are generated, as indicated, when the received value is higher or lower than expected. The message is not processed if the Sequence Number received is lower than expected. Sequence numbers are only verified for Call Processing Class messages.
- e. The valid range of Operation/Response Types is #A1-#A4. Operation/Response Types outside of that range generate a log. The Reject of a Session Begin will result in an action according to Table OAFNDISP.
- f. A Reject will only be returned for an inappropriate Operation/Response Type if the operation can be parsed far enough to determine if a valid InvokeID field exists. A Reject is not sent when an invalid Reject message is received.
- g. The Operation/Response length is validated against the Length of Following Message plus the relative offset. Data block lengths are verified against the end of operation as calculated from the Operation/Response Length to ensure they do not indicate a data length that exceeds the end of the operation.
- h. For incoming Invokes, the valid range of InvokeIDs is #0100-#DFFF. For incoming responses (ReturnResults, ReturnErrors, and Rejects) the specific InvokeID is compared against those of recent outgoing Invokes, ReturnResults, and ReturnErrors.
- i. The length of the data accompanying an operation/result should match the Operation Response Length minus the intervening header field lengths.
- j. Problem Types and Reject Reasons are specified in the OAP Spec. The values in the OAP Spec are a subset of those defined by OSI for ROSE.
- k. Errors are detected for missing mandatory data blocks as well as invalid data blocks. Missing data blocks are detected after message parsing is complete. Unexpected data blocks are valid data blocks, unexpected for the operation/result/error or in invalid combination of data blocks (e.g. more than 1 optional data block for a '1 of 3' expected).
- l. InvokeProblem MistypedArgument, ReturnResultProblem MistypedResult, or ReturnErrorProblem MistypedParameter, depending on Operation/Response Type.
- m. The length of a data block must not indicate data past the calculated end of the operation, else one of the lengths is invalid.

Chapter 7: Call Processing Operations

7.0 Operation Grouping Name (Template)

This section has the templates that are used for defining OAP operations and responses.

The operations themselves (along with responses) are placed in alphabetic order using 7.x as a heading.

7.0.1 Request Operation

Identifier: *This template is used for request operations.*

Description: *What this operation is used for.*

Preconditions: • *Characteristics or information required to successfully perform this operation.*

Post Conditions: • *Changes made as a direct result of successfully performing this operation.*

Side Effects: • *Anything that occurs as a result of successfully performing this request.*

Response(s): • *List of responses if any*

- X Success Response
- X Error Response

Sender(s): • *List of potential senders who can send this message*

- Call Processing Switch
- Active Service Node
- Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): *“None”, if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this request message.*

Duplicate Request: • *Specifies what will happen if this request is received twice by the switch.*

7.0.1.1 Success Response

Interpretations: *Describes what it means to get this response.*

Message Class: Call Processing

Message Type: Return Result

Data Block(s): *“None”, if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this response message.*

7.0.1.2 Error Response

Interpretations: *Describes what it means to get this response.*

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress

Comments: 1. *Additional information that relates to this response message.*

7.0.2 Inform Operation

Identifier: *This template is used for operations that do not receive a response.*

Description: *What this operation is used for.*

Sender(s): • *List of potential senders who can send this message*

- Call Processing Switch
- Active Service Node
- Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): *“None”, if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this response message.*

Lost Message: • *States what will happen if this operation is lost.*

7.1 Operation Groups

This section lists the operations by groups, along with additional information. The operations themselves are described, alphabetically, in the remainder of this chapter.

7.1.1 AMA

List of Operations:

- Append AMA Module Request
 - Append AMA Module Success Response
 - Append AMA Module Error Response
- Generate AMA Record Request
 - Generate AMA Record Success Response
 - Generate AMA Record Error Response

Service nodes can append formatted billing data to a billing record. The switch is unaware of the contents of the billing data (however, the quantity, in hex digits, of billing data should be specified by the service node). The knowledge of the data is in the service node and downstream processor.

Once the service node sends the data to the switch, the switch appends the data to the AMA record. The data is appended using a module code. The switch decides which of the following two module codes to use based on the size of the AMA data sent by the service node.

- Module Code 186, Service Node Formatted - Small: This module code is used for AMA data that is 40 hex digits (20 bytes) or less. It is replicative. It can be appended to an AMA record multiple times.
- Module Code 187, Service Node Formatted - Large: This module code is used for AMA data that is 140 hex digits (70 bytes) or less. It is replicative. It can be appended to an AMA record multiple times.

The total size of all module codes 186 and 187 in one AMA record cannot exceed 1600 hex digits (800 bytes). (Note that if a call consists of multiple call sessions (e.g. due to triggers), the earlier sessions may have appended AMA modules without generating an AMA record, leaving latter sessions with less than the full 1600 hex digits available prior to 'overflow'.) If an attempted module append would cause the AMA buffer limit of 1600 hex digits to be exceeded, the Append AMA Module Request returns an 'AMA Buffer Overflow' error, and the requested AMA module is NOT appended. At this point a service node might want to send a Generate AMA Record Request to clear the buffer, and then resend the failed Append AMA Module Request.

AMA sequence numbers, intended as unique identifiers to avoid instances of duplicate or lost billing, are associated with both the Append AMA Module Request and Generate AMA Record Request operations. These sequence numbers should be initialized upon the start of every session such that the first requests for the respective operations contain a sequence number of one. Subsequent requests for each AMA request type should increment the associated sequence number with each send. The switch will process Append AMA Module Requests and Generate AMA Record Requests if the received sequence number is higher than the prior sequence number for the same request type, but will return an error, and not process the AMA request, if the received sequence number is the same or lower than the prior request.

If the response time from the switch appears excessive, with a potential for a lost message (request or the response), the service node should send the same AMA request again (same AMA sequence number and data, but appropriately updated OAP class header information). If the switch had not received the original AMA request, the repeated AMA request will be processed and responded to normally. If the switch processed the prior request, and the response message was lost, the repeated request will return an AMA sequence error response. If no response is received for this subsequent request, it is possible that the switch is unavailable (e.g. in the process of software upgrade), and the service node should take steps appropriate to the function or service being provided. *In no case should a service node initiate a repeat request until the switch response time gets quite excessive, as response delays are most likely to be greatest when the switch is under heaviest load.*

7.1.2 Billing

List of Operations:

- Billing Number Request
 - Billing Number Success Response
 - Billing Number Error Response
- Calling Number BNS Query Request
 - Calling Number BNS Query Success Response
 - Calling Number BNS Query Error Response
- Class Charge Request
 - Class Charge Success Response
 - Class Charge Error Response

The switch must have the following information in order to successfully class charge a call:

- A party directory number: The Directory Number Request operation can be used to send this information to the switch.
- B party directory number: The Directory Number Request operation can be used to send this information to the switch. The B party DN is not needed for calls requiring directory assistance. The B party is needed if the service type is TA and the class charge is one of the following:
 - collect
 - station special calling, bill to an alternate DN
 - station special calling, bit to non-custom card
- Billing number: The Billing Number Request operation can be used to send this information to the switch. The data block sent (Alternate Billing Directory Number DB, BCD Card Number DB, ASCII Card Number DB) depends on the type of billing number. The Billing Number Request operation does not have to be sent if the switch already has the billing number (e.g. class charged collect and switch has forward party DN).

The service node can request the switch to validate the billing number with the Class Charge request. If validation is requested, the switch is in a “Restricted Input Mode” while performing the validation query. This means that the switch will not accept any requests until the query ends.

There are two types of external database queries that can be performed for billing validation: BNS (Billed Number Screening) and CCV (Calling Card Validation). BNS queries are used for alternate and collect directory numbers. CCV queries are used for calling card numbers. Refer to the TOPS translations guide for more information on what happens when the switch sends a validation query and what checks are performed before the query is sent.

Note that after a call has been class charged and billing is satisfied, the call can then be completed using the Call Float Request operation.

The Calling Number BNS Query Request operation is used when the SN wants the switch to perform a BNS query on the calling number. Both a calling and called number must be present prior to the request. This operation does not set the class charge for the call, it only performs a query.

7.1.3 Call Details

List of Operations:

- Call Details Request
 - Call Details Success Response
 - Call Details Error Response

7.1.4 Call End From the Switch

List of Operations:

- Call End Inform

This operation will end the session between the switch and service node.

7.1.5 Call Float

List of Operations:

- Call Float Request
 - Call Float Success Response
 - Call Float Error Response

The switch must have the following information in order to successfully complete a call:

- A party directory number: The Directory Number Request operation can be used to send this information to the switch.
- B party directory number: The Directory Number Request operation can be used to send this information to the switch.
- Billing number: The Billing Number Request operation can be used to send this information to the switch. The data block sent (Alternate Billing Directory Number DB, BCD Card Number DB, ASCII Card Number DB) depends on the type of billing number.
- Class charge: The Class Charge Request operation can be used to send this information to the switch.

7.1.6 Call Resume

List of Operations:

- Call Resume Request
 - Call Resume Success Response
 - Call Resume Error Response

This operation is used to release a call from the service node when providing OSSAIN preprocessing. Refer to the Functionality Overview section on OSSAIN Preprocessing for more information.

7.1.7 Carrier

List of Operations:

- Carrier Assignment Request
 - Carrier Assignment Success Response
 - Carrier Assignment Error Response

7.1.8 Connecting To Treatment

List of Operations:

- Connect To Treatment Request
 - Connect To Treatment Success Response
 - Connect To Treatment Error Response
- Route To Treatment Request
 - Route To Treatment Success Response
 - Route To Treatment Error Response

The Connect to Treatment operation will connect a party to treatment while the service node remains in control of the call.

The route to treatment operation will end the session between the switch and service node while connecting the party to treatment.

7.1.9 Conferencing

List of Operations:

- Call Merge Inform
- Call Merge Request
 - Call Merge Success Response
 - Call Merge Error Response
- Conference Add Request
 - Conference Add Success Response
 - Conference Add Error Response
- Conference Create Request
 - Conference Create Success Response
 - Conference Create Error Response
- Conference Details Request
 - Conference Details Success Response
 - Conference Details Error Response
- Conference Release Request
 - Conference Release Success Response
 - Conference Release Error Response
- Conference Remove Request
 - Conference Remove Success Response
 - Conference Remove Error Response

Conferencing operations are used to allocate/deallocate conferencing resources, add/remove parties to/from a conference, and to obtain information about the conference. The Call Merge Request is used to merge the A party from one call into the A party slot of another call that has a B party but no A party. Both parties must be in the same conference for the call merge.

For the following operations, a Conference ID DB is sent if the “call” is involved in a “conference”. The service node should use the Conference Details Request if it wishes to receive complete details on each of the parties involved in the “conference” (Conference Details will include information about ALL parties in the “conference”, not just the ones associated with this particular “call”).

- Call Details Success Response
- Session Begin Inform

- Trigger Event Inform

For the following operations, the service node may specify whether a party(ies) or voice link should be added to the conference:

- Call Float Request
- Connect DN Request
- Voice Connect Request

The following operations are affected if a party(ies) or voice link was involved in a conference:

- End Call Request - If the call being ended is in any way involved in a conference, all conference resources associated with the call will be released. If this is the last call associated with a particular conference, the entire conference is ended and all DMS resources for the conference are freed.
- Release DN Request - If the party being released is connected to a conference, that connection will be released. The remaining parties in the conference are unaffected.
- Voice Release Request - If the SN voice link was connected to a conference, that connection will be released. The remaining parties in the conference are unaffected.

7.1.10 Connection Status

List of Operations

- Connection Status Inform

7.1.11 Directory Number

List of Operations:

- Directory Number Request
 - Directory Number Success Response
 - Directory Number Error Response
- ISUP Calling Party Number Update Request
 - ISUP Calling Party Number Update Success Response
 - ISUP Calling Party Number Update Error Response
- LNP Request
 - LNP Success Response
 - LNP Error Response
- LRN Update Request
 - LRN Update Success Response
 - LRN Update Error Response

When the service node requests to update a directory number, the switch is in a “Restricted Input Mode” while updating the directory number (an LNP query might be needed). This means that the switch will not accept any requests until it has completed processing the Directory Number Request.

If the directory number translated to a treatment, the switch will respond with a Return Result, specifying the treatment. A success response is returned because the directory number is successfully stored in the switch.

If the service node wants to route the call to the treatment that was returned in the Return Result, it can send the Route To Treatment Request. The switch will route the call to the treatment specified in the request and release the service node from the call.

If the service node wants the treatment specified in the Directory Number Success Response to be applied to the call but does not want to be released from the call, it can send the Connect DN Request. The switch will apply the treatment. However, the response will be a Connect DN Return Error message, specifying that a treatment was applied. This is due to the fact that the switch did not connect to the directory number, but connected to a treatment. This is consistent with other error reasons for the Connect DN Request.

If the service node wants to release the treatment being applied, it can send the Release DN Request. The switch will release the treatment being applied.

The LNP Request operation is used to obtain the location routing number for the specified party (A party, B party, alt DN, or a line based calling card number). Refer to the Functionality Overview section on Local Number Portability for more information on LRN.

The LRN Update operation is used to set or update the Location Routing Number information associated with the given party.

7.1.12 Directory Number Connection

List of Operations:

- Connect DN Request
 - Connect DN Success Response
 - Connect DN Error Response
- Release DN Request
 - Release DN Success Response
 - Release DN Error Response

7.1.13 DTMF Receiver

List of Operations:

- DTMF Digit Detected Inform
- DTMF Receiver Release Request
 - DTMF Receiver Release Success Response
 - DTMF Receiver Release Error Response

7.1.14 End Call By Active Service Node

List of Operations:

- End Call Request
 - End Call Success Response
 - End Call Error Response

The operation request will end the session between the switch and service node.

7.1.15 Estimate of Charges

List of Operations:

- Estimate of Charges Request
 - Estimate of Charges Success Response
 - Estimate of Charges Error Response

7.1.16 Service Change

List of Operations:

- Service Change Request
 - Service Change Success Response
 - Service Change Error Response

7.1.17 Service Provider Identifier

List of Operations:

- Service Provider ID Assignment Request
 - Service Provider ID Assignment Success Response
 - Service Provider ID Assignment Error Response

SPID information is obtained via the OLNS database or switch Table SPID if the SPID SOC option ENSV0017 is turned on at the switch. The SPID data is sent to the service node if the Operator Services AIN Version 3 SOC option OSAN0003 is turned on.

Both SOC options must be turned on to process the SPID Assignment request.

7.1.18 Session Begin from the Switch to the Service Node

List of Operations:

- Session Begin Inform

The switch sends a session begin message when it is sending a call to the service node.

The next request message from the service node to the switch will inform the switch if the service node has accepted the call:

- If the service node chooses to accept the call, then it processes the call as usual. The request message will be some type of call processing operation request.
- If the active service node does not want to accept the call, then it requests the switch to perform an action on the call. The following lists some call disposition operations that can be used:
 - Call Float Request
 - End Call Request
 - Route to Treatment Request
 - Transfer to Carrier Request
 - Transfer to Control List Request

If the call sent to the service node has a service type of directory assistance (field Service Type in the Call Information DB), then the call may be at the service node to obtain alternate billing (typically to a third number or calling card number). Once billing has been established, the call should be routed to a control list to continue DA processing within the DMS switch environment. These calls should only be disposed of via a Transfer to Control List operation or an End Call operation.

7.1.19 Session Initiation by the Service Node

List of Operations

- Session Initiation Request
 - Session Initiation Success Response
 - Session Initiation Error Response
- Session Recall Request
 - Session Recall Success Response
 - Session Recall Error Response

These operation requests begin a session between the switch and service node.

A Session Recall Request can be used to initiate a session with a floated call, provided the CallID of the call to be recalled is known. In order to initiate a session with a floated call, the service node must send the Session Recall Request on an idle SN-originated session. If the target call already has a session with a service node, the session pool and session id that the call is connected to will be returned in the Return Error message. If the target call is floated, it will be connected to the SN-originated session and a Return Result will be sent with full call details.

Session Initiation Request: Connecting to the A Party

For a Session Initiation, the order of operations is not enforced, and therefore, the appropriate information may not be available when needed for equal access translations. In order for equal access translations to be performed, the following order must be used by the service node.

1. Initiate a session with the switch via the Session Initiation Request.
2. Enter the A party DN via the Directory Number Request.
 - The A party DN could have been sent with the Session Initiation Request.
3. Connect to the A party via the Connect DN Request.
 - The originating information is now available at the switch.
4. Enter the B party DN via the Directory Number Request
 - Equal access translations is performed.
5. Assign a carrier to the call via the Carrier Assignment Request
 - This is optional if a carrier is already assigned and mandatory if this is an interlata call with no carrier assigned.
 - In order to float the call or connect to the B party, the switch must provide operator services for the carrier.
6. Connect to the B party DN via the Connect DN Request

- or float the call via the Call Float Request with option to connect the B party.

Session Initiation Request: Not Connecting the A Party

In some cases, the A party is not connected for a Session Initiation. This is true for a 3-way conferencing scenario where the A party is already connected in the first leg (the A-B party leg) and the second session (initiated by the service node) is used to connect to the C party.

In order for the switch to do equal access translations, the A party trunk information is needed. Since the A party is not connected in the C-party session, this information is not available at the switch. However, the service node has this information from the A-B party session. The service node must send the originating trunk information to the switch prior to updating the B party DN.

The following is the order of operations for initiating a session and not connecting to the A party:

1. Initiate a session with the switch via a Session Initiation Request.
2. Enter the A party DN via the Directory Number Request.
 - Send the Originating Trunk Information DB
 - The A party DN and the Originating Trunk Information DB could have been sent with the Session Initiation Request.
3. Enter the B party DN via the Directory Number Request.
 - Equal access translations is performed since the switch has the originating trunk information.
4. Assign a carrier to the call via the Carrier Assignment Request.
 - This is optional if a carrier is already assigned and mandatory if this is an interlata call with no carrier assigned.
 - If the switch returned a Carrier DB in the Directory Number Success Response for the B party, the service node should compare it to the preferred carrier for the A party returned in the OLNS Information DB sent in the A-B party session. If the carrier is not the same, the service node should update the switch with the correct carrier.
 - In order to float the call or connect to the B party, the switch must provide operator services for the carrier.
5. Connect to the B party DN via the Connect DN Request.

7.1.20 Simultaneous Interactions

List of Operations:

- Accept Control Inform
- Node Release Inform
- Pass Control Request
 - Pass Control Success Response
 - Pass Control Error Response
- Pass Through Inform
- Pass Through Request
 - Pass Through Success Response
 - Pass Through Error Response
- Passive Function Provider Request
 - Passive Function Provider Success Response
 - Passive Function Provider Error Response
- Passive Node Status Inform
- Release Node Request
 - Release Node Success Response
 - Release Node Error Response

A service node which has call control capabilities (known as the “active” node or function provider) may bring a second function provider (either another service node or a TOPS TA operator) into the call. This second function provider (known as the “passive” node or function provider) does not have call control capabilities and is restricted in terms of what it can request from the switch. The only valid OAP requests from a passive node are:

- Voice Connect Request
- Voice Release Request
- Release Node Request
- Pass Through Request

The Release Node Request can be used by the passive node to release itself, without affecting the switch’s connection with the active node. Furthermore, the Release Node Request can be used by the active node to either release the passive node or to release itself. If the active node uses the Release Node Request to release itself, the passive node will receive an Accept Control Inform and it becomes the active node for the call.

7.1.21 Speech Path

List of Operations

- Speech Path Request
 - Speech Path Success Response
 - Speech Path Error Response

7.1.22 Text To Operator

List of Operations:

- Text To Operator Request
 - Text To Operator Success Response
 - Text To Operator Error Response

7.1.23 Timers

- Conversation Timing Request
 - Conversation Timing Success Response
 - Conversation Timing Error Response
- Notification Timer Inform
- Notification Timer Request
 - Notification Timer Success Response
 - Notification Timer Error Response

The Notification Timer Request allows the service node to start/stop/reset the notification timer while the call has a session with the node. The service node may choose to use either the “float-style” notification timer (run only while the call is floated) or the “persistent-style” notification timer (run while the call has a session with the service node, as well as when the call is floated).

If the “persistent” notification timer expires while the call is floated, it will trigger just as the “float-style” notification timer would. If the timer expires while the call has a session with the service node, the Notification Timer Expiration Inform will be sent to the service node.

Note: only billable calls accumulate conversation timing.

7.1.24 Transfer

List of Operations

- Transfer to Carrier Request
 - Transfer to Carrier Success Response
 - Transfer to Carrier Error Response
- Transfer to Control List Request
 - Transfer to Control List Success Response
 - Transfer to Control List Error Response

These operation requests will end the session between the switch and service node.

If a call is being transferred to a DA operator (via Transfer to Control List Request), ADAS eligibility checks are performed and the call is routed to ADAS processing when applicable. Note that this is true for both TA and DA calls being routed to a DA operator. The DMS switch changes the service of TA calls to DA for calls routed to a DA operator.

7.1.25 Trigger Event Inform

List of Operations:

- Trigger Event Inform

The switch sends this message when it is sending a call to the service node due to a trigger event but is not giving call control to the service node. The service node should not respond to the message.

7.1.26 Voice Link

List of Operations:

- Voice Connect Request
 - Voice Connect Success Response
 - Voice Connect Error Response
- Voice Release Request
 - Voice Release Success Response
 - Voice Release Error Response
- Voice Release Inform
- Return Answer Message Request
 - Return Answer Message Success Response
 - Return Answer Message Error Response

7.2 Accept Control Inform^{SV3}

Identifier: #61E

Description: Used to indicate to a passive service node that it is now the active service node for the call.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes
M	Call Information DB	20 bytes
M	Node Identification DB	8 bytes
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
M	Charge Status DB	6 bytes
1 to 2 ^{SV4}	Class Charge DB	10 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2	Directory Number DB	34 bytes
0 to 3	Alternate Billing Directory Number DB	18 bytes
O	BCD Card Number DB	24 bytes
O	ASCII Card Number DB	38 bytes
O	Context Block DB	56 bytes
0 to 2	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information	6 bytes
O to 3	LNP Information DB	20 bytes
0 to 6	SPID DB	12 bytes
O ^{SV4}	ISUP Calling Party Number DB	16 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes
O ^{SV5}	Country Direct Information DB	14 bytes
O ^{SV6}	SN Voice Link DB	12 bytes
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes
O ^{SV6}	Estimate Charges General Information DB	10 bytes
1 of 2 (O) ^{SV6}	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

- Comments:**
1. The Function data block contains the function that should be provided by the service node. It will be the same function that was sent in the Session Begin message that was sent when this node was originally connected.
 2. The Node Identification data block gives information on the service node that relinquished control of this call. In addition to other information, the Node Identification data block will indicate whether that node is still connected in a passive mode or if it is no longer connected to the call.

3. The Originating Trunk Information data block is optional because there may not be an originating party present.
4. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
5. The Directory Number data block is optional because there may not be any parties currently present.
6. If context data has been established by any function provider previously involved in the call, then the Context Block data block is sent to the service node.
7. The Alternate Billing Directory Number DB is sent if there is an alternate (or third number) billing number for the call.
8. The BCD Card Number is sent if there is a domestic calling card number for the call.
9. The ASCII Card Number DB is sent if there is a foreign calling card number for the call.
10. A treatment data block is sent if a treatment is applied to either the A party port or B party port.
11. The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.
12. If LNP information on the directory number(s) (A party, B party, alternate directory number or line based calling card) is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned.
13. The SPID DB is returned if there is any account owner or billing service provider information for the specified party.
14. Up to two Class Charge data blocks may be sent. One Class Charge DB is sent with current class charge information. The current class charge is always sent. A second Class Charge DB is sent if the Session Origination type is Reorigination (the call was RLT'd and reoriginated back to the switch). The previous class charge is sent if SV4 SOC option is on.
15. The ISUP Calling Party Number DB is sent if the number is available and in effect at the switch.
16. The Terminating Trunk Information is sent when the terminating party is present.
17. The Country Direct Information DB is sent if the call origination is of type country direct.
18. If the voice link was selected by the switch, the voice link information is sent in the SN Voice Link DB.

19. The Estimate of Charges General Information, Duration and No Duration DBs are only sent if a successful estimate of charges has been previously performed on the call.

Lost Message: • If this message is lost or the service node does not respond, switch timers will eventually expire. See Chapter 3 for more details on lost message handling by the switch.

7.3 Append AMA Module Request

Identifier: #600

Description: Used by the service node to append its formatted billing data to the base AMA structure in modular form. The switch continuously collects billing information during call progression. Therefore, the service node can send a series of Append AMA Requests as the call proceeds during a call session.

Preconditions: • None.

Post Conditions: • The billing record is changed to reflect the inclusion of the new AMA module.

Side Effects: • If billing record does not exist, one is created

Response(s): • Append AMA Module Success Response
• Append AMA Module Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	AMA Module DB	78 bytes

Comments: 1. The switch returns an AMA Buffer Overflow error response to an Append AMA Module Request if it determines that the new append would cause the sum of the 186 and 187 modules appended to a billing record to exceed 1600 hex digits (800 bytes). If this error is returned to the service node the requested AMA module was NOT appended. The service node may choose to handle this error response by issuing a Generate AMA Record Request, which clears the AMA buffer, and then reissue the failed Append AMA Module Request.

Duplicate Request: • The Append AMA DB contains an Append AMA Sequence Number field. This sequence number is handled as described in 7.1.1 “AMA” on page 181. As noted there, the first time during a session that a service node sends an Append AMA Module Request operation, the sequence number sent should be set to ‘one’. Each subsequent time, during the same session, that a service node sends an Append AMA Module Request the sequence number should be incremented. Note that the sequence number should only

be reset upon the start of a new session, and not upon a Generate AMA Record Request.

7.3.1 Append AMA Module Success Response

Interpretations: The billing data has been successfully appended to the billing data for this call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.3.2 Append AMA Module Error Response

Interpretations: The billing data could not be appended to the existing billing information.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress
0701	AMA buffer overflow
0704	Invalid append AMA sequence number

- Comments:**
1. If an Append AMA Module Request would cause the sum of the accumulated 186 and 187 modules appended to exceed 1600 hex digits (800 bytes), an AMA buffer overflow (0701) error is returned. The AMA module is not appended.
 2. If the Append AMA Sequence Number is the same or lower than that sent with the prior Append AMA Module Request (during the same session), an Invalid append AMA sequence number (0704) error is returned to the service node.

7.4 Billing Number Request

Identifier: #602

Description: Requests that the specified directory number or card number be associated with the call as the billing number.

Preconditions: • If a Routing Index or Routing Number DB is included with this operation, the switch must be using North American translations.

Post Conditions: • The number is stored as the billing number.

Side Effects: • The class charge is set to unspecified.

Response(s): • Billing Number Success Response
• Billing Number Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
1 of 3 (M)	Alternate Billing Directory Number DB	18 bytes
	BCD Card Number DB	24 bytes
	ASCII Card Number DB	38 bytes
1 of 2 (O) ^{SV6}	Routing Index DB	8 bytes
	Routing Number DB	16 bytes
O ^{SV7}	LNP Enhancements DB	6 bytes

- Comments:**
1. Note that some billing numbers overwrite previous billing numbers stored in the switch. The alternate billing directory number overwrites the card number. The card number overwrites the alternate billing directory number. These two types of billing numbers cannot be stored simultaneously for a call.
 2. Directory numbers are validated using BNS queries to the LIDB, if DMS datafill specifies.
 3. Card numbers are validated using CCV queries to the LIDB or RIDB, if DMS datafill specifies.
 4. The Class Charge Request Operation specifies whether the service node wants the switch to validate the billing number. The following lists the appropriate class charge for the billing number:
 - If the billing number is an alternate directory number (e.g. bill to third number), the class charge must be set to Station Special Calling.

- If the billing number is a collect directory number, the class charge must be set to Station Collect or Person Collect.
 - If the billing number is an account code, the class charge must be set to Station Paid.
 - If the billing number is a card number, the class charge must be set to Station Special Calling/Called.
5. If the service node wants to set the class charge for the call, it should send the Class Charge operation subsequent to sending the Billing Number Request operation.
 6. If the billing number is a custom calling card and calling card restrictions apply (refer to the Allowed Billing Set field), the class charge to the custom card will not be blocked by the switch.
 7. A Routing Index or Routing Number DB is only allowed in this operation if the Alternate Billing Directory Number DB is also included. That DB must include a valid alternate billing DN.
 8. When a Routing Index or Routing Number DB is included with this operation, the Connection ID field of either DB must be set to Alt DN Connection.
 9. If a Routing Index DB is sent with a Routing Index of 2046 (the NIL value), the alternate routing index and alternate routing number associated with the alternate DN connection will be cleared.
 10. If a Routing Number DB is sent with a Directory Number Digit Count of 0, the alternate routing number and alternate routing index associated with the alternate DN connection will be cleared.
 11. BNS validation is not performed on a Directory Number in a Routing Number DB.

7.4.1 Billing Number Success Response

Interpretations: The billing number is stored for the call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	BCD Card Information DB	6 bytes
O	LNP Information DB	20 bytes

- Comments:**
1. The service node must send the Class Charge Request if it wants the switch to send a validation query for this billing number.
 2. If the billing number is a BCD Card Number and the BCD Card Number Type field is set to Unknown in the Billing Number Request, then the DMS determines the BCD Card Number Type and returns it in the BCD Card Information DB.
 3. If LNP information was obtained for the B party directory number (which is being used for the billing number), the LNP Information DB is returned.

7.4.2 Billing Number Error Response

Interpretations: There was a problem with the billing number.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0705	Invalid language
0707	Invalid number format
070B	No billing number
070C	Billing number is hotlisted
0713	Third number in B party connection
076C	Invalid Routing Index
076D	North American-only functionality
076E	Connection ID - Party mismatch

Comments: 1. Error ID #076D is sent if the Billing Number Request included a Routing Index DB or Routing Number DB and the switch does not use North American translations.

7.5 Call Details Request

Identifier: #604

Description: The service node is requesting an update on call information. The service node can send this request when it is unsure of what has happened to the call.

Preconditions: • None.

Post Conditions: • None.

Side Effects: • None.

Response(s): • Call Details Success Response
• Call Details Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None

Comments: 1. None.

7.5.1 Call Details Success Response

Interpretations: Current call information is sent to the service node.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes
M	Call Information DB	20 bytes
1 of 2 (O)	Node Identification DB	8 bytes
	Operator Identification DB	8 bytes
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
M	Charge Status DB	6 bytes
1 to 2 ^{SV4}	Class Charge DB	10 bytes
1 of 5 (O)	Trigger Information DB	6 bytes
	Trigger Treatment DB	6 bytes
	Trigger Release Cause DB	8 bytes
	Trigger Ringing DB	8 bytes
	Trigger DTMF DB	8 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2	Directory Number DB	34 bytes
0 to 3	Alternate Billing Directory Number DB	18 bytes
O	BCD Card Number DB	24 bytes
O	ASCII Card Number DB	38 bytes
O	Context Block DB	56 bytes
0 to 2	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information DB	6 bytes
O to 3 ^{SV3}	LNP Information DB	20 bytes
M	Time DB	10 bytes
0 to 6 ^{SV3}	SPID DB	12 bytes
O ^{SV3}	Preprocessing Information DB	8 bytes
O ^{SV4}	Conference ID DB	8 bytes
0 to 2 ^{SV4}	Conference Party Connection DB	10 bytes
O ^{SV4}	Conference SN Voice Connection DB	6 bytes
O ^{SV4}	ISUP Calling Party Number DB	16 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes
O ^{SV5}	Country Direct Information DB	14 bytes
O ^{SV6}	SN Voice Link DB	12 bytes
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes

M/O	Data Blocks	Byte Length
O ^{SV6}	Estimate of Charges General Information DB	10 bytes
1 or 2 (O) ^{SV6}	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

- Comments:**
1. The Function DB contains the function that is currently being provided by the service node.
 2. Information on the previous function provider (i.e. the last function provider before this service node) involved in the call is sent to the service node. If the previous function provider was a service node, then the Node Identification data block is used. If the previous function provider was an operator, then the Operator Identification data block is sent. If there was not a previous function provider for the call (i.e. the service node is the first function provider), then neither data block is sent.
 3. The Originating Trunk Information data block is optional because there may not be an originating party present.
 4. If a trigger event occurred that caused the service node to be brought into the call, then one of the five trigger data blocks will be sent. The trigger data block that is sent depends on the trigger event that occurred. The trigger data blocks contain information on the trigger events that caused the service node to be brought into the call. These data blocks are optional. The service node may not have been brought into the call as a result of a trigger (e.g. the call is transferred to the service node from an operator or another service node).
 5. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
 6. The Directory Number data block is optional because there may not be any parties currently present.
 7. If context data has been established by any function provider involved in the call (including the current one), then the Context Block data block is sent to the service node.
 8. The Alternate Billing Directory Number DB is sent if there is an alternate (or third number) billing number for the call.
 9. The BCD Card Number is sent if there is a domestic calling card number for the call.
 10. The ASCII Card Number DB is sent if there is a foreign calling card number for the call.
 11. A treatment data block is sent if a treatment is applied to either the A party port or B party port.

12. The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.
13. If LNP information on the directory number(s) (A party, B party, alternate directory number or line based calling card) is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned.
14. The SPID DB is returned if there is any account owner or billing service provider information for the specified party.
15. The Preprocessing Information DB is only sent on OSSAIN preprocessing sessions.
16. The Conference ID DB is sent if this call is associated with a conference. The Conference Party Connection DB is sent if a subscriber party is connected to the conference. The Conference SN Voice Connection DB is sent if a SN voice link is connected to the conference.
17. Up to two Class Charge data blocks may be sent. One Class Charge DB is sent with current class charge information. The current class charge is always sent. A second Class Charge DB is sent if the Session Origination type is Reorigination (the call was RLT'd and reoriginated back to the switch). The previous class charge is sent if SV4 SOC option is on.
18. The ISUP Calling Party Number DB is sent if the number is available and in effect at the switch.
19. The Terminating Trunk Information is sent when the terminating party (B party or Alternate DN) is present.
20. The Country Direct Information DB is sent if the call origination is of type country direct.
21. If the voice link was selected by the switch, the voice link information is sent in the SN Voice Link DB.
22. The Estimate of Charges General Information, Duration and No Duration DBs are only sent if a successful estimate of charges has been previously performed on the call.

7.5.2 Call Details Error Response

Interpretations: There was a problem returning the call details to the service node.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress

Comments: 1. None.

7.6 Call End Inform

Identifier: #605

Description: Sent to the service node involved in the call when the switch needs to abort the call for some reason.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments:

1. All outstanding requests from all function providers involved in the call are aborted.
2. All switch resources are released (e.g. service node session, voice link, subscriber trunks, and network connections).
3. The call is not completed by the switch.

Lost Message:

- Regardless of whether or not the service node receives this message, the call has ended at the switch. Any messages the service node sends regarding the terminated call will be disregarded by the switch.
- Although the service node may consider the session used for the terminated call as being still active, the session is available from the switch's perspective. The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.

7.7 Call Float Request

Identifier: #606

Description: Requests that the switch release the service node from the call and connect the parties if not already connected.

- Preconditions:**
- For a float connection indicator of Unconnected or Connected (in the Call Float Indicator DB), the calling party must be connected and off-hook, and a B Party directory number must be present. For a float connection indicator of Talking (in the Call Float Indicator DB), all parties must be connected and off-hook. Note that the float connection indicator is not applicable when floating to a treatment.
 - Billing must be satisfied if switch restrictions are applied to the call. Note that billing requirements are not checked if floating to treatment.
 - The call cannot be floated if the B party is an Alt DN.
 - The service type of the call (field Service Type in the Call Information DB) cannot be directory assistance.
 - For interlata calls, a carrier must be assigned to the call and the switch must provide operator services for the carrier.
 - If a Routing Index or Routing Number DB is included with this operation, the switch must be using North American translations.
 - If a Routing Index or Routing Number DB is included with this operation, the B party must not be connected at the time of the request.

- Post Conditions:**
- Any switch resources associated with the service node are freed (i.e. session, voice link, dtmf receiver).
 - All function providers are released from the call.
 - Switch based float triggering is in effect.

Side Effects: • Previous timing is cancelled if the call has been class charged collect.

- Response(s):**
- Call Float Success Response
 - Call Float Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Call Float Indicator DB	6 bytes
O	Call In Queue DB	6 bytes
O	Context Block DB	56 bytes
O	Trigger Profile Index DB	14 bytes
O	Sequence Number Check DB	6 bytes
0 to 3	Language DB	6 bytes
O	Billing Acceptance DB	6 bytes
O	Operator Services System Action DB	6 bytes
O ^{SV3}	Charge Status DB	6 bytes
O	Branding Indicator DB	6 bytes
O ^{SV3}	RLT Request DB	6 bytes
0 to 2 ^{SV4}	Conference Party Connection DB	10 bytes
1 of 2 (O) ^{SV6}	Routing Index DB	8 bytes
	Routing Number DB	16 bytes
O	Call Update DB	6 bytes

- Comments:**
1. The Sequence Number Check DB should be included by the service node if it is concerned that it received all messages (e.g. notification of a DTMF event) from the switch prior to the float request. If the sequence number indicated does not agree with the last message sent by the switch, an error is returned (#070F) and the float does not occur. This allows the service node to process the intervening message before trying again. If the Sequence Number Check DB is not included in the request, the switch floats the call without any check for sequence number. This should only be done if the service node does not care if any event (such as an on-hook or DTMF digit) that immediately follows the float request is unprocessed (by a service node). Note that a message 'lost' in this way will have occurred with the switch having sent the event message prior to processing the float request, and thus the switch will have no way of knowing that the event went unprocessed.
 2. A language can be specified for each connection in the Language data block. However, the switch can store either the B party language or the Alt DN language, but not both. The last language DB processed between the B party and the Alt DN will be saved in the switch.
 3. A call can be floated to treatment. In this case, the float connection indicator in the Call Float Indicator DB is not applicable and billing requirements are not checked.
 4. The Billing Acceptance DB and Operator Services System Action DB contain billing information required for calls billed to a calling card, a third number, or billed collect.

5. The Branding Indicator DB is sent by the service node to inform the switch that front-end branding has been provided. The omission of this data block is interpreted by the switch as if it received the data block with a branding status of 0 (call not branded).
6. If the call's elapsed conversation time exceeds the notification time specified in the Trigger Profile Index DB then an error response is returned to the SN.
7. The service node must explicitly request RLT for the call. If the call is RLT eligible, but the service node does not request RLT, then RLT is not attempted.
8. If RLT is requested and the call has a non-nil trigger profile index (e.g. the Call Float Request has the RLT Request DB and the Trigger Profile Index DB), then RLT will be attempted. If the call does not meet all RLT requirements, then trigger using the trigger profile index will be in effect.
9. One Conference Party Connection DB should be sent for each party that the service node wishes to place in a conference at float.
10. If the service node specifies that the "float-style" notification timer should be started at float, any "persistent-style" notification timer that might have been running will be stopped and the "float-style" notification timer will be started with the values supplied by the service node.
11. If the service node specifies that the "persistent-style" notification timer should be started at float, any "persistent-style" notification timer that might have been running will be stopped and the new "persistent-style" notification timer will be started with the values supplied by the service node.
12. If this call is involved in a conference, it may be floated with only a B party connection. Otherwise, both the A and B parties must be present..
13. RLT will not be attempted if the call is involved in a conference.
14. When a Routing Index or Routing Number DB is included with this operation, the Connection ID field of either DB must be set to B Party Connection.
15. If a Routing Index DB is sent with a Routing Index of 2046 (the NIL value), the alternate routing index and alternate routing number associated with the B Party connection will be cleared. The call will route using normal translations.
16. If a Routing Number DB is sent with a Directory Number Digit Count of 0, the alternate routing number and alternate routing index associated with the B Party connection will be cleared. The call will route using normal translations.

- Duplicate Request:**
- If the first request was successful, the session with the service node has ended at the switch. Any messages the service node sends regarding the terminated call will be discarded by the switch. From the switch's perspective, the session is now available to present another call to the service node. The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.
 - If the first request failed, the switch will process the second request as normal and return the appropriate response.

7.7.1 Call Float Success Response

Interpretations: The call has been successfully floated.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.7.2 Call Float Error Response

Interpretations: The call could not be floated. The active service node is still in control of the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0705	Invalid Language
0707	Invalid Number Format
070D	The A party is not connected
070F	Sequence number mismatch
0713	Third number in B party connection
0717	The B party is connected
0718	Billing details incomplete
0719	Invalid trigger profile
071E	Operation not valid for DA
0726	The B party is on-hook
0729	The B party is off-hook
072A	Carrier not served by switch
0732	The A Party is on-hook
0734	Translations failed
073E	Carrier not specified for call
0747	Conversation timing exceeds notification timing
0750	Conversation timing cannot be performed for 'No AMA' calls
0751	Call merge in progress
0752	Unknown or invalid conference id
0753	Voice entity already connected to a conference
0756	Too many voice entities in conference
076C	Invalid Routing Index
076D	North American-only functionality
076E	Connection ID - Party mismatch
0774	Class charge missing but required

Comments: 1. If the error reason returned is Billing Details Incomplete (#0718), the Invalid Billing Details DB is returned which will specify the billing information that is invalid or missing.

2. For error 0726 'The B party is on-hook', the B party cannot be on-hook when a call float with the "talking" option (Float Connection Indicator) is requested. Call float with the "unconnected or connected" option should be requested if the B party is on-hook.
3. For error 0729 'The B party is off-hook' the B party cannot be off-hook when a call float with the "unconnected or connected" option (Float Connection Indicator) is requested. Call float with the "talking" option should be requested if the B party is off-hook.
4. If the language specified in the Language DB is invalid, error 0705 is returned, The language for the connection is set to Nil in the switch.
5. The language for the B party cannot be updated if a third party is connected. Error ID 0713 is returned if an attempt is made to change the B party language while the third party is connected.
6. The language for the third party cannot be updated if the B party is connected. Error ID 0717 is returned if an attempt is made to change the third party language while the B party is connected.
7. If Error ID 0753 is returned, the Conference Party Connection DB is returned. If more than one of the voice entities specified are already connected to a conference, only the first error found is returned.
8. Error ID #076D is sent if the Call Float Request included a Routing Index DB or Routing Number DB and the switch does not use North American translations.
9. Error ID #0774 may be sent for calls using the Automatic Carrier Selection feature that do not have a class charge. An Automatic Carrier Selection call may also result in an Error ID #0734 if it is blocked in switch table CCCSOPTS.

7.8 Call Merge Inform^{SV4}

Identifier: #62A

Description: Used to inform the service node when an A party has been merged into this call from another call.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
M	Directory Number DB	34 bytes
O	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information	6 bytes
O	LNP Information DB	20 bytes
0 to 2	SPID DB	12 bytes
M	Call Timing DB	18 bytes

- Comments:**
1. The Originating Trunk Information data block is optional because there may not be an originating party present.
 2. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
 3. A treatment data block is sent if a treatment is applied to the A party port.
 4. The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.
 5. If LNP information on the A party directory number is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned.
 6. The SPID DB is returned if there is any account owner or billing service provider information for the A party.

- Lost Message:**
- If this message is lost, the service node will not know that an A party has been added to the call. However, the service node will receive all of the data blocks included in this message when call details is requested. Also, future Session Begin Informs and/or Trigger Event Informs will include this information.

7.9 Call Merge Request^{SV4}

Identifier: #629

Description: This operation is used to move the A party of this call into the A party slot of another call that only has a B party. The service node specifies the callid of the call to be merged into.

Preconditions:

- The A party of this call must be connected, offhook.
- The B party of the other call must be connected, offhook.
- The other call must not have an A party connected.
- Both the A party and the B party must be in the same conference.

Post Conditions:

- The A party from this call is placed into the A party slot of the specified call.
- This call is ended and all resources used by this call are freed.
- Conference resources used by this call are either freed or reconfigured to another call.

Side Effects:

- If this call has a B party connected, that connection will be terminated as this call is ended.
- If the other call has a session with a service node at the time of this operation, the service node will receive a Call Merge Inform to tell it of the addition of the A party.
- If the other call is floated at the time of this operation, the service node will find out about the merge when/if the specified call triggers back to the service node.

Response(s):

- Call Merge Success Response
- Call Merge Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Callid DB	8 bytes

Comments:

1. The Callid DB contains the callid of the call the service node wants to merge the A party into.
2. The A party and the target B party must be in the same conference. If either is not in a conference or they are in difference conferences, an error is returned.

Duplicate Request: • If this request is sent twice, the second request will be ignored by the switch since the call will have been ended.

7.9.1 Call Merge Success Response

Interpretations: The A party of this call has been placed in the A party slot of the specified call and this call has ended.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.9.2 Call Merge Error Response

Interpretations: The merge was not performed. Both this call and the specified target call remain unaffected by this operation.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
070D	The A party is not connected
0732	The A party is on-hook
0754	Voice entity not connected to a conference
0757	The B party of the specified call is not connected and offhook
0758	The specified call already has an A party connected
0759	Unknown or invalid callid
075E	Target B party is connected to a different conference than the A party conference

Comments: 1. Error ID #0754 is returned if the following cases:

- The A party is not connected to any conference (the Connection ID DB is returned for the A party)
- Target B party is not connected to any conference (the Connection ID DB is returned for the B party in the target call)

2. Error ID #075E is returned if the target B party is connected to a different conference than the A party conference (the Conference ID DB is returned for the B party in the target call).

7.10 Call Resume Request^{SV3}

Identifier: #620

Description: Requests that the switch release the service node and its resources from an OSSAIN preprocessing call session, without terminating the call.

Preconditions: • An OSSAIN preprocessing call session must have been established between the service node and the switch.

Post Conditions: • Any switch resources associated with the node are freed (i.e. session, voice link).
• The call continues to either a TOPS operator or automated system.

Side Effects: • If the optional Branding Indicator DB is not sent, the switch assumes that the service node did not provide front-end call branding during the session. If the branding indicator is improperly set, subsequent operator screen displays indicating the call's branding status may be erroneous. Also, a call could potentially be branded twice, once at the service node and once by the switch.

Response(s): • Call Resume Success Response
• Call Resume Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	Branding Indicator DB	6 bytes
O	Context Block DB	56 bytes
O	Text DB	26 bytes
O	Operator System DB	6 bytes

Comments: 1. None.

Duplicate Request: • If the first request was successful, the call has resumed under switch control and the service node is no longer involved. Any messages the service node sends regarding the call will be ignored by the switch. From the switch's perspective the session is now available to present another call to the service node. The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.

- If the first request failed, the switch will process the duplicate as normal.

7.10.1 Call Resume Success Response

Interpretations: The switch has successfully resumed control of an OSSAIN preprocessed call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.10.2 Call Resume Error Response

Interpretations: The switch could not resume control of an OSSAIN call as requested by the service node.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0609	Operation is restricted for standard session
060B	SOC option not turned on

Comments: 1. If the service node sends a Call Resume Request during a standard OSSAIN call, errorID 0609 will be returned. The Call Resume Request operation is only allowed on preprocessing calls.

7.11 Calling Number BNS Query Request^{SV7}

Identifier: #635

Description: A request to perform a BNS query on the calling number to check for any billing restrictions.

Preconditions: • The calling and called party directory numbers must be present.

Post Conditions: • A BNS query was successfully launched on the calling number.

Side Effects: • None.

Response(s): • Calling Number BNS Query Success Response
• Calling Number BNS Query Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments: 1. None.

Duplicate Request: • The same result should be generated and returned to the SN.

7.11.1 Calling Number BNS Query Success Response

Interpretations: The BNS query on the calling number was successfully launched.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	BNS Query Result DB	6 bytes
M	Billing Number Validation DB	10 bytes
0 to 2	SPID DB	12 bytes

Comments: 1. The SPID data block will be sent if SPID information is returned from the BNS query.

7.11.2 Calling Number BNS Query Error Response

Interpretations: The switch was unable to launch a BNS query on the calling number.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
060B	SOC option not turned on
0709	No calling number
070A	No forward number
0713	Third number in B party connection
073B	Conversation timing in effect

Comments: 1. None.

7.12 Carrier Assignment Request

Identifier: #612

Description: Assign the specified carrier to the call.

Preconditions: • None.

Post Conditions: • A carrier is associated with the call.

Side Effects: • May reset the class charge and/or result in different billing restrictions for the call, reflected in optional data blocks received in the success response.

Response(s): • Carrier Assignment Success Response
• Carrier Assignment Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Carrier Number DB	8 bytes

Comments: 1. None.

7.12.1 Carrier Assignment Success Response

Interpretations: The carrier is updated for the call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Carrier DB	20 bytes
O	Billing Information DB	10 bytes
O	Class Charge DB	10 bytes

- Comments:**
1. If the switch provides operator services for the carrier, then the call can be floated by the service node. If the switch does not provide operator services, the service node must transfer the call to the carrier via the Transfer to Carrier Request.
 2. The Billing Information DB is sent if the request results in different billing restrictions for the call. If the DB is not sent, assume the pre-existing billing restrictions still apply.
 3. The Class Charge DB is sent if the new carrier assignment results in a change in the status of the class charge.

7.12.2 Carrier Assignment Error Response

Interpretations: The carrier for the call was not updated.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0717	The B party is connected
0720	Invalid carrier
072B	Carrier not allowed for this call
0735	Translations failed, treatment applied
073D	Carrier ID is restricted

- Comments:**
1. A ReturnError of “Translations failed, treatment applied” (0735) will return a Treatment DB indicating the specific treatment received.
 2. Any previous carrier information stored at the switch is set to nil if a 07xx error is returned (except 0717), the service type is not DA, and interlata is not restricted to the original carrier.

7.13 Class Charge Request

Identifier: #603

Description: Requests the switch to enter the class charge and handling data for the call. This operation is also used to indicate if the service node wants the switch to validate the billing number.

Preconditions:

- The calling party directory number is required unless the class charge is being set to unspecified.
- The forward party directory number is required if:
 - Service Type is Toll and Assist
 - A query to the LIDB is needed:
 - class charge is collect
 - class charge is station special calling, bill to alternate DN
 - class charge is station special calling, bill to non-custom card
- If the billing number is not the calling party's directory number or the forward party's directory number, then it is required.

Post Conditions:

- New class charge is stored for the call.
- If validation was requested, then billing validation was attempted.

Side Effects: • Switch's view of the class charge is changed to reflect the new class charge.

Response(s):

- Class Charge Success Response
- Class Charge Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Class Charge DB	10 bytes
O	Billing Number Input DB	6 bytes

Comments:

1. This is a restricted operation. If the service node requested validation, the switch will not accept any requests until the validation is complete. The service node should wait until it receives the response for the Class Charge Request.
2. The class charge can be set to nil by sending the Class Charge DB with the Handling Data and Billing Data set to Unspecified and the Validation Request set to Not Applicable.

3. All class charges must be validated except when the service node has previously sent a custom calling card as the billing number to the DMS or is setting the class charge to nil. In all other cases, if the service node request that the DMS does not validate the billing, the DMS will return the “invalid class charge” error.
4. Custom calling cards cannot be validated by the DMS. If the service node requests for the DMS to validate a custom calling card in a class charge request, the DMS will return an “invalid class charge” error.
5. The Billing Number Input DB contains billing information required for calls billed to a calling card, third number, or billed collect.
6. If the billing number is a custom calling card and calling card restrictions apply (refer to the Allowed Billing Set field), the class charge to the custom card will not be blocked by the switch.

7.13.1 Class Charge Success Response

Interpretations: A class charge has been successfully entered for the call. If the service node requested validation, the results are returned.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Billing Number Validation DB	10 bytes
1 of 2 (O)	BNS Query Result DB	6 bytes
	CCV Query Result DB	6 bytes
O ^{SV3}	LNP Information DB	20 bytes

- Comments:**
1. The Billing Number Validation DB is included in the response only if the Class Charge Request indicated that validation was requested.
 2. The BNS Query Result DB is included if all of the following is true:
 - the Class Charge Request indicated that validation was requested
 - the DMS successfully received a BNS query response from a LIDB database
 - the billing number is a line number
 - an error was found with the billing number (i.e. Validation Result field in the Billing Number Validation DB is not Billing Accepted).
 3. The CCV Query Result DB is included if all of the following is true:
 - the Class Charge Request indicated that validation was requested
 - the DMS successfully received a CCV query response from a LIDB database
 - the billing number is a card number.
 4. If LNP information (for alternate directory number or line based calling card) is available through datafill or as a result of an LNP query, the LNP Information DB is returned.

7.13.2 Class Charge Error Response

Interpretations: There was a problem with the class charge request. If the switch attempted to change the class charge but failed (i.e. the error reason returned is not 06xx), the class charge is set to unspecified. If the class charge failed, then the billing number is not validated.

If the class charge passed and an error response was returned, then there was a problem with the validation request. The billing number was not validated.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0706	Invalid number type
0709	No calling number
070A	No forward number
070B	No billing number
070C	Billing number is hotlisted
070E	Invalid class charge
0712	Call is restricted for this billing method
0713	Third number in B party connection
0714	800+ calls must be auto collect
071D	Charge class invalid for call
0739	Auto collect allowed only for 800+ calls.
073B	Conversation Timing In Effect
0760	Previous class charge information cannot be set by the service node

Comments: 1. If an error id of 73B (Conversation Timing In Effect) is returned, then call timing, based upon the connection of one of the parties of the call, has been started by the switch. To cancel timing, the service node must request to release the party responsible for call timing, or send in a Conversation Timing Request to stop timing.

7.14 Conference Add Request^{SV4}

Identifier: #625

Description: This operation is used to move voice connections (i.e. A party, B party, or SN voice link) from the call to a conference.

Preconditions:

- A conference must have been previously created (though not necessarily by this call).
- Any parties and/or the SN voice link must be connected to the call (either onhook or offhook) before they can be added to the conference by this operation.

Post Conditions:

- Conference resources are allocated in the switch and the specified parties and/or the SN voice link are added to the conference.

Side Effects:

- If a party and/or SN voice link is added to a conference, that entity will not be able to hear/speak to any party and/or SN voice link that is not added to the conference. In other words, a party and/or SN voice link is either connected to other entities in a conference or it is connected to other entities in this call, but not both.

Response(s):

- Conference Add Success Response
- Conference Add Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Conference ID DB	8 bytes
0 to 2	Connection ID DB	6 bytes
O	Conference SN Voice Connection DB	6 bytes

Comments:

1. One Connection ID DB should be sent for each party that the service node wishes to add to the conference. If two Connection ID DBs are sent for the same party, the extra Connection ID DB will be ignored.
2. The Conference SN Voice Connection DB allows the SN to specify whether or not the SN voice link is to be placed in the conference. If the SN voice link is not placed in the conference, it will remain connected to the call (which may or may not still contain parties to talk to, depending on whether the service node moved them into the conference or left them in the call as well).

3. The service node has the capability to move voice entities between “the call” and “the conference” as often as it sees fit (via Conference Add Request and Conference Remove Requests).

Duplicate Request: • If this request is sent twice, the second request will fail because the specified voice entities will have already been added to the conference.

7.14.1 Conference Add Success Response

Interpretations: Conference resources have been allocated (if necessary) in the switch and the specified parties and/or the SN voice link have been added to the conference.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.14.2 Conference Add Error Response

Interpretations: None of the requested voice entities were added to the conference.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
070D	The A party is not connected
0727	The B party is not connected
072F	There is no voice connection for this session
0752	Unknown or invalid conference id
0753	Voice entity already connected to a conference
0756	Too many voice entities in conference

- Comments:**
1. Conference calls can contain at most four voice entities (either four subscribers or three subscriber parties and at most one SN voice link). If any of these conditions are exceeded by this operation, Error ID 0756 is returned.
 2. If Error ID 0753 is returned, either the Conference Party Connection DB (for subscriber parties) or the Conference SN Voice Connection DB (for SN voice link) is returned. If more than one of the voice entities specified are already connected to a conference, only the first error found is returned.

7.15 Conference Create Request^{SV4}

Identifier: #624

Description: This operation is used to create a conference. As part of creating the conference, the service node may choose to add the A and/or B parties to the conference, as well as adding the SN voice link to the conference (if these connections exist).

Preconditions:

- There cannot be a conference call already associated with this call (i.e. a second conference call cannot be created in this call until the first conference is ended via a Conference Release Request).
- Any parties and/or the SN voice link must be connected to the call (either onhook or offhook) before they can be added to the conference by this operation.

Post Conditions:

- Conference resources are allocated in the switch and the specified parties and/or the SN voice link are added to the conference.

Side Effects:

- If a party and/or SN voice link is added to a conference, that entity will not be able to hear/speak to any party and/or SN voice link that is not added to the conference. In other words, a party and/or SN voice link is either connected to other entities in a conference or it is connected to other entities in this call, but not both. The SN has the capability to move voice entities between “the call” and “the conference” as often as it sees fit (via Conference Add Request and Conference Remove Requests).

Response(s):

- Conference Create Success Response
- Conference Create Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
0 to 2	Connection ID DB	6 bytes
O	Conference SN Voice Connection DB	6 bytes

Comments:

1. One Connection ID DB should be sent for each party that the SN wishes to add to the conference. If two Connection ID DBs are sent for the same party, the extra Connection ID DB will be ignored.
2. The Conference SN Voice Connection DB allows the SN to specify whether or not the SN voice link is to be placed in the conference. If the SN voice link is not placed in the conference, it will remain connected to

the call (which may or may not still contain parties to talk to, depending on whether the SN moved them into the conference or left them in the call as well).

Duplicate Request: • If this request is sent twice, the second request will fail because a conference has already been created in this call. The SN will receive the Conference ID DB assigned for the first conference in the Return Error response.

7.15.1 Conference Create Success Response

Interpretations: Conference resources have been allocated in the switch and the specified parties and/or the SN voice link have been added to the conference.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Conference ID DB	8 bytes

Comments: 1. The Conference ID returned will be required on all future requests relating to this conference.

7.15.2 Conference Create Error Response

Interpretations: The conference was not created. All parties and the SN voice link remain unchanged.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
060D	Invalid operation for calls involved in a simultaneous interaction
070D	The A party is not connected
0727	The B party is not connected
072F	There is no voice connection for this session
0755	Conference already exists

- Comments:**
1. The Conference ID DB is returned with Error ID 0755 to indicate the conference ID of the existing conference.
 2. If Error ID 060D is returned, the passive function provider must be released before creating a conference.

7.16 Conference Details Request^{SV4}

Identifier: #628

Description: This operation is used to obtain information about ALL voice entities associated with a conference, not just the ones that happen to be in this call.

Preconditions: • A conference must have been previously created (though not necessarily by this call).

Post Conditions: • The conference and all its voice entities remain unaffected by this operation.

Side Effects: • None.

Response(s): • Conference Details Success Response
• Conference Details Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Conference ID DB	8 bytes

Comments: 1. None.

Duplicate Request: • If this request is sent twice, the details will be sent again.

7.16.1 Conference Details Success Response

Interpretations: Conference details about all voice entities in the conference are sent to the service node.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
0 to 4	Conference Party Connection Information DB	12 bytes
O	Conference SN Voice Connection Information DB	8 bytes

- Comments:**
1. One Conference Party Connection Information DB is sent for each party connected to the conference. The Conference Voice Connection Information DB is sent if an SN voice link is connected to a conference.
 2. If no data blocks are sent, then no voice entities are connected to the conference.

7.16.2 Conference Details Error Response

Interpretations: The details about the specified conference could not be obtained.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0752	Unknown or invalid conference

Comments: 1. None.

7.17 Conference Release Request^{SV4}

Identifier: #627

Description: This operation is used to end an association between this call and a specified conference.

Preconditions:

- A conference must have been previously created (though not necessarily by this call).
- This call must have been associated with the conference (either by creating it or by adding a voice entity to it).

Post Conditions:

- All conference resources used by this call are released or reconfigured to another call in the conference.
- All voice entities associated with this call are moved from the conference back into the call (if not already moved)

Side Effects:

- None of the voice entities in this call will be able to communicate with any of the voice entities (from other calls) that may still be in the conference. However, the voice entities in this call will still be connected to each other.

Response(s):

- Conference Release Success Response
- Conference Release Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Conference ID DB	8 bytes

Comments:

1. This operation only ends this call's association with a conference. It does not necessarily end the conference itself. Parties from other calls may still be connected to the conference. If this is the last call associated with the conference, all conference resources are freed and the conference is ended.

Duplicate Request:

- If this request is sent twice, the second request will fail because the specified conference will be unknown by this call.

7.17.1 Conference Release Success Response

Interpretations: Conference resources associated with this call have been freed or reconfigured to other calls in the conference. All voice entities in this call have been removed from the conference. If this was the last call associated with the conference, the conference is ended and all conference resources are reclaimed by the switch.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.17.2 Conference Release Error Response

Interpretations: This call's association with the conference is unchanged and all voice entities in this call that were connected to the conference are still connected to it.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0752	Unknown or invalid conference id

Comments: 1. None.

7.18 Conference Remove Request^{SV4}

Identifier: #626

Description: This operation is used to move voice connections (i.e. A party, B party, or SN voice link) from a conference to the call.

Preconditions:

- A conference must have been previously created (though not necessarily by this call).
- All voice entities that are being removed must be connected to a conference.

Post Conditions:

- Conference resources are deallocated (if appropriate) in the switch and the specified parties and/or the SN voice link are removed from the conference.

Side Effects:

- If a party and/or SN voice link is removed from a conference, that entity will not be able to hear/speak to any party and/or SN voice link that is still in the conference. However, that entity will now be connected to other voice entities in the all (if any).

Response(s):

- Conference Remove Success Response
- Conference Remove Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
0 to 2	Connection ID DB	6 bytes
O	Conference SN Voice Connection DB	6 bytes

Comments:

1. One Connection ID DB should be sent for each party that the service node wishes to remove from the conference. If two Connection ID DBs are sent for the same party, the extra Connection ID DB will be ignored.
2. The Conference SN Voice Connection DB allows the SN to specify whether or not the SN voice link should be removed from the conference.

Duplicate Request:

- If this request is sent twice, the second request will fail because the specified voice entities will have already been removed from the conference.

7.18.1 Conference Remove Success Response

Interpretations: Conference resources have been deallocated (if necessary) in the switch and the specified parties and/or the SN voice link have been removed from the conference.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.18.2 Conference Remove Error Response

Interpretations: None of the requested voice entities were removed from the conference.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0754	Voice entity not connected to a conference

Comments: 1. If Error ID 0754 is returned, either the Connection ID DB (for subscriber parties) or the Conference SN Voice Connection DB (for SN voice link) is returned. If more than one of the voice entities specified were not connected to a conference, only the first error found is returned.

7.19 Connect DN Request

Identifier: #60A

Description: Connects to the specified DN. The requested speech status is indicated in the Speech Path Status field of the Speech Path DB. Translations are performed for this operation. They may fail.

Preconditions:

- The directory number for the specified Connection ID must have been stored for the call.
- If a Routing Index or Routing Number DB is included with this operation, the switch must be using North American translations.

Post Conditions:

- The specified DN is outpulsed.

Side Effects:

- None.

Response(s):

- Connect DN Success Response
- Connect DN Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Speech Path DB	6 bytes
O ^{SV4}	Conference ID DB	8 bytes
1 of 2 (O) ^{SV6}	Routing Index DB	8 bytes
	Routing Number DB	16 bytes
O ^{SV7}	LNP Enhancements DB	6 bytes

Comments:

1. When the service node requests a connection, the switch is in a “Restricted Input Mode” while waiting for trunk messaging. This means that the switch will not accept any requests until the connection attempt ends.
2. If the Conference ID DB is sent, the party will be connected to the specified conference. Otherwise, the party will be connected to this call.
3. When a Routing Index or Routing Number DB is included with this operation, the Connection ID field of either DB must match the value of the Connection ID field of the Speech Path DB. Also, the real directory number associated with the connection must successfully pass translations.
4. If a Routing Index DB is sent with a Routing Index of 2046 (the NIL value), the alternate routing index and alternate routing number associ-

ated with the specified connection will be cleared. The call will route using normal translations.

5. If a Routing Number DB is sent with a Directory Number Digit Count of 0, the alternate routing number and alternate routing index associated with the specified connection will be cleared. The call will route using normal translations.

Duplicate Request: • Either a 'Directory number connection in progress' error or a 'Party already connected' error will be returned, depending on whether the prior request is still in the process of being completed or not.

7.19.1 Connect DN Success Response

Interpretations: The DN specified in the request has been translated, outpulsed, and connected. The status of the connection (e.g. off-hook, busy, etc.), however, should not be assumed until a Connection Status Inform is received (for ISUP trunks, the SS7 Backward Call Indicators ‘may’ indicate ringing, etc., but the party still cannot be determined to be off-hook until a Connection Status Inform message indicates that event). The service node is still involved in the call and is in control.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Connection Progress DB	8 bytes
O ^{SV3}	LNP Information DB	20 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes

- Comments:**
1. Success for non-ISUP trunks is concluded upon successful outpulsing. For ISUP trunks, a success response is returned upon the switch receiving an ACM message (when message is expected by the switch).
 2. After receiving a success response, any further requests for connection to the same party (e.g. A Party, B Party, Alt. DN) will be invalid until the service node has successfully sent a release request for that party.
 3. If LNP information on the directory number is available as a result of an LNP query, the LNP Information DB is returned.
 4. When connecting to the B party or alternate DN, the Terminating Trunk Information DB is returned to the service node.

7.19.2 Connect DN Error Response

Interpretations: The specified DN could not be connected to the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0707	Invalid Number Format
0708	No DN digits entered
072A	Carrier not served by switch
0733	Party already connected
0734	Translations failed
0735	Translations failed, Treatment applied
0736	Outpulsing failure, Treatment applied
0737	Release with cause, Treatment applied
0738	Outpulsing failure, No treatment applied
0752	Unknown or invalid conference id
0756	Too many voice entities in conference
076C	Invalid Routing Index
076D	North American-only functionality
076E	Connection ID - Party mismatch
0774	Class charge missing but required

- Comments:**
1. A ReturnError of 'Translations failed, Treatment applied' (0735) will return a Treatment DB indicating the specific treatment received.
 2. A ReturnError of 'Outpulsing failure, Treatment applied' (0736) will return a Treatment DB indicating the specific treatment received.
 3. A ReturnError of 'Release with Cause, Treatment applied' (0737) will return both a Release Cause data block, which provides the actual release cause information, and a Treatment data block, which will indicate what treatment the switch is applying. This error, and its associated informa-

tion, can only be returned when the request for a connection is on an ISUP trunk.

4. A ReturnError of 'Outpulsing failure, No treatment applied' (0738) will return no data blocks.
5. The Speech Status part of the Connect DN Request will always succeed, provided a valid party/DN is selected (i.e. there are no additional Error-IDs corresponding to Speech Status failures; invalid party/DN is equally an error for Connecting)..
6. Error ID #076D is sent if the Connect Dn Request included a Routing Index DB or Routing Number DB and the switch does not use North American translations.
7. Error ID #0774 may be sent for calls using the Automatic Carrier Selection feature.

7.20 Connect To Treatment Request^{SV6}

Identifier: #632

Description: Connect the specified party to the treatment indicated in the message.

- Preconditions:**
- If the party to receive the treatment is the A-party, then a B- or Alt-party cannot be connected.
 - If the party to receive the treatment is the B- or Alt-party, then the A-party cannot be connected.
 - The party to receive the treatment must be connected.

Post Conditions:

- The party is connected to treatment. The SN is still in control of the call.

Side Effects:

- None.

Response(s):

- Connect To Treatment Success Response
- Connect To Treatment Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Treatment DB	6 bytes
O ^{SV4}	Conference ID DB	8 bytes

Comments: 1. This request requires parallel datafill between the service node and the switch for the treatment numbers.

Duplicate Request:

- If the first request was successful, a 'Party already connected' error will be sent. If the first request was not successful, the operation will be processed as normal by the switch.

7.20.1 Connect To Treatment Success Response

Interpretations: The switch has connected the appropriate party to treatment.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.20.2 Connect To Treatment Error Response

Interpretations: The party could not be connected to the treatment specified in the request operation.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
070D	The A Party is not connected
071A	Invalid treatment
071E	Operation not valid for DA
0727	The B Party is not connected
0733	Party already connected
0752	Unknown or invalid conference id
0756	Too many voice entities in conference

7.21 Connection Status Inform

Identifier: #609

Description: The switch has detected a change in the status of the specified connection.

Preconditions: • A connection was up for the call in progress.

Post Conditions: • Parties may have joined or exited the call. The trunk will have been released for a Connection Event of 'Released'.

Side Effects: • None.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Connection Event DB	6 bytes
O	Release Cause DB	6 bytes
O	Treatment DB	6 bytes
O ^{SV3}	Call Timing	18 bytes

- Comments:**
1. This message informs of on-hooks, off-hooks, hook flash, and released. DTMF events are handled through the DTMF Digit Detected Inform operation/message.
 2. The Treatment DB will be present if, as the result of a network event, the switch is applying treatment. The presence of a Release Cause DB is optional for calls on an ISUP trunk, and will not be present for calls on a non-ISUP trunk.
 3. The Call Timing DB is sent if answer for the call has been detected by the switch.

Lost Message: • The indication of an event will not get passed to the service node. Hook-flashes may get repeated. Lost off-hooks will probably result in subsequent party on-hook. Lost on-hooks and releases may result in switch call timers timing out, resulting in the call being taken down.

7.22 Conversation Timing Request

Identifier: #621

Description: Requests that the switch start or stop conversation timing for a call.

Preconditions: • None.

Post Conditions: • None.

Side Effects: • None.

Response(s): • Conversation Timing Success Response
• Conversation Timing Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Conversation Timing Indicator DB	6 bytes

- Comments:**
1. Conversation timing may be stopped or started throughout the life of the call. Starting conversation timing on a call that already has existing conversation time has the effect of continuing accumulation of conversation time. Stopping conversation time on a call that has existing accrued conversation time has the effect of stopping the accumulation of conversation time.
 2. Stopping conversation timing will suspend the notification timer if it is running and the notification timer type is conversation. The timer will automatically resume if timing is subsequently started again.

Duplicate Request: • If the first request was successful and the call has started timing, a subsequent request to start timing is equivalent to a resume timing and has no effect on call timing. If a stop timing request is sent to the switch and is successful, a subsequent stop timing would have no effect since call timing was already stopped.

7.22.1 Conversation Timing Success Response

Interpretations: The switch has successfully processed the conversation timing request.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.22.2 Conversation Timing Error Response

Interpretations: The conversation timing request could not be processed.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
073F	Timing has started
0750	Conversation timing cannot be performed for 'No AMA' calls
075F	Conversation timing not started

Comments: 1. None.

7.23 DTMF Digit Detected Inform

Identifier: #60E

Description: Sent by the switch when a digit is received by the DTMF receiver that is monitoring the party connected to it.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

M/O	Data Blocks	Byte Length
M	DTMF Digit DB	6 bytes

Comments: 1. Only a single digit will be sent per message. If the party enters a stream of digits, only a single digit will be sent at a time with this inform operation. For example, if the party enters 3 digits, the switch will send 3 inform messages. Each message will have a single digit.

Lost Message:

- If the service node is expecting this message and it gets lost, the service node will not receive the digit it was expecting. The service node may timeout waiting for the digit. It can then respond accordingly (e.g. replay the announcement to the subscriber to make a selection).
- The service node may not be expecting any digits but the subscriber keys some digits. If this message is lost, then the subscriber may key the digits again or may eventually go on-hook.
- Note that the service node will be unable to differentiate between this message being lost and the subscriber not entering any digits.

7.24 DTMF Receiver Release Request

Identifier: #617

Description: The service node is requesting to release the DTMF receiver connected to the call.

Preconditions: • A DTMF receiver is associated with the call.

Post Conditions: • The DTMF receiver is released and no longer connected to the call.

Side Effects: • None.

Response(s): • DTMF Receiver Release Success Response
• DTMF Receiver Release Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments: 1. None.

Duplicate Request: • If there is no DTMF receiver connected to the call, a success response (ReturnResult) will be returned.

7.24.1 DTMF Receiver Release Success Response

Interpretations: The switch has successfully released the DTMF receiver.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.24.2 DTMF Receiver Release Error Response

Interpretations: The DTMF receiver could not be released.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress

Comments: 1. None.

7.25 Directory Number Request

Identifier: #607

Description: A request to enter or update the directory number information for the specified connection (A Party, or B Party).

Preconditions:

- For any update to B party information to be accepted, a Third party must not be connected.
- Only a change of language may be requested if the specified party is already connected.
- If a Routing Index or Routing Number DB is included with this operation, the switch must be using North American translations.

Post Conditions:

- The directory number is stored for the specified connection.
- The language is updated for the specified party.

Side Effects:

- The class charge is reset to unspecified (unless only language or presentation indicator is updated).
- The rating information is updated based on the directory number.

Response(s):

- Directory Number Success Response
- Directory Number Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Directory Number Update DB	30 bytes
O	Originating Trunk Information DB	8 bytes
1 of 2 (O) ^{SV6}	Routing Index DB	8 bytes
	Routing Number DB	16 bytes
O ^{SV7}	LNP Enhancements DB	6 bytes

Comments:

1. If the specified party (and the Third party, as well, in the B party case) is not connected, then changes to the party's language and DN will be processed. If the specified party is connected, only the party's language may be changed.
2. To change the language of a connected party, the service node must include the previously established DN for that party, otherwise the update will be treated as an attempt to change the directory number and an error response will be returned.

3. A DN Digit Count of 0 in the Directory Number Update DB clears the DN for the indicated party.
4. A language of unknown will reset the language of the specified party to nil language.
5. The B party and Third party share a single language setting, but although this is the case, language updates still must be done through the appropriate operation (e.g. If a Third party is connected, the Billing Number Request operation should be used to update the language of the Third party; a Directory Number Request to change the B party language at this time will return an error).
6. Directory Numbers for domestic calls under the North American numbering plan must be no greater than 10 digits (i.e. for Directory Number Type of Domestic, and switch data-filled as North American Dialing Plan, the Directory Number Digit Count, for A or B parties, should be between 0 and 10). Directory numbers for foreign calls under the North American numbering plan (B party only) have a maximum of 15 digits. Directory numbers for all calls and parties under the Open numbering plan have a maximum of 18 digits.
7. The DN Type for the A party must be identified as Domestic, otherwise an Invalid Number Type error (ErrorID #0706) will be returned.
8. Updates to CLI are currently not supported, and thus the CLI fields are ignored by this operation.
9. Directory numbers should not include long distance or international access digits. Country code digits should be included for foreign numbers.
10. This request may cause an LNP query to be launched. This will put the switch in a "Restricted Input Mode" for this call. The service node should wait for the response before sending in other requests.
11. The Originating Trunk Information DB should be sent if updating the A party DN and the A party will not be connected in this session and is already connected in another session (ex: the A party is connected in the first leg of a 3-way call scenario and this session is the session initiated by the service node for the C party). This information is needed to do equal access translations on the B party. The A party DN information should be sent before the B party DN information.
12. When a Routing Index or Routing Number DB is included with this operation, the party in the Connection ID field of either DB must match the Party field of the Directory Number Update DB. Also, the DN in the Directory Number Update DB must successfully pass translations (DN Digit Count cannot equal zero).

13. If a Routing Index DB is sent with a Routing Index of 2046 (the NIL value), the alternate routing index and alternate routing number associated with the DN connection will be cleared.
14. If a Routing Number DB is sent with a Directory Number Digit Count of 0, the alternate routing number and alternate routing index associated with the DN connection will be cleared.
15. The service node can change the presentation indicator status of an existing DN by including the previously established DN for that party along with the new status in the request.

7.25.1 Directory Number Success Response

Interpretations: The directory number information has been stored against the specified party. The class charge for the call is reset to unspecified (unless only language is updated). The service node should send in the class charge for the call via the Class Charge Request operation.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Billing Information DB	10 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
O	Treatment DB	6 bytes
O	Originating Station Information DB	22 bytes
M	LATA Information	6 bytes
O ^{SV3}	LNP Information DB	20 bytes
0 to 2 ^{SV3}	SPID DB	12 bytes
O	Directory Number Information DB	8 bytes

- Comments:**
1. If the Directory Number passes translations and a carrier number is present, the Carrier DB is sent with the response. Directory numbers that are cleared or result in treatment do not result in a Carrier DB being returned. The Carrier DB is not sent if updating the A party directory number.
 2. If the Directory Number translates to a treatment, the treatment value is returned. Treatment is not automatically applied in this case, rather the service node may connect to the treatment (using a Connect DN Request), or it may retry the Directory Number Request with a different number.
 3. Subsequent Directory Number Requests for a party will overwrite earlier information (assuming not connected). To update a single field (i.e. Language) the earlier information (such as Directory Number) should be resent if it is not intended to change.
 4. The Originating Station Information DB will be returned for an A party DN request, providing associated origination information about that directory number.
 5. If LNP information on the directory number is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned.
 6. The SPID DB is returned if there is account owner or billing service provider SPID information for the A party.

7. The Carrier data block is sent for carrier calls or calls arriving on a carrier trunk. The NBEC DB is sent for calls completed by the operating company and is based on the A party DN.
8. The Directory Number Information DB is sent when the B party DN is updated. This information can be used with the Lata Status field. The following are valid combinations:
 - Intralata Local
 - Intralata Non-local
 - Interlata Non-local
 - (Interlata Local is not a valid combination. If the call is interlata, then the DN type is set to non-local.)

7.25.2 Directory Number Error Response

Interpretations: There was a problem with the directory number information for the specified party. Unless otherwise indicated for a given ErrorID, #07xx errors will result in the resetting of the class charge for the call to 'unspecified' (i.e. a new Class Charge Request is required to reestablish a class charge), and the clearing of the directory number information for the specified party (unless the party is connected, in which case no change occurs to the directory number information). The generic #06xx errors do not result in the resetting of the class charge.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0705	Invalid language
0706	Invalid number type
0707	Invalid number format
0709	No calling number
0713	Third number in B party connection.
0716	The A party has been released.
0717	The B party is connected
0724	Call is setup to be transferred to a carrier
0733	Party already connected
0734	Translations failed
076A	Invalid trunk group and member
076C	Invalid Routing Index
076D	North American-only functionality
076E	Connection ID - Party mismatch

- Comments:**
1. An ErrorID #0705 will result in the current language being set to nil language. The class charge and directory number information will not be reset if the requested change was only to update a party's language.
 2. An ErrorID of #0706 indicates that a directory number type of Foreign was specified for the A party (i.e. the DN Type for the A party must be identified as Domestic.).
 3. An ErrorID of #0713 will be returned for a request to update B party information if a Third party is connected. The B party and Third party share a number of physical and logical structures. Updates to the B party directory number information are not allowed while the Third party is connected. The class charge is reset, but the directory number information is not.
 4. Following a release of an A party that has dialed in (i.e. not a service node initiated call), the change of either the A or B party numbers is disallowed, and will return an ErrorID of #0716. The A party may however be reconnected. An Error ID #0716 will reset the class charge to unspecified, and, for a B party request, clear the B party number.
 5. An Error ID #0733 will be returned if the DN Request is sent to modify the directory number for a party that is already connected.
 6. If Error ID #076A or #0709 is returned, the directory number is updated for the party. However, the originating trunk information is not updated.
 7. If Error ID #0717 is returned, then the A party DN information cannot be updated if the B party is connected. This may change the LATA status of the call.
 8. If Error ID #0709 is returned, the originating trunk information cannot be updated if there is no A party DN.
 9. Error ID #076D is sent if the Directory Number Request included a Routing Index DB or Routing Number DB and the switch does not use North American translations.

7.26 End Call Request

Identifier: #608

Description: Indicates the service node wishes to abort the call and release all switch resources associated with the call and the service node.

Preconditions: • A call session must have been established between the active service node and the switch.

Post Conditions: • The call is terminated.

Side Effects: • All switch resources are released (e.g. service node session, voice link, subscriber trunks, and network connections).
• External switch queries are canceled if in progress.
• All function providers are released from the call

Response(s): • End Call Success Response
• End Call Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	End Call Handling DB	6 bytes
O	Billing Acceptance DB	6 bytes
O	Operator Services System Action DB	6 bytes
O	Sequence Number Check DB	6 bytes

- Comments:**
1. The call is not completed by the switch.
 2. The End Call Handling DB indicates if the service node is ending the call due to normal call handling or as a result of an error (i.e. the call is in an unrecoverable state).
 3. The Billing Acceptance DB and Operator Services System Action DB contain billing information required for calls billed to a calling card, a third number, or billed collect.
 4. If this call was involved in a conference, all conference resources associated with this call are either freed or reconfigured to another call involved in the conference. If this is the last call in the conference, all switch resources associated with the conference are freed.
 5. The Sequence Number Check DB should be included by the service node if it does not want to end the call if a call merge is in progress (i.e. the switch is in the process of sending a Call Merge Inform or sent the Call

Merge Inform at the same time the service node is sending the End Call Request). If a call merge is in progress and the sequence number in the data block indicates the service node has not received the Call Merge Inform, then an error is returned (#0751) and the call does not end. This allows the service node to process the Call Merge Inform. If the Sequence Number Check DB is not included in the request, the switch ends the call without any check for sequence number.

- Duplicate Request:**
- If the first request was successful, the call has ended at the switch. Any messages the service node sends regarding the terminated call will be ignored by the switch. From the switch's perspective, the session is now available to present another call to the service node. The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.
 - If the first request failed, the switch will process the duplicate request as normal.

7.26.1 End Call Success Response

Interpretations: The switch has successfully ended the call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O ^{SV3}	Call Timing DB	18 bytes

Comments: 1. None.

7.26.2 End Call Error Response

Interpretations: The switch could not end the call requested by the service node.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0751	Call merge in progress

- Comments:**
1. Error ID #751 is returned only if the Sequence Number DB is sent in the End Call Request and it indicates the service node has not had a chance to process the Call Merge Inform.
 2. If the service node wishes to end the call regardless of whether or not a call merge is in progress, the Sequence Number DB should not be sent. The switch will then end the call.

7.27 Estimate of Charges Request^{SV6}

Identifier: #631

Description: A request for an estimate of charges.

Preconditions:

- SOC ENSV0018 must be ON.
- The call must have valid billing.
- The call cannot use external rating and the internal rating tables must be completely datafilled.

Post Conditions:

- None.

Side Effects:

- None.

Response(s):

- Estimate of Charges Success Response
- Estimate of Charges Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	Estimate of Charges Alternate Date DB	6 bytes
O	Estimate of Charges Alternate Time DB	6 bytes
O	Estimate of Charges Duration DB	6 bytes

Comments:

1. If the Estimate of Charges Alternate Time DB is not sent, the switch will use the current switch time when calculating charges.
2. If the Estimate of Charges Alternate Date DB is not sent, the switch will use the current switch date when calculating charges.
3. If the Estimate of Charges Duration DB is not sent, the switch will send the charges for the initial and subsequent periods in the response.
4. See the Estimate of Charges section of the Functionality Overview chapter for more details on time, date and duration interactions.
5. A single Module Code 611 is automatically appended to the AMA record as a result of the *last* Estimate of Charges Request for a call. If multiple estimate requests are made on the same call, the SN must send in a Generate AMA Record Request after each Estimate of Charges Request (except the last one) in order to generate a Module Code 611 for each request.

7.27.1 Estimate of Charges Success Response

Interpretations: The estimate of charges calculation was successful and estimate information is sent to the service node.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Estimate of Charges General Information DB	10 bytes
1 of 2 (M)	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

- Comments:**
1. The Estimate of Charges Duration Information DB will be sent if the optional Estimate of Charges Duration DB was received in the request.
 2. The Estimate of Charges No Duration Information DB will be sent if the optional Estimate of Charges Duration DB was not received in the request.

7.27.2 Estimate of Charges Error Response

Interpretations: The estimate calculation failed.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
060B	SOC option not turned ON
0718	Billing Details Incomplete
0770	Estimate of charges failure
0771	Invalid Estimate Duration for Numbering Plan
0772	Invalid Estimate Date

- Comments:**
1. An ErrorID of #060B can result if either the OSAN0006 SOC or ENSV0018 SOC is not ON.
 2. An ErrorID of #0771 can result if the Estimate Duration exceeds 999 minutes for a call using the North American Numbering Plan.
 3. An ErrorID of #0772 can result if the estimate date is invalid, for example, if an estimate is requested with a specified date of 2/30 (February 30th).

7.28 Generate AMA Record Request

Identifier: #601

Description: A request to generate an AMA Record at the switch.

Interpretations: The switch takes all accumulated billing data collected to this point and formats a Bellcore AMA Record. All formatted billing data collected to this point is included in the AMA Record that is generated.

Preconditions: • None

Post Conditions: • An AMA record is generated.

Side Effects: • The appended billing data buffer (in the switch, used to accumulate modules from the Append AMA Request operations) is cleared allowing further formatted billing data to be sent to the switch by the SN.
• The service node connect time is reset to the current time.

Response(s): • Generate AMA Record Success Response
• Generate AMA Record Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Generate AMA DB	6 bytes
O ^{SV3}	Charge Status DB	6 bytes

Comments: 1. The charge status for the next call segment following a Gen AMA request is automatically marked Billable by the switch.

Duplicate Request: • The Generate AMA DB contains a Generate AMA Sequence Number field. This sequence number is handled as described in 7.1.1 “AMA” on page 181. As noted there, the first time during a session that a service node sends a Generate AMA Record Request operation, the sequence number sent should be set to ‘one’. Each subsequent time, during the same session, that a service node sends a Generate AMA Record Request the sequence number should be incremented.

7.28.1 Generate AMA Record Success Response

Interpretations: The AMA record has been successfully generated.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.28.2 Generate AMA Record Error Response

Interpretations: The AMA record could not be generated.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0702	Invalid generate AMA sequence number
0717	The B party is connected

Comments: 1. If the Generate AMA Sequence Number is the same or lower than that sent with the prior Generate AMA Record Request (during the same session), an Invalid generate AMA sequence number (0702) error is returned to the service node.

7.29 ISUP Calling Party Number Update Request^{SV4}

Identifier: #62E

Description: This operation allows the service node to update the digits sent in the Calling Party Number (CgPN) parameter of the outgoing ISUP Initial Address Message. Normally, the calling digits are sent in the CgPN parameter of the outgoing ISUP IAM.

Preconditions: • None.

Post Conditions: • The digits supplied are stored in the switch.

Side Effects: • If the service node subsequently makes a forward connection and an outgoing ISUP trunk is used, the digits supplied by the service node are sent in the CgPN parameter of the outgoing ISUP IAM.

Response(s): • ISUP Calling Party Number Update Success Response
• ISUP Calling Party Number Update Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	ISUP Calling Party Number DB	16 bytes

- Comments:**
1. An DN digit count of 0 in the ISUP Calling Party Number DB clears any previously entered digits via this operation. The calling digits are sent in the outgoing ISUP IAM.
 2. The number supplied by the service node is only used if the service node subsequently makes a forward connection and an outgoing ISUP trunk is used. If instead of making a forward connection, the service node transfers the call to an operator, TOPS automated system or another service node using OAP 3.0 or lower, the ISUP Calling Party Number is no longer in effect.
 3. The number supplied in this operation is not stored in the AMA record. If the service node wishes to record this number in the AMA record, the OSSAIN custom AMA feature should be used.
 4. In order to update the CgPN parameter, SOC option OSEA0005 must be turned on at the switch.
 5. If the Presentation Indicator of the ISUP Calling Party Number DB is set to Presentation Status Unknown, the Presentation Indicator field in the

outgoing ISUP Calling Number parameter of the IAM will reflect the status of the original calling number.

6. The service node can change the presentation indicator status of the existing ISUP Calling Party Number by including the previously established ISUP CgPN along with the new status in the request.

Duplicate Request: • The digits for the ISUP Calling Party Number is updated again.

7.29.1 ISUP Calling Party Number Update Success Response

Interpretations: The switch has successfully processed the request and the number supplied by the service node is stored at the switch.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.29.2 ISUP Calling Party Number Update Error Response

Interpretations: The operation encountered a problem. The number supplied by the service node is not stored at the switch.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0707	Invalid number format

Comments: 1. Error id #0707 is returned when the number of digits supplied by the service node exceeds the allowed maximum of 18 digits.

7.30 LNP Request^{SV3}

Identifier: #622

Description: This operation returns the portability status of the specified party, and if the specified party's DN is ported, the LRN is also returned.

Preconditions: • The directory number for the specified party must have been stored for the call.

Post Conditions: • An LNP query may be launched if the requested portability information is not already stored in the switch.

Side Effects: • If an LNP query is launched to obtain the LNP information, an AMA module 720 will be appended to the AMA record.

Response(s): • LNP Success Response
• LNP Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Connection ID DB	6 bytes

Comments: 1. If an LNP query is launched, the switch is in a "Restricted Input Mode" while waiting for the query result. The switch will not accept any request until the query is completed.

Duplicate Request: • If an LNP query is in progress, an error will be returned.
• If an LNP query is not in progress and the LNP information is available at the switch, the information will be returned to the service node.

7.30.1 LNP Success Response

Interpretations: The operation was successful. If the specified party's DN is ported, the LRN is returned.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	LNP Information DB	20 bytes

Comments: 1. None.

7.30.2 LNP Error Response

Interpretations: The operation encountered a problem obtaining the LNP information.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0708	No DN digits entered
0749	LNP information not available

Comments: 1. None.

7.31 LRN Update Request^{SV7}

Identifier: #636

Description: A request to set or update the Location Routing Number information associated with the given party.

Preconditions: • The specified party cannot already be connected.

Post Conditions: • The LRN is stored for the given party.

Side Effects: • Data is set that will cause a LNP AMA module to be appended.

Response(s): • LRN Update Success Response
• LRN Update Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	LRN Update DB	8 bytes

Comments: 1. None.

Duplicate Request: • The LRN will take on the value specified in the last successful request received.

7.31.1 LRN Update Success Response

Interpretations: The LRN information was successfully updated for the specified party.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.31.2 LRN Update Error Response

Interpretations: The switch could not update the LRN information for the given party.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
060B	SOC option not turned on
0707	Invalid number format
0709	No calling number
0713	Third number in B party connection
0716	A party has been released
0717	The B party is connected
0733	Party already connected

Comments: 1. None.

7.32 Node Release Inform^{SV3}

Identifier: #61D

Description: Used to inform a service node involved in a simultaneous interaction that it has been released.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments:

1. An active service node would only receive this inform if it was released at the request of a passive operator.
2. A passive service node would receive this inform if it was released as a result of any of the following:
 - an OAP Release Node Request from the active service node requesting that the passive node be released, or
 - a call disposition request (e.g. Transfer to Control List Request) from the active service node, or
 - the service node or session pool associated with the passive session is taken out of service at the switch, or
 - the switch encountered a failure while attempting to send a message to the passive service node.

Lost Message: • If this message is lost, then all requests from the node will be ignored by the switch.

7.33 Notification Timer Inform

Identifier: #62C

Description: Used to inform the service node when the notification timer has expired.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	Call Timing DB	18 bytes

Comments: 1. None.

Lost Message: • If this message is lost, the service node will not know that the notification timer has expired.

7.34 Notification Timer Request

Identifier: #62B

Description: This operation is used to start/stop/reset the persistent-style notification timer.

Preconditions: • If the timer is being stopped, it must be currently running.

Post Conditions: • The notification timer is started/stopped/reset.

Side Effects: • None.

Response(s): • Notification Timer Success Response
• Notification Timer Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Notification Timer DB	10 bytes

- Comments:**
1. Once the persistent-style notification timer is started, it will continue to run until the service node stops it or until it expires. The call can be floated or transferred to other service nodes and the timer will continue to run. Also, the timer will continue to run even if a party goes onhook or is released by the service node.
 2. Like the float-style notification timer, the persistent-style notification timer can run for a relative period of time (Relative Type) or it can be tied to conversation timing (Conversation Type).
 3. If the persistent-style notification timer expires while the call has a session with an service node, the new Notification Timer Inform will be sent.
 4. If the persistent-style notification timer expires while the call is floated, the call will trigger if the trigger profile for the call indicates that a notification timer is a valid trigger. Otherwise, the timer expiration will be ignored and the call will remain floated.
 5. If the SN requests the persistent-style notification timer be started and the timer is already running, it will be reset. In other words, the timer will be stopped and restarted with the new values.

6. Stopping conversation timing will suspend the notification timer if it is running and the notification timer type is conversation. The timer will automatically resume if timing is subsequently started again.

Duplicate Request: • If this request is sent twice, the second request will cause the timer to be reset if the SN requests to start it. If the SN requests to stop the timer, the request will fail because the timer will have been stopped already.

7.34.1 Notification Timer Success Response

Interpretations: The notification timer has been started/stopped/reset, depending on the service node's request.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.34.2 Notification Timer Error Response

Interpretations: The notification timer request was not done. If the timer was not previously running, it still isn't. If the timer was running, it still is.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
075A	Notification timer not running
075F	Conversation timing not started

Comments: 1. None.

7.35 Pass Control Request^{SV3}

Identifier: #61B

Description: Used by an active service node to pass call control capabilities to the passive service node.

Preconditions: • A passive service node must be connected.

Post Conditions: • The active service node becomes the passive service node and the passive service node becomes the active service node.

Side Effects: • An OAP Accept Control Inform has been sent to the passive service node.

Response(s): • Pass Control Success Response
• Pass Control Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None

- Comments:**
1. This operation cannot be used to pass control to a passive operator. The only way to pass control to a passive operator is for the active node to use the Release Node Request to release itself. It should also be noted that passive operators also have the ability to request the release of the active node. If the passive operator used that capability, the active node would receive a Node Release Inform.
 2. Once control is passed, the newly-appointed active service node can decide at some later point to pass control back to the original active service node using this operation. There is no limit on the number of times control can be passed back and forth using this method.

Duplicate Request: • If the first request was received by the switch, then the service node is now the passive service node. An error response is returned with error #0604 (operation not allowed from a passive node).
• If the first request was not received by the switch, then the switch processes the request as normal.

7.35.1 Pass Control Success Response

Interpretations: Call control has been passed to the passive service node. The roles of active and passive have been swapped (i.e. the node receiving this response is now a passive service node).

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.35.2 Pass Control Error Response

Interpretations: Call control was not passed.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0748	A passive SN must be connected

Comments: 1. None.

7.36 Pass Through Inform^{SV3}

Identifier: #61F

Description: Used to send unformatted data to a service node connected in a simultaneous interaction.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Inform

Data Block(s):

M/O	Data Blocks	Byte Length
M	Pass Through DB	106 bytes

- Comments:**
1. Pass through messaging can take place between an active and passive service node or between an active service node and a passive operator.
 2. When the switch receives a Pass Through Request from an active service node, the information is forwarded on to the passive node using the Pass Through Inform.
 3. When the switch receives a pass through message from a passive node or operator, the information is forwarded on to the active service node (using the Pass Through Inform operation).
 4. If an active service node sends the Pass Through Request operation when a passive node is not yet connected (e.g. call is queued for a passive function provider), a return error response is sent to the active service node.
 5. Pass Through messages may be sent at any time when two nodes are connected simultaneously.

Lost Message: • The switch will not take any action if this message is lost. The active and passive nodes may choose to use additional Pass Through messages to provide acknowledgments on the reception of a Pass Through Inform message.

7.37 Pass Through Request^{SV3}

Identifier: #633

Description: This operation is used by the service node to send pass through data to the switch. The pass through data is sent to the function provider connected in a simultaneous connection.

Preconditions: • The service node is connected in a simultaneous connection.

Post Conditions: • The pass through data is sent to the other function provider connected to the call in a simultaneous connection.

Side Effects: • None.

Response(s): • Pass Through Success Response
• Pass Through Error Response

Sender(s): • Active Service Node
• Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Pass Through DB	106 bytes

- Comments:**
1. This is a restricted operation. The service node should wait for a response to this operation before sending another operation to the switch.
 2. Pass through messaging can take place between an active and passive service node or between an active service node and a passive operator.
 3. When the switch receives a Pass Through Request from an active service node, the information is forwarded on to the passive node using the Pass Through Inform.
 4. When the switch receives a pass through message from a passive node or operator, the information is forwarded on to the active service node (using the Pass Through Inform operation).
 5. If an active service node sends the Pass Through Request operation when a passive node is not yet connected (e.g. call is queued for a passive function provider), a return error response is sent to the active service node.

6. Pass Through messages may be sent at any time when two nodes are connected simultaneously.

Duplicate Request: • If this request is lost, the other function provider will not be sent the pass through data.

7.37.1 Pass Through Success Response

Interpretations: The pass through data was successfully sent to the other function provider.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.37.2 Pass Through Error Response

Interpretations: The pass through data was not sent to the other function provider.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available.
0602	Directory number connection in progress
0603	Coin signalling in progress
0606	Passive function provider request in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0741	A passive function provider must be connected.

- Comments:**
1. If error #0601 is returned, the switch could not allocate an extension block to store the pass through data. Switch resources must be increased.
 2. If error #0741 is returned, another function provider (an operator or another service node) must be connected in a simultaneous connection.
 3. The version 3 SOC state must be on for this operation request.

7.38 Passive Function Provider Request^{SV3}

Identifier: #619

Description: Used by an active service node to request that a passive function provider be connected to the call.

Preconditions: • A passive function provider cannot already be connected nor can this call be in queue for a passive function provider.

Post Conditions: • A passive function provider is connected or the call is queued for a passive function provider.

Side Effects: • If the Passive Function Provider Success Response indicates that a passive function provider is connected, the passive function provider will have received an OAP Session Begin (for a passive SN) or an OPP Call Begin (for a passive operator).

Response(s): • Passive Function Provider Success Response
• Passive Function Provider Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes

Comments: 1. This is a restricted operation. The active service node should wait until it receives the response to the Passive Function Provider Request before it sends in additional requests.

2. If the Passive Function Provider Success Response indicates that the call is queued for a passive function provider, a Passive Node Status Inform will be sent to the active service node once the passive function provider is actually connected.

3. If the Passive Function Provider Success Response indicates that the call is queued for a passive function provider, the active service node may cancel the queue request by sending a Release Node Request.

4. If the Passive Function Provider Success Response indicates that the call is queued for a passive function provider and the passive function provider is going to be a TOPS TA operator, the active service node will receive a Passive Node Status Inform when an operator becomes available. This Passive Node Status Inform will indicate that a TOPS operator

is being connected and that no requests should be sent by the active node until the connection is complete. Once the connection is complete, the Passive Node Status Inform described in Comment 2 will be sent to the active node indicating that the operator is fully connected. Once the operator is completely connected, the active node may resume sending requests.

5. An operator cannot be connected in a simultaneous interaction for a call with a session type of Directory Assistance.

7.38.1 Passive Function Provider Success Response

Interpretations: Either a passive function provider is connected or the call is in queue for a passive function provider.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Call Queue Information DB	14 bytes
O 1/2	Node Identification DB	8 bytes
	Operator Identification DB	8 bytes

- Comments:**
1. The Call Queue Information DB will indicate whether the passive function provider is connected or if the call is queued for a passive function provider. Other information about the call queue is also provided in this data block.
 2. If a passive function provider is immediately obtained, then either the Node Identification DB (for passive service node id) or the Operator Identification DB (for operator and terminal id) is sent with the success response.

7.38.2 Passive Function Provider Error Response

Interpretations: A passive function provider was not obtained for this call nor was this call queued for passive function provider.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
060C	Invalid operation for calls involved in a conference
071E	Operation not valid for DA
072E	Voice channel seize failure
0742	Unknown or invalid function
0743	Queue overflow
0744	Queue deflection
0745	Other failure in routing and queuing
0746	Passive function provider already connected or call is already queued for a passive function provider.
074E	Function datafill mismatch
0764	Invalid broadcast configuration

- Comments:**
1. If error ID 0743 (Queue overflow) or 0744 (Queue deflection) is indicated, the Call Queue Information DB is also sent to provide more details on the status of the call queue.
 2. If Error ID 060C is returned, the Conference Release Request must be used to disassociate the call from the conference.
 3. Error ID 0764 can result if the SN has a broadcast voice link and has requested a passive operator.

7.39 Passive Node Status Inform^{SV3}

Identifier: #61C

Description: Used to inform the active service node about changes in the status of the passive function provider during a simultaneous interaction.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Passive Node Status DB	6 bytes
O 1/2	Node Identification DB	8 bytes
	Operator Identification DB	8 bytes

Comments: 1. If the call was queued for a passive function provider and this message is being sent to the active service node to indicate that the passive provider has now be obtained and is connected, then either the Node Identification DB (for passive service node id) or the Operator Identification DB (for operator and terminal id) is sent with this message.

7.40 Release DN Request

Identifier: #60B

Description: Releases the DN from the call and adjusts resources as necessary.

Preconditions: • Specified DN must already be connected to the call.

Post Conditions: • Specified DN is released and resources (e.g. conference ports) have been adjusted.

Side Effects: • 800+ queries aborted.

Response(s): • Release DN Success Response
• Release DN Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Connection ID DB	6 bytes

- Comments:**
1. When the service node requests a release DN, the switch is in a “Restricted Input Mode” while waiting for coin signals to complete (if the call is from a coin phone). This means that the switch will not accept any requests until the release attempt ends.
 2. If the party being released was connected to a conference, that connection will be released. The remaining party connections to the conference are unaffected.
 3. If the B party or Alt DN party is released and the notification timer of type conversation was running, the notification timer will stop.

Duplicate Request: • If the connection is already released, a success response (ReturnResult) will be returned.

7.40.1 Release DN Success Response

Interpretations: The specified DN has been released.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.40.2 Release DN Error Response

Interpretations: The specified party was not connected to the call in the first place.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
073C	Requested party not in forward port

Comments: 1. None.

7.41 Release Node Request^{SV3}

Identifier: #61A

Description: Used to request the release of a particular function provider during a simultaneous interaction.

Preconditions: • A passive function provider must be connected or the call must be in queue for a passive function provider.

Post Conditions: • The node indicated in the Release Node Request is released or the call is taken out of queue if it is queued for a passive function provider and the Release Node Request indicates that the passive node is to be released.

Side Effects: • If the active node requested to release itself, the passive node will have received an Accept Control Inform and it becomes the active node for the call. If the call was in queue for a passive function provider, an OAP Session Begin (for passive SNs) or an OPP Call Begin (for passive operators) will be sent to the newly-appointed active node once one becomes available.

• If the active node requested to release the passive node, then an OAP Node Release Inform (for passive SNs) or an OPP Call End (for passive operators) will have been sent to the passive node.

• If the passive node requested to release itself, an OAP Passive Node Status Inform will have been sent to the active node indicating that the passive node has been released.

Response(s): • Release Node Success Response
• Release Node Error Response

Sender(s): • Active Service Node
• Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Release Node DB	6 bytes

Comments: 1. A passive service node cannot request to have the active node released.

7.41.1 Release Node Success Response

Interpretations: The node indicated in the Release Node Request was released or the call was taken out of queue if it was queued for a passive function provider and the Release Node Request indicated that the passive node was to be released.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.41.2 Release Node Error Response

Interpretations: The switch could not release the function provider requested by the service node.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed from a passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0741	A passive node must be connected or the call must be in queue for a passive node

Comments: 1. Error #0604 is returned to a passive service node if the passive node attempts to release the active node.

7.42 Return Answer Message Request^{SV7}

Identifier: #634

Description: A service node requests that answer be sent to the originator on a call where answer was not previously sent when the voice link connection to the service node was established.

Preconditions: • A party is present.

Post Conditions: • Answer is returned to originator.

Side Effects: • None.

Response(s): • Send Answer Success Response
• Send Answer Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments: 1. None.

Duplicate Request: • A success response will be returned but no processing will be done by the switch.

7.42.1 Return Answer Message Success Response

Interpretations: Answer has been successfully sent to the originator.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None.

Comments: 1. None.

7.42.2 Return Answer Message Error Response

Interpretations: The switch was unable to send answer to the originator.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
060B	SOC option not turned on
0709	No calling number
0711	A subscriber party is not connected
0713	Third number in B party connection

Comments: 1. None.

7.43 Route To Treatment Request

Identifier: #60C

Description: Route the call to the treatment specified in the message. An error has occurred that resulted in the call requiring some form of treatment.

Preconditions: • An A Party must be connected and offhook with no other parties connected.

Post Conditions: • The call is routed to treatment and the service node is released.

Side Effects: • All switch resources associated with the service node are released.
• No triggers are in effect when the call is routed to treatment.

Response(s): • Route To Treatment Success Response
• Route To Treatment Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Treatment DB	6 bytes
O ^{SV3}	Charge Status DB	6 bytes

Comments: 1. This request requires parallel datafill between the service node and the switch for the treatment numbers.
2. The call will not be completed; upon on-hook by the A Party the call is taken down by the switch.

Duplicate Request: • If the first request was successful, the session has ended at the switch. Any messages the service node sends regarding the terminated session will be ignored by the switch.
• The service node may receive a Session Begin Inform message on the same session that was used for the previous call. If this happens, the service node should release the resources for the call and begin the new call. The new call will have a new call identifier.
• If the first request failed, the switch will process the duplicate request as normal.

7.43.1 Route To Treatment Success Response

Interpretations: The switch has routed the call to treatment.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.43.2 Route to Treatment Error Response

Interpretations: The call could not be routed to the treatment specified in the request operation. The active service node is still in control of the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
060C	Invalid operation for calls involved in a conference
070D	The A Party is not connected
0715	Call is set to be transferred
071A	Invalid treatment
071E	Operation not valid for DA
072A	Carrier not served by switch
0731	Party other than A Party is connected
0732	The A Party is on-hook

Comments: 1. If Error ID 060C is returned, the Conference Release Request must be used to disassociate the call from the conference.

7.44 Service Change Request^{SV4}

Identifier: #630

Description: Allows the service node to change the service type of call from Toll and Assistance (TA) to Directory Assistance (DA) or from Directory Assistance to Toll and Assistance.

Preconditions:

- The new (requested) service type must be different from the old (existing) service type.
- There cannot be a forward party connected.

Post Conditions:

- The service type is changed to the new service type.
- Billing information is updated based on the new service type and sent to the service node.
- If the new service type is DA, the service node is updated with the charge status for the call.

Side Effects:

- The forward party directory number is cleared.
- The class charge is reset to unspecified. To satisfy billing for the new service portion of the call, the service node must class charge the call again.
- If the new service type is TA, the billable status is reset to Billable.
- For a DA to TA service change, the Call Origination type is reset to Zero Minus.

Response(s):

- Service Change Success Response
- Service Change Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Requested Service DB	6 bytes

Comments:

1. The Requested Service DB specifies the new base service to be used for the call.
2. There is not an implicit generation of an AMA record on the Service Change Request. If AMA needs to be generated for the old service, the service node must send a Generate AMA Request *prior* to the Service Change Request.

3. Conversation timing is unaffected by any service change. If a Generate AMA Record is not requested prior to the service change, any conversation time existing under the old service will still exist under the new service.

Duplicate Request: • If the first request was successful, the second will result in the switch sending an error response of “New service is the same as the old service” (#0769). If the first request was not successful, the second will be processed as if it was the initial request.

7.44.1 Service Change Success Response

Interpretations: The switch has successfully changed the service type of the call to the service type.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Billing Information DB	10 bytes
O	Charge Status DB	6 bytes

- Comments:**
1. For a TA to DA service change, the Call Origination Type remains what is was prior to the request.
 2. For a DA to TA service change, the Call Origination Type is reset to Zero Minus.
 3. The Billing Information DB is populated with information that is relevant to the new service type.
 4. The Charge Status DB is only sent on a TA to DA service change. The Billable Status field is set based on using the Call Origination Type as part of the index into switch table DABILL.
 5. For a DA to TA service change, the Billable Status of the call is reset to Billable. The Charge Status DB is not sent.

7.44.2 Service Change Error Response

Interpretations: The switch was unable to change the service type of the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0717	The B party is connected
0769	New service is the same as the old service
076B	Operation not allowed for Country Direct call

Comments: 1. None.

7.45 Service Provider ID Assignment Request^{SV3}

Identifier: #623

Description: Assign the SPID to the specified party and type of service provider.

Preconditions: • None.

Post Conditions: • The SPID is associated with the specified party and type of service provider for the duration of the call or until it is overwritten by another request.

Side Effects: • May reset the class charge and/or result in different billing restrictions for the call, reflected in optional data blocks received in the success response.

Response(s): • Service Provider ID Assignment Success Response
• Service Provider ID Assignment Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
1 to 6	SPID DB	12 bytes

Comments: 1. If a SPID is sent for the terminating party or billing party, the SPID is stored at the switch but is not used in any way.

7.45.1 Service Provider ID Assignment Success Response

Interpretations: The SPID for the specified party and type of service provider has been updated for the call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Billing Information DB	10 bytes
O	Class Charge DB	10 bytes

- Comments:**
1. The Billing Information DB is sent if the request results in different billing restrictions for the call. If the DB is not sent, assume the pre-existing billing restrictions still apply.
 2. The Class Charge DB is sent if the new carrier assignment results in a change in the status of the class charge.

7.45.2 Service Provider ID Assignment Error Response

Interpretations: The specified SPID was not assigned to the party and type of service provider specified. Any previous SPID assigned to the party for the type of service provider remains in effect.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress
060B	SOC option not turned on
074F	Invalid service provider id
0761	SPID processing not enabled

Comments: 1. None.

7.46 Session Begin Inform

Identifier: #60D

Description: The switch is sending a call to the service node. If the service node accepts the session begin for the call, the service node is involved in the call.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes
M	Call Information DB	20 bytes
1 of 2 (O)	Node Identification DB	8 bytes
	Operator Identification DB	8 bytes
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
M	Charge Status DB	6 bytes
1 to 2 ^{SV4}	Class Charge DB	10 bytes
1 of 5 (O)	Trigger Information DB	6 bytes
	Trigger Treatment DB	6 bytes
	Trigger Release Cause DB	8 bytes
	Trigger Ringing DB	8 bytes
	Trigger DTMF DB	8 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2	Directory Number DB	34 bytes
0 to 3	Alternate Billing Directory Number DB	18 bytes
O	BCD Card Number DB	24 bytes
O	ASCII Card Number DB	38 bytes
O	Context Block DB	56 bytes
0 to 2	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information DB	6 bytes
O ^{SV3}	Call Timing DB	18 bytes
O to 3 ^{SV3}	LNP Information DB	20 bytes
M	Time DB	10 bytes
0 to 6 ^{SV3}	SPID DB	12 bytes
O ^{SV3}	Preprocessing Information DB	8 bytes
O ^{SV4}	Conference ID DB	8 bytes
0 to 2 ^{SV4}	Conference Party Connection DB	10 bytes
O ^{SV4}	Conference SN Voice Connection DB	6 bytes
O ^{SV4}	ISUP Calling Party Number DB	16 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes
O ^{SV5}	Country Direct Information DB	14 bytes
O ^{SV6}	SN Voice Link DB	12 bytes

M/O	Data Blocks	Byte Length
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes
O ^{SV6}	Estimate of Charges General Information DB	10 bytes
1 or 2 (O) ^{SV6}	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

- Comments:**
1. The service node can deny the session request from the switch. An active service node achieves this by sending a Transfer to Carrier Request, Transfer to Control List Request, Route to Treatment, or an End Call Request. A passive service node achieves this by sending a Release Node Request. A Call Resume Request may be used for a session type of OSSAIN preprocessing.
 2. The Function data block contains the function that should be provided by the service node.
 3. Information on the function provider that was either previously connected or that is currently connected simultaneously to the call is sent to the service node. If the function provider is a service node, then the Node Identification data block is used. If the function provider is an operator, then the Operator Identification data block is sent. If a previous function provider was not connected nor is another function provider currently connected simultaneously to the call, then neither data block is sent.
 4. The Originating Trunk Information data block is optional because there may not be an originating party present.
 5. If a trigger event occurred, one of the five trigger data blocks will be sent. The trigger data block that is sent depends on the trigger event that occurred. The trigger data blocks contain information on the trigger events that caused the service node to be brought into the call. These data blocks are optional. The service node may not have been brought into the call as a result of a trigger (e.g. the call is transferred to the service node from an operator or another service node).
 6. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
 7. The Directory Number data block is optional because there may not be any parties currently present.

- 8.** If context data has been established by any function provider previously involved in the call, then the Context Block data block is sent to the service node.
- 9.** The Alternate Billing Directory Number DB is sent if there is an alternate (or third number) billing number for the call.
- 10.** The BCD Card Number is sent if there is a domestic calling card number for the call.
- 11.** The ASCII Card Number DB is sent if there is a foreign calling card number for the call.
- 12.** A treatment data block is sent if a treatment is applied to either the A party port or B party port.
- 13.** The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.
- 14.** The Call Timing DB is sent if answer for the call has been detected by the switch.
- 15.** If LNP information on the directory number(s) (A party, B party, alternate directory number or line based calling card) is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned. Refer to the Functionality Overview section on LNP for more information.

16. The SPID DB is returned if there is any account owner or billing service provider SPID information for the specified party.
17. The Preprocessing Information DB is only sent on OSSAIN preprocessing sessions.
18. The Conference ID DB is sent if this call is associated with a conference. The Conference Party Connection DB is sent if a subscriber party is connected to the conference. The Conference SN Voice Connection DB is sent if a SN voice link is connected to the conference.
19. Up to two Class Charge data blocks may be sent. One Class Charge DB is sent with current class charge information. The current class charge is always sent. A second Class Charge DB is sent if the Session Origination type is Reorigination (the call was RLT'd and reoriginated back to the switch). The previous class charge is sent if SV4 SOC option is on.
20. The ISUP Calling Party Number DB is sent if the number is available and in effect at the switch.
21. The Terminating Trunk Information is sent when the terminating party is present.
22. The Country Direct Information DB is sent if the call origination is of type country direct.
23. If the switch makes the voice link selection for the session, the SN Voice Link DB is returned to indicate the voice link it selected.
24. The Estimate of Charges General Information, Duration and No Duration DBs are only sent if a successful estimate of charges has been previously performed on the call.

Lost Message: • If this message is lost or the service node does not respond, switch timers will eventually expire. See Chapter 3 for more details on lost message handling by the switch.

7.47 Session Initiation Request

Identifier: #60F

Description: The service node is requesting the switch to begin a session.

- Preconditions:**
- The function id in the class header must be datafilled in switch Table OAF-UNDEF.
 - If a Routing Index or Routing Number DB is included with this operation, the switch must be using North American translations.

Post Conditions:

- If there are no errors with the request, a session is established between the switch and service node. The switch will allocate a Call ID for the call.

Side Effects:

- None.

Response(s):

- Session Initiation Success Response
- Session Initiation Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Network Service DB	6 bytes
M	Charge Status DB	6 bytes
O	Carrier Number DB	8 bytes
0 to 2	Directory Number Update DB	30 bytes
1 of 3 (O)	Alternate Billing Directory Number DB	18 bytes
	BCD Card Number DB	24 bytes
	ASCII Card Number DB	38 bytes
0 to 6 ^{SV3}	SPID DB	12 bytes
O	Originating Trunk Information DB	8 bytes
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes
O ^{SV7}	LNP Enhancements DB	6 bytes

- Comments:**
1. The service node will select the Session ID to be used for the current session.
 2. The service node cannot send any requests to the switch until it receives a response for this operation.
 3. Most data blocks are optional. The service node can send the information with this request or can send it under the specific request for the optional

data blocks (e.g. Directory Number Update DB can be sent with the Directory Number Request operation).

4. An LNP query may be launched for the A party and/or the B party if the number is portable. Refer to the Functionality Overview section on LNP for more information.
5. If any errors are found in any data blocks in the Session Initiation Request, then the session request is denied and a call ID is not allocated.
6. The sequence number of the Session Initiation Request message must be a 1. If it is not, the request is denied and a call ID is not allocated.
7. When a centralized service node initiates a call to an OSAC Remote, the state of the service node and session pool is verified with the OSAC Host. If the service node or session pool is not in service at the Host, then an response message is returned to the service node. Refer to the Functionality Overview section on OSAC for more information.
8. The Originating Trunk Information DB should be sent if the A party will not be connected in this session and is already connected in another session (ex: the A party is connected in the first leg of a 3-way call scenario and this session is being initiated for the C party). This information is needed to do equal access translations on the B party.
9. Directory Numbers (if any Directory Number Update DBs are sent) for domestic calls under the North American numbering plan must be no greater than 10 digits (i.e. for Directory Number Type of Domestic, and switch data-filled as North American Dialing Plan, the Directory Number Digit Count, for A or B parties, should be between 0 and 10). Directory numbers for foreign calls under the North American numbering plan (B party only) have a maximum of 15 digits. Directory numbers for all calls and parties under the Open numbering plan have a maximum of 18 digits.

Duplicate Request: • If a second Session Initiation request is sent on a session which the switch considers already in use by another call, the switch will take the first call down and begin a new call with the data in the second session initiation request. A new callid will be allocated for the new call.

- An AMA record is not generated for the first call.

7.47.1 Session Initiation Success Response

Interpretations: The switch was able to initiate a session to the service node. It has successfully allocated a Call ID for the call. The Call ID is returned in the Call Processing Class header.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Billing Information DB	10 bytes
O	BCD Card Information DB	6 bytes
O	Treatment DB	6 bytes
O	Originating Station Information DB	22 bytes
O	LATA Information	6 bytes
O to 2 ^{SV3}	LNP Information DB	20 bytes
M	Time DB	10 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2 ^{SV3}	SPID DB	12 bytes
O ^{SV6}	SN Voice Link DB	12 bytes
O ^{SV6}	Directory Number Information DB	8 bytes

- Comments:**
1. The Billing Information data block is returned if the Session Initiation Request had a Directory Number Update data block.
 2. The BCD Card Information DB is returned if the billing number is a BCD Card Number and the BCD Card Number Type field is set to Unknown.
 3. If a directory number translates to a treatment, the treatment value is returned in the Treatment DB. The treatment is not applied to the call. The service node may connect to the treatment via the Connect DN request or the Route to Treatment Request.
 4. The Originating Station Information DB will be returned for an A party DN update, providing associated origination information about that directory number.
 5. The LATA Information data block is returned if the Session Initiation Request had a Directory Number Update data block.
 6. If LNP information on the directory number(s) (for A party and/or B party) is available or as a result of an LNP query, the LNP Information DB is returned.
 7. The Carrier data block is sent for carrier calls. The NBEC DB is sent for calls completed by the operating company.

8. The SPID DB is returned if there is account owner or billing service provider SPID information for the A party.
9. If the switch makes the voice link selection for the session, the SN Voice Link DB is returned to indicate the voice link it selected.
10. The Directory Number Information DB is returned if the Session Initiation Request had a Directory Number Update data block for the B party.

7.47.2 Session Initiation Error Response

Interpretations: The switch was unable to initiate a session to the service node.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
060B	SOC option not turned on
0700	Cannot allocate call id
0703	Session not available
0705	Invalid language
0706	Invalid number type
0707	Invalid number format
0708	No DN digits entered
0709	No calling number
070B	No billing number
070C	Billing number is hotlisted
0720	Invalid carrier
072A	Carrier not served by switch
0734	Translation failed
074B	No response from OSAC Host
074C	No host-remote session available
074D	Session pool datafill mismatch
074E	Function datafill mismatch
074F	Invalid service provider id
0761	SPID processing not enabled
076A	Invalid trunk group and member
076C	Invalid Routing Index
076D	North American-only functionality

- Comments:**
1. The first error found is returned in the error response. However, it may not be the only error in the message.
 2. If Error ID #0709 is returned, the originating trunk information cannot be updated if there is no A party DN.
 3. Error ID #076D is sent if the Session Initiation Request included a Routing Index DB or Routing Number DB and the switch does not use North American translations.

7.48 Session Recall Request^{SV4}

Identifier: #62D

Description: This operation is used to initiate a session with a floated call.

Preconditions:

- The SN originated session used to initiate this request must be idle.
- The target call must be floated.

Post Conditions:

- The target call is connected to the service node originated session and is no longer floated.

Side Effects:

- None.

Response(s):

- Session Recall Success Response
- Session Recall Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Callid DB	8 bytes

Comments:

1. This request must be issued on an idle SN originated session. If the request succeeds, complete call details will be returned in the success response. If the target call already has an service node session associated with it at the time of this request, an error response is returned with the session pool id and session id that the call is connected to.
2. If this request is received on an idle subscriber originated session, the request will be ignored.

Duplicate Request:

- If this request is sent twice on the same session, the second request will fail because the SN originated session will no longer be idle.

7.48.1 Session Recall Success Response

Interpretations: The target call is now connected to this session.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes
M	Call Information DB	20 bytes
1 of 2 (O)	Node Identification DB	6 bytes
	Operator Identification DB	8 bytes
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
M	Charge Status DB	6 bytes
M	Class Charge DB	10 bytes
1 of 5 (O)	Trigger Information DB	6 bytes
	Trigger Treatment DB	6 bytes
	Trigger Release Cause DB	8 bytes
	Trigger Ringing DB	8 bytes
	Trigger DTMF DB	8 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2	Directory Number DB	34 bytes
0 to 3	Alternate Billing Directory Number DB	18 bytes
O	BCD Card Number DB	24 bytes
O	ASCII Card Number DB	38 bytes
O	Context Block DB	56 bytes
0 to 2	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information DB	6 bytes
O	Call Timing DB	18 bytes
O to 3	LNP Information DB	20 bytes
M	Time DB	10 bytes
0 to 6	SPID DB	12 bytes
O	Conference ID DB	8 bytes
0 to 2	Conference Party Connection DB	10 bytes
O	ISUP Calling Party Number DB	16 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes
O ^{SV5}	Country Direct Information DB	14 bytes
O ^{SV6}	SN Voice Link DB	12 bytes
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes
O ^{SV6}	Estimate of Charges General Information DB	10 bytes
1 or 2 (O) ^{SV6}	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

- Comments:**
1. The Conference ID DB is sent if this call is associated with a conference. The Conference Party Connection DB is sent if a subscriber party is connected to the conference. The Conference SN Voice Connection DB is sent if a SN voice link is connected to the conference. The service node can deny the session request from the switch. An active service node achieves this by sending a Transfer to Carrier Request, Transfer to Control List Request, Route to Treatment, or an End Call Request. A passive service node achieves this by sending a Release Node Request. A Call Resume Request may be used for a session type of OSSAIN preprocessing.
 2. The Function data block contains the function that should be provided by the service node.
 3. Information on the function provider that was either previously connected or that is currently connected simultaneously to the call is sent to the service node. If the function provider is a service node, then the Node Identification data block is used. If the function provider is an operator, then the Operator Identification data block is sent. If a previous function provider was not connected nor is another function provider currently connected simultaneously to the call, then neither data block is sent.
 4. The Originating Trunk Information data block is optional because there may not be an originating party present.
 5. If a trigger event occurred simultaneously with the Session Recall Request, one of the five trigger data blocks will be sent. The trigger data block that is sent depends on the trigger event that occurred.
 6. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
 7. The Directory Number data block is optional because there may not be any parties currently present.
 8. If context data has been established by any function provider previously involved in the call, then the Context Block data block is sent to the service node.
 9. The Alternate Billing Directory Number DB is sent if there is an alternate (or third number) billing number for the call.
 10. The BCD Card Number is sent if there is a domestic calling card number for the call.
 11. The ASCII Card Number DB is sent if there is a foreign calling card number for the call.
 12. A treatment data block is sent if a treatment is applied to either the A party port or B party port.
 13. The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.

14. The Call Timing DB is sent if answer for the call has been detected by the switch.
15. If LNP information on the directory number(s) (A party, B party, alternate directory number or line based calling card) is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned. Refer to the Functionality Overview section on LNP for more information.
16. The SPID DB is returned if there is any account owner or billing service provider SPID information for the specified party.
17. The Conference ID DB is sent if this call is associated with a conference. The Conference Party Connection DB is sent if a subscriber party is connected to the conference.
18. The ISUP Calling Party Number DB is sent if the number is available and in effect at the switch.
19. The Terminating Trunk Information is sent when the terminating party is present.
20. The Country Direct Information DB is sent if the call origination is of type country direct.
21. If the switch makes the voice link selection for the session, the SN Voice Link DB is returned to indicate the voice link it selected.
22. The Estimate of Charges General Information, Duration and No Duration DBs are only sent if a successful estimate of charges has been previously performed on the call.

7.48.2 Session Recall Error Response

Interpretations: The target call could not be connected to this service node session.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0601	No switch resources available
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
0703	Session not available
0759	Unknown or invalid callid
075B	Session is not idle
075C	Session is not used for SN originations
075D	Target call already has a session with an SN

Comments: 1. If Error ID 075D is returned, the Session Information DB is sent to the service node which indicates which session the target call is connected to.

7.49 Speech Path Request

Identifier: #610

Description: Service nodes send this operation when they need to alter a party's speech status. This operation may be requested only for an existing, connected party (the change will occur instantly).

Preconditions:

- The directory number for the specified connection (through prior Billing or Directory Number Requests) is required.
- The party must be connected.

Post Conditions:

- The speech path is altered.

Side Effects:

- None.

Response(s):

- Speech Path Success Response
- Speech Path Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Speech Path DB	6 bytes

Comments:

1. The service node is not inhibited in any way from altering the speech path status of any party involved in a call that is using a broadcast voice link; however, the switch network connection may limit the effectiveness of some requests. In particular, the one-way network connection will obviously impact the ability of a party to speak or transmit DTMF tones *to the voice link*. For example, on a call with just a calling party and a broadcasting voice link, the speech path status for the calling party might be two-way, but it is limited to one-way, listen-only, by the hardware connection in the network. The subscriber will not be *heard* by the voice link.

Duplicate Request:

- The speech path change will be verified to already be in force. A success response will be returned.

7.49.1 Speech Path Success Response

Interpretations: The specified party's speech path was altered according to the request.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.49.2 Speech Path Error Response

Interpretations: The specified party's speech path was not altered according to the request.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
0711	A subscriber party is not connected
073A	No DN associated with connection

Comments: 1. None.

7.50 Text To Operator Request

Identifier: #611

Description: This operation allows a small block of text to be presented to a compatible operator position once the operator becomes involved in the call.

Preconditions: • None.

Post Conditions: • The text buffer is overwritten with the new text data.

Side Effects: • None.

Response(s): • Text To Operator Success Response
• Text To Operator Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Text DB	26 bytes

Comments: 1. The text buffer is sent only once to an operator.
2. Once the text buffer is sent to the operator, the buffer is cleared.

Duplicate Request: • If the switch can perform the operation (i.e. none of the error reasons apply), the switch will overwrite the text buffer again.

7.50.1 Text To Operator Success Response

Interpretations: The text will be sent to the operator position if the call is transferred to an operator.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.50.2 Text To Operator Error Response

Interpretations: The text buffer has not been changed. If there was text in the buffer before, it has not changed. If the call is transferred to an operator position, it will receive the text currently in the buffer.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress

Comments: 1. None.

7.51 Transfer to Carrier Request

Identifier: #618

Description: Requests that the switch release the service node from the call and transfer the call to the carrier.

Preconditions:

- The switch does not provide operator services for the carrier.
- The forward party or alternate billed party is not connected.

Post Conditions:

- Any switch resources associated with the service node are freed (i.e. session, voice link, dtmf receiver).
- All function providers are released from the call.
- Speech paths are automatically set to two-way.

Side Effects:

- Switch based triggering is not in effect.

Response(s):

- Transfer to Carrier Success Response
- Transfer to Carrier Error Response

Sender(s):

- Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
O	Carrier Number DB	8 bytes
OSV3	Charge Status DB	6 bytes
OSV3	RLT Request DB	6 bytes
OSV6	Routing Index DB	8 bytes

Comments:

1. The carrier number may have been previously specified via the Carrier Request operation or Session Initiation Request.
2. The service node must explicitly request RLT for the call. If the call is RLT eligible, but the service node does not request RLT, then RLT is not attempted.
3. When a Routing Index DB is included with this operation, the Connection ID field must be set to B Party Connection.
4. If a Routing Index DB is sent with a Routing Index of 2046 (the NIL value), the alternate routing index and alternate routing number associated with the B Party connection will be cleared. The call will route to the carrier using normal translations.

- Duplicate Request:** • If the first request was successful, the session with the service node has ended at the switch. Any messages the service node sends regarding the terminated call will be discarded by the switch. From the switch's perspective, the session is now available to present another call to the service node. The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.
- If the first request failed, the switch will process the second request as normal and return the appropriate response.

7.51.1 Transfer to Carrier Success Response

Interpretations: The switch has successfully transferred the call to the carrier.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. Triggering is not in effect.

7.51.2 Transfer to Carrier Error Response

Interpretations: The call could not be transferred to the carrier. The call is still at the service node and the service node is in control of the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
060C	Invalid operation for calls involved in a conference
0713	Third number in B party connection
0717	The B party is connected
071E	Operation not valid for DA
071F	Routing to carrier failed
0720	Invalid carrier
0722	Carrier served by switch
072B	Carrier not allowed for call
0735	Translations failed, treatment applied
073D	Carrier id is restricted
073E	Carrier not specified for call
073F	Timing has started
0740	Incoming trunk group not supported for transfer
076C	Invalid Routing Index
076E	Connection ID - Party mismatch

- Comments:**
1. A ReturnError of 'Translations failed, Treatment applied' (0735) will return a Treatment DB indicating the specific treatment received.
 2. Any previous carrier information stored at the switch is set to nil if a 0720, 72B, 735 error is returned; the service type is not DA; and interlata is not restricted to the original carrier.

3. If Error ID 060C is returned, the Conference Release Request must be used to disassociate the call from the conference.

7.52 Transfer to Control List Request

Identifier: #613

Description: Sent when the service node wishes to direct the call to a specific, new control list.

Preconditions: • A subscriber party must be connected and off hook.

Post Conditions: • All function providers previously involved in the call are released from the call.
• Switch processing directs the call to the first function in the specified control list.

Side Effects: • If the node was providing a function in a specific control list, that control list would be abandoned.

Response(s): • Transfer to Control List Success Response
• Transfer to Control List Error Response

Sender(s): • Active Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Transfer Control List DB	8 bytes
O	Context Block DB	56 bytes
O	Trigger Profile Index DB	14 bytes
0 to 3	Language DB	6 bytes
O	Billing Acceptance DB	6 bytes
O	Operator Services System Action DB	6 bytes
O	Text DB	26 bytes
O ^{SV3}	Charge Status DB	6 bytes
O	Branding Indicator DB	6 bytes
O	Call Update DB	6 bytes

Comments: 1. A language can be specified for each connection in the Language data block. However, the switch can store either the B party language or the Alt DN language, but not both. The last language DB processed between the B party and the Alt DN will be saved in the switch.

2. If the call is transferred to a DA operator, the service node should release the B party DN and generate an AMA record. If the B party is still connected at the transfer, then the switch will release the B party and generate an AMA record before transferring the call to a DA operator.

3. If the call is transferred to a DA operator and there is no A party connected, the switch will generate a log and take the call down.
4. If the call is transferred to an operator and the B party has been connected but has not answered, the service node should release the B party and generate an AMA record. If the B party is still connected, the switch will release the B party and generate an AMA record.
5. The Billing Acceptance DB and the Operator Services System Action DB contain billing information required for calls billed to a calling card number, a third number, or billed collect. Specifically, the Operator Services System Action DB is used to determine whether billing has been satisfied on credit card, third number and collect calls. This information is used on transfers when determining a call's eligibility for automated systems such as ADAS (Automated Directory Assistance Service) as well as affecting billing-related operator screen displays. If this DB is not sent on these calls, the switch assumes that billing was not satisfied.

Duplicate Request: • If the first request was successful, the session between the switch and service node has ended at the switch. Any messages the service node sends on the terminated session regarding the previous call will be ignored by the switch. From the switch's perspective, the session is now available to present another call to the service node (even though the service node may not have received the response). The switch may eventually send a Session Begin Inform message to the service node on the session. If this happens, the service node should take the previous call down and begin the new call for the Session Begin. The new call will have a new call identifier.

- If the first request failed, the switch will process the duplicate request as normal.

7.52.1 Transfer to Control List Success Response

Interpretations: The switch has successfully transferred the call.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.52.2 Transfer to Control List Error Response

Interpretations: The transfer is not allowed and the current node is still in control of the call.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0604	Operation not allowed for passive node
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
0608	Operation is restricted for preprocessing session
060A	OSAC voice connection in progress
060B	SOC option not turned on
060C	Invalid operation for calls involved in a conference
0705	Invalid Language
0711	A subscriber party is not connected
0713	Third number in B party connection
0717	The B party is connected
0719	Invalid trigger profile
071B	Invalid control list
071C	No subscriber parties off hook
072A	Carrier not served by switch
0747	Conversation time exceeds notification time
0750	Conversation timing cannot be performed for 'No AMA' calls

- Comments:**
1. If the language specified in the Language DB is invalid, error 0705 is returned, The language for the connection is set to Nil in the switch.
 2. The language for the B party cannot be updated if a third party is connected. Error ID 0713 is returned if an attempt is made to change the B party language while the third party is connected.
 3. The language for the third party cannot be updated if the B party is connected. Error ID 0717 is returned if an attempt is made to change the third party language while the B party is connected.
 4. If Error ID 060C is returned, the Conference Release Request must be used to disassociate the call from the conference.

7.53 Trigger Event Inform

Identifier: #614

Description: The switch is sending the message to the service node without giving call control to the service node. The call is sent due to a trigger event other than a DTMF digit or a release cause event.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Function DB	6 bytes
M	Call Information DB	20 bytes
1 of 2 (O)	Node Identification DB	6 bytes
	Operator Identification DB	8 bytes
O	Originating Trunk Information DB	8 bytes
M	Billing Information DB	10 bytes
M	Originating Station Information DB	22 bytes
M	Charge Status DB	6 bytes
1 to 2 ^{SV4}	Class Charge DB	10 bytes
1 of 3 (M)	Trigger Information DB	6 bytes
	Trigger Treatment DB	6 bytes
	Trigger Ringing DB	8 bytes
1 of 2 (O)	Carrier DB	20 bytes
	NBEC DB	8 bytes
0 to 2	Directory Number DB	34 bytes
0 to 3	Alternate Billing Directory Number DB	18 bytes
O	BCD Card Number DB	24 bytes
O	ASCII Card Number DB	38 bytes
O	Context Block DB	56 bytes
0 to 2	Treatment DB	6 bytes
O	OLNS Information DB	50 bytes
M	LATA Information DB	6 bytes
O ^{SV3}	Call Timing DB	18 bytes
O to 3 ^{SV3}	LNP Information DB	20 bytes
M	Time DB	10 bytes
0 to 6 ^{SV3}	SPID DB	12 bytes
O ^{SV4}	Conference ID DB	8 bytes
0 to 2 ^{SV4}	Conference Party Connection DB	10 bytes
O ^{SV4}	ISUP Calling Party Number DB	16 bytes
O ^{SV4}	Terminating Trunk Information DB	8 bytes
O ^{SV5}	Country Direct Information DB	14 bytes
0 to 3 ^{SV6}	Routing Index DB	8 bytes
0 to 3 ^{SV6}	Routing Number DB	16 bytes
O ^{SV6}	Estimate of Charges General Information DB	10 bytes
1 or 2 (O) ^{SV6}	Estimate of Charges Duration Information DB	8 bytes
	Estimate of Charges No Duration Information DB	16 bytes

Comments: 1. The service node should not respond to this inform message. Any responses will be ignored by the switch.

2. The Function data block contains the function that the trigger event mapped to in the trigger datafill.
3. Information on the previous function provider (i.e. the last function provider before this service node) involved in the call is sent to the service node. If the previous function provider was a service node, then the Node Identification data block is used. If the previous function provider was an operator, then the Operator Identification data block is sent. If there was not a previous function provider for the call (i.e. the service node is the first function provider), then neither data block is sent.
4. The Originating Trunk Information data block is optional because there may not be an originating party present.
5. The trigger data block that is sent depends on the trigger event that occurred. The trigger data block sent contains information on the trigger event that caused the service node to be brought into the call.
6. The Carrier data block contains information on the carrier associated with the call (for carrier calls or calls arriving on a carrier trunk). The NBEC DB is sent for calls completed by the operating company.
7. The Directory Number data block is optional because there may not be any parties currently present.
8. If context data has been established by any function provider previously involved in the call, then the Context Block data block is sent to the service node.
9. The Alternate Billing Directory Number DB is sent if there is an alternate (or third number) billing number for the call.
10. The BCD Card Number is sent if there is a domestic calling card number for the call.
11. The ASCII Card Number DB is sent if there is a foreign calling card number for the call.
12. A treatment data block is sent if a treatment is applied to either the A party port or B party port.
13. The OLNS Information DB is sent if the DMS query was made to an external OLNS LIDB.
14. The Call Timing DB is sent if answer for the call has been detected by the switch.
15. If LNP information on the directory number(s) (A party, B party, alternate directory number or line based calling card) is available through switch datafill or as a result of an LNP query, the LNP Information DB is returned.
16. The SPID DB is returned if there is any account owner or billing service provider SPID information for the specified party.

17. The Conference ID DB is sent if this call is associated with a conference. The Conference Party Connection DB is sent if a subscriber party is connected to the conference.
18. The ISUP Calling Party Number DB is sent if the number is available and in effect at the switch.
19. The Terminating Trunk Information DB is sent when the terminating party is present.
20. Up to two Class Charge data blocks may be sent. One Class Charge DB is sent with current class charge information. The current class charge is always sent. A second Class Charge DB is sent if the Session Origination type is Reorigination (the call was RLT'd and reoriginated back to the switch). The previous class charge is sent if SV4 SOC option is on.
21. The Country Direct Information DB is sent if the call origination is of type country direct.
22. The Estimate of Charges General Information, Duration and No Duration DBs are only sent if a successful estimate of charges has been previously performed on the call.

Lost Message: • There is no indication at the switch if this message is lost. The service node will not have received the message and will not be able to process the trigger event.

7.54 Voice Connect Request

Identifier: #615

Description: Used by the service node to establish a voice connection for the call and indicate which voice channel between the switch and the service node should be used. This operation is sent as soon as a data-only connection between the switch and the service node needs to be supplemented by a voice connection. If SOC OSAN0006 is ON the SN can have the switch select the voice link instead of specifying a voice channel in the request.

Preconditions:

- No voice connection exists for this call (if SOC OSAN0006 is OFF).
- If the optional Voice Channel DB is sent requesting a non-broadcast voice channel, the voice channel must be available.
- For broadcast voice links, the making of this connection must not exceed the maximum allowable for the voice link (defined in switch table OAVLMAP, field MAXCONNS).
- When requesting a broadcast voice connection during a simultaneous interaction, there cannot be another existing voice link connection of any type (standard or broadcast or passive operator) associated with the other function provider.
- When requesting a standard voice connection during a simultaneous interaction, there cannot be an existing broadcast connection associated with the other function provider.

Post Conditions:

- If switch table OAVLMAP indicates the voice channel is not used for broadcasting (field BCST = N) and the optional Broadcast DB is not received, a two-way connection between the switch and the service node is completed over the specified voice link.
- If switch table OAVLMAP indicates the voice channel is used for broadcasting (field BCST = Y) and the optional Broadcast DB is received, a one-way connection between the switch and the service node is completed over the specified voice link.
- If SOC OSAN0006 is ON and a previous voice link connection existed for the session at the time of the request, that connection is released and replaced by the new connection. If the switch is unable to make the new connection the first is still released. In both cases the SN is informed of the release of the initial voice link in the response.

Side Effects: • None.

Response(s):

- Voice Connect Success Response
- Voice Connect Error Response

Sender(s):

- Active Service Node
- Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M ^{SV2-5} O ^{SV6}	Voice Channel DB	6 bytes
O ^{SV4}	Conference ID DB	8 bytes
O ^{SV4}	Broadcast DB	8 bytes

- Comments:**
1. Parallel datafill between the switch and service node is necessary to guarantee that the specified voice channel maps to the same connection for both agents.
 2. Refer to Section 2.4.3 “Voice Connections” on page 74 and Section 13.13 “Voice Connections” on page 990 for more information on voice connections.
 3. If the switch detects that the service node has gone off-hook towards the switch on a non-broadcast voice connection, then the switch will release the voice link between the switch and service node. The service node should take corrective actions (e.g. send the call to an operator or another service node). The voice link is put into an RMB state in the switch and cannot be used again until an onhook towards the switch is detected.
 4. Offhooks and onhooks on a broadcast voice connection for voice links with switch datafill of IMMEDIATE in field CUTTHRU of table OAVL-MAP are ignored. The voice link connection is unaffected.
 5. Offhooks and onhooks on a broadcast voice connection for voice links with switch datafill of HKCHG in field CUTTHRU of table OAVL-MAP are used to indicate the beginning of a recording cycle. The voice link connection is unaffected unless the switch is counting cycles and the last cycle has been played. In such a case, the voice link connection is released.
 6. For broadcast voice links using the HKCHNG cut thru method, there must be at least one party connected on the call or the detection of the hook state change(s) will not take place.
 7. This is a restricted operation. If voice connections are being made for a call involving a centralized service node and an OSAC Remote, messaging will occur between the OSAC Remote and OSAC Host. The service node should not send any requests until the voice connections are complete.
 8. If the Conference ID DB is sent, the SN voice link will be connected to the specified conference. Otherwise, the SN voice link is connected the parties in this call.

9. If the request is for a broadcast voice link owned by another node, the owning node is not informed of this usage by the switch. It is up to the service nodes to coordinate the sharing of voice link resources.
10. An initial period of silence is needed before playing any prompt to ensure that speech has been established throughout the external network. Typically, 550 milliseconds is a sufficient amount of time to ensure speech has been established.
11. The Voice Channel DB is optional if SOC OSAN0006 is turned on; otherwise it is mandatory.

Duplicate Request:

- If the first request was successful the switch will return an error indicating the voice link is already connected.
- If the first request failed, the switch will perform the second voice connect request as normal.

7.54.1 Voice Connect Success Response

Interpretations: The voice connection between the switch and the service node has been established. The speech path is two-way for non-broadcast connections. For broadcast voice links, the switch network connection is one-way.

Message Class: Call Processing

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O ^{SV4}	Broadcast Count DB	6 bytes
0 to 2 ^{SV6}	SN Voice Link DB	12 bytes

- Comments:**
1. If the connection is to a broadcast voice link, the optional Broadcast Count DB is appended, This data block gives the number of current connections on the broadcast voice link *at the time the success response message was composed*.
 2. If the switch makes the voice link selection for the session, the SN Voice Link DB is returned to indicate the voice link it selected.
 3. If a previous voice link connection was released while processing the Voice Connect Request, the SN Voice Link DB is returned to indicate which voice link was released.

7.54.2 Voice Connect Error Response

Interpretations: The voice connection between the switch and the service node has not been established.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress
060B	SOC option not turned on
072C	Invalid voice channel id
072D	Voice channel already in use
072E	Voice channel seize failure
0730	There is already a voice connection for this session
074A	Host-Remote voice channel not available
0752	Unknown or invalid conference id
0756	Too many voice entities in conference
0762	Attempt to exceed max connections for broadcast voice link
0763	Voice link cutthru method invalid for cycle counting
0764	Invalid broadcast configuration
0766	Not a broadcast voice link
0767	OSAC Host SOC not on
0768	Voice link for broadcast only
076F	SN Voice Link datafill missing
0773	OSAC release too low

- Comments:**
1. If the error ID is #0730, then the switch will return either the Voice Channel DB (if SN selected) or the SN Voice Link DB (if switch selected) with the voice channel ID currently being used for this session.
 2. If the error ID is #0762, then the switch will return the Broadcast Count DB. This data block specifies the number of current connections that exist on the broadcast voice link *at the time the error message was composed*.
 3. If the switch had to release an existing voice link connection as a result of a Voice Connect Request but was unable to make a connection to the new

voice link, the SN Voice Link DB identifying the released voice link will be returned in this error response, along with the appropriate error ID.

4. Error ID #0773 is returned when the SN wants the switch to select a voice link and the voice links are centralized but the version of OSAC protocol between the host and remote does not support switch voice link selection.

7.55 Voice Release Inform

Identifier: #62F

Description: Informs the service node that the switch has removed a voice link from a call. There is no longer a voice connection between the service node and the switch.

Sender(s): • Call Processing Switch

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None

Comments: 1. This message can be received anytime a voice link (non-broadcast or broadcast) is released by the switch as a result of something other than a Voice Release Request from the service node.

Lost Message: • If the message is lost and the service node tries to explicitly release the voice link (Voice Release Request), the switch will respond with error id #072F (There is no voice connection for this session).

7.56 Voice Release Request

Identifier: #616

Description: Used by the service node to indicate that the voice connection between the switch and the service node should be freed. When issued, the switch should perform whatever actions necessary to remove the specified voice link from the call topology.

Preconditions: • There should be a voice connection for the current call.

Post Conditions: • The speech path between the switch and the service node is removed.

Side Effects: • None.

Response(s): • Voice Release Success Response
• Voice Release Error Response

Sender(s): • Active Service Node
• Passive Service Node

Message Class: Call Processing

Message Type: Invoke

Data Block(s): None.

Comments: 1. If the SN voice link was connected to a conference, that connection will be released. The remaining parties in the conference are unaffected.

Duplicate Request: • If the first request was successful, the switch will respond with an error message indicating that there is no voice connection for this session.
• If the first message failed, the switch will perform the operation as normal.

7.56.1 Voice Release Success Response

Interpretations: The voice connection between the switch and the service node has been removed.

Message Class: Call Processing

Message Type: Return Result

Data Block(s): None

Comments: 1. None.

7.56.2 Voice Release Error Response

Interpretations: The voice connection between the switch and the service node could not be removed.

Message Class: Call Processing

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
0600	Database query in progress
0602	Directory number connection in progress
0603	Coin signalling in progress
0605	Voice link request in progress for passive node
0606	Passive function provider request in progress
0607	Passive operator connection in progress
060A	OSAC voice connection in progress
072F	There is no voice connection for this session

Comments: 1. None.

Chapter 8: Non-Call Processing Operations

8.0 Operation Grouping Name (Template)

This section has the templates that are used for defining OAP operations and responses.

The operations themselves (along with responses) are placed in alphabetic order using 7.x as a heading.

8.0.1 Request Operation

Identifier: *This template is used for request operations.*

Description: *What this operation is used for.*

Preconditions: • *Characteristics or information required to successfully perform this operation.*

Post Conditions: • *Changes made as a direct result of successfully performing this operation.*

Side Effects: • *Anything that occurs as a result of successfully performing this request.*

Response(s): • *List of responses if any*

- X Success Response
- X Error Response

Sender(s): • *List of potential senders who can send this message*

- Maintenance Switch
- Service Node
- OSAC Remote Switch

Message Class: Node, Session Pool

Message Type: Invoke

Data Block(s): *“None”, if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this request message.*

Duplicate Request: • *Specifies what will happen if this request is received twice by the switch.*

8.0.1.1 Success Response

Interpretations: *Describes what it means to get this response.*

Message Class: Node, Session Pool

Message Type: Return Result

Data Block(s): *"None", if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this response message.*

8.0.1.2 Error Response

Interpretations: *Describe what it means to get this response.*

Message Class: Node, Session Pool

Message Type: Return Error

.

ErrorID (#)	Value

Comments: 1. *Additional information that relates to this response message.*

8.0.2 Inform Operation

Identifier: *This template is used for operations that do not receive a response.*

Description: *What this operation is used for.*

Sender(s): • *List of potential senders who can send this message*

- Maintenance Switch
- Active Service Node

Message Class: Node, Session Pool

Message Type: Invoke

Data Block(s): *“None”, if no data blocks are needed*

M/O	Data Blocks	Byte Length
M or O		
N to M		
N of X (M/O)		

M = Mandatory Data Block

O = Optional Data Block

N to M = N to M of this data block can be sent with the operation. N is the minimum number of this data block that can be sent and M is the maximum number of this data block that can be sent. If N = 0, the data block is optional (0, 1, 2, or more may be sent). If N = 1 or more, the data block is mandatory.

N of X (M/O) = X represents the number of possible choice data blocks, of which only N of the data blocks may (depending on optionality) be sent. Out of a set of X data blocks, send N. The value in the parenthesis indicates if it's mandatory or optional to send the data block. For example, the value 1 of 3 (M), (where the three data blocks are A, B, and C), indicate that either A, B, or C will be sent, but not all.

Comments: 1. *Additional information that relates to this response message.*

Lost Message: • *States what will happen if this operation is lost.*

8.1 Node Operations

- MIS OSSAIN Inform
- Node Alarm Inform
- Node Audit Request
 - Node Audit Success Response
 - Node Audit Error Response
- Node Busy Request
 - Node Busy Success Response
 - Node Busy Error Response
- Node Log Inform
- Node Return To Service Request
 - Node Return To Service Success Response
 - Node Return To Service Error Response
- Node Test Request
 - Node Test Success Response
 - Node Test Error Response

8.1.1 MIS OSSAIN Inform

Identifier: #010D

Description: The inform message provides the service node with event driven data about OSSAIN agents and call queues.

Sender(s):

- Standalone Switch
- Maintenance Switch

Message Class: MIS

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Information ID DB	14 bytes
0 to 253	OSSAIN Non-Call Session Event DB	16 bytes
0 to 92	OSSAIN Call Queue Event DB	44 bytes
0 to 72	OSSAIN Session Event DB	56 bytes

Comments: 1. The service node should not respond to this inform message. Any responses will be ignored by the switch.

8.1.2 Node Alarm Inform

Identifier: #010E

Description: The service node is informing the DMS to generate an alarm.

Sender(s): • Service Node

Message Class: Node Alarm

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Alarm Information DB	6 bytes
1 to 3	Report Text DB	86 bytes

- Comments:**
1. Critical, major, and minor alarms are independent of each other.
 2. If a critical alarm is on and an inform is sent to turn the same alarm on again, a log will be generated with text. No other action will be taken.
 3. When an alarm is turned off, it is then cleared.
 4. The text from the Report Text DB is printed in the log in the order that the Report Text DBs are sent to the switch.

Lost Message: • Alarm will not be generated if this message is lost.

8.1.3 Node Audit Request

Identifier: #0112

Description: This message is used by the Maintenance Switch to verify the state of the target service node.

Preconditions:

- Service node must be in either the INSV (in service) or ISTB (in service trouble) state.
- The Maintenance Switch only sends this message to a service node if no call processing messages have been received from that service node within a period defined by AUDFREQ in table OANODINV.

Post Conditions:

- None

Side Effects:

- None

Response(s):

- Node Audit Success Response
- Node Audit Error Response

Sender(s):

- Maintenance Switch

Message Class: Node

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Time DB	10 bytes

Comments: 1. None.

Lost Message:

- If a service node fails to respond to the Node Audit Request message within a datafillable amount of time (field AUDRTIMR in table OANODINV), another Node Audit Request message is sent. Once the number of consecutive failures exceeds field AUDTRIES in table OANODINV, the service node state is changed to the SYSB state. This causes all calls that have sessions with the service node to be taken down. All session pool audits to the service node are halted. In addition to this, all INSV session pools on that service node are marked as CBSY (C-side busy).

8.1.3.1 Node Audit Success Response

Interpretations: This signifies that the node has received the Node Audit Request message and is in an INSV or ISTB state.

Message Class: Node

Message Type: Return Result

Data Block(s): None

Comments: 1. None

8.1.3.2 Node Audit Error Response

Interpretations: This signifies that the service node has encountered an error during its audit process.

Message Class: Node

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

- Comments:**
1. Once the number of consecutive failures exceeds field AUDTRIES in table OANODINV, the service node state is changed to the SYSB state.
 2. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the second Node Audit Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.1.4 Node Busy Request

Identifier: #0111

Description: This message is used by the Maintenance Switch to busy a service node or by the service node when it needs to take itself out of service.

Preconditions: • If the service node is either INSV, ISTB or SYSB and is requested to go MANB then a Node Busy Request message is sent to the service node.

Post Conditions: • None

Side Effects: • In service session pools on the target node are placed in a CBSY state.
• Existing calls that have sessions with the affected node are taken down.
• Prevents the routing of new calls to the target service node.

Response(s): • Node Busy Success Response
• Node Busy Error Response

Sender(s): • Maintenance Switch
• Service Node

Message Class: Node

Message Type: Invoke

Data Block(s): None

Comments: 1. If the service node is currently OFFL (offline) and is requested to go MANB (manual busy) then a Node Busy Request message is not sent to the service node.
2. The service node will go MANB regardless of response received. Should a response be received, any text message associated with that response will be displayed at the MAP and logged. If no response is received then an error message will be displayed and logged.
3. The service node is responsible for taking its session pools out of service and cleaning up any call processing resources.

Lost Message: • If a service node fails to send a response within 10 seconds to the Node Busy Request then a response time-out error message is displayed and logged at the Maintenance Switch. The Maintenance Switch will place the service node in the MANB state, move all associated session pools to a CBSY state and inform call processing that the session pools are not available for handling calls.

8.1.4.1 Node Busy Success Response

Interpretations: This response indicates that the service node has completed its processing to go into a busy state.

Message Class: Node

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes

Comments: 1. A maintenance text data block may accompany the Busy Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.1.4.2 Node Busy Error Response

Interpretations: An error occurred on the service node while trying execute its MANB processing.

Message Class: Node

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010C	Invalid state
010E	Maintenance in progress
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Node Busy Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.1.5 Node Log Inform

Identifier: #010F

Description: The service node is informing the DMS to generate a log.

Sender(s): • Service Node

Message Class: Node Log

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
1 to 3	Report Text DB	86 bytes

Comments: 1. The text from the Report Text DB is printed in the log in the order that the Report Text DBs are sent to the switch.

Lost Message: • Log will not be generated if message is lost.

8.1.6 Node Return To Service Request

Identifier: #0110

Description: This message is sent from the Maintenance Switch in order to bring a service node into service.

Preconditions: • Node is in a SYSB or MANB state.

Post Conditions: • The node state is set to ISTB

Side Effects: • Once the Node Return To Service Success Response message is received, the Maintenance Switch performs the following:

- Submits a request to RTS each CBSY session pool supported by the target service node
- Logs the event

Response(s): • Node Return To Service Success Response
• Node Return To Service Error Response

Sender(s): • Maintenance Switch

Message Class: Node

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Time DB	10 bytes

Comments:

1. This message is requested manually if the service node is in the MANB state or by an auto-RTS process in the Maintenance Switch if the service node is in the SYSB state.
2. The Maintenance Switch waits for a response from the service node. The response indicates whether the service node was able to successfully execute its RTS activities.

Lost Message:

- If the RTS message was manually initiated and the service node fails to reply to the Node Return To Service Request message (timeout value based on field RTSTIME in OANODINV), a response time-out error message will be displayed and logged at the Maintenance Switch. The Maintenance Switch will keep the service node in the MANB state.
- If the RTS message was system initiated (auto-RTS), the service node state will remain SYSB and no error messages will be displayed or logs generated.

8.1.6.1 Node Return To Service Success Response

Interpretations: This response indicates that the service node successfully executed its RTS activities.

Message Class: Node

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes

- Comments:**
1. This message indicates that the service node base software is operational and its session pools can now be brought into service.
 2. Upon receipt of this message, the Maintenance Switch performs the following tasks:
 - Sets the node state to ISTB
 - Submits a request to RTS each CBSY session pool supported by the service node
 - Logs the event.
 3. A maintenance text data block may accompany the Test Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.1.6.2 Node Return To Service Error Response

Interpretations: An error occurred while trying to RTS a service node. The included data block specifies the cause of the failure.

Message Class: Node

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Node RTS Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.1.7 Node Test Request

Identifier: #0113

Description: This message indicates that the service node is to execute its set of maintenance diagnostics.

Preconditions: • The node is in a MANB state.

Post Conditions: • None.

Side Effects: • None.

Response(s): • Node Test Success Response
• Node Test Error Response

Sender(s): • Maintenance Switch

Message Class: Node

Message Type: Invoke

Data Block(s): None

Comments: 1. None

Lost Message: • If the service node fails to reply to the Node Test Request message in the period defined by TSTTIME in table OANODINV, a response time-out error message will be displayed and logged at the Maintenance Switch. The service node will remain in the MANB state.

8.1.7.1 Node Test Success Response

Interpretations: This response indicates that the service node successfully executed its set of diagnostics.

Message Class: Node

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes

Comments: 1. A maintenance text data block may accompany the Test Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.1.7.2 Node Test Error Response

Interpretations: This response indicates that the service node failed to execute its set of diagnostics.

Message Class: Node

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Node Test Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.2 Session Pool Maintenance Operations

- Session Pool Alarm Inform
- Session Pool Audit Request
 - Session Pool Audit Success Response
 - Session Pool Audit Error Response
- Session Pool Busy Request
 - Session Pool Busy Success Response
 - Session Pool Busy Error Response
- Session Pool Drain Inform
- Session Pool Log Inform
- Session Pool Return To Service Request
 - Session Pool Return To Service Success Response
 - Session Pool Return To Service Error Response
- Session Pool State Inform
- Session Pool Test Request
 - Session Pool Test Success Response
 - Session Pool Test Error Response
- Session Pool Throttle Operations:
 - Session Pool Throttle Success Response
 - Session Pool Throttle Error Response

8.2.1 Session Pool Alarm Inform

Identifier: #0307

Description: A session pool is informing the DMS to generate an alarm and a log report.

Sender(s): • Service Node

Message Class: Session Pool Alarm

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Alarm Information DB	6 bytes
1 to 3	Report Text DB	86 bytes

- Comments:**
1. All alarms are independent of each other.
 2. If a critical alarm is on and an inform is sent to turn the same alarm on again, a log will be generated with text.
 3. When an alarm is turned off, it is then cleared.

Lost Message: • Alarm will not be generated if message is lost.

8.2.2 Session Pool Audit Request

Identifier: #0302

Description: This message is used by the Maintenance Switch to determine if the target session pool is in service.

Preconditions: • The Maintenance Switch only sends this message to a session pool if:

- the session pool is in service
- no call processing messages have been received from that session pool within a period defined by AUDFREQ in table OANODINV.

Post Conditions: • None

Side Effects: • None

Response(s): • Session Pool Audit Success Response
• Session Pool Audit Error Response

Sender(s): • Maintenance Switch

Message Class: Session Pool

Message Type: Invoke

Data Block(s): None

Comments: 1. None.

Lost Message: • If a session pool fails to respond to the Session Pool Audit Request message within a datafillable amount of time (field AUDRTIMR in table OANODINV), another Session Pool Audit Request message will be sent. Once the number of consecutive failures exceeds field AUDTRIES in table OANODINV, the session pool state is set to SYSB. This causes all calls that have sessions with the session pool to be taken down. All session pool audits to the session pool are halted. The service node will go ISTB.

8.2.2.1 Session Pool Audit Success Response

Interpretations: This signifies that the subject session pool is in an INSV (in service) state.

Message Class: Session Pool

Message Type: Return Result

Data Block(s): None

Comments: 1. None

8.2.2.2 Session Pool Audit Error Response

Interpretations: This indicates that the session pool encountered an error during its audit process.

Message Class: Session Pool

Message Type: Return Error

Data Block(s):

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

- Comments:**
1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the second Session Pool Audit Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.
 2. The Maintenance Switch treats this message as if no response from the session pool was received and will send another Session Pool Audit Request message. Once the number of consecutive failures exceeds field AUDTRIES in table OANODINV, the session pool state is set to SYSB and all calls associated with that session pool will be taken down.

8.2.3 Session Pool Busy Request

Identifier: #0301

Description: A Maintenance Switch issues this message when it wishes to busy a session pool or a service node sends this when it wants to take a session pool out of service.

Preconditions: • The Session pool is INSV (In Service)

Post Conditions: • Existing calls that have sessions with the target session pool are taken down
• Target session pool is placed in a MANB state

Side Effects: • If the service node associated with the session pools is in the INSV state, it is changed to an ISTB state.
• Session pool audits to the target session pool are halted

Response(s): • Session Pool Busy Success Response
• Session Pool Busy Error Response

Sender(s): • Maintenance Switch
• Service Node

Message Class: Session Pool

Message Type: Invoke

Data Block(s): None

Comments: 1. The session pool application on the service node is responsible for cleaning up its own resources (i.e. calls, etc.).
2. The Maintenance Switch waits for a Session Pool Busy Response message from the service node before continuing with its busy session pool processing.
3. If the session pool goes from the CBSY, SYSB or OFFL state to the BSY state, no message is sent to the session pool.

Lost Message: • If a session pool fails to send a response within 10 seconds to the Session Pool Busy Request, a response time-out error is displayed and logged at the DMS MAP. The Maintenance Switch will mark the session pool as MANB.

8.2.3.1 Session Pool Busy Success Response

Interpretations: This response indicates that the session pool was able to execute the activities required to go into a busy state.

Message Class: Session Pool

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes

Comments: • A maintenance text data block may accompany the Busy Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.2.3.2 Session Pool Busy Error Response

Interpretations: An error occurred on the service node while trying to busy the target session pool. The included data block specifies the cause of the failure.

Message Class: Session Pool

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010C	Invalid state
010E	Maintenance in progress
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Session Pool Busy Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.2.4 Session Pool Drain Inform

Identifier: #0305

Description: This inform message notifies a session pool when a `drain` command is entered by a DMS user.

Sender(s): • Maintenance Switch

Message Class: Session Pool

Message Type: Invoke

Data Block(s): None

Comments: 1. The drain command overrides requests to change the number of available sessions. Once a MAP user enters the drain command, the number of available sessions is set to zero, regardless of the number of available sessions. A busy and return to service must be entered at the switch to bring up the number of available sessions for the session pool.

Lost Message: • If this operation is lost, the node will not receive this inform message.
• If the service node never receives the Drain inform (i.e. lost) and continues to initiate calls to the switch, the switch will generate a log and ignore the call processing messages.

8.2.5 Session Pool Log Inform

Identifier: #0306

Description: A session pool is informing the DMS to generate a log.

Sender(s): • Service Node

Message Class: Session Pool Log

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
1 to 3	Report Text DB	86 bytes

Comments: 1. None

Lost Message: • Log will not be generated if message is lost.

8.2.6 Session Pool Return To Service Request

Identifier: #0300

Description: This message is sent by the Maintenance Switch when it wants to bring a session pool into service.

Preconditions:

- Node is in a ISTB state,
- Session pool is MANB, CBSY or SYSB

Post Conditions:

- The session pool is set to INSV

Side Effects:

- None

Response(s):

- Session Pool Return To Service Success Response
- Session Pool Return To Service Error Response

Sender(s):

- Maintenance Switch

Message Class: Session Pool

Message Type: Invoke

Data Block(s): None.

Comments:

1. The Maintenance Switch waits for a response from the session pool before continuing with its RTS processing.
2. Upon receipt of a Session Pool Return To Service Success Response message, the Maintenance Switch will mark the session pool as being INSV. If all session pools on the service node are in service, the state of the service node is changed from ISTB to INSV.
3. Upon receipt of a Session Pool Return To Service Error Response or time out on the outstanding request, the Maintenance Switch logs the error condition and moves the session pool state to SYSB.

Lost Message:

- A session pool that fails to send a response to the Session Pool Return to Service Request is placed in the SYSB state (timeout value based on field RTSTIME in Table OANODINV). If the return to service was manually requested, the Maintenance Switch displays and logs a response time-out error condition.

8.2.6.1 Session Pool Return To Service Success Response

Interpretations: This response indicates that the service node successfully executed its session pool RTS activities.

Message Class: Session Pool

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes
O	Throttle DB	6 bytes

Comments: 1. A maintenance text data block may accompany the Return to Service Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.2.6.2 Session Pool Return To Service Error Response

Interpretations: An error occurred while trying to RTS a session pool. The included data block specifies the cause of the failure.

Message Class: Session Pool

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Session Pool RTS Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.2.7 Session Pool State Inform^{SV3}

Identifier: #0308

Description: An OSAC Remote switch is informing the service node about the state of a session pool at the switch. The state at the Remote is independent of the state of the session pool at the OSAC Host.

Sender(s): • OSAC Remote

Message Class: Session Pool

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Maintenance State DB	6 bytes

Comments: 1. The service node should only initiate sessions with the switch on a session pool that is in service with the OSAC Host and has received this inform from the OSAC Remote with an in service value for the maintenance state.

8.2.8 Session Pool Test Request

Identifier: #0303

Description: This message indicates that a session pool is to execute its set of maintenance diagnostics.

Preconditions: • The session pool is in a MANB state.

Post Conditions: • None

Side Effects: • None

Response(s): • Session Pool Test Success Response
• Session Pool Test Error Response

Sender(s): • Maintenance Switch

Message Class: Session Pool

Message Type: Invoke

Data Block(s): None

Comments: 1. None.

Lost Message: • If a session pool that fails to send a response to the Session Pool Test Request a response time-out error is displayed and logged at the Maintenance Switch (timeout value based on field TSTTIME in Table OANOD-INV).

8.2.8.1 Session Pool Test Success Response

Interpretations: This response indicates that the service node successfully executed its set of diagnostics for the session pool.

Message Class: Session Pool

Message Type: Return Result

Data Block(s):

M/O	Data Blocks	Byte Length
O	Maintenance Text DB	38 bytes

Comments: 1. A maintenance text data block may accompany the Test Success response. Any message in the maintenance text data block will be displayed at the MAP and logged.

8.2.8.2 Session Pool Test Error Response

Interpretations: This response indicates that the Session Pool failed to execute its set of diagnostics.

Message Class: Session Pool

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010B	Invalid unsolicited operation
0110	Maintenance Operation Failed

Comments: 1. A Maintenance Text data block is associated with error ID #0110. Upon receipt of the Session Pool Test Error Response, the Maintenance Switch will generate a log which will include the text from the Maintenance Operation Failed text field.

8.2.9 Session Pool Throttle Request

Identifier: #0304

Description: A session pool is requesting to change the number of available sessions.

Preconditions:

- Session Pool must be in service.
- No other maintenance activity should be in progress.
- Session pool must be datafilled in switch Table OASESNPL as ORIGTYPE = subscriber and not used only for trigger event inform messages or datafilled as ORIGTYPE = SN.

Post Conditions:

- Number of available sessions has been changed.

Side Effects:

- None

Response(s):

- Session Pool Throttle Success Response
- Session Pool Throttle Error Response

Sender(s):

- Service Node

Message Class: Session Pool

Message Type: Invoke

Data Block(s):

M/O	Data Blocks	Byte Length
M	Throttle DB	6 bytes

Comments:

1. When an unsolicited maintenance message is received by OSSAIN data communications, a buffering mechanism is used for storing the message while the request is waiting in the maintenance request queue. If there are no buffers available, the request is discarded.
2. Only one unsolicited maintenance request can be processed at a time. Therefore, a response should be received before sending another request.

Duplicate Request:

- There are no problems with this operation being sent twice. It would just simply change the number of available sessions again.

8.2.9.1 Session Pool Throttle Success Response

Interpretations: The number of session pools has been changed.

Message Class: Session Pool

Message Type: Return Result

Data Block(s): None

8.2.9.2 Session Pool Throttle Error Response

Interpretations: The number of session pools could not be updated.

Message Class: Session Pool

Message Type: Return Error

ErrorID Values:

ErrorID (#)	Value
010A	Operation invalid for session pool type
010C	Invalid state
010D	Throttle request greater than MAXSESN
010E	Maintenance in progress
010F	Drained command entered

Comments: 1. None.

Chapter 9: Data Block Descriptions

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Alarm Information DB	6 bytes	• Node Alarm Inform • Session Pool Alarm Inform
Alternate Billing Directory Number DB	18 bytes	• Accept Control Inform • Billing Number Request • Call Details Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Trigger Event Inform
AMA Module DB	78 bytes	• Append AMA Module Request
ASCII Card Number DB	38 bytes	• Accept Control Inform • Billing Number Request • Call Details Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Trigger Event Inform
BCD Card Information DB	6 bytes	• Billing Number Success Response • Session Initiation Success Response
BCD Card Number DB	24 bytes	• Accept Control Inform • Billing Number Request • Call Details Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Trigger Event Inform
BNS Query Result DB	6 bytes	• Cass Charge Success Response
Billing Acceptance DB	6 bytes	• Call Float Request • End Call Request • Transfer To Control List Request
Billing Information DB	10 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Directory Number Success Response • Service Change Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform
Billing Number Input DB	6 bytes	• Class Charge Request
Billing Number Validation DB	10 bytes	• Class Charge Success Response
Branding Indicator DB	6 bytes	• Call Float Request • Call Resume Request • Transfer To Control List Request
Broadcast DB	8 bytes	• Voice Connect Request
Broadcast Count DB	6 bytes	• Voice Connect Error Response • Voice Connect Success Response

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
CCV Query Result DB	6 bytes	• Class Charge Success Response
Call Float Indicator DB	6 bytes	• Call Float Request
Call In Queue DB	6 bytes	• Call Float Request
Call Information DB	20 bytes	• Accept Control Inform • Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Call Queue Information DB	14 bytes	• Passive Function Provider Error Response • Passive Function Provider Success Response
Call Timing	18 bytes	• Call Merge Inform • Connection Status Inform • End Call Success Response • Notification Timer Inform • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Call Update DB	6 bytes	• Call Float Request • Transfer to Control List Request
Callid DB	8 bytes	• Call Merge Request • Session Recall Request
Carrier DB	20 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Carrier Assignment Success Response • Directory Number Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform
Carrier Number DB	8 bytes	• Carrier Assignment Request • Session Initiation Request • Transfer to Carrier Request
Charge Status DB	6 bytes	• Accept Control Inform • Call Details Success Response • Call Float Request • Generate AMA Record Request • Route to Treatment Request • Service Change Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Transfer to Carrier Request • Transfer to Control List • Trigger Event Inform

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Class Charge DB	10 bytes	• Accept Control Inform • Call Details Success Response • Class Charge Request • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Conference ID DB	8 bytes	• Call Details Success Response • Call Merge Error Response • Conference Add Request • Conference Create Error Response • Conference Create Success Response • Conference Details Request • Conference Release Request • Connect DN Request • Session Begin Inform • Session Recall Success Response • Trigger Event Inform • Voice Connect Request
Conference Party Connection DB	10 bytes	• Call Details Success Response • Call Float Error Response • Call Float Request • Conference Add Error Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Conference Party Connection Information DB	12 bytes	• Conference Details Success Response
Conference SN Voice Connection DB	6 bytes	• Call Details Success Response • Conference Add Error Response • Conference Add Request • Conference Create Request • Conference Remove Error Response • Conference Remove Request • Session Begin Inform
Conference SN Voice Connection Information DB	8 bytes	• Conference Details Success Response
Connection Event DB	6 bytes	• Connection Status Inform
Connection ID DB	6 bytes	• Call Merge Error Response • Conference Add Request • Conference Create Request • Conference Remove Error Response • Conference Remove Request • LNP Request • Release DN Request
Connection Progress DB	8 bytes	• Connect DN Success Response

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Context Block DB	56 bytes	• Accept Control Inform • Call Details Success Response • Call Float Request • Call Resume Request • Session Begin Inform • Session Recall Success Response • Transfer to Control List Request • Trigger Event Inform
Conversation Timing Indicator DB	6 bytes	• Conversation Timing Request
Country Direct Information DB	14 bytes	• Accept Control Inform • Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
DTMF Digit DB	6 bytes	• DTMF Digit Detected Inform
Directory Number DB	34 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Directory Number Information DB	8 bytes	• Directory Number Success Response • Session Initiation Success Response
Directory Number Update DB	30 bytes	• Directory Number Request • Session Initiation Request
End Call Handling DB	6 bytes	• End Call Request
Estimate of Charges Alternate Date DB	6 bytes	• Estimate of Charges Request
Estimate of Charges Alternate Time DB	6 bytes	• Estimate of Charges Request
Estimate of Charges Duration DB	6 bytes	• Estimate of Charges Request
Estimate of Charges Duration Information DB	8 bytes	• Estimate of Charges Success Response • Accept Control Inform • Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Estimate of Charges General Information DB	10 bytes	• Estimate of Charges Success Response • Accept Control Inform • Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Estimate of Charges No Duration Information DB	16 bytes	- Estimate of Charges Success Response - Accept Control Inform - Call Details Success Response - Session Begin Inform - Session Recall Success Response - Trigger Event Inform
Function DB	6 bytes	- Accept Control Inform - Call Details Success Response - Passive Function Provider Request - Session Begin Inform - Session Recall Success Response - Trigger Event Inform
Generate AMA DB	6 bytes	Generate AMA Record Request
ISUP Calling Party Number DB	16 bytes	- Accept Control Inform - Call Details Success Response - ISUP Calling Party Number Request - Session Begin Inform - Session Recall Success Response - Trigger Event Inform
Information ID DB	14 bytes	MIS OSSAIN Inform
Invalid Billing Details	6 bytes	Call Float Error Response
LATA Information	6 bytes	- Accept Control Inform - Call Details Success Response - Call Merge Inform - Directory Number Success Response - Session Begin Inform - Session Initiation Success Response - Session Recall Success Response - Trigger Event Inform
LNP Information DB	20 bytes	- Accept Control Inform - Billing Number Success Response - Call Details Success Response - Call Merge Inform - Class Charge Success Response - Connect DN Success Response - Directory Number Success Response - LNP Success Response - Session Begin Inform - Session Initiation Success Response - Session Recall Success Response - Trigger Event Inform
LNP Enhancements DB	6 bytes	- Billing Number Request - Connect DN Request - Directory Number Request - Session Initiation Request
LRN Update DB	12 bytes	LRN Update Request
Language	6 bytes	- Call Float Request - Transfer To Control List Request

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Maintenance State	6 bytes	• Session Pool State Inform
Maintenance Text DB	38 bytes	• Node Audit Error Response • Node Busy Success Response • Node Busy Error Response • Node Return to Service Success Response • Node Return to Service Error Response • Node Test Success Response • Node Test Error Response • Session Pool Audit Error Response • Session Pool Busy Success Response • Session Pool Busy Error Response • Session Pool Return to Service Success Response • Session Pool Return to Service Error Response • Session Pool Test Success Response • Session Pool Test Error Response
NBEC DB	8 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Directory Number Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform
Network Service DB	6 bytes	• Session Initiation Request
Node Identification DB	8 bytes	• Accept Control Inform • Call Details Success Response • Passive Function Provider Success Response • Passive Node Status Inform • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Notification Timer DB	10 bytes	• Notification Timer Request
OLNS Information DB	50 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
OSSAIN Call Queue Event DB	44 bytes	• MIS OSSAIN Inform
OSSAIN Session Event DB	56 bytes	• MIS OSSAIN Inform
OSSAIN Non-Call Session Event DB	16 bytes	• MIS OSSAIN Inform

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Operator Identification DB	8 bytes	• Call Details Success Response • Passive Function Provider Success Response • Passive Node Status Inform • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Operator Services System Action DB	6 bytes	• Call Float Request • End Call Request • Transfer To Control List Request
Operator System DB	6 bytes	• Call Resume Request
Originating Station Information DB	22 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Directory Number Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform
Originating Trunk Information DB	8 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Directory Number Request • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Trigger Event Inform
Pass Through DB	106 bytes	• Pass Through Inform
Passive Node Status DB	6 bytes	• Passive Node Status Inform
Preprocessing Information DB	8 bytes	• Call Details Success Response • Session Begin Inform
RLT Request DB	6 bytes	• Call Float Request • Transfer To Carrier Request
Release Cause DB	6 bytes	• Connect DN Error Response • Connection Status Inform
Release Node DB	6 bytes	• Release Node Request
Report Text DB	86 bytes	• Node Alarm Inform • Node Log Inform • Session Pool Alarm Inform • Session Pool Log Inform
Requested Service DB	6 bytes	• Service Change Request

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Routing Index DB	8 bytes	• Accept Control Inform • Billing Number Request • Call Details Success Response • Call Float Request • Connect DN Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Transfer to Carrier Request • Trigger Event Inform
Routing Number DB	16 bytes	• Accept Control Inform • Billing Number Request • Call Details Success Response • Call Float Request • Connect DN Success Response • Session Begin Inform • Session Initiation Request • Session Recall Success Response • Trigger Event Inform
SN Voice Link DB	12 bytes	• Accept Control Inform • Call Details Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Voice Connect Success Response • Voice Connect Error Response
SPID DB	12 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Directory Number Success Response • Service Provider ID Assignment Request • Session Begin Inform • Session Initiation Request • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform
Sequence Number Check DB	6 bytes	• Call Float Request • End Call Request
Session Information DB	8 bytes	• Session Recall Error Response
Speech Path DB	6 bytes	• Connect DN Request • Speech Path Request
Terminating Trunk Information DB	8 bytes	• Accept Control Inform • Call Details Success Response • Connect DN Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Text DB	26 bytes	• Call Resume Request • Text To Operator Request • Transfer to Control List Request
Throttle DB	6 bytes	• Session Pool Return to Service Success Response • Session Pool Throttle Request
Time DB	10 bytes	• Call Details Success Response • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Trigger Event Inform • Node Audit Request • Node Return to Service Request
Transfer Control List DB	8 bytes	• Transfer to Control List Request
Treatment DB	6 bytes	• Accept Control Inform • Call Details Success Response • Call Merge Inform • Carrier Assignment Error Response • Connect DN Error Response • Connection Status Inform • Directory Number Success Response • Route To Treatment Request • Session Begin Inform • Session Initiation Success Response • Session Recall Success Response • Transfer to Carrier Error Response • Trigger Event Inform
Trigger DTMF DB	8 bytes	• Call Details Success Response • Session Begin Inform • Session Recall Success Response
Trigger Information DB	6 bytes	• Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Trigger Profile Index DB	14 bytes	• Call Float Request • Transfer To Control List Request
Trigger Release Cause DB	8 bytes	• Call Details Success Response • Session Begin Inform • Session Recall Success Response
Trigger Ringing DB	8 bytes	• Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform
Trigger Treatment DB	6 bytes	• Call Details Success Response • Session Begin Inform • Session Recall Success Response • Trigger Event Inform

Table 9: Data Blocks to Operations Mapping Table

Data Blocks	Data Block Size	Operations
Voice Channel DB	6 bytes	• Voice Connect Error Response • Voice Connect Request

9.0 Data Block Template

Identifier:

Description:

Size: bytes

Data Fields: **Identifier:**

#7F

DB Byte Count:

0 to 65535

Data Block Field 1:

Field 1 Range

Data Block Field 2:

Field 2 Range

Comments: 1.

Byte #	Meaning
0	ID = #
1	
2	DB Byte Count
3	
4	Field 1
5	Field 2

9.1 Alarm Information DB

Identifier: #13B

Description: This data block informs the DMS what type of alarm should turn on or off.

Size: 6 bytes

Data Fields: **Identifier:**

#13B

DB Byte Count:

0 to 65535

Alarm Status:

0 = On

1 = Off

Alarm Severity:

0 = Critical

1 = Major

2 = Minor

Comments: 1. None.

Byte #	Meaning
0	ID = #13B
1	
2	DB Byte Count
3	
4	Alarm Status
5	Alarm Severity

9.2 Alternate Billing Directory Number DB

Identifier: #102

Description: This data block contains information on the billing number which is in the form of a directory number.

Size: 18 bytes

Data Fields: **Identifier**

#102

DB Byte Count:

0 to 65535

Billing Number Type:

0 = Alternate Directory Number

1 = Collect

2 = Account Code

Language:

0 to 99

254 = Unknown

Directory Number Type:

0 = Foreign

1 = Domestic Non-local

2 = Domestic Local

3 = Outgoing Trunk

4 = Service

254 = Unknown

Directory Number Digit Count:

0 to 20

Directory Number:

BCD digits, ranging from 0 to 9

- Comments:**
1. If the call is being billed to the forward party (i.e. class charge is collect) and the directory number for the forward party is already stored in the switch, then this data block does not need to be sent to the switch (unless other information needs to be updated, such as Language).
 2. The Language and Directory Number Type fields are only applicable if the Billing Number Type is Alternate Directory Number or Collect.
 3. If this data block is sent by the service node to the switch and the value in the Language field differs from what is stored in the switch, then the value in this data block overwrites what is stored in the switch.

4. Directory numbers are validated using BNS queries to the LIDB, if DMS datafill specifies it.
5. If the service node sends this data block with a Directory Number Digit Count of 0, the switch will nil out the alternate billing number stored in the switch and set the class charge to unspecified.
6. The Class Charge Request Operation specifies whether the service node wants the switch to validate the billing number. The following lists the appropriate class charge for the billing number:
 - If the billing number is an alternate directory number (e.g. bill to third number), the class charge must be set to Station Special Calling.
 - If the billing number is a collect directory number, the class charge must be set to Station Collect or Person Collect.
 - If the billing number is an account code, the class charge must be set to Station Paid.
7. A value of Foreign for the Directory Number Type field indicates that the directory number must be prepended with a valid country code.
8. A Foreign alternate directory number must start with the "011" access code and then have the country code and directory number.
9. Exactly four digits must be sent for a billing number of type Account Code. If the account code number is smaller than four digits, the billing number must be padded in front with zeros (e.g. 0123 for account code of 123).
10. Directory Numbers for domestic calls under the North American numbering plan must be 10 digit numbers (i.e. for Directory Number Type of Domestic and switch datafilled as North American Dialing Plan, the Directory Number Digit Count, for Alternate Directory Number or Collect, should equal 10). Zero, to clear the number, is the only other value for Directory Number Digit Count that will be accepted by the switch. Directory numbers for foreign calls under the North American numbering plan (Billing Number Type of Collect) have a maximum of 15 digits. Directory numbers for all calls and parties under the Open numbering plan have a maximum of 18 digits.

Byte #	Meaning
0	ID = #102
1	
2	DB Byte Count
3	
4	Billing Number Type
5	Language
6	Directory Number Type
7	Directory Number Digit Count
8	Directory Number
9	
10	
11	
12	
13	
14	
15	
16	
17	

9.3 AMA Module DB

Identifier: #103

Description: Formatted billing information from the service node.

Size: 78 bytes

Data Fields: **Identifier:**

#103

DB Byte Count:

0 to 65535

Append AMA Sequence Number:

0 to 254

AMA Handling:

0 = Do Not Clear Previously Appended AMA

1 = Clear Previously Appended AMA

AMA Hex Digit Count (in digits):

0 to 140

AMA Data:

Up to 140 hex digits of AMA billing data.

- Comments:**
1. The switch records the formatted billing data into one of two Bellcore AMA Modules: Service Node Formatted-Small, module code 186 or Service Node Formatted-Large, module code 187. The Bellcore AMA Module chosen is dependent on the size of the formatted AMA data sent in this data block. The two sizes of AMA modules supported are 40 and 140 hex digits. If the formatted billing data sent is not on one of the size boundaries (i.e. 40 or 140), the data will be padded to the end with nil (#F).
 2. The Append AMA Sequence Number is intended as a unique identifier to avoid instances of lost or duplicate billing. This sequence number should be initialized to '1' at the start of every session and incremented with each Append AMA DB. The switch expects the Append AMA Sequence Number to increment with every request within a session. The switch will process AMA Module DBs only if their sequence number is equal to (or greater than) the sequence number expected. Sequence numbers that are less than that expected will return an 'Invalid append AMA sequence number' error. Note that the Append AMA Sequence Number is independent of the Generate AMA Sequence Number (a Generate AMA Record does not affect the sequencing of the Append AMA Sequence Number).
 3. The AMA buffer is limited to 1600 hex digits (800 bytes) of 186 and 187 modules. If too many AMA modules are appended (i.e. the 1600 hex digit

buffer size is exceeded), an AMA buffer overflow error will be returned, and the AMA module will not be appended. A subsequent Generate AMA operation will, if successfully processed, generate a billing record with any Append AMA Modules, and clear the AMA buffer for this call. This will allow the AMA buffer to be cleared so that the failed append can be resubmitted.

4. If the AMA Handling field is set to 'Clear Previously Appended AMA' then the DMS will remove any previously stored custom AMA data from its buffer (i.e. the removed data will not be appended to any AMA record). The new AMA Data in the current Append AMA will be put into the buffer after the prior data has been cleared (to clear the buffer without introducing any new AMA, the AMA Hex Digit Count field should be set to 0).
5. There is a method defined for identifying the format of the custom AMA data. Refer to the TOPS Custom AMA document (NIS Q259-1).

Byte #	Meaning
0	ID = #103
1	
2	DB Byte Count
3	
4	Append AMA Sequence Number
5	AMA Handling
6	Padding Field
7	AMA Hex Digit Count
8	AMA Data
9	
10	
11	
12	
13	
14	
...	
77	

9.4 ASCII Card Number DB

Identifier: #113

Description: This is the ASCII format card number. If the billing number contains ASCII characters, this data block must be used.

Size: 38 bytes

Data Fields:

Identifier:	#113
DB Byte Count:	0 to 65535
ASCII Card Number Type:	0 = Foreign 1 = Custom
ASCII PIN Length:	0 to 30
ASCII Card Number Length:	0 to 30
ASCII Card Number:	ASCII characters

- Comments:**
1. Refer to the field definition of ASCII Card Number Type for the valid formats of the card types
 2. The PIN is included in the ASCII card number at the end. The ASCII card number length includes the PIN.
 3. If the billing number is a card number, the class charge must be set to Station Special Calling/Called.
 4. No database queries are done by the DMS for ASCII Cards.
 5. The PIN, as specified in this data block, is at the end of the card number and will not be displayed in AMA records.
 6. If the service node sends this data block with an ASCII Card Number Length of zero, the switch will nil out the card number stored in the switch and set the class charge to unspecified.
 7. Currently, no more than 23 digits for the ASCII card number and pin number is supported.

Byte #	Meaning
0	ID = #113
1	
2	DB Byte Count
3	
4	ASCII Card Number Type
5	Padding Field
6	ASCII PIN Length
7	ASCII Card Number Length
8	ASCII Card Number
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	

9.5 BCD Card Information DB

Identifier: #120

Description: This data block contains the type of BCD card number being used for billing.

Size: 6 bytes

Data Fields: **Identifier**

#120

DB Byte Count:

0 to 65535

BCD Card Number Type:

0 = ISO

1 = Directory Number Card

2 = RAO Number Card

3 = Custom Card

254 = Unknown

Comments: 1. Refer to the Field Definition on BCD Card Number Type for the formats for each type of Card Number.

Byte #	Meaning
0	ID = #120
1	
2	DB Byte Count
3	
4	BCD Card Number Type
5	Padding Field

9.6 BCD Card Number DB

Identifier: #108

Description: This data block contains information on the BCD card number being used for billing.

Size: 24 bytes

Data Fields: **Identifier**

#108

DB Byte Count:

0 to 65535

BCD Card Number Type:

0 = ISO

1 = Directory Number Card

2 = RAO Number Card

3 = Custom Card

254 = Unknown

BCD PIN Digit Count:

0 to 32

BCD Card Number Digit Count:

0 to 32

BCD Card Number:

BCD digits, ranging from 0 to 9

- Comments:**
1. Refer to the Field Definition on BCD Card Number Type for the valid formats for each type of BCD Card Number.
 2. Directory numbers are validated using BNS queries to the LIDB, if DMS datafill specifies it.
 3. Calling card numbers are validated using CCV queries to the LIDB or RIDB, if DMS datafill specifies it.
 4. The Class Charge Request Operation specifies whether the service node wants the switch to validate the billing number. If the billing number is a card number, the class charge must be set to Station Special Calling.
 5. The PIN, as specified in this data block, is at the end of the card number and will not be displayed in AMA records.
 6. If the service node sends this data block with a BCD Card Number Digit Count of zero, the switch will nil out the card number stored in the switch and set the class charge to unspecified.
 7. If the BCD Card Number Type is set to Unknown, then the BCD PIN Digit Count field is ignored by the switch.

8. Currently, no more than 23 digits for the BCD card number and pin number is supported.

Byte #	Meaning
0	ID = #108
1	
2	DB Byte Count
3	
4	BCD Card Number Type
5	Padding Field
6	BCD PIN Digit Count
7	BCD Card Number Digit Count
8	BCD Card Number
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	

9.7 BNS Query Result DB

Identifier: #12C

Description: This data block indicates some of the results of a BNS LIDB query.

Size: 6 bytes

Data Fields: **Identifier:**

#12C

DB Byte Count:

0 to 65535

BNS Query Response:

0 = Miscellaneous

1 = Vacant Code

2 = Semi-Public Coin Phone

3 = Public Coin Phone

4 = Public Coinless Phone

5 = Non Working Number

Comments: 1. This is a sub-list of query results.

Byte #	Meaning
0	ID = #12C
1	
2	DB Byte Count
3	
4	BNS Query Response
5	Padding Field

9.8 Billing Acceptance DB

Identifier: #130

Description: This data block indicates the method of billing acceptance.

Size: 6 bytes

Data Fields: **Identifier:**

#130

DB Byte Count:

0 to 65535

Billing Acceptance Response Method:

0 = DTMF

1 = Verbal to Operator

2 = Derived

3 = Voice Recognition

254 = Unknown

Comments: 1. None.

Byte #	Meaning
0	ID = #130
1	
2	DB Byte Count
3	
4	Billing Acceptance Response Method
5	Padding Field

9.9 Billing Information DB

Identifier: #104

Description: Billing information for the current call.

Size: 10 bytes

Data Fields: **Identifier:**

#104

DB Byte Count:

0 to 65535

Network Service ID:

0 to 32766

Allowed Billing Set:

Bit 0 = Station Paid
Bit 1 = Station Collect
Bit 2 = Station Special Calling Card
Bit 3 = Station Special Calling 3rd Number
Bit 4 = Station Special Called Card
Bit 5 = Station Special Called 3rd Number
Bit 6 = Person Paid
Bit 7 = Person Collect
Bit 8 = Person Special Calling Card
Bit 9 = Person Special Calling 3rd Number
Bit 10 = Person Special Called Card
Bit 11 = Person Special Called 3rd Number

Restricted Billing Index:

0 to 100

Terminating Code Allowed Set:

Bit 0 = Card
Bit 1 = Collect
Bit 2 = Third Number

- Comments:**
1. The Network Service ID is obtained from switch table OACTLDEF and is based on the control list identifier. It can also be set by the service node with a Session Initiation Request. For OSSAIN preprocessing calls, the Network Service ID comes from switch table OAINPRE.
 2. For calls that trigger to OSSAIN without obtaining a Network Service ID previously, the max Network Service ID (32766) is used. For example, if a call that is floated by an operator then subsequently triggers to a service node via an OSSAIN function would not have obtained a Network Service ID from table OACTLDEF, so the max Network Service ID is used.

3. A Restricted Billing Index of 100 is the 'nil' value (i.e. unrestricted originator).

Byte #	Meaning
0	ID = #104
1	
2	DB Byte Count
3	
4	Network Service ID
5	
6	Allowed Billing Set
7	
8	Restricted Billing Index
9	Terminating Code Allowed Set

9.10 Billing Number Input DB

Identifier: #131

Description: This data block indicates the billing number input method.

Size: 6 bytes

Data Fields: **Identifier:**

#131

DB Byte Count:

0 to 65535

Billing Number Input Method:

0 = DTMF

1 = Verbal to Operator

2 = Derived

3 = Voice Recognition

254 = Unknown

Comments: 1. None.

Byte #	Meaning
0	ID = #131
1	
2	DB Byte Count
3	
4	Billing Number Input Method
5	Padding Field

9.11 Billing Number Validation DB

Identifier: #105

Description: This data block contains the results of a billing number validation.

Size: 10 bytes

Data Fields: **Identifier:**

#105

DB Byte Count:

0 to 65535

Validation Sequence Number:

0 to 254

Validation Result:

0 = Validation Not Performed

1 = Billing Denied

2 = Manual Validation Required

3 = Billing Acceptance Required

4 = Billing Accepted

5 = No Mutual Honoring Agreement

Validation Result Determined by DMS Datafill:

0 = Validation Result NOT Determined by DMS Datafill

1 = Validation Result Determined by DMS Datafill

RAO Digit Count:

0 to 3

RAO Code:

3 BCD Digits

Byte #	Meaning
0	ID = #105
1	
2	
3	DB Byte Count
4	
5	Validation Sequence Number
6	Validation Result
7	Validation Result Determined by DMS Datafill
8	RAO Digit Count
9	
	RAO Code

9.12 Branding Indicator DB

Identifier: #13E

Description: The data block specifies the branding status of the call. This refers to front end branding only, not branding provided after a call is floated.

Size: 6 bytes

Data Fields: **Identifier:**

#13E

DB Byte Count:

0 to 65535

Branding Status:

0 = Call has not received front-end branding

1 = Call has received front-end branding

- Comments:**
1. The omission of the branding indicator DB is to be interpreted as if the DB had been sent with a branding status of 0.
 2. If the branding status is improperly set (or the DB is not sent), subsequent operator screen displays indicating the call's branding status may be erroneous.

Byte #	Meaning
0	ID = #13E
1	
2	DB Byte Count
3	
4	Branding Status
5	Padding Field

9.13 Broadcast DB

Identifier: #153

Description: The data block provides broadcast information for the voice connection between the switch and the service node.

Size: 8 bytes

Data Fields:

Identifier:	#153
DB Byte Count:	0 to 65535
Node ID:	0 to 65535
Broadcast Cycle Count:	0 to 255

- Comments:**
1. The switch associates voice links with a particular service node. The Node ID identifies the owner of the voice link resource. This allows a service node to use the voice links of another service node.
 2. For broadcast voice links, specifying a non-zero number for the Broadcast Cycle Count field transfers responsibility for the release of the voice link to the switch. The switch will monitor for the correct number of cycles to be played and then release the voice link immediately. A Voice Release Inform message is then sent to the service node. Note that the service node can request the release of the voice link at any time prior to the release by the switch.
 3. Cycle counting is only applicable to voice links datafilled in switch table OAVLMAP with a value of HKCHG in field CUTTHRU.
 4. Once started, cycle counting continues as long as the voice link is connected to at least one party in the call (either calling or called). If both parties drop off, cycle counting stops. If another party is connected on the call and the voice link is still connected, counting continues from the last number; the number of cycles is not reset to 0. The only way to reset the cycle count to 0 is for the service node to request the release of the voice link, followed by a request to reconnect to it.

Byte #	Meaning
0	ID = #153
1	
2	DB Byte Count
3	
4	Node ID
5	
6	Broadcast Cycle Count
7	Padding Field

9.14 Broadcast Count DB

Identifier: #155

Description: Specifies the number of current connections to a particular broadcast voice channel.

Size: 6 bytes

Data Fields: **Identifier:**

#155

DB Byte Count:

0 to 65535

Current Broadcast Connections:

0 to 1023

Comments: 1. When this data block comes from the switch, the Current Broadcast Connections field reflects the value at the time the message is composed. This number may be different by the time the service node receives the message, depending on whether another call connects or releases the same voice link.

Byte #	Meaning
0	ID = #155
1	
2	DB Byte Count
3	
4	Current Broadcast Connections
5	

9.15 CCV Query Result DB

Identifier: #12D

Description: This data block indicates the results of a CCV LIDB query.

Size: 6 bytes

Data Fields: **Identifier:**

#12D

DB Byte Count:

0 to 65535

CCV Query Response:

0 = Miscellaneous
1 = PIN Mismatch
2 = PIN Unrestricted
3 = PIN Restricted
4 = PSDI of 2 - Threshold Exceeded
5 = PSDI of 3 - Non-Payment
6 = PSDI of 4 - Service Restriction
7 = CSDI of 3 - Service Denial

Comments: 1. None.

Byte #	Meaning
0	ID = #12D
1	
2	DB Byte Count
3	
4	CCV Query Response
5	Padding Field

9.16 Call Float Indicator DB

Identifier: #128

Description: This data block indicates whether a Float Call Request should connect the B Party upon float.

Size: 6 bytes

Data Fields: **Identifier:**

#128

DB Byte Count:

0 to 65535

Float Connection Indicator:

0 = Unconnected or Connected

1 = Talking

Comments: 1. The service node should set the Float Connection Indicator according to its interpretation of the status of the call. This will ensure that the switch properly interprets the float request.

- If value 0 (Unconnected or Connected) is selected, then, if necessary (i.e. if the connection is not already made), the B Party will be connected upon float (the A Party must be connected and off-hook, the B Party DN must be entered, and the B Party must not have answered).
- If value 1 (Talking) is selected, the B party should already be connected. The switch will not connect the B party upon float (all parties must be connected and off-hook).
- The value for the float connection indicator is not applicable if floating to a treatment.

Byte #	Meaning
0	ID = #128
1	
2	DB Byte Count
3	
4	Float Connection Indicator
5	Padding Field

9.17 Call In Queue DB

Identifier: #106

Description: This data block indicates whether the call should be taken down if all parties go on hook while the call is in queue for a function provider, if such queueing subsequently occurs.

Size: 6 bytes

Data Fields: **Identifier:**

#106

DB Byte Count:

0 to 65535

Call In Queue Take Down:

0 = Do Not Take Call Down

1 = Take Call Down

Comments: 1. None.

Byte #	Meaning
0	ID = #106
1	
2	DB Byte Count
3	
4	Call In Queue Take Down
5	Padding Field

9.18 Call Information DB

Identifier: #107

Description: This data block contains information on the call.

Size: 20 bytes

Data Fields: **Identifier:**

#107

DB Byte Count:

0 to 65535

Session Origination Type:

0 = Initial Call Setup

1 = Transfer

2 = Recall/Trigger

3 = Reorigination

Service Node Mode:

0 = Passive

1 = Active

2 = Node currently not connected

Control List ID:

0 to 4095

Trigger Profile Index:

0 to 2047

Call Origination Type:

0 = Unspecified

1 = Zero Minus

2 = Zero Plus

3 = One Plus

4 = Delay

5 = 555

6 = Toll Station

7 = Toll Subscriber

8 = 211

9 = 311

10 = 411

11 = 511

12 = 611

13 = 711

14 = 811

15 = 911

16 = 999

17 = HOM555

18 = FOR555
19 = INTS (Service Interception)
20 = Alarm
21 = Mobile
22 = Attended Pay Station
23 = CAMA
24 = 121
25 = 131
26 = 141
27 = 151
28 = 161
29 = 171
30 = 181
31 = 191
32 = 1150
33 = 1151
34 = 1152
35 = 1153
36 = 1154
37 = 1155
38 = 1156
39 = 1157
40 = 1158
41 = 1159
42 = 1160
43 = 1161
44 = 1162
45 = SPARE1
46 = SPARE2
47 = SPARE3
48 = SPARE4
49 = SPARE5
50 = Country Direct

Call In Queue Take Down:

0 = Do Not Take Call Down
1 = Take Call Down

DTMF Receiver:

0 = Not Connected
1 = Connected

Service Type:

0 = Toll and Assistance
1 = Directory Assistance

Session Type:

0 = Standard OSSAIN session

1 = Preprocessing OSSAIN session

Conversation Timing Status:

0 = Stop Call Timing

1 = Start Call Timing

Branding Status:

0 = Call has not received front-end branding

1 = Call has received front-end branding

Notification Timer Status:

0 = Notification timer running

1 = Notification timer not running

SPID Processing^{SV4}:

Bit 0 = A Party DN

Bit 1 = B Party DN

Bit 2 = Alt DN

Number of Operator to SN Handoffs:

0 to 10

254 = Nil

- Comments:**
1. If this data block is sent to the service node after a session is established by the service node (i.e. a session initiated by the service node), then the Session Origination Type field is set to Initial Call Setup.
 2. The Control List ID field indicates which control list is currently being used for the call. If there is no control list associated with the call, the value is nil (i.e. control list id is 4095). The service node can change the control list being used for the call via the Transfer to Control List Request operation.
 3. The Trigger Profile Index field indicates the current trigger profile being used for the call. The service node may change the index for call via the Trigger Profile Index data block which can be sent in the Call Float Request operation and the Transfer To Control List operation.
 4. The Call In Queue Take Down field indicates the status of this field when the call is sent to a service node. This informs the service node what will happen to the call if it releases the call and the call is subsequently queued for a service node. This field is referenced if all parties go on-hook. The service node can change the value of this field when it releases the call.
 5. The Service Type field indicates the base service applied to the call. Directory assistance calls are limited in functionality in that the service node can only dispose of the call via a Transfer to Control List or an End Call request.
 6. Two notes about the Branding Status field:

- The branding status is set to 1 if branding is provided internally by the TOPS switch. The branding status may also be 1 if the SN has previously sent the Branding Indicator DB with a branding status of 1.
 - At the point in a DA call that the calling party elects to have TOPS automatically complete a call to the requested number (Automated Directory Assistance Call Completion), the switch resets the branding status to 0.
7. Call Origination Types 20-44 are only valid for OSSAIN preprocessing calls.
8. The Notification Timer Status field indicates whether or not a "persistent-style" notification timer is running.

Byte #	Meaning
0	ID = #107
1	
2	DB Byte Count
3	
4	Session Origination Type
5	Service Node Mode
6	Control List ID
7	
8	Trigger Profile Index
9	
10	Call Origination Type
11	Call In Queue Take Down
12	DTMF Receiver
13	Service Type
14	Session Type
15	Conversation Timing Status
16	Branding Status
17	Notification Timer Status
18	SPID Processing
19	Number of Operator to SN Handoffs

9.19 Call Queue Information DB

Identifier: #13A

Description: This data block contains information on the status of a call queue in the switch.

Size: 14 bytes

Data Fields: **Identifier:**

#13A

DB Byte Count:

0 to 65535

Queuing System:

0 = OSSAIN service nodes

1 = TOPS operators

Call Queue:

0 to 8190

Calls In Queue:

0 to 32767

Predicted Wait Time (in tenths of seconds):

0 to 32767

Serving Agent Status:

0 = No agents are defined to serve this call queue

1 = Agents are defined to serve this call queue

Function Provider Request Status:

0 = Function provider connected to call

1 = Call queued for function provider

2 = Call deflected (i.e. Predicted Wait Time too high)

3 = Call queue overflow (i.e. Too many calls in this call queue)

4 = Queuing System overflow (i.e. Too many calls queued in the switch for all call queues in the given Queuing System combined)

- Comments:**
1. The Queuing System field indicates whether the function provider request resulted in a request for a session with a service node (Queuing System = 0) or a request for a TOPS operator (Queuing System = 1).
 2. The Call Queue field indicates the call queue that was used for the function provider request. Call Queue numbers are only unique within a particular Queuing System (e.g. Call Queue 1 in the OSSAIN queuing system is a different queue than Call Queue 1 in the TOPS operator queuing system). The combination of Queuing System and Call Queue uniquely identify a call queue in the switch.

3. The Serving Agent Status field indicates whether any queuing agents have been made known to the queuing system in the switch.

If the Queuing System field is 0 (OSSAIN service nodes), a Serving Agent Status of 0 (No agents are defined to serve this call queue) means that all service nodes that serve that call queue are out of service. If at least one service node that serves that call queue is in service, the Serving Agent Status field will have a value of 1 (Agents are defined to serve this call queue).

If the Queuing System field is 1 (TOPS operators), a Serving Agent Status of 0 (No agents are defined to serve this call queue) means that there are no operators logged in that serve that call queue. If at least one operator is logged in that serves that call queue, the Serving Agent Status field will have a value of 1 (Agents are defined to serve this call queue).

Byte #	Meaning
0	ID = #13A
1	
2	DB Byte Count
3	
4	Queuing System
5	Padding Field
6	Call Queue
7	
8	Calls In Queue
9	
10	Predicted Wait Time
11	
12	Serving Agent Status
13	Function Provider Request Status

9.20 Call Timing DB

Identifier: #13F

Description: Indicates a call's answer and conversation timings from the switch's perspective.

Size: 18 bytes

Data Fields:

Identifier:	#13F
DB Byte Count:	0 to 65535
Answer Date - Year:	0 to 9
Answer Date - Month:	1 to 12
Answer Date - Day:	1 to 31
Answer Time - Hour:	0 to 23
Answer Time - Minutes:	0 to 59
Answer Time - Seconds:	0 to 59
Answer Time - Tenths Seconds	0 to 9
Conversation Time - Elapsed Minutes	0 to 99,999
Conversation Time - Seconds:	0 to 59
Conversation Time - Tenths Seconds	0 to 9

- Comments:**
1. Conversation Time is zero for calls where answer has just occurred.
 2. The range for the Answer-Year field is 0 to 9 for this data block. The Year value represents the last digit of the Year (e.g. Answer-Year = "6" for 1996).

Byte #	Meaning
0	ID = #13F
1	
2	DB Byte Count
3	
4	Answer Date - Year
5	Answer Date - Month
6	Answer Date - Day
7	Answer Time - Hours
8	Answer Time - Minutes
9	Answer Time - Seconds
10	Answer Time - Tenths Seconds
11	Padding Field
12	Conversation Time - Elapsed Minutes
13	
14	
15	Conversation Time - Seconds
16	Conversation Time - Tenths Seconds
17	Padding Field

9.21 Call Update DB

Identifier: #165

Description: This data block is sent by the service node to update information pertaining to an OSSAIN call.

Size: 6 bytes

Data Fields: **Identifier:**

#165

DB Byte Count:

0 to 65535

Number of Operator to SN Handoffs:

0 to 10

254 = Nil

Comments: 1. The number of operator to SN handoffs represents the maximum number of handoffs allowed for this call.

Byte #	Meaning
0	ID = #
1	
2	DB Byte Count
3	
4	Number of Operator to SN Handoffs
5	Padding Field

9.22 Callid DB

Identifier: #149

Description: Identifies a particular call.

Size: 8 bytes

Identifier:

#149

DB Byte Count:

0 to 65535

Call ID:

0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Comments: 1. None.

Byte #	Meaning
0	ID = #149
1	
2	DB Byte Count
3	
4	Call ID
5	
6	
7	

9.23 Carrier DB

Identifier: #109

Description: Information on the active carrier for the call.

Size: 20 bytes

Data Fields: **Identifier:**

#109

DB Byte Count:

0 to 65535

Carrier Assignment Source Indicator:

0 = Unknown

1 = Switch table TRKGRP or TOPEATRK

2 = Trunk signalling

3 = OLNS

4 = Switch table DNPIC

5 = TOPS operator

6 = OSSAIN service node

7 = Subscriber dialled

Carrier Identification Digit Count:

0 to 4

Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

Service Type Provided by Switch:

0 = Transfer only

1 = Complete operator services

Carrier Number Status:

0 = None

1 = Valid

2 = Invalid

3 = Invalid for this call only

Carrier Service Status:

0 = Valid

1 = Invalid

Carrier Name Length:

0 to 8

Carrier Name:

Up to 8 ASCII Characters

Comments: 1. None.

Byte #	Meaning
0	ID = #109
1	
2	DB Byte Count
3	
4	Carrier Assignment Source Indicator
5	Carrier Identification Digit Count
6	Carrier Identification Code
7	
8	Service Type Provided by Switch
9	Carrier Number Status
10	Carrier Service Status
11	Carrier Name Length
12	Carrier Name
13	
14	
15	
16	
17	
18	
19	

9.24 Carrier Number DB

Identifier: #135

Description: Specifies the carrier ID to be used for the call.

Size: 8 bytes

Data Fields: **Identifier:**

#135

DB Byte Count:

0 to 65535

Carrier Assignment Source Indicator:

0 = Unknown

1 = Switch table TRKGRP or TOPEATTRK

2 = Trunk Signalling

3 = OLNS

4 = Switch table DNPIC

5 = TOPS operator

6 = OSSAIN service node

7 = Subscriber dialled

Carrier Identification Digit Count:

0 to 4

Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

Comments: 1. When the service node sends this data block to the switch, it sets the Carrier Assignment Source Indicator field to indicate how the carrier number was obtained. This value gets put in the AMA record (module code 53 and module code 54).

Byte #	Meaning
0	ID = #135
1	
2	
3	DB Byte Count
4	
5	Carrier Assignment Source Indicator
6	Carrier Identification Digit Count
7	Carrier Identification Code

9.25 Charge Status DB

Identifier: #10A

Description: This data block contains information on the billing status of the call.

Size: 6 bytes

Data Fields: **Identifier:**

#10A

DB Byte Count:

0 to 65535

Billable Status:

0 = Not Billable

1 = Billable

Comments: 1. None.

Byte #	Meaning
0	ID = #10A
1	
2	DB Byte Count
3	
4	Billable Status
5	Padding Field

9.26 Class Charge DB

Identifier: #10B

Description: Status on the class charge and handling data for the call. The service node can use this data block to request the switch to validate the billing number.

Size: 10 bytes

Data Fields: **Identifier:**

#10B

DB Byte Count:

0 to 65535

Handling Data:

0 = Station

1 = Person

2 = Auto

254 = Unspecified

Billing Data:

0 = Paid

1 = Collect

2 = Special Calling

3 = Special Called

254 = Unspecified

Validation Request:

0 = Not Applicable

1 = Do not validate

2 = Validate

Validation Sequence Number:

0 to 254

Class Charge Selector:

0 = Current class charge

1 = Previous class charge

- Comments:**
1. If this data block is sent from the switch to the service node, it contains the billing status of the call. If the data block is sent from the service node to the switch, it contains what the values for the field should be.
 2. The Validation Request is set to Not Applicable when the Class Charge data block is sent from the switch to the service node. In this case, the validation sequence number is not meaningful.
 3. When the Class Charge data block is sent from the service node to the switch, the value of the Validation Request field can be set to either Do Not Validate (if the service node does not want the switch to send a vali-

dation query for the billing number) or Validate (if the service node wants the switch to send a validation query for the billing number).

4. The Validation Sequence Number should be set to 1 by the service node for the first time at each new session when validation is requested by the service node.
5. The Validation Request field can be set to the value 'Do not validate' only when the billing data is 'Special Calling' or 'Special Called'.
6. The class charge can be set to unspecified by sending the Class Charge DB with a Handling Data and Billing Data of Unspecified and Validation Request set to not applicable.
7. This data block is sent by the switch with Class Charge Selector field set to 'Previous' if Session Origination type is 'Reorigination'.
8. If this data block is sent by the service node to the switch, the Class Charge Selector field must be set to 'Current'.

Byte #	Meaning
0	ID = #10B
1	
2	DB Byte Count
3	
4	Handling Data
5	Billing Data
6	Validation Request
7	Validation Sequence Number
8	Class Charge Selector
9	Padding Field

The following table lists the valid combinations of the Handling Data field and the Billing Data field.

Table 10: Valid Class Charges

Handling Data	Billing Data
Unspecified	Unspecified
Station	Paid
Station	Collect
Station	Special Calling

Table 10: Valid Class Charges

Handling Data	Billing Data
Station	Special Called
Person	Paid
Person	Collect
Person	Special Calling
Person	Special Called
Auto	Collect

9.27 Conference ID DB

Identifier: #14A

Description: Identifies a particular conference.

Size: 8 bytes

Identifier:

#14A

DB Byte Count:

0 to 65535

Conference ID:

0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Comments: 1. None.

Byte #	Meaning
0	ID = #14A
1	
2	DB Byte Count
3	
4	Conference ID
5	
6	
7	

9.28 Conference Party Connection DB

Identifier: #14B

Description: Specifies a conference party connection for a party in a call.

Size: 10 bytes

Identifier:

#14B

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Conference ID:

0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Comments: 1. None.

Byte #	Meaning
0	ID = #14B
1	
2	DB Byte Count
3	
4	Connection ID
5	Padding Field
6	Conference ID
7	
8	
9	

9.29 Conference Party Connection Information DB

Identifier: #14C

Description: Provides information on a conference party connected to a conference.

Size: 12 bytes

Identifier:

#14C

DB Byte Count:

0 to 65535

Call Id:

0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Connection Status:

0 = Not Connected

1 = Connected, Off Hook

2 = Connected, On Hook

3 = Connection is treatment

Speech Path Status

0 = Zero-Way

1 = Two-Way

2 = One-Way (Listen Only)

3 = One-Way (Speak Only)

Comments: 1. The call id in this data block indicates the call that contains the party being described.

Byte #	Meaning
0	ID = #14C
1	
2	DB Byte Count
3	
4	Call ID
5	
6	
7	
8	Connection ID
9	Connection Status
10	Speech Path Status
11	Padding Field

9.30 Conference SN Voice Connection DB

Identifier: #14D

Description: Specifies the status of the SN voice link connection in the call.

Size: 6 bytes

Identifier:

#14D

DB Byte Count:

0 to 65535

SN Voice Conferencing Status:

0 = SN voice link in call

1 = SN voice link in conference

Comments: 1. If this DB is sent in an SN request, then the SN is telling the switch what to do with the SN voice link. If this DB is sent from the switch, it indicates what the SN voice link is connected to.

Byte #	Meaning
0	ID = #14D
1	
2	DB Byte Count
3	
4	SN Voice Conferencing Status
5	Padding Field

9.31 Conference SN Voice Connection Information DB

Identifier: #14E

Description: Provides information on the SN voice link connected to a conference.

Size: 8 bytes

Identifier:

#14E

DB Byte Count:

0 to 65535

Call Id:

0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Comments: 1. None.

Byte #	Meaning
0	ID = #14E
1	
2	DB Byte Count
3	
4	Call Id
5	
6	
7	

9.32 Connection Event DB

Identifier: #10C

Description: Identifies a connection status change.

Size: 6 bytes

Identifier:

#10C

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Connection Event:

0 = On-hook

1 = Off-hook

2 = Hook flash

3 = Released

Comments: 1. A 'Released' can occur on ISUP trunks where the trunk is released when all parties have gone on-hook.

Byte #	Meaning
0	ID = #10C
1	
2	DB Byte Count
3	
4	Connection ID
5	Connection Event

9.33 Connection ID DB

Identifier: #10D

Description: Identifies the connection associated with the message.

Size: 6 bytes

Data Fields:

Identifier:

#10D

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Comments: 1. None.

Byte #	Meaning
0	ID = #10D
1	
2	DB Byte Count
3	
4	Connection ID
5	Padding Field

9.34 Connection Progress DB

Identifier: #10E

Description: Provides SS7 connection status information for ISUP trunk status. The service node may use the indicators to determine call waiting status of a call, ISUP continuity, etc.

Size: 8 bytes

Data Fields:

Identifier:

#10E

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

SS7 Backward Call Indicators:

A two byte SS7 set field containing SS7 information such as:

Charge Indicator, Called Party's Status Indicator, Called Party's Category Indicator, End-to-end Method Indicator, Interworking Indicator, IAM Segmentation Indicator, ISDN User Part Indicator, Holding Indicator, ISDN Access Indicator, Echo Control Device Indicator (See TR-NWT-000246 for details).

Comments: 1. None.

Byte #	Meaning
0	ID = #10E
1	
2	DB Byte Count
3	
4	Connection ID
5	Padding Field
6	SS7 Backward Call Indicators
7	

9.35 Context Block DB

Identifier: #110

Description: This data block contains miscellaneous information on the call that is sent to a function provider (i.e. a service node or an operator) on a transfer.

Size: 56 bytes

Data Fields: **Identifier:**

#110

DB Byte Count:

0 to 65535

Context Block Length:

0 to 50

Context Block Data:

up to 50 bytes

- Comments:**
1. The context data provided by the service node overwrites the existing context block data stored in the switch.
 2. The context block is not sent to a TOPS Automated System.
 3. The service nodes are responsible for the contents of the context block. The format of the context block data is unknown by the switch.
 4. The format of the context block data area is stored in a field within the context block data. The first field in the context block is the Context Type. The value of this field indicates the structure of the data area. Refer to the TOPS Context Block document for more information (NIS Q258-1).

Byte #	Meaning
0	ID = #110
1	
2	DB Byte Count
3	
4	Padding Field
5	Context Block Length
6	Context Block Data
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
...	
55	

9.36 Conversation Timing Indicator DB

Identifier: #140

Description: This data block contains the start/stop timing indication for an established call.

Size: 6 bytes

Data Fields: **Identifier:**

#140

DB Byte Count:

0 to 65535

Conversation Timing Status:

0 = Stop Call Timing

1 = Start Call Timing

Comments: 1. In general, voice connectivity between the parties in the call should be established prior to performing start timing.

Byte #	Meaning
0	ID = #140
1	
2	DB Byte Count
3	
4	Conversation Timing Status
5	Padding Field

9.37 Country Direct Information DB

Identifier: #15B

Description: This data block contains information on the country direct call.

Size: 14 bytes

Data Fields: **Identifier:**

#15B

DB Byte Count:

0 to 65535

Country Direct Access Code Number:

up to 4 BCD digits, ranging from 0 to 9

Country Direct Access Code Count:

0 to 4

Country Direct Country Code Count:

0 to 3

Country Direct Country Code Number:

up to 3 BCD digits, ranging from 0 to 9

Country Direct Carrier Code Number:

up to 4 BCD digits, ranging from 0 to 9

Country Direct Carrier Code Count:

0 to 4

Comments: 1. When a call is a country direct call (as identified by having this data block appear in any of the applicable operations) there are billing restrictions in place. Station paid and person paid billing are never allowed. There are other billing restrictions associated with the country in DMS datafill.

Byte #	Meaning
0	ID = #15B
1	
2	DB Byte Count
3	
4	Country Direct Access Code Number
5	
6	Country Direct Access Code Count
7	Country Direct Country Code Count
8	Country Direct Country Code Number
9	
10	Country Direct Carrier Code Number
11	
12	Country Direct Carrier Code Count
13	Padding Field

9.38 DTMF Digit DB

Identifier: #116

Description: This data block contains the digit received by the DTMF receiver monitoring the specified party.

Size: 6 bytes

Data Fields: **Identifier:**

#116

DB Byte Count:

0 to 65535

DTMF Digit:

0 - 9 = 0 to 9

10 = '*'

11 = '#'

Party:

1 = A Party

2 = B Party

Comments: 1. DTMF digits will only be reported for the A party at this time.

Byte #	Meaning
0	ID = #116
1	
2	DB Byte Count
3	
4	DTMF Digit
5	Party

9.39 Directory Number DB

Identifier: #111

Description: Directory Number information for a specified party. This data block is passed from the switch to service nodes to inform them of directory number related information.

Size: 34 bytes

Data Fields: **Identifier:**

#111

DB Byte Count:

0 to 65535

Party:

1 = A Party

2 = B Party

Directory Number Type:

0 = Foreign

1 = Domestic Non-local

2 = Domestic Local

3 = Outgoing Trunk

4 = Service

254 = Unknown

Language:

0 to 99

254 = Unknown

ANI Status:

0 = ANI Success

1 = Operator Number Identification (ONI)

2 = ANI Office Failure

3 = ANI Failure Detected

4 = Not Applicable

254 = Unknown

Connection Status:

0 = Not Connected

1 = Connected, Off Hook

2 = Connected, On Hook

3 = Connection is treatment

Speech Path Status:

0 = Zero-Way

1 = Two-Way

2 = One-Way (Listen Only)

3 = One-Way (Speak Only)

Presentation Indicator:

0 = Presentation Allowed

1 = Presentation Restricted

255 = Presentation Status Unknown

Directory Number Digit Count:

0 to 20

Directory Number:

BCD digits, ranging from 0 to 9

Unenforced Restrictions^{SV6}:

0 = No DN unenforced restrictions

1 = DN unenforced restrictions

Calling Line Identifier Digit Count:

0 to 20

Calling Line Identifier Number:

BCD digits, ranging from 0 to 9

- Comments:**
1. With SS7 signalling, two numbers may be sent for the A party. There may be a charge number and a calling line identifier number. If both numbers are received from SS7 signalling, the charge number will be in the Directory Number field and the calling line identifier number will be in the Calling Line Identifier Number Field. If only the charge number or the calling line identifier number is received from SS7 signalling, then the number will be in the Directory Number field. For other types of signalling (e.g. MF), the calling party directory number will be in the Directory Number field.
 2. If the party is not connected, then the Speech Path Status field does not apply.
 3. Directory numbers will not include long distance or international access digits. Country code digits will be included for foreign numbers.
 4. The Directory Number Type for the A party defaults to Domestic Local. This field indicates the type of B party DN the A party DN is calling. The B party directory number type is from the perspective of the A party calling the B party.
 5. The Unenforced Restrictions field only applies to B party Domestic numbers. Because the Directory Number DB may also apply to the A party, the default value of the unenforced restrictions field for the A party number (as well as for B party Foreign numbers) is always No DN unenforced restrictions. If the OSAN0006 SOC is ON, screening is enabled (i.e.

TOPSPARM CLD_DNSCRN_ENABLE is set to Y), and the B party number has a TOPS screening attribute (or any combination of attributes) datafilled against it in Table DNSCRN, then the unenforced restrictions field is set to DN unenforced restrictions. Otherwise, the unenforced restrictions field will be set to No DN unenforced restrictions. The unenforced restrictions field is informational only and has no effect on the Service Node's ability to float a call

6. The Presentation Indicator field does not apply to B-party numbers. For B-party numbers the field will be set to Presentation Status Unknown.
7. If a Calling Line Identifier Number exists, the Presentation Indicator applies to the number in the Calling Line Identifier Number field.

Byte #	Meaning
0	ID = #111
1	
2	DB Byte Count
3	
4	Party
5	Directory Number Type
6	Language
7	ANI Status
8	Connection Status
9	Speech Path Status
10	Presentation Indicator
11	Directory Number Digit Count
12	Directory Number
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	Unenforced Restrictions
23	Calling Line Identifier Digit Count
24	Calling Line Identifier Number
25	
26	
27	
28	
29	
30	
31	
32	
33	

9.40 Directory Number Information DB

Identifier: #156

Description: Directory Number type information for a specified party. This data block is passed from the switch to service nodes.

Size: 8 bytes

Data Fields: **Identifier:**

#156

DB Byte Count:

0 to 65535

Party:

1 = A Party

2 = B Party

Directory Number Type:

0 = Foreign

1 = Domestic Non-local

2 = Domestic Local

3 = Outgoing Trunk

4 = Service

254 = Unknown

Unenforced Restrictions^{SV6}:

0 = No DN unenforced restrictions

1 = DN unenforced restrictions

Comments: 1. The Unenforced Restrictions field only applies to B party Domestic numbers. Because the Directory Number Information DB may also apply to the A party, the value of the unenforced restrictions field for the A party number (as well as for B party Foreign numbers) is always No DN unenforced restrictions. If the OSAN0006 SOC is ON, screening is enabled (i.e. TOPSPARM CLD_DNSCRN_ENABLE is set to Y), and the B party number has a TOPS screening attribute (or any combination or attributes) datafilled against it in Table DNSCRN, then the unenforced restrictions field is set to DN unenforced restrictions. Otherwise, the unenforced restrictions field will be set to No DN unenforced restrictions. The unenforced restrictions field is informational only and has no effect on the Service Node's ability to float a call.

Byte #	Meaning
0	ID = #156
1	
2	DB Byte Count
3	
4	Party
5	Directory Number Type
6	Unenforced Restrictions
7	Padding

9.41 Directory Number Update DB

Identifier: #112

Description: Directory Number information to be updated for a specified party. This data block is sent from service nodes to the switch to request updating of directory number related information.

Size: 30 bytes

Data Fields:

Identifier:

#112

DB Byte Count:

0 to 65535

Party:

1 = A Party

2 = B Party

Language:

0 to 99

254 = Unknown

Directory Number Type:

0 = Foreign

1 = Domestic Non-local

2 = Domestic Local

3 = Outgoing Trunk

4 = Service

254 = Unknown

Directory Number Digit Count:

0 to 20

Directory Number:

BCD digits, ranging from 0 to 9

Presentation Indicator:

0 = Presentation Allowed

1 = Presentation Restricted

255 = Presentation Status Unknown

Calling Line Identifier Digit Count:

0 to 20

Calling Line Identifier Number:

BCD digits, ranging from 0 to 9

- Comments:**
1. With SS7 signalling, two numbers may be sent for the A party. There may be a charge number and a calling line identifier number. If both numbers are received, the charge number will be in the Directory Number field and the calling line identifier number will be in the Calling Line Identifier Number Field. If only the charge number or the calling line identifier number is received, then the number will be in the Directory Number field. For other types of signalling (e.g. MF), the A party directory number will be in the Directory Number field and nothing will be in the Calling Line Identifier Number. ****Note: CLI updates by the service node are not currently supported****
 2. If the directory number is for any party other than the A party, there will be no data in the Calling Line Identifier Number. ****Note: CLI updates by the service node are not currently supported****
 3. If this data block is sent with a Directory Number Digit Count of 0, then the directory number for the respective party is cleared.
 4. If this data block is sent with a Language of unknown, then the Language is cleared in the switch.
 5. If other information about the party needs to be updated (e.g. language), then the service node should resend all previously entered information (e.g. directory number) in this data block along with the new information (i.e. information in a Directory Number Update DB for a party will overwrite any information from prior Directory Number Update DBs for that party).
 6. To change any field other than the Language field the party must not be connected. To change the Language field while connected, the Directory Number Type, Digit Count, and Digits must be unchanged.
 7. For Directory Number Type, the only appropriate values for the Directory Number Update DB are Foreign, Domestic Non-local and Unknown. The A party should always be indicated as Domestic Non-local or Unknown.
 8. Directory Numbers for domestic calls under the North American numbering plan must be no greater than 10 digits (i.e. for Directory Number Type of Domestic, and switch data-filled as North American Dialing Plan, the Directory Number Digit Count, for A or B parties, should be between 0 and 10). Directory numbers for foreign calls under the North American numbering plan (B party only) have a maximum of 15 digits. Directory numbers for all calls and parties under the Open numbering plan have a maximum of 18 digits.
 9. Directory numbers should not include long distance or international access digits. Foreign directory numbers should be prepended with country code digits.

10. The Presentation Indicator field only applies to A-party numbers. For B-party updates the field is ignored.

Byte #	Meaning
0	ID = #112
1	
2	
3	DB Byte Count
4	
5	Party
6	Language
7	Directory Number Type
8	Directory Number Digit Count
9	
10	Directory Number
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	Presentation Indicator
21	
22	Calling Line Identifier Digit Count
23	
24	Calling Line Identifier Number
25	
26	
27	
28	
29	

9.42 End Call Handling DB

Identifier: #132

Description: This data block indicates the reasons why the service node is ending the call.

Size: 6 bytes

Data Fields: **Identifier:**

#132

DB Byte Count:

0 to 65535

Call Handling:

0 = Normal Call Handling - bill the call

1 = Normal Call Handling - cancel billing

2 = Error Recovery

- Comments:**
1. This data block is invoked with Call Handling = 0 when the call needs to be taken down as part of a normal call flow (e.g. the A party went on-hook) and the AMA record should record billable conversation time.
 2. This data block is invoked with Call Handling = 1 when the call needs to be taken down as part of a normal call flow and AMA record should not record any billable conversation time.
 3. This data block is invoked with Call Handling = 2 when the call is in an unrecoverable state with the service node (e.g. the service node could not allocate a resource or transfer to another node or an operator, and it had already acknowledged the Call Begin message).

Byte #	Meaning
0	ID = #132
1	
2	DB Byte Count
3	
4	Call Handling
5	Padding Field

9.43 Estimate of Charges Alternate Date DB

Identifier: #15F

Description: Specifies an alternate date for the switch to use for an estimate of charges calculation.

Size: 6 bytes

Data Fields: **Identifier:**

#15F

DB Byte Count:

0 to 65535

Month:

1 - January

2 - February

.....

11 - November

12 - December

Day:

1 to 31

Byte #	Meaning
0	ID = #15F
1	
2	DB Byte Count
3	
4	Month
5	Day

9.44 Estimate of Charges Alternate Time DB

Identifier: #160

Description: Specifies an alternate time for the switch to use for an estimate of charges calculation.

Size: 6 bytes

Data Fields: **Identifier:**

#160

DB Byte Count:

0 to 65535

Hour:

0 to 23

Minutes:

0 to 59

Byte #	Meaning
0	ID = #160
1	
2	DB Byte Count
3	
4	Hour
5	Minutes

9.45 Estimate of Charges Duration DB

Identifier: #161

Description: Specifies a duration value for the switch to use in an estimate of charges calculation.

Size: 6 bytes

Data Fields: **Identifier:**

#161

DB Byte Count:

0 to 65535

Estimate Duration:

1 to 65400

Byte #	Meaning
0	ID = #161
1	
2	DB Byte Count
3	
4	Estimate Duration
5	

- Comments:**
1. The Estimate Duration is in units of minutes.
 2. The Estimate Duration cannot exceed 999 minutes for a call using the North American Numbering Plan.

9.46 Estimate of Charges Duration Information DB

Identifier: #162

Description: The calculated results of an estimated charge request when a duration was specified by the SN.

Size: 8 bytes

Data Fields: **Identifier:**

#162

DB Byte Count:

0 to 65535

Estimated Charges:

0 to 4,294,967,295 (#0000 0000 to #FFFF FFFF)

Byte #	Meaning
0	ID = #162
1	
2	DB Byte Count
3	
4	Estimated Charges
5	
6	
7	

Comments: 1. Charges are in generic units. They are not specific to a currency.

9.47 Estimate of Charges General Information DB

Identifier: #163

Description: Date and Time specific results of an estimated charge calculation.

Size: 10 bytes

Data Fields: **Identifier:**

#163

DB Byte Count:

0 to 65535

Month:

1 - January

2 - February

.....

11 - November

12 - December

Day:

1 to 31

Year:

0 to 99

Hour:

0 to 23

Minutes:

0 to 59

Byte #	Meaning
0	ID = #163
1	
2	DB Byte Count
3	
4	Month
5	Day
6	Year
7	Hour
8	Minutes
9	Padding

9.48 Estimate of Charges No Duration Information DB

Identifier: #164

Description: The calculated results of an estimated charge request when a duration was not specified by the SN.

Size: 16 bytes

Data Fields: **Identifier:**

#164

DB Byte Count:

0 to 65535

Initial Period - Minutes:

0 to 59

Initial Period - Seconds:

0 to 59

Initial Period - Estimated Charges:

0 to 4,294,967,295

Subsequent Period - Minutes:

0 to 59

Subsequent Period - Seconds:

0 to 59

Subsequent Period - Estimated Charges:

0 to 4,294,967,295

Byte #	Meaning
0	ID = #164
1	
2	DB Byte Count
3	
4	Initial Period - Minutes
5	Initial Period - Seconds
6	Initial Period - Estimated Charges
7	
8	
9	
10	Subsequent Period - Minutes
11	Subsequent Period - Seconds
12	Subsequent Period - Estimated Charges
13	
14	
15	

- Comments:**
1. Charges are in generic units. They are not specific to a currency.
 2. The maximum Initial and Subsequent Period value used by the switch is 15 minutes and 0 seconds.

9.49 Function DB

Identifier: #114

Description: Identifies the function for the operation.

Size: 6 bytes

Data Fields: **Identifier:**

#114

DB Byte Count:

0 to 65535

Function ID:

0 to 1023

Comments: 1. If the operation is from the switch to the service node, the Function data block identifies the function the service node should provide for the call. If the operation is from the service node to the switch, the Function data block identifies the next function that should be provided for the call.

Byte #	Meaning
0	ID = #114
1	
2	DB Byte Count
3	
4	Function ID
5	

9.50 Generate AMA DB

Identifier: #115

Description: Specifies information needed when generating an AMA record.

Size: 6 bytes

Data Fields: **Identifier:**

#115

DB Byte Count:

0 to 65535

Generate AMA Sequence Number:

0 to 254

- Comments:**
1. The Generate AMA Sequence Number is intended as a unique identifier to avoid instances of lost or duplicate billing. This sequence number should be initialized to '1' at the start of every session and incremented with each Generate AMA DB. The switch expects the Generate AMA Sequence Number to go up with every request within a session. The switch will process Generate AMA DBs only if their sequence number is equal to (or greater than) the sequence number expected (i.e the sequence number is greater than the prior Generate AMA Sequence Number processed within the current session). Sequence numbers that are less than that expected will return an 'Invalid generate AMA sequence number' error.
 2. The Generate AMA DB will, if successfully processed, generate a billing record, including any AMA modules appended through the Append AMA Module Request, and clear the call's AMA buffer. If an Append AMA Module Request returns a buffer overflow error (the buffer has a max of 1600 hex digits total), the Generate AMA Record Request will allow the AMA buffer to be cleared so that the failed append can be resubmitted.

Byte #	Meaning
0	ID = #115
1	
2	DB Byte Count
3	
4	Generate AMA Sequence Number
5	Padding Field

9.51 ISUP Calling Party Number DB

Identifier: #151

Description: Specifies the number to be used as the ISUP Calling Party Number parameter in the outgoing ISUP IAM.

Size: 16 bytes

Data Fields: **Identifier:**

#151

DB Byte Count:

0 to 65535

Presentation Indicator:

0 = Presentation Allowed

1 = Presentation Restricted

255 = Presentation Status Unknown

Directory Number Digit Count:

0 to 20

Directory Number:

BCD digits, ranging from 0 to 9

- Comments:**
1. If the service node sends this data block with a Directory Number Digit Count of 0, then the number stored for the ISUP Calling Party Number parameter is cleared.
 2. The maximum number of digits supported for the ISUP Calling Party Number parameter is 18.

Byte #	Meaning
0	ID = #151
1	
2	DB Byte Count
3	
4	Presentation Indicator
5	Directory Number Digit Count
6	Directory Number
7	
8	
9	
10	
11	
12	
13	
14	
15	

9.52 Information ID DB

Identifier: #157

Description: This data block is sent at the beginning of a MIS message buffer. It indicates the MIS application ID as well as time of day information from the DMS.

Size: 14 bytes

Data Fields:

Identifier:	#157
DB Byte Count:	0 to 65535
Application ID:	0 = TOPS 1 = OSSAIN
Time - Hour:	0 to 23
Time - Minutes:	0 to 59
Time - Seconds:	0 to 59
Date - Year:	0 to 9
Date - Month:	1 to 12
Date - Day:	1 to 31
Restart Severity:	0 = No Restart 1 = Warm 2 = Cold 3 = Reload
Tenths of Seconds Timestamp:	0 to 65535

Comments: 1. None.

Byte #	Meaning
0	ID = #157
1	
2	DB Byte Count
3	
4	Application ID
5	Time - Hour
6	Time - Minutes
7	Time - Seconds
8	Date - Year
9	Date - Month
10	Date - Day
11	Restart Severity
12	Tenths of Seconds Timestamp
13	

9.53 Invalid Billing Details DB

Identifier: #12E

Description: This data block specifies which billing details are missing which cause billing to be unsatisfied.

Size: 6 bytes

Data Fields: **Identifier:**

#12E

DB Byte Count:

0 to 65535

Invalid Billing Set:

Bit 0 = Invalid or Missing Class Charge

Bit 1 = Invalid or Missing A Party DN

Bit 2 = Invalid or Missing B Party DN

Bit 3 = Invalid or Missing Billing Number

Comments: 1. None.

Byte #	Meaning
0	ID = #12E
1	
2	DB Byte Count
3	
4	Invalid Billing Set
5	

9.54 LATA Information DB

Identifier: #136

Description: This data block associates a lata status for the call.

Size: 6 bytes

Data Fields: **Identifier:**

#136

DB Byte Count:

0 to 65535

LATA Status:

0 = IntraLATA

1 = InterLATA

LATA Restrictions:

0 = No Restrictions

1 = Non-Conforming End Office

2 = All InterLATA is Restricted

Comments: 1. None.

Byte #	Meaning
0	ID = #136
1	
2	DB Byte Count
3	
4	LATA Status
5	LATA Restrictions

9.55 LNP Information DB

Identifier: #142

Description: Identifies the LRN and the portability status of the specified party.

Size: 20 bytes

Data Fields: **Identifier:**

#142

DB Byte Count:

0 to 65535

Portability Status:

0 = DN Not Ported

1 = DN Ported

254 = Unknown

Connection ID:

0 = Alt DN Connection

1 = A Party

2 = B Party

LRN Digit Count:

0 to 10

LRN:

BCD digits, ranging from 0 to 9

LRN Digit Count - CLI:

0 to 10

LRN - CLI:

BCD digits, ranging from 0 to 9

- Comments:**
1. If the specified party's DN is not ported, the LRN Digit Count field is set to 0.
 2. If the LRN is obtained through switch datafill instead of an LNP query, the Portability Status field is set to 254 (Unknown).
 3. If the LRN is obtained for the billing DN (third number or line-based calling card number), the Connection ID field is set to 0.
 4. With SS7 signalling, two numbers may be sent for the A party. There may be a charge number and a calling line identifier number. If both numbers are received from SS7 signalling, and if the LRN is available for either or both numbers, the LRN for the charge number will be in the LRN field and the LRN for the calling line identifier number will be in the LRN -

CLI field. If both LRNs are present, the LRN-CLI field determines the value of the portability status. If only the charge number or the calling line identifier number is received, then the LRN for this number will be in the LRN field.

Byte #	Meaning
0	ID = #142
1	
2	DB Byte Count
3	
4	Portability Status
5	Connection ID
6	Padding
7	LRN Digit Count
8	LRN
9	
10	
11	
12	
13	LRN Digit Count - CLI
14	LRN - CLI
15	
16	
17	
18	
19	Padding

9.56 LNP Enhancements DB

Identifier: #167

Description: This data block is used to modify standard LNP behavior of an OAP operation.

Size: 6 bytes

Data Fields: **Identifier:**

#167

DB Byte Count:

0 to 65535

LNP Query:

0 = Perform LNP query

1 = Skip LNP query

Comments: 1. None.

Byte #	Meaning
0	ID = #
1	
2	DB Byte Count
3	
4	LNP Query
5	Padding

9.57 LRN Update DB

Identifier: #166

Description: Contains Location Routing Number information for a specified party. This data block can be passed from the service node to the switch to update the LRN information.

Size: 12 bytes

Data Fields: **Identifier:**

#166

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party

2 = B Party

LRN Digit Count:

0 to 10

LRN:

BCD Digits, ranging from 0 to 9

Comments: 1. None.

Byte #	Meaning
0	ID = #
1	
2	DB Byte Count
3	
4	Connection ID
5	LRN Digit Count
6	LRN
7	
8	
9	
10	
11	Padding

9.58 Language DB

Identifier: #12F

Description: This data block associates a language to the specified party.

Size: 6 bytes

Data Fields: **Identifier:**

#12F

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Language:

0 to 99

254 = Unknown

Comments: 1. None.

Byte #	Meaning
0	ID = #12F
1	
2	DB Byte Count
3	
4	Connection ID
5	Language

9.59 Maintenance State DB

Identifier: #147

Description: This data block indicates the state of the node or session pool.

Size: 6 bytes

Data Fields: **Identifier:**

#147

DB Byte Count:

0 to 65535

Maintenance State:

0 = Out of Service

1 = In Service

Comments: 1. None.

Byte #	Meaning
0	ID = #147
1	
2	DB Byte Count
3	
4	Maintenance State
5	Padding Field

9.60 Maintenance Text DB

Identifier: #101

Description: This data block may accompany a maintenance message to the switch. This block is used so that the service node or session pool can send a message to be displayed at the MAP or logged at the switch.

Size: 38 bytes

Data Fields: **Identifier:**

#101

DB Byte Count:

0 to 65535

Maintenance Text Length:

0 to 32

Maintenance Text:

ASCII, 0 to 32 alpha-numeric characters (including space)

Comments: 1. None.

Byte #	Meaning
0	ID = #101
1	
2	DB Byte Count
3	
4	Padding Field
5	Maintenance Text Length
6	Maintenance Text
7	
8	
9	
10	
11	
...	
37	

9.61 NBEC DB

Identifier: #148

Description: Identifies the Non-Bell Exchange Carrier.

Size: 8 bytes

Data Fields: **Identifier:**

#148

DB Byte Count:

0 to 65535

NBEC Code Digit Count:

0 to 4

NBEC Code:

up to 4 BCD digits ranging from 0 to 9

Comments: 1. None.

Byte #	Meaning
0	ID = #148
1	
2	DB Byte Count
3	
4	Padding Field
5	NBEC Code Digit Count
6	NBEC Code
7	

9.62 Network Service DB

Identifier: #10F

Description: The network service ID for the current service.

Size: 6 bytes

Data Fields: **Identifier:**

#10F

DB Byte Count:

0 to 65535

Network Service ID:

0 to 32766

Comments: 1. None.

Byte #	Meaning
0	ID = #10F
1	
2	DB Byte Count
3	
4	Network Service ID
5	

9.63 Node Identification DB

Identifier: #117

Description: Identifies the OSSAIN Node.

Size: 8 bytes

Data Fields: **Identifier:**

#117

DB Byte Count:

0 to 65535

Node ID:

0 to 65535

Service Node Mode:

0 = Passive

1 = Active

2 = Node not currently connected

Comments: 1. None.

Byte #	Meaning
0	ID = #117
1	
2	DB Byte Count
3	
4	Node ID
5	
6	Service Node Mode
7	Padding Field

9.64 Notification Timer DB

Identifier: #14F

Description: Specifies the start/stop/reset information for the "persistent-style" notification timer.

Size: 10 bytes

Data Fields: **Identifier:**

#14F

DB Byte Count:

0 to 65535

Notification Time - Elapsed Minutes:

0 to 99, 999

Notification Time - Seconds:

0 to 59

Notification Time Type:

0 = Relative Time

1 = Conversation Time

Notification Timer Action:

0 = Start notification timer

1 = Stop notification timer

- Comments:**
1. If the Timer Action is set to Stop Notification Timer, the Notification Time fields are ignored by the switch.
 2. If the Timer Action is set to Start Notification Timer and a notification timer is already running for this call, the previous timer will be stopped and the timer will be restarted with the new timer values supplied in this data block.
 3. A Notification Time type of 'Relative' indicates that the timer should be started with the Notification Time fields supplied in this data block.
 4. A Notification Time type of 'Conversation' indicates that any elapsed conversation time should be subtracted from the Notification Time fields in this DB and the resulting time should be used to set the notification timer.
 5. Conversation timing will stop if conversation stops. Conversation timing will start if conversation resumes.

Byte #	Meaning
0	ID = #14F
1	
2	DB Byte Count
3	
4	Notification Time - Elapsed Minutes
5	
6	
7	Notification Time - Seconds
8	Notification Time Type
9	Notification Timer Action

9.65 OLNS Information DB

Identifier: #134

Description: Calling number information which was sent by the OLNS (Originating Line Number Screening) Database.

Size: 50 bytes

Data Fields: **Identifier:**

#134

DB Byte Count:

0 to 65535

OLNS TA Sent Paid Indicator:

0 = Screened

1 = Allowed

2 = Allowed for Domestic

3 = Allowed for IntraLATA Calls Due to Non-Payment

4 = Allowed for IntraLATA Calls at Customer Request

5 = Not Allowed

254 = Unknown

OLNS TA Collect Indicator:

0 = Screened

1 = Allowed

2 = Allowed for Domestic

3 = Not Allowed

254 = Unknown

OLNS TA Third Number Indicator:

0 = Screened

1 = Allowed

2 = Allowed for Domestic

3 = Not Allowed

4 = Allowed with Operator Verification

5 = Allowed with Operator or Automated Verification

254 = Unknown

OLNS TA Credit Card Indicator:

0 = Screened

1 = Allowed

2 = Allowed for Domestic

3 = Not Allowed

254 = Unknown

OLNS TA Special BNS Indicator:

0 = Screened

1 = Allowed

2 = Not Allowed

254 = Unknown

OLNS DACC Indicator:

0 = Screened

1 = Allowed

2 = Not Allowed

3 = Allowed with Billing Restrictions

4 = Allowed for Local, Non-Toll Calls

5 = Allowed with Billing Restrictions, No Sent Paid

254 = Unknown

OLNS DA Sent Paid Indicator:

0 = Screened

1 = Allowed from this line

2 = Not allowed from this line

254 = Unknown

OLNS DA Third Number Indicator:

0 = Screened

1 = Allowed from this line

2 = Not allowed from this line

3 = Allowed with operator verification

4 = Allowed with operator or automated verification

254 = Unknown

OLNS DA Credit Card Indicator:

0 = Screened

1 = Allowed from this line

2 = Not allowed from this line

254 = Unknown

OLNS DA Special BNS Indicator:

0 = Screened

1 = Allowed from this line

2 = Not allowed from this line

254 = Unknown

OLNS InterLATA Carrier Indicator:

0 = Screened

1 = IC Not Indicated

2 = IC Indicated

3 = IC Denied

254 = Unknown

OLNS InterLATA Carrier Identification Digit Count:

0 to 4

OLNS InterLATA Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

OLNS IntraLATA Toll Carrier Indicator:

- 0 = Screened
- 1 = IntraLATA Toll CIC Not Indicated
- 2 = IntraLATA Toll CIC Indicated
- 3 = IntraLATA Toll CIC Denied
- 4 = IntraLATA Toll CIC Not Indicated - ILP does not exist
- 5 = IntraLATA toll CIC Not Indicated - IntraLATA toll calls not allowed
- 254 = Unknown

OLNS IntraLATA Toll Carrier Identification Digit Count:

0 to 4

OLNS IntraLATA Toll Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

OLNS International Carrier Indicator:

- 0 = Screened
- 1 = INC Not Indicated
- 2 = INC Indicated
- 3 = INC Denied
- 254 = Unknown

OLNS International Carrier Identification Digit Count:

0 to 4

OLNS International Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

OLNS Foreign Language

- 0 = Screened
- 1 = Spanish
- 2 = French
- 3 = German
- 4 = Italian
- 5 = Mandarin
- 6 = Tagalog
- 7 = Polish
- 8 = Korean
- 9 = Vietnamese
- 10 = Portuguese
- 11 = Japanese
- 12 = Greek
- 13 = Arabic
- 14 = Hindi (Urdu)
- 15 = Russian

16 = Yiddish
17 = Thai (Laotian)
18 = Persian
19 = French Creole
20 = Armenian
21 = Navaho
22 = Hungarian
23 = Hebrew
24 = Dutch
25 = Mon-Khmer (Cambodian)
26 = Gujarathi
27 = Ukrainian
28 = Czech
29 = Pennsylvania Dutch
30 = Miao (Hmong)
31 = Norwegian
32 = Slovak
33 = Swedish
34 = Serbo-Croatian
35 = Kru
36 = Rumanian
37 = Lithuanian
38 = Finnish
39 = Panjabi
40 = Formosan
41 = Croatian
42 = Turkish
43 = Ilocano
44 = Bengali
45 = Danish
46 = Syriac
47 = Samoan
48 = Malayalam
49 = Cajun
50 = Amharic
51 = Cantonese
254 = Unknown

OLNS Alphanumeric String Length:

0 to 8

OLNS Alphanumeric String:

up to 8 characters of ASCII text

OLNS Service/Equipment Indicator:

0 = Screened

1 = POTS Line - Business/Residential

2 = LEC Public - Standard Interface - Postpay Overtime
 3 = POTS Line - Residential - Message Rate 1
 4 = POTS Line - Residential - Message Rate 2
 5 = 5 = LEC Semi-Public
 6 = POTS Line - Business - Flat Rate
 7 = POTS Line - Business - Message Rate 1
 8 = Coinless (non-IPP)
 9 = Coinless (IPP)
 10 = LEC Prepaid Telecommunications Card Station
 11 = POTS Line - Business - Message Rate 2
 12 = LEC Public Standard Interface - Prepay Overtime
 13 = LEC Public - Alternate Interface
 14 = IC Public - Standard Interface
 15 = IC Public - Alternate Interface
 16 = POTS Line - Residential - Flat Rate
 17 = Voice Quote - without tax - hotel
 18 = Voice Quote - with tax - hotel
 19 = IPP - Standard Interface
 20 = IPP - Alternate Interface
 21 = Hospital
 22 = Prison (non-IPP)
 23 = Auto Quote - without tax - hotel
 24 = Auto Quote - with tax - hotel
 25 = Dormitory Line
 26 = Centrex Line
 27 = PBX Line
 28 = Prison (IPP)
 29 = WATS Line
 30 = Cellular
 31 = Pager
 32 = Personal Communication Service (PCS)
 33 = FGA (Feature Group A)
 34 = Mobile
 35 = LEC Public - Special Billing - Post-Pay Overtime
 36 = LEC Public - Special Billing - Pre-Pay Overtime
 37 = Public - Incompatible Network Interface (No ACTS)
 38 = Cellular- Rate 1
 39 = Cellular- Rate 2
 40 = POTS Line- Business- Single-line
 41 = POTS Line- Business- Multi-line
 42 = Public- Post-pay
 254 = Unknown

OLNS Treatment Indicator:

0 = Screened
 1 = Automated Treatment- Provide Alerting Tone

- 2 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 1
- 3 = Operator Treatment- Operator Handling Due to Station Limitations
- 4 = Operator Treatment- Operator Handling at Customer Request
- 5 = Special Treatment- Handicapped
- 6 = Special Treatment- Deaf
- 7 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 2
- 8 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 3
- 9 = Operator Treatment- High Fraud
- 10 = Automated Treatment- Provide Alerting Tone - No Operator Access
- 11 = Automated Treatment- Provide Alerting Tone and Operator Announcement 4 - No Operator Access
- 12 = Automated Treatment- Provide Alerting Tone and Operator Announcement 5 - No Operator Access
- 13 = Automated Treatment- Provide Alerting Tone and Operator Announcement 6 - No Operator Access
- 14 = Automated Treatment- Provide Alerting Tone and Operator Announcement 7
- 15 = Automated Treatment- Provide Alerting Tone and Operator Announcement 8
- 16 = Automated Treatment- Provide Alerting Tone and Operator Announcement 9
- 17 = Automated Treatment- Provide Alerting Tone and Operator Announcement 10- No Operator Access
- 18 = Automated Treatment- Provide Alerting Tone and Operator Announcement 11- No Operator Access
- 19 = Automated Treatment- Provide Alerting Tone and Operator Announcement 12- No Operator Access
- 20 = Automated Treatment- Special Handling 1
- 21 = Automated Treatment- Special Handling 2
- 22 = Automated Treatment- Special Handling 3
- 23 = Automated Treatment- Special Handling 4
- 24 = Automated Treatment- Special Handling 5
- 25 = Automated Treatment- Provide Alerting Tone and Operator Announcement 13
- 26 = Automated Treatment- Provide Alerting Tone and Operator Announcement 14- No Operator Access
- 254 = Unknown

OLNS Account Owner Length^{SV3}:

0 to 4

OLNS Account Owner^{SV3}:

up to 4 characters of ASCII text

OLNS Billing Service Provider Length^{SV3}:

0 to 4

OLNS Billing Service Provider^{SV3}:

up to 4 characters of ASCII text

- Comments:**
1. Refer to the OLNS Protocol document for more information on OLNS values.
 2. This data block is a snapshot of the OLNS data returned from the OLNS LIDB. Corresponding values in other OAP fields may be different from that indicated in this data block. For example, the carrier IDs associated with the calling number in the OLNS database are sent in this data block. However, the carrier that is active for this call is returned in the Carrier DB.
 3. The following OLNS fields are used to determine corresponding OAP field values:
 - OLNS TA/DA billing indicators: Used to determine Restricted Billing Index.
 - OLNS Interlata Carrier Indicator: Used to determine Carrier Identification Code if it's an interlata call.
 - OLNS Service/Equipment Indicator: Used to determine Phone Class Type.
 - OLNS Account Owner: Used to determine Service Provider ID for account owner.
 - OLNS Billing Service Provider: Used to determine Service Provider ID for billing service provider.
 4. The OLNS International Carrier Indicator is currently always set to INC Not Indicated.
 5. The OLNS Interlata Carrier Indicator and OLNS Intralata Carrier Indicator fields are either set to Indicated or Not Indicated.

Byte #	Meaning
0	ID = #134
1	
2	
3	DB Byte Count
4	OLNS TA Sent Paid Indicator
5	OLNS TA Collect Indicator
6	OLNS TA Third Number Indicator
7	OLNS TA Credit Card Indicator
8	OLNS TA Special BNS Indicator
9	OLNS DACC Indicator
10	OLNS DA Sent Paid Indicator
11	OLNS DA Third Number Indicator
12	OLNS DA Credit Card Indicator
13	OLNS DA Special BNS Indicator
14	OLNS InterLATA Carrier Indicator
15	OLNS InterLATA Carrier ID Digit Count
16	OLNS InterLATA Carr ID Code
17	
18	OLNS IntraLATA Toll Carrier Indicator
19	OLNS IntraLATA Toll Carrier ID Dig Count
20	OLNS IntraLATA Toll Carrier ID Code
21	
22	OLNS International Carrier Indicator
23	OLNS International Carrier ID Digit Count
24	OLNS International Carrier ID Code
25	
26	OLNS Foreign Language
27	OLNS Alphanumeric String Length
28	OLNS Alphanumeric String
29	
30	
31	
32	
33	
34	
35	
36	OLNS Service/Equipment Indicator
37	OLNS Treatment Indicator

Byte #	Meaning
38	Padding Field
39	OLNS Account Owner Length
40	OLNS Account Owner
41	
42	
43	
44	Padding Field
45	OLNS Billing Service Provider Length
46	OLNS Billing Service Provider
47	
48	
49	

9.66 OSSAIN Call Queue Event DB

Identifier: #158

Description: This data block is sent by the DMS when a call event pertaining to a queue occurs.

Size: 44 bytes

Data Fields: **Identifier:**

#158

DB Byte Count:

0 to 65535

Queueing Event:

0 = Call in Queue

1 = Call Abandoned

2 = Call Presented to Agent

3 = Call Deflected - wait time to large

4 = Call Deflected - max queue size exceeded

5 = Call Queue Overflow

Service Type:

0 = Toll Assistance (TA)

1 = Directory Assistance (DA)

Node ID - Source:

0 to 65535

Trunk Group ID:

0 to 8191

Trunk Member ID:

0 to 10000

Call Type for Queueing:

0 to 2045

Call Queue:

0 to 8190

Time in Queue:

0 to 65535

Calls in Queue:

0 to 32767

Call ID:

0 to 4294967295

Tenths of Seconds Timestamp:

0 to 65535

Access Indicator:

0 = Operating Company

1 = Carrier

Service Provider ID Length:

4

Service Provider ID:

ASCII Characters

MIS Sequence Number:

0 to 255

Carrier Identification Digit Count:

0 to 4

Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

Session Pool ID:

0 to 65535

Session ID:

0 to 1023

Node ID - Service Node:

0 to 65535

Function ID:

0 to 1023

- Comments:**
1. The Source Node ID is the ID of the switch where the call originated.
 2. The Service Node ID is the ID of the service node to which the call was routed.

Byte #	Meaning
0	ID = #158
1	
2	DB Byte Count
3	
4	Queueing Event
5	Service Type
6	Source Node ID
7	
8	Trunk Group ID
9	
10	Trunk Member ID
11	
12	Call Type for Queueing
13	
14	Call Queue
15	
16	Time in Queue
17	
18	Calls In Queue
19	
20	Call ID
21	
22	
23	
24	Tenths of Seconds Timestamp
25	
26	Access Indicator
27	Service Provider ID Length
28	Service Provider ID
29	
30	
31	
32	MIS Sequence Number
33	Carrier Identification Digit Count
34	Carrier Identification Code
35	
36	Session Pool ID
37	
38	Session ID
39	
40	Service Node ID
41	
42	Function ID
43	

9.67 OSSAIN Non-Call Session Event DB

Identifier: #15A

Description: This data block is sent by the DMS when a non call event occurs that affects an OSSAIN session.

Size: 16 bytes

Data Fields: **Identifier**

#15A

DB Byte Count:

0 to 65535

Reset Time Flag:

0 = Do Not Reset Time

1 = Reset Time

Session State:

0 = Out of Service

1 = Not Available for CallP

2 = Available for CallP

Queue Peg Indicators:

Bit 0 = Peg Initial Session Seizures

Bit 1 = Peg Service Node Origination Seizures

Bit 2 = Peg Trigger Session Seizures

Bit 3 = Peg Transfer Session Seizures

Bit 4 = Peg Non Call Work Volume

Bit 5 = Peg Idle Time

Bit 6 = Peg Call Busy Work Volume

MIS Sequence Number:

0 to 255

Tenths of Seconds Timestamp:

0 to 65535

Session Pool ID:

0 to 65535

Session ID:

0 to 1023

Node ID - Service Node:

0 to 65535

Comments: 1. None.

Byte #	Meaning
0	ID = #15A
1	
2	DB Byte Count
3	
4	Reset Time Flag
5	Session State
6	Queue Peg Indicators
7	MIS Sequence Number
8	Tenths of Seconds Timestamp
9	
10	Session Pool ID
11	
12	Session ID
13	
14	Service Node ID
15	

9.68 OSSAIN Session Event DB

Identifier: #159

Description: This data block is sent by the DMS when call related event occurs that affects an OSSAIN session.

Size: 56 bytes

Data Fields: **Identifier:**

#159

DB Byte Count:

0 to 65535

Reset Time Flag:

0 = Do Not Reset Time

1 = Reset Time

Session State:

0 = Out of Service

1 = Not Available for CallP

2 = Available for CallP

Tenths of Seconds Timestamp:

0 to 65535

Call Queue:

0 to 8190

Queue Peg Indicators:

Bit 0 = Peg Initial Session Seizures

Bit 1 = Peg Service Node Origination Seizures

Bit 2 = Peg Trigger Session Seizures

Bit 3 = Peg Transfer Session Seizures

Bit 4 = Peg Non Call Work Volume

Bit 5 = Peg Idle Time

Bit 6 = Peg Call Busy Work Volume

Service Peg Indicators:

Bit 0 = Peg Service Initiation

Bit 1 = Peg Service Work Volume

Language:

0 to 99

254 = Unknown

Handling Data:

0 = Station

1 = Person

2 = Auto

254 = Unspecified

Billing Data:

0 = Paid

1 = Collect

2 = Special Calling

3 = Special Called

254 = Unspecified

Special Number Validity:

0 = Valid

1 = Verify

2 = Invalid

3 = Hotlisted

254 = Unspecified

Special Number Format:

0 = Domestic

1 = Overseas

2 = CCITT Calling Card

254 = Unspecified

Special Number Type:

0 = Third Number

1 = Special Number

2 = Calling Card

254 = Unspecified

Service Type:

0 = Toll Assistance (TA)

1 = Directory Assistance (DA)

Phone Class Type:

0 = Undetermined

1 = Station

2 = Prepay CCF

3 = Prepay CDF

4 = Postpay Coin

5 = Hotel/Motel

6 = Hotel/Motel With No Charge

7 = Attended Pay Station

8 = Marine

9 = Mobile

10 = Institution

Call ID:

0 to 4294967295

Directory Number Type - B Party:

0 = Foreign
1 = Domestic Non-local
2 = Domestic Local
3 = Outgoing Trunk
4 = Service
254 = Unknown

Call Origination Type:

0 = Unspecified
1 = Zero Minus
2 = Zero Plus
3 = One Plus
4 = Delay
5 = 555
6 = Toll Station
7 = Toll Subscriber
8 = 211
9 = 311
10 = 411
11 = 511
12 = 611
13 = 711
14 = 811
15 = 911
16 = 999
17 = HOM555
18 = FOR555
19 = INTS (Service Interception)
20 = Alarm
21 = Mobile
22 = Attended Pay Station
23 = CAMA
24 = 121
25 = 131
26 = 141
27 = 151
28 = 161
29 = 171
30 = 181
31 = 191
32 = 1150
33 = 1151
34 = 1152
35 = 1153
36 = 1154
37 = 1155

38 = 1156
39 = 1157
40 = 1158
41 = 1159
42 = 1160
43 = 1161
44 = 1162
45 = SPARE1
46 = SPARE2
47 = SPARE3
48 = SPARE4
49 = SPARE5
50 = Country Direct

Restricted Billing Index:

0 to 100

Carrier Identification Digit Count:

0 to 4

Carrier Identification Code:

up to 4 BCD digits ranging from 0 to 9

Trunk Group ID:

0 to 8191

Trunk Member ID:

0 to 10000

Node ID - Source Node:

0 to 65535

MIS Sequence Number:

0 to 255

Access Indicator:

0 = Operating Company
1 = Carrier

Service Provider ID Length:

4

Service Provider ID:

ASCII Characters

Call Type for Queueing:

0 to 2045

Network Service ID:

0 to 32766

Session Pool ID:

0 to 65535

Session ID:

0 to 1023

Node ID - Service Node:

0 to 65535

Function ID:

0 to 1023

- Comments:**
1. The Source Node ID is the ID of the switch where the call originated.
 2. The Service Node ID is the ID of the service node to which the call was routed.

Byte #	Meaning
0	ID = #159
1	
2	DB Byte Count
3	
4	Reset Time Flag
5	Session State
6	Tenths of Seconds Timestamp
7	
8	Call Queue
9	
10	Queue Peg Indicators
11	Service Peg Indicators
12	Language
13	Handling Data
14	Billing Data
15	Special Number Validity
16	Special Number Format
17	Special Number Type
18	Service Type
19	Phone Class Type
20	Call ID
21	
22	
23	
24	Directory Number Type
25	Call Origination Type
26	Restricted Billing Index
27	Carrier Identification Digit Count
28	Carrier Identification Code
29	
30	Trunk Group ID
31	
32	Trunk Member ID
33	
34	Source Node ID
35	
36	MIS Sequence Number
37	Padding Field
38	Access Indicator
39	Service Provider ID Length
40	Service Provider ID
41	
42	
43	
44	Call Type for Queueing
45	

Byte #	Meaning
46	Network Service ID
47	
48	Session Pool ID
49	
50	Session ID
51	
52	Service Node ID
53	
54	Function ID
55	

9.69 Operator Identification DB

Identifier: #118

Description: Identifies the operator.

Size: 8 bytes

Data Fields: **Identifier:**

#118

DB Byte Count:

0 to 65535

Operator ID:

0 to 9997

Terminal ID:

0 to 9999

Comments: 1. See the definition of each operation that contains this data block for specifics on how the operator identifier is used.

Byte #	Meaning
0	ID = #118
1	
2	DB Byte Count
3	
4	Operator ID
5	
6	Terminal ID
7	

9.70 Operator Services System Action DB

Identifier: #133

Description: This data block indicates what action is taken by the Operator Services System. This information is placed in AMA records.

Size: 6 bytes

Data Fields: **Identifier:**

#133

DB Byte Count:

0 to 65535

Operator Services System Action:

0 = Unknown

1 = Automatically Accepted

2 = Automatically Accepted from operator methods

3 = Verified Accepted

4 = Verified Rejected

5 = Provided Restricted Billing Treatment

6 = Transferred to IC Operator Services System

7 = Provided Intercept Service

8 = Inward Service

9 = Disconnected

Comments: 1. None.

Byte #	Meaning
0	ID = #133
1	
2	DB Byte Count
3	
4	Operator Services System Action
5	Padding Field

9.71 Operator System DB

Identifier: #141

Description: Indicates a call's eligibility for TOPS automated system processing.

Size: 6 bytes

Data Fields: **Identifier:**

#141

DB Byte Count:

0 to 65535

Operator System Selector:

0 = Call may only be handled by an operator

1 = Call may be handled by either an operator or TOPS automated system

Comments: 1. If operator-only is indicated in the Operator System Selector field, all TOPS automated systems will be bypassed and the call will be assigned to an operator.

Byte #	Meaning
0	ID = #141
1	
2	DB Byte Count
3	
4	Operator System Selector
5	Padding Field

9.72 Originating Station Information DB

Identifier: #119

Description: Service type information for the call.

Size: 22 bytes

Data Fields: **Identifier:**

#119

DB Byte Count:

0 to 65535

Phone Class Type:

0 = Undetermined
1 = Station
2 = Prepay CCF
3 = Prepay CDF
4 = Postpay Coin
5 = Hotel/Motel
6 = Hotel/Motel With No Charge
7 = Attendant Pay Station
8 = Marine
9 = Mobile
10 = Institution

Coin Overtime Type:

0 = Not Applicable
1 = Postpay Overtime
2 = Prepay Overtime

Coin Key Pad State:

0 = Undetermined
1 = Enabled
2 = Disabled

Hotel Room Number Length:

0 to 6

Hotel Room Number:

up to 6 ASCII characters

Authorization Number Length:

0 to 6

Authorization Number:

up to 6 ASCII characters

Comments: 1. The Hotel Room Number field may be used if the Phone Class Type is Hotel/Motel or Hotel/Motel With No Charge.

2. The Authorization Number field may be used if the Phone Class Type is Hotel/Motel, Hotel/Motel With No Charge, or Institution.

Byte #	Meaning
0	ID = #119
1	
2	
3	DB Byte Count
4	
5	Phone Class Type
6	Coin Overtime Type
7	Coin Key Pad State
8	Hotel Room Number Length
9	
10	
11	
12	
13	
14	Hotel Room Number
15	Padding Field
16	Authorization Number Length
17	Authorization Number
18	
19	
20	
21	

9.73 Originating Trunk Information DB

Identifier: #11A

Description: Identifies the trunk that the call originated on.

Size: 8 bytes

Data Fields:

Identifier:
#11A

DB Byte Count:
0 to 65535

Trunk Group ID:
0 to 8191

Trunk Member ID:
0 to 10000

Comments: 1. None.

Byte #	Meaning
0	ID = #11A
1	
2	DB Byte Count
3	
4	Trunk Group ID
5	
6	Trunk Member ID
7	

9.74 Pass Through DB

Identifier: #138

Description: This data block is used to pass unformatted data between active and passive function providers during a simultaneous interaction call.

Size: 106 bytes

Data Fields: **Identifier:**

#138

DB Byte Count:

0 to 65535

Pass Through Data Length:

0 to 100

Pass Through Data:

up to 100 bytes

Comments: 1. The function providers are responsible for the contents of the pass through data. The format of the pass through data is unknown by the switch.

Byte #	Meaning
0	ID = #138
1	
2	
3	DB Byte Count
4	
5	Padding Field
6	Pass Through Data Length
7	
8	Pass Through Data
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
...	
105	

9.75 Passive Node Status DB

Identifier: #137

Description: This data block is used to inform the active service node about the status of the passive function provider during a simultaneous interaction.

Size: 6 bytes

Data Fields: **Identifier:**

#137

DB Byte Count:

0 to 65535

Passive Node Status:

0 = Passive function provider connected

1 = Passive function provider released

2 = Passive function provider performing restricted operation

3 = Passive function provider completed restricted operation

4 = Passive operator being connected

- Comments:**
1. If the active service node receives this data block with Passive Node Status = 2 (Passive function provider performing restricted operation), the active service node should refrain from sending in any requests until the switch sends this data block back with Passive Node Status = 3 (Passive function provider completed restricted operation). If the active service node attempts to send in a request during the restricted window, it will receive an Error Response indicating which restricted operation is being performed by the passive function provider.
 2. If the active service node receives this data block with Passive Node Status = 4 (Passive operator being connected), the active service node should refrain from sending in any requests until the switch sends this data block back with Passive Node Status = 0 (Passive function provider connected). If the active service node attempts to send in a request during the operator connection window, it will receive an Error Response indicating that a passive operator connection is in progress.

Byte #	Meaning
0	ID = #137
1	
2	DB Byte Count
3	
4	Passive Node Status
5	Padding Field

9.76 Preprocessing Information DB

Identifier: #146

Description: This data block contains information about an OSSAIN preprocessing call. Included is the type of operator system that handled the call just prior to the preprocessing session and the system to which the call will resume to after preprocessing.

Size: 8 bytes

Data Fields: **Identifier:**

#146

DB Byte Count:

0 to 65535

Operator System Type (Pre-OSSAIN):

0 = Nil

1 = TOPS Operator

2 = AABS

3 = ACTS

4 = ADAS

5 = MCCA

Operator System Type (Post Resume):

0 = Nil

1 = TOPS Operator

2 = AABS

3 = ACTS

4 = ADAS

5 = MCCA

ADACC Transition Indicator:

0 = Automated Directory Assistance Call Completion Transition
False

1 = Automated Directory Assistance Call Completion Transition
True

- Comments:**
1. The ADACC Transition Indicator is used to distinguish between a new origination and a call that is transitioning from a DA call to a toll call for ADACC.
 2. Notes about the Pre-OSSAIN Operator System Type field:
 - A new call origination that has not been to any operator system will have a 0 (NIL) value. The ADACC Transition Indicator field will be False.

- A DA call that has transitioned to toll for ADACC and has not been to any operator system for toll billing service will have a 0 (NIL) value. In this case, the ADACC Transition Indicator field will be True.
 - A value of 1 (TOPS Operator) is not valid for the Pre-OSSAIN instance of this field.
 - Values 2 thru 5 are used for calls which were not successfully handled by a TOPS automated system. The value identifies the automated system that was handling the call prior to preprocessing.
3. A NIL value for the Post-Resume Operator System Type field is not possible.

Byte #	Meaning
0	ID = #146
1	
2	DB Byte Count
3	
4	Operator System Type (Pre-OSSAIN)
5	Operator System Type (Post Resume)
6	ADACC Transition Indicator
7	Padding Field

9.77 RLT Request DB

Identifier: #144

Description: Used to indicate whether RLT should be attempted for the call.

Size: 6 bytes

Data Fields: **Identifier:**

#144

DB Byte Count:

0 to 65535

Requested RLT Action:

0 = Do not attempt RLT

1 = Attempt RLT

- Comments:**
1. If this data block is not sent, the default is set to Do not attempt RLT.
 2. If the SN sends this data block with Do not attempt RLT, the RLT is turned off permanently for the call. RLT will not be able to be turned on again for the call by either an operator or an SN.

Byte #	Meaning
0	ID = #144
1	
2	DB Byte Count
3	
4	Requested RLT Action
5	Padding Field

9.78 Release Cause DB

Identifier: #11C

Description: Contains the SS7 Release Cause for a failed Connect DN Request or for a Connection Status Inform with an Event type of Released. This data block accompanies these messages for ISUP trunks only. The Release Cause DB indicates why the connection, indicated in the Connection ID, was released. In the switch a treatment is typically associated with the particular cause.

Size: 6 bytes

Data Fields: **Identifier:**

#11C

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

SS7 Release Cause:

0 to 127 (See TR-NWT-000246 for details.)

Comments: 1. None.

Byte #	Meaning
0	ID = #11C
1	
2	DB Byte Count
3	
4	Connection ID
5	SS7 Release Cause

9.79 Release Node DB

Identifier: #139

Description: This data block is used to indicate which function provider is to be released during a simultaneous interaction.

Size: 6 bytes

Data Fields: **Identifier:**

#139

DB Byte Count:

0 to 65535

Release Node Indicator:

0 = Self

1 = Other node

Comments: 1. None.

Byte #	Meaning
0	ID = #139
1	
2	DB Byte Count
3	
4	Release Node Indicator
5	Padding Field

9.80 Report Text DB

Identifier: #13C

Description: This data block stores one line of a log report text.

Size: 86 bytes

Data Fields: **Identifier:**

#13C

DB Byte Count:

0 to 65535

Report Text Length:

0 to 80

Report Text:

ASCII Text

Comments: 1. None.

Byte #	Meaning
0	ID = #13C
1	
2	DB Byte Count
3	
4	Padding Field
5	Report Text Length
6	Report Text
7	
8	
9	
10	
...	
...	
...	
85	

9.81 Requested Service DB

Identifier: #154

Description: This data block indicates the service type requested for the call.

Size: 6 bytes

Data Fields: **Identifier:**

#154

DB Byte Count:

0 to 65535

Service Type:

0 = Toll and Assistance

1 = Directory Assistance

Comments: 1. None.

Byte #	Meaning
0	ID = #154
1	
2	DB Byte Count
3	
4	Service Type
5	Padding Field

9.82 Routing Index DB

Identifier: #15C

Description: Identifies an index into switch table OAINRTE to be used to obtain an alternate route for connecting to the specified party. The resulting route overrides the route obtained by the switch when performing translations on the DN.

Size: 8 bytes

Data Fields: **Identifier:**

#15C

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Routing Index:

0 to 2046

Comments: 1.

Byte #	Meaning
0	ID = #15C
1	
2	DB Byte Count
3	
4	Connection ID
5	Padding Field
6	Routing Index
7	

9.83 Routing Number DB

Identifier: #15D

Description: Identifies an alternate routing number that is being used in place of the directory number associated with a particular party (A, B or Alt). The alternate routing number will be used by translations and screening to determine a different route for a call than would otherwise be taken.

Size: 16 bytes

Data Fields: **Identifier:**

#15D

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Directory Number Digit Count:

0 to 20

Directory Number:

BCD Digits, ranging from 0 to 9

Comments: 1.

Byte #	Meaning
0	ID = #15D
1	
2	DB Byte Count
3	
4	Connection ID
5	Directory Number Digit Count
6	Directory Number
7	
8	
9	
10	
11	
12	
13	
14	
15	

9.84 SN Voice Link DB

Identifier: #15E

Description: This datablock contains the voice link information.

Size: 12 bytes

Data Fields: **Identifier:**

#15E

DB Byte Count:

0 to 65535

SN Voice Link Status:

0 = Connected

1 = Released

Trunk Group ID:

0 to 8191

Trunk Member ID:

0 to 10000

Logical Voice Channel ID:

0 to 8192

- Comments:**
1. Logical Voice Channel ID 8192 is the NIL voice channel ID.
 2. For a switch-selected voice link the Logical Voice Channel ID will be the NIL voice channel ID.

Byte #	Meaning
0	ID = #15E
1	
2	DB Byte Count
3	
4	SN Voice Link Status
5	Padding Field
6	Trunk Group ID
7	
8	Trunk Member ID
9	
10	Logical Voice Channel ID
11	

9.85 SPID DB

Identifier: #145

Description: Contains a Service Provider Identifier (SPID) for the specified party and type of service provider.

Size: 12 bytes

Data Fields: **Identifier:**

#145

DB Byte Count:

0 to 65535

Service Provider Party Identifier:

- 1 = Originating party directory number data
- 2 = Terminating party directory number data
- 3 = Billing party directory number data

Type of Service Provider:

- 1 = Account owner
- 2 = Billing service provider

Service Provider ID Length:

4

Service Provider ID:

ASCII Characters

Comments: 1. SPIDs for the terminating party and billing party are currently not supported.

Byte #	Meaning
0	ID = #145
1	
2	
3	DB Byte Count
4	
5	Service Provider Party Identifier
6	Type of Service Provider
7	Padding Field
8	Service Provider ID Length
9	
10	
11	

9.86 Sequence Number Check DB

Identifier: #129

Description: Contains the sequence number of the last switch-originated message prior to a service node sending this data block. The switch uses this sequence number to determine if the service node has had an opportunity to process all messages (such as DTMF events) sent by the switch prior to sending this data block.

Size: 6 bytes

Data Fields:

Identifier:	#129
DB Byte Count:	0 to 65535
Sequence Number:	0 to 65535

Comments: 1. None.

Byte #	Meaning
0	ID = #129
1	
2	DB Byte Count
3	
4	Sequence Number
5	

9.87 Session Information DB

Identifier: #150

Description: Specifies a particular SN session.

Size: 8 bytes

Data Fields: **Identifier:**

#150

DB Byte Count:

0 to 65535

Session Pool ID:

0 to 65535

Session ID:

0 to 1023

Comments: 1. None.

Byte #	Meaning
0	ID = #150
1	
2	DB Byte Count
3	
4	Session Pool ID
5	
6	Session ID
7	

9.88 Speech Path DB

Identifier: #11B

Description: This data block contains the desired speech status of the party specified by the connection ID.

Size: 6 bytes

Data Fields: **Identifier:**

#11B

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Speech Path Status:

0 = Zero-Way

1 = Two-Way

2 = One-Way (Listen Only)

3 = One-Way (Speak Only)

Comments: 1. None.

Byte #	Meaning
0	ID = #11B
1	
2	DB Byte Count
3	
4	Connection ID
5	Speech Path Status

9.89 Terminating Trunk Information DB

Identifier: #152

Description: Identifies the trunk that the call terminated to for the B party or alternate DN.

Size: 8 bytes

Data Fields: **Identifier:**

#152

DB Byte Count:

0 to 65535

Trunk Group ID:

0 to 8191

Trunk Member ID:

0 to 10000

Comments: 1. None.

Byte #	Meaning
0	ID = #152
1	
2	DB Byte Count
3	
4	Trunk Group ID
5	
6	Trunk Member ID
7	

9.90 Text DB

Identifier: #11D

Description: A block of text that can be sent to a compatible operator position.

Size: 26 bytes

Data Fields: **Identifier:**

#11D

DB Byte Count:

0 to 65535

Text Block Length:

0 to 20

Text Block:

ASCII Text

Comments: 1. None.

Byte #	Meaning
0	ID = #11D
1	
2	DB Byte Count
3	
4	Padding Field
5	Text Block Length
6	Text Block
7	
8	
9	
...	
25	

9.91 Throttle DB

Identifier: #13D

Description: This data block indicates the number of available sessions for a session pool.

Size: 6 bytes

Data Fields: **Identifier:**

#13D

DB Byte Count:

0 to 65535

Available Sessions:

0 to 1023

Comments: 1. None

Byte #	Meaning
0	ID = #13D
1	
2	DB Byte Count
3	
4	Available Sessions
5	

9.92 Time DB

Identifier: #11E

Description: Indicates the current time from the switch's perspective. If the data block is included in a maintenance operation, the time is from the maintenance switch perspective. If the data block is included in a call processing operation, the time is from that call processing switch that sent the message (Standalone switch or OSAC Remote switch).

Size: 10 bytes

Data Fields:	Identifier:
	#11E
	DB Byte Count:
	0 to 65535
	Date - Year:
	0 to 99
	Date - Month:
	1 to 12
	Date - Day:
	1 to 31
	Time of Day - Hour:
	0 to 23
	Time of Day - Minutes:
	0 to 59
	Time of Day - Seconds:
	0 to 59

Comments: 1. None.

Byte #	Meaning
0	ID = #11E
1	
2	DB Byte Count
3	
4	Year
5	Month
6	Day
7	Hours
8	Minutes
9	Seconds

9.93 Transfer Control List DB

Identifier: #11F

Description: Contains the index of the control list that the current node is attempting to transfer to.

Size: 8 bytes

Data Fields: **Identifier:**

#11F

DB Byte Count:

0 to 65535

Control List ID:

0 to 4095

Call In Queue Take Down:

0 = Do Not Take Call Down

1 = Take Call Down

Comments: 1. None.

Byte #	Meaning
0	ID = #11F
1	
2	DB Byte Count
3	
4	Control List ID
5	
6	Call In Queue Take Down
7	Padding Field

9.94 Treatment DB

Identifier: #121

Description: Indicates the treatment type (per switch table TMTCNTL) the call is receiving. For ISUP trunks, the SS7 Release Cause value provides this index.

Size: 6 bytes

Data Fields: **Identifier:**

#121

DB Byte Count:

0 to 65535

Connection ID:

0 = Alt DN Connection

1 = A Party Connection

2 = B Party Connection

Treatment:

0 to 254

- Comments:**
1. When sent by the switch, the Connection ID field specifies which connection is receiving the treatment. When sent by the SN, specifies which party is to receive treatment
 2. The service node may stop treatment application through releasing the connection associated with the treatment (i.e. issuing a Release DN Request for the Connection ID indicated in the Treatment DB).

Byte #	Meaning
0	ID = #121
1	
2	DB Byte Count
3	
4	Connection ID
5	Treatment

9.95 Trigger DTMF DB

Identifier: #122

Description: This data block indicates information regarding the switch trigger DTMF event that initiated this session with the service node.

Size: 8 bytes

Data Block(s): **Identifier:**

#122

DB Byte Count:

0 to 65535

Trigger Detection Point:

0 = Connect

1 = Talking

2 = Disconnect

Party:

1 = A Party

2 = B Party

DTMF Digit:

0 - 9 = 0 to 9

10 = *

11 = #

- Comments:**
1. The DTMF Digit field specifies the digit that caused the trigger.
 2. In the talking state, only the '*' and the '#' trigger events are reported to the service node.

Byte #	Meaning
0	ID = #122
1	
2	DB Byte Count
3	
4	Trigger Detection Point
5	Party
6	DTMF Digit
7	Padding Field

9.96 Trigger Information DB

Identifier: #123

Description: This data block indicates information regarding the switch trigger event that initiated this session with the service node.

Size: 6 bytes

Data Block(s): **Identifier:**

#123

DB Byte Count:

0 to 65535

Trigger Detection Point:

0 = Connect

1 = Talking

2 = Disconnect

Trigger Event:

0 = Outpulsing Failure

1 = No Answer

2 = Flash

3 = Answer

4 = Notification

5 = Calling (A Party) Disconnect

6 = Party Disconnect

- Comments:**
1. This data block is used if the trigger event is not treatment, release cause, ringing, or DTMF.
 2. If the trigger event requires additional information, then a specific trigger event data block will be sent instead (e.g. Trigger Treatment DB, Trigger Release Cause DB, Trigger Ringing, Trigger DTMF DB).

Byte #	Meaning
0	ID = #123
1	
2	DB Byte Count
3	
4	Trigger Detection Point
5	Trigger Event

The following table lists the valid Trigger Events for each Trigger Detection Point.

Table 11: Trigger Events for each Trigger Detection Point

Connect	Talking	Disconnect
• Outpulsing Failure • Ringing Detected • Release Cause • Treatment • No Answer • DTMF Digit • Flash • Answer • Calling Disconnect	• DTMF Digit • Flash • Notification • Calling Disconnect • Party Disconnect	• DTMF Digit • Flash • Calling Disconnect

9.97 Trigger Profile Index DB

Identifier: #124

Description: This data block contains information regarding the trigger profile index to be applied to the call.

Size: 14 bytes

Data Fields: **Identifier:**

#124

DB Byte Count:

0 to 65535

No Answer Time:

0 to 120

Trigger Profile Index:

0 to 2047

Notification Time - Elapsed Minutes:

0 to 99, 999

Notification Time - Seconds:

0 to 59

Notification Time Type:

0 = Relative Time

1 = Conversation Time

Notification Timer Mode:

0 = Float-style notification timer

1 = Persistent-style notification timer

2 = Do not change timer

- Comments:**
1. If trigger processing is turned off (i.e. set to the nil trigger profile index, 2047), then the No Answer Time field and the Notification Time field is ignored by the switch.
 2. If the trigger profile does not require the No Answer Timer or the Notification Timer, then the No Answer Time field and/or the Notification Time field is ignored by the switch.
 3. When no notification time is specified (i.e. 0 minutes and 0 seconds) this indicates that the switch use the default value or the timer specified by the previous function provider.
 4. A Notification Time type of 'Relative' indicates the call triggers back to the SN following call float or answer, whichever occurs last, after the duration of the Notification Time specified.

5. A Notification Time type of 'Conversation' indicates the call triggers back to the SN following the call's elapsed conversation time equal to the notification time specified.
6. If the Notification Timer Mode is set to "float-style", any "persistent-style" notification timer that might have been running will be stopped and the "float-style" notification timer will be started with the values supplied in this data block.
7. If the Notification Timer Mode is set to "persistent-style", any "persistent-style" notification timer that might have been running will be stopped and the new "persistent-style" notification timer will be started with the values supplied in this data block.
8. If the Notification Timer Mode is set to "Do not change timer", then no timer information is changed. If timers were set previously, then those will be used when the call is floated.

Byte #	Meaning
0	ID = #124
1	
2	DB Byte Count
3	
4	Padding Field
5	No Answer Time
6	Trigger Profile Index
7	
8	Notification Time - Elapsed Minutes
9	
10	
11	Notification Time - Seconds
12	Notification Time Type
13	Notification Timer Mode

9.98 Trigger Release Cause DB

Identifier: #125

Description: This data block indicates information regarding the switch trigger release cause event that initiated this session with the service node.

Size: 8 bytes

Data Block(s): **Identifier:**

#125

DB Byte Count:

0 to 65535

Trigger Detection Point:

0 = Connect

1 = Talking

2 = Disconnect

SS7 Release Cause:

0 to 127

Treatment:

0 to 254

- Comments:**
1. The SS7 Release Cause field specifies the release cause value that caused the trigger.
 2. The release cause trigger event only applies during the connect state. In other trigger detection points (i.e. talking and disconnect), the release cause is not reported to the service node.

Byte #	Meaning
0	ID = #125
1	
2	DB Byte Count
3	
4	Trigger Detection Point
5	SS7 Release Cause
6	Treatment
7	Padding Field

9.99 Trigger Ringing DB

Identifier: #12A

Description: This data block indicates information regarding the backward call indicator that initiated this session with the service node.

Size: 8 bytes

Data Block(s): **Identifier:**

#12A

DB Byte Count:

0 to 65535

Trigger Detection Point:

0 = Connect

1 = Talking

2 = Disconnect

SS7 Backward Call Indicators:

A two byte SS7 set field containing SS7 information such as:

Charge Indicator, Called Party's Status Indicator, Called Party's Category Indicator, End-to-end Method Indicator, Interworking Indicator, IAM Segmentation Indicator, ISDN User Part Indicator, Holding Indicator, ISDN Access Indicator, Echo Control Device Indicator (See TR-NWT-000246 for details).

Comments: 1. The ringing trigger event is associated with the B party. Therefore, the SS7 Backward Call Indicators field is for the B party.

Byte #	Meaning
0	ID = #12A
1	
2	DB Byte Count
3	
4	Trigger Detection Point
5	Padding Field
6	SS7 Backward Call Indicators
7	

9.100 Trigger Treatment DB

Identifier: #126

Description: This data block indicates information regarding the switch trigger treatment event that initiated this session with the service node.

Size: 6 bytes

Data Block(s): **Identifier:**

#126

DB Byte Count:

0 to 65535

Trigger Detection Point:

0 = Connect

1 = Talking

2 = Disconnect

Treatment:

0 to 254

- Comments:**
1. The Treatment field specifies the treatment that caused the trigger.
 2. The treatment field is associated with the A party.

Byte #	Meaning
0	ID = #126
1	
2	DB Byte Count
3	
4	Trigger Detection Point
5	Treatment

9.101 Voice Channel DB

Identifier: #127

Description: Voice channel information for the voice connection between the switch and the service node.

Size: 6 bytes

Data Fields: **Identifier:**

#127

DB Byte Count:

0 to 65535

Voice Channel ID:

0 to 8191

Comments: 1. None.

Byte #	Meaning
0	ID = #127
1	
2	DB Byte Count
3	
4	Voice Channel ID
5	

Chapter 10: Field Definitions

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
ADACC Transition Indicator	1 byte	• Preprocessing Information DB
Access Indicator	1 byte	• OSSAIN Call Queue Event DB • OSSAIN Session Event DB
Alarm Severity	1 byte	• Alarm Information DB
Alarm Status	1 byte	• Alarm Information DB
Allowed Billing Set	2 bytes	• Billing Information DB
AMA Data	70 bytes	• AMA Module DB
AMA Handling	1 byte	• AMA Module DB
AMA Hex Digit Count	1 byte	• AMA Module DB
ANI Status	1 byte	• Directory Number DB
Append AMA Sequence Number	1 byte	• AMA Module DB
Application ID	1 byte	• Information ID DB
ASCII Card Number	30 bytes	• ASCII Card Number DB
ASCII Card Number Length	1 byte	• ASCII Card Number DB
ASCII Card Number Type	1 byte	• ASCII Card Number DB
ASCII PIN Length	1 byte	• ASCII Card Number DB
Authorization Number	6 bytes	• Originating Station Information DB
Authorization Number Length	1 byte	• Originating Station Information DB
Available Sessions	2 bytes	• Throttle DB
BCD Card Number	16 bytes	• BCD Card Number DB
BCD Card Number Digit Count	1 byte	• BCD Card Number DB
BCD Card Number Type	1 byte	• BCD Card Information DB • BCD Card Number DB
BCD PIN Digit Count	1 byte	• BCD Card Number DB
BNS Query Response	1 byte	• BNS Query Result DB
Billable Status	1 byte	• Charge Status DB
Billing Acceptance Response Method	1 byte	• Billing Acceptance DB
Billing Data	1 byte	• Class Charge DB • OSSAIN Session Event DB
Billing Number Input Method	1 byte	• Billing Number Input DB
Billing Number Type	1 byte	• Alternate Billing Directory Number DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Branding Status	1 byte	• Branding Indicator DB • Call Information DB
Broadcast Cycle Count	1 byte	• Broadcast DB
CCV Query Response	1 byte	• CCV Query Result DB
Call Handling	1 byte	• End Call Handling DB
Call ID	4 bytes	• CallID DB • Conference Party Connection Information DB • Conference SN Voice Connection Information DB • OAP Call Processing Class Header • OSSAIN Call Queue Event DB • OSSAIN Session Event DB
Call In Queue Take Down	1 byte	• Call In Queue DB • Call Information DB • Transfer Control List DB
Call Origination Type	1 byte	• Call Information DB • OSSAIN Session Event DB
Call Queue	2 bytes	• Call Queue Information DB • OSSAIN Call Queue Event DB • OSSAIN Session Event DB
Call Type for Queuing	2 bytes	• OSSAIN Call Queue Event DB • OSSAIN Session Event DB
Calls In Queue	2 bytes	• Call Queue Information DB • OSSAIN Call Queue Event DB
Calling Line Identifier Digit Count	1 byte	• Directory Number DB • Directory Number Update DB
Calling Line Identifier Number	10 bytes	• Directory Number DB • Directory Number Update DB
Carrier Assignment Source Indicator	1 byte	• Carrier DB • Carrier Number DB
Carrier Identification Code	2 bytes	• Carrier DB • Carrier Number DB • OSSAIN Call Queue Event DB • OSSAIN Session Event DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Carrier Identification Digit Count	1 byte	- Carrier DB - Carrier Number DB - OSSAIN Call Queue Event DB - OSSAIN Session Event DB
Carrier Name	8 bytes	Carrier DB
Carrier Name Length	1 byte	Carrier DB
Carrier Number Status	1 byte	Carrier DB
Carrier Service Status	1 byte	Carrier DB
Class Charge Selector	1 byte	Class Charge DB
Coin Key Pad State	1 byte	Originating Station Information DB
Coin Overtime Type	1 byte	Originating Station Information DB
Conference ID	4 bytes	- Conference ID DB - Conference Party Connection DB
Connection Event	1 byte	Connection Event DB
Connection ID	1 byte	- Conference Party Connection DB - Conference Party Connection Information DB - Connection Event DB - Connection ID DB - Connection Progress DB - LNP Information DB - Language DB - Release Cause DB - Speech Path DB - Treatment DB
Connection Status	1 byte	- Conference Party Connection Information DB - Directory Number DB
Context Block Data	50 bytes	Context Block DB
Context Block Length	1 byte	Context Block DB
Control List ID	2 bytes	- Call Information DB - Transfer Control List DB
Conversation Timing Status	1 byte	- Call Information DB - Conversation Timing Indicator DB
Country Direct Access Code Count	1 byte	Country Direct Information DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Country Direct Access Code Number	2 bytes	• Country Direct Information DB
Country Direct Carrier Code Count	1 byte	• Country Direct Information DB
Country Direct Carrier Code Number	2 bytes	• Country Direct Information DB
Country Direct Country Code Count	1 byte	• Country Direct Information DB
Country Direct Country Code Number	2 bytes	• Country Direct Information DB
Current Broadcast Connections	2 bytes	• Broadcast Count DB
DB Byte Count	2 bytes	• All Data Blocks
DTMF Digit	1 byte	• DTMF Digit DB • Trigger DTMF DB
DTMF Receiver	1 byte	• Call Information DB
Data Block ID	2 bytes	• All Data Blocks
Day	1 byte	• Call Timing DB • Information Id DB • Time DB
Directory Number	10 bytes	• Alternate Billing Directory Number DB • Directory Number DB • Directory Number Update DB • ISUP Calling Party Number DB
Directory Number Digit Count	1 byte	• Alternate Billing Directory Number DB • Directory Number DB • Directory Number Update DB • ISUP Calling Party Number DB
Directory Number Type	1 byte	• Alternate Billing Directory Number DB • Directory Number DB • Directory Number Information DB • Directory Number Update DB • OSSAIN Session Event DB
Elapsed Minutes	3 bytes	• Call Timing DB • Notification Timer DB • Trigger Profile Index DB
Error ID	2 bytes	• Return Error Operation Header
Estimated Charges	4 bytes	• Estimate of Charges Duration Information DB • Estimate of Charges No Duration Information DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Estimate Duration	2 bytes	• Estimate of Charges Duration DB
Float Connection Indicator	1 byte	• Call Float Indicator DB
Function ID	2 bytes	• Call Processing Class Header • Function DB • OSSAIN Call Queue Event DB • OSSAIN Session Event DB
Function Provider Request Status	1 byte	• Call Queue Information DB
Generate AMA Sequence Number	1 byte	• Generate AMA DB
Handling Data	1 byte	• Class Charge DB • OSSAIN Session Event DB
Hotel Room Number	6 bytes	• Originating Station Information DB
Hotel Room Number Length	1 byte	• Originating Station Information DB
Hour	1 byte	• Call Timing DB • Information Id DB • Time DB
Invalid Billing Set	2 bytes	• Invalid Billing Details DB
Invoke ID	2 bytes	• All Operation Headers
LATA Restrictions	1 byte	• LATA Information DB
LATA Status	1 byte	• LATA Information DB
LNP Query	1 byte	• LNP Enhancements DB
LRN	5 bytes	• LNP Information DB
LRN Digit Count	1 byte	• LNP Information DB
Language	1 byte	• Alternate Billing Directory Number DB • Directory Number DB • Directory Number Update DB • Language DB • OSSAIN Session Event DB
Logical Voice Channel ID	2 bytes	• SN Voice Link DB
Maintenance State	1 byte	• Maintenance State DB
Maintenance Text	32 bytes	• Maintenance Text DB
Maintenance Text Length	1 byte	• Maintenance Text DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Minutes	1 byte	• Call Timing DB • Information Id DB • Time DB
MIS Sequence Number ID	1 byte	• OSSAIN Call Queue Event DB • OSSAIN Call Session Event DB • OSSAIN Non-Call Session Event DB
Month	1 byte	• Call Timing DB • Information Id DB • Time DB
NBEC Code	2 bytes	• NBEC DB
NBEC Code Digit Count	1 byte	• NBEC DB
Network ID	2 bytes	• MIS Class Header
Network Service ID	2 bytes	• Billing Information DB • Network Service DB • OSSAIN Session Event DB
No Answer Time	1 byte	• Trigger Profile Index DB
Node ID	2 bytes	• Node Identification DB • OSSAIN Call Queue Event DB • OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB
Notification Time Type	1 byte	• Notification Timer DB • Trigger Profile Index DB
Notification Timer Action	1 byte	• Notification Timer DB
Notification Timer Mode	1 byte	• Trigger Profile Index DB
Notification Timer Status	1 byte	• Call Information DB
Number of Operator to SN Handoffs	1 byte	• Call Information DB • Call Update DB
OLNS Account Owner	4 bytes	• OLNS Information DB
OLNS Account Owner Length	1 byte	• OLNS Information DB
OLNS Alphanumeric String	8 bytes	• OLNS Information DB
OLNS Alphanumeric String Length	1 byte	• OLNS Information DB
OLNS Billing Service Provider	4 bytes	• OLNS Information DB
OLNS Billing Service Provider Length	1 byte	• OLNS Information DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
OLNS DA Credit Card Indicator	1 byte	• OLNS Information DB
OLNS DA Sent Paid Indicator	1 byte	• OLNS Information DB
OLNS DA Special BNS Indicator	1 byte	• OLNS Information DB
OLNS DA Third Number Indicator	1 byte	• OLNS Information DB
OLNS DACC Indicator	1 byte	• OLNS Information DB
OLNS Foreign Language	1 byte	• OLNS Information DB
OLNS InterLATA Carrier Indicator	1 byte	• OLNS Information DB
OLNS InterLATA Carrier ID Digit Count	1 byte	• OLNS Information DB
OLNS InterLATA Carrier Identification Code	2 bytes	• OLNS Information DB
OLNS International Carrier Indicator	1 byte	• OLNS Information DB
OLNS International Carrier ID Digit Count	1 byte	• OLNS Information DB
OLNS International Carrier Identification Code	2 bytes	• OLNS Information DB
OLNS IntraLATA Toll Carrier Indicator	1 byte	• OLNS Information DB
OLNS IntraLATA Toll Carrier ID Digit Count	1 byte	• OLNS Information DB
OLNS IntraLATA Toll Carrier ID Code	2 bytes	• OLNS Information DB
OLNS Service/Equipment Indicator	1 byte	• OLNS Information DB
OLNS TA Collect Indicator	1 byte	• OLNS Information DB
OLNS TA Credit Card Indicator	1 byte	• OLNS Information DB
OLNS TA Sent Paid Indicator	1 byte	• OLNS Information DB
OLNS TA Special BNS Indicator	1 byte	• OLNS Information DB
OLNS TA Third Number Indicator	1 byte	• OLNS Information DB
OLNS Treatment Indicator	1 byte	• OLNS Information DB
Operation ID	2 bytes	• Invoke Operation Header
Operator ID	2 bytes	• Operator Identification DB
Operator Services System Action	1 byte	• Operator Services System Action DB
Operator System Selector	1 byte	• Operator System DB
Operator System Type	1 byte	• Preprocessing Information DB
Party	1 byte	• DTMF Digit DB • Directory Number DB • Directory Number Information DB • Directory Number Update DB • Trigger DTMF DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Pass Through Data	100 bytes	• Pass Through DB
Pass Through Data Length	1 byte	• Pass Through DB
Passive Node Status	1 byte	• Passive Node Status DB
Phone Class Type	1 byte	• OSSAIN Session Event DB • Originating Station Information DB
Portability Status	1 byte	• LNP Information DB
Predicted Wait Time	2 bytes	• Call Queue Information DB
Presentation Indicator	1 byte	• Directory Number DB • Directory Number Update DB • ISUP Calling Party Number DB
Protocol Version	2 bytes	• OAP Class Headers
Queue Peg Indicators	1 byte	• OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB
Queuing Event	1 byte	• OSSAIN Call Queue Event DB
Queuing System	1 byte	• Call Queue Information DB
RAO Code	2 bytes	• Billing Number Validation DB
RAO Digit Count	1 byte	• Billing Number Validation DB
Release Node Indicator	1 byte	• Release Node DB
Report Text	80 bytes	• Report Text DB
Report Text Length	1 byte	• Report Text DB
Requested RLT Action	1 byte	• RLT Request DB
Reset Time Flag	1 byte	• OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB
Restart Severity	1 byte	• Information Id DB
Restricted Billing Index	1 byte	• Billing Information DB • OSSAIN Session Event DB
Routing Index	2 bytes	• Routing Index DB
SN Voice Conferencing Status	1 byte	• Conference SN Voice Connection DB
SN Voice Link Status	1 byte	• SN Voice Link DB
SPID Processing	1 byte	• Call Information DB
SS7 Backward Call Indicators	2 bytes	• Connection Progress DB • Trigger Ringing DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
SS7 Release Cause	1 byte	- Release Cause DB - Trigger Release Cause DB
Seconds	1 byte	- Call Timing DB - Information Id DB - Notification Timer DB - Time DB - Trigger Profile Index DB
Sequence Number	2 bytes	- All Class Headers - Sequence Number Check DB
Service Node Mode	1 byte	- Call Information DB - Node Identification DB
Service Peg Indicators	1 byte	OSSAIN Session Event DB
Service Provider ID	4 bytes	- OSSAIN Call Queue Event DB - OSSAIN Session Event DB - SPID DB
Service Provider ID Length	1 byte	- OSSAIN Call Queue Event DB - OSSAIN Session Event DB - SPID DB
Service Provider Party Identifier	1 byte	SPID DB
Service Type	1 byte	- Call Information DB - OSSAIN Call Queue Event DB - OSSAIN Session Event DB - Requested Service DB
Service Type Provided by Switch	1 byte	Carrier DB
Serving Agent Status	1 byte	Call Queue Information DB
Session ID	2 bytes	- Call Processing Class Header - OSSAIN Call Queue Event DB - OSSAIN Non-Call Session Event DB - OSSAIN Session Event DB - Session Information DB
Session Origination Type	1 byte	Call Information DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Session Pool ID	2 bytes	• Call Processing Class Header • OSSAIN Call Queue Event DB • OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB • Session Information DB • Session Pool Class Header • Session Pool Alarm Class Header • Session Pool Log Class Header
Session State	1 byte	• OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB
Session Type	1 byte	• Call Information DB
Solicitor Number	2 bytes	• Node Class Header • Node Alarm Class Header • Node Log Class Header • Session Pool Class Header • Session Pool Alarm Class Header • Session Pool Log Class Header
Special Number Format	1 byte	• OSSAIN Session Event DB
Special Number Type	1 byte	• OSSAIN Session Event DB
Special Number Validity	1 byte	• OSSAIN Session Event DB
Speech Path Status	1 byte	• Conference Party Connection Information DB • Directory Number DB • Speech Path DB
Tenths of Seconds Timestamp	2 bytes	• Information Id DB • OSSAIN Call Queue Event DB • OSSAIN Non-Call Session Event DB • OSSAIN Session Event DB
Tenths Seconds	1 byte	• Call Timing DB
Terminal ID	2 bytes	• Operator Identification DB
Terminating Code Allowed Set	1 byte	• Billing Information DB
Text Block	20 bytes	• Text DB
Text Block Length	1 byte	• Text DB
Time In Queue	2 bytes	• OSSAIN Call Queue Event DB

Table 12. Fields To Data Blocks Mapping Table

Fields	Size	Data Blocks
Treatment	1 byte	- Treatment DB - Trigger Release Cause DB - Trigger Treatment DB
Trigger Detection Point	1 byte	- Trigger DTMF DB - Trigger Information DB - Trigger Release Cause DB - Trigger Ringing DB - Trigger Treatment DB
Trigger Event	1 byte	Trigger Information DB
Trigger Profile Index	2 bytes	- Call Information DB - Trigger Profile Index DB
Trunk Group ID	2 bytes	- Originating Trunk Information DB - OSSAIN Call Queue Event DB - OSSAIN Session Event DB - Terminating Trunk Information DB - SN Voice Link DB
Trunk Member ID	2 bytes	- Originating Trunk Information DB - OSSAIN Call Queue Event DB - OSSAIN Session Event DB - Terminating Trunk Information DB - SN Voice Link DB
Type of Service Provider	1 byte	SPID DB
Unenforced Restrictions	1 byte	- Directory Number DB - Directory Number Information DB
Validation Request	1 byte	Class Charge DB
Validation Result	1 byte	Billing Number Validation DB
Validation Result Determined by DMS Datafill	1 byte	Billing Number Validation DB
Validation Sequence Number	1 byte	- Billing Number Validation DB - Class Charge DB
Voice Channel ID	2 bytes	Voice Channel DB
Year	1 byte	- Call Timing DB - Information Id DB - Time DB

10.1 ADACC Transition Indicator

Description: Specifies whether a DA call has transitioned to a toll call for Automated Directory Assistance Call Completion. A call is considered transitioned if a subscriber has elected to have the TOPS switch automatically complete a call to the DA requested number.

Length: 1 byte

Format: Binary

Range: 0 = Automated Directory Assistance Call Completion Transition False
1 = Automated Directory Assistance Call Completion Transition True

Definition: This field identifies if a DA call has transitioned TA for ADACC service.

- Value 0 = A call has not transitioned from DA to toll for Automated DA Call Completion
- Value 1 = A DA call has transitioned to a toll call for Automated DA Call Completion

Example: Automated Directory Assistance Call Completion Transition True

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.2 Access Indicator

Description: Indicates whether the call is to be completed by the operating company or by a carrier.

Length: 1 byte

Format: Binary

Range: 0 = Operating Company
1 = Carrier

Definition: The following are the values for this field:

- Value 0 = The call will be completed by an operating company.
- Value 1 = The call will be completed by a carrier.

Example: The call will be completed by a carrier.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.3 Alarm Severity

- Description: Specifies the alarm status.
- Length: 1 Byte
- Format: Binary
- Range: 0 = Critical
1 = Major
2 = Minor
- Definition: Indicates the severity of the software alarm that needs to be turned on or off.
- Example: Alarm severity is major.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.4 Alarm Status

- Description: Specifies if an alarm is/should be turned on or off.
- Length: 1 Byte
- Format: Binary
- Range: 0 = On
1 = Off
- Definition: Indicates if an alarm is/should be turned on or off.
- Example: Alarm is turned on.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.5 Allowed Billing Set

Description: Indicates which type of billing methods the billed party may choose from when specifying billing for the call.

Length: 2 bytes

Range: Set

Format: Bit 0 = Station Paid
Bit 1 = Station Collect
Bit 2 = Station Special Calling Card
Bit 3 = Station Special Calling 3rd Number
Bit 4 = Station Special Called Card
Bit 5 = Station Special Called 3rd Number
Bit 6 = Person Paid
Bit 7 = Person Collect
Bit 8 = Person Special Calling Card
Bit 9 = Person Special Calling 3rd Number
Bit 10 = Person Special Called Card
Bit 11 = Person Special Called 3rd Number

Definition: This field indicates the types of billing that are valid for the billed party. The value for this field is determined using switch tables REGION, DOMBILL, and OVSBIL. The restricted billing index value is also used to determine the value for the allowed billing set.

- A bit set to one indicates the billing associated with that bit is allowed and may be specified by the billed party. A bit set to zero indicates the billing is restricted and may not be selected.
- Note that bit 12 - 15 are unused and will have a value of zero.
- Bit 0, Station Paid = Call is station to station, billing to calling party's directory number is allowed.
- Bit 1, Station Collect = Call is station to station, billing to called party's directory number is allowed.
- Bit 2, Station Special Calling Card = Call is station to station, calling party can request billing the call to a calling card number.
- Bit 3, Station Special Calling 3rd Number = Call is station to station, calling party can request billing the call to a third number.
- Bit 4, Station Special Called Card = Call is station to station, called party can request billing the call to a calling card number.

- Bit 5, Station Special Called 3rd Number = Call is station to station, called party can request billing the call to a third number.
- Bit 6, Person Paid = Call is person to person, billing to calling party's directory number is allowed.
- Bit 7, Person Collect = Call is person to person, billing to called party's directory number is allowed.
- Bit 8, Person Special Calling Card = Call is person to person, calling party can request billing the call to a calling card number.
- Bit 9, Person Special Calling 3rd Number = Call is person to person, calling party can request billing the call to a third number.
- Bit 10, Person Special Called Card = Call is person to person, called party can request billing the call to a calling card number.
- Bit 11, Person Special Called 3rd Number = Call is person to person, called party can request billing the call to a third number.
- Auto Collect is always allowed for dialling toll-free numbers.

Example: Station Paid and Person Paid billing methods are allowed.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1

10.6 AMA Data

Description: Specifies the formatted AMA data to be appended to the billing record.

Length: 70 bytes (140 hex digits)

Format: Unknown to the switch

Range: None.

Definition: The AMA data for billing. Refer to the AMA Hex Digit Count field for the size of the AMA data.

- There is a method defined for identifying the format of the custom AMA data. Refer to the TOPS Custom AMA document (NIS Q259-1).

Example: A block of AMA data whose field format is unknown to the switch. (Each byte containing up to two hex digits, as indicated by the AMA Hex Digit

Count field.) Digit string 'B0E334...95'.

7	4	3	0	Byte #
B			0	0
E			3	1
3			4	2
...			...	...
9			5	69

10.7 AMA Handling

Description: Indicates whether any AMA previously buffered by the DMS should be cleared before the new AMA is added to the DMS buffer.

Length: 1 byte

Format: Binary

Range: 0 = Do Not Clear Previously Appended AMA

1 = Clear Previously Appended AMA

Definition: This field is used to provide a method to clear previously appended AMA. By setting the field to Clear Previously Appended AMA, any AMA that was previously appended will be cleared prior to the addition of new AMA data. (Note: Following a Generate AMA Record Request, the AMA data in the DMS buffer will have been recorded and the buffer automatically cleared. The AMA Handling 'clear' must be used prior to a Generate AMA Record, or the call end, if the desire is to avoid recording of some prior Append AMA Requests.) The following are the values for the field:

- Value 0 = The accompanying new AMA data will be appended to any previously appended AMA data in the DMS buffer.
- Value 1 = Any previously appended AMA will be cleared from the DMS buffer prior to the addition of the accompanying new AMA data.

Example: Do Not Clear Previously Appended AMA.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.8 AMA Hex Digit Count

Description: Specifies the number of hex digits of billing data in the AMA Module Data Block (in the AMA Data field).

Length: 1 byte

Format: Binary

Range: 0 to 140 hex digits

Definition: This field indicates the number of hex digits of formatted billing data to append to the billing record. A value of 0 adds no new AMA (may be used with the AMA Handling field to clear all buffered AMA).

Example: AMA Hex Digit Count of 48 (which will take up 24 bytes in the AMA Data field).

7	6	5	4	3	2	1	0
0	0	1	1	0	0	0	0

10.9 ANI Status

Description: Specifies the Automatic Number Identification (ANI) status for the A party.

Length: 1 byte

Format: Binary

Range: 0 = ANI Success
1 = Operator Number Identification (ONI)
2 = ANI Office Failure
3 = ANI Failure Detected
4 = Not Applicable
254 = Unknown

Definition: The following are the values for this field.

- Value 0 = The directory number was successfully received through signalling.
- Value 1 = The directory number was not received through signalling. ONI means the end office does not normally send the directory number for the A party. An example is a party line where the end office is unable to detect which party has gone off-hook.
- Value 2 = The directory number was not received through signalling. ANI office failure means the end office normally send the directory number for the A party, but was unable to send it for this particular call. ANI office failure usually indicates equipment problems in the end office.
- Value 3 = The directory number was not received through signalling. The end office attempted to send the directory number but the switch did not receive the proper format.
- Value 4 = The ANI status is not applicable. The ANI status only applies to the A party.
- Value 254 = The ANI status is unknown.

Example: ANI status is success.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.10 Application ID

Description: Specifies the QMS MIS application.

Length: 1 byte

Format: Binary

Range: 0 = TOPS
1 = OSSAIN

Definition: The QMSMIS application is datafilled in table QMSMIS.

Example: The application is OSSAIN.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.11 Append AMA Sequence Number

Description: Specifies the number of times a service node has requested an AMA module be appended. This number is used by the switch to avoid duplicate or lost billing. Sequential requests must have an increasing sequence number or the switch will not process them, but will instead return an error.

Length: 1 byte

Format: Binary

Range: 0 to 254

Definition: A sequence number field specifying the number of times an append AMA has been requested.

Example: Sequence Number is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.12 ASCII Card Number

Description: Specifies the ASCII card number.

Length: 30 bytes

Format: ASCII

Range: Alphanumeric ASCII characters {A-Z, a-z, 0-9}

Definition: This field contains the ASCII card number and the PIN, if any, used for billing. It is variable in length. The ASCII Card Number Length field will indicate the length of the card number and the PIN, if any.

Example: ASCII card number is '33M12345'

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
3	33	0	0	1	1	0	0	1	1	0
3	33	0	0	1	1	0	0	1	1	1
M	4D	0	1	0	0	1	1	0	1	2
1	31	0	0	1	1	0	0	0	1	3
2	32	0	0	1	1	0	0	1	0	4
3	33	0	0	1	1	0	0	1	1	5
4	34	0	0	1	1	0	1	0	0	6
5	35	0	0	1	1	0	1	0	1	7
nil	#FF	1	1	1	1	1	1	1	1	8
nil	#FF	1	1	1	1	1	1	1	1	...
nil	#FF	1	1	1	1	1	1	1	1	29

10.13 ASCII Card Number Length

Description: The length of the ASCII Card Number, including the PIN.

Length: 1 byte

Format: Binary

Range: 0 to 30

Definition: The number of characters in the ASCII card number.

Example: Length is 22.

7	6	5	4	3	2	1	0
0	0	0	1	0	1	1	0

10.14 ASCII Card Number Type

Description: Specifies the ASCII card number type

Length: 1 byte

Format: Binary

Range: 0 = Foreign
1 = Custom

Definition: The types of ASCII cards are:

- Value 0 = ASCII card is a Foreign format card:
 - alphanumeric, variable length digit string
 - begins with a country code that is 2-3 digits in length
 - contains an alphabetical character, following the country code, that indicates the dates during which the credit card number is valid.
 - contains an account number that follows the alphabetical character; the length of the account number varies, depending on the issuing country.
- Value 1 = ASCII card has a custom format, the switch does no format checking

Example: ASCII card number type is Foreign

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.15 ASCII PIN Length

Description: Specifies the length of the PIN contained in the ASCII card number.

Length: 1 byte

Format: Binary

Range: 0 to 30

Definition: The length of the PIN in the ASCII Card Number field

Example: Length is 6.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0

10.16 Authorization Number

Description: This field indicates the authorization number of the A party.

Length: 6 bytes

Format: ASCII

Range: {0 to 9, A to Z}

Definition: An authorization number for the call. This field may be required if the phone class type is Institution, Hotel/Motel, Hotel/Motel With No Charge. If there is not an authorization number, then the field value is set to the nil value (#FFF).

- Up to six characters can be supplied. Numerics are treated as one character each, and the upper case alphas are treated as two characters each.
- “123456”, “W1212”, “WW12”, and “WWW” are acceptable combinations (in any order or combination).
- “WWWW”, “WWW1”, and “WW123” are not acceptable combinations because they exceed the six character limitation.

Example: An authorization number of A13C

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
A	41	0	1	0	0	0	0	0	1	0
1	31	0	0	1	1	0	0	0	1	1
3	33	0	0	1	1	0	0	1	1	2
C	43	0	1	0	0	0	0	1	1	3
nil	FF	1	1	1	1	1	1	1	1	4
nil	FF	1	1	1	1	1	1	1	1	5

10.17 Authorization Number Length

Description: Specifies the number of characters in the Authorization Number field.

Length: 1 byte

Format: Binary

Range: 0 to 6

Definition: This field specifies the number of characters in the Authorization Number Field.

Example: Length is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.18 Available Sessions

Description: Specifies the absolute number of available sessions for a session pool.

Length: 2 Bytes

Format: Binary

Range: 0 to 1023

Definition: Indicates the number of sessions in a session pool that can be used for call processing, thereby adjusting the maximum number of calls that can be handled on the given session pool. A value of 0 indicates that no calls should be handled on the given session pool. When a value above 0 is used, the maximum session ID that can be used for call processing is set to the value of this field minus 1.

Example: Session number is 6.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	1

10.19 BCD Card Number

Description: Specifies the card number for billing

Length: 16 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: The card number being used for billing, including the PIN, if any. The PIN, if present, is at the end of the number.

Example: Billing number is 91999167892000

7	4	3	0	Byte #
9		1		0
9		9		1
9		1		2
6		7		3
8		9		4
2		0		5
0		0		6
F		F		7
F		F		...
F		F		15

10.20 BCD Card Number Digit Count

Description: Specifies the number of digits in the BCD card number, including the PIN, if any.

Length: 1 byte

Format: Binary

Range: 0 to 32

Definition: The number of digits in the calling card number.

Example: BCD card number digit count is 12.

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0

10.21 BCD Card Number Type

Description: Specifies the type of BCD card number.

Length: 1 byte

Format: Binary

Range: 0 = ISO Card
1 = Directory Number Card
2 = RAO Number Card
3 = Custom Card
254 = Unknown

Definition: The types of calling card are:

- Value 0 = Card is an ISO format card:
 - conforms to ISO standard 7812
 - are between 5 and 23 digits in length, including the PIN.
- Value 1 = Card has directory number format:
 - fixed length of 14 digits
 - begins with a valid NPA
 - fourth digit is in the range of '2' to '9'
 - has a 4 digit PIN, at the end of the number.
- Value 2 = Card has an RAO (Regional Accounting Office) format:
 - fixed length of 14 digits
 - begins with a 3 digit series (in the range of 600 to 799) that represents the RAO
 - fourth digit is either a '0' or '1'

— Note: The switch does not enforce the range to be between 600-799. It recognizes an RAO card by the fact that the fourth digit is '0' or '1' and the card is 14 digits long.

- Value 3 = Card is a custom card. If this value is sent to the switch, the switch does not perform any format checks on the card number.
- Value 254 = The number type is unknown. The switch determines the type of card, performs format checks and returns the type of card to the service node.

Example: Card number type is a directory number format card.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.22 BCD PIN Digit Count

Description: Specifies the size of the BCD card PIN within the BCD card number (at the end)

Length: 1 byte

Format: Binary

Range: 0 to 32

Definition: The size of the PIN within the BCD card number.

Example: PIN has size of 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.23 BNS Query Response

Description: Specifies the response received from a BNS billing validation query.

Length: 1 byte

Format: Binary

- Range:** 0 = Miscellaneous
1 = Vacant Code
2 = Semi Public Coin Phone
3 = Public Coin Phone
4 = Public Coinless Phone
5 = Non Working Number
- Definition:** The BNS validation query response.
- Value 0 = Any response not listed below.
 - Value 1 = Billing number is a vacant code.
 - Value 2 = Equipment is semi public coin phone.
 - Value 3 = Equipment is public coin phone.
 - Value 4 = Equipment is public coinless phone.
 - Value 5 = Invalid Network Number.
- Example:** The billed number is a Public Coin Phone.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.24 Billable Status

- Description:** Specifies whether or not to mark the call billable in the AMA record
- Length:** 1 byte
- Format:** Binary
- Range:** 0 = Not Billable
1 = Billable
- Definition:** When sent by the switch, the default value for this field is based on the CT4Q. The value is datafilled in switch table CT4QNAMS, field NOAMA.
- Value 0 = The call will be marked as not billable on the AMA record.
 - Value 1 = The call will be marked billable on the AMA record.
- Example:** The call should be marked billable on the AMA record.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.25 Billing Acceptance Response Method

Description: Specifies the method by which billing acceptance was obtained from the billed party.

Length: 1 byte

Format: Binary

Range: 0 = DTMF
1 = Verbal to Operator
2 = Derived
3 = Voice Recognition
254 = Unknown

Definition: Billing Acceptance Method

- Value 0 = Billing acceptance response was given using DTMF digits.
- Value 1 = Billing acceptance given to an operator.
- Value 2 = Billing acceptance should be set to derived for station collect and third number billed calls for which the billed party is not connected for billing acceptance. This includes cases where billing acceptance is accepted or rejected as a result of a validation query, Service Node data-fill, or for calls in which the called party is not connected due to time-of-day or other restrictions.
- Value 3 = Billing acceptance response was obtained by the use of voice recognition.
- Value 254 = This is the correct field setting for all calling card calls, including both rejected and accepted calling cards.

Example: Billing acceptance by voice recognition.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.26 Billing Data

Description: Specifies the class charge billing for the call.

Length: 1 byte

Format: Binary

Range: 0 = Paid
1 = Collect
2 = Special Calling
3 = Special Called
254 = Unspecified

Definition: This field contains the billing data for the call. The Handling Data contains the class charge type.

- Value 0 = The A party is paying for the call.
- Value 1 = The B party is paying for the call.
- Value 2 = The A party requests an alternate billing number be used to pay for the call. The alternate billing number can be either a third number or a calling card number.
- Value 3 = The B party requests an alternate billing number be used to pay for the call. The alternate billing number can be either a third number or a calling card number.
- Value 254 = The class charge is unspecified.

Example: Class charge billing is special calling.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.27 Billing Number Input Method

Description: Specifies how the billing number was received.

Length: 1 byte

Format: Binary

Range: 0 = DTMF

1 = Verbal
2 = Derived
3 = Voice Recognition
254 = Unknown

Definition: How the billing number was received for the call.

Example: The billing number was obtained verbally.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.28 Billing Number Type

Description: Specifies the billing number type.

Length: 1 byte

Format: Binary

Range: 0 = Alternate Directory Number
1 = Collect
2 = Account Code

Definition: The following are the values for this field:

- Value 0 = The billing number is an alternate directory number (e.g. third party directory number).
- Value 1 = The billing number is the forward party's directory number (e.g. a collect call).
- Value 2 = The call is being billed to the A party. However, the billing is for a specific account number. The account number must be 4 digits in length. The number should be padded in front with zeros if the account code is smaller than four digits (e.g. 0123 for an account code number of 123).

Example: Billing number type is alternate directory number.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.29 Branding Status

Description: Indicates whether a call has received front-end branding.

Length: 1 byte

Format: Binary

Range: 0 = Call not branded
1 = Call branded

Definition: The front-end branding status for the call.

- Value 0 = Call has not received any front-end branding.
- Value 1 = Call has received front-end branding. The branding status is set to 1 if branding is provided internally by the TOPS switch. The branding status may also be 1 if the SN has previously sent the Branding Indicator DB with a branding status of 1.
- At the point in a DA call that the calling party elects to have TOPS automatically complete a call to the requested number (Automated Directory Assistance Call Completion), the switch resets the branding status to 0.

Example: The call has received front-end branding from either the switch or service node.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.30 Broadcast Cycle Count

Description: Indicates if cycle counting should be done by the switch.

Length: 1 byte

Format: Binary

Range: 0 to 255

- Definition:** Indicates the number of cycles of a broadcast recording to be monitored by the switch. This field has two purposes:
- indicates whether the DMS is to count cycles and have the responsibility of releasing the voice link.
 - if the DMS is to count cycles, indicates the number of cycles of a recording to be heard by parties on a broadcast voice link before it is released.
 - Cycle counting is only applicable to voice links datafilled in switch table OAVLMAP with a value of HKCHG in field CUTTHRU.
 - Value 0 = cycle counting is not required.
 - Value 1 to 255 = the number of cycles of a recording to be heard by parties on a broadcast voice link before it is released.
- Example:** The subscriber is to hear 2 cycles of a broadcast recording before the voice link is released by the switch.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.31 CCV Query Response

- Description:** Specifies the result received from a CCV billing validation query.
- Length:** 1 byte
- Format:** Binary
- Range:**
- 0 = Miscellaneous
 - 1 = PIN Mismatch
 - 2 = PIN Unrestricted
 - 3 = PIN Restricted
 - 4 = PSDI of 2 - Threshold Exceeded
 - 5 = PSDI of 3 - Non-Payment
 - 6 = PSDI of 4 - Service Restriction
 - 7 = CSDI of 3 - Service Denial
- Definition:** CCV Query response
- Value 0 = Any response not listed below.
 - Value 1 = PIN included in the calling card number is not valid.

- Value 2 = The calling card number may be used to bill calls to any destination.
- Value 3 = The calling card number may be used to bill calls to the directory number indicated by the last ten digits of the calling card number.
- Value 4 = PIN Service Denial for threshold exceeded, used for fraud detection.
- Value 5 = PIN Service Denial for non payment.
- Value 6 = PIN Service Denial for service restriction. Service for this PIN is blocked due to geographic, time or other restriction.
- Value 7 = Card Service Denial, the calling card has been deactivated.

Example: The PIN is Unrestricted

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.32 Call Handling

Description: Specifies whether the call is being aborted as a result of normal call handling or due to an error.

Length: 1 byte

Format: Binary

Range: 0 = Normal Call Handling - Bill the call
 1 = Normal Call Handling - Cancel Billing
 2 = Error Recovery

Definition: The following are the values for this field:

- Value 0 = The call needs to be taken down as part of a normal call flow (e.g. the A party went on-hook). A billable AMA record is produced.
- Value 1 = The call needs to be taken down as part of normal call flow. A non-billable AMA record is produced.
- Value 2 = The call is in an unrecoverable state (e.g. the service node could not allocate a resource or transfer to another node or an operator, and it had already acknowledged the Session Begin message).

Example: Ending the call as part of normal handling, bill the call.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.33 Call ID

- Description:** A unique number assigned by the switch to identify the new call.
- Length:** 4 bytes
- Format:** Binary
- Range:** 0 to 4294967295 (#0000 0000 to #FFFF FFFF)
- Definition:** This field uniquely identifies the call the message pertains to. This identifier is included in every message and associates the message with a particular call. The hexadecimal value #FFFF FFFF will be used as a NIL value. The first byte provides the low order bits of the Call ID. The fourth byte provides the high order bits.
- Example:** Call ID is 16938 (#0000 422A)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
00	0	0	0	0	0	0	0	0	1
42	0	1	0	0	0	0	1	0	2
2A	0	0	1	0	1	0	1	0	3

10.34 Call In Queue Take Down

- Description:** Specifies whether or not the switch should immediately take the call down if all subscribers are on-hook. This is only applicable when the switch is solely in control of the call (i.e. no service node or operator is connected) and the call is queued for a service node.

Length: 1 byte

Format: Binary

Range: 0 = Do Not Take Call Down

1 = Take Call Down

Definition: The following are the values for this field.

- Value 0 = The switch will not immediately take the call down if all parties go on hook while the call is in queue. However, a Queued sanity timer is initiated. If this timer expires while the call is still in queue, then the call is taken down by the switch and a log is generated. The Queued sanity timer is cleared (i.e. cancelled) if a session becomes available to serve the call. The call is then presented to the service node. The duration of the Queued sanity timer is 10 seconds. Note that a subscriber going back off-hook has no effect on the timer.
- Value 1 = The switch will take the call down if all parties go on hook while the call is in queue. If at least one party is connected (either the A party or a forward party), the switch will not take the call down. This is the default value for this field.
- If a call is queued, and all parties on that call are on facilities that require at least one party to be present (e.g. ISUP trunks), the facility will be released if all parties go on-hook. However, the call will remain in queue based on the value of this field.

Example: Take the call down.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.35 Call Origination Type

Description: Specifies the call origination type for the call.

Length: 1 byte

Format: Binary

Range: 0 = Unspecified

1 = Zero Minus (Operator Handled)

2 = Zero Plus (Operator Assisted)

3 = One Plus (Direct Dialed)
4 = Delay
5 = 555
6 = Toll Station
7 = Toll Subscriber
8 = 211
9 = 311
10 = 411
11 = 511
12 = 611
13 = 711
14 = 811
15 = 911
16 = 999
17 = HOM555
18 = FOR555
19 = INTS (Service Interception)
20 = Alarm
21 = Mobile
22 = Attended Pay Station
23 = CAMA
24 = 121
25 = 131
26 = 141
27 = 151
28 = 161
29 = 171
30 = 181
31 = 191
32 = 1150
33 = 1151
34 = 1152

35 = 1153
36 = 1154
37 = 1155
38 = 1156
39 = 1157
40 = 1158
41 = 1159
42 = 1160
43 = 1161
44 = 1162
45 = SPARE1
46 = SPARE2
47 = SPARE3
48 = SPARE4
49 = SPARE5
50 = Country Direct
51 = Special LRN

Definition: This gives information on how the call was dialled by the subscriber. The call origination type is set through translations.

- For North American dialling plan, HOM555 refers to calls dialled NPA-555-1212 within the calling subscriber's home area code. FOR555 refers to calls dialled with any other area code (i.e. a foreign area code). For non-North American dialling plan, HOM555 refers to DA calls within the calling subscriber's home area, while FOR555 refers to calls not within the calling subscriber's area (i.e. a foreign area).
- Call Origination Types 24 thru 44 are typically used for operator-dialed inward calls.
- Call Origination Types 20-44 are only valid for OSSAIN preprocessing calls.

Example: Call origination type is zero plus.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.36 Call Queue

Description: Specifies the call queue number used for an function provider request.

Length: 2 bytes

Format: Binary

Range: 0 to 8190

Description: The call queue number used for a given function provider request. Call queue numbers are unique within a particular queuing system (e.g. Call Queue 1 in one queuing system is a different queue than Call Queue 1 in another queuing system). Therefore, this field should always be used in conjunction with the Queuing System field since it is the combination of Queuing System and Call Queue that uniquely defines a call queue in the switch. Call queues for all queuing systems are defined in table QMSCQDEF (QMS Call Queue Definition table) which is indexed by a two-part key made up of Queuing System and Call Queue.

- 0 to 8190 are valid call queues.

Example: Call Queue 212 (#00D4) was used.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
D4	1	1	0	1	0	1	0	0	1

10.37 Call Type for Queuing

Description: This field identifies the call type for queuing assigned to a call.

Length: 2 bytes

Format: Binary

Range: 0 to 2047

Definition: This field identifies the call type for queuing or CT4Q assigned to a call by QMS. The CT4Q assigned is arrived at by a stepwise refinement approach which compares characteristics of the call against datafillable queuing criteria. This allows incoming traffic to be separated into different categories based on call attributes. Upon passing through all of the

refinement criteria, a given call has a CT4Q assigned. CT4Qs are defined in table CT4QNAMS.

- Values 0 to 2046 are valid CT4Q assignments.
- Value 2047 is the unspecified or nil CT4Q.

Example: CT4Q is 120 (#0078).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
78	0	1	1	1	1	0	0	0	1

10.38 Calls In Queue

Description: Number of calls currently in queue for a particular call queue.

Length: 2 bytes

Format: Binary

Range: 0 to 32767

Definition: This field specifies how many calls are currently queued in the switch in a particular call queue. The maximum number of calls that can be queued in a particular call queue is datafilled on a call queue basis in the switch table QMSCQDEF (QMS Call Queue Definition table). Once the number of calls in a particular call queue exceeds the maximum datafilled for that call queue, all additional calls that attempt to queue in that call queue will result in a call queue overflow (i.e. the calls will not queue and the service nodes making the agent requests will receive an indication of the error).

- 0 to 32766 are valid counts for the number of calls in queue.
- 32767 is a nil value. It can be received if there are no queuing elements allocated to the call queue.

Example: 4 calls are currently queued in this call queue.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
04	0	0	0	0	0	1	0	0	1

10.39 Calling Line Identifier Digit Count

Description: Specifies the number of digits in the calling line identifier number.

Length: 1 byte

Format: Binary

Range: 0 to 20

Definition: The number of digits in the Calling Line Identifier (CLI) Number field.

Example: CLI number digit count is 10.

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

10.40 Calling Line Identifier Number

Description: Specifies the calling line identifier number of the A party.

Length: 10 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: This field contains the digits that comprise the CLI number of the A party, in the case that the CLI number is different from the party's charge number. The CLI and charge number is received from SS7 signalling. The size of the field is determined by the value contained in the Calling Line Identifier Number Digit Count field.

Example: Calling Line Identifier for party is 6135553683.

7	4 3	0	Byte #
6	1	0	
3	5	1	
5	5	2	
3	6	3	
8	3	4	
F	F	5	
F	F	6	
F	F	...	
F	F	9	

10.41 Carrier Assignment Source Indicator

Description: Identifies the source of the carrier assignment for a call.

Length: 1 byte

Format: Binary

Range: 0 = Unknown
1 = Switch table TRKGRP or TOPEATRK
2 = Trunk signalling
3 = OLNS
4 = Switch table DNPIC
5 = TOPS operator
6 = OSSAIN service node
7 = Subscriber dialled

Definition: This field indicates the source for the carrier assignment.

- Value 0 = The source is unknown.
- Value 1 = The carrier assigned is the default carrier from the switch table TRKGRP or TOPEATRK. The datafill is based on incoming trunk group.

- Value 2 = The carrier was signalled to the switch from the preceding office at call arrival.
- Value 3 = The carrier assignment comes from information returned in an OLNS query.
- Value 4 = The carrier assigned is from table DNPIC. The datafill is based on the A party directory number.
- Value 5 = The carrier information was provided by a TOPS operator.
- Value 6 = The carrier information was provided by a service node.
- Value 7 = The subscriber dialled the carrier access code (e.g. 10XXX, 950) at origination.
- The precedence order is as follows:
 - TOPS operator/service node
 - subscriber dialled/trunk signalling
 - OLNS
 - table DNPIC
 - table TOPEATTRK

Example: Carrier assignment was signalled to the switch from the preceding office at call arrival.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.42 Carrier Identification Code

Description: Specifies the inter/intra LATA carrier number for the call.

Length: 2 bytes

Format: BCD

Range: 4 BCD digits, each digit ranging from 0 to 9 (0000 to 9999)

Definition: The default carrier for the call can be obtained from signalling, datafill, or the operator or service node could enter a carrier for the call.

- The carrier number is datafilled in switch Table TOPEACAR.
- The carrier number may contain leading zeros (e.g. '0595').

- If the carrier number is less than 4 digits, #F will be added to the end of the number (e.g. 123F).

Example: Carrier Identification Code is 6789.

7	4	3	0	Byte #
6	7			0
8	9			1

10.43 Carrier Identification Digit Count

Description: Specifies the number of digits in the carrier identification code.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: This field specifies the number of digits contained in the Carrier Identification Code field.

Example: Carrier identification digit count is 3.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.44 Carrier Name

Description: Specifies the inter/intra LATA carrier name for the call.

Length: 8 bytes

Format: ASCII

Range: 8 alphanumeric characters in the range {A-Z, a-z, 0-9}

Definition: This carrier name associated with the carrier identification code in switch table TOPEACAR.

- When this field is sent by the service node to the switch, the switch will ignore this field.

Example: Carrier name is "CARR11".

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
C	43	0	1	0	0	0	0	1	1	0
A	41	0	1	0	0	0	0	0	1	1
R	52	0	1	0	1	0	0	1	0	2
R	52	0	1	0	1	0	0	1	0	3
1	31	0	0	1	1	0	0	0	1	4
1	31	0	0	1	1	0	0	0	1	5
nil	FF	1	1	1	1	1	1	1	1	6
nil	FF	1	1	1	1	1	1	1	1	7

10.45 Carrier Name Length

Description: Specifies the number of characters in the Carrier Name field.

Length: 1 byte

Format: Binary

Range: 0 to 8

Definition: This field specifies the number of characters in the Carrier Name field.

Example: Length is 6.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0

10.46 Carrier Number Status

Description: Specifies the validity of the InterLATA Carrier number of the call.

Length: 1 byte

Format: binary

Range: 0 = None

1 = Valid

2 = Invalid

3 = Invalid for this call only

Definition: The following are the values for this field:

- Value 0 = None, indicates there is no carrier associated with the call.

Value 1 = Valid, indicates the carrier number is valid and can be used for the call.

Value 2 = Invalid, indicates the carrier number is not valid.

Value 3 = Invalid for this call only, indicates the carrier is a valid carrier in the office but it is not allowed to complete this call.

Example: Carrier number is valid.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.47 Carrier Service Status

Description: Specifies the validity of a service node action with regard to the type of service provided for a carrier.

Length: 1 byte

Format: Binary

Range: 0 = Valid

1 = Invalid

Definition: Value 0 = Valid, indicates the action complies with the type of service being provided by the operating company on behalf of the carrier.

Value 1 = Invalid, indicates the action does not comply with the type of service being provided by the operating company on behalf of the carrier.

Example: Carrier service status is valid

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.48 Class Charge Selector

Description: Specifies if this is the current or previous class charge information.

Length: 1 byte

Format: Binary

Range: 0 = Current class charge
1 = Previous class charge

Definition: This field indicates if the class charge is current for the call or was the previous class charge for the call.

- Value 0 = This is the current class charge for the call
- Value 1 = This is the previous class charge for the call. The previous class charge information is sent only by the switch. It is sent if the call was RLT'd and the Session Origination Type field is set to Reorigination.

Example: This is the previous class charge information.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.49 Coin Key Pad State

Description: Specifies the state of the DTMF key pad for the A party.

Length: 1 byte

Format:	Binary
Range:	0 = Undetermined 1 = Enabled 2 = Disabled
Definition:	This field indicates the DTMF key pad state. If the phone is not a coin phone, then the value is set to Undetermined.
Example:	The pad state is enabled.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.50 Coin Overtime Type

Description:	Specifies whether overtime charges are to be collected before or after the overtime period for a coin call.
Length:	1 byte
Format:	Binary
Range:	0 = Not Applicable 1 = postpay overtime 2 = prepay overtime
Definition:	The value for this field may be obtained through switch datafill if the originating trunk is datafilled in table TOPSCOIN. The coin overtime type field specifies whether it is postpay or prepay. If the originating trunk is not datafilled in table TOPSCOIN, then the value is set to postpay overtime. • Value 0 = The call is not from a coin phone. • Value 1 = Charges for the overtime period are to be collected after the overtime period. • Value 1 = Charges for the overtime period are to be collected before the overtime period.
Example:	Overtime type is prepay overtime.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.51 Conference ID

Description: Identifies a particular conference.

Length: 4 bytes

Format: Binary

Range: 0 to 4294967295 (#0000 0000 to #FFFF FFFF)

Definition: This field uniquely identifies a conference. The first byte provides the low order bits of the conference ID. The fourth byte provides the high order bits.

Example: Conference ID is 23692 (#0000 5C8C)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
00	0	0	0	0	0	0	0	0	1
5C	0	1	0	1	1	1	0	0	2
8C	1	0	0	0	1	1	0	0	3

10.52 Connection Event

Description: Specifies the change that has occurred in a connection.

Length: 1 byte

Format: Binary

Range: 0 = On-hook
1 = Off-hook
2 = Hook flash

3 = Released

Definition: This field specifies the change in a connection that has occurred on a call controlled by a node. The information is sent to the node to indicate the event. Released occurs where all parties go on-hook on a trunk type (e.g. ISUP) that auto-releases, independent of the switch.

Example: On-hook.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.53 Connection ID

Description: Specifies a connection/party type. Used to associate directory numbers, speech path status, connect/disconnect requests, events, etc., with a specific connection.

Length: 1 byte

Format: Binary

Range: 0 = Alt DN Connection
1 = A Party Connection
2 = B Party Connection

Definition: The ID of the specified party/DN associated with the data block. Note: The Alt DN Connection and B Party Connection must not be requested concurrently (i.e. one party or the other must be disconnected first).

Example: A Party Connection.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.54 Connection Status

Description: Specifies the connection status of the connected party.

Length: 1 byte

Format: Binary

Range: 0 = Not Connected
1 = Connected, Off Hook
2 = Connected, On Hook
3 = Connection is treatment

Definition: The following are the values for this field:

- Value 0 = The party is not connected. The speech path status is set to zero-way.
- Value 1 = The party is connected but off hook. The speech path status field indicates the type of connection.
- Value 2 = The party is connected and on hook. The speech path status field indicates the type of connection.
- Value 3 = A treatment is being applied to this connection. Other parties connected in a one-way, listen only or two-way speech path status will be able to hear the treatment being applied.

Example: Value 1 = The party is connected and off hook.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.55 Context Block Data

Description: A block of data pertaining to the call.

Length: 50 bytes

Format: Unknown to switch

Range: #0 - #FF

Definition: This field includes a block of data pertaining to the call. This field will either contain the data or specify a pointer to where the data is stored.

- The format of the context block data area is stored in a field within the context block data. The first field in the context block is the Context Type. The value of this field indicates the structure of the data area. Refer to the TOPS Context Block document for more information (NIS Q258-1).

Example: A 50 byte block with data whose format is unknown to the switch.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
	1	0	0	0	1	0	0	0	0
	0	1	0	1	0	1	0	1	1
	0	0	1	0	0	0	1	0	2
	0	0	0	1	0	0	0	1	3
	1	0	1	0	1	0	1	0	:
	0	1	0	0	0	1	0	0	:
	0	0	0	0	0	0	0	0	:
	1	0	0	0	1	0	0	1	48
	0	0	1	0	0	0	1	0	49

10.56 Context Block Length

Description: The length of the Context Block Data.

Length: 1 byte

Format: Binary

Range: 0 to 50

Definition: This field contains the length for the Context Block Data field.

Example: Length is 20.

7	6	5	4	3	2	1	0
0	0	0	1	0	1	0	0

10.57 Control List ID

Description: Identifies the control list.

Length: 2 bytes

Format: Binary

Range: 0 to 4095

Definition: A control list is a list of functions that OSSAIN applies sequentially to the call. This is the index to table OACTLDEF, which contains the list of all allowable control lists.

- Values 0 to 4094 are valid control list indices.
- Value 4095 is the nil control list index.

Example: Control list index of 212 (#00D4).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
D4	1	1	0	1	0	1	0	0	1

10.58 Conversation Timing Status

Description: Specifies whether the conversation time recording for the call be started or stopped.

Length: 1 byte

Format: Binary

Range: 0 = Stop Call Timing

1 = Start Call Timing

Definition: If the field contains 1, the billable conversation time for the call is started. If the field is 0, billable conversation time accumulation for the call is stopped.

Example: Start call timing

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.59 Country Direct Access Code Count

Description: Specifies the number of digits in the Country Direct Access Code field.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: This field contains the number of digits in the Country Direct Access Code field.

Example: Access Code digit count is 3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.60 Country Direct Access Code Number

Description: The access code used for the country direct call.

Length: 2 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: This field contains the Country Code Access Code Number. This number is

signalled in from the gateway switch. The switch does not manipulate the number in any way. The size of the field is determined by the value contained in the Country Direct Access Code count field.

Example: Access Code number is 800

7	4	3	0	Byte #
8	0			0
0	F			1

10.61 Country Direct Carrier Code Count

Description: Specifies the number of digits in Country Direct Carrier Code Number field.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: This field contains the number of digits in the Country Direct Carrier Code field.

Example: Carrier Code digit count is 4

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.62 Country Direct Carrier Code Number

Description: The carrier code number used for the country direct call.

Length: 2 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: This field contains the Country Code Carrier Code Number. This number is signalled in from the gateway switch. The switch does not manipulate the number in any way. The size of the field is determined by the value contained in the Country Direct Carrier Code count field.

Example: Carrier Code number is 0002

7	4	3	0	Byte #
0	0			0
0	2			1

10.63 Country Direct Country Code Count

Description: Specifies the number of digits in Country Direct Country Code Number field.

Length: 1 byte

Format: Binary

Range: 0 to 3

Definition: This field contains the number of digits in the Country Direct Country Code field.

Example: Country Code digit count is 3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.64 Country Direct Country Code Number

Description: The country code digits used for the country direct call.

Length: 2 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: This field contains the Country Code Country Code Number. This number is signalled in from the gateway switch. The switch does not manipulate the number is any way. The size of the field is determined by the value contained in the Country Direct Country Code count field.

Example: Country Code number is 123

7	4	3	0	Byte #
1	2			0
3	F			1

10.65 Current Broadcast Connections

Description: Indicates the current number of broadcast connections existing on a voice link.

Length: 2 bytes

Format: Binary

Range: 0 to 1023

Definition: The number of connections currently made to the broadcast voice link.

Example: There are 20 broadcast connections on the voice link.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	1	0	0
D4	0	0	0	1	0	1	0	0	1

10.66 DB Byte Count

Description: Specifies the number of bytes to follow in a data block.

Length: 2 bytes

Format: Binary

Range: 0 to 65535

Definition: This field indicates the number of bytes contained in a data block. The count does not include the data block identifier (3 bytes) or the DB Byte Count field itself. It indicates the number of bytes following this field.

Example: DB byte count is 5.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1	1

10.67 DTMF Digit

Description: The digit received by the DTMF receiver.

Length: 1 byte

Format: Binary

Range: 0 to 9 = digits 0 to 9

10 = '*'

11 = '#'

Definition: The field contains up to one digit received by the DTMF receiver.

Example: Party entered an asterisk (10, #A)

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

10.68 DTMF Receiver

Description: Specifies if the DTMF receiver is still connected.

Length: 1 byte

Format: Binary

Range: 0 = Not Connected
1 = Connected

Definition: This field specifies if the DTMF receiver is still connected to the call.

Example: DTMF receiver is connected.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.69 Data Block ID

Description: Uniquely identifies a data block.

Length: 2 bytes

Format: Binary

Range: Hex value: 0000 to FFFF

Definition: Uniquely identifies an OAP data block.

- Identifiers for OAP data blocks defined in this document range from #0000 to #DFFF, inclusive. Identifiers ranging from #E000 to #FFFF are reserved for external use (e.g. service node to service node use).

Example: Data block ID is #016C.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	1	0
0	1	1	0	1	1	0	0	1

10.70 Day

Description: Specifies the current day of the month.

Length: 1 byte

Format: Binary

Range: 1 to 31

Definition: The day of the month.

Example: Day is 22nd.

7	6	5	4	3	2	1	0
0	0	0	1	0	1	1	0

10.71 Directory Number

Description: Specifies the directory number of the specified party.

Length: 10 bytes (Maximum of 20 digits.)

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: This field contains the digits that comprise the directory number of the specified party. The size of the field is determined by the value contained in the Directory Number Digit Count field.

- Directory numbers sent to the switch should not include the long distance access (e.g. '1') or international access prefix digits for the A or B parties.
- Foreign alternate DNs sent to the switch require the "011" international access prefix digits.

Example: Directory Number for party is 6135553684.

7	4	3	0	Byte #
6	1			0
3	5			1

7	4	3	0	Byte #
5	5			2
3	6			3
8	4			4
F	F			5
F	F			6
F	F			...
F	F			9

10.72 Directory Number Digit Count

Description: Specifies the number of digits in the directory number.

Length: 1 byte

Format: Binary

Range: 0 to 20

Definition: The number of digits in the Directory Number field. Actual switch limits for this value may be less, depending on call type, numbering plan, and party.

Example: Directory number digit count is 10.

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

10.73 Directory Number Type

Description: Specifies the type of directory number associated with the message.

Length: 1 byte

Format: Binary

Range: 0 = Foreign

1 = Domestic Non-local

2 = Domestic Local

3 = Outgoing Trunk

4 = Service

254 = Unknown

Definition: The number type of the specified party in the data block.

- Value 0 = Foreign is used for all calls not within in the A party's dialing plan. The Directory Number field includes country code digits.
- Value 1 = Domestic non-local number type is used for all non-local calls within the A party's dialing plan. The Directory Number field does not include country code digits.
- Value 2 = Domestic local number type is used for all calls that are local to the A party's dialing plan.
- Value 3 = Outgoing Trunk (OGT) keyed by an operator.
- Value 4 = Calls dialed to a service agency (e.g. 411, 611).

Example: A number type of Domestic Local.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.74 Elapsed Minutes

Description: Specifies the current elapsed conversation time in minutes.

Length: 3 bytes

Format: Binary

Range: 0 to 99,999

Definition: The duration in the number of minutes.

Example: Elapsed minutes is 68090 (#0109FA)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
01	0	0	0	0	0	0	0	1	0
09	0	0	0	0	1	0	0	1	1
FA	1	1	1	1	1	0	1	0	2

10.75 Error ID

Description: Identifies a unique error value.

Length: 2 bytes

Format: Binary

Range: Hex value: 0100 to FFFF

Definition: Uniquely identifies an OAP error value. The OAP error values are listed in Chapter 11: "Error ID Values".

10.76 Estimated Charges

Description: Specifies the charges for a call.

Length: 4 bytes

Format: Binary

Range: 0 to 4,294,967,295

Definition: This field specifies the estimated charges for a call. The charges are in generic units. They are not specific to any currency.

Example: A charge of five hundred units (#0000 #01F4)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
00	0	0	0	0	0	0	0	0	1
01	0	0	0	0	0	0	0	1	2
F4	1	1	1	1	0	1	0	0	3

10.77 Estimate Duration

Description: Specifies the length (in minutes) to be used when estimating charges for a call.

Length: 2 bytes

Format: Binary

Range: 1 to 65400

Definition: This field specifies the duration of a call used when computing an estimate of charges.

- Values 1 to 999 are valid minutes that can be used for duration estimates of calls using North American Numbering.
- Values 1 to 65400 are valid minutes that can be used for duration estimates of calls using Open Numbering.

Example: A duration of 14 minutes.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
0E	0	0	0	0	1	1	1	0	1

10.78 Float Connection Indicator

- Description:** Specifies whether the B Party is to be connected upon call float.
- Length:** 1 byte
- Format:** Binary
- Range:** 0 = Unconnected or Connected
1 = Talking
- Definition:** Ensures that the switch properly interprets whether the B Party is to be connected upon float.
- Value 0: The parties are not connected or the B party is connected but has not answered.
 - Value 1: The parties are connected and talking.
- Example:** Parties are talking. No attempt will be made to (re)connect the B Party.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.79 Function ID

- Description:** Specifies the function being referenced in the message.
- Length:** 2 bytes
- Format:** Binary
- Range:** 0 to 1023
- Definition:** A function represents a service or a portion of a service provided by either a service node, a TOPS operator, or an existing TOPS automated system. Table OAFUNDEF contains the definitions of all allowable functions.
- 0 to 1022 are valid Function IDs.
 - 1023 is the nil Function ID.
- Example:** Function is 54 (#0036).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
36	0	0	1	1	0	1	1	0	1

10.80 Function Provider Request Status

Description: Specifies the status of a request by a service node to connect another function provider to the call.

Length: 1 byte

Format: Binary

Range: 0 = Function provider connected to call
1 = Call queued for function provider
2 = Call deflected
3 = Call queue overflow
4 = Queuing System overflow

Definition: The status of a function provider request by a service node.

- Value 0 = A function provider has been connected to the call.
- Value 1 = The call has been queued for a function provider because no function providers were immediately available and the Predicted Wait Time for the call did not exceed the maximum allowable Predicted Wait Time datafilled in the switch.
- Value 2 = No function provider was available to serve the call and the Predicted Wait Time exceeded the maximum allowable Predicted Wait Time datafilled in the switch. Therefore, the call was not queued and was instead deflected from the requested call queue.
- Value 3 = No function provider was available to serve the call and the number of calls in the requested call queue exceeded the maximum allowed for that call queue datafilled in switch table QMSCQDEF (QMS Call Queue Definition table). Therefore, the call could not be queued and call queue overflow occurred.
- Value 4 = No function provider was available to serve the call and the number of calls queued in all call queues associated with the Queuing System exceeded the maximum datafilled for that Queuing System data-

filled in switch table QAPLNDEF (QMS Application Definition table). Therefore, the call could not be queued and Queuing System overflow occurred.

Example: A function provider was connected to the call.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.81 Generate AMA Sequence Number

Description: Specifies the number of times a service node has requested an AMA record be generated. This number is used by the switch to avoid duplicate or lost billing. Sequential requests must have an increasing sequence number or the switch will not process them, but will instead return an error.

Length: 1 byte

Format: Binary

Range: 0 to 254

Definition: A sequence number field specifying the number of times a generate AMA has been requested.

Example: Sequence number is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.82 Handling Data

Description: Specifies the class charge handling type for the call.

Length: 1 byte

Format: Binary

Range: 0 = Station

1 = Person

2 = Auto

254 = Unspecified

Definition: The class charge handling type for the call. The Billing Data field contains the actual class charge billing data.

- Value 0 = A station to station call. The calling subscriber is not requesting a specific person to answer the forward number.
- Value 1 = A person to person call. The calling subscriber requests to talk to a specific person. The operator (or service node) connects the forward party and remains on the line until the person answers the call.
- Value 2 = A call can be completed to the appropriate party without first contacting the party for verification. This is a special type of handling. An example of its use is 800 toll-free calls which are always paid for by the 800 number.
- Value 254 = The class charge is unspecified.

Example: Class charge handling is person.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.83 Hotel Room Number

Description: Specifies the room number.

Length: 6 bytes

Format: ASCII

Range: {0 to 9}, {A to Z}

Definition: The hotel room number. This field may be required if the phone class type is Hotel/Motel or Hotel/Motel With No Charge. If there is no hotel room number, then the field value is set to the nil value (#FFF).

- Up to six characters can be supplied. Numerics are treated as one character each (half a byte of store), and the upper case alphas are treated as two characters each (one byte of store).
- “123456”, “W1212”, “WW12”, and “WWW” are acceptable combinations (in any order or combination).

- “WWWW”, “WWW1”, and “WW123” are not acceptable combinations because they exceed the six character limitations.

Example: Hotel room number is AB12.

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
A	41	0	1	0	0	0	0	0	1	0
B	42	0	1	0	0	0	0	1	0	1
1	31	0	0	1	1	0	0	0	1	2
2	32	0	0	1	1	0	0	1	0	3
nil	FF	1	1	1	1	1	1	1	1	4
nil	FF	1	1	1	1	1	1	1	1	5

10.84 Hotel Room Number Length

Description: Specifies the number of characters in the Hotel Room Number field.

Length: 1 byte

Format: Binary

Range: 0 to 6

Definition: This field specifies the number of characters in the Hotel Room Number Field.

Example: Length is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.85 Hour

Description: Specifies the current hour.

Length: 1 byte
Format: Binary
Range: 0 to 23
Definition: The hour of the day.
Example: Hour is 12 (#0C).

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0

10.86 Invalid Billing Set

Description: Indicates which billing details are invalid.

Length: 2 bytes

Format: Set

Range: Bit 0 = Invalid or Missing Class Charge
Bit 1 = Invalid or Missing A Party DN
Bit 2 = Invalid or Missing B Party DN
Bit 3 = Invalid or Missing Billing Number

Definition: This field indicates which billing details are invalid.

- A bit set to one indicates that the information associated with that bit is invalid or missing. A bit set to zero indicates the information associated with the bit is valid.
- Bits 4 - 15 are currently unused and will have a value of zero.

Example: The class charge is missing.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1

10.87 Invoke ID

- Description:** Specifies the solicitor number used for an operation.
- Length:** 2 bytes
- Format:** Binary
- Range:** Hex value: 0100 to FFFF
- Definition:** A solicitor identification number used to map particular response messages to specific operations.
- Session Pool Class, Session Pool Alarm Class, Session Pool Log Class: the range of invoke ids for service node originated requests/informs is #0100 to #01FF.
- Example:** Invoke id is #02C8

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	1

10.88 LATA Restrictions

- Description:** This field specifies if there are any restrictions with the A party and equal access.
- Length:** 1 byte
- Format:** Binary
- Range:** 0 = No Restrictions
1 = Non-Conforming End Office
2 = All InterLATA is Restricted
- Definition:** Equal access provides the ability for a subscriber to have a presubscribed carrier or to override their presubscribed carrier by dialling 10XXX. An end office that offers equal access is known as a conforming or equal access end office. An end office that does not offer equal access is known as a non-conforming end office.
- Value 0 = The call originated from an equal access end office. No restrictions apply to equal access processing.

- Value 1 = The call originated from a non-conforming end office. The end office does not provide equal access. The carrier may be changed for the call. However, subscribers who do not have a choice at the end office should not be allowed to have a choice when the call is at the service node.
- Value 2 = The call originated from an equal access end office, but the subscriber is not allowed to complete interLATA calls.

Example: All interlata is restricted.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.89 LATA Status

Description: Specifies if the call is an intraLATA or interLATA call.

Length: 1 byte

Format: Binary

Range: 0 = IntraLATA
1 = InterLATA

Definition: Indicates if the call is intralata or interLATA. This information is known when the switch has the A party directory number and the B party directory number. This field defaults to intraLATA if the B party directory number is not present.

Example: Intralata call

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.90 LNP Query

Description: Specify whether or not to override the LNP query launched by default for certain OAP operation

Length: 1 byte

Format: Binary

Range: 0 = Perform LNP Query
1 = Skip LNP Query

Definition:

- Value 0 = The service node has specified that an LNP query can be launched as is normally done for this operation.
- Value 1 = The service node has requested that no LNP query is performed for this operation.

Example: A LNP query should be performed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.91 LRN

Description: Specifies the location routing number of the specified directory number.

Length: 5 bytes

Format: BCD

Range: BCD digits, ranging from 0 to 9

Definition: The LRN of the specified DN.

Example: LRN is 2012201111

7	4	3	0	Byte #
2	0			0
1	2			1
2	0			2
1	1			3
1	1			4

10.92 LRN Digit Count

Description: Specifies the number of digits in the LRN field.

Length: 1 byte

Format: Binary

Range: 0 to 10

Definition: The number of digits in the LRN field.

Example: LRN digit count is 10

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

10.93 Language

Description: Specifies what languages are in effect for a party on a call.

Length: 1 byte

Format: Binary

Range: 0 to 99

254 = Unknown

Definition: The language that applies to the specified party. The language indexes into DMS Table TOPSLANG.

Example: Language is 12.

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0

10.94 Logical Voice Channel ID

Description: Specifies the logical voice channel number for the voice connection between the switch and the service node.

Length: 2 bytes
 Format: Binary
 Range: 0 to 8192
 Definition: This field specifies the voice channel that is selected for use during this session. The voice channels are data filled on a per node basis in switch Table OAVLMAP. Voice channel 8192 is the NIL voice channel ID.
 Example: Voice Channel ID is 725 (#02D5)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
02	0	0	0	0	0	0	1	0	0
D5	1	1	0	1	0	1	0	1	1

10.95 Maintenance State

Description: Specifies the state of the node or session pool.
 Length: 1 Byte
 Format: Binary
 Range: 0 = Out of Service
 1 = In Service
 Definition: State of the node or session pool specified in the message.

- Value 0: The state is out of service and no call processing messaging can occur with the switch.
- Value 1: The state is in service.

 Example: State is in service

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.96 Maintenance Text

Description: Used to send messages to be displayed at the MAP or logged.

Length: 32 bytes

Format: ASCII

Range: A-Z, a-z, 0-9, space

Definition: Alpha-numeric message to be displayed at the MAP or logged.

Example: Message to tell maintenance individual to “Call Service”.

ASCII	Hex	7	6	5	4	3	2	1	0	Byte #
C	43	0	1	0	0	0	0	1	1	0
a	61	0	1	1	0	0	0	0	1	1
l	6C	0	1	1	0	1	1	0	0	2
l	6C	0	1	1	0	1	1	0	0	3
	20	0	0	1	0	0	0	0	0	4
S	53	0	1	0	1	0	0	1	1	5
e	65	0	1	1	0	0	1	0	1	6
r	72	0	1	1	1	0	0	1	0	7
v	76	0	1	1	1	0	1	1	0	8
i	69	0	1	1	0	1	0	0	1	9
c	63	0	1	1	0	0	0	1	1	10
e	65	0	1	1	0	0	1	0	1	11
nil	#FF	1	1	1	1	1	1	1	1	...
nil	#FF	1	1	1	1	1	1	1	1	31

10.97 Maintenance Text Length

Description: Indicates the length of the maintenance text string.

Length: 1 byte

Format: Binary
Range: 0 to 32
Definition: Indicates the number of characters in the maintenance text string.
Example: From above maintenance text, length is 12 bytes

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	0

10.98 Minutes

Description: Specifies the current minutes.
Length: 1 byte
Format: Binary
Range: 0 to 59
Definition: The minutes of the hour.
Example: Minutes is 39 (#27)

7	6	5	4	3	2	1	0
0	0	1	0	0	1	1	1

10.99 MIS Sequence Number ID

Description: This field provides a numerical sequence identifier for session and call queue event DBs used in the MIS data stream.
Length: 1 byte
Format: Binary
Range: 0 to 255
Definition: The DMS maintains a sequence count for each queue and session. The count is incremented each time a queue event or session event is composed for that queue or session. After a restart, the sequence number may begin at

any value.

Example: Third sequence number.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.100 Month

Description: Specifies the current month.

Length: 1 byte

Format: Binary

Range: 1 to 12

Definition: The month of the year.

Example: Month is February (2).

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.101 NBEC Code

Description: Specifies the Non-Bell Exchange Carrier the call.

Length: 2 bytes

Format: BCD

Range: up to 4 BCD digits, each digit ranging from 0 to 9 (0000 to 9999)

Definition: This field specifies the NBEC code. The NBEC code is assigned on a per trunk group basis or per calling directory number basis (from table NBECCODE).

- If the NBEC code is less than 4 digits, #F is padded at the end (e.g. 123F).

Example: NBEC code is 1592.

7	4	3	0	Byte #
1	5			0
9	2			1

10.102 NBEC Code Digit Count

Description: Specifies the number of digits in the NBEC code.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: This field specifies the number of digits contained in the NBEC Code field.

Example: Carrier identification digit count is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.103 Network ID

Description: This identifies the MIS network.

Length: 2 bytes

Format: Binary

Range: 0 to 32766

Definition: This is the MIS network ID. The value is obtained from table QMSMIS.

Example: Network ID is 200 (#00C8).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
C8	1	1	0	0	1	0	0	0	1

10.104 Network Service ID

Description: Specifies the service identifier corresponding to the service being provided during a call.

Length: 2 bytes

Format: Binary

Range: 0 to 32766

Definition: The identifier is retrieved by the switch from switch table OACTLDEF, based upon the control list identifier in effect when the call is presented to a function provider; or is provided by a service node at Session Initiation. Each time an AMA record is produced for a call, the current Network Service ID is placed in it. With future development, this identifier may be used on a network-wide basis to drive real time rating of calls, and to allow restrictions to be placed upon which types of services a particular subscriber is eligible to receive.

- For OSSAIN preprocessing calls, the Network Service ID comes from switch table OAINPRE.
- For calls that trigger to OSSAIN without obtaining a Network Service ID previously, the max Network Service ID (32766) is used. For example, if a call that is floated by an operator then subsequently triggers to an OSSAIN service node via an OSSAIN function would not have obtained a Network Service ID from table OACTLDEF, so the max Network Service ID is used.

Example: Network Service ID is 500 (#01F4).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
01	0	0	0	0	0	0	0	1	0

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
F4	1	1	1	1	0	1	0	0	1

10.105 No Answer Time

Description: Specifies the amount of time, in seconds, before reporting the no answer trigger.

Length: 1 byte

Format: Binary

Range: 0 to 120

Definition: Indicates the no answer time value.

- Value 0 = No timer specified. Use Switch default value or the timer specified by previous function provider.
- Values 1-120 = No answer time of 1-120 seconds.

Example: Report no answer in 8 seconds.

7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0

10.106 Node ID

Description: Uniquely identifies an OSSAIN node in the network.

Length: 2 bytes

Format: Binary

Range: 0 to 65535

Definition: This field identifies an OSSAIN node. Table OANODINV contains a list of all valid node identifiers.

- 0 to 767 are valid node ids that can be datafilled in Table OANODINV.
- 65535 is the nil node id.

Example: Node ID is 250 (#00FA).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
FA	1	1	1	1	1	0	1	0	1

10.107 Notification Time Type

Description: Identifies the basis for starting a trigger notification timer.

Length: 1 byte

Format: Binary

Range: 0 = Relative Time
1 = Conversation Time

Definition: Indicates the point of reference from which to start the notification timer.

- Value 0 = 'Relative Time' indicates the call triggers back to the SN after call float or answer, whichever occurs last, for the duration of the notification timer specified.
- Values 1 = 'Conversation Time' indicates the call triggers back to the SN after call float when the call's elapsed conversation time equals the notification time specified.

Example: Conversation Time

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.108 Notification Timer Action

Description: Specifies what should be done with the notification timer.

Length: 1 byte

Format: Binary

- Range: 0 = Start notification timer
1 = Stop notification timer
- Definition: This field indicates whether the notification timer should be started or stopped.
- If the timer is to be stopped and it is already running, the previous timer will be stopped and the timer will be restarted.
- Example: Stop the notification timer

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.109 Notification Timer Mode

- Description: Specifies whether the notification timer is “float-style” or “persistent-style”.
- Length: 1 byte
- Format: Binary
- Range: 0 = Float-style notification timer
1 = Persistent-style notification timer
2 = Do not change timer
- Definition: This field indicates which mode of operation should be applied to the notification timer.
- Value 0 = This value means the notification timer can only run while the call is floated. If the call triggers, the float-style notification timer is stopped.
 - Value 1 = This value means that the notification timer can run while the call has a session with an SN, as well as when the call is floated. When using this mode of notification timer, the timer will continue to run until it expires or until the SN explicitly stops it. It will continue to run even when the call triggers or is transferred to another SN.
 - Value 2 = This value means that the notification time specified in the message must not affect the current times (the notification timer will be unaffected).
- Example: Float-style notification timer

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.110 Notification Timer Status

Description: Specifies whether or not the notification timer is currently running.

Length: 1 byte

Format: Binary

Range: 0 = Notification timer running
1 = Notification timer not running

Definition: This field indicates whether or not the notification timer is running.

Example: The notification timer is running

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.111 Number of Operator to SN Handoffs

Description: Indicates the number of Operator to SN handoffs allowed for the call.

Length: 1 byte

Format: Binary

Range: 0 to 10
254 = Nil

Definition: This field is used to indicate the number of Operator to SN handoffs allowed for this call.

- Value 0 to 10 = The number of successful Operator to SN handoffs allowed is less than or equal to this value.
- Value 254 = The nil value. There is no limit to the number of Operator to SN handoffs. However, the number of transitions is limited by standard OSSAIN transition processing.

Example: 2 Operator to SN Handoffs allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.112 OLNS Account Owner

Description: The account owner associated with the calling number which is returned for the OLNS LIDB.

Length: 4 bytes

Format: ASCII

Range: All ASCII characters are supported.

Definition: The account owner

Example: '2A45'

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
2	32	0	0	1	1	0	0	1	0	0
A	41	0	1	0	0	0	0	0	1	1
4	34	0	0	1	1	0	1	0	0	2
5	35	0	0	1	1	0	1	0	1	3

10.113 OLNS Account Owner Length

Description: The length of the account owner returned by the OLNS LIDB.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: The number of characters in the OLNS account owner field.

Example: The length is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.114 OLNS Alphanumeric String

Description: The alphanumeric string associated with calling number which is returned the from the OLNS LIDB.

Length: 8 bytes

Format: ASCII

Range: All ASCII characters are supported.

Definition: The alphanumeric string.

Example: '33M12'

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
3	33	0	0	1	1	0	0	1	1	0
3	33	0	0	1	1	0	0	1	1	1
M	4D	0	1	0	0	1	1	0	1	2
1	31	0	0	1	1	0	0	0	1	3
2	32	0	0	1	1	0	0	1	0	4
nil	#FF	1	1	1	1	1	1	1	1	5
nil	#FF	1	1	1	1	1	1	1	1	6
nil	#FF	1	1	1	1	1	1	1	1	7

10.115 OLNS Alphanumeric String Length

Description: The length of the alphanumeric string returned by the OLNS database.

Length: 1 byte
 Format: Binary
 Range: 0 to 8
 Definition: The number of characters in the OLNS alphanumeric string.
 Example: The length is 5.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1

10.116 OLNS Billing Service Provider

Description: The billing service provider associated with the calling number which is returned from the OLNS LIDB.
 Length: 4 bytes
 Format: ASCII
 Range: All ASCII characters are supported.
 Definition: The billing service provider.
 Example: '2BE5'

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
2	32	0	0	1	1	0	0	1	0	0
B	42	0	1	0	0	0	0	1	0	1
E	45	0	1	0	0	0	1	0	1	2
5	35	0	0	1	1	0	1	0	1	3

10.117 OLNS Billing Service Provider Length

Description: The length of the billing service provider returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: The number of characters in the OLNS billing service provider field.

Example: The length is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.118 OLNS DA Credit Card Indicator

Description: The DA calling card billing restrictions for the calling number returned from the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed from this line
2 = Not allowed from this line
254 = Unknown

Definition: The DA calling card billing restrictions.

Example: DA calling card billing is allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.119 OLNS DA Sent Paid Indicator

Description: The DA sent paid billing restrictions for the calling number returned from the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed from this line
2 = Not allowed from this line
254 = Unknown

Definition: The DA sent paid billing restrictions.

Example: DA sent paid billing is not allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.120 OLNS DA Special BNS Indicator

Description: The DA special BNS billing restrictions for the calling number returned from the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed from this line
2 = Not allowed from this line
254 = Unknown

Definition: The DA special BNS billing restrictions.

Example: DA special BNS billing is allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.121 OLNS DA Third Number Indicator

Description: The DA third number billing restrictions for the calling number returned

from the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed from this line
2 = Not allowed from this line
3 = Allowed with operator verification
4 = Allowed with operator or automated verification
254 = Unknown

Definition: The DA third number billing restrictions.

Example: DA third number billing is allowed with operator or automated verification.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.122 OLNS DACC Indicator

Description: The DACC restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Not Allowed
3 = Allowed with Billing Restrictions
4 = Allowed for Local, Non-Toll Calls
5 = Allowed with Billing Restrictions, No Sent Paid
254 = Unknown

Definition: The DACC restrictions.

- Value 3 indicates that the other TA billing restrictions (TA collect, third, etc.) define the allowed billing types for DACC.

- Value 4 indicates that DACC is only allowed for calls which are local and free.
- Value 5 indicates that the other TA billing restrictions (TA collect, third, etc.) define the allowed billing types for DACC. Additionally, sent paid billing is not allowed for DACC.

Example: DACC is not allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.123 OLNS Foreign Language

Description: The foreign language of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range:

- 0 = Screened
- 1 = Spanish
- 2 = French
- 3 = German
- 4 = Italian
- 5 = Mandarin
- 6 = Tagalog
- 7 = Polish
- 8 = Korean
- 9 = Vietnamese
- 10 = Portuguese
- 11 = Japanese
- 12 = Greek
- 13 = Arabic
- 14 = Hindi (Urdu)
- 15 = Russian
- 16 = Yiddish

17 = Thai (Laotian)
18 = Persian
19 = French Creole
20 = Armenian
21 = Navaho
22 = Hungarian
23 = Hebrew
24 = Dutch
25 = Mon-Khmer (Cambodian)
26 = Gujarathi
27 = Ukrainian
28 = Czech
29 = Pennsylvania Dutch
30 = Miao (Hmong)
31 = Norwegian
32 = Slovak
33 = Swedish
34 = Serbo-Croatian
35 = Kru
36 = Rumanian
37 = Lithuanian
38 = Finnish
39 = Panjabi
40 = Formosan
41 = Croatian
42 = Turkish
43 = Ilocano
44 = Bengali
45 = Danish
46 = Syriac
47 = Samoan
48 = Malayalam

49 = Cajun

50 = Amharic

51 = Cantonese

254 = Unknown

Definition: The foreign language of the calling party. If the value is unknown, English should be assumed.

Example: The language is Cantonese.

7	6	5	4	3	2	1	0
0	0	1	1	0	0	1	1

10.124 OLNS InterLATA Carrier Indicator

Description: The interLATA carrier indicator of the calling party returned by the OLNS LIDB.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = IC Not Indicated
2 = IC Indicated
3 = IC Denied
254 = Unknown

Definition: The interLATA carrier indicator of the calling party.
— This field is set to either not indicated or indicated. The other values are not currently used.

Example: The interLATA carrier is indicated.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.125 OLNS InterLATA Carrier Identification Digit Count

Description: The number of digits in the interLATA carrier code of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: The number of digits in the interLATA carrier code.

Example: The number of digits is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.126 OLNS InterLATA Carrier Identification Code

Description: The interLATA carrier identification code of the calling party returned by the OLNS database.

Length: 2 bytes

Format: BCD

Range: 4 BCD digits, each digit ranging from 0 to 9 (0000 to 9999)

Definition: The interLATA carrier identification code.

Example: 6789.

7	4	3	0	Byte #
6	7			0
8	9			1

10.127 OLNS International Carrier Indicator

Description: The international carrier indicator of the calling party returned by the OLNS LIDB.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = IC Not Indicated
2 = IC Indicated
3 = IC Denied
254 = Unknown

Definition: The international carrier indicator of the calling party.
— This field is currently always set to not indicated.

Example: The international carrier is not indicated.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.128 OLNS International Carrier Identification Digit Count

Description: The number of digits in the international carrier code of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: The number of digits in the international carrier code.

Example: The number of digits is 0.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.129 OLNS International Carrier Identification Code

Description: The international carrier identification code of the calling party returned by

the OLNS database.

Length: 2 bytes

Format: BCD

Range: 4 BCD digits, each digit ranging from 0 to 9 (0000 to 9999)

Definition: The international carrier identification code.

Example: There are no digits international carrier digits.

7	4	3	0	Byte #
#F	#F	#F	#F	0
#F	#F	#F	#F	1

10.130 OLNS IntraLATA Toll Carrier Indicator

Description: The intraLATA toll carrier indicator of the calling party returned by the OLNS LIDB.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = IntraLATA Toll CIC Not Indicated
2 = IntraLATA Toll CIC Indicated
3 = IntraLATA Toll CIC Denied
4 = IntraLATA Toll CIC Not Indicated - IntraLATA toll pre-subscription does not exist
5 = IntraLATA Toll CIC Not Indicated - IntraLATA toll calls not allowed
254 = Unknown

Definition: The intraLATA toll carrier indicator of the calling party.
— This field is set to either not indicated or indicated. The other values are currently not used.

Example: The intraLATA toll carrier is not indicated.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.131 OLNS IntraLATA Toll Carrier Identification Digit Count

Description: The number of digits in the intraLATA toll carrier code of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 to 4

Definition: The number of digits in the intraLATA toll carrier code.

Example: The number of digits is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.132 OLNS IntraLATA Toll Carrier Identification Code

Description: The intraLATA toll carrier identification code of the calling party returned by the OLNS database.

Length: 2 bytes

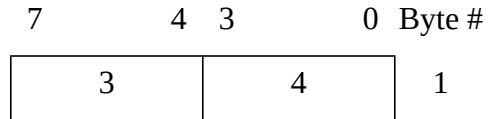
Format: BCD

Range: 4 BCD digits, each digit ranging from 0 to 9 (0000 to 9999)

Definition: The intraLATA toll carrier identification code.

Example: 1234.

7	4	3	0	Byte #
1	2		0	



10.133 OLNS Service/Equipment Indicator

Description: The service/equipment indicator of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = POTS Line - Business/Residential
2 = LEC Public - Standard Interface - Postpay Overtime
3 = POTS Line - Residential - Message Rate 1
4 = POTS Line - Residential - Message Rate 2
5 = LEC Semi-Public
6 = POTS Line - Business - Flat Rate
7 = POTS Line - Business - Message Rate 1
8 = Coinless (non-IPP)
9 = Coinless (IPP)
10 = LEC Prepaid Telecommunications Card Station
11 = POTS Line - Business - Message Rate 2
12 = LEC Public Standard Interface - Prepay Overtime
13 = LEC Public - Alternate Interface
14 = IC Public - Standard Interface
15 = IC Public - Alternate Interface
16 = POTS Line - Residential - Flat Rate
17 = Voice Quote - without tax - hotel
18 = Voice Quote - with tax - hotel
19 = IPP - Standard Interface
20 = IPP - Alternate Interface
21 = Hospital
22 = Prison (non-IPP)

23 = Auto Quote - without tax - hotel
24 = Auto Quote - with tax - hotel
25 = Dormitory Line
26 = Centrex Line
27 = PBX Line
28 = Prison (IPP)
29 = WATS Line
30 = Cellular
31 = Pager
32 = Personal Communication Service (PCS)
33 = FGA (Feature Group A)
34 = Mobile
35 = LEC Public - Special Billing - Post-Pay Overtime
36 = LEC Public - Special Billing - Pre-Pay Overtime
37 = Public - Incompatible Network Interface (No ACTS)
38 = Cellular- Rate 1
39 = Cellular- Rate 2
40 = POTS Line- Business- Single-line
41 = POTS Line- Business- Multi-line
42 = Public- Post-pay
254 = Unknown

Definition: The service/equipment type of the calling party.

Example: The service/equipment type is cellular rate 1.

7	6	5	4	3	2	1	0
0	0	1	0	0	1	1	0

10.134 OLNS TA Collect Indicator

Description: The TA collect billing restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Allowed for Domestic
3 = Not Allowed
254 = Unknown

Definition: The TA collect billing restrictions.

Example: TA collect is not allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.135 OLNS TA Credit Card Indicator

Description: The TA calling card billing restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Allowed for Domestic
3 = Not Allowed
254 = Unknown

Definition: The TA calling card restrictions.

Example: TA calling card calls are only allowed for domestic calls.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.136 OLNS TA Sent Paid Indicator

Description: The TA sent paid billing restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Allowed for Domestic
3 = Allowed for IntraLATA Calls Due to Non-Payment
4 = Allowed for IntraLATA Calls at Customer Request
5 = Not Allowed
254 = Unknown

Definition: The TA sent paid restrictions.

Example: TA sent paid billing is allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.137 OLNS TA Special BNS Indicator

Description: The TA special BNS restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Not Allowed
254 = Unknown

Definition: The TA special BNS restrictions.

Example: TA special BNS billing is allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.138 OLNS TA Third Number Indicator

Description: The TA third number billing restrictions for the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Allowed
2 = Allowed for Domestic
3 = Not Allowed
4 = Allowed with Operator Verification
5 = Allowed with Operator or Automated Verification
254 = Unknown

Definition: The TA third number billing restrictions.

Example: Third number billing is allowed with operator verification.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.139 OLNS Treatment Indicator

Description: The treatment value of the calling party returned by the OLNS database.

Length: 1 byte

Format: Binary

Range: 0 = Screened
1 = Automated Treatment- Provide Alerting Tone

- 2 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 1
- 3 = Operator Treatment- Operator Handling Due to Station Limitations
- 4 = Operator Treatment- Operator Handling at Customer Request
- 5 = Special Treatment- Handicapped
- 6 = Special Treatment- Deaf
- 7 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 2
- 8 = Automated Treatment- Provide Alerting Tone and Prompting Announcement 3
- 9 = Operator Treatment- High Fraud
- 10 = Automated Treatment- Provide Alerting Tone - No Operator Access
- 11 = Automated Treatment- Provide Alerting Tone and Operator Announcement 4 - No Operator Access
- 12 = Automated Treatment- Provide Alerting Tone and Operator Announcement 5 - No Operator Access
- 13 = Automated Treatment- Provide Alerting Tone and Operator Announcement 6 - No Operator Access
- 14 = Automated Treatment- Provide Alerting Tone and Operator Announcement 7
- 15 = Automated Treatment- Provide Alerting Tone and Operator Announcement 8
- 16 = Automated Treatment- Provide Alerting Tone and Operator Announcement 9
- 17 = Automated Treatment- Provide Alerting Tone and Operator Announcement 10- No Operator Access
- 18 = Automated Treatment- Provide Alerting Tone and Operator Announcement 11- No Operator Access
- 19 = Automated Treatment- Provide Alerting Tone and Operator Announcement 12- No Operator Access
- 20 = Automated Treatment- Special Handling 1
- 21 = Automated Treatment- Special Handling 2
- 22 = Automated Treatment- Special Handling 3
- 23 = Automated Treatment- Special Handling 4
- 24 = Automated Treatment- Special Handling 5

25 = Automated Treatment- Provide Alerting Tone and Operator Announcement 13

26 = Automated Treatment- Provide Alerting Tone and Operator Announcement 14- No Operator Access

254 = Unknown

Definition: The treatment value.

Example: The treatment value is special treatment- deaf.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0

10.140 Operation ID

Description: Uniquely identifies an OAP operation.

Length: 2 bytes

Format: Binary

Range: Hex value: 0100 to FFFF

Definition: Uniquely identifies an OAP request operation or inform operation. Each operation identifier is independent of class or direction.

- The identifiers for the OAP operations defined in this document range from #0100 to #DFFF, inclusive. Identifiers ranging from #E000 to #FFFF are reserved for external use (e.g. service node to service node use).

Example: Operation ID is #03B4.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	1	1	0
1	0	1	1	0	1	0	0	1

10.141 Operator ID

Description: Specifies the operator.

Length: 2 bytes

Format: Binary

Range: 0 to 9997

Definition: The operator identifier points to a specific operator. Table TQOPROF and Table OPRDAT contain a list of all valid operators.

Example: Operator ID is 640 (#0280).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
02	0	0	0	0	0	0	1	0	0
80	1	0	0	0	0	0	0	0	1

10.142 Operator Services System Action

Description: Specifies the action taken by the Operator Services System (OSS) for AMA recording purposes.

Length: 1 byte

Format: Binary

Range: 0 = Unknown

1 = Automatically Accepted

2 = Automatically Accepted due to operator methods.

3 = Verified Accepted

4 = Verified Rejected

5 = Provided Restricted Billing Treatment

6 = Transferred to IC Operator Services System

7 = Provided Intercept Service

8 = Inward Service

9 = Disconnected

Definition: Operator Services System Action

- Value 0 = No applicable OSS action has been taken. This value is used for all calls referred to an operator.
- Value 1 = Calling card, collect and third number billed calls for which billing is automatically accepted because the database query responds with "always accept" and Account Code billing calls which do not require database query but require other checks which have been passed.
- Value 2 = Third number and collect calls that have billing accepted without verification with the billed party because time of day or other restrictions apply.
- Value 3 = Third number and collect calls for which the billing is verified and accepted by the billed party.
- Value 4 = Third number and collect calls for which the billing is verified but rejected by the billed party.
- Value 5 = Calls on which the billing validation query or Service Node datafill indicates that the billing number is denied.
- Value 6 = Calls that are transferred to a carrier for which operator services are not provided after billing has been chosen.
- Value 7 = Third number and collect billed calls for which the billing number is on intercept as indicated by the billing validation query.
- Value 8 = Operator to operator calls.
- Value 9 = Calls on which the billing number is denied due to the results of the billing validation query or Service node datafill and the call is terminated without a retry.

Example: A third number billed call is automatically accepted due to operator methods during a time of day in which billing acceptance from the billed party is performed.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.143 Operator System Selector

Description: Indicates whether a call is eligible for TOPS automated systems or if it is only allowed to be handled by an operator.

Length: 1 byte

Format: Binary

Range: 0 = Operator only handling
1 = Operator or TOPS automated system handling

Definition:

- Value 0 = Call may only be handled by an operator, bypassing any TOPS automated systems.
- The call may be handled by either an operator or TOPS automated system.

Example: The call should bypass any TOPS automated systems and go directly to an operator.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.144 Operator System Type

Description: Identifies the type of operator system used by the switch for handling a TOPS call.

Length: 1 byte

Format: Binary

Range: 0 = Nil
1 = TOPS Operator
2 = AABS (Automated Alternate Billing Service)
3 = ACTS (Automated Coin Toll Service)
4 = ADAS (Automated Directory Assistance Service)
5 = MCCA (Mechanized Calling Card Service)

Definition: Identifies the operator system used for the call.

- Value 0 = Call has received no operator system handling
- Value 1 = Call handling provided by a TOPS operator
- Value 2 = Call handling provided by the AABS system.
- Value 3 = Call handling provided by the ACTS system.
- Value 4 = Call handling provided by the ADAS system.
- Value 5 = Call handling provided by the MCCA system.

Example: Call handling provided by the AABS system.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.145 Party

- Description: Identifies the party associated with the data block.
- Length: 1 byte
- Format: Binary
- Range: 1 = A Party
2 = B Party
- Definition: This field associates the information in the data block with a specific party. The party field values are parties that can be involved in call completion.
- Example: Party data is for B Party.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.146 Pass Through Data

- Description: A block of data that is passed between two function providers connected to the call.
- Length: 100 bytes
- Format: Binary
- Range: None.
- Definition: This field contains a block of data that the switch will receive from one function provider connected to a call and forward on to the other function provider connected to the call.
- Example: A 100 byte block with data whose format is unknown to the switch.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	1
	0	1	0	0	0	0	0	0	2
	0	0	0	0	0	0	1	0	:
	0	1	1	1	0	0	0	0	:
	0	0	0	0	0	0	0	0	98
	0	0	0	1	0	0	0	0	99

10.147 Pass Through Data Length

Description: The length of the Pass Through Data

Length: 1 byte

Format: Binary

Range: 0 to 100

Definition: This field contains the length for the Pass Through Data field.

Example: Length is 20.

7	6	5	4	3	2	1	0
0	0	0	1	0	1	0	0

10.148 Passive Node Status

Description: Indicates the status of a passive function provider during a simultaneous interaction.

Length: 1 byte

Format: Binary

- Range:**
- 0 = Passive function provider connected
 - 1 = Passive function provider released
 - 2 = Passive function provider performing restricted operation
 - 3 = Passive function provider completed restricted operation
 - 4 = Passive operator being connected
- Definition:** The status of a passive function provider during a simultaneous interaction.
- Value 0 = The passive function provider has been connected.
 - Value 1 = The passive function provider has been released.
 - Value 2 = The passive function provider is performing a restricted operation (e.g. Voice Connect Request). The active service node should refrain from sending any requests until the restricted operation is complete. For most requests, the active service node will receive an Error Response if they attempt to make a request while a restricted operation is outstanding for the passive function provider.
 - Value 3 = The restricted operation that was being processed for the passive function provider has completed. The active node may resume sending requests.
 - Value 4 = A passive operator is being connected. The active service node should refrain from sending any requests while the passive operator is being connected. Once the operator is fully connected, the active service node will receive this field with Value = 0 (Passive function provider connected). For most requests, the active service node will receive an Error Response if they attempt to make a request while the passive operator is being connected.
- Example:** A passive function provider has been connected.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.149 Phone Class Type

- Description:** Indicates the phone class type of the call.
- Length:** 1 byte
- Format:** Binary
- Range:** 0 = Undetermined

- 1 = Station
- 2 = Prepay CCF
- 3 = Prepay CDF
- 4 = Postpay Coin
- 5 = Hotel/Motel
- 6 = Hotel/Motel With No Charge
- 7 = Attended Pay Station
- 8 = Marine
- 9 = Mobile
- 10 = Institution

Definition: The value for this field is obtained from the originating trunk. The following are the values for this field:

- Value 0 = The phone class type is not known for the call.
- Value 1 = The call is from a normal station phone, sometimes known as a plain old telephone station (POTS) line.
- Value 2 = The call is from a prepay coin-coin first (CCF) telephone. A coin deposit is required before dial tone is returned. A prepay phone has an intermediate container, called a hopper, which can be tipped in one of two directions. The coins can either be deposited in the collection box or returned to the subscriber.
- Value 3 = The call is from a prepay coin dial tone first (CDF) telephone. Dial tone is received as soon as the A party goes off hook. A prepay phone has an intermediate container, called a hopper, which can be tipped in one of two directions. The coins can either be deposited in the collection box or returned to the subscriber.
- Value 4 = The call is from postpay coin telephone. The deposit slots are connected directly to the collection box. The subscriber deposits money only after the forward party has been reached. This type of phone can accept coin deposits, but is not equipped to return coins.
- Value 5 = The call is from a hotel or motel phone. If the call is charged to the room number, then the room number and/or authorization number is required. If the call is not charged to the room number, then the room number is not required.
- Value 6 = The call is from a hotel or motel phone. Regardless of how the call is billed, the room number is required since the Hotel/Motel may charge the occupant of the room for the phone access.
- Value 7 = The call is from an attendant pay station where payment is made to an attendant.

- Value 8 = The call is from a marine unit.
- Value 9 = The call is from a mobile unit.
- Value 10 = The call is from an institution (e.g. lawyer's office, hospital).
If the call is a direct dialed call, then an authorization number is required.

Example: Phone Class Type is hotel/motel.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1

10.150 Portability Status

Description: Specifies the portability status of the directory number.

Length: 1 byte

Format: Binary

Range: 0 = DN not ported
1 = DN ported
254 = Unknown

Definition: This field is used to indicate whether or not the specified party's DN is ported. Refer to the Functionality Overview section on LNP for more information on DN portability. The following are the values for this field:

- Value 0 = The switch determines the DN is not ported based on datafill or the LNP query result.
- Value 1 = The switch determines the DN is ported based on the LNP query result.
- Value 254 = An LNP query is not made, the switch does not know whether the DN is ported or not. This value is used when the LRN information is obtained through switch datafill instead of an LNP query.

Example: The DN is ported

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.151 Predicted Wait Time

Description: Specifies the predicted wait time (in tenths of seconds) for a given function provider request.

Length: 2 bytes

Format: Binary

Range: 0 to 32767

When a function provider request is made, it is possible that a function provider may not be immediately available to serve the call. If this is the case, the Predicted Wait Time field indicates how long (in tenths of seconds) the switch projects it will take for a function provider to become available to serve the call.

Each call queue in the switch has datafill in table QMSCQDEF (QMS Call Queue Definition table) that indicates the maximum Predicted Wait Time that will be allowed before calls start deflecting from the call queue (i.e. if it is going to be too long of a wait before a function provider will become available, the call will not queue and the service node making the function provider request will receive an indication of the error).

The maximum Predicted Wait Time can also be customized on a session pool basis through datafill in table OQCQPROF (OSSAIN Call Queue Profile table). If the maximum Predicted Wait Time is customized for a given session pool in table OQCQPROF, it overrides the maximum Predicted Wait Time datafilled for the call queue in table QMSCQDEF.

- 0 to 32766 are valid Predicted Wait Times
- 32767 is the NIL value. This value can be used if the session pool serving the call queue associated with the passive function provider is not in-service.

Example: The Predicted Wait Time for the call is 16 tenths of a second.

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	1

10.152 Presentation Indicator

Description: Indicates the presentation status of an associated directory number.

Length: 1 byte

Format: Binary

Range: 0 = Presentation Allowed
1 = Presentation Restricted
255 = Presentation Status Unknown

Definition: This field is used to indicate whether or not the specified DN is eligible for display on another party's terminal equipment.

- Value 0 = The number is allowed to be displayed.
- Value 1 = The number is restricted from being displayed.
- Value 255 = The presentation status of the number is unknown.

Example: The number is Presentation Restricted.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.153 Protocol Version

Description: Specifies the protocol version (X.Y).

Length: 2 bytes

Format: Binary

Range: 0 to 255 for protocol release; 0 to 255 for protocol increment

Definition: The protocol version consists of the release (X) and increment (Y).

Example: Protocol version is 2.3.

	7	6	5	4	3	2	1	0	Byte #
Protocol Version	0	0	0	0	0	0	1	0	0
Protocol Increment	0	0	0	0	0	0	1	1	1

10.154 Queue Peg Indicators

Description: Specifies what session statistics should be incremented.

Length: 1 byte

Format: Set

Range:

- Bit 0 = Peg Initial Session Seizure (ISS)
- Bit 1 = Peg Service Node Origination Seizure (SNOS)
- Bit 2 = Peg Trigger Session Seizure (TGSS)
- Bit 3 = Peg Transfer Session Seizure (TXSS)
- Bit 4 = Peg Non Call Work Volume (NCWV)
- Bit 5 = Peg Idle Time (IDLT)
- Bit 6 = Peg Call Busy Work Volume (CBWV)

Definition: The following are the statistics:

- Bit 0 = Peg ISS: A subscriber originated call has arrived at the service node for the first time.
- Bit 1 = Peg SNOS: A service node originated call has been processed by the node for the first time.
- Bit 2 = Peg TGSS: This is a call that was previously floated and has hit a trigger event causing the call to route to a service node.
- Bit 3 = Peg TXSS: This is a call that arrived at a service node due to a transfer from another service node or operator.
- Bit 4 = Peg NCWV: Peg of the time spent while the service node is not available to accept calls and no calls are at this service node.
- Bit 5 = Peg IDLT: Peg of the time spent available to accept calls but with no calls at this service node.
- Bit 6 = Peg CBWV: Peg of the time spent processing a call at this service node.
- Bit 7 = This bit is not used and will always be set to 0.

Example: Peg ISS and CBWV.

7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1

10.155 Queuing Event

Description: Specifies how the call was handled when queued.

Length: 1 byte

Format: Binary

Range: 0 = Call in Queue
1 = Call Abandoned
2 = Call Presented to Agent
3 = Call Deflected - wait time too large
4 = Call Deflected - maximum queue size exceeded
5 = Call Queue Overflow

Definition: The following are the possible events:

- Value 0 = Call placed in queue.
- Value 1 = Calling subscriber abandoned while the call was in queue (e.g. calling party when onhook).
- Value 2 = Call presented to an OSSAIN session.
- Value 3 = Call deflected from queue due to the predicted wait time for the queue being longer than the maximum queue time allowed by datafill in table QMSCQDEF.
- Value 4 = Call deflected from queue due to queue size exceeded (field CQMAXSIZ, table QMSCQDEF).
- Value 5 = Call queue overflow. There are no more queuing elements (table QAPLNDEF).

Example: Call is in queue.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.156 Queuing System

Description: Specifies which queuing system was used in the switch for a given function provider request.

Length: 1 byte

- Format:** Binary
- Range:** 0 = OSSAIN service nodes
1 = TOPS operators
- Definition:** The switch uses different call queues, agent (function provider) queues, and call queue profiles for service nodes than it does for TOPS operators. For a given function provider request, this field will indicate whether a service node was requested for the call or whether a TOPS operator was requested for the call.
- Value 0 = The function provider request resulted in a request for a session with a service node from the OSSAIN service node queuing system.
 - Value 1 = The function provider request resulted in a request for a TOPS operator from the TOPS operator queuing system.
- Example:** The OSSAIN service node queuing system was used.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.157 RAO Code

- Description:** Specifies the Regional Accounting Office Code.
- Length:** 2 bytes
- Format:** BCD
- Range:** 3 BCD digits, ranging from 0 to 9
- Definition:** The RAO code for the calling card number. The number of digits in the RAO Code is specified by the RAO Digit Count field.
- RAO codes range from 200 - 999.
 - RAO code of 000 is a nil RAO code.
- Example:** RAO is '678'.

7	4	3	0	Byte #
6	7			0

7	4	3	0	Byte #
8	F			1

10.158 RAO Digit Count

Description: Specifies the number of digits in the RAO Code field.

Length: 1 byte

Format: Binary

Range: 0 to 3

Definition: This field specifies the number of digits in the RAO Code field.

Example: RAO digit count is 3.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.159 Release Node Indicator

Description: Specifies which function provider to release from the call.

Length: 1 byte

Format: Binary

Range: 0 = Self
1 = Other Node

Definition: The Release Node Indicator is used to indicate whether a service node wishes to release itself or the other function provider from the call. This field is only applicable when two nodes are connected to the call.

- Value 0 = The service node wishes to release itself from the call, thus leaving the other function provider as the only function provider connected to the call.
- Value 1 = The service node wishes to release the other function provider from the call, thus leaving itself as the only function provider connected to the call.

Example: Release the other function provider from the call.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.160 Report Text

Description: Specifies one line of a log report text.

Length: 80 bytes

Format: ASCII

Range: All ASCII characters with the exception of control characters.

Definition: This field specifies the printable text to be printed on one line of a log report.

Example: OAIN 700 ERROR

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
O	4F	0	0	1	1	0	0	0	0	0
A	41	0	1	0	0	0	0	0	1	1
I	49	0	1	0	0	1	0	0	1	2
N	4E	0	1	0	0	1	1	1	0	3
space	20	0	0	1	0	0	0	0	0	4
7	37	0	0	1	1	0	1	1	1	5
0	30	0	0	1	1	0	0	0	0	6
0	30	0	0	1	1	0	0	0	0	7
space	20	0	0	1	0	0	0	0	0	8
E	45	0	1	0	0	0	1	0	1	9
R	52	0	1	0	1	0	0	1	0	10

ASCII	Hex	Binary								
R	52	0	1	0	1	0	0	1	0	11
O	4F	0	0	1	1	0	0	0	0	12
R	52	0	1	0	1	0	0	1	0	13
nil	F	F	F	F	F	F	F	F	F	.
nil	F	F	F	F	F	F	F	F	F	.
nil	F	F	F	F	F	F	F	F	F	78
nil	F	F	F	F	F	F	F	F	F	79

10.161 Report Text Length

Description: Specifies the number of characters on one line of a log report.

Length: 1 Byte

Format: Binary

Range: 0 to 80

Definition: This field contains the length of printable ascii characters on one line of a log report.

Example: Report text length is 60.

7	6	5	4	3	2	1	0
0	0	1	1	1	1	0	0

10.162 Requested RLT Action

Description: Specifies if the call should attempt RLT.

Length: 1 byte

Format: Binary

Range: 0 = Do not attempt RLT
1 = Attempt RLT

Definition: Specifies whether or not the switch should attempt RLT.

- Value 0 = RLT should not be attempted by the switch.
- Value 1 = RLT should be attempted by the switch.

Example: Attempt RLT.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.163 Reset Time Flag

Description: Indicates whether the MIS node should reset the timestamp associated with the session id specified in the data block.

Length: 1 byte

Format: Binary

Range: 0 = Do not reset time
1 = Reset time

Definition: This field specifies whether or not the MIS node should reset the timestamp associated with the session id specified in the data block. If the value of this field is reset time, then the MIS node should store the timestamp given in the data block against the session id. When the DMS sends a request to peg idle time or work volume, the timestamp stored in used to calculate the correct amount of idle time or work volume.

Example: Reset time.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.164 Restart Severity

Description: Indicates the severity of a restart, if one occurred before the current information was composed.

Length: 1 byte

Format:	Binary
Range:	0 = No Restart 1 = Warm 2 = Cold 3 = Reload
Definition:	The restart level that occurred in the DMS prior to the events being reported. • Value 0 = No restart has occurred since the last information data block was composed. • Value 1 = A warm restart has occurred since the last information data block was composed. Calls remain at service node and in queue. • Value 2 = A cold restart has occurred since the last information data block was composed. All calls at service node and all calls in queue were taken down. • Value 3 = A reload restart has occurred since the last information data block was composed. All calls at service node and all calls in queue were taken down.
Example:	No restart.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.165 Restricted Billing Index

Description:	Specifies the index of the Restricted Billing Indicator.
Length:	1 byte
Format:	Binary
Range:	0 to 100
Definition:	This is the AMARBC value from Table RESTBIL which identifies classes of restricted originators (e.g. call originated from a prison) if desired. The value for this field is based on the originating trunk (from table TRKGRP) or on the A party's directory number (table SPLDNID or DNSCRN). A value of 100 is the 'nil' value (i.e. unrestricted originator).

Example: Index of 5.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1

10.166 Routing Index

Description: Identifies an alternate routing index to be used by the switch to obtain a route. The resulting route overrides the route obtained by the switch when performing translations.

Length: 2 bytes

Format: Binary

Range: 0 to 2046

Definition: Contains an index into switch table OAINRTE which maps the index to an office route in one of the OFRx tables. For a Routing Index to be valid, it must be datafilled in table OAINRTE.

- 0 to 2045 are valid Routing Indices that could map to an office route
- 2046 is a NIL value

Example: Routing Index is 212 (#00D4)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
D4	1	1	0	1	0	1	0	0	1

10.167 SN Voice Conferencing Status

Description: Specifies whether or not the SN voice link is connected to a conference.

Length: 1 byte

Format: Binary

- Range:** 0 = SN voice link in call
1 = SN voice link in a conference
- Definition:** This field specifies what the SN voice link should be connected to (if the field is included in an SN request) or it specifies what the SN voice link is connected to (if the field is sent from the switch).
- Value 0: The voice link should be/is connected to a call.
 - Value 1: The voice link should be/is connected to a conference.
- Example:** Voice link should be/is connected to a conference.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.168 SN Voice Link Status

- Description:** Specifies whether the voice link is connected or released.
- Length:** 1 Byte
- Format:** Binary
- Range:** 0 = Connected
1 = Released
- Definition:** This field indicates whether the voice link is connected or released.
- Value 0 = The voice link is connected
 - Value 1 = The voice link is released
- Example:** The voice link is released

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.169 SPID Processing

- Description:** Indicates if SPID processing is enabled in the switch for a specific DN.

Length:	1 byte
Format:	Set
Range:	Bit 0 = A Party DN Bit 1 = B Party DN Bit 2 = Alt DN
Definition:	This field indicates if SPID processing is enabled for a DN. It is enabled based on trunk group datafill in switch table TOPSTOPT. • A bit set to one indicates that SPID processing is enabled at the switch for the DN. The service node may send in SPID information for the DN. • SPID processing applies to both the Account Owner SPID (AO) and Billing Service Provider SPID (BSP). • SPID processing enabled does not mean the service node will receive SPID information for the DN. • Bits 3 - 7 are currently unused and will have a value of zero.
Example:	SPID processing for the A party DN is enabled at the switch.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.170 SS7 Backward Call Indicators

Description:	Provides SS7 call status information on an off-hook or as the result status of a successful connection request on an ISUP trunk. This is the Backward Call Indicators field value returned by the ANM (ANswer Message on ISUP trunks) or ACM (Address Complete Message on ISUP trunks).
Length:	2 bytes
Format:	Binary
Definition:	Specifies several SS7 Backward Call Indicators values. See TR-NWT-000246 for details. The values of each bit is shown below:

	7	6	5	4	3	2	1	0	Byte #
Bits 0 - 7	H	G	F	E	D	C	B	A	0
Bits 8 - 15	P	O	N	M	L	K	J	I	1

- Bits BA: Charge Indicator
 - Value 00 = No Indication (default)
 - Value 01 = No Charge
 - Value 10 = Charge
 - Value 11 = Spare
- Bits DC: Called Party's Status Indicator
 - Value 00 = No Indication (default)
 - Value 01 = Subscriber free
 - Value 10 = Connect when free
 - Value 11 = Excessive delay
- Bits FE: Called Party's Category Indicator
 - Value 00 = No Indication (default)
 - Value 01 = Ordinary (non-payphone) subscriber
 - Value 10 = Payphone
 - Value 11 = Spare
- Bits HG: End-to-End Method Indicator
 - Value 00 = No end-to-end method available
 - Value 01 = Pass along method available
 - Value 10 = SCCP method available
 - Value 11 = pass along and SCCP methods available
- Bit I: Interworking Indicator
 - Value 0 = No interworking encountered
 - Value 1 = Interworking encountered
- Bit J: IAM Segmentation Indicator
 - Value 0 = No indication

- Value 1 = Additional information has been received and incorporated into call setup
- Bit K: ISDN User Part Indicator
 - Value 0 = ISDN User Part not used all the way
 - Value 1 = ISDN User Part used all the way
- Bit L: Holding Indicator
 - Value 0 = Holding not required (default)
 - Value 1 = Holding required (No procedure specified for US networks)
- Bit M: ISDN Access Indicator
 - Value 0 = Terminating access non-ISDN
 - Value 1 = Terminating access ISDN
- Bit N: Echo Control Device Indicator
 - Value 0 = Incoming half echo control device not included (default)
 - Value 1 = Incoming half echo control device included
- Bits PO: SCCP Method Indicator
 - Value 00: No Indication (default)
 - Value 01 = Connectionless method available (No procedures specified for US networks)
 - Value 10 = Connection oriented method available (No procedure specified for US networks)
 - Value 11 = Connectionless and Connection Oriented methods available (No procedure specified for US networks)

Example: Charge call (10), B party is 'subscriber free' (01), No indication on Called party (00), No end-to-end method (00), no interworking (0), No indication on IAM Segmentation (0), ISDN User Part used all the way (1), holding not required (0), terminating access ISDN (1), Incoming half echo control device not included (0), No indication on SCCP method (00)

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0	1

10.171 SS7 Release Cause

Description: Specifies the SS7 Release Cause value indicating the release reason received from an ISUP trunk terminating party for unsuccessful connections. Note that the switch relates Treatment to SS7 Release Cause values through switch table TMTMAP.

Length: 1 byte

Format: Binary

Range: 0 to 127

Definition: Specifies one of the SS7 Release Cause values for ISUP trunks. SS7 Release Cause values are specified in TR-NWT-000246.

Example: Outpulsing failure, trigger release cause number 1, Unallocated Number.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.172 Seconds

Description: Specifies the current seconds.

Length: 1 byte

Format: Binary

Range: 0 to 59

Definition: The seconds of the minute.

Example: Seconds is 42 (#2A).

7	6	5	4	3	2	1	0
0	0	1	0	1	0	1	0

10.173 Sequence Number

Description: Identifies the number of messages, including the current one, the message

sender has sent.

Length: 2 bytes

Format: Binary

Range: 0 to 65535

Definition: Identifies the number of messages each node has sent to another node. The sequence number in the class header identifies the number of messages, including the current one, that the message source node has sent to the destination node.

- This number is used to determine if a message was lost previously in the call.
- Zero is the nil sequence number.

Example: Sequence number is 10 (#000A).

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	1

10.174 Service Node Mode

Description: Specifies if the connection mode of the service node.

Length: 1 byte

Format: Binary

Range: 0 = Passive

1 = Active

2 = Node not currently connected

Definition: This field specifies the connection mode of the service node.

- Value 0 = The service node is the passive function provider for the call. It does not have call control and is limited in terms of what requests it can send to the switch.
- Value 1 = The service node is the active function provider for the call. It has call control capabilities for the call.
- Value 2 = The service node is not currently connected to the call.

Example: The service node is the active function provider for the call.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.175 Service Peg Indicators

Description: Indicates what service measurements to peg.

Length: 1 byte

Format: Set

Range: Bit 0 = Peg Service Initiation
Bit 1 = Peg Service Work Volume

Definition: Service measurements:

- Bit 0 = A service, such as TA or DA, has been initiated.
- Bit 1 = Work volume pertaining to a service has been accumulated.
- Bits 2 to 7 are not used and are always set to 0.

Example: Peg both.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.176 Service Provider ID

Description: Specifies the Service Provider Identifier for an associated service provider party and type of service provider.

Length: 4 bytes

Format: ASCII

Range: 4 alphanumeric characters in the range {A-Z, 0-9}

Definition: A service provider ID known by the switch via Table SPID.

Example: 'ABCD'

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
A	41	0	0	1	0	0	0	0	1	0
B	42	0	0	1	0	0	0	1	0	1
C	43	0	0	1	0	0	0	1	1	2
D	44	0	0	1	0	0	1	0	0	3

10.177 Service Provider ID Length

Description: Specifies the length of the Service Provider ID field.

Length: 1 byte

Format: Binary

Range: 4

Definition: The length of the Service Provider ID field.

Example: Length is 4.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

10.178 Service Provider Party Identifier

Description: Specifies the party to which a SPID applies.

Length: 1 byte

Format: Binary

Range: 1 = Originating party directory number data
 2 = Terminating party directory number data
 3 = Billing party directory number data

- Definition: Specifies the party to which an associated SPID applies.
- Value 1 = The service provider applies to the A party.
 - Value 2 = The service provider applies to the B party.
 - Value 3 = The service provider applies to the billed party.

Example: Originating party directory number data.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.179 Service Type

Description: Specifies the base service assigned to the call.

Length: 1 byte

Format: Binary

Range: 0 = Toll and Assistance (TA)
1 = Directory Assistance (DA)

Definition: The following are the values for this field:

- Value 0 = Toll and Assistance service. Service node has no restrictions on how to dispose of the call.
- Value 1 = Directory Assistance service. Call requiring directory assistance may be sent to the service node to obtain alternate billing (typically to a third number or calling card number). Once billing has been established, the call should be routed to a control list to continue DA processing within the DMS switch environment. These calls should only be disposed of via a Transfer to Control List operation or an End Call operation.

Example: Service type is TA.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.180 Service Type Provided by Switch

Description: Specifies the type of service the switch provides on behalf of the carrier.

Length: 1 byte

Format: Binary

Range: 0 = Transfer only
1 = Complete operator services

Definition: Service type provided by the switch for the carrier.

- Value 0 = Indicates the call must be transferred to the carrier. The specified carrier is not served by the switch (NOSERV). If the call is transferred to the carrier, triggering is not in effect.
- Value 1 = Indicates the operating company has contracted with the specified carrier to provide full operator services on behalf of the carrier. The carrier is served by the switch (SERV).
- Refer to the Northern Telecom Translations Guide for more information on SERV/NOSERV carriers.

Example: Service Type is Complete Operator Services

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.181 Serving Agent Status

Description: Specifies whether or not there are any agent defined in the switch to serve the given call queue.

Length: 1 byte

Format: Binary

Range: 0 = No agents are defined to serve this call queue.
1 = Agents are defined to serve this call queue.

Definition: For OSSAIN service nodes, agents (i.e. sessions) are defined in the switch when the session pool containing those agents is brought into service. For TOPS operators, agents (i.e. operators) are defined in the switch when the operator logs into their terminal. When agents are defined, they are associated with one or more call queues. The Serving Agent Status field is

used to indicate if there is at least one agent defined in the switch that serves the given call queue.

- Value 0 = If the given call queue was in the OSSAIN service node queuing system, this value indicates that no session pools that serve the given call queue are in service. If the given call queue was in the TOPS operator queuing system, this value indicates that no operators that serve the given call queue are logged in.
- Value 1 = If the given call queue was in the OSSAIN service node queuing system, this value indicates that at least one session pool that serves the given call queue is in service. If the given call queue was in the TOPS operator queuing system, this value indicates that at least one operator that serves the given call queue is logged in.

Example: Agents are defined to serve this call queue.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.182 Session ID

Description: Identifies the session being used for the message.

Length: 2 bytes

Format: Binary

Range: 0 to 1023

Definition: The session ID for the message. The max number of sessions for each session pool id is datafilled in switch Table OASESNPL.

Example: ID is 20 (#0014)

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	1

10.183 Session Origination Type

Description: Identifies how the service node is being brought into the call.

Length: 1 byte

Format: Binary

Range: 0 = Initial Call Setup

1 = Transfer

2 = Recall/Trigger

3 = Reorigination

Definition: The reason the service node is being brought into the call.

- Value 0 = The service node is the first function provider processing the call.
- Value 1 = Another service node or operator transferred the call to a function, which resulted in sending the call to the receiving service node.
- Value 2 = The call had been floated. However, a trigger was hit that caused the call to be sent to the service node.
- Value 3 = The call was released from the switch (i.e. RLT), came back to the switch, and routed to the service node.

Example: Origination Type is Recall/Trigger.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.184 Session Pool ID

Description: Specifies the session pool for this session.

Length: 2 bytes

Format: Binary

Range: 0 to 65535

Definition: Identifies the session pool for this message. Session pool IDs are unique within a given OSSAIN network. Session Pool IDs are datafilled in switch Table OASESNPL. This table also contains the maximum number of sessions for each Session Pool ID.

- 0 to 4094 are valid session pool ids that can be datafilled in Table OASESNPL.

Example: ID is 312 (#0138)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
01	0	0	0	0	0	0	0	1	0
38	0	0	1	1	1	0	0	0	1

10.185 Session State

Description: The current state of the specified service node session.

Length: 1 byte

Format: Binary

Range: 0 = Out of service
1 = Not available for call processing
2 = Available for call processing

Definition: Indicates the current state of the session.

- Value 0 = The session is out of service and not available due to some maintenance activity on the session pool or service node.
- Value 1 = The session is not available for call processing as a result of a throttling or drain action on the session pool.
- Value 2 = The session is in service and available to process calls.

Example: The session is available for call processing.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.186 Session Type

- Description:** Identifies the type of OSSAIN processing allowed for the call.
- Length:** 1 byte
- Format:** Binary
- Range:** 0 = Standard OSSAIN session
1 = Preprocessing OSSAIN session
- Definition:** Type type of processing the service node should provide for the call. Refer to the Functionality Overview section for more information on OSSAIN Preprocessing.
- Example:** Standard OSSAIN processing

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.187 Solicitor Number

- Description:** This field is used to identify the maintenance task that is running at the service node or the switch.
- Length:** 2 Bytes
- Format:** Binary
- Range:** 0 to 65535
- Definition:** Identifies the maintenance task for the service node or the switch.
- Range 0 - 250 identifies the maintenance task running at the switch. The service node should echo the value back in the response.
 - Range 256 - 65534 identifies the maintenance task running at the service node. The switch will echo the value back in the response.
 - Session Pool Class, Session Pool Alarm Class, Session Pool Log Class: the solicitor number for service node originated requests/ informs is limited to the range 256 - 511.
 - 65535 is the nil value.
- Example:** Solicitor Number is 250.

7	6	5	4	3	2	1	0	Byte #
0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	0	1

10.188 Special Number Format

Description: The format of the billing number.

Length: 1 byte

Format: Binary

Range: 0 = Domestic

1 = Overseas

2 = CCITT Calling Card

254 = Unspecified

Definition: The type of billing number when the billing number is either a calling card or alternate directory number.

Example: The billing number is domestic.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.189 Special Number Type

Description: The type of billing number used for a call.

Length: 1 byte

Format: Binary

Range: 0 = Third Number

1 = Special Number

2 = Calling Card

254 = Unspecified

- Definition:** The type of billing number with a special class charge.
- Value 0 = An alternate directory number.
 - Value 1 = A ten digit RAO billing number. This type of billing number is not used for OSSAIN calls.
 - Value 2 = A type of calling card.
 - Value 254 = unspecified
- Example:** The billing number is a calling card.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.190 Special Number Validity

- Description:** Specifies the validity of the billing number.
- Length:** 1 byte
- Format:** Binary
- Range:**
- 0 = Valid
 - 1 = Verify
 - 2 = Invalid
 - 3 = Hotlisted
 - 254 = Unspecified
- Definition:** This specifies the validity of the billing number when the billing number is either a calling card or alternate directory number.
- Value 0 = The billing number is valid.
 - Value 1 = The billing number needs to be verified.
 - Value 2 = The billing number is invalid.
 - Value 3 = The billing number is marked for special treatment (e.g. stolen or unpaid).
 - Value 254 = unspecified
- Example:** The billing number needs to be verified.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.191 Speech Path Status

Description: Specifies the status of the voice connection for the specified party.

Length: 1 byte

Format: Binary

Range: 0 = Zero-Way (Neither Speak nor Listen)

1 = Two-Way (Both Speak and Listen)

2 = One-Way (Listen Only)

3 = One-Way (Speak Only)

Definition: This field specifies the voice connection for the specified party.

- Value 0 = The specified party can neither speak nor listen.
- Value 1 = The specified party can both speak and listen.
- Value 2 = The specified party can only listen.
- Value 3 = The specified party can only speak.

Example: Specified party can only listen (one-way connection).

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.192 Tenths of Seconds Timestamp

Description: An absolute timestamp taken by the DMS CPU tick counter and converted into tenths of seconds.

Length: 2 bytes

Format: Binary

Range: 0 to 65535

Definition: Indicates the timestamp value. Invariant over external clock changes (e.g. daylight savings).

Example: 125 tenths of seconds (#007D).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
7D	0	1	1	1	1	1	0	1	1

10.193 Tenths Seconds

Description: Specifies the specified time in tenths of seconds.

Length: 1 byte

Format: Binary

Range: 0 to 9

Definition: The time in tenths of seconds

Example: Tenths of seconds is 9.

7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	1

10.194 Terminal ID

Description: The identifier for the operator position.

Length: 2 bytes

Format: Binary

Range: 0 to 9999

Definition: This field identifies the operator terminal for the operator specified in the Operator ID field. Table TOPSPOS contains a list of all valid terminal identifiers.

Example: Terminal ID is 301 (#012D).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
01	0	0	0	0	0	0	0	1	0
2D	0	0	1	0	1	1	0	1	1

10.195 Terminating Code Allowed Set

Description: Indicates whether Terminating Code Screening datafill in the switch allows the billing types in the specified set for a given A party to B party call completion.

Length: 1 byte

Format: Set

Range: Bit 0 = Card
 Bit 1 = Collect
 Bit 2 = Third Number

Definition: When both an A party and a B party DN have been defined for a call, the Terminating Code Allowed Set can be determined by the switch, indicating which billing types are allowed for completion of the call. Effectively, when a type of billing is disallowed, it indicates the operating company does not want that billing type to be used to complete calls from a specific DN to a specific region or country. The relevant switch tables are DOMBILL, OVSBILL, and TERMSCRN.

If a subscriber selects a type of billing which is not in the allowed set, the Service Node should transfer the call to an operator for manual, alternative billing procedures. The intent is to prevent fraud when completing calls at an automated node, and to discourage further potentially fraudulent attempts by bringing an operator into the call.

While the handling described above is the expected processing for the call, the Service Node does have the option of taking its own actions to inform the subscriber of the invalid billing, and to attempt to bill the call in an alternative manner, in lieu of directing the call to an operator.

Example: Collect and Third Number billing is allowed.

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0

10.196 Text Block

Description: A block of text data.

Length: 20 bytes

Format: ASCII

Range: full ASCII character set

Definition: The text data.

Example: Text data is “Mr. Smith”

ASCII	Hex	Binary								Byte #
		7	6	5	4	3	2	1	0	
M	4D	0	1	0	0	1	1	0	1	0
r	72	0	1	1	1	0	0	1	0	1
.	2E	0	0	1	0	1	1	1	0	2
space	20	0	0	1	0	0	0	0	0	3
S	53	0	1	0	1	0	0	1	1	4
m	6D	0	1	1	0	1	1	0	1	5
i	69	0	1	1	0	1	0	0	1	6
t	74	0	1	1	1	0	1	0	0	7
h	68	0	1	1	0	1	0	0	0	8
nil	#FF	1	1	1	1	1	1	1	1	9
nil	#FF	1	1	1	1	1	1	1	1	...
nil	#FF	1	1	1	1	1	1	1	1	19

10.197 Text Block Length

Description: This field is the length of the text data.

Length: 1 byte

Format: Binary

Range: 0 to 20

Definition: This field contains the length of the Text Block field.

Example: Length is 13 (#0D).

7	6	5	4	3	2	1	0
0	0	0	0	1	1	0	1

10.198 Time in Queue

Description: Specifies the total time a call was in queue (in tenths of seconds).

Length: 2 bytes

Format: binary

Range: 0 to 65535

Definition: When a call abandons or a call is presented to an agent, this field indicates the amount of time the call spent in queue.

Example: Time in queue for the call was 35 tenths of seconds (#0023).

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
23	0	0	1	0	0	0	1	1	1

10.199 Treatment

Description: Specifies the switch treatment.

Length: 1 byte

Format: Binary

Range: 0 to 254

Definition: This field indexes into switch table TMTCNTL for the treatment. This field specifies the treatment applied in the switch.

- This requires parallel datafill between the switch and the service node if the service node needs to make application decisions based on treatment.
- 0 to 217 are defined treatments. Refer to Appendix B for the mapping of specific treatments.
- 218 to 254 are undefined treatments.

Example: Treatment is 10.

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0

10.200 Trigger Detection Point

Description: Specifies the Trigger Detection Point (TDP) in which the trigger event occurred.

Length: 1 byte

Format: Binary

Range: 0 =Connect

1 = Talking

2 = Disconnect

Definition: This field specifies the TDP (state of the call) from which this session was initiated.

Example: Connect TDP

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

10.201 Trigger Event

Description: Specifies the trigger event that caused trigger processing to route the call to this node.

Length: 1 byte

Format: Binary

Range: 0 = Outpulsing Failure
1 = No Answer
2 = Flash
3 = Answer
4 = Notification
5 = Calling (A Party) Disconnect
6 = Party Disconnect

Definition: This field specifies the trigger event that initiated this session with the node.

- The No Answer and Answer trigger events applies to the B party.
- The Flash trigger event applies to the A party.
- The party disconnect applies to any party other than the A party.

Example: Outpulsing failure.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.202 Trigger Profile Index

Description: Identifies the trigger profile index.

Length: 2 bytes

- Format:** Binary
- Range:** 0 to 2047
- Definition:** This index is used by the switch to access call floated trigger datafill. The trigger profile index is datafilled in switch table OATPREFIX. This index indicates what triggers will be in effect if the call is floated.
- Values 0 to 2046 are valid trigger profile indices.
 - Value 2047 is the nil trigger profile index. No trigger processing is performed.
- Example:** Profile Index of 42. (#002A)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
2A	0	0	1	0	1	0	1	0	1

10.203 Trunk Group ID

- Description:** Specifies the trunk group identifier for the originating, terminating or SN voice link trunk.
- Length:** 2 bytes
- Format:** Binary
- Range:** 0 to 8191
- Definition:** The originating, terminating or SN voice link trunk group for the call. The trunk group is obtained from switch Table CLLI, field ADNUM.
- Example:** Trunk ID is 120 (#0078)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
78	0	1	1	1	1	0	0	0	1

10.204 Trunk Member ID

Description: Specifies the trunk member identifier for the originating, terminating or SN voice link trunk.

Length: 2 bytes

Format: Binary

Range: 0 to 10000

Definition: The originating, terminating or SN voice link trunk group member for the call. The member number is obtained from switch table TRKMEM.

- Values 0 to 9999 are valid trunk member numbers.
- Value 10000 is the unspecified or nil value.

Example: Trunk ID is 121 (#0079)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
00	0	0	0	0	0	0	0	0	0
79	0	1	1	1	1	0	0	1	1

10.205 Type of Service Provider

Description: Specifies the type of SPID.

Length: 1 byte

Format: Binary

Range: 1 = Account Owner
2 = Billing Service Provider

Definition: Specifies the type of SPID.

- Value 1 = The service provider represents the company that is responsible for the end user's service.
- Value 2 = The service provider represents the company that is involved in some combination of the following:
 - processing billing records

- billing the end user
- intercompany settlements (e.g. for calling card and bill to alternate number calls)

Example: The account owner SPID.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.206 Unenforced Restrictions

Description: Specifies whether the associated party directory number has either an UNPAID, BLCKCALL, or CCR attribute (or any combination) datafilled against it in Table DNSCRN.

Length: 1 byte

Format: Binary

Range: 0 = No DN unenforced restrictions associated with party number
1 = DN unenforced restrictions associated with party number

Definition:

- Value 0 = The associated DN party does not have any attributes datafilled against it in Table DNSCRN.
- Value 1 = The associated DN party does have attributes (or any combination of attributes) datafilled against it in Table DNSCRN.

Example: There is an attribute (or combination of attributes) datafilled against the associated directory number in Table DNSCRN.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.207 Validation Request

Description: Specifies whether the switch should validate the billing number.

Length: 1 byte

Format: Binary

Range: 0 = Not Applicable

1 = Do not validate

2 = Validate

Definition: This field indicates whether or not the switch should validate the billing number.

- Value 0 = This field is not applicable at this time.
- Value 1 = The switch should not perform its validation and not send a query to an external data base.
- Value 2 = The switch should perform its validation, including sending a query to an external data base if applicable.

Example: Validate billing number.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

10.208 Validation Result

Description: Specifies the results of the billing validation.

Length: 1 byte

Format: Binary

Range: 0 = Validation Not Performed

1 = Billing Denied

2 = Manual Validation Required

3 = Billing Acceptance Required

4 = Billing Accepted

5 = No Mutual Honoring Agreement

Definition: This field states in a succinct form the results of a billing validation.

- Value 0 = Validation cannot be performed because the billing number does not have a format that can be processed by the database. Currently this can happen in the following scenarios:
 - The Service Node requests collect billing and the called number is a foreign number.

When one of these scenarios occurs, the DMS accepts the requested billing and the Service Node can successfully float the call, even though no database query was actually made.

- Value 1 = Billing is denied.
- Value 2 = The billing number needs to be transferred to an operator for manual validation.
- Value 3 = The billing number has been validated. However, the billed party needs to verbally accept billing for the call.
- Value 4 = The billing number has been validated and the billing is accepted for the call.
- Value 5 = The Local Exchange Carrier (LEC) refuses to accept the calling card which is issued by a particular InterLATA Carrier (IC). A validation query is not sent to the calling card database in this case. This value is determined from switch Table DCICSET and Table DCICDEF. This applicable for CIID cards (i.e. a type of RAO card) and CCITT cards (i.e. a type of ISO card).

Example: Validation result is billing acceptance is required.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.209 Validation Result Determined by DMS Datafill

Description: Specifies if the billing number validation result was determined by datafill on the DMS because the DMS did not receive a definite response from the billing validation database.

Length: 1 byte

Format: Binary

Range: 0 = Validation Result NOT Determined by DMS Datafill
1 = Validation Result Determined by DMS Datafill

Example: Validation Result Determined by DMS Datafill

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

10.210 Validation Sequence Number

Description: Specifies which validation attempt this is for the call.

Length: 1 byte

Format: Binary

Range: 0 to 254

Definition: This field indicates which validation attempt this is for the call. This is set by the service node to help synchronize the validation response to a validation request. The switch only echoes this value back to the service node.

Example: Third validation for the call.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

10.211 Voice Channel ID

Description: Specifies the logical voice channel number for the voice connection between the switch and the service node.

Length: 2 bytes

Format: Binary

Range: 0 to 8191

Definition: This field specifies the voice channel that is selected for use during this session. The voice channels are data filled on a per node basis in switch Table OAVLMAP.

Example: Voice Channel ID is 725 (#02D5)

Hex	Binary								Byte #
	7	6	5	4	3	2	1	0	
02	0	0	0	0	0	0	1	0	0
D5	1	1	0	1	0	1	0	1	1

10.212 Year

Description: Specifies the current year.

Length: 1 byte

Format: Binary

Range: 0 to 99

Definition: The current year.

Example: The year is '95 (#5F)

7	6	5	4	3	2	1	0
0	1	0	1	1	1	1	1

Chapter 11: Error ID Values

Table 13: Non-Call Processing Class Error IDs

Error ID (#)	Value	Error Description
010A	Operation invalid for session pool type	This operation cannot be requested for the specified session pool due to the type of session pool.
010B	Invalid unsolicited operation	This operation cannot be requested by the service node.
010C	Invalid state	The service node or session pool is not in a valid state for this operation.
010D	Throttle request is greater than MAXSESN	The requested number of available sessions is greater than MAXSESN in Table OASESNPL. MAXSESN is the maximum number of available sessions for a session pool.
010E	Maintenance in progress	A maintenance activity is in progress.
010F	Drain command entered	The Drain command has been entered at the switch so this operation cannot be performed.
0110- DB #101	Maintenance operation failed	Text provides information as to failure encountered.

Table 14: Call Processing Error ID Values Which Do Not Result in Call Updates

Error ID (#)	Value	Error Description
0600	Database query in progress	An external database query is in progress.
0601	No switch resources available	There are no switch resources available to perform the request.
0602	Directory number connection in progress	The switch is attempting a requested directory number connection.
0603	Coin signalling in progress	The switch is waiting for coin signalling to complete while releasing a party.
0604	Operation not allowed for passive node	The operation is not allowed to be requested by a passive service node.
0605	Voice link request in progress for passive node	Voice connections are being made for a passive service node. The voice connections may involve OSAC processing. This error is returned to the active service node.
0606	Passive function provider request in progress	A function provider is being connected for a simultaneous interaction.
0607	Passive operator connection in progress	An operator is being connected (after the call was queued for one) for a simultaneous interaction.
0608	Operation is restricted for preprocessing session	The attempted operation is not valid for calls with a session type of Preprocessing.
0609	Operation is restricted for standard session	The attempted operation is not valid for calls with a session type of Standard.
060A	OSAC voice connection in progress	An osac voice connection is in progress. No other operations are allowed while the voice connections are being made with the OSAC Host.
060B	SOC option not turned on	The SOC option for the operation or a data block within the operation is not turned on at the switch.
060C	Invalid operation for calls involved in a conference	The specified operation is not allowed because the call is currently associated with a conference.
060D	Invalid operation for calls involved in a simultaneous interaction	The specified operation is not allowed because the call is currently involved in a simultaneous interaction.

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
0700	Cannot allocate call id	The switch could not allocate a callid for the new call that was originated by the service node.
0701	AMA buffer overflow	Not enough free space exists in the AMA buffer to do the requested Append AMA.
0702	Invalid generate AMA sequence number	The value of the generate ama sequence number field in the message is less than or equal to the value stored in the switch. An AMA record will not be generated.
0703	Session not available	This session is not available to be used for the call.
0704	Invalid append AMA sequence number	The value of the append ama sequence number field in the message is less than or equal to the value stored in the switch. The AMA module will not be appended.
0705	Invalid language	The language is not datafilled in switch table TOPSLANG and, therefore, is not valid.
0706	Invalid number type	The type of billing number is invalid for this call (e.g. class charge is set to collect and number type is set to account code). For a Directory Number request, this error indicates that a Directory Number Type of Foreign was improperly specified for the A party.
0707	Invalid number format	The format of the number is invalid for this call (e.g. it does not agree with the number type field).
0708	No DN digits entered	No directory number digits were sent for the specified connection.
0709	No calling number	A calling number is required for the class charge request.
070A	No forward number	A forward number is required for the class charge request.
070B	No billing number	A billing number was not received with the Billing Number Request operation. The billing number is required for the class charge request.
070C	Billing number is hotlisted	The billing number is hotlisted and cannot be used for billing.
070D	The A party is not connected	The A party has gone onhook and the service node has requested the call be floated.
070E	Invalid class charge	The class charge is invalid for this call.
070F	Sequence number mismatch	The sequence number sent with a float request does not match that of the sequence number of the last message sent by the switch.
0710	Invalid network service id	The network service ID is invalid.
0711	A subscriber party is not connected	There must be a party connected for this operation request.
0712	Call is restricted for this billing method	Billing method cannot be used for this call.
0713	Third number in B party connection	The B party connection is occupied by a third party. The class charge cannot be changed while a third number is connected to the call. Floating to a third party is not allowed. A DN entry cannot be processed for the B party at this time.

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
0714	800+ calls must be auto collect	The class charge for an 800 call should be auto collect.
0715	Call is set to be transferred	The requested operation cannot be performed because the call is set to be transferred.
0716	The A party has been released.	A new directory number cannot be entered for a call while the A party is in a released state (e.g. the service node has sent a Release DN for the A Party). The A party may, however, be reconnected, and once reconnected, the B party and Alt DN can then again be modified (if not already connected with an established DN).
0717	The B party is connected	The operation cannot be performed due to the B party being connected.
0718 - DB #12E	Billing details incomplete (Invalid Billing Details DB)	DMS restrictions are being applied and billing is not satisfied to perform the operation request.
0719	Invalid trigger profile	The trigger profile specified is invalid. The profile number must be datafilled in switch table OATPREFIX.
071A	Invalid treatment	The treatment specified is invalid.
071B	Invalid control list	The control list specified is invalid.
071C	No subscriber parties off hook	A subscriber party must be off hook for this operation request to be successful.
071D	Charge class invalid for call	The class charge is invalid for the particular call.
071E	Operation not valid for DA	The attempted operation is not valid for calls with a service type of Directory Assistance.
071F	Routing to carrier failed	The call could not be routed to the carrier.
0720	Invalid carrier	The carrier specified is invalid.
0721		
0722	Carrier served by switch	The switch is providing complete operator services for the call. The call cannot be transferred to the carrier for service.
0723	Screening Failed	The billing number or connecting number failed screening checks.
0724	Call is setup to be transferred to a carrier	The directory number cannot be changed since the call is ready to be transferred to a carrier.
0725	Forward agent is present for BLV and overseas calls	The directory number for the forward party cannot be changed because the forward party is present.
0726	The B party is on-hook	The B party cannot be on-hook when a call float with the "parties already connected" option (Call Float Indicator) is requested. Call float with the "connect parties" option should be requested if the B party is on-hook.
0727	The B party is not connected	The B party is not connected and a call float with the "parties already connected" option (Call Float Indicator DB) has been requested. Call float with the "connect parties" option should be requested if the B party is not connected.
0729	The B party is off-hook	The operation requested is disallowed after the B party has answered.

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
072A	Carrier not served by switch	The call cannot be completed by the switch because the switch is not providing billing for the call. The call should be transferred to a carrier for service.
072B	Carrier not allowed for call	This carrier cannot be changed for a DA serve call, the primary carrier cannot handle the call and an alternate carrier is not available, or the carrier cannot terminate to certain areas.
072C	Invalid voice channel id	The voice channel ID specified is invalid.
072D	Voice channel already in use	The voice channel ID specified is already in use.
072E	Voice channel seize failure	There was a failure when seizing the voice channel that was specified.
072F	There is no voice connection for this session	There is no voice connection for this session and therefore the operation cannot be performed.
0730 - DB #127	There is already a voice connection for this session (Voice Channel DB)	A voice connection currently exists for this call.
0731	Party other than A Party is connected	Call cannot be routed to treatment if parties other than the A Party are still connected.
0732	The A Party is on-hook	Call cannot be floated if the A Party is on-hook.
0733	Party already connected	A party for this connection is already established.
0734	Translations failed	DMS translations tables do not result in a valid destination DN. Also used if an Automatic Carrier Selection call is blocked in switch table CCCSOPTS.
0735 - DB #121	Translations failed, Treatment applied (Treatment DB)	DMS translations tables do not result in a valid destination DN. A treatment was set and the switch has applied the treatment.
0736 - DB #121	Outpulsing failed, Treatment applied (Treatment DB)	There is a problem reaching this particular number (e.g. vacant number). The switch has applied a treatment.
0737 - DB #11C	Release with Cause, Treatment applied (Release Cause DB, Treatment DB)	An SS7 Release Cause has been received in attempting a connection. Treatment has been applied. The cause value should be interpreted per Bellcore GR-317-CORE.
0738	Outpulsing failed, No treatment applied	A problem with the connection resulted in the DMS being unable to outpulse the requested DN. No treatment applied.
0739	Auto collect allowed only for 800+ calls.	The service node attempted to assign a class charge other than auto collect to an 800+ call. Or, the SN is class charging a non-800 call as auto collect.
073A	No DN associated with connection	The Directory Number digits for the connection have not been entered. The DN must be entered prior to connection manipulation.
073B	Conversation Timing In Effect	The class charge request cannot be processed because the DMS has already begun timing the call.
073C	Requested party not in forward port	The service node attempted to release the B party when a special number was in the forward port, or attempted to release the special number when a B party was in the forward port.
073D	Carrier ID is restricted	The call is restricted from going to an interlata carrier.

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
073E	Carrier not specified for call	A carrier has not been specified for this call and the operation requires a carrier.
073F	Timing has started	Timing has started on the call and the operation cannot be performed.
0740	Incoming trunk group not supported for transfer	The incoming trunk group does not support transferring a call to a carrier trunk.
0741	A passive function provider must be connected or the call must be in queue for a passive function provider.	This operation requires a passive function provider.
0742	Unknown or invalid function	The specified function is not valid at the switch. Parallel datafill for function ids are required.
0743 - DB #13A	Queue overflow	The maximum number of calls that can be queued for this function is already queued. A new call cannot be added to the queue.
0744 - DB #13A	Queue deflection	The predicted wait time to obtain a function provider is too long.
0745	Other failure in routing or queuing	An internal error has occurred obtaining a function provider or there are too many calls in queue for OSSAIN processing.
0746	Passive function provider already connected or call is already queued for a passive function provider	At most two function providers can be connected to a call in a simultaneous interaction. Another function provider cannot be added.
0747	Conversation time exceeds notification time	The current elapsed conversation time exceeds the conversation notification time requested by the Trigger Profile Index DB.
0748	A passive SN must be connected	This operation requires a passive service node to be connected in a simultaneous interaction for the operation to be successfully processed.
0749	LNP information not available	The switch could not obtain the LNP information for the specified directory number.
074A	Host-Remote voice channel not available	A host-remote voice channel is not available to make voice connections for this OSAC call.
074B	No response from OSAC Host	The OSAC Remote was waiting for a acknowledgment from the OSAC Host before successfully proceeding with the operation request. However, the switch timed out waiting for the response.
074C	No host-remote session available	An acknowledgment is required from the OSAC Host for the operation to be successfully processed. However, there are no sessions available to message the OSAC Host.
074D	Session pool datafill mismatch	The session pool datafill in Table OASESNPL between the OSAC Host and OSAC Remote does not match.
074E	Function datafill mismatch	The function datafill in Table OAFUNDEF between the OSAC Host and OSAC Remote does not match. This could also be returned if there is datafill mismatch between Table OAFUNDEF and OASESNPL (e.g. function is datafilled as centralized and the session pool used for session initiation is datafilled as standalone).

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
074F	Invalid service provider id	The service provider specified is not datafilled in switch Table SPID.
0750	Conversation timing cannot be performed for 'No AMA' call	Conversation notification timing, as indicated by the trigger profile index, cannot be performed on calls marked as 'No AMA'.
0751	Call merge in progress	The service node requested an operation while a call merge is in progress.
0752	Unknown or invalid conference id	The conference id was unrecognized by the switch
0753 - DB #14B or DB #14D	Voice entity already connected to a conference	The SN has requested to add a voice entity to a conference when that voice entity is already connected to a conference. DB #14B is returned if the voice entity is a subscriber party. DB 14D is returned if the voice entity is a SN voice link.
0754 - DB #10D or DB #14D	Voice entity not connected to a conference	The SN has requested to remove a voice entity from a conference when that voice entity is not connected to a conference. DB #10D is returned if the voice entity is a subscriber party. DB 14D is returned if the voice entity is a SN voice link.
0755 - DB #14A	Conference already exists	The SN attempted to create a conference when one already exists. Before a new conference can be created, the previous conference must be ended by a sending a Conference Release Request.
0756	Too many voice entities in conference	The conference has reached the maximum size. Another voice entity cannot be added until one of the existing conferees is released.
0757	The B party of the specified call is not connected and offhook	The operation cannot be performed because the B party of the specified call is not connected and offhook.
0758	The specified call already has an A party connected	The operation cannot be performed because the A party of the specified call is already connected.
0759	Unknown or invalid callid	The callid was not found to be currently in use by the switch
075A	Notification timer not running	The SN attempted to stop the notification timer when it wasn't running.
075B	Session is not idle	The operation could not be performed because it was not received on an idle session.
075C	Session is not used for SN origina-tions	The operation could not be performed because it was not received on an SN originated session.
075D - DB #150	Target call already has a session with an SN	The SN attempted to issue a Session Recall Request for a call that already has a session with an SN.
075E - DB #14A	Target B party is connected to a different conference than the A party conference	The SN requested to merge the A party from one conference with the B party which is connected to a difference conference.
075F	Conversation timing not started	The SN attempted to perform an operation involving conversation timing.
0760	Previous class charge information cannot be set by the service node	The SN attempted to set the previous class charge information. This information is only set by the switch.

Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
0761	SPID processing not enabled	The SN attempted to update SPID information for a DN when SPID processing is not enabled at the switch for the DN.
0762 - DB #155	Attempt to exceed max connections for broadcast voice link	The maximum number of connections for the requested broadcast voice link, defined in switch table OAVLMAP, would be exceeded if this operation were processed.
0763	Voice link cutthru method invalid for cycle counting	The voice cut through method defined in switch table OAVLMAP is not valid for counting cycles. The cutthru method should be set to HKCHG for cycle counting.
0764	Invalid broadcast configuration	Connection to the requested voice link (either standard or broadcast) would result in an unsupported broadcast configuration.
0765		
0766	Not a broadcast voice link	The service node requested a broadcast voice connection with a voice link that is not datafilled in the switch with broadcast capabilities. Check datafill in table OAVLMAP.
0767	OSAC Host SOC not on	The SOC option necessary to perform the requested operation is not on at the OSAC Host switch.
0768	Voice link for broadcast only	The service node requested a regular voice connection with a voice link that is datafilled in the switch as a broadcast voice link. Check datafill in table OAVLMAP.
0769	New service is the same as the old service	The service node requested a service change but the new service type is the same as the current service type of the call.
076A	Invalid trunk group and member	The trunk information is not datafilled in table TRKMEM.
076B	Invalid Country Direct Operation	The operation is not allowed on a Country Direct Call (Call Origination Type = Country Direct)
076C	Invalid Routing Index	The specified Routing Index is not present in table OAIN-ROUTE or maps to an OFRx office route which either does not exist or is in the specified OFRx table as a non-RTE tuple.
076D	North American-only functionality	The requested functionality is only available in North American switch loads.
076E	Connection ID - Party mismatch	The Connection ID in either the Routing Number or Routing Index DB does not match the party associated with the operation.
076F	SN Voice Link datafill missing	The service node requested the switch to select and connect to a voice link but the voice link datafill is missing from table SNVLGRP.
0770	Estimate of Charges Failure	The request for an estimate of charges has failed. This may be due to incomplete internal rating datafill, or if the call is using external rating.
0771	Invalid Estimate Duration for Numbering Plan	This will result if the estimate duration is out of range for the numbering plan. For North American numbering plan, the estimate duration range must be between 1 and 999 minutes.
0772	Invalid Estimate Date	This will result if the month and day received from the service node for the estimate is an invalid date (i.e, Feb. 30).

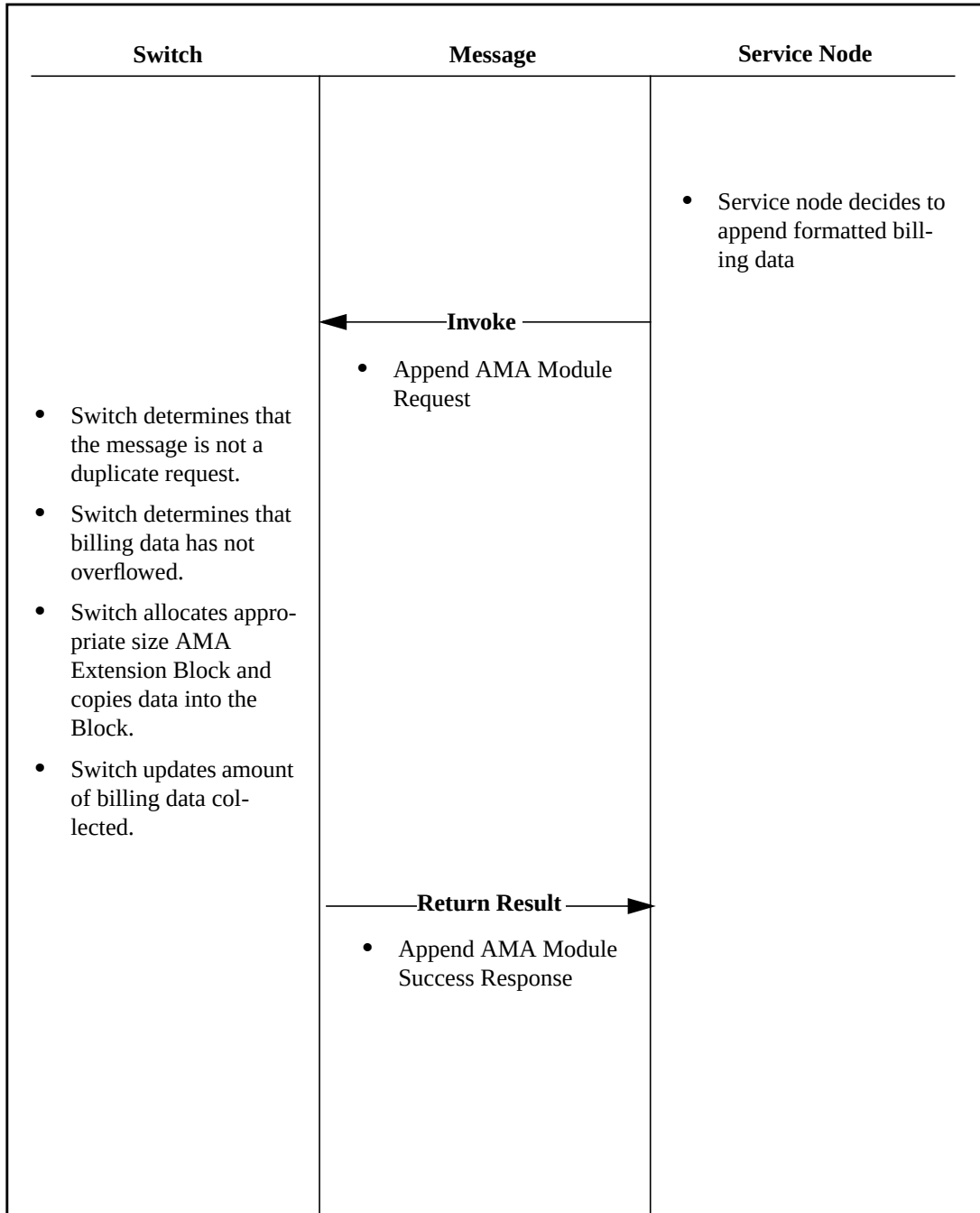
Table 15: Call Processing Error ID Values Which May Result in Call Updates

Error ID (#)	Value	Error Description
0773	OSAC release too low	The version of OSAC protocol is too low to process the operation
0774	Class charge missing but required	The requested operation is not allowed without the call having a valid class charge.

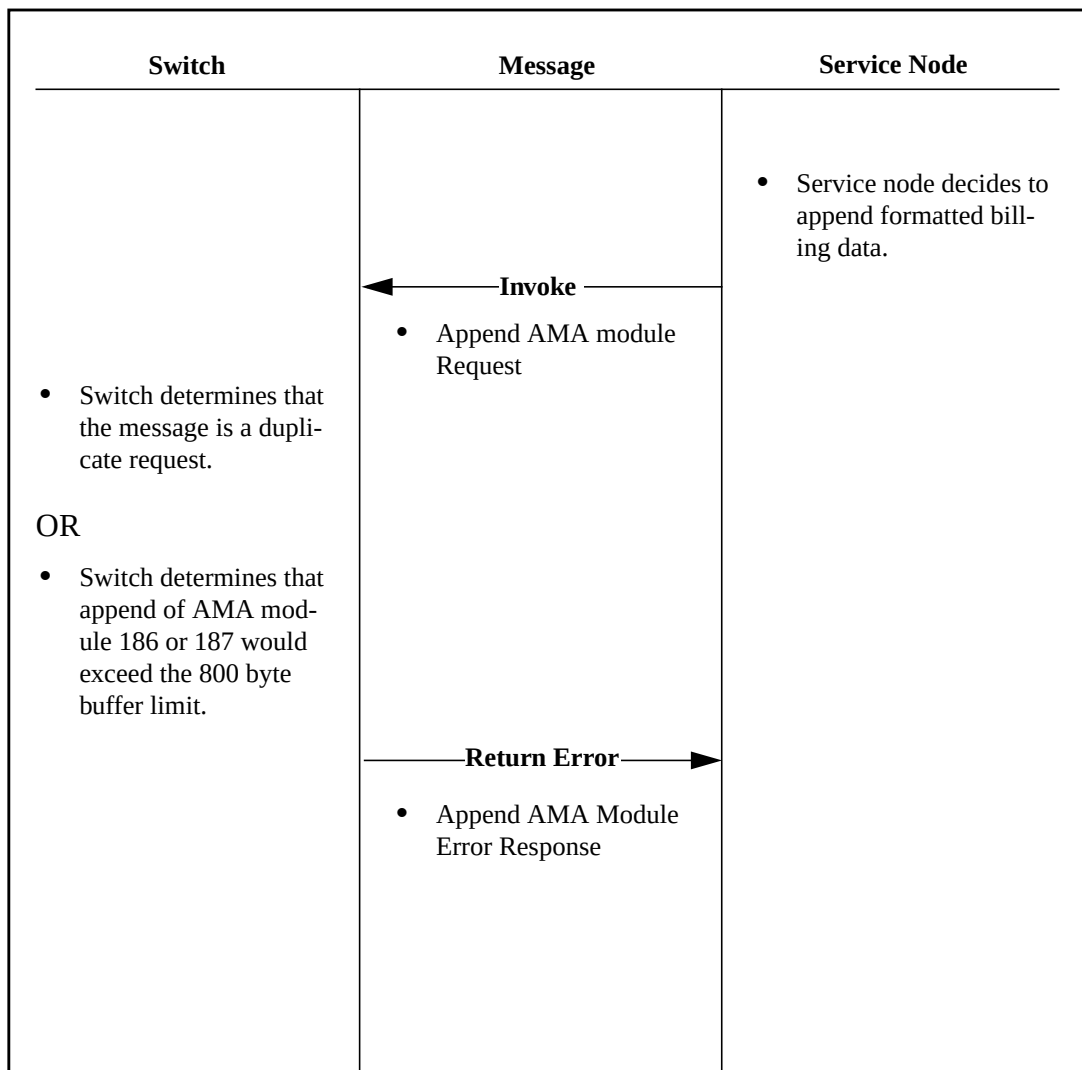
Chapter 12: Message Flows

12.1 AMA Message Flows

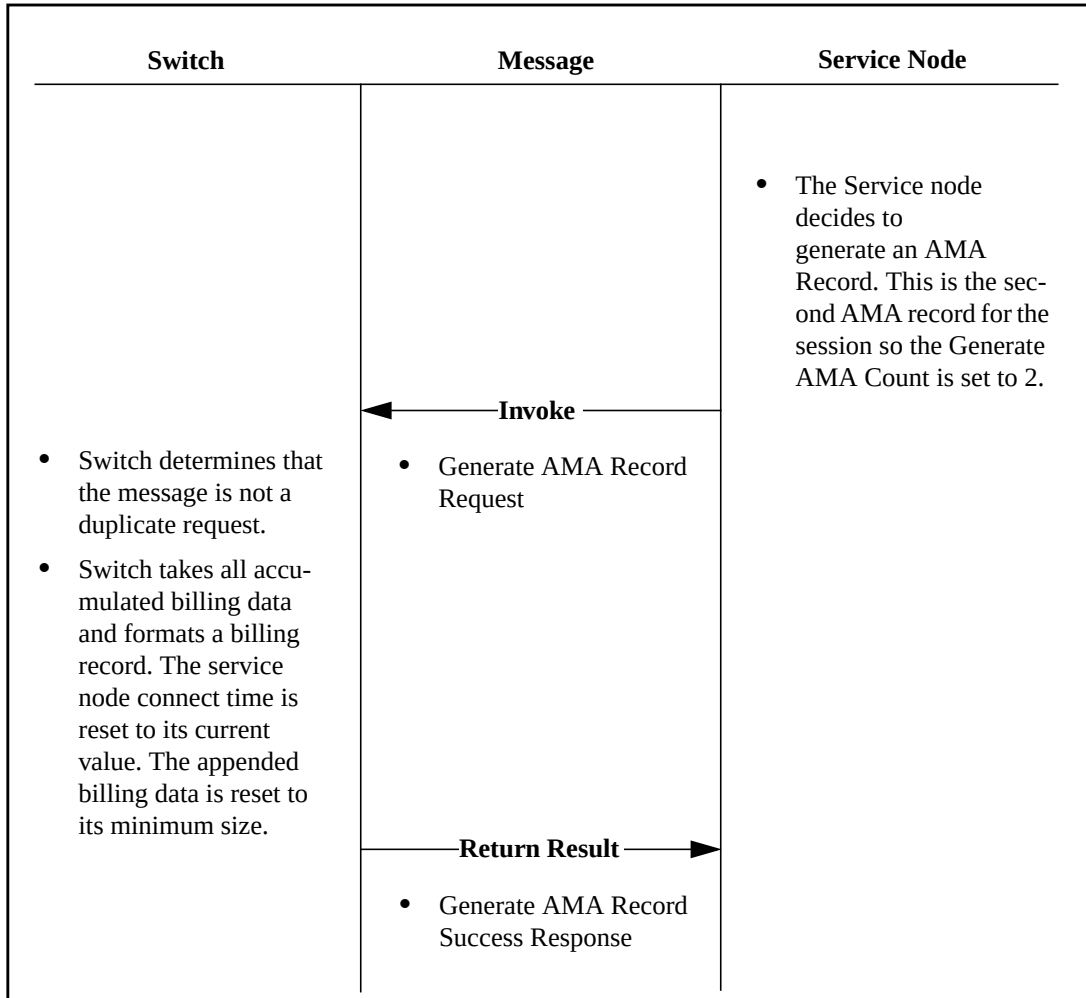
12.1.1 Append Billing Data (Success)



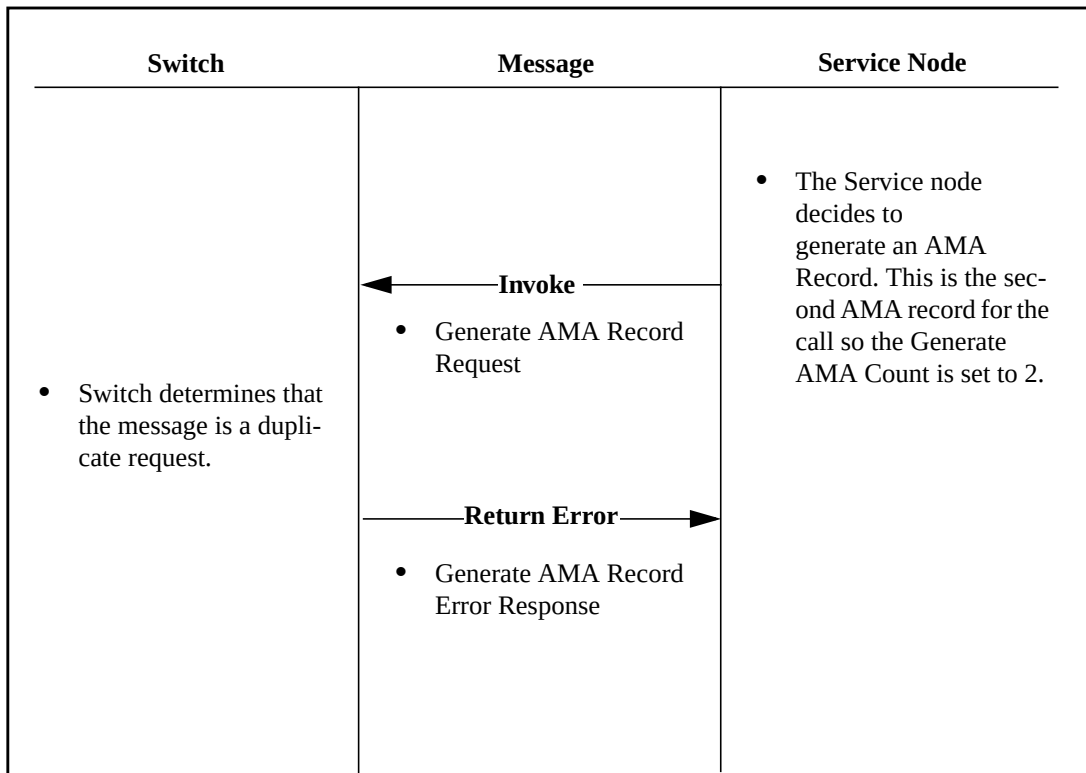
12.1.2 Append Billing Data (Failure)



12.1.3 Gen AMA (Success)

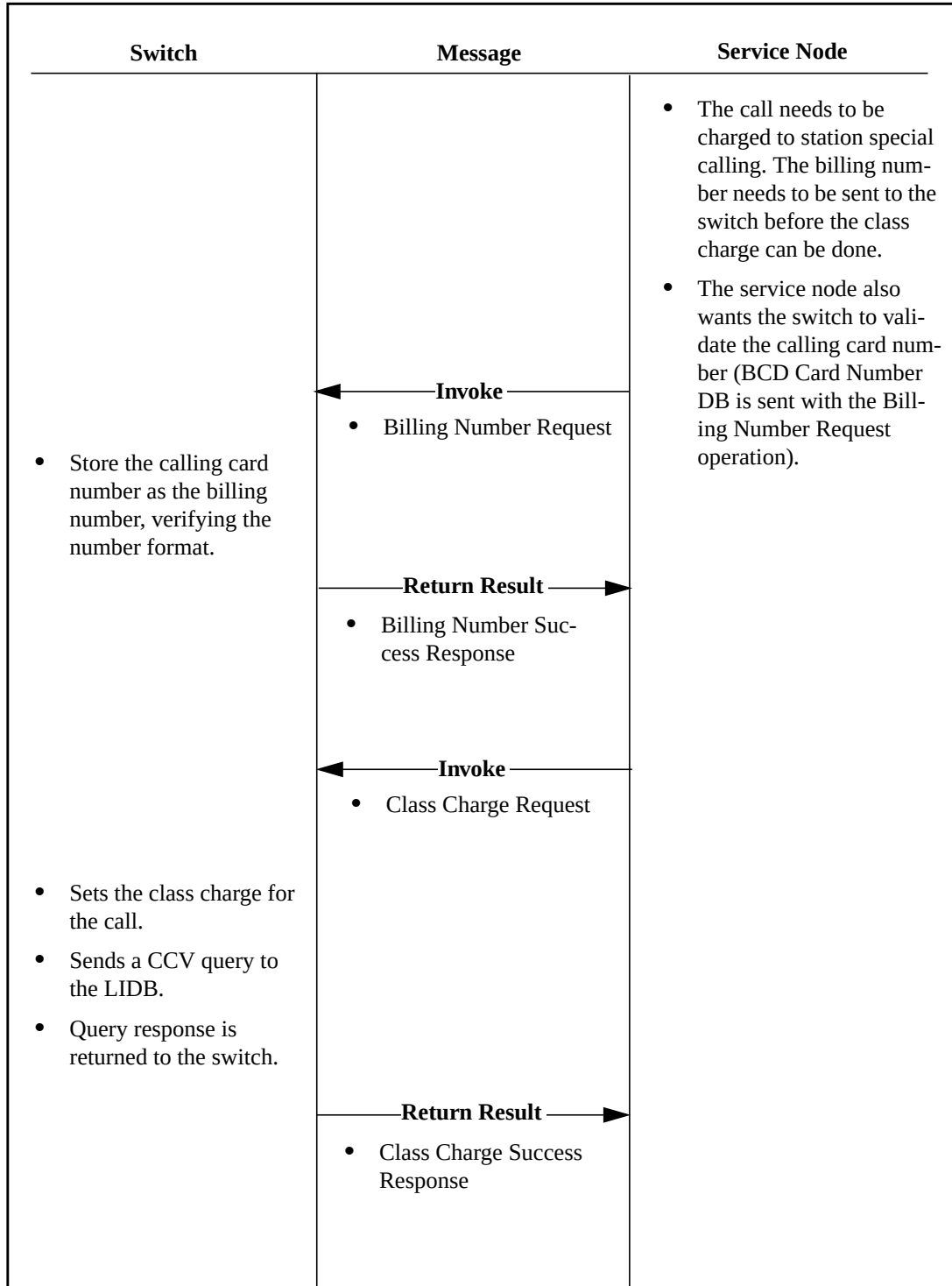


12.1.4 Gen AMA (Failure)



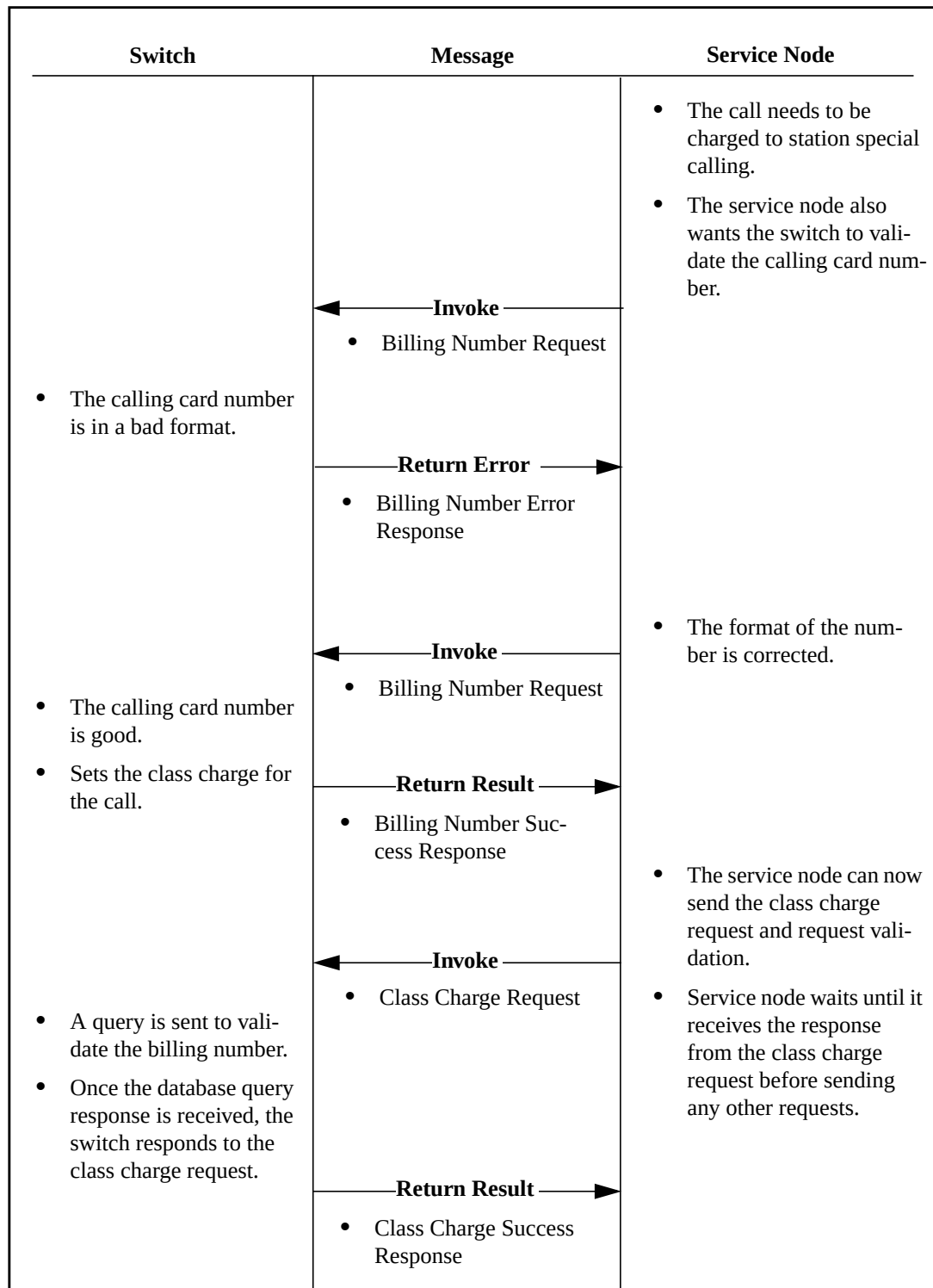
12.1.5 Billing Number and Class Charge Request Call Flow (Success)

The service node needs to class charge the call and have the switch validate the billing number.



12.1.6 Billing Number and Class Charge Request Call Flow (Failure)

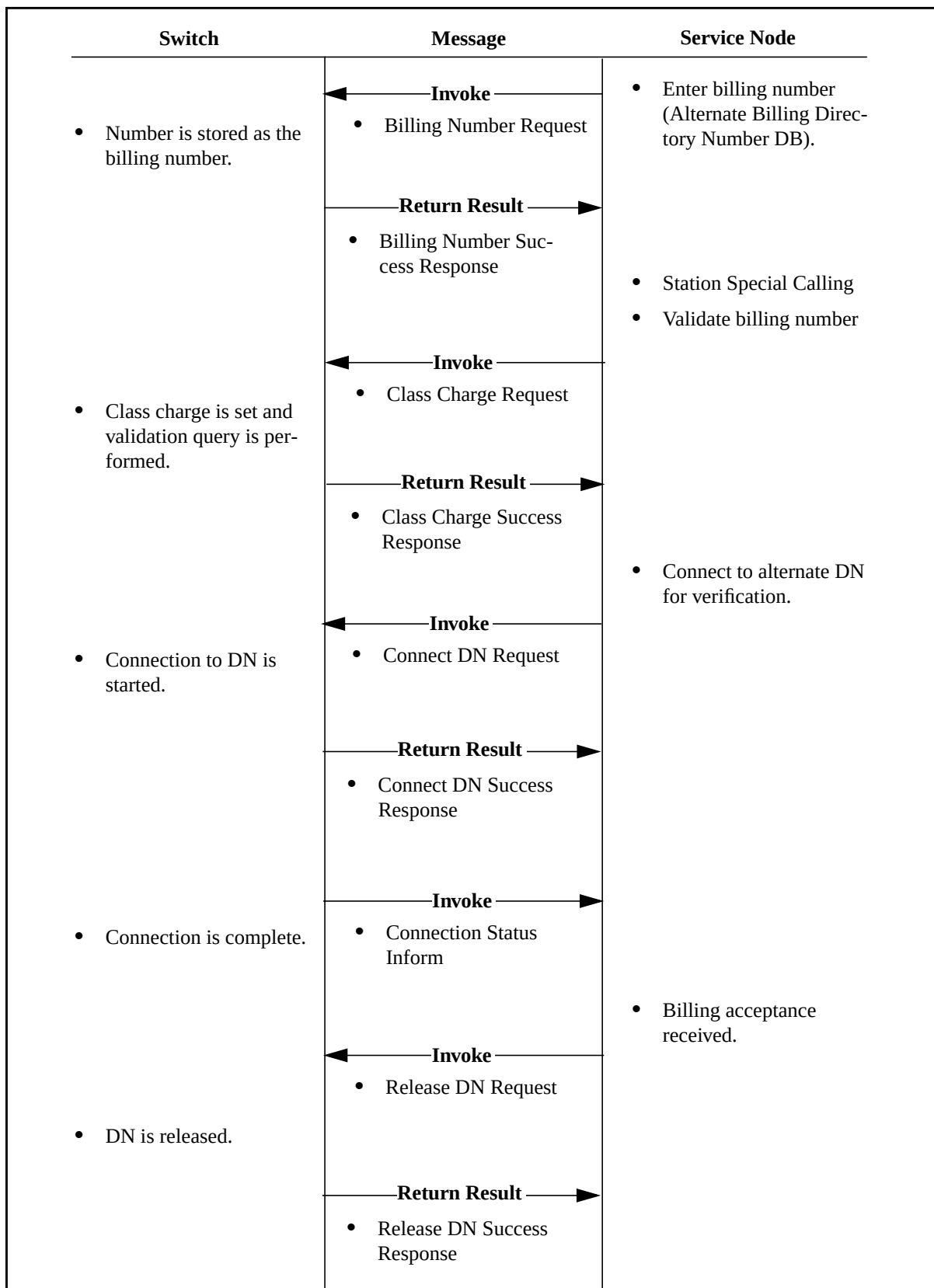
The service node needs to class charge the call and have the switch validate the billing number. However, the billing number is in a bad format.



12.1.7 Bill to Alternate Directory Number, Verification Required

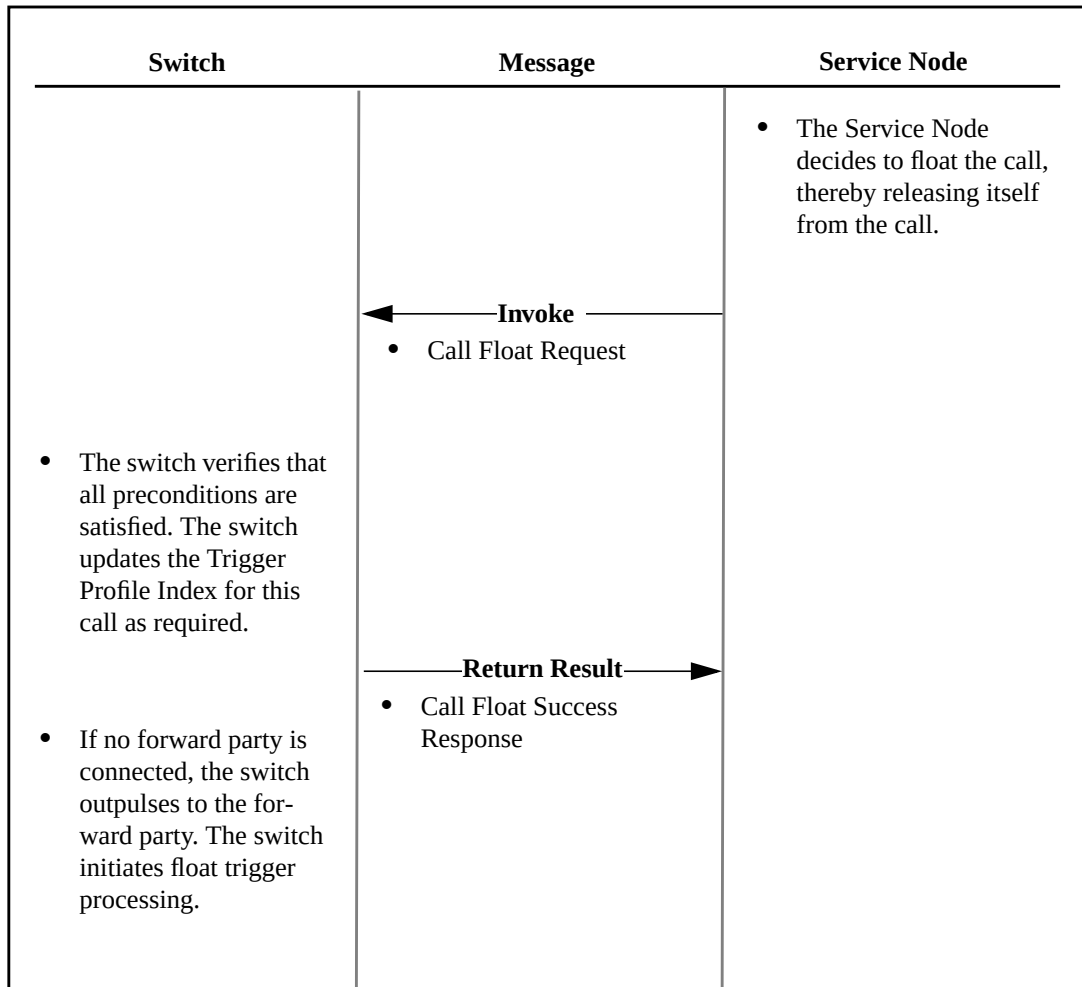
The service node wants the switch to validate the billing number. To do this, it sends the billing number, which is in a directory number format, to the switch. It then sends a class charge request to set the class charge to station special calling. In the class charge request, validate billing number is set to true, which informs the switch to validate the billing number. The switch sends the BNS query to the LIDB.

Once the billing number is successfully validated, the switch returns the query result to the service node. The query result indicates that billing acceptance is required. The service node sends a request to connect to the directory number (i.e. the billing number). Once verification is received, then the service node sends a release request.

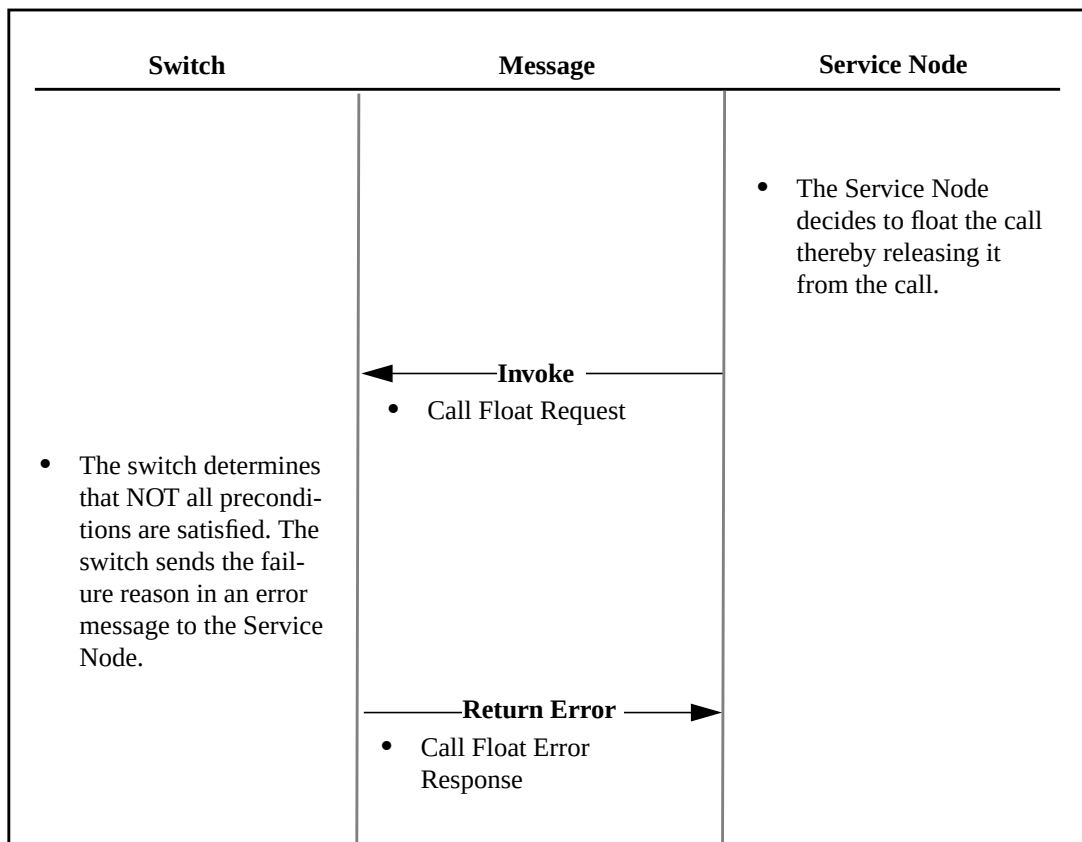


12.2 Call Float Call Flows

12.2.1 Call Float (Success)



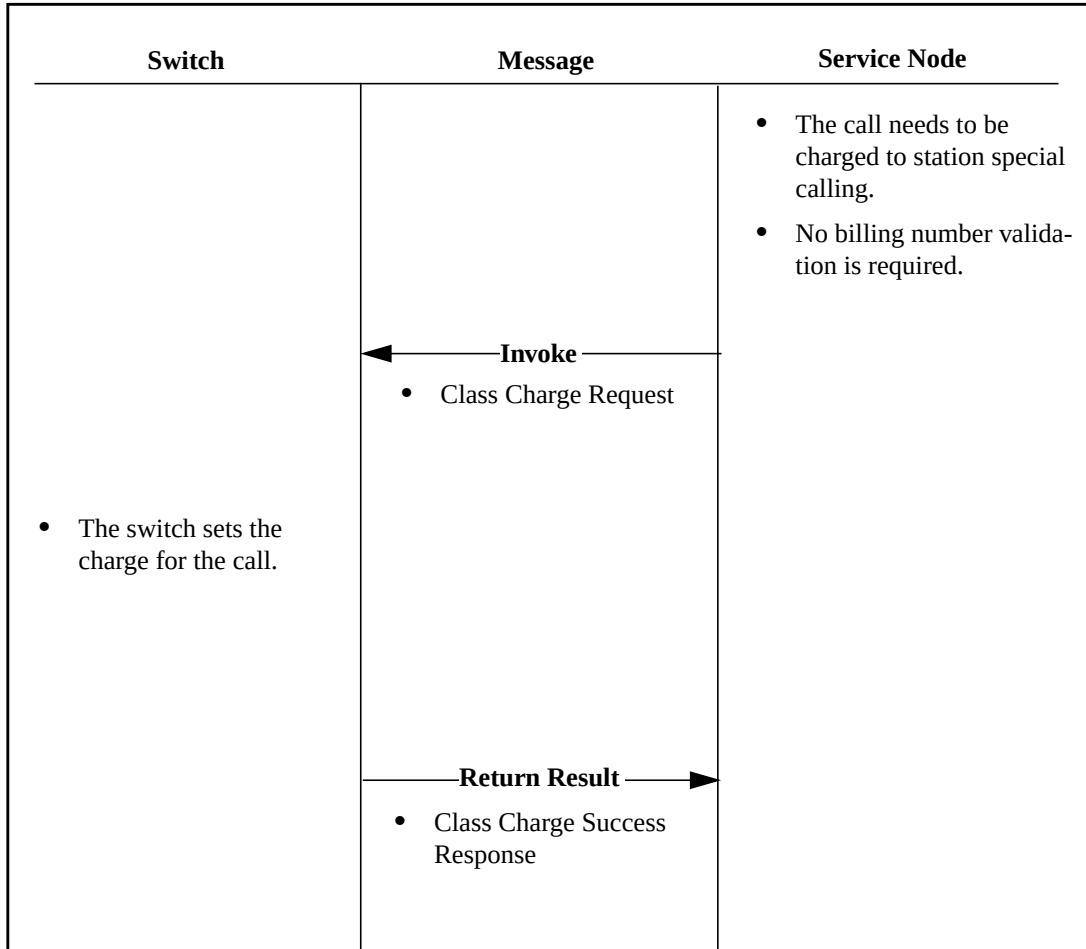
12.2.2 Call Float (Failure)



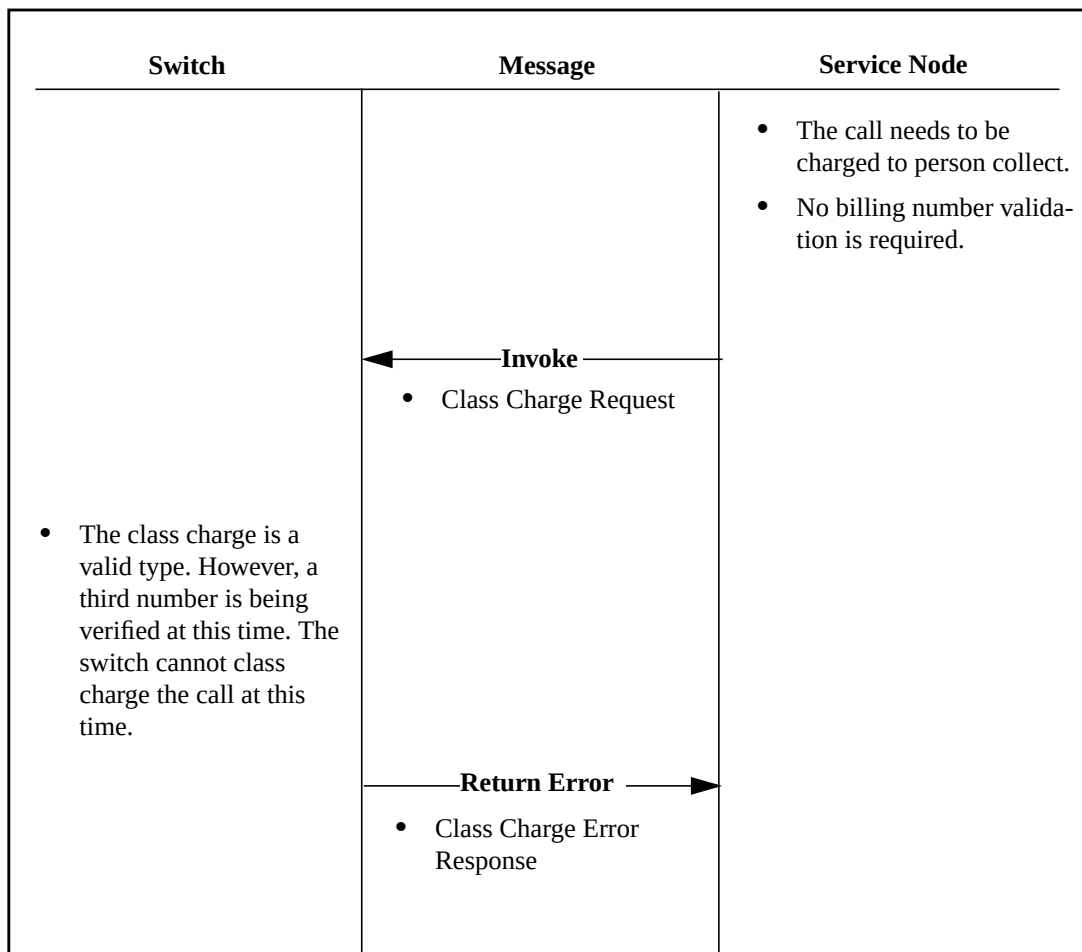
12.3 Class Charge Call Flows

12.3.1 Class Charge (Success)

The service node needs to class charge the call with the switch.



12.3.2 Class Charge (Failure)



12.4 Conferencing

This section contains example call flows that illustrate one way that the new OAP conferencing operations can be used. In order to fully understand the example call flows, this introduction gives a brief description of the example functionality from both a subscriber and an OAP standpoint.

The discussion of conferencing begins with a floated two-party call. Presumably, the SN would have floated this call with a trigger profile index that contains a trigger (perhaps the '*' or '#' key in the talking state) that indicates a subscriber's wish to initiate a 3-way conference call. When the subscriber initiates the conferencing trigger, the call is presented to the SN (via Session Begin Inform) for processing. The first of the example conferencing message flows will pick up the call at this point.

Once the two-party call triggers back, the SN will place the B party on hold (via Speech Path Request) so that it may communicate with the A party. In order to obtain the DN digits for the third conferee, the SN will connect a voice link to the call (via Voice Connect Request) and prompt the subscriber for the digits. While the interaction with the subscriber is taking place, the SN proceeds to create the conference (via Conference Create Request) and simultaneously add both the A and B parties as well as the SN voice link to the newly created conference.

Once the third conferee's digits are obtained, the SN can proceed to initiate a new call (via Session Initiation Request) that will contain the third conferee (hereto referred to as the 'C' party, although it is more precisely known as the B party of another call). As part of the Session Initiation Request, the SN will have to provide the billing information for the A to C connection. It should be noted that two separate billing records will be generated for this conference, one for the A to B connection (AMA from call containing A and B parties) and one for the A to C connection (AMA from call containing C party).

Finally, the SN instructs the switch to connect the C party (via Connect DN Request). As part of the request, the SN instructs the switch to place the C party in the conference so that the A party can hear the connection progress (e.g. ringing or busy). After answer, the A and C parties are talking. This is known as the **consult** phase of the conference call.

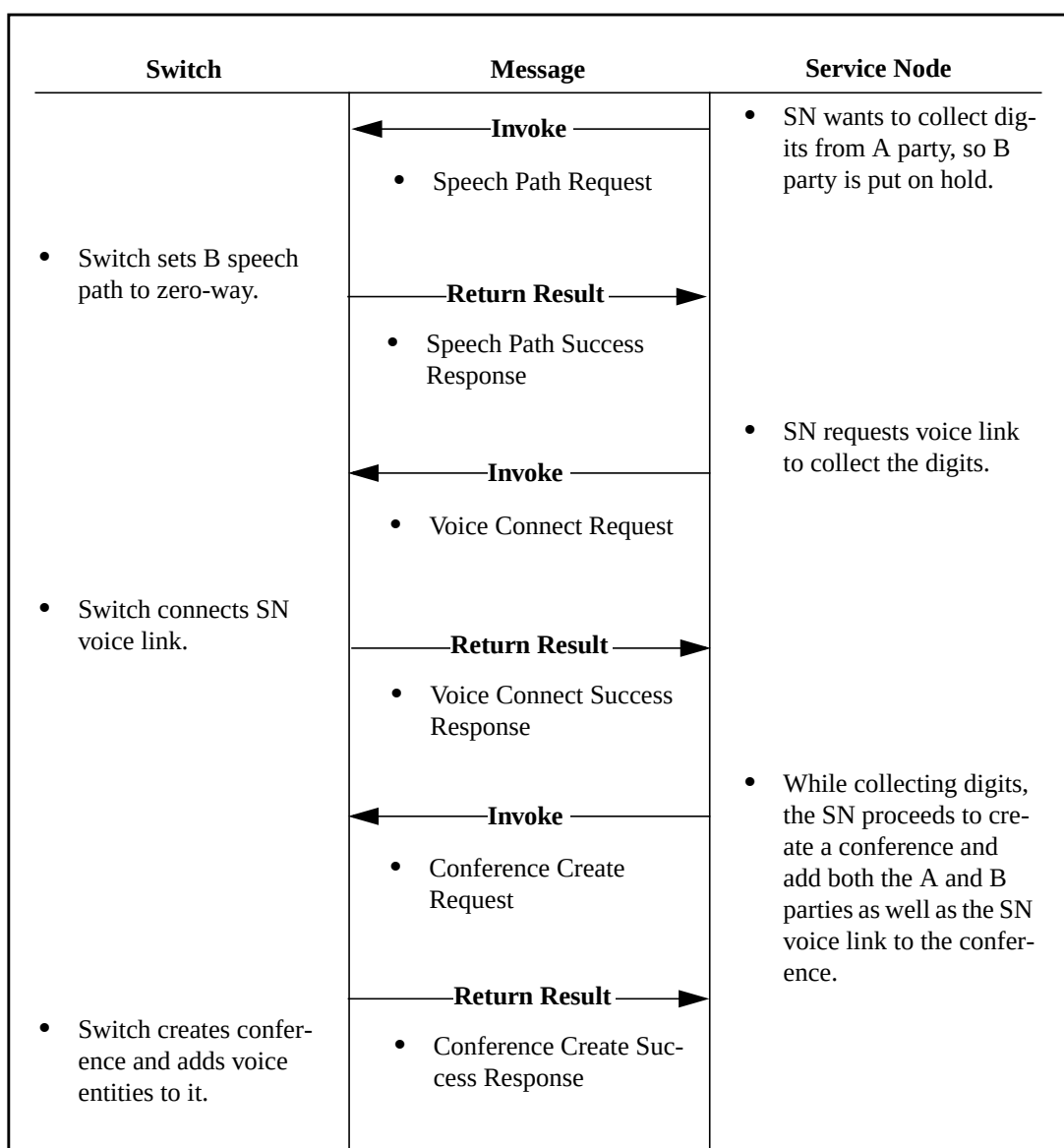
The consult phase continues until the A party signals (via DTMF, e.g.) to have the B party added into the conference. At this point, the SN floats both calls (via Call Float Request in each call), and speech paths are restored to two-way on float. The period of time after the call is floated is known as the **conference** phase of the conference call.

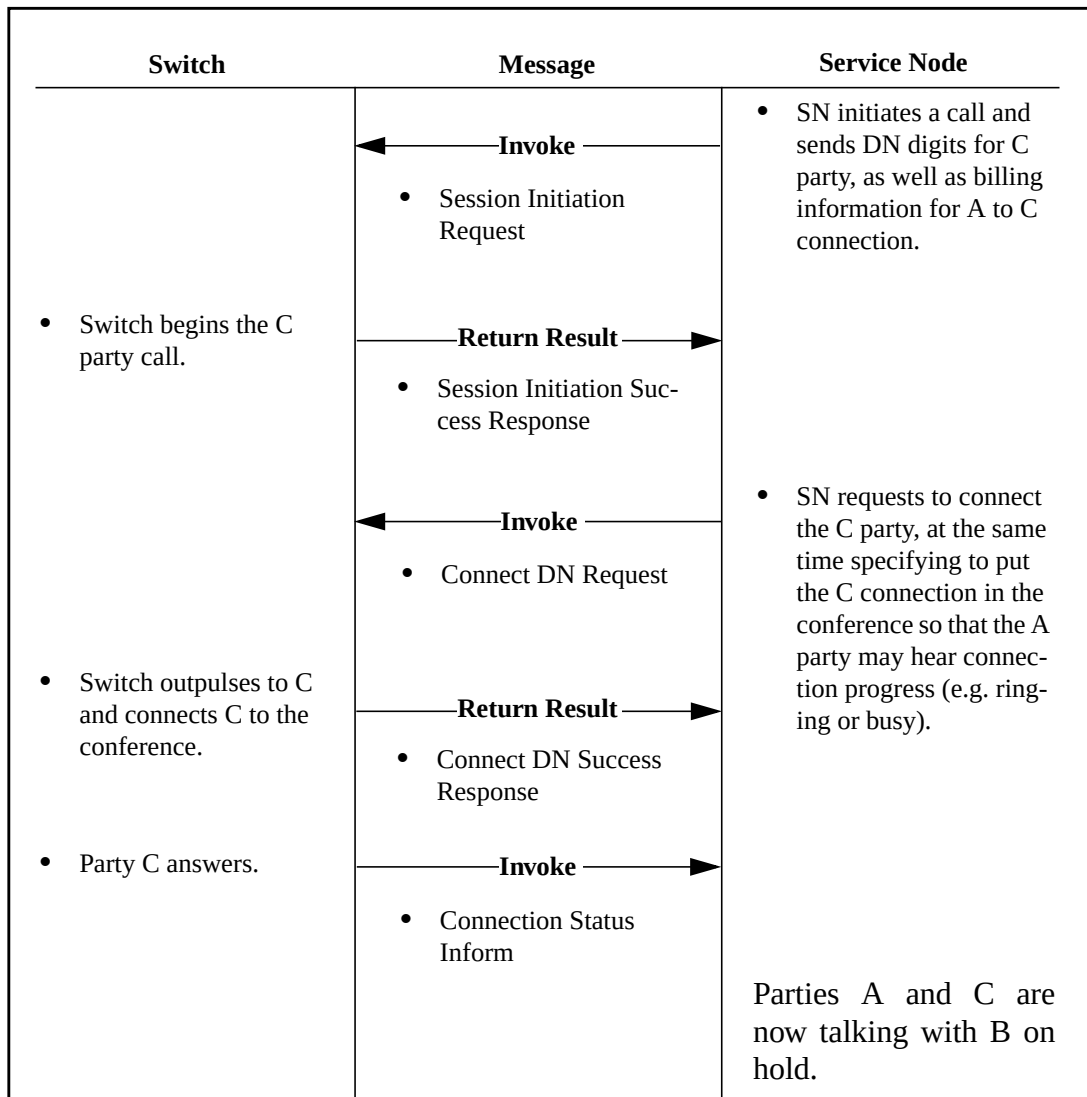
The following example flows will cover success paths for both the consult and conference phases; as well as example flows for A, B, or C disconnects during each phase.

12.4.1 Consult Phase

This flow assumes that a two-party call has already triggered back to SN after the subscriber initiated a trigger for a 3-way conference call. The flow picks up after the Session Begin Inform has been sent and continues through the consult phase of the conference, just before both calls are floated to start the conference phase.

Since this flow includes two different calls (one for the A and B parties, one for the C party), it has been divided into two flows. The first flow corresponds to the session with the A/B call and the second flow to the session with the C call. It should be noted that the first flow would occur sequentially before the second flow since the interaction with the A party must occur before the call for the C party can be created.

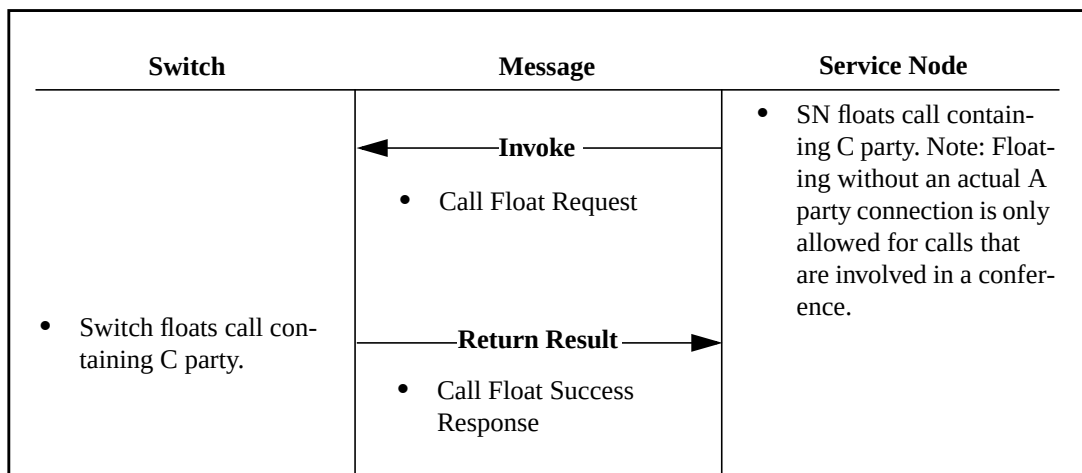
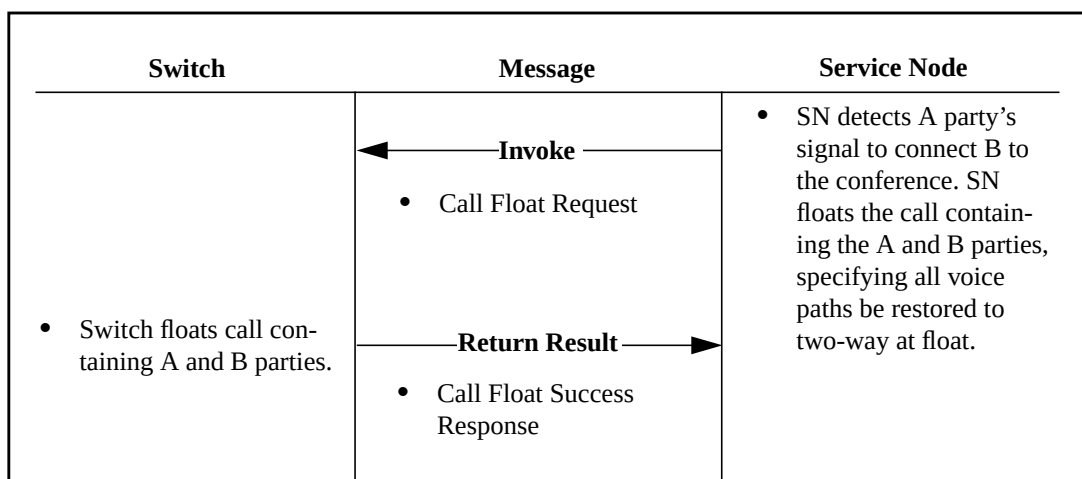




12.4.2 Conference Phase

This flow picks up where the last flow left off. At the end of the consult phase, the A and C parties were talking with the B party on hold. Both the call containing the A and B parties and the call containing the C party have sessions with the SN. The conference phase is started by the subscriber's signal to add the B party to the conference (via DTMF, e.g.). At that point, the SN proceeds to float both calls with all speech paths restored to two-way.

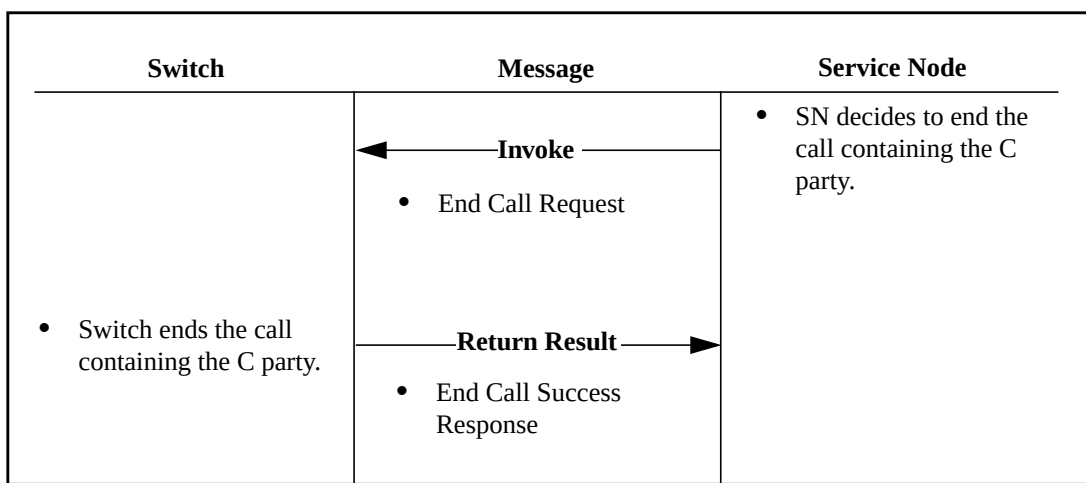
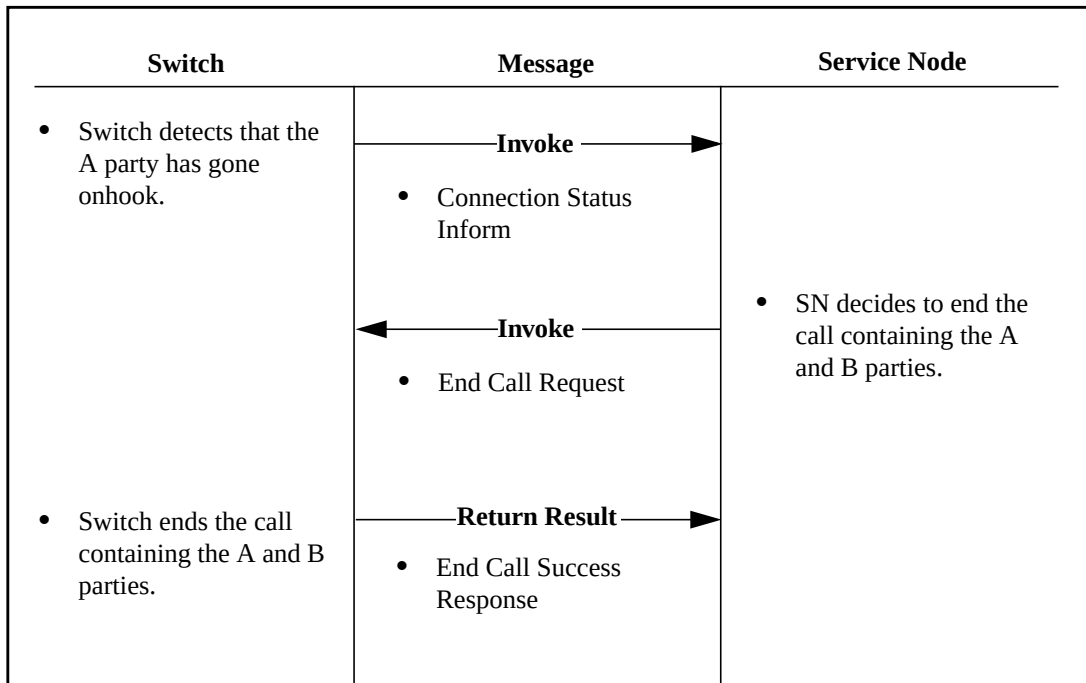
As with the previous flow, this flow also involves two calls. Again, the first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. However, in the overall conference phase flow, both floats could be occurring simultaneously.



12.4.3 Party A disconnects during consult phase

This flow depicts an A party disconnect at any point during the consult phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to end both calls involved in the conference as a result of the A party disconnect.

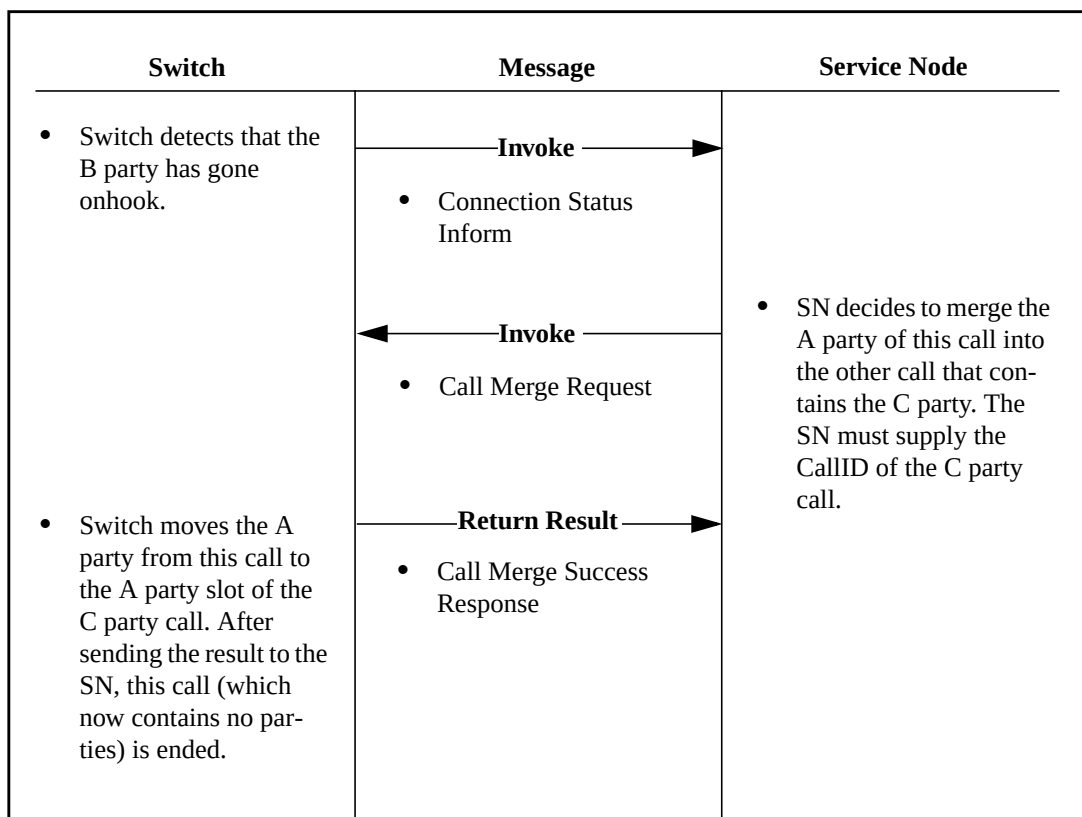
As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for an A party disconnect during consult, both calls could be ended simultaneously.

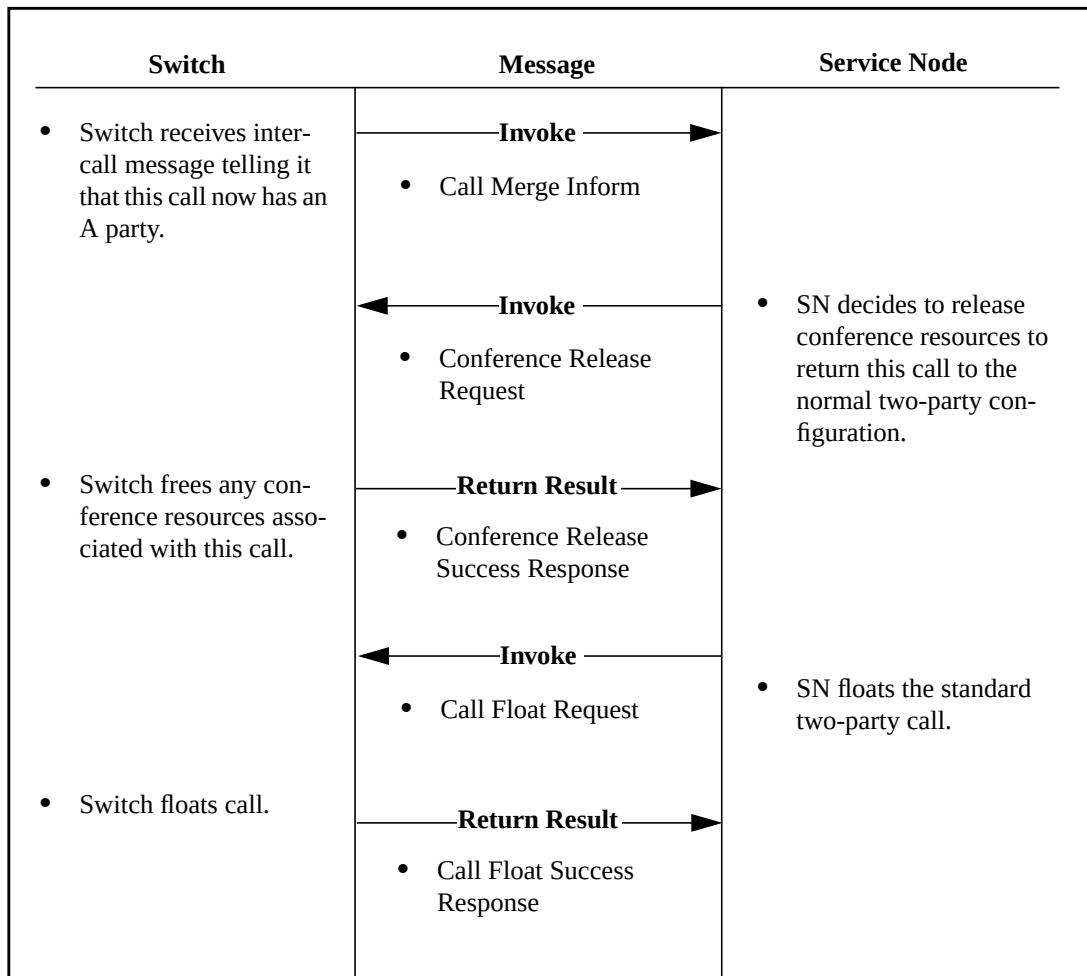


12.4.4 Party B disconnects during consult phase

This flow depicts a B party disconnect at any point during the consult phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to merge the call containing the A party into the call containing the C party. After merging the calls, the call now containing both the A and C parties is floated as a standard two-party call.

As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A party and the second flow corresponds to the session with the C party call. In the overall flow for a B party disconnect during consult, the flow for the A party call would occur before the flow for the C party call.

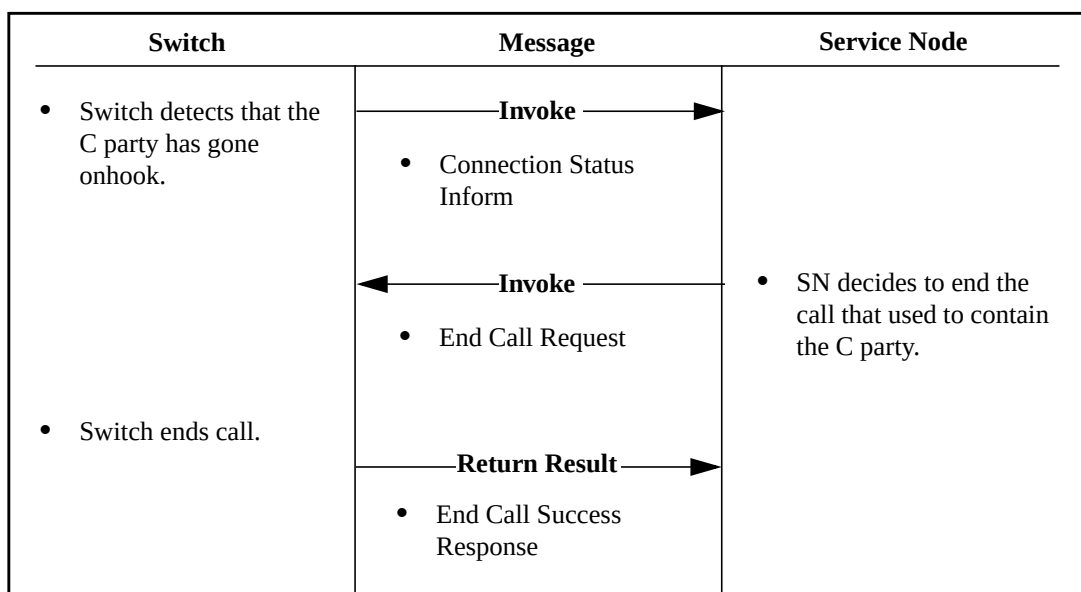
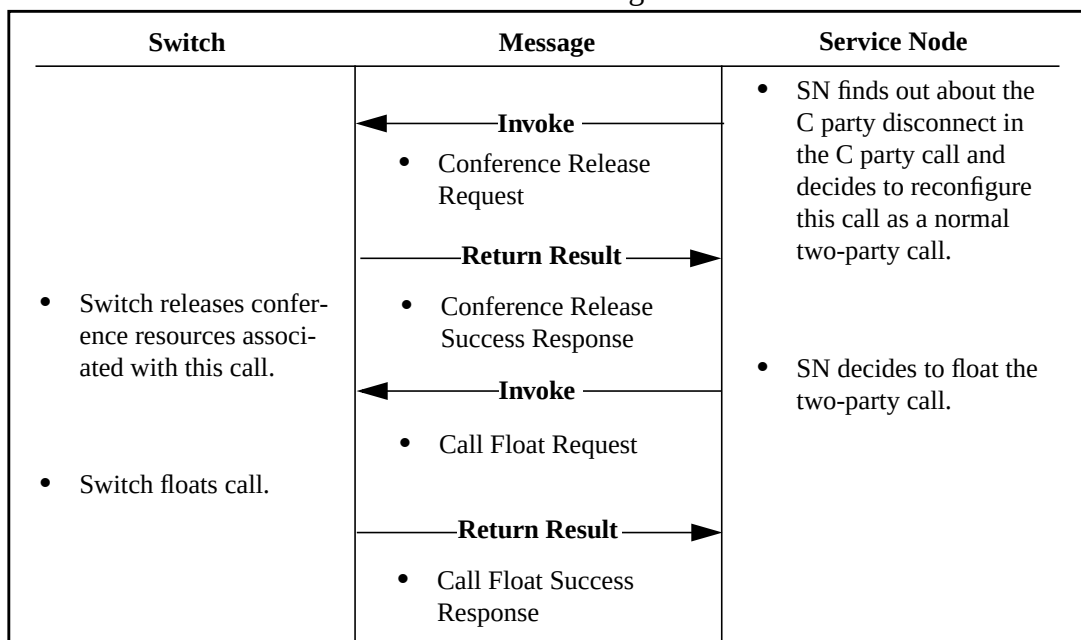




12.4.5 Party C disconnects during consult phase

This flow depicts a C party disconnect at any point during the consult phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to end the call containing the C party and float the call containing the A and B parties as a normal two-party call.

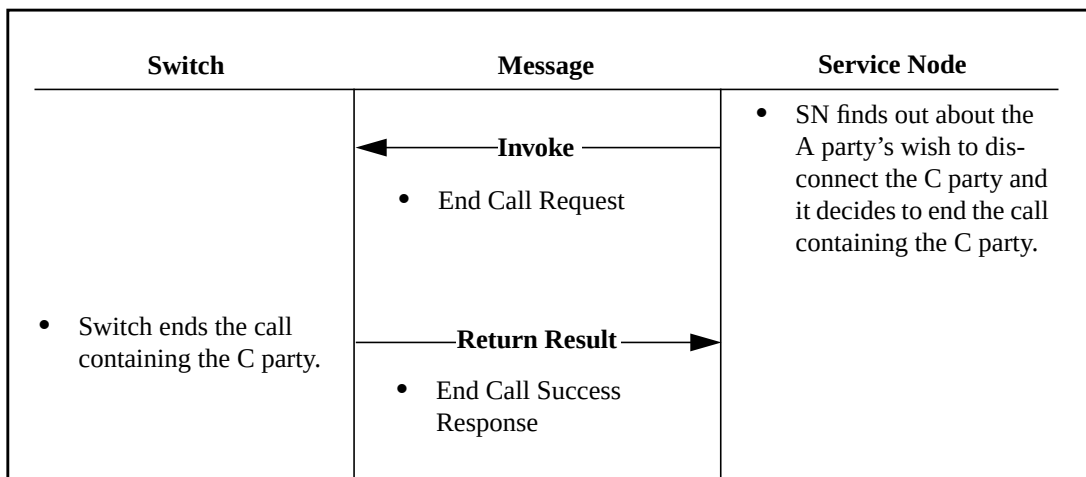
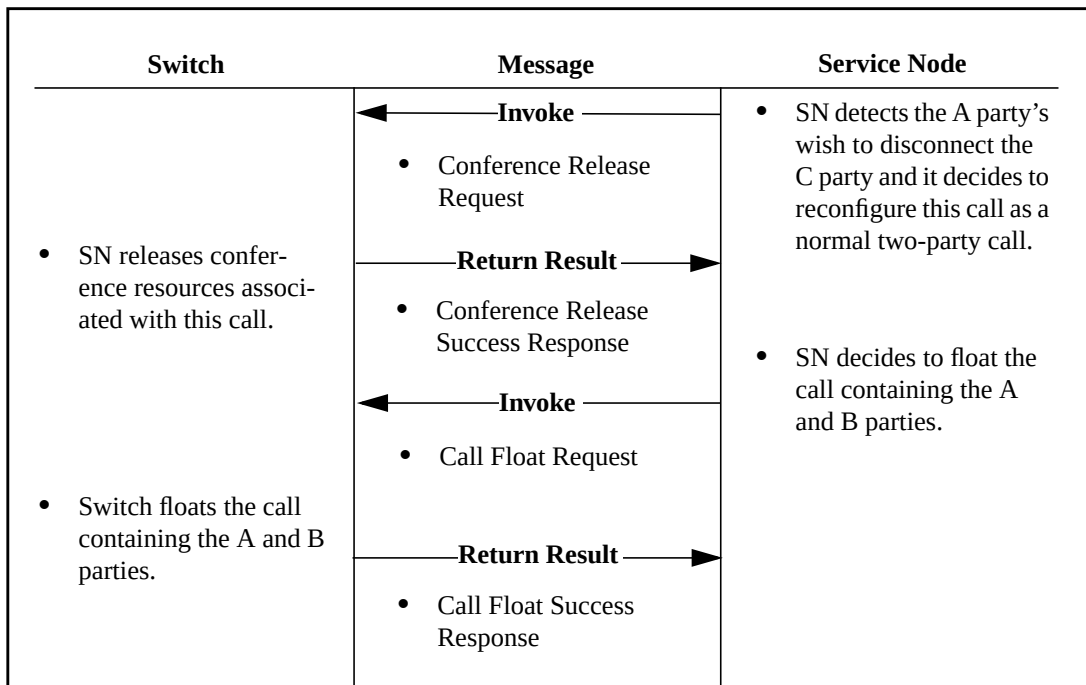
As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for a C party disconnect during consult, the Connection Status Inform received in the C call would be the first message in the overall flow. After that, the A/B call could be floated at the same time that the C call is being ended.



12.4.6 Party A releases party C consult phase

In this example, the A party signals (via DTMF, e.g.) to release the C party during the consult phase (perhaps because the C line was busy). For the purpose of this example flow, we will assume that the SN wishes to end the call containing the C party and float the call containing the A and B parties as a normal two-party call.

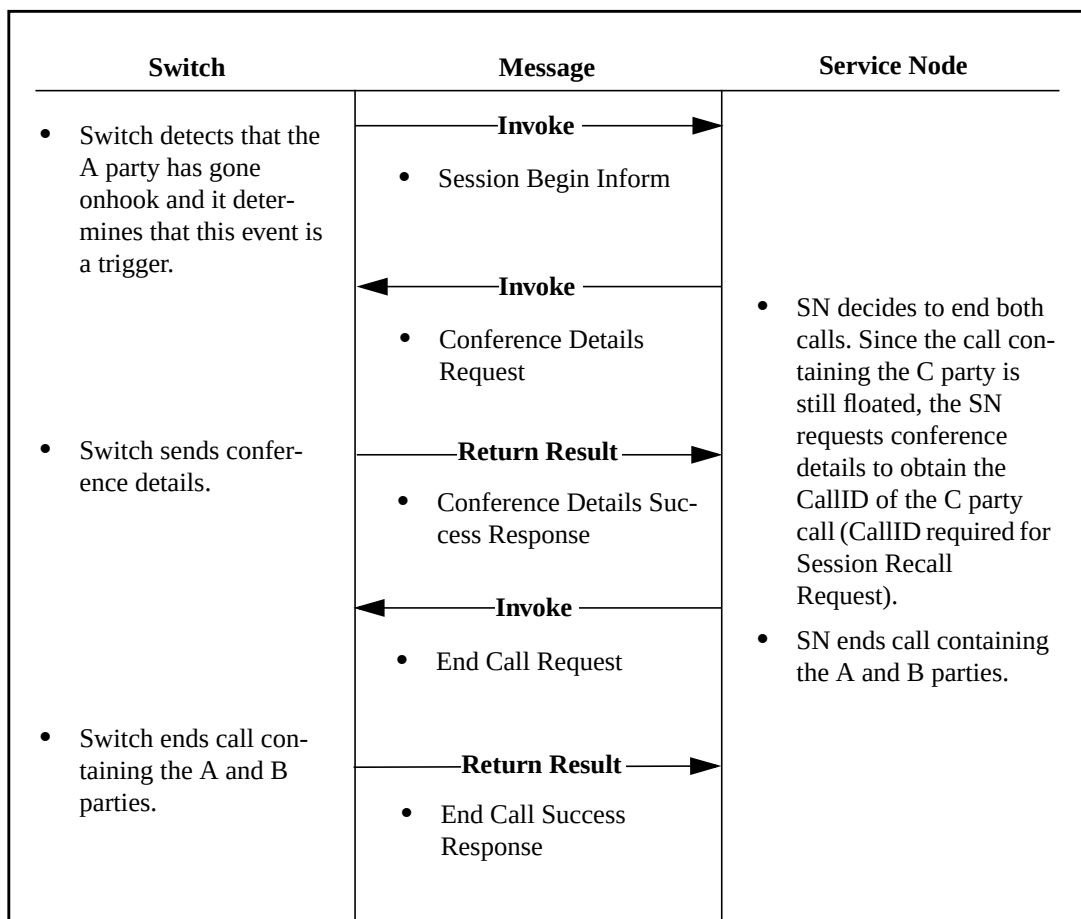
As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for an A party releasing C party during consult, the flow starts with the SN detection of an A party event in the call containing the A and B parties.

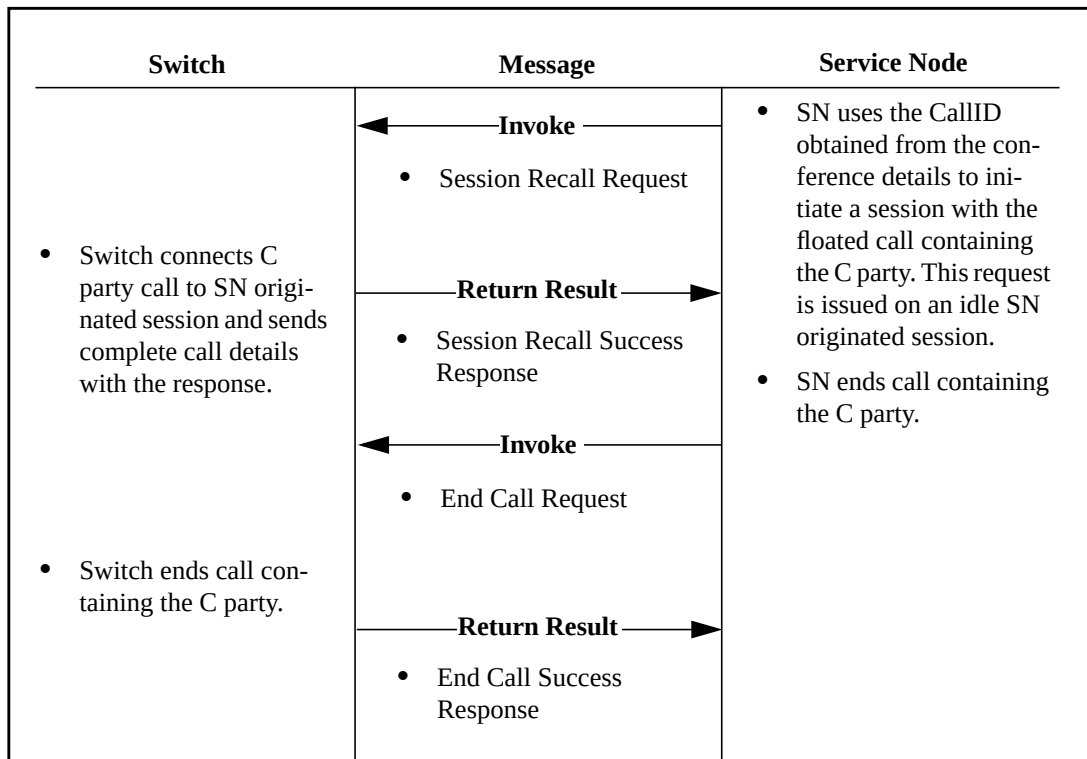


12.4.7 Party A disconnects during conference phase

This flow depicts an A party disconnect at any point during the conference phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to end both calls as a result of the A party disconnect.

As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for an A party disconnect during the conference phase, the flow starts with the A party disconnect trigger in the call containing the A and B parties.

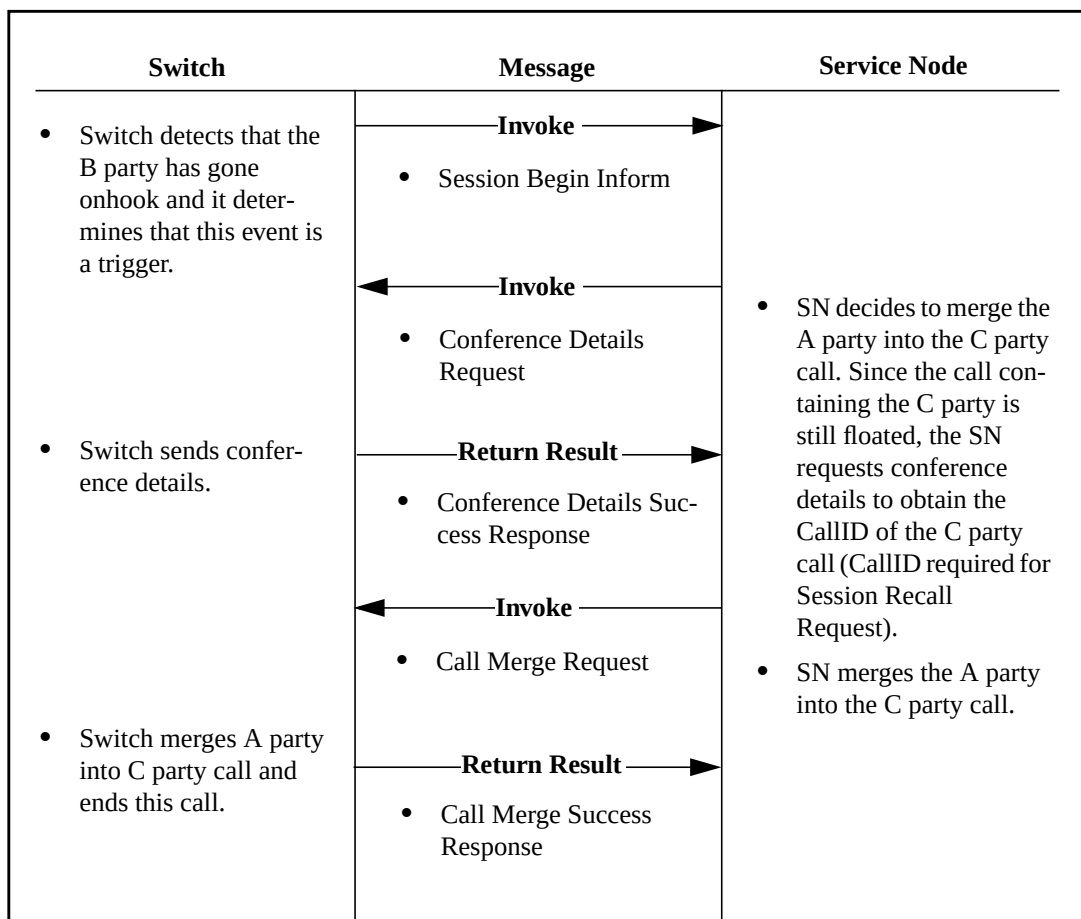


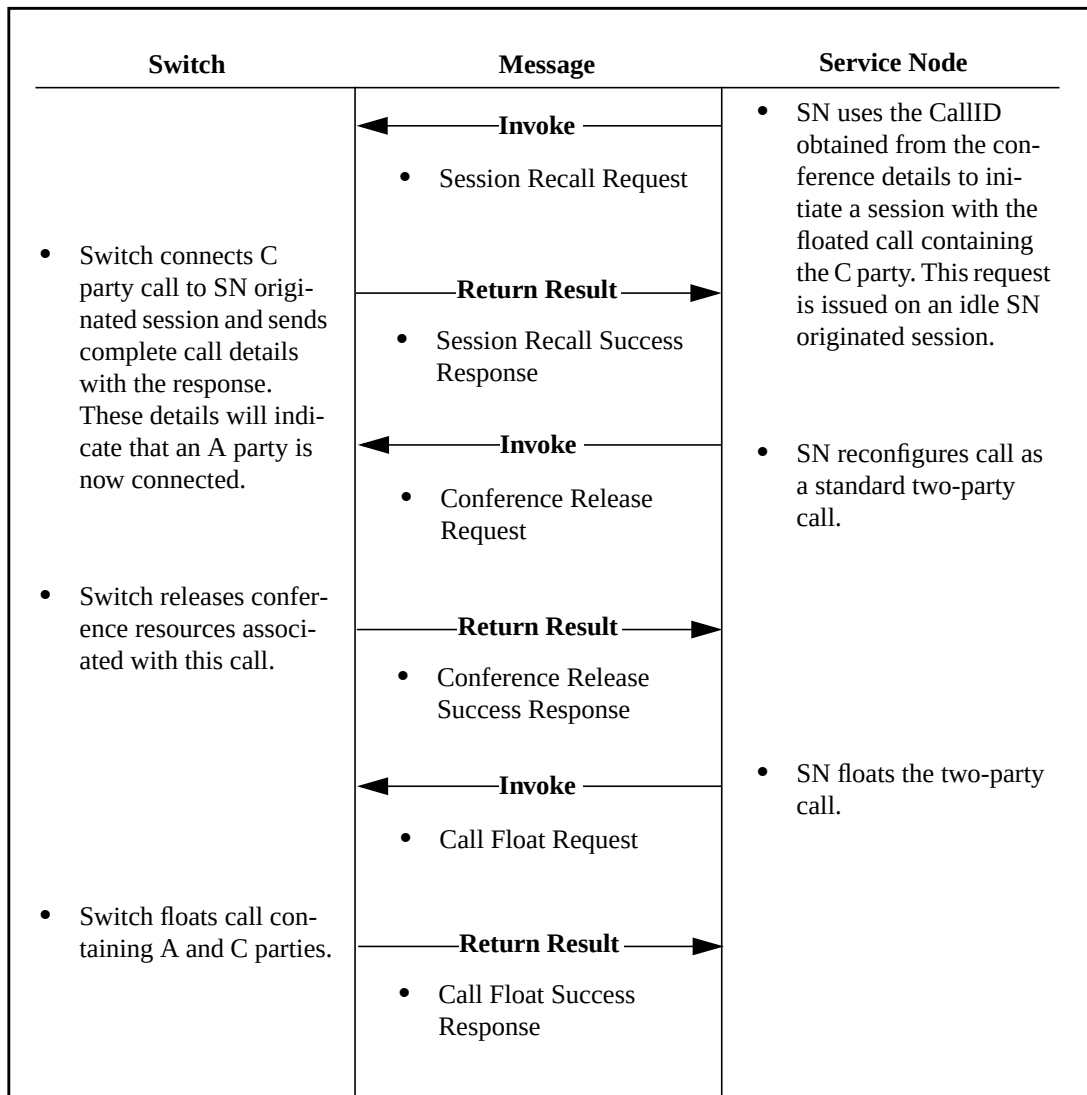


12.4.8 Party B disconnects during conference phase

This flow depicts a B party disconnect at any point during the conference phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to merge the A party into the C party call and float the call as a two-party call.

As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for a B party disconnect during conference, the flow starts with the B party disconnect trigger in the call containing the A and B parties.

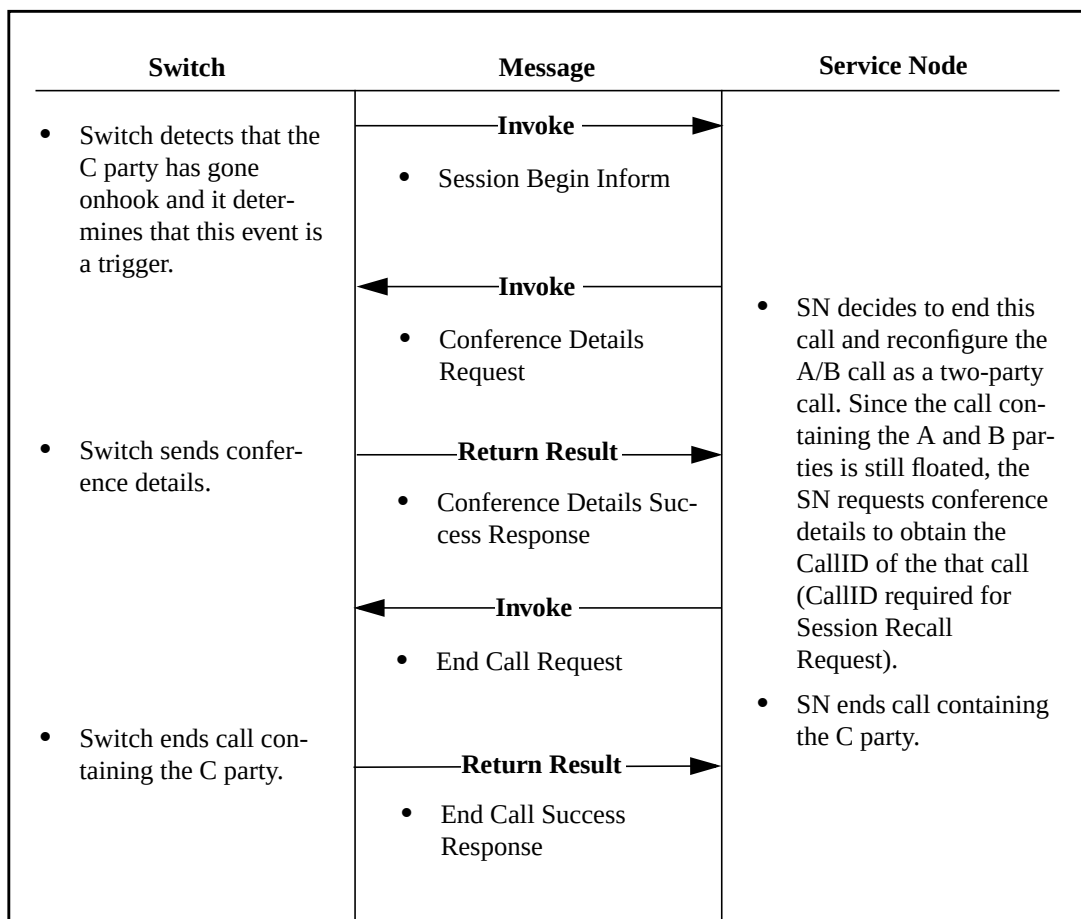


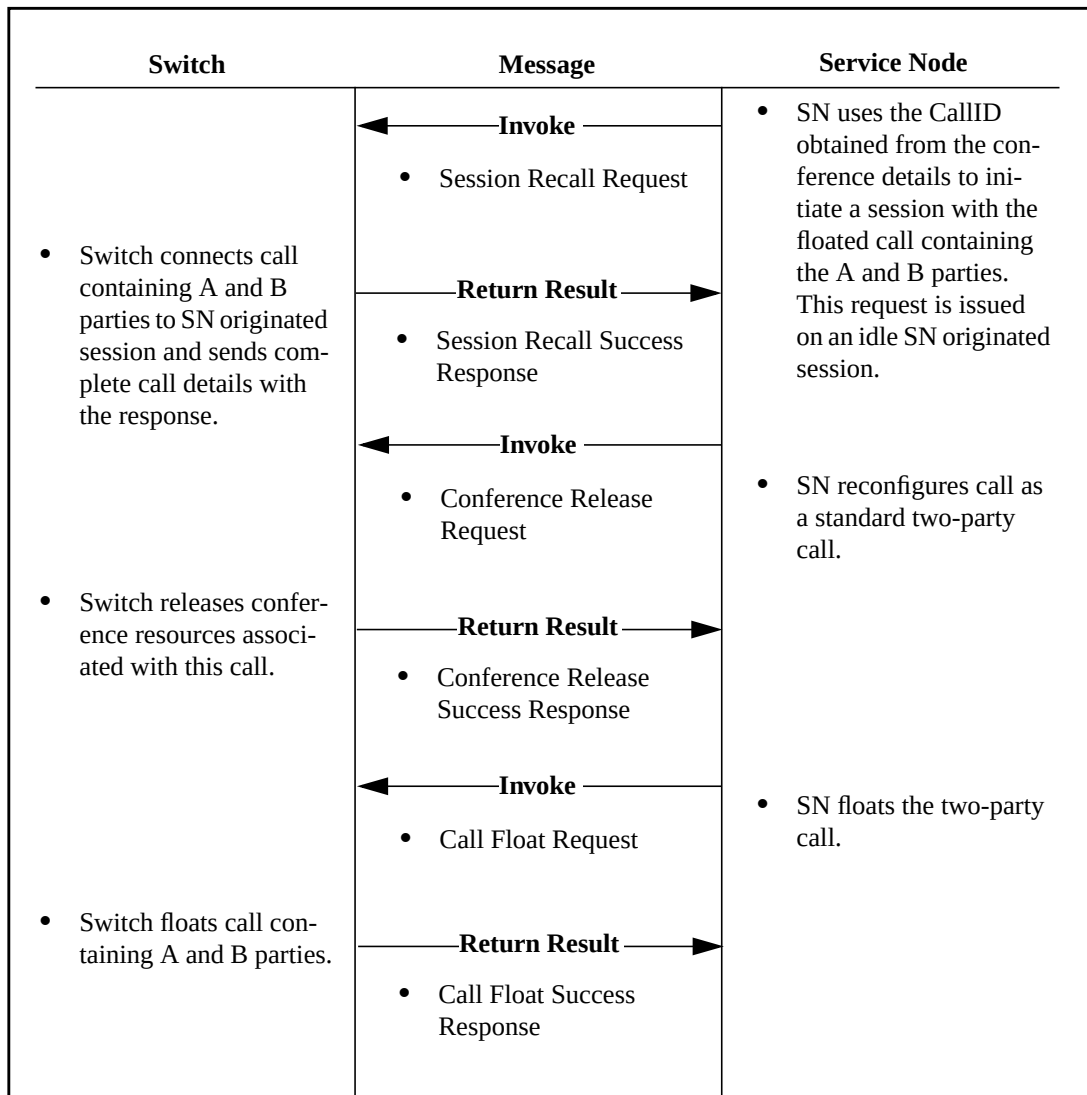


12.4.9 Party C disconnects during conference phase

This flow depicts a C party disconnect at any point during the conference phase of the conference. For the purpose of this example flow, we will assume that the SN wishes to end the call containing the C party and reconfigure the call containing the A and B parties as a two-party call.

As with previous flows, this flow involves two calls. The first flow corresponds to the session with the call containing the A and B parties and the second flow corresponds to the session with the C party call. In the overall flow for a C party disconnect during conference, the flow starts with the C party disconnect trigger in the call containing the C party.





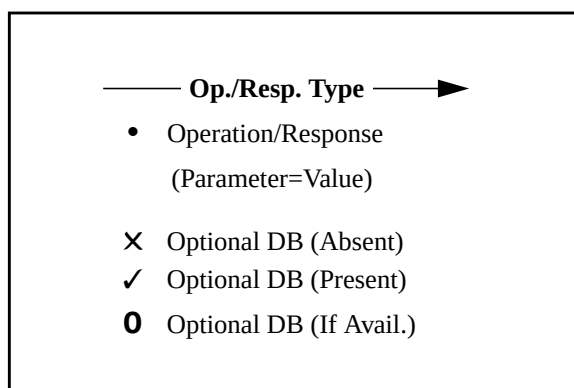
12.5 Connection Status Call Flows

As general usage of Connection Status Informs is covered fairly substantially in the DN Connection call flows, only one flow relating to ISUP continuity is presented here.

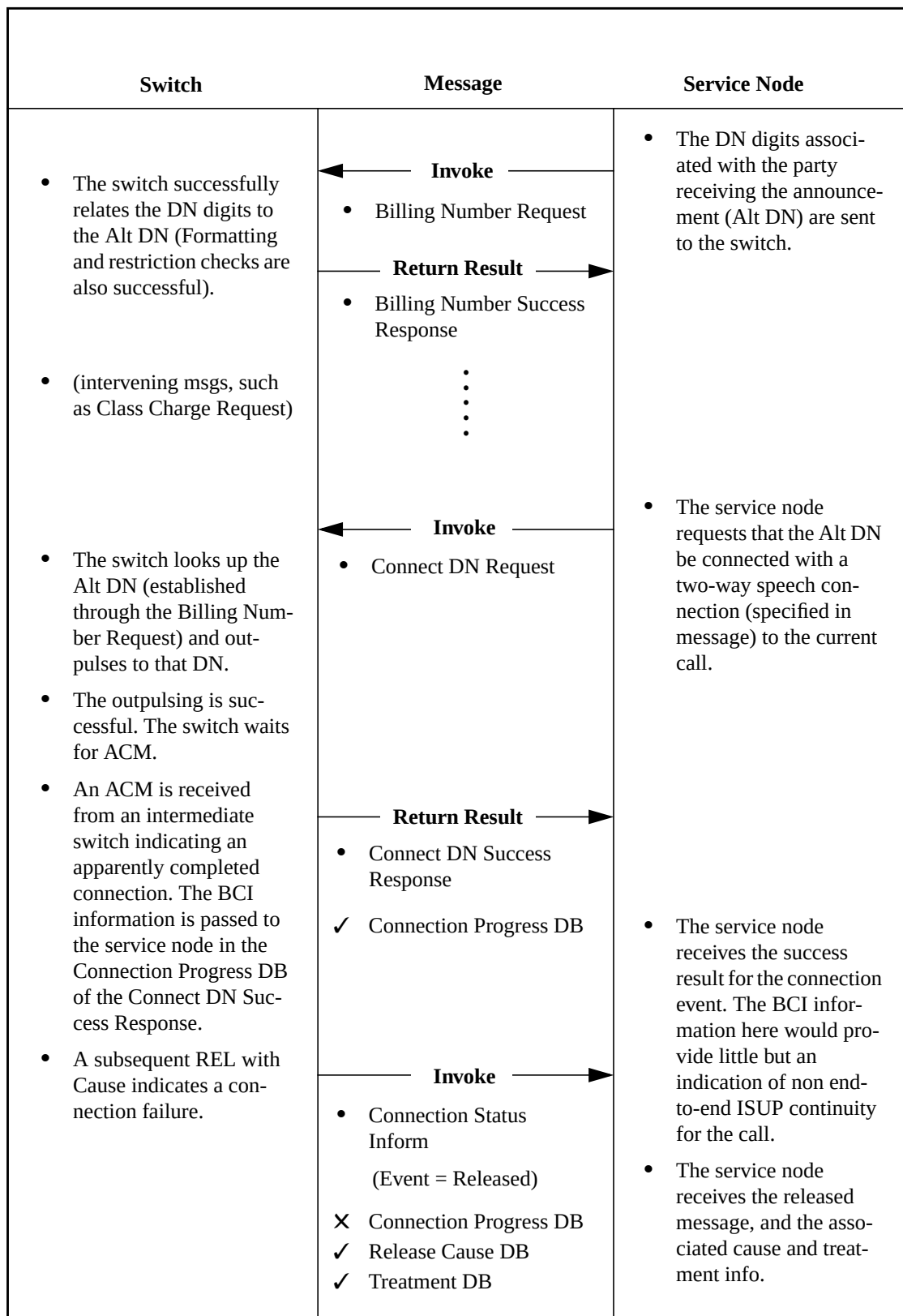
12.5.1 Connection Status Inform (Release after ACM)

The following automated third-billed scenario shows an example where an intermediate switch in an ISUP network hasn't waited for an ACM to be returned from the end office before returning an ACM to the originator (in this case, the TOPS DMS). In this scenario, the initial connection appears to have succeeded, but actually is determined shortly thereafter to have failed.

The following flow diagrams include significant parameters and optional data blocks, in addition to the usual operations/response type information, for purposes of call flow clarification. For individual messages in the call flow figures, these are shown as follows:



Note: Optional DBs marked 'O' are sent if, under the particular network condition, the switch has the optional data available to send.



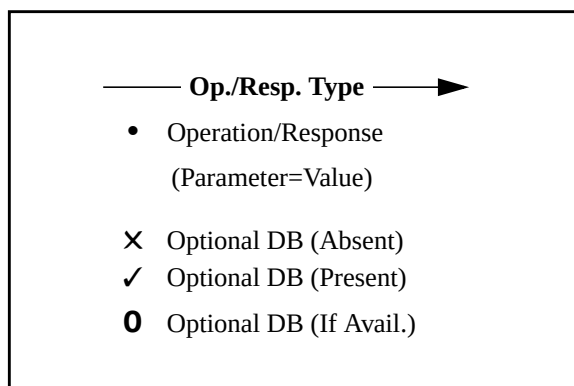
12.6 DN Connection Call Flows

The following call flows highlight making (Connect DN Request) and breaking (Release DN Request) connections. Success and failure cases are typically shown for both ISUP and non-ISUP scenarios.

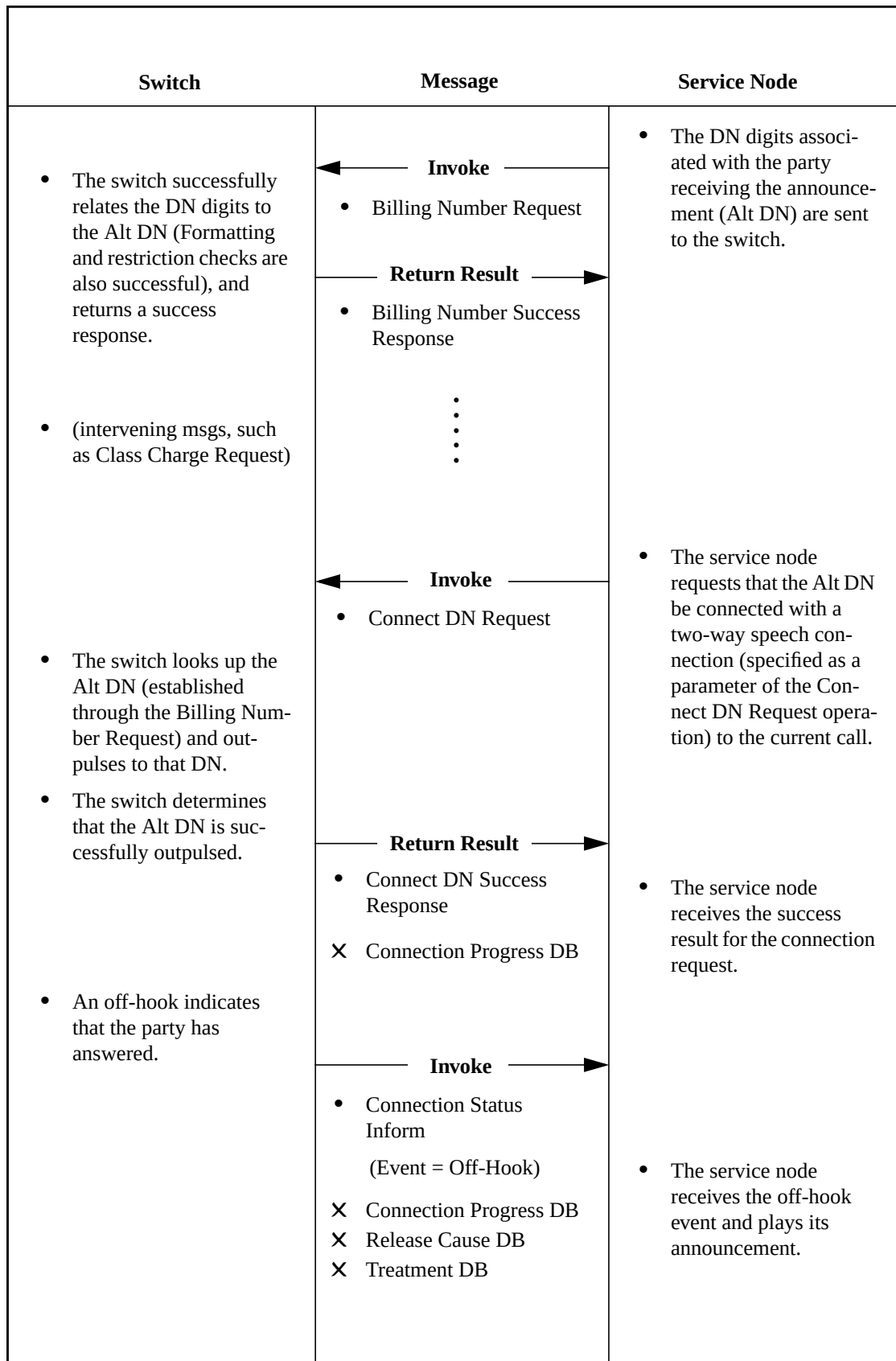
12.6.1 Connect DN Request (Non-ISUP Trunk, Success)

The following flow fragment is from an automated third-billed call (automated billing acceptance in response to a service node announcement prompt). A Connect DN message is sent by the service node to connect the alternate DN party who will be asked for billing acceptance. Upon outpulsing to the alternate DN, the switch identifies that the outpulsed digits were successfully sent, and then sends a Return Result of Success to the service node. As shown, the service node still needs to wait for an off-hook (in the Connection Event field of a Connection Status Inform message) to ensure a completed connection and an available party (Note: the speech status for the connect party is selected in the Connect DN Request).

The following flow diagrams include significant parameters and optional data blocks, in addition to the usual operations/response type information, for purposes of call flow clarification. For individual messages in the call flow figures, these are shown as follows:



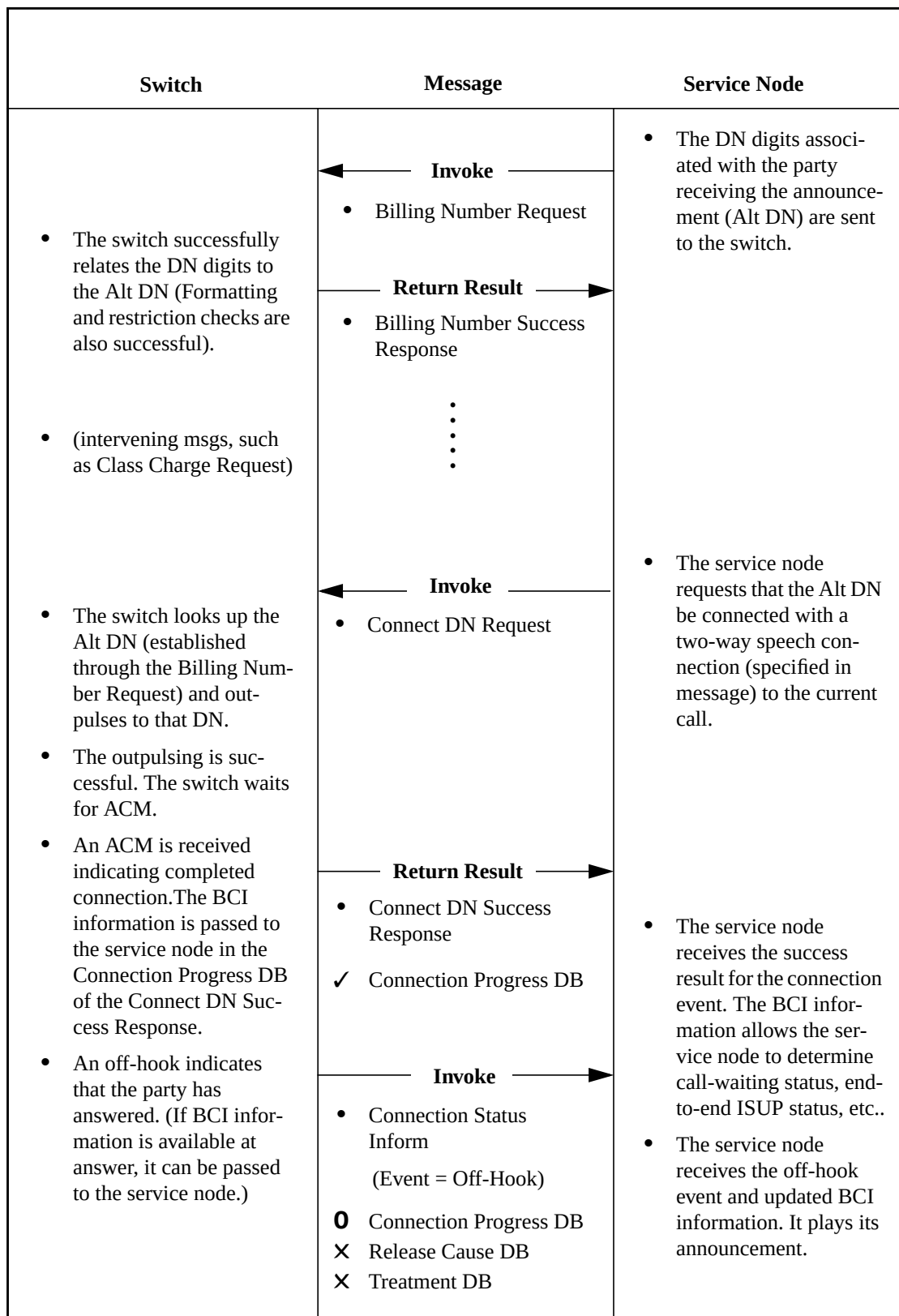
Note: Optional DBs marked ‘O’ are sent if, under the particular network condition, the switch has the optional data available to send.



12.6.2 Connect DN Request (ISUP Trunk Outpulse Forward, Success)

The following flow shows the same call flow for an ISUP trunk. In the ISUP case, for outpulsing forward, the switch identifies a successful connection upon receipt of a successful Address Complete Message (ACM). The switch then sends a Return Result of Success, which includes the SS7 Backward Call Indicators information ('backward' here, is unrelated to DMS port usage) provided through the ACM message in the Backward Call Indicators field of the optional Connection Progress DB. There is no attempt to interpret the Backward Call Indicators; they are simply passed through to the service node. The service node can then determine call waiting, ringing state, etc. from the subfields of the Backward Call Indicators field.

One evolutionary issue here is the completeness of the information in the Backward Call Indicators. If the call is not 'end-to-end' ISUP, then the Backward Call Indicators will indicate this lack of ISUP continuity, but no ringing status. A similar situation exists if any switch in the ISUP chain does not wait for an ACM from its far side before returning an ACM towards the originating requestor. In the former case, ringing status will never be received. In the latter case, it might be (in a follow-up message to the ACM; the Call Progress Message or CPG). Thus in a network with little end-to-end ISUP continuity, the BCI information returned to a service node in a Connect DN Success Response will rarely provide call progression value. As the network matures to full end-to-end ISUP, the BCI information will become increasingly more useful. The important point to be noted here is that scenarios exist wherein the switch receives an ACM suggesting a completed connection (passing the BCI information from the ACM to a service node in the Connection Progress DB) but then subsequently receives a Release with Cause, indicating a connection error, prior to answer. This odd sequence of events will be covered in a Connection Status Inform scenario.

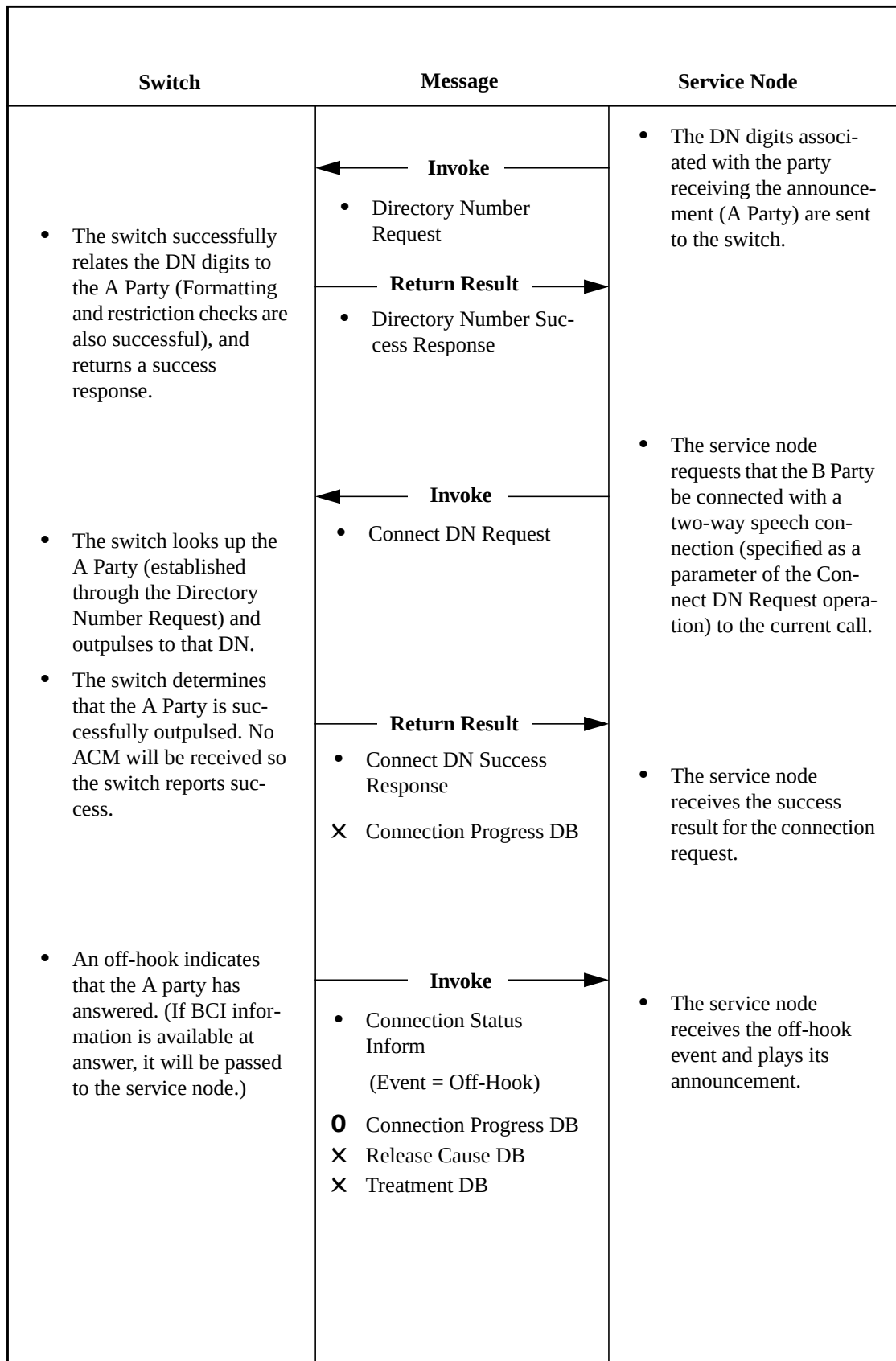


12.6.3 Connect DN Request (ISUP Trunk Outpulse Backward, Success)

The following flow shows a Booked Call Database call where the service node needs to connect the A Party on the backward port.

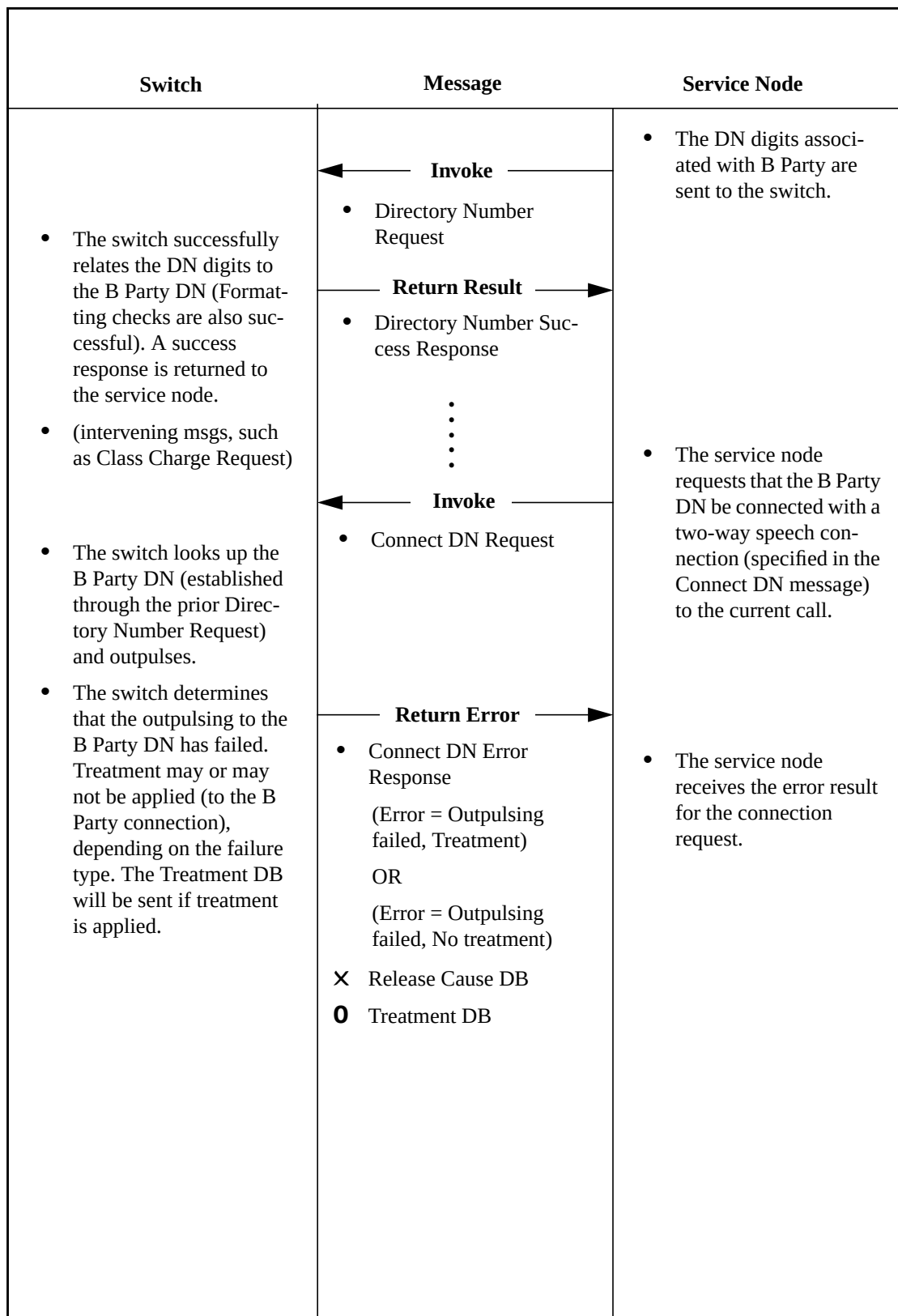
The distinction here is that the switch will not receive ACM on outpulsing back, even though this is an ISUP trunk, thus the switch must report success based on outpulsing, rather than on receipt of ACM. No BCI information will be available to pass to the service node.

This call operates much like a non-ISUP call.



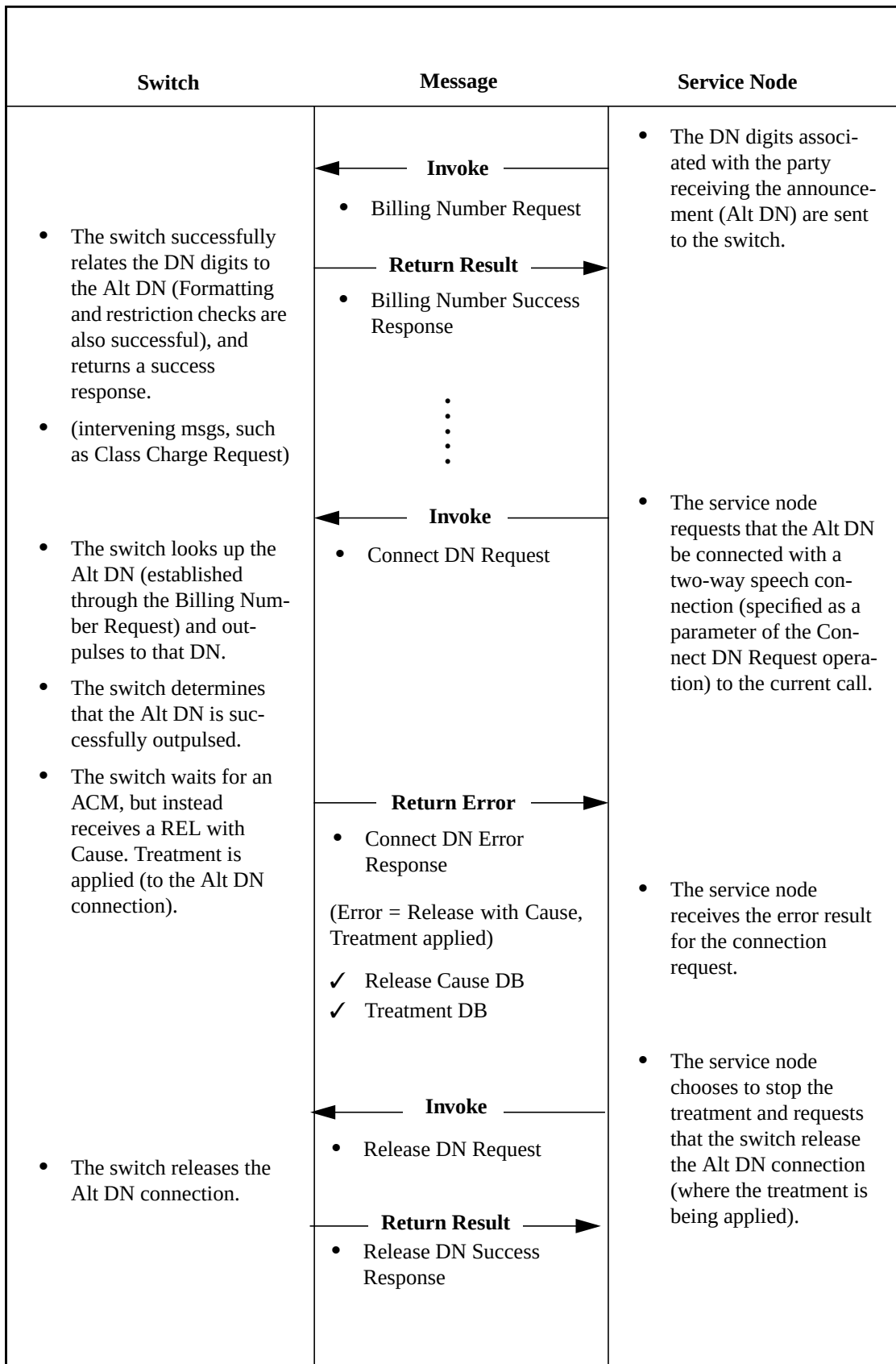
12.6.4 Connect DN Request (Failure on Outpulsing)

The following automated Collect-billed call scenario covers an outpulsing failure. Note that treatment may or may not be applied, depending on the failure type. The scenario is the same for both ISUP and non-ISUP trunks.



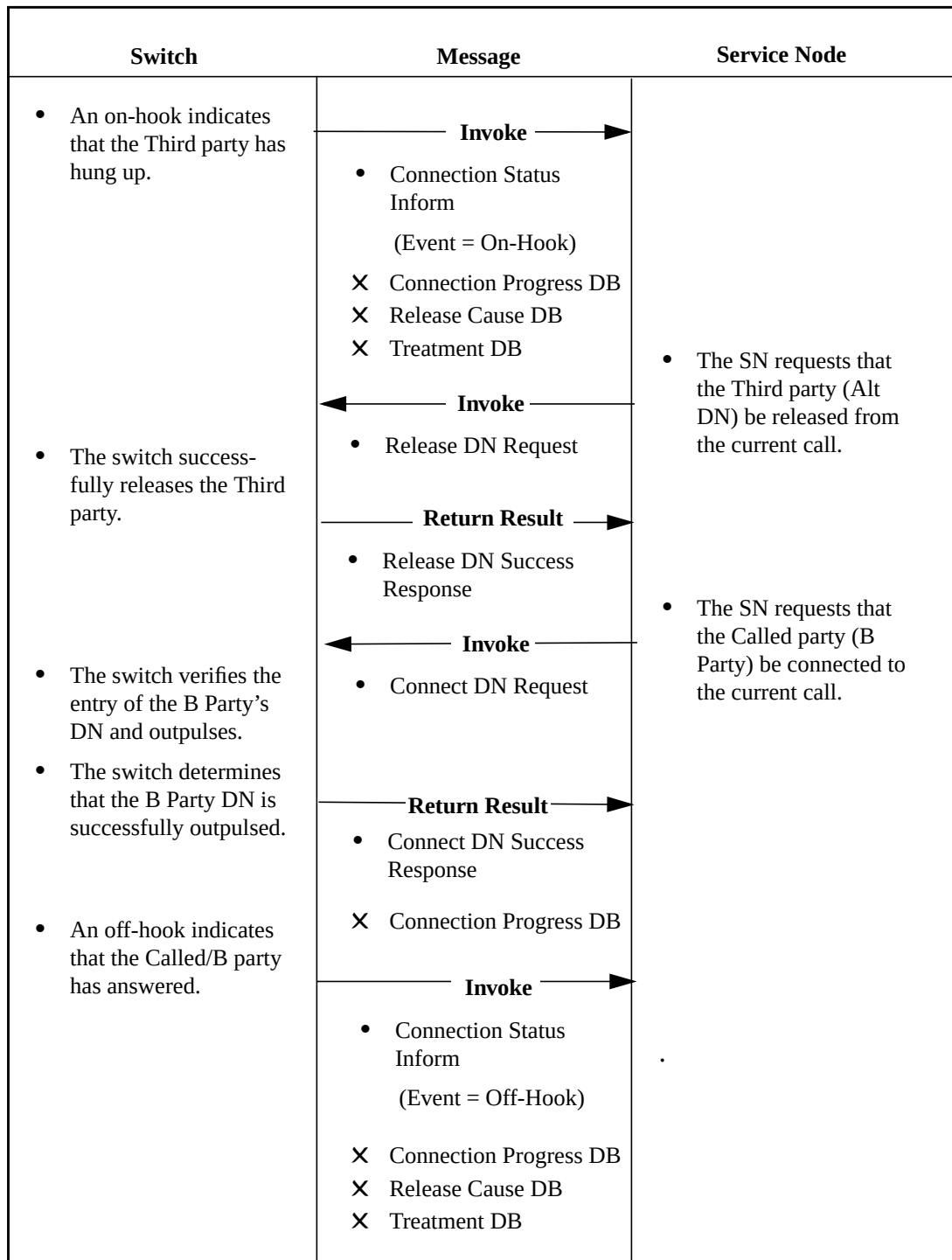
12.6.5 Connect DN Request (ISUP Trunk, Failure, Release with Cause)

The following call flow again shows part of an automated Third-billed call scenario where the Third party (Alt DN) needs to be called for billing acceptance. In this scenario a failure occurs due to an SS7 error, with the network returning a Release with Cause. In response, the switch will apply the treatment associated with the cause value (e.g. a reorder tone). The switch will pass through the cause value to the service node (which will provide the service node with specifics, such as whether the network is busy), as well as an indication of what treatment is being applied. It is up to the service node to decide whether to interrupt the treatment, or to allow it to proceed.



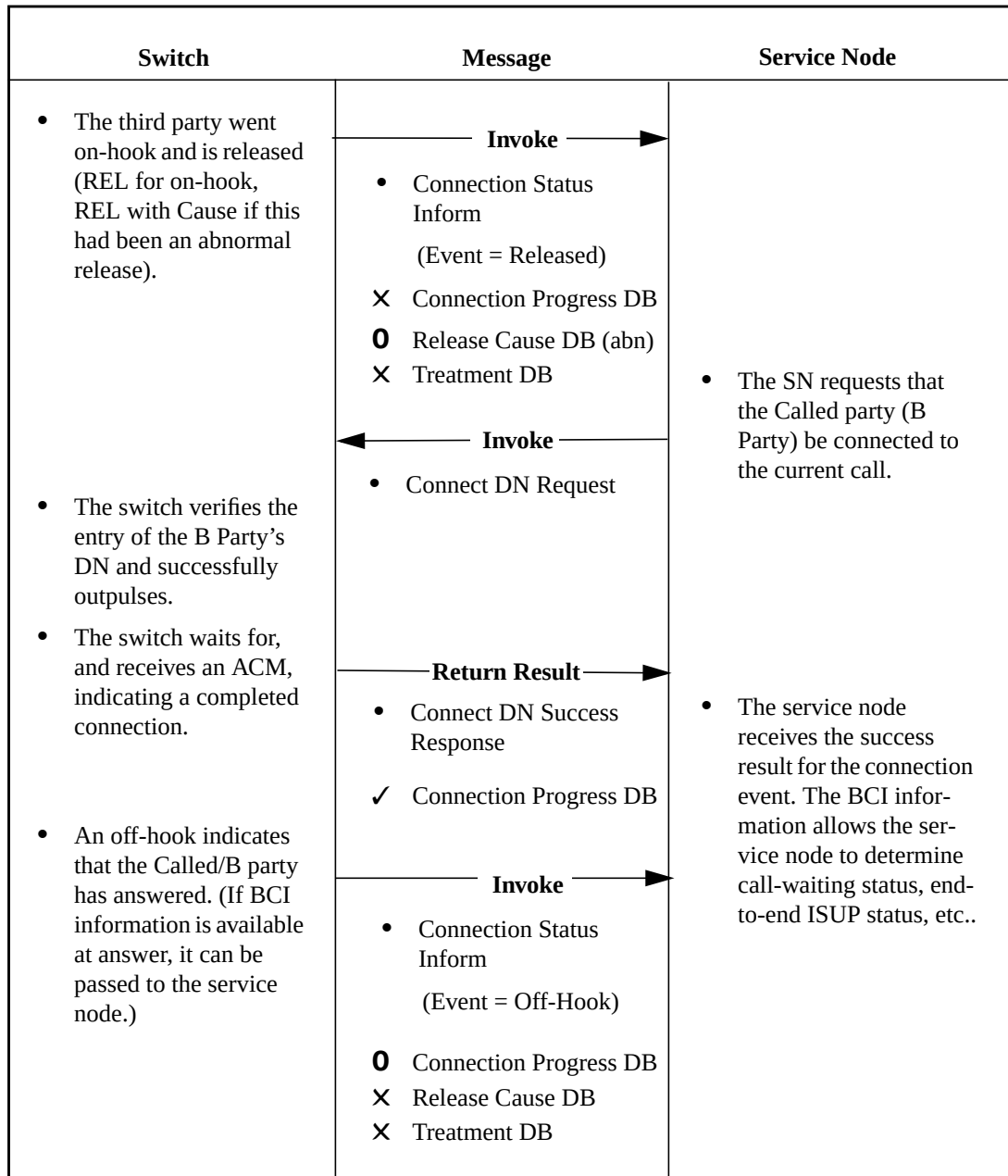
12.6.6 Release DN Request (Success, non-ISUP Trunk)

The following flow fragment is also from an automated Third-billed call. Both the Alt DN and the Called's DN have been provided to the switch. The Third party has accepted the billing. At this point, the connection to the Third party is released and a new connection to the Called party is requested.



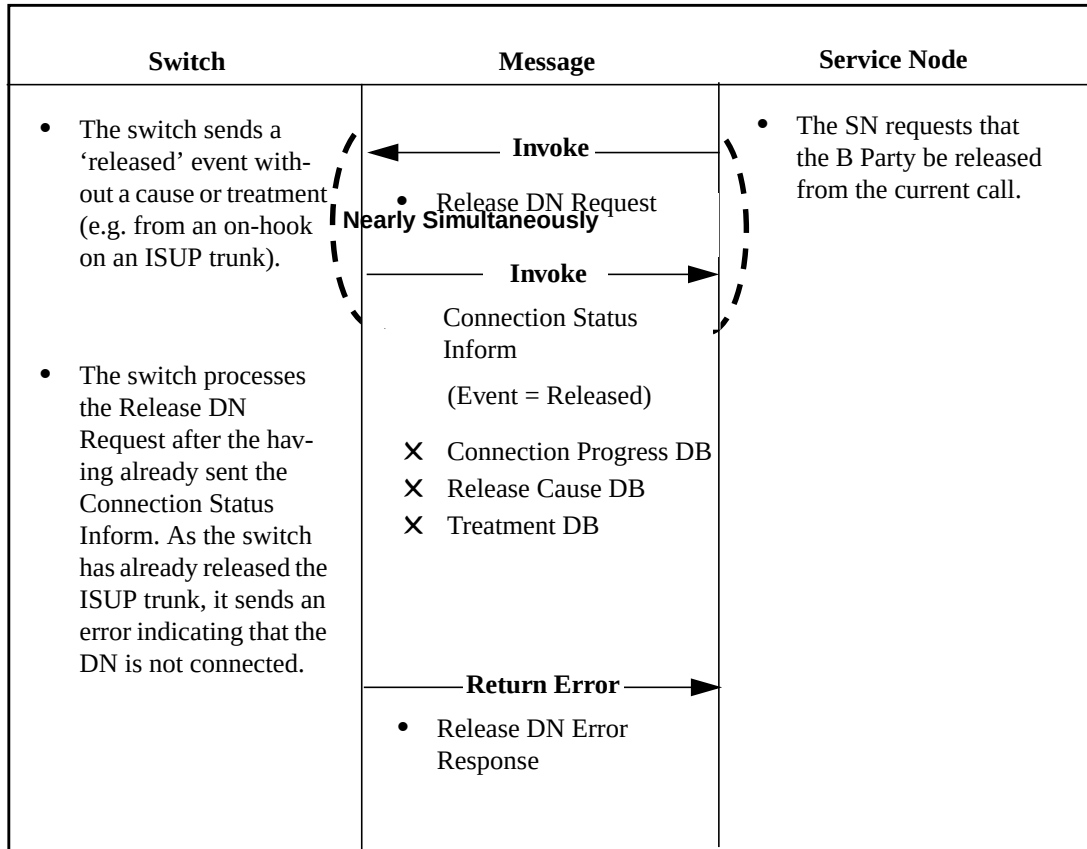
12.6.7 Release DN Request ‘equivalent’ (Success, ISUP Trunk)

This is the same flow fragment as for the previous scenario, but both parties are connected via ISUP trunks. Note that the third party was immediately released upon going on-hook (i.e. the Release DN Request is not actually necessary in this scenario as there is no operator hold to retain the call once the Third party went on-hook.), however ISUP can also simulate an on-hook through a suspend message (SUS), in which case the initial part of the call scenario would have followed the non-ISUP scenario.



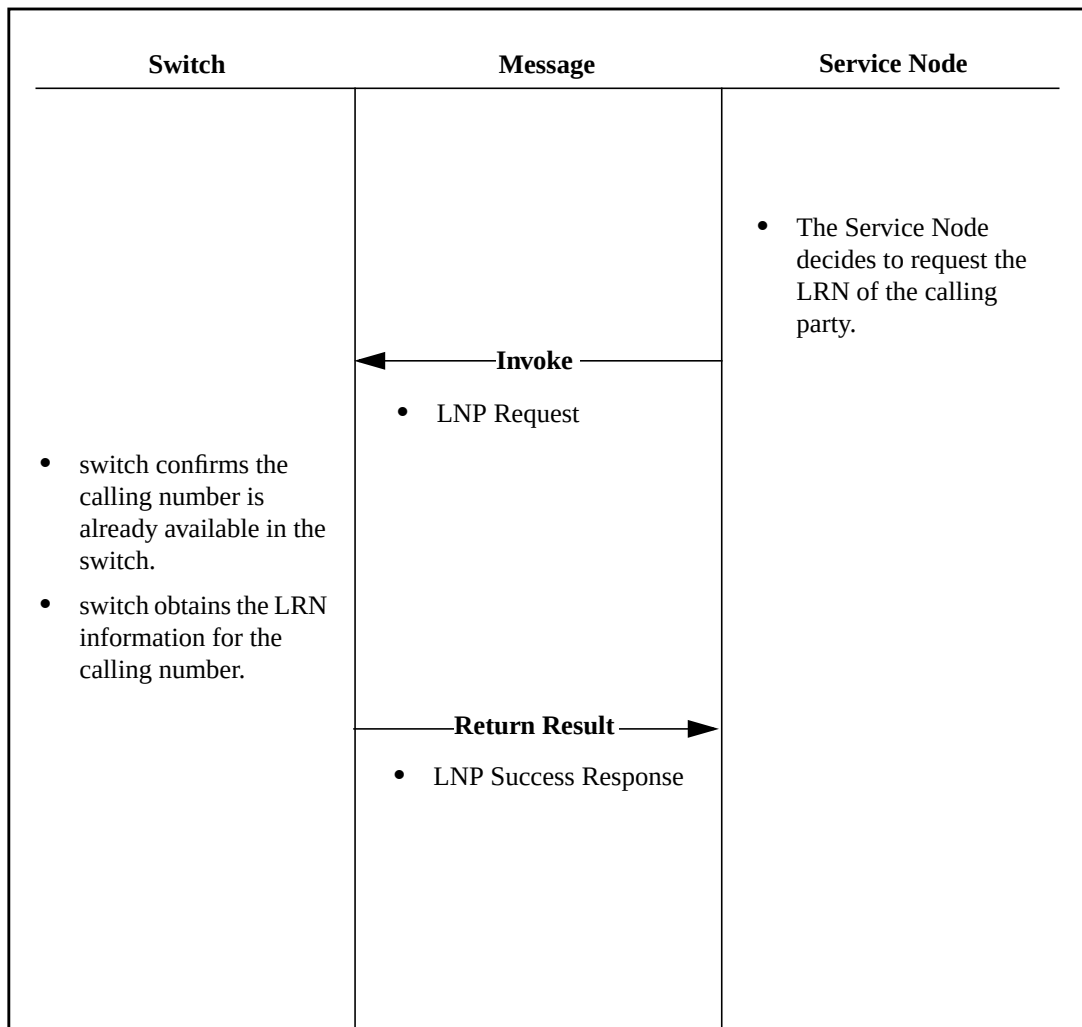
12.6.8 Release DN Request (Failure)

The following could follow a scenario where there was a race condition between an ISUP 'Release' event and a 'Release DN Request'. Both the Release DN Request from the service node and the Connection Status Event from the switch are transmitted nearly simultaneously. The switch receives and processes the Release Request after already having released the trunk. (Non-race condition scenarios might indicate the service node has lost track of 'who' is connected 'where'.)

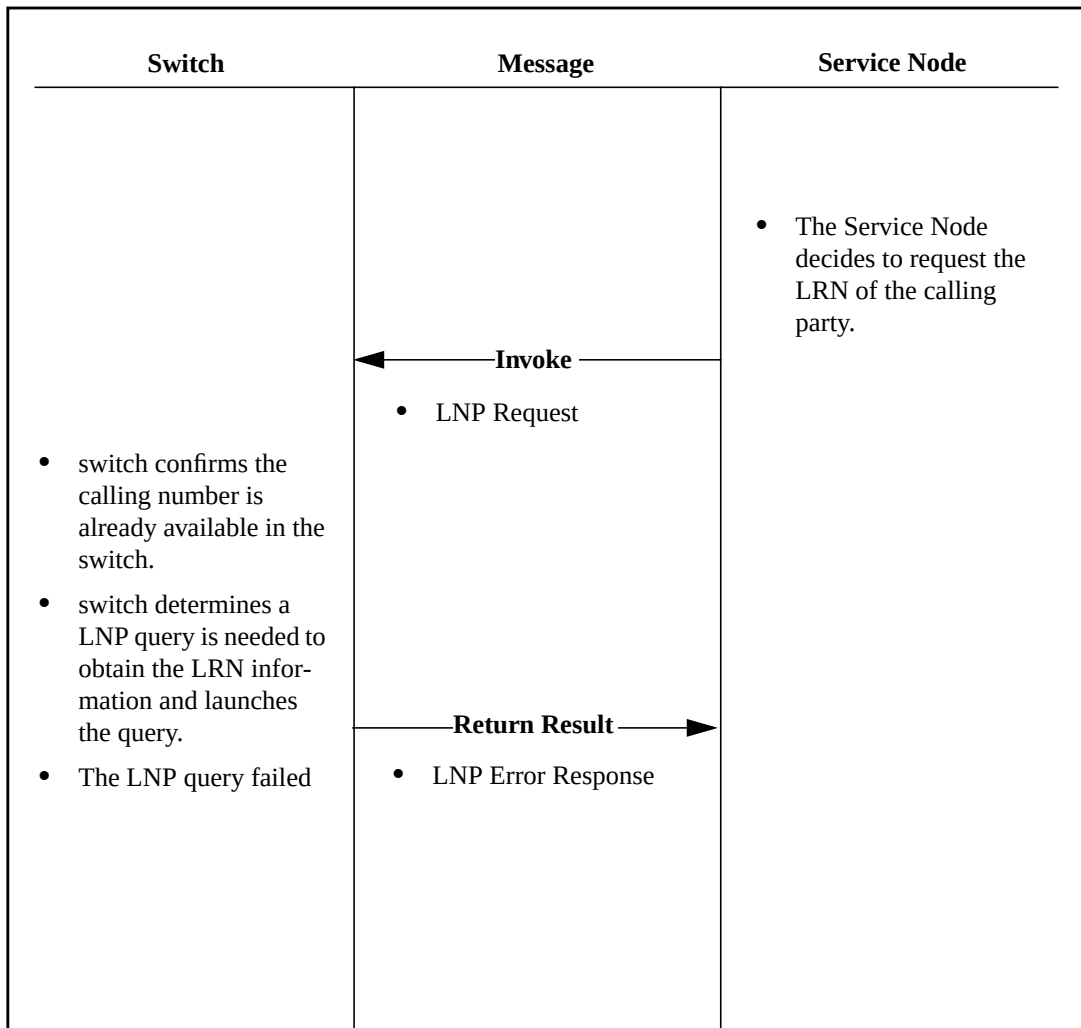


12.7 LNP Request

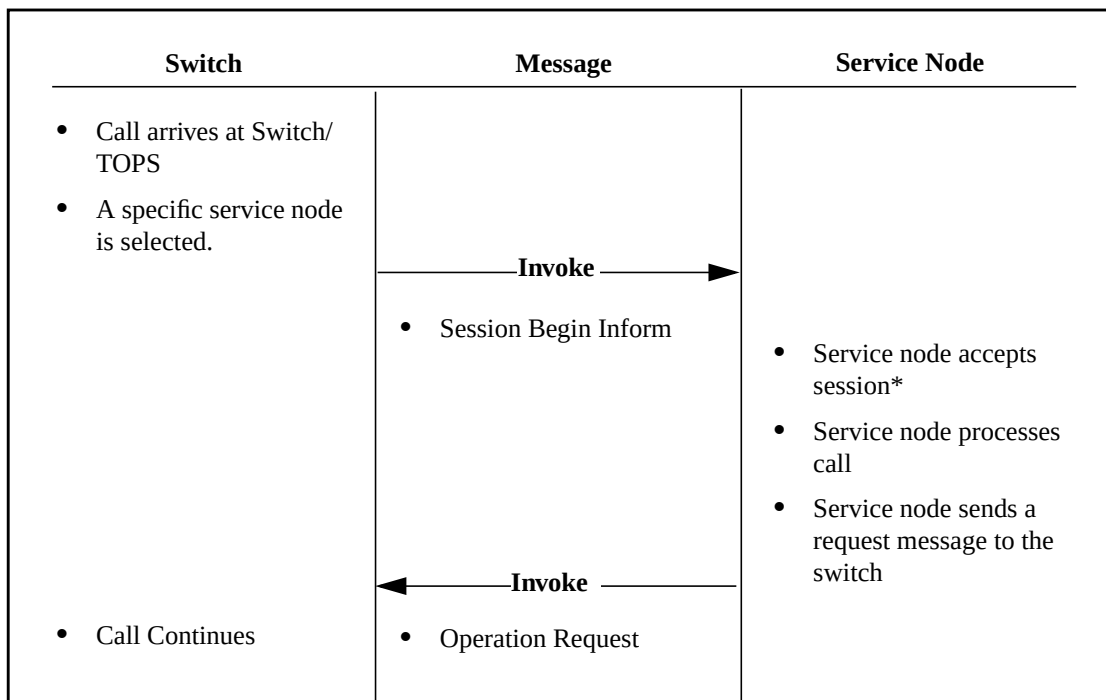
12.7.1 LNP Request (Success)



12.7.2 LNP Request (Failure)

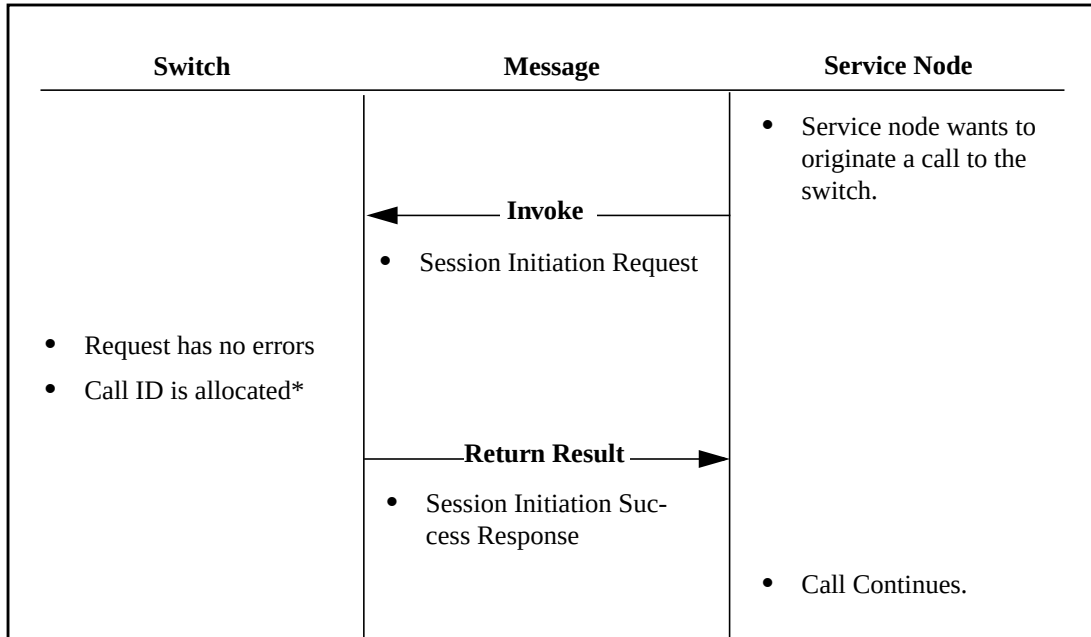


12.8 Session Origination From Switch to Service Node Call Flow



* If the service node cannot handle the call, then it will send an operation request that will instruct the switch to either transfer or abort the call.

12.9 Session Origination From Service Node to Switch Call Flow



* If an error is found in the request, the switch will not allocate a Call ID. Instead a NIL Call ID will be sent in the Session Initiation Error Response.

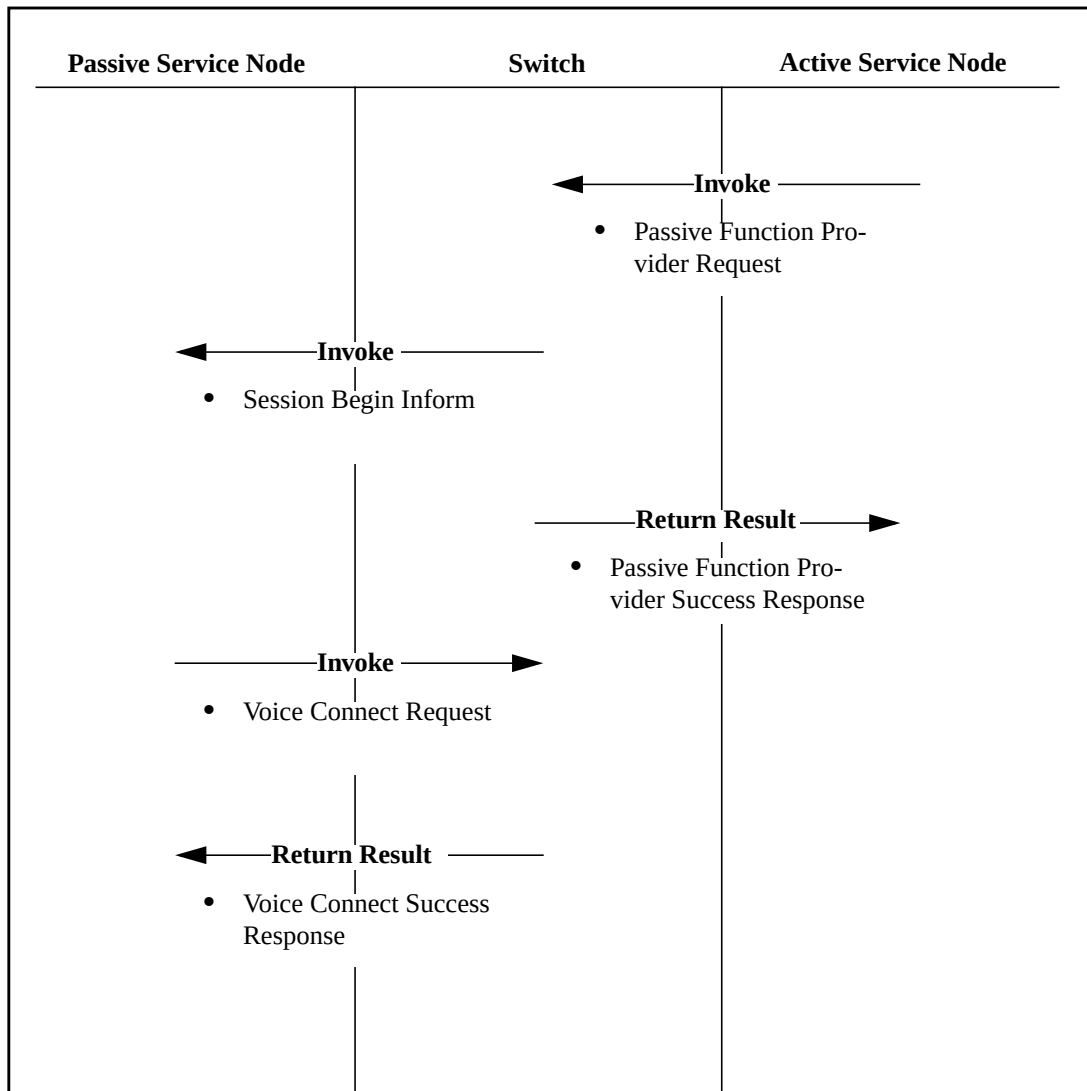
12.10 Simultaneous Interaction Call Flows

12.10.1 Connecting a Passive Function Provider

In order to initiate a simultaneous interaction, the active service node must send a Passive Function Provider request to the switch. The switch will use the function identifier supplied by the active service node to select a passive function provider for the call. The passive function provider may be either another service node or a TOPS operator. The message flows for each are shown in this section.

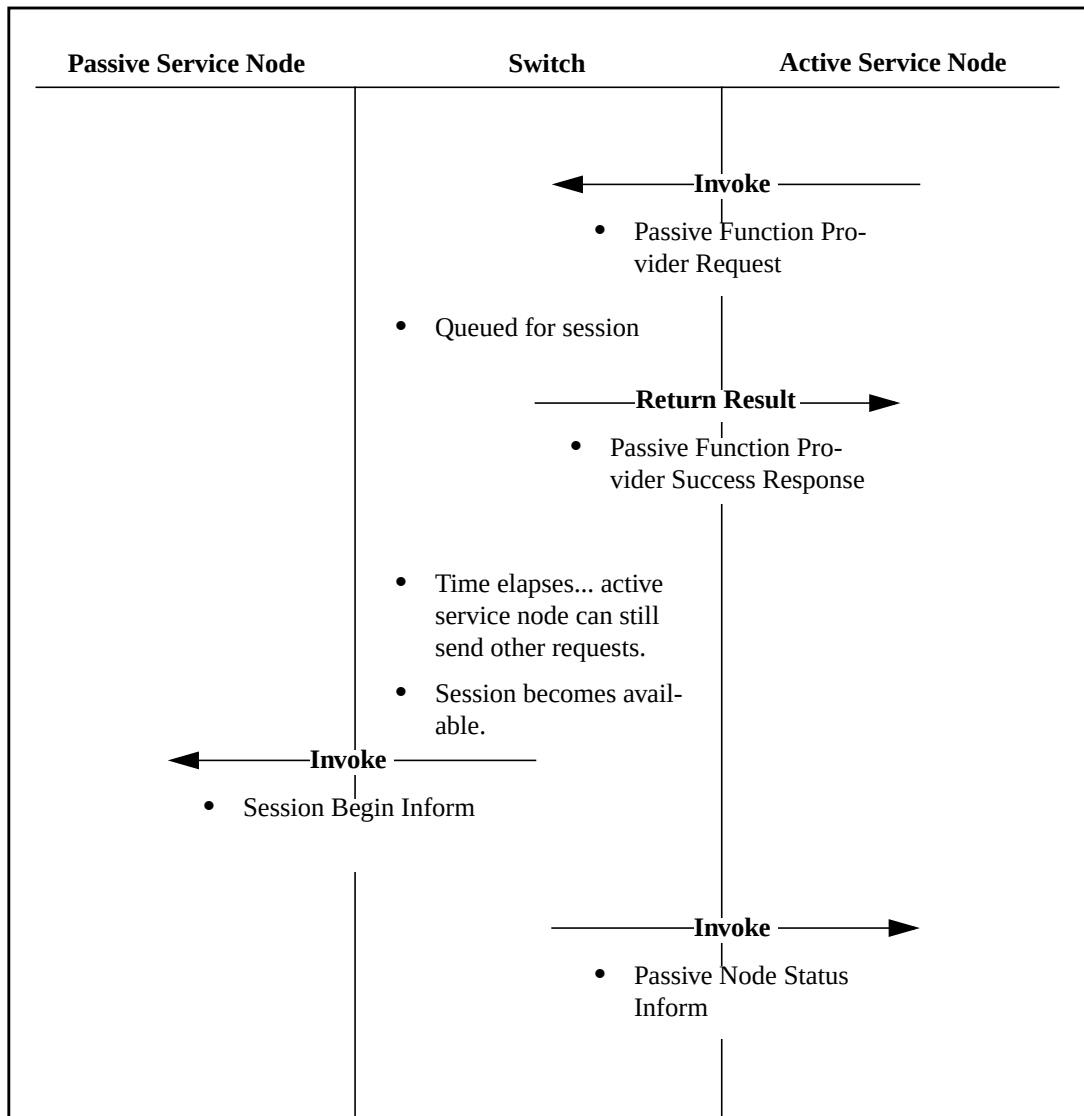
12.10.1.1 Connecting a Passive Service Node - Session Obtained Immediately

This flow shows a Passive Function Provider request that resulted in a session with a passive service node being requested. In this case, the session with the passive service node was obtained immediately (i.e. no queuing in the switch was required).



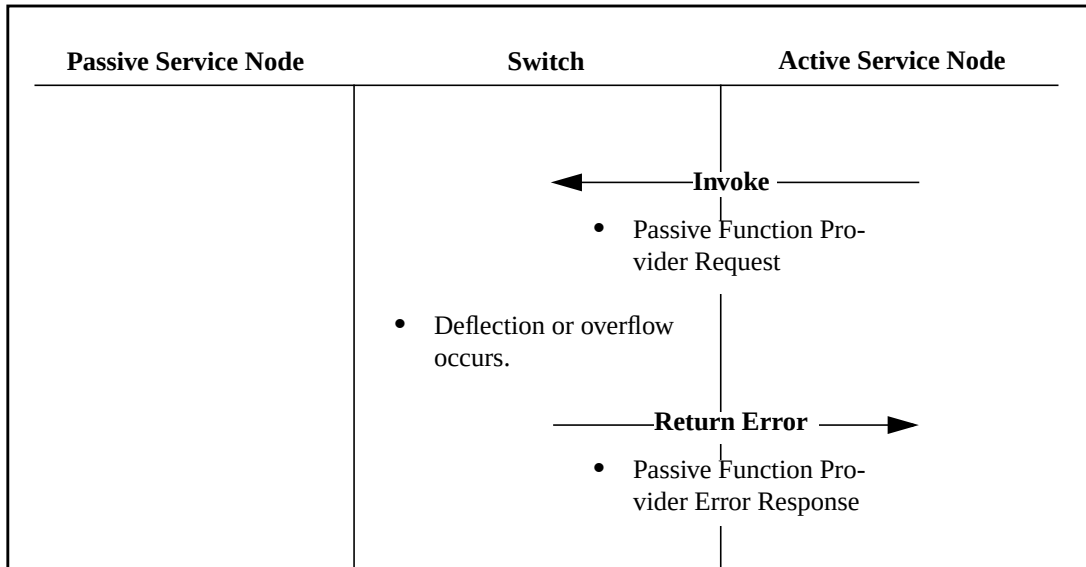
12.10.1.2 Connecting a Passive Service Node - Call Queues for Session

This flow shows a Passive Function Provider request that resulted in a session with a passive service node being requested. In this case, a session with a passive service node was not immediately available. The switch queues the call for a passive service node until one becomes available.



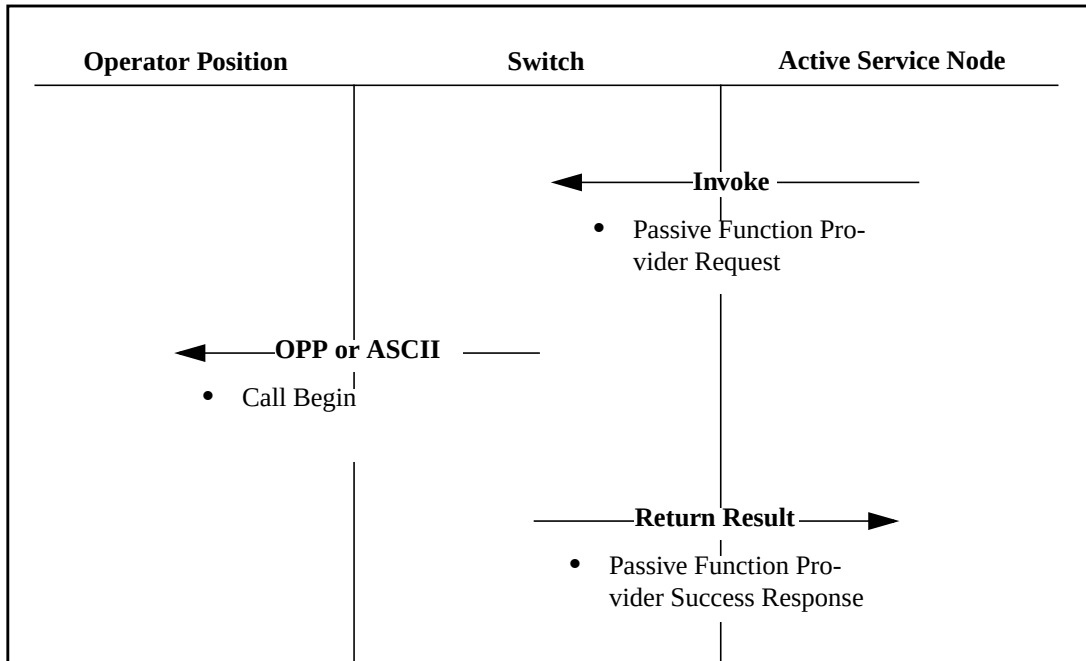
12.10.1.3 Connecting a Passive Service Node - Deflection or Overflow

This flow shows a Passive Function Provider request that resulted in a session with a passive service node being requested. In this case, a session was not immediately available and a deflection or overflow condition occurred in the switch when an attempt was made to queue the call.



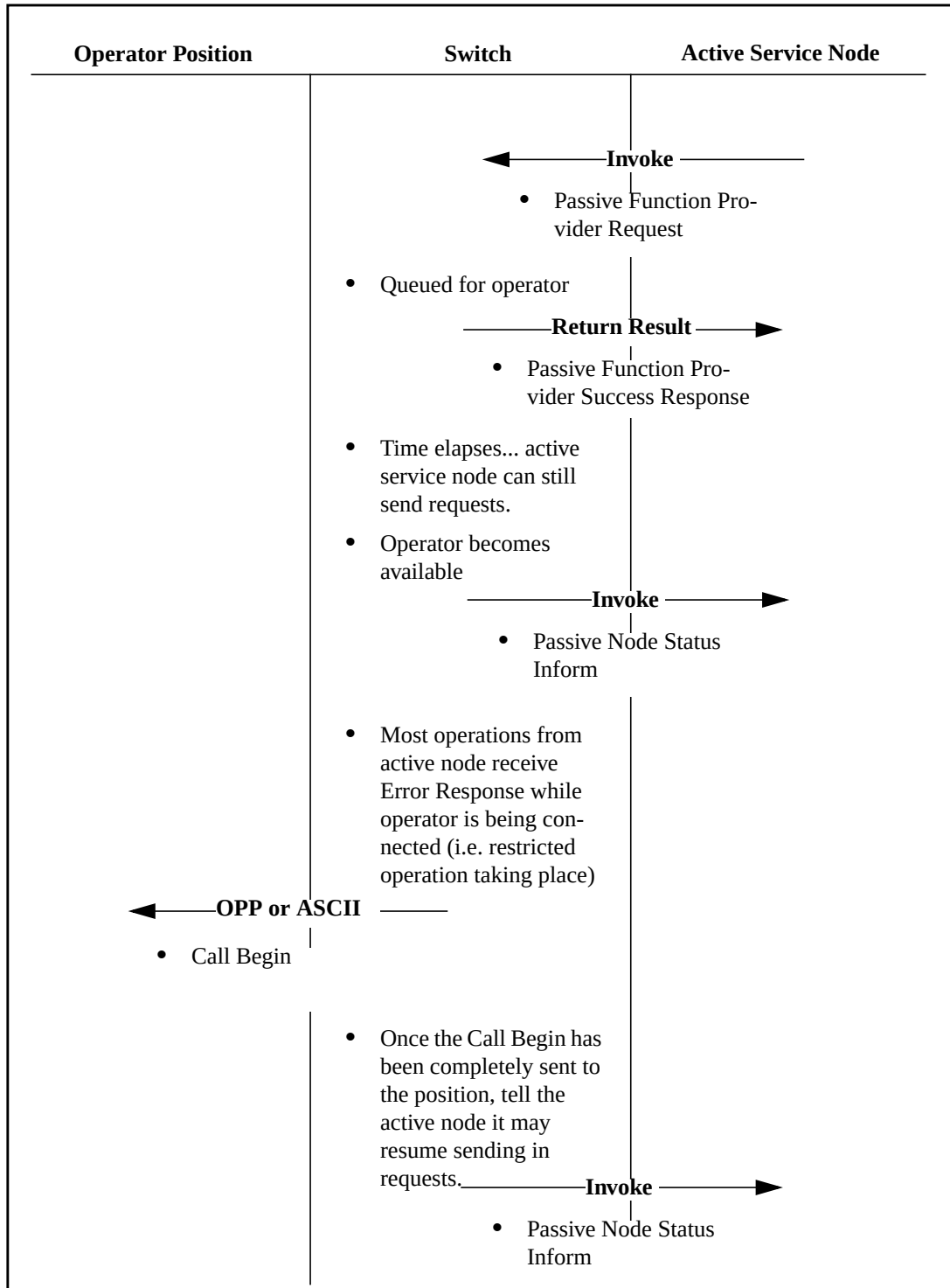
12.10.1.4 Connecting a Passive Operator - Operator Obtained Immediately

This flow shows a Passive Function Provider request that resulted in a TOPS operator being requested. In this case, an operator was obtained immediately (i.e. no queuing in the switch was required).



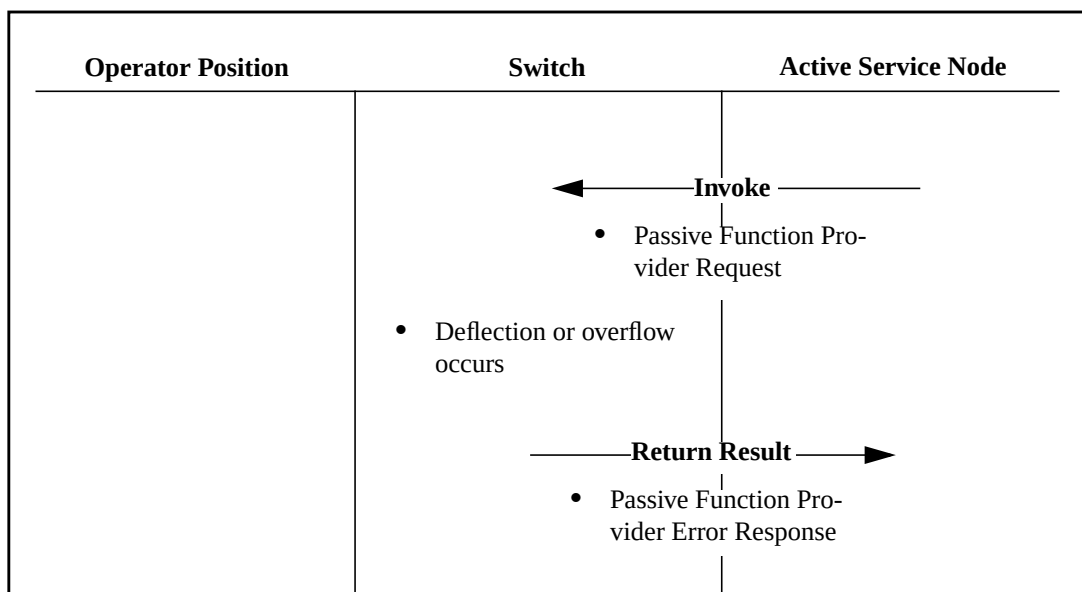
12.10.1.5 Connecting a Passive Operator - Call Queues for Operator

This flow shows a Passive Function Provider request that resulted in a TOPS operator being requested. In this case, an operator was not immediately available for the call. The switch queues the call until an operator becomes available.



12.10.1.6 Connecting a Passive Operator - Deflection or Overflow

This flow shows a Passive Function Provider request that resulted in a TOPS operator being requested. In this case, an operator was not immediately available and a deflection or overflow condition occurred in the switch when an attempt was made to queue the call.



12.10.2 Releasing a Passive Function Provider

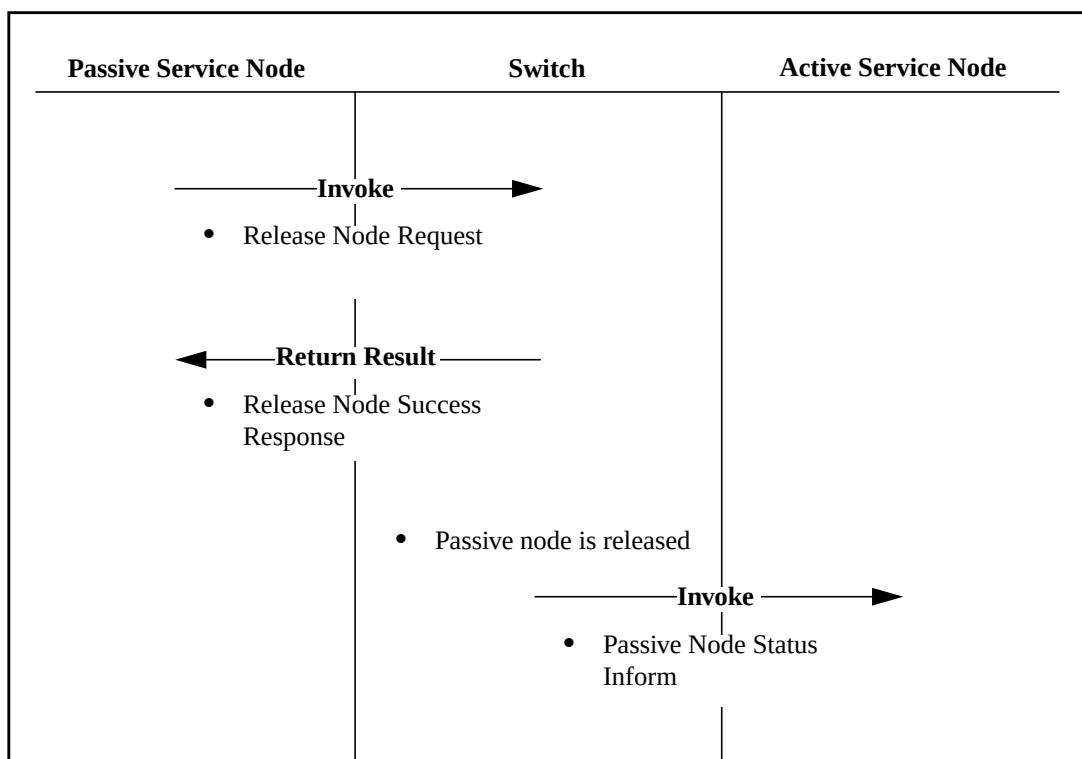
There are four scenarios in which a passive function provider will be released:

- The passive function provider can request to release itself.
- The active service node can request to have the passive function provider released.
- The passive function provider is automatically released by the switch when the active service node requests a call disposition (e.g. Float Call Request).
- The switch will release the passive function provider if it encounters an error while attempting to send a message to it.

A call flow for each of the scenarios listed above is given in this section.

12.10.2.1 Releasing a Passive Node at the Passive Node's Request

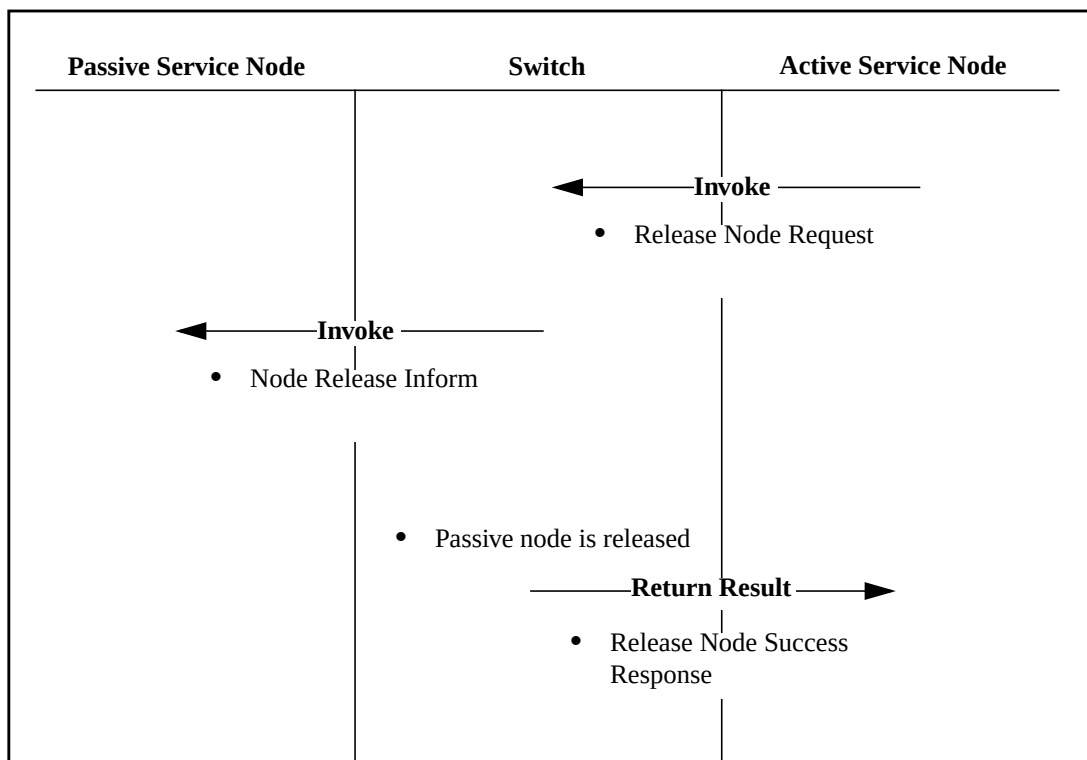
A passive node may request to have itself released from the call. The active service node is informed of the release of the passive node and the call continues. The message flow shown below assumes that a passive service node was connected to the call, although the passive function provider could have been a TOPS operator. However, since the call flow from the active service node's point of view is the same in both cases, the operator release flow is not shown.



12.10.2.2 Releasing a Passive Node at the Active Node's Request

The active service node may request to have the passive node released from the call. The passive node is informed that it is being released and the call continues. The message flow shown below assumes that a passive service node was connected to the call, although the passive function provider could have been a TOPS operator. However, since the call flow from the active service node's point of view is the same in both cases, the operator release flow is not shown.

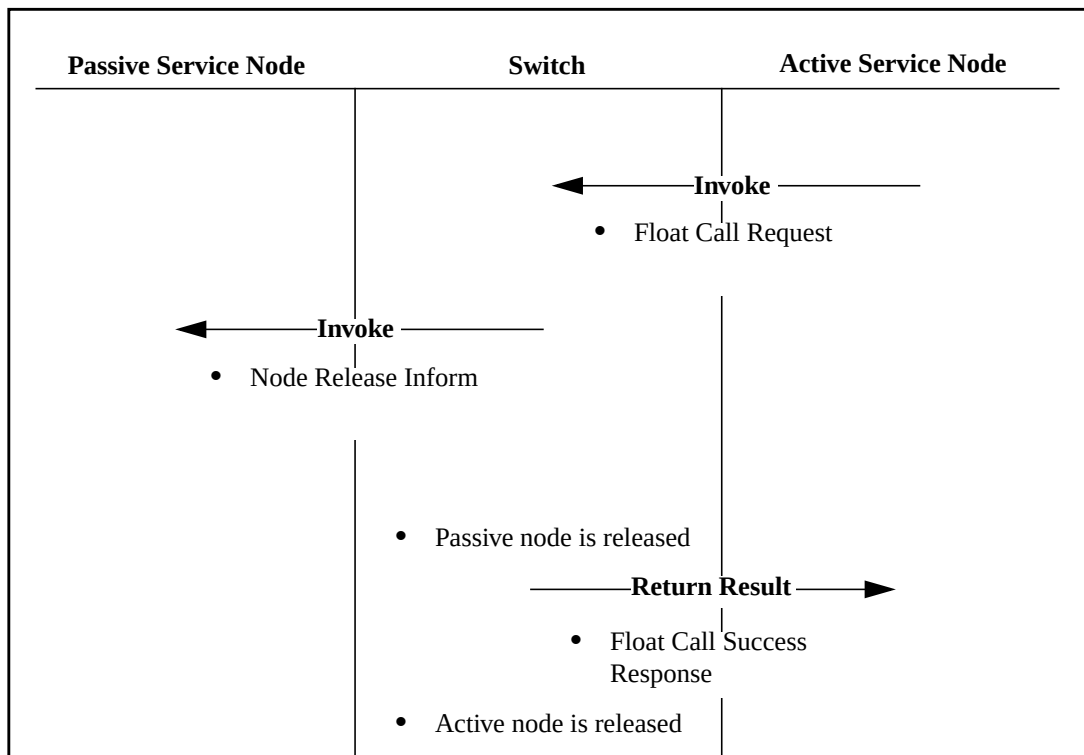
Although this flow assumes that a passive service node was connected to the call, it is also possible that the call could be queued in the switch awaiting the next available session with the passive node. If this were the case, the message flow would look the same to the active service node, but no messaging would take place with the passive node since it hadn't been connected yet.



12.10.3 Releasing a Passive Node Due to a Call Disposition

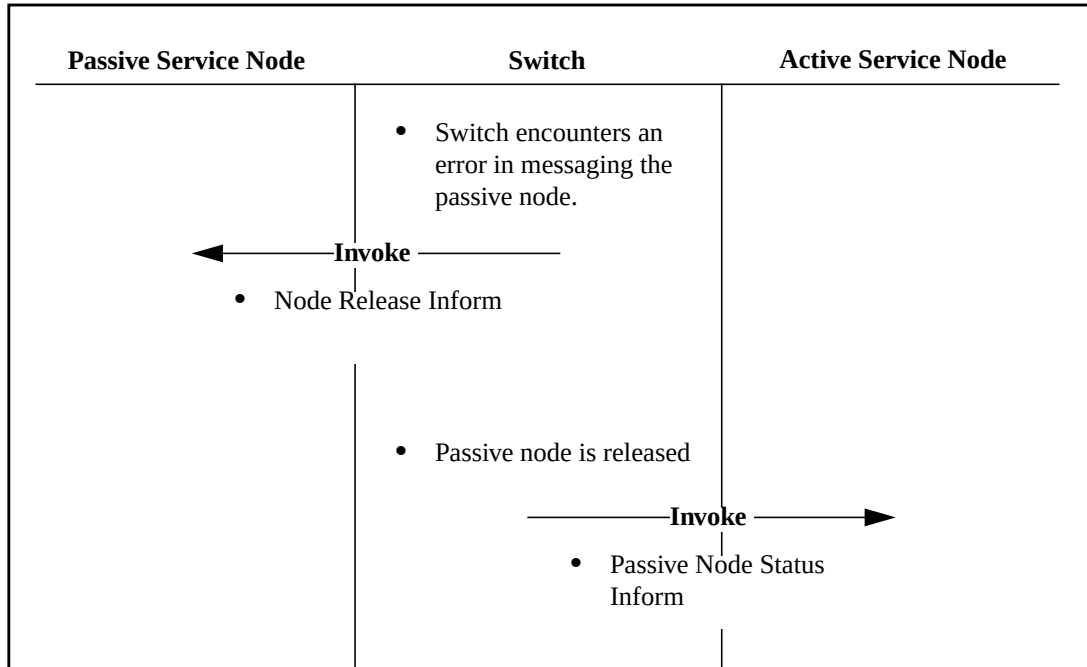
The active service node may request a call disposition (e.g. Float Call Request), which results in both the active and passive node being released from the call. The passive node is informed that it is being released and the call disposition continues. The message flow shown below assumes that a passive service node was connected to the call, although the passive function provider could have been a TOPS operator. However, since the call flow from the active service node's point of view is the same in both cases, the operator release flow is not shown.

Although this flow assumes that a passive service node was connected to the call, it is also possible that the call could be queued in the switch awaiting the next available session with the passive node. If this were the case, the message flow would look the same to the active service node, but no messaging would take place with the passive node since it hadn't been connected yet.



12.10.4 Switch Releasing a Passive Node Due to an Error Condition

The switch will release the passive node if it encounters an error in trying to send a message to it. The passive node is informed that it is being released and the active service node is informed that the passive node is being released. The message flow shown below assumes that a passive service node was connected to the call, although the passive function provider could have been a TOPS operator. However, since the call flow from the active service node's point of view is the same in both cases, the operator release flow is not shown.



12.10.5 Passing Call Control

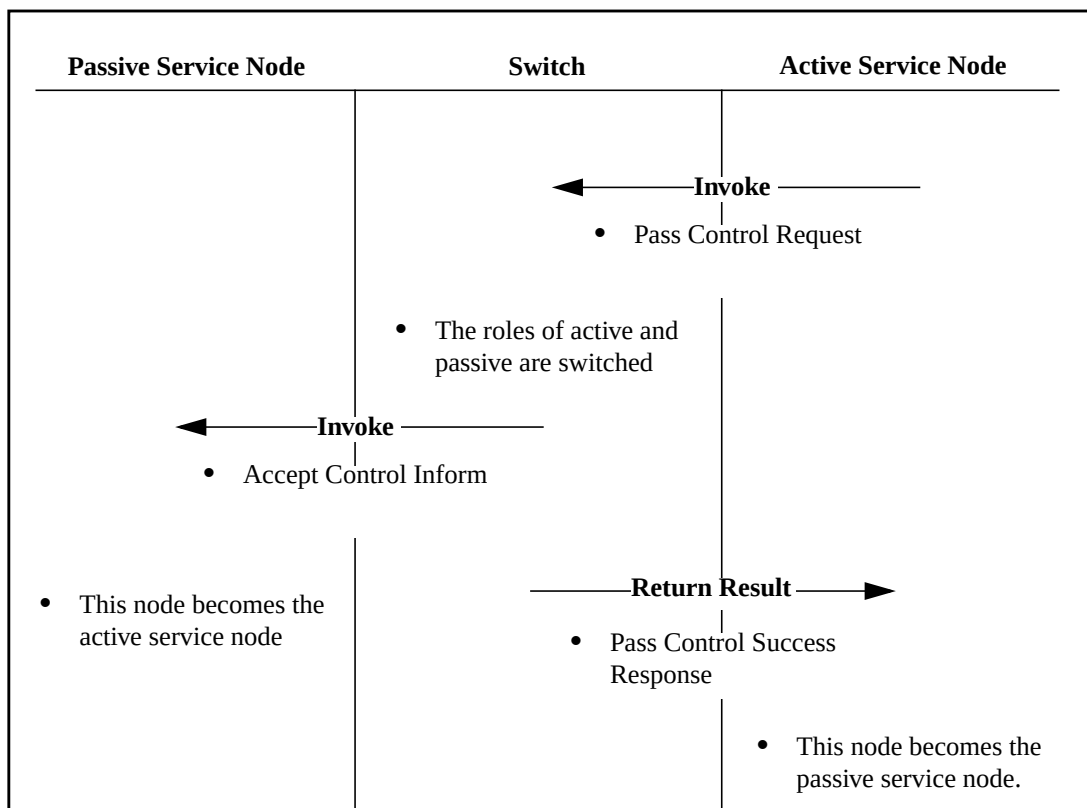
Call control capabilities can be passed from the active service node to the passive node in one of three ways:

- The active service node can request to pass call control capabilities to the passive node. This can only be done if the passive node is a service node (i.e. this form of passing control is not allowed between an active service node and a passive operator). If control is passed in this way, the active service node becomes the passive service node and the passive service node becomes the active service node.
- The active service node can request to release itself, thus leaving the passive node in complete control of the call. This form of passing call control is allowed with both passive service nodes and passive operators.
- A passive operator can request to have the active service node released from the call, thus leaving the operator in complete control of the call. This form of passing call control is not allowed from passive service nodes.

12.10.6 Pass Control at Active Service Node's Request

If both nodes connected to the call are service nodes (i.e. a passive operator is not connected), the active service node can request to have call control passed to the passive service node. The passive service node receives complete call details and becomes the active service node for the call.

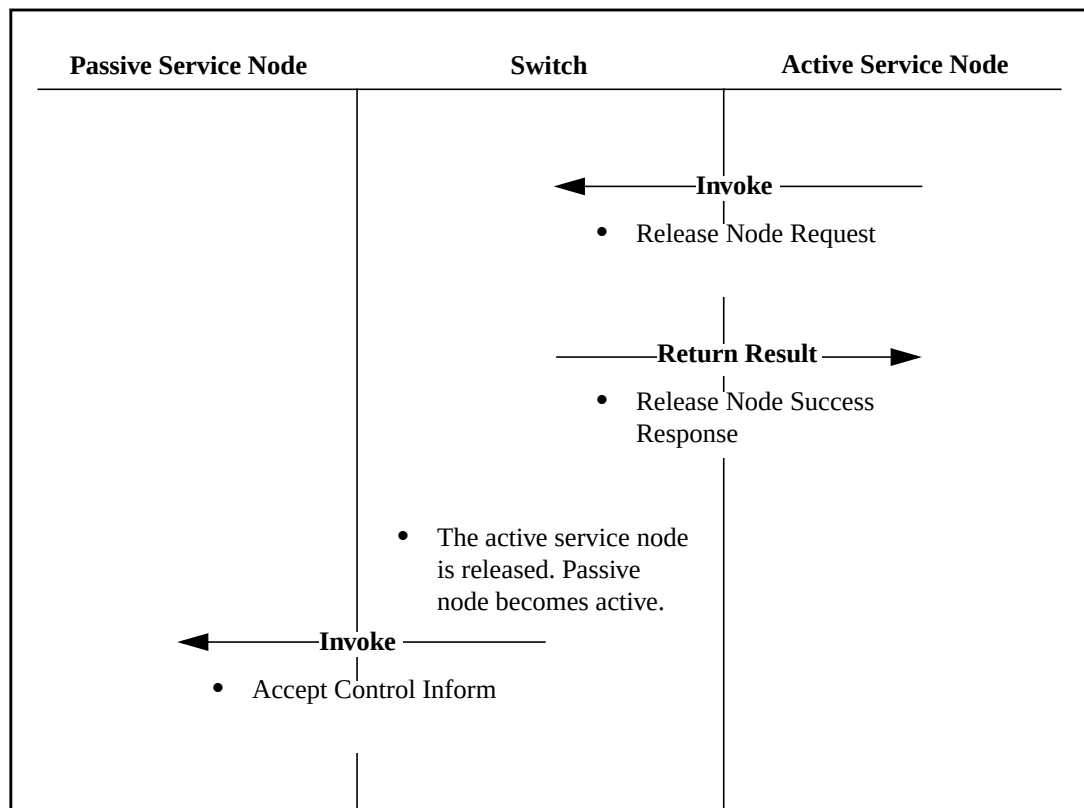
This call flow assumes that the passive service node was already connected to the call, although it is possible that the call could have been queued for a session with the passive node. If that were the case, the message flow would look the same to the active service node, but the Accept Control Inform would not be sent to the passive service node since it is not yet connected. Once a session with the passive service node is established, the passive service node will receive a Session Begin which will have an indication that it is the active service node for the call.



12.10.7 Pass Control Due to Active Service Node Release

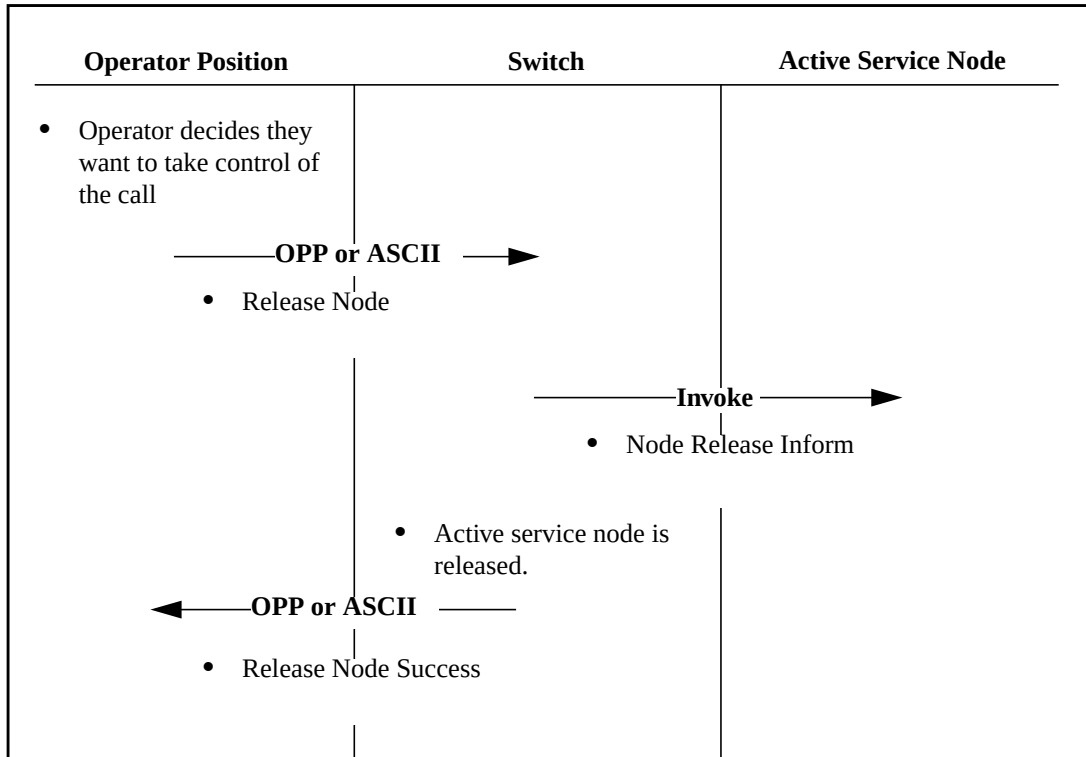
The active service node can pass control to the passive node by releasing itself from the call. The passive node receives an indication that it is now the only node connected to the call, thus giving it complete call control capabilities. The following flow assumes that a passive service node was connected to the call, although it is possible that a passive operator could have been connected instead. Since the message flow from the active service node's point of view is the same in both cases, the passive operator flow was not shown.

This call flow assumes that the passive node was already connected to the call, although it is possible that the call could have been queued for a passive node. If that were the case, the message flow would look the same to the active service node, but no messaging would take place with the passive node since it is not yet connected. Once the passive node is connected, it will receive either an OAP Session Begin (for a service node) or a Call Begin (for an operator) indicating that it is the only node involved in the call.



12.10.8 Pass Control at Passive Operator's Request

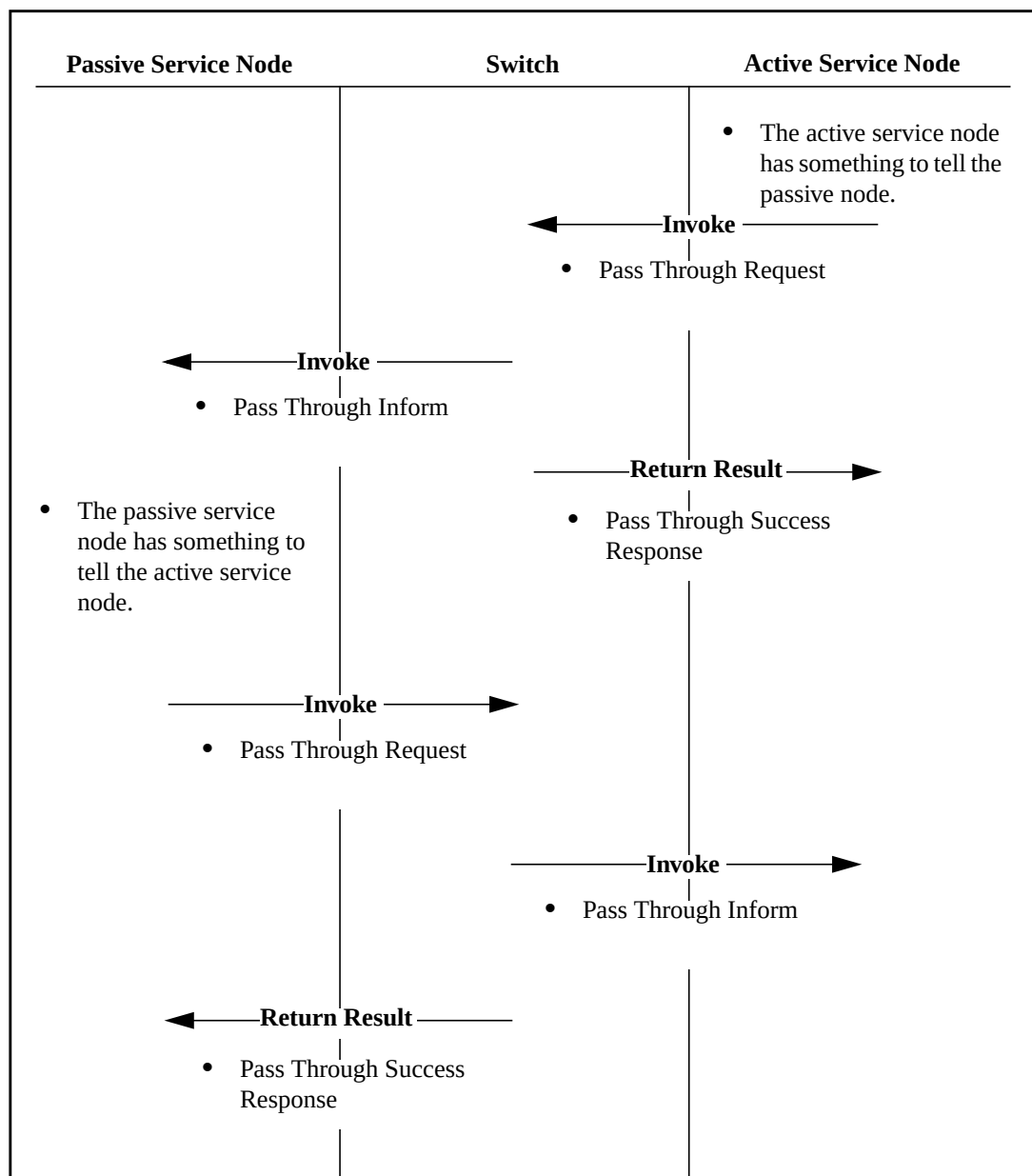
A passive operator may request that the active service node be released from the call, thus leaving the operator in complete control of the call. The active service node is informed that it is being released.



12.10.9 Pass Through Messaging

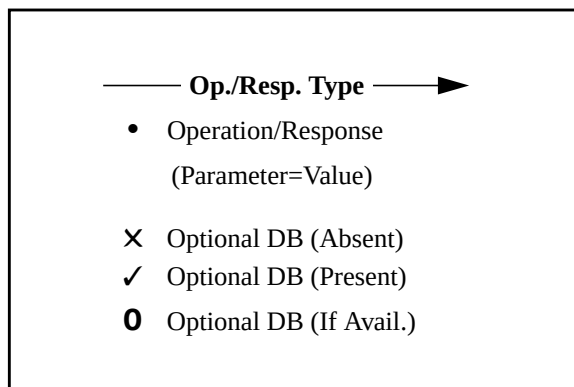
Pass through messages can be exchanged between an active service node and a passive service node or between an active service node and a passive operator. The format of the data in the pass through message is unknown to the switch, the switch merely acts as an intermediary that forwards the pass through message to the appropriate node.

Although this flow shows pass through messaging between two service nodes, pass through messaging could also take place between an active service node and a passive operator. Since the messaging from the active service node's point of view is the same in either case, the passive operator flow is not shown.



12.11 Speech Path Call Flows

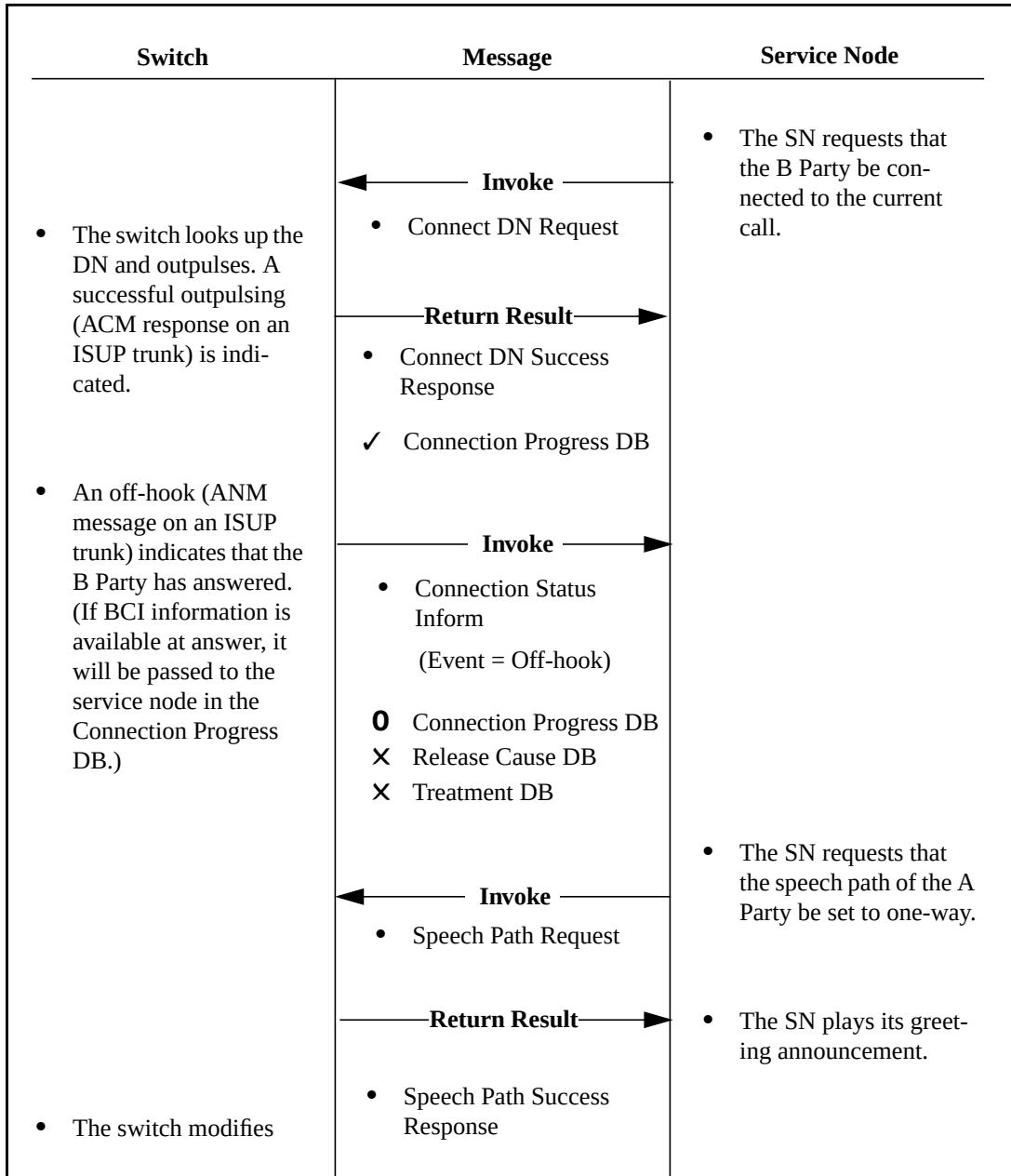
The following flow diagrams include significant parameters and optional data blocks, in addition to the usual operations/response type information, for purposes of call flow clarification. For individual messages in the call flow figures, these are shown as follows:



Note: Optional DBs marked 'O' are sent if, under the particular network condition, the switch has the optional data available to send.

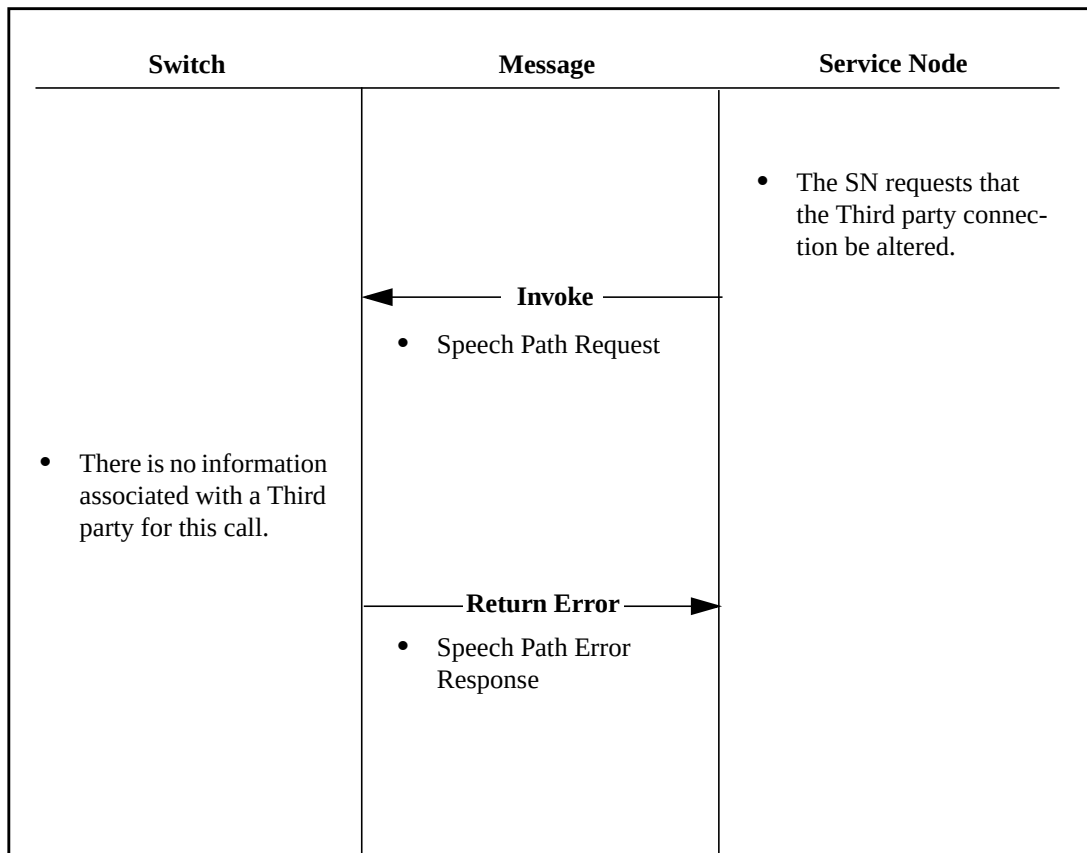
12.11.1 Speech Path Request (Success)

This following flow fragment is from an automated Collect call. Following a successful connection to the B Party, but prior to billing acceptance, the SN might alter the speech status to one-way for the A Party (can listen, but not speak). In this scenario, the B Party's DN has already been forwarded to the switch.



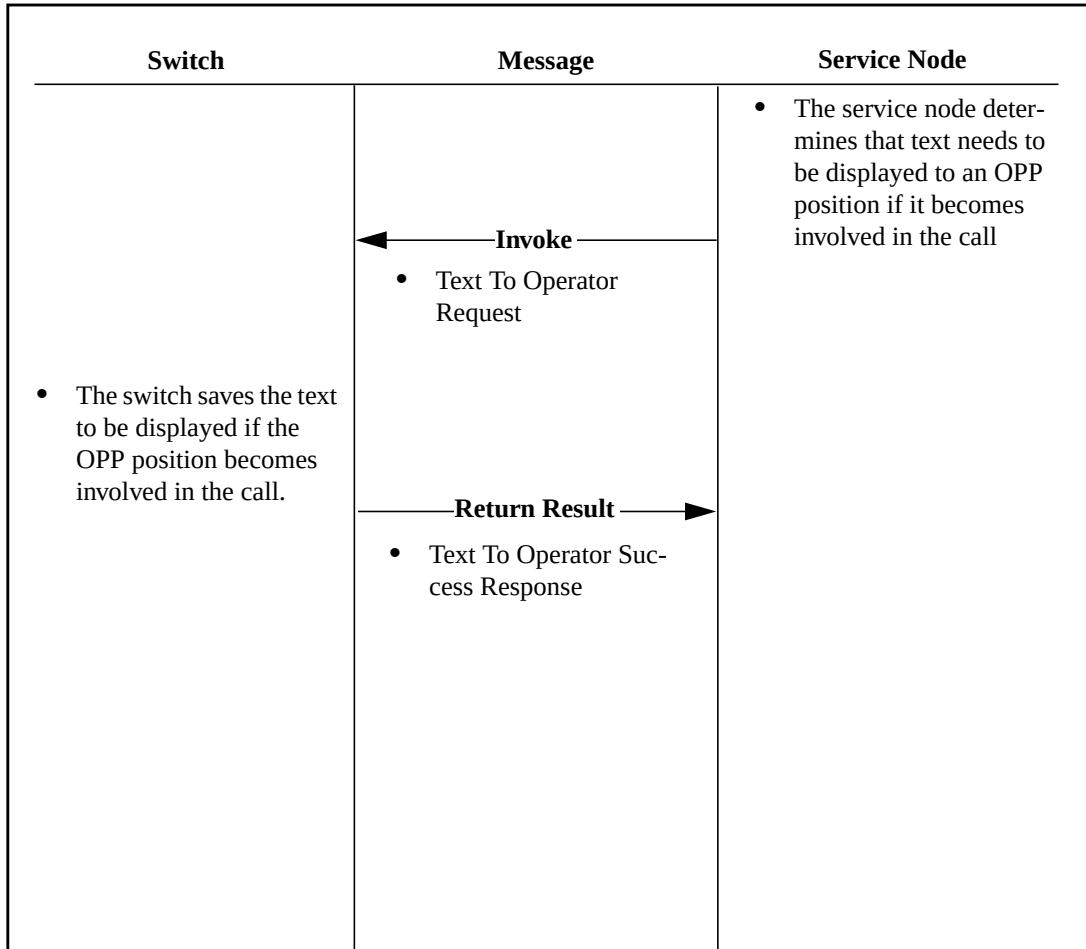
12.11.2 Speech Path Request (Failure)

The SN needs to indicate the appropriate party to be affected by the Speech Path Request. In the following scenario the SN failed to select a valid party.

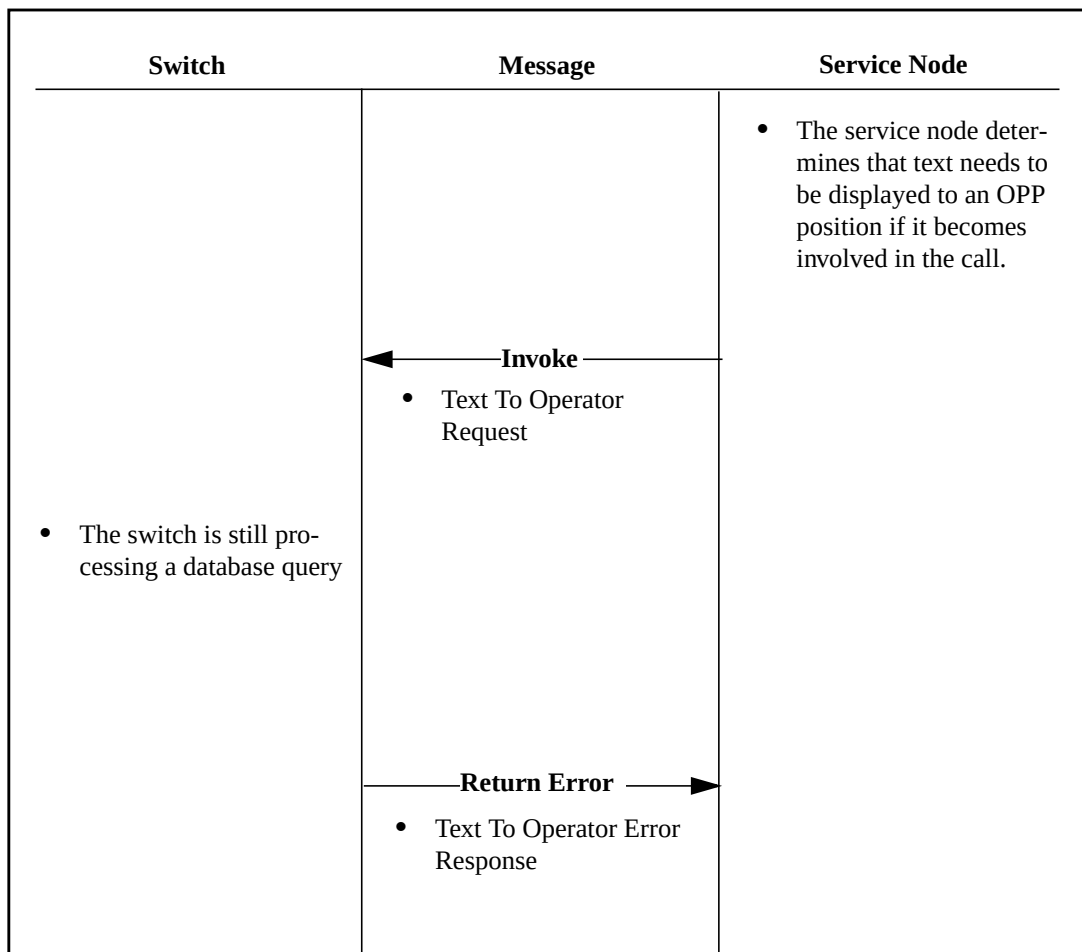


12.12 Text To Operator Message Flows

12.12.1 Text To Operator (Success)



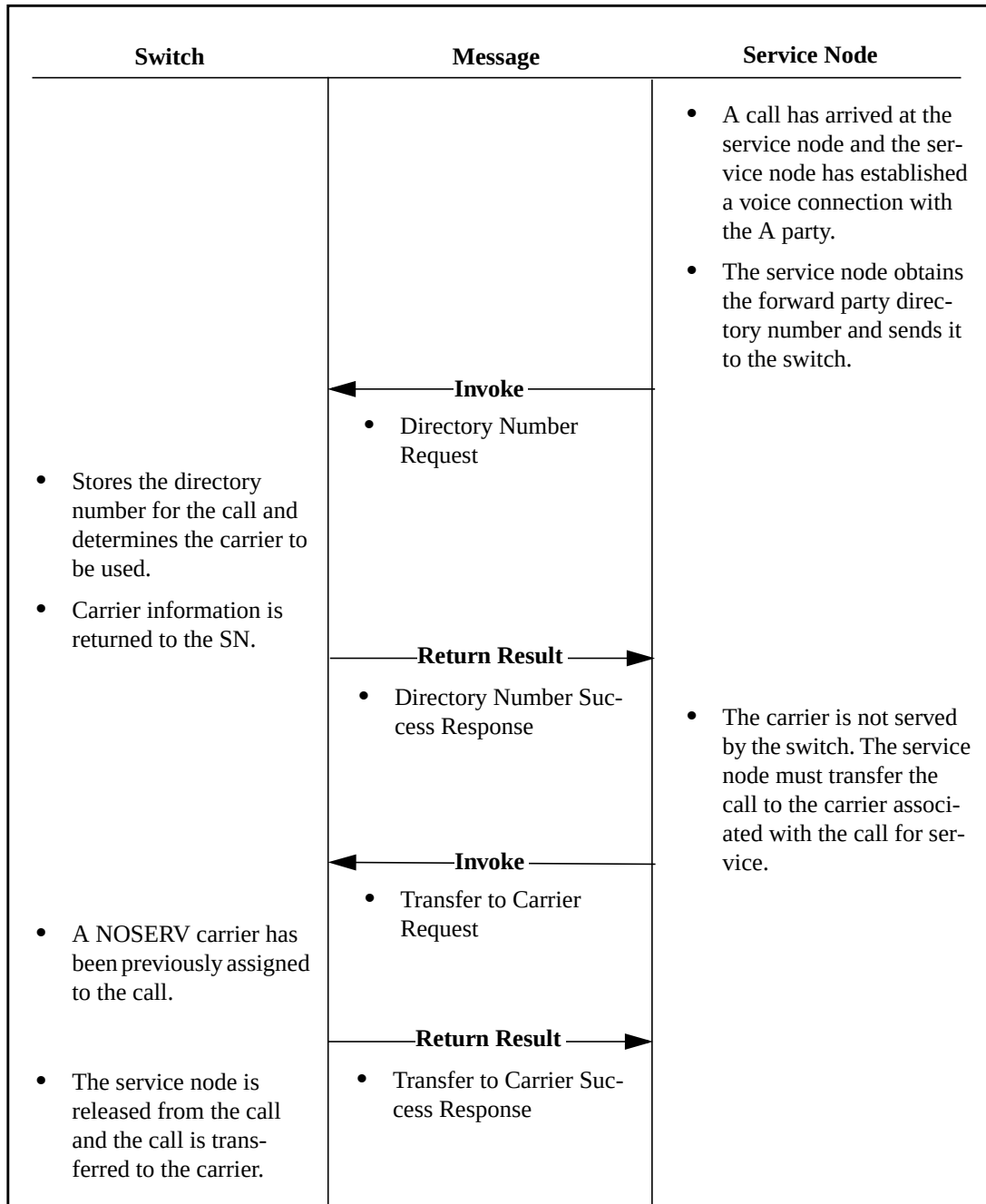
12.12.2 Text To Operator (Failure)



12.13 Transfer to Carrier Message Flows

12.13.1 Transfer to Carrier (Success)

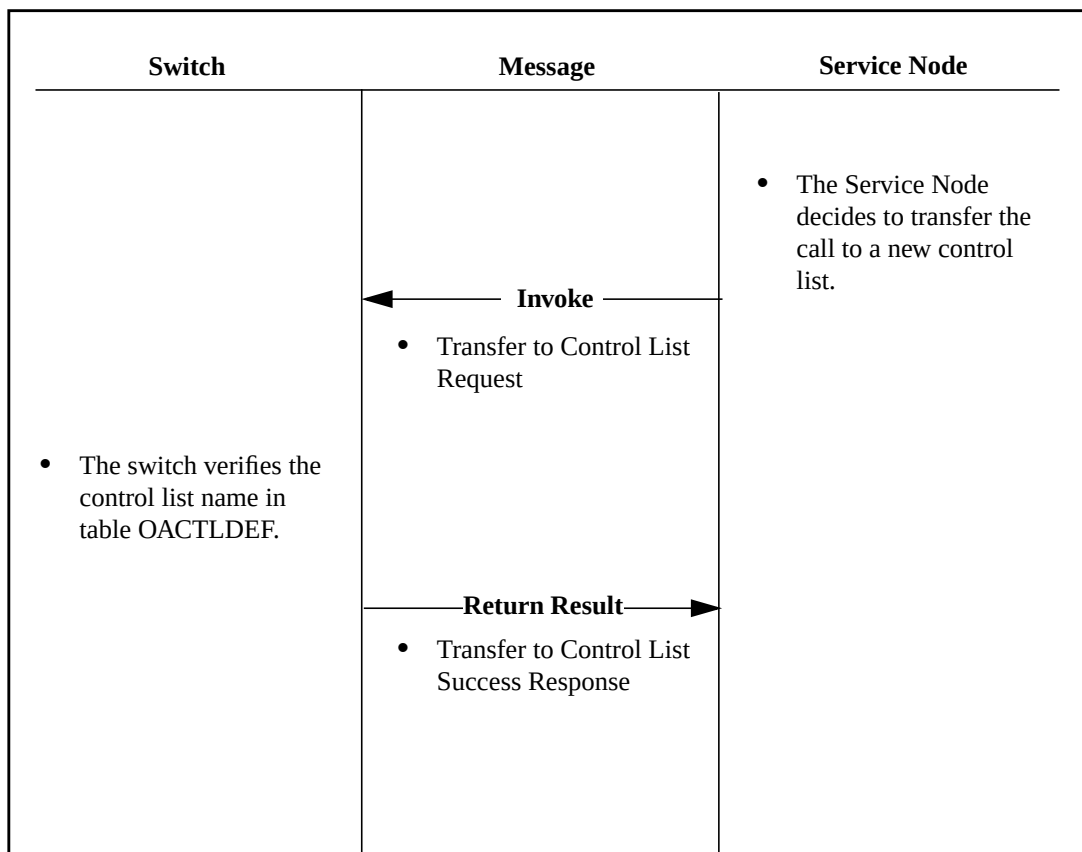
The Service Node wants the call to be transferred to the carrier for service.



12.14 Transfer to Function Provider Message Flows

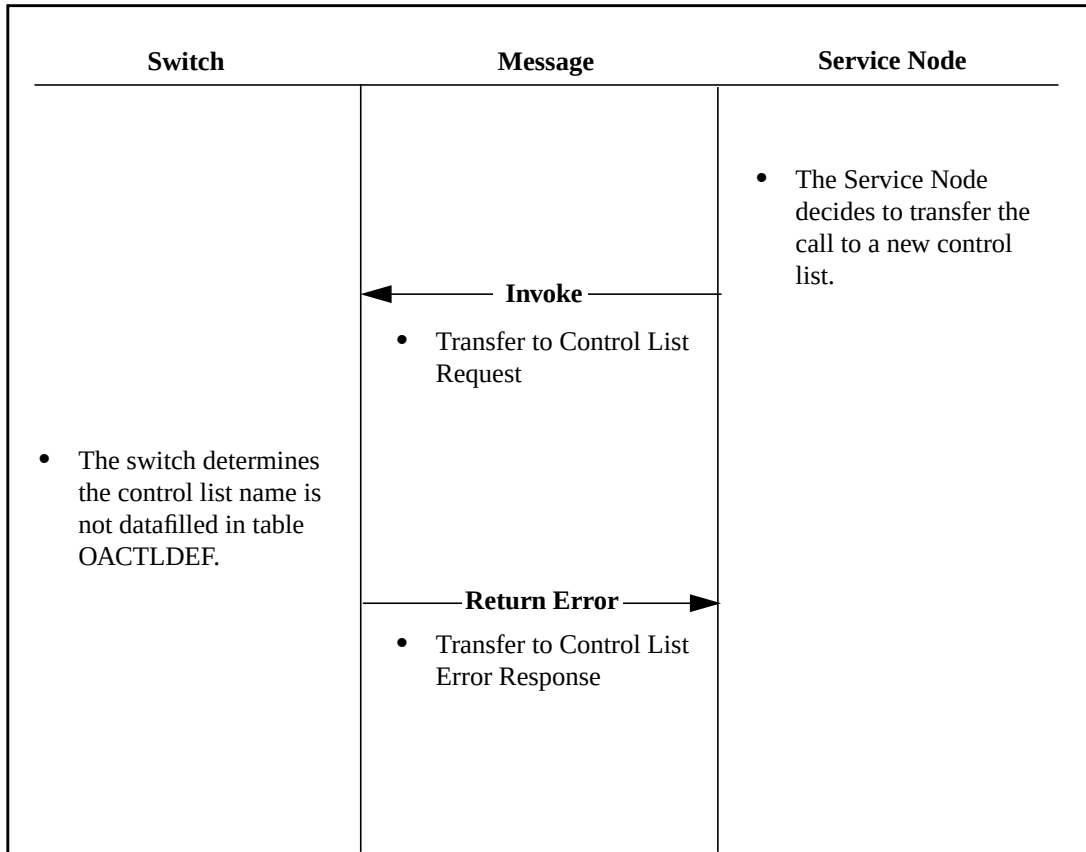
12.14.1 Transfer to Control List (Success)

The Service Node wants the call flow to move explicitly to a new control list and the unconfirmed transfer is allowed.



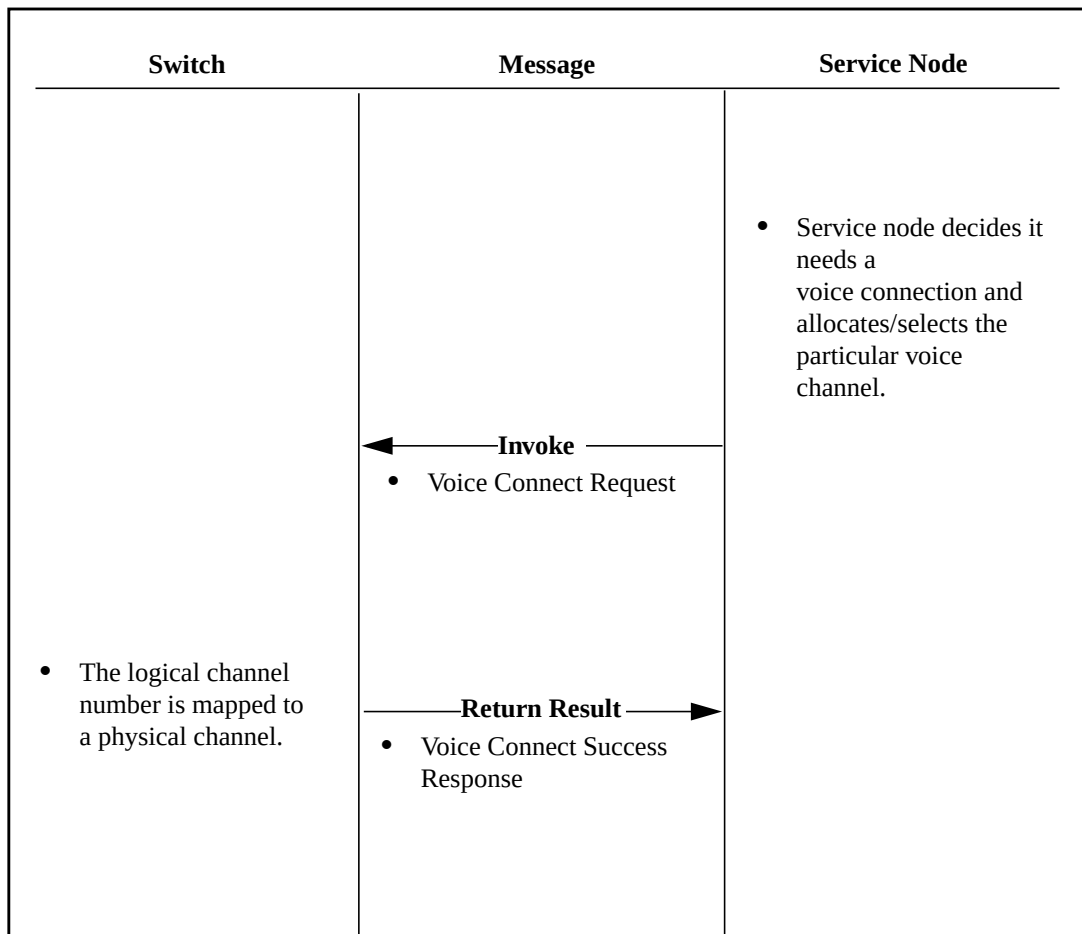
12.14.2 Transfer to Control List (Failure)

The Service Node wants the call flow to move explicitly to a new control list and the unconfirmed transfer is not allowed.

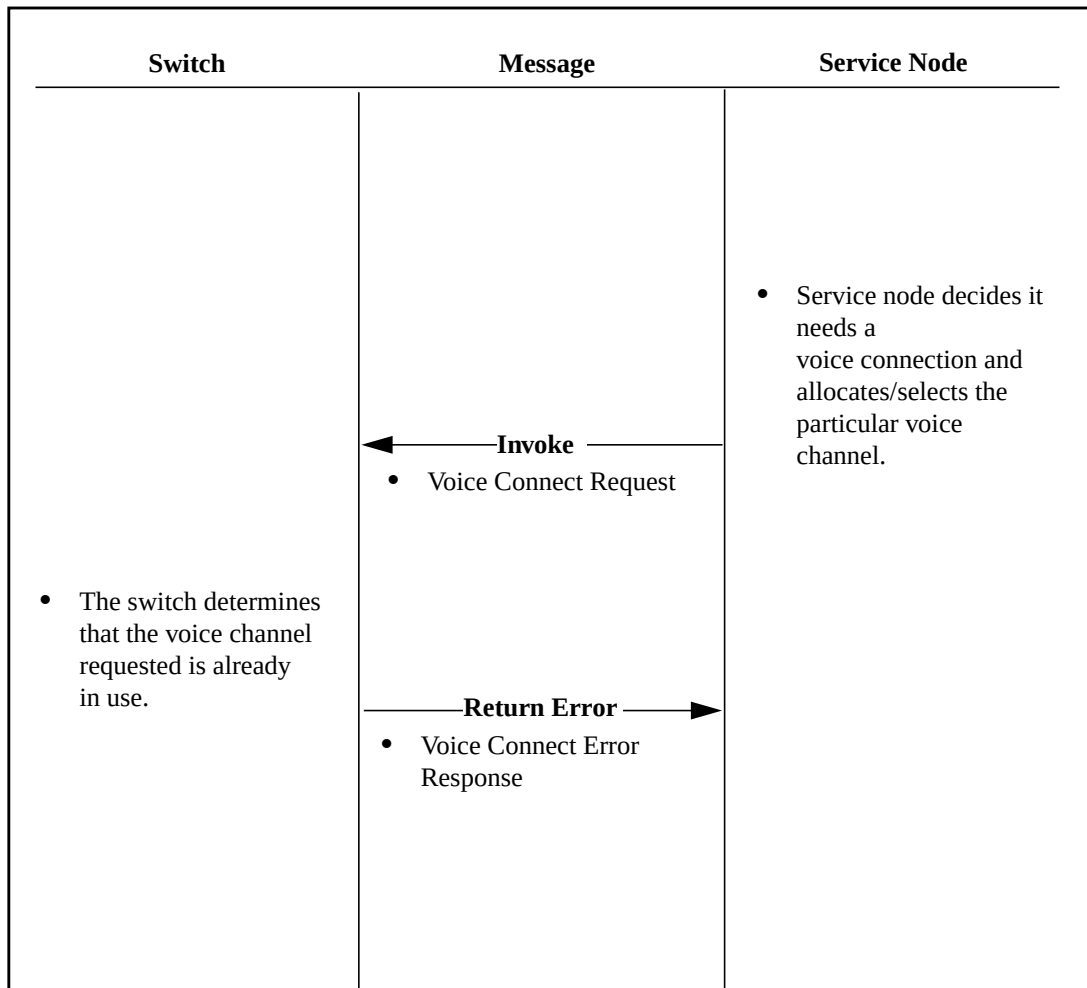


12.15 Voice Link Message Flows

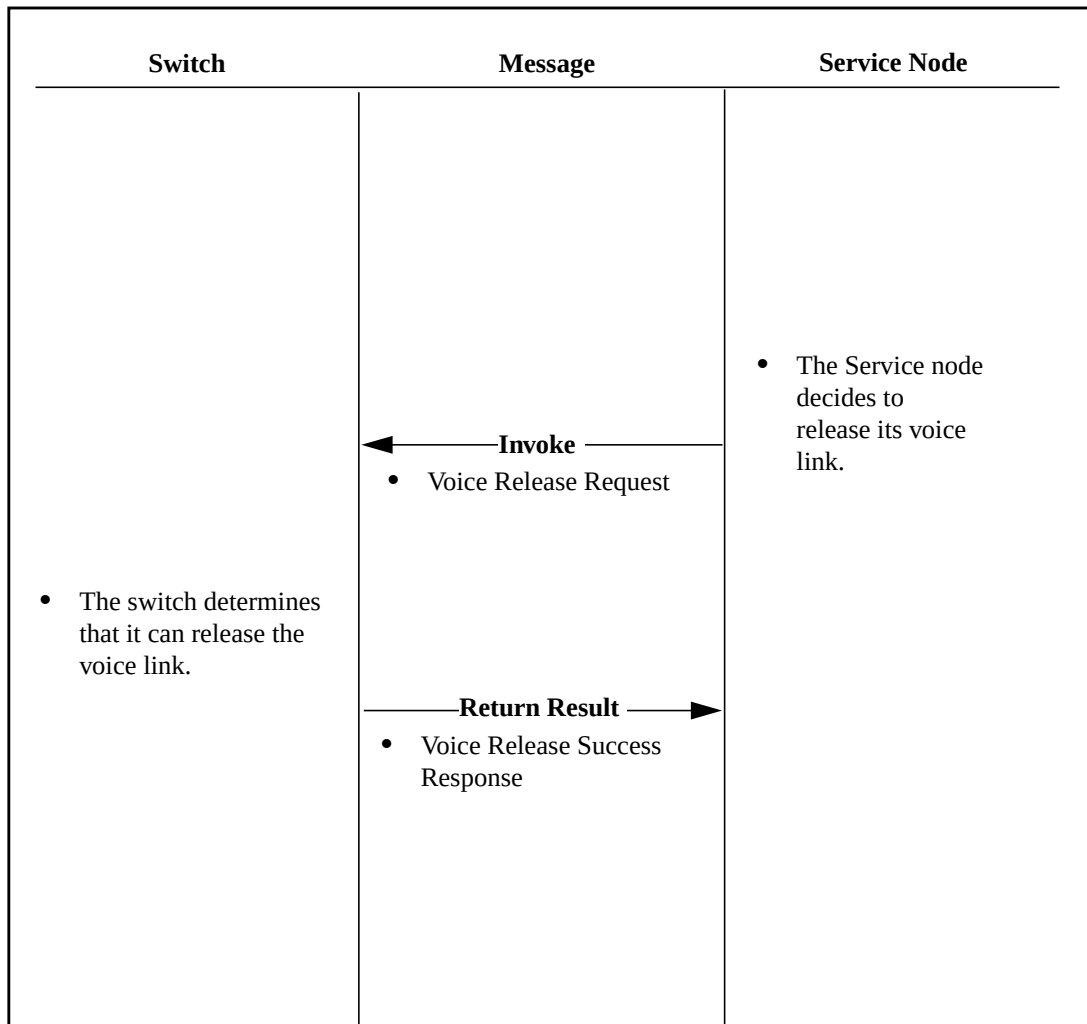
12.15.1 Connect Voice Link (Success)



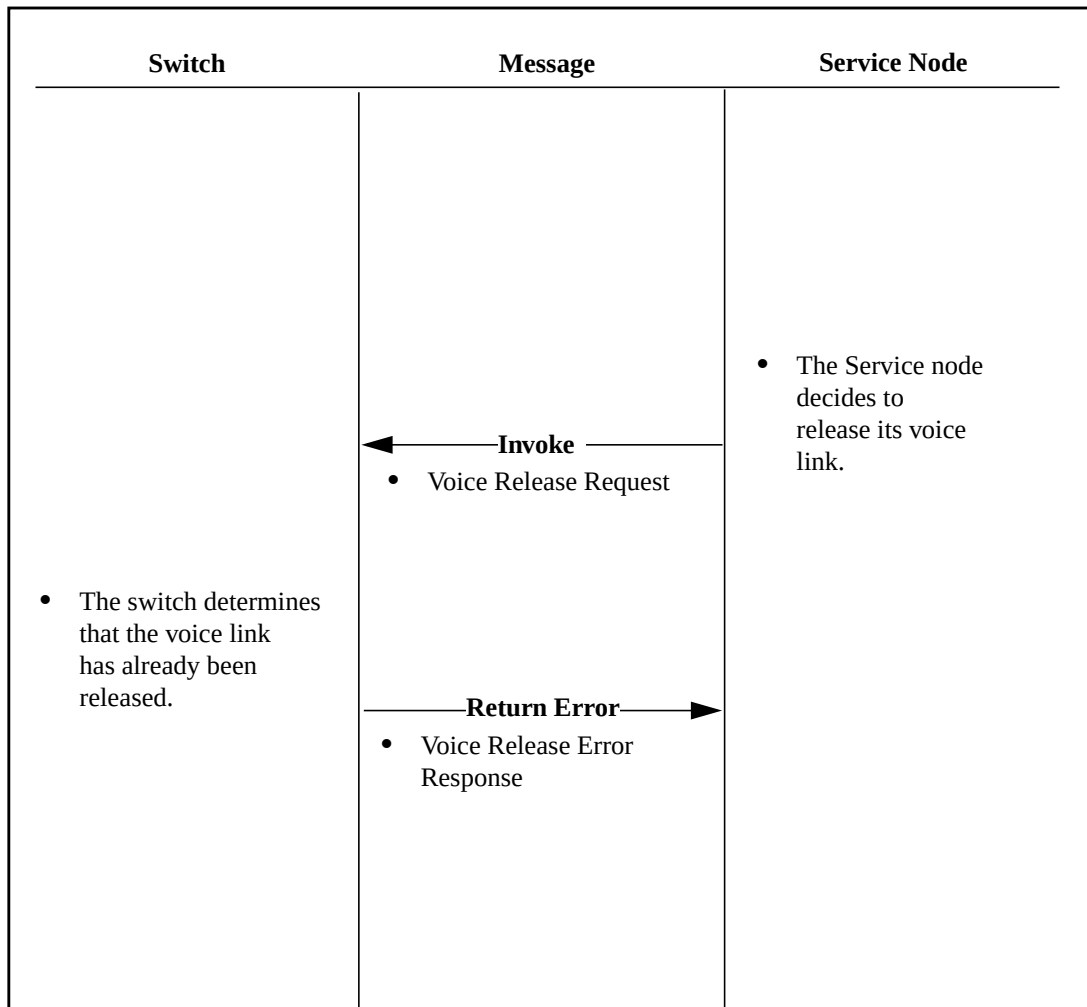
12.15.2 Connect Voice Link (Failure)



12.15.3 Release Voice Link (Success)



12.15.4 Release Voice Link (Failure)



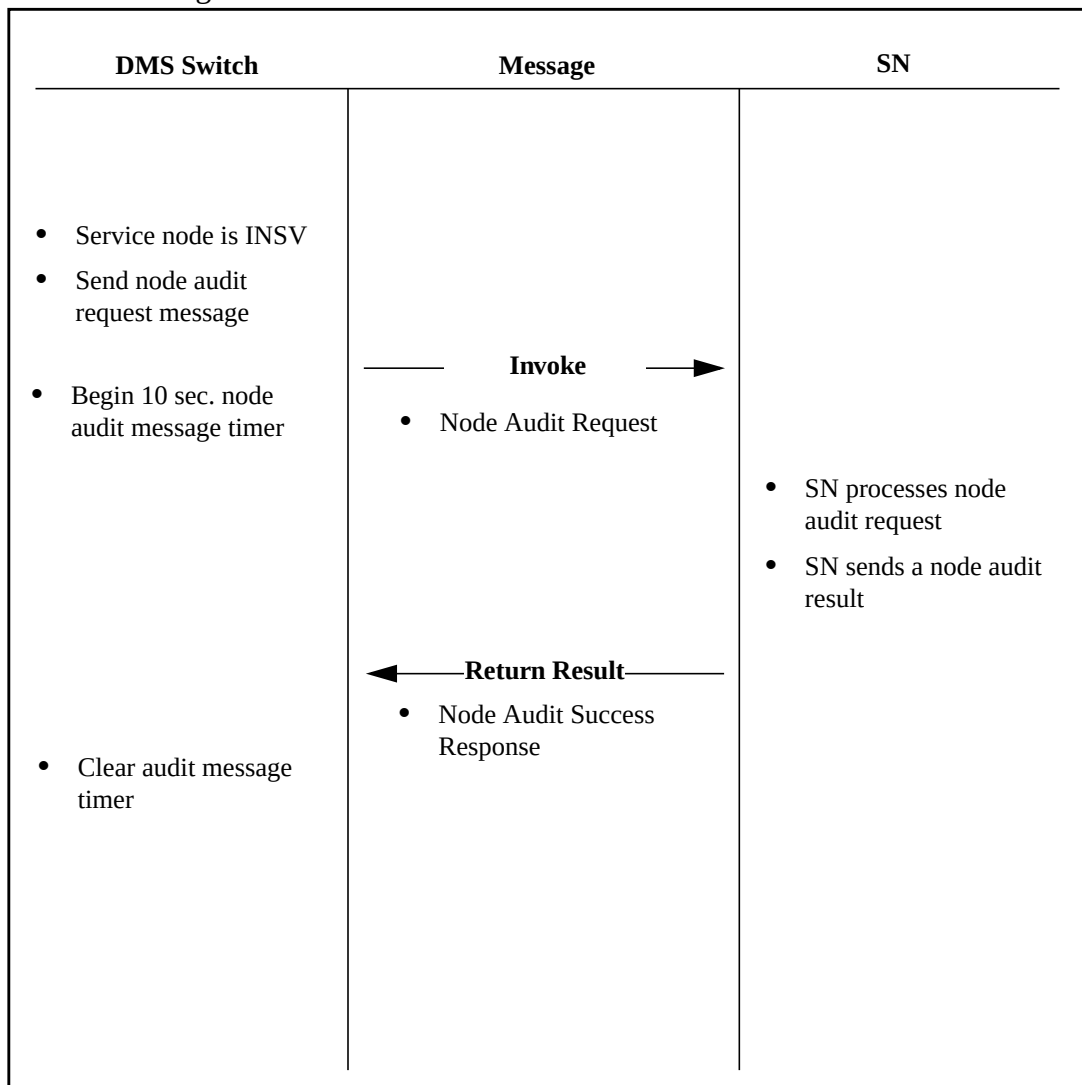
12.16 Non-Callp Node Based Message Flows

12.16.1 Node Audit Request - (Success)

The Maintenance Switch only sends this message to a service node if:

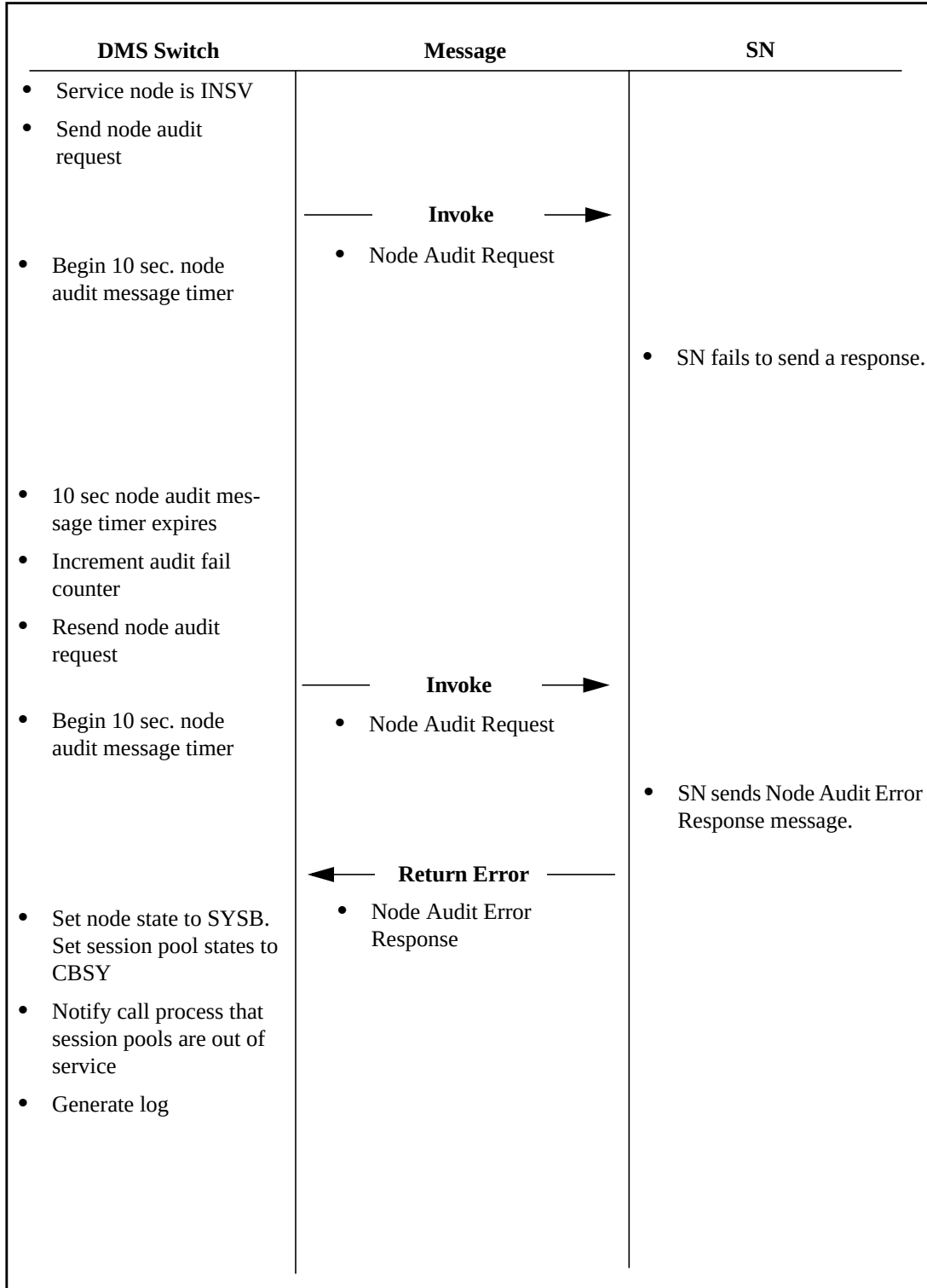
- no call processing messages have been received from that service node within a period defined by AUDFREQ in table OANODINV or,
- no audit messages have been sent to that service node for 5 times the time period defined by AUDFREQ in table OANODINV.

If messages have been received, the SN is considered to be in service. Additionally, the SN must be in either INSV or ISTB state in order to receive a node audit message. This example shows the case where no messages have been received and the SN is in the INSV state.



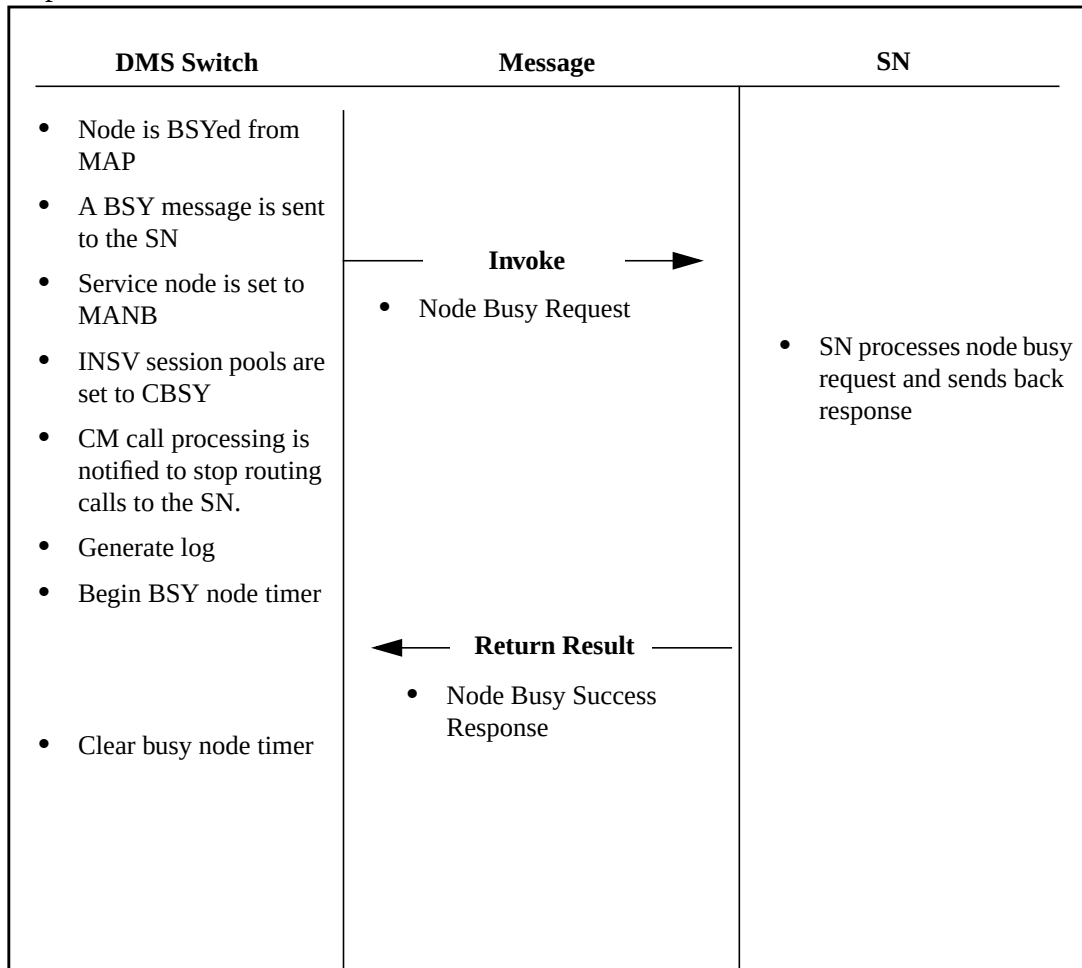
12.16.2 Node Audit Request - (Failure)

If a DMS Switch originated node audit fails, calls that have sessions with the unresponsive SN are taken down and new calls are prevented from being routed to the SN until it can be brought back into service.



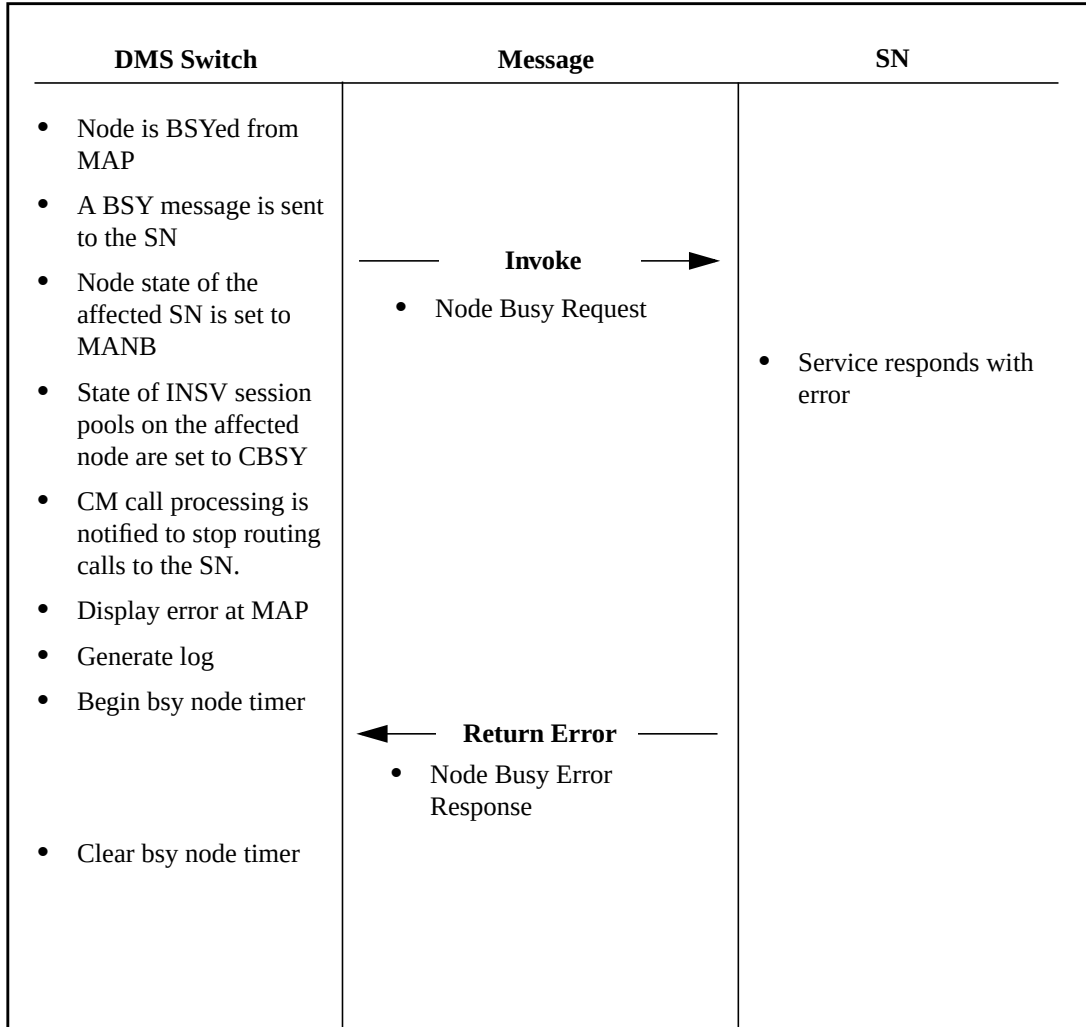
12.16.3 Node Busy Request by Switch - (Success)

This flow shows the activity in busying a SN at the switch. A crafts person enters a BSY command at the DMS Switch MAP to manually busy a SN. When a service node is busied, node level auditing of the service node is halted. In addition, any INSV session pools resident on the affected node are put into CBSY state.



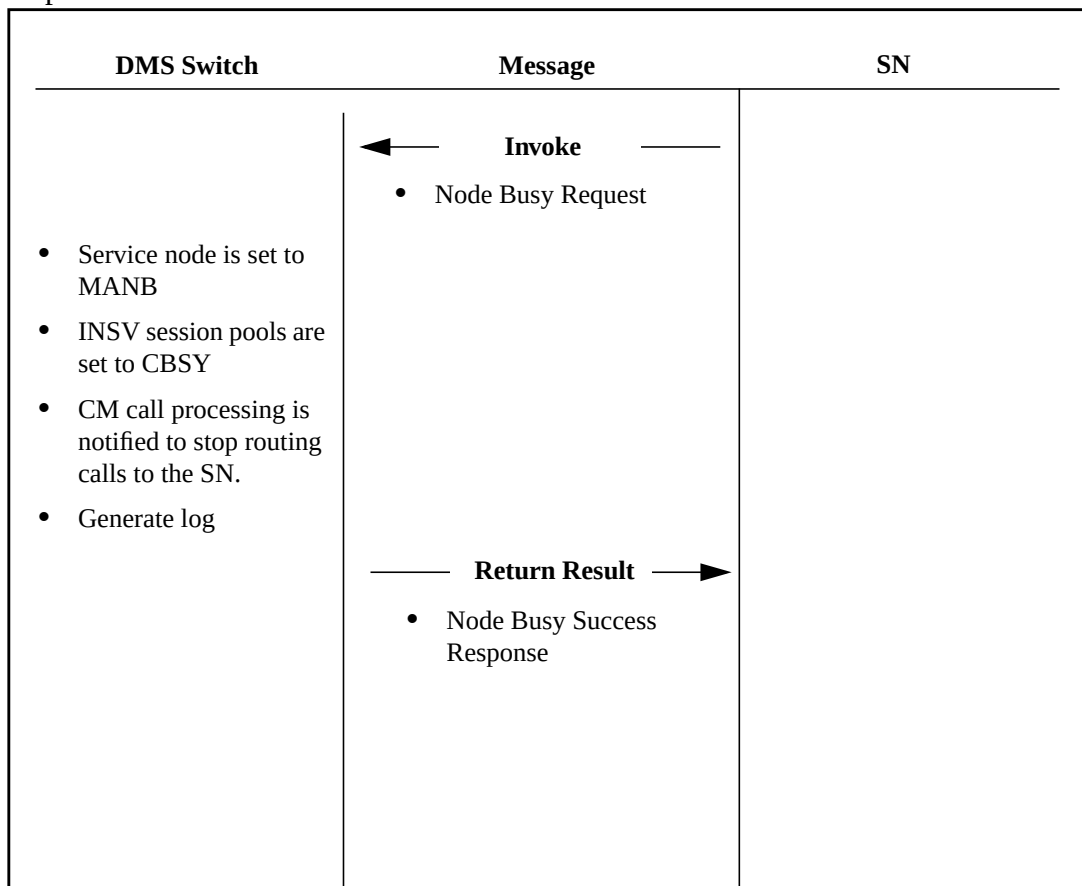
12.16.4 Node Busy Request by Switch - (Failure)

If no response to a Node Busy Request is received, or an error response is received, the DMS Switch will mark the SN as MANB and any INSV session pools resident on the affected node are put into CBSY state.



12.16.5 Node Busy Request by SN - (Success)

This flow shows the activity when an SN busy's itself. When a service node is busied, node level auditing of the service node is halted. In addition, any INSV session pools resident on the affected node are put into CBSY state.



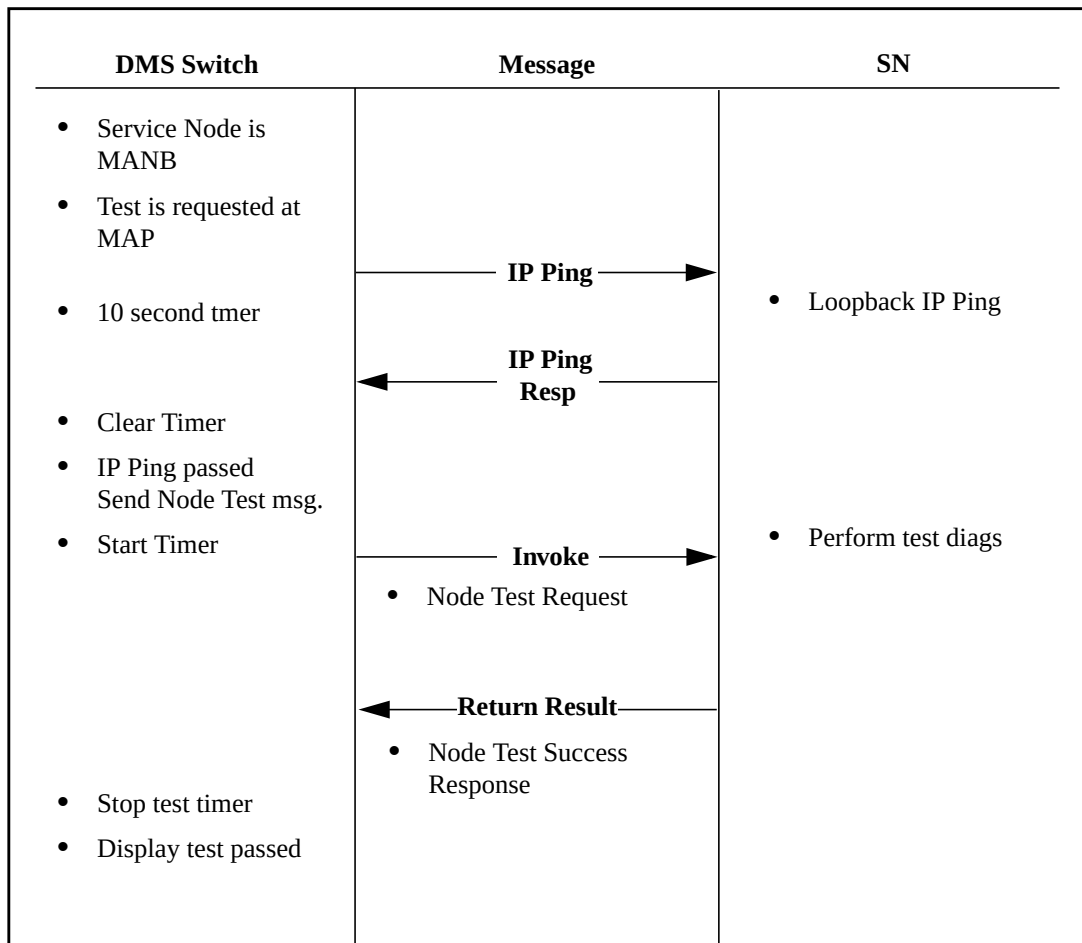
12.16.6 Node Test Request - (Success)

The DMS switch sends a node test message to the service node whenever a manual test (TST) is requested.

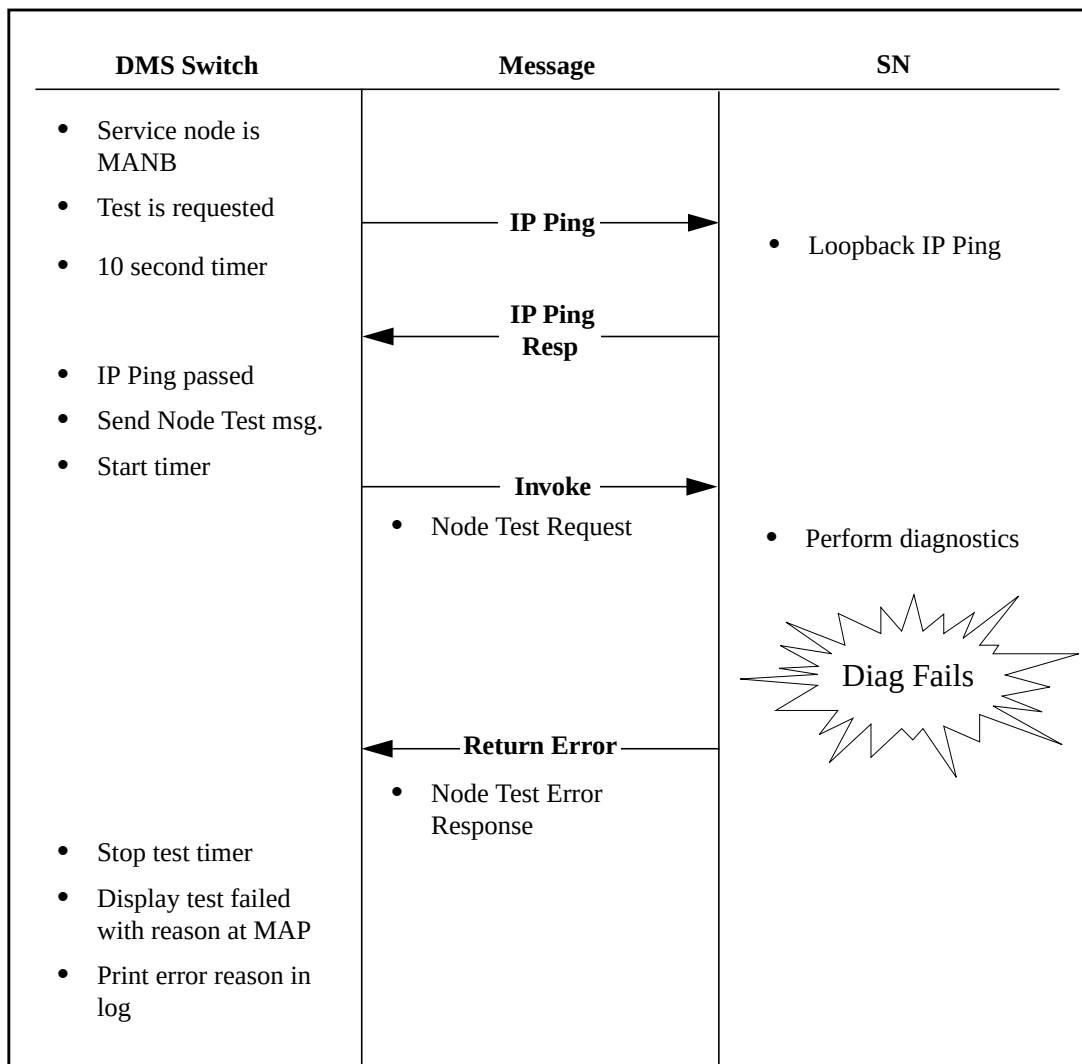
Prior to the node test message, an ICMP echo message (IP ping) will be sent from the DMS switch to the service node. Upon return of the IP ping, the node test message will be sent to the service node. The TSTTIME parameter in table OANODINV defines the amount of time the DMS switch will wait for a response from the service node.

Whenever the service node receives the node test message, it has three options:

- Respond - test passed. At this point the test result is displayed at the MAP.
- Respond - test failed. The test failed message will include Maintenance Text data block to diagnostic troubleshooting information. A test failed notification is displayed at the MAP along with the Maintenance Text message. A log will also be generated which will include the Maintenance Text.
- No response - the DMS Switch will time out and will display a time-out error message at the MAP.

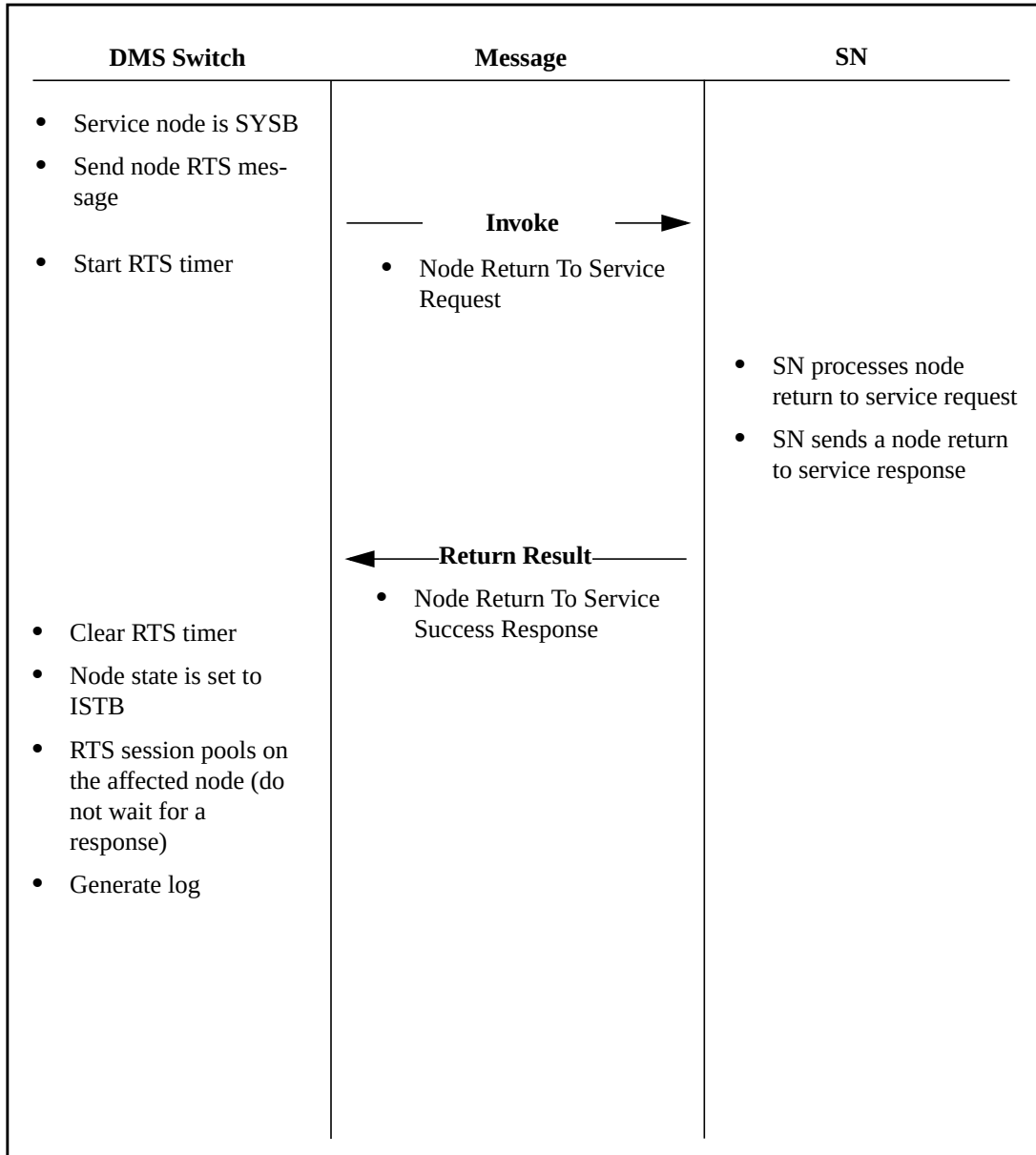


12.16.7 Node Test Request - (Failure)



12.16.8 Node Return To Service Request - Automated(Success)

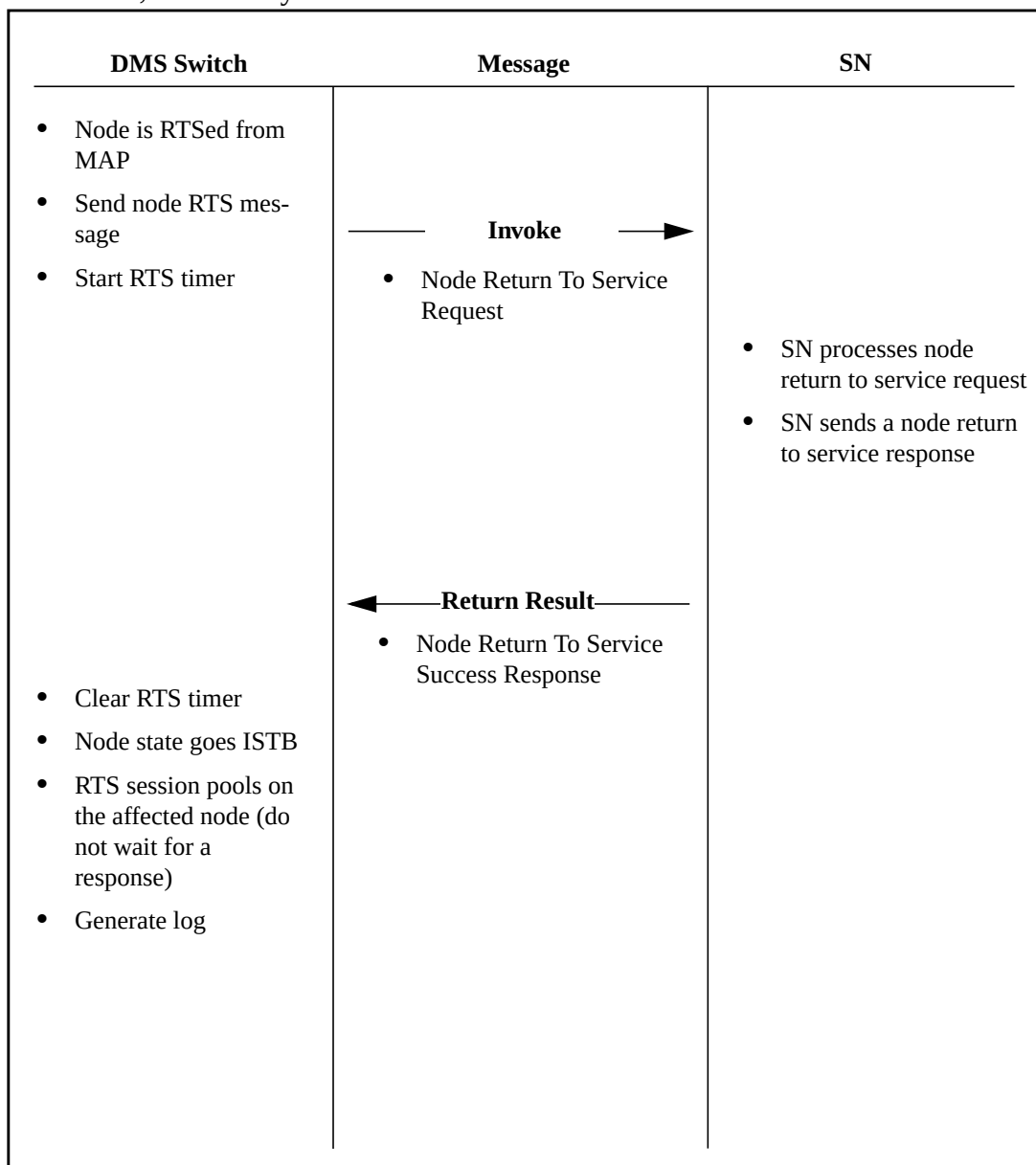
On a periodic basis, the DMS switch sends an automated node return to service message to every service node in the SYSB state. Once the SN is brought into service, a RTS request is sent to each session pool supported by the SN that is in the CBSY state. The SN is placed in the ISTB until all session pools are RTS'ed. If any of the session pools fail to RTS, the SN stays ISTB.



Note: Refer to 12.17.8 “Session Pool Return To Service Request - Automated (Success)” on page 873

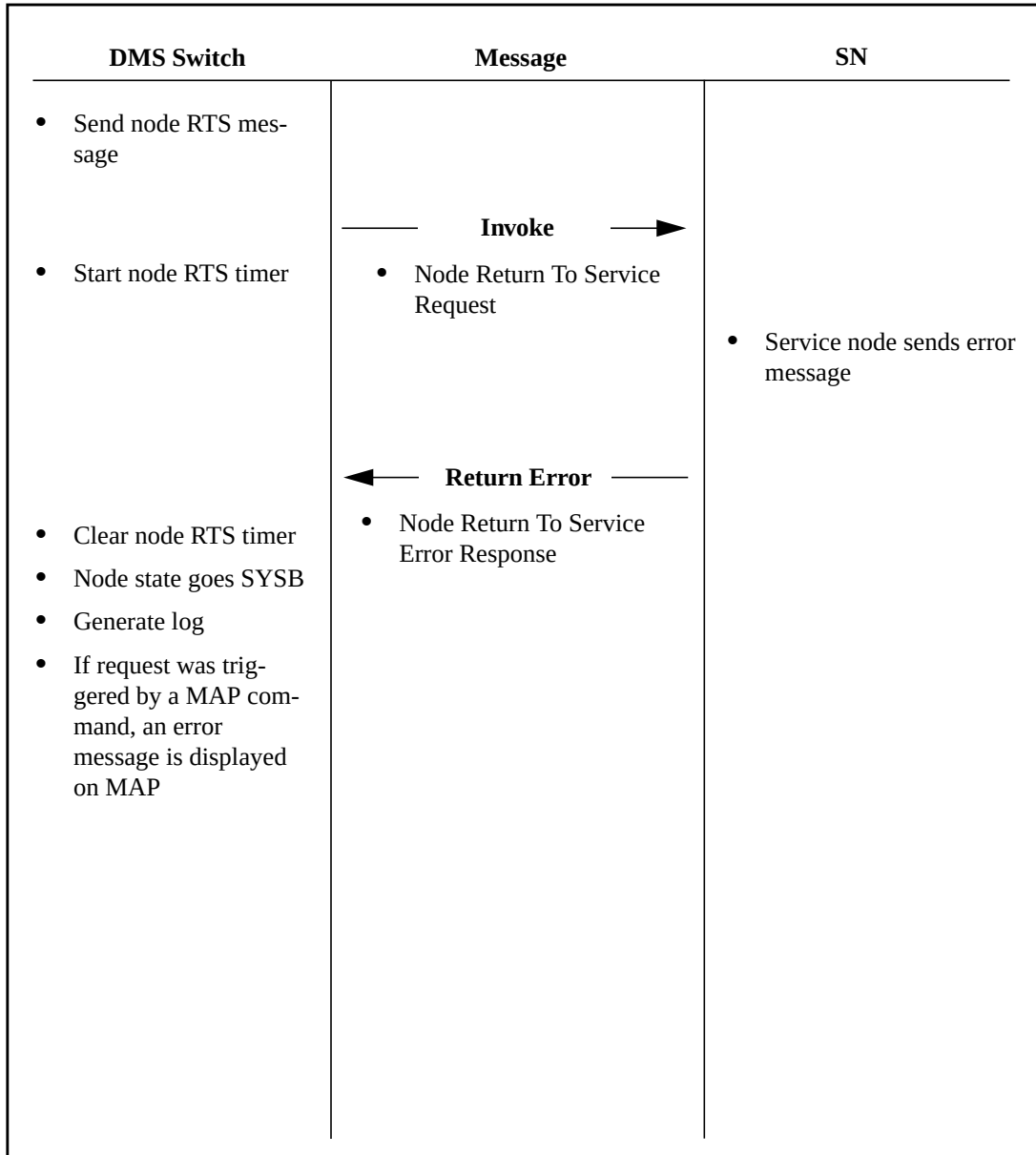
12.16.9 Node Return To Service Request - Manual (Success)

This example shows a DMS Switch requesting to RTS driven by a MAP request. Once the SN is brought into service, a RTS request is sent to each session pool supported by the SN that is in the CBSY state. The SN is placed in the ISTB until all session pools are RTS'd. If any of the session pools fail to RTS, the SN stays ISTB.



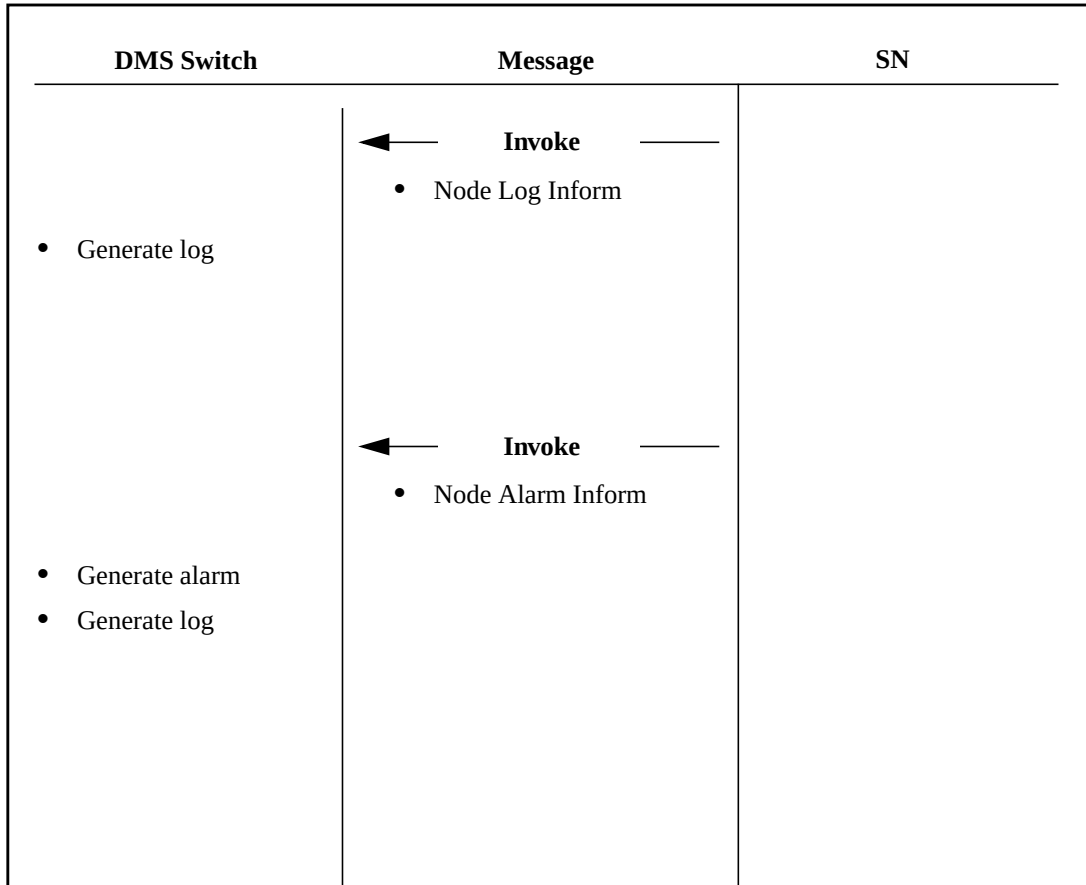
12.16.10 Node Return To Service Request - (Failure)

This example shows a DMS Switch requesting to RTS driven by a MAP request or expiration of RTSTIME from table OANODINV. In this flow the request fails. The SN node goes SYSB and the failure is logged. If the RTS is driven by a MAP request, a failure message is displayed at the MAP.



12.16.11 Node Log/Alarm Inform

This flow shows the activity when an SN wants to generate a log or an alarm at the switch.



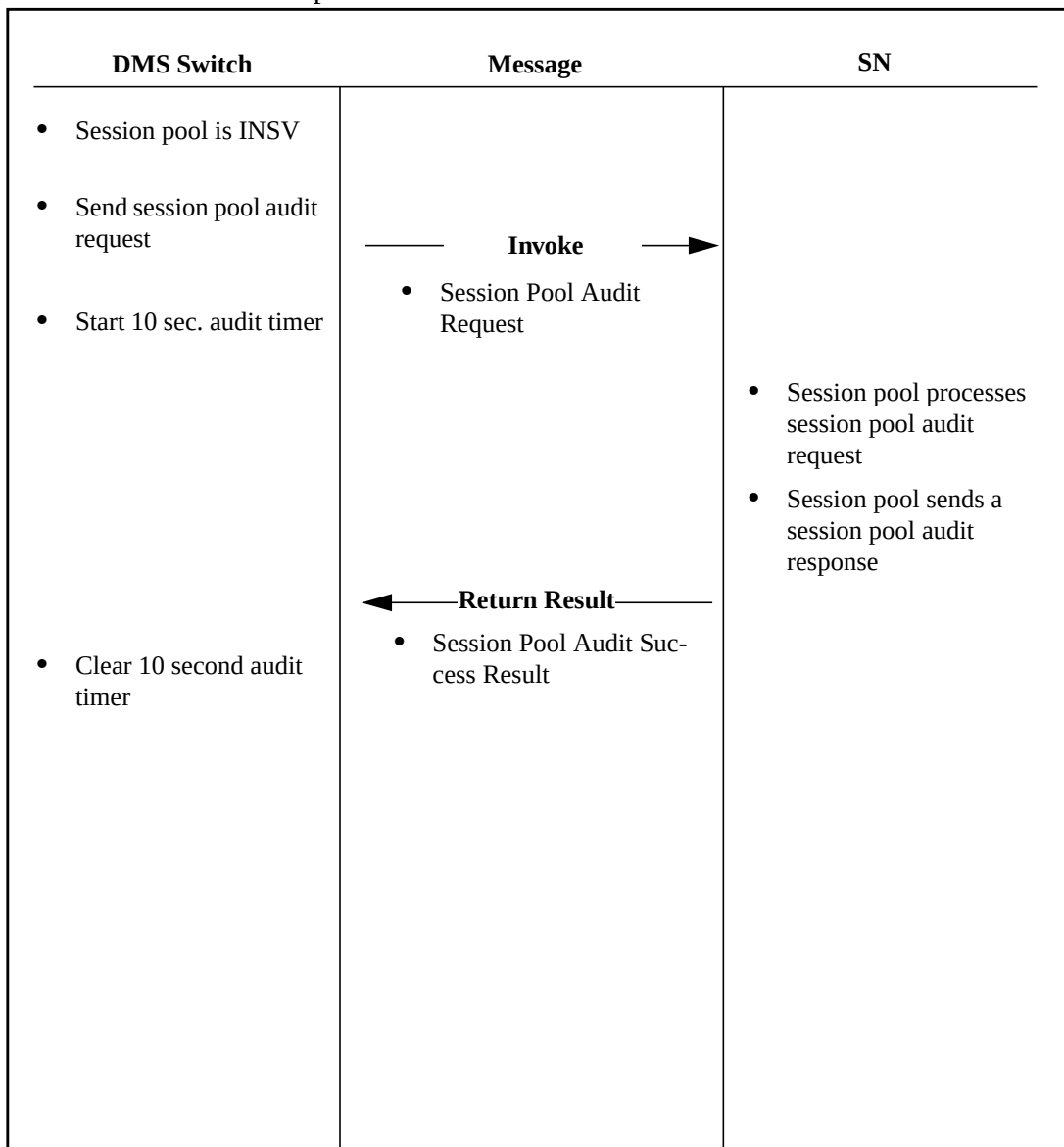
12.17 Non-Call Session Pool Based Message Flows

12.17.1 Session Pool Audit Request (Success)

The Maintenance Switch only sends this message to a session pool if:

- no call processing messages have been received from that session pool within a period defined by AUDFREQ in table OANODINV or,
- no audit messages have been sent to that session pool for 5 times the time period defined by AUDFREQ in table OANODINV.

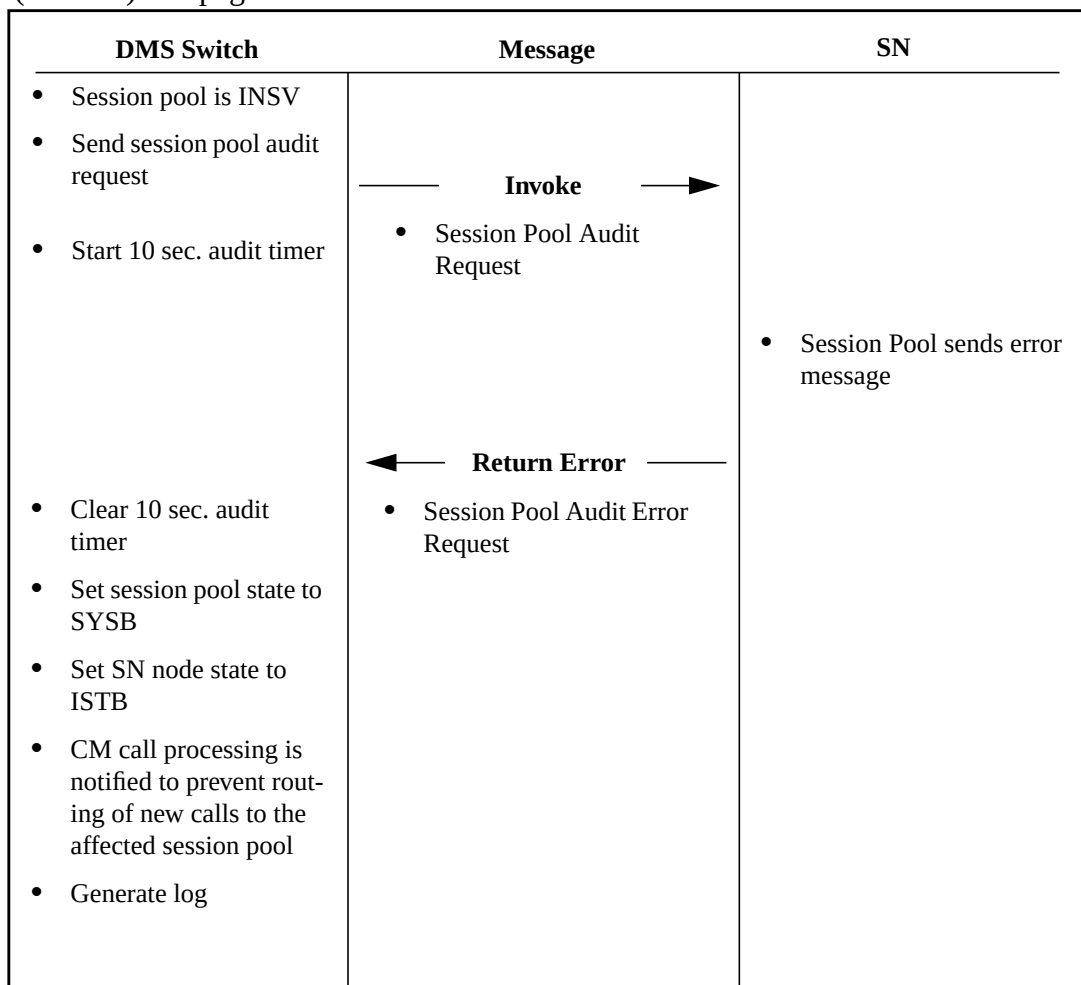
The audit is not performed if a message has been received from the service node across the subject session pool or if the session pool is not in an INSV state. This flow illustrates a successful session pool audit of an INSV session pool.



12.17.2 Session Pool Audit Request - (Failure)

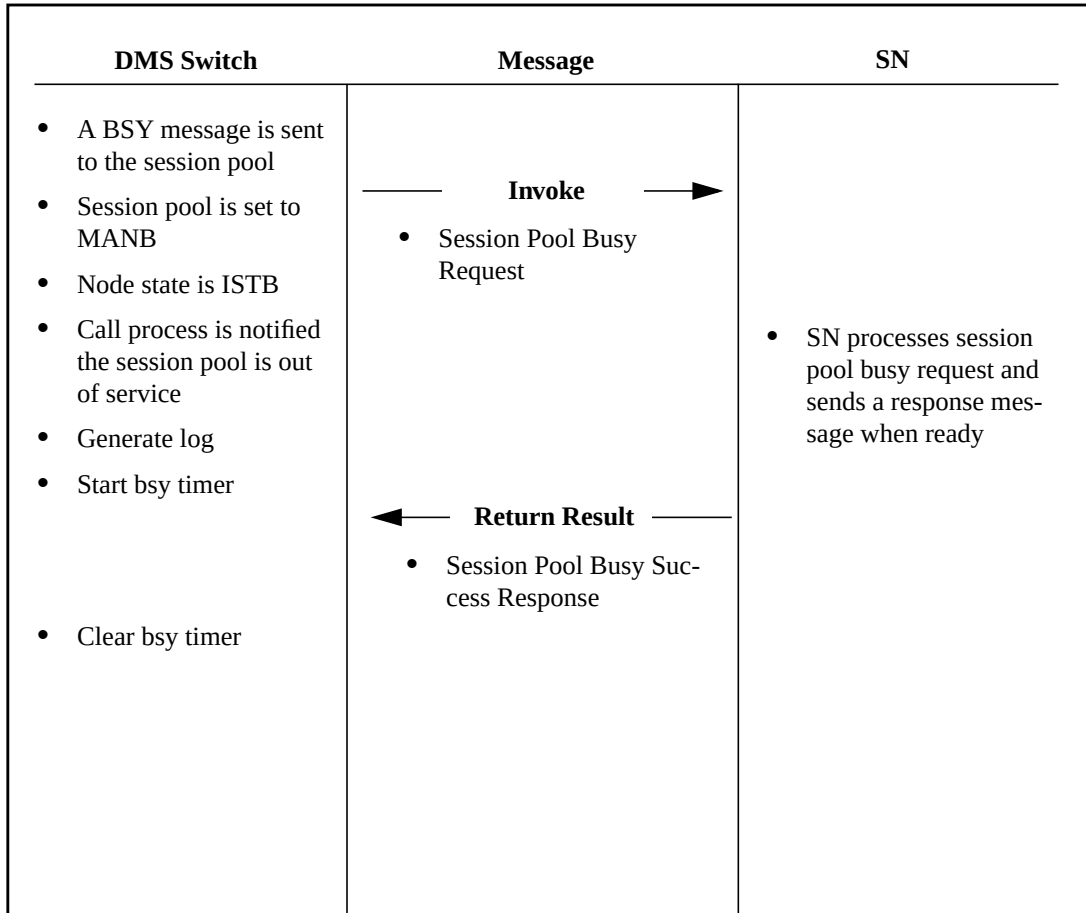
In this flow, the session pool fails to respond to the audit request. The session pool is placed into a SYSB state and calls with sessions to that session pool are taken down. The SN state is set to ISTB. New calls will not be routed to the out of service session pool.

The switch does not audit out of service session pools but, instead, the switch will periodically attempt to RTS the SYSB session pool. See “Session Pool Return To Service Request - Automated (Success)” on page 873.



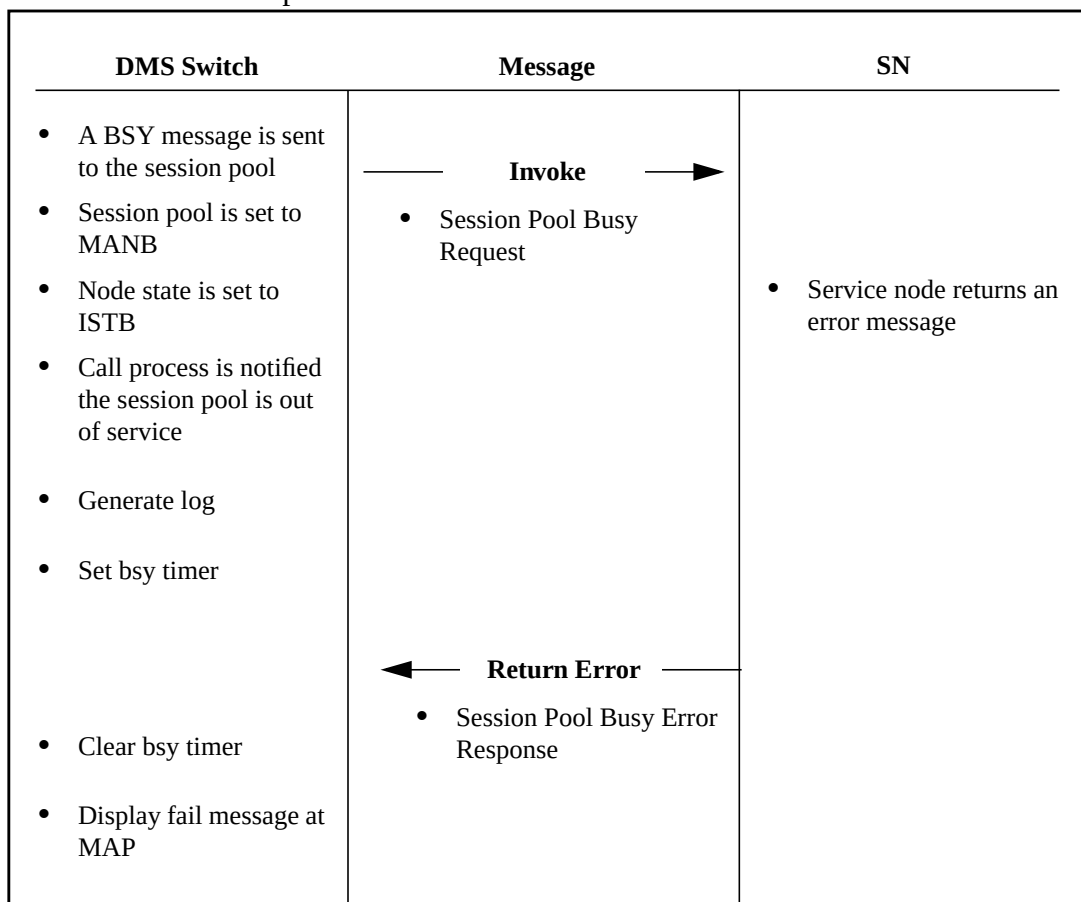
12.17.3 Session Pool Busy Request by Switch - (Success)

This flow shows the activity involved in busying out a session pool at the switch. When this occurs, the DMS Switch takes down all calls that have sessions with the target session pool. New calls are prevented from being routed to the target session pool. The session pool state is set to MANB and the node state of the SN that supports the session pool is set to ISTB.



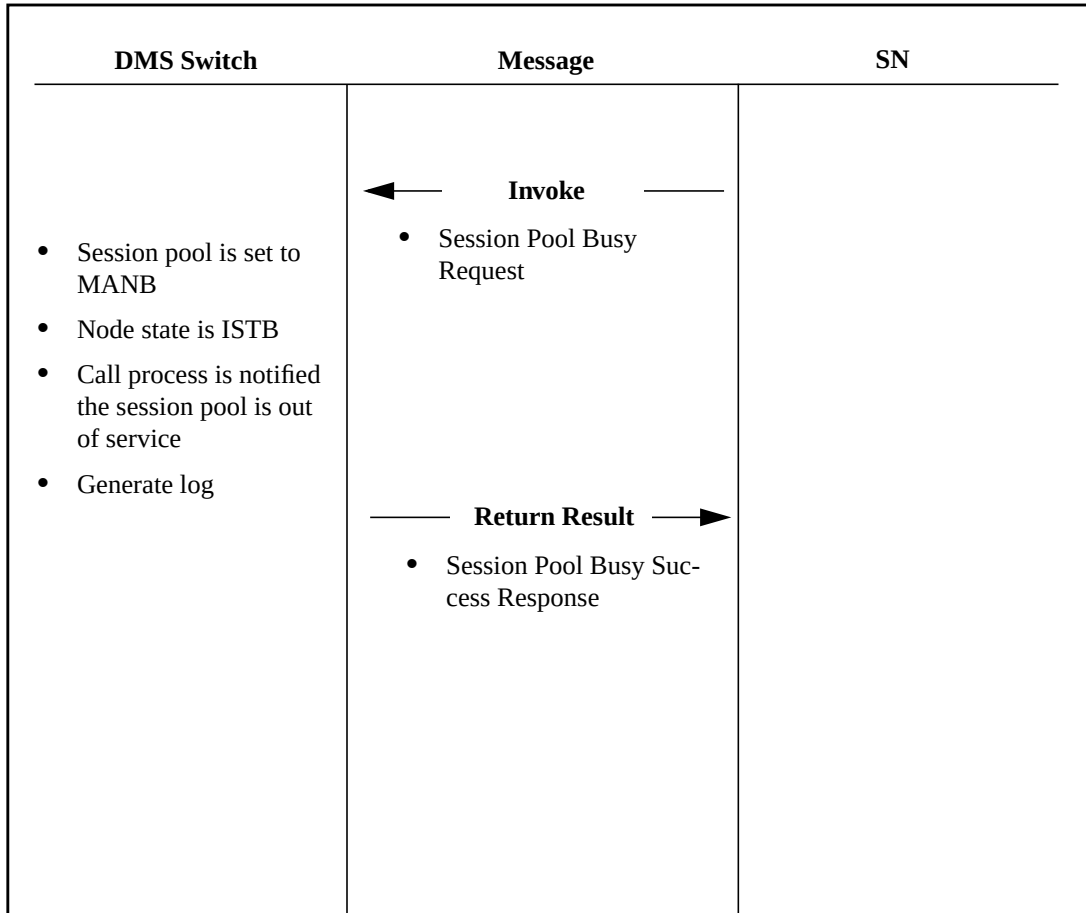
12.17.4 Session Pool Busy Request - (Failure)

In this flow, the session pool fails to respond to the session pool BSY request. The session pool is placed in MANB state and calls with sessions to that session pool are taken down. The SN state is set to ISTB. New calls will not be routed to the out of service session pool. The switch does not audit out of service session pools.



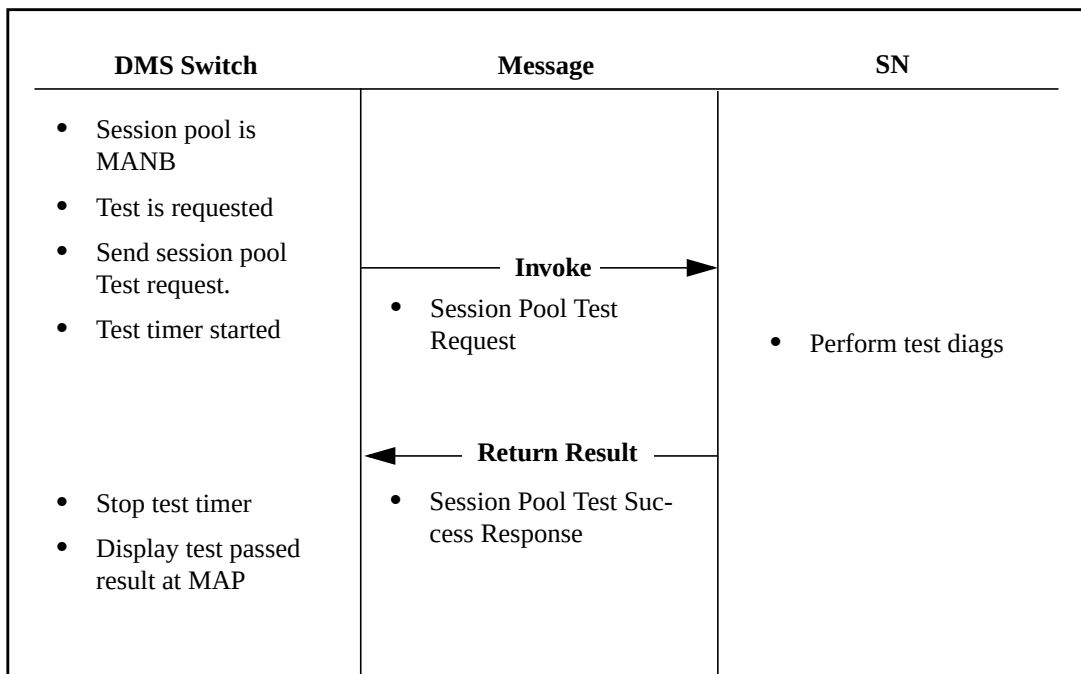
12.17.5 Session Pool Busy Request by SN - (Success)

This flow shows the activity involved in busying out a session pool by the SN. When this occurs, the DMS Switch takes down all calls that have sessions with the target session pool. New calls are prevented from being routed to the target session pool. The session pool state is set to MANB and the node state of the SN that supports the session pool is set to ISTB.



12.17.6 Session Pool Test Request - (Success)

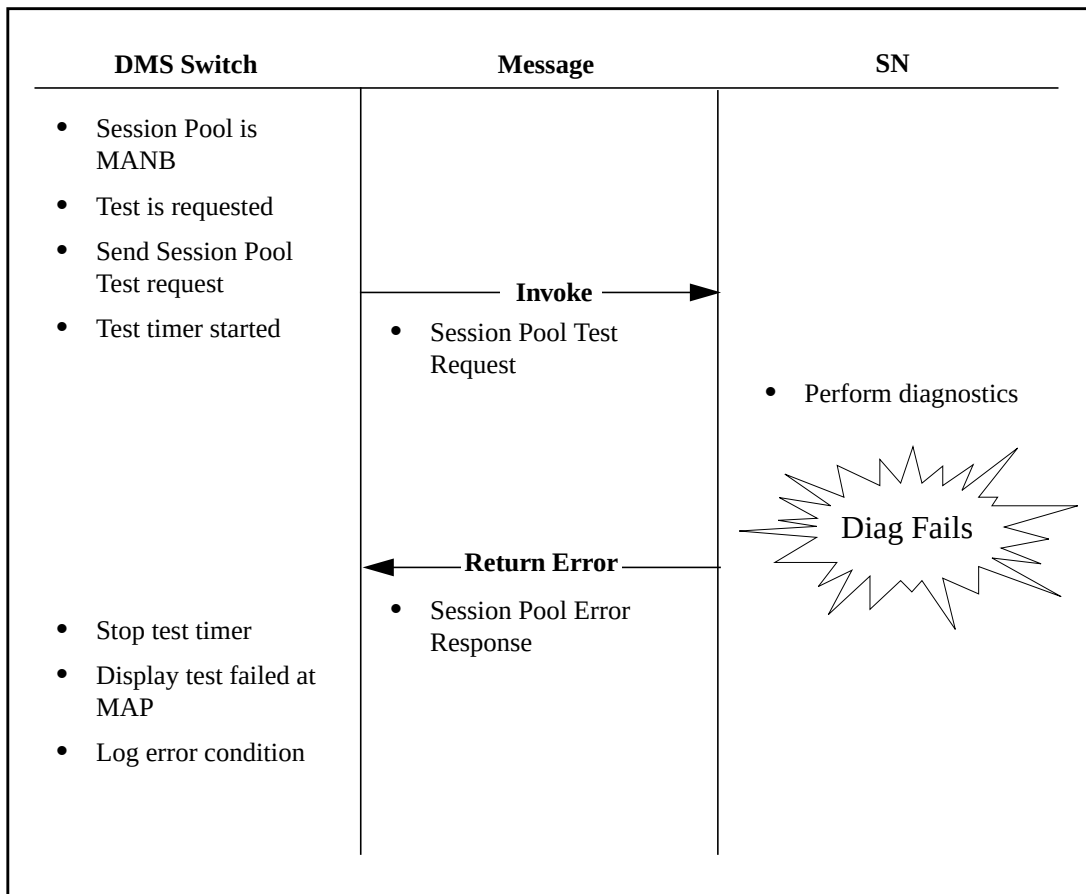
The DMS switch sends a session pool test message to a specific session pool whenever a manual session pool test (TST) is requested. The session pool must be in the MANB state prior to requesting the test.



12.17.7 Session Pool Test Request - (Failure)

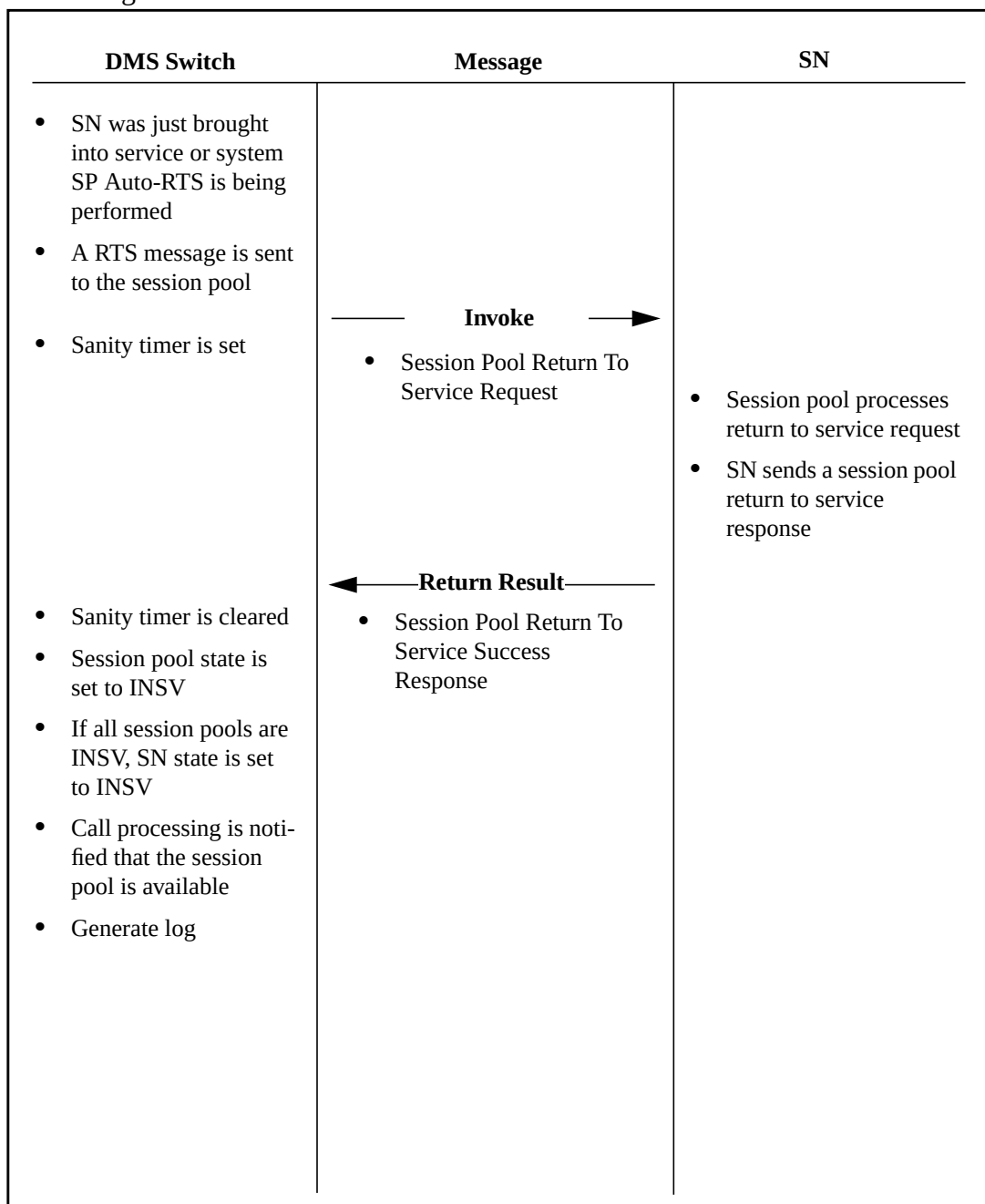
This example shows a DMS Switch originated session pool TST that fails. The session pool stays MANB. Two error scenarios are supported:

- Response with test failed. The test failed message will include a Maintenance Text data block to identify the diagnostic procedure that failed. A test failed notification is displayed at the MAP along with the Maintenance Text message from the session pool. A log will be generated which will also include the Maintenance Text.
- No response. The DMS switch will time out and indicate that no reply was received at the MAP.



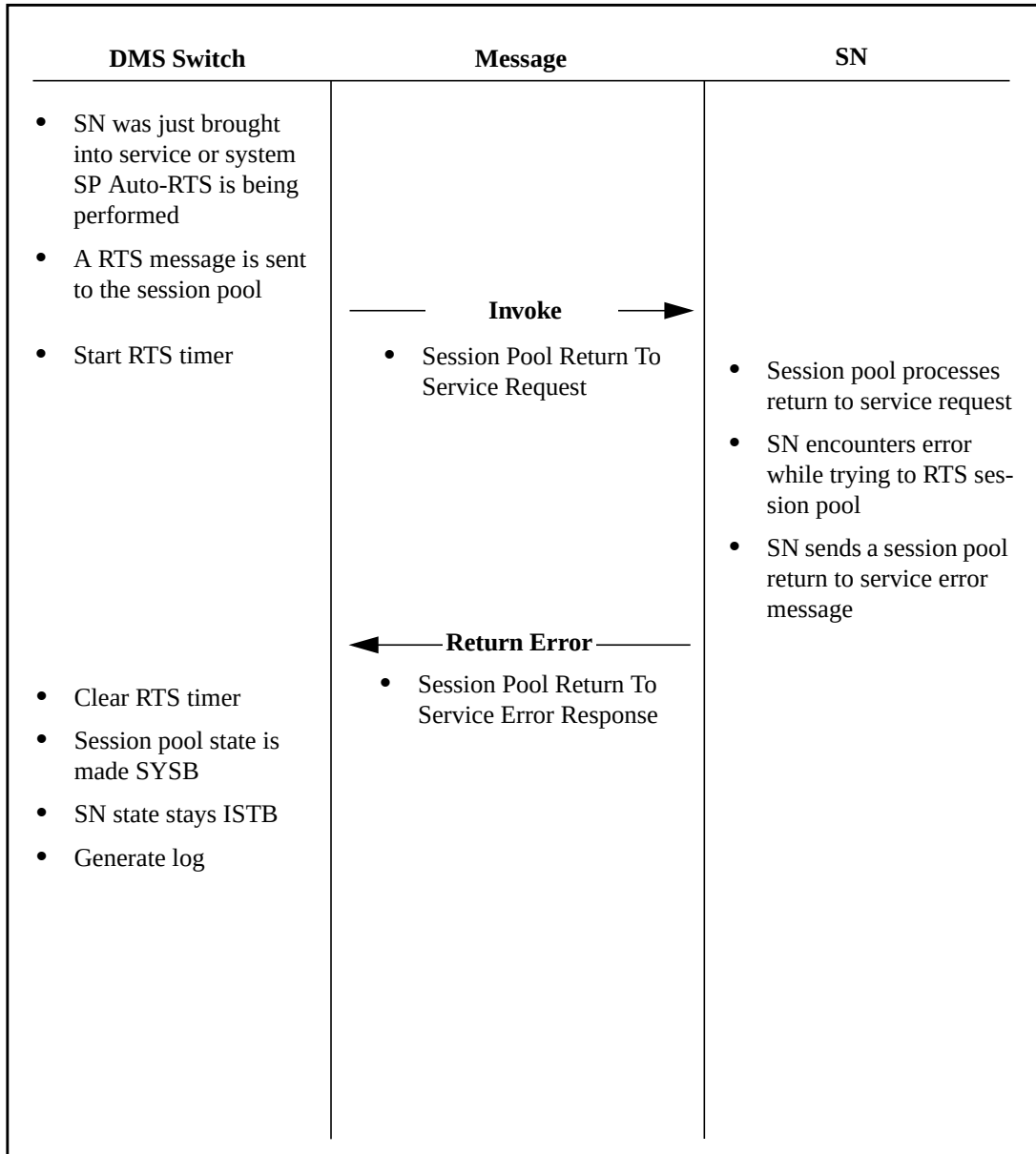
12.17.8 Session Pool Return To Service Request - Automated (Success)

This flow illustrates a DMS Switch originated RTS of a session pool. The RTS is initiated by either a SN being brought into service or by the start of a session pool audit cycle as defined by AUDFREQ in table OANODINV. Once the session pool responds to the RTS request the DMS Switch sets the session pool state to INSV. If all session pools supported by the SN are INSV, the SN state is changed from ISTB to INSV.



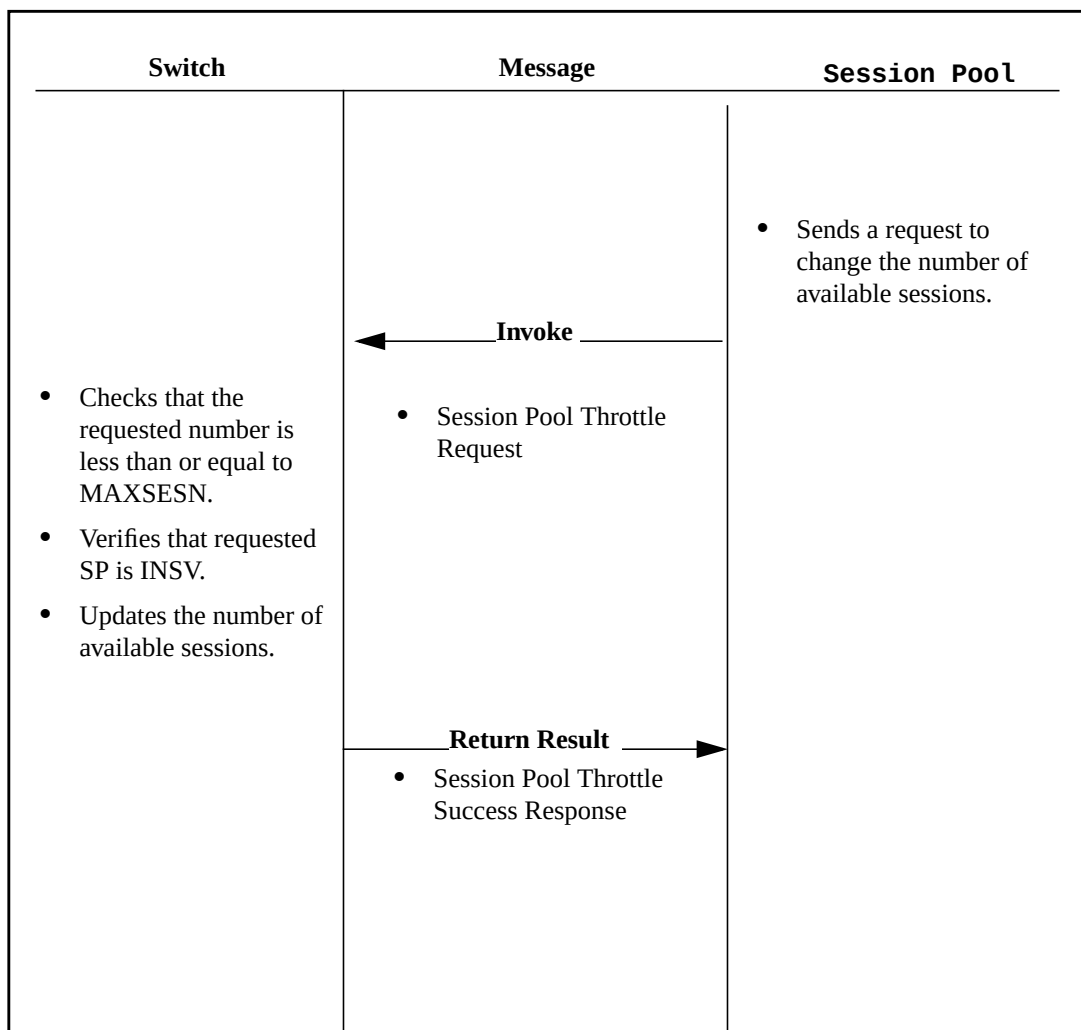
12.17.9 Session Pool Return To Service Request - Automated (Failure)

This example shows a DMS Switch originated session pool RTS that fails. The session pool state to either go SYSB and the SN state to stay ISTB.



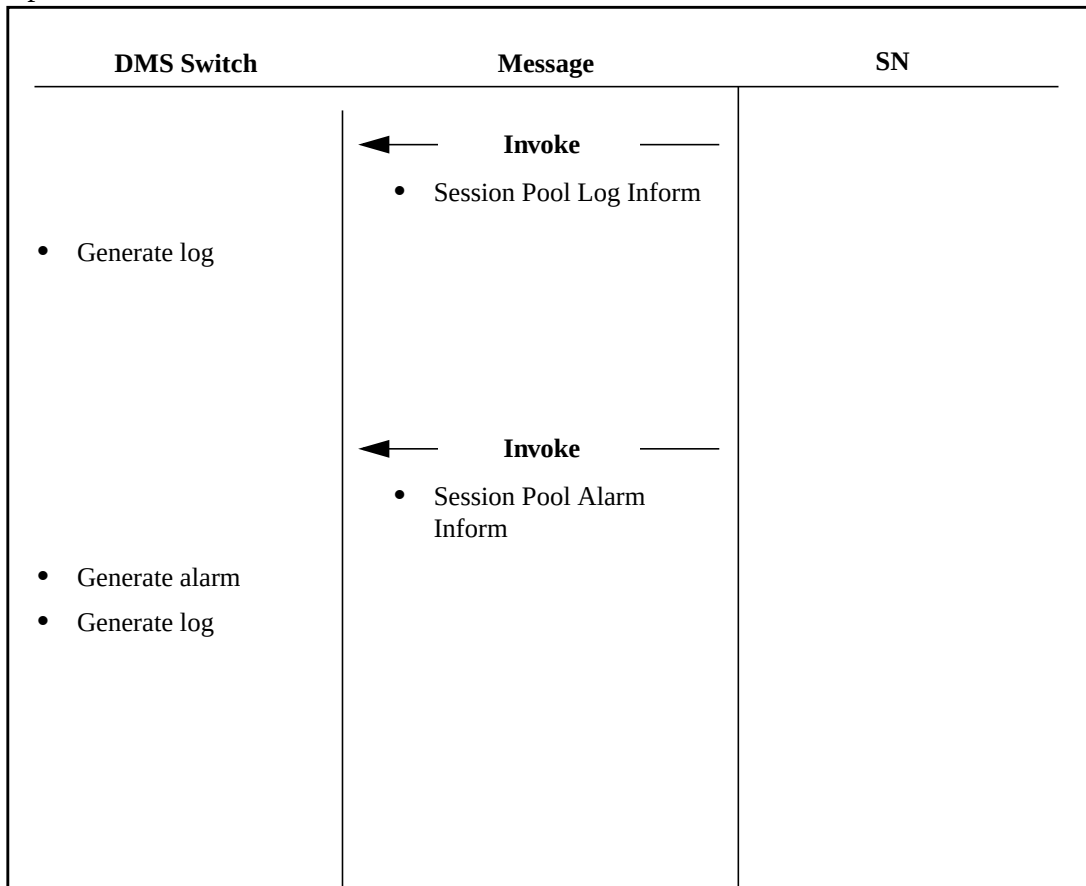
12.17.10 Session Pool Throttle Request

This flow shows the activity when a SP wants to change the number of sessions available for call processing. It sends a request to the DMS to change the number of available sessions. If the request is valid, the DMS changes the number of available sessions in the SP. After updating the number of available sessions for the SP, the DMS sends a success reply back to the SP.



12.17.11 Session Pool Log/Alarm Inform

This flow shows the activity when an SN wants to generate a log or an alarm at the switch for a session pool.



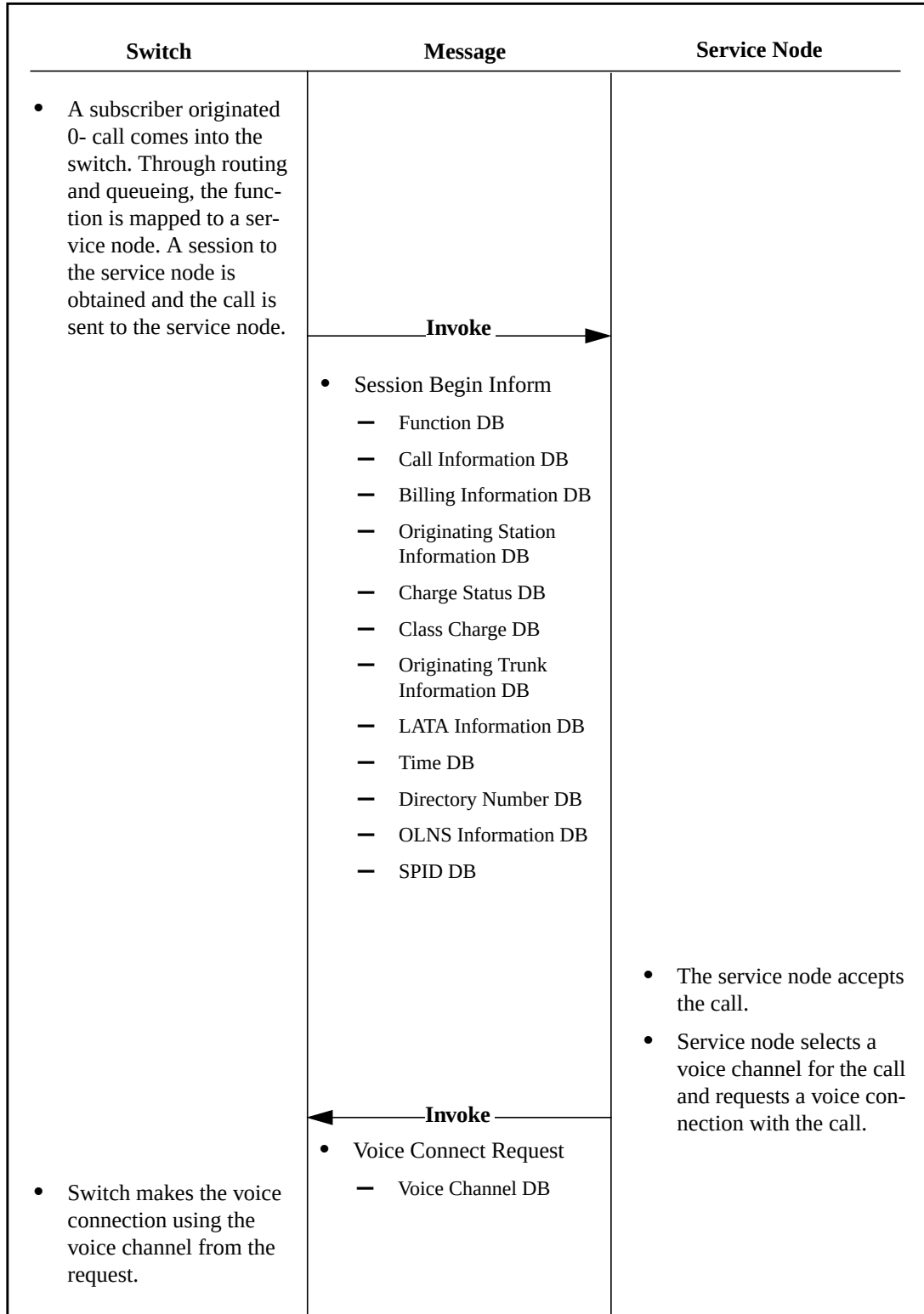
12.18 Example Call Flows

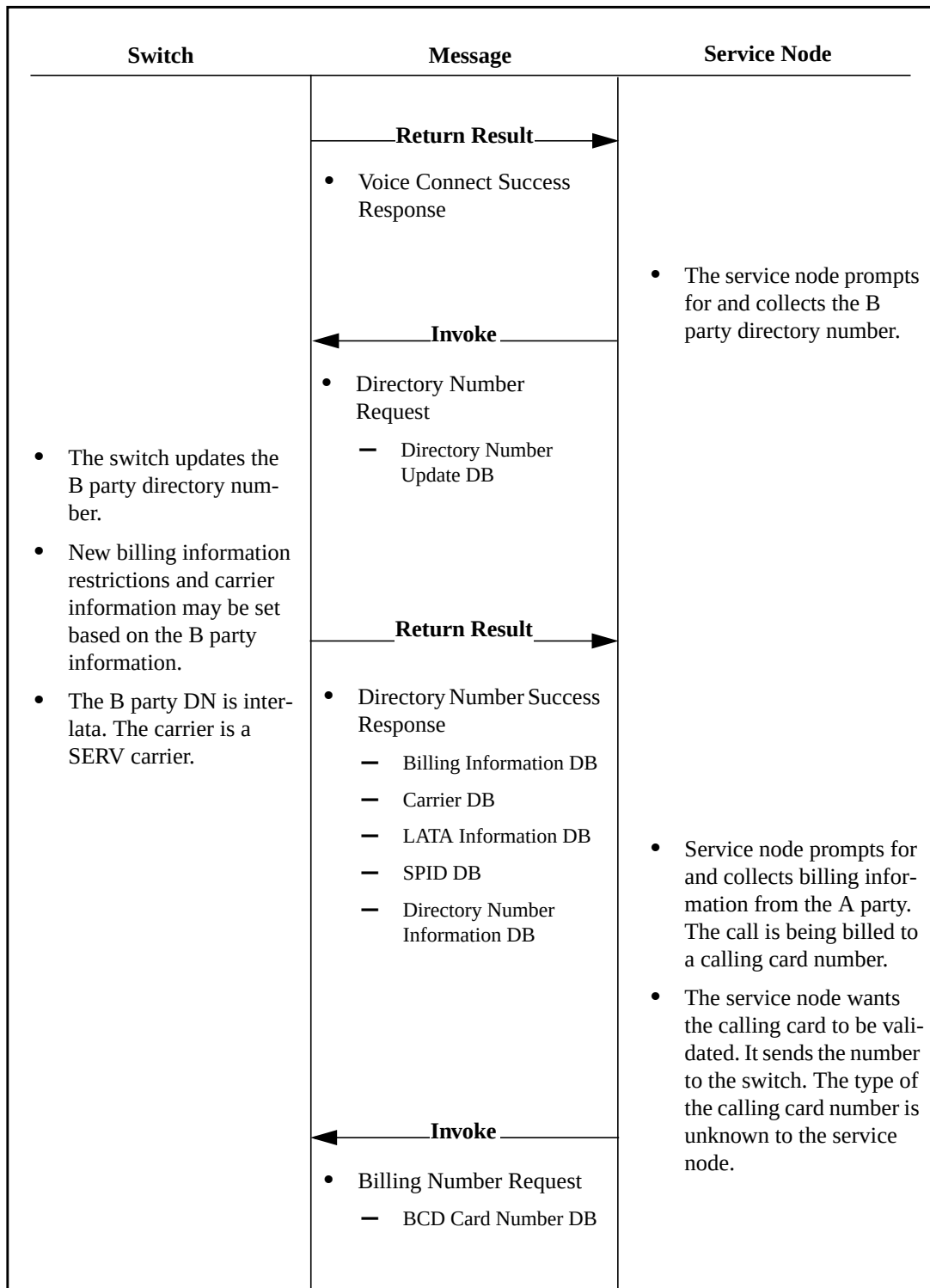
The following sections describe example call scenarios. For each call scenario, the following information is provided:

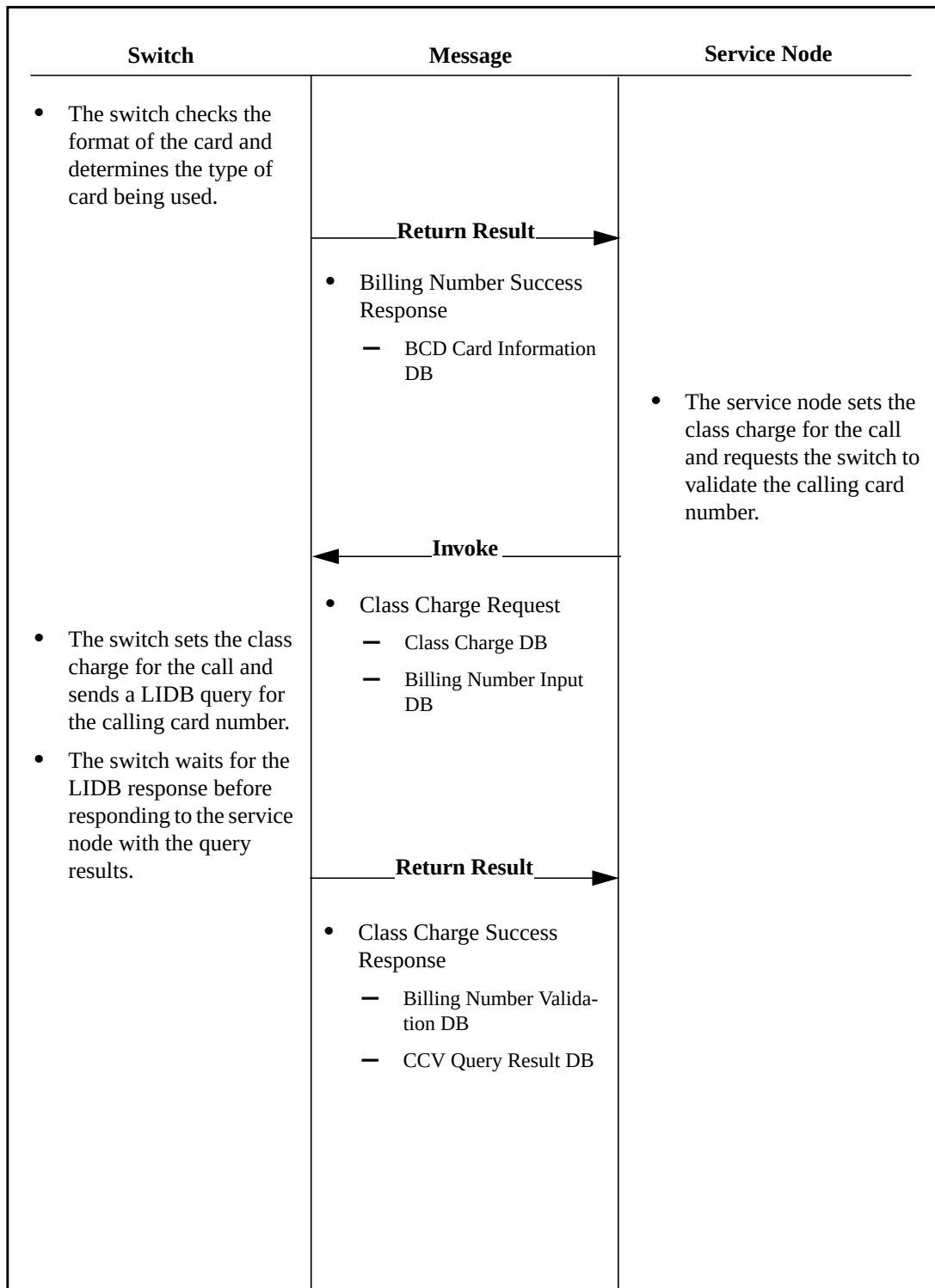
- Message Flow
- Raw hex message for each OAP message
- Message layout for each OAP message

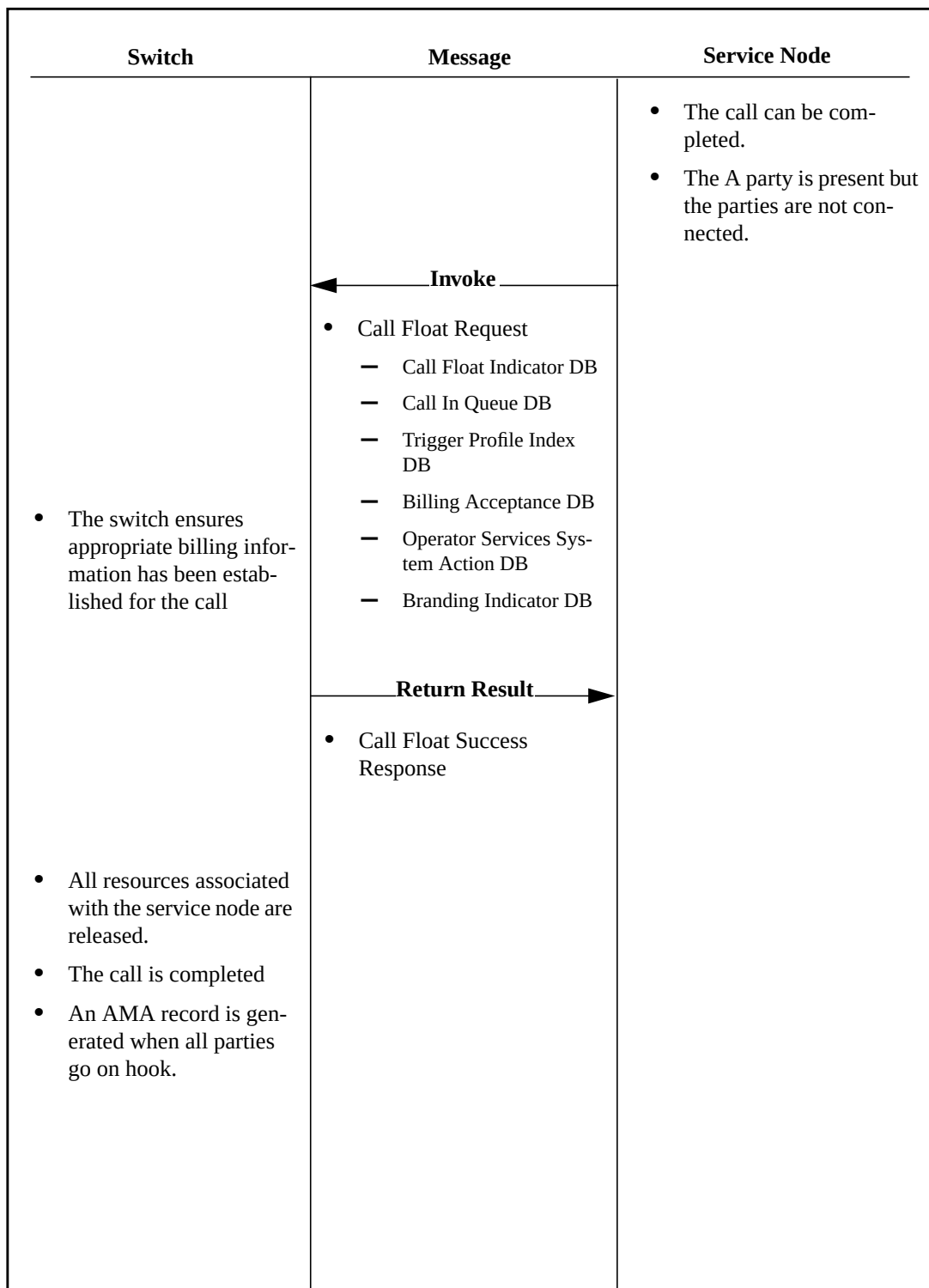
12.18.1 Call Billed to a Calling Card

The following call scenario is a basic call completion call billed to a calling card number.









Session Begin for Call (Hex Dump):

```
0005000000EE001A04000400005A0001000100060001000000BBA18200E2020201000202
060D048200CA011400020001010700100001000100000010100000000000101FE010400060
0150FFFF64070119001201000000FFFFFFFFFFFFFFFF00FFFFFFFFFFFFF010A000201FF01
0B0006FEFE000000FF011A000400E2000401480004FF0423450304011101010008433131
3120202020013600020000011E000661040E0A28050111001E0101FE000101FF0A619322
6565FFFFFFFFF0000FFFFFFFFFFFFFFFFFFFFFFFFF0134002E0101010101010101010102040
1110100FFFF0100FFFF0200FFFFFFFFFFFFFFFFFFFF0107FF0441424344FF00FFFFFFFFF014
500080101FF0441424344
```

- Call Processing Class Header
- Invoke ROSE Header
- Function DB
- Call Information DB
- Billing Information DB
- Originating Station Information DB
- Charge Status DB
- Class Charge DB
- Originating Trunk Information DB
- NBEC DB
- LATA Information DB
- Time DB
- Directory Number DB
- OLNS Information DB
- SPID DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#00EE	238 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0001	1
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#00E2	226 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, Length = 2
33	Invoke ID Value	#0100	#0100
34			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, Length = 2
37	Operation ID Value	#060D	Session Begin Inform
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#00CA	202 bytes
42			
43	Data Block ID	#0114	Function DB
44			
45	DB Byte Count	#0002	2 bytes
46			
47	Function ID	#0001	1 (Table OAFUNDEF)
48			
49	Data Block ID	#0107	Call Information DB
50			
51	DB Byte Count	#0010	16 bytes
52			
53	Session Origination Type	#00	Initial Call Setup
54	Service Node Mode	#01	Active
55	Control List ID	0001	1 (Table OACTLDEF)
56			
57	Trigger Profile Index	0000	0 (Table OATPRFIX)
58			
59	Call Origination Type	01	Zero Minus
60	Call In Queue Take Down	01	Take Call Down
61	DTMF Receiver	00	Not Connected
62	Service Type	00	TA
63	Session Type	00	Standard Session
64	Conversation Timing Status	00	Stop Call Timing
65	Branding Status	00	No Branding
66	Notification Timer Status	#01	Not Running
67	SPID Processing	#01	A party enabled
68	Number of Operator to SN Handoffs	#FE	Nil

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
69	Data Block ID	#0104	Billing Information DB
70			
71	DB Byte Count	#0006	6 bytes
72			
73	Network Service ID	#0015	21 (Table OACTLDEF)
74			
75	Allowed Billing Set	#0FFF	All billing is allowed
76			
77	Restricted Billing Index	#64	100 (nil value, unrestricted)
78	Terminating Code Allowed Set	#07	Calling Card, Third number, and Collect billing is allowed
79	Data Block ID	#0119	Originating Station Information DB
80			
81	DB Byte Count	#0012	18 bytes
82			
83	Phone Class Type	#01	Station
84	Coin Overtime Type	#00	Not Applicable
85	Coin Key Pad State	#00	Undetermined
86	Hotel Room Number Length	#00	0
87	Hotel Room Number	#FF #FF #FF #FF #FF #FF	Nil
88			
89			
90			
91			
92			
93	Padding Field	#FF	Nil
94	Authorization Number Length	#00	0
95	Authorization Number	#FF #FF #FF #FF #FF #FF	Nil
96			
97			
98			
99			
100			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
101	Data Block ID	#010A	Charge Status DB
102			
103	DB Byte Count	#0002	2 bytes
104			
105	Billable Status	#01	Billable
106	Padding Field	#FF	Nil
107	Data Block ID	#010B	Class Charge DB
108			
109	DB Byte Count	#0006	6 bytes
110			
111	Handling Data	#FE	Unspecified
112	Billing Data	#FE	Unspecified
113	Validation Request	#00	Not Applicable
114	Validation Sequence Number	#00	0
115	Class Charge Selector	#00	Current
116	Padding Field	#FF	Nil
117	Data Block ID	#011A	Originating Trunk Information DB
118			
119	DB Byte Count	#0004	4 bytes
120			
121	Trunk Group ID	#00E2	226 (Table CLLI, field ADNUM)
122			
123	Trunk Member ID	#0004	4 (Table TRKMEM)
124			
125	Data Block ID	#0148	NBEC DB
126			
127	DB Byte Count	#0004	4 bytes
128			
129	Padding field	#FF	Nil
130	NBEC Code Digit Count	#04	4 digits
131	NBEC Code	#23	2345
132		#45	
133	Data Block ID	#0136	LATA Information DB
134			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
135	DB Byte Count	#0002	2 Bytes
136			
137	LATA Status	#00	IntraLATA
138	LATA Restrictions	#00	No Restrictions
139	Data Block ID	#011E	Time DB
140			
141	DB Byte Count	#0006	6 bytes
142			
143	Year	#61	97
144	Month	#04	04
145	Day	#0E	14
146	Hour	#0A	10
147	Minutes	#28	40
148	Seconds	#05	05
149	Data Block ID	#0111	Directory Number DB
150			
151	DB Byte Count	#001E	30 bytes
152			
153	Party	#01	A Party
154	Directory Number Type	#01	Domestic
155	Language	#FE	Unknown
156	ANI Status	#00	ANI Success
157	Connection Status	#01	Connected, Off Hook
158	Speech Path Status	#01	Two-Way
159	Presentation Indicator	#FF	Presentation Status Unknown
160	Directory Number Digit Count	#0A	10 digits

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
161	Directory Number	#61 #93 #22 #65 #65 #FF #FF #FF #FF #FF	619-322-6565
162			
163			
164			
165			
166			
167			
168			
169			
170			
171	Unenforced Restrictions	#00	No DN unenforced restrictions
172	Calling Line Identifier Digit Count	#00	0 digits
173	Calling Line Identifier Number	#FF #FF #FF #FF #FF #FF #FF #FF #FF #FF	Nil
174			
175			
176			
177			
178			
179			
180			
181			
182			
183	Data Block ID	#0134	OLNS Information DB
184			
185	DB Byte Count	#002E	46 bytes
186			
187	OLNS TA Sent Paid Indicator	#01	Allowed
188	OLNS TA Collect Indicator	#01	Allowed
189	OLNS TA Third Number Indicator	#01	Allowed
190	OLNS TA Credit Card Indicator	#01	Allowed
191	OLNS Special BNS Indicator	#01	Allowed
192	OLNS DACC Indicator	#01	Allowed
193	OLNS DA Sent Paid Indicator	#01	Allowed
194	OLNS DA Third Number Indicator	#01	Allowed
195	OLNS DA Credit Card Indicator	#01	Allowed

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
196	OLNS DA Special BNS Indicator	#01	Allowed
197	OLNS InterLATA Carrier Indicator	#02	IC Indicated
198	OLNS InterLATA Carrier ID Digit Count	#04	4 digits
199	OLNS InterLATA Carrier ID Code	#01	0111
200		#11	
201	OLNS IntraLATA Toll Carrier Indicator	#01	Not Indicated
202	OLNS IntraLATA Toll Carrier ID Digit Count	#00	0 digits
203	OLNS IntraLATA Toll Carrier ID Code	FFFF	Nil
204			
205	OLNS International Carrier Indicator	#01	Not Indicated
206	OLNS International Carrier ID Digit Count	#00	0 digits
207	OLNS International Carrier ID Code	FFFF	Nil
208			
209	OLNS Foreign Language	#02	2, French
210	OLNS Alphanumeric String Length	#00	0 characters
211	OLNS Alphanumeric String	#FF #FF #FF #FF #FF #FF #FF #FF	Nil
212			
213			
214			
215			
216			
217			
218			
219	OLNS Service/Equipment Indicator	#01	POTS
220	OLNS Treatment Indicator	#07	Automated Treatment
221	Padding Field	#FF	Nil
222	OLNS Account Owner Length	#04	4 characters
223	OLNS Account Owner	#41	ABCD
224		#42	
225		#43	
226		#44	
227	Padding Field	#FF	Nil
228	OLNS Billing Service Provider Length	#00	0 characters

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
229	OLNS Billing Service Provider	#FF	FFFF
230		#FF	
231		#FF	
232		#FF	
233	Data Block ID	#0145	SPID DB
234			
235	DB Byte Count	#0008	8 bytes
236			
237	Service Provider Party Identifier	#01	Originating Party
238	Type of Service Provider	#01	Account Owner
239	Padding Field	#FF	Nil
240	Service Provider ID Length	#04	4 characters
241	Service Provider ID	#41	ABCD
242		#42	
243		#43	
244		#44	

Voice Connect Request (Hex Dump):

00050000002A001A0400040000060001000100060001000000BBA18200120202010002020
615048200060127000203FD

- Call Processing Class Header
- Invoke ROSE Header
- Voice Channel DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#002A	42 bytes
6			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0001	1
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0012	18 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, Length = 2
33	Invoke ID Value	#0100	#0100
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, Length = 2
37	Operation ID Value	#0615	Voice Connect Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0006	6 bytes
42			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
43	Data Block ID	#0127	Voice Channel DB
44			
45	DB Byte Count	#0002	2 bytes
46			
47	Voice Channel ID	#03FD	1021 (Table OAVLMAP for each service node)
48			

Voice Connect Success Response (Hex Dump):

000500000020001A04000400005A0002000100060001000000BBA28200080202010004820000

- Call Processing Class Header
- Return Result ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0002	2
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0008	8 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0100	#0100 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

Directory Number Request (Hex Dump):

000500000042001A0400040000060002000100060001000000BBA182002A0202010102020
6070482001E0112001A0211010A2012201236FFFFFFFFFFFFFFFF00FFFFFFFFFFFFFFFFFFFF
F

- Call Processing Class Header
- Invoke ROSE Header
- Directory Number Update DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0042	66 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0002	2
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#002A	42 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0101	#0101
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0607	Directory Number Request
38			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#001E	30 bytes
42			
43	Data Block ID	#0112	Directory Number Update DB
44			
45	DB Byte Count	#001A	26 bytes
46			
47	Party	#02	B Party
48	Language	#11	17 (Table TOPSLANG)
49	Directory Number Type	#01	Domestic
50	Directory Number Digit Count	#0A	10 digits
51	Directory Number	#20 #12 #20 #12 #36 #FF #FF #FF #FF #FF	201-220-1236
52			
53			
54			
55			
56			
57			
58			
59			
60			
61	Presentation Indicator	#FF	Presentation Status Unknown
62	Calling Line Identifier Digit Count	#00	0 digits
63	Calling Line Identifier Number	#FF #FF #FF #FF #FF #FF #FF #FF #FF #FF	Nil
64			
65			
66			
67			
68			
69			
70			
71			
72			

Directory Number Success Response (Hex Dump):

000500000058001A04000400005A0003000100060001000000BBA28200540202010104820
0360104000600150FFF6407010900100304011101010008433131312020202001360002010
014500080201FF044142434401560004020100FF

- Call Processing Class Header
- Return Result ROSE Header
- Billing Information DB
- Carrier DB
- LATA Information DB
- SPID DB
- Directory Number Information DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0058	88 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0003	3
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#54	84 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0101	#0101 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0036	54 bytes
38			
39	Data Block ID	#0104	Billing Information DB
40			
41	DB Byte Count	#0006	6 bytes
42			
43	Network Service ID	#0015	21 (Table OACTLDEF)
44			
45	Allowed Billing Set	#0FFF	All billing is allowed
46			
47	Restricted Billing Index	#64	100 (nil value, unrestricted)
48	Terminating Code Allowed Set	#07	Calling Card, Third number, and Collect billing is allowed
49	Data Block ID	#0109	Carrier DB
50			
51	DB Byte Count	#0010	16 bytes
52			
53	Carrier Assignment Source Indicator	#03	OLNS
54	Carrier Identification Digit Count	#04	4 digits
55	Carrier Identification Code	#01	0111
56		#11	

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
57	Service Type Provided by Switch	#01	Complete Operator Svcs
58	Carrier Number Status	#01	Valid
59	Carrier Service Status	#00	Valid
60	Carrie Name Length	#08	8 characters
61	Carrier Name	#43 #31 #31 #31 #20 #20 #20 #20	C111 space space space space
62			
63			
64			
65			
66			
67			
68			
69	Data Block ID	#0136	LATA Information DB
70			
71	DB Byte Count	#0002	2 bytes
72			
73	LATA Status	#01	InterLATA
74	LATA Restrictions	#00	No Restrictions
75	Data Block ID	#0145	SPID DB
76			
77	DB Byte Count	#0008	8 bytes
78			
79	Service Provider Party Identifier	#02	Terminating Party
80	Type of Service Provider	#01	Account Owner
81	Padding Field	#FF	Nil
82	Service Provider ID Length	#04	4 characters
83	Service Provider ID	#41 #42 #43 #44	ABCD
84			
85			
86			
87	Data Block ID	#0156	Directory Number Information DB
88			
89	DB Byte Count	#0002	4 bytes
90			
91	Party	#02	B Party
92	Directory Number Tytpe	#01	Domestic Non-Local

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
93	Unenforced Restrictions	#00	No DN unenforced restrictions
94	Padding Field	#FF	Nil

Billing Number Request (Hex Dump):

00050000003C001A0400040000060003000100060001000000BBA182002402020102020206020482001801080014FEFF000E082B081B1B080AF5F5F5F5F5F5F5F5F5F5

- Call Processing Class Header
- Invoke ROSE Header
- BCD Card Number DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#003C	60 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0003	3
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0024	36 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0102	#0102
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0602	Billing Number Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0018	24 bytes
42			
43	Data Block ID	#0108	BCD Card Number DB
44			
45	DB Byte Count	#0014	20 bytes
46			
47	BCD Card Number Type	#FE	Unknown Card Type
48	Padding Field	#FF	Nil
49	BCD PIN Digit Count	#00	0 digits
50	BCD Card Number Digit Count	#0E	14 digits

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
51	BCD Card Number	#08 #2B #08 #1B #1B #08 #0A #F5 #F5 #F5 #F5 #F5 #F5 #F5 #F5	Encrypted BCD Number
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			

Billing Number Success Response (Hex Dump):

000500000026001A04000400005A0004000100060001000000BBA282000E020201020482
00060120000201FF

- Call Processing Class Header
- Return Result ROSE Header
- BCD Card Information DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0026	38 bytes
6			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0004	4
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#000E	14 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, short, length = 2
33	Invoke ID Value	#0102	#0102 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0006	6 bytes
38			
39	Data Block ID	#0120	BCD Card Information DB
40			
41	DB Byte Count	#0002	2 bytes
42			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
43	BCD Card Number Type	#01	Directory Number Card
44	Padding Field	#FF	Nil

Class Charge Request (Hex Dump):

000500000034001A0400040000060004000100060001000000BBA182001C0202010302020
60304820010010B00060002020000FF0131000200FF

- Call Processing Class Header
- Invoke ROSE Header
- Class Charge DB
- Billing Number Input DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0034	52 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0004	4
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#001C	28 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0103	#0103
34			
35	Operation ID Type	#02	#02, Fixed
36	Operation ID Length Format	#02	#02, Fixed
37	Operation ID Value	#0603	Class Charge Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0010	16 bytes
42			
43	Data Block ID	#010B	Class Charge DB
44			
45	DB Byte Count	#0006	6 bytes
46			
47	Handling Data	#00	Station
48	Billing Data	#02	Special Calling
49	Validation Request	#02	Validate
50	Validation Sequence Number	#00	0
51	Class Charge Selector	#00	Current
52	Padding Field	#FF	Nil
53	Data Block ID	#0131	Billing Number Input DB
54			
55	DB Byte Count	#0002	2 bytes
56			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
57	Billing Number Input Method	#00	DTMF
58	Padding Field	#FF	Nil

Class Charge Success Response (Hex Dump):

000500000030001A04000400005A0005000100060001000000BBA28200180202010304820
010010500060004000302F1012D000202FF

- Call Processing Class Header
- Return Result ROSE Header
- Billing Number Validation DB
- CCV Query Result

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0030	48 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0005	5
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0018	24 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0103	#0103 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0010	16 bytes
38			
39	Data Block ID	#0105	Billing Number Validation DB
40			
41	DB Byte Count	#0006	6 bytes
42			
43	Validation Sequence Number	#00	0
44	Validation Result	#04	Billing Accepted
45	Validation Result Determined By DMS Datafill	#00	Validation Result Not Determined by DMS Datafill
46	RAO Digit Count	#03	3 digits
47	RAO Code	#02 #F1	201
48			
49	Data Block ID	#012D	CCV Query Result
50			
51	DB Byte Count	#0002	2 bytes
52			
53	CCV Query Response	#02	PIN Unrestricted
54	Padding Field	#FF	Nil

Call Float Request (Hex Dump):

000500000050001A0400040000060005000100060001000000BBA18200380202010402020
6060482002C0128000200FF0106000201FF0124000AFF00000200000000000000130000200
FF0133000201FF013E000201FF

- Call Processing Class Header
- Invoke ROSE Header
- Call Float Indicator DB
- Call In Queue DB
- Trigger Profile Index DB
- Operator Services System Action DB
- Branding Indicator DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0050	80 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0005	5
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0038	56 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0104	#0104
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0606	Call Float Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#2C	44 bytes
42			
43	Data Block ID	#0128	Call Float Indicator DB
44			
45	DB Byte Count	#0002	2 bytes
46			
47	Float Connection Indicator	#00	Unconnected or Connected
48	Padding Field	#FF	Nil
49	Data Block ID	#0106	Call In Queue
50			
51	DB Byte Count	#0002	2 bytes
52			
53	Call In Queue Take Down	#01	Take Call Down
54	Padding Field	#FF	Nil

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
55	Data Block ID	#0124	Trigger Profile Index DB
56			
57	DB Byte Count	#000A	10 bytes
58			
59	Padding Field	#FF	Nil
60	No Answer Time	#00	0 seconds
61	Trigger Profile Index	#0002	2 (table OATPRFIX)
62			
63	Notification Time - Elapsed Minutes	#00	0 minutes
64		#00	
65		#00	
66	Notification Time - Seconds	#00	0 seconds
67	Notification Time Type	#00	Relative Time
68	Notification Timer Mode	#00	Float Style
69	Data Block ID	#0130	Billing Acceptance DB
70			
71	DB Byte Count	#0002	2 bytes
72			
73	Billing Acceptance Response Method	#00	DTMF
74	Padding Field	#FF	Nil
75	Data Block ID	#0133	Operator Services System Action DB
76			
77	DB Byte Count	#0002	2 bytes
78			
79	Operator Services System Action	#01	Automatically Accept
80	Padding Field	#FF	Nil
81	Data Block ID	#013E	Branding Indicator DB
82			
83	DB Byte Count	#0002	2 bytes
84			
85	Branding Status	#01	Call has received front-end branding
86	Padding Field	#FF	Nil

Call Float Success Response (Hex Dump):

000500000020001A04000400005A0006000100060001000000BBA28200080202010404820000

- Call Processing Class Header
- Return Result ROSE Header

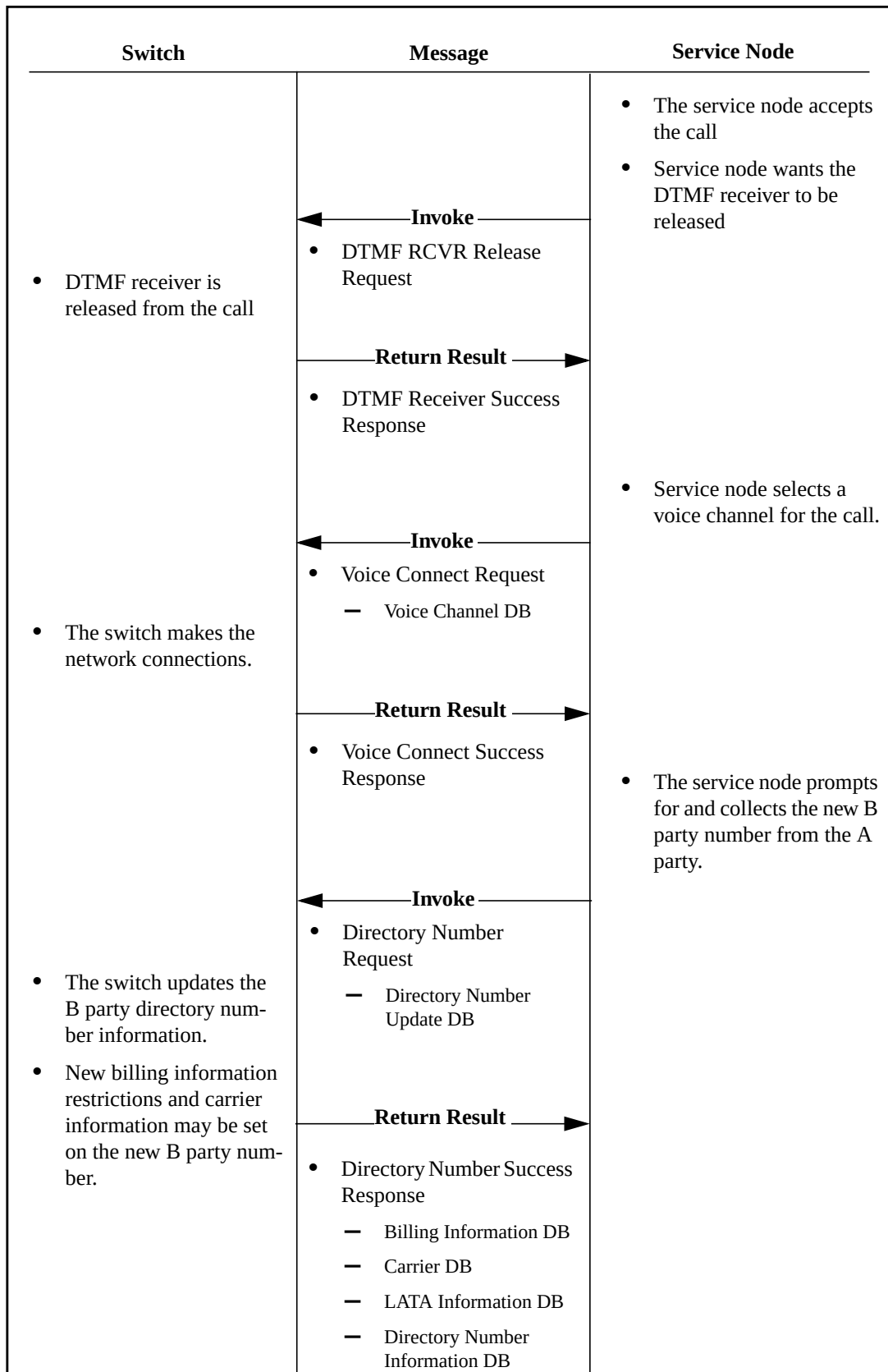
Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0006	6
16			
17	Function ID	#0001	1 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0001	1
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length

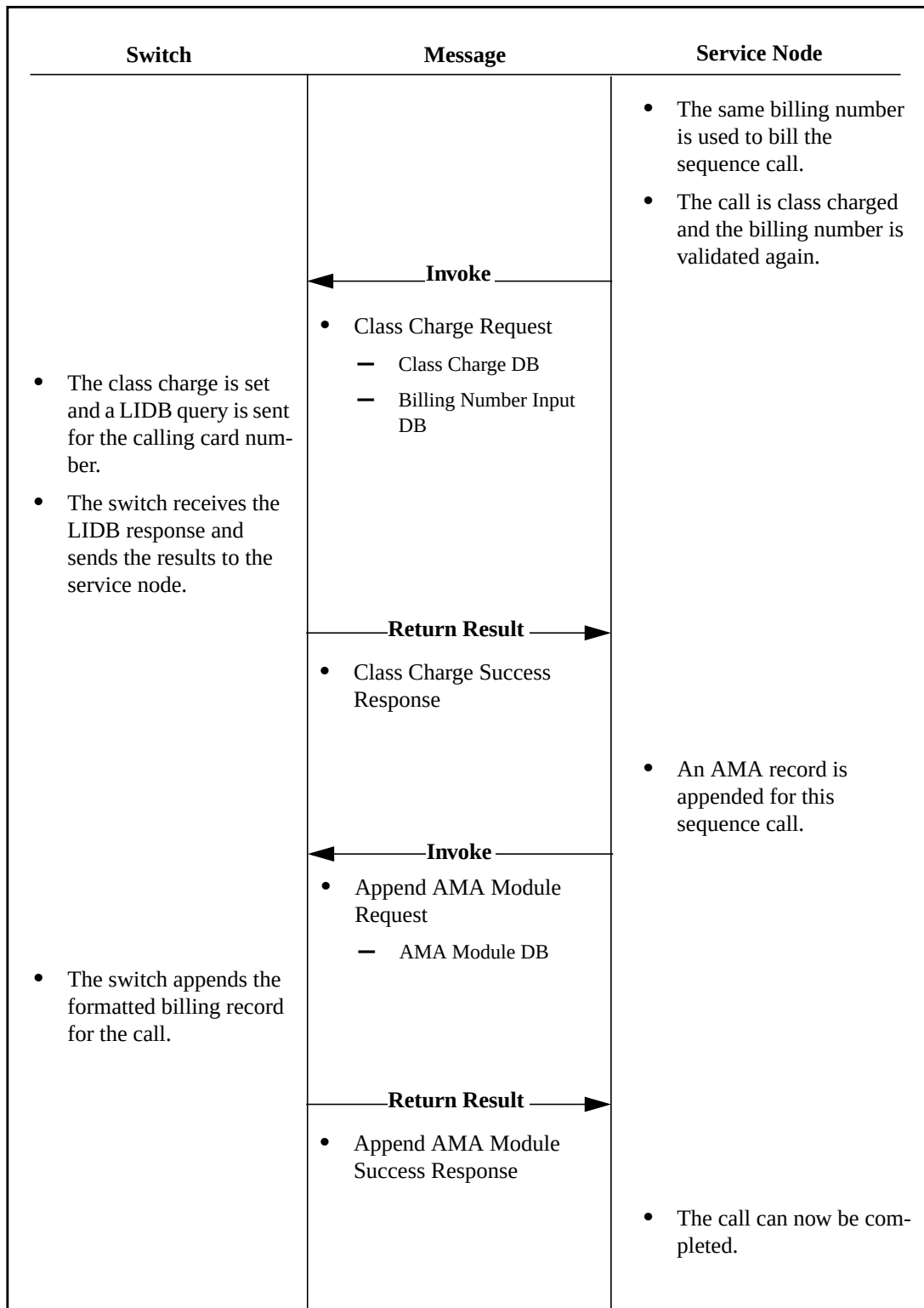
Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
29	Operation/Response Length (remaining header plus data)	#0008	8 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0104	#0104 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

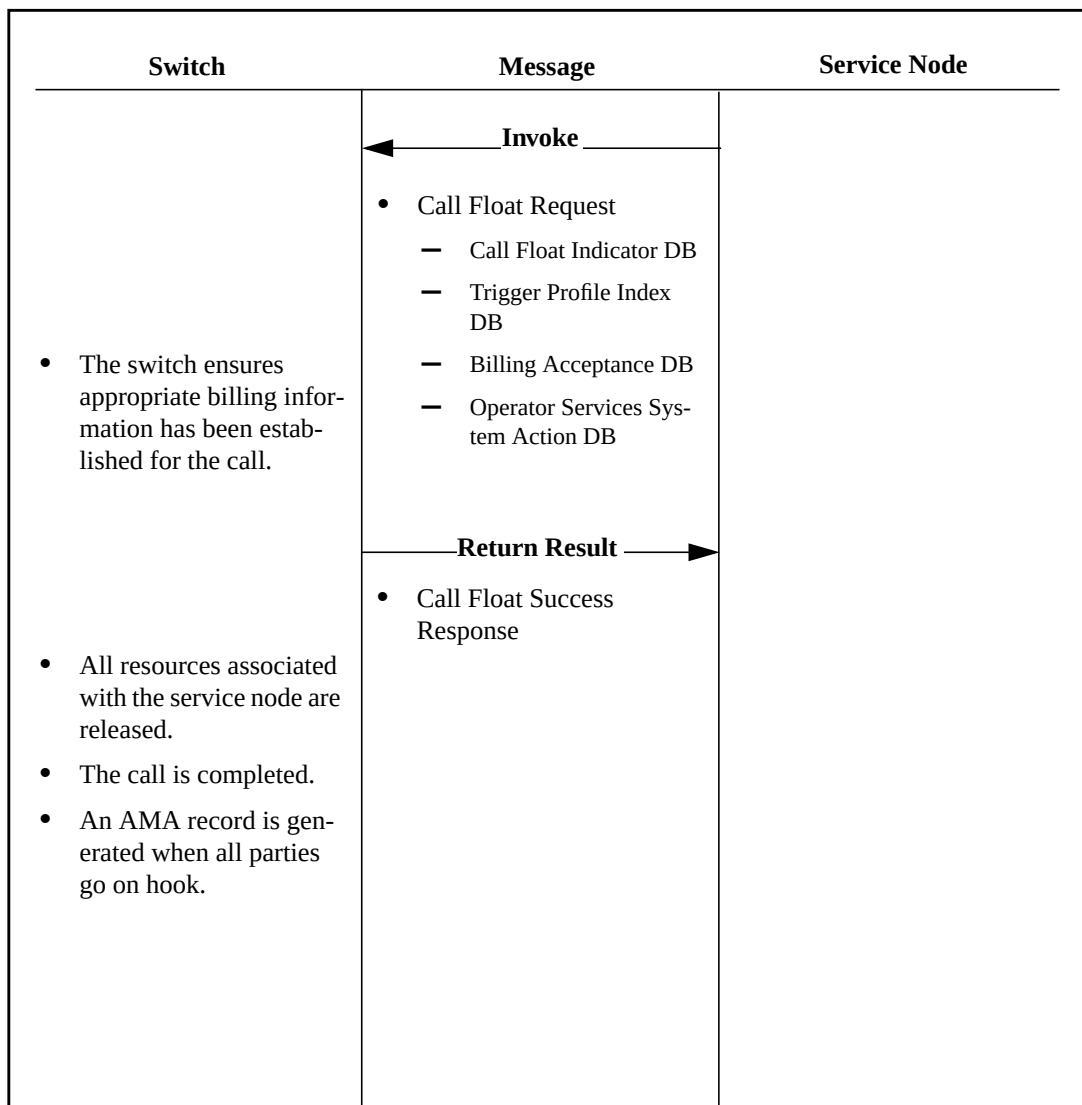
12.18.2 Sequence Call

The following message flow continues from the message flow in Section 12.18.1 “Call Billed to a Calling Card” on page 878. After the parties have been talking for awhile, the B party goes on hook and the A party begins a sequence call.

Switch	Message	Service Node
• A and B parties talking • B party goes on hook • A party begins sequence call (DTMF digit). • B party is released. • An AMA record is generated (based on DTMF trigger datafill). • A session is selected and the call is sent to a service node.	Invoke → • Session Begin Inform — Function DB — Call Information DB — Billing Information DB — Originating Station Information DB — Charge Status DB — Class Charge DB — Carrier DB — LATA Information DB — Time DB — Originating Trunk Information DB — Directory Number DB — Directory Number DB — OLNS Information DB — SPID DB — SPID DB — Node Identification DB — Trigger DTMF DB — BCD Card Number DB	







Session Begin for Sequence Call (Hex Dump):

```
000500000150001A04000400005A0001000500060000000000BBA18201380202010002020
60D048201400114000200050107001002010FFF0002010101000000010101FE0104000600
150FFFF64070119001201000000FFFFFFFFFFFFFFFF00FFFFFFFFFFFFFFFF010A000201FF010
B00060002000000FF01090010030401110101000843313131202020200111001E0101FE00
0101FF0A6193226565FFFFFFFFF0000FFFFFFFFFFFFFFFFFFFFFFFF0111001E020111FE00
01FF0A2012201236FFFFFFFFF0000FFFFFFFFFFFFFFFFFFFFFFFF01220004020102FF0108
001401FF000E082B081B1B080AF5F5F5F5F5F5F5F5F5F5013600020100011E000661040E0
B1E0F011A000400E200040134002E01010101010101010101020401110100FFF0100FFF
F0200FFFFFFFFFFFFFFFFF0107FF0441424344FF00FFFFFFFFF014500080101FF04414243
44014500080201FF044142434401170004000601FF
```

- Call Processing Class Header
- Invoke ROSE Header
- Function DB
- Call Information DB
- Billing Information DB
- Originating Station Information DB
- Charge Status DB
- Class Charge DB
- Carrier DB
- LATA Information DB
- Time DB
- Originating Trunk Information DB
- Directory Number DB
- Directory Number DB
- OLNS Information DB
- SPID DB
- SPID DB
- Node Identification DB
- Trigger DTMF DB
- BCD Card Number DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0150	336 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0001	1
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0138	312 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, Length = 2
33	Invoke ID Value	#0100	#0100
34			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, Length = 2
37	Operation ID Value	#060D	Session Begin Inform
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#012C	300 bytes
42			
43	Data Block ID	#0114	Function DB
44			
45	DB Byte Count	#0002	2 bytes
46			
47	Function ID	#0005	5 (Table OAFUNDEF)
48			
49	Data Block ID	#0107	Call Information DB
50			
51	DB Byte Count	#0010	16 bytes
52			
53	Session Origination Type	#02	Recall/Trigger
54	Service Node Mode	#01	Active
55	Control List ID	#0FFF	4095, NIL (Table OACTLDEF)
56			
57	Trigger Profile Index	#0002	2 (Table OATPRFIX)
58			
59	Call Origination Type	#01	Zero Minus
60	Call In Queue Take Down	#01	Take Call Down
61	DTMF Receiver	#01	Connected
62	Service Type	#00	TA
63	Session Type	#00	Standard Session
64	Conversation Timing Status	#00	Stop Call Timing
65	Branding Status	#01	Call has received front-end branding
66	Notification Timer Status	#01	Not Running
67	SPID Processing	#01	A Party enabled
68	Number of Operator to SN Handoffs	#FE	Nil

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
69	Data Block ID	#0104	Billing Information DB
70			
71	DB Byte Count	#0006	6 bytes
72			
73	Network Service ID	#0015	21 (from Table OACTL-DEF)
74			
75	Allowed Billing Set	#0FFF	All billing is allowed
76			
77	Restricted Billing Index	#64	100 (nil value, unrestricted)
78	Terminating Code Allowed Set	#07	Calling Card, Third number, and Collect billing is allowed
79	Data Block ID	#0119	Originating Station Information DB
80			
81	DB Byte Count	#0012	18 bytes
82			
83	Phone Class Type	#01	Station
84	Coin Overtime Type	#00	Not Applicable
85	Coin Key Pad State	#00	Undetermined
86	Hotel Room Number Length	#00	0
87	Hotel Room Number	#FF #FF #FF #FF #FF #FF	Nil
88			
89			
90			
91			
92			
93	Padding Field	#FF	Nil
94	Authorization Number Length	#00	0
95	Authorization Number	#FF #FF #FF #FF #FF #FF	Nil
96			
97			
98			
99			
100			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
101	Data Block ID	#010A	Charge Status DB
102			
103	DB Byte Count	#0002	2 bytes
104			
105	Billable Status	#01	Billable
106	Padding Field	#FF	Nil
107	Data Block ID	#010B	Class Charge DB
108			
109	DB Byte Count	#0006	6 bytes
110			
111	Handling Data	#00	Station
112	Billing Data	#02	Special Calling
113	Validation Request	#00	Not Applicable
114	Validation Sequence Number	#00	0
115	Class Charge Selector	#00	Current
116	Padding Field	#FF	Nil
117	Data Block ID	#0109	Carrier DB
118			
119	DB Byte Count	#0010	16 bytes
120			
121	Carrier Assignment Source Indicator	#03	OLNS
122	Carrier Identification Digit Count	#04	4 digits
123	Carrier Identification Code	#01	0111
124		#11	
125	Service Type Provided by Switch	#01	Complete Operator Services
126	Carrier Number Status	#01	Valid
127	Carrier Service Status	#00	Valid
128	Carrier Name Length	#08	8 characters

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
129	Carrier Name	#43 #31 #31 #31 #20 #20 #20 #20	C111 space space space space
130			
131			
132			
133			
134			
135			
136			
137	Data Block ID	#0111	Directory Number DB
138			
139	DB Byte Count	#001E	30 bytes
140			
141	Party	#01	A Party
142	Directory Number Type	#01	Domestic
143	Language	#FE	Unknown
144	ANI Status	#00	ANI Success
145	Connection Status	#01	Connected, Off Hook
146	Speech Path Status	#01	Two-Way
147	Presentation Indicator	#FF	Presentation Status Unknown
148	Directory Number Digit Count	#0A	10 digits
149	Directory Number	#61 #93 #22 #65 #65 #FF #FF #FF #FF #FF	619-322-6565
150			
151			
152			
153			
154			
155			
156			
157			
158			
159	Unenforced Restrictions	#00	No DN unenforced restrictions
160	Calling Line Identifier Digit Count	#00	0 digits

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
161	Calling Line Identifier Number	#FF #FF #FF #FF #FF #FF #FF #FF #FF #FF	Nil
162			
163			
164			
165			
166			
167			
168			
169			
170			
171	Data Block ID	#0111	Directory Number DB
172			
173	DB Byte Count	#001E	30 bytes
174			
175	Party	#02	B Party
176	Directory Number Type	#01	Domestic
177	Language	#11	17 (from Table TOPS-LANG)
178	ANI Status	#FE	Unknown
179	Connection Status	#00	Not Connected
180	Speech Path Status	#01	Two-Way
181	Presentation Indicator	#FF	Presentation Status Unknown
182	Directory Number Digit Count	#0A	10 digits
183	Directory Number	#20 #12 #20 #12 #36 #FF #FF #FF #FF #FF	201-220-1236
184			
185			
186			
187			
188			
189			
190			
191			
192			
193	Unenforced Restrictions	#00	No DN unenforced restrictions

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
194	Calling Line Identifier Digit Count	#00	0 digits
195	Calling Line Identifier Number	#FF #FF #FF #FF #FF #FF #FF #FF #FF	Nil
196			
197			
198			
199			
200			
201			
202			
203			
204			
205	Data Block ID	#0122	Trigger DTMF DB
206			
207	DB Byte Count	#0004	4 bytes
208			
209	Trigger Detection Point	#02	Disconnect
210	Party	#01	A Party
211	DTMF Digit	#02	2
212	Padding Field	#FF	Nil
213	Data Block ID	#0108	BCD Card Number DB
214			
215	DB Byte Count	#0014	20 bytes
216			
217	BCD Card Number Type	#01	Directory Number Card
218	Padding Field	#FF	Nil
219	BCD PIN Digit Count	#00	0 digits
220	BCD Card Number Digit Count	#0E	14 digits

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
221	BCD Card Number	#08 #2B #08 #1B #1B #08 #0A #F5 #F5 #F5 #F5 #F5 #F5 #F5 #F5	Encrypted BCD Number
222			
223			
224			
225			
226			
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237	Data Block ID	#0136	LATA Information DB
238			
239	DB Byte Count	#0002	2 Bytes
240			
241	LATA Status	#01	InterLATA
242	LATA Restrictions	#00	No Restrictions
243	Data Block ID	#011E	Time DB
244			
245	DB Byte Count	#0006	6 bytes
246			
247	Year	#61	97
248	Month	#04	04
249	Day	#0E	14
250	Hour	#0B	11
251	Minutes	#1E	30
252	Seconds	#0F	15
253	Data Block ID	#011A	Originating Trunk Information DB
254			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
255	DB Byte Count	#0004	4 bytes
256			
257	Trunk Group ID	#00E2	226 (Table CLLI, field ADNUM)
258			
259	Trunk Member ID	#0004	4 (Table TRKMEM)
260			
261	Data Block ID	#0134	OLNS Information DB
262			
263	DB Byte Count	#002E	46 bytes
264			
265	OLNS TA Sent Paid Indicator	#01	Allowed
266	OLNS TA Collect Indicator	#01	Allowed
267	OLNS TA Third Number Indicator	#01	Allowed
268	OLNS TA Credit Card Indicator	#01	Allowed
269	OLNS Special BNS Indicator	#01	Allowed
270	OLNS DACC Indicator	#01	Allowed
271	OLNS DA Sent Paid Indicator	#01	Allowed
272	OLNS DA Third Number Indicator	#01	Allowed
273	OLNS DA Credit Card Indicator	#01	Allowed
274	OLNS DA Special BNS Indicator	#01	Allowed
275	OLNS InterLATA Carrier Indicator	#02	IC Indicated
276	OLNS InterLATA Carrier ID Digit Count	#04	4 digits
277	OLNS InterLATA Carrier ID Code	#01	0111
278		#11	
279	OLNS IntraLATA Toll Carrier Indicator	#01	Not Indicated
280	OLNS IntraLATA Toll Carrier ID Digit Count	#00	0 digits
281	OLNS IntraLATA Toll Carrier ID Code	FFFF	Nil
282			
283	OLNS International Carrier Indicator	#01	Not Indicated
284	OLNS International Carrier ID Digit Count	#00	0 digits
285	OLNS International Carrier ID Code	FFFF	Nil
286			
287	OLNS Foreign Language	#02	2, French
288	OLNS Alphanumeric String Length	#00	0 characters

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
289	OLNS Alphanumeric String	#FF #FF #FF #FF #FF #FF #FF #FF	Nil
290			
291			
292			
293			
294			
295			
296			
297	OLNS Service/Equipment Indicator	#01	POTS
298	OLNS Treatment Indicator	#07	Automated Treatment
299	Padding Field	#FF	Nil
300	OLNS Account Owner Length	#04	4 characters
301	OLNS Account Owner	#41 #42 #43 #44	ABCD
302			
303			
304			
305	Padding Field	#FF	Nil
306	OLNS Billing Service Provider Length	#00	0 characters
307	OLNS Billing Service Provider	#FF #FF #FF #FF	FFFF
308			
309			
310			
311	Data Block ID	#0145	SPID DB
312			
313	DB Byte Count	#0008	8 bytes
314			
315	Service Provider Party Identifier	#01	Originating Party
316	Type of Service Provider	#01	Account Owner
317	Padding Field	#FF	Nil
318	Service Provider ID Length	#04	4 characters
319	Service Provider ID	#41 #42 #43 #44	ABCD
320			
321			
322			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
323	Data Block ID	#0145	SPID DB
324			
325	DB Byte Count	#0008	8 bytes
326			
327	Service Provider Party Identifier	#02	Terminating Party
328	Type of Service Provider	#01	Account Owner
329	Padding Field	#FF	Nil
330	Service Provider ID Length	#04	4 characters
331	Service Provider ID	#41	ABCD
332		#42	
333		#43	
334		#44	
335	Data Block ID	#0117	Node Identification DB
336			
337	DB Byte Count	#0004	4 bytes
338			
339	Node ID	#0006	6 (Table OANODNAM)
340			
341	Service Node Mode	#01	Active
342	Padding Field	#FF	Nil

DTMF RCVR Release Request (Hex Dump)

000500000024001A0400040000060001000500060000000000BBA182000C0202010002020
61704820000

- Call Processing Class Header
- Invoke ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
5	Length in bytes of following Message	#0024	36 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0001	1
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#000C	12 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, Length = 2
33	Invoke ID Value	#0100	#0100
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, Length = 2
37	Operation ID Value	#0617	DTMF RCVR Release Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
41	Argument Length	#0000	0 bytes
42			

DTMF RCVR Release Success Response (Hex Dump):

000500000020001A04000400005A0002000500060000000000BBA28200080202010004820000

- Call Processing Class Header
- Return Result ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0002	2
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length	#0008	8 bytes
30	(remaining header plus data)		
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0100	#0100 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

Voice Connect Request (Hex Dump):

00050000002A001A0400040000060002000500060000000000BBA18200120202010102020
615048200060127000203FE

- Call Processing Class Header
- Invoke ROSE Header
- Voice Channel DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#002A	42 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0002	2
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0012	18 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, Length = 2
33	Invoke ID Value	#0101	#0101
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, Length = 2
37	Operation ID Value	#0615	Voice Connect Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0006	6 bytes
42			
43	Data Block ID	#0127	Voice Channel DB
44			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
45	DB Byte Count	#0002	2 bytes
46			
47	Voice Channel ID	#03FE	1022 (from Table OAV-LMAP for each service node)
48			

Voice Connect Success Response (Hex Dump):

000500000020001A04000400005A0003000500060000000000BBA28200080202010104820000

- Call Processing Class Header
- Return Result ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0003	3
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0008	8 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0101	#0101 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

Directory Number Request (Hex Dump):

000500000042001A0400040000060003000500060000000000BBA182002A0202010202020
6070482001E0112001A0211010A2012202456FFFFFFFFFFFFFFFF00FFFFFFFFFFFFFFFFFFFF
F

- Call Processing Class Header
- Invoke ROSE Header
- Directory Number Update DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
5	Length in bytes of following Message	#0042	66 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0003	3
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#002A	42 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0102	#0102
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0607	Directory Number Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
41	Argument Length	#001E	30 bytes
42			
43	Data Block ID	#0112	Directory Number Update DB
44			
45	DB Byte Count	#001A	26 bytes
46			
47	Party	#02	B Party
48	Language	#11	17 (Table TOPSLANG)
49	Directory Number Type	#01	Domestic
50	Directory Number Digit Count	#0A	10 digits
51	Directory Number	#20 #12 #20 #24 #56 #FF #FF #FF #FF #FF	201-220-2456
52			
53			
54			
55			
56			
57			
58			
59			
60			
61	Presentation Indicator	#FF	Presentation Status Unknown
62	Calling Line Identifier Digit Count	#00	0 digits
63	Calling Line Identifier Number	#FF #FF #FF #FF #FF #FF #FF #FF #FF #FF	Nil
64			
65			
66			
67			
68			
69			
70			
71			
72			

Directory Number Success Response (Hex Dump):

00050000004C001A04000400005A0004000500060000000000BBA2820034020201020482
002A0104000600150FFF640701090010030401110101000843313131202020200136020100
01560002020100FF

- Call Processing Class Header
- Return Result ROSE Header
- Billing Information DB
- Carrier DB
- LATA Information DB
- Directory Number Information DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#004C	76 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0004	4
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0034	52 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0102	#0102 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#002A	42 bytes
38			
39	Data Block ID	#0104	Billing Information DB
40			
41	DB Byte Count	#0006	6 bytes
42			
43	Network Service ID	#0015	21 (from Table OACTL-DEF)
44			
45	Allowed Billing Set	#0FFF	All billing is allowed
46			
47	Restricted Billing Index	#64	100 (nil value, unrestricted)
48	Terminating Code Allowed Set	#07	Calling Card, Third number, and Collect billing is allowed
49	Data Block ID	#0109	Carrier DB
50			
51	DB Byte Count	#0010	16 bytes
52			
53	Carrier Assignment Source Indicator	#03	OLNS
54	Carrier Identification Digit Count	#04	4 digits
55	Carrier Identification Code	#01	0111
56		#11	

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
57	Service Type Provided by Switch	#01	Complete Operator Services
58	Carrier Number Status	#01	Valid
59	Carrier Service Status	#00	Valid
60	Carrie Name Length	#08	8 characters
61	Carrier Name	#43 #31 #31 #31 #20 #20 #20 #20	C111 space space space space
62			
63			
64			
65			
66			
67			
68			
69	Data Block ID	#0136	LATA Information DB
70			
71	DB Byte Count	#02	2 bytes
72			
73	LATA Status	#01	InterLATA
74	LATA Restrictions	#00	No Restrictions
75	Data Block ID	#0156	Directory Number Information DB
76			
77	DB Byte Count	#0002	4 bytes
78			
79	Party	#02	B Party
80	Directory Number Tytpe	#01	Domestic Non-Local
81	Unenforced Restrictions	#00	No DN unenforced restrictions
82	Padding Field	#FF	Nil

Class Charge Request (Hex Dump):

```
000500000034001A0400040000060004000500060000000000BBA182001C0202010302020
60304820010010B00060002020100FF0131000200FF
```

- Call Processing Class Header
- Invoke ROSE Header
- Class Charge DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0034	52 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODINV)
14			
15	Sequence Number	#0004	4
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#001C	28 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0103	#0103
34			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
35	Operation ID Type	#02	#02, Fixed
36	Operation ID Length Format	#02	#02, Fixed
37	Operation ID Value	#0603	Class Charge Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0010	16 bytes
42			
43	Data Block ID	#010B	Class Charge DB
44			
45	DB Byte Count	#0006	6 bytes
46			
47	Handling Data	#00	Station
48	Billing Data	#02	Special Calling
49	Validation Request	#02	Validate
50	Validation Sequence Number	#01	1
51	Class Charge Selector	#00	Current
52	Padding Field	#FF	Nil
53	Data Block ID	#0131	Billing Number Input
54			
55	DB Byte Count	#0002	2 bytes
56			
57	Billing Number Input Method	#00	DTMF
58	Padding Field	#FF	Nil

Class Charge Success Response (Hex Dump):

000500000030001A04000400005A000500060000000000BBA282001802020103048200100
10500060104000302F1012D000202FF

- Call Processing Class Header
- Return Result ROSE Header
- Billing Number Validation DB
- CCV Query Result

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0030	48 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0005	65
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (from Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#0018	24 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0103	#0103 (from request)
34			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0010	16 bytes
38			
39	Data Block ID	#0105	Billing Number Validation DB
40			
41	DB Byte Count	#0006	6 bytes
42			
43	Validation Sequence Number	#01	1
44	Validation Result	#04	Billing Accepted
45	Validation Result Determined By DMS Datafill	#00	Validation Result Not Determined by DMS Datafill
46	RAO Digit Count	#03	3 digits
47	RAO Code	#02	201
48		#F1	
49	Data Block ID	#012D	CCV Query Result
50			
51	DB Byte Count	#0002	2 bytes
52			
53	CCV Query Response	#02	PIN Unrestricted
54	Padding Field	#FF	Nil

Append AMA Module Request (Hex Dump):

```
000500000072001A0400040000060005000500060000000000BBA182005A0202010402020
6000482004E0103004A0100FF42001C001CF3666C001C1C001C001CFFFFFFF002C
001CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF
FFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF000CFFFFFFF
```

- Call Processing Class Header
- Invoke ROSE Header
- AMA Module DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0072	114 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0005	5
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#005A	90 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0104	#0104
34			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0600	Append AMA Module Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#004E	78 bytes
42			
43	Data Block ID	#0103	AMA Module DB
44			
45	DB Byte Count	#004A	74 bytes
46			
47	Append AMA Sequence Number	#01	1
48	AMA Handling	#00	Do Not Clear Previously Appended AMA
49	Padding Field	#FF	Nil
50	AMA Hex Digit Count	#42	66 hex digits

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
51	AMA Data	#00 #1C #00 #1C #F3 #66 #6C #00 #1C #1C #00 #1C #00 #1C #FF #FF #FF #FF #FC #00 #2C #00 #1C #FF #FF #FF #FF #FF #FF #FF #FC #00 #0C #FF #FF #FF	Block of AMA data (format unknown to switch)
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
87	AMA Data (continued)	#FF	Block of AMA data continued
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			

Append AMA Module Success Response (Hex Dump):

000500000020001A04000400005A0006000500060000000000BBA28200080202010404820000

- Call Processing Class Header
- Return Result ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0006	6
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
29	Operation/Response Length (remaining header plus data)	#0008	8 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, short, length = 2
33	Invoke ID Value	#0104	#0104 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

Call Float Request (Hex Dump):

000500000044001A0400040000060006000500060000000000BBA182002C0202010502020
606048200200128000200FF0124000AFF00000100000000000000130000200FF0133000201
FF

- Call Processing Class Header
- Invoke ROSE Header
- Call Float Indicator DB
- Trigger Profile Index DB
- Billing Acceptance DB
- Operator Services System Action DB

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0044	68 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#0006	6 (Table OANODNAM)
14			
15	Sequence Number	#0006	6
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A1	Invoke
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length (remaining header plus data)	#002C	44 bytes
30			
31	Invoke ID Type	#02	#02, Integer
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0105	#0105
34			
35	Operation ID Type	#02	#02, Integer
36	Operation ID Length Format	#02	#02, Short, length = 2
37	Operation ID Value	#0606	Call Float Request
38			
39	Argument Type	#04	#04, Octet String
40	Argument Length Format	#82	#82, Long, 2 byte length
41	Argument Length	#0020	32 bytes
42			
43	Data Block ID	#0128	Call Float Indicator DB
44			

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
45	DB Byte Count	#0002	2 bytes
46			
47	Float Connection Indicator	#00	Unconnected or Connected
48	Padding Field	#FF	Nil
49	Data Block ID	#0124	Trigger Profile Index DB
50			
51	DB Byte Count	#000A	10 bytes
52			
53	Padding Field	#FF	Nil
54	No Answer Time	#00	0 seconds
55	Trigger Profile Index	#0001	1 (table OATPRFIX)
56			
57	Notification Time - Elapsed Minutes	#00	0 Minutes
58		#00	
59		#00	
60	Notification Time - Seconds	#00	0 Seconds
61	Notification Time Type	#00	Relative Time
62	Notification Timer Mode	#00	Float Style
63	Data Block ID	#0130	Billing Acceptance DB
64			
65	DB Byte Count	#0002	2 bytes
66			
67	Billing Acceptance Response Method	#00	DTMF
68	Padding Field	#FF	Nil
69	Data Block ID	#0133	Operator Services System Action DB
70			
71	DB Byte Count	#0002	2 bytes
72			
73	Operator Services System Action	#01	Automatically Accept
74	Padding Field	#FF	Nil

Call Float Success Response (Hex Dump):

000500000020001A04000400005A0007000500060000000000BBA28200080202010504820000

- Call Processing Class Header
- Return Result ROSE Header

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
1	OAP Class Header ID	#0005	Call Processing Class Header
2			
3	Options	#0000	No bits set
4			
5	Length in bytes of following Message	#0020	32 bytes
6			
7	Offset (from Class Header ID to Operation Data)	#001A	26 bytes
8			
9	Message Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
10			
11	Sender's Max Protocol Version	#0600	Msg Release = 6 Msg Increment = 0
12			
13	Source Node ID	#005A	90 (Table OANOD-NAM)
14			
15	Sequence Number	#0007	7
16			
17	Function ID	#0005	5 (Table OAFUNDEF)
18			
19	Session Pool ID	#0006	6 (Table OASESNPL)
20			
21	Session ID	#0000	0
22			
23	Call ID	#0000 #00BB	#0000 00BB
24			
25			
26			
27	Operation/Response Type	#A2	Return Result
28	Operation/Response Length Format	#82	#82, Long, 2 byte length
29	Operation/Response Length	#0008	8 bytes
30	(remaining header plus data)		
31	Invoke ID Type	#02	#02, Integer

Byte #	OAP Field Name	Actual Numeric Value	Actual Symbolic Value
32	Invoke ID Length Format	#02	#02, Short, length = 2
33	Invoke ID Value	#0105	#0105 (from request)
34			
35	Result Type	#04	#04, Octet String
36	Result Length Format	#82	#82, Long, 2 byte length
37	Result Length	#0000	0 bytes
38			

Chapter 13: Functionality Overview

13.1 Automated Alternate Billing Service (AABS) - Transition from an SN

Transitions from a service node to the TOPS AABS system are accomplished through the OAP Transfer to Control List operation. TOPS AABS can be used to either:

1. Obtain the billing information (i.e. collect, third number, or calling card number) from the calling subscriber and obtain billing acceptance (for collect and third calls) from the called/third party. The call is then floated from the AABS node.
2. If the billing information has already been obtained (i.e. the billing number is present and the call has been class charged), AABS can be used to obtain the billing acceptance (for collect and third calls) and then float the call. This use of TOPS AABS is often referred to as a “hand-off”.

The switch will determine whether the call is being sent to AABS for full billing of the call or if the call is being sent to AABS as a hand-off. This determination is made by checking the call details to see if the call has been class charged (i.e. if the call has been class charged, then it is treated as a hand-off).

Before a service node can transfer the call to the TOPS AABS system, it is responsible for verifying that call details are correct for AABS. If the details are not correct, the call will be sent to a compatible operator. The following list identifies the requirements that must be satisfied to send a call to TOPS AABS:

- The calling number must be present and the calling party must be connected.
- A valid called/third number must be present, but the called/third party can not be connected to the call.
- The set of valid billing options for the called/third number must include at least one of the three billing types handled by TOPS AABS (calling card, collect, and third number).
- The called/third number must not be restricted to Person Only calls (i.e. they must be capable of receiving a Station class charge).
- The originating station cannot be a rotary service.
- If the call is an equal access call which is to be routed to a carrier which has contracted for operator services with the Operating Company, then they must have switch datafill that indicates they wish to have zero-plus calls automated.
- The incoming trunk facility must be datafilled in the switch as one which can receive AABS service if there is no class charge.
- For hand-off calls, the call must be class charged as either collect or bill to third and the valid billing types for the called/third number must include the desired class charge.

13.2 Directory Assistance

Calls receiving Directory Assistance service are typically sent to the service node to obtain alternate billing to a third number or calling card number. Once billing has been established, the call should be routed to a control list to continue DA processing within the DMS switch environment. These calls should only be disposed of via a Transfer to Control List operation or an End Call operation. Any other attempt to dispose of the call via operations such as Call Float, Transfer to Carrier, or Route to Treatment is denied by the DMS. Another restriction for DA calls is that they cannot be class charged as collect.

For calls being transferred to a DA operator, ADAS/ADASPLUS eligibility checks are performed and the call is routed to ADAS processing when applicable. Note that this is true for both TA and DA calls being routed to a DA operator. The DMS switch changes the service of TA calls to DA when the call is routed to a DA operator. Likewise, the switch changes the service of DA calls to TA when the call is routed to a TA operator.

13.3 Equal Access

Service nodes are given the capability of assigning a carrier to an interlata call. The switch will return information about the carrier, indicating whether or not the switch provides operator services for the carrier (in the Carrier DB). The switch will also return carrier information anytime a carrier is assigned for the call through signalling, translations, or OLNS data (Carrier Assignment Source Indicator field in the Carrier DB).

If the switch provides operator services for the carrier, the service node may request to float the call (if all other preconditions for float are met). The call cannot be transferred to the carrier if the switch provides operator services for the carrier. The switch will return an error response if the service node sends the Transfer to Carrier Request.

If the switch does not provide operator services for the carrier, the service node must transfer the call to the carrier (Transfer to Carrier Request). The switch will return an error response if the service node attempts to float the call.

If the call is interlata and there is no carrier assigned to the call, the service node must enter a carrier in order to float (the switch must provide operator services for the carrier). Otherwise, the service node can either end the call or transfer the call to another agent (Transfer to Control List Request).

Note that a zero plus interlata call to a noserv carrier will not route to a service node.

Calls assigned in switch table LATAXLA as Corridor or Privilege are interlata calls that are completed over local operating company trunks, not carrier trunks. Such calls, while technically interlata, are presented to the SN with a LATA status of intralata. The handling of these calls is identical to intralata calls (i.e., no carrier needs to be assigned, they are floated instead of transferred to carrier).

The switch performs equal access translations if the following conditions are met:

- originating trunk information is available
- the originating trunk is either an MF TOPS trunk or an ISUP TOPS trunk
- the call is billable

For a subscriber originated session, the originating trunk information is available immediately. Equal access translations is performed when the B party DN is updated and connected.

For a service node initiated session, this information may or may not be available at the appropriate time. The following sections describe what must be done in order for the switch to perform equal access translations.

13.3.1 Service Node Initiated Session: Connecting to the A Party

For a Session Initiation, the order of operations is not enforced, and therefore, the appropriate information may not be available when needed for equal access translations. In order for equal access translations to be performed, the following order must be used by the service node.

1. Initiate a session with the switch via the Session Initiation Request.
2. Enter the A party DN via the Directory Number Request.
 - The A party DN could have been sent with the Session Initiation Request.
3. Connect to the A party via the Connect DN Request.
 - The originating information is now available at the switch.
4. Enter the B party DN via the Directory Number Request
 - Equal access translations is performed.
5. Assign a carrier to the call via the Carrier Assignment Request
 - This is optional if a carrier is already assigned and mandatory if this is an interlata call with no carrier assigned.
 - In order to float the call or connect to the B party, the switch must provide operator services for the carrier.
6. Connect to the B party DN via the Connect DN Request
 - or float the call via the Call Float Request with option to connect the B party.

Note: If a service node initiated session is used to connect to only one party, the party should be treated as if it is the A party.

13.3.2 Service Node Initiated Session: Not Connecting the A Party

In some cases, the A party is not connected for a Session Initiation. This is true for a 3-way conferencing scenario where the A party is already connected in the first leg (the A-B party leg) and the second session (initiated by the service node) is used to connect to the C party.

In order for the switch to do equal access translations, the A party trunk information is needed. Since the A party is not connected in the C-party session, this information is not available at the switch. However, the service node has this information from the A-B party session. The service node must send the originating trunk information to the switch prior to updating the B party DN.

The following is the order of operations for initiating a session and not connecting to the A party:

1. Initiate a session with the switch via a Session Initiation Request.
2. Enter the A party DN via the Directory Number Request.
 - Send the Originating Trunk Information DB
 - The A party DN and the Originating Trunk Information DB could have been sent with the Session Initiation Request.
3. Enter the B party DN via the Directory Number Request.
 - Equal access translations is performed since the switch has the originating trunk information.
4. Assign a carrier to the call via the Carrier Assignment Request.
 - This is optional if a carrier is already assigned and mandatory if this is an interlata call with no carrier assigned.
 - If the switch returned a Carrier DB in the Directory Number Success Response for the B party, the service node should compare it to the preferred carrier for the A party returned in the OLNS Information DB sent in the A-B party session. If the carrier is not the same, the service node should update the switch with the correct carrier.
 - In order to float the call or connect to the B party, the switch must provide operator services for the carrier.
5. Connect to the B party DN via the Connect DN Request.

13.4 Estimated Charges

This feature allows the Service Node to request that the switch perform an estimate of charges for a call. Once all of the information required to successfully rate the call has been obtained from the subscriber, the SN requests an estimate of charges be performed. The switch performs the calculation and the Service Node is informed of the charges.

Estimating charges functionality is only applicable to calls using internal switch rating. The determination as to whether a call is using internal or external rating is based on datafill in switch table COMPCODE or table TOPSPARM parameter RTRS_DEFAULT_SYSTEM. Requests for estimating charges on an externally-rated call will result in an error response.

All estimate requests are divided into two types:

- Duration Estimate. For this type, the SN has specified a duration to be used when determining charges for the call. The switch response will contain the calculated charges for a call of the specified duration.
- No Duration Estimate. In this type, the SN does not specify a duration. The switch response will contain the calculated charges for the initial and subsequent periods.

Along with the type, estimate requests take the form of one of the following regarding the time and date:

- Calculate the charges for a call using the current date and time. In this type of request the SN does not need to specify a date or time.
- Calculate the charges using the current date and some other SN-specified time.
- Calculate the charges using the current time and some other SN-specified date.
- Calculate the charges using some other SN-specified date and time.

Other attributes besides date, time and duration that affect the rating of a call include the following: calling number, called number (domestic and overseas), call origination type, station class (for example, hotel, coin, non-coin), class charge (for example, station paid, per collect) and rate step.

13.4.1 Estimate Time

If the estimate is for the current time, the SN does not need to specify the hour and minute in the request. For an estimate of a call that is some time other than the current time, the service node must specify an alternate hour and minute to use in the estimation. The time is based on a 24-hour clock and consists of an hour and minute.

Note that if the current time is 1500 hours and the SN specifies 1300 hours, then the estimate is done for 1300 hours of the *same* day.

If the Service Node does not specify a time, then current switch time is assumed.

13.4.2 Estimate Date

If the estimate is for the current date, the SN does not need to specify the day and month in the request. For an estimate of a call that is some date other than the current date, the Service Node must specify an alternate day and month to use in the estimation.

Note that for the date, the TOPS switch will do a rolling twelve month look-ahead depending on the input. For example, if the current month is November and the service node specifies an October date, TOPS assumes October of the *following* year.

If the service node does not enter a date, then the current switch date is assumed.

13.4.2.1 Estimate Time and Date Interactions

The following interactions exist between the time and date depending on what information is sent by the service node. Note that the switch does not do any backwards checks. If information entered is less than the current time/date, TOPS assumes next day or year. The following table details the interactions.

The service node cannot specify a year; however, because the estimate can be done for a 12-month rolling look ahead, the year may need to be incremented by one; for example; it's December and the SN wants to do an estimate for January. This incrementing of the year will be handled by the switch.

The accuracy of the estimate will depend largely on how up-to-date the Operating Company keeps their rating tables.

Table 16 Date and Time Interactions

Case #	Month	Day	Hour	Minutes	Action Taken by the Switch
1	not specified	not specified	not specified	not specified	When the service node does not enter any date or time information, then the current switch date and time are used for the estimated call.
2	not specified	not specified	<i>specified</i>	<i>specified</i>	The service node specified the time for the estimated call but no date. There are two possible scenarios that can result here: 1. The time specified is greater than the current time, therefore, use the time specified, plus the current date for the estimate. 2. The time specified is less than the current time (for example, it's currently 1500 hours and the service node specified 1200 hours), then assume time entered is for the same day plus the current date for the estimate.

Table 16 Date and Time Interactions

Case #	Month	Day	Hour	Minutes	Action Taken by the Switch
3	<i>specified</i>	<i>specified</i>	not specified	not specified	The service node specified the date for estimated call but not the time. There are several possible scenarios that can result: 1. The month specified is greater than the current month, use the month, day, and current time for the estimate. 2. The month specified is less than the current month, assume same month, day, and current time in following year. 3. The month is the same as the current month, but the day is less than the current day, assume same month/day of the following year. 4. The month is the same as the current month and the day is greater than the current day, use input as is.
4	<i>specified</i>	<i>specified</i>	<i>specified</i>	<i>specified</i>	The service node specified time and date. The handling of this is the same as the handling of the case 3 in addition to the following: 1. The month and day specified are the same as the current month and day, but the time specified is less than the current time, then assume estimate is for the same day. 2. The month and day specified are the same as the current month and day and the time is greater than the current time, use input as is.

13.4.3 Estimates with/without a specified duration

The Service Node can optionally specify the length of the call to be estimated. For example, the SN may specify that the duration of the estimated call be 30. The switch then calculates the total charges for a call lasting 30 minutes with the existing call attributes.

For North American switches, the largest duration that can be handled by rating is 999 minutes. Anything larger than 999 minutes will fail to rate.

If a duration is not specified, then the Service Node is informed of the charges and length of the initial period and the charges and length of the subsequent period for a call with the existing attributes.

13.4.4 AMA considerations

A single Module Code 611 is automatically appended to the AMA record as a result of the *last* Estimate of Charges Request for a call. If multiple estimate requests are made on the same call, the SN must send in a Generate AMA Record Request after each Estimate of Charges Request (except the last one) in order to generate a Module Code 611 for each request.

13.5 Local Number Portability

13.5.1 Background

Local Number Portability (LNP) is a circuit switched network capability that allows a user on one switching exchange (i.e., end office) to keep the same public directory number when moving to a different switching exchange. Information needed to process calls in an LNP environment is maintained in an external database.

TOPS LNP is based on requirements developed by the Illinois Number Portability Workshop and published in the Illinois Commerce Commission (ICC) Generic Operator Services Switching Requirements for Number Portability.

LNP allows a subscriber to keep the same directory number (DN) when moving to a different switch. A directory number that has been moved from one switch to another is referred to as ported. Other users can connect to ported numbers without any changes to their dialing procedures.

Historically, all DNs with the same first six digits (i.e., the NPA-NXX) belonged to a single telephone switch. With the introduction of LNP, individual DNs in one NPA-NXX may be ported to different telephone switches. This has tremendous implications for call routing, as the first six digits of a DN no longer uniquely identify the switch that hosts the DN.

Hence the concept of a Location Routing Number (LRN) was introduced. Each switch which hosts ported numbers has a ten-digit LRN assigned to it. The LRN is used in routing calls to subscribers whose numbers have been ported. With ISUP trunks, both the LRN and the dialed digits are signaled. With MF trunks, only the dialed digits are signaled.

An NPA-NXX is referred to as portable if it may include ported DNs. When datafill indicates that a number's NPA-NXX is portable, the DMS switch sends a query to an LNP SCP (Service Control Point) database to obtain portability information about the number. The LNP SCP sends a response to the DMS switch with the LRN, if the number is ported, or with the queried number if it is not ported. This information is then used for AMA recording and/or to route the call to its correct destination.

For TOPS, LNP affects the way numbers are translated and outpulsed, and it creates a need for additional information when they are used in billing records. As a result, LNP particularly affects alternate billing services (ABS) and busy line verification (BLV). ABS calls (e.g., collect, person-to-person, third party billed, calling card) are impacted because of the verification required on the alternative billing information, and the additional information required for billing.

Number portability may be defined as Service Provider Portability, Location Portability, and Service Portability.

- Service Provider Portability allows an end user to change access/service providers without changing telephone number.

- Location Portability allows the end user to change locations without changing telephone number.

Service Portability allows the end user to change service mix (e.g., POTS to ISDN) without changing telephone number. Currently with TOPS, LNP functionality includes facility-based Service Provider Portability and Location Portability within a rate center. The AMA record identifies only the facility-based service provider. Service Portability, as defined above, is not applicable to operator services switches.

13.5.2 LNP Information

The Service Node may request the LNP information for the A, B, and billed parties using the LNP Request Operation. If the LNP information is not readily available from a previous LNP query, an LNP query will be launched for the DN.

When the switch launches an LNP query, the switch is in a “Restricted Input Mode” while waiting for the query result. This means that the switch will not accept any request until the query is completed.

If the LNP information is known for the A, B and/or the billed party, the information will be included via the LNP Information DB, with the following messages:

- Session Initiation Success Response
- Call Details Success Response
- Session Begin Inform
- Trigger Event Inform
- Accept Control Inform

An LNP Information DB for the DN involved in the operation may be included with the following messages:

- Billing Number Success Response (B party DN)
- Class Charge Success Response (third or line-based calling card number)
- Connect DN Success Response (A, B or billed party DN)
- Directory Number Success Response (A or B party DN)

13.5.3 LNP SOC

LNP queries are launched only if the LNP SOC option OSEA0008 is turned on at the switch. The LNP information is sent to the service node only if the Operator Services AIN Version 3 SOC option ENSV0020 is turned on.

If the LNP SOC option is turned on but the OSSAIN Version 3 SOC option is not, then the call will be routed using the LRN information obtained from the LNP query. However, no

LNP information is returned to the service node. Therefore, the service node will not have an indication that the LNP information is being used for routing the call.

If the OSSAIN Version 3 SOC option is turned on but the LNP SOC is not, then the service node will get an error response for the LNP Request operation indicating that LNP information is not available.

13.6 Maintenance

If the switch receives a large percentage of invalid messages (i.e. messages which cannot be completely processed by the switch) from a service node, the switch will take the service node out of service. Messaging with the service node will begin again once the service node is returned to service.

13.7 OSSAIN Centralization

OSSAIN Centralization (OSAC) gives operating companies the option to place service nodes in either a standalone configuration or a centralized configuration, independent of operator positions.

In a standalone configuration, one switch performs call distribution (ACD), voice connections, and maintenance for a given service node. This switch functions in an OSSAIN standalone mode for this service node. It does not need to communicate with other switches regarding the service node. The service node is communicating with one switch for call processing.

In a centralized configuration, one or more service nodes are centralized at an OSAC Host switch. The call distribution, voice connections, and maintenance for those service nodes are centralized at the OSAC Host switch. In this configuration, service node sessions and voice links can be shared among several OSAC remote switches. The service node may be communicating with more than one switch for call processing. However, it still communicates with only one switch for maintenance.

A switch functions as an OSAC Host for a service node when it provides the call distribution and voice connections to that service node for one or more OSAC Remote switches. Maintenance for centralized service nodes is coordinated with the OSAC Host switch only.

A switch functions as an OSAC Remote for a service node when it uses an OSAC Host switch for the call distribution and voice connections to the service node. OSAC Remote switches only have a call processing view with the service node. There is no maintenance between the OSAC Remote switch and the centralized service node except to ensure connectivity.

Service nodes can initiate calls to switches that function as both Standalone or Remote for that service node.

A given switch can perform as both a Standalone and Host for some service nodes while acting as a Remote switch for others.

The following table summarizes the aspects in a standalone and centralized topology.

Table 17. OSAC Components

	Standalone Switch	OSAC Host Switch	OSAC Remote Switch
Call Control	✓		✓
Call Distribution	✓	✓	
Node Maintenance	✓	✓	
Session Pool Maintenance	✓	✓	
Voice Channel Terminations	✓	✓	
Monitor Connectivity	✓	✓	✓

13.8 OSAC Call Scenario

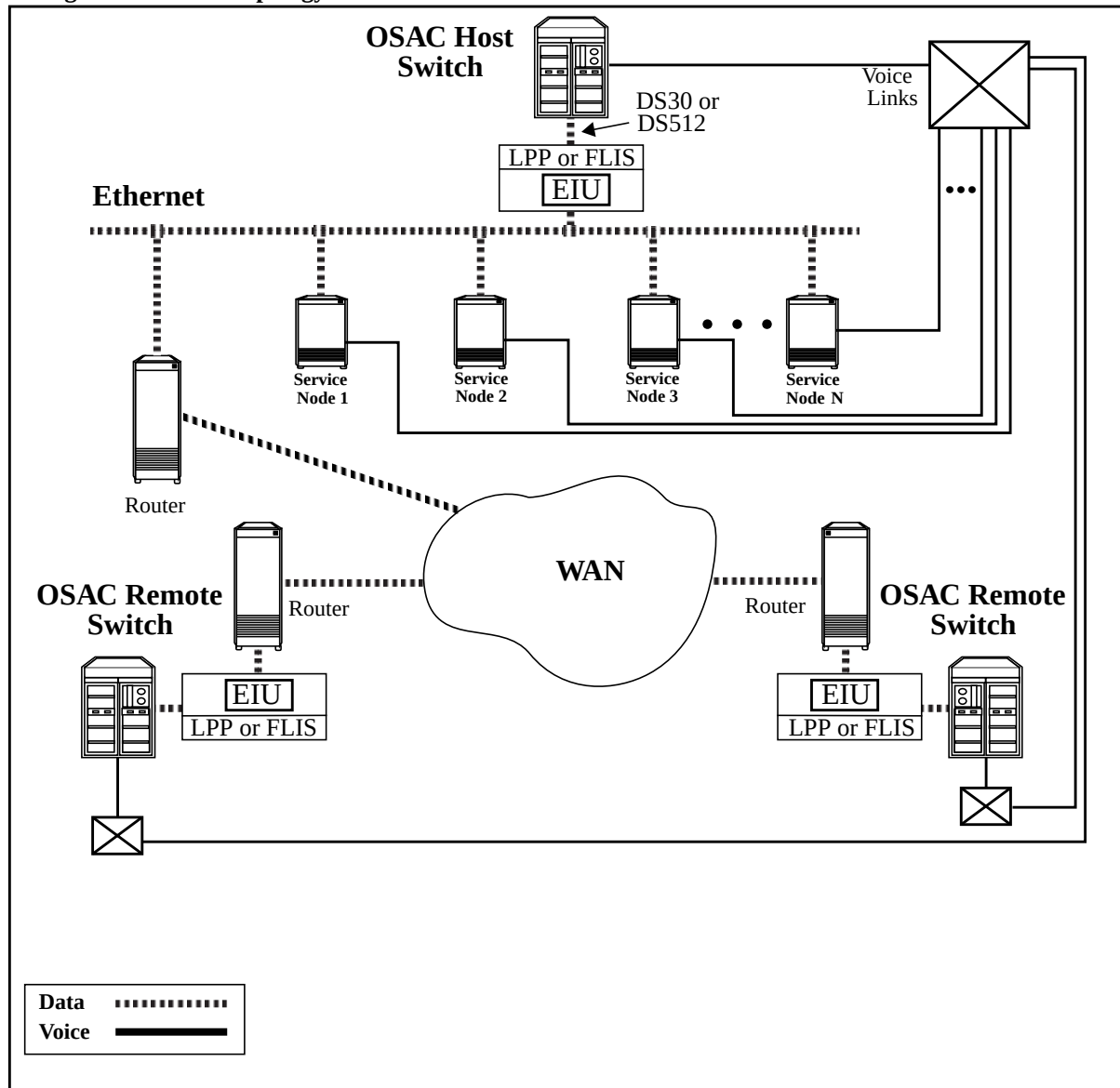
When a call comes in at an OSAC Remote requiring a function provided by a centralized service node, the Remote must query the OSAC Host for a session to the service node. After a service node session is obtained from the Host, the Remote begins the call with the service node. Call control messages are exchanged between the OSAC Remote and service node. For this call, the switch in which the call comes in is processing the call in an OSAC Remote mode for this session. The Host switch for the service node is processing the call in an OSAC Host mode. The call is processed in an OSAC environment for the service node session.

The OSAC Host also provides the voice connections to the centralized service node. When a service node requests a voice connection for the call, it sends an OAP Voice Connect Request to the OSAC Remote. The request has the logical service node voice channel. For an OSAC call, the service node voice channel is the voice channel between the service node and the OSAC Host. The logical voice channel must be datafilled in Table OAVLMAP in the Host. This table is used in the OSAC Host to map the service node voice channel to the actual voice circuits. Parallel datafill of logical voice channels are needed between the service node and OSAC Host.

The fact that the switch is acting as a Remote is transparent to the service node. From the service node's perspective, the call is just like a standalone call (except for protocol version negotiation). Service nodes are not aware that any OSAC processing is occurring. Service nodes should respond to the sending IP address included in the message from the switch.

Figure 20: "OSAC Topology" shows an example of an OSSAIN topology with centralization. For simplicity, this example shows only two Remote TOPS switches connected to the Host switch. However, there could be more Remote switches sharing the centralized service nodes. Note that service nodes are centralized at a Host switch independent of the location of operator positions.

Figure 18. OSAC Topology



OSAC functionality is provided with the Operator Services AIN Version 3 SOC option.

13.8.1 Parallel Datafill

There are no extra parallel datafill required between the OSAC Host and service node with OSAC. The service node IDs and session pool IDs still need to be datafilled at the Host. The logical voice channel IDs used with the service node must be datafilled at the OSAC Host only.

The same parallel datafill required between a Standalone switch and a service node is required between an OSAC Remote and a centralized service node, except for voice channel ID's (e.g. control list, function, language, treatment). The voice channel IDs are datafilled at the Host instead of the Remote in an OSAC environment.

13.9 OSSAIN Preprocessing

OSSAIN preprocessing sessions are used to allow calls normally denied OSSAIN processing to access a service node in a limited mode. The service node is informed of the type of session (standard or preprocessing) in the Session Begin Inform message. During preprocessing, the operations available are limited to the following:

- Append AMA Module Request
- Call Details Request
- Call End Inform
- End Call Request
- Session Begin Inform
- Text To Operator Request
- Voice Connect Request
- Voice Release Request
- Connection Status Inform
- Call Resume Request
- Service Provider ID Assignment Request

The switch will inform the SN of any changes in the connection status of all parties involved in the call. Call disposition is at the discretion of the service node. Once preprocessing is completed, the SN may request that the switch either terminate the call (if, for example, the calling party went onhook) or resume it towards its original destination (TOPS operator or automated system).

For OSSAIN standard processing, the only operation that is not allowed is the Call Resume Request.

OSSAIN Preprocessing is provided with the Operator Services AIN Version 3 SOC option.

An OSSAIN preprocessing session may occur at the following points in a TOPS call:

- On a new call origination, prior to receiving service from any TOPS automated system such as AABS, ACTS, ADAS or M CCS
- On a new call origination, prior to receiving service from a TOPS operator
- On a new call origination, upon time out at a TOPS automated system without satisfying that system's requirements
- On a new call origination, upon receiving a request for a live operator from a subscriber at a TOPS automated system
- During Automated DA Call Completion, prior to receiving service from a TOPS automated billing system such as AABS, ACTS or M CCS
- During Automated DA Call Completion, prior to receiving service from a TOPS operator

- During Automated DA Call Completion, upon time out at a TOPS automated system without satisfying billing requirements
- During Automated DA Call Completion, upon receiving a request for a live operator from a subscriber at a TOPS automated billing system

13.10 Queue Management System Management Information System

The Queue Management System Management Information System (QMS MIS) is responsible for providing event driven data about QMS sessions and call queues for use in off-board reporting facilities. The call queue and session event data is captured within OAP data blocks which are buffered prior to being sent by the DMS. The MIS buffer is sent on a periodic basis or whenever the buffer becomes full. Data connectivity to the external MIS system is achieved via Transmission Control Protocol/Internet Protocol (TCP/IP) over an Ethernet Interface Unit (EIU). The external MIS system uses the data from these messages to calculate its own reports. The external MIS vendor decides what statistics to report as well as the format of the reports.

13.10.1 General Message Description

The following is a general description of the events captured by the DMS MIS data blocks showing when they occur. There are two basic types of events reported: queue events and session events. Session events can be further divided between session events generated as a result of call activities and those generated as a result of maintenance activities. A description of each follows:

- Queue Event - occurs when a call accesses the OSSAIN queueing system. This event provides information about the queueing system such as the number of calls in queue, time in queue, and whether the call was deflected or overflowed from queue.
- Session Event - this event is reported for calls that have obtained a session. It contains numerous information about the call as well as a timestamp that can be used to calculate the duration of the call at the session.
- Non Call Session Event - this event is reported for sessions performing various state transitions between being idle, busy, or throttled. It contains a timestamp that can be used to calculate the time a session spends in a particular state.

In addition to the events described above the MIS message buffer begins with a single Information data block that contains basic information from the switch with respect to the MIS application, the time and date the buffer was started, a timestamp, and the restart severity.

13.10.2 Pegging Strategy

The MIS events reported provide the information needed by the MIS vendor to peg queue and/or session event data. Queue events are generated at the point the call accesses the queueing system. Session events are generated at the points in time a session is obtained or released from a call and when sessions are made available or unavailable due to maintenance activities. The MIS vendor is told what to peg (if anything) and what the new session state is.

13.10.2.1 Work Volume and Idle Time Pegging

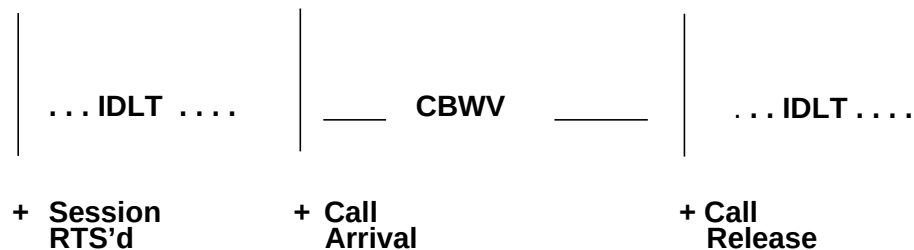
The session event data blocks contain a timestamp that can be used to calculate the duration that a session was in a particular state. Idle time and work volume pegs are reported in both call and non call session events. They are described as follows:

- Call Busy Work Volume (CBWV) - the accumulation of time from call arrival to a session to call release from the session.
- Non Call Work Volume (NCWV) - the amount of time a session is in service but not available to handle calls. Pegging of this time occurs for sessions that are drained or throttled.
- Idle Time (IDLT) - the amount of time a session is in service and available to handle calls.

In order for the DMS to provide timestamps for session events, the events are reported after the event has been completed. For example, CBWV is pegged at the point in time the call is being released from a session. IDLT is pegged at the point in time an idle session is being assigned to a call. The duration of a particular event is calculated by subtracting the previous timestamp recorded for that session from the current timestamp being reported for that session. The following diagrams illustrate the pegging of IDLT, CBWV and NCWV.

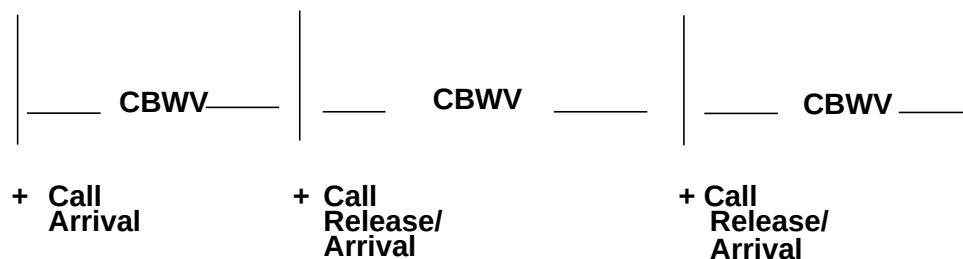
Call Arrival - to an idle session

A request is sent to peg IDLT at Call Arrival. When the call is released CBWV stops accumulating against the call and a request is sent to peg CBWV.



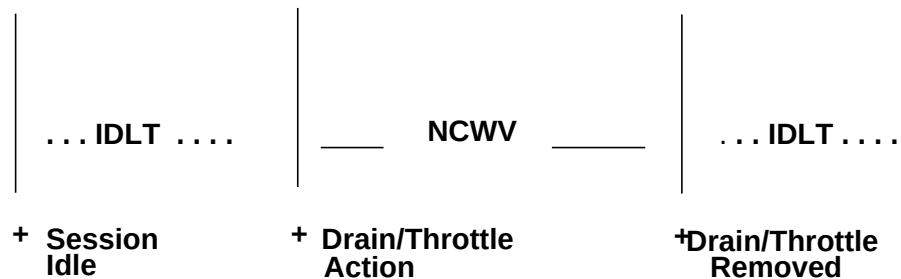
Continuous Call Arrival

This occurs when multiple calls are queued for a session. As each call is released CBWV is pegged for that call. The session pegs no IDLT.



Draining/Throttling a Session Pool

NCWV is accumulated when a session pool is drained or throttled and the session can no longer accept calls. When the drain or throttle action is removed NCWV is pegged.



It should be noted that peg events are only reported on in service sessions. Therefore, time duration is not pegged for sessions while in the busy or offline state. When a session is brought in service a non call session event is generated with no indication to peg CBWV, NCWV, or IDLT. However, the timestamp sent in this non call session event should be recorded to serve as a basis from which to calculate a time duration for the next session event.

13.10.2.2 Time of Day and Tenths of Seconds Pegging

Timestamps sent with queue event and session event messages are measured in tenths of second units. The timestamps are taken from the DMS CPU clock. This is because, unlike time of day, CPU clock ticks remain unchanged by restarts and clock adjustments (e.g. daylight savings adjustments). CPU clock ticks provide a reliable means of accurately measuring time accumulations irrespective of restarts and clock adjustments.

The time of day may be checked against CPU clock timestamps each time an Information data block is sent. The Information DB contains the time of day, year, as well as a current timestamp. Adjustments in time of day can be detected from the Information DB by determining if the time delta of the time of day is equal to the time delta of the CPU clock ticks. If the two values are not equal, a time of day clock adjustment has occurred.

The tenths of seconds timestamp is reset to 0 after reaching its maximum possible value, 2 to the power of 16, minus 1. This must be considered when pegging time accumulation. For instance, if the MIS pegs a stretch of idle time that occurred between 65467 and 324, the amount of time passed is 39.3 seconds.

13.10.2.3 Suggested Algorithm for MIS Pegging

The processing of the MIS message stream requires a particular approach be taken by the MIS node in order to peg events appropriately. The algorithm which follows is a general one which describes the recommended approach for pegging events and time intervals. This algorithm requires data structures for the OSSAIN application to record timestamps and sequence numbers for each session as well as each call queue.

```

CASE data_block_id
IN
{ossain_call_queue_event_db}:
    %% Note: we won't recognize the failure case
    %% where we miss a multiple of 256 call queue
    %% event messages for a particular call queue...
    if (cq_seq_num[datablock.call_queue] + 1) MOD 256
        = datablock.sequence_number
    then
        %% All is well - the datablock is in sequence.
    else
        %% One or more call queue event messages for this
        %% call queue have been lost (specifically, at
        %% least ((msg.sequence_number + 256) -
        %% cq_seq_num[msg.call_queue]) MOD 256 were
        %% lost). Average answer stats may be impacted,
        %% and call queue length counts may be off.
        <log the failure>
    endif;
    %% Process the call queue event.
CASE msg.queuing_event
IN
{call_placed_in_queue}:
    %% no pegging required for this case.
{call_abandoned, call_presented_to_agent,
 call_deflected, call_overflow}:
    %% for the queue specified in datablock.call_queue:
    <peg the counter for the type of call queue event>
    <peg the register accumulating the time spent
    in queue>
endcase;
%% for the queue specified in datablock.call_queue:
<update the calls in queue counter from
datablock.current_call_queue_size>
<update the calls waiting indicator>
%% Update the stored sequence number from
datablock.sequence_number

{ossain_non_call_session_event,
ossain_session_event}:
    %% Note we won't recognize the failure case
    %% where we miss a multiple of 256 position
    %% event messages for a particular position...
    if (session_seq_num[datablock.session_number]+1)MOD 256
        = datablock.sequence_number;
    then
        %% All is well - the message is in sequence.
        %% Process the session event:
        if datablock.reset_time
        then
            if <any pegs indicated>
            then
                %% calculate the interval and peg as
                %% indicated in the datablock peg fields.

```

```
        dcl delta_time timestamp;
        datablock.timestamp-stored_time[session_number]
            -> delta_time;
        <peg any indicated time-based registers
        (CBWV, NCWV, IDLT, SWV) by delta_time
        for the specified session, call type,
        service, etc.>
        <peg any indicated non-time-based
        registers (ISS, SNOS, TGSS, TXSS, SI)
        by a simple increment-by-one>
    endif;
    %% reset the timestamp.
    datablock.time ->
        stored_time[datablock.session_number];
else
    %% if the reset time flag was not set, then
    %% the message was only to inform the MIS of
    %% an MIS state change, and not to peg anything.
endif;
<update state counters as required using
MIS Session State, Session ID, and Session>
else
    %% One or more session event messages for this
    %% session have been lost (specifically, at
    %% least ((msg.sequence_number + 256) -
    %% session_seq_num[datablock.session_number]) MOD 256
    %% were lost). The data for the interval from
    %% stored_time[datablock.session_number] to
    %% datablock.time has been lost for this position.
    %% Log the failure, and reset the time. (Don't
    %% peg anything - the interval is meaningless)
    <log the failure>
    datablock.time -> stored_time[dat-
        ablock.session_number];
endif;
%% Update the stored sequence number for this session.
datablock.sequence_number ->
    session_seq_num[datablock.session_number];
endcase; %% data_block_id
```

13.10.3 MIS and the OSAC Environment

In an OSAC environment, the OSAC HOST switch supports data connectivity to the off-board MIS system. Therefore, when a call originates in an OSAC Remote switch, the call information required by the MIS system is sent to the OSAC HOST switch for inclusion in the MIS OAP data blocks. In the OSAC environment, the Host to Remote messaging is performed via the OSAC protocol.

13.10.4 MIS message buffer

The OSSAIN MIS application in the DMS supports a message buffer which is filled with the Queue Event, Session Event and Non Call Session Event DBs as they are generated. The MIS datablocks are stored in a 4k buffer internal to the DMS. The buffered datablocks are sent to the MIS when the buffer is filled or when a buffer time-out period expires.

The first datablock in each buffer is the Information ID datablock. The Information ID DB is inserted immediately when the DMS decides to begin filling a buffer. The time and date given in the Information ID DB are the time and date at the time that the Information ID DB is inserted into the buffer.

When the DMS is ready to send the MIS buffer it is encapsulated within a TCP message header. The 4K MIS buffer consists of the MIS Class Header, MIS Inform Operation and the MIS datablocks. Refer to Table 18, “Example MIS Buffer with TCP Header,” .

Table 18. Example MIS Buffer with TCP Header

TCP Header
MIS Class Header
MIS Inform Operation
Information ID DB
OSSAIN Call Queue Event DB [Call 1 - queued]
OSSAIN Call Queue Event DB [Call 2 - queued]
OSSAIN Call Queue Event DB [Call 1 - got agent]
OSSAIN Non-Call Session Event DB [Call 1 - peg IDLT]
OSSAIN Call Queue Event DB [Call 3 - deflected]
OSSAIN Session Event DB [Call 1 - peg CBWV]
.
.
.

A 4K byte message buffer is always sent to the MIS node even though it may only be partially filled. The length field, in the MIS Class Header, must be used by the MIS node to accurately parse the MIS message buffer.

13.10.5 MIS data block generation

The following figures provide call flows illustrating when the Queue Event, Session Event and Non Call Session Event DBs are generated and placed into the MIS buffer. Most OSSAIN call scenarios generate three OAP MIS data blocks per session. Figure 19: “OSSAIN Call gets session no queuing” illustrates an OSSAIN call that immediately obtains a session without queueing. Note that one OSSAIN Call Queue Event data block, one OSSAIN Non-Call Session Event data blocks and one OSSAIN Session Event data block are generated.

Figure 19. OSSAIN Call gets session no queuing

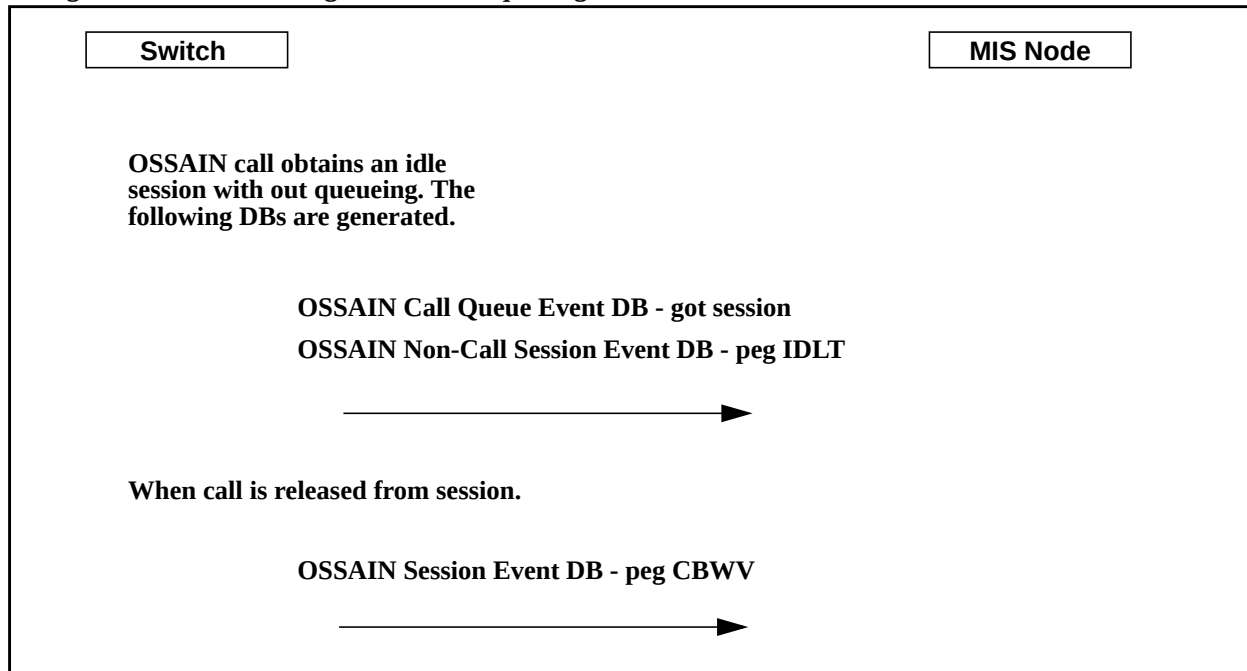
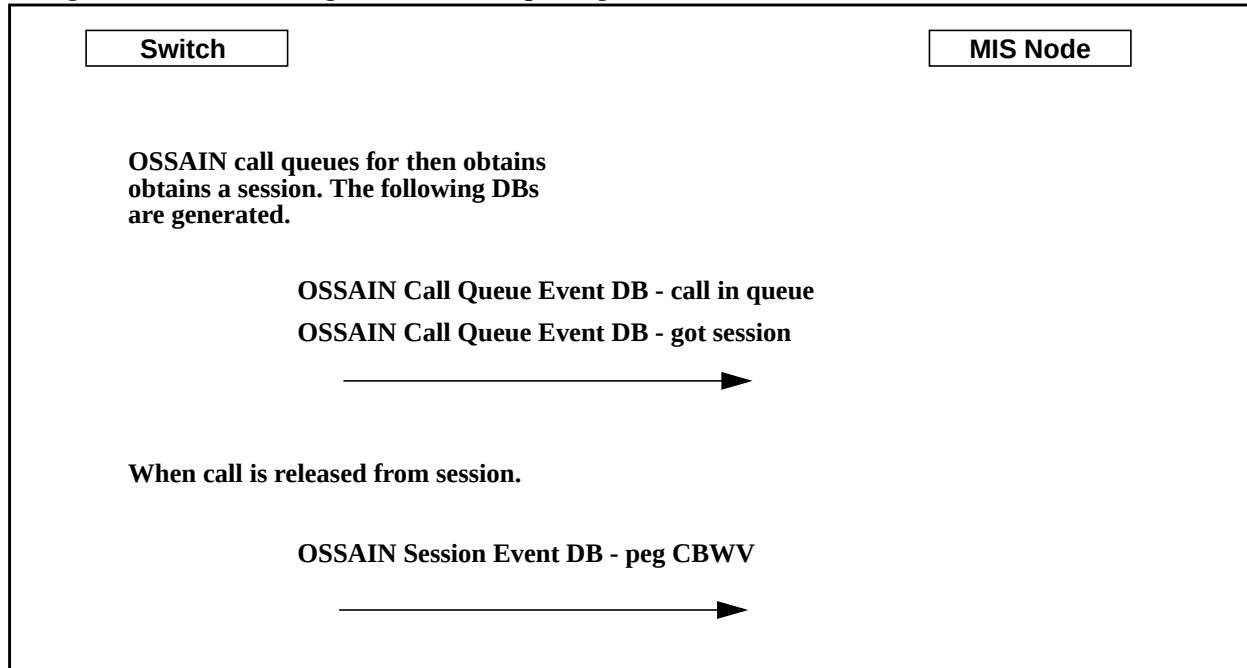


Figure 20: “OSSAIN Call gets session with queuing” illustrates an OSSAIN call that is queued for a session before obtaining one. In this scenario two OSSAIN Call Queue Event DBs are generated, one when the call is queued and one when the session is obtained. One OSSAIN Session Event is generated when the call is released from the session. Note there is no OSSAIN Non-Call Session Event to peg IDLT for this scenario.

Figure 20. OSSAIN Call gets session with queuing

The remaining figures 20-22 illustrate the generation of individual MIS events by the DMS. These can be generated by an assortment of call scenarios and/or maintenance activities on the switch. These figures also illustrate the pegging actions to be taken by the MIS Node.

Figure 21. Example MIS data block flows.

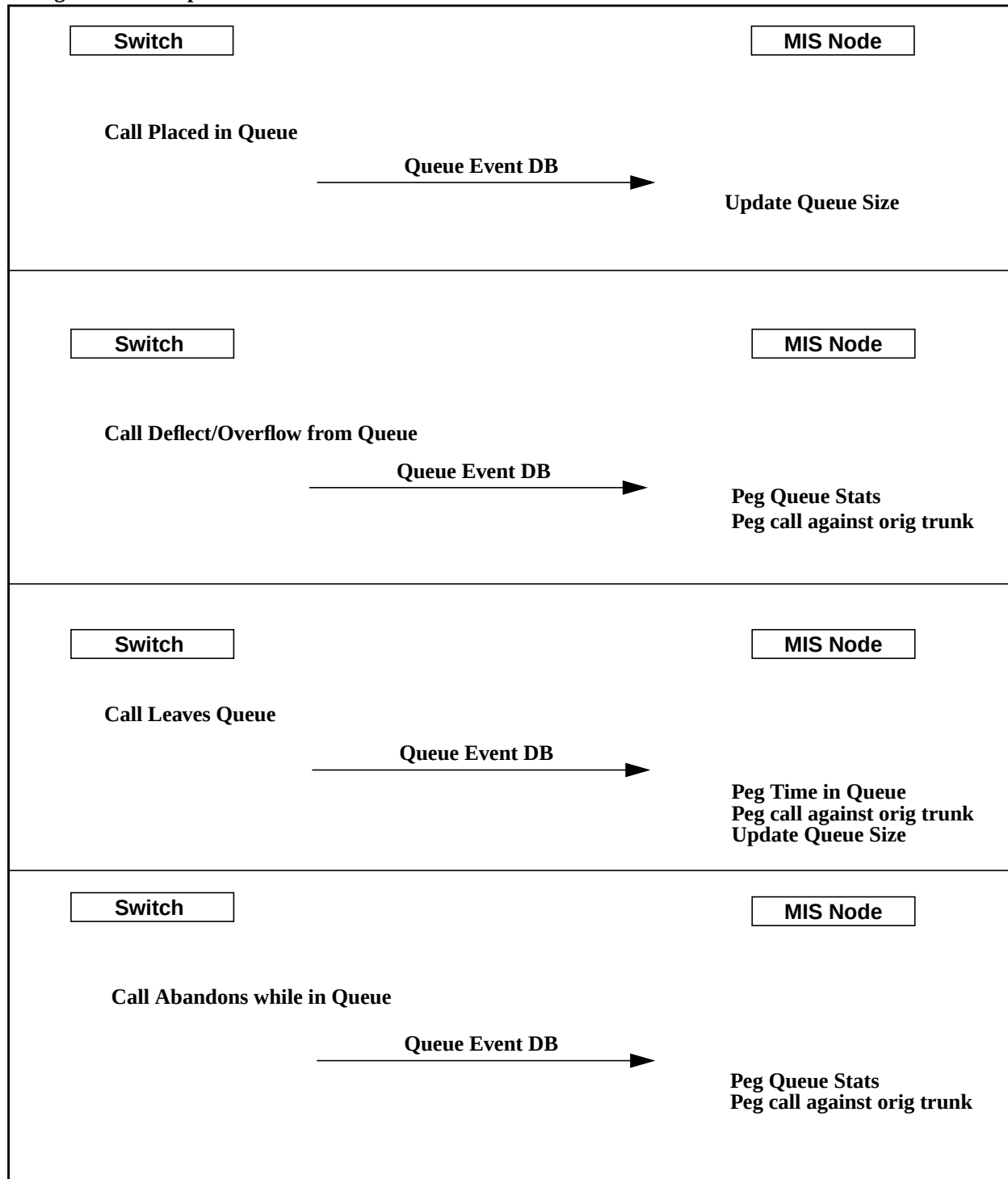


Figure 22. Example MIS message flows continued.

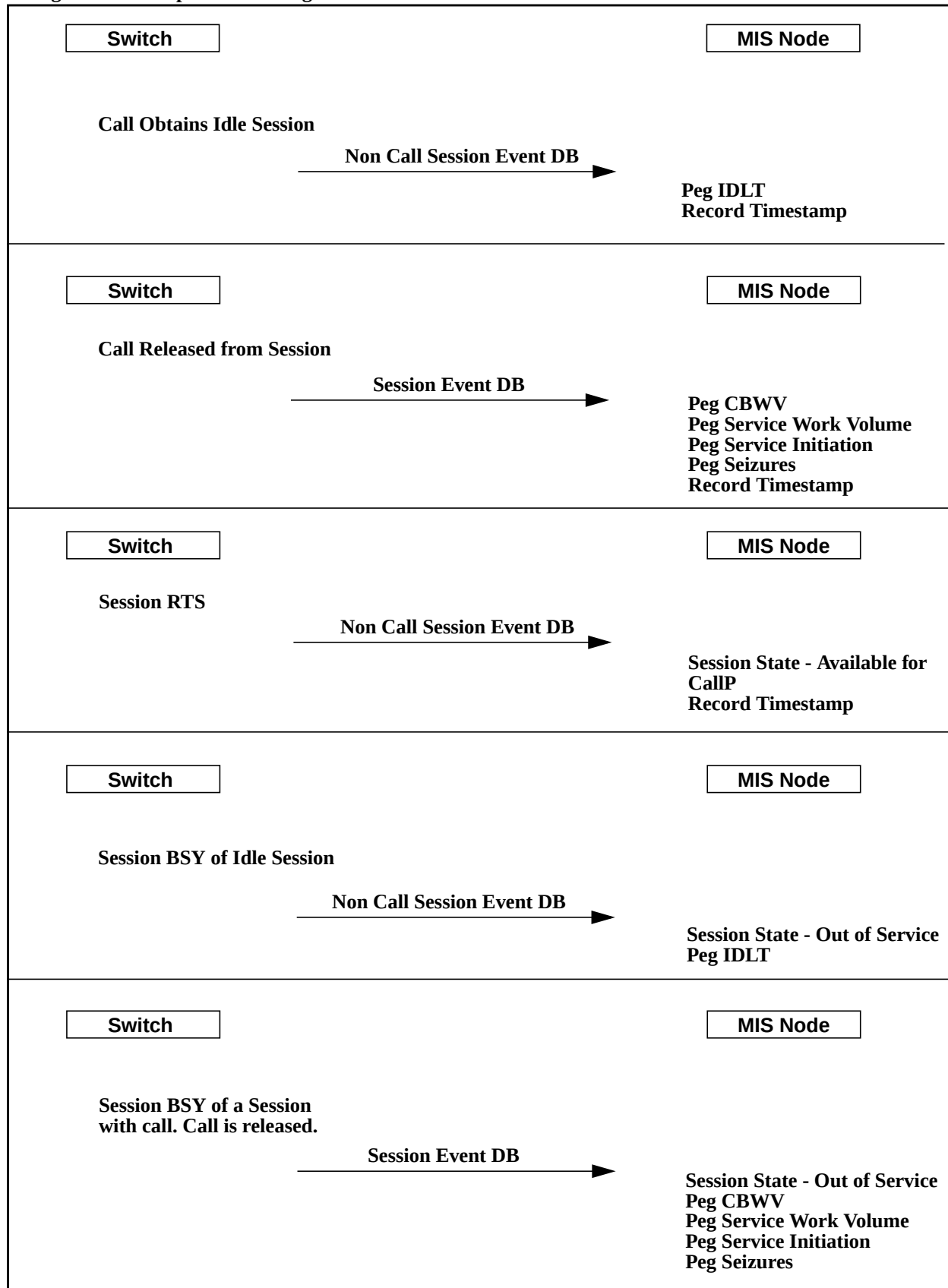
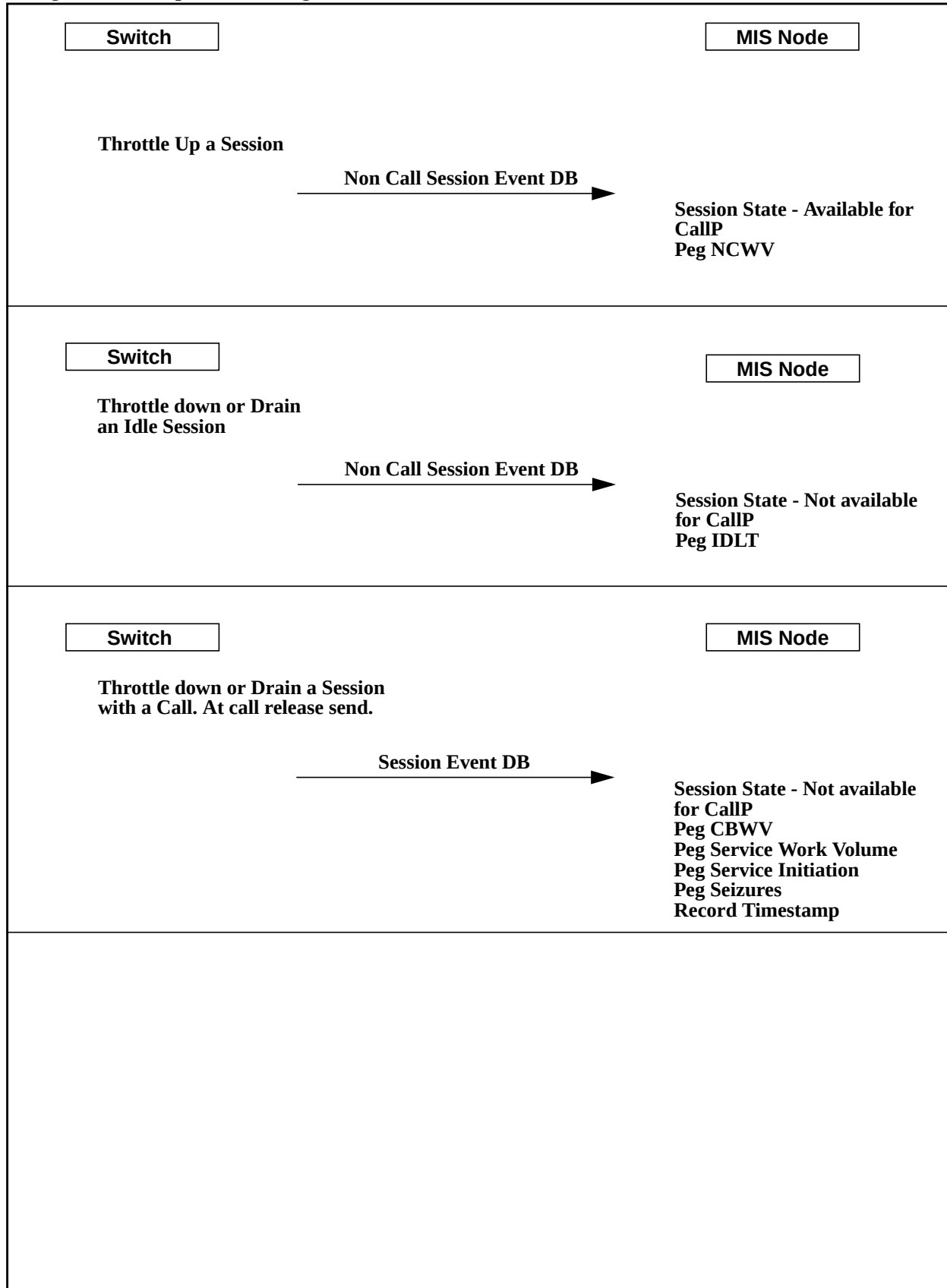


Figure 23. Example MIS message flows continued.



13.11 Release Link Trunking

Release Link Trunking (RLT) minimizes the usage of Signalling System 7 (SS7) ISUP trunks by releasing the ISUP connections between the TOPS switch and the previous switch in the network for a given call. After the RLT messaging between two switches is complete, the ISUP connections are released. This frees the ISUP trunks for additional traffic and reduces the number of ports being used. Without RLT, ISUP connections must be maintained between the TOPS switch and the previous switch for the duration of the call.

RLT between the TOPS switch and the DMS250 switch has been deployed in two phases. RLT Phase I was released in TOP07, and RLT Phase II was released in TOP09. Separate SOC options control the two pieces of functionality.

RLT Phase I applied to a limited set of call types.

- collect billing
- third number billing
- station paid billing

RLT Phase II applied to a broader set of call types.

- all the call types supported in Phase I
- calling card billing

For calls to be RLT eligible, there must be proper RLT trunk datafill with the variant and version of RLT. For a bridge call, the trunk datafill (variant and version) must be the same for the incoming and outgoing trunk.

An understanding of RLT is important to the OSSAIN developer, because calls are released back to the previous switch with RLT. This has an impact on post float processing. Specifically, a call that has been RLT'd but triggered back to an SN will appear (e.g. the CALLID will change) different when compared to a call that has not RLT'd but triggered back. The difference is in the data sent to the SN by the TOPS switch. Special attention must be given with respect to assumptions about a call's data when using RLT and triggers. The reason is that a call that has NOT been RLT'd but triggers will have been maintained as a single entity within the TOPS switch. On the other hand, a call that has been RLT'd but triggers requires two entities within the TOPS switch. This is due to the fact that the re-originated call from the previous office looks like a new call to the TOPS switch.

Note that in RLT Phase I, no post float processing is supported. RLT Phase II introduces the ability to detect a sequence call request by the subscriber (e.g. keying #), which enables calling card calls to be supported. Specifically, TOPS can RLT a call even if one of the following triggers is in effect:

- #: Note that switch datafill must indicate that the forward party is released upon detection of the #. If the release forward indicator is not set, TOPS will not RLT the call.
- called disconnect

If post float processing is required on a call (other than the triggers listed above), then this call is not a candidate for RLT. This is because the DMS250 controls RLT'd calls, and there is no mechanism for the DMS250 to detect or inform the TOPS switch about the trigger. As an example, debit card calls are not candidates for RLT, because of the need for notification when the time period expires.

While RLT Phase II supports a small set of triggers, the OSSAIN developer must use care in designing an application for these calls. When the call is returned from the DMS250 to the TOPS switch, the call will be a reoriginated call as stated above. A reoriginated call contains the same data as the original call just prior to it being RLT'd, with a few key exceptions.

The following information will always be changed on the reorigination of a sequenced RLT call.

- callid

The following information will always remain unchanged on the reorigination of a sequenced RLT call.

- class charge
- account code
- custom pin length
- billing number
- billing number type
- billing number digit count
- sequence number (# of sequence calls)
- trigger profile index
- whether the call has been previously branded (Y/N)
- language
- context block

For each application, the OSSAIN developer must decide if and for which calls to activate RLT. A saving of trunking facilities is a clear benefit to an Operating Company that deploys an OSSAIN application which uses RLT. For this reason, developers should make an effort to use this functionality when feasible. The service node requests RLT by including an RLT Request data block in the Call Float Request or the Transfer to Carrier Request.

The preconditions for RLT are listed below:

- Both the incoming trunk and the outgoing trunk terminate to the same end office.

- Both the incoming trunk and the outgoing trunk support RLT
- Both the incoming trunk and the outgoing trunk support the same variant of RLT.

13.11.1 RLT SOC

RLT Phase I functionality at the switch is provided by the RLT I SOC option OSEA0007. The RLT datablock will be accepted by the switch only if the Operator Services AIN Version 3 SOC option OSAN0003 is turned on.

If the RLT I SOC option is turned on but the OSSAIN Version 3 SOC option is not, then RLT will not be used for the call.

If the OSSAIN Version 3 SOC option is turned on but the RLT I SOC is not, then the RLT eligibility checks will fail and RLT will not be used for the call.

RLT Phase II functionality at the switch is provided by the RLT II SOC option OSEA0009.

13.12 Simultaneous Interactions

OSSAIN provides a framework in which at most two function providers (i.e. a TOPS operator and a service node) can interact simultaneously in a call to provide a function. However, only one function provider is the active function provider (i.e. in control of the call). The other, passive function provider has minimal capabilities. Control of the call can pass back and forth between the function providers without the switch ever taking control.

Note that only a service node and a TOPS TA operator can interact simultaneously. An operator cannot be connected to a call in a simultaneous interaction for a session type of directory assistance. An existing TOPS automated system cannot be connected to a call in a simultaneous interaction.

Simultaneous Interaction functionality is provided with the Operator Services AIN Version 3 SOC option.

13.12.1 Active Function Provider

The active function provider has control of the call. It has the full set of OAP capabilities.

For simultaneous interactions, the active function provider requests the switch to bring another function provider (i.e. a TOPS TA operator or another service node) into the call. The switch responds in one of the following ways:

- the function provider is available and connected to the call
- the call is queued for a function provider
- a function provider is not available (i.e. no switch resources are available to obtain another function provider).

If a second function provider is available, it is connected to the call in a passive mode. The active function provider can pass control to the passive function provider. The passive function provider then becomes the active provider and the active provider becomes passive.

13.12.2 Passive Function Provider

The second function provider connected to the call is the passive function provider. When it is first connected, it will receive call origination information. However, after the initial update, it will not receive further call information (i.e. unsolicited events and call updates by the active function provider), with the exception of calling/called party hook status changes. The passive function provider can be updated using one of the following means:

- the two function providers communicate via a non-switch connection
- the active function provider can pass data through the switch to the passive function provider via the OAP Pass Through Request operation

- the active function provider can also pass control of the call to the passive function provider. Once the passive provider has control of the call, it will receive the current call details (i.e. the same information sent in the response to the Call Details Request Operation).

The passive function provider has a restricted set of OAP capabilities. The capabilities allowed are listed below:

- release itself from the call
- pass data through the switch to the active function provider
- request/release a voice link for itself

Any other actions requested by the passive function provider is considered to be an error by the switch. If the passive function provider is a service node, the switch will respond with an OAP error message.

13.12.3 Non-Switch Connection (NSC)

A non-switch connection is a connection between the two function providers that does not involve messaging through the switch. This type of connection should be used when large volumes of data need to be passed between the two function providers involved in the call.

Transferring large amounts of data would be too slow through the switch. Therefore, the amount of information allowed via the OAP Pass Through Request operation has a limited data size. Refer to the description of the operation for the defined size.

Development work may be required at the operator position to allow for a non-switch connection with a service node.

13.12.4 Simultaneous Interaction Between TOPS TA Operator and A Service Node

As stated earlier, a service node and a TOPS TA operator can be connected to a call simultaneously. Note that the protocol used between the operator position and the switch is either OPP (for OPP compliant position types) or ASCII (for TOPS MP positions).

When an operator is connected to a call in the passive mode, it will have a restricted set of capabilities. The service node connected to the call is the active function provider. The operator connected in the passive mode will be able to do one of the following:

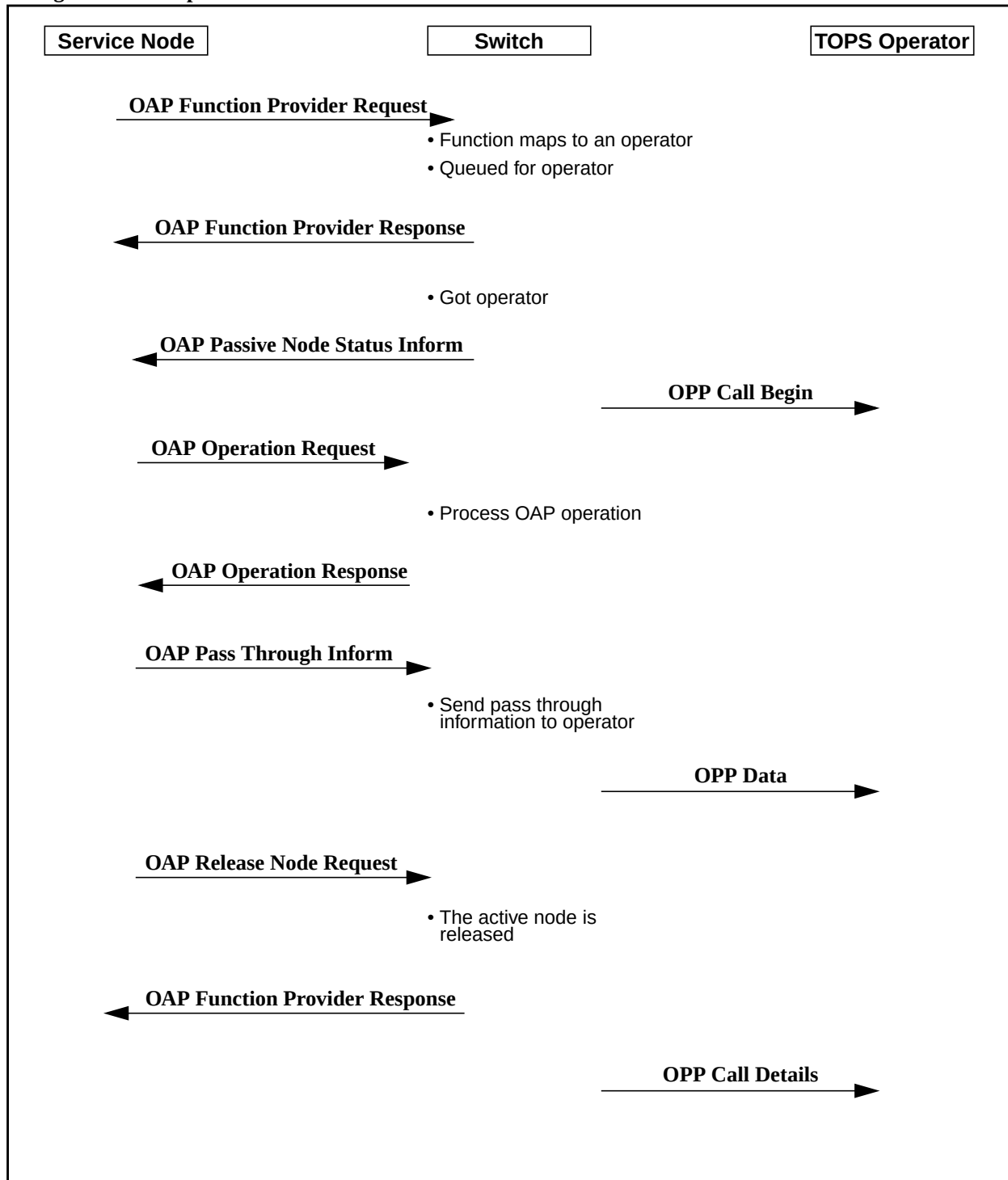
- Release itself from the call (OPP/ASCII message from operator to switch), which causes a message to be sent to the service node indicating that the passive function provider has been released from the call (OAP message from switch to service node).
- Send and receive pass through information through the switch with the service node (OPP to/from operator; OAP to/from service node).
- Send and receive information through the NSC connection with a service node.

- Release the active node from the call (OPP/ASCII message from operator to switch)

All other OPP action requests from the operator will be blocked by the switch. However, an operator connected in the passive mode will still be able to hear and speak to the subscriber and listen to what the service node is providing.

Figure 24: “Example call flow for simultaneous interaction” illustrates how a TOPS operator and service node interact simultaneously in a call.

Figure 24. Example call flow for simultaneous interaction



13.13 Voice Connections

As of OAP version 4.0 there are two main categories of voice links:

- two-way, non-broadcast voice links¹, referred to as **standard** voice links.
- one-way, **broadcast** voice links.

In addition, as of OAP 6.0 standard voice link operation can be further categorized by their method of selection:

- **SN requested/SN selected** - explicit Voice Connect Request sent by SN, selection of voice circuit by SN from its own pool of voice links.

Each service node that is selecting standard voice links is associated in switch datafill with its own set of voice channels. This set of voice channels is considered as a resource that is 'owned' by a particular Service Node, at least as far as the switch is concerned². For standard SN-selected voice links, an SN is only allowed to use its own voice link resources.

When a service node requests a voice connection for a particular session, it must specify a logical voice channel number. The node ID of the SN, along with the logical voice channel number, are mapped to an actual voice circuit at the switch using table OAVLMAP. The switch then connects the voice circuit to the call and sends back a success response.

- **SN requested/switch selected** - explicit Voice Connect Request sent by SN, selection of voice circuit by switch from a group of voice links associated with the SN and function.

When the SN sends in a request for a voice link but does not specify a logical channel number, the switch will select the voice link. The voice circuit is selected from a trunk group associated with a particular node and function in table SNVLGRP. The switch selects an available member, connects it to the call and informs the SN of its selection in the Voice Connect Success Response.

- **Switch auto voice link selection** - automatic voice link selection and connection of voice circuit by the switch at the start of the session.

In this case, the call arrives at the SN with a voice link already attached. Functions can be datafilled in the switch to automatically select and connect a voice link to the call prior to starting the session with the SN. The voice circuit is selected from a trunk group associated with the node and function in table SNVLGRP. The switch selects an available member, connects it to the call and informs the SN of its selection in the first message of the session.

Broadcast voice links are limited to the following method of selection (OAP 4.0 and above):

-
1. All voice links prior to version 4.0 were of the two-way, non-broadcast type.
 2. Voice link resource owning is only a concept that the switch uses to map a logical voice channel to a voice link circuit. The DMS makes no assumptions about how voice circuits are terminated in the network. Voice circuits (either standard or broadcast) can be connected to the service node itself, to another service node, or to any other external system.

- **SN requested/SN selected** - explicit Voice Connect Request sent by SN, selection of voice circuit by SN from its *own* pool of voice links or from *another* SN's pool of voice links.

For broadcast connections, an SN is allowed to request the voice channel resources of another Service Node. Whenever a Service Node requests a broadcast voice connection for a particular call, it must specify a node ID (indicating the voice link resource owner) as well as the logical voice channel number. This is true whether it is using its own or another SN's voice links. By allowing Service Nodes to share broadcast voice link resources, common recordings such as those that play music or provide a branding announcement can be used by several different nodes.

For broadcast voice links, the received node ID (voice link resource owner) and logical voice channel are mapped to the actual voice circuit in table OAVLMAP. The switch then connects the voice circuit to the call and sends back a success response.

13.13.1 Switch/Service Node parallel functionality/datafill

It is important that the switch and the Service Node agree on a voice link's functionality with regards to the following:

- *the type of voice link* (broadcast or standard). Due to switch network configuration constraints, individual voice links can function as only one type or the other. In the switch this is determined by the BCST field in table OAVLMAP.
- *the method of selection* (SN or switch). SN selected voice links must be datafilled in table OAVLMAP, while switch-selected voice links must have the trunk group associated with the service node and function defined in table SNVLGRP.
- *the point at which the voice path is cut through on broadcast voice links*. The switch can allow speech transmission from the voice link to be cut through immediately (controlled by setting the CUTTHRU field in table OAVLMAP to IMMEDIATE) or it can wait for a signal from the voice trunk (a momentary offhook) before cutting through speech (controlled by setting the CUTTHRU field to HKCHG).
- *the maximum number of connections that can be made to a single broadcast voice link*. Again, because of hardware limitations in the switch network, the total amount of broadcasting connections must be factored into the engineering of the switch. For this reason the number of broadcast connections is stored on a per voice link basis in field MAXCONNS in table OAVLMAP. This value can range up to 1023 for trunks using the IMMEDIATE method. For trunks using HKCHG, the maximum allowable is 255.
- *who controls the release of a broadcast voice link*. Two options exist:
 - the Service Node can request the release
 - the switch can release the voice link independently of the Service Node.

The option used is dependent on information received in the OAP Voice Connect Request and can change from call to call for an individual broadcast voice link.

These items are discussed in more detail in the following sections.

13.13.2 Standard voice link operation

When voice processing is required to collect information from a party involved in a call at a Service Node, the call is connected to an individual SN voice link. In this situation, a unique, two-way Network connection³ is necessary in order for the SN to obtain any voice or DTMF input from the call participants. Standard voice links are used for this purpose.

For SN-selected voice links, the switch will set up a two-way connection when mapping of the logical channel and node ID result in a voice link that has the BCST filed set to N. For switch-selected voice links, the switch will setup a two-way connection after selecting an idle voice circuit from a group of circuits associated with the node and function.

Release of a standard voice link during a session is under the control of the Service Node. An exception to this rule is if the switch detects a problem on the circuit and has to release it. In that case, the SN will receive a Voice Release Inform message.

13.13.2.1 Signalling on standard voice links

Information signalled on standard voice links is limited to the 'Available/Not Available' status of the voice circuit. This information is transmitted as either an onhook (available) or an offhook (not available). If either side goes offhook the voice link is no longer available for use by call processing. If the offhook is from the SN the switch puts the voice link into the RMB (Remote Make Bsy) state. It remains RMB until an onhook is detected from the SN. Likewise, a voice link that is offhook in the switch must remain unused by the SN until the switch sends an onhook. If the SN attempts to select a trunk in a busy state, a switch log will be generated and the failure will be indicated to the Service Node.

NOTE: An offhook of longer than 40 millisecond duration from the SN while a voice link is involved in a call results in the call being taken down as well as the voice link being put into the RMB state.

13.13.3 Broadcast voice link operation

In some cases it may only be necessary for the SN to play a recording that is not unique to that particular call, such as a branding announcement or music, without collecting any information from any party. While this type of call could be handled using a standard two-way, non-broadcast voice link, the fact that the direction of the information is one-way only (transmitted from the voice link to the call participants) and is somewhat generic lends itself to being more of a broadcasted type of transmission. In such a situation, voice link broadcasting may be used to connect not just an individual call, but several calls simultaneously to the same voice link. The switch network connection is a one-way broadcast transmission from the SN voice link to multiple calls. This results in a significant savings in voice link facilities between the switch and the Service Node.

3. The exact speech path status (one-way, two-way, zero-way) of each port in an OSSAIN call is controlled by the Service Node.

Figure 25: “Voice Link Broadcasting to 5 separate calling parties” shows an example of how the network connections might look in the above scenario. In this diagram, a single Service Node Voice Link (SNVL) is connected to 5 different calling parties simultaneously. Each party has a one-way network connection to the SNVL.

Figure 25. Voice Link Broadcasting to 5 separate calling parties

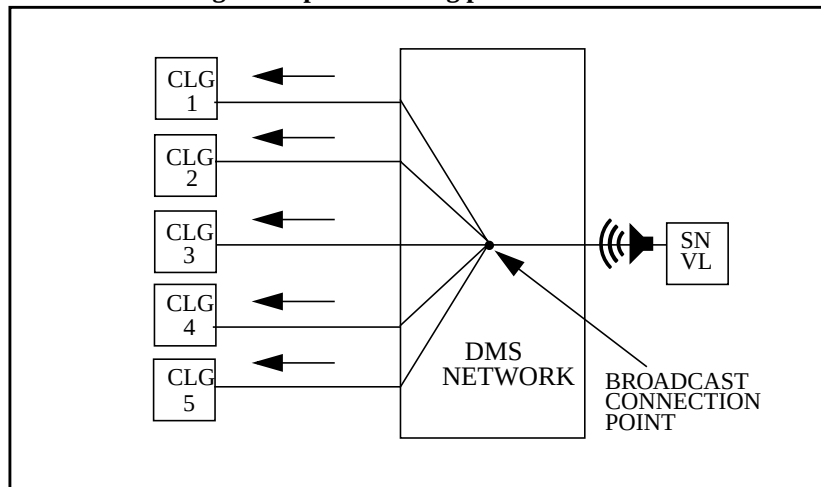
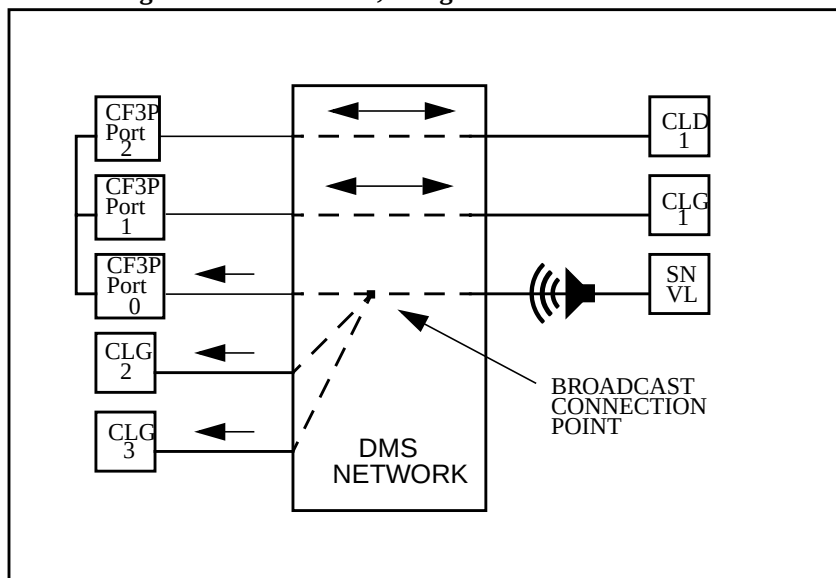


Figure 26: “VL broadcasting with CF3P in 1 call, along with 2 other calls” shows 3 calls connected to a single broadcasting voice link, including a call using a 3-port conference circuit (CF3P). The arrows indicate the direction of each of the Network connections.

Figure 26. VL broadcasting with CF3P in 1 call, along with 2 other calls



Broadcast voice links are selected by the SN but their release could be under the control of either the switch or the SN. This is determined when the Voice Connect Request is received and is based on the Broadcast Cycle Count field in the Broadcast DB. Note that the Broadcast DB is only received when the Voice Connect Request is for a broadcast voice channel.

When a Service Node requests a connection to a broadcast voice link, the DMS maps the given voice channel (from the Voice Channel DB) and node ID (from the Broadcast DB) to a physical trunk group and member number using table OAVLMAP. If the voice channel's BCST field is set to Y and connection to the trunk does not exceed the maximum number of simultaneous connections allowed for the voice link (field MAXCONS), a one-way broadcast network connection is made.

If a non-zero number is received in the Broadcast Cycle Count field of the Broadcast DB, the switch will keep track of the number of cycles played and release the voice link once that count is reached. The SN is informed of the release by a Voice Release Inform message.

If zero is received for the count, the switch will not count cycles and it is up to the Service Node to release the voice link with a Voice Release Request message.

While the DMS Network connection is made when the voice link is selected, the enabling of the transmission of Pulse-Code Modulated (PCM) speech samples from the broadcasting voice link occurs in one of two possible modes:

- immediate PCM cut-through when the connection is made. This is similar to the way a standard voice link operates.
- PCM cut-through disabled until the DMS detects an offhook of a least 10 milliseconds on the voice link.

The mode used is determined by datafill of field CUTTHRU in table OAVLMAP. The rationale for providing two modes of connecting is to allow for different types of broadcasting announcements. In some cases, such as for playback of music, it is unimportant as to when the call participants actually begin hearing the recording. For announcements of this type, immediate voice cut-through may be specified by setting field CUTTHRU to IMMED.

On the other hand, for a branding announcement it is usually important that the recording not be heard unless it is at the beginning of the recording. For such recordings, the voice transmission path should not be cut thru until a change of state on the circuit is received, indicating the beginning of the recording. This functionality is obtained by setting the CUTTHRU field to HKCHG.

For broadcast voice links the period of time that the PCM stream, as well as the voice link itself, remains connected is determined at the time of the request to connect a voice link. The SN can specify one of the following:

- the DMS is to disable the PCM stream and remove the voice link from the call after a certain number of cycles⁴ have been played, without waiting for instruction from the SN. In this case, the DMS will inform the SN that the specified number of cycles has been played and that the voice link has been removed from the call.

4. A cycle is considered to be 1 complete playing of a repetitive recording - see later sections for a discussion of cycles.

- An explicit request from the SN is required to remove the voice link from the call. The DMS will not monitor the number of cycles played, nor will it disconnect PCM from the voice link until the voice link is removed by a request from the SN.

This latter method of releasing voice links is the way it works for **all standard** voice links.

13.13.3.1 Signalling on broadcast voice links

Signalling is used on broadcast voice links to convey not only an 'Available/Not Available' status but also to signal the beginning of an announcement playback cycle.

Like the standard voice link, an idle voice link (one not used by any calls) is considered out of service if either the switch or the SN transmits an offhook; however, unlike standard voice links, an offhook received by the switch *during* a call will not result in the call being taken down, nor will the trunk go RMB at that time. Offhooks received from the SN during a call can be used to indicate the start of a new cycle, therefore they are either ignored (if the CUTTHRU type is IMMED) or are used to enable PCM (if CUTTHRU = HKCHG) and keep track of cycles (if CYCLE COUNT field in OAP message is other than 0). The offhook cannot be interpreted to mean 'Not Available' for broadcast voice links while at least one call is using the voice link. If the offhook still exists once the voice link is no longer in use by any calls it will at that time be put RMB.

The following table shows the possible hook-state transitions for broadcast voice links and the interpretation of each. For a transition to be considered valid, it must be in the new state for a minimum of 10 milliseconds if there are calls using the voice link, or 40 milliseconds if there are no calls on the link.

Table 19. Signalling used on broadcast voice links

SIGNAL	FROM	CALL ON VOICE LINK?	MEANING
OFFHK	SWITCH	NO	Remove from service
OFFHK	SWITCH	YES	Remove from service ^a
ONHK	SWITCH	NO	Available for service
ONHK	SWITCH	YES	Available for service
OFFHK	SN	NO	Remove from service
OFFHK	SN	YES	Start of playback cycle

Table 19. Signalling used on broadcast voice links

SIGNAL	FROM	CALL ON VOICE LINK?	MEANING
ONHK	SN	NO	Put back in service
ONHK	SN	YES	Playback begins

a. Call is taken down.

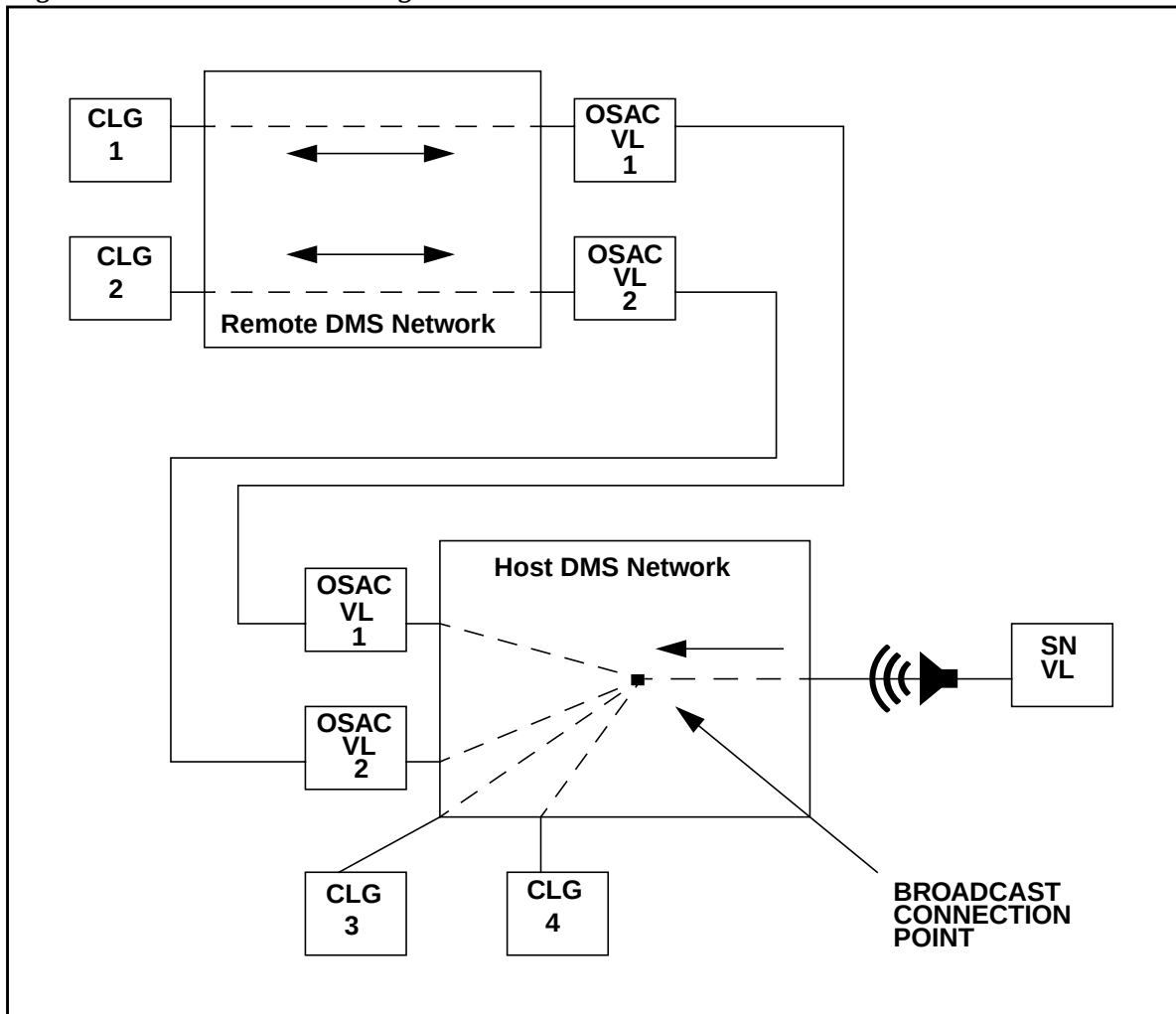
13.13.3.2 Broadcast voice links in simultaneous interactions

When a call has a broadcast voice connection, it must be the only voice connection up for the call. A call cannot have a broadcast connection for one function provider and a standard, non-broadcast connection for another function provider at the same time (in a simultaneous interaction mode). This includes both passive service nodes and passive operators. Note that a request for a passive operator when a broadcast voice link is attached to the active SN will result in a return error. The request for a passive SN is allowed; however, the passive SN cannot connect a voice link (standard or broadcast).

13.13.3.3 Voice Link Broadcasting in OSAC

Voice Link Broadcasting is supported when using the OSSAIN Centralization (OSAC) feature. Note that only Service Node voice links (voice links between a Host or Standalone switch and an SN) can be set up for broadcasting in table OAVLMAP. It is not allowed on the OSAC voice links (voice links between the Host and Remote switches).

Figure 27: “Voice Link Broadcasting in an OSAC environment.” illustrates how broadcasting works for OSAC. The Host office also functions as a Standalone OSSAIN office. Four separate calls are shown connected to a single broadcasting SNVL. Two of these originated in the OSAC Remote (calls 1 and 2) while the other two are Standalone calls that originated in the same switch serving as the OSAC Host (calls 3 and 4). Note that broadcasting for the Remote still requires an individual OSAC voice link for each call.

Figure 27. Voice Link Broadcasting in an OSAC environment.

13.13.4 Network delay allowance for all voice links

For both standard and broadcast voice links, an initial period of silence is needed before playing any prompt to ensure that speech has been established throughout the external network. Typically, 550 milliseconds is a sufficient amount of time to ensure speech has been established.

13.13.5 Trunk guard timing interactions for all voice links

When a voice link is released from a call and it is no longer in use by any other calls, it must remain unused for a certain amount of time before it can be reseeded. The amount of time is determined by the setting of the TRKGRDTM field for the voice link in table TRKSGRP. It is recommended that this field be set to the minimum possible value, which is 20 (200 milliseconds).

The trunk guard timer is started when a trunk state in the switch goes from Call Processing Busy (CPB) to Idle and it applies to both standard and broadcast voice links. For this reason,

the SN must wait before requesting use of the same voice link again. Failure to wait the specified amount of time will cause erratic behaviour on the trunk.

For broadcast voice links, the timer is only started when a call releases the voice link and there are no other calls using the voice link. At this point, the trunk state will go from CPB to Idle. If a call releases a broadcast voice link and other calls are still using it, the trunk state will remain CPB so the trunk guard timer is not started. The SN can continue to connect and release the voice link from calls without regard to the timer as long as the releasing call is not the last call using the voice link.

For standard voice links, any release of the voice link will start the trunk guard timer since the trunk state will always be changing from CPB to Idle.

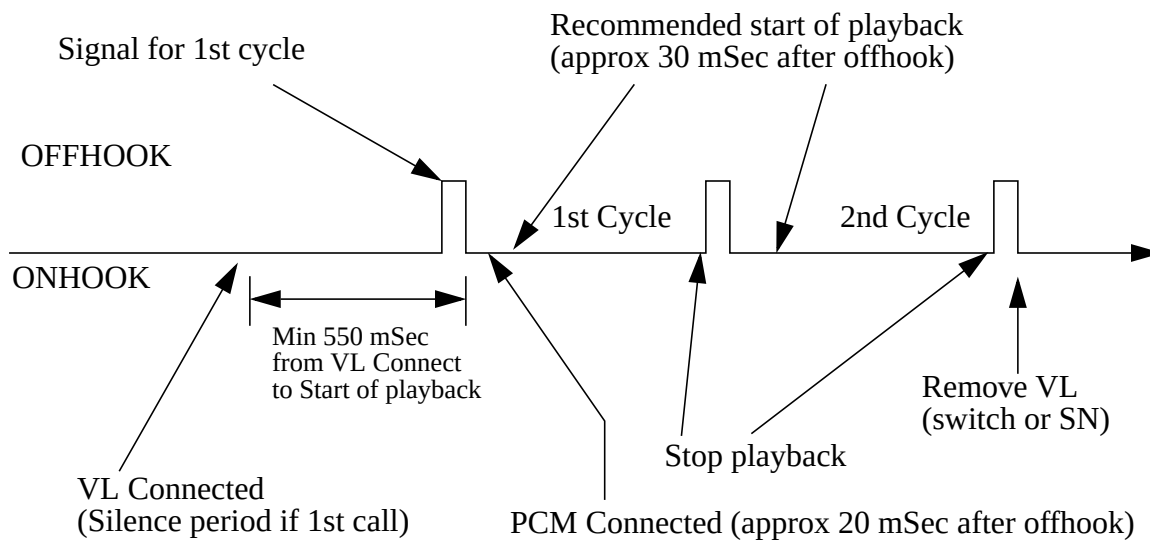
13.13.6 Increased OAP messaging for broadcast voice link failure

Broadcast voice links with large MAXCONNS values have the potential to generate a sudden, temporary burst of OAP messaging when a failure occurs on the voice link, such as a loss of carrier. Each call receiving the broadcast will detect the problem at roughly the same time and inform the SN of a change in status of the call. With the possibility of broadcasting to up to 1023 calls, a link failure could generate a message to each of the 1023 sessions currently using the voice link, in a very short period of time. For this reason, the message handling capacity of the SN should be kept in mind when determining the value of MAXCONNS.

13.13.7 Recommendations for cycling broadcast recordings

For broadcast voice links using the HKCHG method for determining when PCM is cut through, care must be taken when engineering the external system that will provide the recordings so that a caller is not subject to long periods of silence, or connected in the middle of a recording. It is envisioned that in such a system a recording will continuously repeat itself when at least 1 subscriber is still connected to it. A single playing of the entire recording is considered to be a cycle. During a particular cycle, any new callers added on will have to wait until the next cycle begins (indicated by the offhook) before PCM will be cut through to those terminals. This means that a caller may experience a period of silence while waiting for the beginning of the next cycle. The length of this period depends on the length of 1 cycle; therefore, it is recommended that cycle times be kept as short as possible.

The following diagram shows the possible sequence of events on a broadcast voice link using HKCHG. This shows a single call connecting to the voice link for a playback of two cycles.



Note that on all transitions from onhook to offhook, the offhook must be maintained for a minimum of 10 mSecs. Anything less than this will not be interpreted as the start of a cycle.

It is important that the recording NOT start playing while the voice link is offhook. Calls that are connected while the voice link is offhook are connected to PCM if that offhook has a duration of more than 10mSecs. If the recording is playing back while the voice link is offhook, new callers may be connected in the middle of the recording.

By not starting the playback until the onhook (plus 20 mSec filter), calls connected while the voice link is offhook will connect PCM (which will be silence at this point) if their 10mSec filter timer expires. If it doesn't expire before the onhook, PCM will not be connected until the next cycle. Either way they are guaranteed not to be connected in the middle of the recording. All PCM-connected parties will hear the recording start at the same time. Those that did not connect PCM will wait until the next cycle.

In general, the offhook (after expiration of a 10mSec filter) should be interpreted as a 'start PCM' signal to the switch. The onhook (followed by the expiration of a 20 mSec filter), should be interpreted as the 'start playback' point for the SN.

In all, a 30 mSec delay between the initial offhook and the start of the playback is recommended in order for the offhook to be detected and filtered by the connected party and to allow time for connecting the PCM.

The recommended sequence of events for connecting callers to an idle broadcast link:

- Voice Connect Success Response is received by the SN.

- Voice link remains onhook and silent for a minimum of 540 mSec. This is to partially allow for establishment of the external network. In the diagram the total duration of this delay for the network (approximately 570 mSecs) is the period of time between the 'VL Connected' and 'PCM Connected'.
- Voice link goes offhook for a minimum of 10mSecs. Any new callers added to the voice link will have PCM cut through if this offhook still exists after their 10mSec filter timer expires.
- Voice link goes back onhook. After a minimum of 30 mSecs has elapsed from the time the VL initially went offhook, playback starts.
- Playback stops at end of cycle.
- Last 3 steps are repeated for each cycle until the last caller is removed from VL, at which point the VL goes back to an idle state.

Along with short cycles, another efficient way of limiting 'dead air' time is to have a group of voice links that provide the same recording but at staggered intervals. New callers could then be connected to the trunk in the group that is nearest to completing a cycle (assuming that it will not exceed the maximum number of connections allowed for that voice link).

A slightly less efficient but simpler method, also using a group of voice links providing the same recording, is to select a trunk by sequencing through each member in the group. Each new call goes to the next member of the group, regardless of where it is in the cycle. As an example, the following 3 voice links are set up to play the same branding recording, with a cycle time of 5 seconds:

- BCSTVL 0
- BCSTVL 1
- BCSTVL 2

All voice links are initially idle and onhook. Nine calls arrive within a 5 second time frame that need to have the same branding announcement played. The first call is connected to BCSTVL 0, which goes offhook toward the DMS to indicate the beginning of the recording (cycle 1). The remaining calls are connected to the voice links shown in Table 20, "Example order of broadcast voice link selection," , each going offhook when its connection is first made.

Table 20. Example order of broadcast voice link selection

CALL 1	BCSTVL 0
CALL 2	BCSTVL 1
CALL 3	BCSTVL 2
CALL 4	BCSTVL 0
CALL 5	BCSTVL 1
CALL 6	BCSTVL 2

Table 20. Example order of broadcast voice link selection

CALL 7	BCSTVL 0
CALL 8	BCSTVL 1
CALL 9	BCSTVL 2

When call 4 is initially connected to BCSTVL 0, the voice link is in the middle of its first 5-second cycle. Call 4 will receive silence until the end of that cycle; however, its period of silence should be fairly short since subsequent calls were added to other members (ensuring that some period of time will have elapsed since the 1st caller was connected to the 1st voice link). In contrast, if only 1 voice link were used, the second call would have to wait until the 2nd cycle (most likely) and could conceivably be connected earlier in the cycle, therefore receiving more ‘dead air’ time.

Obviously, this is a fairly simple method. More complex algorithms could be used to achieve greater efficiency. The main point is that selection of a voice link should take into account the fact that the subscriber could be subjected to a certain period of silence while waiting for the recording to begin. Keep in mind that this only affects voice links using the HKCHG method. If IMMEDIATE is used PCM is cut through immediately. No ‘dead air’ time should occur.

Another problem that can exist with cycled recordings is the task of providing the quick removal of the PCM from the voice link. If a caller should hear only two cycles of a recording, it is critical that PCM be removed in a timely manner such that the subscriber does not hear the beginning of the next cycle before it is removed. To speed up the process of disconnecting PCM the DMS can be given the responsibility of monitoring the number of cycles and then cutting off PCM once all cycles have been played. Along with this, the DMS will release the voice link and inform the SN of this event.

At the time of the request, if the SN specifies a non-zero number for the number of cycles be played, the switch will take control of counting the cycles. Once all cycles have been played PCM is disconnected *and* the voice link is released from the call. The DMS will inform the SN of such an event (after the VL is released).

At the time of the connect request by the SN, determination is made as to whether the DMS will count cycles, and as a result have control of the release of the voice link. If the DMS is not to keep track of cycles, PCM (once cut through) will remain on the terminal until the Service Node requests the removal of the voice link. This means that if a caller should hear two cycles of a recording and no more than that, the SN needs to request voice link removal immediately at the end of the 2nd cycle. In these situations it is highly recommended that there be a short interval of silence between the offhook state change and the beginning of the audio. This would help in ‘syncing’ up the caller with the start and end of the recording when delays occur between a request and the actual disconnecting of PCM.

Appendix A: ASN.1 Encoding of the OAP Operation Header

A.1 ASN.1 Syntax & Encoding

ASN.1 refers specifically to a syntax for describing the structure of Application Layer data (i.e. the data in our Bundles/Operations), although it may be used for other layers, as well. In CCITT X.409, ASN.1 also referred to a method of encoding that application data. In ISO 8824 (ASN.1) and 8825 (BER - Basic Encoding Rules), the two discussions were separated, with the consideration that alternate encodings might exist in the future. These two areas describe the content/intent of the OSI Presentation Layer (Layer 6).

A.1.1 ASN.1 Definitions of ROSE Operations

The ROSE Operation types and their parameters are defined in ASN.1 terms. This section displays ASN.1 encoding of the ROSE Operations and their parameters. X.409/X.410 syntax and descriptions are used (the X.409/X.410 syntax/descriptions are more concise and readable, if not as all-inclusive of new options/descriptions).

One Application Entity (AE) invokes a remote operation in another AE through an exchange of Operation Protocol Data Units (OPDUs). There are four kinds of OPDU:

OPDU ::= CHOICE {[1] Invoke, [2] ReturnResult, [3] ReturnError, [4] Reject}

A.1.1.1 ASN.1 Definition of Invoke Operation

The Invoke Operation type is defined as a type with the three ‘sequential’ parameters discussed:

Invoke ::= SEQUENCE {invokeID INTEGER, OPERATION, argument ANY}

OPERATION is an ASN.1 macro which boils down to an INTEGER value, assigned by the user, with an accompanying syntax referencing a Result type and a set of valid Error types. Send Operations have neither Result nor Error references (i.e. depending on Operation class, one or both may be optional). The ‘argument’ may be Simple (i.e. a basic predefined type, examples of which include INTEGER, REAL, and NULL) or Constructed (i.e. data structures). We use a Simple ‘argument’ type of OctetString to encode our operation data (see the Operation Data Formatting section of this document for encoding formats). This encoding minimizes processing and message size overheads.

A.1.1.2 ASN.1 Definition of ReturnResult Operation

ReturnResult ::= SEQUENCE {invokeID INTEGER, result ANY}

(The ‘result’ may be Simple (i.e. a basic predefined type, examples of which include INTEGER, REAL, and NULL) or Constructed (i.e. data structures). We use a Simple ‘result’ type of OctetString to encode our operation result data (see the Operation Data Formatting section of this document for encoding formats). This encoding minimizes processing and message size overheads.

A.1.1.3 ASN.1 Definition of ReturnError Operation

The ReturnError OperationType is defined as a sequence of three data fields:

ReturnError ::= SEQUENCE {invokeID INTEGER, ERROR, parameter ANY}

ERROR must be one of the error types referenced/specified in the Invoke Operation description.

The error 'parameter' may be Simple (i.e. a basic predefined type, examples of which include INTEGER, REAL, and NULL) or Constructed (i.e. data structures). We use a Simple 'parameter' type of OctetString to encode our operation error data (see the Operation Data Formatting section of this document for encoding formats). This encoding minimizes processing and message size overheads.

A.1.1.4 ASN.1 Definition of Reject Operation

The Reject Operation is defined as a sequence of two fields, one of which has a refinement:

Reject ::= SEQUENCE {invokeID CHOICE {INTEGER, NULL}
 {problem CHOICE {
 [0] IMPLICIT GeneralProblem,
 [1] IMPLICIT InvokeProblem,
 [2] IMPLICIT ReturnResultProblem,
 [3] IMPLICIT ReturnErrorProblem}}}

An example of RejectReasons for GeneralProblem include (this list has grown over time):

GeneralProblem ::= INTEGER {
 unrecognizedOPDU(0),
 mistypedOPDU(1),
 badlyStructuredOPDU(2)}

IMPLICIT indicates the following field type is overridden through predefinition as a predeclared type, thus, for the Reject operation, the Reject Reason and Problem fields should be handled as one field, with the Type field indicating the problem (0.3), and a Value field containing the RejectReason.

A.1.2 General ASN.1 Encoding (Basic Encoding Rules)

Encoding for ASN.1 data types, following BER are of the following format:

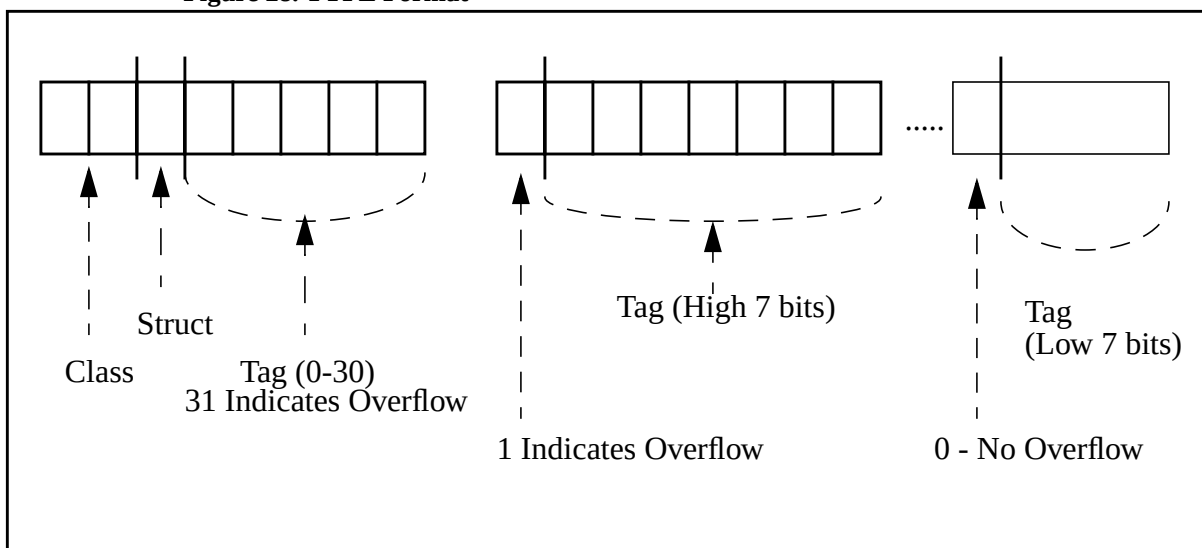
Table 21: ASN.1 General Format

TYPE	LENGTH	VALUE

A.1.2.1 Encoding of TYPE

TYPE is defined by one or more octets/bytes in the following manner:

Figure 28. TYPE Format



Class: 00 - Universal (Defined in ASN.1)

01 - Application-Wide (Unique within application)

10 - Context-Specific (Unique only within context of use)

11 - Private (Private agreement between users)

Struct: 0 - Simple type (i.e. not a SEQUENCE, SET OF, etc.)

1 - Constructed type (SEQUENCE OF, SET, etc.)

Tag: 0-30 1 Octet

31-127 2 Octets

128-16k 3 Octets

...

...

Note: BER indicates Tag values must be encoded in fewest Octets encodable.

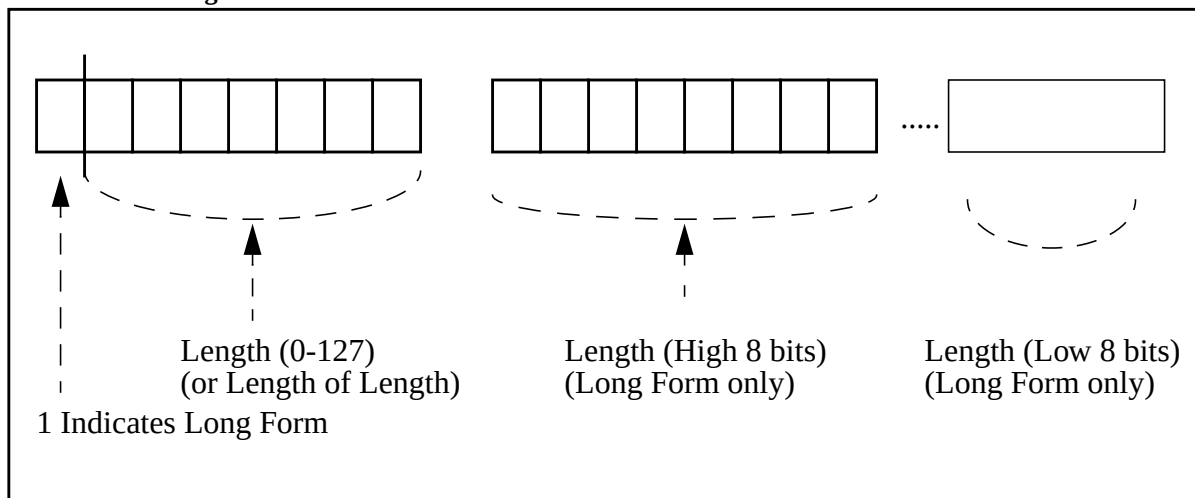
Table 22: Some Universal-Class Types (Defined in ASN.1)

ID Code (Decimal)	Data Type
0	End-of-contents
1	Boolean
2	Integer
3	Bit String
4	Octet String
5	Null
16	Sequence
17	Set
18	Numeric String
19	Printable String
20	T.61 String
21	Videotex String
22	IA5 String
23	UTC Time
24	Generalized Time

A.1.2.2 Encoding of LENGTH

LENGTH is defined by one or more octets/bytes in the following manner:

Figure 29. LENGTH Format



Form: 0 - Short Form (Length of Value is 0-127 bytes)

1 - Long Form: Value = 0: Indeterminate Length, (Constructed Types only)

End-of-Contents Indicator in Value field

(End-of-Contents Indicator is 2 Octets of 0 in Value Field)

Value = 1-126: Length of Length is 1-126 bytes,

Subsequent n octets contain Length of Value

Length: 0-127 bytes 1 Octet (Short Form)

0-256 bytes 2 Octets (Long Form)

256-64K bytes 3 Octets (Long Form)

... ..

Note: BER indicates that for Integer Type, Short Form may not indicate a length longer than Value requires. Long Form does not suffer from this restriction.

Implications: OperationIDs, InvokeIDs, and ErrorIDs (INTEGERS of Length 2 bytes) must use only Values of 256 or greater (i.e. Range of 256 - 64K).

Appendix B: Message Set

B.1 Summary of Operations

The following tables list the operations by various groups. The details of each operation are found in previous chapters of this document.

B.1.1 Call Processing Request Operations

These operations are call related. For these operations, the sender is requesting an action of the receiver and is expecting a response. The possible responses are defined with the operation. Refer to Chapter 7: “Call Processing Operations” for a detailed description of each operation.

All the operations in this table are in the **Call Processing Class** and are **Invoke** type messages.

Table 23. Call Processing Request Operations

Operation	Sender(s)			Receiver(s)		
	Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Append AMA Module Request		X		X		
Billing Number Request		X		X		
Call Details Request		X		X		
Call Float Request		X		X		
Call Merge Request		X		X		
Call Resume Request		X		X		
Calling Number BNS Query Request		X		X		
Carrier Assignment Request		X		X		
Class Charge Request		X		X		
Conference Add Request		X		X		
Conference Create Request		X		X		
Conference Details Request		X		X		
Conference Release Request		X		X		
Conference Remove Request		X		X		
Connect DN Request		X		X		
Connect to Treatment Request		X		X		
Conversation Timing Request		X		X		
DTMF Receiver Release Request		X		X		
Directory Number Request		X		X		
End Call Request		X		X		
Estimate of Charges Request		X		X		
Generate AMA Record Request		X		X		
ISUP Calling Party Number Update Request		X		X		
LNP Request		X		X		
LRN Update Request		X		X		
Notification Timer Request		X		X		

Table 23. Call Processing Request Operations

Operation	Sender(s)			Receiver(s)		
	Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Pass Control Request		X		X		
Passive Function Provider Request		X		X		
Pass Through Request		X	X	X		
Release DN Request		X		X		
Release Node Request		X	X	X		
Return Answer Message Request		X		X		
Route To Treatment Request		X		X		
Service Change Request		X		X		
Service Provider ID Assignment Request		X		X		
Session Initiation Request		X		X		
Session Recall Request		X		X		
Speech Path Request		X		X		
Text To Operator Request		X		X		
Transfer To Carrier Request		X		X		
Transfer To Control List		X		X		
Voice Connect Request		X	X	X		
Voice Release Request		X	X	X		

B.1.2 Call Processing Inform Operations

These operations convey information about the call to the receiver. They are not requests, and therefore, they do not have corresponding success and error responses. However, these operations can cause protocol violations. Refer to Chapter 7: “Call Processing Operations” for a detailed description of each operation.

All the operations in this table are in the **Call Processing Class** and are **Invoke** type messages.

Table 24. Call Processing Inform Operations

Operation	Sender(s)			Receiver(s)		
	Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Accept Control Inform	X					X
Call End Inform	X				X	X
Call Merge Inform	X				X	
Connection Status Inform	X				X	
DTMF Digit Detected Inform	X				X	
Node Release Inform	X				X	X
Notification Timer Inform	X				X	
Pass Through Inform	X				X	X
Passive Node Status Inform	X				X	
Session Begin Inform	X				X	X
Trigger Event Inform	X				X	
Voice Release Inform	X				X	X

B.1.3 Non-Call Processing Request Operations

These operations are non-call related. For these operations, the sender is requesting an action of the receiver and is expecting a response. The possible responses are defined with the operation. Refer to Chapter 8: “Non-Call Processing Operations” for a detailed description of each operation.

All the operations in this table are **Invoke** type messages.

Table 25. Non-Call Processing Request Operations

Operation	Class		Sender(s)		Receiver(s)	
	Node	Session Pool	Maintenance Switch	Service Node	Maintenance Switch	Service Node
Node Audit Request	X		X			X
Node Busy Request	X		X	X	X	X
Node Return To Service Request	X		X			X
Node Test Request	X		X			X
Session Pool Audit Request		X	X			X
Session Pool Busy Request		X	X	X	X	X
Session Pool Return To Service Request		X	X			X
Session Pool Test Request		X	X			X

B.1.4 Restricted Operations

This table lists the operations that are considered Restricted Operations. The restriction is listed along with the operation.

Service nodes should not send any operations to the switch until they have received a response for the restricted operation.

All the operations in this table are in the **Call Processing Class** and are **Invoke** type messages.

Table 26. Restricted Operations

Operation	Restriction	Sender(s)			Receiver(s)		
		Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Billing Number Request	If a B party DN was entered for the billing number, an LNP query may be launched to obtain LNP information for this DN. The service node should not send another request until it receives a response for the Billing Number Request.		X		X		
Calling Number BNS Query Request	The switch needs to wait for the query response from the external data base. The service node should not send another request until it receives a response for the Calling Number BNS Query Request.		X		X		
Class Charge Request	If validation is requested, the switch needs to wait for the query response from the external data base. The service node should not send another request until it receives a response for the Class Charge Request.		X		X		

Table 26. Restricted Operations

Operation	Restriction	Sender(s)			Receiver(s)		
		Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Connect DN	The switch generally waits for an ACM message upon attempting to make a connection on ISUP trunks. The service node should not send another request until it receives a response for the Connect DN Request.		X		X		
Directory Number Request	Updating the A party directory number may cause the switch to launch an OLNS query to an external data base. The service node should not send another request until it receives the response from the Directory Number Request.		X		X		
LNP Request	This operation requests the LNP information for the specified directory number. This may result in the switch sending a query to the LNP database.		X		X		
Passive Function Provider Request	This operation requests a passive function provider for a simultaneous interaction.		X		X		
Pass Through Request	The maximum number of bytes that can be sent in 1 message to an operator providing a Passive Function is 25 bytes. Since the Pass Through DB can be 100 bytes the service node must wait while multiple messages are sent to the operator position.		X	X		X	X

Table 26. Restricted Operations

Operation	Restriction	Sender(s)			Receiver(s)		
		Call Processing Switch	Active Service Node	Passive Service Node	Call Processing Switch	Active Service Node	Passive Service Node
Release DN Request	When releasing the back party, the switch has to do a blind coin return if the call is from a coin phone. This may take a few seconds because the switch has to wait for coin signals to complete. The service node should not send another request until it receives the response from the Release DN Request.		X		X		
Voice Connect Request	If voice connections are being made for a call involving a centralized service node, and an OSAC Remote, messaging will occur between the OSAC Host and OSAC Remote. The service node (active or passive) should not send any requests until the voice connection is complete		X	X	X		

B.2 Identifier Cross Referencing

The following tables cross reference a specific operation or data block identifier to the actual operation or data block.

B.2.1 Operation Identifiers

Table 27. Identifier to Operation Cross Reference

Identifier (#)	Operation	Class						
		Call Processing	MIS	Node	Node Alarm	Node Log	Session Pool	Session Pool Alarm
010D	MIS OSSAIN Inform		X					
010E	Node Alarm Inform				X			
010F	Node Log Inform					X		
0110	Node RTS Request			X				
0111	Node Busy Request			X				
0112	Node Audit Request			X				
0113	Node Test Request			X				
0300	Session Pool RTS Request						X	
0301	Session Pool Busy Request						X	
0302	Session Pool Audit Request						X	
0303	Session Pool Test Request						X	
0304	Session Pool Throttle Request						X	
0305	Session Pool Drain Inform						X	
0306	Session Pool Log Inform							X
0307	Session Pool Alarm Inform							X
0308	Session Pool State Inform						X	
0600	Append AMA Module Request	X						
0601	Generate AMA Record Request	X						
0602	Billing Number Request	X						
0603	Class Charge Request	X						
0604	Call Details Request	X						
0605	Call End Inform	X						
0606	Call Float Request	X						
0607	Directory Number Request	X						
0608	End Call Request	X						
0609	Connection Status Inform	X						
060A	Connect DN Request	X						
060B	Release DN Request	X						
060C	Route to Treatment	X						

Table 27. Identifier to Operation Cross Reference

Identifier (#)	Operation	Class						
		Call Processing	MIS	Node	Node Alarm	Node Log	Session Pool	Session Pool Alarm
060D	Session Begin Inform	X						
060E	DTMF Digit Detected Inform	X						
060F	Session Initiation Request	X						
0610	Speech Path Request	X						
0611	Text to Operator Request	X						
0612	Carrier Assignment Request	X						
0613	Transfer to Control List Request	X						
0614	Trigger Event Inform	X						
0615	Voice Connect Request	X						
0616	Voice Release Request	X						
0617	DTMF Receiver Release Request	X						
0618	Transfer to Carrier Request	X						
0619	Passive Function Provider Request	X						
061A	Release Node Request	X						
061B	Pass Control Request	X						
061C	Passive Node Status Inform	X						
061D	Node Release Inform	X						
061E	Accept Control Inform	X						
061F	Pass Through Inform	X						
0620	Call Resume Request	X						
0621	Conversation Timing Request	X						
0622	LNP Request	X						
0623	Service Provider ID Assignment Request	X						
0624	Conference Create Request	X						
0625	Conference Add Request	X						
0626	Conference Remove Request	X						
0627	Conference Release Request	X						
0628	Conference Details Request	X						
0629	Call Merge Request	X						
062A	Call Merge Inform	X						
062B	Notification Timer Request	X						
062C	Notification Timer Inform	X						
062D	Session Recall Request	X						

Table 27. Identifier to Operation Cross Reference

Identifier (#)	Operation	Class						
		Call Processing	MIS	Node	Node Alarm	Node Log	Session Pool	Session Pool Alarm
062E	ISUP Calling Party Number Update Request	X						
062F	Voice Release Inform	X						
0630	Service Change Request	X						
0631	Estimate of Charges Request	X						
0632	Connect to Treatment Request	X						
0633	Pass Through Request	X						
0634	Return Answer Message Request	X						
0635	Calling Number BNS Query Request	X						
0636	LRN Update Request	X						

B.2.2 Data Block Identifiers

Table 28. Identifier to Data Block Cross Reference

Identifier (#)	Data Block
0100	Abort Call DB - removed in OAP 2.0
0101	Maintenance Text DB
0102	Alternate Billing Directory Number DB
0103	AMA Module DB
0104	Billing Information DB
0105	Billing Number Validation DB
0106	Call In Queue DB
0107	Call Information DB
0108	BCD Card Number DB
0109	Carrier DB
010A	Charge Status DB
010B	Class Charge DB
010C	Connection Event DB
010D	Connection ID DB
010E	Connection Progress DB
010F	Network Service DB
0110	Context Block DB
0111	Directory Number DB
0112	Directory Number Update DB
0113	ASCII Card Number DB
0114	Function DB
0115	Generate AMA DB
0116	DTMF Digit DB
0117	Node Identification DB
0118	Operator Identification DB
0119	Originating Station Information DB
011A	Originating Trunk Information DB
011B	Speech Path DB
011C	Release Cause DB
011D	Text DB
011E	Time DB
011F	Transfer Control List DB
0120	BCD Card Information DB
0121	Treatment DB
0122	Trigger DTMF DB
0123	Trigger Information DB

Table 28. Identifier to Data Block Cross Reference

Identifier (#)	Data Block
0124	Trigger Profile Index DB
0125	Trigger Release Cause DB
0126	Trigger Treatment DB
0127	Voice Channel DB
0128	Call Float Indicator DB
0129	Sequence Number Check DB
012A	Trigger Ringing DB
012B	Connect DN Information DB - removed in OAP 3.0
012C	BNS Query Result DB
012D	CCV Query Result DB
012E	Invalid Billing Details DB
012F	Language DB
0130	Billing Acceptance DB
0131	Billing Number Input DB
0132	End Call Handling DB
0133	Operator Services System Action DB
0134	OLNS Information DB
0135	Carrier Number DB
0136	LATA Information DB
0137	Passive Node Status DB
0138	Pass Through DB
0139	Release Node DB
013A	Call Queue Information DB
013B	Alarm Information DB
013C	Report Text DB
013D	Throttle DB
013E	Branding Indicator DB
013F	Call Timing DB
0140	Conversation Timing Indicator DB
0141	Operator System DB
0142	LNP Information DB
0143	RLT Eligible DB - removed in OAP 4.0
0144	RLT Request
0145	SPID DB
0146	Preprocessing Information DB
0147	Maintenance State DB
0148	NBEC DB
0149	Callid DB
014A	Conference ID DB
014B	Conference Party Connection DB

Table 28. Identifier to Data Block Cross Reference

Identifier (#)	Data Block
014C	Conference Party Connection Information DB
014D	Conference SN Voice Connection DB
014E	Conference SN Voice Connection Information DB
014F	Notification Timer DB
0150	Session Information DB
0151	ISUP Calling Party Number DB
0152	Terminating Trunk Information DB
0153	Broadcast DB
0154	Requested Service DB
0155	Broadcast Count DB
0156	Directory Number Information DB
0157	Information ID DB
0158	OSSAIN Call Queue Event DB
0159	OSSAIN Session Event DB
015A	OSSAIN Non-Call Session Event DB
015B	Country Direct Information DB
015C	Routing Index DB
015D	Routing Number DB
015E	SN Voice Link DB
015F	Estimate of Charges Alternate Date DB
0160	Estimate of Charges Alternate Time DB
0161	Estimate of Charges Duration DB
0162	Estimate of Charges Duration Information DB
0163	Estimate of Charges General Information DB
0164	Estimate of Charges No Duration Information DB
0165	Call Update DB
0166	LRN Update DB
0167	LNP Enhancements DB

B.3 Treatments

The following is a mapping of current treatments.

- 0 = Undefined Treatment
- 1 = No Service Circuit
- 2 = Partial Dial
- 3 = Permanent Signal
- 4 = Invalid Account Code
- 5 = Coin Denied Term
- 6 = Vacant Code
- 7 = Misdirected CAMA Call
- 8 = Misdirected Local
- 9 = Network Block, Heavy Traffic
- 10 = Network Block, Normal Traffic
- 11 = Emergency 1
- 12 = Emergency 2
- 13 = Unauthorized CAMA Code
- 14 = System Failure
- 15 = CAMA Queue Overflow
- 16 = HNPA Code Intercept
- 17 = Unassigned Number
- 18 = Blank Directory Number
- 19 = Busy Line
- 20 = Unauthorized Outwats
- 21 = Toll Denied
- 22 = Unauthorized Inwats
- 23 = Start Signal Time Out
- 24 = No Circuit
- 25 = Reorder
- 26 = Manual Line
- 27 = Orig Susp Serv
- 28 = Term Susp Serv
- 29 = Regular Intercept

30 = Trouble Intercept
31 = Machine Intercept
32 = Perm Sign No ROH
33 = Denied Termination
34 = Near End Cong
35 = Far End Cong
36 = Orig Rev Multi Coded
37 = Toll Overload
38 = Confirm Tone
39 = Rev Ring PFXA
40 = Orig Rev Freq
41 = Term Rev Freq
42 = Orig Rev Coded
43 = Orig Rev Multi Freq
44 = Single Rev Ring
45 = Disconnect Timeout Treatment
46 = Unauthorized Precedence
47 = Blocked Precedence Call
48 = Emergency 3
49 = Emergency 4
50 = No Coin
51 = Preempt Tone
52 = Storage Overflow Reorder
53 = Invalid Authorized Code
54 = Temporarily Invalid
55 = Coin Overtime Treatment
56 = Disallowed Coin Free Call
57 = Priority Screen Fail
58 = Generalized No Circuit
59 = Attendant Busy
60 = Music On Hold
61 = Deny Orig Data Terminal

62 = Testdesk Bridged
 63 = Restricted Date Time
 64 = Premature Trunk Offering
 65 = Vacant Speed Number
 66 = Answer Timeout
 67 = No Metering Zone
 68 = Feature Not Allowed
 69 = Unregistered Mobile
 70 = Trunk Perm Ground
 71 = Signal Time Out BOC
 72 = Signal Time Out IC INC
 73 = Emergency 5
 74 = Emergency 6
 75 = Invalid OIC Code
 76 = ANI Account Status Not Allowed
 77 = CFW Verification
 78 = Feature Action NACK
 79 = Carr ACC Code Error
 80 = Dial 950
 81 = No Dial 950
 82 = InterLATA RES
 83 = No Dial Access Code
 84 = Dial Access Code
 85 = ANI Database Failure
 86 = Mobile Page Timeout
 87 = AIOD Failure
 88 = First Digit Not Zero
 89 = Calling Card Timeout
 90 = Calling Card Invalid
 91 = Calling Card Not Allowed
 92 = Feature Data Error
 93 = No Software Resource

94 = Customer Group Resource Overflow
95 = Vacant Country Code
96 = Local Call Area Barred
97 = Invalid City code
98 = Connection Not Possible
99 = Changed Number, Intercept
100 = DBS Communication Failure
101 = NCS Incoming Exclusion
102 = NCS Invalid ID Code
103 = NCS Translation Failure
104 = Not On Network
105 = NCS Unexpected Error
106 = ATD Timeout
107 = ANI FGB Block
108 = Invalid Corridor Call
109 = Service Currently Unavailable
110 = Invalid PIN Digits
111 = NPA Restricted
112 = IDDD Prohibited
113 = Call Not Accepted
114 = Clear Back Tone
115 = MFC Timeout Or Confusion
116 = Ann Failer
117 = Changed Number Announcement
118 = Changed Number Forwarded
119 = Operator Services Voice Response
120 = N00 Call Blocked
121 = CFW Simultaneous Overflow
122 = ILR Restricted
123 = COS Exceeded
124 = Carrier Access Code Block
125 = Service Intercept

126 = International Wakeup Call
 127 = INB Treatment
 128 = NCS 800 Failure
 129 = Feature Reorder
 130 = CCS7 Application Failure
 131 = Datafill Error
 132 = STR Blue Box Fraud Detected
 133 = No Terminal Responding
 134 = Call Rejected
 135 = Universal Public Access Block
 136 = STN Orig Restriction Error
 137 = Call Not Allowed
 138 = Vacant Prefix Code
 139 = Credit Card Announcement Prompt
 140 = Authcode Prompt
 141 = Credit Card Number Invalid
 142 = Address Digits Prompt
 143 = Credit Card Dial Tone
 144 = Unpaid Credit Card Number
 145 = Call Back Destination Number
 146 = NCS 900 Database Failure
 147 = NCS 900 Number Out Of Band
 148 = NCS 900 Not In Service
 149 = Carrier Call Comp Fail
 150 = SCRJ Call Reject
 151 = Invalid Conference Code
 152 = LEC Calling Card Valid
 153 = LEC Calling Card Not Valid
 154 = CUG Service Not Allowed
 155 = VPN Service Not Allowed
 156 = POTS Service Not Allowed
 157 = SCA Call Reject

158 = NCS Treatment 0
159 = NCS Treatment 1
160 = Channel NEG Fail
161 = BC Not Implemented
162 = Audible Ringing Tone
163 = Justified ALT Calling Knowledge
164 = ITC Invalid
165 = Not Very Important Person
166 = ACRJ Call Reject
167 = Facility Not Implemented
168 = Protocol Error
169 = Invalid Message
170 = Serv Opt Not Impl
171 = CDA ACT Successful
172 = CDA ACT Failed
173 = CDA Deact Successful
174 = CDA Deact Failed
175 = ANI Account Recently Disallowed
176 = DSCWID Discon
177 = Global Foncard Not Valid
178 = Local Busy
179 = Toll Busy
180 = Invalid Info Element Contents
181 = No BC Available
182 = No Routing Available
183 = Protocol Error 1
184 = Protocol Error 2
185 = Protocol Error 3
186 = Protocol Error 4
187 = Protocol Error 5
188 = CUG Error 1
189 = WUCR Call Treatment

190 = Mobile Trouble
 191 = Mobile Weak Power
 192 = RMT Feat CNTL Success
 193 = RMT Feat CNTL Denied
 194 = RMT Feat CNTL Error
 195 = MTX ERV Originator
 196 = MTX ERV Terminator
 197 = MTX ERV Timeout
 198 = Trigger Block
 199 = Manufacturer ESN Fraud
 200 = Mobile Inactive
 201 = Q33 Failure
 202 = Q33 Fail2
 203 = AIN Disconnect Call
 204 = AIN Final Treatment
 205 = Invalid Redir Feat Code
 206 = REDIR Tandem Threshold Exceeded
 207 = International Toll Denied
 208 = Profile Timeout
 209 = Mobile IS Being Paged
 210 = Call Is Being Forwarded
 211 = LDSA (Long Distance Signal Access) Activated
 212 = LDSA (Long Distance Signal Access) Deactivated
 213 = Overflow From Ring To Busy
 214 = AVP Authentication Fail
 215 = AVP Feat 2
 216 = Black Listed Mobile
 217 = Exchange Routing Error

Index

A

Accept Control Inform 205, 212
Access Indicator 625
ADACC Transition Indicator 625
Alarm Information DB 459
Alarm Severity 626
Alarm Status 626
Allowed Billing Set 627
Alternate Billing Directory Number DB 460
AMA Data 628
AMA Handling 629
AMA Hex Digit Count 630
AMA Module DB 463
ANI Status 630
Append AMA DB 216
Append AMA Module Error Response 219
Append AMA Module Request 181, 212, 216
Append AMA Module Success Response 218
Append AMA Sequence Number 216, 632
Application ID 631
ASCII Card Number 632
ASCII Card Number DB 465
ASCII Card Number Length 633
ASCII Card Number Type 634
ASCII PIN Length 634
Authorization Number 635
Authorization Number Length 636
Available Sessions 636

B

BCD Card Information DB 467
BCD Card Number 637
BCD Card Number DB 468
BCD Card Number Digit Count 637
BCD Card Number Type 222, 638
BCD PIN Digit Count 639
Billable Status 640
Billing Acceptance DB 471

Billing Acceptance Response Method 641
Billing Data 642
Billing Information DB 472
Billing Number Error Response 223
Billing Number Input DB 474
Billing Number Input Method 642
Billing Number Request 183, 220
Billing Number Success Response 222
Billing Number Type 643
Billing Number Validation DB 475
BNS Query Response 639
BNS Query Result DB 470
Branding Indicator DB 476
Branding Status 644
Broadcast Count 479
Broadcast Cycle Count 644
Broadcast DB 477

C

Call Details Error Response 229
Call Details Request 185, 224
Call Details Success Response 225
Call End Inform 186, 230
Call Float Error Response 236
Call Float Indicator DB 481
Call Float Request 187, 188, 208, 231
Call Float Success Response 235
Call Handling 646
Call ID 647
Call In Queue DB 482
Call In Queue Take Down 647
Call Information DB 483
Call Merge Error Response 244
Call Merge Inform 191, 239
Call Merge Request 191, 241
Call Merge Success Response 243
Call Origination Type 648
Call Queue 651
Call Resume Error Response 248

Call Resume Request 245
Call Resume Success Response 247
Call Timing DB 489
Call Type for Queuing 651
Call Update DB 491
Callid DB 492
Calling Line Identifier Digit Count 653
Calling Line Identifier Number 653
Calling Number BNS Query Error Response 251
Calling Number BNS Query Request 183, 249
Calling Number BNS Query Success Response 250
Calls In Queue 652
Carrier Assignment Error Response 254
Carrier Assignment Request 189, 191, 249, 252
Carrier Assignment Source Indicator 654
Carrier Assignment Success Response 253
Carrier DB 493
Carrier Identification Code 655
Carrier Identification Digit Count 656
Carrier Name 656
Carrier Name Length 657
Carrier Number Status 658
Carrier Service Status 658
CCV Query Response 645
CCV Query Result DB 480
Charge Status DB 496
Class Charge DB 497
Class Charge Error Response 258
Class Charge Request 183, 220, 255
Class Charge Selector 659
Class Charge Success Response 257
Coin Key Pad State 659
Coin Overtime Type 660
Conference Add Error Response 262
Conference Add Request 191, 259
Conference Add Success Response 261
Conference Create Error Response 266
Conference Create Request 191, 263
Conference Create Success Response 265
Conference Details Error Response 269
Conference Details Request 191, 267
Conference Details Success Response 268
Conference ID 661
Conference ID DB 500
Conference Party Connection DB 501
Conference Party Connection Information DB 502
Conference Release Error Response 272
Conference Release Request 191, 270
Conference Release Success Response 271
Conference Remove Error Response 275
Conference Remove Request 191, 273
Conference Remove Success Response 274
Conference SN Voice Connection DB 504
Conference SN Voice Connection Information DB 505
Connect DN Error Response 279
Connect DN Request 194, 196, 276
Connect DN Success Response 278
Connect to Treatment Error Response 283
Connect To Treatment Request 190
Connect To Treatment Success Response 282
Connection Event 661
Connection Event DB 506
Connection ID 662
Connection ID DB 507
Connection Progress DB 508
Connection Status 663
Connection Status Inform 193, 284
Connection To Treatment Request 281
Context Block Data 663
Context Block DB 509
Context Block Length 664
Control List ID 665
Conversation Timing Error Response 287
Conversation Timing Indicator DB 511
Conversation Timing Request 285
Conversation Timing Status 665
Conversation Timing Success Response 286
Country Direct Access Code Count 666
Country Direct Access Code Number 666
Country Direct Carrier Code Count 667
Country Direct Carrier Code Number 667
Country Direct Country Code Count 668
Country Direct Country Code Number 668
Country Direct Information DB 512
Current Broadcast Connections 669

D

Data Block ID 671
Day 672
DB Byte Count 670
Directory Number 672
Directory Number DB 515
Directory Number Digit Count 673
Directory Number Error Response 297
Directory Number Information DB 519
Directory Number Request 194, 292
Directory Number Success Response 295
Directory Number Type 673
Directory Number Update DB 521
DTMF Digit 670
DTMF Digit DB 514
DTMF Digit Detected Inform 197, 284, 288
DTMF Receiver 671
DTMF Receiver Release Error Response 291
DTMF Receiver Release Request 197, 289
DTMF Receiver Release Success Response 290

E

Elapsed Minutes 674
End Call Error Response 302
End Call Handling DB 524
End Call Request 198, 202, 299
End Call Success Response 301
Error ID 675
ession Pool State Inform 439
Estimate Duration 676
Estimate of Charges Alternate Date DB 525
Estimate of Charges Alternate Time DB 526
Estimate of Charges Duration DB 527
Estimate of Charges Duration Information DB 528
Estimate of Charges Error Response 305
Estimate of Charges General Information DB 529
Estimate of Charges No Duration Information DB 530
Estimate of Charges Request 199, 303
Estimate of Charges Success Response 304
Estimated Charges 675

F

Float Connection Indicator 677
Function DB 532
Function ID 677
Function Provider Request Status 678

G

Generate AMA DB 533
Generate AMA Record Error Response 308
Generate AMA Record Request 181, 216, 306
Generate AMA Record Success Response 307
Generate AMA Sequence Number 679

H

Handling Data 679
Hotel Room Number 680
Hotel Room Number Length 681
Hour 681

I

Information ID DB 536
Invalid Billing Details DB 538
Invalid Billing Set 682
Invoke ID 683
ISUP Calling Party Number DB 534
ISUP Calling Party Number Update Error Response 312
ISUP Calling Party Number Update Request 194, 309
ISUP Calling Party Number Update Success Response 311

L

Language 686
Language DB 544
LATA Information DB 539
LATA Restrictions 683
LATA Status 684
LNP Enhancements DB 542
LNP Error Response 315
LNP Information DB 540
LNP Query 684

LNP Request 194, 313
LNP Success Response 314
Logical Voice Channel ID 686
LRN 685
LRN Digit Count 686
LRN Update DB 543
LRN Update Error Response 318
LRN Update Request 194, 316
LRN Update Success Response 317

M

Maintenance State 687
Maintenance State DB 545
Maintenance Text 688
Maintenance Text DB 546
Maintenance Text Length 688
Minutes 689
MIS Sequence Number ID 689
Month 690

N

NBEC Code 690
NBEC Code Digit Count 691
NBEC DB 547
Network ID 691
Network Service DB 548
Network Service ID 692
No Answer Time 693
Node Alarm Inform 411
Node ID 693
Node Identification DB 549
Node Log Inform 419
Node Release Inform 205, 319
Notification Time Type 694
Notification Timer Action 694
Notification Timer DB 550
Notification Timer Error Response 324
Notification Timer Inform 320
Notification Timer Mode 695
Notification Timer Request 208, 321
Notification Timer Status 696
Notification Timer Success Response 323
Number of Operator to SN Handoffs 696

O

OLNS Account Owner 697
OLNS Account Owner Length 697
OLNS Alphanumeric String 698
OLNS Alphanumeric String Length 698
OLNS Billing Service Provider 699
OLNS Billing Service Provider Length 699
OLNS DA Credit Card Indicator 700
OLNS DA Sent Paid Indicator 700
OLNS DA Special BNS Indicator 701
OLNS DA Third Number Indicator 701
OLNS DACC Indicator 702
OLNS Foreign Language 703
OLNS Information DB 552
OLNS InterLATA Carrier Identification Code 706
OLNS InterLATA Carrier Identification Digit Count 706
OLNS InterLATA Carrier Indicator 705
OLNS International Carrier Identification Code 707
OLNS International Carrier Identification Digit Count 707
OLNS International Carrier Indicator 706
OLNS IntraLATA Toll Carrier Identification Code 709
OLNS IntraLATA Toll Carrier Identification Digit Count 709
OLNS IntraLATA Toll Carrier Indicator 708
OLNS Service/Equipment Indicator 710
OLNS TA Collect Indicator 711
OLNS TA Credit Card Indicator 712
OLNS TA Sent Paid Indicator 713
OLNS TA Special BNS Indicator 713
OLNS TA Third Number Indicator 714
OLNS Treatment Indicator 714
Operation ID 716
Operator ID 717
Operator Identification DB 573
Operator Services System Action 717
Operator Services System Action DB 574
Operator System DB 575
Operator System Selector 718
Operator System Type 719
Originating Station Information DB 576

Originating Trunk Information DB 578
 OSSAIN Call Queue Event DB 561
 OSSAIN Non-Call Session Event DB 564
 OSSAIN Session Event DB 566

P

Party 720
 Pass Control Error Response 327
 Pass Control Request 205, 325
 Pass Control Success Response 326
 Pass Through Data 720
 Pass Through Data Length 721
 Pass Through DB 579
 Pass Through Error Response 332
 Pass Through Inform 205, 328
 Pass Through Request 205, 329
 Pass Through Success Response 331
 Passive Function Provider Error Response 336
 Passive Function Provider Request 205, 333
 Passive Function Provider Success Response 335
 Passive Node Status 721
 Passive Node Status Inform 205, 337
 Phone Class Type 722
 Portability Status 724
 Predicted Wait Time 725
 Preprocessing Information DB 583
 Presentation Indicator 726
 Protocol Version 726

Q

Queue Peg Indicators 727
 Queuing Event 728
 Queuing System 728

R

RAO Code 729
 RAO Digit Count 730
 Release Cause DB 586
 Release DN Error Response 340
 Release DN Request 194, 196, 338
 Release DN Success Response 339
 Release Node Error Response 343
 Release Node Indicator 730

Release Node Request 205, 341
 Release Node Success Response 342
 Report Text 731
 Report Text DB 588
 Report Text Length 732
 Requested RLT Action 732
 Requested Service DB 589
 Reset Time Flag 733
 Restart Severity 733
 Restricted Billing Index 734
 Restricted Input Mode 183, 194, 276, 293, 338
 Return Answer Message Error Response 346
 Return Answer Message Request 211, 344
 Return Answer Message Success Response 345
 RLT Request DB 585
 Route to Treatment Error Response 349
 Route To Treatment Request 190, 194, 201
 Route to Treatment Request 202, 347
 Route To Treatment Success Response 348
 Routing 735
 Routing Index 735
 Routing Index DB 590
 Routing Number DB 591

S

Seconds 740
 Sequence Number 740
 Sequence Number Check DB 594
 Service Change Error Response 353
 Service Change Request 200, 350
 Service Change Success Response 352
 Service Node Mode 741
 Service Peg Indicators 742
 Service Provider ID 742
 Service Provider ID Assignment Error Response 356
 Service Provider ID Assignment Request 354
 Service Provider ID Assignment Success Response 355
 Service Provider ID Length 743
 Service Provider Party Identifier 743
 Service Type 744
 Service Type Provided by Switch 745
 Serving Agent Status 745

Session Begin Inform 202, 230, 300, 357, 385
Session ID 746
Session Information DB 595
Session Initiation Error Response 366
Session Initiation Request 203, 362
Session Initiation Success Response 364
Session Origination Type 747
Session Pool Alarm Inform 427
Session Pool Drain Inform 434
Session Pool ID 747
Session Pool Log Inform 435
Session Pool Throttle Error Response 445
Session Pool Throttle Request 443
Session Pool Throttle Success Response 444
Session Recall Error Response 372
Session Recall Request 367
Session Recall Success Response 368
Session State 748
Session Type 749
SN Voice Conferencing Status 735
SN Voice Link DB 592
SN Voice Link Status 736
Solicitor Number 749
Special Number Format 750
Special Number Type 750
Special Number Validity 751
Speech Path DB 596
Speech Path Error Response 375
Speech Path Request 206, 373
Speech Path Status 752
Speech Path Success Response 374
SPID DB 593
SPID Processing 736
SS7 Backward Call Indicators 737
SS7 Release Cause 740

T

Tenths of Seconds Timestamp 752
Tenths Seconds 753
Terminal ID 753
Terminating Code Allowed Set 754
Terminating Trunk Information 597
Text Block 755
Text Block Length 756
Text DB 598

Text To Operator Error Response 378
Text To Operator Request 207, 376
Text To Operator Success Response 377
Throttle DB 599
Time DB 600
Time in Queue 756
Transfer Control List DB 602
Transfer to Carrier Error Response 382
Transfer to Carrier Request 209, 379
Transfer to Carrier Success Response 381
Transfer to Control List Error Response 387
Transfer to Control List Request 202, 209, 384
Transfer To Control List Success Response 386
Treatment 757
Treatment DB 603
Trigger Detection Point 757
Trigger DTMF DB 604
Trigger Event 758
Trigger Event Inform 210, 388
Trigger Information DB 605
Trigger Profile Index 758
Trigger Profile Index DB 607
Trigger Release Cause DB 609
Trigger Ringing DB 610
Trigger Treatment DB 611
Trunk Group ID 759
Trunk Member ID 760
Type of Service Provider 760

U

Unenforced Restrictions 761

V

Validation Request 761
Validation Result 762
Validation Result Determined by DMS Datafill 763
Validation Sequence Number 764
Voice Channel DB 612
Voice Channel ID 764
Voice Connect Error Response 396
Voice Connect Request 211, 392
Voice Connect Success Response 395
Voice Release Error Response 401

Voice Release Inform 211, 398
Voice Release Request 211, 399
Voice Release Success Response 400

Y

Year 765