NORTEL NETWORKS

Passport 7400, 15000, 20000

Configuring IP

241-5701-810

Passport 7400, 15000, 20000

Configuring IP

Publication: 241-5701-810
Document status: Standard
Document version: 5.2S2
Document date: December 2003

Printed in Canada

Publication history

December 2003

5.2S2 Standard
General availability. Contains information on Passport 7400, Passport 15000, and Passport 20000 for the PCR5.2 GA release.

Contents

About this document **25**
Who should read this document and why 25
What you need to know 26
How this document is organized 26
What's new in this document 27
IP multicast 28
Passport IP differentiated services for the gigabit Ethernet FP 28
Passport virtual router redundancy protocol 28
Text conventions 29
Procedure conventions 29
Operational mode 30
Provisioning mode 30
Activating configuration changes 31
Related documents 32
Passport documents 32
Request for comments (RFCs) 33
How to get more help 35

Chapter 1
IP configuration work flow **37**

Chapter 2
ATM MPE configuration for IP over ATM **41**
Prerequisites to ATM MPE configuration 41
ATM MPE configuration task flow 41
Configuring an ATM MPE interface for IP traffic 43
Configuring an ATM PVC for an ATM MPE interface 45

Configuring an ATM soft PVC for an ATM MPE interface 48

Chapter 3
Frame relay DTE configuration for IP over frame relay 53

Prerequisites to frame relay DTE configuration 53
Frame relay DTE configuration task flow 53
Configuring a frame relay DTE interface for IP traffic 56
Configuring a physical (hairpin) connection for a frame relay DTE interface 57
Configuring a logical connection for a frame relay DTE interface 60
Configuring a direct connection for a frame relay DTE interface 63
Customizing the Frame Relay link emission queue (Leq) component 66
Customizing the data link connection identifier (Dlci) component 68
Customizing the Hq subcomponent 70

Chapter 4
IP-optimized DLCI configuration for IP over frame relay 71

Prerequisites to IP-optimized DLCI configuration 71
IP-optimized DLCI configuration task flow 72
Configuring an IP-optimized DLCI 74
Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface 76

Chapter 5
Frame relay DTE to IP-optimized DLCI migration 79

Prerequisites to frame relay DTE to IP-optimized DLCI migration 79
Frame relay DTE to IP-optimized DLCI migration task flow 79
Migrating from a physical (hairpin) or logical connection with service interruption 81
Migrating from a physical (hairpin) or logical connection without service interruption 84
Migrating from a direct connection 87

Chapter 6
Gigabit Ethernet configuration for IP over GigE 89
Prerequisites to gigabit Ethernet configuration 89
Gigabit Ethernet configuration task flow 90
Configuring an Ethernet interface for IP traffic 91

Chapter 7
Point-to-point protocol configuration for IP over PPP 93
Prerequisites to PPP configuration 93
PPP configuration task flow 93
Configuring a PPP interface for IP traffic 96
Configuring a hardware connection for a PPP interface 97
Customizing the Link component 99
Customizing the link quality monitor (Lqm) component 102
Customizing the link emission queue (Leq) component 103

Chapter 8
Configuring PPP/ATM interworking for Passport 7400 105

Chapter 9
Virtual media configuration 109
Virtual media configuration task flow 109
Adding a virtual media component 111
Configuring connectivity between virtual routers 113

Chapter 10
IP capabilities configuration on Passport 115
Prerequisites to IP capabilities configuration 115
IP capabilities configuration task flow 115
Configuring a virtual router 118
Configuring IP on a virtual router 120
Configuring and linking a protocol port to a media interface 122
Enabling IP on a protocol port 124
Associating a single IP logical interface with a single subconnection 126

Chapter 11
Routing information protocol (RIP) configuration 129
RIP configuration task flow 129
Configuring a routing information protocol (RIP) interface 132
Configuring RIP import policy 135
Configuring RIP export policy 137
Configuring an always-up IP interface for RIP 140
Migrating from RIPv1 to RIPv2 142

Chapter 12
Open shortest path first (OSPF) configuration 147
OSPF configuration task flow 147
Adding OSPF to a virtual router 150
Configuring OSPF virtual links 153
Configuring an OSPF interface 155
Configuring OSPF export policy 159
Configuring an always-up IP interface for OSPF 162

Chapter 13
Intermediate System to Intermediate System (ISIS) configuration 165
ISIS configuration task flow 165
Configuring ISIS routing protocol 167
Configuring the ISIS interface 170

Chapter 14
Migrating from RIP to OSPF 173

Chapter 15
BGP-4 configuration 175
Prerequisites to BGP-4 configuration 175
BGP-4 configuration task flow 175
Configuring a BGP-4 instance 178
Configuring a BGP-4 peer 180
Configuring BGP-4 import policy 184
Configuring BGP-4 export policy 189
Configuring BGP-4 AS weight policy 192

Configuring BGP-4 aggregate policy 193
Configuring AS path attributes for export policy 196
Configuring multi-exit discrimination for export policy 198
Inserting a dummy AS in update messages 200
Specifying destination networks for export policy 201
Configuring an always-up IP interface for BGP-4 203

Chapter 16
Configuring static ARP 205

Chapter 17
Static route configuration 209
Static routes configuration task flow 209
Defining static routes 211
Configuring route entry discard for specific destinations 214

Chapter 18
Configuring bootstrap protocol 215

Chapter 19
Configuring IP multicast 219
IP multicast configuration task flow 219
Configuring IP multicast 222
Configuring a policy for a multicast group 223
Configuring IP multicast using static routes 225
Configuring IP multicast using IGMP 226
Configuring IP multicast using PIM-SM 227
Linking a candidate RP router and a candidate BSR router to an always-up logical interface 229

Chapter 20
Virtual router redundancy protocol configuration 233
Virtual router redundancy protocol configuration task flow 233
Adding the VRRP feature 236
Enabling VRRP on the participating Passport VRs 237
Creating a VRRP virtual router 238
Setting the VRRP advertisement interval 240
Defining critical IP interfaces 241

Chapter 21
IP class of service (CoS) configuration 243
Prerequisites to IP CoS configuration 243
IP CoS configuration task flow 243
Configuring IP CoS policy groups 246
Configuring packet classification policies 249
Configuring IP CoS for frame relay DTE 252
Configuring IP CoS for IP-optimized DLCI 254
Configuring IP CoS for ATM MPE 257
Configuring IP CoS for Ethernet 259
Configuring IP CoS for point-to-point protocol (PPP) 261
Activating the CoS policy group family on the IpPort 262

Chapter 22
IP DiffServ configuration 265
Prerequisites to IP DiffServ configuration 265
IP DiffServ configuration work flow 265
IP DiffServ domain configuration 267
Prerequisites to IP DiffServ domain configuration 267
IP DiffServ domain configuration task flow 267
IP DiffServ interface configuration 270
Prerequisites to IP DiffServ interface configuration 270
IP DiffServ interface configuration task flow 271
Adding a DiffServ domain to the virtual router 274
Modifying per-hop behaviors 275
Modifying DSCP marking for locally generated packets 277
Configuring an ATM link for IP DiffServ 279
Configuring an Ethernet link for IP DiffServ 281
Adding an IP DiffServ interface profile to a virtual router 283
Configuring ingress DSCP translation 285
Configuring egress DSCP translation 287
Modifying IP DiffServ drop precedence mode 289
Activating the IP DiffServ interface on the IpPort 291

Chapter 23
IP CoS to IP DiffServ migration 293
IP CoS to IP DiffServ migration task flow 294
Preparing IP DiffServ interface profiles for IP CoS migration 297
Activating IP DiffServ interface profiles for IP CoS migration 300
Removing IP CoS 302

Chapter 24
Configuring IP flow filters 305

Chapter 25
Configuring point-to-point tunnels 309

Chapter 26
IP monitoring and testing 313
Monitoring the ATM MPE configuration 314
ATM MPE component states 314
ATM MPE soft PVC component states 315
Monitoring the AtmMpe component 315
Monitoring the AtmConnection subcomponent 316
Monitoring the IlsFwdr component 317
Testing ATM MPE soft PVC connectivity 317
Testing ATM MPE soft PVC data flow 318
Clearing or optimizing an ATM MPE soft PVC 319
Monitoring the frame relay DTE configuration 321
Frame relay DTE component states 321
Frame relay DTE remote group (Rg) component states 322
Frame relay DTE data link connection identifier (Dlci) component states 323
Monitoring the FrDte component 324
Monitoring the StDlci subcomponent 325
Monitoring the Rg subcomponent 325
Monitoring the PPP configuration 327
PPP component states 327
Monitoring the Ppp component 328
Monitoring the Link subcomponent 328
Monitoring the Lqm subcomponent 329

Monitoring the Leq subcomponent 330
Monitoring the IP and virtual router configuration 331
Monitoring the IP component 331
Monitoring the IpInterfaceEntry subcomponent 333
Monitoring the IP cache subcomponent 333
Monitoring the ICMP subcomponent 334
Testing connectivity using the ICMP subcomponent 335
Monitoring the TCP subcomponent 337
Monitoring the UDP subcomponent 338
Monitoring the IpPort component 338
Monitoring the Arp subcomponent 339
Monitoring the BootpPort component 340
Monitoring the RelayBroadCast subcomponent 341
Monitoring the IP routing management configuration 342
Monitoring the virtual media configuration 345
Monitoring the RIP configuration 347
Monitoring the OSPF configuration 349
Monitoring the BGP-4 configuration 353
Monitoring the static route configuration 355
Monitoring the IP multicast configuration 356
Monitoring the virtual router redundancy protocol configuration 358
Displaying VRRP operational information 358
Locking and unlocking the VRRP component 358
Monitoring the IP CoS configuration 360
Using the ping command with IP CoS 360
Monitoring the IP CoS configuration 361
Monitoring IP DiffServ configuration 362
Confirming IP DiffServ interface profile usage 363
Confirming IP DiffServ interface profile configuration 365
Confirming connection class of connected media 367
Displaying per-hop behaviors 369
Pinging ICMP with IP DiffServ 371
Monitoring the IP flow filters configuration 373
Monitoring the filter component 373
Monitoring the filterFlow subcomponent 373

Monitoring the IP tunnel configuration 375

Chapter 27
Troubleshooting **379**

Troubleshooting ATM MPE 380
Troubleshooting frame relay DTE 383
Troubleshooting PPP 384
Troubleshooting PPP/ATM interworking 388

List of figures

Figure 1 IP configuration work flow 38
Figure 2 ATM MPE configuration task flow 42
Figure 3 Configuring an ATM MPE interface for IP traffic component hierarchy 44
Figure 4 Configuring an ATM PVC for an ATM MPE interface component hierarchy 47
Figure 5 Configuring an ATM soft PVC for an ATM MPE interface component hierarchy 51
Figure 6 Frame relay DTE configuration task flow 54
Figure 7 Configuring a physical (hairpin) connection for a frame relay DTE interface 59
Figure 8 Configuring a logical connection for a frame relay DTE interface 62
Figure 9 Configuring a direct connection for a frame relay DTE interface 65
Figure 10 IP-optimized DLCI configuration task flow 72
Figure 11 Configuring an IP-optimized DLCI component hierarchy 75
Figure 12 Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface component hierarchy 77
Figure 13 Frame relay DTE to IP-optimized DLCI migration task flow 80
Figure 14 Migrating from a physical (hairpin) or logical connection with service interruption 83
Figure 15 Migrating from a physical (hairpin) or logical connection without service interruption 86
Figure 16 Migrating from a direct connection 88
Figure 17 Gigabit Ethernet configuration task flow 90
Figure 18 Configuring an Ethernet interface for IP traffic component hierarchy 92
Figure 19 Point-to-point protocol configuration for task flow 94
Figure 20 Configuring a hardware connection for a PPP interface component hierarchy 98
Figure 21 Configuring PPP/ATM interworking for Passport 7400 component hierarchy 108
Figure 22 Virtual media configuration task flow 110
Figure 23 Configuring virtual media component hierarchy 112
Figure 24 Configuring connectivity between virtual routers component hierarchy 114

Figure 25 IP capabilities configuration task flow 116
Figure 26 Configuring a virtual router component hierarchy 119
Figure 27 Configuring IP on a virtual router component hierarchy 121
Figure 28 Configuring and linking protocol ports to the media interface component hierarchy 123
Figure 29 Enabling IP on a protocol port component hierarchy 125
Figure 30 Associating a single IP logical interface with a single subconnection component hierarchy 127
Figure 31 RIP configuration task flow 130
Figure 32 Configuring a RIP interface component hierarchy 134
Figure 33 Configuring RIP import policy component hierarchy 136
Figure 34 Configuring RIP export policy component hierarchy 139
Figure 35 Configuring an always-up IP interface for RIP component hierarchy 141
Figure 36 Example migration from RIPv1 to RIPv2 using two Passport nodes 143
Figure 37 OSPF configuration task flow 148
Figure 38 Adding OSPF to a virtual router component hierarchy 152
Figure 39 Configuring OSPF virtual links component hierarchy 154
Figure 40 Configuring an OSPF interface component hierarchy 158
Figure 41 Configuring OSPF export policy component hierarchy 161
Figure 42 Configuring an always-up IP interface for OSPF component hierarchy 163
Figure 43 Configuring ISIS task flow 166
Figure 44 Configuring ISIS routing protocol 169
Figure 45 Configuring ISIS Interface 171
Figure 46 BGP-4 configuration task flow 176
Figure 47 Configuring a BGP-4 instance component hierarchy 179
Figure 48 Configuring a BGP-4 peer component hierarchy 183
Figure 49 Configuring a BGP-4 import policy component hierarchy 188

Figure 50 Configuring BGP-4 export policy component hierarchy 191
Figure 51 Configuring BGP-4 AS weight policy component hierarchy 192
Figure 52 Configuring BGP-4 aggregate policy component hierarchy 195
Figure 53 Configuring AS path attributes for export policy component hierarchy 197
Figure 54 Configuring multi-exit discrimination for export policy component hierarchy 199
Figure 55 Inserting a dummy AS in update messages component hierarchy 200
Figure 56 Specifying destination networks for export policy component hierarchy 202
Figure 57 Configuring an always-up IP interface for BGP-4 component hierarchy 204
Figure 58 Configuring static ARP component hierarchy 208
Figure 59 Static route configuration task flow 210
Figure 60 Defining static routes component hierarchy 213
Figure 61 Configuring route entry discard for specific destinations component hierarchy 214
Figure 62 Configuring bootstrap protocol component hierarchy 217
Figure 63 IP multicast configuration task flow 220
Figure 64 Configuring IP multicast component hierarchy 231
Figure 65 VRRP configuration task flow 234
Figure 66 VRRP associated components 242
Figure 67 IP CoS configuration task flow 244
Figure 68 Configuring IP CoS policy groups component hierarchy 248
Figure 69 Configuring IP CoS packet classification policies 251
Figure 70 Configuring IP CoS for frame relay DTE component hierarchy 253
Figure 71 Configuring IP CoS for IP-optimized DLCI component hierarchy 256
Figure 72 Configuring IP CoS for ATM MPE component hierarchy 258
Figure 73 Configuring IP CoS for gigabit Ethernet component hierarchy 260
Figure 74 Configuring IP CoS for PPP component hierarchy 261

Figure 75 Activating the CoS policy group family on the IpPort component hierarchy 263
Figure 76 IP DiffServ configuration work flow 266
Figure 77 IP DiffServ domain configuration task flow 268
Figure 78 Adding a DiffServ domain to the virtual router component hierarchy 274
Figure 79 Modifying per-hop-behaviors component hierarchy 276
Figure 80 Modifying DCSP marking for locally generated packets component hierarchy 278
Figure 81 Configuring an ATM link for IP DiffServ component hierarchy 280
Figure 82 Configuring an Ethernet link for IP DiffServ component hierarchy 282
Figure 83 Adding IP DiffServ interface profile to a Vr component hierarchy 284
Figure 84 Configuring ingress DSCP translation component hierarchy 286
Figure 85 Configuring egress DSCP translation component hierarchy 288
Figure 86 Configuring IP DiffServ drop precedence component hierarchy 290
Figure 87 Activating the IP DiffServ interface on the IpPort component hierarchy 292
Figure 88 IP CoS to IP DiffServ migration task flow 295
Figure 89 Preparing IP DiffServ interface profiles for IP CoS migration component hierarchy 299
Figure 90 Activating IP DiffServ interface profiles for IP CoS migration component hierarchy 301
Figure 91 Removing IP CoS component hierarchy 303
Figure 92 Configuring IP flow filters component hierarchy 308
Figure 93 Configuring point-to-point tunnels component hierarchy 311
Figure 94 Confirming IP DiffServ interface profile usage component hierarchy 364
Figure 95 Confirming DiffServ configuration component hierarchy 366
Figure 96 Confirming connection class of connected media component hierarchy 368
Figure 97 Displaying per-hop-behaviors component hierarchy 370

Figure 98 Pinging ICMP with IP DiffServ component hierarchy 372

List of tables

Table 1 Customizing the FrDte Leq subcomponent 66
Table 2 Customizing the FrDte Dlci subcomponent 68
Table 3 Customizing the Link component 99
Table 4 Customizing the Lqm component 102
Table 5 Customizing the Leq component 103
Table 6 Example migration: RIP behavior on two Passport nodes with different RIP configuration 144
Table 7 ATM MPE component states 314
Table 8 AtmConnection component states 315
Table 9 Monitoring the AtmMpe component 316
Table 10 Monitoring the AtmConnection subcomponent 317
Table 11 Monitoring the IlsFwdr component 317
Table 12 Monitoring AtmMpe soft PVC connectivity 318
Table 13 Monitoring AtmMpe soft PVC data flow 319
Table 14 Clearing the AtmMpe soft PVC 320
Table 15 Frame relay DTE component states 321
Table 16 Frame relay DTE remote group (Rg) component states 322
Table 17 Frame relay DTE Dlci component states 323
Table 18 Monitoring the FrDte component 324
Table 19 Monitoring the StDlci subcomponent 325
Table 20 Monitoring the Rg subcomponent 325
Table 21 PPP component states 327
Table 22 Monitoring the Ppp component 328
Table 23 Monitoring the Link subcomponent 329
Table 24 Monitoring the Lqm subcomponent 329
Table 25 Monitoring the Leq subcomponent 330
Table 26 Locking and unlocking the IP component 332
Table 27 Monitoring the IP component 332
Table 28 Monitoring the IpInterfaceEntry subcomponent 333
Table 29 Monitoring the IP cache subcomponent 334
Table 30 Clearing IP cache table entries 334
Table 31 Monitoring the ICMP subcomponent 335
Table 32 Testing connectivity using the ICMP subcomponent 336
Table 33 Monitoring the TCP subcomponent 337
Table 34 Displaying the TCP connection table 337
Table 35 Monitoring the UDP subcomponent 338
Table 36 Displaying the UDP listen table 338

Table 37 Locking and unlocking the IpPort component 339
Table 38 Monitoring the IpPort component 339
Table 39 Monitoring the Arp subcomponent 340
Table 40 Clearing ARP table dynamic host entries 340
Table 41 Monitoring the BootpPort component 341
Table 42 Monitoring the RelayBroadCast subcomponent 341
Table 43 Monitoring the Ip ForwardTable component 342
Table 44 Monitoring the Ip RouteDataBaseEntry component 344
Table 45 Locking and unlocking the Vm If component 345
Table 46 Monitoring the Vm component 345
Table 47 Monitoring the Vm If component 346
Table 48 Locking and unlocking the Rip component 347
Table 49 Monitoring the Rip component 347
Table 50 Monitoring the Rip If component 348
Table 51 Monitoring RIP import and export policy 348
Table 52 Locking and unlocking the Ospf component 349
Table 53 Monitoring the Ospf component 350
Table 54 Monitoring the OspfIf component 350
Table 55 Monitoring OSPF export policy 350
Table 56 Monitoring OSPF areas 350
Table 57 Monitoring OSPF hosts 351
Table 58 Monitoring OSPF virtual links 351
Table 59 Monitoring OSPF stub areas 351
Table 60 Monitoring OSPF neighbors 351
Table 61 Monitoring the OSPF link state database 352
Table 62 Locking and unlocking the Bgp and Bgp Peer components 353
Table 63 Monitoring the Bgp and Bgp Peer components 353
Table 64 Monitoring routes in the BGP routing information base (RIB) 354
Table 65 Locking and unlocking the Ip Static component 355
Table 66 Monitoring the Ip Static component 355
Table 67 Monitoring IP multicast and IGMP 356
Table 68 Monitoring PIM-SM 357
Table 69 Displaying the VRRP operational information 358
Table 70 Locking and unlocking the VRRP component 359
Table 71 Using the ping command with IP CoS 360
Table 72 Monitoring IP CoS 361
Table 73 Monitoring the filter component 373
Table 74 Monitoring the filterFlow subcomponent 374
Table 75 Locking and unlocking the Tunnel component 375

Table 76 Monitoring the Tunnel component 376
Table 77 Monitoring the ProtocolPort component for an IP tunnel 376
Table 78 Monitoring the IpPort component for an IP tunnel 377
Table 79 Monitoring the LogicalIf component for an IP tunnel 377
Table 80 Handling problems with ATM MPE 380
Table 81 Handling problems with frame relay DTE 383
Table 82 Handling problems with PPP 384

About this document

This user guide describes how to configure virtual routers, the Internet protocol (IP), and other protocols and services related to IP on the Passport system.

The following topics are discussed in this section:

- "Who should read this document and why" (page 25)
- "What you need to know" (page 26)
- "How this document is organized" (page 26)
- "What's new in this document" (page 27)
- "Text conventions" (page 29)
- "Procedure conventions" (page 29)
- "Related documents" (page 32)
- "How to get more help" (page 35)

Who should read this document and why

This guide is for anyone who performs the following tasks for IP on the Passport system:

- planning
- installing and provisioning
- operating and maintaining

What you need to know

This guide assumes that you are familiar with the concepts of internetworking, particularly with the IP suite and its common uses.

See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information about IP on Passport.

How this document is organized

241-5701-810 *Passport 7400, 15000, 20000 Configuring IP* contains the following sections:

- "IP configuration work flow" (page 37)
- "ATM MPE configuration for IP over ATM" (page 41)
- "Frame relay DTE configuration for IP over frame relay" (page 53)
- "IP-optimized DLCI configuration for IP over frame relay" (page 71)
- "Frame relay DTE to IP-optimized DLCI migration" (page 79)
- "Gigabit Ethernet configuration for IP over GigE" (page 89)
- "Point-to-point protocol configuration for IP over PPP" (page 93)
- "Configuring PPP/ATM interworking for Passport 7400" (page 105)
- "IP capabilities configuration on Passport" (page 115)
- "Routing information protocol (RIP) configuration" (page 129)
- "Open shortest path first (OSPF) configuration" (page 147)
- "Intermediate System to Intermediate System (ISIS) configuration" (page 165)
- "Migrating from RIP to OSPF" (page 173)
- "BGP-4 configuration" (page 175)
- "Configuring static ARP" (page 205)
- "Static route configuration" (page 209)
- "Configuring bootstrap protocol" (page 215)
- "Configuring IP multicast" (page 219)

- “Virtual router redundancy protocol configuration” (page 233)
- “IP class of service (CoS) configuration” (page 243)
- “IP DiffServ configuration” (page 265)
- “IP CoS to IP DiffServ migration” (page 293)
- “Configuring IP flow filters” (page 305)
- “Configuring point-to-point tunnels” (page 309)
- “IP monitoring and testing” (page 313)
- “Troubleshooting” (page 379)

What’s new in this document

The following features were added to this document:

- “IP multicast” (page 28)
- “Passport IP differentiated services for the gigabit Ethernet FP” (page 28)
- “Passport virtual router redundancy protocol” (page 28)

Other changes made to this document include the following.

- The section “IP class of service (CoS) configuration” (page 243) was updated with a new procedure, “Activating the CoS policy group family on the IpPort” (page 262), and a change to the task flow, “IP CoS configuration task flow” (page 244).
- For CR Q00643576, the section “Prerequisites to IP capabilities configuration” (page 115) was updated.
- From the Release Notes 5.1.1, the section “Customizing the data link connection identifier (Dlci) component” (page 68) was updated to indicate that rate enforcement for FrDte only applies to PQC-based FPs and to indicate the affected attributes.
- The section “Configuring a logical connection for a frame relay DTE interface” (page 60) was updated with information about configuring virtual framers on 4-port DS3Ch and 1-port STM1Ch FPs.

- The procedures “Configuring IP CoS for ATM MPE” (page 257) and “Configuring an ATM link for IP DiffServ” (page 279) were updated with information about the behavior of the *AtmMpe Ac* component when a VCC connection is lost.
- In “IP class of service (CoS) configuration” (page 243) and “IP DiffServ configuration” (page 265), “discard priority” was changed to “drop precedence”.
- The figure “Adding a DiffServ domain to the virtual router component hierarchy” (page 274) was updated to show support for wireless UMTS.
- The sections “Modifying per-hop behaviors” (page 275) and “Modifying DSCP marking for locally generated packets” (page 277) were updated to reflect current component and attribute names.
- The section “Activating the IP DiffServ interface on the IpPort” (page 291) was updated to include information on when the procedure is required and to replace <assign_profile> with attribute *Vr Ip DiffServ linkToDiffServUsers*.

IP multicast

The following sections were added for this feature:

- “IP configuration work flow” (page 38)
- “Configuring IP multicast” (page 219)
- “Monitoring the IP multicast configuration” (page 356)

Passport IP differentiated services for the gigabit Ethernet FP

The following sections were updated for this feature:

- “Configuring an Ethernet link for IP DiffServ” (page 281)
- “Configuring egress DSCP translation” (page 287)

Passport virtual router redundancy protocol

The following sections were added for this feature:

- “IP configuration work flow” (page 38)
- “Virtual router redundancy protocol configuration” (page 233)

- "Monitoring the virtual router redundancy protocol configuration" (page 358)

Text conventions

This document uses the following text conventions:

- **`nonproportional spaced bold type`**

  Nonproportional spaced bold type represents words that you should type or that you should select on the screen.

- *italics*

  Words that appear in italics indicate a software component or attribute name.

- `[optional_parameter]`

  Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.

- `<general_term>`

  Words in angle brackets represent variables which are to be replaced with specific values.

Procedure conventions

This document uses the following procedure conventions:

- You can enter commands using full component and attribute names, or you can abbreviate them. The commands used in the procedures contain the full component and attribute names in the first instance. In the second instance, the component and attribute names are abbreviated. For more information on abbreviating component and attribute names, see 241-5701-060 *Passport 7400, 15000, 20000 Components*. All component and attribute names are formatted in italics.
- The introduction of every procedure states whether you must perform the procedure in operational mode or provisioning mode. For more information on these modes, see "Operational mode" (page 30) or "Provisioning mode" (page 30).

- When you complete a procedure, you can verify your changes and then activate them as the new node configuration. For more information on completing configuration changes and exiting provisioning mode, see “Activating configuration changes” (page 31).

Operational mode

Procedures contained within this document can either be performed in operational mode or provisioning mode. When you initially log into a Passport node, you are in operational mode. Passport uses the following command prompt when you are in operational mode:

```
#>
```

where:
is the current command number

In operational mode, you work with operational components and attributes. In operational mode, you can

- list operational components and display operational attributes to determine the current operating parameters for the node
- control the state of parts of the node by locking and unlocking components
- set certain operational attributes and enter commands to perform diagnostic tests

Provisioning mode

To change from operational mode to provisioning mode, type the following command at the operator prompt:

```
start Prov
```

Only one user can be in provisioning mode at a time. Passport uses the following command prompt whenever you are in provisioning mode:

```
PROV #>
```

where:

is the current command number

In provisioning mode, you work with the provisionable components and attributes that contain the current and future configurations of the node. You can add and delete components, and display and set provisionable attributes. For information on completing the configuration changes, exiting provisioning mode, and returning to operational mode see “Activating configuration changes” (page 31).

For information on operational and provisionable attributes, see 241-5701-060 *Passport 7400, 15000, 20000 Components*.

Activating configuration changes

Several procedures in this document ask that you complete the configuration changes. When you complete the configuration changes, you are activating the configuration changes, confirming that you want to activate them, and saving the changes. You are instructed to complete the configuration changes only at the end of procedures that you perform in provisioning mode.

CAUTION

Activating a provisioning view can affect service

Activating a provisioning view can result in a CP reload or restart, causing all services on the Passport node to fail. See 241-5701-050 *Passport 7400, 15000, 20000 Commands*, for more information.

1. Verify that the provisioning changes you have made are acceptable:

   ```
   check Prov
   ```

   Correct any errors and then verify the provisioning changes again.

2. If you want to store the provisioning changes in a file, save the provisioning view:

   ```
   save Prov
   ```

3. If you want these changes as well as other changes made in the edit view to take effect immediately, activate, confirm, and commit the provisioning changes:

   ```
   activate Prov
   ```

   ```
   confirm Prov
   ```

   ```
   commit Prov
   ```

4 End the provisioning session:

```
end Prov
```

Related documents

For the complete list of documents in the Passport documentation library, see 241-5701-001 *Passport 7400, 15000, 20000 Documentation Guide*.

The following sections contain documents related to the information in this guide:

- "Passport documents" (page 32)
- "Request for comments (RFCs)" (page 33)

Passport documents

The following documents containing information related to IP and the Passport system are available from Nortel Networks:

- 241-1001-506 *DPN-100 Alarm Console Indications*
- 241-5701-001 *Passport 7400, 15000, 20000 Documentation Guide*
- 241-5701-005 *Passport 7400, 15000, 20000 List of Terms*
- 241-5701-030 *Passport 7400, 15000, 20000 Overview*
- 241-5701-050 *Passport 7400, 15000, 20000 Commands*
- 241-5701-060 *Passport 7400, 15000, 20000 Components*
- 241-5701-500 *Passport 6400, 7400, 15000, 20000 Alarms*
- 241-5701-600 *Passport 7400, 15000, 20000 Configuration Guide*
- 241-5701-615 *Passport 7400, 15000, 20000 FP Configuration Reference*
- 241-5701-700 *Passport 7400, 15000, 20000 ATM Overview*
- 241-5701-702 *Passport 7400, 15000, 20000 ATM Routing and Signaling Fundamentals*
- 241-5701-901 *Passport 7400, 15000, 20000 Frame Relay Fundamentals*
- 241-5701-902 *Passport 7400, 15000, 20000 Configuring Frame Relay*
- 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*

- 241-5701-581 *Passport 7400, 15000, 20000 Basics: VPN Fundamentals*
- 241-5701-582 *Passport 7400, 15000, 20000 VPN Configuration Management*
- 241-1501-200 *Passport 15000, 20000 Hardware Description*
- 241-1501-205 *Passport 15000, 20000 Site Requirements and Preparation Guide*
- 241-1501-210 *Passport 15000, 20000 Hardware Installation Guide*
- 241-1501-215 *Passport 15000, 20000 Hardware Maintenance Guide*
- 241-7401-200 *Passport 7400 Hardware Description*
- 241-7401-240 *Passport 7400 Hardware Installation, Maintenance and Upgrade*

Request for comments (RFCs)

The following Requests for Comments (RFCs) containing information related to IP are available from numerous sources including Internet Network Information Center (NIC) servers:

- RFC761, *DoD standard Transmission Control Protocol*
- RFC768, *User Datagram Protocol*
- RFC0791, *Internet Protocol*
- RFC792, *Internet Control Message Protocol*
- RFC793, *Transmission Control Protocol*
- RFC815, *IP Datagram Reassembly Algorithms*
- RFC821, *Simple Mail Transfer Protocol*
- RFC826, *An Ethernet Address Resolution Protocol*
- RFC854, *Telnet Protocol Specifications*
- RFC904, *Exterior Gateway Protocol Formal Specification*
- RFC950, *Internet Standard Subnetting Procedure*
- RFC951, *Bootstrap Protocol (BootP)*
- RFC959, *File Transfer Protocol*

- RFC1009, *Requirements for Internet Gateways*
- RFC1038, *Draft Revised IP Security Option*
- RFC1042, *Standard for Transmission of IP Datagrams over IEEE 802 Networks*
- RFC1122, *Requirements for Internet Hosts - Communication Layers*
- RFC1157, *Management Information Base for Network Management of TCP/IP-based Internets*
- RFC1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
- RFC1253, *OSPF Version 2 Management Information Base*
- RFC1354, *IP Forwarding Table MIB*
- RFC1517, *Applicability Statement For the Implementation of Classless Inter-Domain Routing (CIDR)*
- RFC1518, *An Architecture for IP Address Allocation with CIDR*
- RFC1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*
- RFC1541, *Dynamic Host Configuration Protocol*
- RFC1577, *Classical IP and ARP over ATM*
- RFC1583, *OSPF Version 2*
- RFC1657, *Border Gateway Protocol version 4 (BGP-4) MIB*
- RFC1701, *Generic Routing Encapsulation*
- RFC1702, *Generic Routing Encapsulation over IPv4 networks*
- RFC1723, *RIP Version 2 Carrying Additional Information*
- RFC1724, *RIP Version 2 MIB Extension*
- RFC1745, *BGP4/IDRP for IP-OSFP Interaction*
- RFC1771, *Border Gateway Protocol 4 (BGP-4*)
- RFC1772, *Application of the Border Gateway Protocol in the Internet*
- RFC2003, *IP Encapsulation within IP*

- RFC2334, *Server Cache Synchronization Protocol*
- RFC2474, *DiffServ Field Definition*
- RFC2597, *Assured Forwarding PHB Group*
- RFC3246, *An Expedited Forwarding PHB*

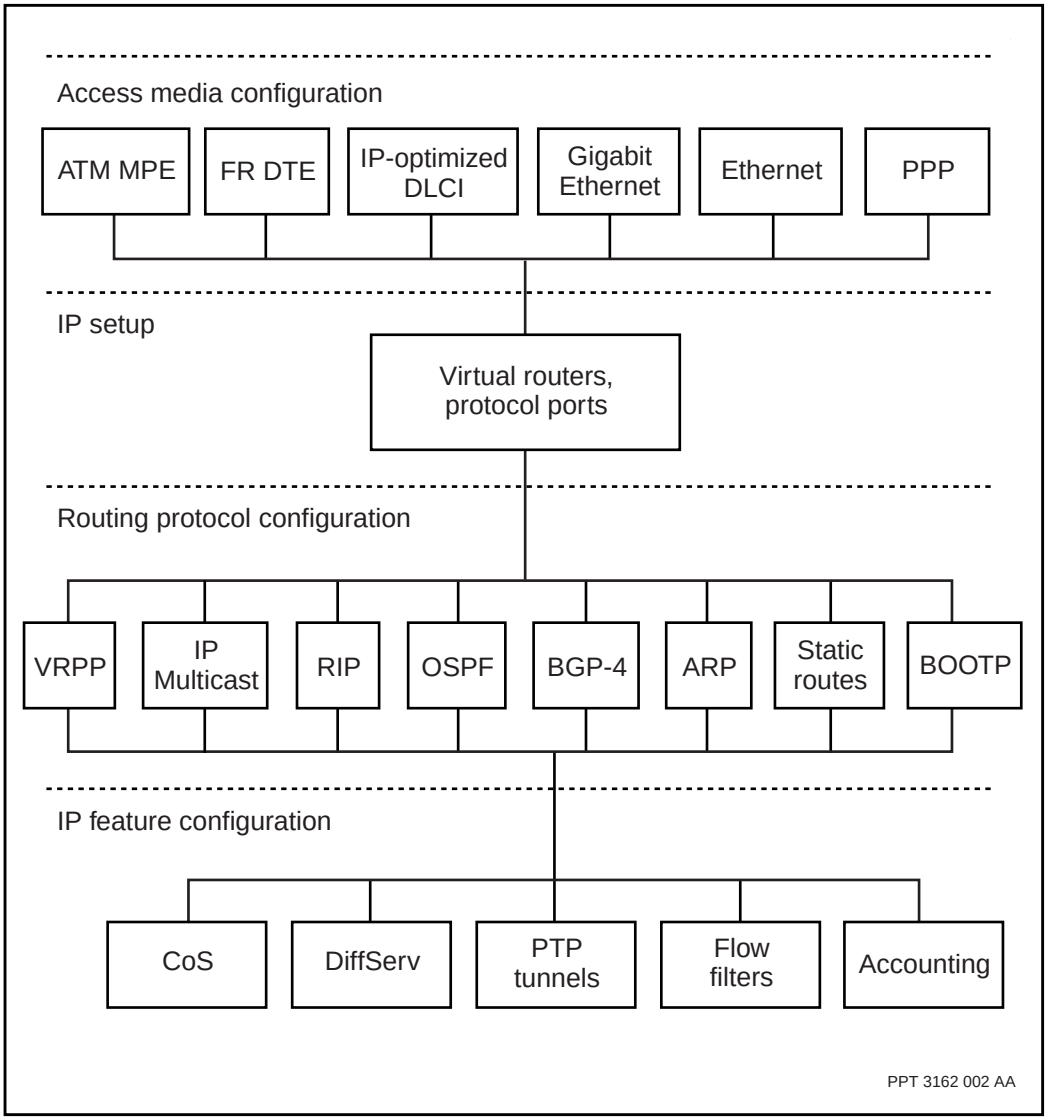
How to get more help

For information on training, problem reporting, and technical support, see the "Nortel Networks works support services" section in the product overview document.

Chapter 1
IP configuration work flow

For a detailed view of the sequence of tasks you perform to configure IP on Passport see the figure “IP configuration work flow” (page 38). Each box in the work flow represents a task that comprises one or more procedures. Each task has a corresponding section in this guide that contains the relevant procedures. To link to any task, go to “Navigation links” (page 38) following the task flow.

Figure 1
IP configuration work flow



Navigation links

- “ATM MPE configuration for IP over ATM” (page 41)

- “Frame relay DTE configuration for IP over frame relay” (page 53)
- “IP-optimized DLCI configuration for IP over frame relay” (page 71)
- “Gigabit Ethernet configuration for IP over GigE” (page 89)
- “Point-to-point protocol configuration for IP over PPP” (page 93)
- “Configuring PPP/ATM interworking for Passport 7400” (page 105)
- “IP capabilities configuration on Passport” (page 115)
- “Routing information protocol (RIP) configuration” (page 129)
- “Open shortest path first (OSPF) configuration” (page 147)
- “Migrating from RIP to OSPF” (page 173)
- “BGP-4 configuration” (page 175)
- “Configuring static ARP” (page 205)
- “Static route configuration” (page 209)
- “Configuring bootstrap protocol” (page 215)
- “Configuring IP multicast” (page 219)
- “Virtual router redundancy protocol configuration” (page 233)
- “IP class of service (CoS) configuration” (page 243)
- “IP DiffServ configuration” (page 265)
- “Configuring IP flow filters” (page 305)

Chapter 2
ATM MPE configuration for IP over ATM

Configure ATM MPE as a first step to enabling the Passport to carry IP over ATM.

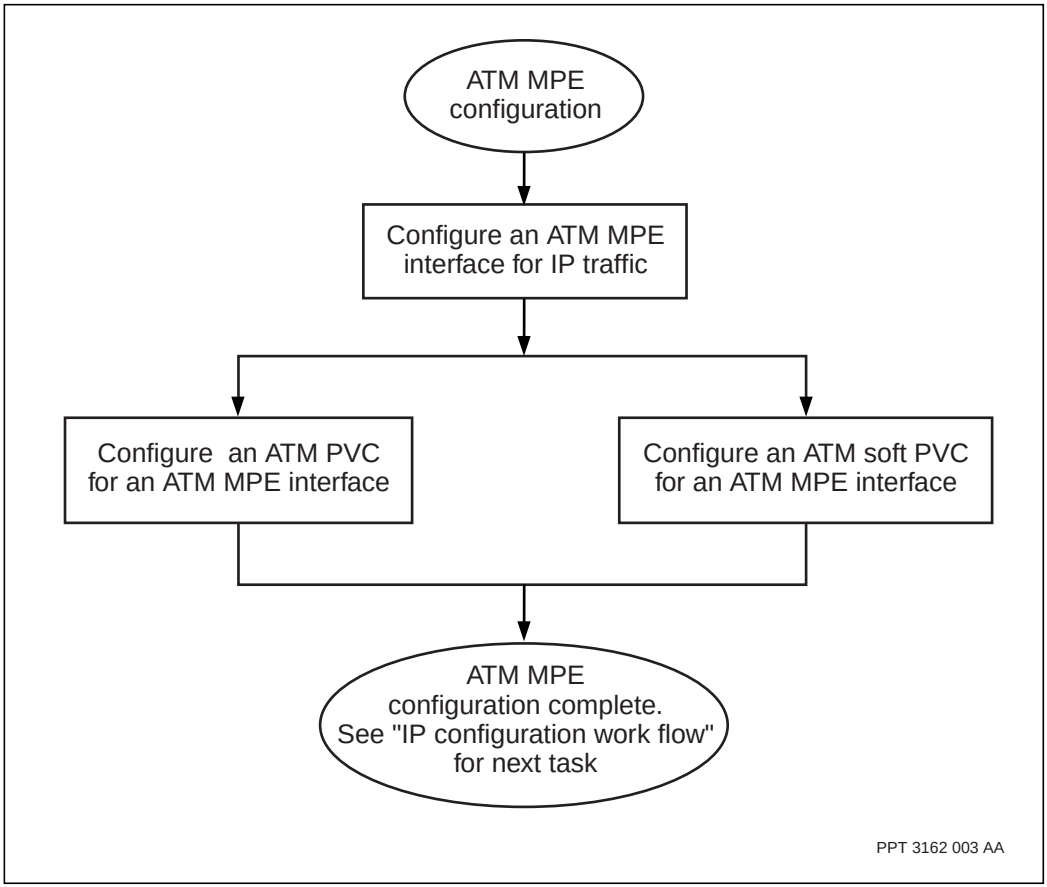
Prerequisites to ATM MPE configuration

- Configure the required ATM interfaces and connections. See 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.
- See the figure “IP configuration work flow” (page 38) to understand how ATM MPE fits into the overall IP configuration task flow.
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information.

ATM MPE configuration task flow

This task flow shows you the sequence of procedures you perform to configure ATM MPE. To link to any procedure, go to the list that follows the task flow.

Figure 2
ATM MPE configuration task flow



Navigation links

- “Configuring an ATM MPE interface for IP traffic” (page 43)
- “Configuring an ATM PVC for an ATM MPE interface” (page 45)
- “Configuring an ATM soft PVC for an ATM MPE interface” (page 48)
- For information about the next task, see “IP configuration work flow” (page 38)

Configuring an ATM MPE interface for IP traffic

Configure an ATM MPE interface for IP traffic to provide an ATM MPE connection between the Passport node and the IP network.

Prerequisites

- Configure the required ATM interfaces and connections. See 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.

Procedure steps

1 Create an instance of the ATM MPE service.

```
add AtmMpe/<n>
```

When you create the *AtmMpe* component, the Passport system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component.

2 Specify the maximum transmission unit (MTU) size to be used for ATM connections on this interface.

```
set AtmMpe/<n> mtu <size>
```

3 Specify the encapsulation type to be used for ATM connections on this interface.

If you set the encapsulation type to *ipVcEncap*, your must configure a static ARP entry to ensure IP connectivity across the ATM network. See “Configuring static ARP” (page 205).

The *ipVcEncap* attribute value is supported only for ATM MPE PVCs, not for soft PVCs.

```
set AtmMpe/<n> encapType <type>
```

Variable definitions

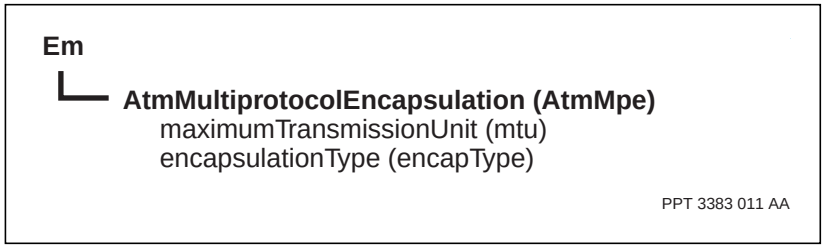
Variable	Value
<n>	The instance number of the ATM MPE interface. The maximum number of *AtmMpe* components you can provision as soft PVC interfaces is 256 on a single Passport node.

(Sheet 1 of 2)

Variable	Value
<size>	The size of the largest datagram that can be sent on the interface.
<type>	The encapsulation type defined for ATM connections on the interface.
(Sheet 2 of 2)	

Procedure job aid

Figure 3
Configuring an ATM MPE interface for IP traffic component hierarchy



Configuring an ATM PVC for an ATM MPE interface

Configure an ATM PVC for an ATM MPE interface to support full-mesh connectivity between VRs or virtual connection gateways (VCGs) using nailed-up connection points.

Prerequisites

- ATM MPE over PVCs is supported on CQC and PQC-based FPs.
- Each *AtmMpe* component must have at least one *AtmConnection* component associated with a VCC that connects to every other Passport node that uses the ATM MPE service.
- Each VCC (VPI.VCI) can be associated with the ATM interface or with a virtual path terminator (VPT). If the VCC is associated with a VPT, the VPI value is the instance of the *Vpt* component and the VCI value is the instance of the *Vcc* component. See 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.
- If you are using VC-based multiplexing for IP traffic, you must configure a static ARP entry for the ATM VCC to ensure IP connectivity across the ATM network. Since VC-based multiplexing supports traffic for only one protocol type (in this case, IP), ARP packets cannot transmit on the ATM MPE service. See “Configuring static ARP” (page 205).

Procedure steps

1 Optionally, create an instance of an ATM connection for the ATM MPE service.

```
add AtmMpe/<n> Ac/<conn>
```

When you create the *AtmMpe* component, the Passport system automatically adds an instance of the *AtmConnection* (*Ac*) component, *Ac/1*, under the *AtmMpe* component. Add the *Ac* component only when you need another connection after *Ac/1*.

Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.

2 Configure a VCC under an ATM interface, if one is not already available. This VCC is directly associated with the ATM MPE service and resides on the same Passport node.

```
add AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc>
```

3 Create a nailed-up endpoint (NEP) for the VCC, if one is not already available.

```
add AtmIf/<i> [Vpt/<Vpi>] Vcc/<vc> Nep
```

4 Link the ATM MPE service to the ATM VCC.

```
set AtmMpe/<n> Ac/<conn> atmConnection AtmIf/<i>
[Vpt/<Vpi>] Vcc/<vc> Nep
```

5 If you are using an ILS forwarder FP, link the ATM MPE service to the ILS forwarder card.

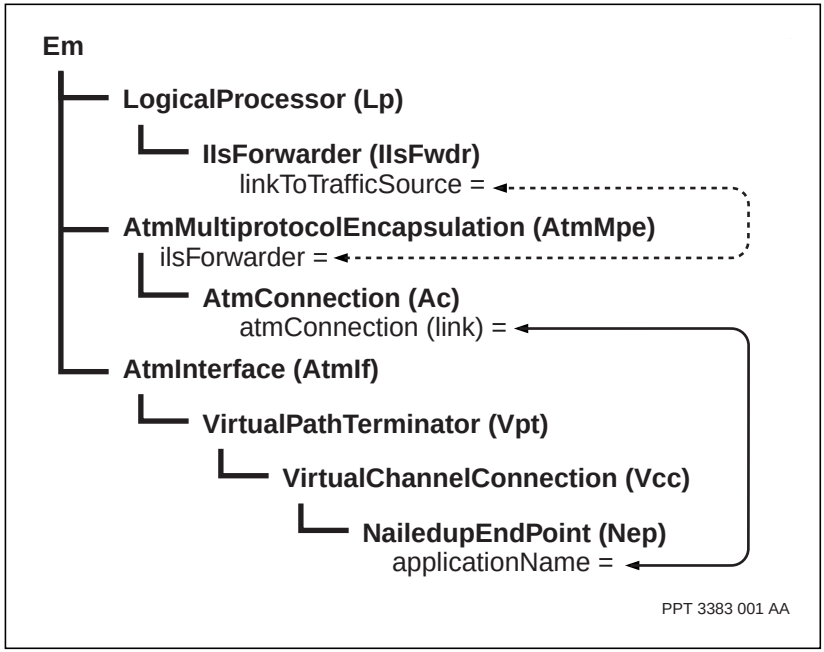
```
set AtmMpe/<n> ilsForwarder Lp/<z> IlsFwdr/<f>
```

Variable definitions

Variable	Value
<conn>	The instance number of the ATM connection on the ATM MPE interface.
<i>	The instance number of the ATM interface.
<vc>	The instance value of the VCC. If the virtual channel is associated with a VPT, this value is the VCI value.
[Vpt/<Vpi>]	The *VirtualPathTerminator* (*Vpt*) component instance.

Procedure job aid

Figure 4
Configuring an ATM PVC for an ATM MPE interface component hierarchy



Configuring an ATM soft PVC for an ATM MPE interface

Configure an ATM soft PVC for an ATM MPE interface to support full-mesh connectivity between customer VRs in a PNNI network, or virtual connection gateways (VCGs) across the backbone.

Prerequisites

- ATM MPE over soft PVCs is supported on PQC-based FPs.
- Configure the required ATM routing components. 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.
- Configure the *Pnni* component under the *AtmRouting* and *AtmInterface* components. See 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.

Procedure steps

1 Optionally, create an instance of an ATM connection for the ATM MPE service.

```
add AtmMpe/<n> Ac/<conn>
```

When you create the *AtmMpe* component, the Passport system automatically adds an instance of the *AtmConnection* (*Ac*) component (*Ac/1*) under the *AtmMpe* component. Add the *Ac* component only when you need another connection after *Ac/1*.

Do not change the attribute *AtmMpe Ac mplsSig* from its default value of *shared*. Setting this attribute to *dedicated* dedicates the connection to MPLS signalling.

2 Create the soft PVC termination point.

```
add AtmMpe/<n> Stp
```

3 If this is the calling end of the soft PVC, define the retry period. This timer determines how long the calling end should wait after failure before making the next setup request.

```
set AtmMpe/<n> Stp retry <period>
```

If you have several ATM MPE applications provisioned on a Passport node, you should provision staggered *retry* timers. This practice ensures that the CP is not stressed by several ATM MPE applications simultaneously trying to reestablish their soft PVCs after failure of a common interface.

4 Optionally, define the local NSAP address if you are not using the default address. You should normally use the default NSAP address for each *AtmMpe* component instance. This practice ensures that any hierarchical routing in the PNNI network is maintained.

```
set AtmMpe/<n> Stp laddr <nsap_addr>
```

5 Provision the endpoint as the calling or called endpoint of the soft PVC.

```
add AtmMpe/<n> Ac/<conn> SrcPvc
```

or

```
add AtmMpe/<n> Ac/<conn> DstPvc
```

When you provision the endpoint as a source, or calling, endpoint, the Passport system automatically adds a *TrafficManagement* (*Tm*) component under the *SrcPvc* component.

6 If this is the calling end of the soft PVC, set the remote address of the source to the NSAP address at the remote end of the connection. This address must match the provisioned or default NSAP address at the remote end.

Note: To determine the NSAP address at the remote end of the connection, display the *opLocalAddress* attribute of the *AtmMpe Stp* component at the remote end.

```
set AtmMpe/<n> Ac/<conn> SrcPvc raddr <nsap_addr>
```

7 If this is the calling end of the soft PVC, set the remote connection identification. The *remoteCi* is the instance of the *AtmMpe Ac* component at the remote end of the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc rci <remote_conn>
```

8 If this is the calling end of the soft PVC, define the ATM service category for the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm service <category>
```

When you define a service category for the soft PVC, the BBC IE parameters are derived from the service category. For more information on ATM traffic management, see 241-5701-705 *Passport 7400, 15000, 20000 ATM Traffic Management Fundamentals*.

9 If this is the calling end of the soft PVC, define the peak cell rate (PCR) for both directions of the connection.

```
set AtmMpe/<n> Ac/<conn> SrcPvc Tm pcr <rate>
```

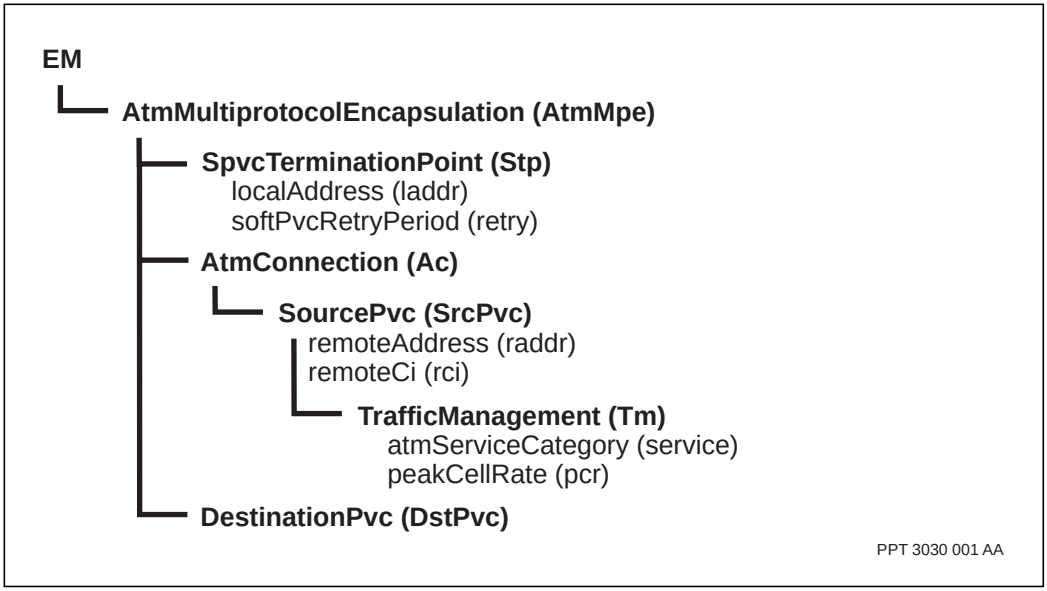
When you define a PCR for the soft PVC, the PCR0+1 traffic descriptor IE parameter is configured. All other parameters are derived from the service category. For more information on ATM traffic management, see 241-5701-705 *Passport 7400, 15000, 20000 ATM Traffic Management Fundamentals*.

Variable definitions

Variable	Value
<category>	*unspecifiedBitRate* (*ubr*), *rtVariableBitRate* (*rtvbr*), *nrtVariableBitRate* (*nrtvbr*), or *constantBitRate* (*cbr*). The default is *ubr*.
<conn>	The instance number of the ATM connection on the ATM MPE interface.
<nsap_addr>	A valid 40-digit NSAP address for the *AtmMpe* component. The default is an empty string. If you do not enter an address, the system supplies a default. For more information on NSAP addressing, see 241-5701-702 *Passport 7400, 15000, 20000 ATM Routing and Signaling Fundamentals*.
<period>	An integer in the range 10 to 3600 seconds. The default is 20 seconds.
<rate>	An integer in the range 0 to 2 147 483 647. The default is 0.
<remote_conn>	An integer in the range 1 to 255. The default is 1.

Procedure job aid

Figure 5
Configuring an ATM soft PVC for an ATM MPE interface component hierarchy



Chapter 3
Frame relay DTE configuration for IP over frame relay

Configure frame relay DTE as a first step to enabling the Passport to carry IP over frame relay.

Frame relay DTE is an alternative as an access media to “IP-optimized DLCI configuration for IP over frame relay” (page 71).

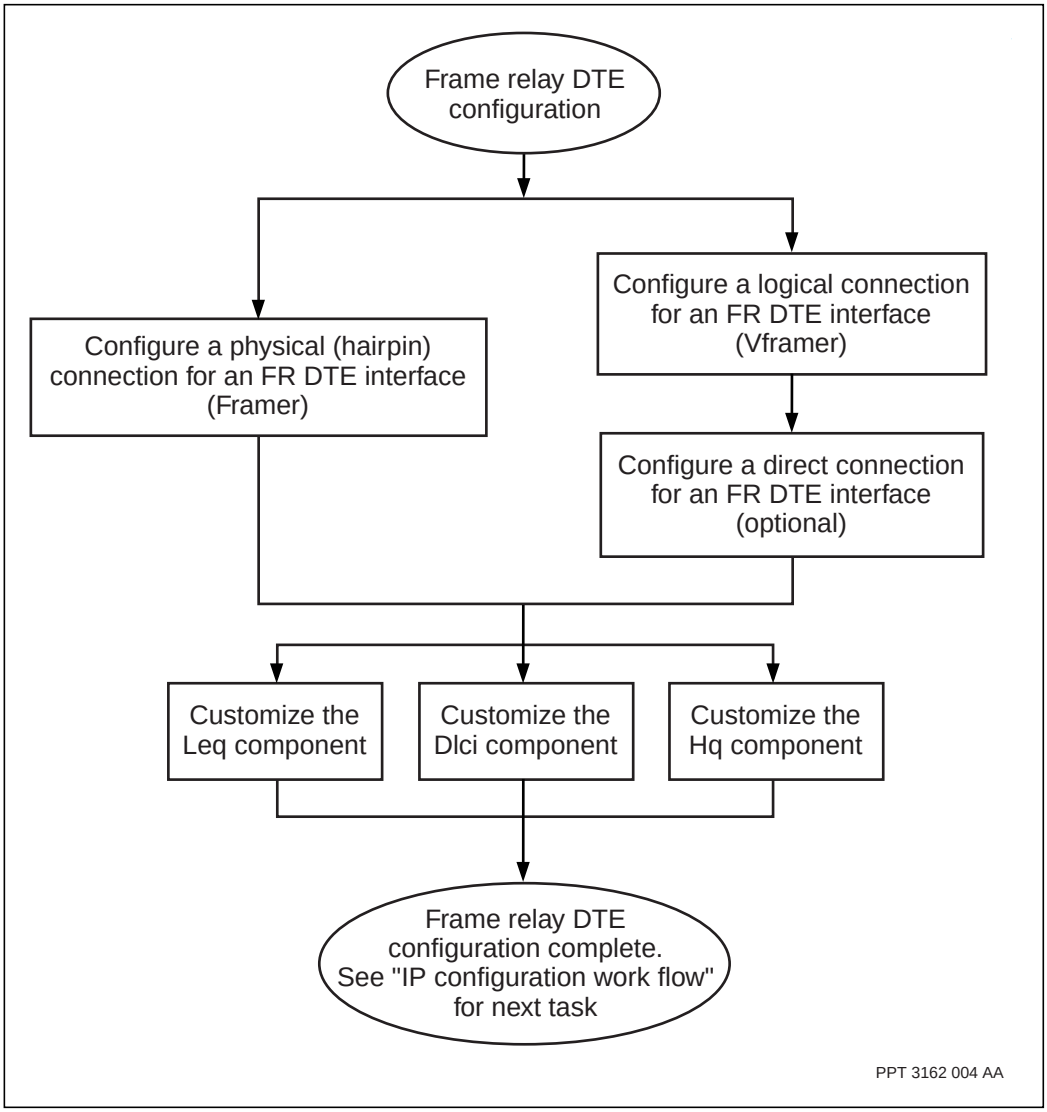
Prerequisites to frame relay DTE configuration

- Configure the required frame relay interfaces and connections. See 241-5701-902 *Passport 7400, 15000, 20000 Configuring Frame Relay*.
- See the figure “IP configuration work flow” (page 38) to understand how frame relay fits into the overall IP configuration task flow.
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information.

Frame relay DTE configuration task flow

This task flow shows you the sequence of procedures you perform to configure frame relay DTE. To link to any procedure, go to the list that follows the task flow.

Figure 6
Frame relay DTE configuration task flow



Navigation links

- “Configuring a frame relay DTE interface for IP traffic” (page 56)

- “Configuring a physical (hairpin) connection for a frame relay DTE interface” (page 57)
- “Configuring a logical connection for a frame relay DTE interface” (page 60)
- “Configuring a direct connection for a frame relay DTE interface” (page 63)
- “Customizing the Frame Relay link emission queue (Leq) component” (page 66)
- “Customizing the data link connection identifier (Dlci) component” (page 68)
- “Customizing the Hq subcomponent” (page 70)
- For information about the next task, see “IP configuration work flow” (page 38)

Configuring a frame relay DTE interface for IP traffic

Configure a frame relay DTE interface for IP traffic to provide a frame relay connection between the Passport node and the IP network.

Prerequisites

- Configure the required frame relay interfaces and connections. See 241-5701-902 *Passport 7400, 15000, 20000 Configuring Frame Relay*.

Procedure steps

1. Add an FrDte link level protocol interface application to the root component. The attributes associated with this new interface have default values assigned automatically. One remote group (Rg/1) is created automatically when FrDte component is added.

   `add FrDte/<n>`

2. List the subcomponents to determine what has been added. Modify the default values as required.

   `list FrDte/<n>`

3. Display the Passport provisionable attributes.

   `display FrDte/<n>`

Variable definitions

Variable	Value
<n>	The number of the FrDte instance.

Configuring a physical (hairpin) connection for a frame relay DTE interface

Configure a physical (hairpin) connection between a frame relay DTE interface and a frame relay UNI interface as an alternative to a logical or direct connection.

Prerequisites

- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information about frame relay DTE physical connections.

Procedure steps

1. Add a logical processor (Lp) to the root component, if you have not already done so:

   ```
   add Lp/<e>
   ```

2. Add a v.35 port to the logical processor:

   ```
   add Lp/<e> V35/<f>
   ```

3. Link the FrDte application to the hardware component:

   ```
   set FrDte/<a> Framer interfaceName Lp/<e> V35/<f>
   ```

4. Add static DLCI components in the range from 16 to 1007 decimal:

   ```
   add FrDte/<a> StDlci/<d>
   ```

5. Link the DLCI to a remote group:

   ```
   set FrDte/<a> StDlci/<d> linkToRemoteGroup FrDte/<a>
   Rg/<g>
   ```

6. Add DLCI and DirectCall subcomponents to both FrUni components:

   ```
   add FrUni/<b> Dlci/<d>
   add FrUni/<c> Dlci/<d>
   ```

   Note: When you add a *Dlci* component, the Passport system automatically creates a *DirectCall (Dc)* subcomponent.

7. Set the attributes of FrUni/<b> Dlci/<d> Dc:

   ```
   set FrUni/<b> Dlci/<d> Dc type master
   set FrUni/<b> Dlci/<d> Dc rdna <dna of Fruni/<c>>
   set FrUni/<b> Dlci/<d> Dc rdlci <d>
   ```

8. Set the attributes of FrUni/<c> Dlci/<d> Dc:

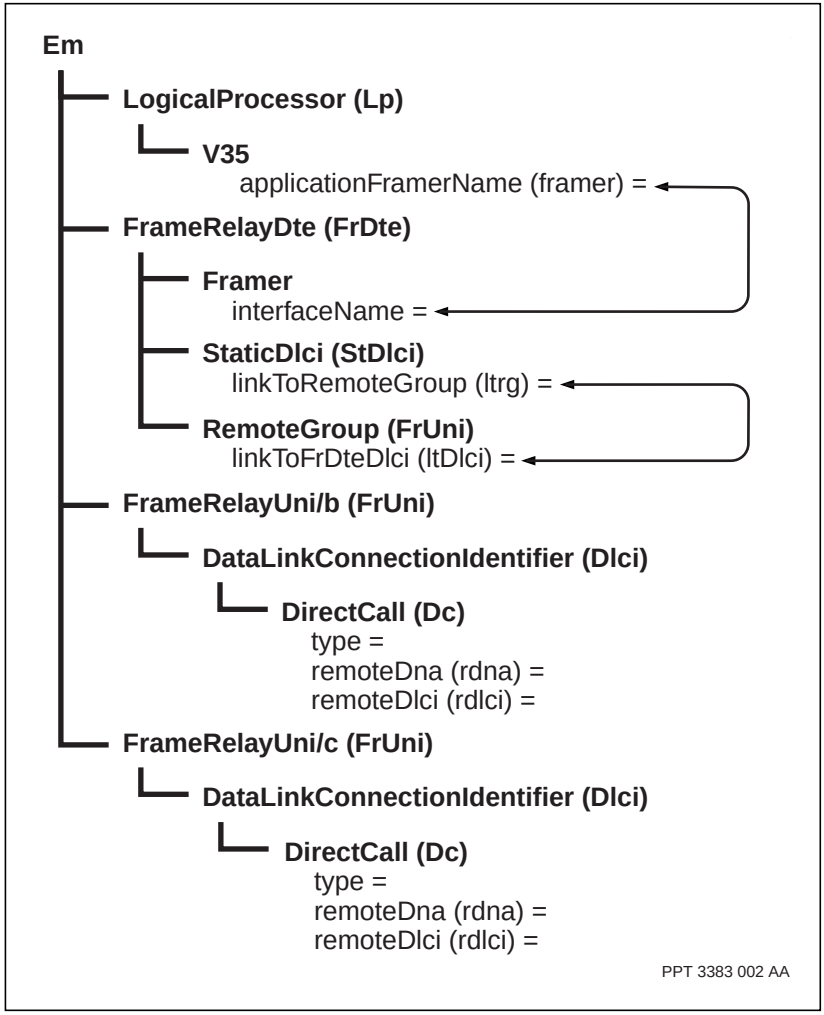
```
set FrUni/<c> Dlci/<d> Dc type slave
set FrUni/<c> Dlci/<d> Dc rdna <dna of Fruni/<b>>
set FrUni/<c> Dlci/<d> Dc rdlci <d>
```

Variable definitions

Variable	Value
<a>	The number of the FrDte instance.
<b>	The number of the FrUni instance.
<d>	The number of the static DLCI instance.
<e>	The number of the Lp instance.
<f>	The number of the V35 instance.
<g>	The number of the remote group instance.

Procedure job aid

Figure 7
Configuring a physical (hairpin) connection for a frame relay DTE interface



Configuring a logical connection for a frame relay DTE interface

Configure a logical connection between a frame relay DTE interface and a frame relay UNI interface as an alternative to a physical or direct connection. Not using a physical connection frees a port on the card.

Prerequisites

- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information about frame relay DTE logical connections.

Procedure steps

1. Delete the framer component and add a virtual framer component to the FrDte:

   ```
   delete FrDte/<a> Framer
   add FrDte/<a> VFramer
   ```

2. Delete the framer component and add a virtual framer subcomponent to FrUni/b:

   ```
   delete FrUni/<b> Framer
   add FrUni/<b> VFramer
   ```

3. Link the FrDte and FrUni virtual framers:

   ```
   set FrDte/<a> VFramer otherVirtualFramer FrUni/<b>
   VFramer
   ```

   ```
   set FrUni/<b> VFramer otherVirtualFramer FrDte/<a>
   VFramer
   ```

   If the virtual framer is configured on the 4-port DS3Ch or 1-port STM-1Ch FPs, the pair of virtual framers must be on the same LP.

4. Link the FrDte application to the Lp:

   ```
   set FrDte/<a> VFramer lp Lp/<e>
   ```

5. Add a static Dlci:

   ```
   add FrDte/<a> StaticDlci/<d>
   ```

6. Link the static Dlci to a remote group:

   ```
   set FrDte/<a> StaticDlci/<d> linkToRemoteGroup
   FrDte/<a> Rg/<g>
   ```

7 Add DLCI and DirectCall subcomponents to both FrUni components:

```
add FrUni/<b> Dlci/<d>
add FrUni/<c> Dlci/<d>
```

Note: When you add a *Dlci* component, the Passport system automatically creates a *DirectCall (Dc)* subcomponent.

8 Set the attributes of FrUni/<b> Dlci/<d> Dc:

```
set FrUni/<b> Dlci/<d> Dc type master
set FrUni/<b> Dlci/<d> Dc rdna <dna of Fruni/<c>>
set FrUni/<b> Dlci/<d> Dc rdlci <d>
```

9 Set the attributes of FrUni/<c> Dlci/<d> Dc:

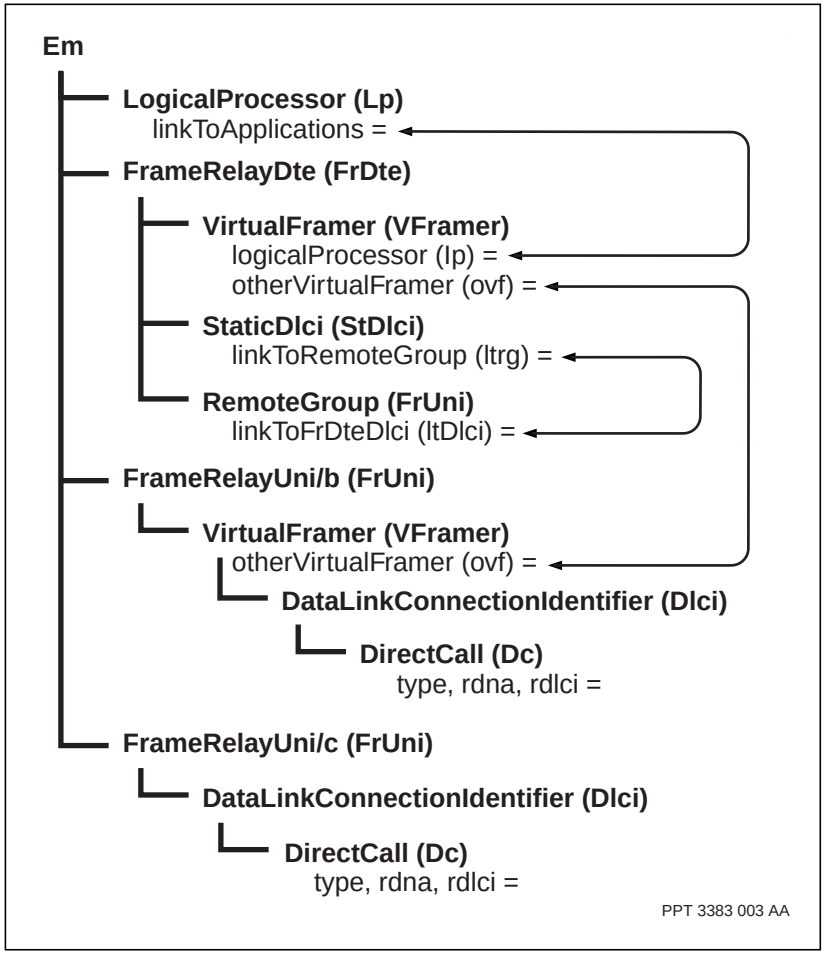
```
set FrUni/<c> Dlci/<d> Dc type slave
set FrUni/<c> Dlci/<d> Dc rdna <dna of Fruni/<b>>
set FrUni/<c> Dlci/<d> Dc rdlci <d>
```

Variable definitions

Variable	Value
<a>	The number of the FrDte instance.
<b>	The number of the FrUni instance.
<c>	The number of the FrUni instance.
<d>	The number of the StaticDlci instance.
<e>	The number of the Lp instance.
<g>	The number of the remote group instance.

Procedure job aid

Figure 8
Configuring a logical connection for a frame relay DTE interface



Configuring a direct connection for a frame relay DTE interface

Configure a direct connection between a frame relay DTE interface and a frame relay UNI interface as an alternative to a physical or logical connection.

Prerequisites

- "Configuring a logical connection for a frame relay DTE interface" (page 60).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information about frame relay DTE direct connections.

Procedure steps

1. Set the LocalManagementInterface (LMI) components on the FrDte and the FrUni to which it interfaces so that no LMI procedures are running:

   ```
   set FrDte/<a> Lmi procedures none
   set FrUni/<b> Lmi procedures none
   ```

2. Turn accounting data collection off for the DLCI of the FrUni that interfaces with the FrDte:

   ```
   set FrUni/<b> Dlci/<d> Sp accounting off
   ```

3. Add a direct connection component to the FrDte:

   ```
   add FrDte/<a> Dconn
   ```

4. Add a direct connection component to the customer-facing FrUni:

   ```
   add FrUni/<c> Dconn
   ```

5. Link the direct connection between the FrDte and the customer-facing FrUni:

   ```
   set FrDte/<a> Dconn directFrUniConnection FrUni/<c>
   Dconn
   ```

Variable definitions

Variable	Value
<a>	The number of the FrDte instance.
<b>	The number of the FrUni instance.

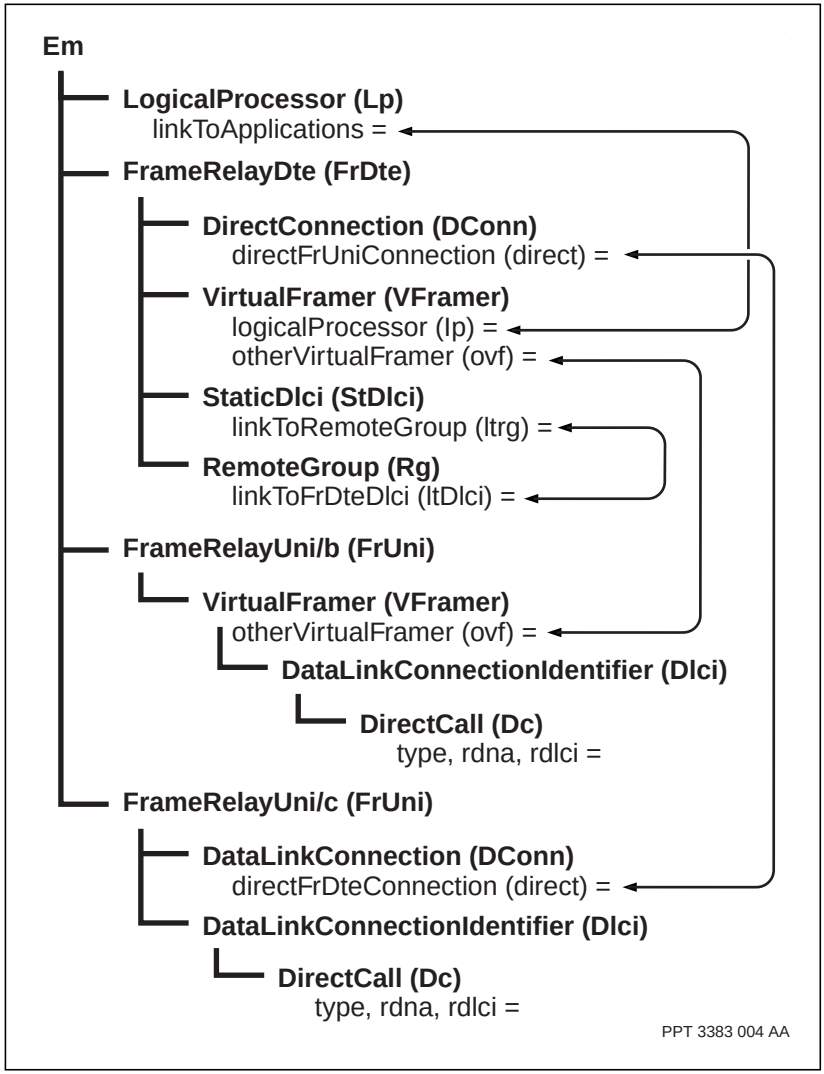
(Sheet 1 of 2)

Variable	Value
<c>	The number of the FrUni instance.
<d>	The number of the Dlci.

(Sheet 2 of 2)

Procedure job aid

Figure 9
Configuring a direct connection for a frame relay DTE interface



Customizing the Frame Relay link emission queue (Leq) component

On a Passport 7400 with SBIC-based FPs only, you can add a link emission queue (*Leq*) subcomponent to the *FrDte* component to provide more elasticity (packet queuing) in the transmit data path, especially for slow frame relay data rates that tend to be overdriven by higher speed LANs. *Leq* also allows prioritization of traffic for certain applications, guaranteeing bandwidth for a particular traffic flow.

> ***Note:*** The *HibernationQueue* (*Hq*) component provides the same services as the *Leq* component. See "Customizing the Hq subcomponent" (page 70).

When you add a Leq component to an instance of frame relay, all of the provisionable attributes default to values that allow normal data path operation. The table "Customizing the FrDte Leq subcomponent" (page 66) provides examples of ways you can customize these values to provide additional capabilities.

Example procedure

> ***Note:*** This is an example procedure. The values you use in your configuration can differ from the values shown here. Consult your network engineer to ensure the values you are using are accurate for your configuration.

Table 1
Customizing the FrDte Leq subcomponent

Customization	Example command
Define the maximum number of packets that can be enqueued in the Leq:	`set FrDte/1 Leq maxPackets 200`
Define the maximum amount of data in milliseconds that can be enqueued at any given time:	`set FrDte/1 Leq maxMsecData 1000`
(Sheet 1 of 2)	

Table 1 (continued)
Customizing the FrDte Leq subcomponent

Customization	Example command
Restrict the percentage of multicast packets allowed to be enqueued to prevent a multicast flooding situation that could engulf the entire queue:	`set FrDte/1 Leq maxPercentMulticast 20`
Ensure old packets on the queue are discarded if they never get an opportunity for transmission:	`set FrDte/1 Leq timeToLive 5000`

(Sheet 2 of 2)

Customizing the data link connection identifier (Dlci) component

The table “Customizing the FrDte Dlci subcomponent” (page 68) provides examples of ways you can customize the provisionable attributes of the *frDte* data link connection identifier (*Dlci*) subcomponent to provide additional capabilities.

You can provision dynamic DLCIs using commands if you are in operational mode. Static DLCIs are provisioned using commands in provisioning mode. See “Operational mode” (page 30) and “Provisioning mode” (page 30).

Since FrDte rate enforcement only applies to PQC-based FPs, the attribute *committedBurst* is ignored for a static DLCI. The length of the first leaky bucket is provided by the attribute *committedInformationRate*, and the length of the second leaky bucket is provided by the attribute *excessBurst*.

Example procedure

This is an example procedure. The values you use in your configuration can differ from the values shown here. Consult your network engineer to ensure the values you are using are accurate for your configuration.

Table 2
Customizing the FrDte Dlci subcomponent

Customization	Example command
Define the average number of bits to be transferred per second over the DLCI to the DCE:	**add FrDte/1 StDlci/100** **set FrDte/1 StDlci/100 committedInformationRate 3800** or **set FrDte/1 DynDlciDef committedInformationRate 3800**
Enable or disable a rate enforcement policy on the DLCI:	**add FrDte/1 StDlci/100**

(Sheet 1 of 2)

Table 2 (continued)
Customizing the FrDte Dlci subcomponent

Customization	Example command
Define the committed burst size (in bits) to which the *Dlci* component wants to subscribe:	**add FrDte/1 StDlci/100** **set FrDte/1 StDlci/100 committedBurst 1000000** or **set FrDte/1 DynDlciDef committedBurst 1000000**
Define the excess bursts (in bits) to which the *Dlci* component wants to subscribe:	**add FrDte/1 StDlci/100** **set FrDte/1 StDlci/100 excessBurst 3000** or **set FrDte/1 DynDlciDef excessBurst 3000**
Specify the action taken when *committedBurst* size has been exceeded on the *Dlci* component but the *excessBurst* size has not:	**add FrDte/1 StDlci/100** **set FrDte/1 StDlci/100 excessBurstAction <action>** or **set FrDte/1 DynDlciDef excessBurstAction <action>** If multiple DLCIs are available for a packet's next IP hop on a frame relay DTE, IP CoS uses the DLCI with the ipCos attribute value that matches the packet's CoS. For example: **add FrDte/1 StDlci/25** **set FrDte/1 StDlci/25 ipCos 0**

(Sheet 2 of 2)

Customizing the Hq subcomponent

The *Hq* subcomponent under the *StDlci* component provides the same type of service as the *Leq* component, except the service is performed on a per DLCI basis. The *Hq* subcomponent is supported only on a Passport 7400 with SBIC-based FPs.

Procedure steps

The following command adds the *Hq* subcomponent under *StDlci/100*:

```
add FrDte/1 StDlci/100 Hq
```

Example procedure

Note: This is an example procedure. The values you use in your configuration can differ from the values shown here. Consult your network engineer to ensure the values you are using are accurate for your configuration.

The provisionable attributes for *Hq* are the same as those for *Leq*, including the defaults and ranges for values. See “Customizing the Frame Relay link emission queue (Leq) component” (page 66). The following example shows the command to provision the *maxMsecData* attribute under *Hq* to 1000 milliseconds:

```
set FrDte/1 StDlci/100 Hq maxMsecData 1000
```

Chapter 4
IP-optimized DLCI configuration for IP over frame relay

Configure an IP-optimized DLCI to enable the Passport to carry IP over frame relay.

IP-optimized DLCI is an alternative as an access media to “Frame relay DTE configuration for IP over frame relay” (page 53).

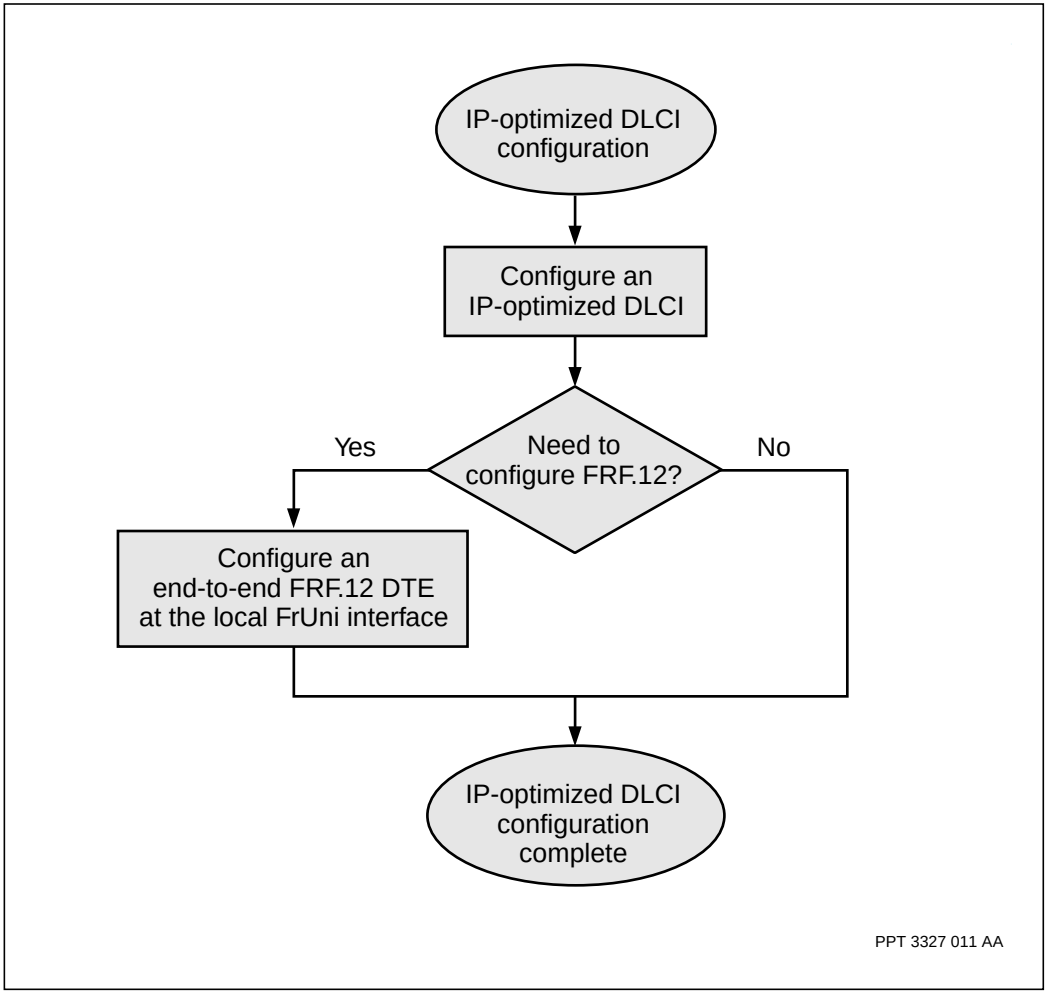
Prerequisites to IP-optimized DLCI configuration

- Configure the required frame relay interfaces. See 241-5701-902 *Passport 7400, 15000, 20000 Configuring Frame Relay*.
- See the figure “IP configuration work flow” (page 38) to understand how IP-optimized DLCI fits into the overall IP configuration task flow.
- See 241-5701-615 *Passport 7400, 15000, 20000 FP Configuration Reference* for information on which FPs support this service.
- Use the procedures in 241-5701-270 *Passport 7400, 15000, 20000 Software Installation Guide* to load any required features. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for information on application and feature names for IP on Passport.
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information.

IP-optimized DLCI configuration task flow

This task flow shows you the sequence of procedures you perform to configure IP-optimized DLCI. To link to any procedure, go to “Navigation links” (page 73).

Figure 10
IP-optimized DLCI configuration task flow



Navigation links

- “Configuring an IP-optimized DLCI” (page 74)
- “Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface” (page 76)
- For information about the next task, see the figure “IP configuration work flow” (page 38).

Configuring an IP-optimized DLCI

Configure an IP-optimized DLCI to create a direct frame relay access to a Passport network.

Prerequisites

- When you are using an IP-optimized DLCI at the edge of the network, it is recommended that you set attribute *FrUni Lmi side* to network.

Procedure steps

1. Add the IpDlciGroup. Subcomponent *IpDlciGroup Frc/1* is automatically created.

   add IpDlciGroup/<n>

2. Add the DLCI and IP connection to the FrUni.

   add FrUni/<o> Dlci/<p>

   del FrUni/<o> Dlci/<p> Dc

   add FrUni/<o> Dlci/<p> IpConnection

   Note: When you add a *Dlci* component, a *Dc* subcomponent is automatically created. *Dc* and *IpConnection* subcomponents are mutually exclusive.

3. Link the DLCI to its FrConnection.

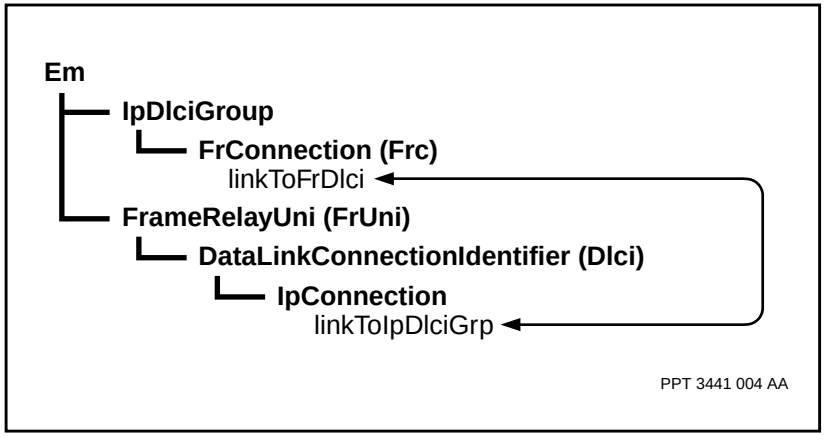
   set FrUni/<o> Dlci/<p> IpConnection linkToIpDlciGrp IpDlciGroup/<n> Frc/<q>

Variable definitions

Variable	Value
<n>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<o>	The instance value of the FrUni.
<p>	The instance value of the DLCI.
<q>	The instance value of the frame relay connection.

Procedure job aid

Figure 11
Configuring an IP-optimized DLCI component hierarchy



Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface

Configure an end-to-end FRF.12 DTE at the local FrUni interface to control delay variation when voice is carried across the same interface as data.

Prerequisites

- For conceptual information see 241-5701-901 *Passport 7400, 15000, 20000 Frame Relay Fundamentals*.

Procedure steps

1. Turn fragmentation on.

   ```
   set FrUni/<o> Dlci/<p> Sp frf12EndToEnd on
   ```

2. Set the data frame size.

   ```
   set FrUni/<o> Dlci/<p> Sp frf12FragmentSize
   <frag_size>
   ```

3. Set the CoS index.

   ```
   set IpDlciGroup/<n> Frc/<q> ipCoS <cos>
   ```

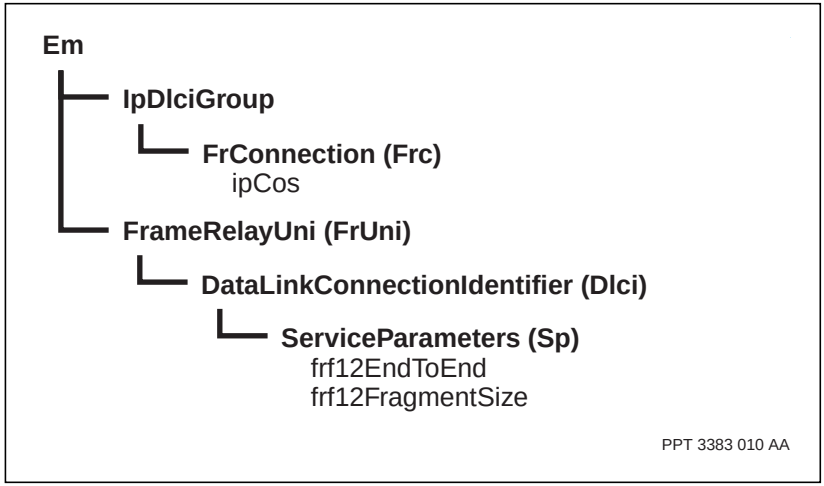
   Note: The Cos index is used as the value for the emission priority (EP).

Variable definitions

Variable	Value
<cos>	The CoS value associated with the DLCI.
<n>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<o>	The instance value of the FrUni.
<p>	The instance value of the DLCI.
<q>	The instance value of the frame relay connection.

Procedure job aid

Figure 12
Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface component hierarchy



Chapter 5
Frame relay DTE to IP-optimized DLCI migration

Migrate from FrDte-based to IP-optimized DLCI IP over frame relay to increase traffic throughput and simplify provisioning for IP over frame relay.

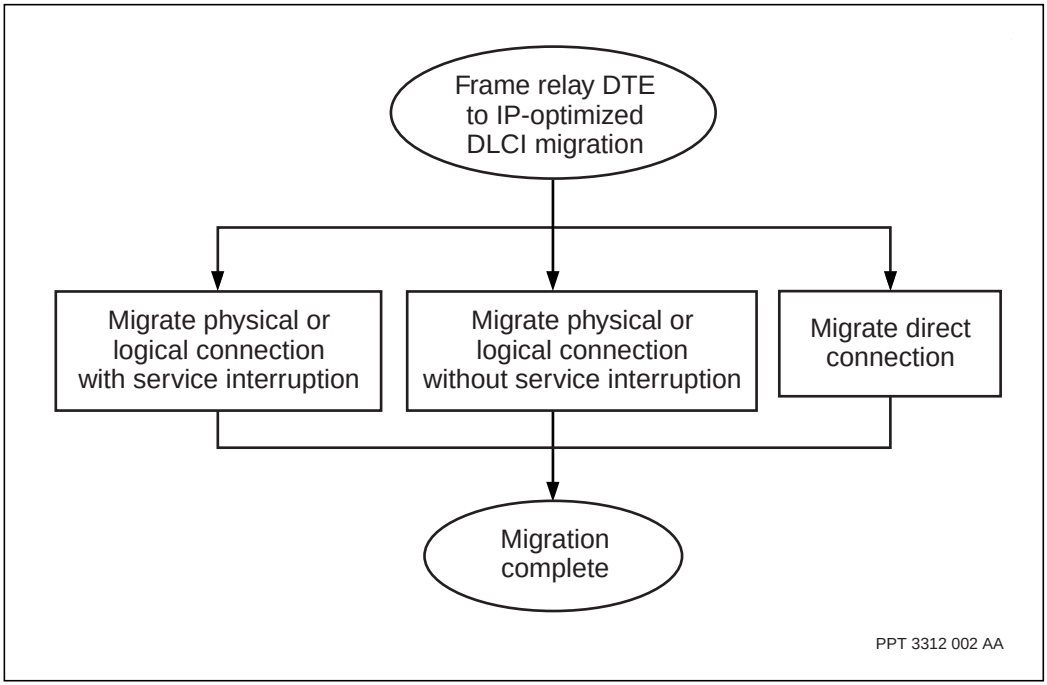
Prerequisites to frame relay DTE to IP-optimized DLCI migration

- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information.

Frame relay DTE to IP-optimized DLCI migration task flow

This task flow shows you the sequence of procedures you perform to migrate frame relay DTE to IP-optimized DLCI. To link to any procedure, go to “Navigation links” (page 80).

Figure 13
Frame relay DTE to IP-optimized DLCI migration task flow



Navigation links

- "Migrating from a physical (hairpin) or logical connection with service interruption" (page 81)
- "Migrating from a physical (hairpin) or logical connection without service interruption" (page 84)
- "Migrating from a direct connection" (page 87)
- For information about the next task, see the figure "IP configuration work flow" (page 38).

Migrating from a physical (hairpin) or logical connection with service interruption

In this procedure, the existing DLCI is converted to an IP-optimized DLCI.

With this approach, the migration is done in one provisioning session and no provisioning is required on the end user equipment. There is a small disruption in service until the DLCI becomes enabled.

Prerequisites

- See “Configuring a physical (hairpin) connection for a frame relay DTE interface” (page 57) or “Configuring a logical connection for a frame relay DTE interface” (page 60) for information on the existing connection.

Procedure steps

1. Remove the components that are no longer required.

   ```
   del FrUni/<c> Dlci/<d> Dc
   del FrUni/<b>
   del FrDte/<a>
   ```

2. Add the IP DLCI group.

   ```
   add IpDlciGroup/<e>
   ```

3. Link the IP DLCI group to the protocol port that was linked to the FrDTE.

   ```
   set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
   ```

4. Add the IP connection.

   ```
   add FrUni/<c> Dlci/<d> IpConnection
   ```

5. Link the DLCI to its FrConnection.

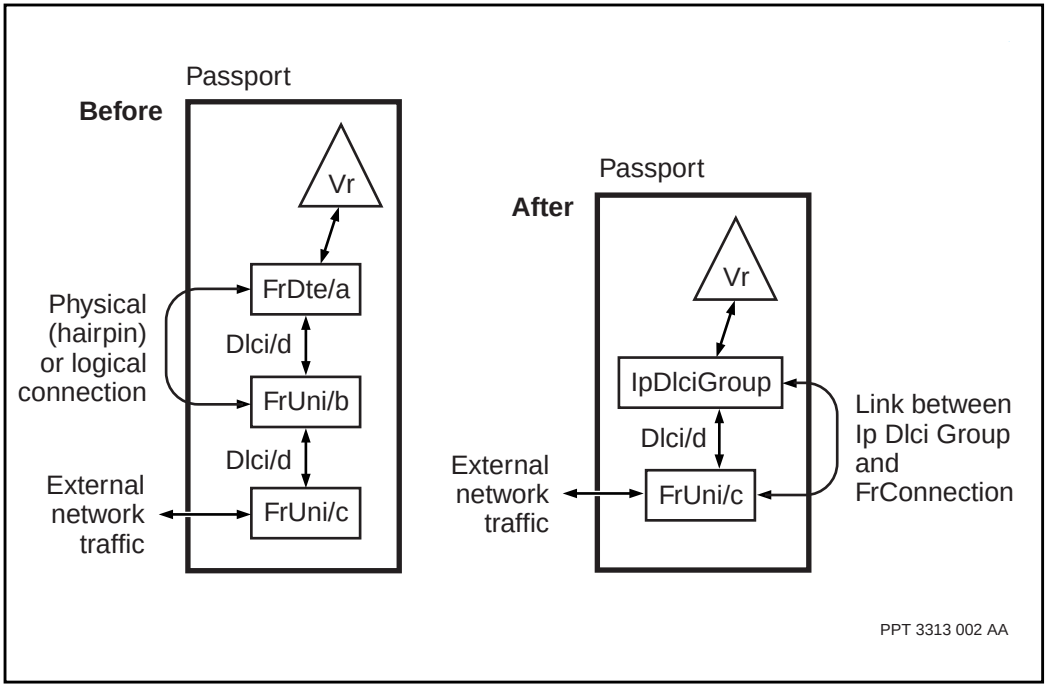
   ```
   set FrUni/<c> Dlci/<d> IpConnection linkToDlciGrp
   IpDlciGroup/<e> Frc/<h>
   ```

Variable definitions

Variable	Value
<a>	The instance value of the FrDTE.
<b>	The instance value of the FrUni between the FrDTE and the customer-facing FrUni.
<c>	The instance value of the customer-facing FrUni.
<d>	The instance value of the DLCI.
<e>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	The name of the virtual router.
<g>	The name of the protocol port.
<h>	The instance value of the frame relay connection.

Procedure job aid

Figure 14
Migrating from a physical (hairpin) or logical connection with service interruption



Migrating from a physical (hairpin) or logical connection without service interruption

In this procedure, a new IP-optimized DLCI is created.

With this approach, the existing DLCI can continue to provide service while the IP-optimized DLCI is being created. Once the IP-optimized DLCI has been tested, the old DLCI can be deleted.

Prerequisites

- See “Configuring a physical (hairpin) connection for a frame relay DTE interface” (page 57) or “Configuring a logical connection for a frame relay DTE interface” (page 60) for information on the existing connection.

Procedure steps

1. Add the IP DLCI group.

   ```
   add IpDlciGroup/<e>
   ```

2. Add the DLCI and IP connection to the FrUni.

   ```
   add FrUni/<c> Dlci/<new>

   del FrUni/<c> Dlci/<new> Dc

   add FrUni/<c> Dlci/<new> IpConnection
   ```

3. Link the DLCI to its FrConnection.

   ```
   set FrUni/<c> Dlci/<new> IpConnection linkToIpDlciGrp
   IpDlciGroup/<e> Frc/<h>
   ```

4. Link the IP DLCI group to the protocol port.

   ```
   set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
   ```

5. Ensure that you have set up a logical interface and network mask. See “Enabling IP on a protocol port” (page 124).

6. Lock the original DLCI to force traffic on to the IP-optimized DLCI.

   ```
   lock FrUni/<c> Dlci/<old>
   ```

7. When you are sure the IP-optimized DLCI is working properly, remove the components that are no longer required.

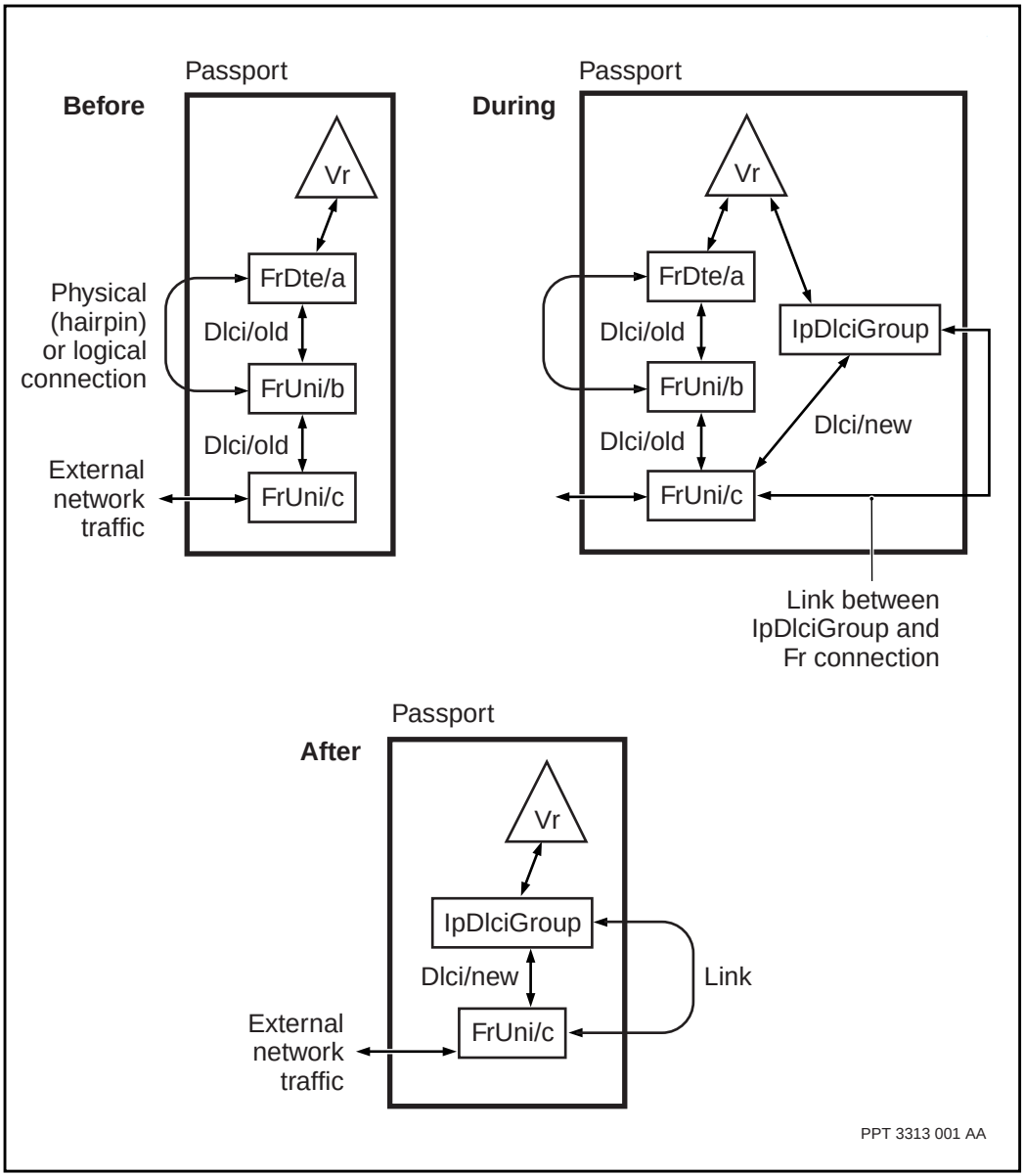
```
del FrUni/<c> Dlci/<old>
del FrUni/<b>
del FrDte/<a>
del Vr/<f> Pp/<i>
```

Variable definitions

Variable	Value
<a>	The instance value of the FrDTE.
<b>	The instance value of the FrUni between the FrDTE and the customer-facing FrUni.
<c>	The instance value of the customer-facing FrUni.
<e>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	The name of the virtual router.
<g>	The name of the protocol port for the IP-optimized DLCI.
<h>	The instance value of the frame relay connection.
<i>	The name of the protocol port for the frame relay DTE.
<new>	The instance value of the IP-optimzed DLCI.
<old>	The instance value of the old DLCI.

Procedure job aid

Figure 15
Migrating from a physical (hairpin) or logical connection without service interruption



Migrating from a direct connection

In this procedure, the existing DLCI is converted to an IP-optimized DLCI.

With this approach, the migration is done in one provisioning session and no provisioning is required on the end user equipment. There is a small disruption in service until the DLCI becomes enabled.

Prerequisites

- See "Configuring a direct connection for a frame relay DTE interface" (page 63) for information on the existing connection.

Procedure steps

1. Remove the components that are no longer required.

   ```
   del FrUni/<c> Dconn
   del FrUni/<c> Dlci/<d> Dc
   del FrUni/<b>
   del FrDte/<a>
   ```

2. Add the IP DLCI group.

   ```
   add IpDlciGroup/<e>
   ```

3. Link the IP DLCI group to the protocol port that was linked to the FrDTE.

   ```
   set IpDlciGroup/<e> linkToProtocolPort Vr/<f> Pp/<g>
   ```

4. Add the IP connection.

   ```
   add FrUni/<c> Dlci/<d> IpConnection
   ```

5. Link the DLCI to its FrConnection.

   ```
   set FrUni/<c> Dlci/<d> IpConnection linkToDlciGrp
   IpDlciGroup/<e> Frc/<h>
   ```

Variable definitions

Variable	Value
<a>	The instance value of the FrDTE.
<b>	The instance value of the FrUni between the FrDTE and the customer-facing FrUni.

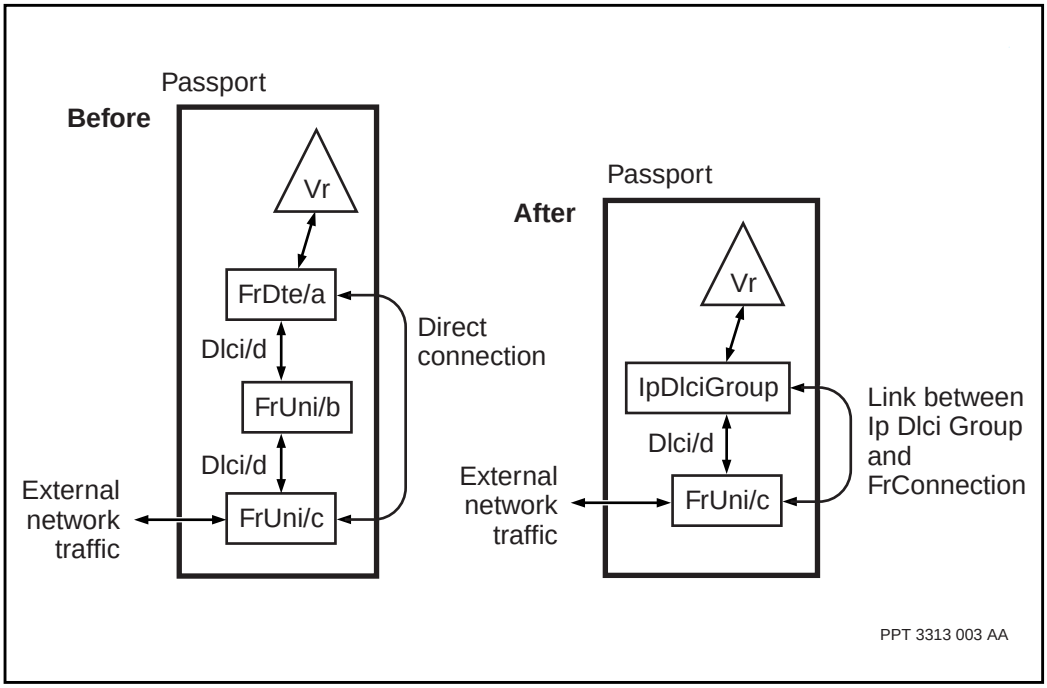
(Sheet 1 of 2)

Variable	Value
<c>	The instance value of the customer-facing FrUni.
<d>	The instance value of the DLCI.
<e>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<f>	The name of the virtual router.
<g>	The name of the protocol port.
<h>	The instance value of the frame relay connection.

(Sheet 2 of 2)

Procedure job aid

Figure 16
Migrating from a direct connection



Chapter 6
Gigabit Ethernet configuration for IP over GigE

Configure gigabit Ethernet as a first step to enabling Passport to carry IP over GigE.

The procedures in this chapter also apply to configuring 10BaseT Ethernet and 100BaseT Ethernet.

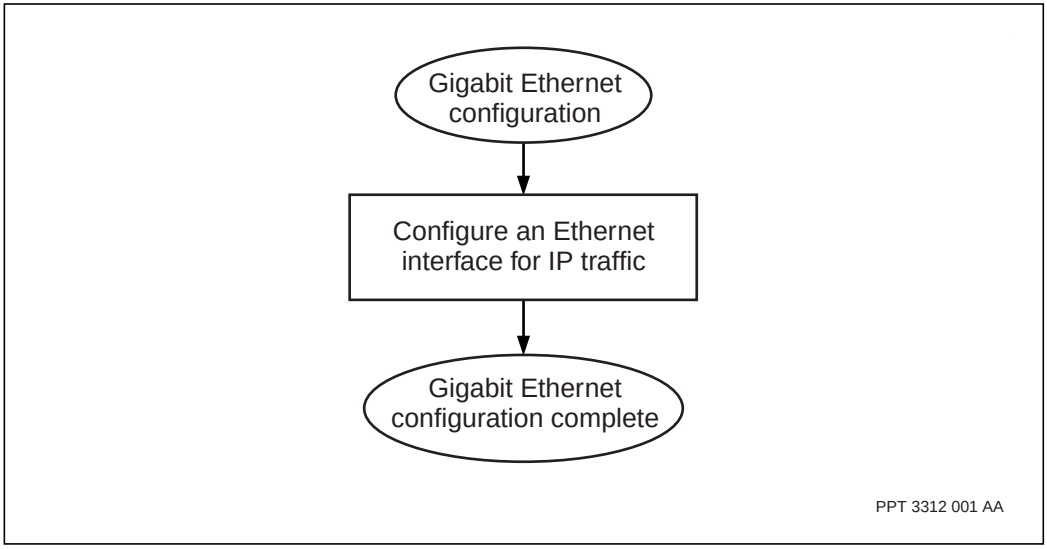
Prerequisites to gigabit Ethernet configuration

- See the figure “IP configuration work flow” (page 38) to understand how gigabit Ethernet fits into the overall IP configuration task flow.
- See 241-5701-615 *Passport 7400, 15000, 20000 FP Configuration Reference* for information on which FPs support this service.
- Use the procedures in 241-5701-270 *Passport 7400, 15000, 20000 Software Installation Guide* to load any required features. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for information on application and feature names for IP on Passport.
- You need to load feature atmMpe on the feature list of the gigabit Ethernet FP in order to forward traffic to ATM MPE media.
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for supporting information.

Gigabit Ethernet configuration task flow

This task flow shows you the sequence of procedures you perform to configure gigabit Ethernet. To link to any procedure, go to "Navigation links" (page 90).

Figure 17
Gigabit Ethernet configuration task flow



Navigation links

- "Configuring an Ethernet interface for IP traffic" (page 91)
- For information about the next task, see the figure "IP configuration work flow" (page 38).

Configuring an Ethernet interface for IP traffic

Configure an Ethernet interface for IP traffic to provide an Ethernet connection between the Passport node and the IP network.

Procedure steps

1. Add a LAN media application. The *La Framer* subcomponent is added automatically.

   ```
   add La/<x>
   ```

2. Link the LAN media application to a physical port.

   ```
   set La/<x> Framer interfaceName Lp/<y> Ethernet/<z>
   ```

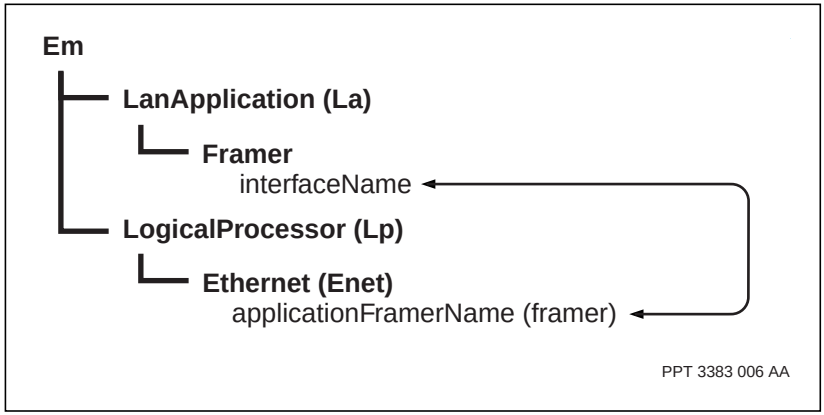
   Note: If you are configuring 100BaseT Ethernet, replace component *Ethernet* with *Ethernet100BaseT* (*Eth100*).

Variable definitions

Variable	Value
<x>	The number of the LAN media application instance.
<y>	The number of the logical processor instance.
<z>	The number of the FP port to which the LAN media application attaches.

Procedure job aid

Figure 18
Configuring an Ethernet interface for IP traffic component hierarchy



Chapter 7
Point-to-point protocol configuration for IP over PPP

Configure point-to-point protocol (PPP) as a first step to enabling the Passport to carry IP over PPP.

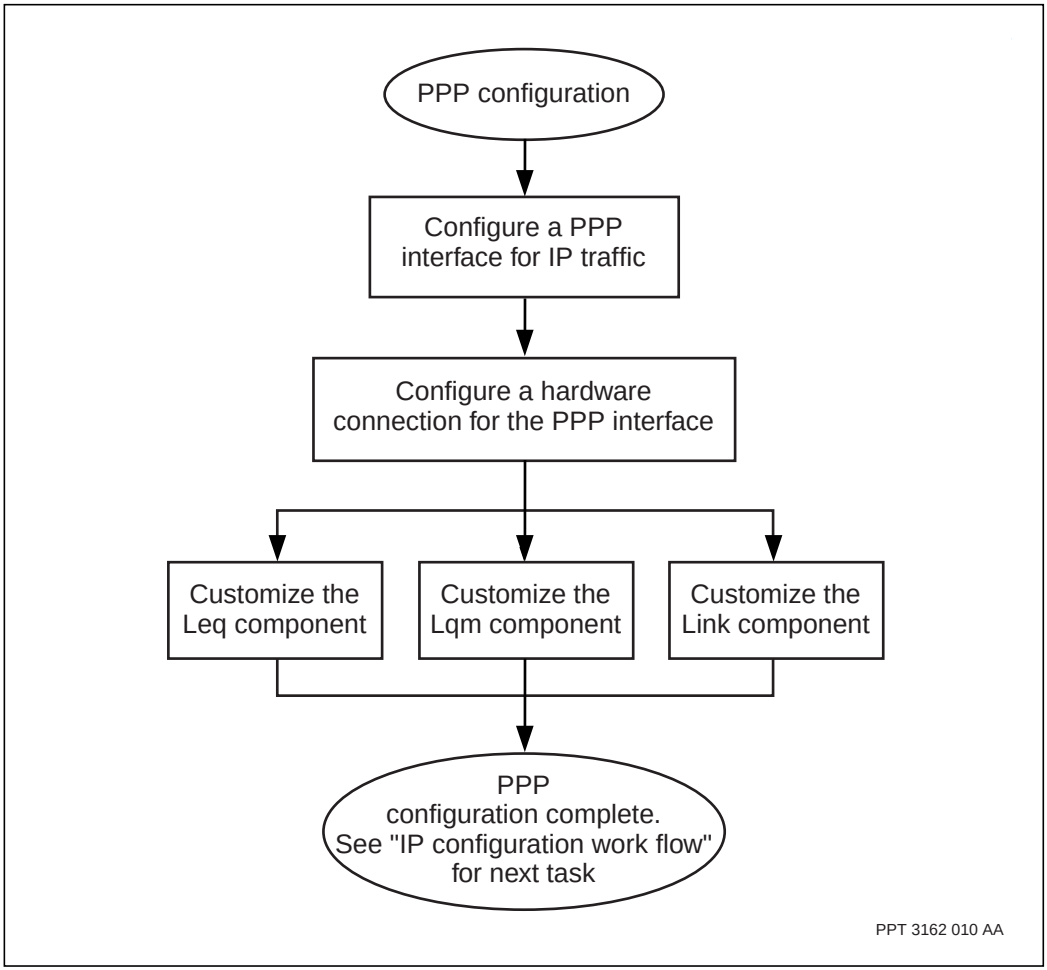
Prerequisites to PPP configuration

- Configure the required PPP interfaces and connections.

PPP configuration task flow

This task flow shows you the sequence of procedures you perform to configure PPP. To link to any procedure, go to “Navigation links” (page 94).

Figure 19
Point-to-point protocol configuration for task flow



Navigation links

- “Configuring a PPP interface for IP traffic” (page 96)
- “Configuring a hardware connection for a PPP interface” (page 97)
- “Customizing the Link component” (page 99)
- “Customizing the link quality monitor (Lqm) component” (page 102)

- “Customizing the link emission queue (Leq) component” (page 103)
- For information about the next task, see the figure “IP configuration work flow” (page 38)

Configuring a PPP interface for IP traffic

Configure a PPP interface for IP traffic to provide a PPP connection between the Passport node and the IP network.

Procedure steps

1. Add a PPP link level protocol interface application to the root component. The attributes associated with this new interface have default values assigned automatically.

   `add Ppp/<n>`

2. List the subcomponents to determine what has been added. Modify the default values as required.

   `list Ppp/<n>`

3. Display the PPP provisionable attributes:

   `display Ppp/<n>`

Variable definitions

Variable	Value
<n>	The number of the PPP instance.

Configuring a hardware connection for a PPP interface

Configure a hardware connection for a PPP interface to route IP packets directly to the link queue of the card where PPP resides without software intervention on the FP. This dramatically improves full duplex packet switching performance over PPP.

Procedure steps

Note: This example uses a Passport 7400 MSA32 E1 card.

1. Add a logical processor (LP) to the root component, if you have not already done so.

   ```
   add Lp/<a>
   ```

2. Add a port to the logical processor. In this example, an E1 port is added. The Passport software automatically adds channel 0 when you add a port.

   ```
   add Lp/<a> E1/<b>
   ```

3. Link the PPP service to the hardware component.

   ```
   set Ppp/<n> Framer interfaceName Lp/<a> E1/<b> Chan/0
   ```

   or

   ```
   set Lp/<a> E1/<b> Chan/0 framer Ppp/<n> Framer
   ```

4. Configure the timeslots of the hardware connection.

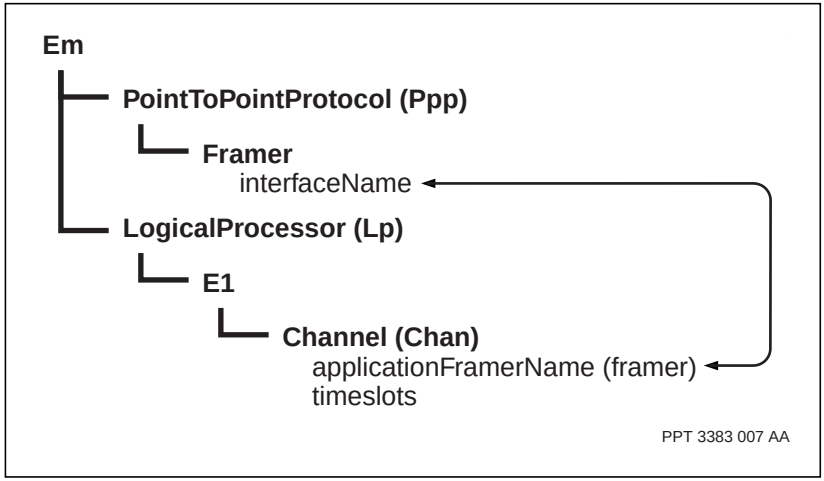
   ```
   set Lp/<a> E1/<b> Chan/0 timeslots 1 2 3 4 5 6 7 8 9
   10 11 12 13 14 15 17 18 19 20 21 22 23 24 25 26 27 28
   29 30 31
   ```

Variable definitions

Variable	Value
<a>	The number of the Lp instance.
<b>	The number of the port instance.
<n>	The number of the Ppp instance.

Procedure job aid

Figure 20
Configuring a hardware connection for a PPP interface component hierarchy



Customizing the Link component

The table "Customizing the Link component" (page 99) provides examples of ways you can customize the provisionable attributes of the *Ppp Link* subcomponent to provide additional capabilities.

Example procedure

Note: This is an example procedure. The values you use in your configuration might differ from the values shown here. Consult you network engineer to ensure the values you are using are accurate for your configuration.

Table 3
Customizing the Link component

Customization	Example command
Add a Link component:	**`add Ppp/1 Link`**
Change the maximum receive unit (MRU) negotiated with the peer PPP application:	**`set Ppp/1 Link configInitialMru <n>`** where: <n> is the size in bytes of the MRU
Use magic number negotiation to detect looped back *link* connections:	**`set Ppp/1 Link configMagicNumber enabled`** where: <n> is either enabled or disabled
Disable the *link* continuity monitor (LCM):	**`set Ppp/1 Link continuityMonitor <n>`** where: <n> is either enabled or disabled
Customize the elapsed time after which PPP attempts to connect with its peer:	**`set Ppp/1 Link restartTimer <n>`** where: <n> is the time in milliseconds

(Sheet 1 of 3)

Table 3 (continued)
Customizing the Link component

Customization	Example command
Customize the number of LCP configuration request retries before entering the stopped state:	**set Ppp/1 Link configureRequestTries <n>** where: <n> is the number of configuration request retries
Customize the number of LCP negative acknowledgements (Naks) that the PPP application sends out when it receives LCP configure requests that are not expected:	**set Ppp/1 Link negativeAckTries <n>** where: <n> is the number of negative acknowledgements the PPP application sends out before entering the stopped state.
Customize the number of LCP terminate request packets a PPP application sends out when it terminates a PPP connection:	**set Ppp/1 Link terminateRequestTries <n>** where: <n> is the number of terminate request packets sent out.
When LQM is enabled, set the quality threshold to a percentage of good packets required to maintain the *link* connection:	**set Ppp/1 Link qualityThreshold <n>** where: <n> is the number of good packets required to maintain the link connection before the link is disabled.
(Sheet 2 of 3)	

Table 3 (continued)
Customizing the Link component

Customization	Example command
Customize the quality window during which LQM examines link quality reporting (LQR) samples to determine what the quality has been over the history of the PPP connection:	**`set Ppp/1 Link qualityWindow <n>`** where: <n> is the time in seconds of the quality window
(Sheet 3 of 3)	

Customizing the link quality monitor (Lqm) component

The table “Customizing the Lqm component” (page 102) provides examples of ways you can customize the provisionable attributes of the *Ppp Lqm* subcomponent to provide additional capabilities.

Example procedure

Note: This is an example procedure. The values you use in your configuration might differ from the values shown here. Consult you network engineer to ensure the values you are using are accurate for your configuration.

Table 4
Customizing the Lqm component

Customization	Example command
Add an Lqm component:	**`add Ppp/1 Lqm`**
Change the status of the Lqm component:	**`set Ppp/1 Lqm configStatus <n>`** where: <n> is either enabled or disabled
Customize the LQM reporting period used by the PPP connection:	**`set Ppp/1 Lqm configPeriod <n>`** where: <n> is a value in the range of zero to 180,000 centiseconds

Customizing the link emission queue (Leq) component

The table “Customizing the Leq component” (page 103) provides examples of ways you can customize the provisionable attributes of the *Ppp Leq* subcomponent to provide additional capabilities.

The *Ppp Leq* subcomponent is only available on a Passport 7400 with SBIC-based FPs.

Example procedure

Note: This is an example procedure. The values you use in your configuration might differ from the values shown here. Consult you network engineer to ensure the values you are using are accurate for your configuration.

Table 5
Customizing the Leq component

Customization	Example command
Add an Leq component:	**`add Ppp/1 Leq`**
Define the maximum number of packets that can be queued in the LEQ:	**`set Ppp/1 Leq maxPackets <n>`** where: <n> is a value in the range of zero to 2048 packets ***Note:*** If zero (default) is specified, the maximum number is determined by the following formula: transmit data rate /1000 = default data rate
Define the maximum amount of data that can be enqueued at any given time:	**`set Ppp/1 Leq maxMsecData <n>`** where: <n> is a value from 100-60,000 milliseconds

(Sheet 1 of 2)

Table 5 (continued)
Customizing the Leq component

Customization	Example command
Restrict the percentage of multicast packets allowed to be enqueued to prevent a multicast flooding situation that could engulf the entire queue:	**set Ppp/1 Leq maxPercentMulticast <n>**
	where:
	<n> is a value from 1%-100%
Ensures that old packets on the queue are discarded if they never get an opportunity for transmission:	**set Ppp/1 Leq timeToLive <n>**
	where:
	<n> is a value from 10,000-60,000 milliseconds

(Sheet 2 of 2)

Chapter 8
Configuring PPP/ATM interworking for Passport 7400

Configure the PPP/ATM interworking on Passport 7400 32-port MSA function processors to enable IP transport between PPP-attached user devices and ATM-attached routers.

Prerequisites

- Load the PppIwf feature onto the ATM FP, which also needs to be configured with PNNI.
- All configuration steps, including the configuration of PNNI or IISP, must be configured in the ATM network to enable the establishment of an SPVC connection. See 241-5701-710 *Passport 7400, 15000, 20000 ATM Configuration Guide*.
- Configure the required PPP interfaces and connections on the CPE router, connecting to the E1 or DS1 port configured with the *PppIwf* component.
- Connect the core router to a port where the SPVC terminates and with the atmMpe attribute matching the encapType attribute of the *PppIwf* component.
- Configure the static ARP entry on the core router with an IP address that matches the PPP IP address. See “Configuring static ARP” (page 205).

Procedure steps

1 Update the application version list (AVL) with the selected versions of the software.

```
set Sw avl wanDte_<version>
```

When you update the AVL, you must verify the configuration twice: once before you can activate it, and again when the node is running the new software. For more information about changing the AVL, see 241-5701-270 *Passport 7400, 15000, 20000 Software Installation Guide*.

2 Add the logical processor types for the MSA 32 FP. For illustrative purposes only, the name MSA32PIWF is used.

```
add Sw Lpt/<MSA32PIWF>
```

3 Add the MSA32 FP.

```
add Lp/1

set Lp/1 main shelf card/1

set Shelf card/1 cardType <MSA32cardType>
```

4 Set the logical processor types for the MSA 32 FP. For illustrative purposes only, the name MSA32PIWF is used.

```
set Sw Lpt/MSA32PIWF feature pppIwf
```

5 Add a DS1 or E1 port and provision the timeslots for its channels component.

For DS1 function processors:

```
add Lp/<Lp1> DS1/<Ds1>

set Lp/<Lp1> Ds1/<Ds1> Chan/<DS_Chan1> timeslots 1 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
```

For E1 function processors:

```
add Lp/<Lp1> E1/<E1>

set Lp/<Lp1> E1/<E1> Chan/<E1_Chan1> timeslots 1 2 3 4
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

6 Add the *PppIwf* component.

```
add PppIwf/1100
```

7 Link the *Framer* component to the physical TDM port.

```
set PppIwf/1100 int lp/1 ds1/0 chan/0
```

8 Add the *AtmAp* component.

```
add PppIwf/1100 atmap
```

9 Set the *AtmAp addressToCall* attribute to the remote ATM interface.

```
set PppIwf/1100 atmap addresstocall
47000000000000000000000010020480D000B00
```

10 Set the *AtmAp* component called VpiVci to that of the connection required to be set up on the remote ATM interface.

```
set PppIwf/1100 atmap calledvpivci 0.200
```

11 Optionally, set the *atmServiceCategory* attribute for the SPVC:

```
set PppIwf/1100 PppIwf/211 atm tm atmServiceCategory
<atmServiceCategory>
```

12 If you are using UPC in your ATM PNNI network, you must set the *peakCellRate* attribute:

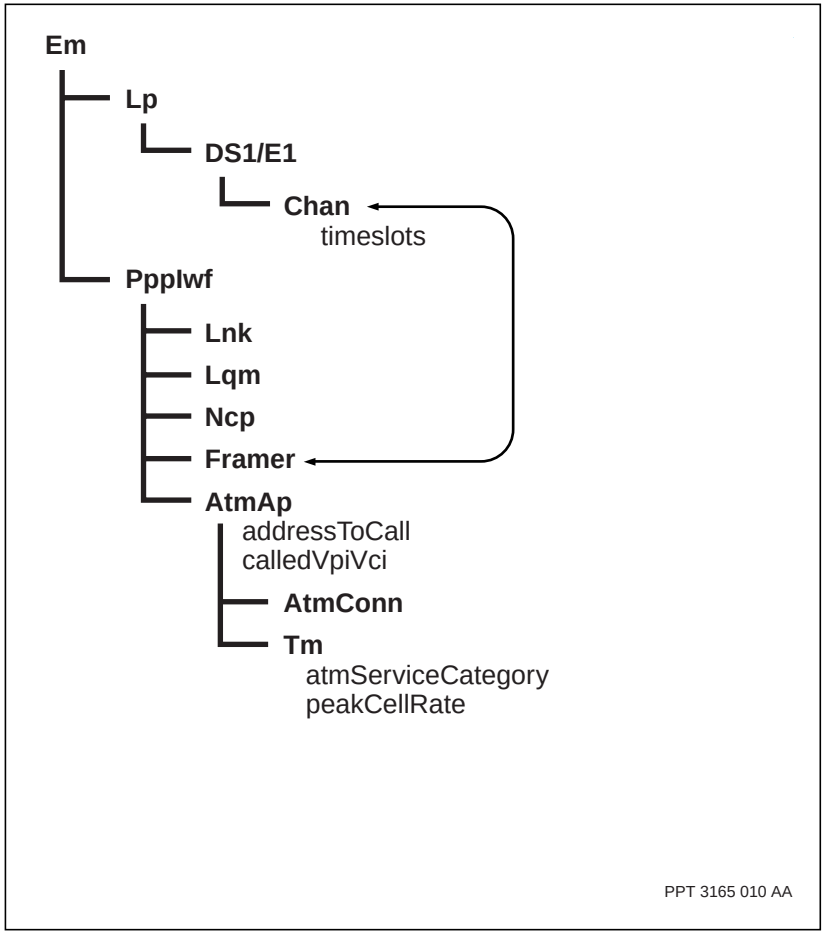
```
set PppIwf/1100 PppIwf/211 atm tm peakCellRate
<pcrValue>
```

Variable definitions

Variable	Value
<atmServiceCategory>	cbr, rtVbr, nrtVbr, or ubr.
<Ds1>	The instance number of the DS1 tributary, having a value between 0 and 31.
<DS_Chan1>	The instance number of the DS1 channel, having a value between 0 and 23.
<E1>	The instance number of the E1 channel, having a value between 0 and 31.
<E1_Chan1>	The instance number of the E1 channel, having a value between 0 and 31.
<Lp1>	The instance number of the logical processor associated with the function processor.
<MSA32cardType>	Either 32pDS1MSA32 or 32pE1MSA32.
<pcrValue>	The peak cell rate value.

Procedure job aid

Figure 21
Configuring PPP/ATM interworking for Passport 7400 component hierarchy



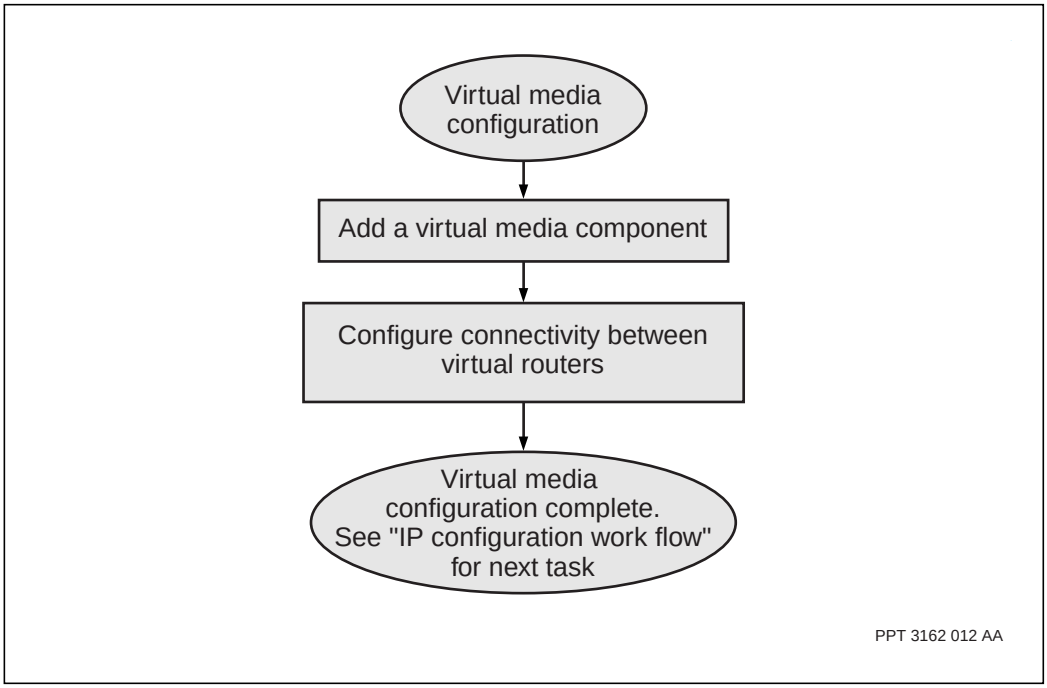
Chapter 9
Virtual media configuration

Configure virtual media to connect two or more virtual routers on the same Passport node.

Virtual media configuration task flow

This task flow shows you the sequence of procedures you perform to configure virtual media on Passport. To link to any procedure, go to “Navigation links” (page 110).

Figure 22
Virtual media configuration task flow



Navigation links

- “Adding a virtual media component” (page 111)
- “Configuring connectivity between virtual routers” (page 113)
- For information about the next task, see “IP configuration work flow” (page 38)

Adding a virtual media component

Add a virtual media to provide virtual, rather than physical, next-hop connectivity between VRs, or if you intend to configure an always-up IP interface for RIP, OSPF, or BGP-4.

Procedure steps

1. Add one or more *Vm* component(s).

   ```
   add Vm/<m>
   ```

2. If required, add another *If* component as a subcomponent of the *Vm* component.

   ```
   add Vm/<m> If/<n>
   ```

3. If required, set the interface mode.

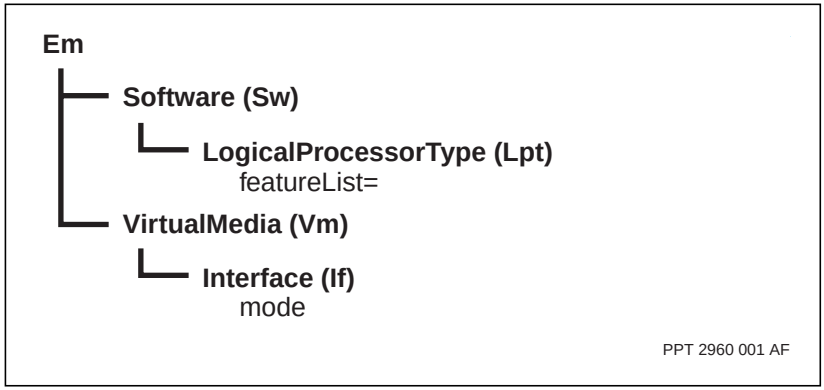
   ```
   set Vm/<m> If/<n> mode <interface_mode>
   ```

Variable definitions

Variable	Value
<interface_mode>	The interface mode. The interface modes alwaysUpInterface and alwaysUpSummary are similar. However when linked to the protocol port of a VR as described in "Configuring connectivity between virtual routers" (page 113), alwaysUpInterface can be configured for a host address or host on a subnet address and alwaysUpSummary can be configured only for a subnet address.
<m>	The instance value of the *Vm* component (decimal 0..15). When you add a *Vm* component, the Passport system automatically creates an *If* subcomponent, and assigns it an instance value of 0. For example, if you add *Vm*/0, the system automatically creates *If*/0 (*Vm*/0 *If*/0).
<n>	The number you assign to the new instance of the *If* component (decimal 1..15). You can add additional *If* components (up to a maximum of 16, including *If*/0) to the *Vm* component.

Procedure job aid

Figure 23
Configuring virtual media component hierarchy



Configuring connectivity between virtual routers

Configure connectivity between two different VRs on the same Passport node to allow them to communicate.

Prerequisites

- Two virtual routers and their protocol ports have been provisioned. See "IP capabilities configuration on Passport" (page 115).

Procedure steps

1 Link one *Vm If* component to the protocol port of a VR.

```
set Vm/<m> If/<n> linktoProtocolPort Vr/<vr_name1> Pp/
<pp_name1>
```

Note: To enable connectivity between different VRs, the *Vm* component instance must be the same for the VRs you want to connect, but the *Vm If* component instances must be different.

2 Link the next *Vm If* component to the protocol port of a VR.

```
set Vm/<m> If/<n+1> linktoProtocolPort Vr/<vr_name2>
Pp/<pp_name2>
```

For example, if you want to enable communication between VR 1 and VR 2, use the following commands:

```
set Vm/0 If/0 linktoProtocolPort Vr/1 Pp/5
```

```
set Vm/0 If/1 linktoProtocolPort Vr/2 Pp/3
```

3 Set the *mode* attribute for both *Vm If* components to enable inter-VR connectivity.

```
set Vm/<m> If/<n> mode interVrConnection
```

```
set Vm/<m> If/<n+1> mode interVrConnection
```

Variable definitions

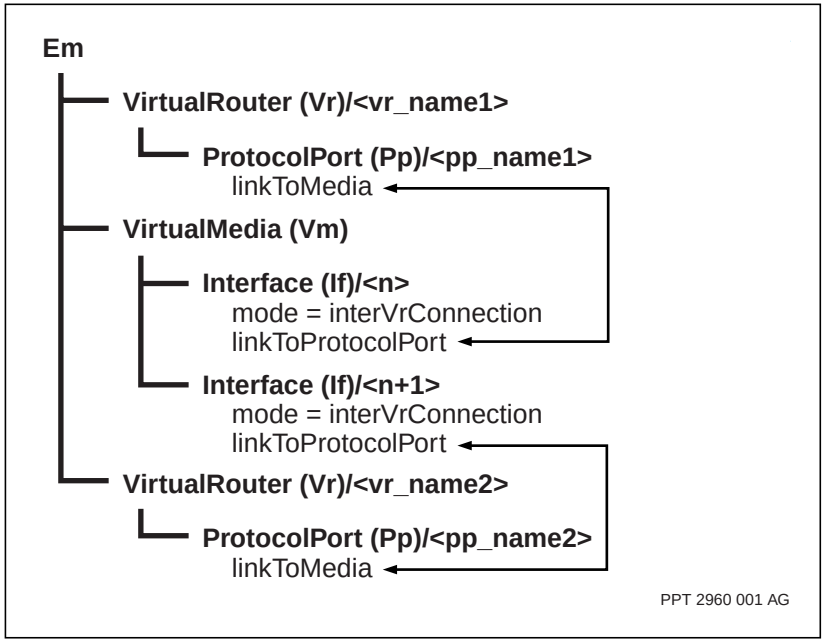
Variable	Value
<m>	The instance value of the virtual media component.
<n>	The instance value of the interface component.
<pp_name1>	The name of the protocol port on the first virtual router.

(Sheet 1 of 2)

Variable	Value
<pp_name2>	The name of the protocol port on the second virtual router.
<vr_name1>	The name of the first virtual router.
<vr_name2>	The name of the second virtual router.
(Sheet 2 of 2)	

Procedure job aid

Figure 24
Configuring connectivity between virtual routers component hierarchy



Chapter 10
IP capabilities configuration on Passport

Configure the Passport to provide IP capabilities.

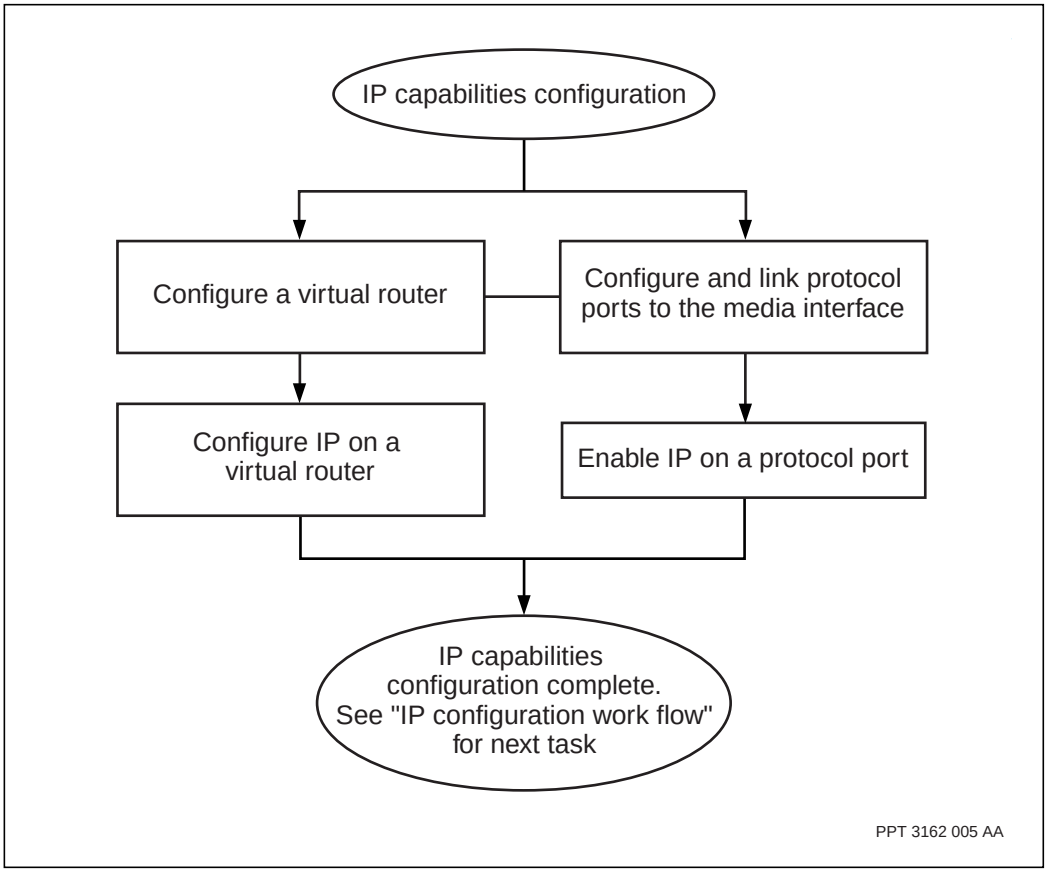
Prerequisites to IP capabilities configuration

- Download all required software applications. See 241-5701-270 *Passport 7400, 15000, 20000 Software Installation Guide*. For information on software applications and their associated feature names for IP on Passport, see 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.
- The 2-port 100baseT FP supports up to two VRs, one per port

IP capabilities configuration task flow

This task flow shows you the sequence of procedures you perform to configure IP capabilities on Passport. To link to any procedure, go to “Navigation links” (page 116).

Figure 25
IP capabilities configuration task flow



Navigation links

- “Configuring a virtual router” (page 118)
- “Configuring IP on a virtual router” (page 120)
- “Configuring and linking a protocol port to a media interface” (page 122)
- “Enabling IP on a protocol port” (page 124)
- “Associating a single IP logical interface with a single subconnection” (page 126)

- For information about the next task, see “IP configuration work flow” (page 38)

Configuring a virtual router

Configure a virtual router on Passport to emulate a physical router in software.

Prerequisites

CAUTION

Moving, deleting, or locking the management VR

The first VR that you create on a Passport node is, by default, the management VR. Once you have activated your provisioning (edit) view, you cannot designate any other VR on the node as the management VR. Deleting or locking the management VR once you have activated your provisioning (edit) view results in loss of connectivity to the Passport node.

- Where Passport supports multiple VRs, choose a name that easily identifies each VR.

Procedure steps

1 Add a *Vr* component:

```
add Vr/<vr_name>
```

2 Specify where the virtual router resides:

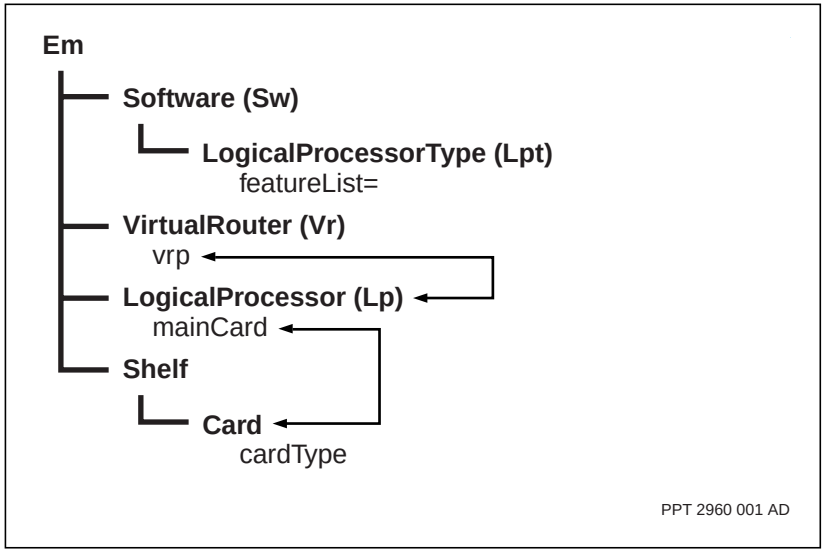
```
set Vr/<vr_name> vrp lp/<vr_lp>
```

Variable definitions

Variable	Value
<vr_name>	The name of the virtual router.
<vr_lp>	The instance value of the logical processor that is linked to the shelf card on which the virtual router resides.

Procedure job aid

Figure 26
Configuring a virtual router component hierarchy



Configuring IP on a virtual router

Configure IP on a virtual router to give IP connectivity to a Passport node.

Procedure steps

1. Add an *Ip* component as a subcomponent of the *Vr* component:

   ```
   add Vr/<vr_name> Ip
   ```

2. If you want to change the default cache table size, provision a cache table size for a single LP:

   ```
   set Vr/<vr_name> Ip cacheTableSize <lp_id>
   <cache_size>
   ```

3. If you want to set the source route attribute:

   ```
   set Vr/<vr_name> Ip sourceRoute <SR_option>
   ```

4. To configure the DSCP value for locally generated BGP, RIP, and OSPF packets, set the *dscpRoutingSource* attribute.

   ```
   set Vr/<vr_name> Ip dscpr <dscp_value>
   ```

5. To configure the DSCP value for any locally generated packets other than BGP, RIP, or OSPF, set the *dscpGeneralSource* attribute.

   ```
   set Vr/<vr_name> Ip dscpg <dscp_value>
   ```

Variable definitions

Variable	Value
<cache_size>	The value of the cache size you want to provision for a particular LP. If you choose not to use the default values, you must make cache table size adjustments in multiples of 100. For more information on managing the cache table, see the section on cache table size in 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.
<lp_id>	The instance value assigned in the IP subcomponent to a particular LP.

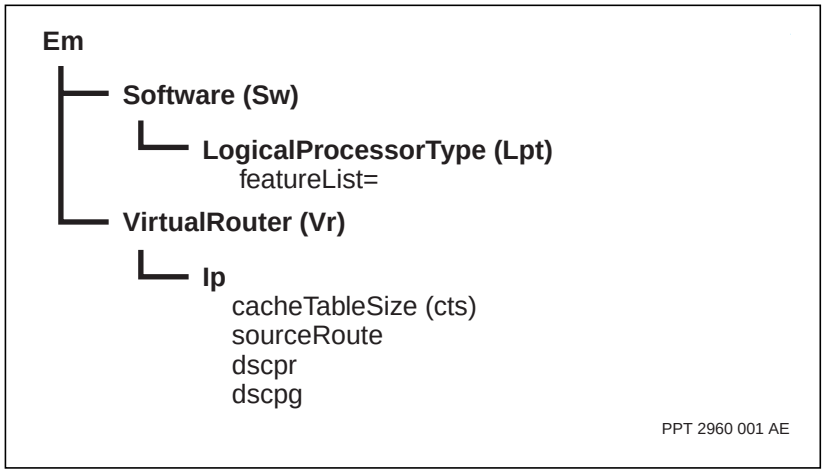
(Sheet 1 of 2)

Variable	Value
<SR_option>	The provisionable attribute *sourceRoute* under the *Ip* component allows you to enable or disable the processing of input datagrams that have a source route IP option on a VR basis. The default value is disabled. For more information, see 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.
<vr_name>	The name of the virtual router.
<dscp_value>	The specific DSCP value you wish to assign for that packet type (Default = 0(df) for dscpg and 48 (cs6) for dscpr).

(Sheet 2 of 2)

Procedure job aid

Figure 27
Configuring IP on a virtual router component hierarchy



Configuring and linking a protocol port to a media interface

Configure and link a protocol port to a media interface to represent a physical instance of a data link or media protocol. You can configure multiple protocol ports on a virtual router.

Procedure steps

1. Add a *ProtocolPort* component as a subcomponent of the *Vr* component:

   `add Vr/<vr_name> ProtocolPort/<pp_name>`

2. Configure the association between the media interface and the *ProtocolPort* component by setting the *linkToProtocolPort* attribute:

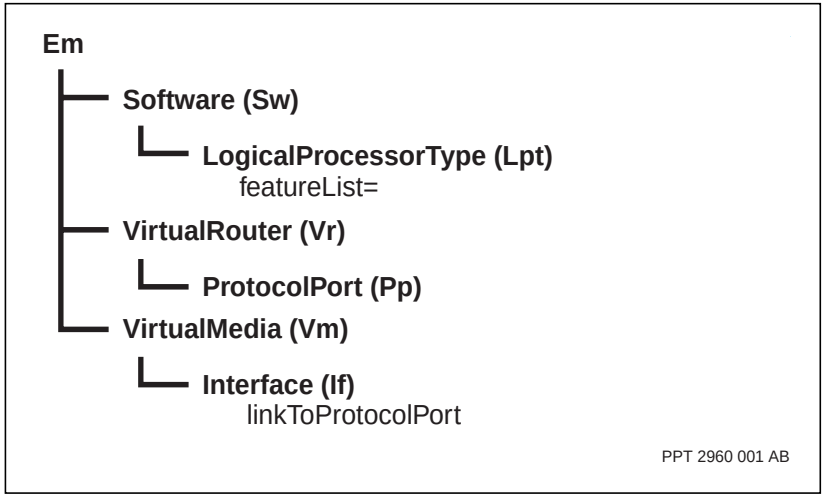
   `set <media_interface> linkToProtocolPort Vr/<vr_name> ProtocolPort/<pp_name>`

Variable definitions

Variable	Value
<media_interface>	The interface name created during the provisioning of a particular medium. Application name values are formatted as a component type/instance value, for example, LA/31. If you are configuring IP over frame relay using frame relay DTE, the media interface is a frame relay remote group, for example, *FrDte/<n> Rg/1*. For more information, see "Frame relay DTE configuration for IP over frame relay" (page 53). If you are configuring IP over frame relay using an IP-optimized DLCI, the media interface is an IP DLCI group, for example, *IpDlciGroupr/<n>*. For more information, see "IP-optimized DLCI configuration for IP over frame relay" (page 71). If you are configuring an IP tunnel, the media interface is the IP tunnel interface, for example, *Vr/<vr_name> IP Tunnel Sep/<sep_id>*. For more information, see "Configuring point-to-point tunnels" (page 309).
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 28
Configuring and linking protocol ports to the media interface component hierarchy



Enabling IP on a protocol port

Add an *IpPort* component to the *ProtocolPort* component to enable IP routing on that port.

Prerequisites

- Specify at least one IP address for the protocol port through the *LogicalIf* subcomponent to enable IP to function.
- Specify an IP address and a network mask to identify the IP subnet to which the IP interface belongs.

Procedure steps

1 Add an *IpPort* component as a subcomponent of the *ProtocolPort* component.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
```

2 Add an IP logical interface.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr>
```

Note: You can configure an IP logical interface against a specific media subconnection, which gives each subconnection its own subnet. See “Associating a single IP logical interface with a single subconnection” (page 126) to complete this procedure.

3 Provision a network mask for the protocol port. For networks that contain routers running RIP version I, the netmask should be the same for all RIP interfaces.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> netMask <netmask>
```

4 Provision a broadcast address for the *ProtocolPort* component.

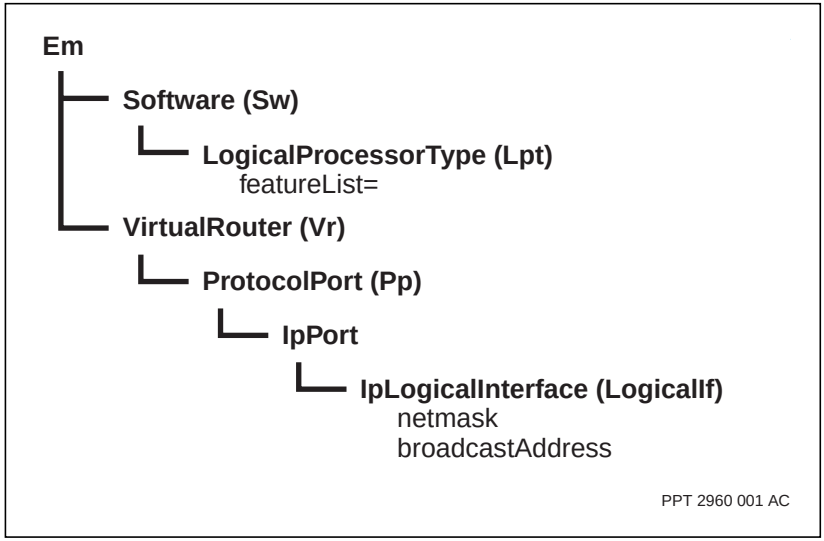
```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> broadcastAddress <broadcast_addr>
```

Variable definitions

Variable	Value
<broadcast_addr>	The broadcast address of the attached IP network or subnetwork.
<ip_addr>	The 32-bit IP address assigned to this logical interface.
<netmask>	The network mask to be used with the IP address.
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 29
Enabling IP on a protocol port component hierarchy



Associating a single IP logical interface with a single subconnection

Associate a single IP logical interface with a single subconnection when you want each subconnection to support a specific IP subnet.

Prerequisites

- The required access media has been provisioned. See one of the following:
  - — "Frame relay DTE configuration for IP over frame relay" (page 53)
  - — "IP-optimized DLCI configuration for IP over frame relay" (page 71)
  - — "ATM MPE configuration for IP over ATM" (page 41)

Procedure steps

1 Configure the association between the IP logical interface and the subconnection.

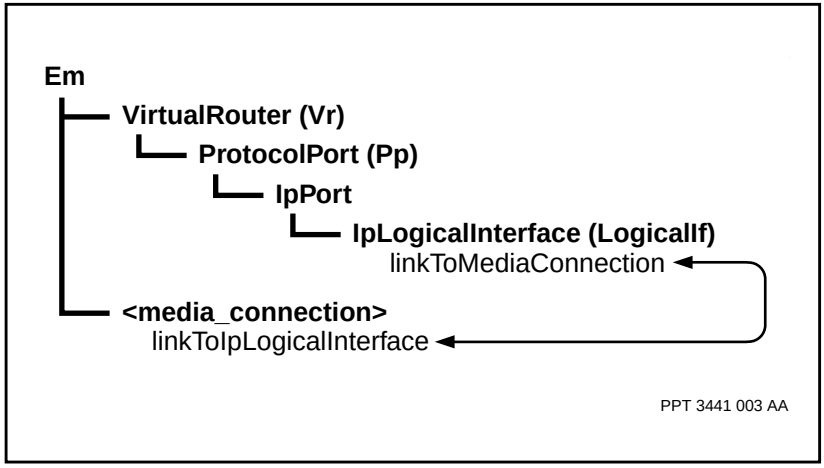
```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ip_addr> linkToMediaConnection
<media_subconnection>
```

Variable definitions

Variable	Value
<ip_addr>	The 32-bit IP address assigned to this logical interface.
<media_subconnection>	The required media subconnection, which can be • for FR DTE, component *FrDte StaticDlci* • for IP-optimized DLCI, component *IpDlciGroup Frc* • for ATM MPE, component *AtmMpe Ac*
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 30
Associating a single IP logical interface with a single subconnection component hierarchy



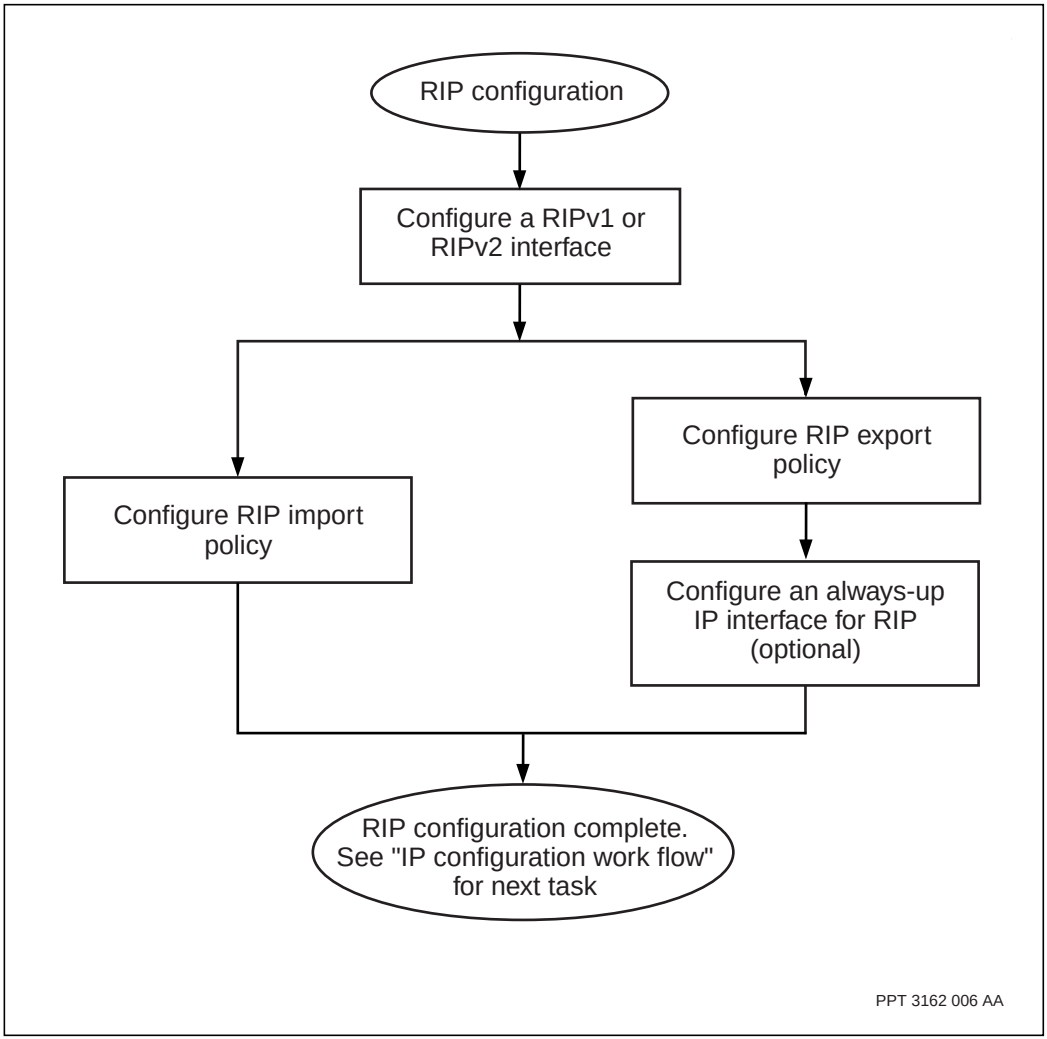
Chapter 11
Routing information protocol (RIP) configuration

Configure the routing information protocol (RIP) to exchange routing information within a network or between networks using RIP.

RIP configuration task flow

This task flow shows you the sequence of procedures you perform to configure RIP. To link to any procedure, go to “Navigation links” (page 130).

Figure 31
RIP configuration task flow



Navigation links

- “Configuring a routing information protocol (RIP) interface” (page 132)
- “Configuring RIP import policy” (page 135)
- “Configuring RIP export policy” (page 137)

- “Configuring an always-up IP interface for RIP” (page 140)
- “Migrating from RIPv1 to RIPv2” (page 142)
- For information about the next task, see “IP configuration work flow” (page 38)

Configuring a routing information protocol (RIP) interface

Configure a RIPv1 or RIPv2 interface to connect the protocol port on the Passport node to your network.

Prerequisites

- Configure protocol ports for each interface to be included in RIP routing exchanges. See “Configuring and linking a protocol port to a media interface” (page 122).
- Configure *LogicalIf* components for each interface to be included in RIP routing exchanges. See “Enabling IP on a protocol port” (page 124).

CAUTION

Changing the value of the *ifConfSend* or *ifConfReceive* attribute

Changing the value of the *IfConfSend* or *ifConfReceive* attribute will cause a brief interruption of service on the interface. For information on the behavior of these attributes, see 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.

Procedure steps

1 Add a *Rip* component as a subcomponent of the *Ip* component:

```
add Vr/<vr_name> Ip Rip
```

2 If required, change the route preference. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information.

```
set Vr/<vr_name> Ip Rip defaultRipRtePref <route_pref>
```

3 Add a *RipIf* component to at least one logical interface:

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf
```

4 Set the version of RIP updates to send from this RIP interface.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf ifConfSend <tx_value>
```

5 Set the version of RIP updates to receive (accept) on this RIP interface.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf ifConfReceive <rx_value>
```

6 Optionally, add the *Neighbor* subcomponent. The *Neighbor* subcomponent of the *RipIf* component describes the RIP neighbor for this logical interface. If this protocol port is configured as non-broadcast/multi-access (NBMA), as specified by the *lanModel* attribute under the *IpPort* subcomponent, then the *Neighbor* subcomponent must be provisioned.

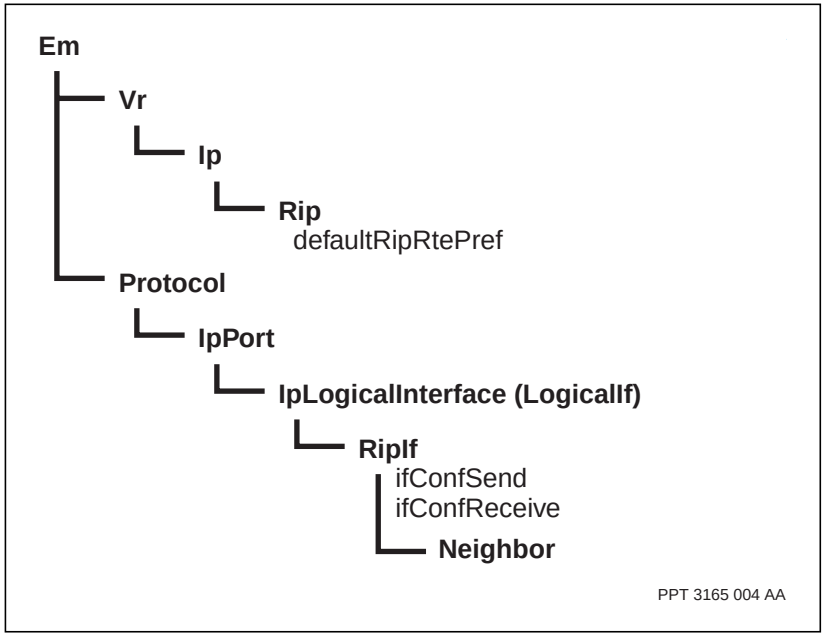
```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> RipIf Neighbor/
<remote_IpAddress>
```

Variable definitions

Variable	Value
<ipAddress>	The IP address of the local interface.
<pp_name>	The name of the protocol port.
<remote_IpAddress>	The IP address of the remote neighbor interface.
<route_pref>	The route preference. The attribute range is 1 to 254. The attribute default is 82.
<rx_value>	The value you assign to the *ifConfReceive* attribute. This can be *v1* (allow reception of RIP 1 packets only), *v2* (allow reception of RIP 2 packets only), *both* (allow reception of both RIP 1 and RIP 2 packets), or *reject* (block receipt of RIP packets). The default value is *both*.
<tx_value>	The value you assign to the *ifConfSend* attribute. This can be *v1* (send RIP 1 packets only), *v2* (support for RIP 2 multicast mode), *v2b* (support RIP 2 broadcast mode), or *silent* (block transmission of RIP packets). The default value is *v2b*.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 32
Configuring a RIP interface component hierarchy



Configuring RIP import policy

Configure RIP import policy to define which learned set of routing information is given to the RIP routing process, which metrics the RIP routing process uses, and which routing processes provide routing information to the RIP routing process.

Procedure steps

1. Add an *Import* component to the *Rip* component:

   ```
   add Vr/<vr_name> Ip Rip Import/<import_policy_number>
   ```

2. Set the *usageFlag* attribute for the *Import* component to enable or disable the import policy:

   ```
   set Vr/<vr_name> Ip Rip Import/<import_policy_number>
   usageflag <usageflag_toggle>
   ```

3. Set the *interface* attribute if you want all routes learned from RIP updates on a particular interface to be entered in the routing table:

   ```
   set Vr/<vr_name> Ip Rip Import/<import_policy_number>
   interface <ipAddress>
   ```

4. Set the *neighbor* attribute if you want all routes learned from a particular neighbor to be installed in the routing table:

   ```
   set Vr/<vr_name> Ip Rip Import/<import_policy_number>
   neighbor <ipAddress>
   ```

5. Set an import metric if you do not want to use the default import metrics:

   ```
   set Vr/<vr_name> Ip Rip Import/<import_policy_number>
   importMetric <cost>
   ```

6. Configure the *Network <ipaddress> <ipmask>* component if you want to restrict the effects of this import policy to specified network ranges. You can set more than one pair of *<ipaddr> <ipmask>*.

   ```
   add Vr/<vr_name> Ip Rip Import/<import_policy_number>
   Network/<net_instance>
   ```

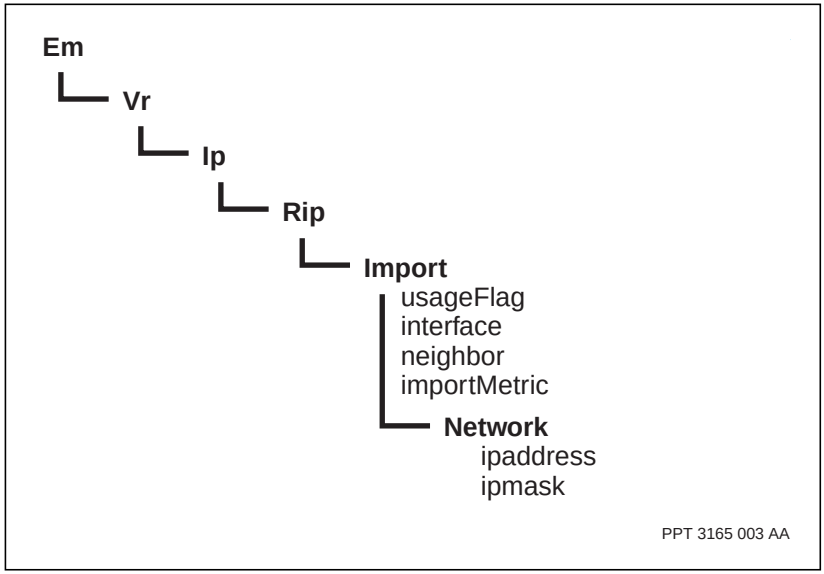
   ```
   set Vr/<vr_name> Ip Rip Import/<import_policy_number>
   Network/<net_instance> ipaddress <ip_addr>, ipmask
   <ip_mask>
   ```

Variable definitions

Variable	Value
<cost>	A decimal metric cost for those interfaces or neighbors that match the criteria defined in this import policy.
<import_policy_number>	The numeric designation assigned to the import policy.
<ip_addr>	The IP address of the network.
<ipAddress>	The IP address of the interface from which the routing information is derived.
<ip_mask>	The mask used by the network.
<net_instance>	The numeric designation assigned to this instance of the *Network* component.
<usageflag_toggle>	Set to either use (enable) or ignore (disable).
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 33
Configuring RIP import policy component hierarchy



Configuring RIP export policy

Configure RIP export policy to define how to advertise routing information on specific interfaces, and which routing processes learned from a specific interface can be exported.

Procedure steps

1 Add an *Export* component to the *Rip* component:

```
add Vr/<vr_name> Ip Rip Export/<export_policy_number>
```

2 Set the *advertiseStatus* attribute for the *Export* component to enable or disable the export policy:

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
advertiseStatus <advertise_toggle>
```

3 Set the *ripInterface* attribute if you want to enable the RIP export policy for this particular interface:

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
ripinterface <ipAddress>
```

Note: If you specify a RIP interface, set the *protocol* attribute to all or rip when you complete step 4.

4 Set the *protocol* attribute if you want to specify which routing protocols are advertised by RIP:

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
protocol <protocol_type>
```

5 If you want to export bgpInternal routes and you set the *protocol* attribute to all in step 4, set the *redistributeIbgp* attribute as follows:

```
set Vr/<vr_name> Ip Rip redistributeIbgp true
```

Note: If you set the *protocol* attribute to bgpInternal in step 4, you do not need to set the *redistributeIbgp* attribute.

6 Set the *exportMetric* attribute if you want to assign a cost to the routes advertised by RIP. This cost is used to judge the relative value of the routes advertised by this router.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
exportMetric <cost>
```

7 To control the usage of specified networks, configure the *Network <ipaddrress> <ipmask>* subcomponent to restrict the effects of this export

policy to the specified network ranges. You can list more than one pair of *<ipaddr> <ipmask>*.

```
add Vr/<vr_name> Ip Rip Export/<export_policy_number1>
Network/<net_instance>
```

8 To block outbound routing information for a specific network, enter

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number1>
outif <interface_instance>, advertiseStatus block,
protocol rip
```

9 To specify OSPF learned routes to be exported, enter

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number2>
protocol ospfExternal
```

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number1>
Network/<net_instance> ipaddress <ip_addr>, ipmask
<ip_mask>
```

Variable definitions

Variable	Value
<advertise_toggle>	Set to either send (enable) or block (disable).
<cost>	A decimal metric cost for routes advertised on those interfaces or to those neighbors that match the defined criteria in this export policy. Change the metric for all routing entries that match the policy criteria to this metric cost, default=0 <use routing table metric>.
<export_policy_number>	The numeric designation assigned to the export policy.
<interface_instance>	The IP address of the interface.
<ip_addr>	The IP address of the network.
<ipAddress>	The IP address of the interface.
<ip_mask>	The mask used by the network.
<net_instance>	The numeric designation assigned to this instance of the *Network* component.

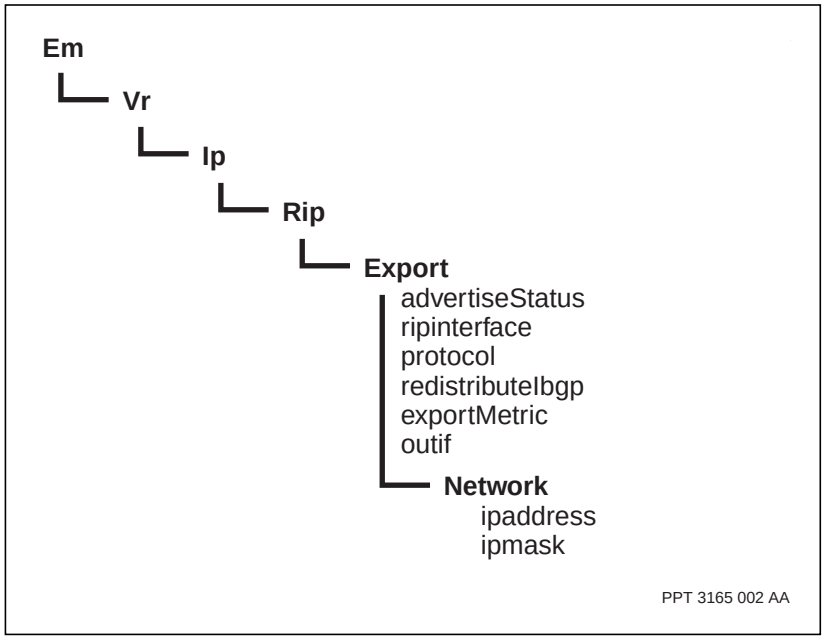
(Sheet 1 of 2)

Variable	Value
<protocol_type>	All, egp, rip, ospfInternal, ospfExternal, staticLocal, staticRemote, bgpInternal, or bgpExternal. If you set the *protocol* attribute to all, the export policy applies to all routes in the forwarding table except bgpInternal and bgpExternal. If you want to export bgpInternal routes, create an export policy with the *protocol* attribute set to bgpInternal, or see step 5. If you want to export bgpExternal routes, create an export policy with the *protocol* attribute set to bgpExternal.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 34
Configuring RIP export policy component hierarchy



Configuring an always-up IP interface for RIP

Prerequisites

- "Virtual media configuration" (page 109)
- Configure an export policy for RIP. For more information, see "Configuring RIP export policy" (page 137).

Procedure steps

1 Set the *protocol* attribute to *staticLocal*.

```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
protocol staticLocal
```

2 Add the *Network* component.

```
add Vr/<vr_name> Ip Rip Export/<export_policy_number>
Network/<net_instance>
```

3 Set the ip address and mask to match against.

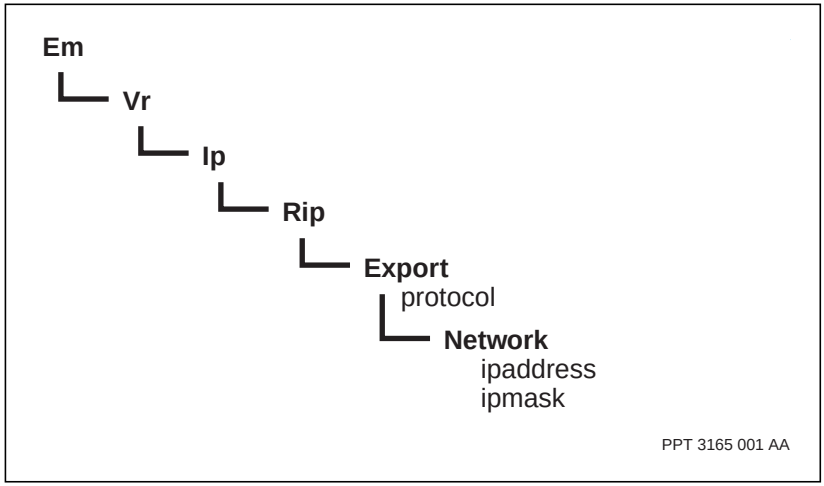
```
set Vr/<vr_name> Ip Rip Export/<export_policy_number>
Network/<net_instance> ipaddress <ip_addr>, ipmask
<ip_mask>
```

Variable definitions

Variable	Value
<export_policy_number>	The numeric designation assigned to the export policy.
<ip_addr>	The same IP address as that of the *LogicalIf* component of the VR's IP port.
<ip_mask>	The mask used by the network.
<net_instance>	The numeric designation of the *Network* component (decimal 0..65535).
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 35
Configuring an always-up IP interface for RIP component hierarchy



Migrating from RIPv1 to RIPv2

You can optimize the RIP configuration by migrating from RIPv1 to RIPv2.

Prerequisites

- To migrate the Passport nodes in a network from RIPv1 to RIPv2, all Passport nodes in the network must be running a release of software that supports RIPv2 (R5.1 and later).

Procedure steps

Migrate all Passport nodes on a link-by-link basis, until all the nodes in the network are set to RIPv2. The steps correspond to the figure "Example migration from RIPv1 to RIPv2 using two Passport nodes" (page 143).

1 Add RIPv2 to Passport 2, but provision the RIP interface on Passport 2 to be backwards compatible with RIPv1 on Passport 1.

For example, set the ifConfSend attribute on the RIP interface of Passport 2 to v2b.

See 241-5701-810 *Passport 7400, 15000, 20000 Configuring IP* for details on configuring the *ifConfSend* and *ifConfReceive* attributes under the *RipIf* component. See the table "Example migration: RIP behavior on two Passport nodes with different RIP configuration" (page 144) for the meaning of the values of the *ifConfSend* and *ifConfReceive* attributes.

The table "Example migration: RIP behavior on two Passport nodes with different RIP configuration" (page 144) also illustrates the behavior of the RIP interface on these two nodes for different combinations of attribute values provisioned for the *ifConfSend* and *ifConfReceive* attributes. This table can be useful in helping you planning your migration. The attribute values appear in the table in italics.

2 Configure the RIP interface on Passport 1 to support RIPv2 only.

For example, set the ifConfSend attribute on the RIP interface of Passport 1 to v2, and the ifConfReceive attribute to v2.

3 Change the RIP interface on Passport 2 to support RIPv2 only.

For example, set the ifConfSend attribute on the RIP interface of Passport 2 to v2, and the ifConfReceive attribute to v2.

4 Remove all RIPv1 components from Passport 2.

Procedure job aid

Figure 36
Example migration from RIPv1 to RIPv2 using two Passport nodes

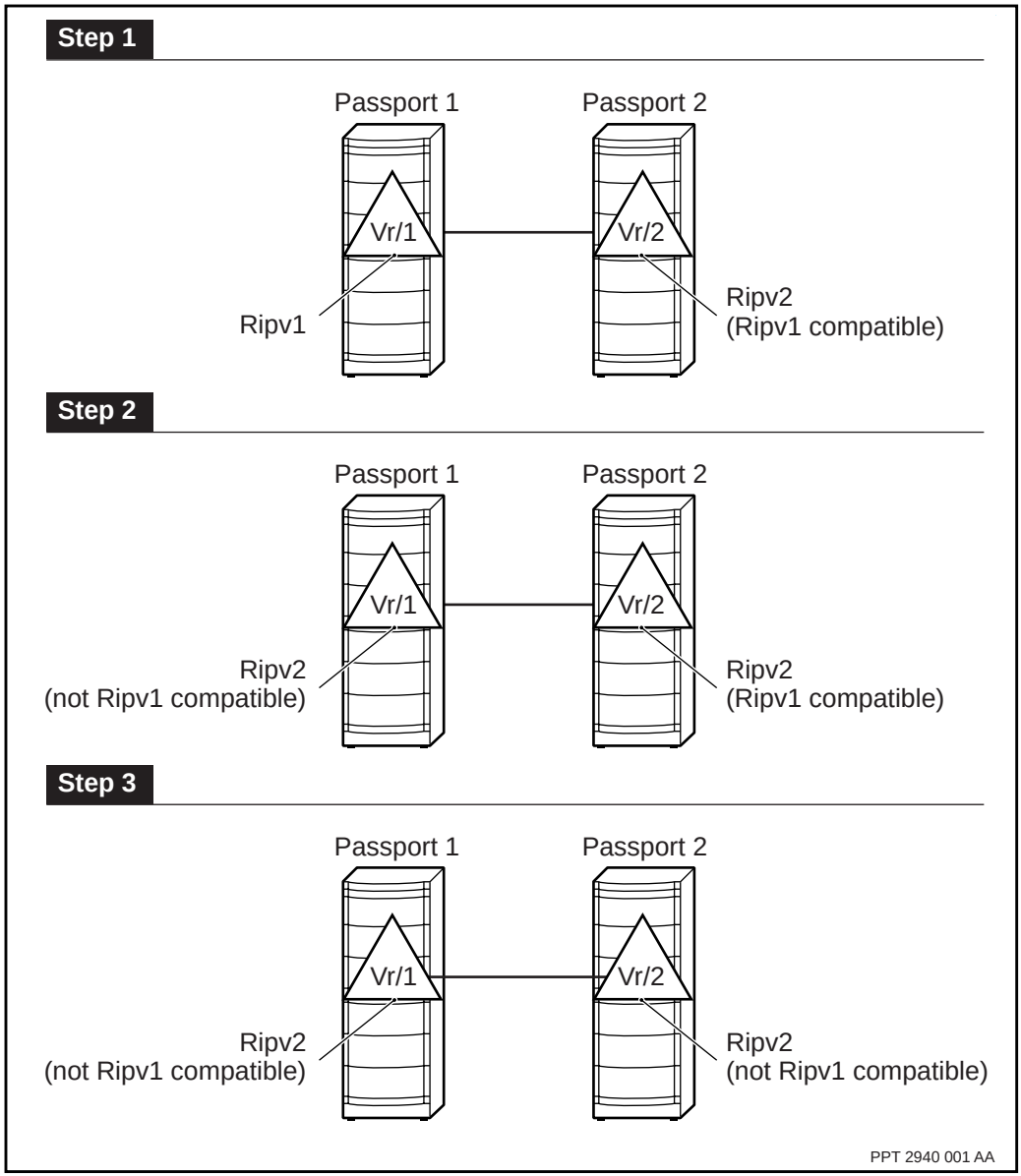


Table 6
Example migration: RIP behavior on two Passport nodes with different RIP configuration

ifConfSend attribute value on Passport 1 (Vr/1) (transmitting)	**ifConfReceive attribute value on Passport 2 (Vr/2) (receiving)**			
	***v1* (RIP 1)**	***v2* (RIP 2)**	***both* (RIP 1 or 2)**	***reject* (do not accept)**
silent (do not send)	No transmission	No transmission	No transmission	No transmission/ updates are rejected.
v1 (RIP 1)	RIP 1 updates broadcast by Vr/1. RIP 1 updates accepted by Vr/2.	RIP 1 updates broadcast by Vr/1. RIP 1 updates rejected by Vr/2.	RIP 1 updates broadcast by Vr/1. RIP 1 updates accepted by Vr/2. The Vr/2 RIP interface processes the updates as RIP 1 updates.	RIP 1 updates broadcast by Vr/1. RIP 1 updates are rejected by Vr/2.
v2b (RIP 1 compatible)	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. The Vr/2 RIP interface processes the RIP 2 updates as RIP 1 updates. (The Vr/2 RIP interface ignores the subnet mask and next hop fields in the RIP 2 update.)	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates broadcast by Vr/1. RIP 2 updates accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates broadcast by Vr/1. RIP 2 updates are rejected by Vr/2.

(Sheet 1 of 2)

Table 6 (continued)
Example migration: RIP behavior on two Passport nodes with different RIP configuration

ifConfSend attribute value on Passport 1 (Vr/1) (transmitting)	**ifConfReceive attribute value on Passport 2 (Vr/2) (receiving)**			
	***v1* (RIP 1)**	***v2* (RIP 2)**	***both* (RIP 1 or 2)**	***reject* (do not accept)**
v2 (RIP 2)	RIP 2 updates are multicast by Vr/1. Because the Vr/2 RIP interface is set for RIP 1 only, Vr/1 will not send RIP 2 updates to Vr/2.	RIP 2 updates are multicast by Vr/1. RIP 2 updates are accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	RIP 2 updates are multicast by Vr/1. RIP 2 updates are accepted by Vr/2. Vr/2 RIP interface will process the subnet mask and next hop fields in the RIP 2 updates.	Updates are rejected

(Sheet 2 of 2)

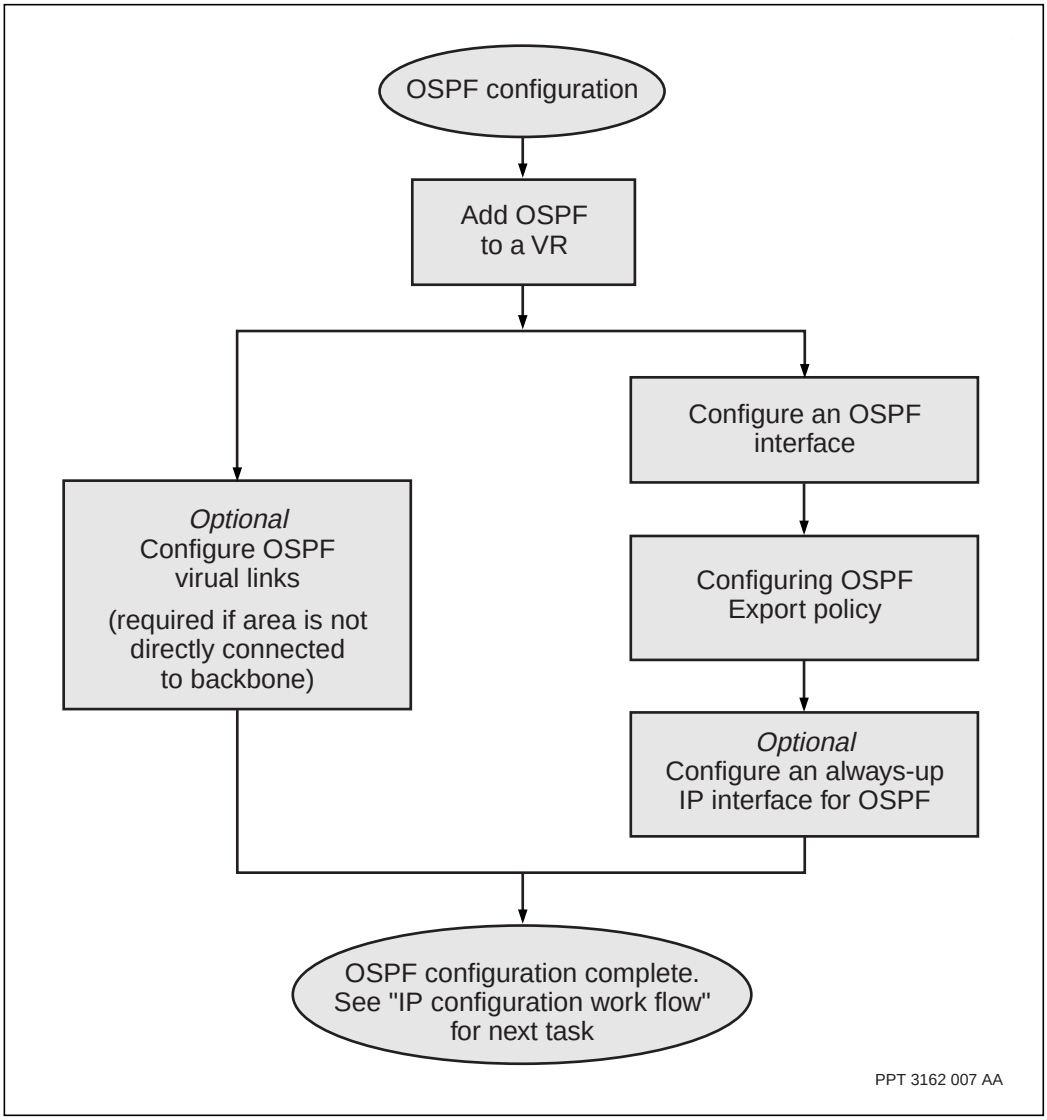
Chapter 12
Open shortest path first (OSPF) configuration

Configure the open shortest path first (OSPF) routing protocol to exchange routing information within a network or between networks using OSPF.

OSPF configuration task flow

This task flow shows you the sequence of procedures you perform to configure OSPF. To link to any procedure, go to “Navigation links” (page 148).

Figure 37
OSPF configuration task flow



Navigation links

- “Adding OSPF to a virtual router” (page 150)

- “Configuring an OSPF interface” (page 155)
- “Configuring OSPF export policy” (page 159)
- “Configuring an always-up IP interface for OSPF” (page 162)
- “Configuring OSPF virtual links” (page 153)
- For more information about the next task, see “IP configuration work flow” (page 38)

Adding OSPF to a virtual router

Add OSPF to a virtual router to enable you to connect the Passport node to a customer network through the protocol port.

Prerequisites

- If you are going to set attribute *Vr Ip Ospf spareInstance* to enable in step 4 of the procedure, first set attribute *Shelf cpEquipmentProtection* to hot. Be aware that setting this attribute to hot causes the spare CP to restart.

Procedure steps

1 Add an *Ospf* component as a subcomponent of the *Ip* component.

```
add Vr/<vr_name> Ip Ospf
```

2 Specify the IP address of the OSPF instance to identify it in the AS.

```
set Vr/<vr_name> Ip Ospf routerId <x.x.x.x>
```

3 Specify the estimated counts if you do not want to use the system defaults. Configuring the value of these attributes to accurate estimates improves performance and does not limit functionality.

```
set Vr/<vr_name> Ip Ospf
estimatedNumberOfInternalOSPFRoutes <int_routes>,
estimatedNumberOfExternalOSPFRoutes <ext_routes>,
estimatedNumberOfInterfacesPerArea <ifs>,
estimatedNumberOfNeighborsPerInterface <nbrs>
```

4 Set the spare instance attribute according to whether you want a synchronized OSPF instance maintained on the standby card.

```
set Vr/<vr_name> Ip Ospf spareInstance
<sparing_action>
```

5 If required, change the default setting for the alarm generator, which specifies what OSPF alarms are generated.

```
set Vr/<vr_name> Ip Ospf alarmGeneration <alarm>
```

6 If required, change the default for the Dijkstra timer.

```
set Vr/<vr_name> Ip Ospf spfHoldTime <hold_time>
```

7 If required, change the default for the ECMP setting.

```
set Vr/<vr_name> Ip Ospf ecmpStatus <ecmp>
```

8 If required, change the route preference. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information.

To change the route preference attribute for OSPF internal routes, use the following command:

```
set Vr/<vr_name> Ip Ospf defaultOspfIntRtePref
<route_pref>
```

To change the route preference attribute for OSPF external type 1 routes, use the following command:

```
set Vr/<vr_name> Ip Ospf defaultOspfExt1RtePref
<route_pref>
```

To change the route preference attribute for OSPF external type 2 routes, use the following command:

```
set Vr/<vr_name> Ip Ospf defaultOspfExt2RtePref
<route_pref>
```

Variable definitions

Variable	Value
<alarm>	The level of alarm generation for this instance of OSPF.
<ecmp>	Specifies whether equal cost multi-path (ECMP) is enabled for OSPF learned routes.
<ext_routes>	The average estimated number of external routes to be exported into OSPF.
<hold_time>	The time interval, in seconds, between OSPF Dijkstra calculations for this instance of OSPF. This is an important scaling factor during peak load times.
<ifs>	The average estimated number of interfaces in each area of the AS.
<int_routes>	The average estimated number of internal OSPF routes to be managed by this OSPF instance.
<nbrs>	The average estimated number of neighbors for each OSPF interface.

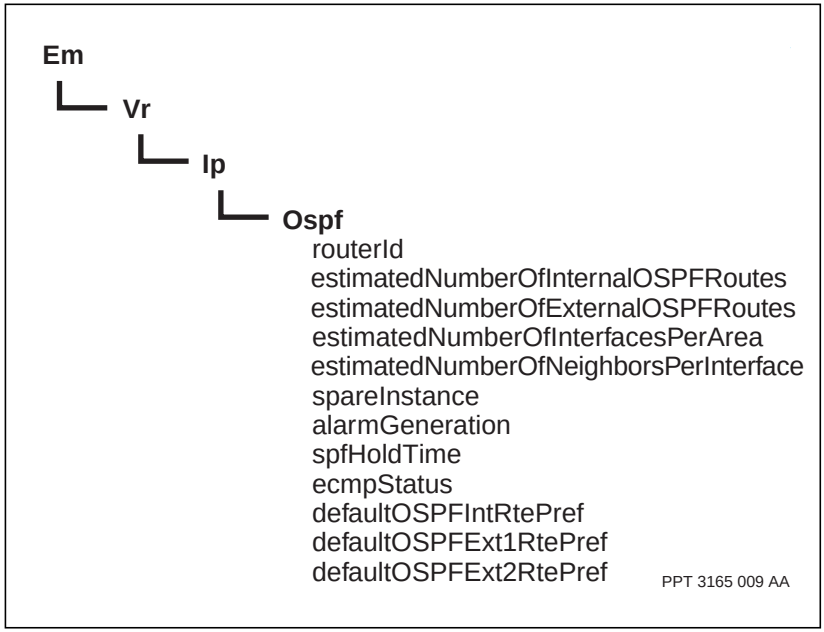
(Sheet 1 of 2)

Variable	Value
<route_pref>	The route preference. The attribute default is 30 for internal routes, 80 for external type 1 routes, and 120 for external type 2 routes.
<sparing_action>	The spare instance setting. When set to disable, a synchronized OSPF instance is not maintained on the standby card.
<vr_name>	The name of the virtual router.
<x.x.x.x>	The 32-bit IP address that uniquely identifies the OSPF router in the AS.

(Sheet 2 of 2)

Procedure job aid

Figure 38
Adding OSPF to a virtual router component hierarchy



Configuring OSPF virtual links

Configure a virtual link to connect an isolated area border router, through an attached area, to the backbone.

Procedure steps

1. Add a *VirtIfEntry* subcomponent under the *Ospf* subcomponent:

   add Vr/<vr_name> Ip Ospf VirtIfEntry/ <localAreaIdIpAddress>,<nbrRouterIdIpAddress>

2. Change the frequency of the OSPF hello protocol exchanges if you do not want to use the hello exchange interval default of 10 seconds. You can choose to adjust the interval depending upon the media bandwidth and traffic conditions. All routers on that media must use the same hello interval.

   set Vr/<vr_name> Ip Ospf VirtIfEntry/ <localAreaIdIpAddress>,<nbrRouterIdIpAddress> helloInterval <interval>

3. Configure the OSPF simple authentication key if you want to provide a higher level of security to OSPF routing exchanges. Ensure that it matches the *authenticationkey* attribute currently in use in the autonomous system, otherwise routing exchanges are discarded.

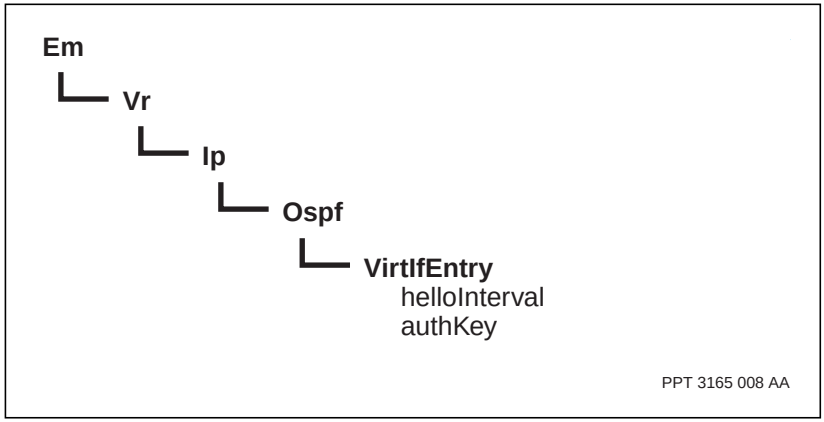
   add Vr/<vr_name> Ip Ospf VirtIfEntry/ <localAreaIdIpAddress>,<nbrRouterIdIpAddress> authKey <authentication_string>

Variable definitions

Variable	Value
<authentication_string>	A hexadecimal string of 1 to 8 bytes (Hex representation of ASCII character set, actually 16 Hex digits).
<interval>	A decimal value from 1 through 3600.
<localAreaIdIpAddress>	The OSPF area ID of the local end of the virtual link.
<nbrRouterIdIpAddress>	The OSPF Router ID of the remote end of the virtual link.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 39
Configuring OSPF virtual links component hierarchy



Configuring an OSPF interface

Configure an OSPF interface to connect the protocol port on the Passport node to your network.

Prerequisites

- It is recommended that each OSPF area on the VR be provisioned on an OSPF interface.
- Configure protocol ports for each interface to be included in OSPF routing exchanges. See “Configuring and linking a protocol port to a media interface” (page 122).
- Configure *LogicalIf* components for each interface to be included in OSPF routing exchanges. See “Enabling IP on a protocol port” (page 124).
- If you are configuring OSPF on IP logical interfaces with links to individual connections (see “Associating a single IP logical interface with a single subconnection” (page 126)), it is recommended that you do not set attribute *Vr ProtocolPort IpPort LogicalIf OspfIf ifType* to broadcast. See step 3 of the procedure.

Procedure steps

1. Add at least one *AreaEntry* component as a subcomponent of the *Ospf* component.

   ```
   add Vr/<vr_name> Ip Ospf AreaEntry/<area_id>
   ```

2. To enable OSPF on Passport, add the *OspfIf* component to each IP logical interface that is taking part in the OSPF process.

   ```
   add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
   LogicalIf/<ipAddress> OspfIf
   ```

3. Select the OSPF interface type.

   ```
   set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
   LogicalIf/<ipAddress> OspfIf ifType <if_type>
   ```

4. Select the mode of operation if the interface type is point-to-multipoint.

   ```
   set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
   LogicalIf/<ipAddress> OspfIf ifType pointToMultipoint
   <mode_of_operation>
   ```

Note: It is necessary to configure the neighbors for a point-to-multipoint interface in a non-broadcast network.

5 Set the *areaId* attribute for the OSPF *LogicalIf* component to define the OSPF area to which the interface connects:

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf areaId <area_id>
```

6 Add a neighbor component for each OSPF neighbor if the interface type is point-to-multipoint non-broadcast or NBMA.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
LogicalIf/<ipAddress> OspfIf Neighbor/<ipAddr>
```

Note: For broadcast networks, the OSPF neighbors are discovered automatically by the Hello protocol.

7 Add a *StubAreaEntry* subcomponent to the *Ospf* component if you want to configure an area as a stub. If you configure an area as a stub, the corresponding *Vr IP Ospf AreaEntry importAsExtern* attribute must be set to *noExternal* or *nssa*.

```
add Vr/<vr_name> Ip Ospf StubAreaEntry/
<stub_area_Id>,<service_type>
```

8 Add an *AreaAggregateEntry* subcomponent to the *Ospf* component if you want to define the extent of summarization for a particular area. This subcomponent is valid only for area border routers.

```
add Vr/<vr_name> Ip Ospf AreaAggregateEntry/
<area_Id>,<lsdbType>,<aggregateNet>,<aggregateMask>
```

9 Add a *HostEntry* component to the *Ospf* component if you want to advertise a particular host address:

```
add Vr/<vr_name> Ip Ospf HostEntry/<ipAd>,<tos>
```

Variable definitions

Variable	Value
<aggregateMask>	A wildcard mask indicating how much of the network and subnetwork address is common to all subnetworks in the area defined by <area_id>.
<aggregateNet>	The 32-bit IP address defining the contiguous network/ subnetwork range to be advertised.

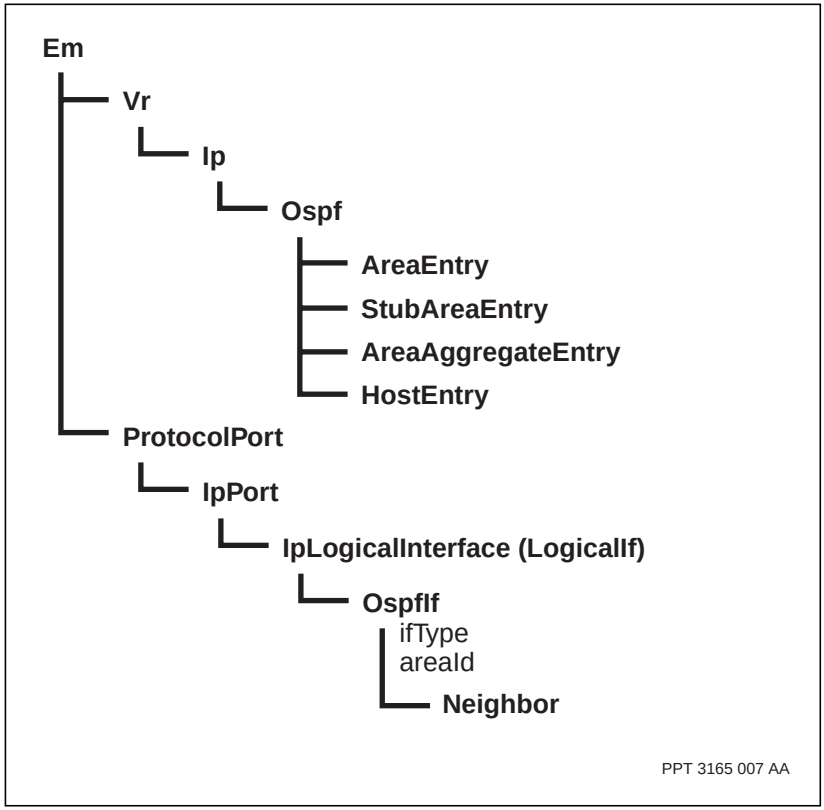
(Sheet 1 of 2)

Variable	Value
<area_id>	The 32-bit IP address for the OSPF area in which this router is located. If you do not specify the area ID, the system will assume that the port is in the backbone, area ID 0.0.0.0.
<if_type>	The type of interface used.
<ipAd>	The host address to be advertised.
<ipAddr>	The 32-bit address assigned to the OSPF neighbor.
<ipAddress>	The 32-bit IP address assigned to this logical interface.
<lsdbType>	The link state advertisement (LSA) type to be used. Type values are summaryLink or nssaExternalLink.
<mode_of_operation>	A value of broadcast or non-broadcast. A value of non-broadcast limits the interface to sending unicast packets whereas a broadcast value enables sending multicast OSPF Hello packets to dynamically discover neighbors.
<pp_name>	The name of the protocol port.
<service_type>	The IP type of service.
<stub_area_Id>	The OSPF area number for the stub network. It must identify an existing *AreaEntry* component.
<tos>	Specifies the ToS byte value assigned to the packet.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 40
Configuring an OSPF interface component hierarchy



Configuring OSPF export policy

Configure OSPF export policy to define how routing information is shared between routing processes.

Procedure steps

1. Add an *Export* component to the *Ospf* component:

   ```
   add Vr/<vr_name> Ip Ospf Export/<export_policy_number>
   ```

2. Set the *advertiseStatus* attribute to enable or disable the *Ospf* Export policy:

   ```
   set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
   advertiseStatus <advertise_toggle>
   ```

3. Set the *asBdrRtrStatus* attribute to enable the OSPF export policy. If you set this attribute to false, which is the default, OSPF uses the default export policy which blocks the export of all non-OSPF learned routes.

   ```
   set Vr/<vr_name> Ip Ospf asBdrRtrStatus true
   ```

4. Set the *ripInterface* attribute if you want to limit the exported OSPF information to a particular RIP learned interface's networks:

   ```
   set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
   ripInterface <ipAddress>
   ```

   Note: If you provision the *ripInterface* attribute, you must set the *protocol* attribute to all or rip. See step 5.

5. Set the *protocol* attribute if you want to specify which routing protocol networks are advertised by this policy:

   ```
   set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
   protocol <protocol_type>
   ```

6. If you want to export bgpInternal routes and you set the *protocol* attribute to all in step 5, set the *redistributeIbgp* attribute as follows:

   ```
   set Vr/<vr_name> Ip Ospf redistributeIbgp true
   ```

   Note: If you set the *protocol* attribute to bgpInternal in step 5, you do not need to set the *redistributeIbgp* attribute.

7. Set the *ripNeighbor* attribute if you want to limit the exported OSPF information to a specific RIP neighbor:

   ```
   set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
   ripNeighbor <ipAddr>
   ```

8 Set the *metric* attribute if you want to change the default cost metrics for exported routes:

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
metric <cost>
```

9 Configure a *NetworkList* <*ipaddress*> <*ipmask*> component if you want to advertise the information defined in this policy about those networks and subnetworks contained in the *NetworkList* component:

```
add Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance>
```

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance> ipaddress <ipAd>, ipmask
<ip_mask>
```

10 Set the *egpAsId* attribute if you want to limit the EGP routes to a specific AS:

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
egpAsId <as_id>
```

Variable definitions

Variable	Value
<advertise_toggle>	Set to send (enable) or blocked (disable).
<as_id>	The autonomous system number of the network routing from which the EGP route was learned.
<cost>	The OSPF cost that is attached to exported routes.
<export_policy_number>	The numeric designation assigned to this export policy.
<ipAd>	The network or subnetwork address of the network.
<ipAddr>	The 32-bit IP address of the specific router to which the information defined in this policy is advertised.
<ipAddress>	The local IP address of the RIP interface.
<ip_mask>	The subnet mask used by the network.
<net_instance>	The numeric designation assigned to this instance of the *NetworkList* component.

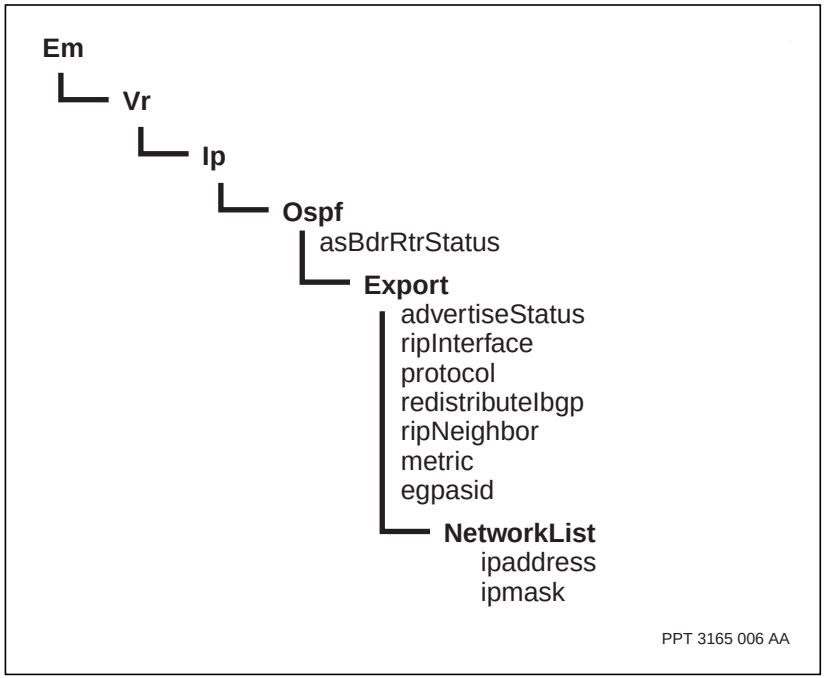
(Sheet 1 of 2)

Variable	Value
<protocol_type>	All, egp, rip, staticLocal, staticRemote, bgpInternal, or bgpExternal. If you set the *protocol* attribute to all, the export policy applies to all routes in the forwarding table except bgpInternal and bgpExternal. If you want to export bgpInternal routes, create an export policy with the *protocol* attribute set to bgpInternal, or see step 6. If you want to export bgpExternal routes, create an export policy with the *protocol* attribute set to bgpExternal.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 41
Configuring OSPF export policy component hierarchy



Configuring an always-up IP interface for OSPF

Prerequisites

- "Virtual media configuration" (page 109)
- Configure an export policy for OSPF. See "Configuring OSPF export policy" (page 159).

Procedure steps

1 Set the *protocol* attribute to *staticLocal*.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
protocol staticLocal
```

2 Add the *NetworkList* component.

```
add Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance>
```

3 Set the ip address to match against.

```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance> ipaddress <ip_addr>
```

4 Set the network mask.

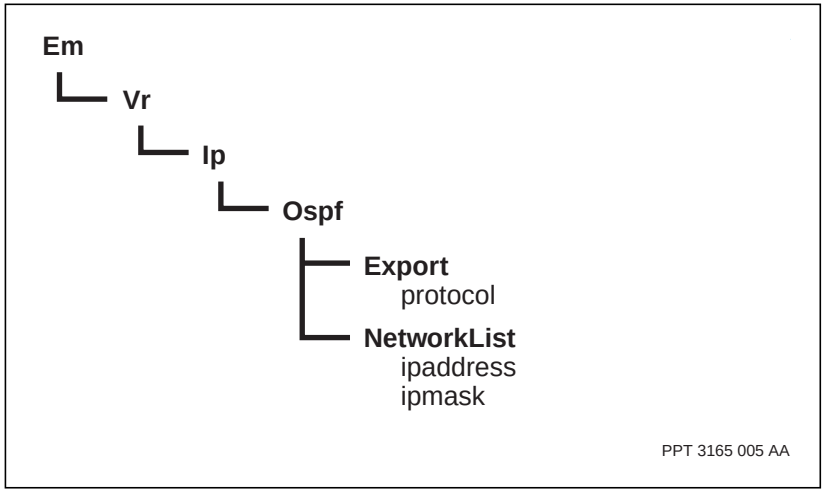
```
set Vr/<vr_name> Ip Ospf Export/<export_policy_number>
NetworkList/<net_instance> ipmask <ip_mask>
```

Variable definitions

Variable	Value
<export_policy_number>	The numeric designation you assign to this export policy.
<ip_addr>	The same IP address as that of the *LogicalIf* component of the VR's IP port.
<ip_mask>	The mask used by the network.
<net_instance>	The numeric designation of the *NetworkList* component (decimal 0..65535).
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 42
Configuring an always-up IP interface for OSPF component hierarchy



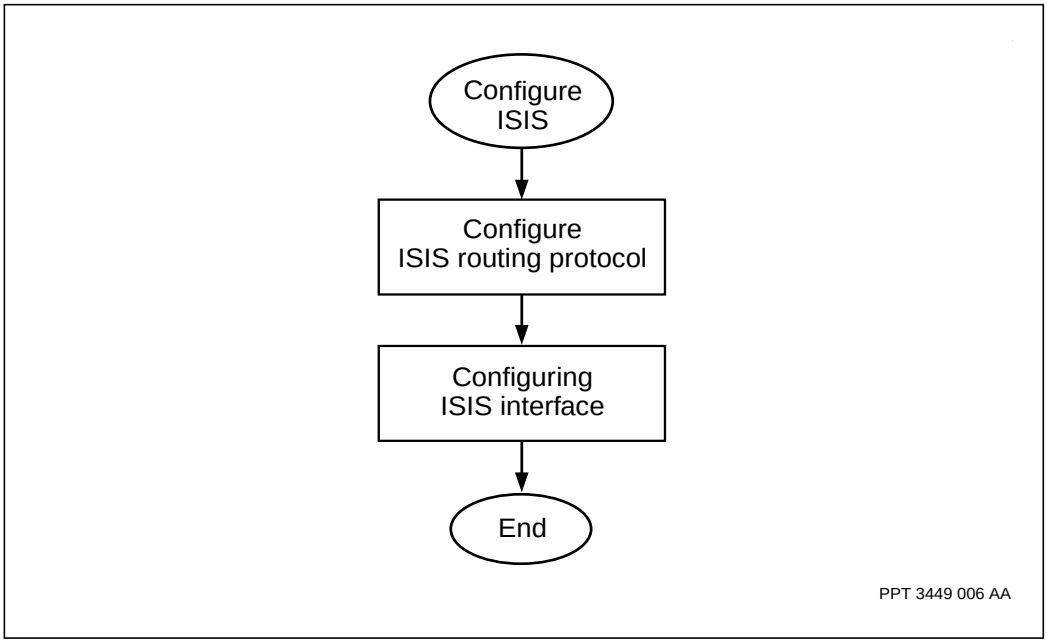
Chapter 13
Intermediate System to Intermediate System (ISIS) configuration

Configure the Intermediate System to Intermediate System (ISIS) routing protocol to exchange routing information within a network or between networks using ISIS.

ISIS configuration task flow

This task flow shows you the sequence of procedures you perform to configure ISIS. To link to any procedure, go to the list that follows the task flow.

Figure 43
Configuring ISIS task flow



Navigation links

- “Configuring ISIS routing protocol” (page 167)
- “Configuring the ISIS interface” (page 170)

Configuring ISIS routing protocol

Configure ISIS to enable the ISIS routing protocol on the router.

Prerequisites

- The ISIS routing component is a subcomponent of the Router component. A component of the type Router/<router_name> must already exist.

Procedure steps

1 Add the *ISIS* protocol under the Router.

```
add router/<router_name> isis
```

2 Set the *ISIS* component attributes.

```
set router/<router_name> Isis <attributes>
```

3 Add the *Network Entity Title* (*Net*) component.

```
add router/<router_name> isis Net/<net_instance>
```

4 Add the *ISIS Level* component.

```
add router/<router_name> isis Level/<level>
```

5 Set the *ISIS Level* attributes.

```
set router/<router_name> Isis Level/<level> preference
<preference>
```

Variable definitions

Variable	Value
<attributes>	Represents the ISIS attributes and values
<level>	Represents the ISIS protocol level specific capability under the router ISIS component. Level/1 only is supported

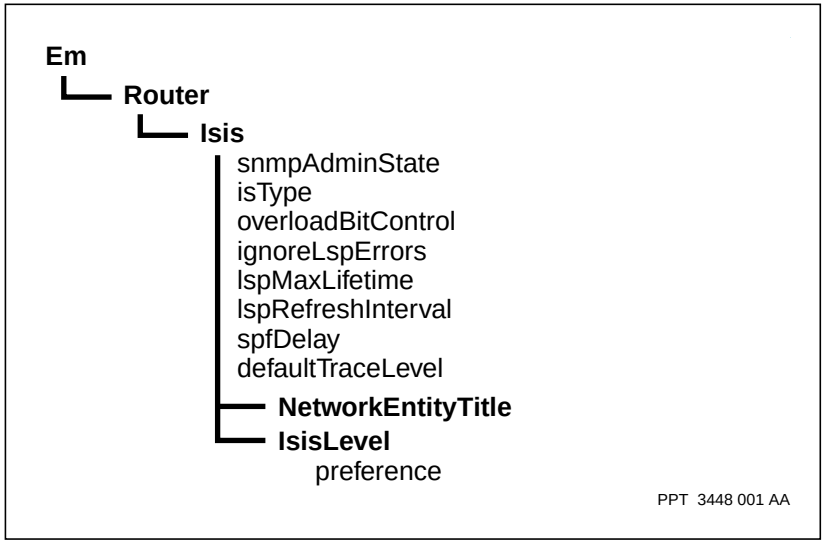
(Sheet 1 of 2)

Variable	Value
<net_instance>	The *value assigned to the NetworkEntityTitle* component that identifies the ISIS router. The NetworkEntityTitle consists of three parts: Area Address, System Identifier, and Network Selector (NSEL). Passport supports up to 3 NETs. On a router, all provisioned NETs must have the same System Identifier. The following address illustrates the NET format: 49.000001.1234.5678.90ab.00 The first part (49.0001) is the Area Address. The Area Address can be from 1 to 13 bytes in length. The first byte of the Area Address is referred to as the Authority and Format Identifier (AFI). The next 6 bytes (1234.5678.90ab) are the System ID. The System ID can be any 6 bytes which allow the ISIS node to be uniquely identified within the domain. The last byte (00) is the NetworkSelector and this is always 0.
<preference>	1 - 255
<router_name>	Any mnemonic (for example, RTR1)

(Sheet 2 of 2)

Procedure job aid

Figure 44
Configuring ISIS routing protocol



Configuring the ISIS interface

Configure the ISIS interface to enable the ISIS routing protocol on a router interface.

Prerequisites

- The ISIS interface component is a subcomponent of the Router Interface component. A component of the type Router/<router_name> must already exist.

Procedure steps

1 Add the *IsisIf* component.

```
add router/<router_name> interface/<ip_address> IsisIf
```

2 Set the *IsisIf* component attributes.

```
set router/<router_name> interface/<ip_address> IsisIf
<isisif_attributes>
```

3 Add the *IsisIf Level* component.

```
add router/<router_name> interface/<ip_address> IsisIf
Level/<if_level>
```

4 Set the *IsisIf Level* component attributes.

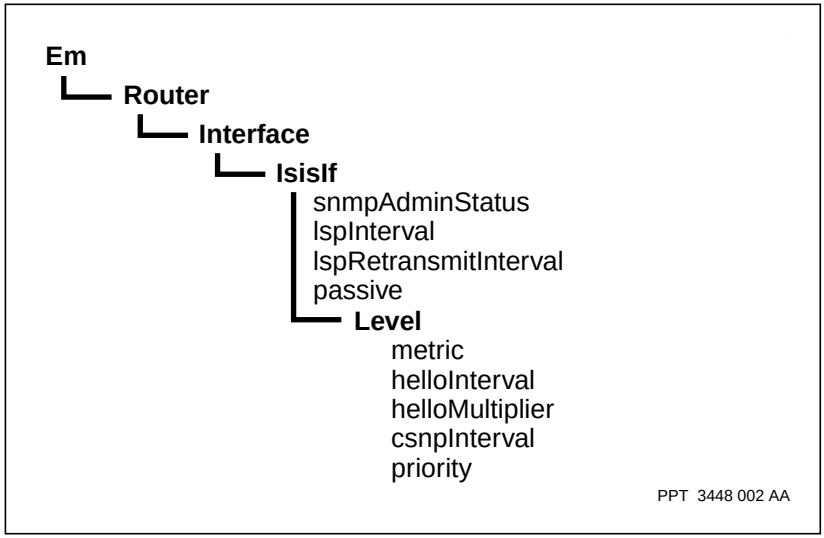
```
set router/<router_name> interface/<ip_address> IsisIf
<isisif_attributes>
```

Variable definitions

Variable	Value
<if_level>	Represents the ISIS Level functionality that is associated with a single logical port. Level/1 only is supported.
<ip_address>	32 bit address assigned to the interface from which the routing information is derived.
<router_name>	Any mnemonic (for example, RTR1)

Procedure job aid

Figure 45
Configuring ISIS Interface



Chapter 14
Migrating from RIP to OSPF

Use the procedures in this section to migrate from RIP to OSPF.

Prerequisites

- To migrate from RIP to OSPF, either use *migrateRip* or change the route preference of RIP or OSPF internal to prefer RIP routes. Changing the route preference is the preferred method because enabling and disabling *migrateRip* restarts the protocol. See the following:
  - — “Configuring a routing information protocol (RIP) interface” (page 132)
  - — “Adding OSPF to a virtual router” (page 150)
- After enabling *migrateRip*, any subsequent route preference changes are not enabled until *migrateRip* is disabled.

Procedure steps

1. Configure the migration state between RIP and OSPF:

   ```
   set Vr/<vr_name> Ip Ospf migrateRip enabled
   ```

   ```
   set Vr/<vr_name> Ip Rip migrateRip enabled
   ```

2. Once the migration is complete and OSPF is stable, change the migration state between RIP and OSPF by disabling the *migrateRip* attribute:

   ```
   set Vr/<vr_name> Ip Ospf migrateRip disabled
   ```

   ```
   set Vr/<vr_name> Ip Rip migrateRip disabled
   ```

Variable definitions

Variable	Value
<vr_name>	The name of the virtual router.

Chapter 15
BGP-4 configuration

Configure the border gateway protocol 4 (BGP-4) to exchange routing information within a network or between networks using BGP-4, or to populate OSPF or RIP information through an IP tunnel.

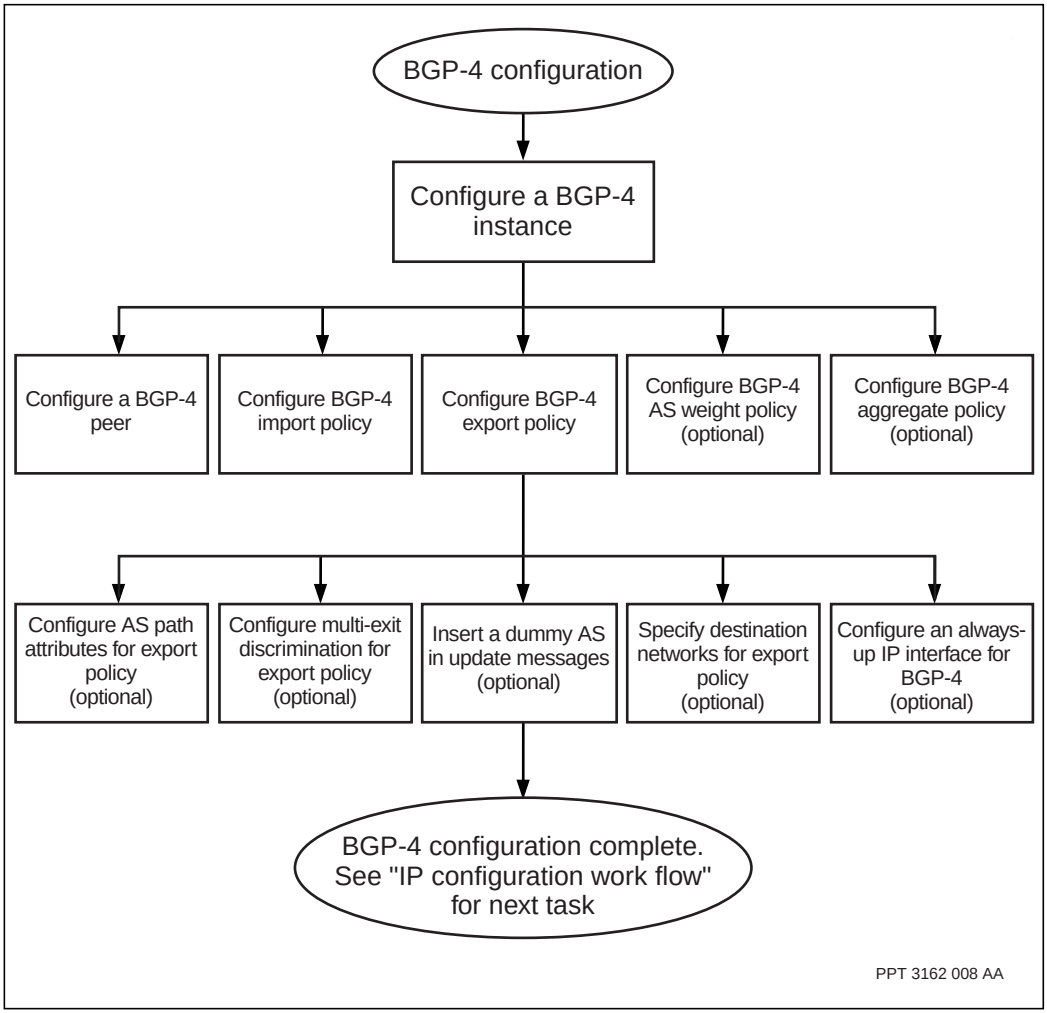
Prerequisites to BGP-4 configuration

- Install a PM2 function processor and CP2 control processor in the Passport node. Passport supports BGP-4 on these cards only.
- Configure one VR for each BGP-4 instance that you intend to configure. You can create only one BGP-4 instance for each VR on the Passport node. See “Configuring a virtual router” (page 118).

BGP-4 configuration task flow

This task flow shows you the sequence of procedures you perform to configure BGP-4. To link to any procedure, go to “Navigation links” (page 176).

Figure 46
BGP-4 configuration task flow



Navigation links

- “Configuring a BGP-4 instance” (page 178)
- “Configuring a BGP-4 peer” (page 180)
- “Configuring BGP-4 import policy” (page 184)

- “Configuring BGP-4 export policy” (page 189)
- “Configuring BGP-4 AS weight policy” (page 192)
- “Configuring BGP-4 aggregate policy” (page 193)
- “Configuring AS path attributes for export policy” (page 196)
- “Configuring multi-exit discrimination for export policy” (page 198)
- “Inserting a dummy AS in update messages” (page 200)
- “Specifying destination networks for export policy” (page 201)
- “Configuring an always-up IP interface for BGP-4” (page 203)
- For information about the next task, see “IP configuration work flow” (page 38)

Configuring a BGP-4 instance

Configure a BGP-4 instance if you need to use BGP-4 to communicate between the Passport node and your network, or to populate OSPF or RIP information through an IP tunnel.

Prerequisites

- Configure one VR for each BGP-4 instance that you intend to configure. You can create only one BGP-4 instance for each VR on the Passport node. See “Configuring a virtual router” (page 118).

Procedure steps

1. Create an instance of the BGP-4 protocol.

   ```
   add Vr/<vr_name> Ip Bgp
   ```

2. Specify the autonomous system (AS) number of the BGP-4 instance.

   ```
   set Vr/<vr_name> Ip Bgp localAs <asNo>
   ```

3. Specify the router identifier for the BGP-4 instance.

   ```
   set Vr/<vr_name> Ip Bgp bgpIdentifier <ipAddress>
   ```

4. Specify the local preference for routes received from external peers, if you do not want to use the default value.

   ```
   set Vr/<vr_name> Ip Bgp locPrf <loc_pref>
   ```

5. Specify the MED value for routes sent to external peers, if you do not want to use the default value.

   ```
   set Vr/<vr_name> Ip Bgp med <metric>
   ```

6. Configure BGP-4 as a route reflector within the AS, if you do not want to use the default value.

   The default cluster identifier, as specified in the *routeReflectorCluster* attribute, has the same IP address as the router ID.

   ```
   set Vr/<vr_name> Ip Bgp rr <true_false>
   ```

7. If required, change the default for the alarm generator, which specifies what BGP alarms are generated.

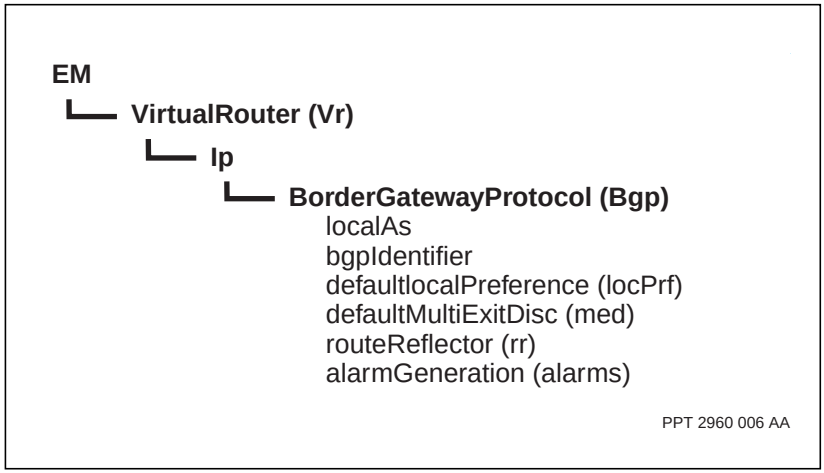
   ```
   set Vr/<vr_name> Ip Bgp alarmGeneration <alarm>
   ```

Variable definitions

Variable	Value
<alarm>	The level of alarm generation for this instance of BGP.
<asNo>	The autonomous system (AS) to which the BGP-4 instance belongs.
<ipAddress>	The 32-bit IP address that identifies the BGP-4 router.
<loc_pref>	The local preference.
<metric>	The MED value.
<true_false>	Specifies whether the BGP-4 instance behaves as a route reflector within the AS.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 47
Configuring a BGP-4 instance component hierarchy



Configuring a BGP-4 peer

Configure BGP-4 peers to exchange routing information about reachable destinations in different autonomous systems (ASs). The BGP-4 peers use this routing information to construct a map of AS connectivity that allows them to eliminate routing loops and enforce policy decisions at the AS level.

Prerequisites

- Since Passport does not support EBGP multi-hop, you must not use virtual media (loopback) addresses for EBGP peering. If you are configuring an EBGP peering session, you must use the addresses of the directly connected interfaces to ensure proper exchange of routing information.

Procedure steps

1 Create an instance of a BGP-4 peer under the BGP-4 instance.

The component *Vr Ip Bgp Peer Desc* is automatically created.

```
add Vr/<vr_name> Ip Bgp Peer/<peer>
```

2 Specify the AS number for the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc peerAs <asNo>
```

3 Specify the local IP address for the BGP-4 peer.

For EBGP peering, you must use the addresses of the directly connected interfaces to ensure proper exchange of routing information. For IBGP peering, you can use an always-up IP interface. If you are using an always-up IP interface, the local IP address of the BGP-4 peer must match the IP address of the virtual router's associated virtual media logical interface.

If you specify the local IP address as 0.0.0.0, TCP chooses a local IP interface based on the remote IP address of this peer connection and the IP forwarding table.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc lac <lac_addr>
```

Note: If you are using an always-up interface, make sure attribute *Vm If mode* is set to alwaysUpInterface. See "Virtual media configuration" (page 109).

4 Specify the frequency of keep alive message retransmissions from the BGP-4 peer.

When you set this attribute to 0, BGP-4 does not send keep alive messages.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc kac
<kac_timer>
```

5 Specify the maximum length of time between BGP-4 keep alive messages from the BGP-4 peer before the BGP-4 instance considers the connection down.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc htc
<htc_timer>
```

6 Specify the minimum length of time TCP waits before re-attempting to establish a BGP-4 connection.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc connectRetry
<con_tmr>
```

7 Specify the minimum length of time the BGP-4 peer waits before sending route updates to a neighbor AS.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc minAsOrig
<orig_tmr>
```

8 Specify the length of time after which BGP-4 can re-advertise route information to peers in other ASs.

BGP-4 ignores the *minAsOrigTime* and *minRouteAdvTime* attributes for routes that are withdrawn.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc minRouteAdv
<adv_tmr>
```

9 Specify whether the BGP-4 peer is a route reflector client.

You must configure the BGP-4 instance as a route reflector if you set the *isRouteReflectorClient* attribute to true.

```
set Vr/<vr_name> Ip Bgp peer/<peer> Desc isRrClient
<true_false>
```

10 Enable dynamic default aggregation (DDA) for routes learned from the EBGP peer if you want to prioritize dynamic default routes.

Specifying a MED metric other than the default value enables DDA for the BGP-4 peer. BGP-4 sets the MED path attribute of the default route to the value you configure in the *defaultInAggMed* (*diaMed*) attribute.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc diaMed
<metric>
```

11 Specify whether the private AS number is removed from the AS path attribute of routes sent to the EBGP peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc remPrivate
<rem_priv>
```

12 Advertise the route availability to the BGP-4 peer.

```
set Vr/<vr_name> Ip Bgp Peer/<peer> Desc nhs <nhs>
```

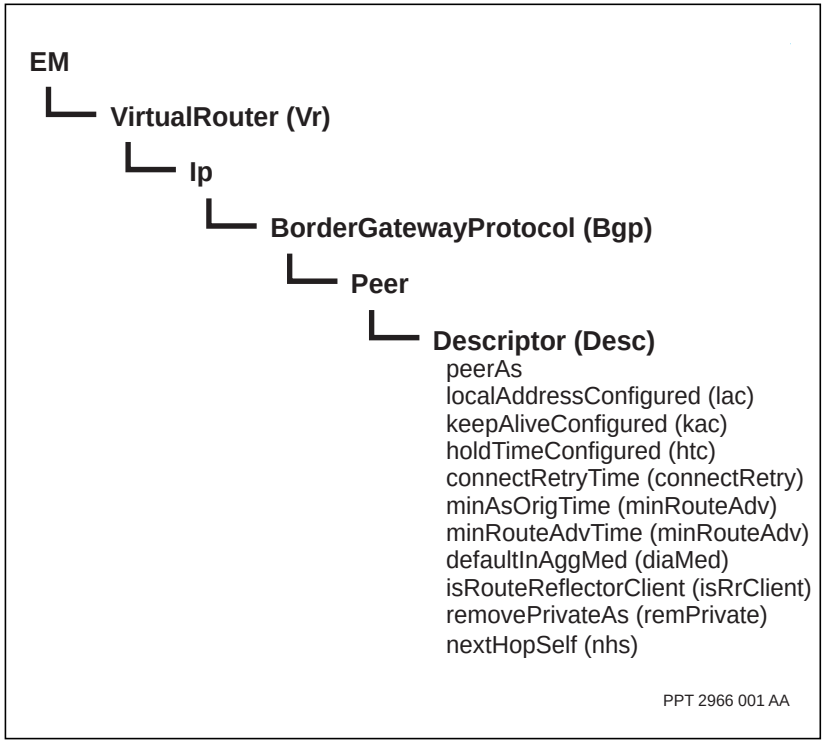
13 Activate your configuration changes. For more information, see “Activating configuration changes” (page 31).

Variable definitions

Variable	Value
<adv_tmr>	The time interval in seconds.
<asNo>	The AS to which the BGP-4 peer belongs.
<con_tmr>	The time interval in seconds.
<htc_timer>	The time interval in seconds.
<kac_timer>	The time interval in seconds.
<lac_addr>	The local IP address for this BGP-4 peer connection.
<metric>	The MED path attribute value assigned to the DDA route.
<nhs>	Indicates if the next-hop-self is enabled or disabled when a route is sent to the peer. For details on the syntax for this attribute, see 241-5701-060 *Passport 7400, 15000, 20000 Components*.
<orig_tmr>	The time interval in seconds.
<peer>	The IP address of the BGP-4 peer. To use an always-up IP interface, the IP address of the BGP-4 peer must be the same as the associated virtual media logical interface. See “Configuring an always-up IP interface for BGP-4” (page 203).
<rem_priv>	Indicates whether the private AS number removal is enabled or disabled when the route is advertised to another AS.
<true_false>	Specifies whether the BGP-4 peer is a route reflector client.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 48
Configuring a BGP-4 peer component hierarchy



Configuring BGP-4 import policy

Configure BGP-4 import policy to specify what routing information BGP-4 allows into or blocks from the IP routing database.

Procedure steps

1. Create an instance of an import policy. The instance number only identifies the policy and is not related to policy preference.

   ```
   add Vr/<vr_name> Ip Bgp Import/<im_plcy>
   ```

2. Specify the AS number of the BGP-4 peer from which routes are learned. If you set this value to 0, the policy matches any AS number.

   ```
   set Vr/<vr_name> Ip Bgp Import/<im_plcy> peerAs <asNo>
   ```

3. Specify the IP address of the BGP-4 peer from which routes are learned. If you set this value to 0.0.0.0, the policy matches any IP address.

   ```
   set Vr/<vr_name> Ip Bgp Import/<im_plcy> peerIpAddress
   <addr>
   ```

4. Specify the AS that originated the routes learned over the BGP-4 peer. If you set this value to 0, the policy matches any AS number.

   ```
   set Vr/<vr_name> Ip Bgp Import/<im_plcy> originAs
   <or_asNo>
   ```

5. Specify a regular expression that identifies AS paths from which BGP-4 accepts route updates if you do not want to use the default value. For details on the syntax for this attribute, 241-5701-060 *Passport 7400, 15000, 20000 Components*.

   ```
   set Vr/<vr_name> Ip Bgp Import/<im_plcy> asExpr
   <path_expr>
   ```

6. Specify a regular expression that identifies community paths from which BGP-4 accepts route updates if you do not want to use a default value. For details on the syntax for this attribute, 241-5701-060 *Passport 7400, 15000, 20000 Components*.

   ```
   set Vr/<vr_name> Ip Bgp Import/<im_plcy> comExpr
   <com_expr>
   ```

7. If you have configured an AS path and community path expression for the import policy, specify a preference for the policy.

When the expression attributes of two import policies match the same AS or community, BGP-4 uses the preference metric to select a preferred policy. A higher value indicates a higher preference.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> exprPref
<pref>
```

8 Specify the protocol that originated the routes learned over the BGP-4 peer:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy>
originProtocol <protocol>
```

9 Specify whether BGP-4 uses or ignores information in routing updates if it meets the criteria specified in the policy:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> usage <flag>
```

10 If required, change the route preference for BGP internal routes. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information.

Either change the route preference attribute *defaultIbgpRtePref*, or override the route preference by changing attribute *ibgpRtePref.*

To change the route preference attribute, use the following command:

```
set Vr/<vr_name> Ip Bgp defaultIbgpRtePref
<ibgp_route_pref>
```

To override the route preference, use the following command:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy>
ibgpRtePreference <ibgp_override>
```

11 If required, change the route preference for BGP external routes. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information.

Either change the route preference attribute *defaultEbgpRtePref*, or override the route preference by changing attribute *ebgpRtePref.*

To change the route preference attribute, use the following command:

```
set Vr/<vr_name> Ip Bgp defaultEbgpRtePref
<ebgp_route_pref>
```

To override the route preference, use the following command:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy>
ebgpRtePreference <override>
```

12 Specify a preference for routes that match the import policy:

If you do not set this value, BGP-4 applies the local preference configured under the BGP-4 instance in the *defaultLocalPreference* attribute to routes that meet the import policy criteria.

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> locPrf
<loc_pref>
```

13 Specify the community number that BGP-4 inserts in the community path attribute for routes that match the criteria of this import policy:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> appCom
<com_no>
```

14 Specify a destination network for this BGP-4 import policy if you want to restrict the destination networks:

You must create an instance of the *Network* (*Net*) subcomponent for each destination network associated with the import policy. If you do not specify any destination networks, BGP-4 applies the import policy to all networks.

```
add Vr/<vr_name> Ip Bgp Import/<imp_plcy> Net/<net_no>
```

15 Specify the network prefix for the destination network associated with this BGP-4 import policy:

```
set Vr/<vr_name> Ip Bgp Import/<im_plcy> Net/<net_no>
prefix <prefix>
```

16 Specify the network prefix length for the destination network associated with this BGP-4 import policy:

```
set Vr/<vr_name> Ip Bgp Import/<imp_plcy> Net/<net_no>
length <length>
```

Variable definitions

Variable	Value
<addr>	The IP address of the BGP-4 peer.
<asNo>	The AS to which the BGP-4 peer belongs.
<com_expr>	A regular expression identifying community paths to match.
<com_no>	The community number added to the community path attribute.

(Sheet 1 of 3)

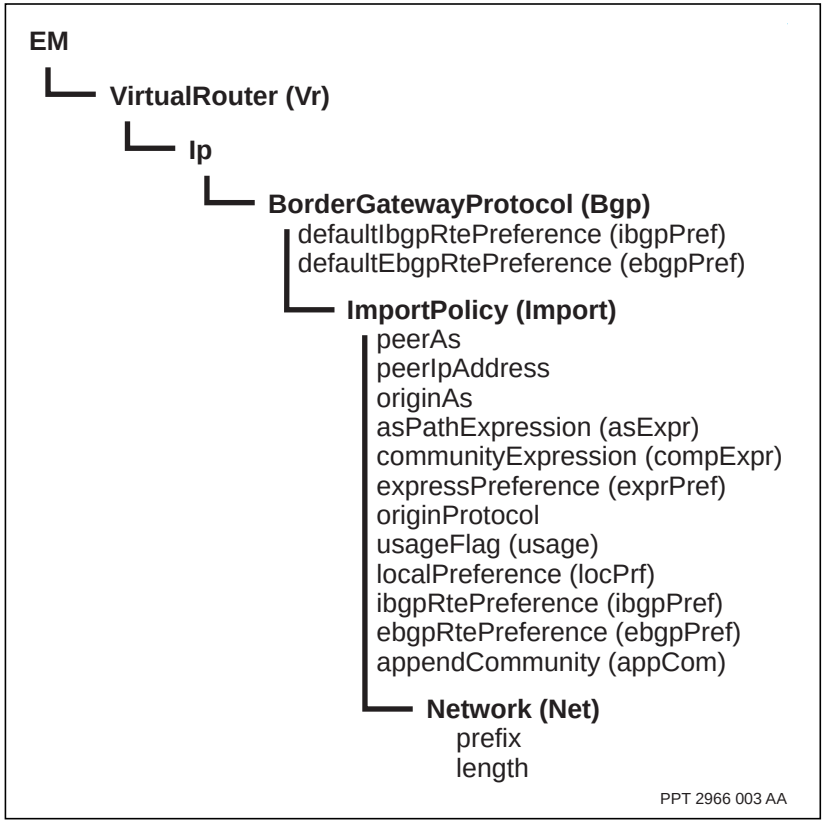
Variable	Value
<ebgp_override>	The override route preference. Attribute range is 1 to 253. Attribute default is sameAsBgp, which means use the value of attribute *defaultEbgpRtePref* for the route preference. To prefer BGP external routes over OSPF internal routes, the recommended setting for *ebgpRtePreference* is 6.
<ebgp_route_pref>	The route preference. Attribute range is 1 to 253. Attribute default is 70.
<flag>	Indicates whether BGP-4 uses or ignores received routing updates.
<ibgp_override>	The override route preference. Attribute range is 1 to 253. Attribute default is sameAsBgp, which means use the value of attribute *defaultIbgpRtePref* for the route preference. To prefer BGP internal routes over OSPF internal routes, the recommended setting for *ibgpRtePreference* is 6.
<ibgp_route_pref>	The route preference. Attribute range is 1 to 253. Attribute default is 122.
<im_plcy>	The instance number of the import policy.
<length>	Specifies the length of the network prefix.
<loc_pref>	The relative preference for routes that match the import policy's criteria
<net_no>	Identifies the destination network associated with the import policy.
<or_asNo>	The number of the AS that originated the learned routes.
<path_expr>	A regular expression identifying AS paths to match.
<pref>	The relative preference of a path-based policy.
<prefix>	The network prefix, in the form of an IP address.

(Sheet 2 of 3)

Variable	Value
<protocol>	Identifies the protocol that originated the routes.
<vr_name>	The name of the virtual router.
(Sheet 3 of 3)	

Procedure job aid

Figure 49
Configuring a BGP-4 import policy component hierarchy



Configuring BGP-4 export policy

Configure BGP-4 export policy to specify what IP routing information BGP-4 distributes to other BGP-4 peers.

Procedure steps

1 Create an instance of an export policy. The instance number only identifies the policy and is not related to policy preference.

```
add Vr/<vr_name> Ip Bgp Export/<ex_plcy>
```

2 Specify the AS number of the BGP-4 peer to which routes are advertised. If you set this value to 0, BGP-4 advertises to all peer ASs.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> peerAs <asNo>
```

3 Specify the IP address of the peer AS to which routes are advertised. If you set this value to 0.0.0.0, the policy matches any IP address.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> peerIpAddress
<addr>
```

4 Specify the protocol to which the export policy applies:

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> protocol
<prtcl>
```

a. If you specified EGP in the *protocol* attribute, specify the number of the EGP AS to which the export policy applies:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or EGP.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> egpAs
<egp_id>
```

b. If you specified BGP in the *protocol* attribute, specify the number of the BGP-4 AS to which the export policy applies:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all, bgpInternal or bgpExternal.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> bgpAs
<bgp_id>
```

c. If you specified OSPF in the *protocol* attribute, specify the OSPF tag for the OSPF routes to which the export policy applies:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or ospfExternal.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ospfTag
<ospf_tag>
```

d. If you specified RIP in the *protocol* attribute, specify the local RIP interface from which RIP routes were learned. If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP.

The IP address of the RIP interface corresponds to the IP address of a *IplogicalIf* component configured under the IP port. For more information, see “Enabling IP on a protocol port” (page 124).

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ripIf
<rip_if>
```

e. If you specified RIP in the *protocol* attribute, specify the RIP neighbor from which RIP routes were learned:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP.

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> ripNbr
<rip_nbr>
```

f. Specify a preference for routes that match the export policy:

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy) locPref
<loc_pref>
```

5 Specify whether BGP-4 advertises or blocks routes that meet the export policy’s criteria:

```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> advertise
<send_block>
```

Variable definitions

Variable	Value
<addr>	The IP address of the BGP-4 peer.
<asNo>	Identifies the AS to which the BGP-4 peer belongs.
<bgp_id>	The number of the BGP AS.
<egp_id>	The number of the EGP AS.
<ex_plcy>	The instance number of the export policy.

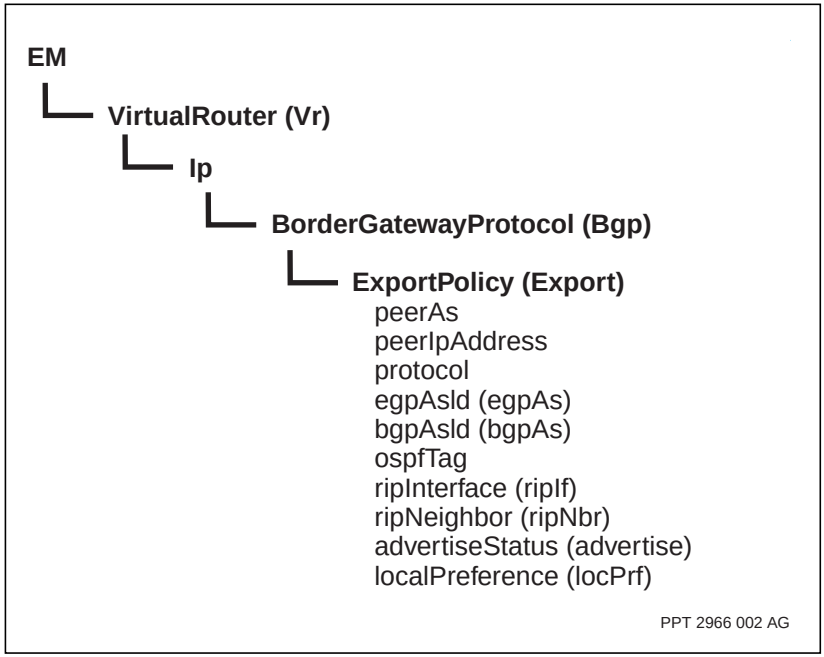
(Sheet 1 of 2)

Variable	Value
<loc_pref>	The preference for routes that match the criteria of the export policy.
<ospf_tag>	The OSPF tag stored in OSPF external routes.
<prtcl>	Identifies the protocol type for matching routes.
<rip_if>	The IP address of the RIP interface from which RIP routes were learned.
<rip_nbr>	The IP address of the RIP neighbor from which RIP routes were learned.
<send_block>	Indicates BGP-4 behavior for routes that match the export policy.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 50
Configuring BGP-4 export policy component hierarchy



Configuring BGP-4 AS weight policy

Configure BGP-4 AS weight policy to set a preference for one autonomous system and discriminate against other autonomous systems.

Procedure steps

1. Create an instance of an AS weight policy. The instance number only identifies the policy and is not related to policy preference.

   ```
   add Vr/<vr_name> Ip Bgp As/<as_no>
   ```

2. Specify the weight for the AS of the BGP-4 instance. The lowest weight is preferred in route selection.

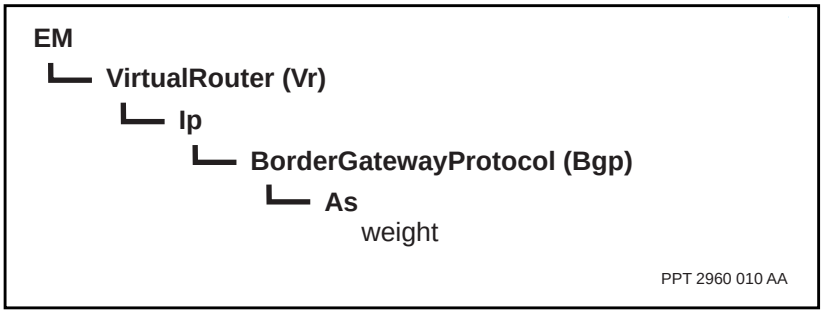
   ```
   set Vr/<vr_name> Ip Bgp As/<as_no> weight <as_wgt>
   ```

Variable definitions

Variable	Value
<as_no>	The instance number of the AS.
<as_wgt>	The weight assigned to the AS.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 51
Configuring BGP-4 AS weight policy component hierarchy



Configuring BGP-4 aggregate policy

Configure BGP-4 aggregate policy to enable BGP-4 to combine the characteristics of different routes and advertise the combination as a single route. Aggregation reduces the data a BGP-4 speaker stores and exchanges with other BGP-4 speakers.

Procedure steps

1. Create an instance of an BGP-4 aggregate policy. The instance number represents the network prefix and prefix length, separated by a comma.

   ```
   add Vr/<vr_name> Ip Bgp Aggregate/<aggr>
   ```

2. Define a set of routes if you want to specify the routes that BGP-4 aggregates or advertises with the aggregated route policy:

   ```
   add Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no>
   ```

3. Specify the network prefix for the destination network associated with this BGP-4 aggregate policy:

   Note: If you do not specify the network prefix for the destination network, route aggregation will not take place.

   ```
   set Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no>
   prefix <prefix>
   ```

4. Specify the network prefix length for the destination network associated with this BGP-4 aggregate policy:

   Note: If you do not specify the network prefix length for the destination network, route aggregation will not take place.

   ```
   set Vr/<vr_name> Ip Bgp Aggregate/<aggr> Net/<net_no>
   length <length>
   ```

5. Specify the protocol to which the aggregate policy applies:

   ```
   set Vr/<vr_name> Ip Bgp Aggregate/<aggr> protocol
   <prtcl>
   ```

6. If you specified EGP in the *protocol* attribute, specify the number of the EGP AS to which the aggregate policy applies:

   If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or EGP.

   ```
   set Vr/<vr_name> Ip Bgp Aggregate/<aggr> egpAs
   <egp_id>
   ```

7 If you specified BGP in the *protocol* attribute, specify the number of the BGP AS to which the aggregate policy applies:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all, bgpInternal or bgpExternal.

```
set Vr/<vr_name> Ip Bgp Aggregate/<aggr> bgpAs
<bgp_id>
```

8 If you specified OSPF in the *protocol* attribute, specify the OSPF tag for the OSPF routes to which the aggregate policy applies:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or ospfExternal.

```
set Vr/<vr_name> Ip Bgp Aggregate/<aggr> ospfTag
<ospf_tag>
```

9 If you specified RIP in the *protocol* attribute, specify the local RIP interface from which RIP routes were learned:

If you set this attribute to a non-zero value, the *protocol* attribute must be set to all or RIP.

The IP address of the RIP interface corresponds to the IP address of a *IpLogicalIf* component configured under the IP port. For more information, see “Enabling IP on a protocol port” (page 124).

```
set Vr/<vr_name> Ip Bgp Aggregate/<aggr> ripIf
<rip_if>
```

10 Specify whether BGP-4 aggregates or advertises routes that meet the aggregate policy’s criteria:

```
set Vr/<vr_name> Ip Bgp Aggregate/<aggr> action
<agg_adv>
```

Variable definitions

Variable	Value
<agg_adv>	Indicates BGP-4 behavior, aggregate or advertise, for routes that match the aggregate policy.
<aggr>	The aggregate policy, in the format <prefix, length>.
<bgp_id>	The number of the BGP AS.
<egp_id>	The number of the EGP AS.

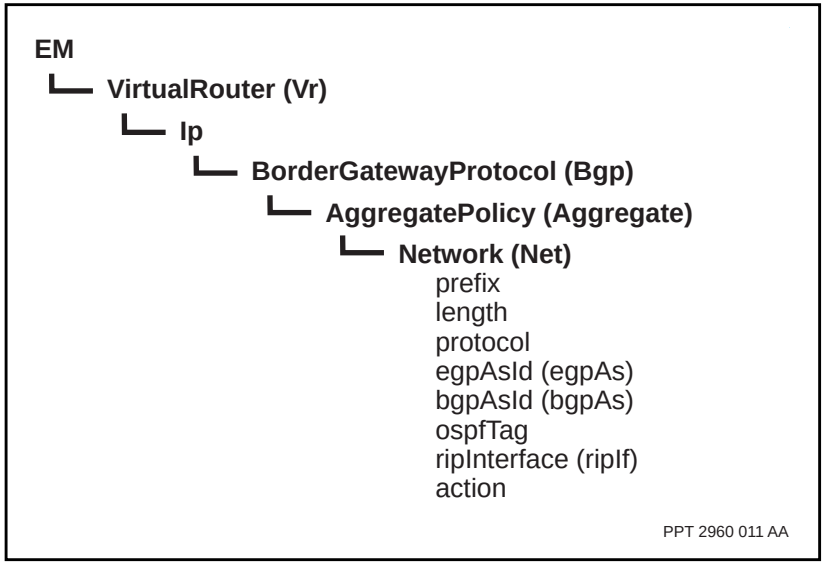
(Sheet 1 of 2)

Variable	Value
<length>	The length of the network prefix.
<net_no>	The routes associated with the aggregate policy.
<ospf_tag>	The OSPF tag stored in OSPF external routes.
<prefix>	The network prefix, in the form of an IP address.
<prtcl>	The protocol type for matching routes.
<rip_if>	The IP address of the RIP interface from which RIP routes were learned.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 52
Configuring BGP-4 aggregate policy component hierarchy



Configuring AS path attributes for export policy

Configure AS path attributes for export policy to provide detailed information about an advertised route. Routers can use path attribute information when making policy decisions.

Procedure steps

1. Specify a regular expression that identifies AS paths to which BGP-4 advertises route updates if you do not want to use the default value. For details on the syntax for this attribute, 241-5701-060 *Passport 7400, 15000, 20000 Components*.

   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> asExpr <path_expr>

2. Specify a regular expression that identifies community paths to which BGP-4 advertises route updates if you do not want to use the default value. For details on the syntax for this attribute, 241-5701-060 *Passport 7400, 15000, 20000 Components*.

   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> comExpr <com_expr>

3. If you have configured an AS path and community path expression for the export policy, specify a preference for the policy:

   When the expression attributes of two export policies match the same AS or community, BGP-4 uses the preference metric to select a preferred policy. A higher value indicates a higher preference.

   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> exprPref <pref>

4. Specify the community number that BGP-4 inserts in the community path attribute before advertising routes identified by this policy:

   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> sendCom <com_no>

Variable definitions

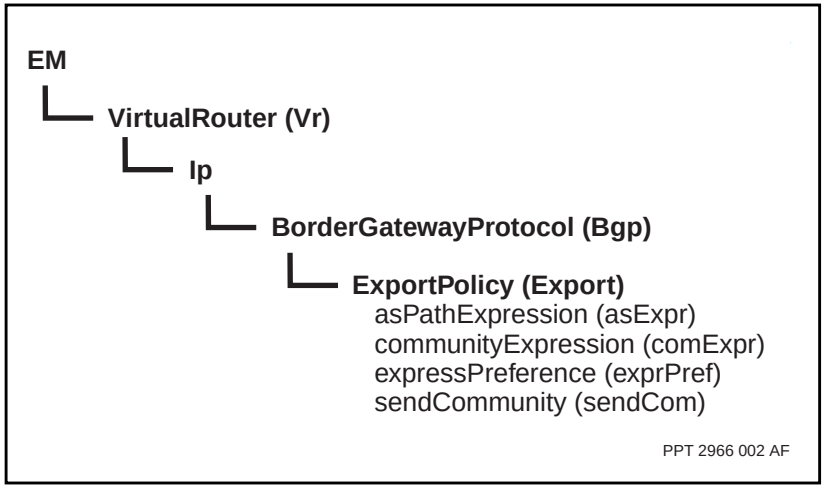
Variable	Value
<com_expr>	A regular expression identifying community paths to match.
<com_no>	The community number added to the community path attribute.

(Sheet 1 of 2)

Variable	Value
<path_expr>	A regular expression identifying AS paths to match.
<pref>	The relative preference of a path-based policy.
<vr_name>	The name of the virtual router.
(Sheet 2 of 2)	

Procedure job aid

Figure 53
Configuring AS path attributes for export policy component hierarchy



Configuring multi-exit discrimination for export policy

Configure multi-exit discrimination for export policy to include the preferred entry point to the autonomous system in updates to external peers.

Procedure steps

1. Specify the metric that BGP-4 uses for this export policy to discriminate between multiple exit points to an adjacent AS:

   `set Vr/<vr_name> Ip Bgp Export/<ex_plcy> med <med_value>`

2. Specify whether BGP-4 includes the MED value in updates to EBGP peers:

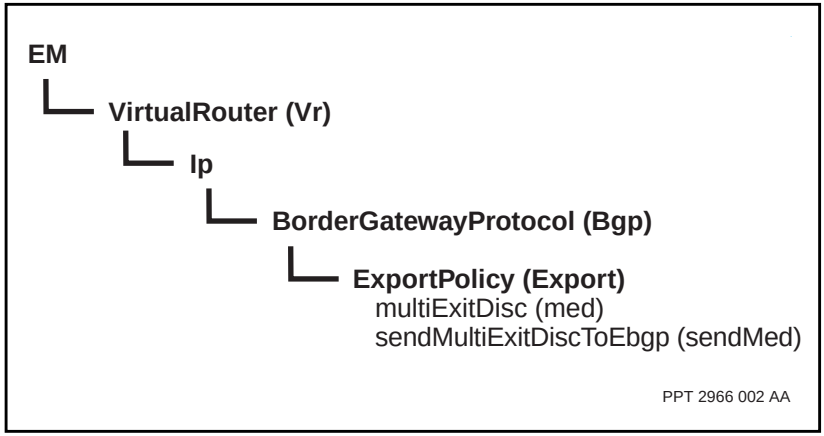
   `set Vr/<vr_name> Ip Bgp Export/<ex_plcy> sendMed <true_false>`

Variable definitions

Variable	Value
<med_value>	The multi-exit discrimination (MED) metric.
<true_false>	Indicates whether BGP-4 includes the MED metric in updates.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 54
Configuring multi-exit discrimination for export policy component hierarchy



Inserting a dummy AS in update messages

Insert a dummy AS in update messages to alter the AS path of the outgoing route.

Procedure steps

Specify a sequence of AS numbers to be inserted before the local AS number in the AS path attribute if you do not want to use the default value:

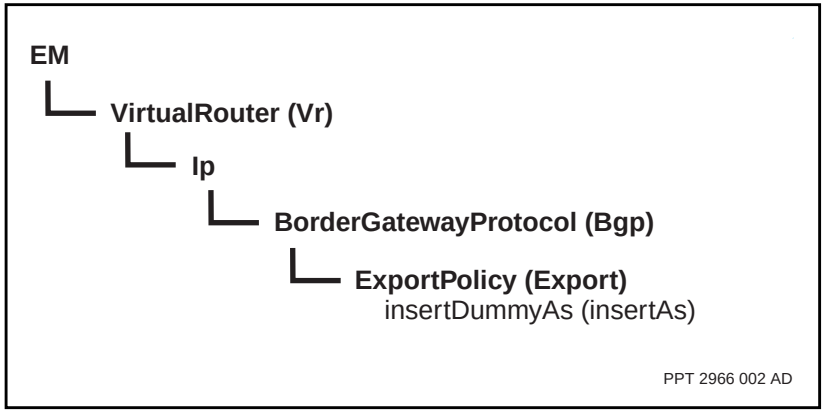
```
set Vr/<vr_name> Ip Bgp Export/<ex_plcy> insertAs
<AS_seq>
```

Variable definitions

Variable	Value
<AS_seq>	A sequence of AS numbers, separated by a period.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 55
Inserting a dummy AS in update messages component hierarchy



Specifying destination networks for export policy

Specify destination networks for export policy to define which networks to export BGP-4 routing information to.

Procedure steps

1. Specify a destination network for this BGP-4 export policy if you want to restrict the destination networks:

   You must create an instance of the *Network* (*Net*) subcomponent for each destination network associated with the export policy. If you do not specify any destination networks, BGP-4 applies the export policy to all networks.

   ```
   add Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
   ```

2. Specify the network prefix for the destination network associated with this BGP-4 export policy:

   ```
   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
   prefix <prefix>
   ```

3. Specify the network prefix length for the destination network associated with this BGP-4 export policy:

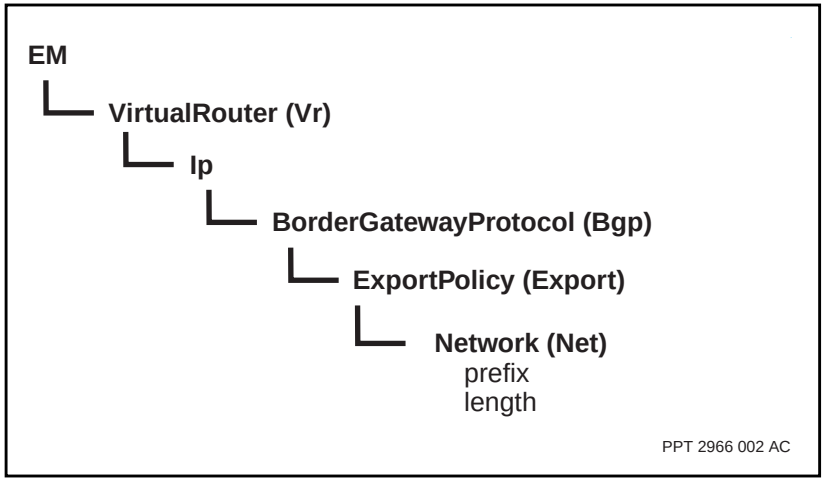
   ```
   set Vr/<vr_name> Ip Bgp Export/<ex_plcy> Net/<net_no>
   length <length>
   ```

Variable definitions

Variable	Value
<length>	The length of the network prefix.
<net_no>	The destination network associated with the export policy.
<prefix>	The network prefix, in the form of an IP address.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 56
Specifying destination networks for export policy component hierarchy



Configuring an always-up IP interface for BGP-4

Configure an always-up IP interface for BGP-4 if, in a VPN network, a virtual connection gateway (VCG) has multiple interfaces to the backbone and internal BGP (IBGP) is required on the VCG.

Prerequisites

- The always-up IP interface is only supported between IBGP peers.
- Configure a *VirtualMedia* (*Vm*) component and set attribute *Vm If mode* to alwaysUpInterface. See “Virtual media configuration” (page 109).
- Do not link the always-up interface, which is the virtual media interface, to the virtual media protocol port used to identify the tunnel end point public source addresses (i.e., it should be unique to BGP). For more information on configuring VCGs and tunnel end points, see 241-5701-582 *Passport 7400, 15000, 20000 VPN Configuration Management*.
- Under the *BorderGatewayProtocol Peer Desc* component, set the *localAddressConfigured* attribute to the IP address of the virtual media.

Procedure steps

1 Add the *Network* component.

```
add Vr/<vr_name> Ip Bgp Export/<export_policy_number>
Network/<net_instance>
```

2 Set the ip address to match against.

```
set Vr/<vr_name> Ip Bgp Export/<export_policy_number>
Network/<net_instance> ipaddress <ip_addr>
```

3 Set the network mask.

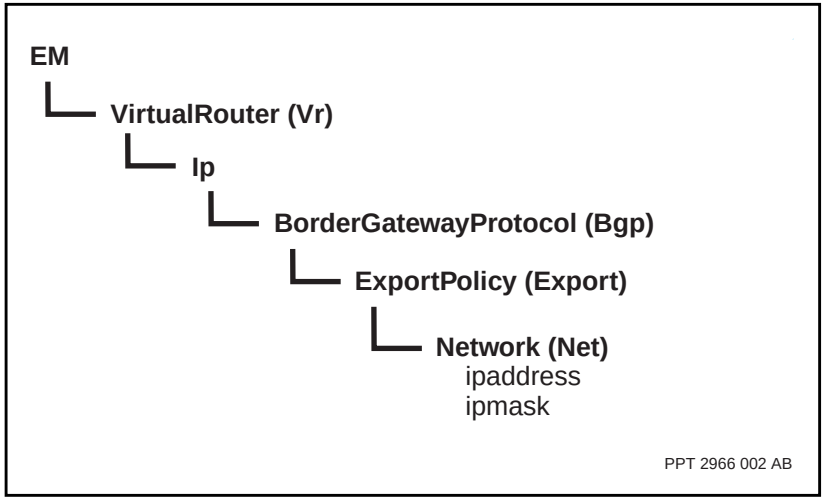
```
set Vr/<vr_name> Ip Bgp Export/<export_policy_number>
Network/<net_instance> ipmask <ip_mask>
```

Variable definitions

Variable	Value
<ip_addr>	The same IP address as that of the *LogicalIf* component of the VR's IP port.
<ip_mask>	The mask used by the network.
<net_instance>	The numeric designation of the *Network* component (decimal 0..65535).
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 57
Configuring an always-up IP interface for BGP-4 component hierarchy



Chapter 16
Configuring static ARP

Configure static ARP for IP traffic to control ARP-related attributes. ARP maps 32-bit IP addresses to physical hardware addresses.

To eliminate the need to ARP a particular host, configure the *HostEntry* subcomponent of the *Ip* component to define static ARP entries. Static host entries take precedence over dynamic entries learned through the ARP process.

You also configure static ARP host entries to ensure IP connectivity across the ATM network when you are using VC-based multiplexing for IP traffic.

Prerequisites

- Configure a media interface for IP traffic. See
  - — "Configuring an ATM MPE interface for IP traffic" (page 43). Specify the VC-based multiplexing as the encapsulation type to be used on the ATM MPE interface.
  - — "Configuring a frame relay DTE interface for IP traffic" (page 56)
- Associate the media interface with a connection. See
  - — "Configuring an ATM PVC for an ATM MPE interface" (page 45). Associate the ATM MPE interface with an ATM VCC.
  - — "Configuring a physical (hairpin) connection for a frame relay DTE interface" (page 57)
  - — "Configuring a logical connection for a frame relay DTE interface" (page 60)

— "Configuring a direct connection for a frame relay DTE interface" (page 63)

- Create an IP port under the protocol port associated with the media interface. See "Enabling IP on a protocol port" (page 124).
- Do not configure static and dynamic ARP entries for the same IP address on the same VCC. You must configure both ends of the connection to be the same. An ATM interface configured with a static ARP entry for an IP address will not respond to an inverse ARP request from a dynamic ARP.

Procedure steps

1. Create a static ARP entry for IP routing to ensure IP connectivity across the network.

   ```
   add Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>
   ```

2. Provision the physical address of the host entry. If the media type is IP.

   ```
   set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>
   physAddress <MAC_address>
   ```

   If the media type is ATM or frame relay:

   ```
   set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>
   permanentVirtualCircuitNumber <pvc_number>
   ```

3. If required, set the *maximumTransmissionUnit* (MTU) attribute.

   ```
   set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>
   maxTxUnit <mtu_size>
   ```

4. If required, set the *encapsulationType* attribute to a non-default value. The encapsulation default auto causes the correct encapsulation type to be chosen based on the media application (for example, Ethernet or IEEE802.3).

   ```
   set Vr/<vr_name> Ip Arp HostEntry/<hostAddress>,<cos>
   encap <encapsulation_type>
   ```

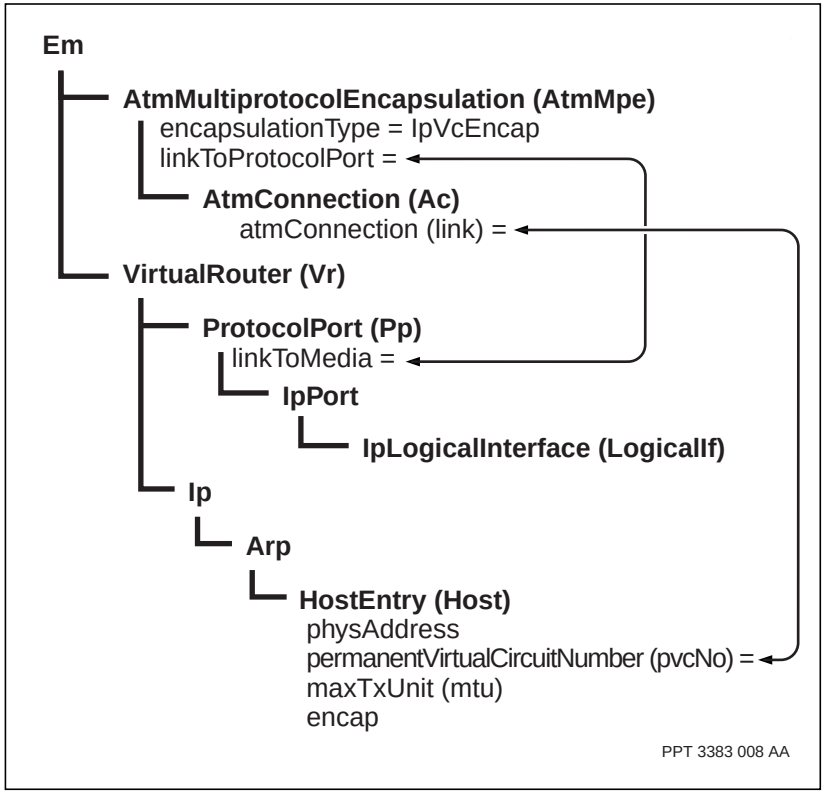
   Note: This attribute can only be set for Ethernet media. All other media must use auto.

Variable definitions

Variable	Value
<cos>	The CoS index assigned to the packet. If you are using a VC-based media (FrDte, AtmMpe), the CoS index must be 0, 1, 2, or 3. For all other media types, use na.
<encapsulation_type>	One of auto, ieee8023, or Ethernet
<hostAddress>	The IP address of the static host being defined
<MAC_address>	The 48-bit MAC address of the host being defined. It is formatted as zero to eight pairs of hex digits separated by dashes. The default address is 00-00-00-00-00-00-00-00.
<mtu_size>	The size in bytes of the maximum transmission unit, or largest datagram, that the host can accept. The MTU must fall within the valid range for the media on which the host is located.
<pvc_number>	The PVC for the static host entry. If the media type is frame relay, then <pvc_number> is the frame relay data link connection identifier. If the media type is ATM, then <pvc_number> is the instance number of the *AtmConnection* component on the ATM MPE interface.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 58
Configuring static ARP component hierarchy



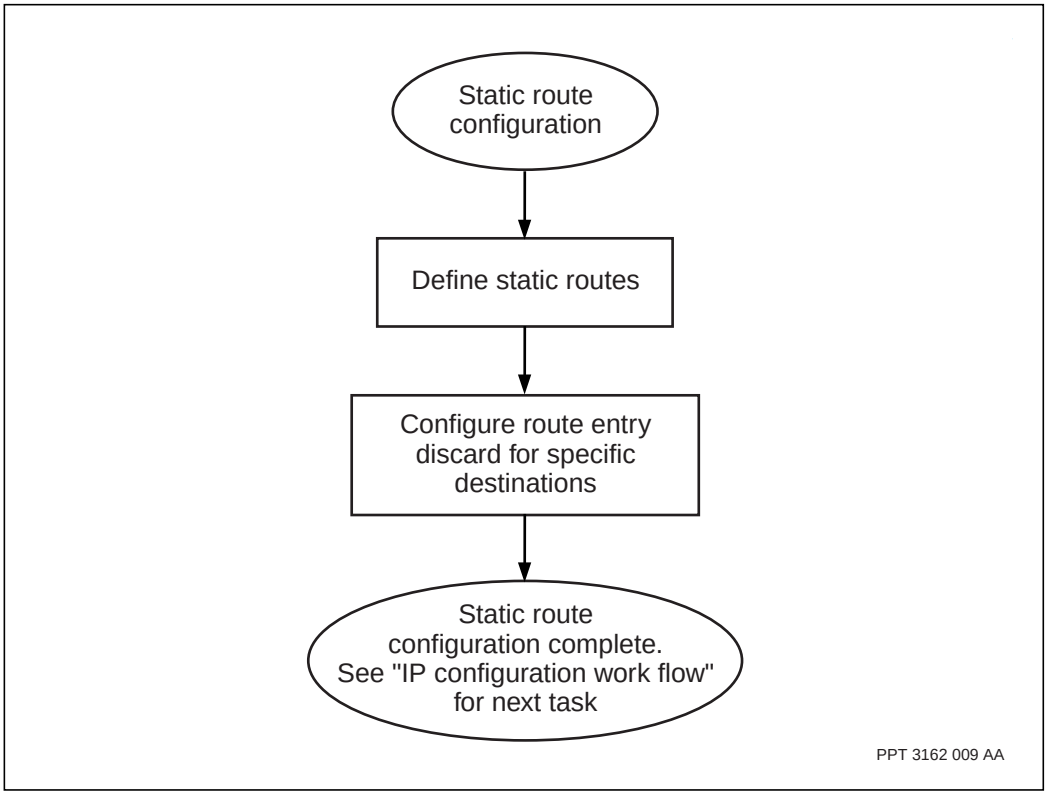
Chapter 17
Static route configuration

Use the procedures in this section to configure static routes.

Static routes configuration task flow

This task flow shows you the sequence of procedures you perform to configure static routes. To link to any procedure, go to “Navigation links” (page 210)

Figure 59
Static route configuration task flow



Navigation links

- “Defining static routes” (page 211)
- “Configuring route entry discard for specific destinations” (page 214)
- For information about the next task, see “IP configuration work flow” (page 38)

Defining static routes

Define static routes to allow the Passport system identify specific routes to remote IP networks or hosts. The definition includes a destination address, address mask, and one or more next hop addresses (gateways).

Procedure steps

1 Add a *Static* component as a subcomponent of the *Ip* component:

```
add Vr/<vr_name> Ip Static
```

2 Add static routes to the route table:

```
add Vr/<vr_name> Ip Static RouteEntry/
<ipAddress>,<destMask>,<serviceType>
```

Note 1: If `<ipAddress>` specifies a host then provision `<destMask>` as 255.255.255.255.

Note 2: Provision locally attached hosts as *Arp HostEntry* components instead of a *Static* component entry.

3 Provision at least one *NextHop* component for each defined static route. The NextHop_ipAddress parameter must denote a locally attached host, but need not be explicitly provisioned as an *Arp HostEntry*. You can add up to three *nextHop* components.

```
add Vr/<vr_name> Ip Static RouteEntry/
<ipAddress>,<destMask>,<serviceType> NextHop/
<nextHop_ipAddress>
```

4 If required, set the metric for the route:

```
set Vr/<vr_name> Ip Static RouteEntry/
<ipAddress>,<destmask>,<tos>
NextHop/<nextHop_ipAddress> metric <cost>
```

5 If required, change the route preference for static remote routes:

Either change the route preference attribute *defaultStaticRemoteRtePref*, or override the route preference by changing attribute *staticRemoteRtePreference*.

When you change the attribute *defaultStaticRemoteRtePref*, services related to that route are disrupted during activation.

To change the route preference attribute, use the following command:

```
set Vr/<vr_name> Ip Static defaultStaticRemoteRtePref
<route_pref>
```

To override the route preference, use the following command:

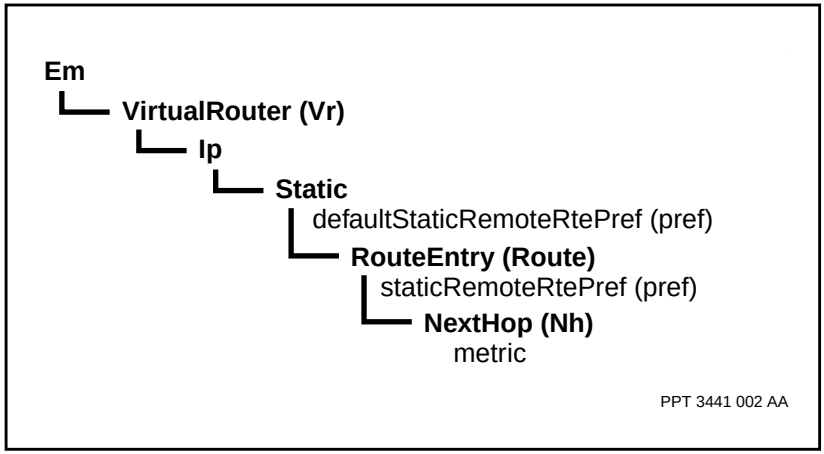
```
set Vr/<vr_name> Ip Static RouteEntry <ip_address>
staticRemoteRtePreference <override>
```

Variable definitions

Variable	Value
<cost>	A relative metric value (ranges from -1 to +65535) assigned to the static route—the assigned cost judges route preference
<destMask>	The subnetwork mask of the remote node used with the IP address.
<ipAddress>	The IP address of the remote node. It can refer either to a specific node or to a network.
<nextHop_ipAddress>	The IP address of the next router in the path to the destination. Since this is a specific node and cannot be a network, there is no subnetwork mask.
<override>	The override route preference. Attribute range is 1 to 253. Attribute default is sameAsStatic, which means use the value of attribute *defaultStaticRemoteRtePref* for the route preference. To prefer static remote routes over OSPF internal routes, the recommended setting for *staticRemoteRtePreference* is 5.
<route_pref>	The route preference. Attribute range is 1 to 253. Attribute default is 72.
<serviceType>	The type of service. Currently, only the default value of 0 is supported.
<tos>	The ToS byte value assigned to the packet.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 60
Defining static routes component hierarchy



Configuring route entry discard for specific destinations

Configure route entry discard for specific destinations to identify destination networks and nodes that do not receive packets through IP, and to discard packets addressed to these destinations.

Procedure steps

Add a *DiscardRoute* component for each route that is not to receive packets through this router:

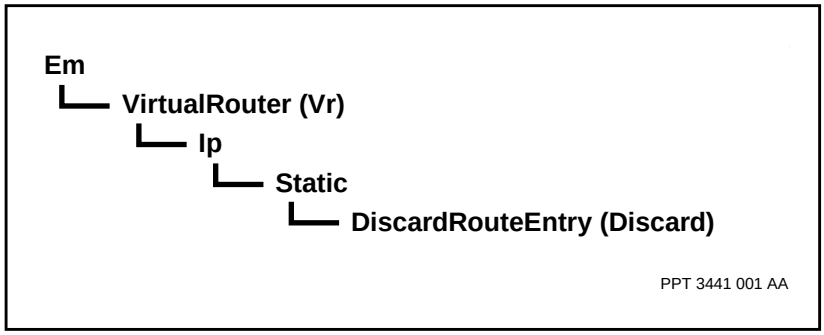
```
add Vr/<vr_name> Ip Static DiscardRouteEntry/
<destAddress>,<destMask>
```

Variable definitions

Variable	Value
<destAddress>	The IP address of the host or route whose packets are discarded.
<destMask>	The subnetwork mask associated with the destination address.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 61
Configuring route entry discard for specific destinations component hierarchy



Chapter 18
Configuring bootstrap protocol

Configure bootstrap protocol (BOOTP) to allow dynamic configuration of a booting host.

Prerequisites

- The Passport system supports the BOOTP relay agent functionality described in RFC951 and RFC1542.
- See the section on Bootstrap protocol (BOOTP) in the 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.

Procedure steps

1 Create the BOOTP component process so that the IP/UDP accepts packets destined for the BOOTP server port

```
add Vr/<vr_name> Ip BootpRelayAgent
```

2 Add a *BootpPort* subcomponent to all IP interfaces on the *Vr* component.

```
add Vr/<vr_name> ProtocolPort/<pp_name> IpPort
BootpPort
```

3 Set the *relayForwardStatus* attribute to control how BOOTP request packets, received on another port, will be flooded out this port.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
BootpPort relayForwardStatus <status_value>
```

4 Identify the logical interface whose address is to be the GIADDR of this BOOTP port.

```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
BootpPort bootpLogicalInterface <portIf__value>
```

5 Identify a list of relay addresses to receive BOOTP request messages when they are received on this port. BOOTP requests are also broadcast out all other BOOTP configured ports according to the *relayForwardStatus* attribute of the outgoing port.

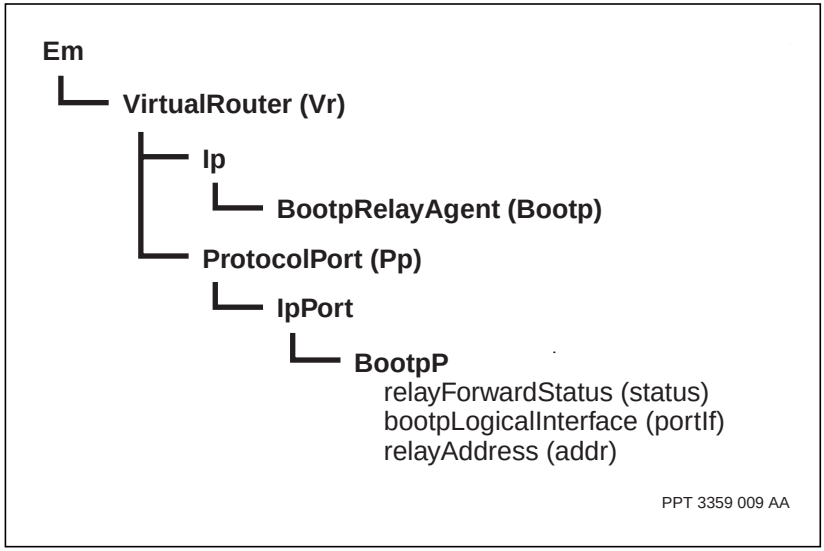
```
set Vr/<vr_name> ProtocolPort/<pp_name> IpPort
BootpPort relayAddress <addr__value>
```

Variable definitions

Variable	Value
<addr_value>	The value for the *relayAddress* attribute. There is no default value.
<portIf_value>	The value for the *bootpLogicalInteface* attribute. The default IP address is 0.0.0.0.
<pp_name>	The name of the protocol port.
<status_value>	The value for the *relayForwardStatus* attribute. The default value is disabled.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 62
Configuring bootstrap protocol component hierarchy



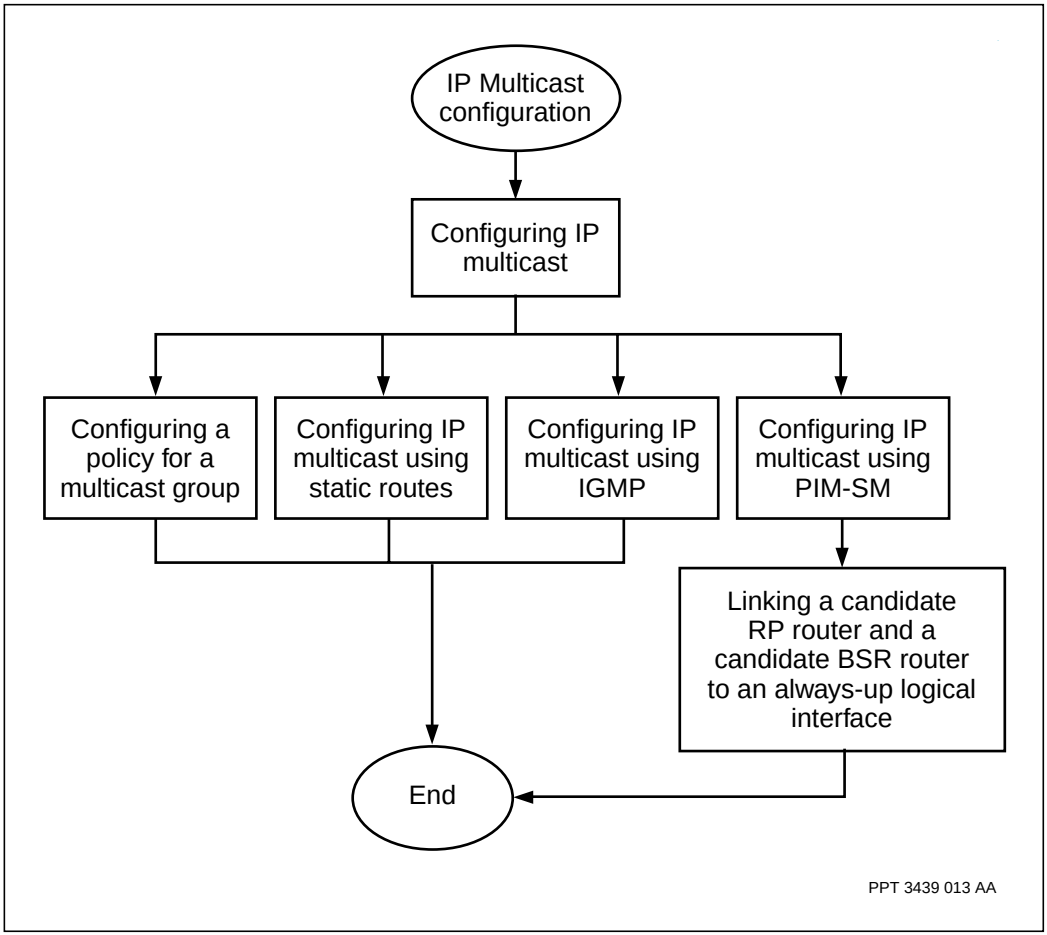
Chapter 19
Configuring IP multicast

Configure IP multicast totransmit multicast data packets to multiple receivers.

IP multicast configuration task flow

The “IP multicast configuration task flow” (page 220) shows you the sequence of procedures you perform to configure IP multicast. To link to any procedure, go to “Navigation links” (page 220).

Figure 63
IP multicast configuration task flow



Navigation links

- “Configuring IP multicast” (page 222)
- “Configuring a policy for a multicast group” (page 223)
- “Configuring IP multicast using static routes” (page 225)
- “Configuring IP multicast using IGMP” (page 226)
- “Configuring IP multicast using PIM-SM” (page 227)

- "Linking a candidate RP router and a candidate BSR router to an always-up logical interface" (page 229)
- For more information about the next task, see "IP configuration work flow" (page 38)

Configuring IP multicast

Configure IP multicast on a VR when you add the *Mcast* component.

Prerequisites

- Configure IP.

Procedure steps

1 Enable multicast on a VR.

```
add Vr/<vr_name> Ip Mcast
```

Variable definitions

Variable	Value
<vr_name>	The name of the virtual router.

Configuring a policy for a multicast group

Configure a policy for a multicast group to control multicast forwarding on specified interfaces. Policy groups can be configured to include a range of multicast group addresses and an action attribute which is set to allow or deny the forwarding of the specified groups. Policy groups may be linked to one or more interfaces.

Note: A new policy will not take effect if the multicast group addressed by the policy is already in the multicast forwarding table and if one of the OIFs for the group in the multicast forwarding table is one of the *linkToPolicyUser* ports.

Prerequisites

- Configure IP multicast.

Procedure steps

1 Add a *PolicyGroup* component.

```
add Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name>
```

2 Add a *Group* component.

```
add Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name> group/<ipAddress>,<netmask>
```

3 Set the action performed by the policy.

```
set Vr/<vr_name> Ip Mcast PolicyGroup/
<policy_group_name> action <action>
```

Variable definitions

Variable	Value
<action>	is either *allow* or *deny.*
<ip_address>	The 32 bit class D multicast group IP address for a local subnet.
<netmask>	The the 32 bit network mask you assign to this IP address.

(Sheet 1 of 2)

Variable	Value
<policy_group_name>	The descriptive name you assign to identify a particular policy group.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Configuring IP multicast using static routes

Configure IP multicast using static routes to enable the forwarding of multicast traffic without the use of multicast routing protocols. Static routes may be used alone or in combination with routing protocols. For example, PIM-SM will initiate joins for multicast groups specified by static entries as well as those learned through the use of IGMP.

Prerequisites

- Configure IP multicast.

Procedure steps

1. Configure a static route for a multicast group.

   ```
   add Vr/<vr_name> Ip Mcast Static
   add Vr/<vr_name> Ip Mcast Static Route/
   <ipAddress>,<domain>
   ```

2. Configure an out interface to a neighboring multicast router for a static route.

   ```
   set Vr/<vr_name> Ip Mcast Static Route/<ipaddress>,
   <domain> OutInterface Vr/<vr_name> Pp/<pp_id> Ipp
   logicalIf/<ipAddress>
   ```

Variable definitions

Variable	Value
<domain>	The 32 bit network mask you assign to this IP address.
<ip_address>	The 32 bit class D multicast group IP address for a local subnet.
<pp_id>	The instance name you assign to this protocol port.
<vr_name>	The name of the virtual router.

Configuring IP multicast using IGMP

Configure IP multicast to run the IGMP protocol on a virtual router.

Prerequisites

- Configure IP multicast.

Procedure steps

1. Enable the IGMP protocol on the VR by add the *Igmp* component.

   ```
   add Vr/<vr_name> Ip Mcast Igmp
   ```

2. Enable IGMP on an interface that will communicate with IGMP hosts by adding an *IgmpIf* component.

   ```
   add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/
   <ipAddress> IgmpIf
   ```

Variable definitions

Variable	Value
<ip_address>	The 32 bit address you assign to this logical interface.
<pp_id>	The instance name you assign to this protocol port.
<vr_name>	The name of the virtual router.

Configuring IP multicast using PIM-SM

Configure IP multicast with PIM-SM to route multicast traffic to sparsely populated receivers.

Prerequisites

- Configure IP multicast.

Procedure steps

1 Enable PIM-SM by adding the *PimSm* component.

```
add Vr/<vr_name> Ip Mcast PimSm
```

2 Configure a PIM-SM multicast domain on the VR by adding a *Domain* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n>
```

3 Configure a VR as a candidate RP router in a domain by adding the *CandidateRp (CRp)* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n> CandidateRp
```

4 Set the range of class D multicast addresses for the candidate RP router using the *CRp* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n> CRp Group/
<group_address>, <mask>
```

5 Configure a VR as a candidate BSR router in a domain by adding the *CandidateBsr (CBsr)* component.

```
add Vr/<vr_name> Ip Mcast PimSm Domain/<n>
CandidateBsr
```

6 Enable the PIM-SM protocol on the IP interface of the applicable protocol port by adding the *PimSmIf* component.

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/
<ipAddress> PimSmIf
```

Variable definitions

Variable	Value
<group_address>	The instance value of the group address.
<ip_address>	The 32 bit address you assign to this logical interface.
<mask>	The instance value of the mask.
<n>	The number assigned to this instance of the *Domain* component (decimal 0..3). The default value is 0.
<pp_id>	The instance name you assign to this protocol port.
<vr_name>	The name of the virtual router.

Linking a candidate RP router and a candidate BSR router to an always-up logical interface

Link a candidate RP router and a candidate BSR router to an always-up logical interface to allow them to communicate.

Note: Nortel Networks recommends that you link your *CRp* and *CBsr* components to the *Vm* component logical interface, which is an always-up interface. The *Vm* component is always reachable because it is not linked to a physical interface, thus providing a reliable, always-up service.

Prerequisites

- Configure a PIM domain.

Procedure steps

1 To link the *CRp* and *CBsr* components to an always-up logical interface, first add a *VirtualMedia* (*Vm*) component.

```
add Vm/<m>
```

Note: When you add a *Vm* component, the Passport system automatically creates an *If* subcomponent, and assigns it an instance value of 0. For example, if you add *Vm*/0, the system automatically creates *If*/0 (*Vm*/0 *If*/0)

2 Add an *IpPort* component, and set the network mask.

a. Add an *IpPort* component

```
add Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/
<ipAddress>
```

b. Set the network mask for the *IpPort* component

```
set Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/
<ipAddress> netmask <netmask>
```

3 Link the *Vm* component to the *IpPort* component you configured in the step above.

```
set Vm/<m> if/<n> linkToProtocolPort Vr/<vr_name> Pp/
<pp_id>
```

4 Set the candidate RP address such that it is the same as the address of the *Vm* component's IpPort.

```
set Vr/<vr_name> Ip Mcast PimSm Domain/<n> Crp address
Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ipAddress>
```

5 Set the candidate BSR address such that it is the same as the address of the *Vm* component's IpPort.

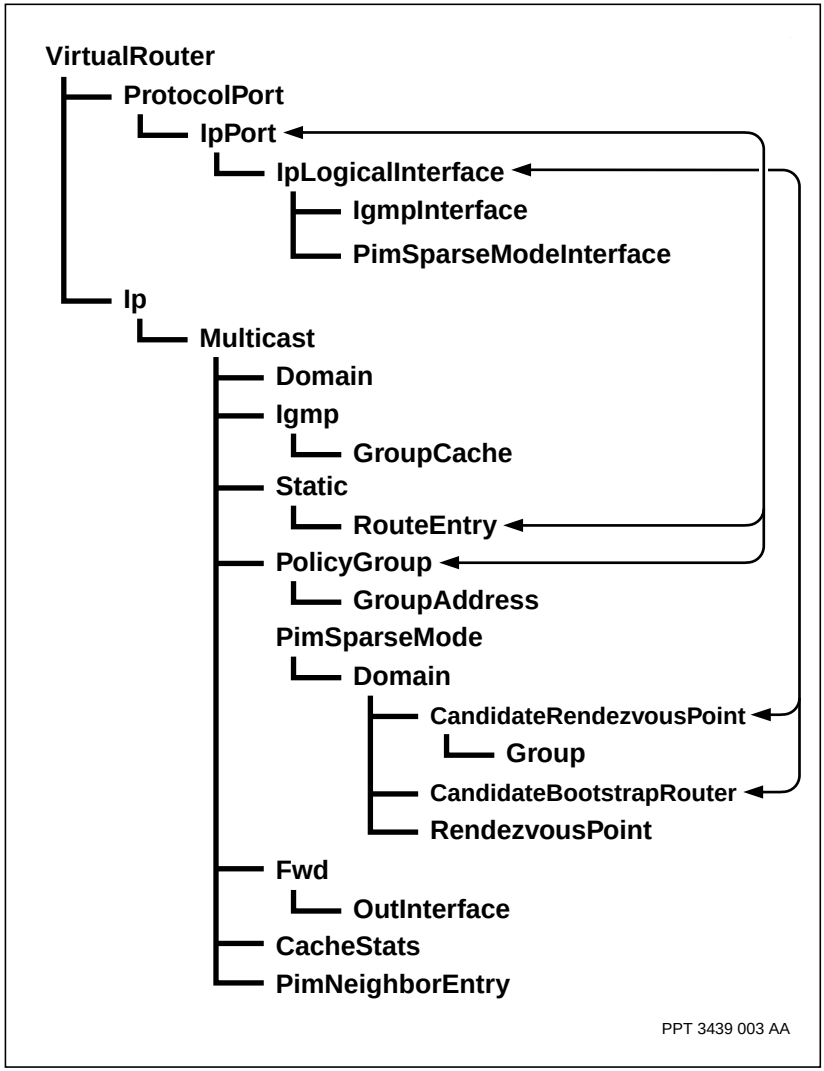
```
set Vr/<vr_name> Ip Mcast PimSm Domain/<n> CBsr address
Vr/<vr_name> Pp/<pp_id> IpPort LogicalIf/<ipAddress>
```

Variable definitions

Variable	Value
<ip_address>	The 32 bit address you assign to this logical interface.
<m>	The instance value of the *Vm* component (decimal 0..15)
<n>	The number assigned to this instance of the *Domain* component (decimal 0..3). The default value is 0.
<netmask>	The network mask to be used with the IP address.
<pp_id>	The instance name you assign to this protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 64
Configuring IP multicast component hierarchy



Chapter 20
Virtual router redundancy protocol configuration

Configure the virtual router redundancy protocol (VRRP) to enable router redundancy and availability to IP routing.

Prerequisites to virtual router redundancy protocol configuration

- The Passport system supports the VRRP functionality described in RFC 2338.
- See the section on VRRP in the 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.

CAUTION

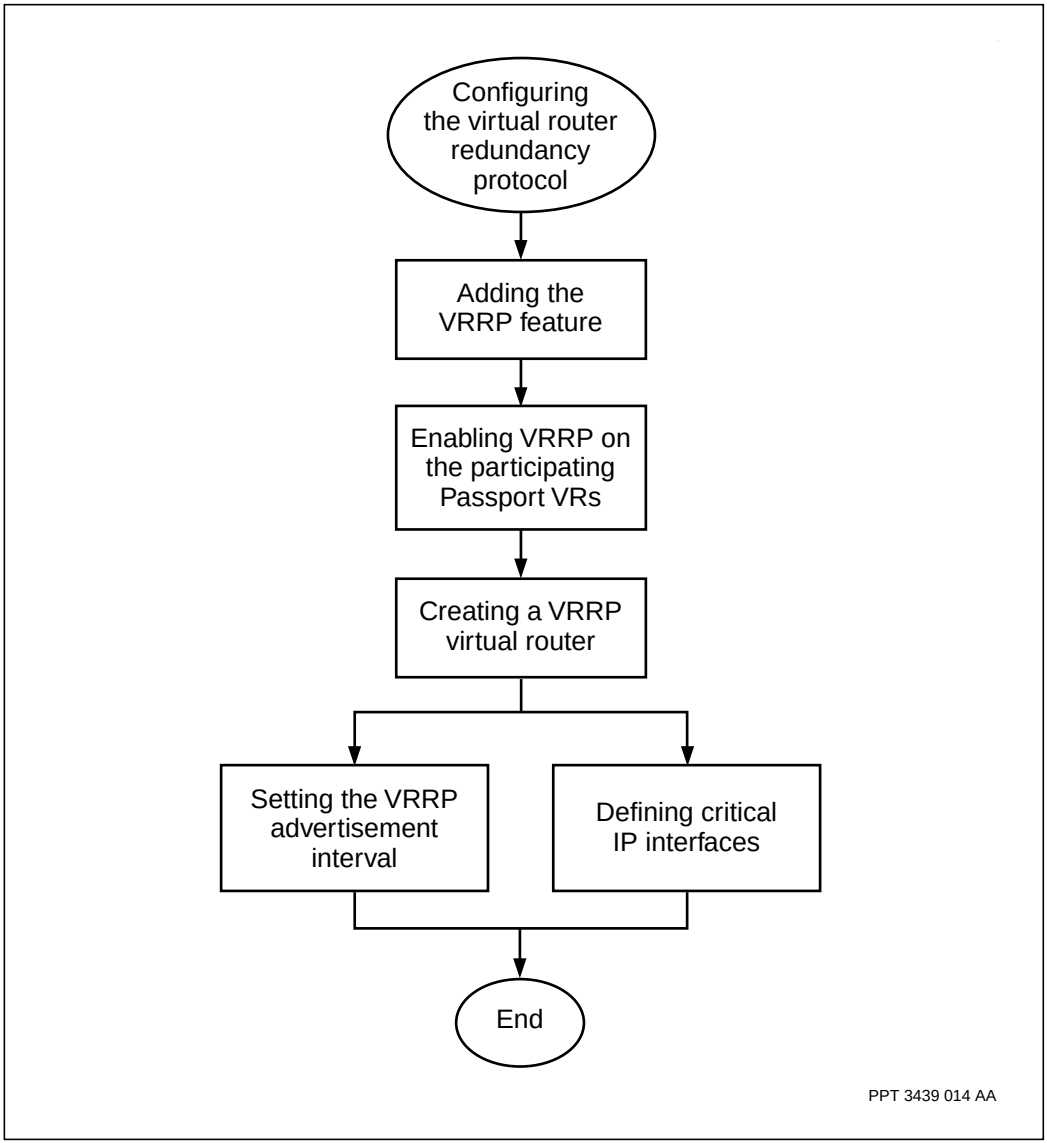
A provisioning change can affect service

Activation of provisioning changes to the *VrrpPort* could result in a brief interruption of service if the *VrrpPort* has a current *virtualRouterState* of master and does not own an IP address (priority not equal to 255).

Virtual router redundancy protocol configuration task flow

The “VRRP configuration task flow” (page 234) displays the sequence of procedures you must perform to configure VRRP. To link to any procedure, go to “Navigation links” (page 234).

Figure 65
VRRP configuration task flow



Navigation links

- “Adding the VRRP feature” (page 236)

- “Enabling VRRP on the participating Passport VRs” (page 237)
- “Creating a VRRP virtual router” (page 238)
- “Setting the VRRP advertisement interval” (page 240)
- “Defining critical IP interfaces” (page 241)

Adding the VRRP feature

Add the VRRP feature to the logical processors linked to the 100BaseT Ethernet cards.

Prerequisites

- Cards supporting VRRP also require the IP feature on the participating LPs.

Procedure steps

1 Add the VRRP feature.

```
set sw lpt/<lp_name> featureList vrrp
```

2 Check, activate and confirm the provisioning file to enable the VRRP software.

```
check Prov
```

```
activate Prov
```

```
conf Prov
```

Variable definitions

Variable	Value
<lp_name>	The ASCII string name of the lpt.

Enabling VRRP on the participating Passport VRs

Enable VRRP on the participating Passport VRs to enable the VRRP process on the participating Passport VRs. The VRRP virtual router requires the participation of two or more Passport VRs. These Passport VRs would typically be on different Passport nodes and connected over a 100 BaseT ethernet LAN segment. Alternatively, the VRRP virtual router requires the participation of at least one Passport VR and one non-Passport router compliant to RFC2338. Also, VRRP on one Passport VR can interwork with an external router implementing RFC2338-compliant VRRP on the same Ethernet LAN segment.

Prerequisites

- Add the VRRP feature.

Procedure steps

1. Enable the VRRP process on a designated master router.

   ```
   add Vr/<router_A> Ip Vrrp
   ```

2. Enable the VRRP process on a second Passport router.

   ```
   add Vr/<router_B> Ip Vrrp
   ```

3. Repeat the above step for each additional Passport VR participating in the VRRP virtual router.

Variable definitions

Variable	Value
<router_A>	An instance name of a Passport VR on Passport node A.
<router_B>	An instance name of a Passport VR on Passport node B.

Creating a VRRP virtual router

Create a VRRP virtual router to emulate a physical router in software.

Prerequisites

- Enable VRRP on the participating Passport VRs. The VRRP virtual router requires the participation of two or more Passport VRs, or at least one Passport VR and a non-Passport RFC2338-compliant router.
- The Passport VRs need to be connected over a 100 BaseT ethernet LAN segment.

Procedure steps

1 Add the *VRRP* component to the Passport VR acting as the VRRP master router.

```
add Vr/<router_A> Protocolport/<enet> IPport Vrrp/
<VRid>
```

2 Set the Ip addresses for the VRRP virtual router with which it is associated.

```
set vr/<router_A> Protocolp/<enet> IPport Vrrp/<VRid>
IpAdresses <ipaddress> <ipaddress>
```

3 Set priority of the VRRP master router.

```
set vr/<router_A> Protocolport/<n> IPport VRRP/<VRid>
Priority <priority>
```

4 Add the *VRRP* component to the Passport VR acting as the VRRP backup router.

```
add Vr/<router_B> Protocolport/<n> IPport Vrrp/<VRid>
```

5 Optionally, set the Ip addresses of VRRP virtual router.

```
set Vr/<backup> Protocolport/<enet> IPport Vrrp/<VRid>
IpAdresses <ipaddess> <ipaddress>
```

6 Set the priority of the VRRP backup router.

```
Set vr/<backup> Protocolport/<enet> IPport Vrrp/<VRid>
Priority <priority>
```

7 Optionally, repeat steps 4-6 for each additional VRRP backup router. If the priority values are the same for multiple backups, selection of a replacement master involves criteria outside the VRRP protocol.

Variable definitions

Variable	Value
<backup>	The name of the Passport VR designated as a VRRP backup router
<enet>	The instance of the ethernet protocol port
<ipaddress>	The IP addresses of the VRRP virtual router. Typically, these are the same as the IP addresses of the VRRP master router
<priority>	A decimal value between 1..254.
<router_A>	An instance name of a Passport VR on Passport node A.
<router_B>	An instance name of a Passport VR on Passport node B.
<VRid>	The decimal VRRP virtual router identifier between 1 - 255

Setting the VRRP advertisement interval

Optionally, configure the advertisement interval on both the master and backup VRRP routers through the *advertisementInterval* attribute. The master and backup routers require the same value.

Prerequisites

- Create a VRRP virtual router.

Procedure steps

1. Set the advertisement timer interval on the VRRP master router.

   ```
   set vr/<router_A> Protocolport/<enet> IPport Vrrp/
   <VRid> AdverInterval <ad_inter>
   ```

2. Set the advertisement timer interval on the VRRP backup router.

   ```
   set vr/<router_B> Protocolport/<enet> IPport Vrrp/
   <VRid> AdverInterval <ad_inter>
   ```

Variable definitions

Variable	Value
<ad_inter>	The advertisement interval time in seconds
<enet>	The instance of the ethernet protocol port
<router_A>	An instance name of a Passport VR on Passport node A.
<router_B>	An instance name of a Passport VR on Passport node B.
<VRid>	The decimal VRRP virtual router identifier between 1 - 255

Defining critical IP interfaces

Define an IP interface on the local router as critical to cause a role switch to that VRRP virtual router.

- Create a VRRP virtual router.

Procedure steps

1. Add a critical IP interface.

   add Vr/<vr_name> Protocolport/<pp_id> IPport criticalIP/<cip_id>

2. Link the critical IP interface to the VRRP virtual router.

   set Vr/<master> Protocolport/<enet> IPport Vrrp/<VRid> linktoCriticalIP Vr/<vr_name> Protocolport/<pp_id> IPport criticalIP/<vr_name>

   or

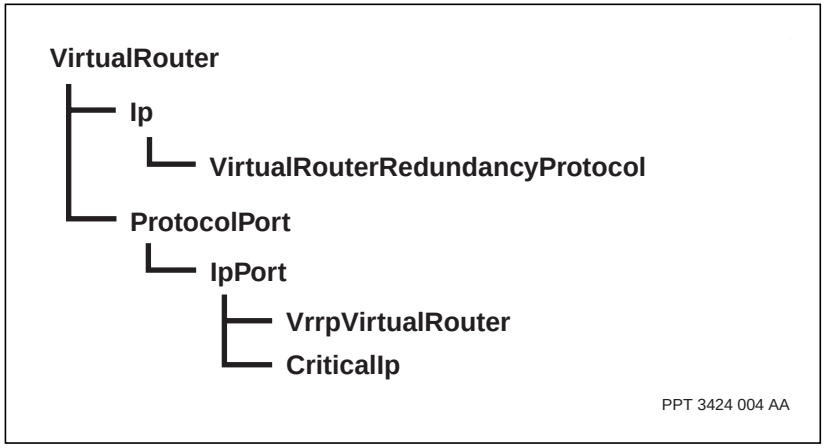
   set Vr/<vr_name> Protocolport/<pp_id> IPport criticalIP/<cip_id> linktoVrrp Vr/<master> Protocolport/<enet> IPport Vrrp/<vr_name>

Variable definitions

Variable	Value
<cip_id>	The instance (0...255) of the critical IP interface.
<enet>	The instance of the ethernet protocol port
<master>	The name of the Passport VR acting as the VRRP master router
<pp_id>	The instance value assigned to the protocol port running the IP traffic
<VRid>	The decimal VRRP virtual router identifier between 1 - 255
<vr_name>	The name of the Passport VR owning the interface

Procedure job aid

Figure 66
VRRP associated components



Chapter 21
IP class of service (CoS) configuration

Configure IP CoS to define packet classification policies and packet treatment options based on CoS values.

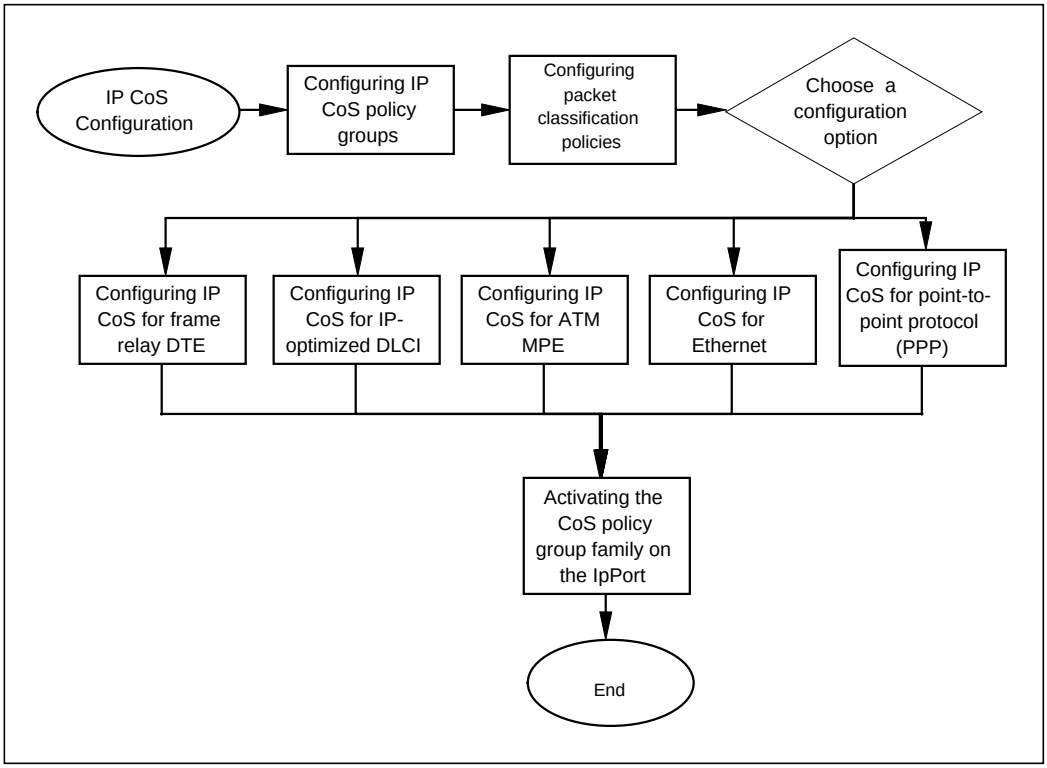
Prerequisites to IP CoS configuration

- Configure IP tunnels if you intend to configure IP CoS over IP tunnels. See “Configuring point-to-point tunnels” (page 309).

IP CoS configuration task flow

This task flow shows you the sequence of procedures you perform to configure IP CoS. To link to any procedure, go to “Navigation links” (page 244).

Figure 67
IP CoS configuration task flow



Navigation links

- “Configuring IP CoS policy groups” (page 246)
- “Configuring packet classification policies” (page 249)
- “Configuring IP CoS for frame relay DTE” (page 252)
- “Configuring IP CoS for IP-optimized DLCI” (page 254)
- “Configuring IP CoS for ATM MPE” (page 257)
- “Configuring IP CoS for Ethernet” (page 259)
- “Configuring IP CoS for point-to-point protocol (PPP)” (page 261)
- “Activating the CoS policy group family on the IpPort” (page 262)

- For information about the next task, see “IP configuration work flow” (page 38)

Configuring IP CoS policy groups

Configure IP CoS policy groups to define packet classification policies and specify packet treatment options for policy matches.

Prerequisites

- Create policy groups for all protocol ports where flow-based classification is done on ingress and/or DSCP marking, or where emission priority assignment is done on egress.

Procedure steps

1 Create an instance of an IP CoS policy group for the virtual router (VR).

When you add the *CosPolicyGroup (Pg)* component, the system automatically creates four CoS treatment profiles under the CoS policy group for CoS indices 0, 1, 2 and 3.

```
add Vr/<vr_name> Ip Pg/<grp>
```

2 If you want the VR or a specific protocol port to mark packets with a specific CoS value on egress, configure IP CoS to change the packet's DSCP field under the appropriate CoS treatment profile.

When you enable packet marking, IP CoS updates the packet's ToS byte based on the values configured for the *tos* and *tosMask* attributes.

```
set Vr/<vr_name> Ip Pg/<grp> EgressCosTreatment/<n>
setTosByte <yes_no>
```

3 If you have enabled packet marking, specify the value of the DSCP field to be assigned to the packet.

If packet marking is not enabled under the *setTosByte* attribute, IP CoS ignores the values configured for the *tos* attribute.

```
set Vr/<vr_name> Ip Pg/<grp> EgressCosTreatment/<n>
tos <tos>
```

4 If you have enabled packet marking, specify the bits of the DSCP field that are to be updated. If packet marking is not enabled under the *setTosByte* attribute, IP CoS ignores the ToS mask value.

All CoS treatment profiles under the same policy group must have the same ToS mask.

IP CoS also uses this value to determine which bits to examine when applying a DSCP-based classification policy. A mask of 0 is invalid.

On PQC FPs all values are treated as OxFC.

set Vr/<vr_name> Ip Pg/<grp> EgressCosTreatment/<n> tosMask <mask>

5 If you want the VR to assign an emission priority (on applicable media) to packets with a specific CoS value, set an emission priority value.

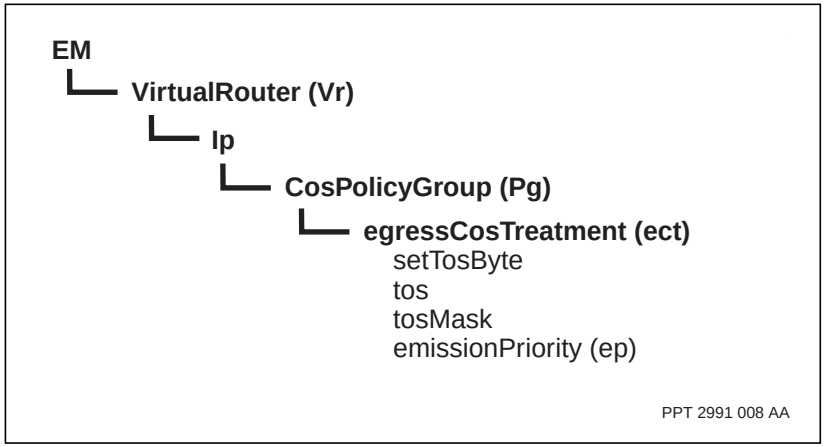
set Vr/<vr_name> Ip Pg/<grp> EgressCosTreatment/<n> ep <ep>

Variable definitions

Variable	Value
<ep>	The emission priority assigned to the packet.
<grp>	The identifier for a a common set of IP CoS policies on the VR.
<n>	The CoS value.
<tos>	The value of the DSCP value assigned to the packet.
<tosMask>	The bits in the DSCP field that are updated.
<vr_name>	The name of the virtual router.
<yes_no>	Specifies whether DSCP marking is enabled.

Procedure job aid

Figure 68
Configuring IP CoS policy groups component hierarchy



Configuring packet classification policies

Configure packet classification policies to define the criteria that IP CoS uses to determine when the policy is applied.

Procedure steps

1. If one does not already exist, create an instance of an IP CoS policy group under the virtual router.

   ```
   add Vr/<vr_name> Ip Pg/<grp>
   ```

2. Create an instance of a CoS policy under the policy group:

   ```
   add Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
   ```

3. Assign a CoS value to the policy. This value corresponds to a packet treatment profile defined under the parent policy group.

   ```
   set Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
   assignedCos <n>
   ```

4. If you want to configure DSCP-based packet classification, create an instance of a mapping policy and specify one or more DSCP field values for a policy match.

   IP CoS uses the mask specified under the policy's associated CoS treatment profile (in the same policy group) to determine which bits to examine for DSCP-based classification.

   Note: IP CoS also uses the ToS mask to determine which bits of the DSCP field to update when packet marking is applied.

   ```
   add Vr/<vr_name> Ip Pg/<grp> Policy/<policy> TosMap
   ```

   ```
   set Vr/<vr_name> Ip Pg/<grp> Policy/<policy> TosMap
   tos <tos>
   ```

5. If you want to configure flow-based packet classification, create one or more instances of a flow identification policy.

   ```
   add Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
   IpAddrLayer4Flow/<flow>
   ```

6. If you want to configure flow-identification based on the traffic's layer 4 protocol, specify a protocol for a policy match.

   ```
   set Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
   IpAddrLayer4Flow/<flow> proto <protocol>
   ```

7 If you want to configure flow-identification based on source or destination IP addressing, specify a prefix and prefix length for a policy match. An address of 0.0.0.0 or a prefix length of 0 signifies any IP address.

If the traffic flow's protocol type has been set to ICMP, IP CoS ignores the values configured for address-based classification.

```
set Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
IpAddrLayer4Flow/<flow> prefix <x.x.x.x>
```

```
set Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
IpAddrLayer4Flow/<flow> len <prlen>
```

8 If you want to configure flow-identification based on TCP or UDP port numbers, specify a port number (or range of port numbers) for a policy match. A port number of 0 signifies any port number.

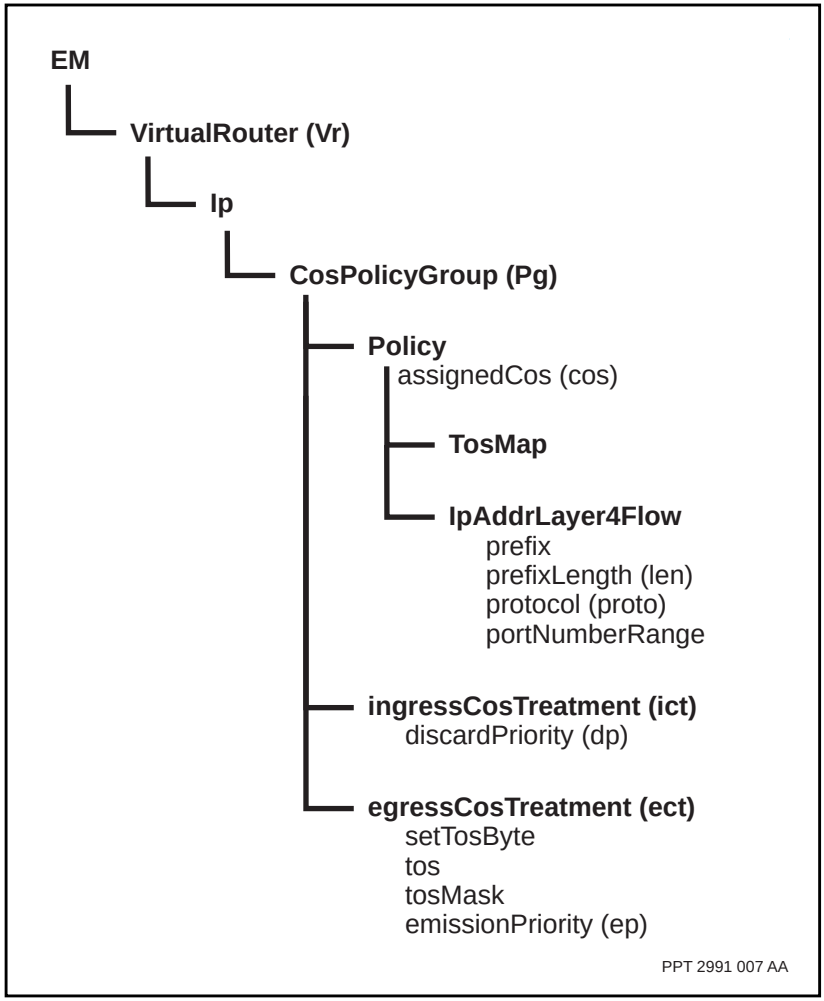
```
set Vr/<vr_name> Ip Pg/<grp> Policy/<policy>
IpAddrLayer4Flow/<flow> portNumberRange <min> <max>
```

Variable definitions

Variable	Value
<flow>	The instance of the flow identification policy.
<grp>	The identifier for a a common set of IP CoS policies on the VR.
<max>	The upper limit of the range of port numbers for which the policy applies.
<min>	The lower limit of the range of port numbers for which the policy applies.
<n>	A CoS value.
<policy>	The policy within the policy group.
<prlen>	The number of most significant bits in the IP address prefix.
<protocol>	The layer 4 protocol for which the policy applies.
<tos>	The DSCP field value(s) for which the policy applies.
<vr_name>	The name of the virtual router.
<x.x.x.x>	The IP address prefix for which the policy applies.

Procedure job aid

Figure 69
Configuring IP CoS packet classification policies



Configuring IP CoS for frame relay DTE

Configure IP CoS for frame relay DTE to enable support for CoS to QoS mapping over multiple DLCIs or a single DLCI.

Prerequisites

- If you add a static DLCI with a different CoS to a *FrDte Remote Group* component that has other DLCIs in operation, lock and unlock the *IpPort* component of the frame relay DTE interface's linked protocol port.
- If you clear some (but not all) IP ARP entries for the same IP address, lock and unlock the *IpPort* component of the *FrDte* component's linked protocol port, so that the system automatically relearns all relevant ARP entries.
- If there are multiple frame relay connections under a frame relay DTE interface with the same CoS value, only one registers with the IP ARP table.

Procedure steps

1 If one does not already exist, create a static DLCI to the next IP hop for a specific CoS priority.

For details on configuring frame relay DTE static DLCIs, see "Frame relay DTE configuration for IP over frame relay" (page 53).

```
add FrDte/<fr> StDlci/<stdlci_no>
```

2 Assign an IP CoS value to the DLCI.

When an IP packet arrives on this DLCI, Passport assigns the configured IP CoS value to the packet. Passport also uses this value to select a DLCI for outgoing IP packets with a matching CoS value.

```
set FrDte/<fr> StDlci/<stdlci_no> ipCos <cos>
```

Variable definitions

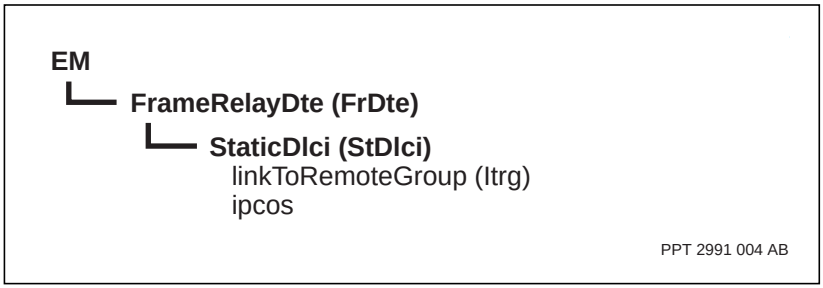
Variable	Value
<cos>	The CoS value associated with packets on the static DLCI.

(Sheet 1 of 2)

Variable	Value
<fr>	The instance number of the frame relay DTE interface.
<stdlci_no>	The instance number of the static DLCI.
(Sheet 2 of 2)	

Procedure job aid

Figure 70
Configuring IP CoS for frame relay DTE component hierarchy



Configuring IP CoS for IP-optimized DLCI

Configure IP CoS for IP-optimized DLCI to support CoS to QoS mapping.

Prerequisites

- If you change the CoS value for an FrConnection, lock and unlock the *IpDlciGroup Frc* component before the change can take effect.
- If you are provisioning FRF.12 on an IP-optimized DLCI, you can provision CoS using this procedure, except to specify an emission priority. Set the attribute *IpDlciGroup Frc ipCos*, see “Configuring an end-to-end FRF.12 DTE at the local Passport FrUni interface” (page 76) for more information.
- If there are multiple frame relay connections under an IP-optimized DLCI with the same CoS value, only one registers with the IP ARP table.

Procedure steps

1 If one does not already exist, create an IP-optimized DLCI to the next IP hop for a specific CoS priority.

For details on configuring IP-optimzed DLCIs, see “IP-optimized DLCI configuration for IP over frame relay” (page 71).

```
add IpDlciGroup/<dlci_grp>
```

2 Assign an IP CoS value to the DLCI.

When an IP packet arrives on this DLCI, Passport assigns the configured IP CoS value to the packet. Passport also uses this value to select a DLCI for outgoing IP packets with a matching CoS value.

```
set IpDlciGroup/<dlci_grp> Frc/<frc> ipCos <cos>
```

3 If you want, specify a drop precedence, or discard priority, at ingress for both DE0 and DE1 traffic.

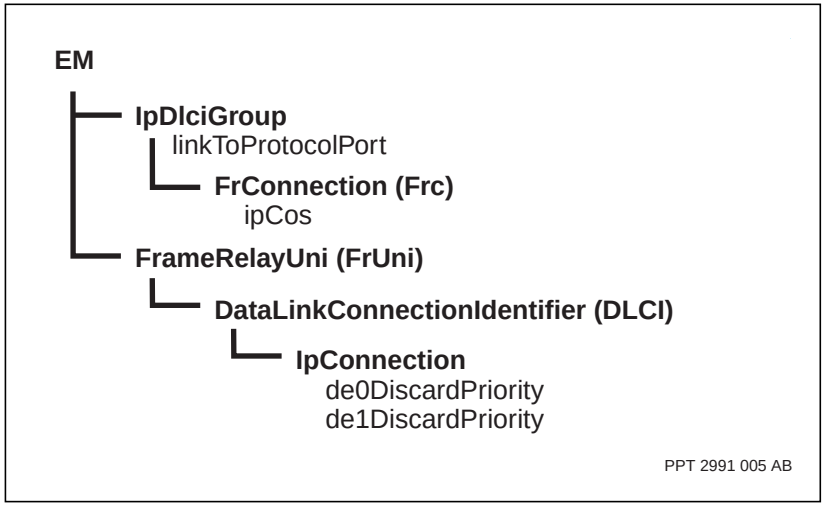
```
Set FrUni/<o> Dlci/<p> IpConnection de0DiscardPriority
<de0_dp>, de1DiscardPriority <de1_dp>
```

Variable definitions

Variable	Value
<cos>	The CoS value associated with packets on the DLCI.
<de0_dp>	The drop precedence assigned at ingress for DE0 traffic.
<de1_dp>	The drop precedence assigned at ingress for DE1 traffic.
<dlci_grp>	The instance value of the interface between the virtual router protocol port and the IP-optimized DLCI.
<frc>	The instance value of the frame relay connection.
<o>	The instance value of the FrUni.
<p>	The instance value of the DLCI.

Procedure job aid

Figure 71
Configuring IP CoS for IP-optimized DLCI component hierarchy



Configuring IP CoS for ATM MPE

Configure IP CoS for ATM MPE to support CoS to QoS mapping for IP packets transmitted and received over multiple virtual channel connections (VCCs) or over a single VCC.

Prerequisites

- If you change the CoS value for an ATM MPE VCC, lock and unlock the *AtmMpe Ac* component before the changes can take effect.
- If you create or delete an *AtmMpe Ac* component under an ATM MPE interface with other VCCs in operation, lock and unlock the *Vr Pp IpPort* component under the protocol port associated with the ATM MPE interface to enable service on the new VCC.
- If there are multiple ATM MPE VCCs under an ATM MPE interface with the same CoS value, only one registers with the IP ARP table. If that VCC goes down, another *AtmMpe Ac* component with that CoS value will register with the IP ARP table.

Procedure steps

1 If one does not already exist, create an ATM MPE VCC to the next IP hop for a specific CoS priority.

For details on configuring ATM MPE VCCs, see “Configuring an ATM PVC for an ATM MPE interface” (page 45).

```
add AtmMpe/<mpe> Ac/<ac>
```

2 Assign an IP CoS value to the VCC.

When an IP packet arrives on this VCC, Passport assigns the configured IP CoS value to the packet. Passport also uses this value to select a VCC for outgoing IP packets with a matching CoS value.

```
set AtmMpe/<mpe> Ac/<ac> ipCos <cos>
```

Variable definitions

Variable	Value
<ac>	The instance number of the VCC.
<cos>	The CoS value associated with packets on the VCC.

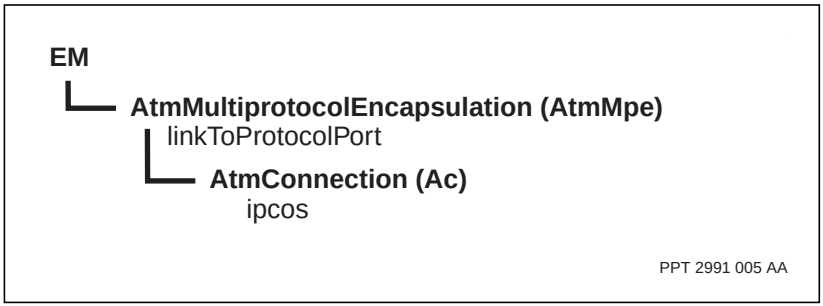
(Sheet 1 of 2)

Variable	Value
<mpe>	The instance number of the ATM MPE interface.

(Sheet 2 of 2)

Procedure job aid

Figure 72
Configuring IP CoS for ATM MPE component hierarchy



Configuring IP CoS for Ethernet

Configure IP CoS for Ethernet to allow flows of IP packets that will be transmitted over an Ethernet interface to be differentiated by QoS characteristics.

This procedure is valid for

- 10BaseT Ethernet
- 100BaseT Ethernet
- gigabit Ethernet

  Note: Gigabit Ethernet only supports layer 2 CoS classification, so the remaining CoS procedures do not apply to it, namely, "Configuring IP CoS policy groups" (page 246) and "Configuring packet classification policies" (page 249).

Prerequisites

- For traffic management information on the FP that supports gigabit Ethernet, see 241-5701-615 *Passport 7400, 15000, 20000 FP Configuration Reference*.

Procedure steps

1 Assign a CoS value to the Ethernet protocol port. Passport assigns this CoS value to all IP packets arriving on the Ethernet interface.

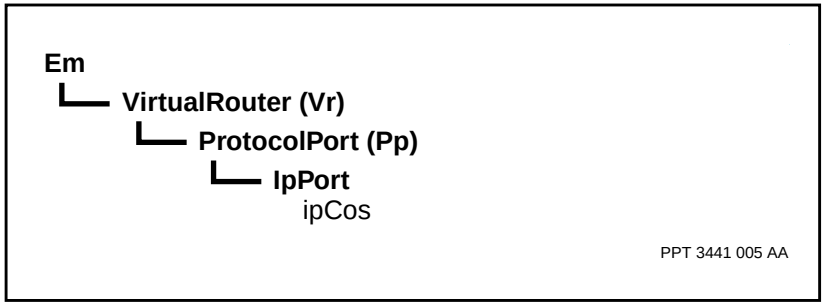
```
set Vr/<vr_name> Pp/<pp_name> IpPort ipCos <cos>
```

Variable definitions

Variable	Value
<cos>	The CoS value to be assigned to incoming packets.
<pp_name>	The name of the protocol port associated with the Ethernet interface.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 73
Configuring IP CoS for gigabit Ethernet component hierarchy



Configuring IP CoS for point-to-point protocol (PPP)

Configure IP CoS for point-to-point protocol (PPP) to allow flows of IP packets that will be transmitted over a PPP interface to be differentiated by QoS characteristics.

Procedure steps

1. Assign a CoS value to the PPP protocol port. Passport assigns this CoS value to all IP packets arriving on the PPP interface.

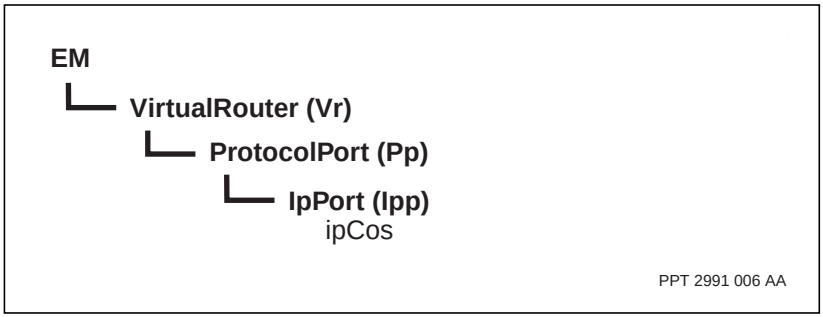
```
set Vr/<vr_name> Pp/<pp_name> IpPort ipCos <cos>
```

Variable definitions

Variable	Value
<cos>	The CoS value to be assigned to incoming packets.
<pp_name>	The name of the protocol port associated with the PPP interface.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 74
Configuring IP CoS for PPP component hierarchy



Activating the CoS policy group family on the IpPort

Activate the CoS policy group family on the IpPort to link the provisioned ingress and egress classification and treatment schemes to the IP packets that go through the specified IpPort.

Note: This procedure does not apply to Gigabit Ethernet interfaces.

Prerequisites

- Determine if you want to activate a CoS policy group family on a specific IpPort or all the IpPorts of a virtual router.

Procedure steps

1 Activate a cos Policy Assignment profile on all IpPorts of a virtual router.

```
set Vr/<vr_name> Ip cosPolicyAssignment Vr/<vr_name>
Ip CosPolicyGroup/<cos_pg1>
```

2 Activate a cos Policy Assignment profile on a specific IpPort of the virtual router. This assignment overrides the assignment for the virtual router at this IpPort.

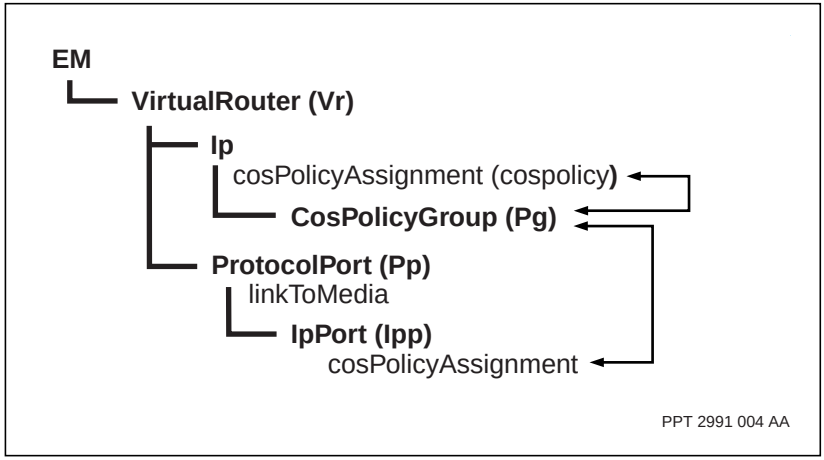
```
set Vr/<vr_name> Pp/<pp_name> IpPort
cosPolicyAssingment Vr/<vr_name> Ip CosPolicyGroup/
<cos_pg2>
```

Variable definitions

Variable	Value
<cos_pg1>	The instance of the IP CoS component whose interface profile applies to all IpPorts on the virtual router.
<cos_pg2>	The instance of the IP Cos component whose interface profile applies to a specific IpPort.
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 75
Activating the CoS policy group family on the IpPort component hierarchy



Chapter 22
IP DiffServ configuration

Configure IP DiffServ to provide a framework that allows you to provide IP traffic management at each virtual router in an IP network.

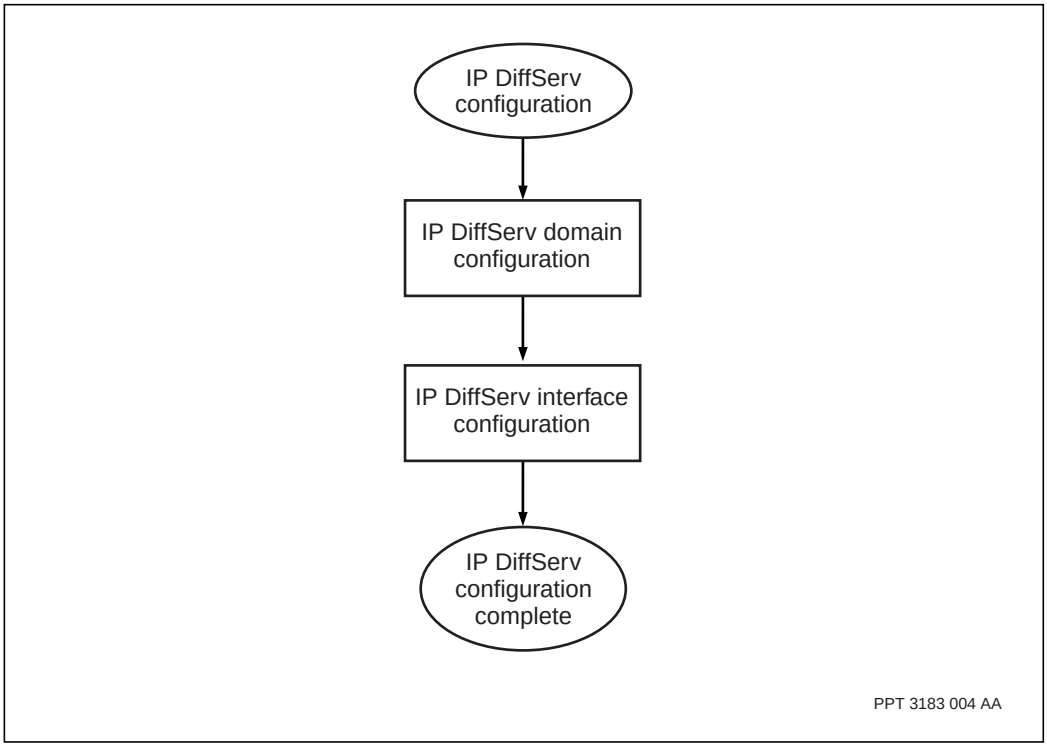
Prerequisites to IP DiffServ configuration

- The appropriate software release is installed according to the procedures in the 241-5701-270 *Passport 7400, 15000, 20000 Software Installation Guide*.
- The appropriate software, applications and features have been configured according to the procedures in the 241-5701-600 *Passport 7400, 15000, 20000 Configuration Guide*.
- IP services framework of virtual routers and media connections must be provisioned according to the “IP configuration work flow” (page 37).

IP DiffServ configuration work flow

This task flow shows you the sequence of procedures you must perform to configure IP DiffServ. To link to any procedure, go to “Work flow navigation” (page 266).

Figure 76
IP DiffServ configuration work flow



Work flow navigation

- "IP DiffServ domain configuration" (page 267)
- "IP DiffServ interface configuration" (page 270)
- Return to "IP configuration work flow" (page 38)

IP DiffServ domain configuration

Configure an IP differentiated services domain to apply differentiated services and a DiffServ domain to a virtual router, locally generated packets and the connected media links.

- "Prerequisites to IP DiffServ domain configuration" (page 267)
- "IP DiffServ domain configuration task flow" (page 267)

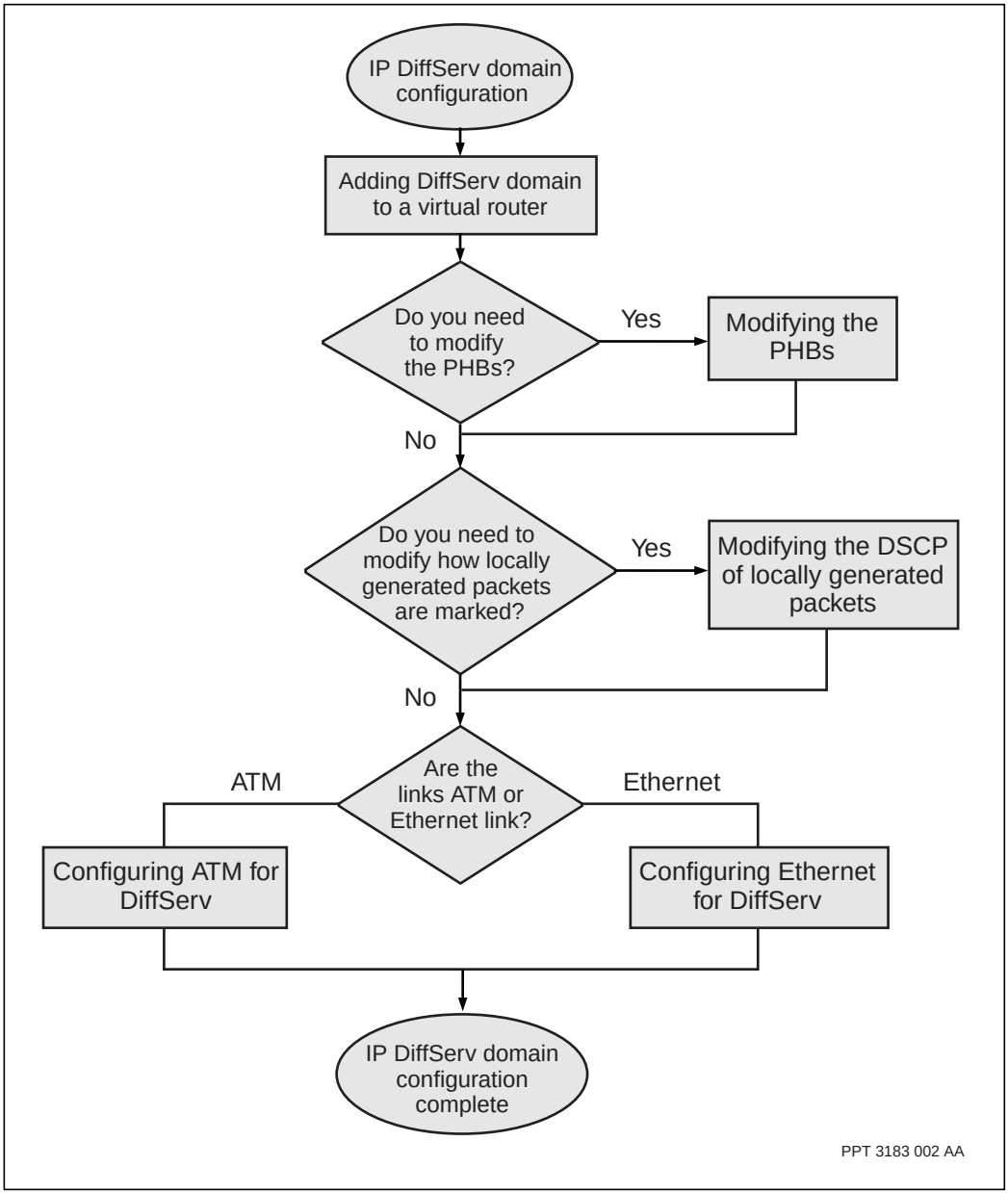
Prerequisites to IP DiffServ domain configuration

- You have a network plan for your differentiated services according to 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.

IP DiffServ domain configuration task flow

This task flow shows you the sequence of procedures you perform to enable IP DiffServ. To link to any procedure, go to "Task flow navigation" (page 269).

Figure 77
IP DiffServ domain configuration task flow



Task flow navigation

- “Adding a DiffServ domain to the virtual router” (page 274)
- “Modifying per-hop behaviors” (page 275)
- “Modifying DSCP marking for locally generated packets” (page 277)
- “Configuring an ATM link for IP DiffServ” (page 279)
- “Configuring an Ethernet link for IP DiffServ” (page 281)
- Return to “IP DiffServ configuration work flow” (page 266)

IP DiffServ interface configuration

Configure an IP DiffServ interface to apply differentiated services to IP packets as they are received and transmitted at an IpPort on the virtual router.

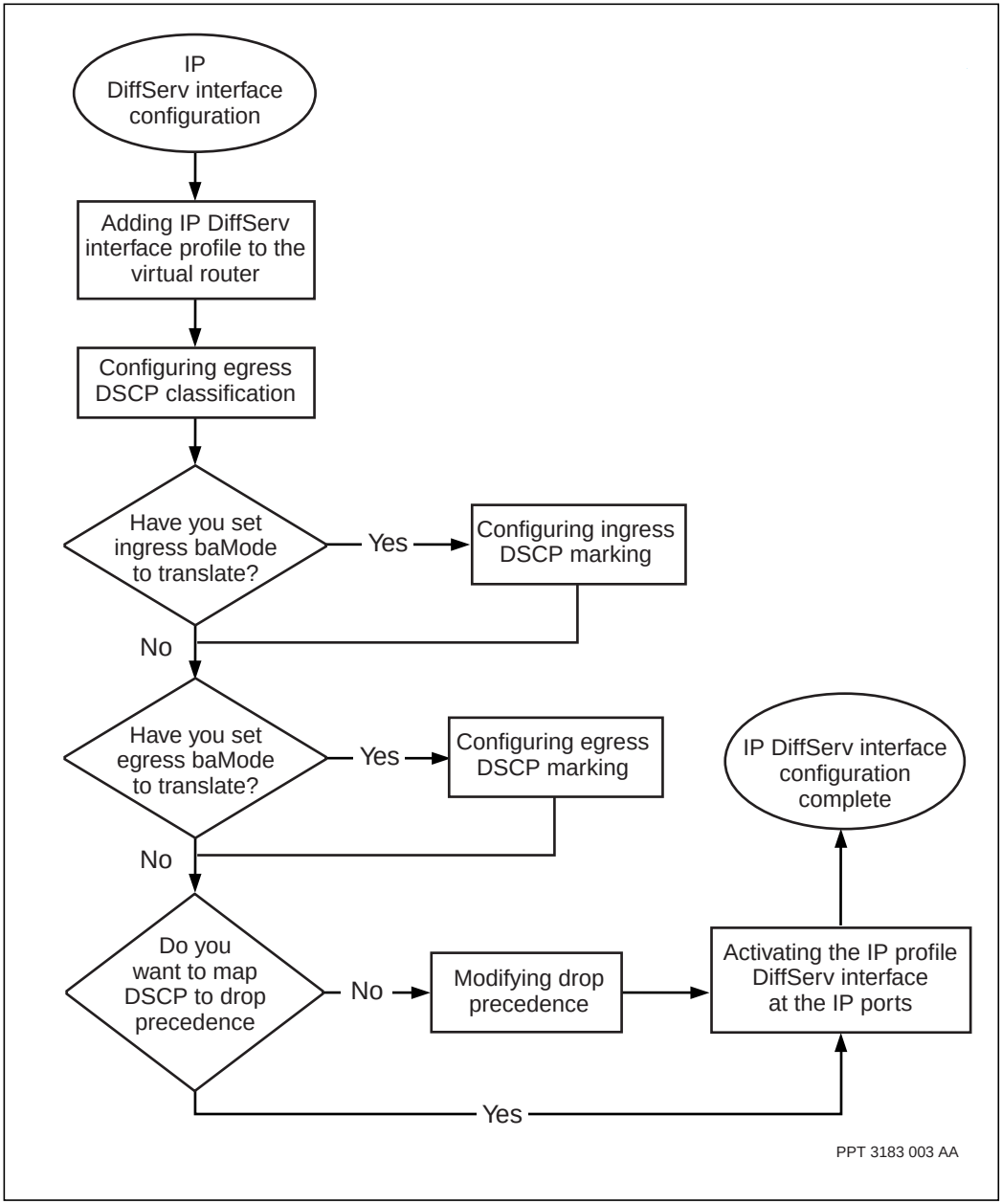
- "Prerequisites to IP DiffServ interface configuration" (page 270)
- "IP DiffServ interface configuration task flow" (page 271)

Prerequisites to IP DiffServ interface configuration

- You have a network plan for your differentiated services according to 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.
- You have identified whether each IpPort will be an IP DiffServ domain edge, domain core, or domain boundary interface.

IP DiffServ interface configuration task flow

This task flow shows you the sequence of procedures you perform to configure an IP DiffServ interface. To link to any procedure, go to “Task flow navigation” (page 273).


IP
DiffServ interface
configuration
Adding IP DiffServ
interface profile to the
virtual router
Configuring egress
DSCP classification
Have you set
ingress baMode
to translate?
Yes
Configuring ingress
DSCP marking
No
Have you set
egress baMode
to translate?
Yes
Configuring egress
DSCP marking
No
Do you
want to map
DSCP to drop
precedence
No
Modifying drop
precedence
Activating the IP profile
DiffServ interface
at the IP ports
IP DiffServ interface
configuration
complete
Yes
PPT 3183 003 AA

Task flow navigation

- “Adding an IP DiffServ interface profile to a virtual router” (page 283)
- “Configuring ingress DSCP translation” (page 285)
- “Configuring egress DSCP translation” (page 287)
- “Modifying IP DiffServ drop precedence mode” (page 289)
- “Activating the IP DiffServ interface on the IpPort” (page 291)
- Return to “IP DiffServ configuration work flow” (page 266)

Adding a DiffServ domain to the virtual router

Add a DiffServ domain to a virtual router to support IP DiffServ and to make the virtual router part of a group (domain) of routers that have the same per-hop-behavior definitions.

Procedure steps

1. To set the per hop behaviors (PHBs) for IP DiffServ, add the DiffServ domain to the virtual router.

   ```
   add Vr/<vr_name> Dsd/<domain_type>
   ```

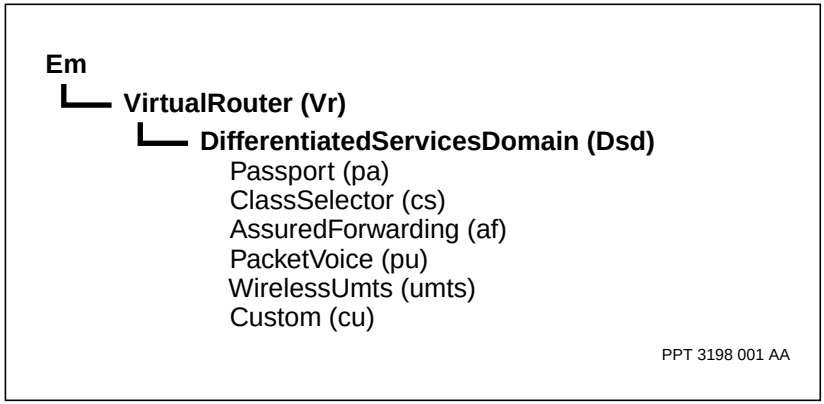
2. Complete the next procedure in the "IP DiffServ domain configuration task flow" (page 267) before activating the provisioning view.

Variable definitions

Variable	Value
<domain_type>	The type of the DiffServ domain you require.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 78
Adding a DiffServ domain to the virtual router component hierarchy



Modifying per-hop behaviors

Modify per-hop behaviors (PHB) to deliver specific differentiated service treatments for your network.

Prerequisites

- You must know which Per-hop-behaviors you want to modify.
- You must know how you want the Per-hop-behaviors to be scheduled and discarded relative to other PHBs in the same domain.

Procedure steps

1. Add a PHB to the domain.

   ```
   add Vr/<vr_name> Dsd/<domain_type> Phb/<phb_value>
   ```

2. Set the traffic class for your PHB.

   ```
   set Vr/<vr_name> Dsd/<domain_type> Phb/<phb_value>
   trafficClass <traffic_value>
   ```

3. Set the drop precedence for your PHB.

   ```
   set Vr/<vr_name> Dsd/<domain_type> Phb/<phb_value>
   dropPrecedence <drop_value>
   ```

4. Complete the next procedure in the “IP DiffServ domain configuration task flow” (page 267) before activating the provisioning view.

Variable definitions

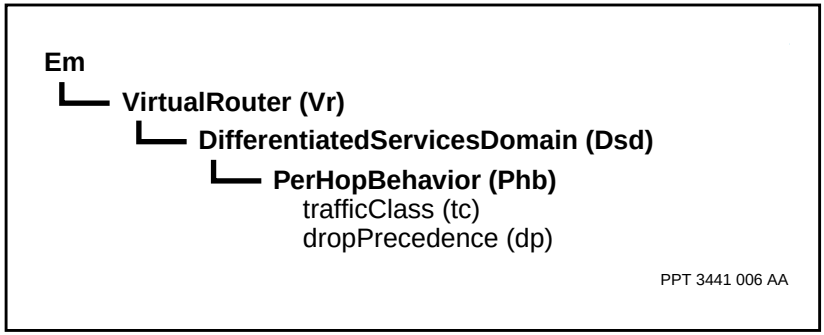
Variable	Value
<domain_type>	The type of differentiated services domain being used.
<drop_value>	The setting that specifies how this PHB is dropped relative to other PHBs.
<phb_value>	The DSCP or RFC representation of the value you are modifying.

(Sheet 1 of 2)

Variable	Value
<traffic_value>	The value that specifies how this PHB is scheduled relative to other PHBs.
<vr_name>	The name of the virtual router.
(Sheet 2 of 2)	

Procedure job aid

Figure 79
Modifying per-hop-behaviors component hierarchy



Modifying DSCP marking for locally generated packets

Modify DSCP marking for locally generated packets to change how packets generated at the virtual router are classified for differentiated services.

Prerequisites

- Identify the PHB you want to assign to the locally generated packets.

  Note: To pass provisioning semantic checks, the PHB used for locally generated packets must match a PHB in your differentiated service domain.

Procedure steps

1. To configure the PHB value for locally generated BGP, RIP, and OSPF packets, set the *phbRoutingSource* attribute.

   Note: This attribute should be left at its default value to be compliant with RFC791.

   ```
   set Vr/<vr_name> Dsd/<domain_type> phbr <phb_value>
   ```

2. To configure the PHB value for any locally generated packets other than BGP, RIP, or OSPF, set the *phbGeneralSource* attribute.

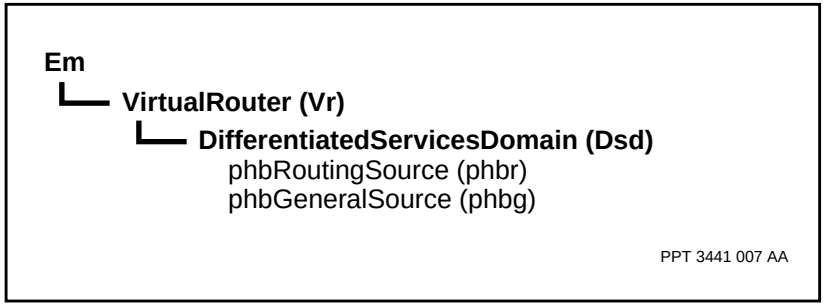
   ```
   set Vr/<vr_name> Dsd/<domain_type> phbg <phb_value>
   ```

Variable definitions

Variable	Value
<domain_type>	The type of differentiated services domain being used.
<phb_value>	The specific DSCP value you want to assign for that packet type (Default = 0 (df) for *phbGeneralSource* and 48 (cs6) for *phbRoutingSource*).
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 80
Modifying DCSP marking for locally generated packets component hierarchy



Configuring an ATM link for IP DiffServ

Configure an ATM link for IP DiffServ to set the DSCP that will be assigned to the packet if the ingress services *baMode* is set to *link* and the connected media is *AtmMpe*.

Prerequisites

- You must have all ATM connections configured according to "ATM MPE configuration for IP over ATM" (page 41).
- You must have completed the procedure "Adding a DiffServ domain to the virtual router" (page 274).
- You must know what DSCP and connection class values you want to associate with the ATM links.
- If you change the *ipDscp* or *ipCos* value for an ATM MPE VCC, lock and unlock the *AtmMpe Ac* component before the changes can take effect.
- If you create or delete an *AtmMpe Ac* component under an ATM MPE interface with other VCCs in operation, lock and unlock the *Vr Pp IpPort* component under the protocol port associated with the ATM MPE interface to enable service on the new VCC.
- If there are multiple ATM MPE VCCs under an ATM MPE interface with the same connection class value, only one registers with the IP ARP table. If that VCC goes down, another *AtmMpe Ac* component with that connection class value will register with the IP ARP table.

Procedure steps

1. Set the link mode *ipDscp* attribute for the ATM link. Passport uses this value to mark IP packets when the ingress classification and marking mode is set to link on the IP DiffServ interface profile.

   ```
   set AtmMpe/<atmmpe_inst> Ac/<ac_inst> ipDscp
   <ipdscp_value>
   ```

2. Set the connection class value for the same ATM link. Passport uses this value to select a VCC for outgoing IP packets that are assigned a matching connection class value.

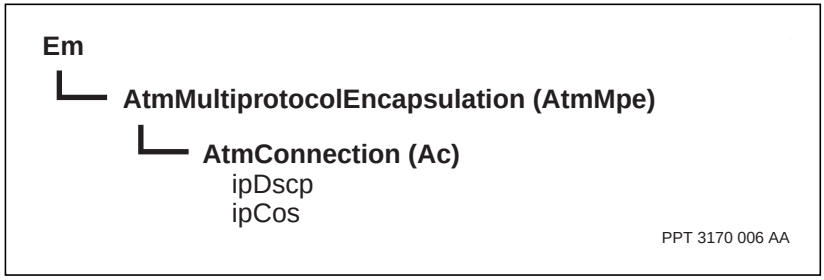
   ```
   set AtmMpe/<atmmpe_inst> Ac/<ac_inst> ipCos
   <ipcos_value>
   ```

Variable definitions

Variable	Value
<ac_inst>	The instance assigned to the AtmConnection.
<atmmpe_inst>	The instance assigned to the AtmMpe.
<ipcos_value>	The connection class value assigned for the ATM link.
<ipdscp_value>	In link mode, the DSCP value assigned to packets arriving on the link.

Procedure job aid

Figure 81
Configuring an ATM link for IP DiffServ component hierarchy



Configuring an Ethernet link for IP DiffServ

Configure an Ethernet link for IP DiffServ to set the DSCP value that will be assigned to the packet if the ingress services *baMode* is set to *link* and the connected media is *Ethernet*.

This procedure is valid for

- gigabit Ethernet
- 100BaseT Ethernet FPs that support IP DiffServ

Prerequisites

- You must have Ethernet connections configured according to the “IP configuration work flow” (page 37).
- You must have completed the procedure “Adding a DiffServ domain to the virtual router” (page 274).
- You must know what DSCP and connection class values you want to associate with the Ethernet links.

Procedure steps

1. Set the link mode *ipDscp* attribute for the Ethernet link.

   ```
   set Vr/<vr_name> Pp/<pp_name> IpPort ipDscp
   <ipdscp_value>
   ```

2. Set the connection class value for the IP port.

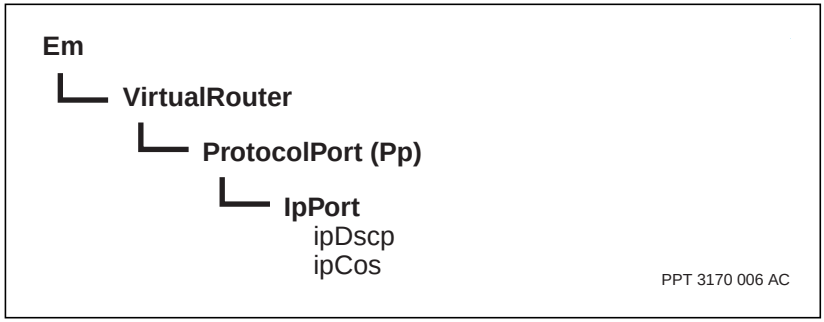
   ```
   set Vr/<vr_name> Pp/<pp_name> IpPort ipCos
   <ipcos_value>
   ```

Variable definitions

Variable	Value
<ipcos_value>	The connection class value assigned to the link.
<ipdscp_value>	In link mode, the DSCP value assigned to packets arriving on the link.
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 82
Configuring an Ethernet link for IP DiffServ component hierarchy



Adding an IP DiffServ interface profile to a virtual router

Add an IP DiffServ interface profile to a virtual router to support IP DiffServ ingress and egress interface profiles on the IpPorts of a virtual router.

Prerequisites

- For an IP DiffServ boundary interface you must know if you need to translate the DSCP on the ingress or egress flow.

Procedure steps

1. Add an IP DiffServ interface profile to the virtual router.

   ```
   add Vr/<vr_name> Ip Diffserv/<ds_inst>
   ```

2. Set the ingress services DSCP handling mode.

   ```
   set Vr/<vr_name> Ip Diffserv/<ds_inst> Is baMode
   <treatment>
   ```

3. Set the egress services DSCP handling mode.

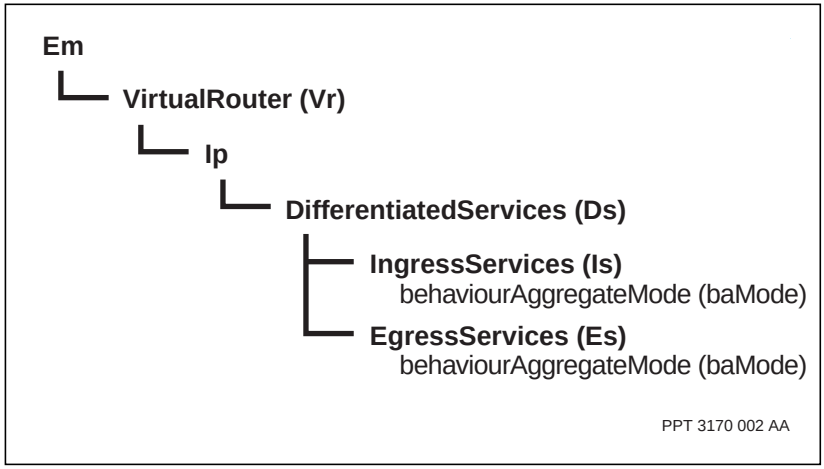
   ```
   set Vr/<vr_name> Ip Diffserv/<ds_inst> Es baMode
   <treatment>
   ```

Variable definitions

Variable	Value
<ds_inst>	The instance of the IP DiffServ component.
<treatment>	*Preserve*, *link* or *translate* depending if this is a domain core, domain edge or domain boundary interface.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 83
Adding IP DiffServ interface profile to a Vr component hierarchy



Configuring ingress DSCP translation

Configure ingress DSCP translation to set what DSCP values you want to identify and change.

Prerequisites

- You must have completed the procedure "Adding an IP DiffServ interface profile to a virtual router" (page 283) for a boundary interface and set the ingress services to *translate*.
- You must know the DSCP values you want to change and the DSCP values to which you want to map the existing values.

Procedure steps

1. Add a behavior aggregate component.

   ```
   Add Vr/<vr_name> Ip DiffServ/<ds_inst> Is Ba/<ba_inst>
   ```

2. Identify the DSCP value you want to change.

   ```
   set vr/<vr_name> Ip Diffserv/<ds_inst> Is Ba/<ba_inst>
   dscpToMatch <match_list>
   ```

3. Set the new DSCP value you want to have.

   ```
   set vr/<vr_name> Ip Diffserv/<ds_inst> Is Ba/<ba_inst>
   dscpToMark <mark_value>
   ```

   Note: Repeat this procedure as many times as necessary to translate to the DSCPs arriving from the adjacent domain.

Variable definitions

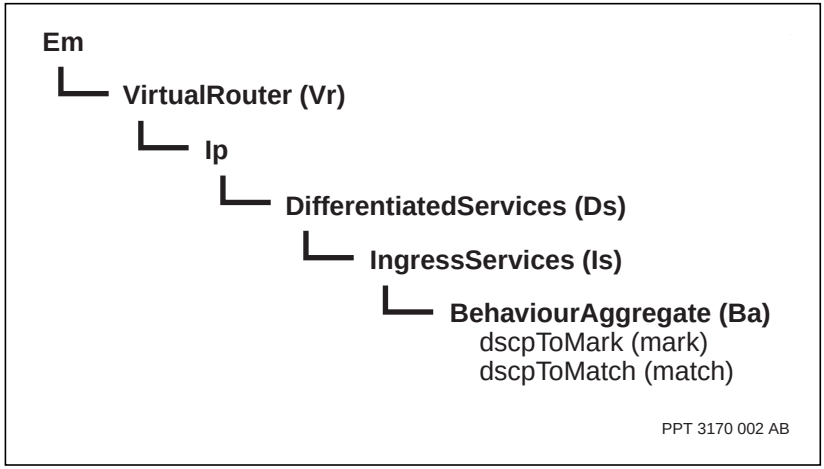
Variable	Value
<ba_inst>	The instance of the behavior aggregate
<ds_inst>	The instance of the IP DiffServ component.
<mark_value>	The DSCP value you want to set on the packets identified by the *dscpToMatch* attribute.

(Sheet 1 of 2)

Variable	Value
<match_list>	A list of one or more DSCP values you want to change to a value specified by the *dscpToMark* attribute. You can specify up to 64 DSCP values to match.
<vr_name>	The name of the virtual router.
(Sheet 2 of 2)	

Procedure job aid

Figure 84
Configuring ingress DSCP translation component hierarchy



Configuring egress DSCP translation

Configure egress DSCP translation to set what DSCP values you want to identify and change.

Prerequisites

- You must have completed the procedure "Adding an IP DiffServ interface profile to a virtual router" (page 283) for a boundary interface and have set the egress services to *translate*.
- You must know the DSCP values you want to change and the DSCP values to which you want to map the existing values.
- This process is not supported when the ingress interface that receives the packet is physically located on a 4pGe FP. For more information, see the section on domain boundary interfaces in 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP*.

Procedure steps

1. Add a behavior aggregate component.

   ```
   Add Vr/<vr_name> Ip DiffServ/<ds_inst> Es Ba/<ba_inst>
   ```

2. Identify the DSCP value you want to change.

   ```
   set Vr/<vr_name> Ip DiffServ/<ds_inst> Es Ba/<ba_inst>
   dscpToMatch <match_list>
   ```

3. Set the new DSCP value you want to have.

   ```
   set vr/<vr_name> Ip DiffServ/<ds_inst> Es Ba/<ba_inst>
   dscpToMark <mark_value>
   ```

   Note: Repeat this procedure as many times as necessary to translate to the DSCPs transmitted to the adjacent domain.

Variable definitions

Variable	Value
<ba_inst>	The instance of the behavior aggregate.
<ds_inst>	The instance of the IP DiffServ component.

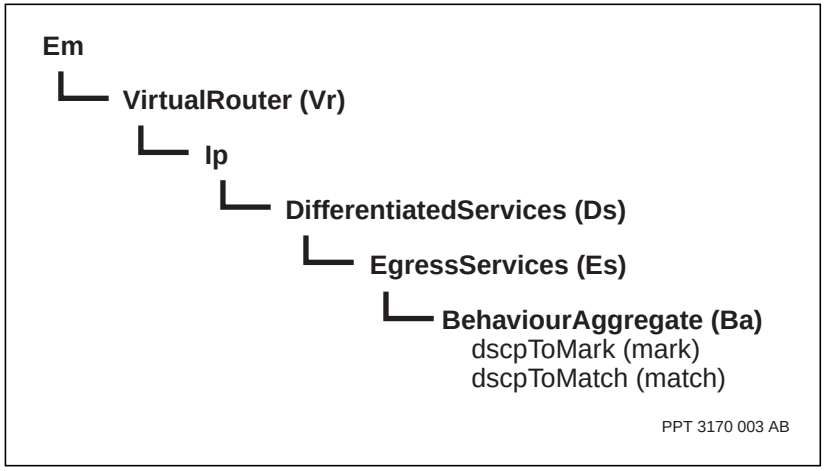
(Sheet 1 of 2)

Variable	Value
<mark_value>	The DSCP value you want to set on the packets identified by the *dscpToMatch* attribute.
<match_list>	A list of one or more DSCP values you want to change to a value specified by the *dscpToMark* attribute. You can specify up to 64 DSCP values to match.
<vr_name>	The name of the virtual router.

(Sheet 2 of 2)

Procedure job aid

Figure 85
Configuring egress DSCP translation component hierarchy



Modifying IP DiffServ drop precedence mode

Modify IP DiffServ drop precedence (or discard priority) mode to determine how IP packets are assigned a drop precedence.

Prerequisites

- You must have completed the procedure "Adding an IP DiffServ interface profile to a virtual router" (page 283).
- Identify whether you want to map the drop precedence to the DSCP or set the drop precedence priority according to the link:
  - For ATM interfaces the CLP bit is mapped to the drop precedence according to AtmIf provisioning.
  - For frame relay interfaces, the DE bit is mapped to the drop precedence according to FRUNI provisioning.
  - For Ethernet interfaces, the drop precedence from the link is always *high*.

Procedure steps

1 Set how IP DiffServ determines drop precedence for ingress services.

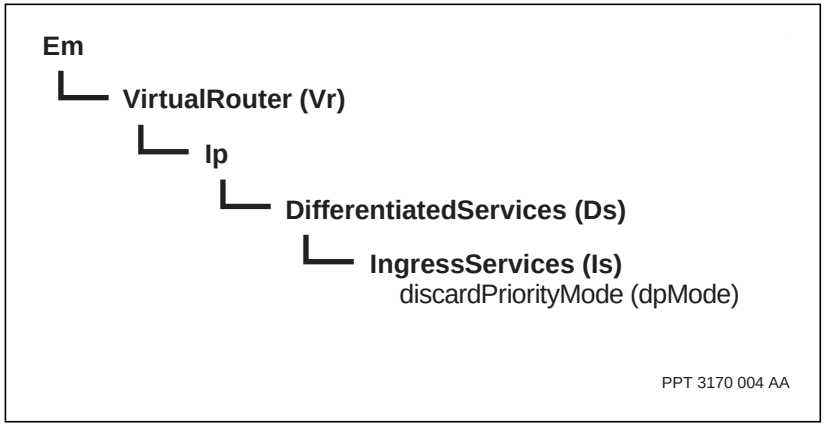
```
set Vr/<vr_name> Ip DiffServ Is dpMode <dp_mode>
```

Variable definitions

Variable	Value
<dp_mode>	The value that sets the drop precedence to match the DSCP or retain the drop precedence of the link.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 86
Configuring IP DiffServ drop precedence component hierarchy



Activating the IP DiffServ interface on the IpPort

Activate the IP DiffServ interface profile on the IpPort to link the provisioned ingress and egress classification and marking schemes to the IP packets that go through the specified IpPort.

This procedure is required when the DSCP of the IP packets received and transmitted at the IpPort have a mode of *link* or *translate*. It is not required for *preserve* mode.

Prerequisites

- You must complete "Adding a DiffServ domain to the virtual router" (page 274).
- For all interfaces you must have completed the procedure "Adding an IP DiffServ interface profile to a virtual router" (page 283).
- For boundary interfaces you must have completed the procedures "Configuring ingress DSCP translation" (page 285) and "Configuring egress DSCP translation" (page 287).
- Determine if you want to activate an IP DiffServ profile on a specific IpPort or all the IpPorts of a virtual router.

Procedure steps

1. Activate an IP DiffServ interface profile on all IpPorts of a virtual router.

   set Vr/<vr_name> Ip diffServAssignment Vr/<vr_name> Ip DiffServ/<ds_inst1)

2. Activate an IP DiffServ interface profile on a specific IpPort of the virtual router. This assignment overrides the assignment for the virtual router at this IpPort.

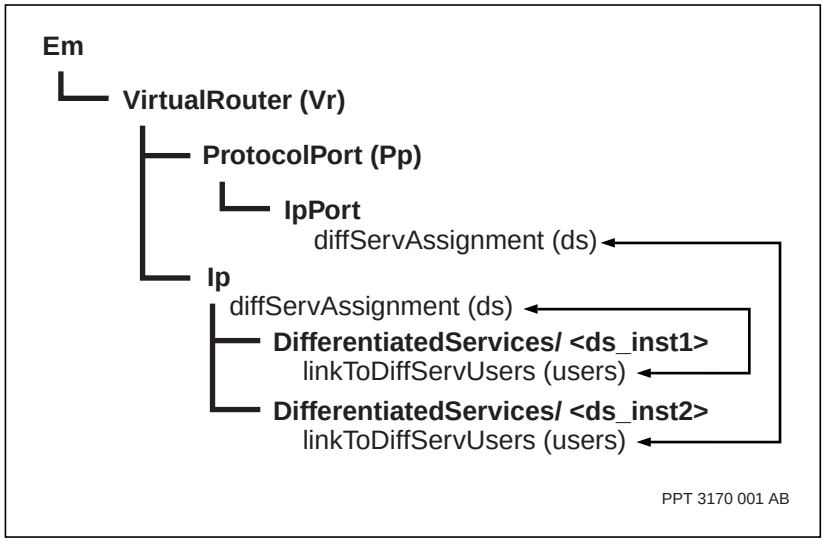
   set Vr/<vr_name> Pp/<pp_name> IpPort diffServAssignment Vr/<vr_name> Ip DiffServ/<ds_inst2>

Variable definitions

Variable	Value
<ds_inst1>	The instance of the IP DiffServ component whose interface profile applies to all IpPorts on the virtual router.
<ds_inst2>	The instance of the IP DiffServ component whose interface profile applies to a specific IpPort.
<pp_name>	The name of the protocol port.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 87
Activating the IP DiffServ interface on the IpPort component hierarchy



Chapter 23
IP CoS to IP DiffServ migration

Migrate IP CoS to IP DiffServ to offer a better framework of IP differentiated services in your existing IP CoS based IP network.

Prerequisites to IP CoS to IP DiffServ migration

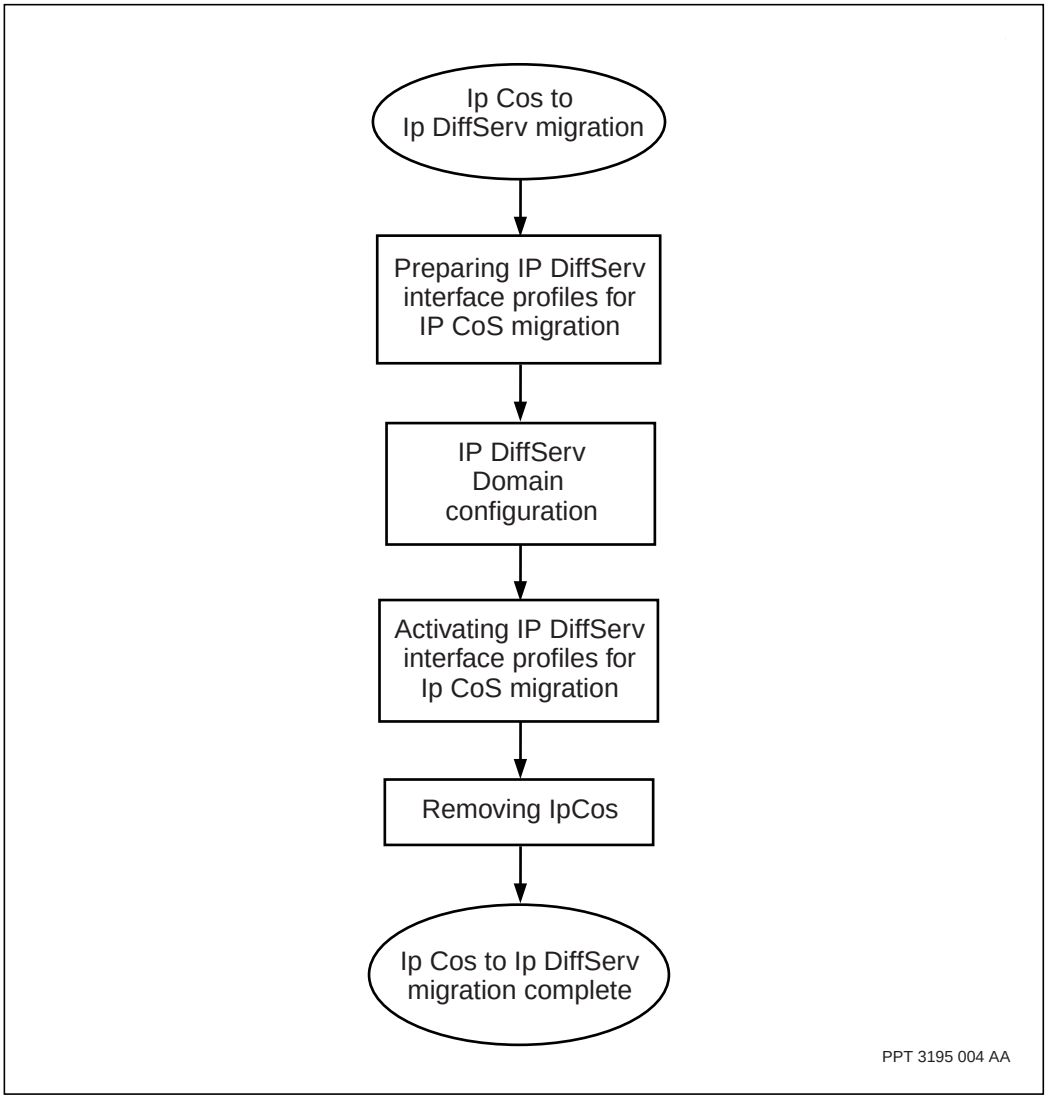
- All existing or desired DiffServ domains and domain interfaces in the network have been identified. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on DiffServ domains.
- All Passport virtual routers in each domain have been identified. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information about virtual routers.
- The *VirtualRouter ProtocolPort linkToMedia* are only deployed on the FPs that support DiffServ. See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on the FPs that support DiffServ.
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* or 241-5701-060 *Passport 7400, 15000, 20000 Components* to ensure the following conditions are satisfied:
  - The virtual router does not have any *Ip Tunnel* subcomponents.
  - The *vpnMode* of the virtual router is set to *customer*.
  - The virtual router does not have *Ip CosPolicyGroup ipAddressLayer4Flow* subcomponents.

- The *ipDiffServ* feature may need to be added to the *Sw Lpt featurelist* for each LP where DiffServ is required. See the section on Application and feature names for IP on Passport in 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information.
- Migrate all virtual routers in a DiffServ domain before proceeding to the virtual routers in another DiffServ domain.

IP CoS to IP DiffServ migration task flow

This task flow shows you the sequence of procedures you must perform to migrate IP CoS to IP DiffServ. To link to any procedure, go to “Navigation links” (page 295).

Figure 88
IP CoS to IP DiffServ migration task flow



Navigation links

- “Preparing IP DiffServ interface profiles for IP CoS migration” (page 297)

- “IP DiffServ domain configuration” (page 267)
- “Activating IP DiffServ interface profiles for IP CoS migration” (page 300)
- “Removing IP CoS” (page 302)
- For information about the next task, see “IP configuration work flow” (page 38)

Preparing IP DiffServ interface profiles for IP CoS migration

Prepare IP DiffServ interfaces for IP CoS migration to control DSCP marking and classification at each IpPort of the virtual router.

Prerequisites

- Migrate all virtual routers in an IP DiffServ domain before proceeding to the virtual routers in another IP DiffServ domain.

Procedure steps

1. Display all the CosPolicyGroup components that have been provisioned on the Vr.

   ```
   d Vr/<vr_name> Ip Pg/*
   ```

2. Add a DiffServ component to replace each CosPolicyGroup component.

   ```
   add Vr/<vr_name> Ip Ds/<ds_inst>
   ```

3. Set the interface classification and marking mode for the ingress services of each DiffServ component.

   ```
   set Vr/<vr_name> Ip Ds/<ds_inst> Is baMode <treatment>
   ```

4. Set the discard priority selection mode for the ingress services of each DiffServ component.

   ```
   set Vr/<vr_name> Ip Ds/<ds_inst> Is dpMode <dp_mode>
   ```

5. Set the interface classification and marking mode for the egress services of each DiffServ component.

   ```
   set Vr/<vr_name> Ip Ds/<ds_inst> Es baMode <treatment>
   ```

6. Add behavior aggregate subcomponents to the ingress services that are translating the DSCP (boundary interfaces).

   ```
   add vr/<vr_name> Ip Ds/<ds_inst> Is Ba/<ba_inst>
   ```

7. Identify which DSCP values to translate for each ingress service that is using translate (boundary interfaces).

   ```
   set vr/<vr_name> Ip Ds/<ds_inst> Is Ba/<ba_inst>
   dscpToMatch <match_list>
   ```

8. Identify the DSCP values to mark each packet with for the ingress services that are using translate (boundary interfaces).

   ```
   set vr/<vr_name> Ip Ds/<ds_inst> Is Ba/<ba_inst>
   dscpToMark <mark_value>
   ```

9 Add behavior aggregate subcomponents to the egress services that are translating the DSCP (boundary interfaces).

```
add vr/<vr_name> Ip Ds/<ds_inst> Es Ba/<ba_inst>
```

10 Identify which DSCP values to translate for each egress service that is using translate.

```
set vr/<vr_name> Ip Ds/<ds_inst> Es Ba/<ba_inst>
dscpToMatch <match_list>
```

11 Identify the DSCP values to mark each packet with for the egress services that are using translate.

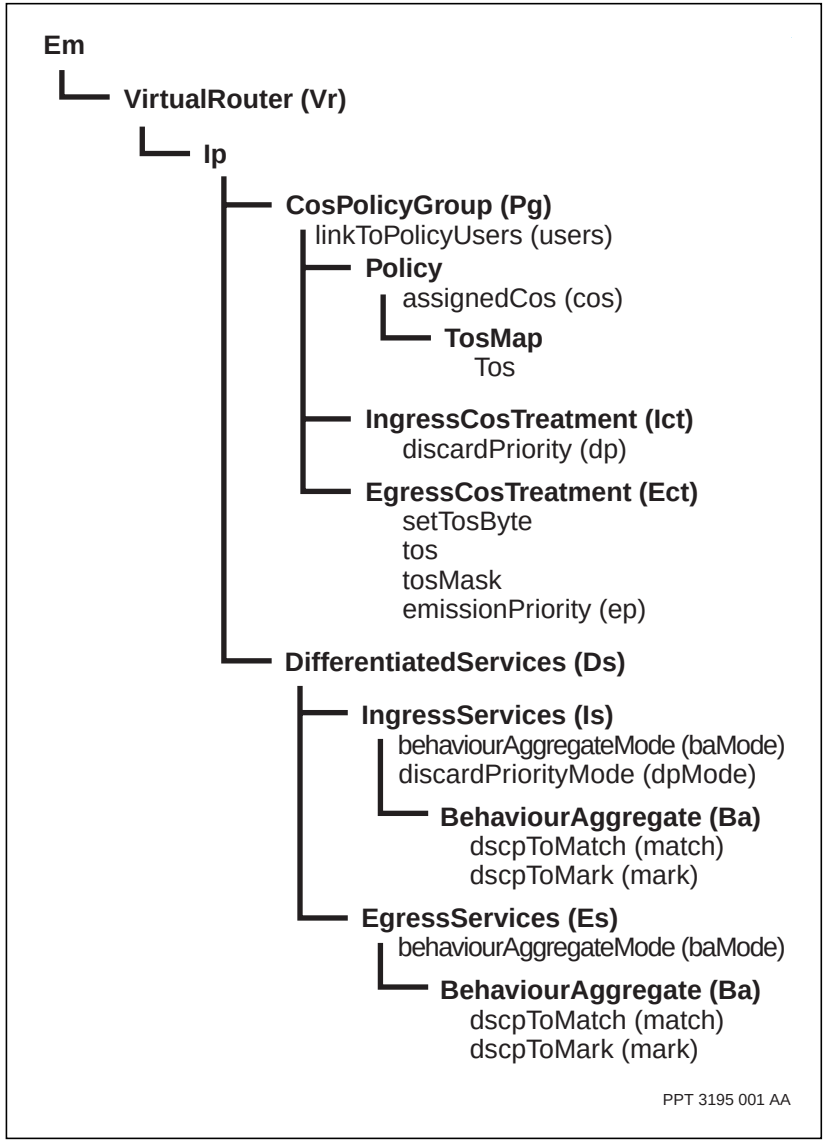
```
set vr/<vr_name> Ip Ds/<ds_inst> Es Ba/<ba_inst>
dscpToMark <mark_value>
```

Variable definitions

Variable	Value
<ba_inst>	The instance of the behavior aggregate
<dp_mode>	The value that sets the discard priority to match the DSCP or retain the discard priority of the link.
<ds_inst>	The instance of the DiffServ component.
<mark_value>	The DSCP value you want to set on the packets identified by the *dscpToMatch* attribute.
<match_list>	A list of one or more DSCP values you want to change to a value specified by the *dscpToMark* attribute.
<treatment>	*Preserve*, *link* or *translate* depending if this is a core, edge or boundary ingress interface.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 89
Preparing IP DiffServ interface profiles for IP CoS migration component hierarchy



Activating IP DiffServ interface profiles for IP CoS migration

Activate IP DiffServ interfaces for IP CoS migration to move the traffic from the IP CoS link to the IP DiffServ interface profiles that have been set up.

Prerequisites

- You must have set up your interfaces according to “Preparing IP DiffServ interface profiles for IP CoS migration” (page 297).
- Migrate all virtual routers in an IP DiffServ domain before proceeding to the virtual routers in another IP DiffServ domain.

Procedure steps

1. Display the IP ports that the CosPolicyGroups are linked to.

   ```
   d Vr/<vr_name> Ip Pg/<policy_name> users
   ```

2. Set the linkToUser for the differentiated services the same as the CosPolicyGroups.

   ```
   set Vr/<vr_name> Ip Ds/<ds_inst> users <user_list>
   ```

3. Clear the linkToUsers attribute for the CosPolicyGroup.

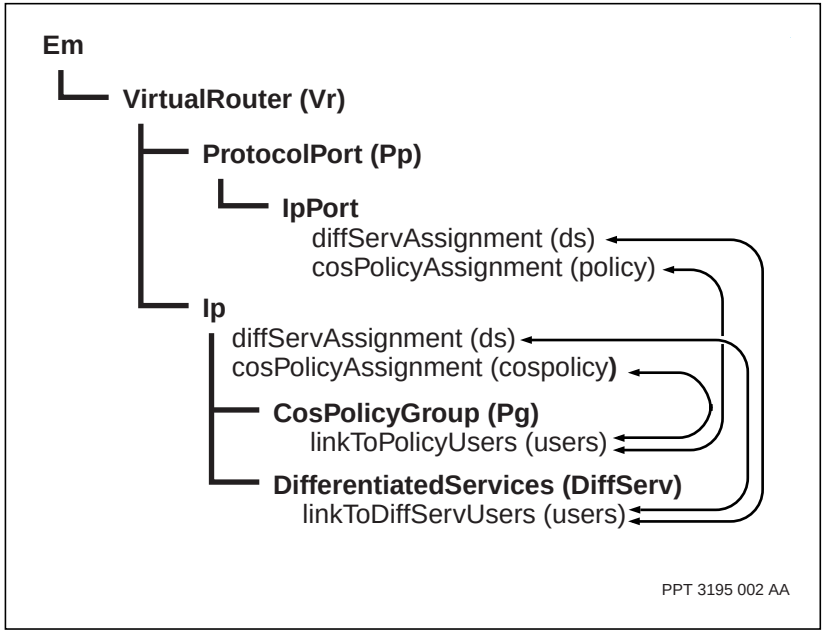
   ```
   set Vr/<vr_name> Ip Pg/<policy_name> users !
   ```

Variable definitions

Variable	Value
<ds_inst>	The instance of the DiffServ component.
<policy_name>	The name of the CoS policy group.
<user_list>	The list of ports that are using the service.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 90
Activating IP DiffServ interface profiles for IP CoS migration component hierarchy



Removing IP CoS

Remove IP CoS to delete any provisioned IP CoS components that have been replaced by DiffServ.

Prerequisites

- You have activated IP DiffServ on all virtual routers that require it according to the procedure “Activating IP DiffServ interface profiles for IP CoS migration” (page 300).
- Migrate all virtual routers in an IP DiffServ domain before proceeding to the virtual routers in another IP DiffServ domain.

Procedure steps

1. Delete all the IP CoS components.

   ```
   del Vr/<vr_name> Ip Pg/<policy_name>
   ```

2. Remove IP CoS from the feature list of all Sw Lpt components.

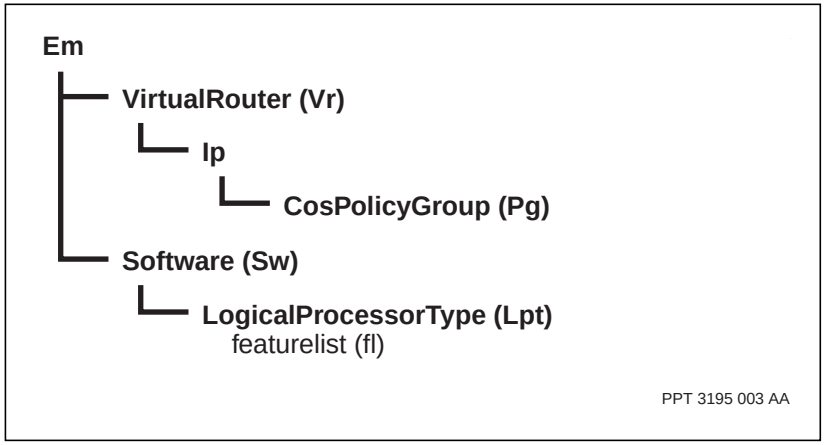
   ```
   set sw Lpt/<lpt_name> fl ~ipCos
   ```

Variable definitions

Variable	Value
<lpt_name>	The name of the logical processor type.
<policy_name>	The name of the CoS policy group.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 91
Removing IP CoS component hierarchy



Chapter 24
Configuring IP flow filters

Configure IP flow filters to create a more secure network by defining which IP packet flows are permitted or denied entry to the network.

Prerequisites

CAUTION

IP flow filters and routing control traffic

A filter configured to permit certain subnets to traverse the network via the port where you apply the filter automatically denies any remaining subnets. This action may deny routing control traffic, which can bring down the routing adjacency in your network.

Consider carefully the effect of any filter you configure on the routing control traffic (e.g., OSPF and BGP parcels) in your network before you assign the filter to the virtual router and IP ports. Always include an *Ip Filter FilterFlow* subcomponent that permits your routing control traffic.

If OSPF is configured to make use of IP Multicast destination addresses, an *IP Filter FilterFlow* subcomponent must be added where the *daPrefix* is set to 224.0.0.4 and the *daPrefixLength* is set to 30. This *FilterFlow* will permit packets destined to 224.0.0.5 (AllSPFRouters) and 224.0.0.6 (AllDRouters).

- Define both the IP CoS and IP flow filters features in a software logical processor type (LPT).

Note: Do not add the IP flow filters feature to the LPT for an SBIC, CQC, or PQC-1 card.

- See the figure "IP configuration work flow" (page 38) to understand how IP flow filters fit into the overall IP configuration task flow.

Procedure steps

1 Add a *Filter* component as a subcomponent of the *Ip* component.

```
add Vr/<vr_name> Ip Filter/<filter_name>
```

2 Add a *FilterFlow* subcomponent to the *Filter* subcomponent.

```
add Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number>
```

3 Set the *action* attribute. If the filter flow action is permit enter

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> action permit
```

If the filter flow action is deny enter

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> action deny
```

4 Set the *saPrefix* attribute of the *FilterFlow* subcomponent.

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> saPrefix <saPrefixAddress>
```

5 Set the *saPrefixLength* attribute of the *FilterFlow* subcomponent.

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> saPrefixLength <saPrefixLength>
```

6 Set the *daPrefix* attribute of the *FilterFlow* subcomponent.

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> daPrefix <daPrefixAddress>
```

7 Set the *daPrefixLength* attribute of the *FilterFlow* subcomponent.

```
set Vr/<vr_name> Ip Filter/<filter_name> FilterFlow/
<filterflow_number> daPrefixLength <daPrefixLength>
```

8 Assign the filter to a protocol port or a virtual router.

To assign the filter to a protocol port enter

set Vr/<vr_name> Pp/<pp_name> IpPort filterAssignment Vr/<vr_name> Filter/<filter_name>

To assign the filter to a virtual router enter

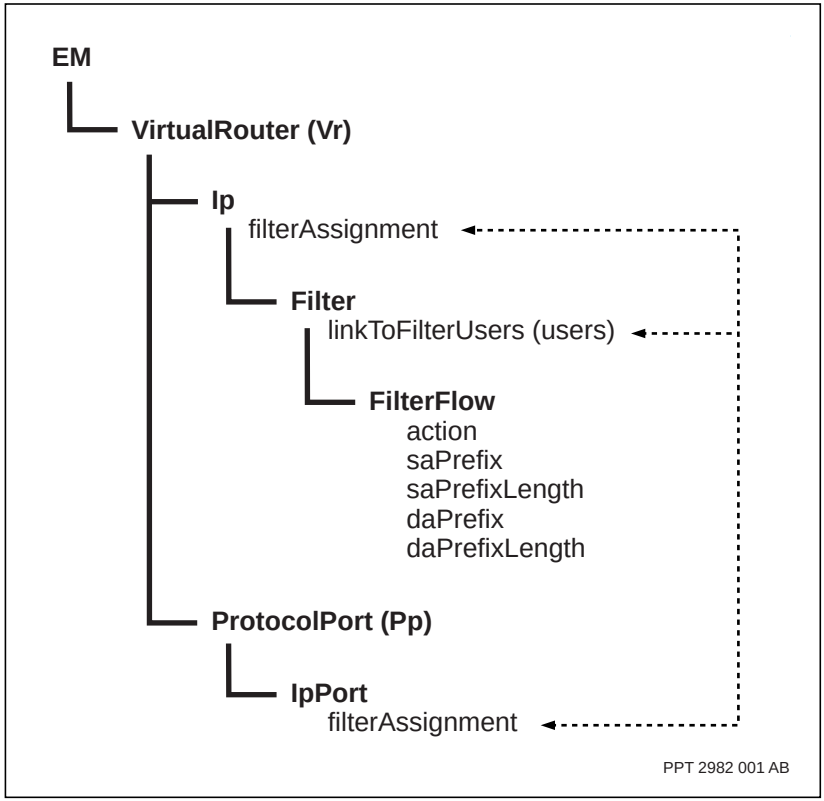
set Vr/<vr_name> Ip filterAssignment Vr/<vr_name> Ip Filter/<filter_name>

Variable definitions

Variable	Value
<daPrefixAddress>	The IP destination address prefix for the flow.
<daPrefixLength>	The number of most significant bits of an IP destination address that you want matched with the IP destination address prefix
<filter_name>	The name of the filter.
<filterflow_number>	The number of the flow filter.
<pp_name>	The name of the protocol port.
<saPrefixAddress>	The IP source address prefix for the flow.
<saPrefixLength>	The number of most significant bits of an IP source address that you want matched with the IP source address prefix.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 92
Configuring IP flow filters component hierarchy



Chapter 25
Configuring point-to-point tunnels

Configure a point-to-point IP tunnel instance to connect two physically separate networks that share the same address space through an IP network with a different address space.

Prerequisites

- "Configuring and linking a protocol port to a media interface" (page 122)
- "Enabling IP on a protocol port" (page 124)
- Configure a RIP or OSPF interface so that the tunnel endpoints can receive information about remote subnetworks. See
  - "Configuring a routing information protocol (RIP) interface" (page 132)
  - "Configuring an OSPF interface" (page 155)
- See the figure "IP configuration work flow" (page 38) to understand how IP tunneling fits into the overall IP configuration task flow.

Procedure steps

1. Create an IP tunnel instance on the virtual router.

   ```
   add Vr/<vr_name> Ip Tunnel
   ```

2. Configure the end point for a point-to-point IP tunnel.

   ```
   add Vr/<vr_name> Ip Tunnel Sep/<sep_id>
   ```

3. Specify the source address of the IP tunnel.

   ```
   set Vr/<vr_name> Ip Tunnel Sep/<sep_id> src <src_addr>
   ```

4 Specify the tunnel destination address.

```
set Vr/<vr_name> Ip Tunnel Sep/<sep_id> dst <dst_addr>
```

5 Specify the tunnel encapsulation type.

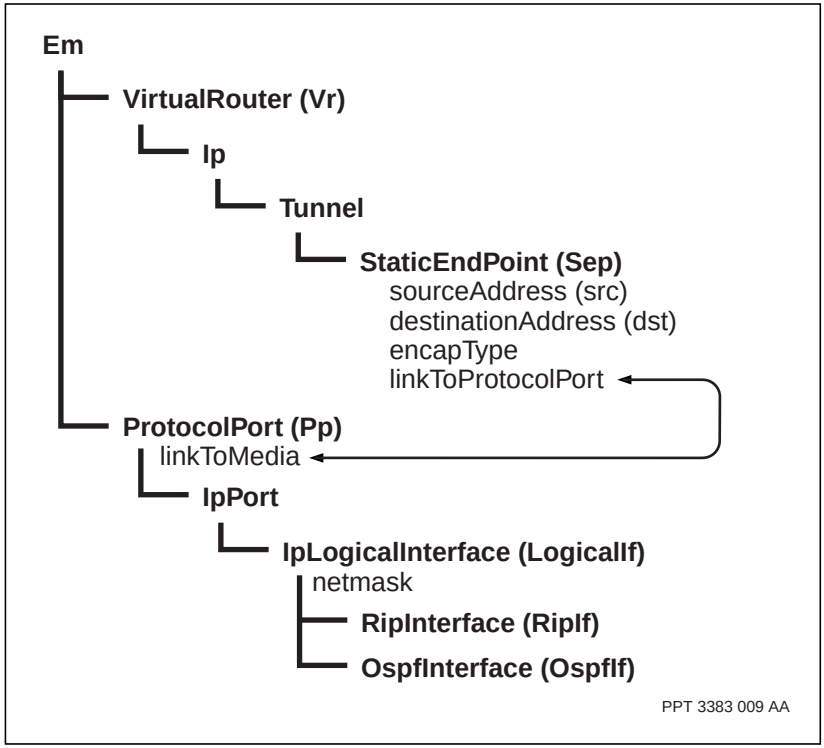
```
set Vr/<vr_name> Ip Tunnel Sep/<sep_id> encapType
<type>
```

Variable definitions

Variable	Value
<dst_addr>	The IP address at the destination end of the tunnel.
<sep_id>	The instance number of the tunnel end point.
<src_addr>	The IP address at the source end of the tunnel.
<type>	The encapsulation type, either IP in IP or GRE.
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 93
Configuring point-to-point tunnels component hierarchy



Chapter 26
IP monitoring and testing

This section contains information about the following:

- “Monitoring the ATM MPE configuration” (page 314)
- “Clearing or optimizing an ATM MPE soft PVC” (page 319)
- “Monitoring the frame relay DTE configuration” (page 321)
- “Monitoring the PPP configuration” (page 327)
- “Monitoring the IP and virtual router configuration” (page 331)
- “Monitoring the IP routing management configuration” (page 342)
- “Monitoring the virtual media configuration” (page 345)
- “Monitoring the RIP configuration” (page 347)
- “Monitoring the OSPF configuration” (page 349)
- “Monitoring the BGP-4 configuration” (page 353)
- “Monitoring the static route configuration” (page 355)
- “Monitoring the IP multicast configuration” (page 356)
- “Monitoring the virtual router redundancy protocol configuration” (page 358)
- “Monitoring the IP CoS configuration” (page 360)
- “Monitoring IP DiffServ configuration” (page 362)
- “Monitoring the IP flow filters configuration” (page 373)
- “Monitoring the IP tunnel configuration” (page 375)

Monitoring the ATM MPE configuration

This section contains the information you need to monitor and maintain the IP over ATM MPE configuration. Issue all commands in operational mode. See “Operational mode” (page 30).

For information on specific components and protocols, see the following sections:

- “ATM MPE component states” (page 314)
- “ATM MPE soft PVC component states” (page 315)
- “Monitoring the AtmMpe component” (page 315)
- “Monitoring the AtmConnection subcomponent” (page 316)
- “Monitoring the IlsFwdr component” (page 317)
- “Testing ATM MPE soft PVC connectivity” (page 317)
- “Testing ATM MPE soft PVC data flow” (page 318)

ATM MPE component states

The table “ATM MPE component states” (page 314) lists the operational states reported by the ATM MPE service.

Table 7
ATM MPE component states

Condition	States reported
ifAdminStatus is provisioned as *down* or *testing*	operational: enabled usage: idle snmpOperStatus: down or testing
ifAdminStatus is provisioned as *up* and the component is locked	operational: enabled usage: idle snmpOperStatus: down
ifAdminStatus is provisioned as *up*, the component is unlocked	operational: enabled usage: active snmpOperStatus: up

ATM MPE soft PVC component states

ATM connection status is independent of the IP datapath status. If the *ifOperStatus* of the *AtmMpe Ac* is up, the soft PVC connection is active and the IP datapath is enabled. If the *atmMpe Ac ifOperStatus* is down, the soft PVC connection may or may not be active, but the IP datapath is disabled. If you lock the *AtmMpe* or *AtmMpe Ac* component, the soft PVC will not be torn down, but the IP forwarding will be disabled for the affected connections.

The table "AtmConnection component states" (page 315) shows the status of IP forwarding for the ATM MPE media in relation to the soft PVC and *AtmMpe Ac* component status.

Table 8
AtmConnection component states

Soft PVC state (AtmMpe Ac SrcPvc/DstPvc)	AtmMpe Ac		IP forwarding status
	OSI admin state	operationalState	
inactive	locked	disabled	disabled
inactive	unlocked	disabled	disabled
active	locked	disabled	disabled
active	unlocked	enabled	enabled

IP forwarding is not enabled immediately after the soft PVC becomes active, but the active state triggers IP forwarding to become enabled. The *ifOperStatus* attribute becomes *up* only after IP forwarding is enabled.

Monitoring the AtmMpe component

The following section describes how to display configuration information and operational statistics for the *AtmMpe* component.

Table 9
Monitoring the AtmMpe component

Action	Command	Legend
List all the AtmMpe instances on a node	**list AtmMpe/***	
Display the status of an AtmMpe instance	**display AtmMpe/<n>**	<n> is the number of the AtmMpe instance
Display the attributes configured under the AtmMpe component	**display -p AtmMpe/<n>**	
Display the status and attributes of a specific AtmMpe component instance using the ifTable	**display Vr/<a> Ift/<b>**	<a> is the number of the virtual router <b> is the number of the ifTable
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the AtmConnection subcomponent

The following section describes how to display configuration information and operational statistics for the *AtmMpe AtmConnection* subcomponent.

Note: If you lock the *atmConnection* component at one end of an ATM MPE VCC, IP traffic destined for that VCC from the remote end is discarded. The remote VR does not receive notification that the VCC is out of service, and continues to transmit traffic on the VCC even though it is not operational.

Table 10
Monitoring the AtmConnection subcomponent

Action	Command	Legend
List all the AtmConnection instances under the AtmMpe component	**list AtmMpe/<n> AtmConn/***	<n> is the number of the AtmMpe instance
Display the status of an AtmConnection instance	**display AtmMpe/<n> AtmConn/<m>**	<n> is the number of the AtmMpe instance <m> is the number of the AtmConn instance
Display the attributes configured under the AtmMpe component	**display -p AtmMpe/<n> AtmConn/<m>**	
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the IlsFwdr component

The following section describes how to display configuration information and operational statistics for the *Lp IlsFwdr* component.

Table 11
Monitoring the IlsFwdr component

Action	Command	Legend
Display the status of an IlsFwdr component	**display Lp/<n> IlsFwdr/<m>**	<n> is the number of the Lp <m> is the number of the IlsFwdr component
Display the attributes configured under the IlsFwdr component	**display -p Lp/<n> IlsFwdr/<m>**	
Note: -p enables you to list provisionable components while in operational mode		

Testing ATM MPE soft PVC connectivity

The following section describes how to verify connectivity in an ATM MPE soft PVC. Check the state attribute of the *SrcPvc* and *DstPvc* components at either end of the soft PVC. If a soft PVC has been established, both endpoints

are in the active state. If the soft PVC has failed, both of the endpoints are in the inactive state. In the case of failure, you can also determine the cause of the connection teardown and check the number of subsequent setup attempts.

Table 12
Monitoring AtmMpe soft PVC connectivity

Action	Command	Legend
Display the status of the calling endpoint of this soft PVC.	**display AtmMpe/<n> Ac/<m> SrcPvc state**	<n> is the number of the AtmMpe instance <m> is the number of the AtmCon instance
Display the status of the called endpoint of this soft PVC.	**display AtmMpe/<n> Ac/<m> DstPvc state**	
Display the reason for the last teardown of the connection at the calling endpoint of this soft PVC.	**display AtmMpe/<n> Ac/<m> SrcPvc lastClearCause**	
Display the reason for the last teardown of the connection at the called endpoint of this soft PVC.	**display AtmMpe/<n> Ac/<m> DstPvc lastClearCause**	
Display the number of times the calling endpoint has attempted to reestablish the connection.	**display AtmMpe/<n> Ac/<m> SrcPvc retryCount**	

Testing ATM MPE soft PVC data flow

The following section describes how to test the flow of data through a soft PVC by monitoring the *AtmMpe* and *AtmIf* component statistics.

Table 13
Monitoring AtmMpe soft PVC data flow

Action	Command	Legend
Display the outPackets and outOctets attributes at the calling endpoint of the soft PVC.	**display AtmMpe/<n> Ac/<m>**	<n> is the number of the AtmMpe instance <m> is the number of the AtmCon instance
Display the inPackets and inOctets attributes at the called endpoint of the soft PVC.	**display AtmMpe/<n> Ac/<m>**	
Display the txCell count at each AtmIf component in the path of the soft PVC.	**display AtmIf/<i> Vcc/<vc> Statistics**	<i> is the number of the AtmIf instance <vc> is the instance value of the VCC
Display the rxCell count at each AtmIf component in the path of the soft PVC.	**display AtmIf/<i> Vcc/<vc> Statistics**	

Clearing or optimizing an ATM MPE soft PVC

The following section describes how to tear down an active ATM MPE soft PVC connection using the clear command. The clear command also resets the *retryCount* attribute of the calling endpoint.

One reason for clearing an active connection is to optimize the path of the soft PVC. If a path better than the current one becomes available, the soft PVC does not take it automatically. You can clear the soft PVC to force it to take a new path. When the soft PVC attempts to reestablish the connection, the PNNI signaling network will find the better path.

Table 14
Clearing the AtmMpe soft PVC

Action	Command	Legend
Clear the ATM MPE soft PVC at the calling end of the connection.	**clear AtmMpe/<n> Ac/<m> SrcPvc**	<n> is the number of the AtmMpe instance <m> is the number of the AtmCon instance
Clear the ATM MPE soft PVC at the called end of the connection.	**clear AtmMpe/<n> Ac/<m> DstPvc**	

Monitoring the frame relay DTE configuration

This section contains the information you need to monitor and maintain the IP over frame relay DTE configuration. Issue all commands in operational mode. See “Operational mode” (page 30).

For information on specific components and protocols, see the following sections:

- “Frame relay DTE component states” (page 321)
- “Frame relay DTE remote group (Rg) component states” (page 322)
- “Frame relay DTE data link connection identifier (Dlci) component states” (page 323)
- “Monitoring the FrDte component” (page 324)
- “Monitoring the StDlci subcomponent” (page 325)
- “Monitoring the Rg subcomponent” (page 325)

Frame relay DTE component states

The table “ATM MPE component states” (page 314) lists the operational states reported by the frame relay DTE service.

Table 15
Frame relay DTE component states

Condition	States reported
The *FrDte* component is waiting for frame relay interface or the hardware interface to become available.	operational: disabled usage: idle administrative: unlocked
The *FrDte* component is ready to provide service, but currently there is no DLCI defined on the interface.	operationalState: enabled usageState: idle administrative: unlocked
A lock command is in effect. The relationship between this component and the frame relay interface or LP does not exist	operational: disabled usage: idle administrative: locked

(Sheet 1 of 2)

Table 15 (continued)
Frame relay DTE component states

Condition	States reported
The component is in use, i.e. there is at least one active DLCI defined on the interface.	operational: enabled usage: active administrative: unlocked
No new DLCI is available, i.e. total number of DLCIs in use >=1024.	operational: enabled usage: busy administrative: unlocked

(Sheet 2 of 2)

Frame relay DTE remote group (Rg) component states

The table “Frame relay DTE remote group (Rg) component states” (page 322) lists the operational states reported by the frame relay DTE remote group (Rg) component.

Table 16
Frame relay DTE remote group (Rg) component states

Condition	States reported
The component is disabled because the corresponding FrDte is locked	operational: disabled usage: idle administrative: unlocked
The remote group is ready to provide service, but currently there is no DLCI linked to it.	operational: enabled usage: idle administrative: unlocked
This remote group is in use, i.e. there is at least one DLCI in use linked to this remote group.	operational: enabled usage: active administrative: unlocked
No new DLCI is available, i.e. total number of DLCIs in use >= 1024.	operational: enabled usage: busy administrative: unlocked

(Sheet 1 of 2)

Table 16 (continued)
Frame relay DTE remote group (Rg) component states

Condition	States reported
The lock command is in effect and the linked protocol port is disabled or the corresponding FrDte is locked.	operational: disabled usage: idle administrative: locked
The lock command is in effect but the linked protocol port is enabled or the corresponding FrDte is unlocked.	operational: enabled usage: idle administrative: locked

(Sheet 2 of 2)

Frame relay DTE data link connection identifier (Dlci) component states

The table "Frame relay DTE Dlci component states" (page 323) lists the operational states reported by the frame relay DTE data link connection identifier (*Dlci*) component.

Table 17
Frame relay DTE Dlci component states

Condition	States reported
The DLCI is disabled because the corresponding remote group is locked or disabled, or the link is down.	operational: disabled usage: idle administrative: unlocked
This DLCI is ready to provide service.	operationalState: enabled usageState: active administrative: unlocked

(Sheet 1 of 2)

Table 17 (continued)
Frame relay DTE Dlci component states

Condition	States reported
The lock command is in effect. Also the DLCI is disabled because the corresponding remote group is locked or disabled, or the link is down.	operational: disabled usage: idle administrative: locked
The lock command is in effect on the corresponding remote group.	operational: enabled usage: idle administrative: locked
(Sheet 2 of 2)	

Monitoring the FrDte component

The following section describes how to display configuration information and operational statistics for the *FrDte* component.

Table 18
Monitoring the FrDte component

Action	Command	Legend
List all the FrDte instances on a node	**list FrDte/***	
Display the status of an FrDte instance	**display FrDte/<n>**	<n> is the number of the FrDte instance
Display the attributes configured under the FrDte component	**display -p FrDte/<n>**	
Display the status and attributes of a specific FrDte component instance using the ifTable	**display Vr/<a> Ift/<b>**	<a> is the number of the virtual router <b> is the number of the ifTable
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the StDlci subcomponent

The following section describes how to display configuration information and operational statistics for the *FrDte StDlci* subcomponent.

Table 19
Monitoring the StDlci subcomponent

Action	Command	Legend
List all the StDlci instances under the FrDte component	**list FrDte/<n> StDlci/***	<n> is the number of the FrDte instance
Display the status of an StDlci instance	**display FrDte/<n> StDlci/<m>**	<n> is the number of the FrDte instance <m> is the number of the StDlci instance
Display the attributes configured under the StDlci subcomponent	**display -p FrDte/<n> StDlci/<m>**	
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the Rg subcomponent

The following section describes how to display configuration information and operational statistics for the *FrDte Rg* subcomponent.

Table 20
Monitoring the Rg subcomponent

Action	Command	Legend
List all the Rg instances under the FrDte component	**list FrDte/<n> Rg/***	<n> is the number of the FrDte instance
Display the status of an Rg instance	**display FrDte/<n> Rg/<m>**	<n> is the number of the FrDte instance <m> is the number of the Rg instance

(Sheet 1 of 2)

Table 20 (continued)
Monitoring the Rg subcomponent

Action	Command	Legend
Display the attributes configured under the Rg subcomponent	**display -p FrDte/<n> Rg/<m>**	
Note: -p enables you to list provisionable components while in operational mode		
(Sheet 2 of 2)		

Monitoring the PPP configuration

This section contains the information you need to monitor and maintain the IP over PPP configuration. Issue all commands in operational mode. See “Operational mode” (page 30).

For information on specific components and protocols, see the following sections:

- “PPP component states” (page 327)
- “Monitoring the Ppp component” (page 328)
- “Monitoring the Link subcomponent” (page 328)
- “Monitoring the Lqm subcomponent” (page 329)
- “Monitoring the Leq subcomponent” (page 330)

PPP component states

The table “ATM MPE component states” (page 314) lists the operational states reported by the PPP service.

Table 21
PPP component states

Condition	States reported
A lock -force operator command is in effect.	operational: disabled usage: idle administrative: locked
The component is unlocked. PPP is not in the Enabled operational state.	operationalState: disabled usageState: idle administrative: unlocked
The component is unlocked and in service. PPP is in the Enabled operational state.	operational: enabled usage: busy administrative: unlocked
PPP is going from the Unlocked state to the Locked state. It is in the process of an orderly shutdown.	operational: enabled usage: busy administrative: shutting down

Monitoring the Ppp component

The following section describes how to display configuration information and operational statistics for the *Ppp* component.

Table 22
Monitoring the Ppp component

Action	Command	Legend
List all the ppp instances on a node	**`list Ppp/*`**	
Display the status of a ppp instance	**`display Ppp/<n>`**	<n> is the number of the Ppp instance
Display the attributes configured under the ppp component	**`display -p Ppp/<n>`**	
Display the status and attributes of a specific ppp component instance using the ifTable	**`display Vr/<a> Ift/<b>`**	<a> is the number of the virtual router <b> is the number of the ifTable
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the Link subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Link* subcomponent.

Table 23
Monitoring the Link subcomponent

Action	Command	Legend
List all the link instances under the ppp component	**list Ppp/<n> Link/***	<n> is the number of the Ppp instance
Display the status of a link instance	**display Ppp/<n> Link/<m>**	<n> is the number of the Ppp instance <m> is the number of the Link instance
Display the attributes configured under the link subcomponent	**display -p Ppp/<n> Link/<m>**	
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the Lqm subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Lqm* subcomponent.

Table 24
Monitoring the Lqm subcomponent

Action	Command	Legend
List all the Lqm instances under the Ppp component	**list Ppp/<n> Lqm/***	<n> is the number of the Ppp instance
Display the status of an Lqm instance	**display Ppp/<n> Lqm/<m>**	<n> is the number of the Ppp instance <m> is the number of the Lqm instance
Display the attributes configured under the Lqm subcomponent	**display -p Ppp/<n> Lqm/<m>**	
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the Leq subcomponent

The following section describes how to display configuration information and operational statistics for the *Ppp Leq* subcomponent.

The *Ppp Leq* subcomponent is only available on a Passport 7400 with SBIC-based FPs.

Table 25
Monitoring the Leq subcomponent

Action	Command	Legend
List all the Leq instances under the Ppp component	**list Ppp/<n> Leq/***	<n> is the number of the Ppp instance
Display the status of an Leq instance	**display Ppp/<n> Leq/<m>**	<n> is the number of the Ppp instance <m> is the number of the Leq instance
Display the attributes configured under the Leq subcomponent	**display -p Ppp/<n> Leq/ <m>**	
Note: -p enables you to list provisionable components while in operational mode		

Monitoring the IP and virtual router configuration

You can perform the tests described in this section to determine whether you have configured IP and virtual routers properly. Issue all commands in operational mode. See “Operational mode” (page 30). For more information, see the following sections:

- “Monitoring the IP component” (page 331)
- “Monitoring the IpInterfaceEntry subcomponent” (page 333)
- “Monitoring the IP cache subcomponent” (page 333)
- “Monitoring the ICMP subcomponent” (page 334)
- “Testing connectivity using the ICMP subcomponent” (page 335)
- “Monitoring the TCP subcomponent” (page 337)
- “Monitoring the UDP subcomponent” (page 338)
- “Monitoring the IpPort component” (page 338)
- “Monitoring the Arp subcomponent” (page 339)
- “Monitoring the BootpPort component” (page 340)
- “Monitoring the RelayBroadCast subcomponent” (page 341)

Monitoring the IP component

The following section describes how to lock and unlock components, display configuration information and operational statistics, and test connectivity for the *Ip* component and its subcomponents.

For more information, see the following tables:

- “Locking and unlocking the IP component” (page 332)
- “Monitoring the IP component” (page 332)

CAUTION

Provisioning session termination

The Ip component can be locked. However, this disables access to IP and all of its subcomponents. It also makes it impossible to further provision using IP applications such as telnet, SNMP, or FTP. IP datagrams are not forwarded while the Ip component is locked. Locking of IP and its related protocols is immediate when you issue the lock command.

Table 26
Locking and unlocking the IP component

Action	Command	Legend
Lock the Ip component. Locking takes the IP component and all of its configured or dynamic subcomponents out of service.	**lock Vr/<vr_name> Ip**	<vr_name> is the name of the virtual router
Unlock the Ip component. Unlocking returns the Ip component and all of its configured or dynamic subcomponents to service.	**unlock Vr/<vr_name> Ip**	

Table 27
Monitoring the IP component

Action	Command	Legend
List the configured components of the IP service.	**list -p Vr/<vr_name> Ip**	<vr_name> is the name of the virtual router
Display the configured components of the IP service.	**display -p Vr/<vr_name> Ip**	
(Sheet 1 of 2)		

Table 27 (continued)
Monitoring the IP component

Action	Command	Legend
Display operational statistics for an IP interface	**display Vr/<vr_name> Ip**	
Note: -p enables you to list provisionable components while in operational mode		
(Sheet 2 of 2)		

Monitoring the IpInterfaceEntry subcomponent

The following section describes how to display operational statistics for the *IpInterfaceEntry* (*If*) subcomponent. For more information, see the table "Monitoring the IpInterfaceEntry subcomponent" (page 333).

Table 28
Monitoring the IpInterfaceEntry subcomponent

Action	Command	Legend
Display operational statistics of the IpInterfaceEntry (if) subcomponent	**display Vr/<vr_name> Ip IpInterfaceEntry/ <Ipaddr>**	<vr_name> is the name of the virtual router <Ipaddr> is the 32-bit IP address assigned to the IP interface
Note: The *hardwareAddress* attribute displays the MAC address in canonical format. The *ncHardwareAddress* attribute displays the MAC address in non-canonical format.		

Monitoring the IP cache subcomponent

The *Cache* component is a dynamic subcomponent of the *Ip* component. It represents the IP cache table on an LP. You can list *Cache* component instances and display operational statistics.

For more information, see the following tables:

- "Monitoring the IP cache subcomponent" (page 334)
- "Clearing IP cache table entries" (page 334)

Table 29
Monitoring the IP cache subcomponent

Action	Command	Legend
List all *Cache* component instances	**list Vr/<vr_name> Ip Cache/***	<vr_name> is the name of the virtual router
Display operational statistics for a particular *Cache* component instance	**display Vr/<vr_name> Ip Cache/<lp_id>**	

Table 30
Clearing IP cache table entries

Action	Command	Legend
Clear all entries from the IP cache table	**clear vr/<vr_name> Ip Cache/<lp_id>**	<lp_id> is the instance value assigned in the IP subcomponent to a particular LP

Monitoring the ICMP subcomponent

Icmp is a subcomponent of the *Ip* component responsible for processing the internet control message protocol (ICMP). ICMP sends control messages to source hosts to indicate special conditions. For more information, see the table “Monitoring the ICMP subcomponent” (page 335).

You can use the ping (packet internet groper) command to test connections to other IP nodes. The ping command issues an echo request to the specified address and then compares the response to the request. For more information see “Testing connectivity using the ICMP subcomponent” (page 335).

Table 31
Monitoring the ICMP subcomponent

Action	Command	Legend
Display the ICMP provisionable attributes	**display -p Vr/<vr_name> Ip Icmp**	<vr_name> is the name of the virtual router
Display the ICMP subcomponent statistics	**display Vr/<vr_name> Ip Icmp**	

Testing connectivity using the ICMP subcomponent

The *Icmp* component uses the *ping* verb to test connectivity. *Ping* sends an echo request to the specified IP node, compares the reply with the request, and sends a message to the console indicating the result. ICMP packets can range from 64 to 5000 bytes in length.

> ***Note:*** If you are transmitting ICMP packets over an ATM VCC with a CBR service category, the default transmission queue size for the VCC limits the packet size to 3680 bytes. To ensure that ICMP packets are not discarded, reduce the maximum IP packet size to 3600 bytes on ATM VCCs with a service category of CBR. ICMP packets are not discarded if the VCC has a service category of UBR.

The retry option specifies the number of times to ping a specific IP address until an echo packet is returned. If after x pings no echo packet is returned, ping reports that the node is not responding.

The continuous option allows the operator to continuously ping an IP address or a range of IP addresses.

The netmask option, also called performing a ping sweep, allows the operator to ping a range of addresses specified by the -Ipaddr and -netmask options.

The table “Testing connectivity using the ICMP subcomponent” (page 336) specifies the commands you can use to test connectivity.

Table 32
Testing connectivity using the ICMP subcomponent

Action	Command	Legend
View commands related to the ping verb for the Icmp component	**help -v(ping) vr ip icmp**	
Ping an IP interface	**ping -Ipaddr(<IpAddress>) -size(<pingSize>) Vr/<vr_name> Ip Icmp**	<IpAddress> is the 32-bit address of the interface to be pinged <pingSize> is the *Icmp* packet payload (64 to 5000 bytes) carried in the echo request. If you do not choose this option, the default ping size is 64 bytes. <vr_name> is the name of the virtual router
Execute IP traceRoute on a Passport switch	**ping -Ipaddr(<IpAddress>) -size(<pingSize>) -traceRoute Vr/<vr_name> Ip Icmp**	-traceRoute is the option to execute IP trace route
Set the retry option	**ping -Ipaddr(<IpAddress>) -size(<pingSize>) -retry(<retry_number>) Vr/<vr_name> Ip Icmp**	<retry_number> is the number of times to ping the switch until it returns an echo packet. The range is 0-9.
Ping a specific address	**ping -Ipaddr(<IpAddress>) -continuous Vr/<vr_name> Ip Icmp**	-netmask is the network mask used with the IP address
Ping a range of addresses	**ping -Ipaddr(<IpAddress>) -continuous -netmask(<IpAddress>) Vr/<vr_name> Ip Icmp**	

(Sheet 1 of 2)

Table 32 (continued)
Testing connectivity using the ICMP subcomponent

Action	Command	Legend
Set the netmask option	**ping -Ipaddr(<IpAddress>) -netmask(<IpAddress>) Vr/<vr_name> Ip Icmp**	
Stop a continuous ping, or a ping sweep (the netmask option)	**ping -stop Vr/<vr_name> Ip Icmp**	
Note: You can also stop a continuous ping or a ping sweep by typing Ctrl-C.		
Support ping through certain local IP interfaces	**ping -src(<IpAddress>) Vr<vr_name> Ip Icmp**	
(Sheet 2 of 2)		

Monitoring the TCP subcomponent

The following tables describe how to display TCP provisioning, statistics, and connections:

- “Monitoring the TCP subcomponent” (page 337)
- “Displaying the TCP connection table” (page 337)

Table 33
Monitoring the TCP subcomponent

Action	Command	Legend
Display *Tcp* subcomponent statistics	**display Vr/<vr_name> Ip Tcp**	

Table 34
Displaying the TCP connection table

Action	Command	Legend
Display the TCP connection table	**display Vr/<vr_name> Ip Tcp Tcpentry/***	<vr_name> is the name of the virtual router

Monitoring the UDP subcomponent

The following tables describe how to display user datagram protocol (UDP) statistics and listen tables:

- “Monitoring the UDP subcomponent” (page 338)
- “Displaying the UDP listen table” (page 338)

Table 35
Monitoring the UDP subcomponent

Action	Command	Legend
Display *Udp* subcomponent statistics	**display Vr/<vr_name> Ip Udp**	<vr_name> is the name of the virtual router

Table 36
Displaying the UDP listen table

Action	Command	Legend
Display the UDP listen table	**display Vr/<vr_name> Ip Udp listen/***	<vr_name> is the name of the virtual router

Monitoring the IpPort component

The following tables describe how to lock, unlock, and display provisioning and operational statistics for the *IpPort* component:

- “Locking and unlocking the IpPort component” (page 339)
- “Monitoring the IpPort component” (page 339)

CAUTION
Locking IP port stops IP routing
The IpPort component can be locked. However, this stops IP routing on the locked port. IP datagrams are not forwarded through that port while the IpPort component is locked. Locking of the IpPort component is immediate when you issue the lock command.

Table 37
Locking and unlocking the IpPort component

Action	Command	Legend
Lock the IP port	**lock Vr/<vr_name> ProtocolPort/<pp_name> IpPort**	<vr_name> is the name of the virtual router
Unlock the IP port	**unlock Vr/<vr_name> ProtocolPort/<pp_name> IpPort**	

Table 38
Monitoring the IpPort component

Action	Command	Legend
Display IpPort LogicalIf addresses	**display Vr/<vr_name> ProtocolPort/ <pp_name> IpPort LogicalIf/***	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display the *IpPort* component information	**display Vr/<vr_name> Pp/<pp_name> IpPort**	

Monitoring the Arp subcomponent

The following tables describe how to monitor and clear ARP entries:

- “Monitoring the Arp subcomponent” (page 340)
- “Clearing ARP table dynamic host entries” (page 340)

Table 39
Monitoring the Arp subcomponent

Action	Command	Legend
Display the ARP static host entries in canonical form.	**display Vr/<vr_name> Ip Arp HostEntry/***	<vr_name> is the name of the virtual router
Display the ARP static host entries in non-canonical form.	**display -noTabular Vr/ <vr_name> Ip Arp HostEntry/***	
Display the ARP dynamic host entries in canonical form.	**display Vr/<vr_name> Ip Arp DynHostEntry/***	
Display the ARP dynamic host entries in non-canonical form.	**display -noTabular Vr/ <vr_name> Ip Arp DynHostEntry/***	
Note: The *physAddress* attribute displays the MAC address in canonical form. To view the MAC address in non-canonical form, use the -noTabular option with the display command.		

Table 40
Clearing ARP table dynamic host entries

Action	Command	Legend
Clear all *DynamicHostEntries* in the ARP table.	**clear vr/<vr_name> ip arp dyn/***	<vr_name> is the name of the virtual router
Clear the *DynamicHostEntries* learned from a specific Ip port that match the specified IP address.	**clear vr/<vr_name> ip arp dyn/<Ip_address>**	
Clear all *DynamicHostEntries* learned from a specific Ip port.	**clear -log(<Ip_address>) vr/ <vr_name> ip arp**	

Monitoring the BootpPort component

The *BootpPort* component contains information about the provisioned BOOTP ports under the *IpPort* component. The table “Monitoring the BootpPort component” (page 341) describes how to display BOOTP port statistics.

Table 41
Monitoring the BootpPort component

Action	Command	Legend
Display BOOTP port statistics using the *BootpPort* component	**display Vr/<vr_name> pp/<protocolPort_name> IpPort BootpP**	<vr_name> is the name of the virtual router

Monitoring the RelayBroadCast subcomponent

The table "Monitoring the RelayBroadCast subcomponent" (page 341) describes how to display RelayBroadCast statistics.

Table 42
Monitoring the RelayBroadCast subcomponent

Action	Command	Legend
Display the *RelayBroadCast* subcomponent global broadcast counter	**display Vr/<vr_name> Ip RelayBC**	<vr_name> is the name of the virtual router
Display the *RelayBroadCast* subcomponent counters on individual interfaces	**display Vr/<vr_name> ProtocolPort/<pp_name> IpPort**	<pp_name> is the name of the protocol port

Monitoring the IP routing management configuration

Passport stores routing information in the IP forwarding table and the routing database. The forwarding table provides information on the routes with the best metric. The routing database provides all sources of routing information. This section describes how to monitor the IP forwarding table and the routing database.

For more information, see the following tables:

- "Monitoring the Ip ForwardTable component" (page 342)
- "Monitoring the Ip RouteDataBaseEntry component" (page 344)

Table 43
Monitoring the Ip ForwardTable component

Action	Command	Legend
Display the forwarding table	**display Vr/<vr_name> Ip Fwd/*,*,*,***	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router
Display the routes in the forwarding table covered by a particular IP address or subnet	**display Vr/<vr_name> Ip Fwd/ <a>.<b>.<c>.<d>,*,*,***	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <a>.<b>.<c>.<d> are the octets in an IP address. They can be numerals or, in the case of a subnet, the wildcard (*). For example, 47.138.*.*

(Sheet 1 of 2)

Table 43 (continued)
Monitoring the Ip ForwardTable component

Action	Command	Legend
Display the routes in the forwarding table for a particular routing protocol.	**display Vr/<vr_name> Ip Fwd/ *(protocol=<pro_name>)**	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <pro_name> is the name of the routing protocol. For example, ospf.
Display the routes in the forwarding table covered by a particular IP address or subnet for a particular routing protocol	**display Vr/<vr_name> Ip Fwd/ <a>[.<b>.<c>.<d>],*,*,* (protocol=<pro_name>)**	<vr_name> is the wildcard (*), to display the routes for all virtual routers or the name of a virtual router, to display the routes for only that virtual router <a> is the first octet in an IP address. It can be numerals or, for all routes, the wildcard (*). [.<b>.<c>.<d>] are the remaining octets in an IP address. They can be numerals or wildcards (*). Once you enter a wildcard, you do not need to complete the address. For example, * instead of *.*.*.*; 47.138.* instead of 47.138.*.* <pro_name> is the name of the routing protocol. For example, ospf.

(Sheet 2 of 2)

Table 44
Monitoring the Ip RouteDataBaseEntry component

Action	Command	Legend
Display the *Ip RouteDataBaseEntry* component	**`display Vr/<vr_name> Ip Rdb/*`**	<vr_name> is the name of the virtual router

Monitoring the virtual media configuration

The *VirtualMedia* (*Vm*) component is an optional root component that you can provision to provide connectivity between VRs or to create an always-up IP interface for RIP, OSPF, or BGP-4. The *Vm* component has no provisionable or operational attributes. The *Vm* component has one subcomponent, the *Interface* (*If*) component.

For more information, see the following tables:

- "Locking and unlocking the Vm If component" (page 345)
- "Monitoring the Vm component" (page 345)
- "Monitoring the Vm If component" (page 346)

Table 45
Locking and unlocking the Vm If component

Action	Command	Legend
Lock an instance of the *Vm If* component.	**lock Vm/<m> If/<n>**	
Unlock an instance of the *Vm If* component.	**unlock Vm/<m> If/<n>**	
Note: Only the *Vm If* component can be locked. The *Vm* component cannot be locked.		

Table 46
Monitoring the Vm component

Action	Command	Legend
Display all *Vm* component instances	**display Vm/***	
Display specific *Vm* component instances	**display Vm/<vm_number>**	

Table 47
Monitoring the Vm If component

Action	Command	Legend
Display all *If* component instances for a *Vm* component in tabular format.	**display Vm/<vm_number> If/***	
Display all *If* component instances for a *Vm* component in non-tabular format.	**display -noTabular Vm/ <vm_number> If/***	
Display all the operational attributes for all instances of the *Vm* and *If* components	**display -o Vm/* If/***	
Display all the operational attributes for a specific instance of the *Vm* component and all instances of its *If* component	**display -o Vm/<vm_name> If/***	

Monitoring the RIP configuration

The following tables describe how to lock and unlock the *Ip Rip* component, monitor the *Ip Rip* and *Ip Rip If* components, and monitor RIP import and export policy. Issue all commands in operational mode. See “Operational mode” (page 30).

- “Locking and unlocking the Rip component” (page 347)
- “Monitoring the Rip component” (page 347)
- “Monitoring the Rip If component” (page 348)
- “Monitoring RIP import and export policy” (page 348)

Table 48
Locking and unlocking the Rip component

Action	Command	Legend
Lock the *Ip Rip* component to take it and all related subcomponents out of service.	**lock Vr/<vr_name> Ip Rip**	
Unlock the *Ip Rip* component to return it all related subcomponents to service.	**unlock Vr/<vr_name> Ip Rip**	

Table 49
Monitoring the Rip component

Action	Command	Legend
Display configurable attributes for the *Rip* component	**display -p Vr/<vr_name> Ip Rip**	
Display operational statistics for the *Rip* component	**display Vr/<vr_name> Ip Rip**	

Table 50
Monitoring the Rip If component

Action	Command	Legend
Display configurable attributes for the *RipIf* component	**display -p Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ipAddress> RipIf**	
Display operational attributes for the *Rip If* component	**display Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ipAddress> RipIf**	

Table 51
Monitoring RIP import and export policy

Action	Command	Legend
Display a RIP import policy	**display Vr/<vr_name> Ip Rip Import/<import_id>**	<import_id> is the instance/ identifier of the particular import policy
Display a RIP export policy	**display Vr/<vr_name> Ip Rip Export/<export_id>**	<export_id> is the instance/ identifier of the particular export policy
Note: To display all import or export polices, use an asterisk (*) as the instance value.		

Monitoring the OSPF configuration

The following tables describe how to lock and unlock the *Ip Ospf* component, monitor the *Ospf* and *OspfIf* components and subcomponents, and monitor the OSPF export policy. Issue all commands in operational mode. See "Operational mode" (page 30).

- "Locking and unlocking the Ospf component" (page 349)
- "Monitoring the OspfIf component" (page 350)
- "Monitoring OSPF export policy" (page 350)
- "Monitoring OSPF areas" (page 350)
- "Monitoring OSPF hosts" (page 351)
- "Monitoring OSPF virtual links" (page 351)
- "Monitoring OSPF stub areas" (page 351)
- "Monitoring OSPF neighbors" (page 351)
- "Monitoring the OSPF link state database" (page 352)

Table 52
Locking and unlocking the Ospf component

Action	Command	Legend
Lock the *Ospf* component to take it and all related subcomponents out of service.	**lock Vr/<vr_name> Ip Ospf**	
Unlock the *Ospf* component to return it and all related subcomponents to service.	**unlock Vr/<vr_name> Ip Ospf**	

Table 53
Monitoring the Ospf component

Action	Command	Legend
Display the *Ip Ospf* subcomponent operational attributes.	**display Vr/<vr_name> Ip Ospf**	

Table 54
Monitoring the OspfIf component

Action	Command	Legend
Display a particular *OspfIf* component (under the *IpPort LogicalIf* component).	**display Vr/<vr_name> Ip Ospf ProtocolPort/ <pp_name> IpPort LogicalIf/<IpAddress> OspfIf**	<pp_name> is the name of the protocol port <IpAddress> is the 32-bit Ip address of the interface

Table 55
Monitoring OSPF export policy

Action	Command	Legend
Display all OSPF export policies.	**display Vr/<vr_name> Ip Ospf export/***	

Table 56
Monitoring OSPF areas

Action	Command	Legend
Display all provisioned OSPF areas attached to this *Vr* component instance.	**display Vr/<vr_name> Ip Ospf AreaEntry/***	
Display all OSPF aggregate areas.	**display Vr/<vr_name> Ip Ospf aggregate/***	

Table 57
Monitoring OSPF hosts

Action	Command	Legend
display Ip OSPF provisioned host entries	**display Vr/<vr_name> Ip Ospf host/***	

Table 58
Monitoring OSPF virtual links

Action	Command	Legend
Display all provisioned OSPF virtual links.	**display -p Vr/<vr_name> Ip Ospf virtif/***	
Display OSPF virtual link operational data.	**display Vr/<vr_name> Ip Ospf virtif/***	

Table 59
Monitoring OSPF stub areas

Action	Command	Legend
Display OSPF stub areas.	**display Vr/<vr_name> Ip Ospf stub/***	

Table 60
Monitoring OSPF neighbors

Action	Command	Legend
Display all OSPF neighbors that have been provisioned or dynamically discovered using the OSPF hello protocol.	**display Vr/<vr_name> Ip Ospf neighbor/***	

Table 61
Monitoring the OSPF link state database

Action	Command	Legend
Display the OSPF link state database.	**display Vr/<vr_name> Ip Ospf Isdb/***	
Display the OSPF external link state database.	**display Vr/<vr_name> Ip Ospf extsdb/***	

Monitoring the BGP-4 configuration

The following tables describe how to lock and unlock the *bgp* component, display BGP operational statistics, and display BGP routes in the databases. Issue all commands in operational mode. See "Operational mode" (page 30).

- "Locking and unlocking the Bgp and Bgp Peer components" (page 353)
- "Monitoring the Bgp and Bgp Peer components" (page 353)
- "Monitoring routes in the BGP routing information base (RIB)" (page 354)

Table 62
Locking and unlocking the Bgp and Bgp Peer components

Action	Command	Legend
Disable a BGP-4 instance for a particular virtual router (VR).	**lock Vr/<vr> Ip Bgp**	
Put the BGP-4 instance back in service.	**unlock Vr/<vr> Ip Bgp**	
Shut down a BGP-4 peer connection.	**lock Vr/<vr> Ip Bgp Peer/<peer>**	
Re-establish the BGP-4 peer connection.	**unlock Vr/<vr> Ip Bgp Peer/<IP_address>**	

Table 63
Monitoring the Bgp and Bgp Peer components

Action	Command	Legend
Display operational statistics for the BGP-4 instance.	**display Vr/<vr> Ip Bgp**	<vr> is the instance identifier of the parent VR.
(Sheet 1 of 2)		

Table 63 (continued)
Monitoring the Bgp and Bgp Peer components

Action	Command	Legend
List all of the subcomponents for the BGP-4 instance.	**`list Vr/<vr> Ip Bgp`**	
Display statistics for a BGP-4 peer under a BGP-4 instance.	**`display Vr/<vr> Ip Bgp Peer/<peer>`**	<peer> is the IP address of the BGP-4 peer. To specify all BGP-4 peers under the BGP-4 instance, enter *

(Sheet 2 of 2)

Table 64
Monitoring routes in the BGP routing information base (RIB)

Action	Command	Legend
Display information for one or more routes received from a BGP-4 peer.	**`display Vr/<vr> Ip Bgp Indb/<in_entry>`**	<vr> is the instance identifier of the parent VR. <in_entry> specifies the input database entry. To specify all entries in the Indb, enter *.
Display information for one or more entries in the BGP-4 local database.	**`display Vr/<vr> Ip Bgp Localdb/<loc_entry>`**	<loc_entry> specifies the local database entry. To specify all entries in the Localdb, enter *.
Display information for one or more entries in the BGP-4 output database.	**`display Vr/<vr> Ip Bgp Outdb/<out_entry>`**	<out_entry> specifies the output database entry. To specify all entries in the Outdb, enter *.
Note: The *inLocal* and *calcLocalPref* attributes apply to routes in the Indb only.		

Monitoring the static route configuration

The following tables describe how to lock and unlock, and monitor the *Ip Static* component. Issue all commands in operational mode. See "Operational mode" (page 30).

- "Locking and unlocking the Ip Static component" (page 355)
- "Monitoring the Ip Static component" (page 355)

Table 65
Locking and unlocking the Ip Static component

Action	Command	Legend
Lock the *Ip Static* component to prevent routing over static routes	**lock Vr/<vr_name> Ip Static**	
Unlock the *Ip Static* component to put it back into a service-providing role	**unlock Vr/<vr_name> Ip Static**	

Table 66
Monitoring the Ip Static component

Action	Command	Legend
Display the *Ip Static* component route entries	**display Vr/<vr_name> Ip Static Route/***	
Display *Ip Static* component discard route entries	**display Vr/<vr_name> Ip Static DiscardRouteEntry/***	

Monitoring the IP multicast configuration

The following tables describe how to lock and unlock, and monitor the *Ip Mcast* component. Issue all commands in operational mode. See "Operational mode" (page 30).

- "Monitoring IP multicast and IGMP" (page 356)
- "Monitoring PIM-SM" (page 357)

Table 67
Monitoring IP multicast and IGMP

Action	Command	Legend
Display the multicast cache.	**Display Vr/<vr_name> Ip Mcast Cache/<n>**	
Display the *Mcast* component operational statistics.	**Display Vr/<vr_name> Ip Mcast**	
Display all configured static routes	**Display Vr/<vr_name> Ip Mcast Static RouteEntry/***	
Lock the *Mcast* component	**Lock Vr/<vr_name> Ip Mcast**	
Unlock the *Mcast* component	**Unlock Vr/<vr_name> Ip Mcast**	
Display the *Igmp* component operational statistics	**Display Vr/<vr_name> Ip Mcast Igmp**	
Display the IGMP group cache	**Display Vr/<vr_name> Ip Mcast Igmp Cache/***	
Lock the *Igmp* component	**Lock Vr/<vr_name> Ip Mcast Igmp**	
Unlock the *Igmp* component	**Unlock Vr/<vr_name> Ip Mcast Igmp**	

Table 68
Monitoring PIM-SM

Action	Command	Legend
Display multicast forwarding table(s)	**Display Vr/<vr_name> Ip Mcast Fwd/***	
Display the outbound interface table(s) for the forwarding table	**Display Vr/<vr_name> Ip Mcast Fwd/* Oif/***	
Display a multicast PIM neighbor	**Display Vr/<vr_name> Ip Mcast PimNbr/***	
Display the PIM-SM Bootstrap router (BSR) and other operational information in a domain	**Display Vr/<vr_name> Ip Mcast PimSm Domain/<n>**	
Display a multicast PIM-SM domain and RP-Set	**Display Vr/<vr_name> Ip Mcast PimSm Domain/<n> RpSet/***	
Lock the *PimSm* component	**Lock Vr/<vr_name> Ip Mcast PimSm**	
Unlock the *PimSm* component	**Unlock Vr/<vr_name> Ip Mcast PimSm**	

Monitoring the virtual router redundancy protocol configuration

This section contains the information you need to monitor and maintain the virtual router redundancy protocol (VRRP) configuration. Issue all commands in operational mode. See “Operational mode” (page 30).

For information on specific components and protocols, see the following sections:

- “Displaying VRRP operational information” (page 358)
- “Locking and unlocking the VRRP component” (page 358)

Displaying VRRP operational information

You can display operational information about VRRP using the table “Displaying the VRRP operational information” (page 358).

Table 69
Displaying the VRRP operational information

Action	Command	Legend
Display the operational attribute defined by the *VRRP* component:	**`display Vr/ <vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid> virtualRouterState`**	<vrrp_router> is the name of the Passport VR running VRRP. <VRid> is the instance of VRRP.

Locking and unlocking the VRRP component

The following section describes how to lock and unlock the VRRP component.

Table 70
Locking and unlocking the VRRP component

Action	Command	Legend
Lock a *Vrrp* component. Locking an active VRRP master router results in the VRRP backup router assuming the master router role.	**lock vr/<vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid>**	<vrrp_router> is the name of the Passport VR running VRRP. <VRid> is the instance of VRRP.
Unlock a *Vrrp* component. Unlocking a designated master router results in return of the acting master to a backup state.	**unlock vr/<vrrp_router> Protocolport/<enet> IPport Vrrp/<VRid>**	

Monitoring the IP CoS configuration

This section contains the information you need to monitor and maintain the IP CoS configuration. Issue all commands in operational mode. See “Operational mode” (page 30).

For information on specific components and protocols, see the following sections:

- “Using the ping command with IP CoS” (page 360)
- “Monitoring the IP CoS configuration” (page 361)

Using the ping command with IP CoS

You can send an ICMP packet to a remote IP address with a specific CoS and ToS value. The table “Testing connectivity using the ICMP subcomponent” (page 336) specifies the commands you can use to test connectivity.

Table 71
Using the ping command with IP CoS

Action	Command	Legend
Send a ping packet with a Cos value to a specific IP address:	**ping -i(<ipaddr>) -cos(<cos>) Vr/<vr> Ip Icmp**	<ipaddr> is the remote IP address. <cos> is the CoS value for the packet, and a value of 0, 1, 2, or 3. <vr> is the instance identifier of the remote VR.
Send a ping packet with a TOS byte value to a specific IP address:	**ping -i(<ipaddr>) -tos(<tos>) Vr/<vr> Ip Icmp**	<tos> is the hexadecimal ToS byte value for the packet.
Send a ping packet with a DSCP value to a specific IP address:	**ping -i(<ipaddr>) -dscp(<dscp_value>) Vr/<vr> Ip Icmp**	<dscp_value> is the decimal DSCP value for the packet.

Monitoring the IP CoS configuration

The following section describes how to display configuration information and operational statistics for IP CoS.

Table 72
Monitoring IP CoS

Action	Command	Legend
Display the configurable attributes for specific classification policies on ingress:	**display -p Vr/<vr> Ip Pg/<grp> Policy/<plcy> TosMap**	<plcy> is the instance identifier of the classification policy.
	display -p Vr/<vr> Ip Pg/<grp> Policy/<plcy> IpAddrLayer4Flow/<flw>	<flw> is the instance identifier of the flow identification policy
	display -p Vr/<vr> Ip Pg/<grp> ingressCosTreatment/*	<grp> is the instance identifier of the CoS policy group
Display the configurable attributes for packet treatment profiles under a specific policy group applied on egress:	**display -p Vr/<vr> Ip Pg/<grp> egressCosTreatment/***	<grp> is the instance identifier of the CoS policy group
Note: -p enables you to list provisionable components while in operational mode		

Monitoring IP DiffServ configuration

Monitor IP DiffServ configuration to help you maintain and monitor your differentiated services configuration.

- “Confirming IP DiffServ interface profile usage” (page 363)
- “Confirming IP DiffServ interface profile configuration” (page 365)
- “Confirming connection class of connected media” (page 367)
- “Displaying per-hop behaviors” (page 369)
- “Pinging ICMP with IP DiffServ” (page 371)

For more information on monitoring connected media see “Monitoring the ATM MPE configuration” (page 314) or “Monitoring the Arp subcomponent” (page 339).

Confirming IP DiffServ interface profile usage

Confirm IP DiffServ interface profile usage to verify how differentiated services have been deployed.

Prerequisites

- You must have completed the procedure "IP DiffServ configuration" (page 265).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on differentiated services.
- See 241-5701-060 *Passport 7400, 15000, 20000 Components* for more information on the components and attributes used in this procedure.

Procedure steps

1. Display the subcomponents that are linked to differentiated services.

   ```
   d -p Vr/<vr_name> Ip Ds <ds_inst> linkToDiffServUsers
   ```

2. Display the differentiated services the Ipport is using.

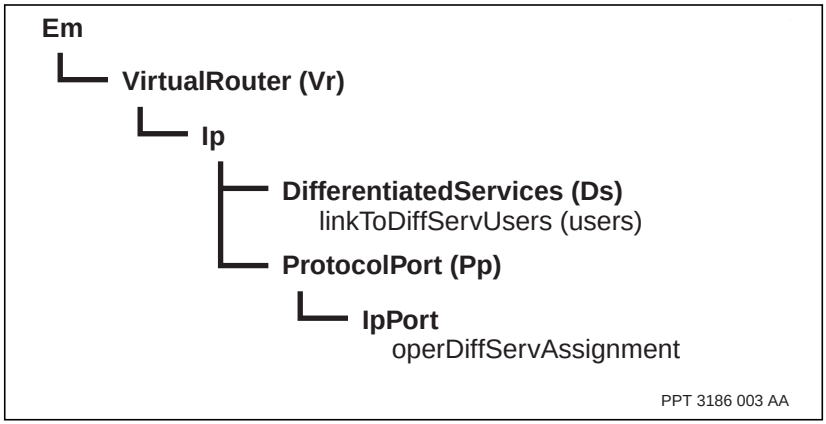
   ```
   d -o Vr/<vr_name> Pp/<pp_name> ipPort
   operDiffServAssignment
   ```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.
<ds_inst>	The instance of the DiffServ component.
<pp_name>	The name of the protocol port.

Procedure job aid

Figure 94
Confirming IP DiffServ interface profile usage component hierarchy



Confirming IP DiffServ interface profile configuration

Confirm IP DiffServ interface profile configuration to verify that the DSCP treatments and discard priority have been configured correctly.

Prerequisites

- You must have completed the procedure "IP DiffServ configuration" (page 265).
- You must have completed the procedure "Adding an IP DiffServ interface profile to a virtual router" (page 283).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on differentiated services.
- See 241-5701-060 *Passport 7400, 15000, 20000 Components* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display the ingress services.

```
d vr/<vr_name> Ip Ds/<ds_inst> Is
```

2 Display the behavior aggregate components of the ingress services.

```
d vr/<vr_name> Ip Ds/<ds_inst> Is Ba/*
```

3 Display the egress services.

```
d Vr/<vr_name> Ip Ds/<ds_inst> Es
```

4 Display the behavior aggregate components of the egress services.

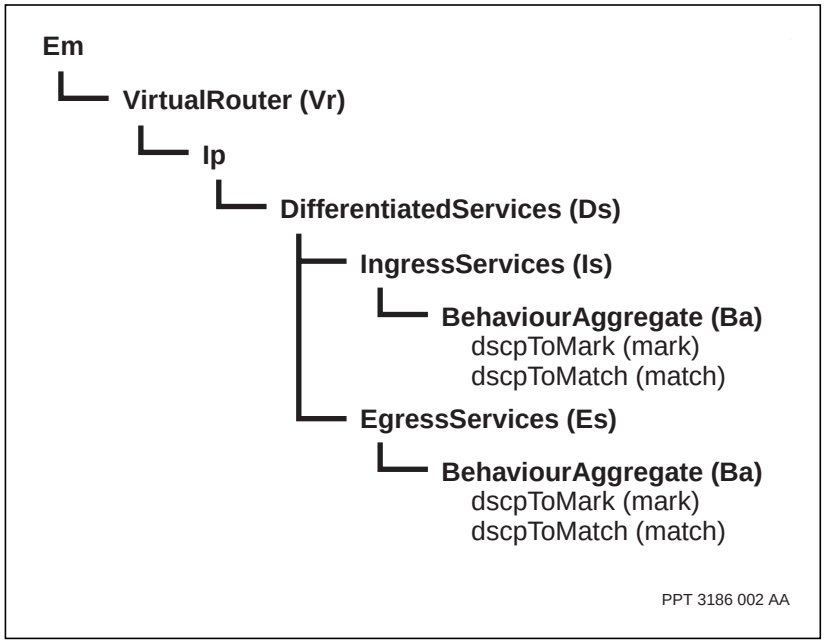
```
d vr/<vr_name> Ip Ds/<ds_inst> Es Ba/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.
<ds_inst>	The instance of the DiffServ component.

Procedure job aid

Figure 95
Confirming DiffServ configuration component hierarchy



Confirming connection class of connected media

Confirm the connection class of connected media to verify that the connection class values specified by *AtmMpe Ac ipCos* matches the connection class value of the connected media.

Prerequisites

- You must have completed the procedure "IP DiffServ configuration" (page 265).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on differentiated services.
- See 241-5701-060 *Passport 7400, 15000, 20000 Components* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display information for all the dynamic host entries.

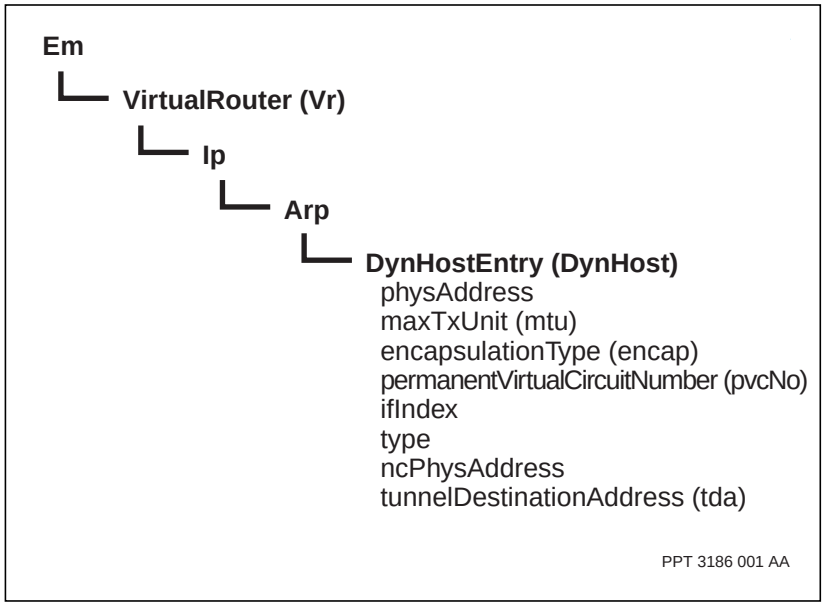
```
d Vr/<vr_name> Ip Arp DynHost/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router.

Procedure job aid

Figure 96
Confirming connection class of connected media component hierarchy



Displaying per-hop behaviors

Display per-hop-behaviors to verify that each PHB is delivering the proper treatment to the IP packets.

Prerequisites

- You must have completed the procedure “IP DiffServ configuration” (page 265).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on differentiated services.
- See 241-5701-060 *Passport 7400, 15000, 20000 Components* for more information on the components and attributes used in this procedure.

Procedure steps

1 Display the per-hop behaviors for the differentiated services domain.

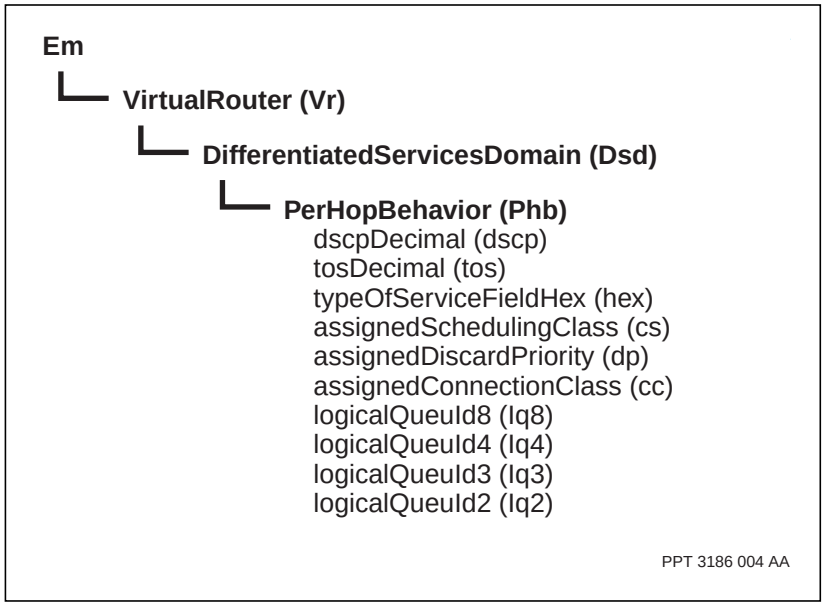
```
d -o Vr/<vr_name> Dsd/<domain_type> Phb/*
```

Variable values

Variable	Value
<vr_name>	The name of the virtual router
<domain_type>	The type of IP DiffServ domain.

Procedure job aid

Figure 97
Displaying per-hop-behaviors component hierarchy



Pinging ICMP with IP DiffServ

Ping ICMP with IP DiffServ to determine the time it takes for a packet to travel to an address in the network.

Prerequisites

- You must complete the procedure “Adding a DiffServ domain to the virtual router” (page 274).
- See 241-5701-805 *Passport 7400, 15000, 20000 Understanding IP* for more information on differentiated services.
- See 241-5701-060 *Passport 7400, 15000, 20000 Components* for more information on the components and attributes used in this procedure.

Procedure steps

1 Ping an IP address with a specified DSCP value.

```
ping -i(<ip_addr>) -dscp (<dscp_value>) Vr/<vr_name>
Ip Icmp
```

2 Ping an IP address with a specified TOS value.

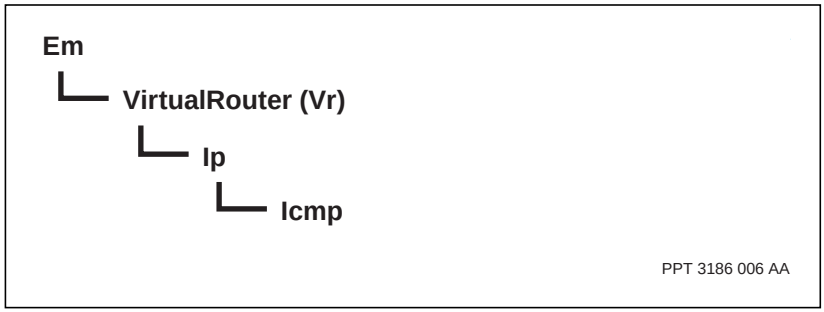
```
ping -i(<ip_addr>) -tos (<tos_value>) Vr/<vr_name> Ip
Icmp
```

Variable values

Variable	Value
<ip_addr>	The IP address you want to ping.
<dscp_value>	The decimal DSCP value you want to mark in the IP packet header.
<tos_value>	The hexadecimal ToS byte value you want to mark in the IP packet header.
<vr_name>	The name of the virtual router

Procedure job aid

Figure 98
Pinging ICMP with IP DiffServ component hierarchy



Monitoring the IP flow filters configuration

You can perform the tests described in this section to determine whether you have configured IP flow filters properly. Issue all commands in operational mode. See “Operational mode” (page 30). For more information, see the following sections:

- “Monitoring the filter component” (page 373)
- “Monitoring the filterFlow subcomponent” (page 373)

Monitoring the filter component

The following section describes how to display operational statistics for the *filter* component. For more information, see the table “Monitoring the IP cache subcomponent” (page 334).

Table 73
Monitoring the filter component

Action	Command	Legend
List all *filter* component instances	**list Vr/<vr_name> Ip filter/***	<vr_name> is the name of the virtual router
List all subcomponents of particular *filter* component instance	**list Vr/<vr_name> Ip filter/<filter_name>**	<filter_name> is the name of the filter
Display operational statistics for a particular *filter* component instance	**display Vr/<vr_name> Ip filter/<filter_name>**	

Monitoring the filterFlow subcomponent

The following section describes how to display operational statistics for the *filterFlow* subcomponent. For more information, see the table “Monitoring the filterFlow subcomponent” (page 374).

Table 74
Monitoring the filterFlow subcomponent

Action	Command	Legend
List all *filter* component instances	**list Vr/<vr_name> Ip filter/* Ip FilterFlow/ ***	<vr_name> is the name of the virtual router
List all subcomponents of particular *filterFlow* subcomponent instance	**list Vr/<vr_name> Ip filter/<filter_name> Ip FilterFlow/ <filterflow_number>**	<filter_name> is the name of the filter <filterflow_number> is the number of the flow filter
Display operational statistics for a particular *filterFlow* subcomponent instance	**display Vr/<vr_name> Ip filter/<filter_name> Ip FilterFlow/ <filterflow_number>**	

Monitoring the IP tunnel configuration

This section contains the information you need to monitor and maintain the IP tunnel configuration. Issue all commands in operational mode. See "Operational mode" (page 30).

- "Locking and unlocking the Tunnel component" (page 375)
- "Monitoring the Tunnel component" (page 376)
- "Monitoring the ProtocolPort component for an IP tunnel" (page 376)
- "Monitoring the IpPort component for an IP tunnel" (page 377)
- "Monitoring the LogicalIf component for an IP tunnel" (page 377)

Table 75
Locking and unlocking the Tunnel component

Action	Command	Legend
Lock the *Ip Tunnel* component to take it and all related subcomponents out of service	**lock Vr/<vr_name> Pp/ <pp_name>**	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Unlock the *Ip Tunnel* component to return it all related subcomponents to service	**unlock Vr/<vr_name> Pp/ <pp_name>**	
Lock all instances of IP tunnel end points for this instance of the *Vr* component	**lock Vr/<vr_name> Ip Tunnel**	
Unlock all instances of IP tunnel end points for this instance of the *Vr* component	**unlock Vr/<vr_name> Ip Tunnel**	

Table 76
Monitoring the Tunnel component

Action	Command	Legend
Display configurable attributes for the *Tunnel* component	**display -p Vr/<vr_name> Tunnel StaticEndPoint/ <endpoint_id>**	
Display operational statistics for the *Tunnel* component	**display Vr/<vr_name> Tunnel StaticEndPoint/ <endpoint_id>**	
Note: -p enables you to list provisionable components while in operational mode		

Table 77
Monitoring the ProtocolPort component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the tunnel *ProtocolPort* component	**display -p Vr/<vr_name> Pp/<pp_name>**	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display operational attributes for the tunnel *ProtocolPort* component	**display Vr/<vr_name> Pp/<pp_name>**	
Note: -p enables you to list provisionable components while in operational mode		

Table 78
Monitoring the IpPort component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the tunnel *IpPort* component	**display -p Vr/<vr_name> Pp/<pp_name> IpPort**	<vr_name> is the name of the virtual router <pp_name> is the name of the protocol port
Display operational attributes for the tunnel *IpPort* component	**display -p Vr/<vr_name> Pp/<pp_name> IpPort**	
Note: -p enables you to list provisionable components while in operational mode		

Table 79
Monitoring the LogicalIf component for an IP tunnel

Action	Command	Legend
Display configurable attributes for the *LogicalIf* component	**display -p Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ip_addr>**	<pp_name> is the name of the tunnel protocol port. <ip_addr> is the IP address of the IP logical interface.
Display operational attributes for the *LogicalIf* component	**display Vr/<vr_name> Pp/<pp_name> IpPort LogicalIf/<ip_address>**	
Note: -p enables you to list provisionable components while in operational mode		

Chapter 27
Troubleshooting

This section contains information about the following:

- “Troubleshooting ATM MPE” (page 380)
- “Troubleshooting frame relay DTE” (page 383)
- “Troubleshooting PPP” (page 384)
- “Troubleshooting PPP/ATM interworking” (page 388)

Troubleshooting ATM MPE

The table “Handling problems with ATM MPE” (page 380) provides guidelines on how to respond to problems that might occur when using the ATM MPE service.

Table 80
Handling problems with ATM MPE

Problem	Possible cause	Solution
ATM MPE does not provide service.	The ATM MPE interface is locked.	Unlock the *AtmMpe* component.
	The ATM MPE VCC is locked.	Unlock the *AtmMpe AtmConnection* component.
	The *ifAdminStatus* of the ATM MPE VCC is down.	Configure the *ifAdminStatus* as up.
	The ILS Forwarder interface is locked	Unlock the *ilsFwdr* component.
	The *ifAdminStatus* of the ILS Forwarder interface is down	Configure the *ifAdminStatus* as *up*.
	The ATM FP or ATM port is locked.	Unlock the ATM FP or ATM port.

(Sheet 1 of 3)

Table 80 (continued)
Handling problems with ATM MPE

Problem	Possible cause	Solution
IP over ATM MPE is not functioning.	The encapsulation type for the *AtmMpe* component is *ipVcEncap*, and there are no static ARP entries provisioned.	Change the encapsulation type to *llcEncap* or add static ARP entries.
	The ARP table has been cleared. This flushes all inverse ARP entries (these are not supplied dynamically).	Lock and unlock the *AtmMpe* component.
	The encapsulation type used by the ATM MPE interfaces at each end of a VCC is not consistent. Both *AtmMpe* instances that terminate a VCC must use the same encapsulation type.	Configure both ends of the VCC to use the same encapsulation type.
	IP is not on the feature list of the ILS Forwarder FP.	Add IP to the ILS Forwarder FP feature list.
An instance of the *AtmMpe* component is disabled, but not locked, and the *ilsFwdr* component is enabled.	The ATM MPE feature is not in the feature list for the ATM FP.	Add the ATM MPE feature to the ATM FP software feature list.
(Sheet 2 of 3)		

Table 80 (continued)
Handling problems with ATM MPE

Problem	Possible cause	Solution
The ATM MPE soft PVC fails to establish.	The called endpoint is either busy or does not exist. You can tell that this is the case because the *AtmMpe Ac SrcPvc retryCount* will continue to increment, and the *lastClearCause* value for that component will be 34 (requested called party soft PVCC not available).	Correct the *remoteAddress* and *remoteCi* attributes for the *SrcPvc* component. Make sure that only one calling endpoint in the ATM network is trying to connect to the called endpoint.
	There is a link, node, or routing failure somewhere in the PNNI network. You can tell that this is the case because the *lastClearCause* will not be 34.	Use the *lastClearCause* to diagnose the problem.

(Sheet 3 of 3)

Troubleshooting frame relay DTE

The table “Handling problems with frame relay DTE” (page 383) provides guidelines on how to respond to problems that might occur when using the frame relay DTE service.

Table 81
Handling problems with frame relay DTE

Problem	Possible cause	Solution
Data transfer over all connections is suspended.	The number of local management interface (LMI) procedure errors within the last monitoredEvents window has exceeded the threshold errorThreshold.	Verify that the network equipment has the LMI protocol enabled. Verify that the LMI parameters set on the network equipment are compatible with those on the router. Turn off the LMI protocol for the frame relay DTE if the network equipment does not support any of the available LMI protocols. Do this by setting the attribute of the LMI component to none.

Troubleshooting PPP

The table “Handling problems with PPP” (page 384) provides guidelines on how to respond to problems that might occur when using the PPP service.

Table 82
Handling problems with PPP

Problem	Possible cause	Solution
The link does not come up. Link Control Protocol (LCP) cannot enter the “open” state.	There is incorrect provisioning data.	Make sure that both ends of the link have compatible provisioning data.
	There is a bad cable or cables.	Test cables using port and line tests and remove bad cables. Look at the *lineCondition* attribute in the *Ppp/n Link* component and make sure the line is correct.
	There has been an operator error.	Check the physical layer attributes to ensure that essential components have not been locked.
	There is no clock from the modem or peer DCE connection.	Check the physical layer statistics to see if the physical layer is operational and ready to provide link service to the PPP application.
	Peer link is not initiating the LCP configure request transmission or not responding with an LCP configure acknowledge.	Check *Ppp/n Link operState* attribute to determine the state of the LCP connection. If it is in *reqsent* then the PPP is trying to connect and there must be something wrong with the physical link connection, or the peer PPP is not talking to us.

(Sheet 1 of 4)

Table 82 (continued)
Handling problems with PPP

Problem	Possible cause	Solution
The link does not come up. Link Control Protocol (LCP) cannot enter the open state.	The quality of the connection may be poor.	If LQM is turned on, then a Bad Line Quality alarm may be set. This can happen only after the *Ppp/n Link operState* attribute is open.
	LCP packets are not getting through to or from peer PPP connection.	Look at the *Ppp/n Framer* statistics of both sides of the link and confirm that there are no CRC errors etc. Look at the ifTable for the physical layer component such as V.35 or DS1 etc. Also look at *Ppp/n Link* attributes to see if PPP is receiving frames that are either too long, too short, of badly formed frames.
A Network Control Protocol (NCP) does not enter the open state.	The network layer protocol is not provisioned on either the local or peer shelf.	Check the provisioning of the network layer protocol on each side of the connection. Look at the *ppp/n ncp state* attributes for the protocol to help determine which side does not want to connect.
	The peer PPP application does not want to open the NCP in question.	Check peer connection and confirm that the network layer protocol does indeed want to connect to the PPP.

(Sheet 2 of 4)

Table 82 (continued)
Handling problems with PPP

Problem	Possible cause	Solution
The link connection keeps going down due to bad line quality. (Note this can only occur when Link Quality Monitoring (LQM) is enabled.)	The link quality could be bad.	Either take measures to improve link quality, or set the quality threshold down in the *ppp/n link qualityThreshold* attribute. Check for CRC errors on both sides of the connection.
	The PPP connection may be over-driving the peer's link, or the peer may be over-driving the side of the link resulting in dropped data or LQM reporting packets. This typically happens only at DS3/E3 link speeds.	Disable LQM or throttle the network layer applications data rate.
	The reporting period for link quality reports is set for too short a period for the data rate being used.	Set the *ppp/n lqm configPeriod* attribute to 100 centiseconds or set the period to zero (default) and allow the period to be determined by the peer or set the value to something over 180,000 centiseconds which allows the reporting period to be calculated based on the data rate.
The link connection keeps going down due to bad line quality. (Note that this can only occur when Link Quality Monitoring (LQM) is enabled.)	The window size may be too small. This is the interval of time during which samples may be accumulated. The window may be provisioned such that very bursty traffic results in an erroneous quality determination.	Set the ppp/n link qualityWindow to a larger value. Typically the 30 second default should be fine for all applications.
	The peer PPP connection might not be sending Link Quality Reports (LQRs).	Check the ppp/n inLqrs counter. If it is not incrementing, the peer PPP application is not responding correctly to LQM.

(Sheet 3 of 4)

Table 82 (continued)
Handling problems with PPP

Problem	Possible cause	Solution
The link connection keeps going down due to loss of link continuity. (Note that this can only happen when Link Continuity Monitoring (LCM) is enabled.)	The link quality could be bad.	Improve link quality. Check for CRC or any other errors in the physical layer statistics on both sides of the link.
	The PPP connection may be over driving the peer's link, or the peer may be over driving the side of the link resulting in dropped LCM packets. This typically only happens at DS3/E3 link speeds.	Disable LCM or throttle the network layer applications data rate.
Network layer traffic gets dropped when bursts of data occur.	The congestion threshold of the link interface is being reached forcing packets to be discarded. Check PPPs ifTable to determine if the ifOutDiscards counter is incrementing. DS3/E3 link speeds may use the direct hardware transmit method. As a result, ifOutDiscards are not incremented when congestion thresholds are reached because congestion management is done in hardware.	On a Passport 7400 with SBIC-based FPs, add an LEQ to the PPP application. This allows more elasticity in the data path for bursty traffic conditions. Adding an LEQ turns off direct hardware transmit because all packets are processed by the outbound FP running the LEQ code.

(Sheet 4 of 4)

Troubleshooting PPP/ATM interworking

The following list contains information on troubleshooting PPP/ATM interworking:

- If an SPVC does not establish, you need to check the *PppIwf* status and the *AtmIf* port status.
- If the port state is enabled and busy, you need to verify the *lastStepFailureCause*. For example, if this attribute is set to 3 and has no route to destination, PNNI cannot find the route to the destination node. Another possible cause is that the PppIwf feature is not loaded on the ATM FP configured with PNNI.
- If the OSPF is in exchange mode on PPP and at the *AtmMpe* end, it is possible that the static ARP entry is not added on the *AtmMpe* end.

Passport 7400, 15000, 20000

Configuring IP

Release 5.2

Publication: 241-5701-810
Document status: Standard
Document version: 5.2S2
Document date: December 2003
Printed in Canada

NORTEL NETWORKS