

Aruba Advanced SwitchingTroubleshooting and Solutions

TRAINING MANUAL

EDU-ASTS-ILT-v20.41



Aruba Education Services

Aruba Advanced Switching Troubleshooting and Solutions Student Guide

December 2020

SKU: EDU-ASTS-ILT-Rev 20.41

Aruba Advanced Switching Troubleshooting and Solutions Student Guide

Copyright
© 2020 Aruba Networks, Inc. AirWave®, Aruba Networks®, Aruba Mobility Management System®, Bluescanner, For Wireless That Works®, Mobile Edge Architecture, People Move. Networks Must Follow., RFProtect, The All Wireless Workplace Is Now Open For Business, and The Mobile Edge Company® are trademarks of Aruba Networks, Inc. All rights reserved. All other trademarks are the property of their respective owners.

Open Source Code
Certain Aruba products include Open Source software code developed by third parties, including software code subject to the GNU General Public License ("GPL"), GNU Lesser General Public License ("LGPL"), or other Open Source Licenses. The Open Source code used can be found at this site:

http://www.arubanetworks.com/open_source

Legal Notice
The use of Aruba Networks, Inc. switching platforms and software, by all individuals or corporations, to terminate other vendors' VPN client devices constitutes complete acceptance of liability by that individual or corporation for this action and indemnifies, in full, Aruba Networks, Inc. from any and all legal actions that might be taken against it with respect to infringement of copyright on behalf of those vendors.

Warranty
This hardware product is protected by the standard Aruba warranty of one year parts/labor. For more information, refer to the ARUBACARE SERVICE AND SUPPORT TERMS AND CONDITIONS.
Altering this device (such as painting it) voids the warranty.

SKU: EDU-ASTS-ILT-v20.41
December 2020

Aruba Advanced Switching Troubleshooting and Solutions

LEARNER GUIDE TABLE OF CONTENTS

Module 0: Course Introduction	1
Introductions	2
Logistics	3
Course Objectives	4
Course Objectives	5
Course Design	6
Agenda – Day 1	7
Agenda – Day 2	8
Agenda – Day 3	9
Agenda – Day 4	10
Agenda – Day 5	11
Lab Activity	12
Module 1: Plan the Wired Networks Solution	13
Objectives	14
Overview	15
Plan the Wired Solutions	16
Gather customer requirements and network design	17
Gather customer requirements and network design cont.	18
Steps in Gathering Information	19
Steps in Gathering Information cont.	20
Wired Network Questionnaire	21
Wired Network Questionnaire cont.	22
Create a Proof of Concept (POC) Plan	23
Create a Proof of Concept (POC) Plan cont.	24
Analyze the processes and results of the POC	25
Primary configuration to move to implementation	26

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Lab Activity	27
Module 2: Troubleshooting	28
Objectives	29
Overview	30
Overview cont.	31
Diagnostic Principles	32
When is an Organized Methodology Most Important	33
Troubleshooting Principles	34
Troubleshooting Principles cont.	35
Components of Effective Troubleshooting	36
Components of Effective Troubleshooting cont.	37
Problem-Solving Methodology	38
Problem Solving Methodology	39
Problem Solving Methodology cont.	40
Log Files and Debugging	41
Assess Current Debug State	42
Assess Current Debug State cont.	43
Set Desired Debug State	44
Remote logging review	45
Remote logging review cont.	46
Remote logging review cont.	47
Event logging	48
Accounting Log	49
Question #1	50
Diagnostics Commands	51
Understanding top CPU command	52
Understanding top CPU command cont.	53
Understanding top CPU command	54
Understanding top CPU command cont.	55
Understanding top CPU command	56
Understanding top CPU command cont.	57
Understanding top CPU command	58
Capacities	59
Diagnostics Troubleshooting Tools	60
Diagnostics Troubleshooting Tools cont.	61
Support Files	62
Support Files cont.	63
Question #2	64

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Traffic Analysis	65
Traffic Mirroring Types	66
Traffic Mirroring Types cont.	67
Mirroring review – What to Capture	68
Mirroring review – What to Capture cont.	69
Local Mirroring Configuration and Verification	70
Mirroring Configuration	71
Mirroring Configuration - Tshark	72
TCPDUMP	73
TCPDUMP cont.	74
Question #3	75
Additional Content	76
Troubleshooting Zones	77
Troubleshooting Zones cont.	78
Identification and Analysis	79
Identification and Analysis cont.	80
Hypothesis and Validation	81
Hypothesis and Validation cont.	82
Top CPU and memory	83
Self Test	84
Self Test cont.	85
Environment	86
Diagnostics - Environment	87
Diag on-demand	88
Lab Activity	89
Module 3: Monitoring and Automation	90
Objectives	91
Overview	92
REST API	93
Network Automation with AOS-CX	94
REST API	95
REST Methods and enabling REST Interface	96
Login/Logout the REST interface	97
Login/Logout the REST interface cont.	98
Sending REST API Requests	99
Swagger Interface	100
REST API versions	101
REST API versions cont.	102

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

REST API versions cont. 103
Swagger Interface 104
Swagger Interface 105
cURL request 106
cURL request cont. 107
cURL request cont. 108
Postman request 109
Postman request 110
Question #1 111
Network Analytic Engine 112
Network Analytic Engine 113
Network Analytic Engine cont. 114
NAE – Scripts and Agents 115
NAE – Scripts and Agents cont. 116
NAE – Script Types and sources 117
NAE – Script Types and sources cont. 118
NAE – supported maximums 119
NAE – Parts of a script 120
NAE – Parts of a script 121
NAE – Parts of a script 122
NAE – Parts of a script 123
NAE – Parts of a script 124
NAE – Parts of a script 125
NAE – Parts of a script 126
Troubleshooting NAE 127
Troubleshooting NAE 128
Question #2 129
Other monitoring 130
IP Service Level Agreement (SLA) 131
IP Service Level Agreement (SLA) cont 132
IP SLA Configuration example 133
IP SLA Configuration example cont. 134
Simple Network Management Protocol – SNMP 135
Simple Network Management Protocol – SNMP cont. 136
SNMPv3 Configuration 137
NetEdit Troubleshooting 138
NetEdit Troubleshooting 139
NetEdit Troubleshooting 140

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

NetEdit Troubleshooting 141
Question #3 142
Additional Content 143
Swagger Interface 144
Agent Functions 145
Agent Functions cont. 146
Baseline Function 147
Baseline Function cont. 148
Baseline Function cont. 149
NetEdit Logs 150
Lab Activity 151
Module 4: VSX, VSF, and L2 Technologies 152
Objectives 153
Overview 154
Overview cont. 155
VSX 156
Virtual Switching Extension – VSX Overview 157
Virtual Switching Extension – VSX Overview cont. 158
VSX Components 159
VSX Components cont. 160
VSX Components cont. 161
VSX Topologies – Aggregation VSX, single routing model 162
VSX Topologies – Aggregation VSX, multiple VRF routing 163
VSX – Virtual MAC and System-MAC Guidance 164
VSX – Virtual MAC and System-MAC Guidance cont. 165
VSX 166
VSX - Best practices 167
VSX - Best practices 168
VSX - Best practices 169
VSX - Best practices 170
VSX - Best practices 171
VSX - Best practices 172
VSX - Best practices 173
VSX - Best practices 174
VSX - Best practices 175
VSX - Best practices 176
VSX - Best practices 177
VSX - Best practices 178

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

VSX - Best practices 179
Question #2 180
Question #3 181
VSX Troubleshooting 182
VSX Troubleshooting – Status and LAG Interface 183
VSX Troubleshooting Config/VSX Sync 184
VSX Troubleshooting - Debug 185
VSX Troubleshooting – Config Consistency 186
VSX Troubleshooting – Link/Interface Status 187
VSX Troubleshooting – MAC / Peer Consistency 188
VSX Troubleshooting – LACP Interfaces 189
VSX Troubleshooting – LACP Interfaces 190
VSX Troubleshooting – Config Parser 191
VSX 192
VSX – Split Brain 193
Additional Content 194
VSF 195
VSF Overview 196
VSF Overview cont. 197
VSF Stack Requirements 198
VSF Member Roles and Links 199
VSF Split Stack – Link Failure 200
VSF Split Stack – Member Failure 201
VSF Link failure 202
Split Detection 203
VSF Troubleshooting 204
VSF Troubleshooting – Status and Members 205
VSF Troubleshooting – Links and Topology 206
VSF Troubleshooting – Removing a VSF member 207
VSF Troubleshooting – Removing a VSF member 208
VSF Troubleshooting – Removing a VSF member 209
VSF Troubleshooting – Removing a VSF member 210
VSX – Split Brain 211
Lab Activity 212
Module 5: Layer 3 Routing and OSPF 213
Objectives 214
Overview 215
Overview cont. 216

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Policy Based Routing 217
PBR Overview 218
PBR Overview 219
PBR Configuration 220
PBR Configuration 221
OSPF Single Area 222
OSPF Adjacencies 223
OSPF Adjacencies 224
OSPF Authentication with Keychains 225
OSPF with BFD 226
OSPF with BFD 227
Relate Route Table to Network Diagram 228
Relate Route Table to Network Diagram 229
Relate OSPF Neighbor Table to Network Diagram 230
Relate OSPF Neighbor Table to Network Diagram 231
Relate OSPF Neighbor Table to Network Diagram 232
Relate OSPF Database to Network Diagram 233
Relate OSPF Database to Network Diagram 234
Relate OSPF Database to Network Diagram 235
Type 1 LSA with Broadcast Networks 236
Question #1 237
Question #2 238
OSPF Multi-Area 239
OSPF LSA Review 240
Multi-area OSPF – No Route Summarization 241
Multi-area OSPF – Route Summarization 242
Multi-area OSPF – Route Filtering 243
OSPF Configuration Best Practices 244
Question #3 – Match Column 1 with Column 2 245
Additional Content 246
Static Routing 247
Static Routes 248
Static Default Route 249
Floating Static Routes 250
Static Routes with BFD 251
Static Routes with BFD 252
PBR Validation: Interface 253
PBR Validation: Summary 254

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

OSPF Single Area 255
OSPF General Operation Review 256
OSPF General Operation Review 257
OSPF General Operation Review 258
OSPF Adjacency Established 259
OSPF Adjacency Established 260
Passive Interfaces 261
Passive Interfaces 262
Setting Interface Costs 263
Setting Interface Costs 264
OSPF Multi-Area 265
Multi-Area OSPF Review 266
Multi-Area OSPF Review 267
Area Border Routers 268
Area Border Routers 269
Area Border Routers 270
Inter-Area Route Aggregation 271
Configure Route Aggregation 272
Configure Route Aggregation 273
Configure Route Aggregation 274
Configure Route Filtering 275
Configure Route Filtering 276
Configuring Cost Value 277
Choosing Best Inter-Area Routes 278
Choosing Best Inter-Area Routes 279
Choosing Best Inter-Area Routes 280
OSPF Troubleshooting Commands Review 281
OSPF Troubleshooting Commands Review 282
Lab Activity 283
Module 6: BGP 284
Objectives 285
Overview 286
Overview 287
BGP Concepts and Peers 288
Six Key Points about BGP Peer Configuration 289
Six Key Points about BGP Peer Configuration 290
Validate BGP Peers 291
BGP Advertisements 292

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

BGP only Advertises Reachable Routes 293
BGP only Advertises Reachable Routes 294
BGP Advertisement: Network Command 295
BGP Advertisement: Aggregate-address Command 296
BGP Advertisement: Aggregate-address Command 297
BGP Routing Table 298
BGP Routing Table 299
Question #1 300
BGP metrics and Tuning 301
BGP Path Selection Criteria 302
BGP Path Selection Criteria 303
BGP Path Selection Criteria 304
Prefix-lists 305
Route Maps 306
Route Maps 307
Weight 308
Validate Weight 309
Local Preference 310
Local Preference: Agg-1/Agg-2 Before Tuning 311
Local Preference: Agg-1/Agg-2 After Tuning 312
BGP AS Path Length Metric 313
BGP AS Path Length Tuning 314
Multiple Exit Discriminator (MED) 315
MED Tuning Results 316
Question #2 317
Route Control 318
Reasons to Control eBGP Routes 319
Mitigating the iBGP Full-Mesh Requirement 320
Mitigating the iBGP Full-Mesh Requirement 321
Route Reflectors: No Need for Full-Mesh iBGP 322
BGP Communities 323
BGP Communities 324
BGP Communities – No Export 325
AS-Path Relax 326
AS-Path Relax 327
Question #3 328
Additional Content 329
BGP Use Case: Multi-Homed ISP Connections 330

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

BGP Features	331
BGP AS	332
BGP AS	333
BGP Sessions	334
BGP Connection States	335
BGP Connection States	336
Peer debug: Successful Establishment	337
Peer debug: Successful Establishment	338
MD5 Authentication	339
iBGP Full-Mesh Peers	340
iBGP Full-Mesh Peers	341
eBGP Multi-hop Peers	342
eBGP Multi-hop Peers	343
Validate BGP Peers	344
BGP Advertisements	345
iBGP Full Mesh Requirement	346
iBGP Route Advertisements	347
iBGP Advertises eBGP Routes	348
Route Control	349
BGP Tuning Review	350
Outbound Route Filter: Before Tuning	351
Outbound Route Filter: After Tuning	352
Control Inbound Routes	353
Filter Outbound Routes to eBGP Peers	354
Filter Outbound Routes to eBGP Peers	355
Inbound Route Filter – Before Tuning	356
Injecting External Routes into OSPF	357
Injecting External Routes into OSPF	358
Route Reflectors: No Need for Full-Mesh iBGP	359
Mitigating the iBGP Full-Mesh Requirement	360
BGP Peer Groups: Configuration Efficiency	361
Route Control Implementation	362
Avoid Becoming a Transit AS	363
Avoid Becoming a Transit AS	364
BGP Communities – No-Advertise	365
BGP Communities – No-Advertise	366
BGP Communities – Local-AS	367
Lab Activity	368

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Module 7: Route Redistribution 369
Objectives 370
Overview 371
Redistributing Static Routes to OSPF 372
Route Redistribution Overview 373
Route Redistribution Overview 374
Scenario: Redistribute Static to OSPF 375
Redistribute Static to OSPF – After redistribution 376
OSPF Type 2 External Routes 377
OSPF Type 1 External Routes 378
Tuning the Default Metric 379
Tuning with Route Maps and Prefix Lists 380
Validate Prefix List Tuning 381
Using Route Tags: Static to OSPF 382
Question #1 383
Redistribute OSPF to BGP 384
Redistribute Between OSPF and BGP – Single link 385
Redistribute Between OSPF and BGP – Single link 386
Redistribute Between OSPF and BGP – Single link 387
Redistribute Between OSPF and BGP – Single link 388
Redistribute OSPF and BGP - Optimization 389
Redistribute OSPF and BGP - Optimization 390
Redistribute OSPF and BGP - Optimization 391
Redistribute OSPF and BGP - Optimization 392
Redistribute Between OSPF and BGP – Dual links 393
Redistribute Between OSPF and BGP – Dual links 394
Redistribute Between OSPF and BGP – Dual links 395
Redistribute Between OSPF and BGP – Dual links 396
Redistribute Between OSPF and BGP – Dual links 397
Redistribute Between OSPF and BGP – Dual links 398
Redistribute Between OSPF and BGP – Dual links 399
Redistribute Between OSPF and BGP – Dual links 400
Route Redistribution Tips 401
Route Redistribution Tips 402
Question #2 403
Lab Activity 404
Additional Content 405
Redistribute Static to OSPF – Before Redistribution 406

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Redistribute Static to OSPF – Validation 407
Redistribute Using the Default Metric 408
Compare Tuning Techniques 409
Improve BGP Convergence Times 410
Improve BGP Convergence Times 411
Module 8: VRF and Route Leaking 412
Objectives 413
Overview 414
VRF 415
VRF Overview 416
VRF OSPF Configuration 417
VRF Verification 418
VRF OSPF Validation: securenet 419
VRF Route Leaking 420
Inter-Virtual Router Forwarding (IVRF) Overview 421
Inter-Virtual Router Forwarding (IVRF) Overview 422
IVRF Restrictions and Limitations 423
Static VRF Route Leaking Configuration 424
Static VRF Route Leaking Configuration 425
Static VRF Route Leaking Configuration 426
Dynamic VRF Route Leaking: Route Targets and BGP 427
Dynamic VRF Route Leaking: Route Targets and BGP 428
Dynamic VRF Route Leaking: Route Targets and BGP 429
Dynamic VRF Route Leaking: Route Targets and BGP 430
Dynamic VRF Route Leaking: Route Targets and BGP 431
Dynamic VRF Route Leaking: Route Targets and BGP 432
Dynamic VRF Route Leaking: Route Targets and BGP 433
Dynamic VRF Route Leaking: Route Targets and BGP 434
Question #1 435
Question #2 436
Additional Content 437
VRF OSPF Validation: default 438
VRF Static Route Configuration 439
Lab Activity 440
Module 9: Multicast Routing 441
Objectives 442
Overview 443
Overview 444

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Network Access and Layer 2 Multicast	445
IGMP	446
IGMPv3 Behavior	447
IGMPv3 Snooping	448
IGMPv3 Snooping	449
IGMP robustness	450
VSX and IGMPv3	451
VSX and IGMPv3	452
Question #1	453
PIM-SM	454
PIM-SM Build-up Process	455
PIM-SM Build-up Process	456
PIM-SM Build-up Process	457
PIM-SM Build-up Process	458
PIM-SM Build-up Process	459
RP Configuration	460
Static RP Configuration	461
Static RP Configuration	462
Dynamic RP Configuration – BSR	463
Dynamic RP Configuration – BSR	464
VSX and PIM	465
VSX and PIM	466
VSX and PIM	467
VSX and PIM	468
VSX and PIM	469
VSX and PIM Configuration	470
VSX and PIM Configuration	471
Question #2	472
Multicast Deployment	473
Multicast Deployment Summary	474
Multicast Troubleshooting Flow	475
Multicast Troubleshooting Flow	476
Multicast Troubleshooting Flow	477
Multicast Troubleshooting Flow	478
Multicast Troubleshooting Flow	479
Multicast Troubleshooting Flow	480
Multicast Troubleshooting Flow	481
Additional Content	482

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Recommended Lecture	483
Multicast	484
Introduction to Multicast	485
Multicast Address Assignments	486
Multicast Address Assignments	487
Switching Multicast Frames	488
MAC Address Mapping	489
MAC Address Mapping (Overlapping)	490
IGMP versions	491
IGMPv3 Messages	492
IGMPv3 Messages	493
IGMPv3 Configuration	494
IGMPv3 Configuration	495
IGMPv3 verification	496
IGMPv3 Snooping Configuration	497
IGMPv3 Snooping Configuration	498
IGMPv3 snooping verification	499
PIM REGISTER Message Wireshark	500
IP Multicast at Layer 3	501
Multicast Routing	502
Multicast Routing	503
Multicast Trees Types	504
Group States	505
Reverse Path Forwarding	506
PIM	507
PIM modes and messages	508
PIM modes and messages	509
PIM Hello message	510
PIM-SM: Designated Router	511
PIM-SM: Designated Router	512
PIM-SM: Rendezvous Point	513
BSR Mechanism	514
BSR Mechanism	515
Bootstrap Message Wireshark	516
BSR Mechanism	517
BSR Mechanism	518
BSR Mechanism	519
PIM Configuration	520

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

PIM Configuration 521
PIM Configuration 522
Lab Activity 523
Module 10: Quality of Service 524
Objectives 525
Overview 526
Quality of Service 527
Quality of Service Overview 528
Traffic Characterization 529
Traffic Characterization 530
QoS Services Models 531
QoS Services Models 532
QoS Processing on AOS-CX Switches 533
QoS Processing on AOS-CX Switches 534
Ingress Stage 535
Ingress Stage – Rate Limiting 536
Configuring Rate Limiting 537
Verifying Rate Limiting 538
Ingress Stage – CIR 539
Ingress Stage – CIR 540
Classifier Policy – Verification 541
Prioritization Stage 542
Prioritization Stage – Classification and Marking 543
Prioritization Stage 544
Prioritization Stage – Trust 545
Prioritization Stage – Configuration 546
Prioritization Stage – Configuration 547
Prioritization Stage – Configuration 548
Question #1 549
Remark QoS 550
Remarking process 551
Queuing Stage 552
Queuing Stage 553
Queuing Stage - Configuration 554
Queuing Stage - Configuration 555
Queuing Stage – Outbound QoS Shaping 556
Scheduler Stage 557
Scheduler Stage – Strict Priority 558

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Scheduler Stage – Strict Priority 559
Scheduler Stage – Deficit Weighted Round Robin 560
Scheduler Stage – Deficit Weighted Round Robin 561
Scheduler Stage – Deficit Weighted Round Robin 562
Scheduler Stage – Deficit Weighted Round Robin 563
Scheduler Stage – Deficit Weighted Round Robin 564
Scheduler Stage – Weighted Fair Queuing 565
Scheduler Stage – Weighted Fair Queuing 566
Scheduling – SP with WFQ or DWRR 567
Scheduling – Configuration 568
Scheduling – Configuration 569
Scheduling – Verification 570
Question #2 571
Additional Content 572
Differentiated Services (DiffServ) Model 573
Differentiated Services (DiffServ) Model 574
Marking on Layer 2 and Layer 3 575
Marking on Layer 2 and Layer 4 576
Prioritization Stage – Configuration 577
Prioritization Stage – Configuration 578
Prioritization Stage – Configuration 579
Lab Activity 580
Module 11: Dynamic Segmentation 581
Objectives 582
Overview 583
Dynamic Segmentation 584
Dynamic Segmentation 585
User Based Tunneling Components 586
User Based Tunneling Components 587
Understanding User Based Tunneling 588
UBT Authentication, Role Assignment, and Redirection 589
UBT Authentication, Role Assignment, and Redirection 590
Tunneled Traffic Infrastructure Requirements 591
Tunneled Traffic Behavior – Multiple clients 592
UBT Verification Commands 593
UBT Verification Commands 594
User Based Tunneling QoS 595
UBT Verification Commands 596

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Question #1 597
Downloadable User Roles 598
User Role Review 599
Downloadable User Roles (DUR) – Flow 600
Downloadable User Roles (DUR) – Flow 601
Question #2 602
Additional Content 603
MAC and 802.1X Authentication 604
MAC and 802.1X Authentication 605
Port-Based vs User-Based Tunneling 606
Port-Based vs User-Based Tunneling 607
UBT Tunnel Establishment 608
UBT Tunnel Establishment 609
Tunneled Traffic – PAPI communication 610
Tunneled Traffic – PAPI communication 611
UBT Configuration 612
UBT Configuration 613
Wireshark Analysis 614
UBT and QoS – Wireshark capture 615
UBT Verification Commands – AOS-CX 616
UBT Verification Commands – Aruba Gateway 617
Downloadable User Roles 618
Downloadable User Roles (DUR) – Configuration 619
Switch Configuration: Certificate Validation 620
Switch Configuration: Clearpass Account 621
Deployment Using NetEdit 622
Dynamic Segmentation Configuration using NetEdit 623
Dynamic Segmentation Configuration using NetEdit 624
Dynamic Segmentation Configuration using NetEdit 625
Dynamic Segmentation Configuration using NetEdit 626
Lab Activity 627
Module 12: Network Security 628
Objectives 629
Overview 630
Access Control List 631
Applying Access Control List 632
Access Control List – Counting action 633
Access Control List – Counting action 634

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Access Control List – Logging action	635
Access Control List – Software and Hardware resources	636
Access Control List – Software and Hardware resources	637
Access Control List – Software and Hardware resources	638
Access Control List – Software and Hardware resources	639
Classifier Policy	640
Classifier Policies	641
Classifier Policies	642
Classifier Policies	643
Classifier Policies	644
Applying Classifier Policy	645
Classifier Policy – Verification	646
Classifier Policy – Verification	647
Classifier Policy – Verification	648
Question #1	649
Control Plane Policing	650
Control Plane Policing	651
Control Plane Policing	652
Control Plane Policing – Configuration	653
Control Plane Policing – Configuration	654
Control Plane Policing – Configuration	655
Control Plane Policing – Configuration	656
Control Plane Policing – Verification	657
Local user and group management	658
User-defined User Groups	659
User-defined User Groups	660
User-defined User Groups	661
Management access using TACACS+	662
Remote Authentication using TACACS+	663
Remote Authorization using TACACS+	664
Remote Authorization using TACACS+	665
Remote Accounting using TACACS+	666
Remote Accounting using TACACS+	667
DHCP Snooping and ARP Inspection	668
DHCP Snooping	669
DHCP Snooping configuration	670
DHCP Snooping verification	671
Dynamic ARP Inspection	672

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Dynamic ARP Inspection	673
Dynamic ARP Inspection	674
Dynamic ARP Inspection	675
Question #2	676
Additional Content	677
Access Control List	678
Access Control List	679
Access Control List	680
Address and Port Object Groups	681
Address and Port Object Groups	682
Access Control List – Verification Commands	683
Access Control List – Logging configuration	684
Local AAA	685
Local AAA	686
Local AAA	687
Built-in User Groups	688
Built-in User Groups	689
Remote AAA using TACACS+	690
Communication with TACACS+ server	691
Communication with TACACS+ server	692
Lab Activity	693
Module 13: IPv6	694
Objectives	695
Overview	696
Overview	697
IPv6	698
IPv6 Header	699
IPv6 Header	700
IPv6 Extension Header	701
IPv6 Neighbor Discovery	702
ICMPv6	703
ICMPv6	704
ICMPv6 – Neighbor Discovery (ND)	705
ICMPv6 – Neighbor Discovery (ND)	706
Manual IPv6 Addressing	707
IPv6 Address - Manual Configuration	708
IPv6 Address AOS-CX Verification	709
Duplicated Address Detection (DAD)	710

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Question #1	711
Dynamic IPv6 Addressing	712
Dynamic Configuration – Router Advertisement	713
Dynamic Configuration – Router Advertisement	714
Dynamic Configuration – RA Address Allocation	715
Dynamic Configuration – RA Address Allocation	716
Stateless Address Autoconfiguration (SLAAC)	717
Interface-ID generation	718
Interface-ID generation – Privacy Extension	719
Autoconfigured Address States and Lifetimes	720
Autoconfigured Address States and Lifetimes	721
SLAAC Configuration	722
SLAAC Verification	723
SLAAC Verification	724
SLAAC Verification	725
DHCPv6	726
DHCPv6	727
SLAAC and Stateless DHCPv6 - Configuration	728
SLAAC and Stateless DHCPv6 - Verification	729
Stateful DHCPv6 configuration	730
Stateful DHCPv6 configuration	731
Stateful DHCPv6 configuration	732
Router Advertisement Verification	733
DHCPv6 Pool Verification	734
Client verification	735
Question #2	736
IPv6 Routing	737
IPv6 Static Routing	738
OSPFv2 vs OSPFv3	739
OSPFv2 vs OSPFv3	740
OSPFv2 vs OSPFv3	741
OSPFv3 LSAs	742
OSPFv3 LSAs	743
Router LSA (Type 1)	744
Router LSA (Type 1)	745
Network LSA (Type 2)	746
Link LSA (Type 8)	747
Link LSA (Type 8)	748

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Link LSA Flooding	749
Intra-Area Prefix LSA (Type 9)	750
Intra-Area Prefix LSA (Type 9)	751
Intra-Area Prefix LSA (Type 9) – Referencing LSA Type 1	752
Intra-Area Prefix LSA (Type 9) – Referencing LSA Type 1	753
Intra-Area Prefix LSA (Type 9) – Referencing LSA Type 2	754
Intra-Area Prefix LSA (Type 9) – Referencing LSA Type 2	755
Basic OSPFv3 configuration	756
OSPFv3 Verification	757
OSPFv3 Verification	758
OSPFv3 Verification	759
Question #3	760
Additional Content	761
Recommended Lecture Order	762
Recommended Lecture Order	763
IPv6 Features	764
IPv6 Features	765
IPv6 Address Representation	766
IPv6 Address Representation	767
IPv6 Address Representation	768
IPv6 Prefix Length Notation	769
IPv6 Address Types	770
IPv6 Address Types - Unicast	771
IPv6 Unicast Addresses	772
IPv6 Unicast Addresses	773
IPv6 Unicast Addresses	774
IPv6 Unicast Addresses	775
IPv6 Address Types - Multicast	776
IPv6 Multicast Address	777
IPv6 Multicast Address	778
IPv6 Multicast Address	779
IPv6 Multicast Address	780
IPv6 Multicast Address	781
Mapping to Ethernet MAC address	782
IPv6 Address Types	783
IPv6 Anycast	784
IPv6 Address Configuration	785
IPv6 Address Host Verification	786

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

Autoconfigured Address States and Lifetimes	787
Router Advertisement - Wireshark analysis	788
Router Advertisement - Wireshark analysis	789
SLAAC and Stateless DHCPv6 Process	790
SLAAC and Stateless DHCPv6 Process	791
SLAAC and Stateless DHCPv6 – RA DNS option	792
Router Advertisement - Wireshark analysis	793
Stateful DHCPv6	794
DHCPv6 – Relay agent	795
DHCPv6 – Relay agent	796
Router Advertisement - Wireshark analysis	797
OSPFv3 – Messages (Message Header)	798
OSPFv3 – Hello Packet	799
OSPFv3 – Database Descriptor Packet	800
OSPFv3 – LSA Headers	801
OSPFv3 – LSA Headers	802
OSPFv3 - Options and Prefix Options Fields	803
OSPFv3 - Options and Prefix Options Fields	804
IPv6 Access Control Lists	805
Basic IPv6 ACL configuration	806
Basic IPv6 ACL verification	807
Lab Activity	808

Rev 20.41 | © Copyright 2020 Hewlett Packard Enterprise Development LP | Confidential – For Training Purposes Only

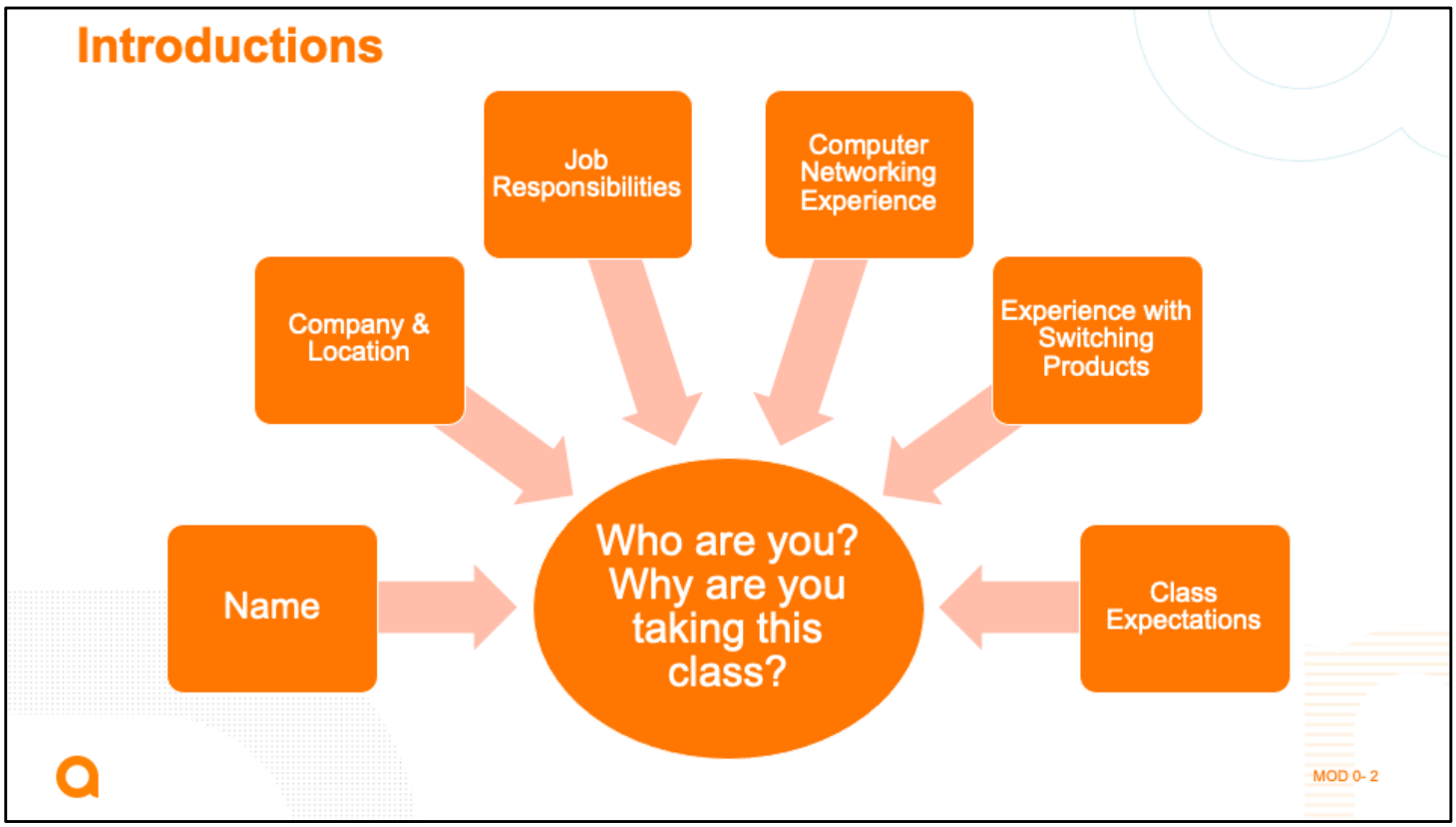
Rev # 20.41

Course Introduction

Aruba Advanced Switching Troubleshooting, and Solutions (ASTS)



aruba
a Hewlett Packard Enterprise company


Introductions
Job Responsibilities
Computer Networking Experience
Company & Location
Experience with Switching Products
Name
Who are you? Why are you taking this class?
Class Expectations
MOD 0- 2

Logistics

Introduction

- Breaks
- Lunch
- Restrooms



MOD 0- 3

Course Objectives

- Describe diagnostic principles
- Use a proven problem-solving methodology
- Describe REST API, NAE, IP SLA, SNMPv3 and NetEdit troubleshooting tools
- Configure, validate and troubleshoot Network Virtualization features
  - VSF
  - VSX
- Configure and validate PBR
- Deploy and troubleshoot multi-area OSPF
- Deploy and troubleshoot BGP

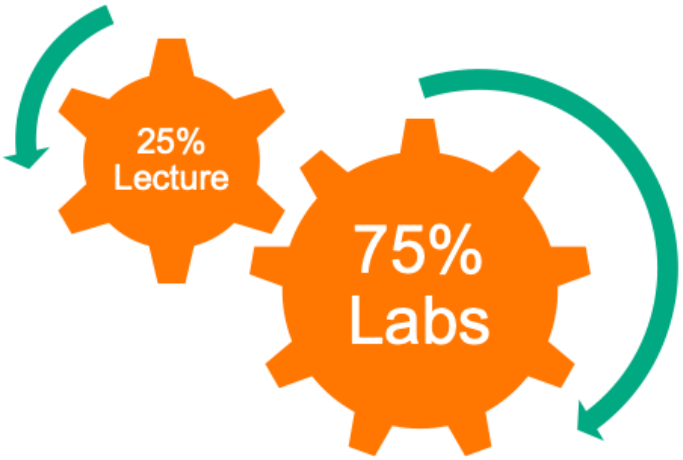


Course Objectives

- Deploy and troubleshoot Route Redistribution
- Deploy and troubleshoot VRF and Route Leaking
- Deploy and troubleshoot Multicast
- Deploy and troubleshoot QoS
- Deploy and troubleshoot Dynamic Segmentation
- Deploy and troubleshoot Network Security
- Deploy and troubleshoot IPv6



Course Design



MOD 0- 6

Agenda – Day 1

1 – Plan the Wired Networks Solution

2 - Troubleshooting

3 – Basic networking with Aruba Solutions

4 - VSX-VSF Layer 2 Technologies



MOD 0- 7

Agenda – Day 2

- 5 – Layer 3 Routing - OSPF
- 6 – BGP
- 7 – Route Redistribution



Agenda – Day 3

- 8 – VRF and Route Leaking
- 9 - Multicast
- 10 - QoS



Agenda – Day 4

- 11 – Dynamic Segmentation
- 12 – Network Security
- 13 – IPv6



Agenda – Day 5



Troubleshooting Tickets


Lab Activity


Rev # 20.41
aruba
Plan the Wired Networks Solution
Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)



After completing this module, you will be able to plan a successful wired network solution deployment.

You will be able to use a 5-step process to identify and document customer requirements, and to use network questionnaires to gather customer requirements.

You will be able to create a Proof of Concept plan, from which you will derive a set of golden configurations.

You will be able to use these golden configurations to test your solution.

Overview

Plan the wired solution



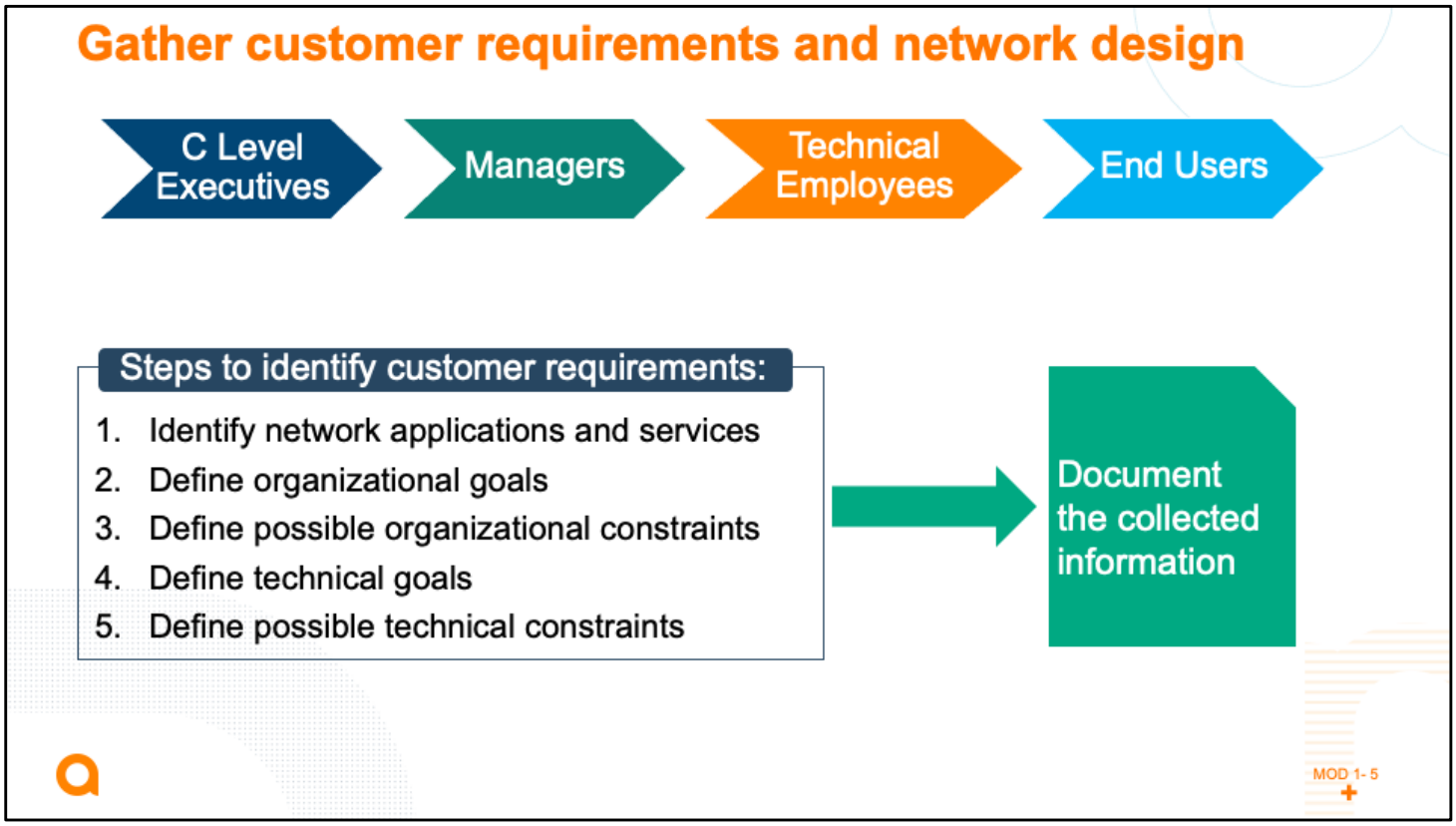
MOD 1 - 3

It is dangerous to deploy a network without thorough planning. This module will provide guidelines and a methodical process for gathering customer requirements and current-state network information, so that you can develop the desired future state solution.

Plan the Wired Solutions

Gather customer requirements, information and POC

aruba
a Hewlett Packard Enterprise company



Information gathering for network design projects requires information from many different sources within the organization. Usually, someone at the executive level within the organization initiates a new or network upgrade project. Other “C” level executives, such as CTO, CEO, or CFO, may also need to “buy-in” and support the project, depending on the level of complexity and cost of the project. Most important are the technical employees – the network administrator or manager – who will need to be involved in the technical aspects of the project. Lastly, if the project will affect end-users, then you will need to provide appropriate training and information to the ultimate end users of the system. It is good to understand what the customer wants to do with the network in terms of the business reasoning behind it, which can enable the designer to discuss the Aruba solutions.

As we mentioned, to obtain customer requirements, you must not only talk to network engineers, but also talk to business unit personnel and company managers.

The steps to identify customer requirements are as follows:

Identify network applications and services. You need to identify current and planned applications and determine the importance of each application. Planned infrastructure services should also be gathered. Network services include security, quality of service (QoS), network management, high availability, mobility, and virtualization.

Define the organizational goals. For organizational goals, you should identify whether the

company's goal is to improve customer support, add new customer services, increase competitiveness, or reduce costs. It might be a combination of these goals, with some of them being more important than others.

Define the possible organizational constraints. Organizational constraints include budget, personnel and internal policy.

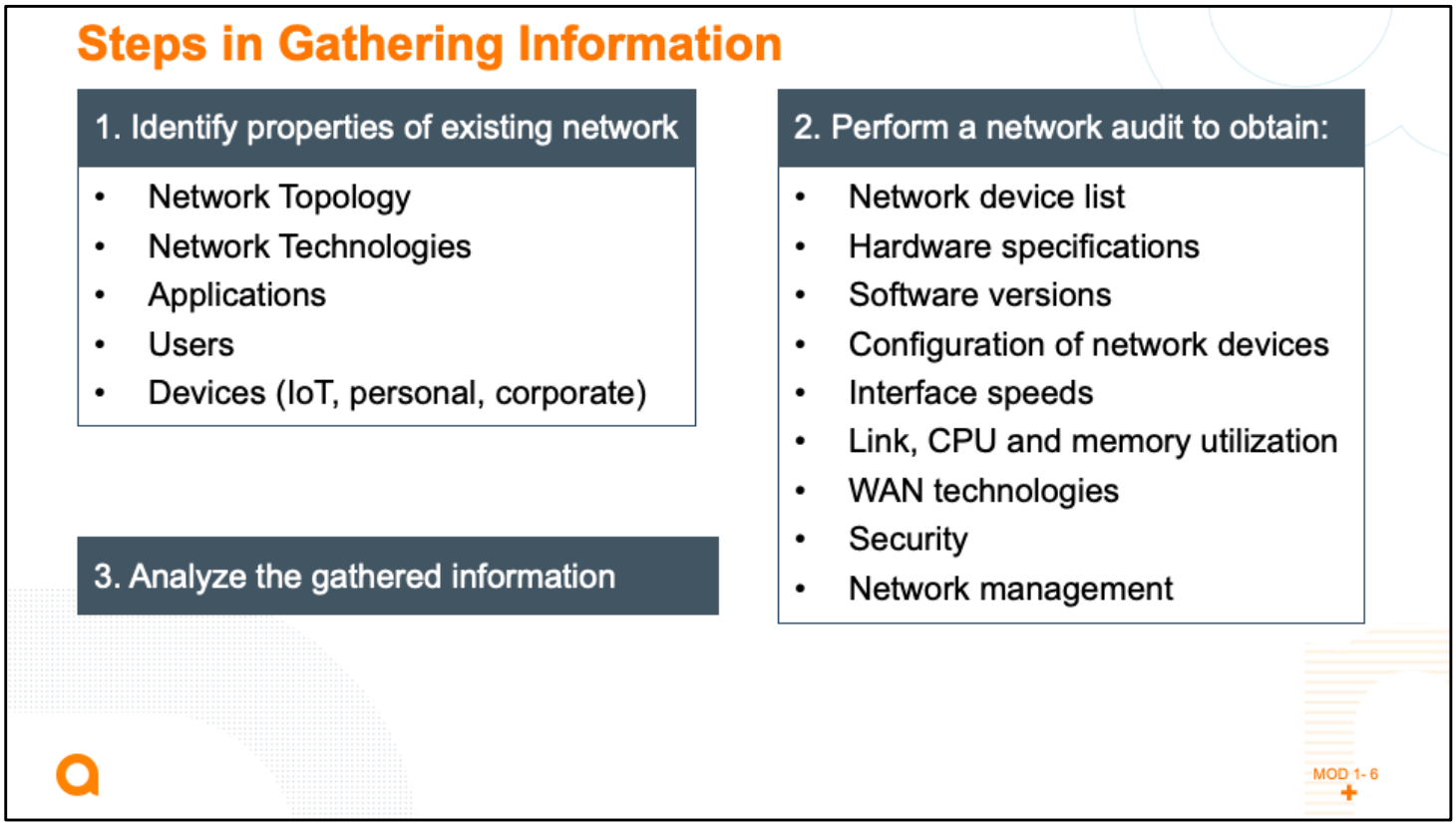
Define the technical goals. Technical goals support the organization's objectives and the supported applications and services. Technical goals include:

- Improve the network's response-time throughput
- Decrease network failures and downtime (high availability)
- Simplify network management
- Improve network security
- Improve reliability of mission-critical applications
- Modernize outdated technologies (technology refresh)
- Improve the network's scalability

Define the possible technical constraints. Technical constraints include:

- Existing wiring does not support new technology.
- Bandwidth might not support new services.

Is important to document all this information for future reference.



Before arriving at a site that has an existing network, you should obtain all existing documentation. Sometimes no documented information exists. Be prepared to use tools to obtain information and get access to log in to the network devices to obtain information. Here are the steps for gathering information:

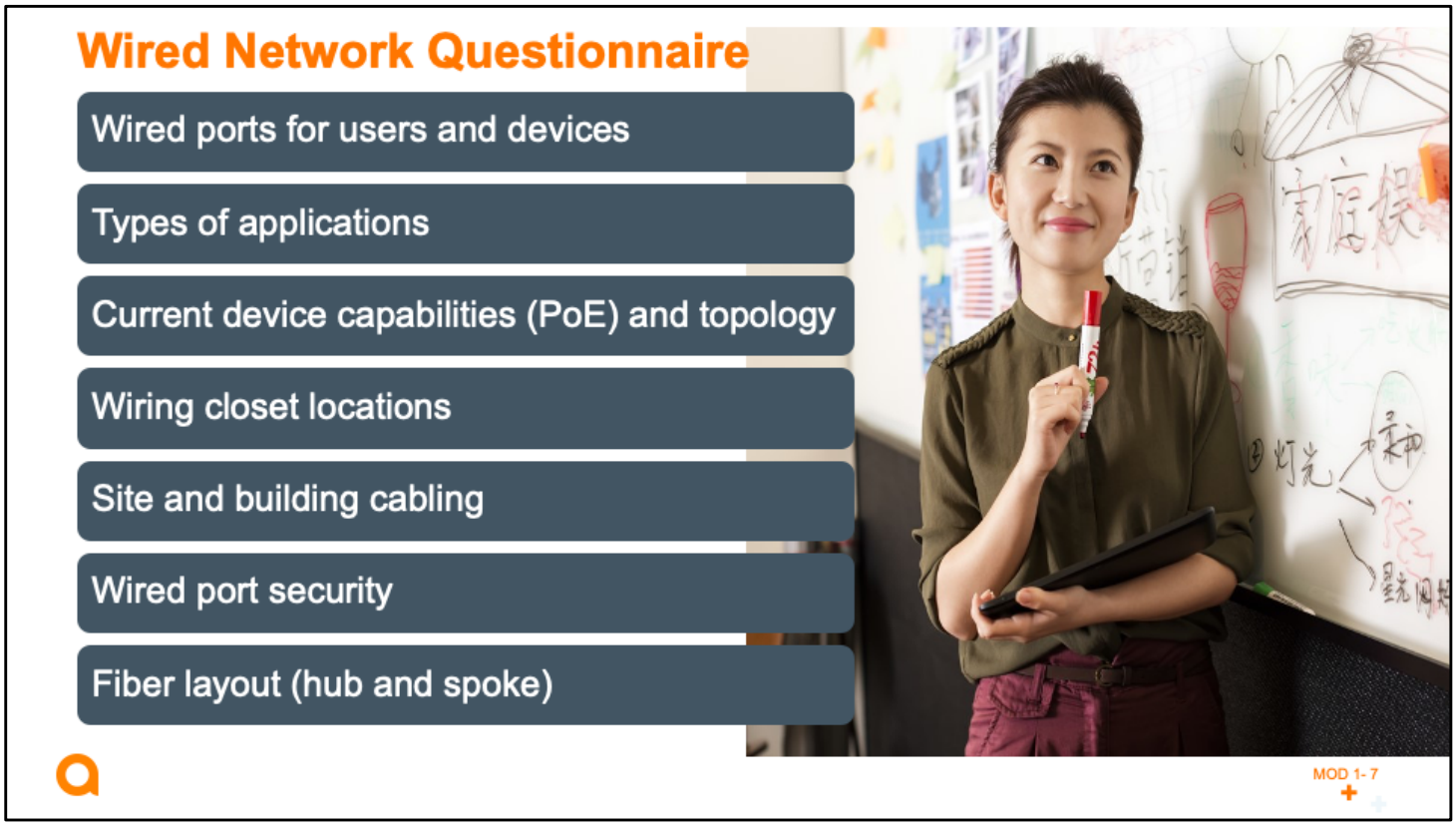
Identify properties of the existing network: network topology, technologies, applications, users and devices. Use existing documentation and organizational input.

Perform a network audit that adds detail to the description of the network. The network audit should provide the following information:

- Network device list
- Hardware specifications
- Software versions
- Configuration of network devices
- Interface speeds
- Link, CPU, and memory utilization
- WAN technology types and carrier information
- Security
- Network Management

Analyze the gathered information.

When gathering existing documentation, you look for site information such as site names, site addresses, site contacts, site hours of operation, and building and room access. Network infrastructure information includes locations and types of servers and network devices, size of each site, data center and closet locations, LAN wiring, WAN technologies and circuit speeds, and power used. Logical network information includes IP addressing, routing protocols, network management, and security access lists used.



You should also investigate the customer's wired network needs by asking how many wired ports the customer requires for users and devices as well as what types of applications run over the network. Other important questions to ask the customer include device capabilities (including PoE), logical topology, physical wiring closet locations, and site cabling. Often, customers will not know the answers to your questions - providing additional motivation to prepare up-to-date documentation.

Typical questions to ask the customer about the wired network are how many wired ports need support, including ports for wired users and other devices such as printers and Internet of Things (IoT) devices. You should also ask what applications users run on the wired network and whether the customer can share information about the current topology and device capabilities. You will also need to assess the physical requirements. In addition to closet locations, you need to gather information about the site's cabling. This will help you to select the correct switches and transceivers, and it might also affect whether you propose switches that support Smart Rate ports.

You should also keep in mind that information that the customer provides about the topology might not be up-to-date. If the customer is open to you passively collecting some information about the network, you could install an evaluation copy of AirWave or use a tool like SolarWinds to map the network.

Wired Network Questionnaire:

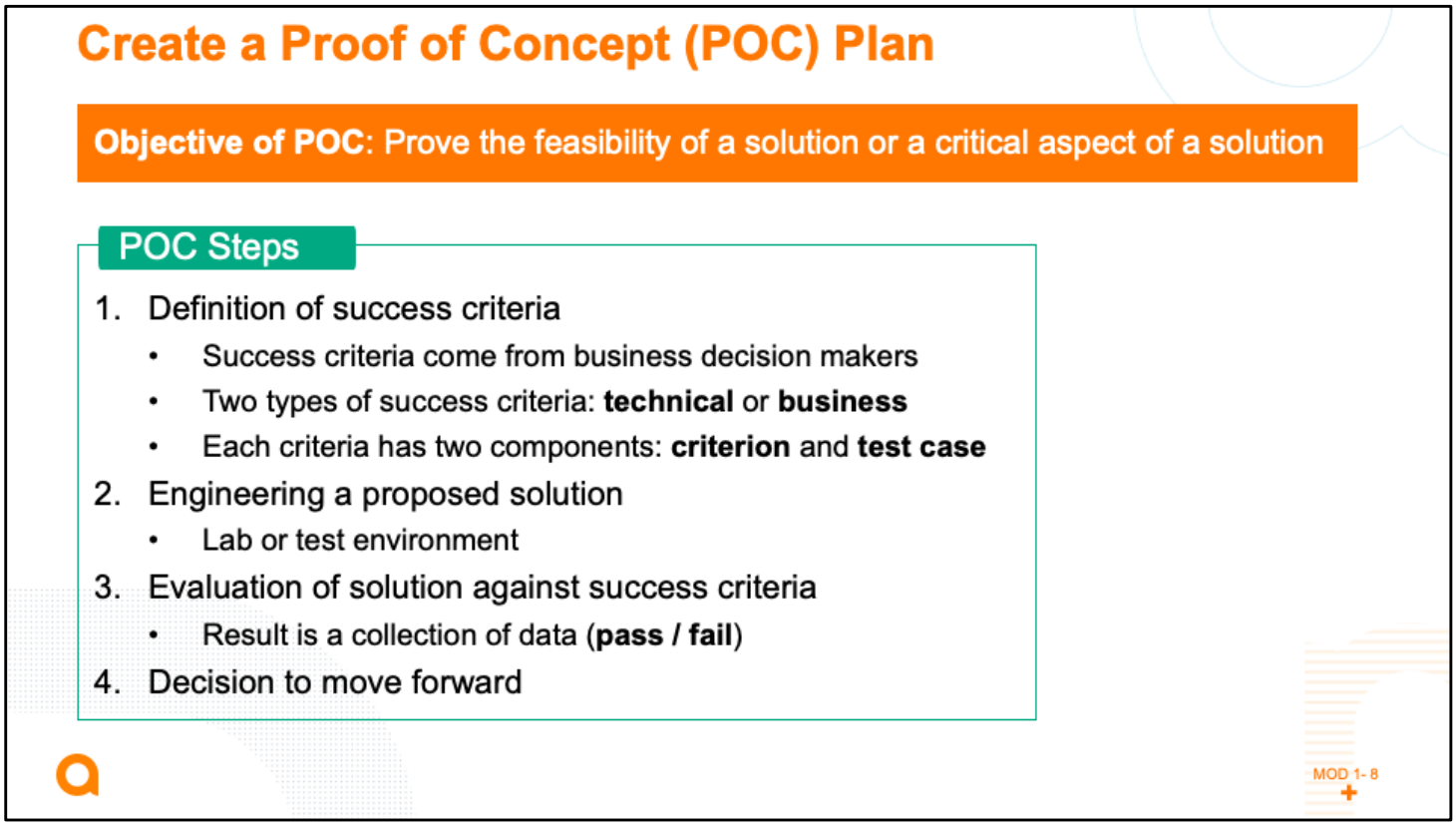
How many wired ports for users and devices?

What types of applications are used on the network?

What are the current device capabilities and topology?

Where are closets located?

How is the site cabled?



Proof of Concept (POC)

It probably sounds intuitive, but the point of a POC is to prove the feasibility of a solution, or the feasibility of a critical aspect of a solution. Typically, a POC is trying to answer questions like the ones below:

Will this technology meet our needs?

Will this product perform as expected?

Will the ultimate solution be feasible?

For a POC to be successful, it must be broken up into the following steps:

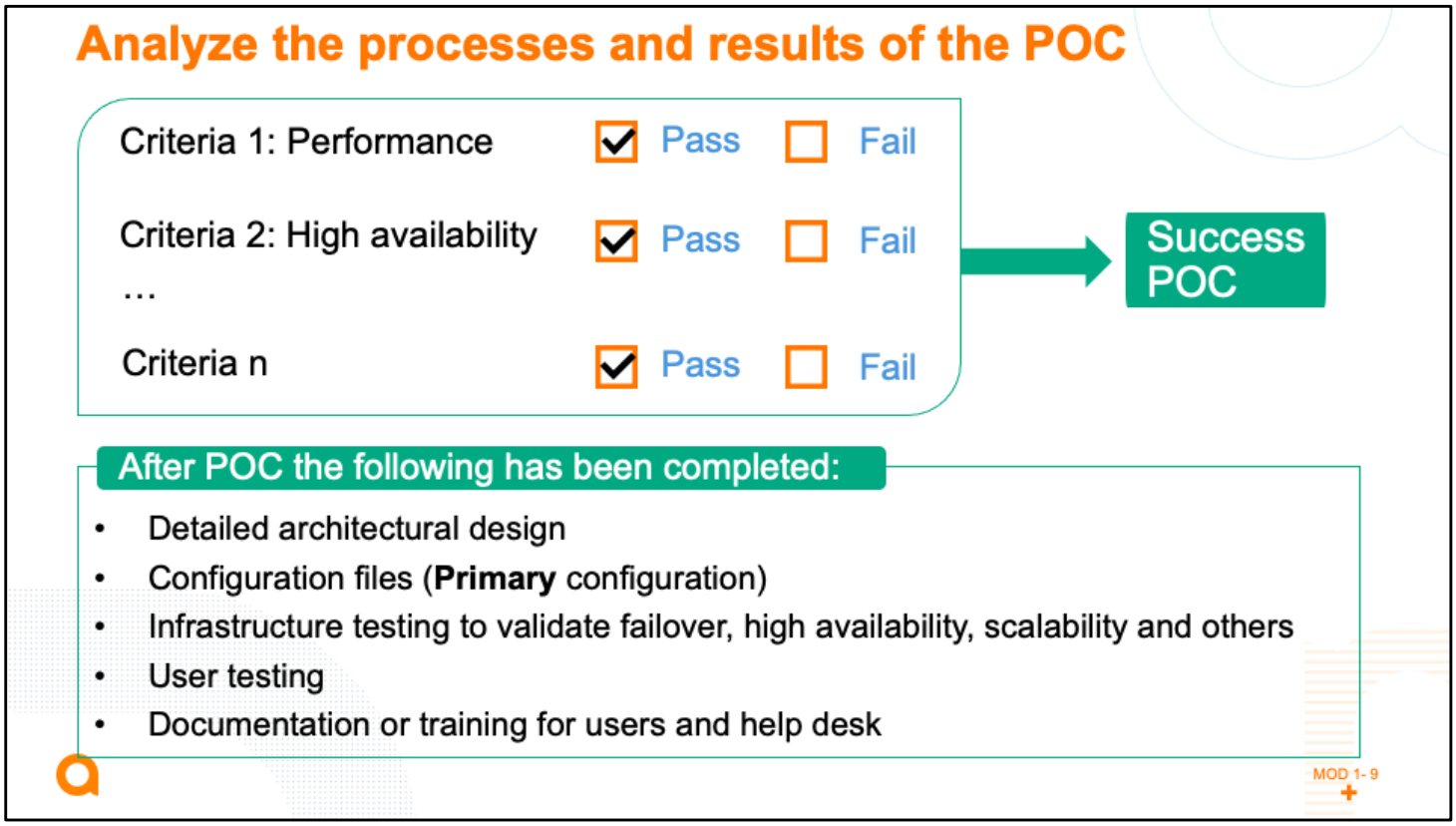
Definition of success criteria. Clearly and specifically defining success criteria will set you up for success. Appropriate success criteria typically come from business decision makers, IT, end-users, or end-user representatives. There are two types of success criteria: technical or business. Each success criteria will have two components: a criterion and a test case. Some criterion, particularly the technical ones, will easily inspire a test case, while others may require some discussions and validation.

Engineering of proposed solution. Once the success criteria and associated test cases are defined, the POC solution can be engineered. This is usually conducted in a lab or test

environment to not have an impact on production systems. The team responsible will typically spend several days to a few weeks configuring the POC environment to achieve the defined success criteria and associated test cases.

Evaluation of solution against success criteria. Once the solution is engineered, it can be evaluated against the test cases defined by the success criteria. The net result is a collection of data (pass / fail for the test cases).

Decision to move forward. The decision makers will access the results from the evaluation portion coupled with implications about the future solution and decide to move forward.



For a POC, a strict methodology focused on the success criteria is critical to achieving a useful result from a POC. If the POC can answer all the questions set out by the success criteria (all pass, all fail, or a mix of pass and fail), then it is successful. A POC is not successful, when the success criteria were not appropriate, or the test cases not detailed enough.

After the POC the following has been completed:

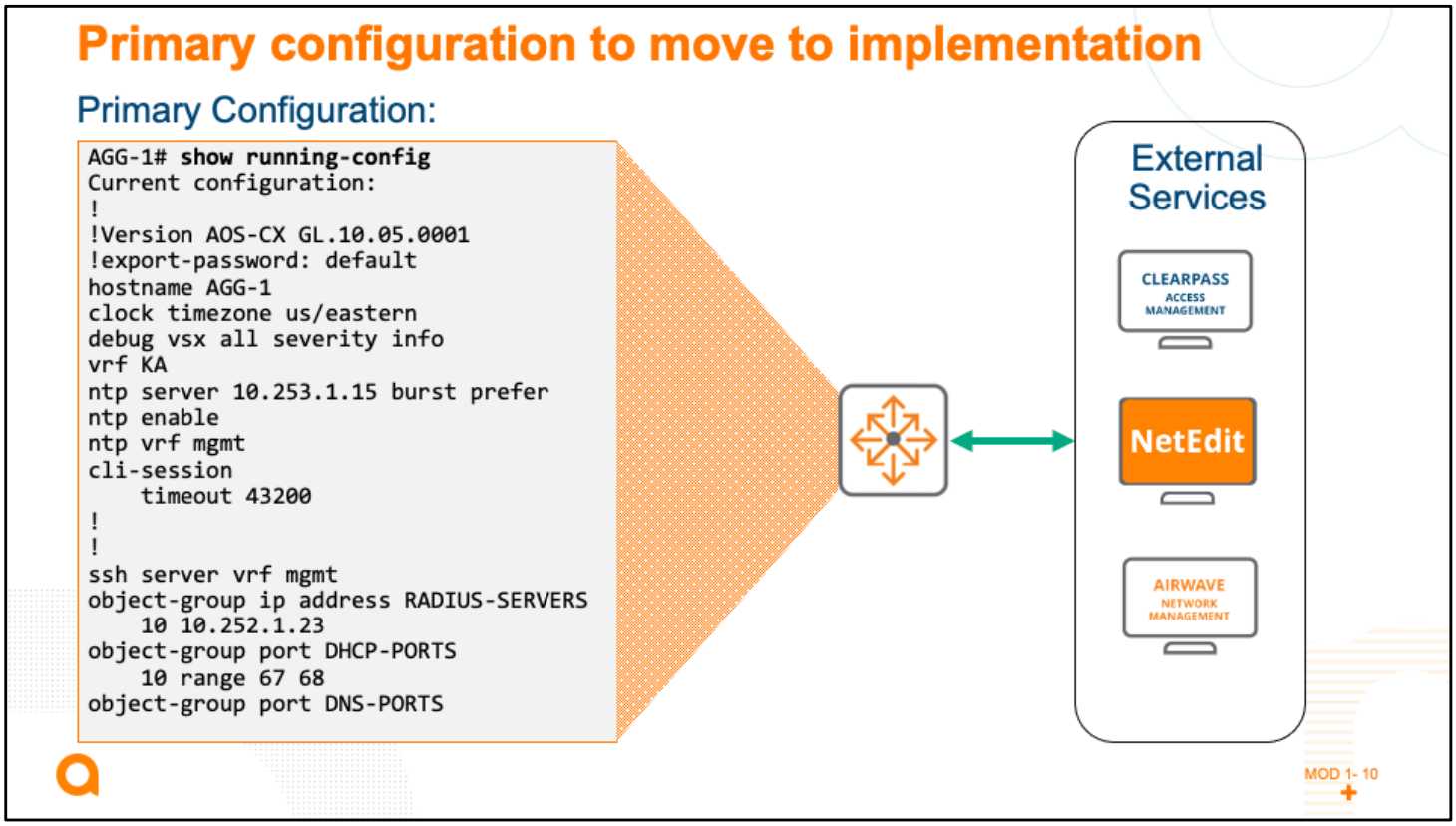
Detailed Architectural Design based on an assessment of business and technical requirements

Configuration files

Infrastructure testing to validate failover, high availability, and possibly scalability

User testing and iterative feedback to optimize the user experience

Documentation or training for users and the help desk



The primary configuration is a result of a successful POC. This configuration is implemented to a subset of users which have stressed the solution by using the 100% of its functionality. This also includes external services that can be used with the switching solution to authenticate or maintain the network, examples include Aruba ClearPass, NetEdit and AirWave.


Lab Activity
Lab 01


Rev 20.41
aruba
a Hewlett Packard Enterprise company
Troubleshooting
Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)



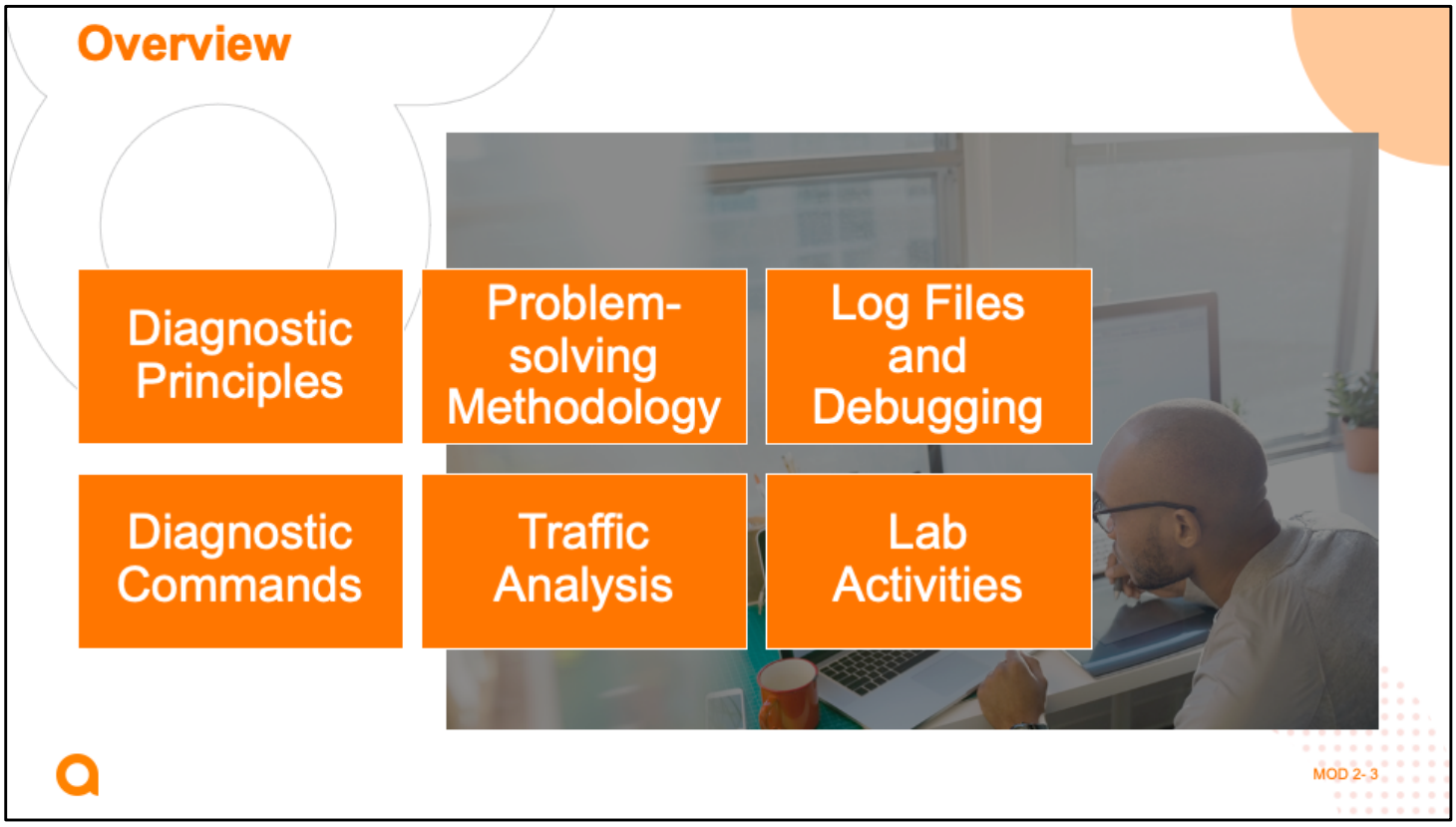
This entire course is focused on elevating your ability to deploy and troubleshoot Aruba Campus LAN solutions – a keen sense of diagnostic ability and critical thinking skills are vital.

After completing this module, you will be able to describe fundamental diagnostic principles. This includes choosing when to use a more involved troubleshooting methodology, and when such an involved methodology may be less important. It also includes the components that you must bring to bear for effective, efficient troubleshooting.

|You will also be able to describe a proven framework and methodology to effectively troubleshoot wired networks.

|Then you will explore how to use several diagnostic CLI-based commands and techniques to help you troubleshoot general Campus LAN-based issues. You will leverage this knowledge throughout the course, and your career.

|Finally, you will learn about several techniques you can use to perform traffic analysis – where to capture packets, what packets to capture, configuration and validation options, along with some helpful tips for setting up and using Wireshark.



You will leverage the information in this module throughout the rest of this course – and perhaps the rest of your career! Your mastery of these core concepts will enable you to maximize your diagnostic and deployment efficiency.

You begin with diagnostic principles – there are certain scenarios when you can be a bit more relaxed in your approach, and others when a very organized, proven troubleshooting methodology is vital – you should know the difference. You learn about troubleshooting principles and zones, and the components that you will bring to bear as an effective diagnostician.

Next you learn about a proven problem-solving methodology – vital for issues with moderate or more complexity, or those with which you are less familiar. Certain commands, log files, and techniques are vital to implementing this methodology. You will learn to use debugging and logging tools more effectively, along with event and account logging, before exploring the use of the service OS.

Then you learn to use some general diagnostic commands related to CPU, memory, self-tests, capacities, and the switch environment. You will also explore diagnostic tools and support files.

Finally, you will explore some more advanced options for traffic analysis – where and when to capture packets, for both local and remote captures, and both basic and policy-based

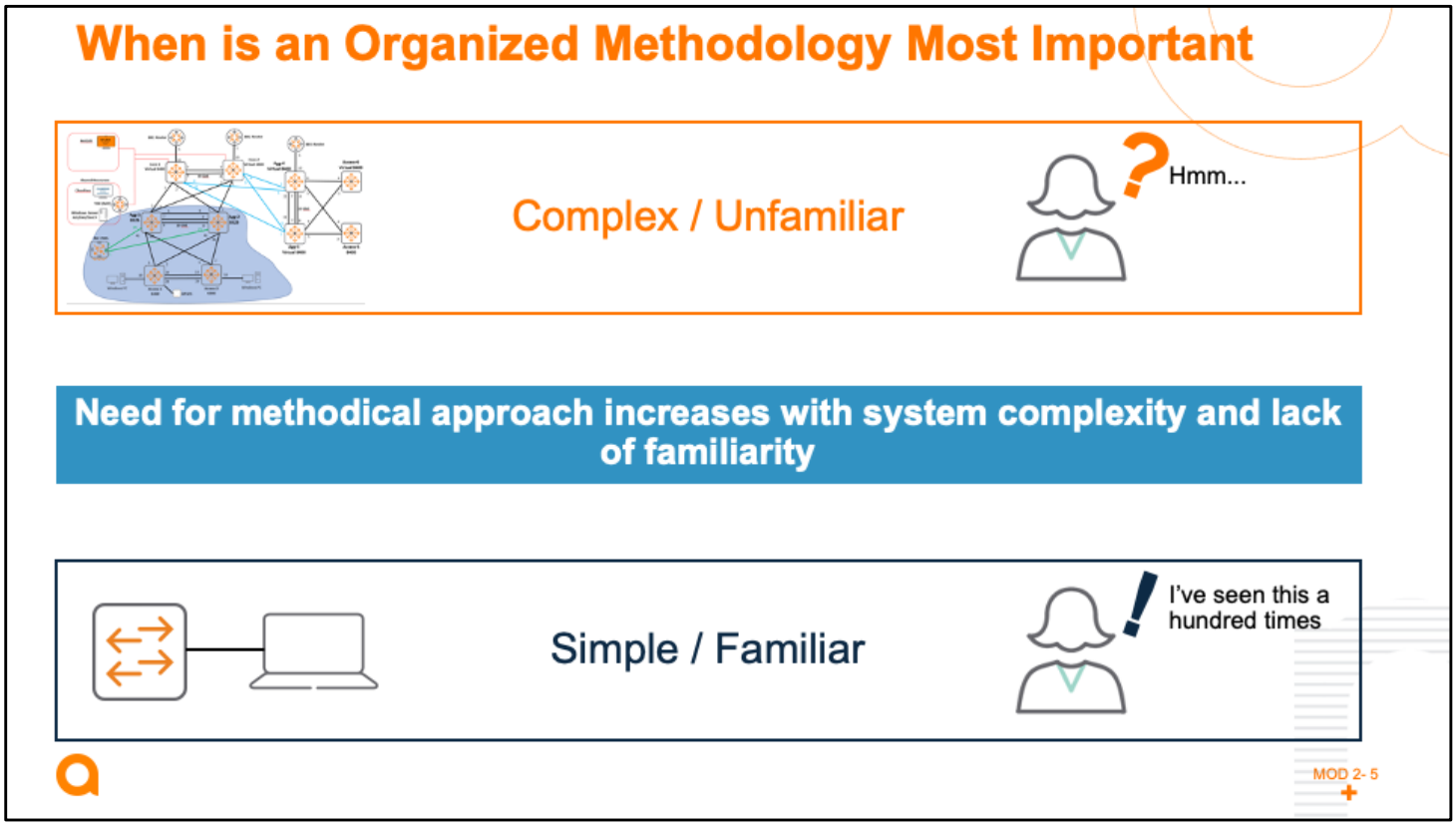
mirroring sessions. You will review some configuration and validation options, before exploring some tips for setting up and using Wireshark.

The module ends with some hands-on lab activities.

Diagnostic Principles

Methodology, Troubleshooting principles, Troubleshooting zones

aruba
a Hewlett Packard Enterprise company

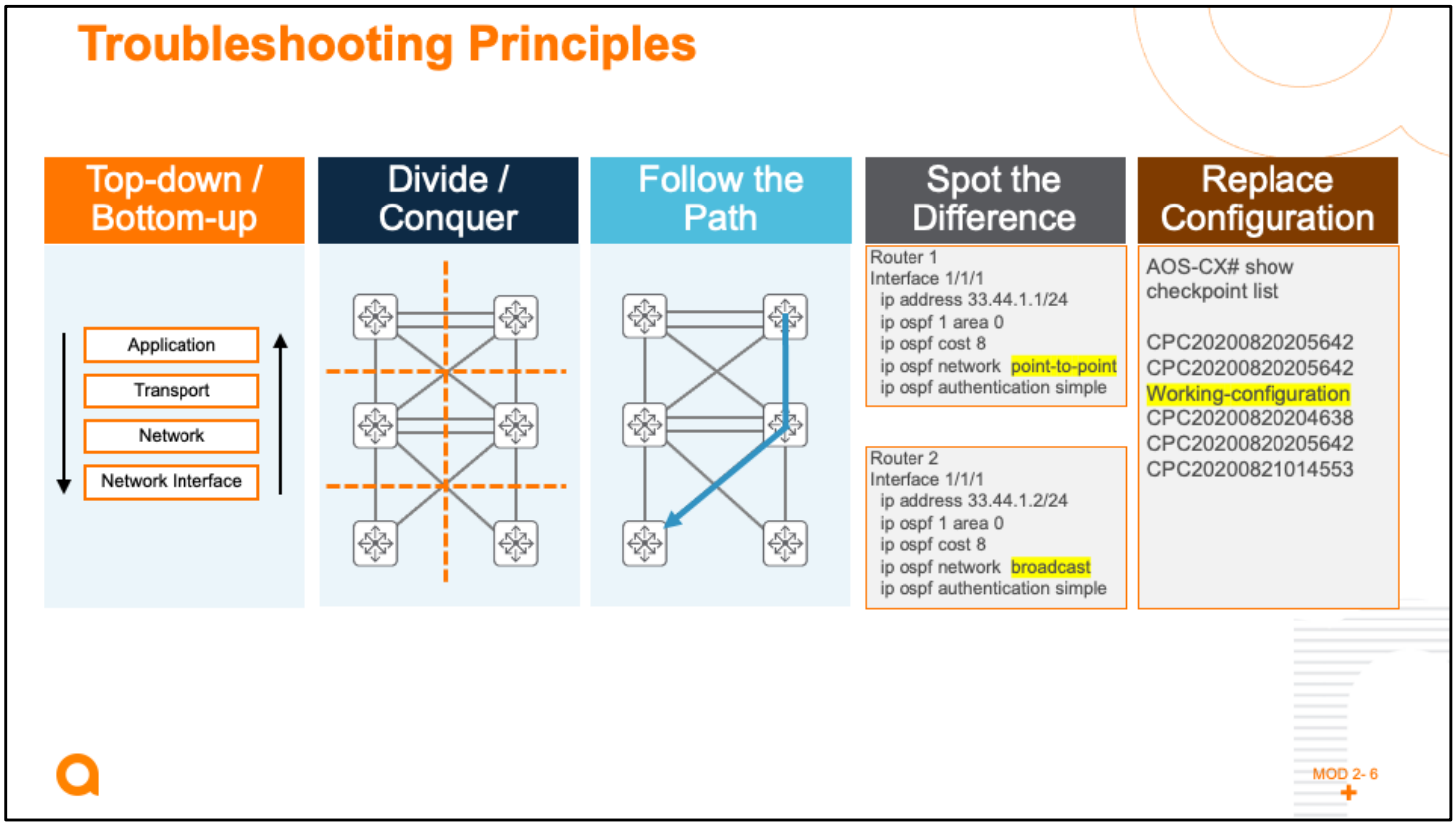


If you are troubleshooting very simple problems, or those with which you have great familiarity, an involved troubleshooting methodology is less important. If you have been working in the same environment for a long time, you become familiar with most of the issues that arise, and so you may rarely use a more involved, systematic approach. This can lead you to believe that such an approach is not necessary or forget how to apply such an approach.

As the complexity of the system increases, it becomes vital to adopt a more scientific approach to troubleshooting. It seems counter-intuitive at first, but the experienced engineers know that a slow, methodical approach often provides quicker resolution than a hurried, random method of part-swapping and configuration changes.

Note that even if you have high familiarity and a simple system, it is often best to use an organized methodology, if for no other purpose than to maintain readiness and build good habits for that day when you do have a more challenging scenario.

The purpose of this section is to propose some effective troubleshooting principles and methods.



Before reviewing an effective troubleshooting methodology, there are several troubleshooting principles to be aware of. You can choose among them and use them in conjunction with one another – depending on the scenario.

Many engineers might use a “Top-down” or “bottom-up” approach. If you think the issue is most likely related to the application or supporting software, you might use a top-down approach. You start by assessing application status, configuration, logs and error messages, gradually working your way down to issues related to TCP or UDP at the Transport layer, then on to routing/network issues, saving the physical and data link layer issues related to network interfaces for last.

Alternatively, you might use a bottom-up approach if you suspect it might save you some steps.

The divide and conquer approach is very often quite effective, especially if implemented properly. Take a simple example of a chassis-based switch that will not properly boot up. You might suspect that one of the line cards is faulty. One approach is to remove one line card, then power it on and check for symptoms. Then replace another line card, and so on. You might swap out or eliminate 7 lines cards, only to find out that it wasn’t related to line cards at all. Perhaps a better approach is to divide the system into two big pieces – “line cards” and “everything else”. Remove all line cards and see if the problem persists. If it does, you have determined that it is not a line card in one step instead of seven steps. If the problem is eliminated, you now know that it is a line card issue. Replace half of the line cards to see if the

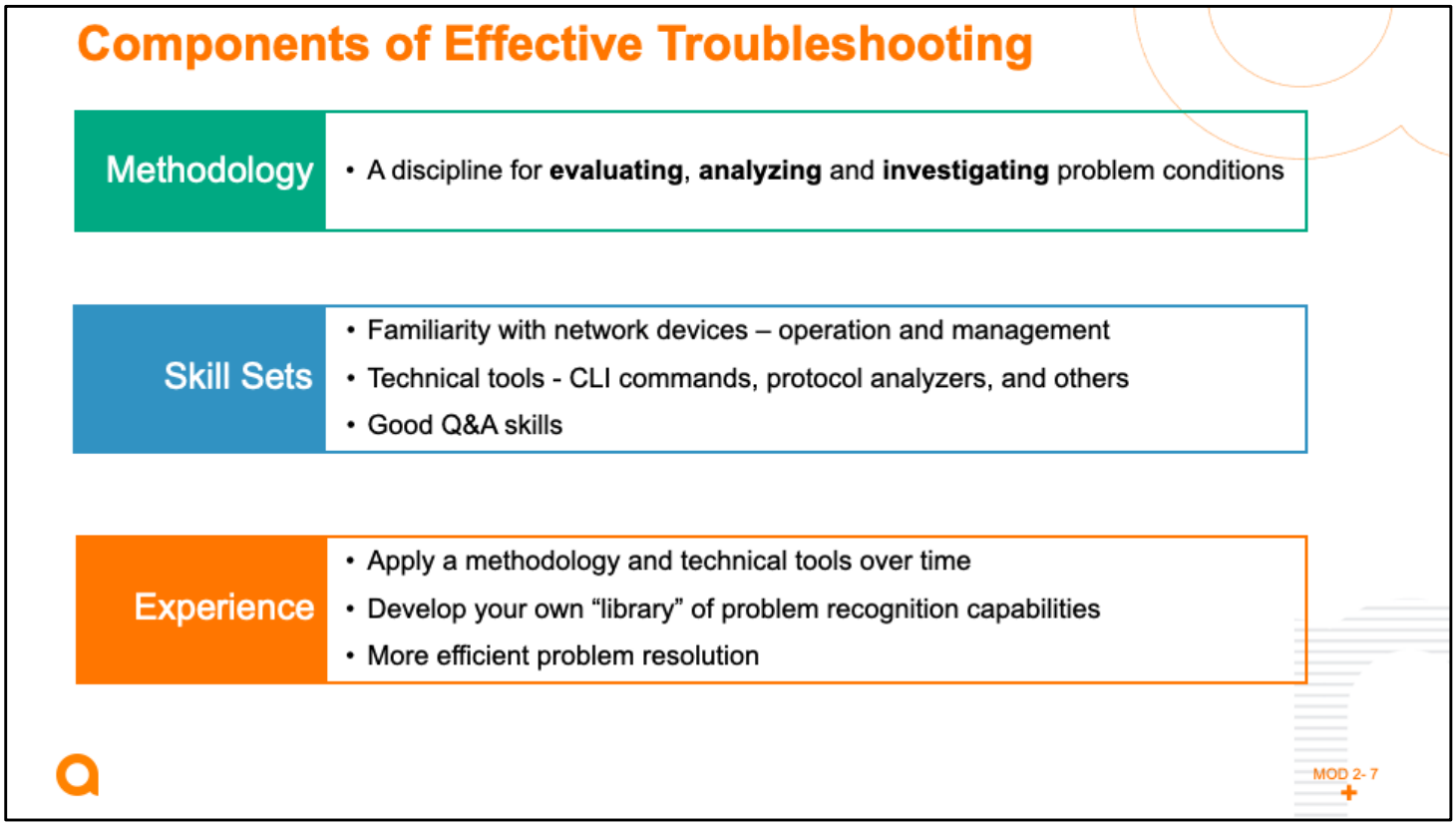
problem re-appears. If it does you know it is one of those line cards. A similar approach can be used on entire networking infrastructures, like the one shown in the figure.

The follow-the-path method asks you to “be the packet”. Determine the flow of that packet (show commands related to LLDP at Layer 1 and 2; spanning tree at Layer 2, Routing at Layer 3) and then follow that path. Where is the packet getting dropped? Your typical ping and traceroute commands are also of great value here. Just remember, many firewalls drop ping/traceroute traffic, so a ping failure may not necessarily mean that the path is bad.

Speaking of firewalls (and mobility controllers), packets may travel through several internal processes within that firewall or mobility controller. This is where solid “theory of operation” knowledge becomes vital – you are better-equipped to follow the data/packet flow and identify the misconfigured sub-system.

You can play the “spot the difference” game – one of these things is not like the other. The figure shows a simple example of comparing two router configurations. Why won’t they form an OSPF adjacency? Ahhh...they are configured for different network types. You can also use this approach to compare one system that is “known good” with another (identical or very similar) malfunctioning system. The difference between them will often lead you toward a solution.

You can also use the “replace components” technique. Instead of determining what is wrong with Router A, simply replace the router and restore the known-good configuration. Or perhaps replace the configuration with the last known-good configuration checkpoint. This gets the customer up more quickly than troubleshooting some internal router issue during a customer outage. Technical curiosity is a trait that many good troubleshooters share. It can also be a trap – the customer does not care about the technical details of their outages as much as eliminating the outage. Get the network up first, then figure out the exact nature of the problem.



The basics of troubleshooting any kind of networking issue might be summarized as "keep eliminating obvious causes until the real cause presents itself." But understanding what this means requires a systematic approach and real discipline when attempting to identify causes from symptoms and apply the right fixes or workarounds.

Troubleshooting is a skill that all networking professionals learn by trial and error. But skipping some of the more painful or obvious errors can make your learning somewhat less trying than it might be otherwise. The most important characteristic to cultivate when solving problems is calmness. If you can keep a clear head when things fail or start degrading seriously, you'll be better able to assess your situation and better equipped to solve whatever problems you discover.

Methodology

Developing problem-solving techniques is often an on-the-job process. Few of us can expect much along the lines of formal network troubleshooting training in our job positions for a number of reasons. These reasons may include:

The relatively fast pace of the day-to-day job tasks and challenges yields little time to pursue formal training on troubleshooting aspects such as technical tools like a protocol analyzer.

Few business environments provide the luxury of a "test lab" and the time to hone your skills where a progression of test problems can be examined, worked through, and resolutions tried out.

In the absence of a more ideal situation, a problem-solving methodology can increase the

effectiveness of support staff by standardizing the approach used to some extent. With a fairly modest amount of discipline, network technicians can improve their problem resolution efficiency in terms of the effort needed and the number of other people that must be directly involved.

Skill Sets

There are a variety of skill sets that can enhance a network technician's success in problem solving. Some of these skills are purely technical in nature. For instance, it is important to understand the fundamentals of how network devices operate and how they are managed. Having proficiency in reading logs or interpreting a protocol analyzer display are examples of having familiarity with the potential tools you may need to call upon from your "toolbox".

Other skills are much less technical, but still very important. As part of the problem investigation process, a network technician may need to talk with various levels of staff. The staff may include non-technical end-users and business unit managers to software and hardware vendor support people. Having sufficient interpersonal skills coupled with good investigative reporter-like skills can expedite the isolation of a problem and eliminate the "noise" that often conceals the real problem.

Proactive IT support groups tend to spend time on developing procedures and tools to facilitate problem resolutions. Some examples of technical tools used by the network technicians are:

Device logs—Archived instances of the logs as well as the current one may provide hints of where the problem may be. At the very least, familiarity with a log file's typical contents helps you differentiate normal from abnormal situations.

Device statistics and status information—Being able to determine the health of a system or the network is important for gathering the "vital" signs. This type of information can include anything from port statistics and CPU utilization to network reachability results.

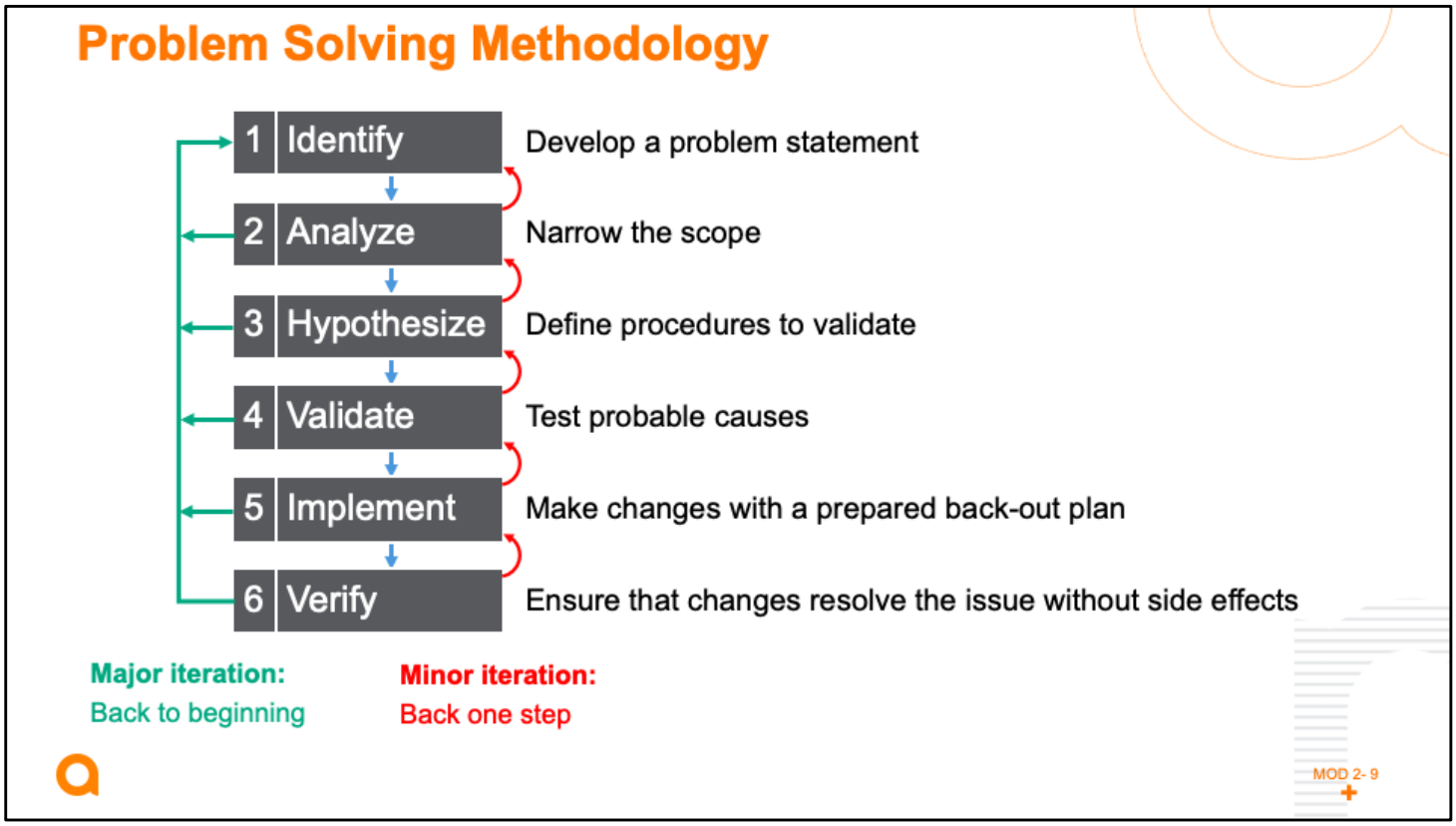
Protocol analyzer—Although this may not be a frequently used tool, it can be invaluable for examining what conversations are or are not occurring between communicating devices.

A problem-solving methodology that is refined over time can be very beneficial to network technicians. Being methodical and learning from the macro and micro levels of mistakes can help network technicians improve problem recognition capabilities and yield a more efficient application of a problem resolution process.

Problem-Solving Methodology

Example of a methodology

aruba
a Hewlett Packard Enterprise company



A problem-solving methodology is a process for managing problem resolution. Although there is no one specific model that may be useful for all problem situations, a general framework can provide guidelines and help ensure efficiency in the efforts made to solve a problem. Applying a methodology can improve the probability of a successful resolution. This graphic illustrates the framework for a general problem-solving methodology that has many applications in today's contemporary network environments.

There are six steps to the problem-solving methodology outlined here. The six steps are:

Identification—Understand and document the problem from both a user and technical perspective. Sometimes it is possible to lose sight of what the potential problem is before searching for a cause when we don't consider multiple perspectives. This step also helps to define success criteria for final verification steps, from a user point of view. You may have resolved the main issue, but their may be lingering aspects that affect the user – "Yes, the network is up, but I still can use my xyz application".

Analysis—Evaluate the situation by investigating, using problem resolution tools, product documentation, and user input.

Hypothesis—Develop possible resolutions based on the analysis and document a possible resolution. This documentation may be fairly informal, but it is important to be able to explain it in writing. Doing so can reveal a hypothesis that is unclear and for which a possible resolution may not be plausible.

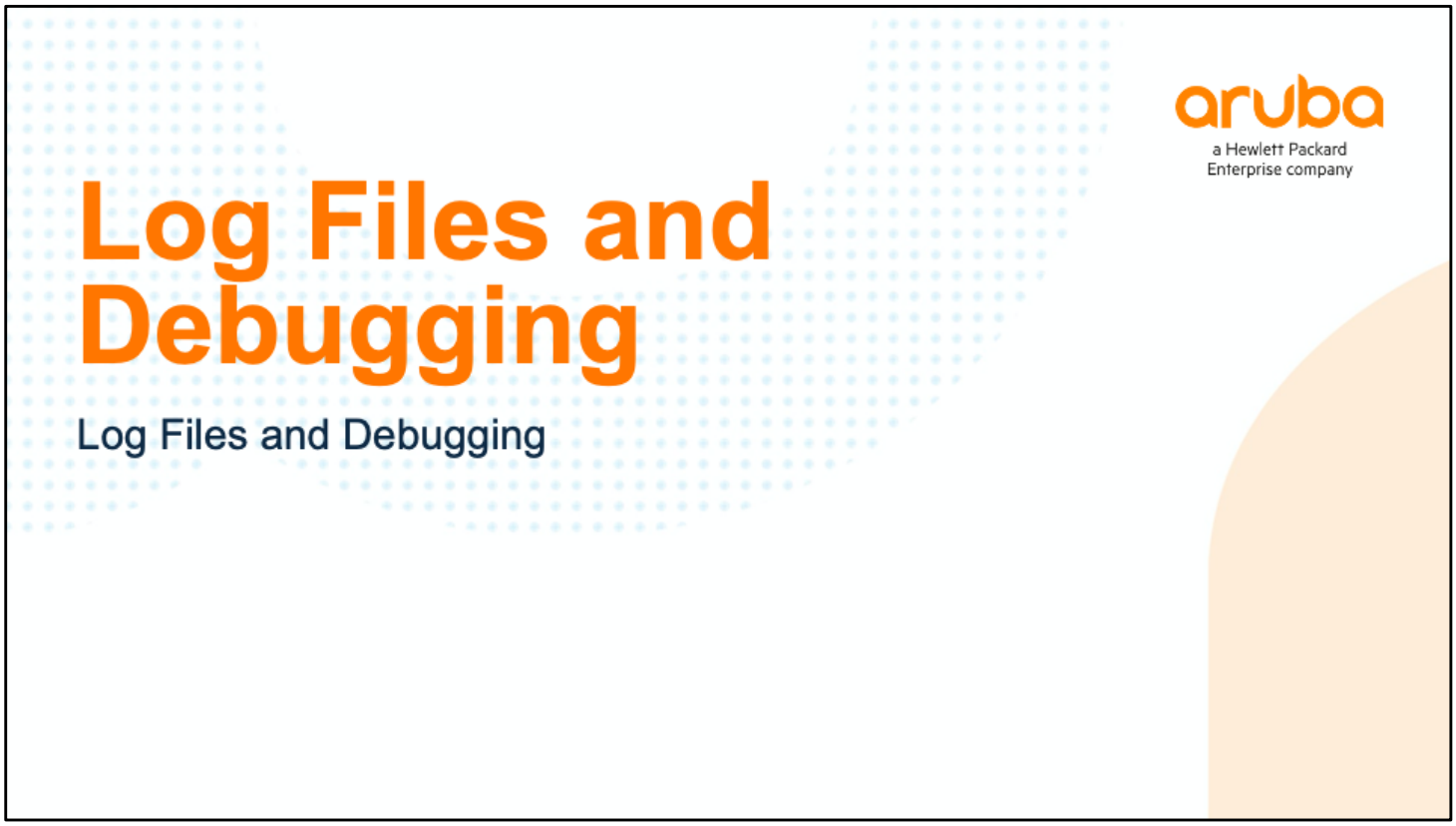
Validation—Run a validation process to prove or disprove the hypothesis. This may not be particularly feasible, for example, if you have no test lab equipment to try out your hypothesis. At the very least, performing a walk-through of the hypothesis in an articulate manner with

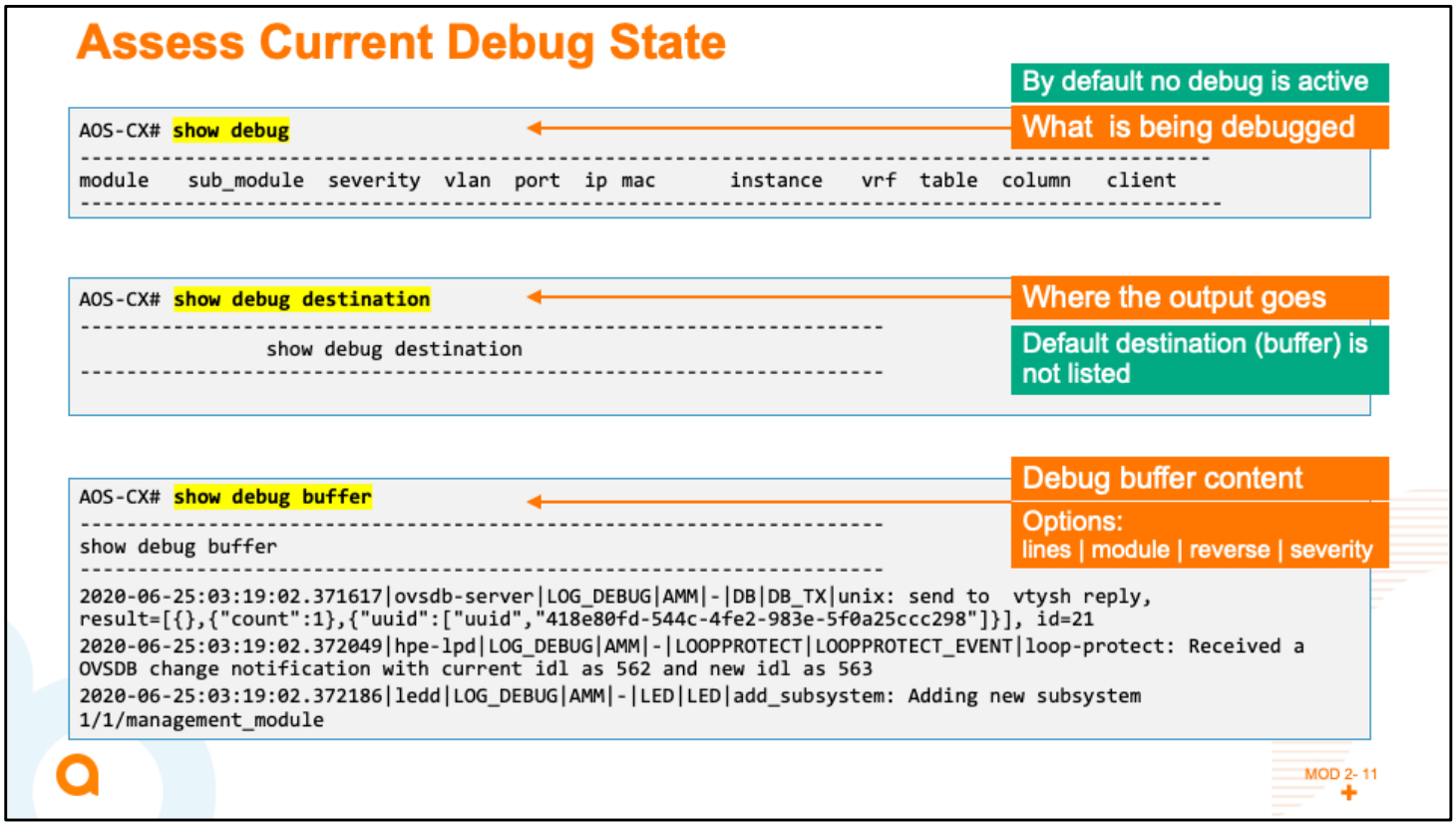
other team members may help.

Implementation—Develop an implementation plan along with a back-out plan, just in case, and then implement the resolution. For example, have a backup configuration and software image readily available.

Verification—Verify the success or failure of the implementation. If it fails, implement the back-out plan.

The steps must be executed in order starting with identification. The rules of the methodology state that if a step fails, you must return to the preceding step above (minor iteration) or possibly return to the top-level step (major iteration).


aruba
a Hewlett Packard
Enterprise company
Log Files and Debugging
Log Files and Debugging



You have learned about the high value of debugging commands for network troubleshooting – just be sure you are using good debugging techniques.

Before using debug tools, you should first assess the current state of the debugging feature. Someone else may have enabled a debug command and neglected to disable it. If you assume that no debugging is active, this could confuse your diagnoses and delay problem resolution. Use show debug to determine current debug state. In the figure's top example, no debugging is currently active. If, say debug lldp events were listed, you would use no debug lldp events to disable it, before doing your diagnoses.

Then use show debug destination to validate where debug output is being sent. The four options are buffer (default), console, file, and syslog. If the default "buffer" option is active, nothing is listed in the output. If, in addition to buffer logging, someone has enabled another option, you should be aware of this. This is especially true if you intend to use CPU-intensive debugging commands – sending heavy debug output to the console or an external syslog service increase resource utilization.

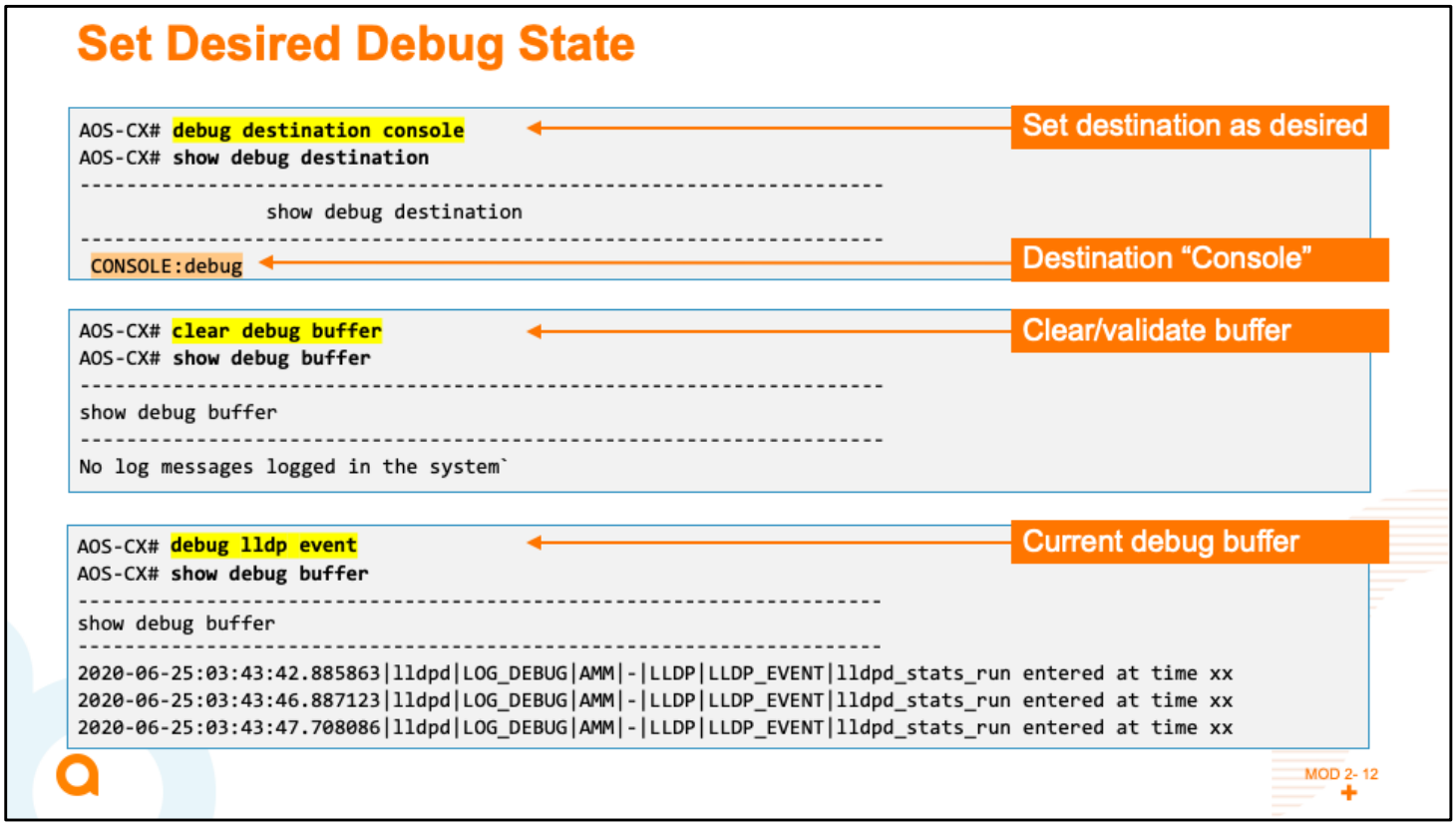
On the other hand, you may be following the best practice of using the lightest, simplest output that provides the needed information. Depending on the nature of the issue, you may benefit from seeing real-time output (debug destination console) as opposed to enabling debug, then using show debug buffer to view the results.

Yes, sending output to the console increases CPU utilization, but this is OK if you use debug commands with less-intensive output. One technique many folks use is to turn debug off BEFORE turning it on. For example, first enter the command no debug ospfv2 all, then enter

debug ospfv2 all. This could be a potentially high-output command in a busy and/or misconfigured environment. If a flurry of output appears, you merely need to press “up arrow” twice, and the command is disabled.

At the extreme end of this “high utilization danger” is debug all, which should almost never be used. Always be careful with using debug tools – the chattiest processes can create a lot of “noise” that can confuse the diagnostic process. You can often resolve many issues with simple show commands. But on those cases where you really need them, they can be invaluable.

Notice that the show debug buffer command has options to only see a certain number of lines of output, or output from a certain module or severity level. You can also choose to see the output in reverse chronological order. These options are useful to narrow your output to only the most important information.



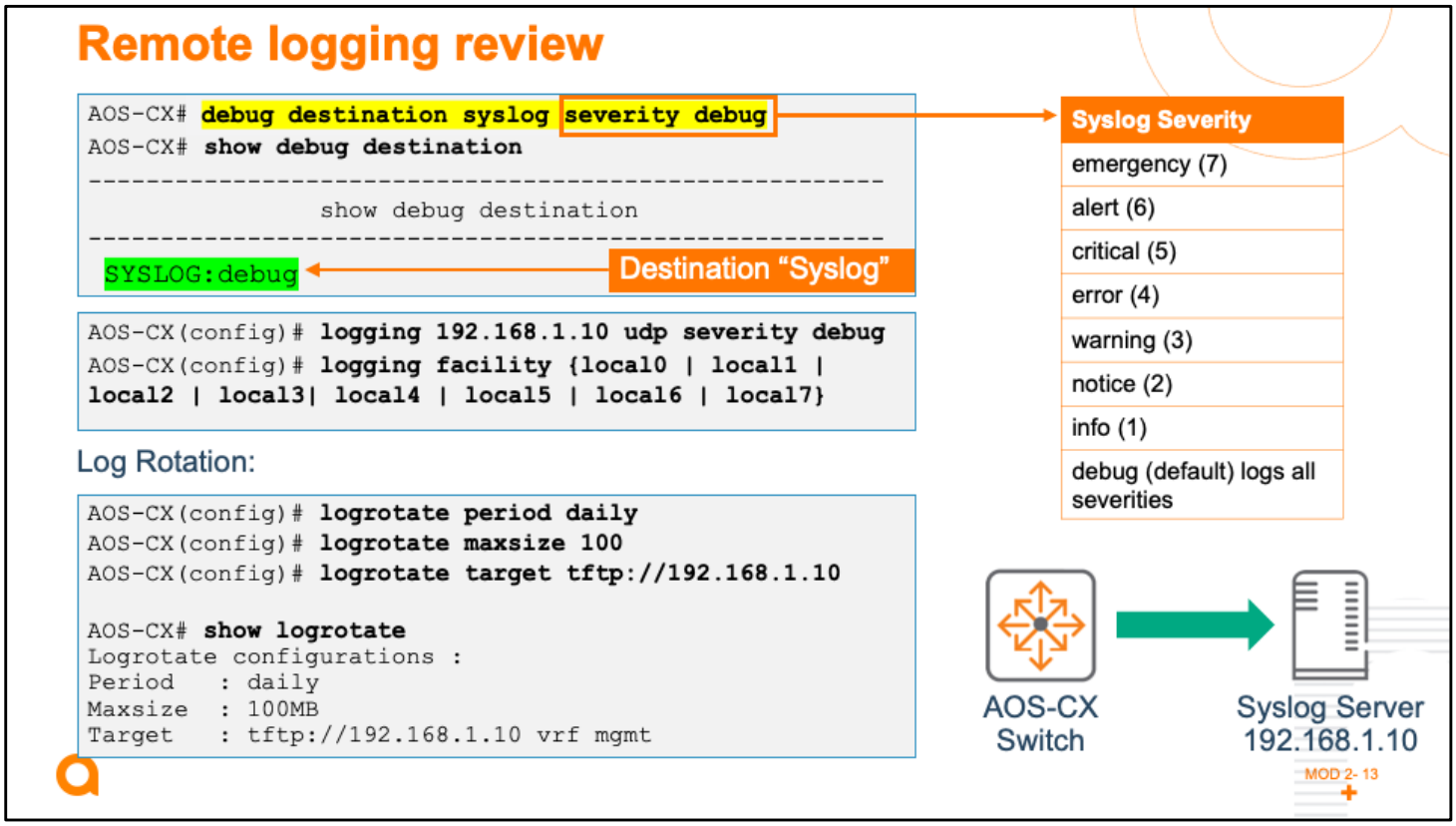
Now that you are completely aware of debug current state, you can set your desired state to begin troubleshooting. You may want to send output to the console for real-time monitoring, or to a syslog server. The syslog server can be nice if you prefer a central repository to store diagnostic information from several devices.

In the figure, debug output shall be sent to the console, in addition to the debug buffer. This is validated with show debug destination.

Next, you might use clear debug buffer to clear the existing debug buffer before starting a new debug operation. It is true that everything is timestamped. Even so, it may be easier to know that all debug output is related to your current situation. Use show debug buffer to validate.

Now issue the desired debugging command. The graphic shows debug lldp event as an example. Since the console was added as a debug destination, you will see the output in real time, and you can view it later with show debug buffer.

Finally, remember what your mother taught you as a child – clean up after yourself! Use no debug lldp event when you are done.



Another option mentioned previously is to send logging information to a remote syslog server, as shown in the figure.

For this option, you must tell the device about the syslog server:

Where the syslog server is - IP address or FQDN

Optionally, how to send it - TCP, TLS, UDP (default), VRF name

Optionally, what to send – severity level(info is default) and/or auditable events.

Optionally, the name of the VRF to use to reach the syslog server

Optionally, you can specify a logging facility to control how entries appear in the syslog server. Facilities have no relationship with severity codes and are used to help organize logged events in some way – literally, to specify the "facility" from which the logged message came. The facility could represent a hardware device, a system module, or a protocol.

One way to use this feature is to assign an arbitrary facility number to various network devices and servers. For example, you might configure "local0" on all servers, "local1" on access switches, "local2" on Access Points (AP), "local3" on distribution switches, and so on. This gives you another filter or search criteria on your syslog server. So, to see only the most egregious issues related to edge switches, you can tell your syslog server to only show entries from "local1" facilities, with severity level 7 (emergency).

AOS-CX permits to export logs from a specific severity value or above, the following output helps to understand the possible values:

```
AOS-CX(config)# debug destination syslog severity ?
  alert        Debug logs with severity 'alert(6)' and above
  crit         Debug logs with severity 'critical(5)' and above
  debug        Debug logs with all severities (Default)
  emer         Debug logs with severity 'emergency(7)' only
  err          Debug logs with severity 'error(4)' and above
  info         Debug logs with severity 'info(1)' and above
  notice       Debug logs with severity 'notice(2)' and above
  warning      Debug logs with severity 'warning(3)' and above
```

See the ArubaOS-CX Diagnostics and Supportability Guides for more details.

Log Rotation

In the figure, the top example Sets the rotate period for the event logs, stored in the /var/log/event.log file, and authentication logs, stored in the /var/log/auth.log file. Defaults to daily, options are hourly, weekly, and monthly.

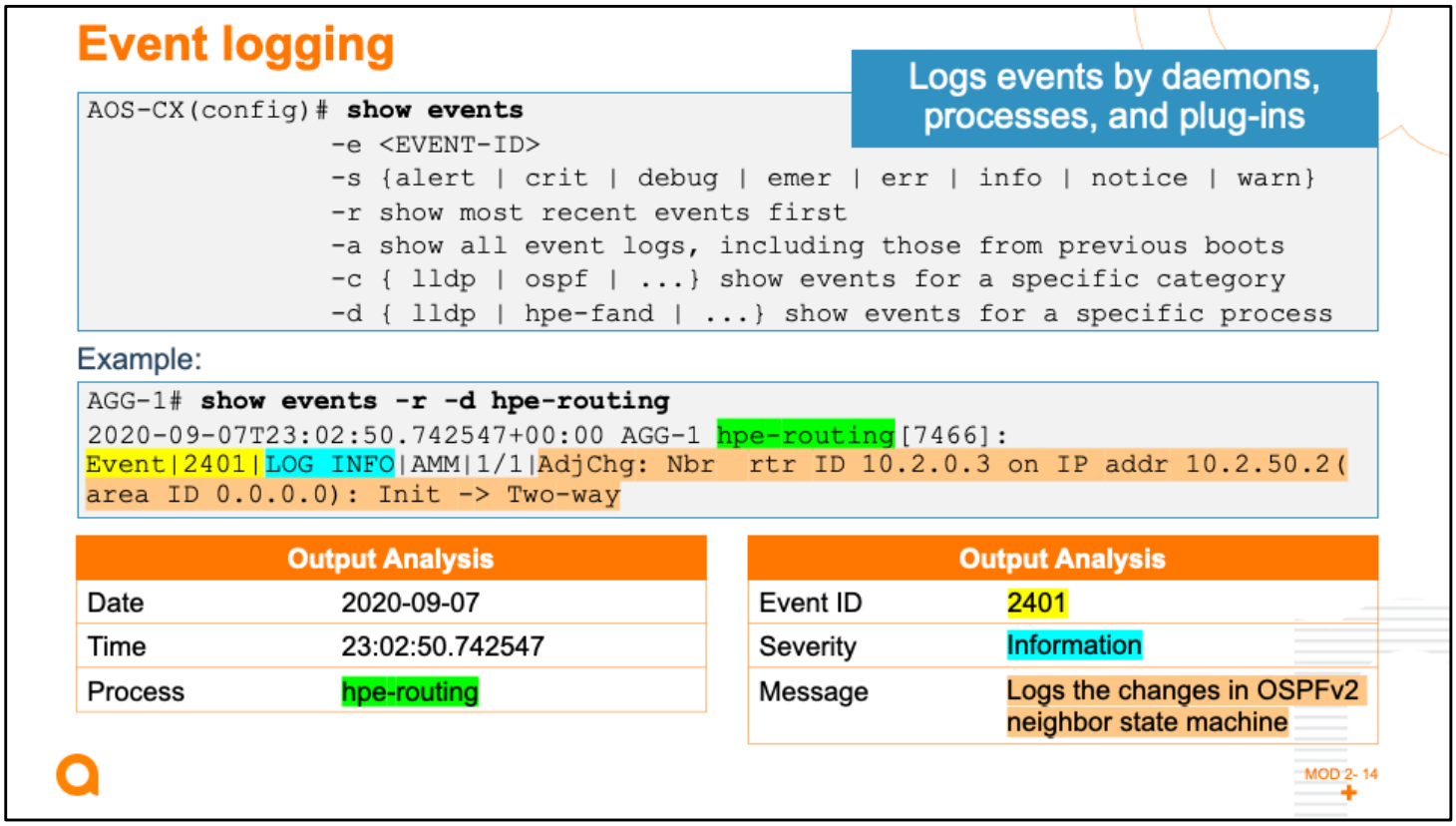
Then you can set the maximum file size log file can reach before it is compressed and stored locally or transferred to a remote host. The size is a value in the range of 10- 200 MB, and it cannot exceed 32 MB for transferred files. 100MB is the default.

The rotation period can have a significant impact on your ability to troubleshoot. You want to balance your need for data with using too many switch resources. If you rotate the log daily, and you don’t notice an issue for three days, the logs are of no help to you. Of course, many Network Management Systems (NMS) can grab the log files daily or weekly for long-term storage, giving you the best of both worlds.

The final command specifies a target TFTP server, and optionally a directory

on that server.

Note. The information sent to the Syslog server can locally be displayed in the AOS-CX switch by issuing the show event command.



Event log entries are generated by daemons, processes, and plug-ins running within the switch software. The event logging framework captures event logs in a system journal by updating the journal fields and metadata.

The figure shows the various filter options to reduce the output to only that which you require.

The output shows how useful the event logging command options can be. This example shows the –r option, which you use to show most recent events first. This is very useful if you have a large event log, and do not wish to clear it, then the –d option is used to narrow the output to a specific process – hpe-routing in this case.

The analysis presented at the bottom of the slide helps you to understand and interpret the information contained in the Log message:

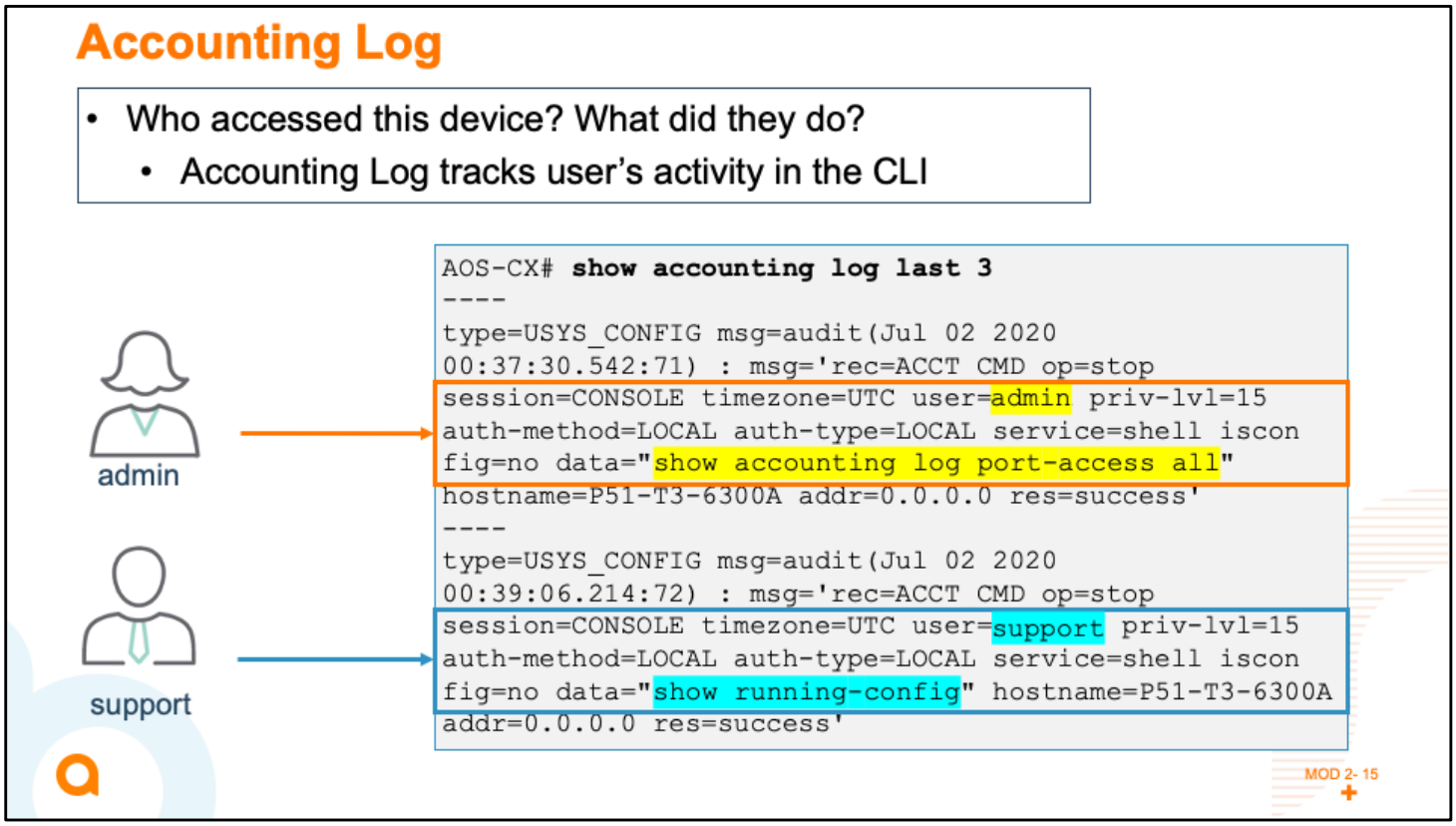
Timestamp section includes date and time when the message was generated. Remember to synchronize the time on the switch with a server using the NTP protocol.

Process ID

Event ID, this is the event ID Aruba has properly documented for the specific message. This information can be reviewed in the ArubaOS-CX Event Log Message Reference Guide document.

Severity of the message

Message generated.



Use the command show accounting log to see accounting activity related to switch login, access, and related activity. It helps you answer the question “Who accessed this device, and what did they do?”

This is especially useful in conjunction with TACACS+ deployments. If certain individuals are abusing their access, or lack training and are making mistakes, you can determine who those folks are, and use TACACS+ to limit their access to only certain commands. This gives you greater granularity to control what specific administrative staff can do when they access the device.

Question #1

```
AOS-CX# show logrotate
Logrotate configurations :
Period   : daily
Maxsize  : 100MB
Target   : tftp://192.168.1.10 vrf mgmt
```

From the output presented, what information will be sent to the Syslog server?

a. Event logs stored in `/var/log/event.log` file
b. Debug information
c. Daily commands issue by the user
d. Authentication events

Knowledge Check

Correct answer:

A

Diagnostics Commands

Top command, self test, capacities, environment

aruba
a Hewlett Packard Enterprise company

Understanding top CPU command

```
ACC-1# top cpu
top - 19:49:25 up 11 days,  1:41,  1 user,  load average: 0.23, 0.35, 0.32
Tasks: 249 total,   1 running, 205 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.4 us,  2.5 sy,  0.0 ni, 94.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  7867456 total,  3060116 free,  1321988 used,  3485352 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  6102424 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2665 root      20   0 1988332  93136  40364 S   6.2  1.2 940:13.07 switchd_ag+
```

First Row	
Current time	19:49:25
Uptime of the switch	11 days, 1:41
User sessions logged in	1
Average load on the system	last minute: 0.23%
	5 minutes: 0.35%
	15 minutes: 0.32%

Second Row	
Total processes running	249
Running processes	1
Sleeping processes	205
Stopped processes	0
Processes waiting to be stopped from the parent process	0 zombie

MOD 2- 18

One issue is defining what “higher than normal” is. Under normal circumstances, most data packets are hardware-switched using Application Specific Integrated Circuits (ASIC), and do not rely on the CPU. Utilization would typically be well below 40 to 50%. For many switches, especially access switches, near this range or higher could be cause for alarm. Certainly, CPU utilization of 60% and higher would be cause for alarm.

However, one of the most power tools in your arsenal is to “baseline” – determine what “normal” is for your devices, in your network. When your network is operating smoothly, with no problems, you have a golden opportunity for baselining. Suppose that your access layer switches normally run at about 5% CPU utilization, and suddenly they are running at 10%. This might be cause for investigation. Perhaps a mis-configuration with Spanning Tree or a routing protocol has caused L2 or L3 loops.

Of course, it could simply be that a “new normal” has been established – the new database application was launched today, or staffing load increased due to a recent merger.

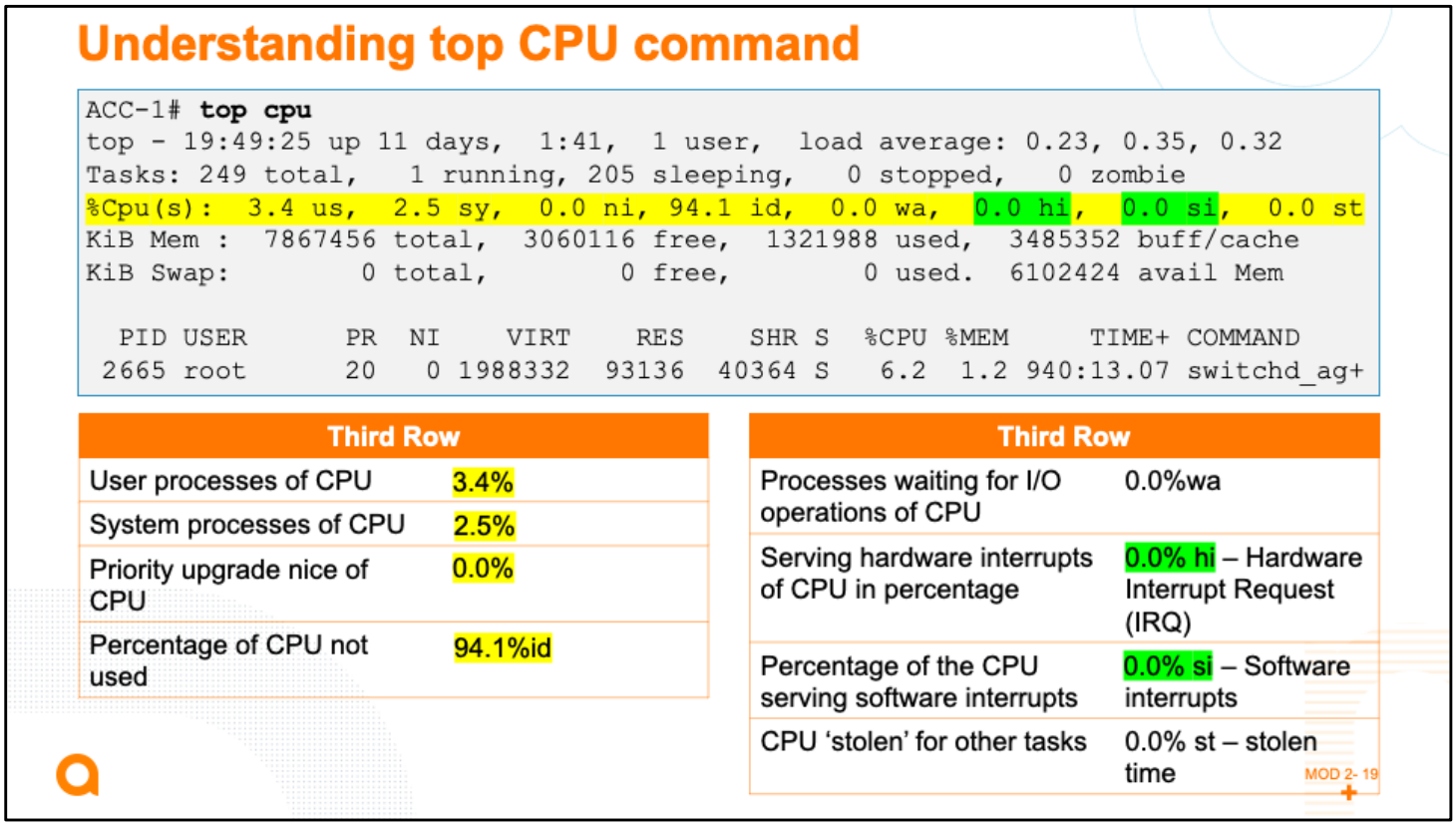
If you have no history or baseline, you can also use a “Compare and Contrast” method of troubleshooting (one of these things is not like the other). All access layer switches are currently running at about 4%, except for the switch named Access_123, which is running at 30% CPU utilization. You might scroll down, identify the offending PID, and troubleshoot from there. If the issue is related to a specific process (spanning tree, OSPF) you begin exploring those issues. If it a related to a switching daemon, you might explore why. Perhaps somebody enabled port mirroring with a destination of “CPU”. This would cause all those packets to be software switched.

Or you might then use the “show interfaces” group of commands to compare transmit/receive

volume on the interfaces. If one interface is suspiciously high, you might be able to follow that interface to its connected device chain to determine the issue.

Use the command top cpu to determine overall CPU utilization. If utilization is higher than normal, you can then scroll down through the list of Process IDs (PID) to determine which process is causing high CPU utilization – to help direct your troubleshooting efforts.

The example explains each part of the output so you can have a better understanding of how interpreter the information presented by the top cpu command.



This slide continues the analysis of each row of the top cpu command and pays special attention to the CPU interrupt information:

Interrupts are events that alter (or interrupt) the normal flow of execution in the system, therefore monitoring its presence in the system can give you important information about the overall performance. It is important to detach the concept of CPU usage and interruption. A system can be considered healthy even if its CPU usage is high and the interrupt percentage is low. On the other hand if you have a CPU usage low but a high interrupt percentage then the system is considered unhealthy.

There are two types of interrupts:

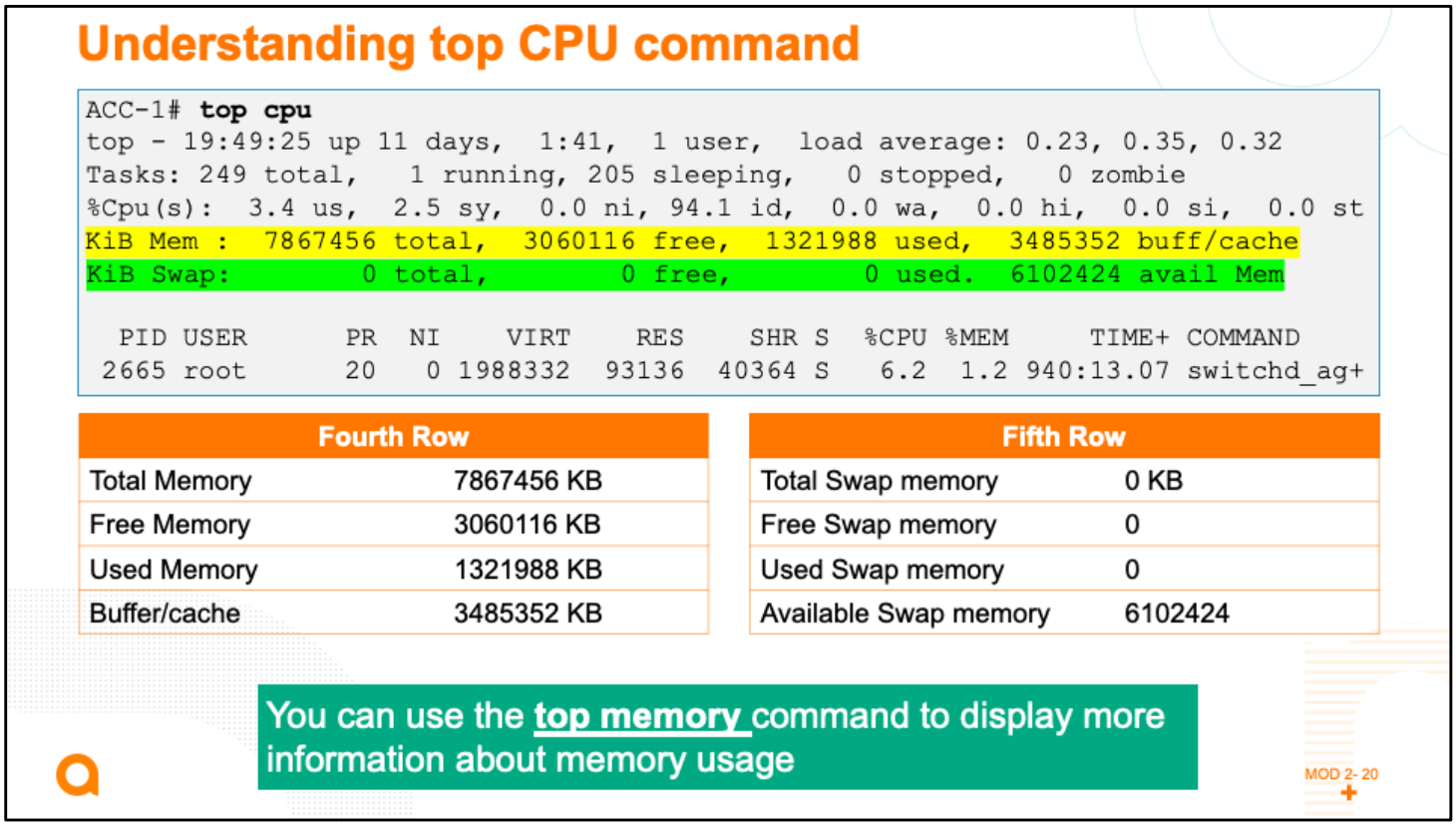
Hardware interrupt:

Are events generated by hardware devices like disk and clocks. Each device or set of hardware devices will have its own Interrupt Request (IRQ) line.

Software interrupt:

A software interrupt, also called an exception, is an interrupt that is caused by software,

usually by a program in user mode. Examples of events that cause them are requests by an application program for certain services from the operating system or the termination of such programs. When it receives a software interrupt signal, the CPU may temporarily switch control to an interrupt handler routine, and the process in the kernel that was suspended by the interrupt will be resumed after the interrupt has been accommodated. Each type of software interrupt is associated with an interrupt handler, which is a software routine that takes control when the interrupt occurs.



The top cpu command on the fourth row displays the information about the memory utilization including total memory, free, used and buffer or cache size. On the next row you can see information about the swap memory.

Swap memory

Memory swapping is a computer technology that enables an operating system to provide more memory to a running application or process than is available in physical RAM. When the physical system memory is exhausted, the operating system can opt to make use of memory swapping techniques to get additional memory.

Memory swapping works by making use of virtual memory and storage space in an approach that provides additional resources when required. In short, this additional memory enables the system to run faster and crunch data better.

With memory swapping, the operating system makes use of storage disk space to provide the functional equivalent of memory storage execution space. The space on the storage device is referred to as "swap space" and is used to run processes that have been swapped out of main physical memory.

Advantages of Memory Swapping

More Memory. Memory swapping is a critical component of memory management, enabling

an operating system to handle requests that would otherwise overwhelm a system.

Continuous Operations. Swap file memory can be written to disk in a continuous manner, enabling faster lookup times for operations.

System Optimization. Application processes of lesser importance and demand can be relegated to swap space, saving the higher performance physical memory for higher value operations.

Limitations of Memory Swapping

Performance. Disk storage space, when called up by memory swapping, does not offer the same performance as physical RAM for process execution.

Disk Limitations. Swap files are reliant on the stability and availability of storage media, which might not be as stable as system memory.

Capacity. Memory swapping is limited by the available swap space that has been allocated by an operating system or hypervisor.

The top memory command would be used in a similar fashion, except of course focused on which processes are causing memory utilization to be abnormally high.

Understanding top CPU command

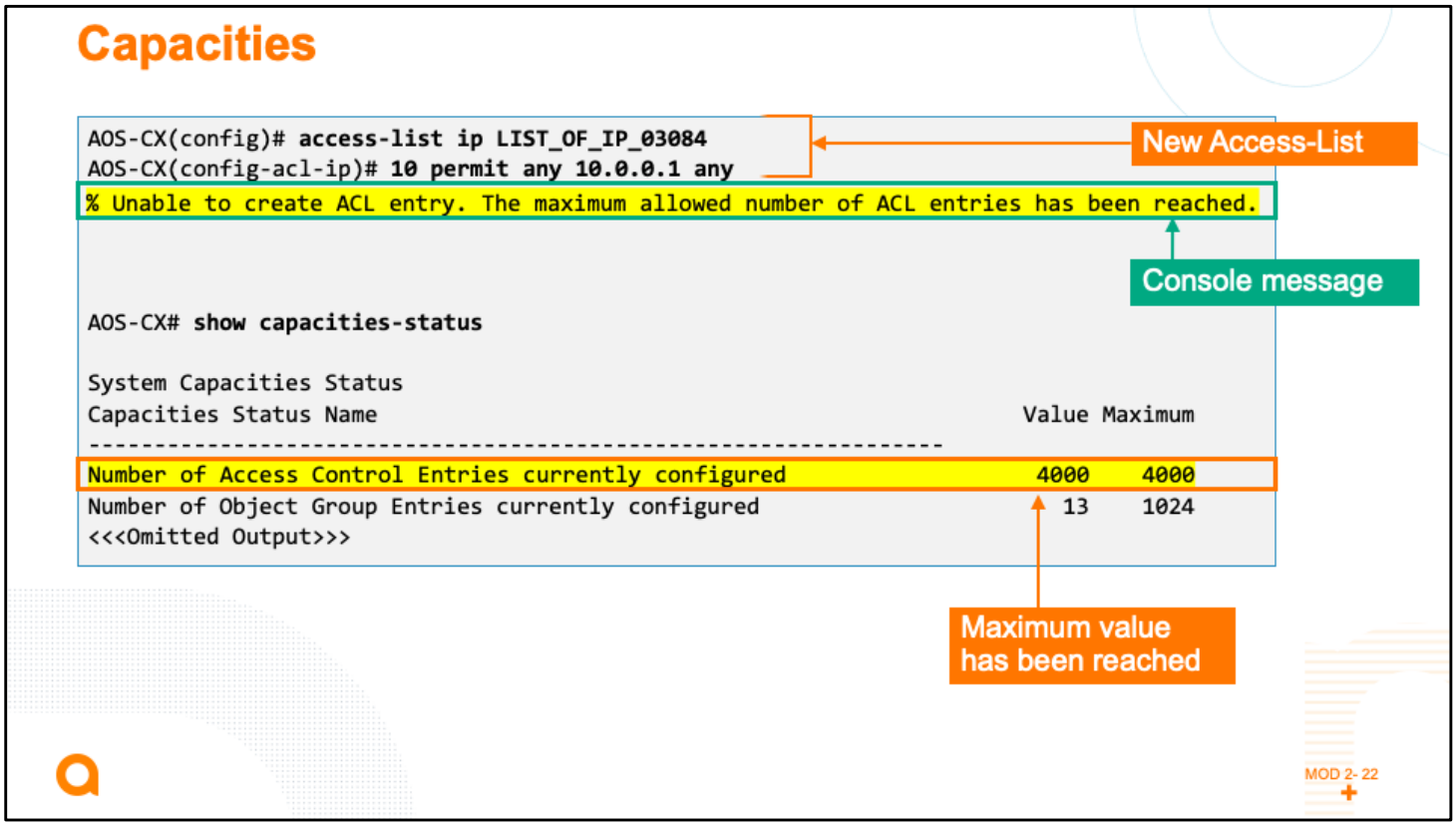
```
ACC-1# top cpu
<<Omitted Output>>
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2665 root      20   0 1988332  93136  40364 S   6.2  1.2 940:13.07 switchd_ag+
```

Processes List	
Process ID (PID)	2665
User that is owner of the process	root
Process Priority (PR)	20
NICE value of the process	0
Virtual memory used by the process (VIRT)	1988332
Physical memory used by the process (RES)	93136
Shared memory of the process	40364

Process List	
Status of the process (S= sleep, R = running, Z= zombie)	S
CPU percentage used by this process	6.2%
Memory percentage used by this process	1.2%
Total time of activity of this process	940:13.07
COMMAND (name of the process)	switchd_ag+

MOD 2- 21

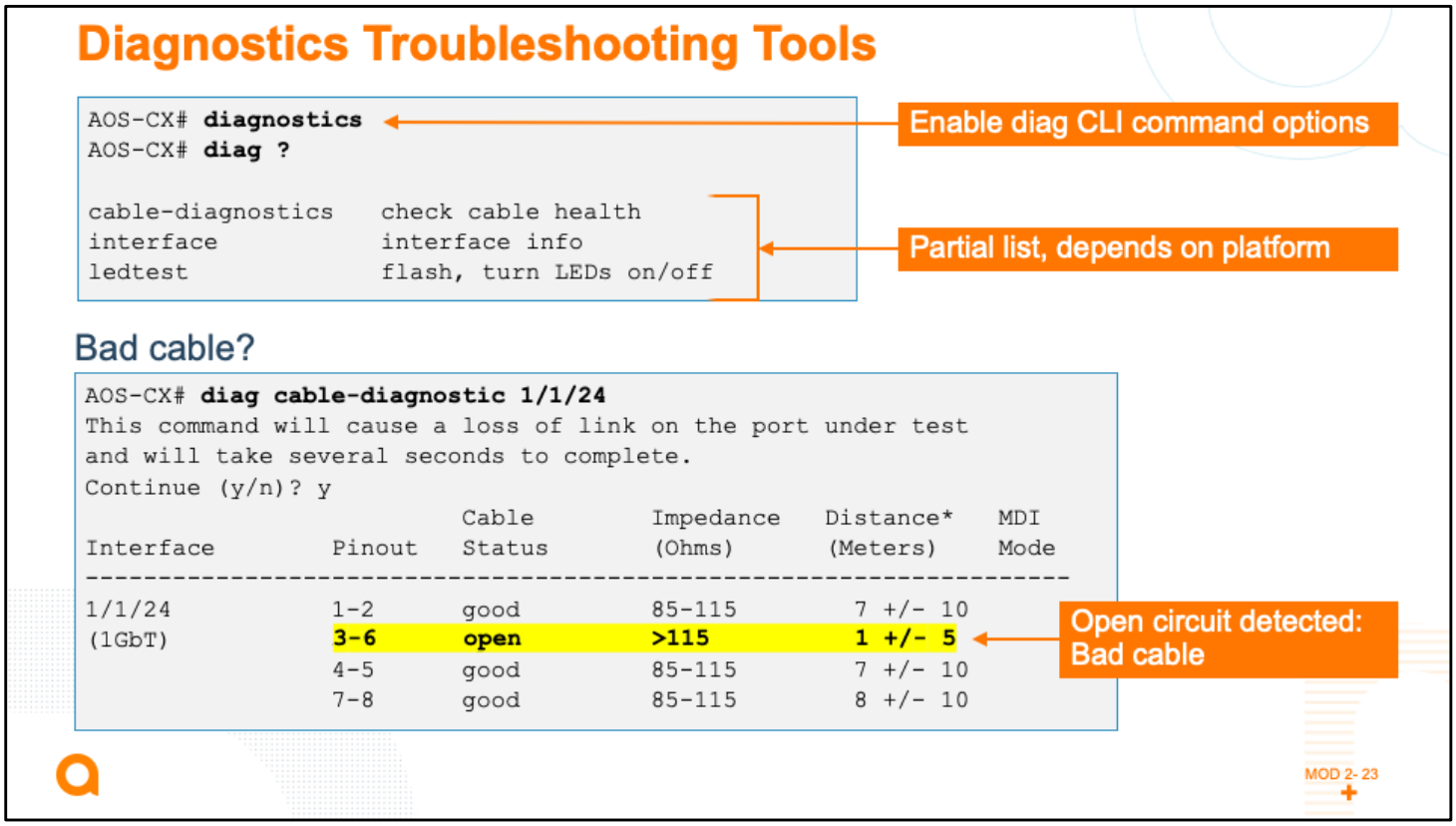
The last information displayed by the top cpu command is the processes list. On this example we present one of the multiple processes that can run in the system and explain its components.



Use show capacities-status to determine if any processes or configuration parameters are at or near maximum values. This could prevent you from configuration desired features or entities.

In this example the administrator wishes to create a new access-list however a message appears when the Access Control Entry is defined, this message indicates the system is unable to create the entry due to a capacity problem.

Often, this command may help you realize that some “clean-up” may be in order. For example, you may have VLANs or Access Control Entries that are no longer used and can be deleted to make room for more current requirements.



ArubaOS-CX includes several built-in diagnostic capabilities. First you enter the command diagnostics to enable the CLI diag command options. Then use the command diag with any of the available options. Many of these options are typically only of use to Aruba Support personnel, but you might find some of them useful.

The figure shows a select portion of available options, which varies based on platform.

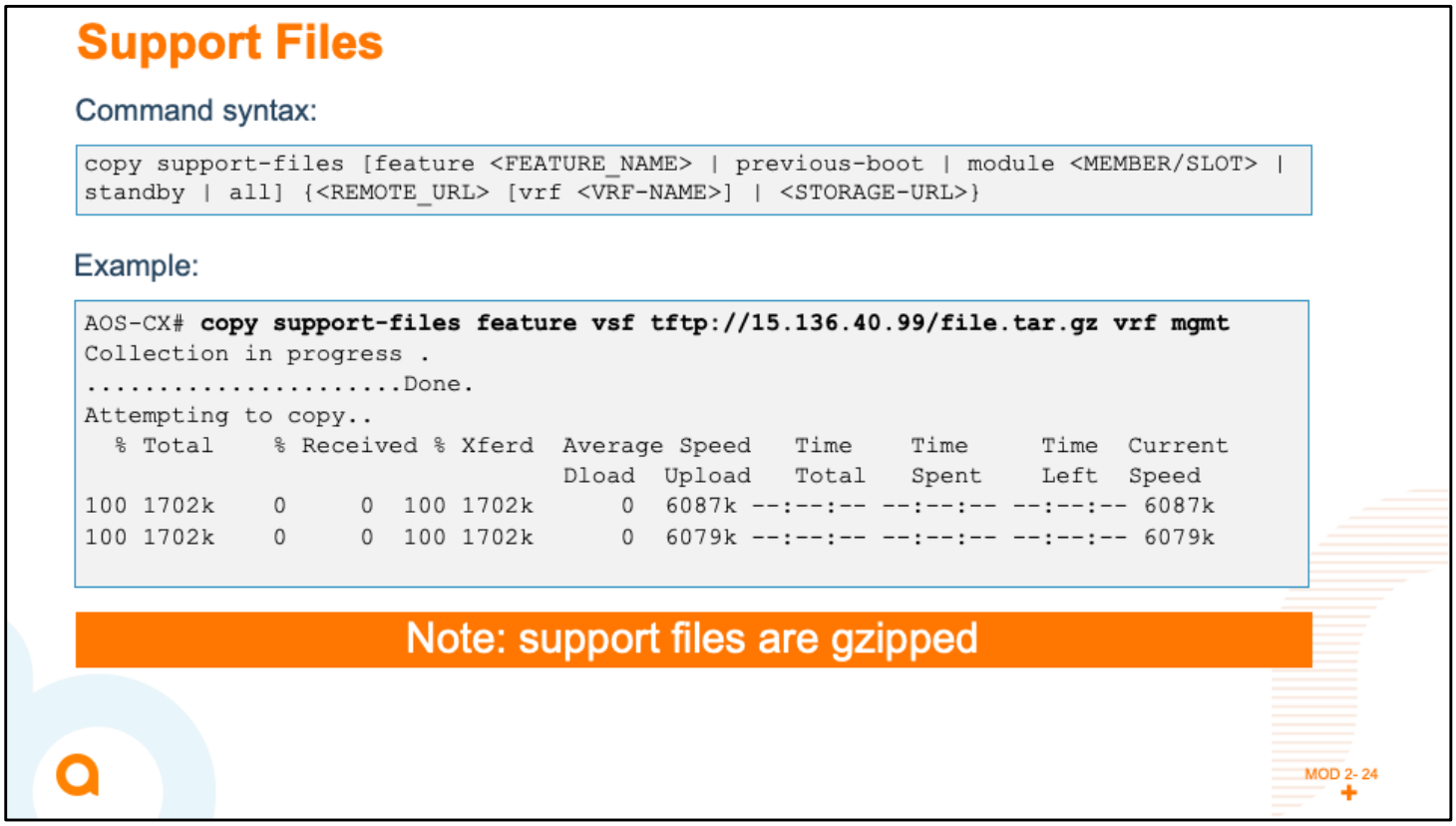
The example shows the cable diagnostics for interface 1/1/24. The cable pairs all test as good. The “distance to fault” is to be ignored in this case, as there is no fault on this cable – the cable status is good.

If you have a more challenging issue related to transceivers and associated cabling, you might also explore the command diag interface command. For example, the command diag interface 1/1/27 transceiver all provides very detailed information on the transceiver module and associated connectivity. This command only works for interfaces that have pluggable transceivers installed.

Note. The following information is displayed for the diag cable-diagnostic command.

Cable Impedance
Status (Ohms) Description
---

good	85-115	No cable faults found
open	>115	Open circuit detected
intra-short	<85	Short-circuit within the same wire pair
inter-short	<85	Short-circuit with another wire pair
high-imp	>115	Cable impedance was higher than expected
low-imp	<85	Cable impedance was lower than expected
unknown	--	Cable test was inconclusive



The figure shows the command to copy support files to an external tftp server. This can be valuable when dealing with Aruba Support staff. You can also copy files to USB and SFTP targets. SFTP is a particularly nice option, because it is TCP-based. If there are network issues during a copy, SFTP may recover more gracefully than TFTP.

The command syntax:

copy support-files [feature <FEATURE_NAME> | previous-boot | module <MEMBER/SLOT> | standby | all] {<REMOTE_URL> [vrf <VRF-NAME>] | <STORAGE-URL>}

Some examples of the usage of the command:

Copying the support files to a remote URL:
switch# copy support-files tftp://10.100.0.12/file.tar.gz

Copying the support files of the lldp feature to a remote URL with a specified VRF:
switch# copy support-files feature lldp tftp://10.100.0.12/file.tar.gz vrf mgmt

Copying the support files from the previous boot to a remote URL with a specified VRF:

switch# copy support-files previous-boot sftp://root@10.0.14.206/file.tar.gz vrf mgmt

Copying the support files to a USB:

switch# copy support-files usb:/file.tar.gz

Copying the files from a module to a remote URL with a specified VRF on an 8400 or 6400 switch:

switch# copy support-files module 1/1 tftp://10.100.0.12/file.tar.gz vrf mgmt

Copying the files from a standby module to a remote URL with a specified VRF on an 8400 or 6400 switch:

switch# copy support-files standby sftp://root@10.0.14.216/file.tar.gz vrf mgmt

Copying all the support files to a remote URL:

switch# copy support-files all sftp://root@10.0.14.216/file.tar.gz vrf mgmt

Copying the support files of the config feature to a USB:

switch# copy support-files feature config usb:/file.tar.gz

Question #2

```
Switch# top cpu
top - 02:20:44 up 17 days,  8:17,  6 users,  load average: 0.47, 0.51, 0.54
Tasks: 311 total,   1 running, 266 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.9 us,  5.9 sy,  0.0 ni, 90.2 id,  0.4 wa,  0.8 hi,  0.2 si,  0.0 st
KiB Mem :  7868152 total,  2489816 free,  1568028 used,  3810308 buff/cache
KiB Swap:  2097148 total,  2096124 free,     1024 used.  5703836 avail Mem
```

From the output, what is the value of Hardware Interrupts?

a. 5.9
b. 0.47
c. 0.8
d. 0.2

a

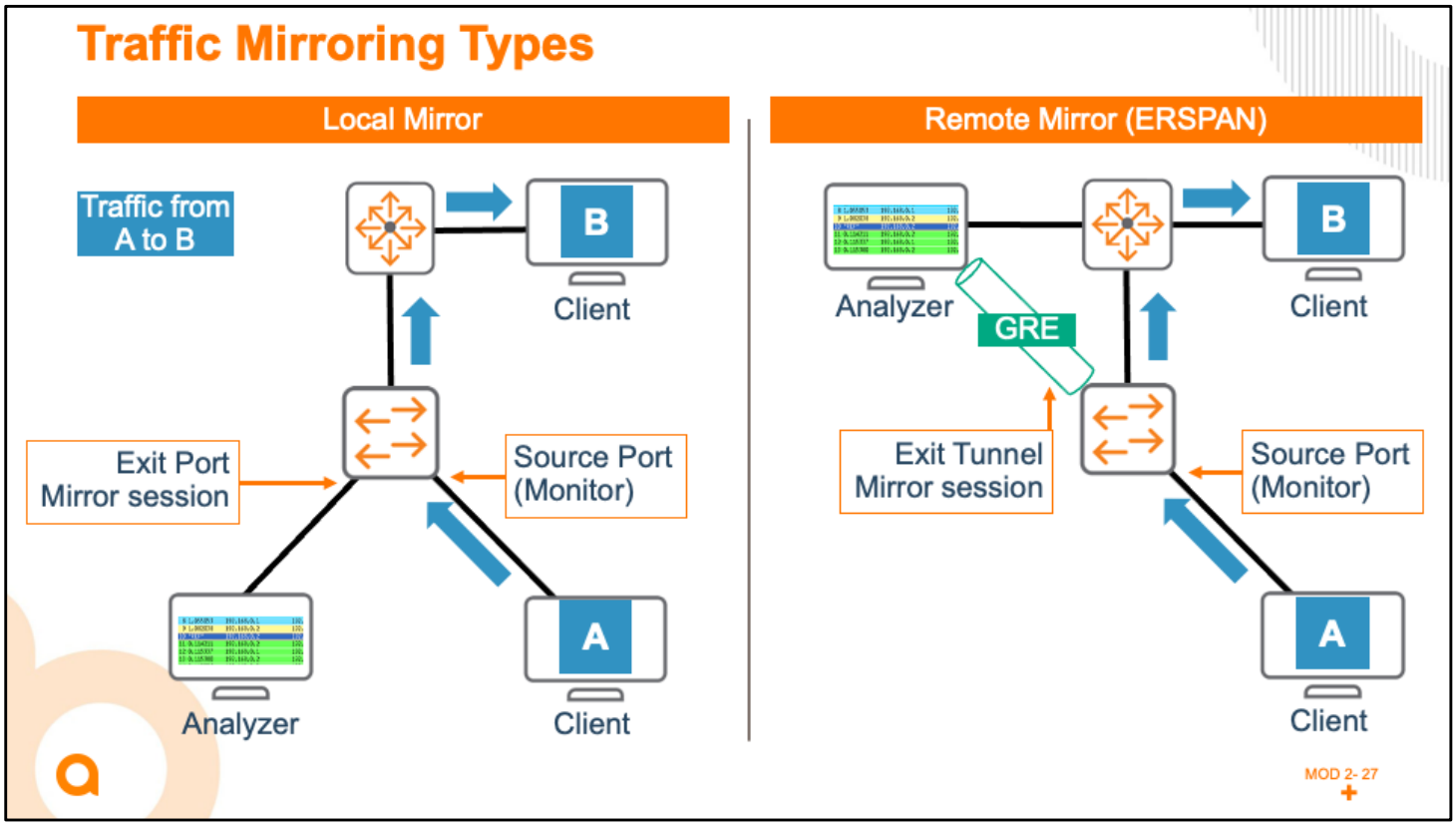
Knowledge Check

Correct answer:
C

aruba
a Hewlett Packard Enterprise company

Traffic Analysis

Mirror Session, TCP Dump and Tshark



Traffic mirroring allows a switch to copy frames that arrive or are sent from a particular port or VLAN and forward them to another port. You can use port mirroring to monitor network traffic to detect threats, troubleshoot problems, or manage the network. For example, you might want to send certain traffic to a security appliance such as an Intrusion Detection System (IDS) device, which can examine the traffic and detect possible threats. Or, you may want to send traffic to a network protocol analyzer, which you can use to examine the traffic and troubleshoot a network problem.

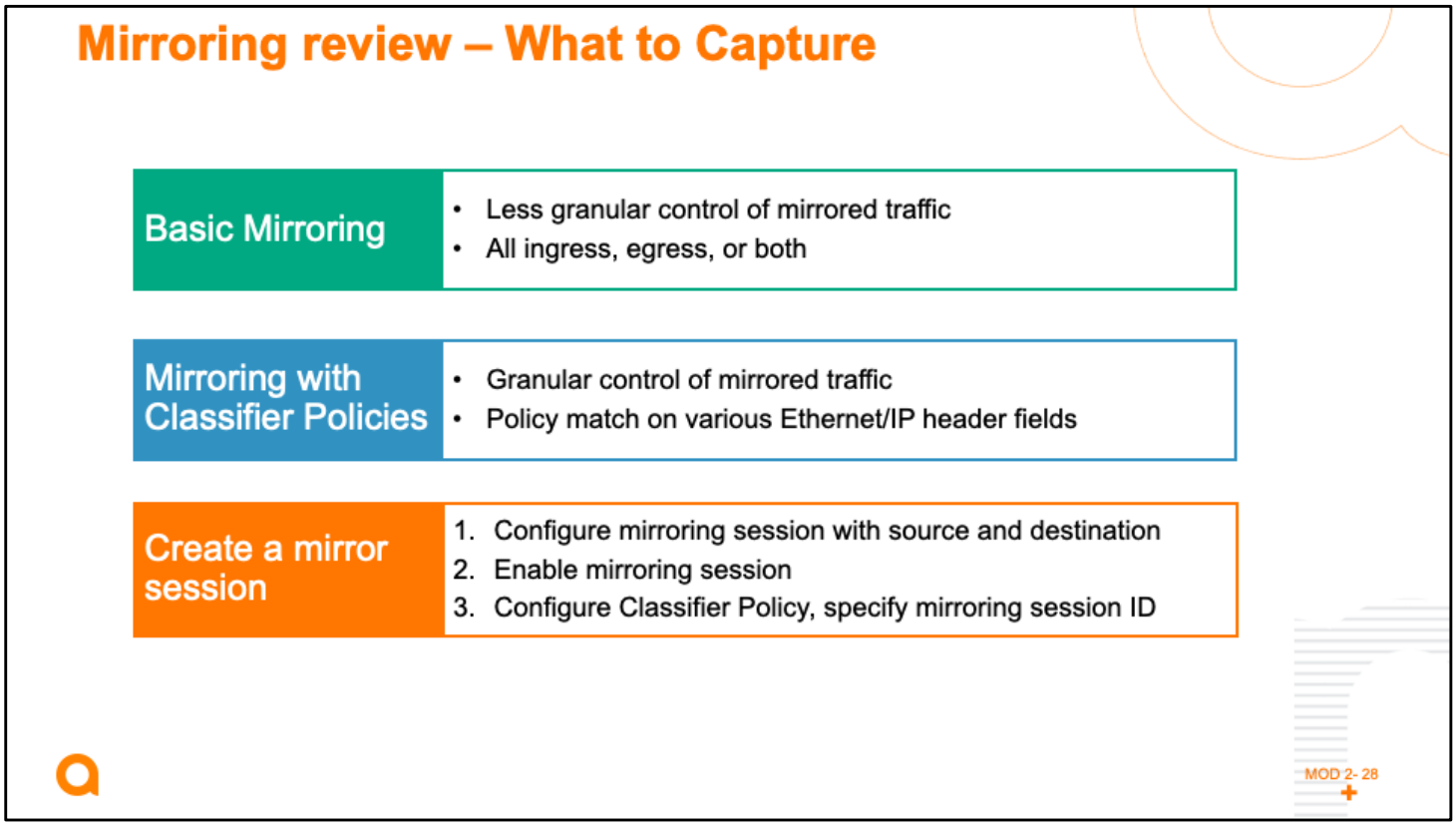
The simplest form of traffic mirroring is local mirroring, in which the switch mirroring the packets sends the copies out a local port.

AOS-CX switches also support remote port mirroring, in which the switch forwards the mirrored packets to a destination device (Analyzer). The switch achieves this by encapsulating the mirrored packets in a GRE header that uses the remote switch's IP address as the destination. The remote, or destination, switch is configured to decapsulate traffic from this GRE tunnel and to forward the traffic out an exit port (see the previous slide on setting up local mirroring).

You may configure multiple mirror sessions with the same source/destination IP address pair, however, only one of those sessions sharing the same source/destination IP address pair can be enabled at a given time.

Remote mirroring is not supported leaving the switch by the OOB port. If VRF management is configured for a Remote session, the session will be in "mirror_err_tunnel_oob_port_not_supported" operation status.

Remote mirroring is not supported leaving the switch encapsulated within another tunnel (e.g. GRE IPv4). When the path to the destination IP address will leave via a tunnel, the session will be in "tunnel_route_resolution_not_populated" operation status.



You just learned that you can control the “where” aspect of mirroring – where to capture from (source port or ports), and where to send traffic to (local or remote switch destination switch port). You can also control the “what” aspect of mirroring – what type of traffic should be captured.

You can choose to create either a basic mirroring session, or mirroring with a classifier policy.

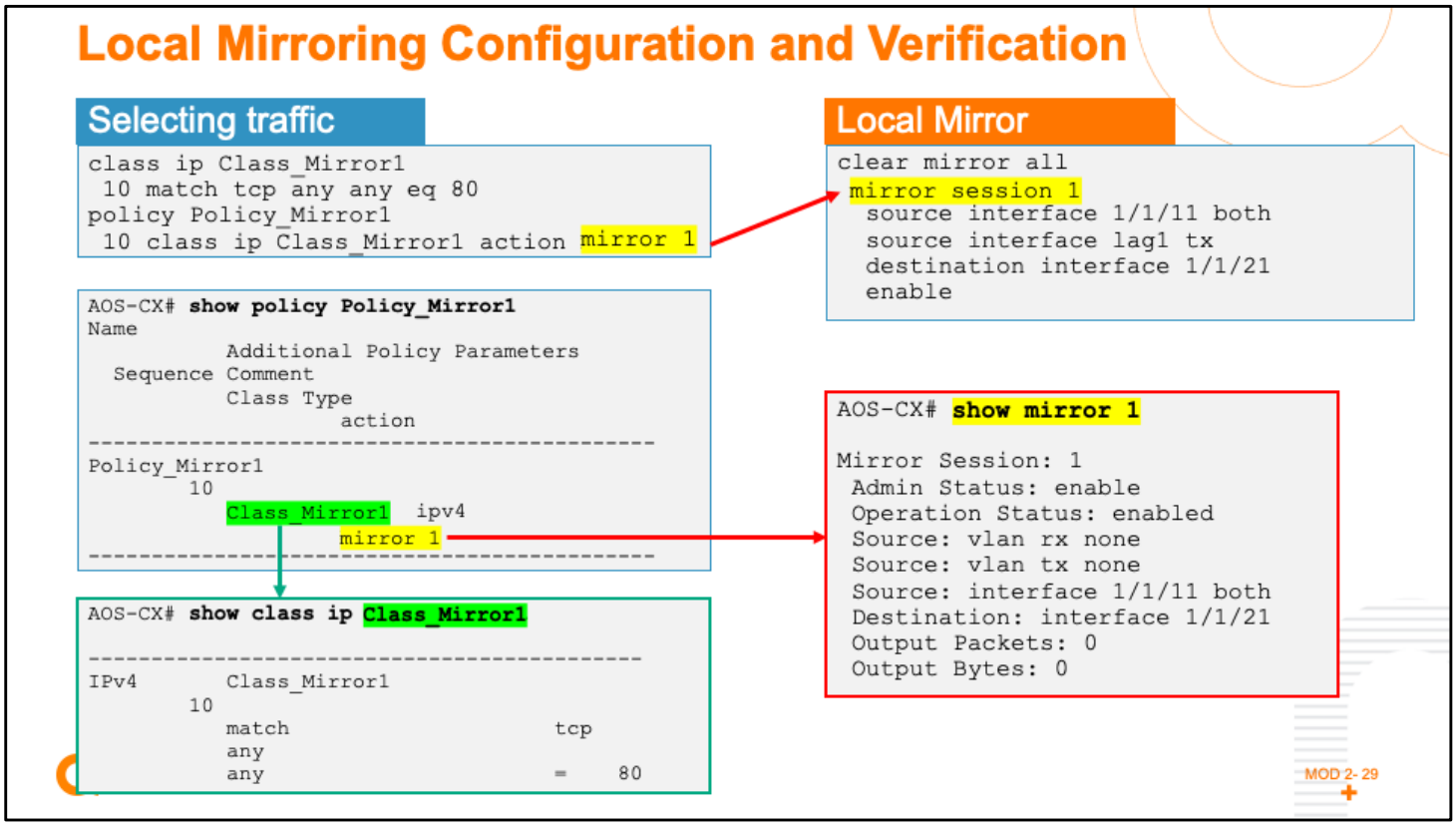
Basic mirroring only gives you coarse, less granular control of which source traffic is mirrored to the destination. You can mirror all ingress traffic, all transmitted egress traffic, or both. The advantage of this method is simplicity – it is easy to configure. If you believe that there is little danger of overwhelming the destination port or switch CPU, this can be good method. In some cases, you are not sure of what you need to capture, and it is often better to have captured traffic that you don’t need than to not have captured traffic that you do need. And remember, utilities like Wireshark have powerful filtering capabilities . This includes the ability to control what Wireshark initially accepts inbound (capture filter), and what Wireshark displays on screen (Display Filter).

On the other hand, there may be times when you have a more precise idea of what needs to be captured, or you may be concerned that capturing all traffic might overwhelm the destination port or the switch CPU. You might choose to configure a classifier policy, which you then apply to the mirroring session. This is a bit like configuring an access list to control what traffic is accepted, and then applying that list to an interface; or like configuring a QoS policy and applying it to an interface. The policy can match on various fields in an Ethernet

and/or IP header.

Creating a mirror session involves three high-level steps:
Configure the mirroring session with a destination interface
Enable the mirroring session
Optionally, configure a classifier policy and specify a mirroring session ID.

Let’s look at some example mirroring sessions.

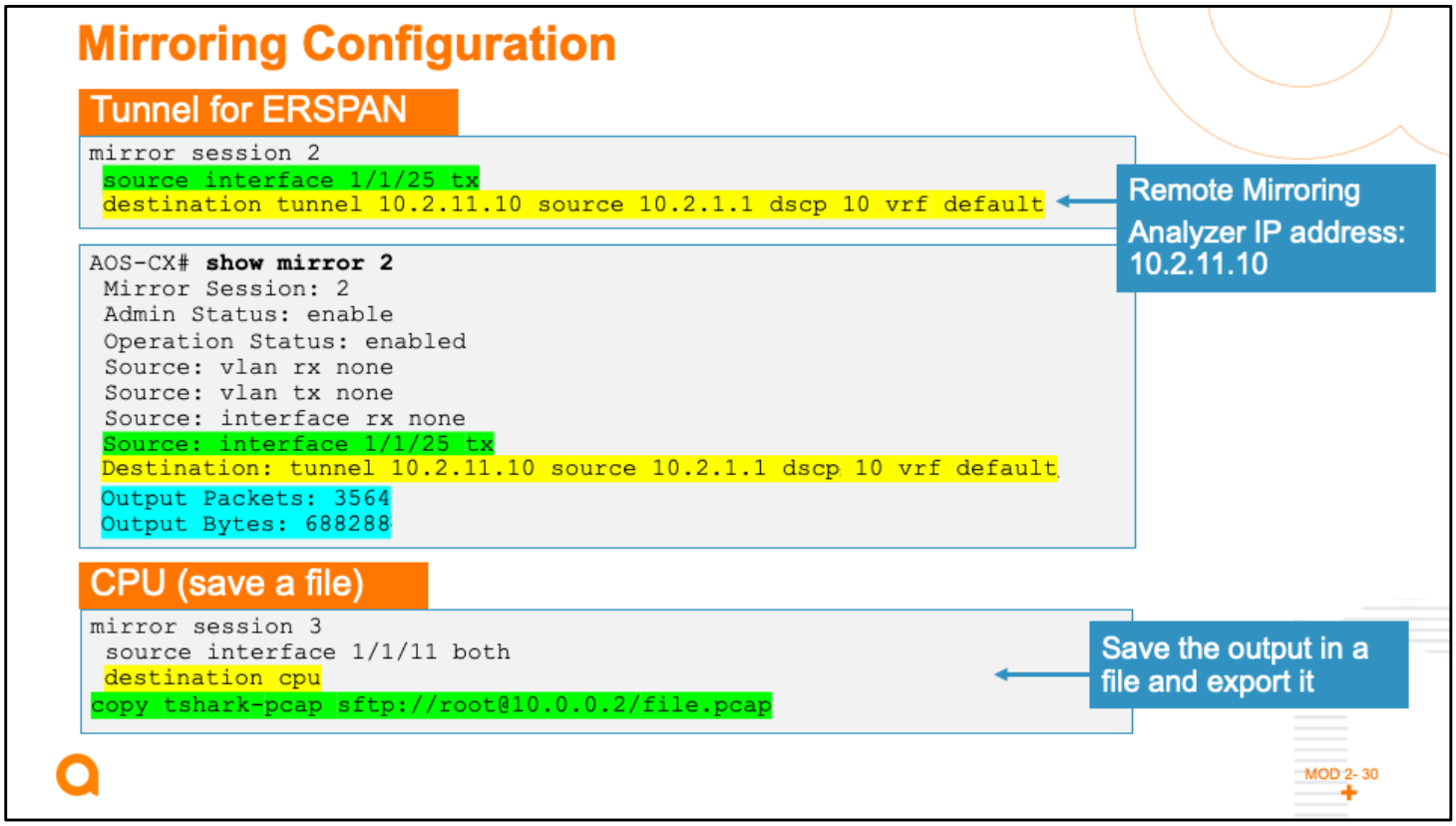


The example shows a local mirroring session with two sources (interface 1/1/11 and LAG1) and a local destination (interface 1/1/21). If you needed more granular control over what is sent to this destination, you can optionally add a Class-based Policy, where you can define what type of traffic should be sent. This policy states that anything that matches the class named “Class_Mirror1” shall be used to limit what is sent to the destination interface for mirror session 1.

You use the show policy command. If you are not aware of the exact policy name, you can use show policy without the specific policy name to see all policies. You are looking for a policy that has an action of “mirror”. When you find such a policy, you will discover which Class is tied to which mirror ID. In the example you see that Class_Mirror1 is associated with Mirror 1. So you can use show class ip Class_Mirror1 to see what traffic shall be forwarded from some source to some destination. Which source? Which destination?

Simply use show mirror 1 to see the defined source and destination ports. The class and policy control WHAT is being mirrored, while the mirror configuration controls WHERE it is being mirrored.

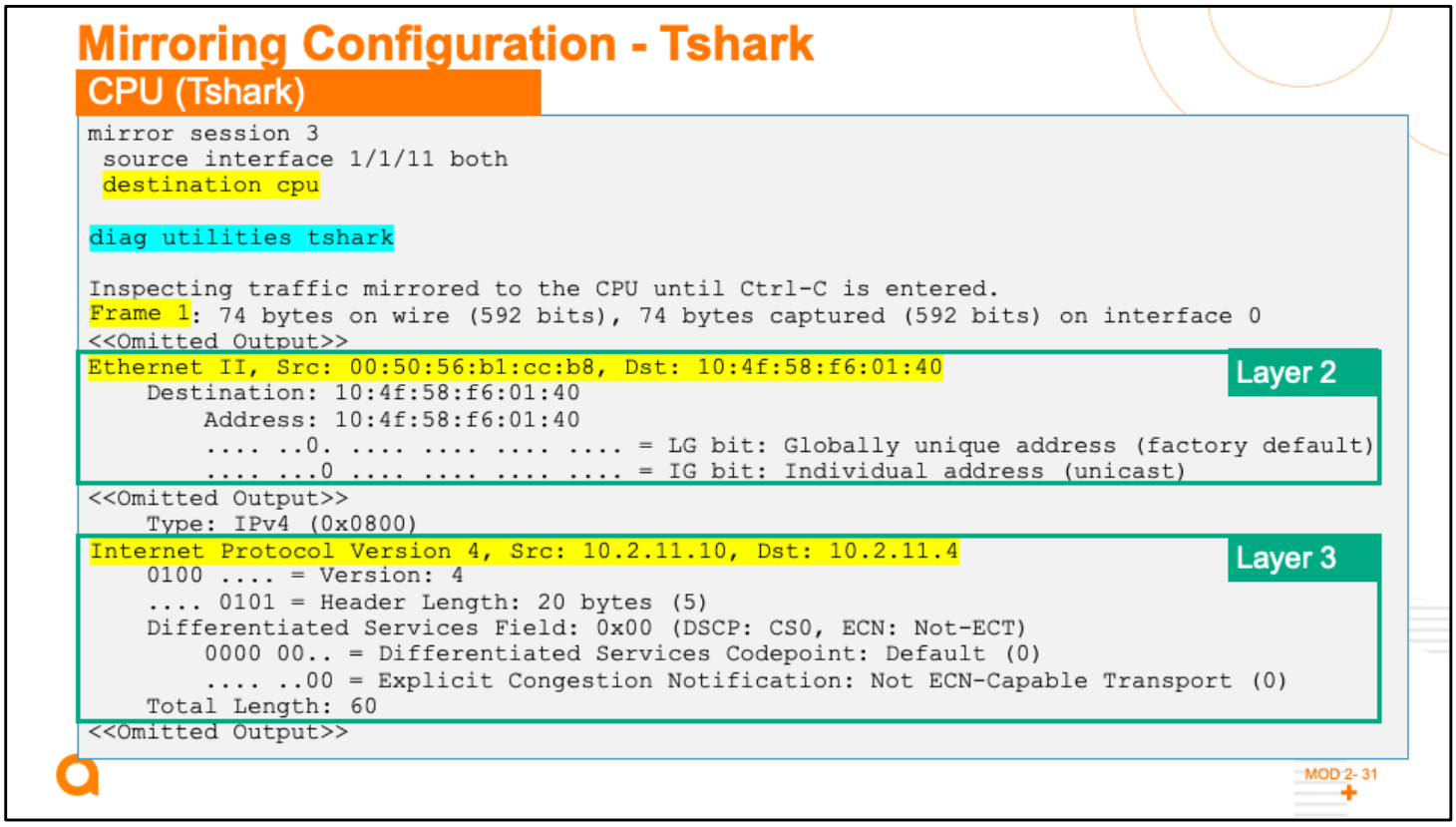
If no policies are configured with a mirror action, you know that any mirroring on this switch is a basic, non-policy-based configuration. If you are unaware of current mirroring configuration, use show mirror to see the list. Then choose which configuration to explore with show mirror 1.



The second example shows an Encapsulated Remote SPAN (ERSPAN) example, in which mirrored traffic is sent to remote device at IP address 10.2.11.10, with a DSCP value of 10, using the default VRF.

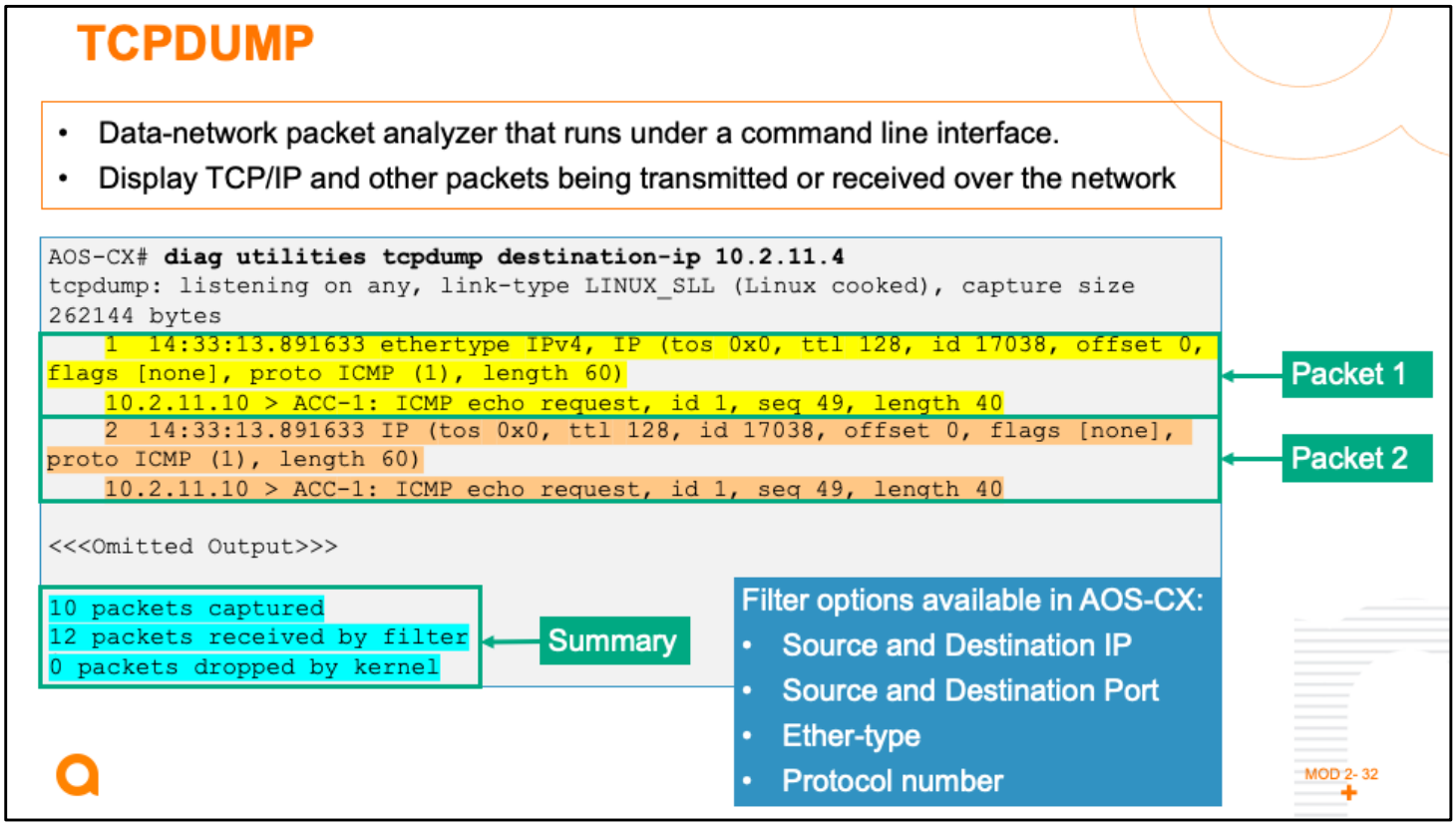
Mirror session 3 defines the switch CPU as the destination so a Tshark1 file can be generated. You would probably not what to use this option unless you could not use the more typical destination port option. You are using more CPU and storage resources on the switch with this option. You might use this if you could not connect a capture device to a switch for some reason. There are some cases where you have traffic that can be pinned down to a smaller volume, this might be a viable option.

1TShark is a network protocol analyzer that uses the command line interface. It lets you capture packet data from a live network, or read packets from a previously saved capture file, either printing a decoded form of those packets to the standard output or writing the packets to a file. TShark's native capture file format is pcapng format, which is also the format used by Wireshark and various other tools.



AOS-CX allows you to display the Tshark output in the console, you must use the diag utilities tshark command.

In the output presented, you can see the packet capture of mirror session 3.



TCPDUMP is a data-network packet analyzer computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over the network. It is distributed under the BSD license TCPDUMP is free software.

TCPDUMP works on most Unix-like operating systems and it is also supported on Windows (called WinDump).

TCPDUMP prints out a description of the contents of packets on a network interface that match the Boolean expression; the description is preceded by a time stamp.

When TCPDUMP finishes capturing packets, it will report counts of:

Packets “captured” (this is the number of packets that tcpdump has received and processed).

Packets “received by filter” (AOS-CX counts packets regardless of whether they were matched by the filter expression and, even if they were matched by the filter expression, regardless of whether TCPDUMP has read and processed them yet)

Packets “dropped by kernel” (this is the number of packets that were dropped, due to a lack of buffer space, by the packet capture mechanism in AOS-CX)

To use this tool in AOS-CX you need to use the diag utilities tcpdump <filter> command.

The slide shows the output for a tcpdump capture. This information displays 2 ICMP packets received sourced by the 10.2.11.10 device and sent to 10.2.11.4. At the end of the output you can find the summary of the capture.

Filter options in AOS-CX include:

- Source and destination IP
- Source and destination port
- Ether-type
- Protocol number

Question #3

Can the traffic mirrored to a remote destination (ERSPAN) be prioritized using DSCP marking?

a. True

b. False

Knowledge Check

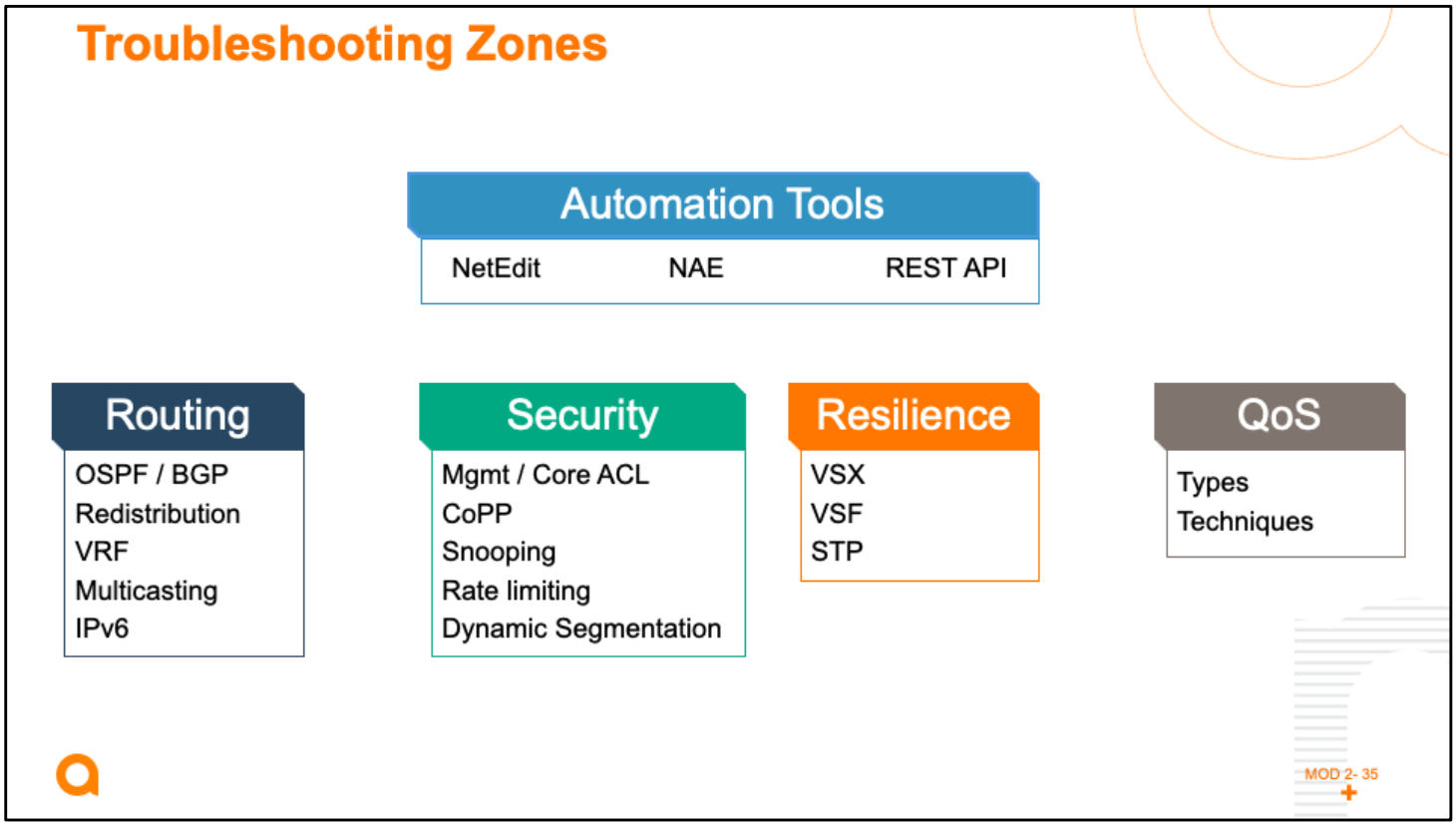
Correct answer:

a. True

aruba
a Hewlett Packard Enterprise company

Additional Content

Module 02



The figure organizes the material covered in this course into five potential troubleshooting zones.

Routing – Devices must be properly addressed, and traffic must be properly routed. You must ensure that routing adjacencies are properly formed, and routing tables are properly built. Complexity increases with multiple routing protocols, route redistribution, and complex BGP policies. Other zones, such as Security and Resilience, can impact routing functionality.

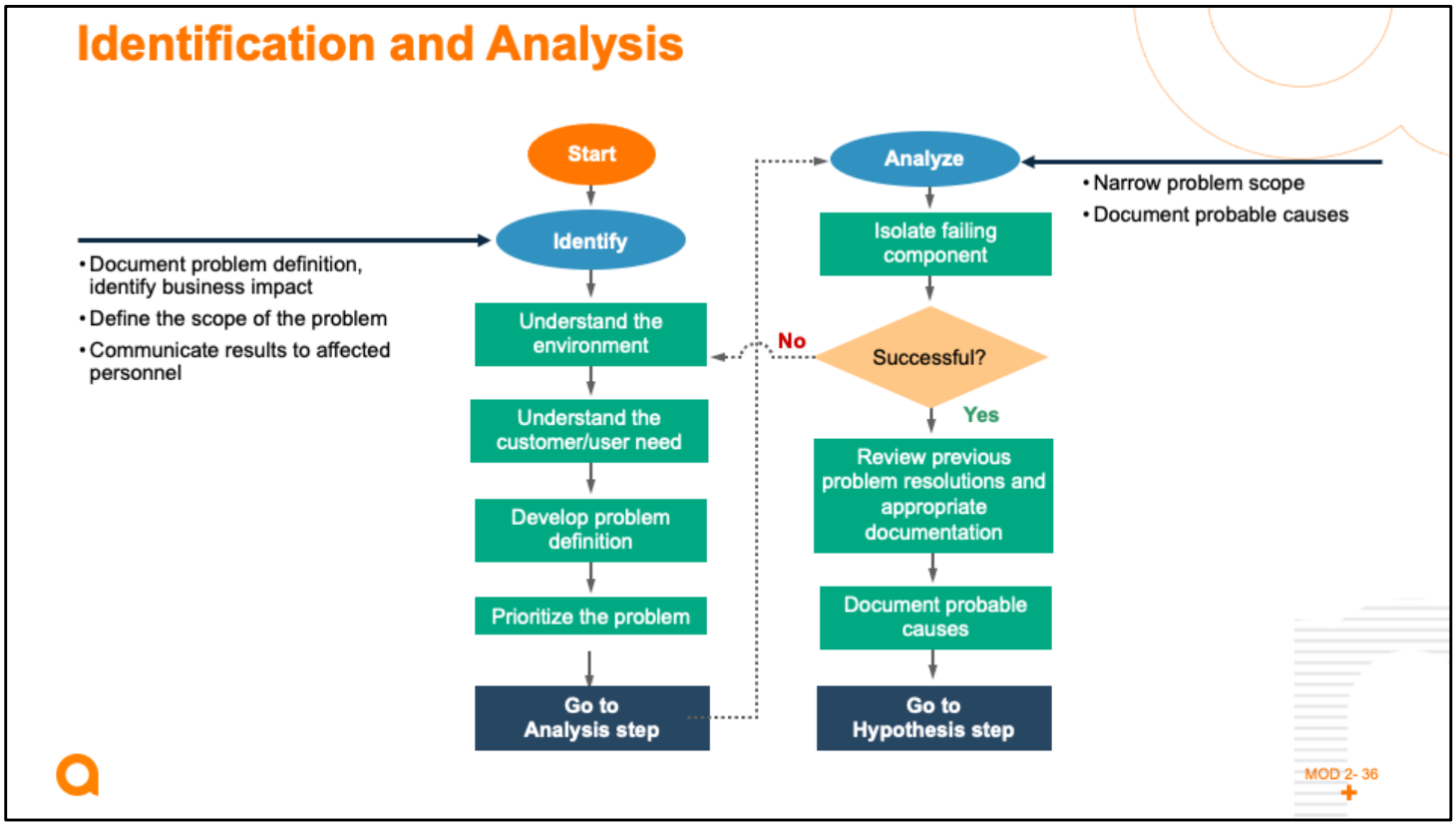
Security – It is a balancing act between locking things down tight enough to mitigate attacks, but open enough to allow required management access, L2/L3 control plane and resilience protocols, and data plane traffic. You must validate that the routing zone is properly getting packets to a security device or process, and that security process is then permitting desired traffic.

Resilience – Features like VSX, VSF, and STP help to mitigate outages, but add a layer of complexity and potential for configuration and deployment errors. You must understand how to properly monitor these features, and how they interact with the Routing and Security Zones.

QoS – Quality of Service mechanisms can provide more granular control and priority mechanisms for business-critical traffic, but can also cause unnecessary bottlenecks, or even drop vital traffic if improperly deployed. Some QoS techniques, such as rate limiting, can also be used to improve security.

Automation tools – Features like NetEdit, NAE, and the REST API can improve your situational awareness and efficiency. They can help you to monitor and diagnose the other four troubleshooting zones. Although not shown in the figure, SNMP is still well-entrenched, and widely used.

Note: Although the Automation Tools topic might be considered merely “tools to help diagnose and monitor the other four zones”, it is still a distinct area of interest, and requires much of the same configuration and diagnostics treatment as the other zones.



The first step of the six-step methodology is identification, which is an observation process. Try to observe everything, not just the apparent problem, and avoid assuming something. Because network troubleshooting primarily involves evaluating and resolving connectivity issues, the general procedure begins with an analysis of symptoms to determine the scope of the issue.

For example, it is important to determine whether the problem is affecting a single host, a group of hosts, or the entire network. If many hosts are affected, determine what they have in common. For instance, if a host can communicate with local hosts, but not remote hosts, verify connectivity with its default gateway. If all hosts in the same VLAN can communicate with local hosts, but not remote hosts, the issue may be a logical problem with the default gateway or a physical problem concerning connectivity with the default gateway. Although the default gateway performs Layer 3 forwarding on behalf of local hosts, their communication with the default gateway is done using Layer 2 addressing.

The identification process consists of doing tasks that can include:

Documenting the physical settings. The specifics will of course vary depending on the problem scenario, but some examples are the following:

- What client, server and network device hardware and software are in use?
- What is the network topology between the client and server?
- Where are the applications and services located? Determine the effects the problem has on the user/customer and the business.

Developing a problem definition—Document probable failures.

Prioritizing the problem—Prioritize based on defined user/customer policies. Is this a problem that must be investigated immediately or can it wait until you can assemble a strategy using the problem-solving methodology?

Step 2 is analysis. Analysis is the process of isolating the problem with the objective to narrow down the different possibilities. The analysis process considers such factors as the following:

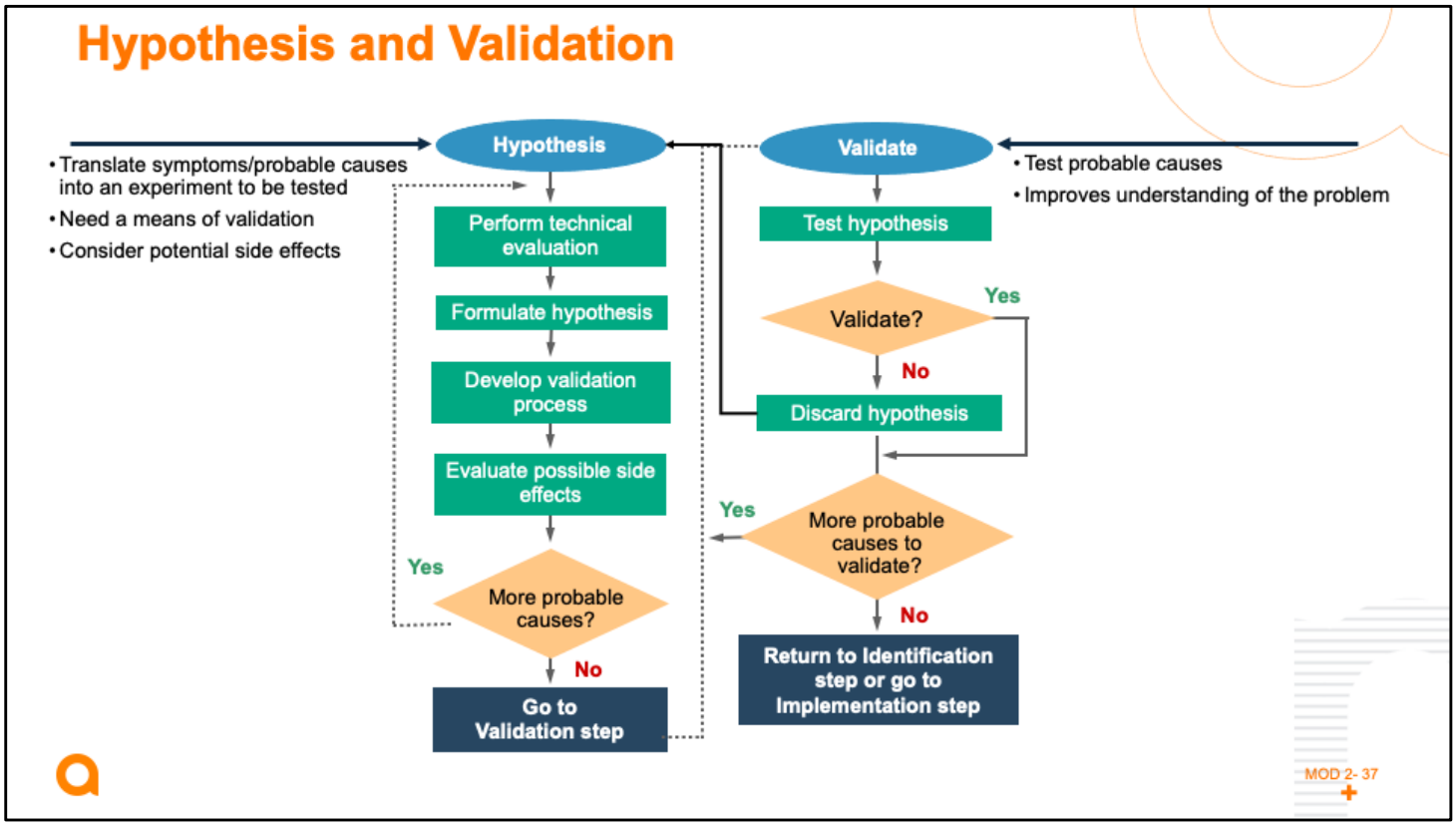
Previous changes to the system

Something new, such as networking equipment, that may have been introduced

Any changes to peripheral equipment that may have been made

Whether the hardware or software is being used correctly

With the scope of the problem having been narrowed down, that can help suggest the type of network troubleshooting tools you may want to use to test probable causes. For example, the problem investigation may involve using simple network reachability tools, such as traceroute or ping, or examination of the logs of multiple switches, or even use of a protocol analyzer.



Step 3 is hypothesis. The hypothesis step involves the evaluation of the information acquired from the analysis step to determine a number of probable causes.

Some things to keep in mind are:

What is the technical reason for the business problem?

You need a validation procedure for the hypothesis to be usable. Although your intuition may prove to be correct at times, in the business world, relying on that primarily makes it difficult for management to feel confident about the process.

Eventual resolution of the problem could create side effects, some that are not immediately obvious.

Validation, step 4, typically involves experimentally determining whether the hypothesis is reasonable. It increases the confidence level that the problem will in fact be resolved after implementation of a potential solution.

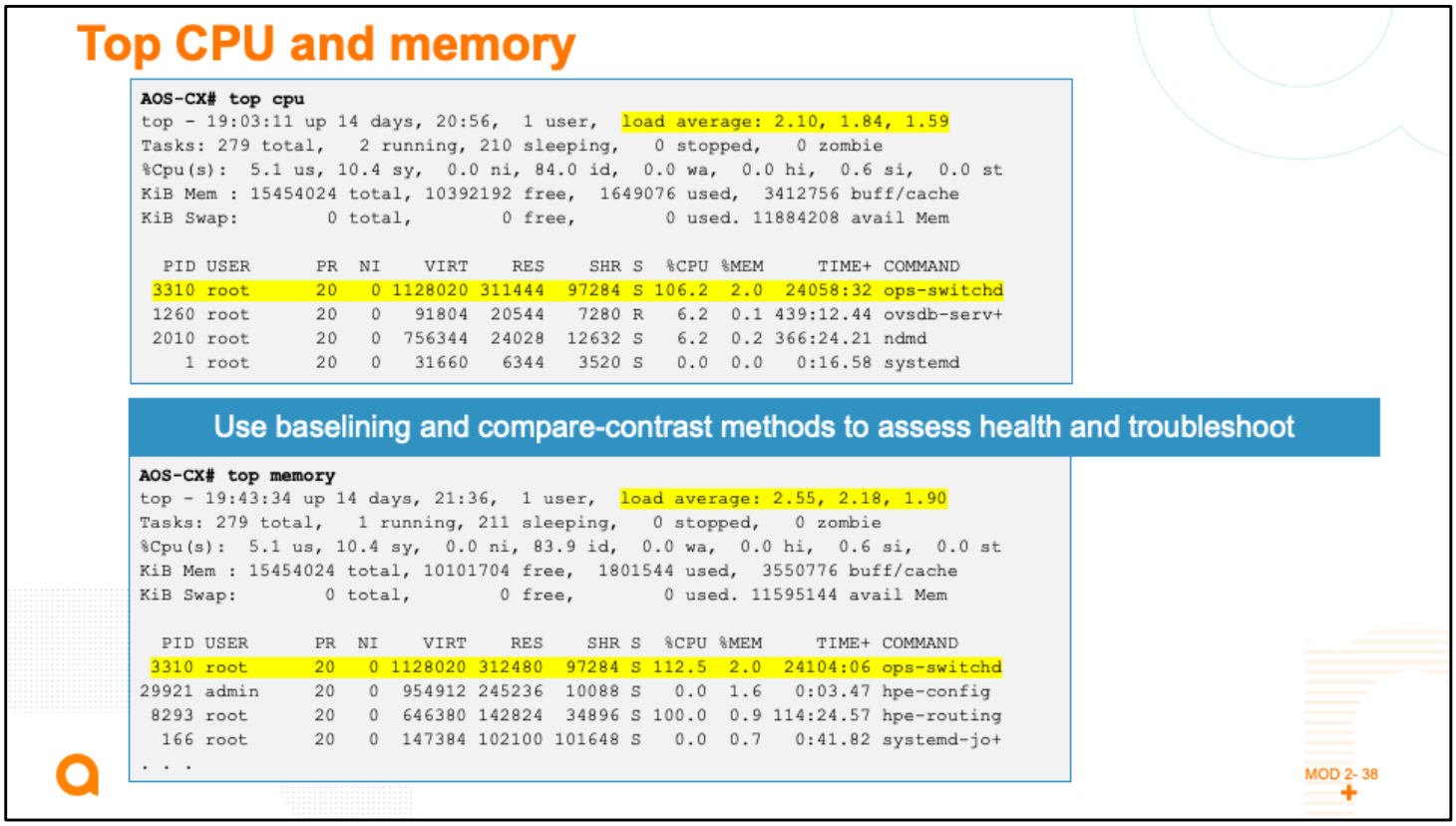
The validation step involves:

Testing each hypothesis until you validate a probable cause with a high degree of certainty. The objective is not necessarily to be 100% sure, but to balance the time criticality of resolving a problem with the information you have available.

If validation fails for all probable causes you developed, then you may need to return to the problem definition phase and start over. Despite what may appear to be time wasted, you will likely have improved your awareness of the problem situation and will have some additional

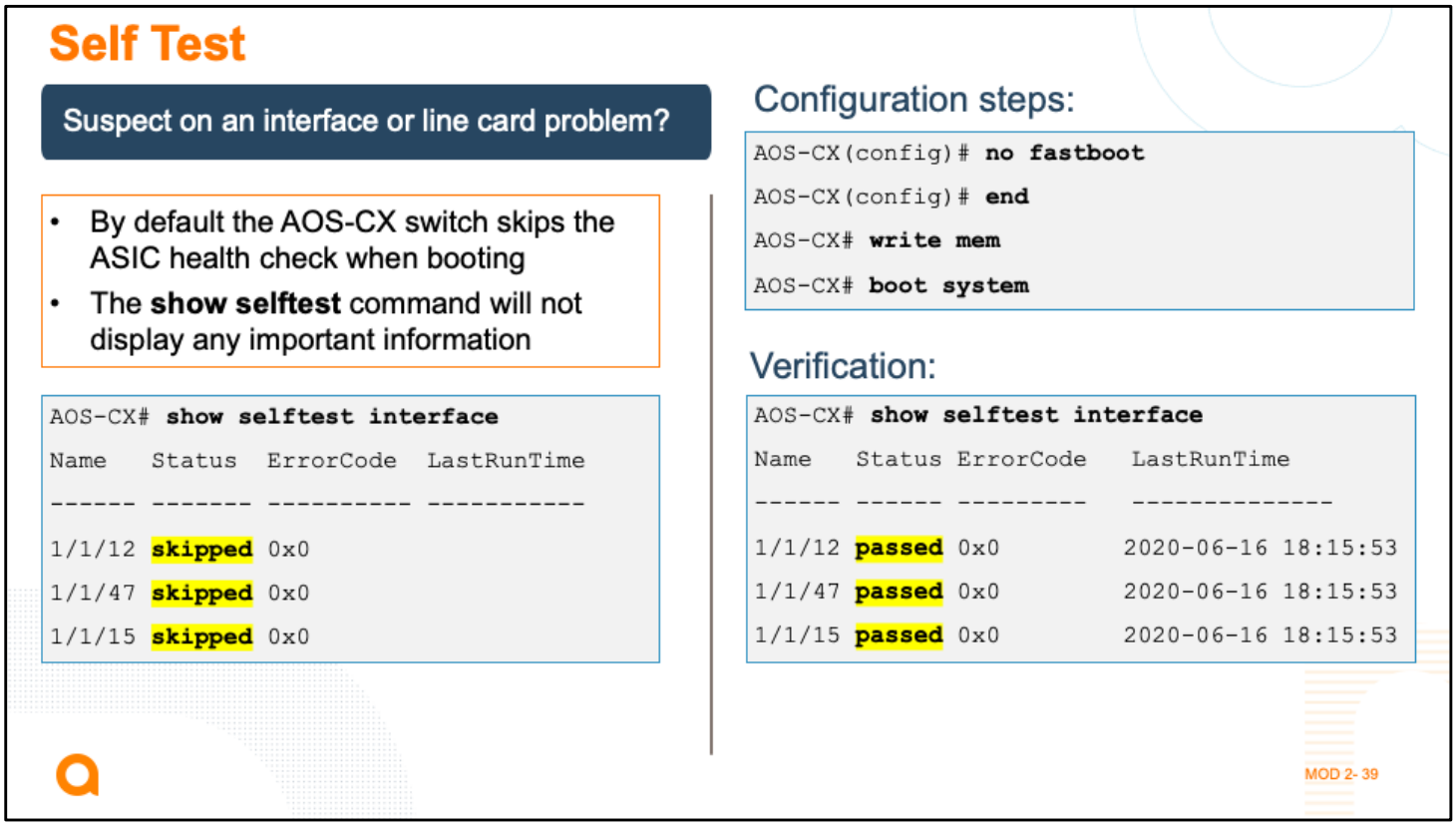
facts to use when you attempt to redefine the problem.

Note the arrow emanating from “Discard hypothesis”. If you discard your original hypothesis, you may need to go back and create a new hypothesis.



Use the command top cpu to determine overall CPU utilization. The figure highlights the 1, 5, and 15-minute averages. If utilization is higher than normal, you can then scroll down through the list of Process IDs (PID) to determine which process is causing high CPU utilization – to help direct your troubleshooting efforts. In the example, CPU utilization seems low, and so further exploration might not be required. However, notice that the process “ops-switchd” is using suspiciously high CPU utilization. This could be due to something like a Layer 2 bridge loop, or some other runaway process. Perhaps the spanning tree protocol is disabled or misconfigured. You might be able to use the command show events –d ops-switchd (or whatever process seems suspicious) to further explore the issue.

The top memory command would be used in a similar fashion, except of course focused on which processes are causing memory utilization to be abnormally high. Again, although overall memory utilization is low, the “ops-switchd” process also shows suspiciously high utilization. As described above, this may be due to a layer 2 bridge loop.



The AOS-CX switches perform a Power On Self Test (POST) during boot. This test includes a register read/write test, to check health of the registers and tables in each ASIC's ingress pipeline. This test is always run during platform initialization.

POST also includes front-end port loopback tests, to validate the physical port front-end interface. Malfunctioning interfaces are marked as "failed" and are thus unavailable for use. This test is only run when no fastboot is configured from global config mode. Therefore the line module and fabric status output of show selftest interface is "skipped" in the left-side example.

In the right, fastboot is disabled, saved, and the switch is rebooted. Thus, the status of show selftest interface is "passed" along with a last runtime.

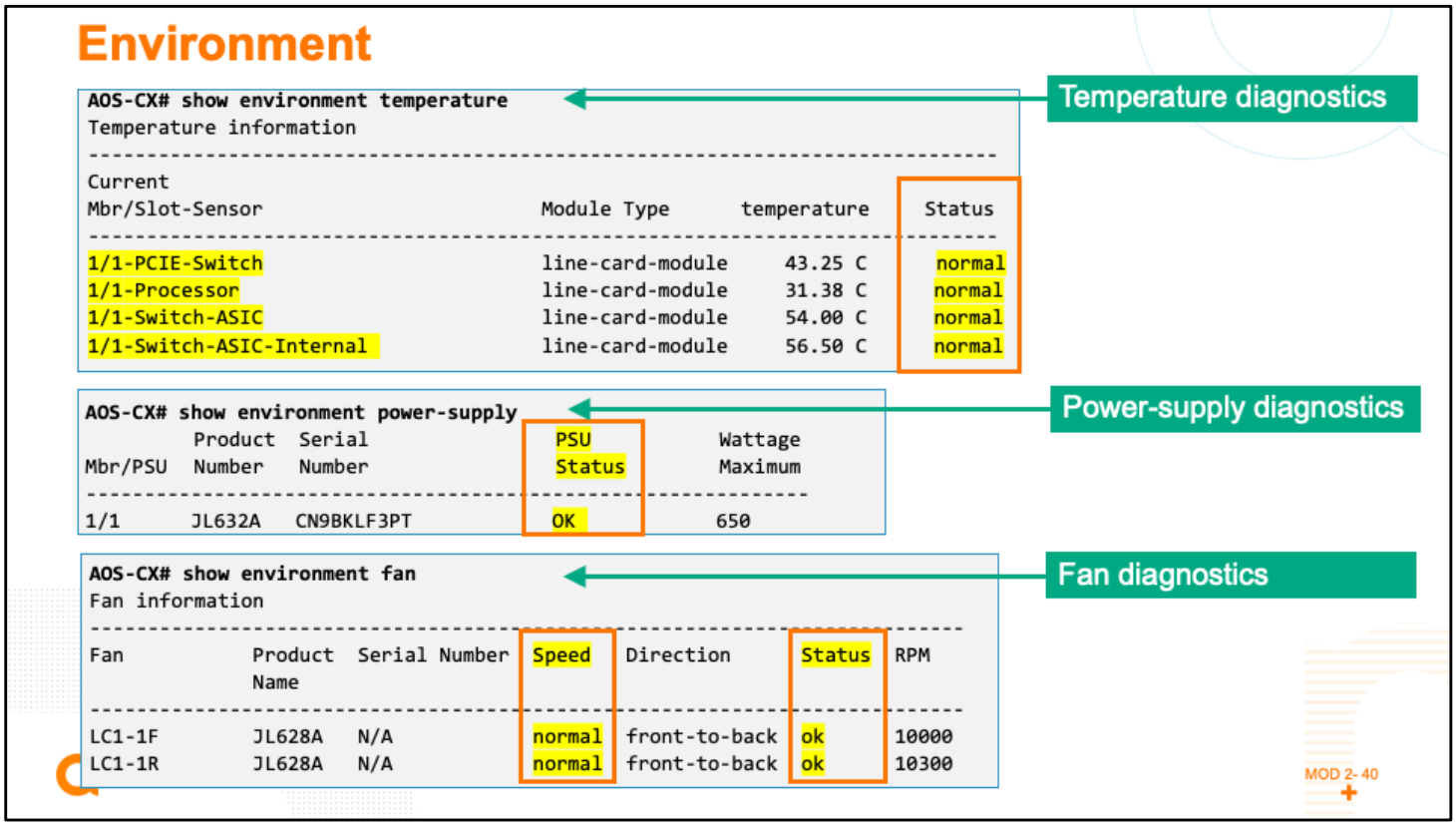
Note that the show selftest command, with no syntax options provides output for the register read/write test for linecards and fabrics, which is run regardless of the configuration of fastboot. The output show selftest interface of course provides interface-specific information, only if fastboot is disabled. You might disable this option if you suspect an interface or line card issue that has gone undetected.

All options for the show selftest command are listed below:

show selftest [brief] [vsx-peer]

show selftest line-module <SLOT-ID>

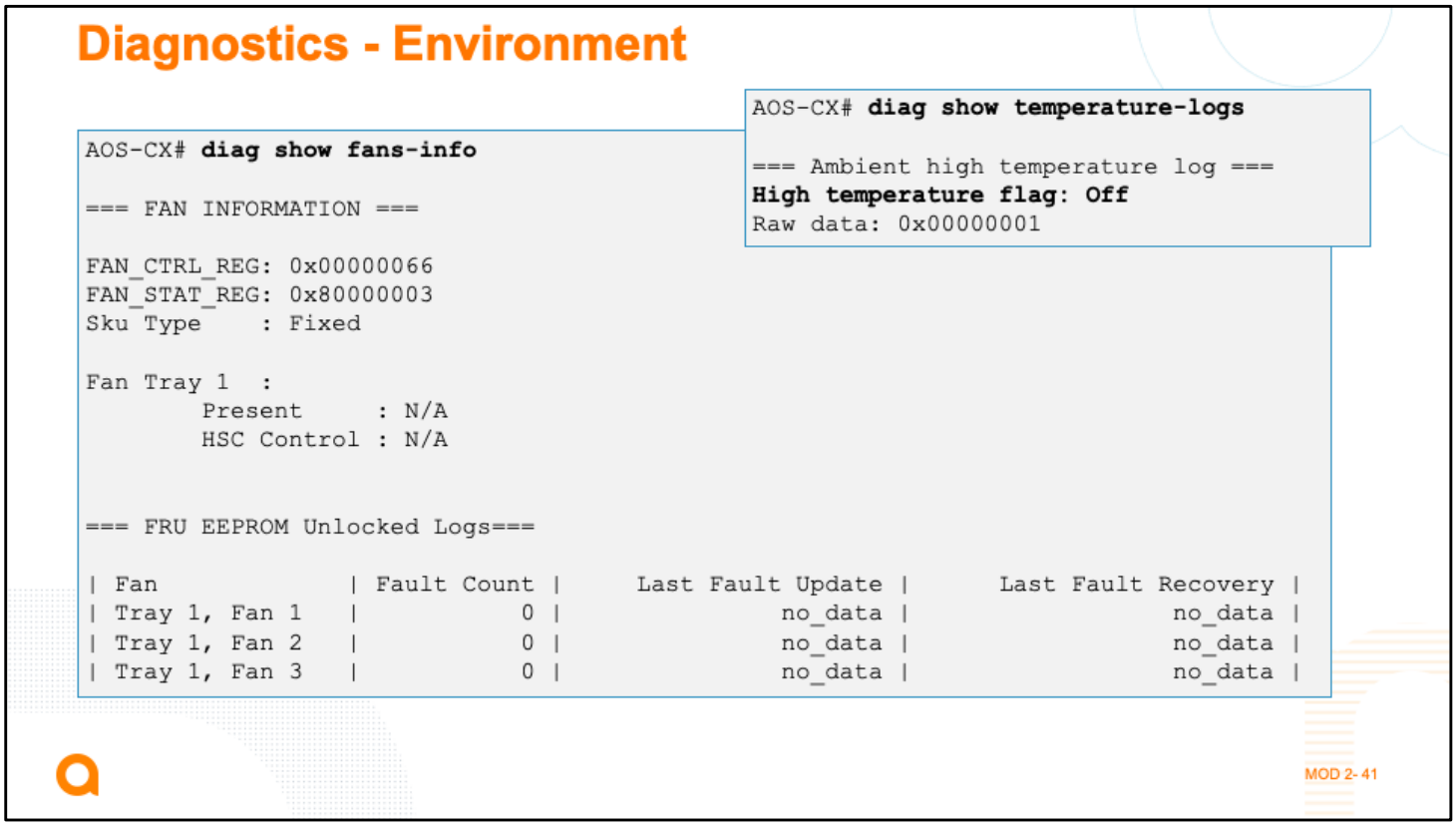
show selftest line-module <SLOT-ID> interface [brief] [vsx-peer]
show selftest interface [<PORT-NUM>] [vsx-peer]



The figure shows the show environment options – temperature, power-supply, fan, and led. There is also a show environment vsx-peer command, not shown. You can use show environment with no options to see all available environment options in a single output – temperature, power-supply, fan, etc.

Use these commands to validate environment health and component status. In the top example, all is OK – status is normal. A status of critical or emergency would be cause for alarm. Assess whether this issue isolated to only this device, or all devices in a particular wiring closet or data center area. You could use show environment temperature detail to see fan status for each mbr/slot sensor, along with current, minimum, and maximum temperatures.

Similar output and options are available for the other commands shown, which you can explore during the lab activities. Extended options for certain commands may be model-specific.



The example shows the fan diagnostic information. There have been no faults logged since this device has been booted.

The inset example shows temperature logs – the high temperature flag is off, meaning that this device has not overheated since being booted.

You might also explore the show diag power-supply command, which provides detailed output related to the power supplies.

Again, remember that option availability is dependent on platform and OS version. For example, AOS-CX v10.04 supports all three of the commands mentioned here on a 6300-series platform, but the 8325 only supports the show diag power-supply option.

Diag on-demand

Bad line module?

```
AOS-CX# diag on-demand line-module 1/1
Performing diagnostic tests.  Please wait ...
Fetching Test results.  Please wait ...

Module                ID    Diagnostics Success
                            Performed
--------------------- ----- ----------- -------
LineModule            1/1             7     100%
```

Bad management module?

```
AOS-CX# diag on-demand management-module
Performing diagnostic tests.  Please wait ...
Fetching Test results.  Please wait ...

Module                ID    Diagnostics Success
                            Performed
--------------------- ----- ----------- -------
ManagementModule      1/1            14     100%
```

MOD 1-42

Depending on the platform, you may be able to perform on-demand diagnostics of certain switch components. For example, the commands shown are available starting with AOS-CX v10.04 on a 6300 and 6400 switch series. However, no on-demand diagnostics are available on other platforms.

You can also the show diagnostic brief command to obtain a similar output, this command as well is available on the 6300 and 6400 switch series.


Lab Activity
Lab 02


Rev #20.41
Monitoring and Automation
Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)
aruba
a Hewlett Packard Enterprise company

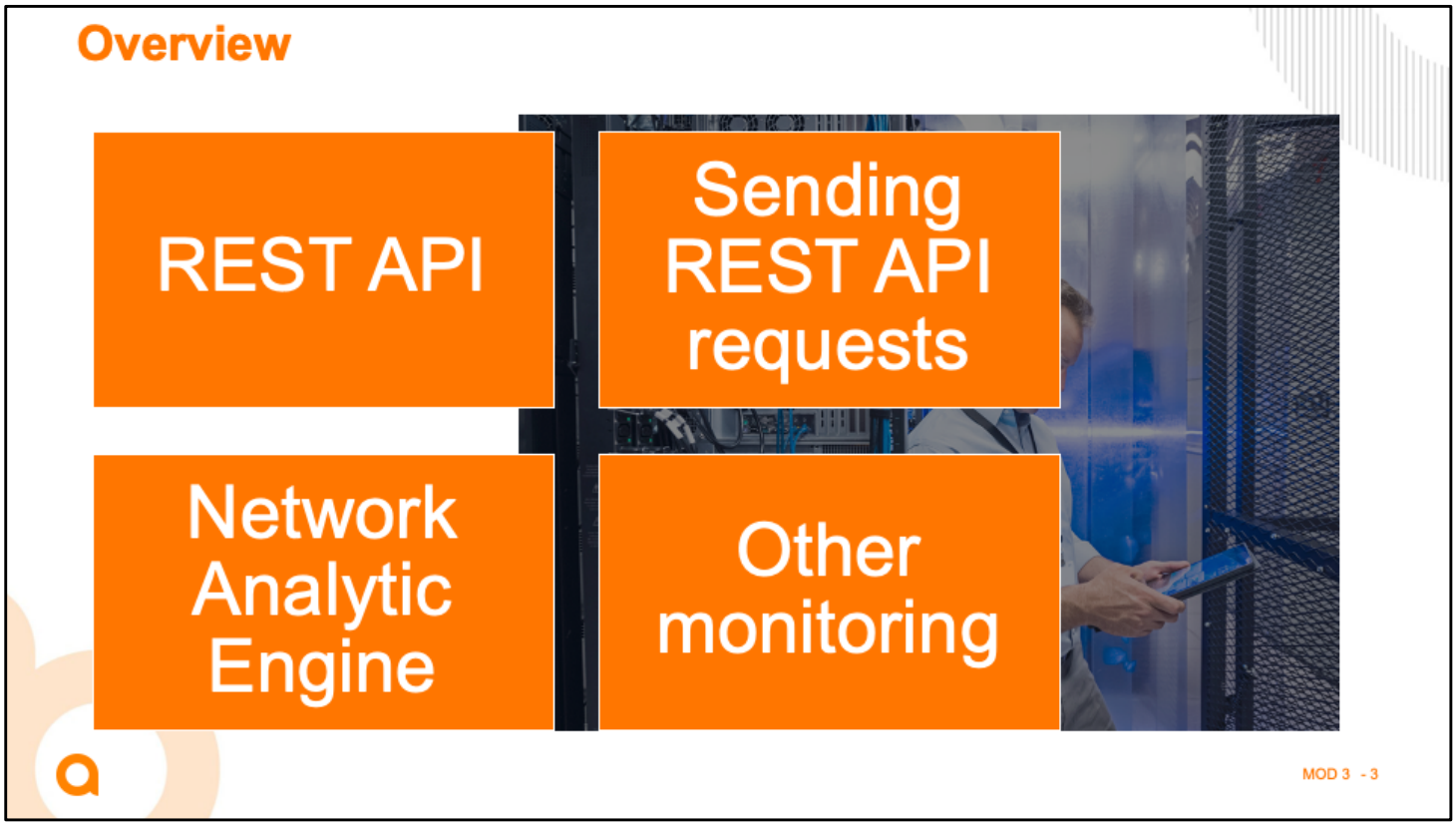


After completing this module, you will be able to describe REST API operation and methods. You will be able to enable, login, and logout of the REST interface, and to describe and use tools to send REST requests.

You will leverage REST API to configure switches and retrieve switch information. This includes the Swagger resource reference and interface, the cURL CLI tool and library, and the Postman interface – used to create/script REST requests.

You will be able to describe Aruba Network Analytics Engine (NAE) scripts, agents, types, and resources. You will also be able to explain script components and functions, and to troubleshoot NAE.

Finally, you will be able to describe and use other monitoring tools such as IP SLA, SNMPv3, and NetEdit.



One of the best things you can do to mitigate risk and maintain healthy systems is to increase your situational awareness – your perception of all network elements and related events over time; your understanding of these systems and events; your ability to detect trends.

This module covers some powerful tools toward this end, starting with the REST API. You will explore REST API operation, and how to use this tool to programmatically configure switches and retrieve switch information.

Then you will learn how to use three common tools to send REST API requests to an AOS-CX device. This includes Swagger, cURL, and Postman.

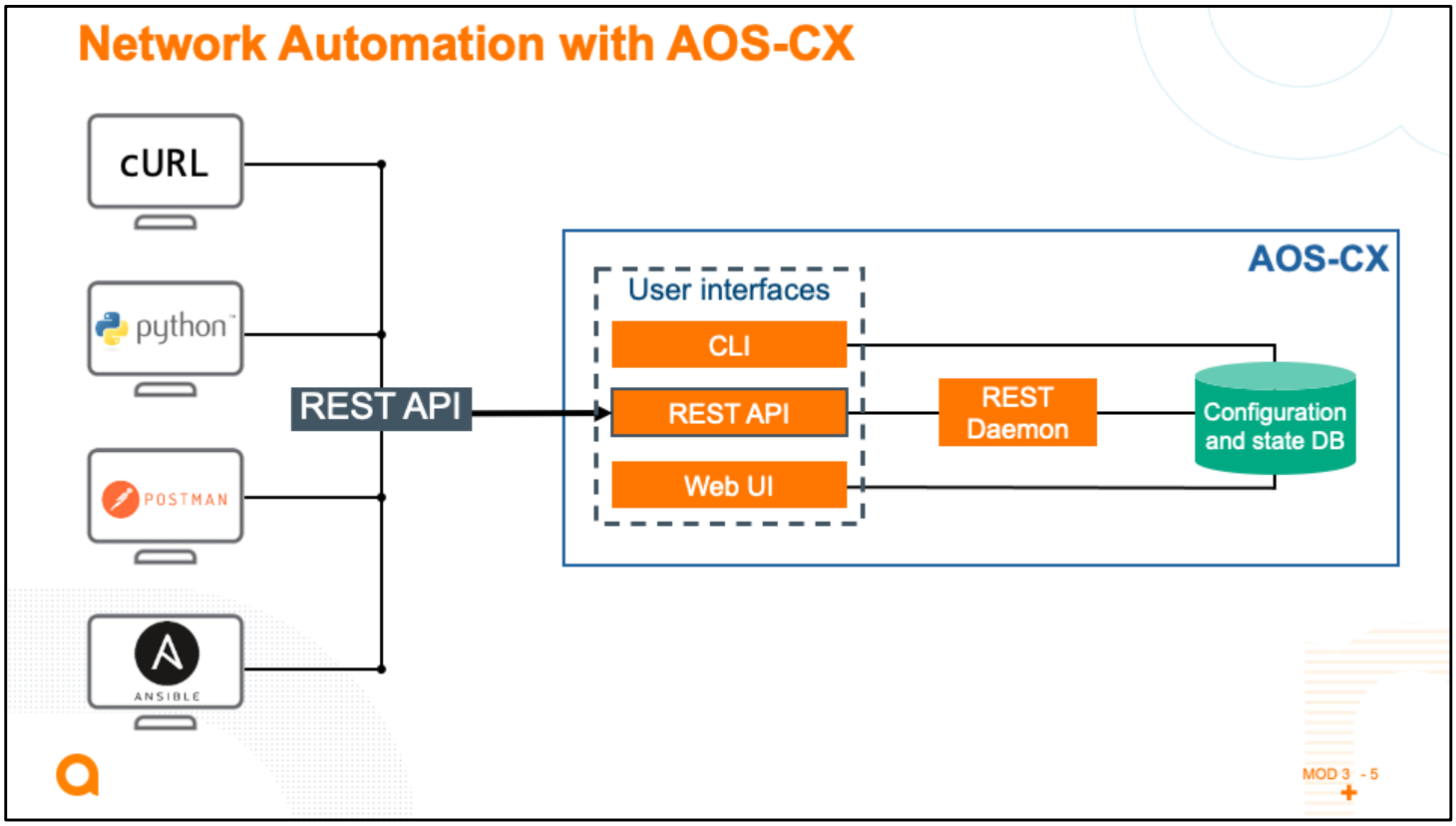
You will learn about Aruba's powerful Network Analytic Engine (NAE). You will learn how to leverage NAE scripts and agents, script types, and script sources. Then you explore NAE components – header, import statements, manifest, parameter definitions, and the agent class constructor. Then you move on to agent functions and NAE troubleshooting.

Finally, you will learn to use other monitoring tools. This includes IP SLA, SNMPv3, and NetEdit topology and logs.

aruba
a Hewlett Packard Enterprise company

REST API

Introduction



The AOS-CX operating system has been built for programmability and automation. The main component in this architecture is the Configuration and State Database which permits efficient access to network functions and states from different user interfaces. Three user interfaces are available:

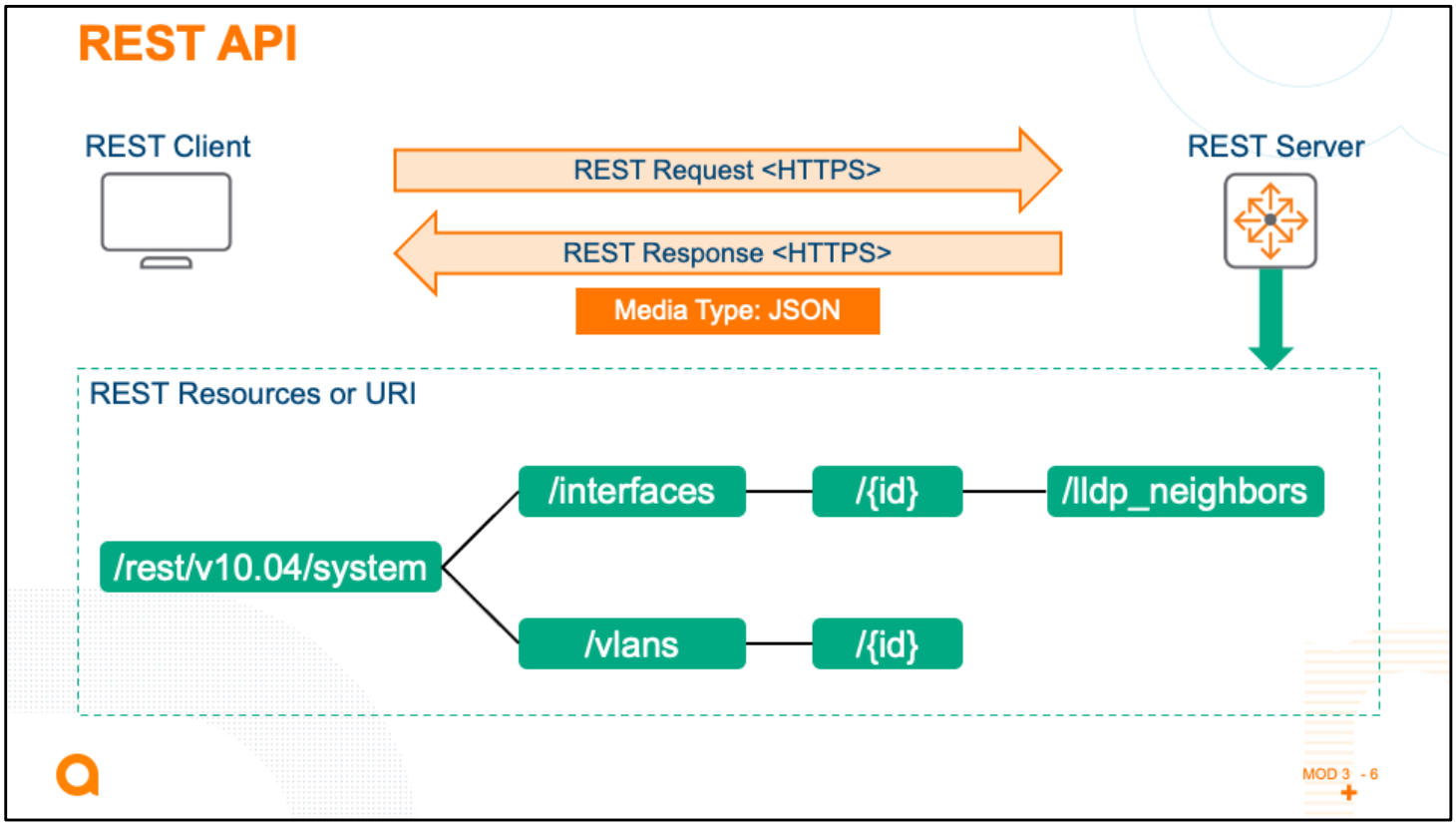
Command Line Interface (CLI): can make changes and retrieve information to the Configuration and State Database directly.

REST API interface also called NGINX : primarily used to receive REST API calls from external entities.

Web UI: The Web UI is of course the graphic interface.

Every aspect of switch configuration and state information is modeled in the AOS-CX switch database and is accessible through the REST API. Developers can access a specific configuration, statistic, or status result for any aspect of the switch through its REST URI using their preferred method such as Ansible, Postman, cURL or external applications running Python scripts.

Note. The NGINX Web Server is an open-source, high-performance HTTP server and reverse proxy. NGINX can service a large amount of requests per second without dropping, and is event based. This means that it doesn't need to spawn new processes or threads for each request, keeping memory usage low.

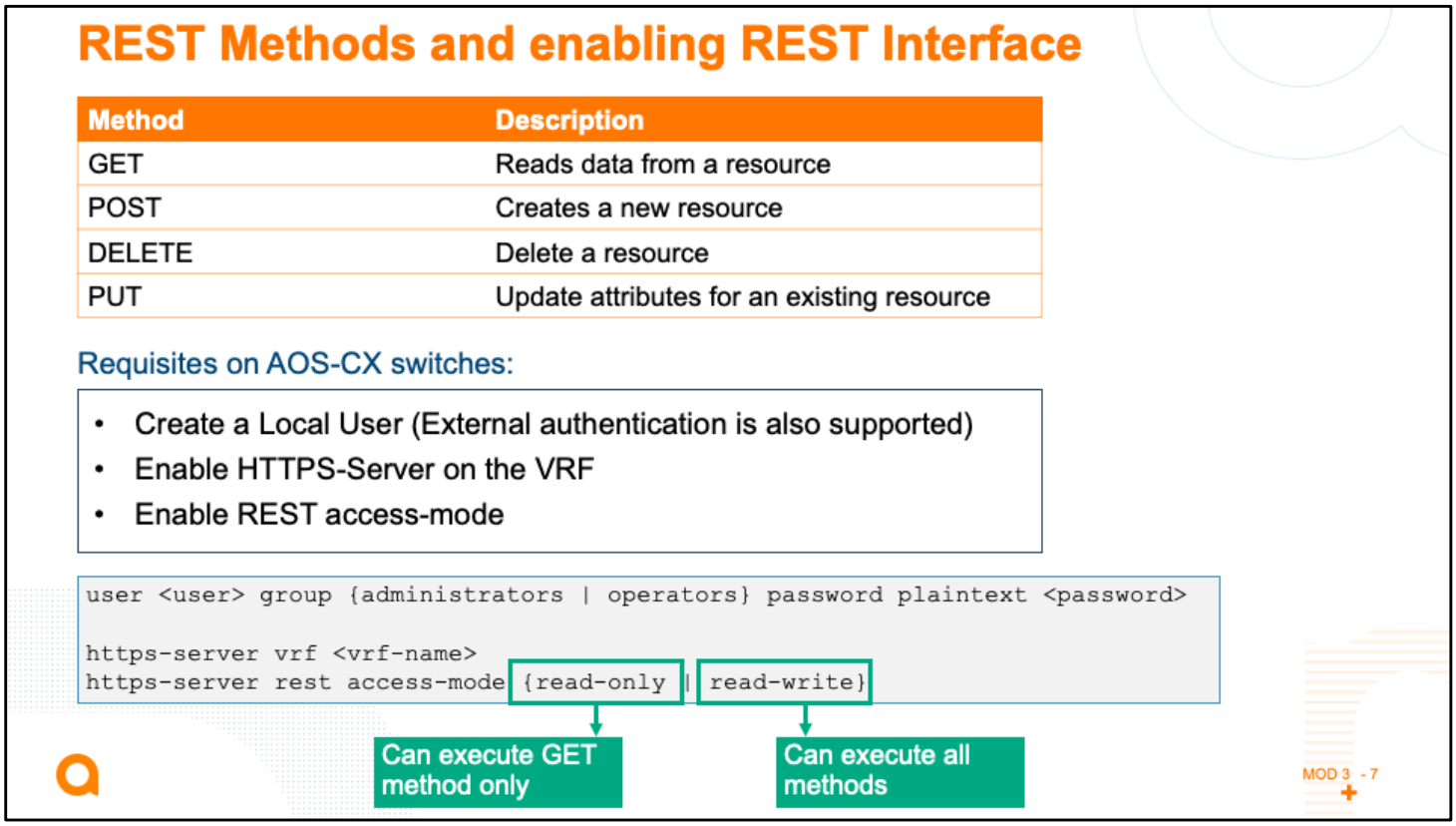


An API is a logical interface; it allows other programs to communicate with the AOS-CX switch to collect information and change settings. It creates a structured system for reporting switch status and statistics, and as for defining switch settings.

REST refers to the type of API. REST uses HTTP/HTTPS as its transport protocol. The client (the program you use to deploy scripts) contacts the server (the AOS-CX switch) and issues basic HTTP commands to create, read, update, and delete switch resources. The requests and responses contain information, which is formatted in a media type. AOS-CX switches use JavaScript Object Notation (JSON), described in RFC 4627 - a standard for representing simple data structures in plain text.

The REST API defines resources, structured into a hierarchical model, each accessible via a specific Uniform Resource Identifier (URI). For example, to collect information about configured switch VLANs, point to that resource using this URI: https://<switch-ip>/rest/v1/system/vlans

A resource can also be a container for other resources. For example, to only collect VLAN 100 information, point to the URI: https://<switch-ip>/rest/v1/system/vlans/100. So the vlan resource is a container for vlan-ids.



Every REST request a client sends is associated with a REST method which is an HTTP command. A GET command reads an existing resource. A POST command creates a resource. A DELETE command removes a resource, and a PUT command replaces attributes for an existing resource.

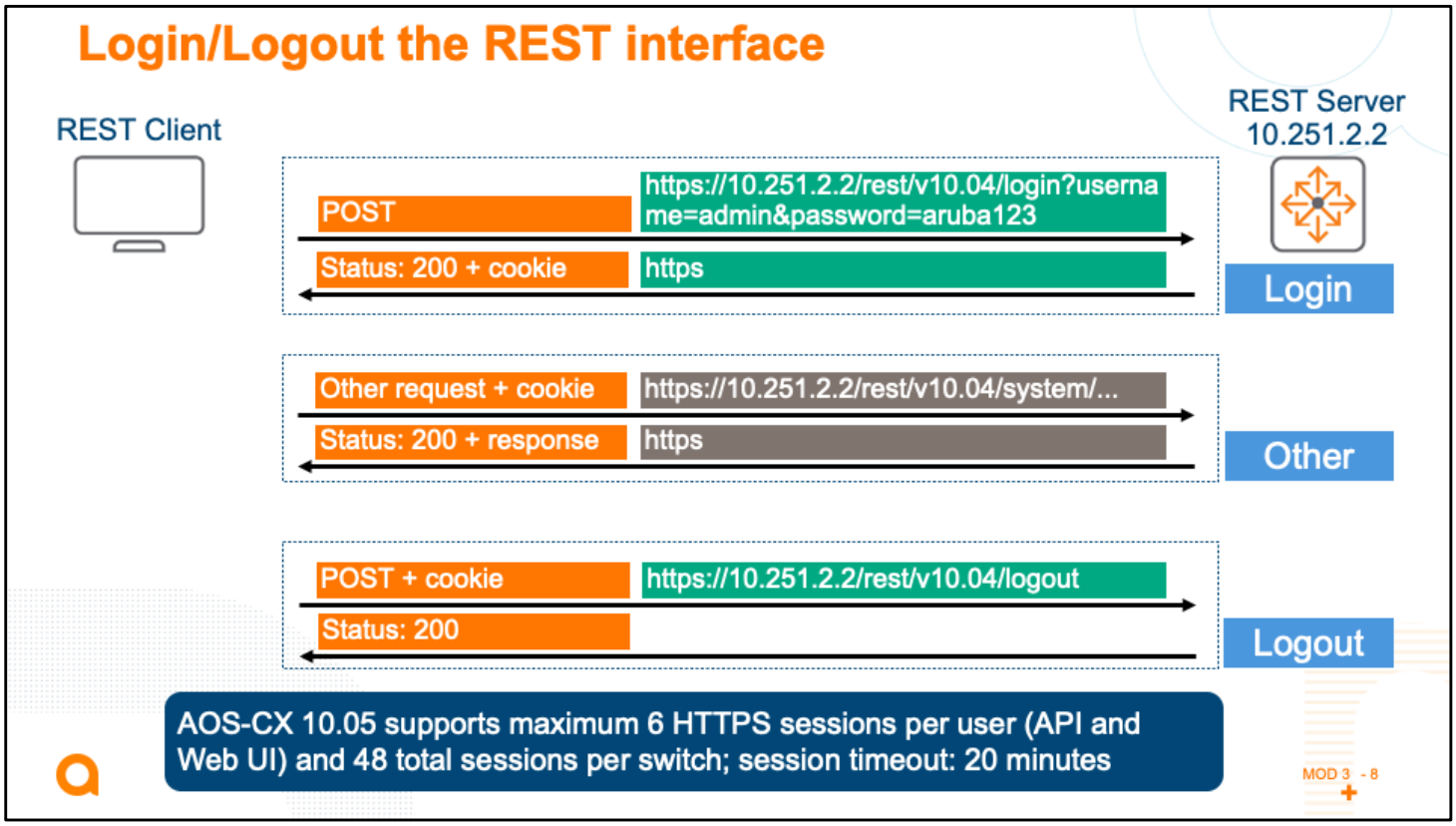
The requisites on an AOS-CX switch to run REST APIs are:

Configure a local user with its appropriate access level.

Enable HTTPS-Server on the proper VRF

Enable REST access-mode. If it is set to read-only mode, only the GET method can be executed. On the other hand if it is set to read-write all the REST API methods can be executed.

Note. If the access mode is set to read-only, executing a non-GET method will return a 404 error.



In an AOS-CX to get access to a resource using the REST API interface the client must authenticate first, this process starts by sending a POST request to rest/v10.04/login resource. Notice that the URI must include a query string with a username and password. In response if the credentials are valid the switch sends a successful status message (200) and a cookie which identifies the session, and which must be included in all subsequent API requests. Login is complete at this moment.

The subsequent requests in the next section could use any API method to read or write information. The types of request authorized depend on the user's role; operators can only use the GET method while managers can use all methods. An exception is that operators can also POST to log out of the interface.

After sending all the desired requests, the script should end by logging out of the interface—that is, sending a request to the logout resource. The AOS-CX 10.05 can only support up to six HTTPS sessions per user and 48 total session at once (this number includes API and Web UI sessions). In AOS-CX a REST API session will timeout after 20 minutes of inactivity but as a good practice you should logout each time you use the REST API interface.

When the maximum session number is reached the system will not create a new session and the system will inform about this situation. To clear the previous HTTPS sessions you can use this command: AOS-CX# https-server session close all

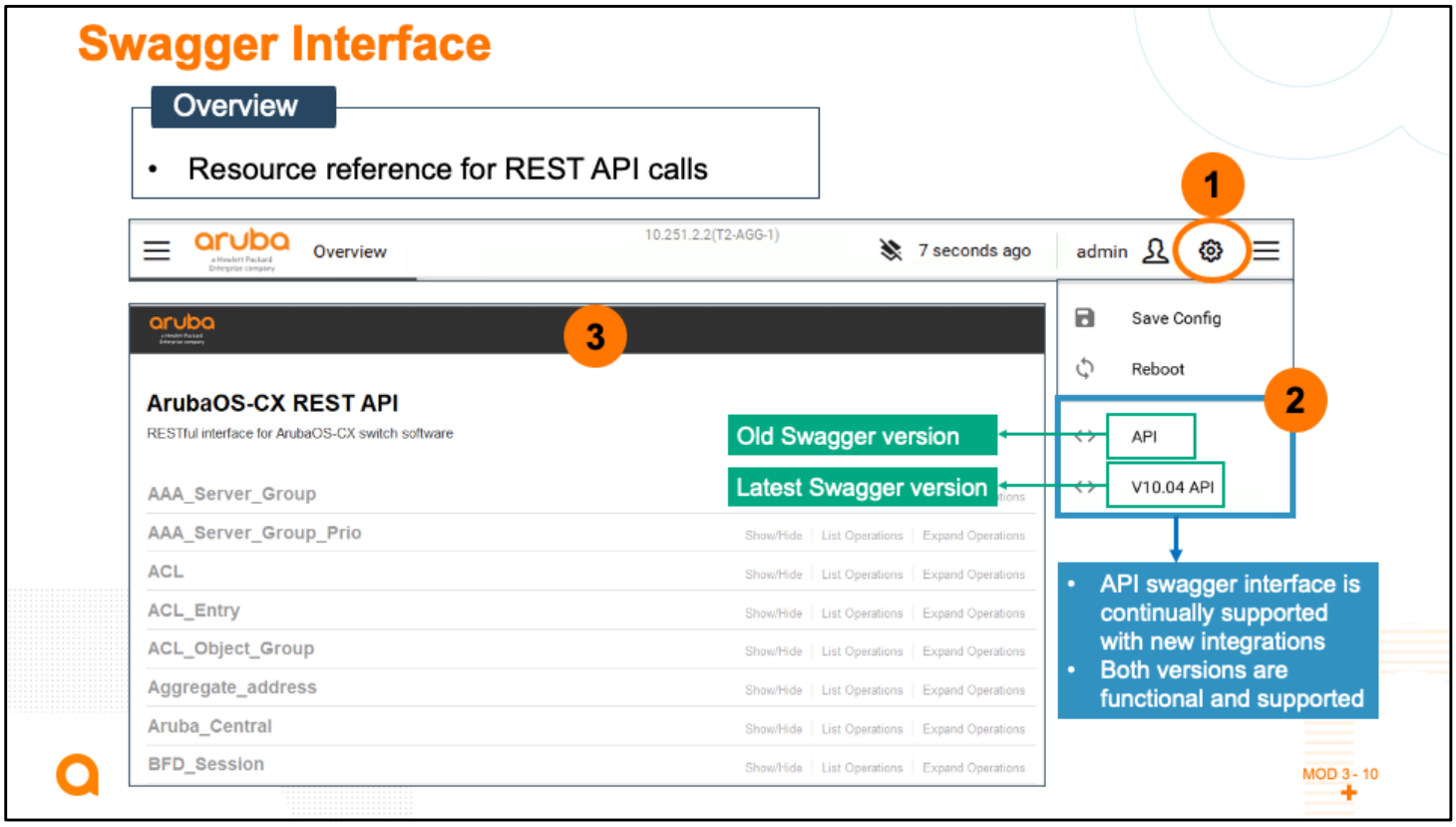
REST API Status Messages:

200: OK
201: Created
204: No Content
304: Not Modified
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
415: Unsupported Media Type
500: Internal Server Error
503: Service Unavailable

Sending REST API Requests

Swagger, cURL and Postman

aruba
a Hewlett Packard Enterprise company

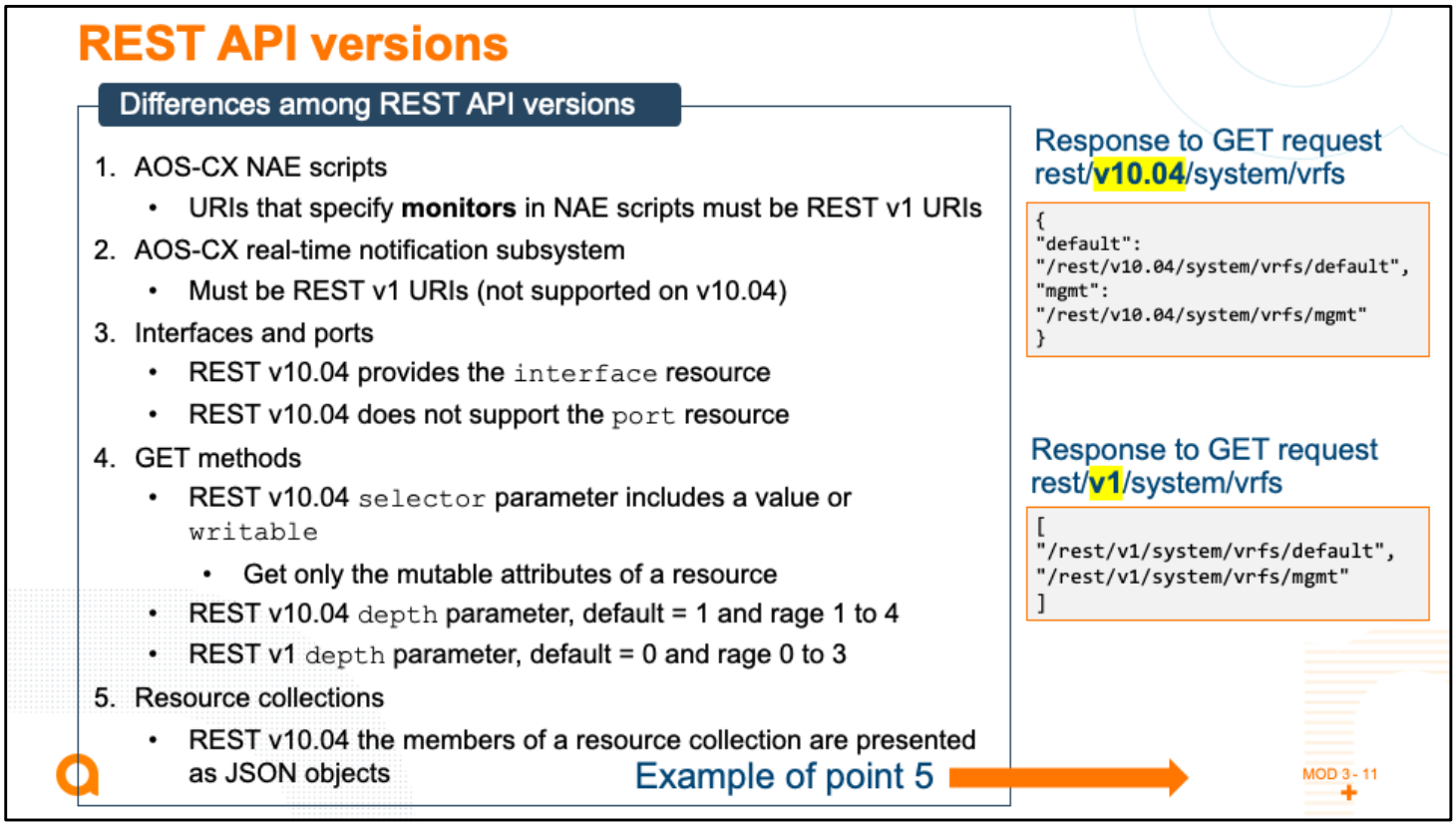


The Swagger interface provides a resource reference to developers. It is for documentation, not managing or monitoring.

Swagger lets you explore all resources in the API, along with their associated URIs. It also helps you to construct valid requests, showing the cURL command to issue that request. Use the cURL command line utility to send and receive HTTP/HTTPS requests. You can try out and validate various formats for request bodies. You can then use this knowledge when scripting an outside program.

However, be aware: actions that you take in the interface are real. A PUT request to /system/interfaces/ID, for example, will change an interface‘s settings . Therefore, you should be careful not to submit actual PUT, POST, and DELETE requests on a live system.

The API is continually supported to help enable newer integrations; this process includes the creation of new versions of the swagger interface. In the image you can see there is a new V10.04 API. Note that this does not necessarily deprecate previous versions, both v1 and v10.04 are functional and supported.



From the AOS-CX release 10.04, the AOS-CX switches support access through multiple versions of the REST API. The REST API versions supported on the AOS-CX switches are v1 and v10.04. The REST API version v10.04 is supported from AOS-CX release 10.04 and later.

The version declared in the REST request must match one of the versions of the REST API supported on the switch. The REST API version is included in the Uniform Resource Identifier (URI) used in REST requests.

In the following example, the REST API version is v10.04: https://192.0.2.5/rest/v10.04/system
In the following example, the REST API version is v1: https://192.0.2.5/rest/v1/system

Each REST API version has its own REST API Guide, you can download the related documents from this site https://asp.arubanetworks.com/downloads;products=Aruba%20Switches

Differences among REST API versions

Resources, attributes, and behaviors might differ among the REST API versions and AOS-CX software release.

AOS-CX Network Analytics Engine scripts. URIs that specify monitors in Network Analytics

Engine scripts must be REST v1 URIs.

AOS-CX real-time notifications subsystem. URIs that identify resources in the AOS-CX real-time notifications subsystem must be REST v1 URIs and are not supported by the REST v10.04 API.

Interfaces and ports. The REST v10.04 API provides the interfaces resource to configure and get information about switch ports and interfaces of all types. The ports resource is not supported by the REST v10.04 API.

GET method. The GET method query parameters differ between REST v1 and REST v10.04:

- The REST v10.04 selector parameter includes a value of writable, which enables you to get only the mutable attributes of a resource.
- The REST v1 depth parameter has a default of 0 and a range of 0 through 3.
- The REST v10.04 depth parameter has a default of 1 and a range of 1 through 4. The REST v10.04 depth=1 is equivalent to the REST v1 depth=0, and so on.

Resource collections. In REST v10.04, the members of a resource collection are represented as JSON objects, where the key is the index, and the value is the URI of the resource.

For example, the response to a GET request to /rest/v10.04/system/vrfs is as follows:

```
{
"default": "/rest/v10.04/system/vrfs/default",
"mgmt": "/rest/v10.04/system/vrfs/mgmt"
}
```

In contrast, the response to a GET request to /rest/v1/system/vrfs on the same switch is as follows:

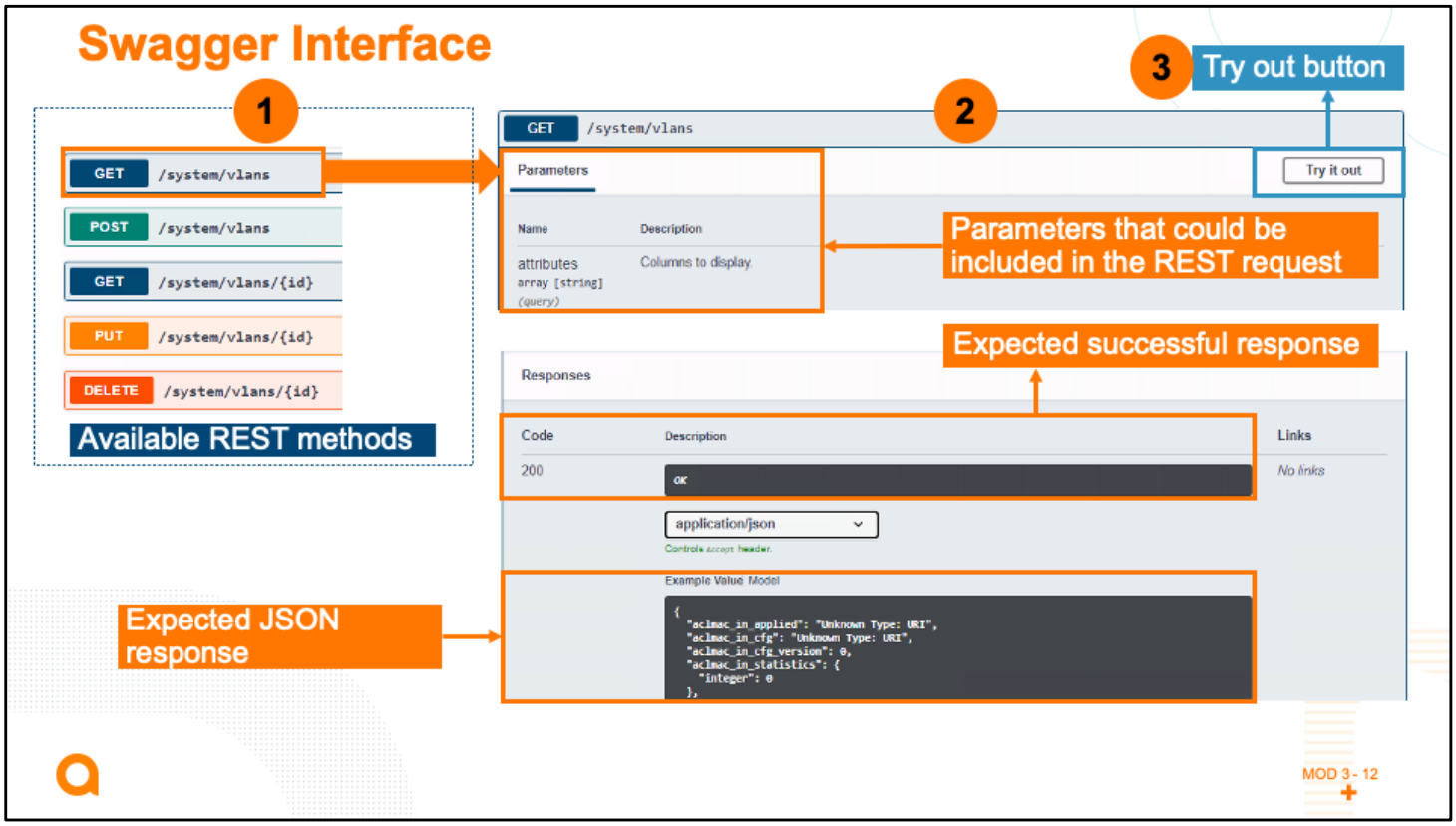
```
[
"/rest/v1/system/vrfs/default",
"/rest/v1/system/vrfs/mgmt"
]
```

About the Real-time notification subsystem

The AOS-CX REST API, combined with built-in databases that provide configuration, state, statistical data, and time-series data for the features and protocols running in the switch, provides a flexible means for switch programmability. Each resource or collection of resources inside the switch is uniquely identified by its URI.

Clients can use the REST API to request information about resources. However, this polling ability does not address the specific use cases in which network management systems need to receive live data or real-time events from the switch. There is a need to have a live notification subsystem that provides the remote network management system with real-time information about any changes that occur in the switch. Timely information about changes is important for troubleshooting and statistical data analyses, as well as for the immediate reaction to real-time events.

The AOS-CX real-time notifications subsystem enables external clients to connect to the switch through a secure WebSocket Protocol connection and to receive real-time notifications about the switch resources and the configuration changes, state changes, and statistical information that interest them.

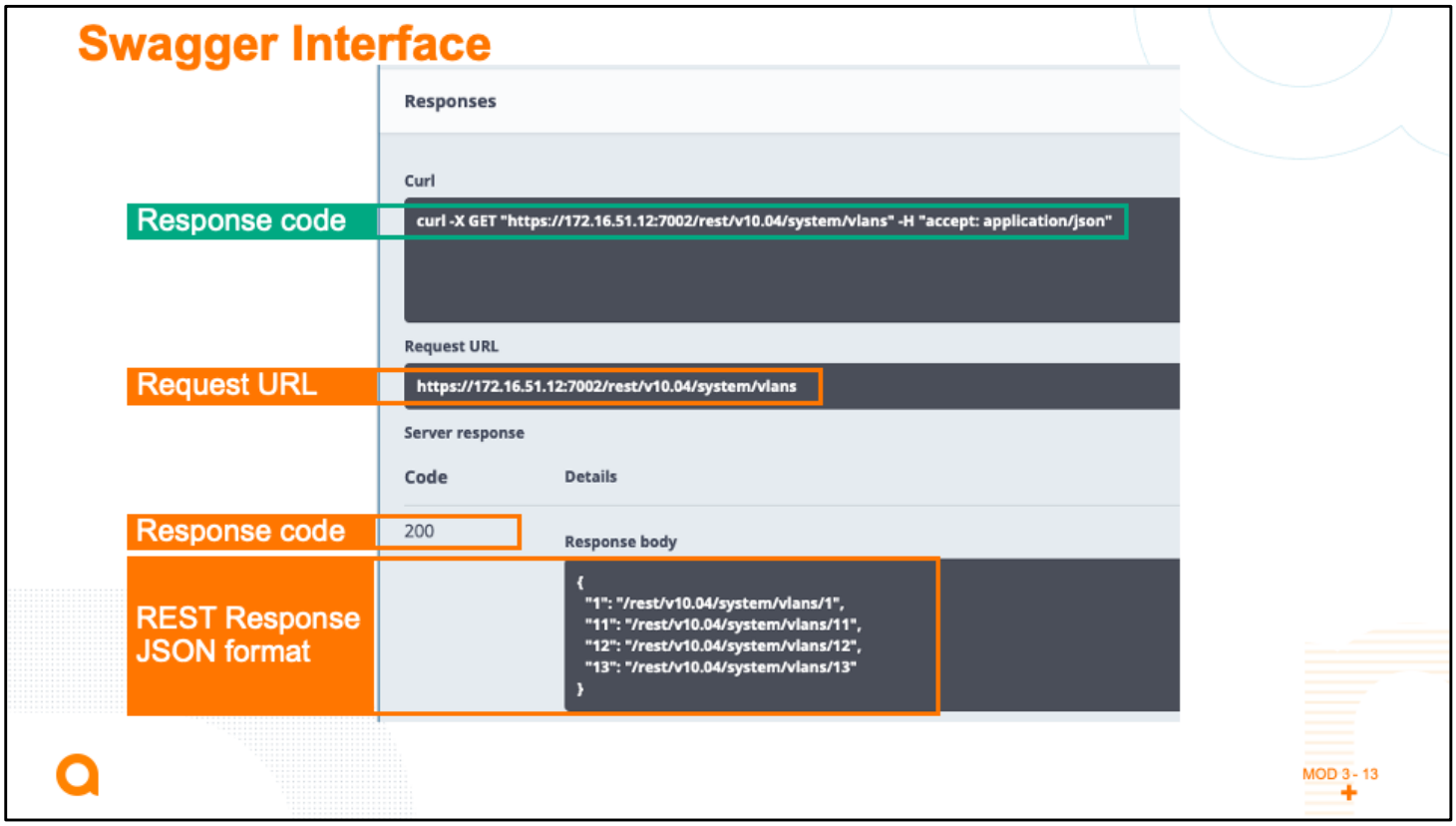


Select a resource to see available REST methods. The figure shows the VLAN resource and five available REST methods:

- GET all VLANs
- POST all VLANs
- GET specific VLAN-ID
- PUT specific VLAN-ID
- DELETE specific VLAN-ID

In this example we select the GET /system/vlans resource to see specific information about this REST method. When you select a method a new section (2) appears on the screen, this section includes the parameters that can be used for a refined REST request (these include attributes, depth, count, selector, and filters for the GET REST method) and all possible responses, in the slide we present the response 200, which correspond to an expected successful response; below this section you can find more information about the expected JSON response.

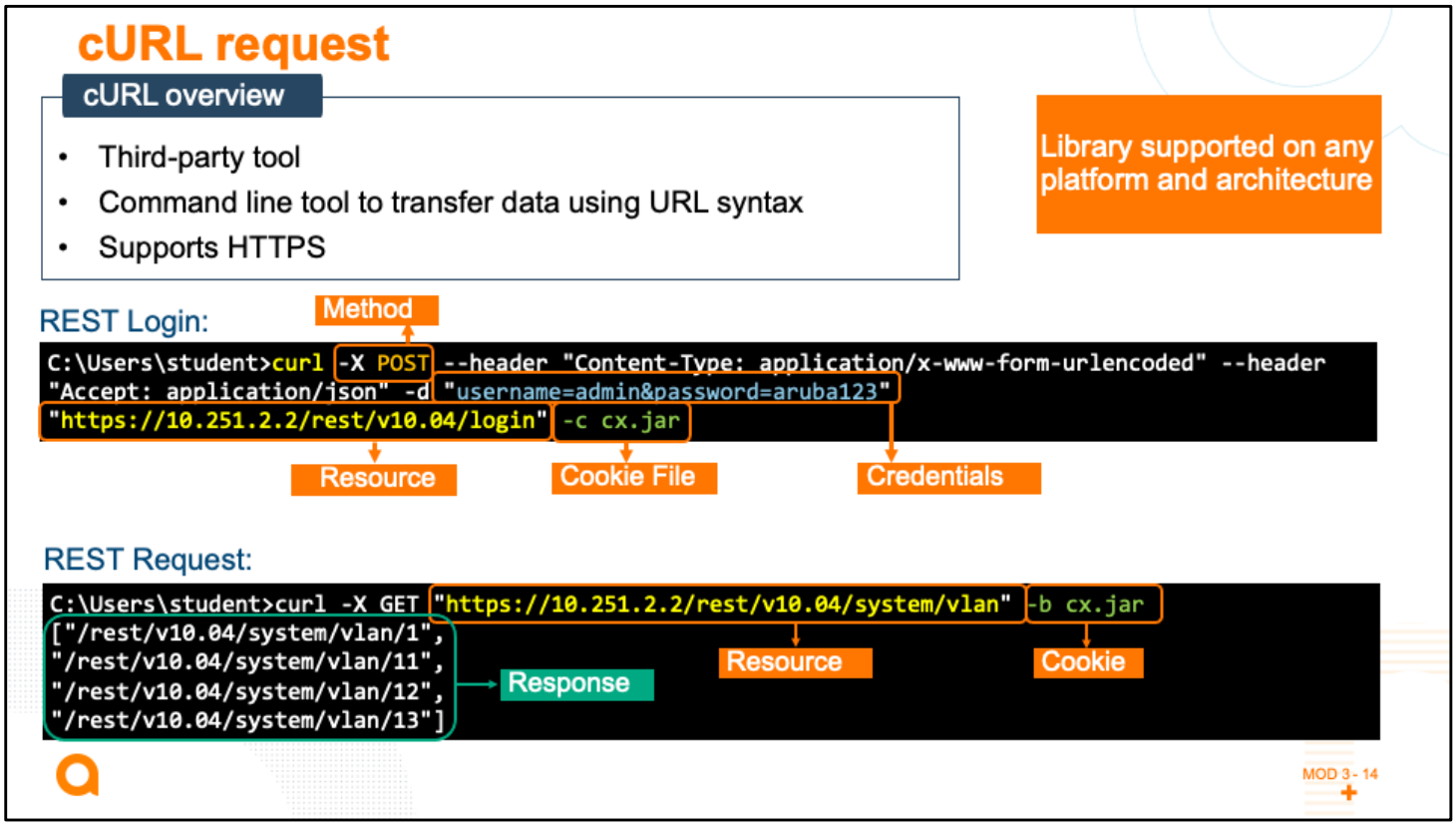
If you require to test this method, you must click on the top right button (Try it out).



The figure shows the response for the GET request to the resource /rest/v1/system/vlans.

Notice that the response includes four VLAN-IDs (1, 11,12 and 13) and the successful response code is 200.

Note. The first section on the output is referring to the cURL command line you can use to request specific resource information - in this case the VLANs. You will explore this more deeply in the next section.



cURL is a third-party tool based on command line that helps to transfer data using URL syntax.

The primary focus of the cURL project is on two products:

curl, the command-line tool

libcurl the transfer library with a C API

Use the command-line tool to get or send data, including files, using URL syntax. cURL supports HTTPS and performs SSL certificate verification by default when you specify a secure protocol like HTTPS.

Use libcurl to add client-side file transfer capabilities to your software, on any platform, any architecture, and for any purpose. libcurl is licensed very liberally, so it is easy to adopt. The libcurl library is portable. It builds and works identically on many platforms, including:

Microsoft Windows

Android

Linux

iOS

MacOS

The figure shows how to use cURL to retrieve VLAN information from the switch via the REST API. This example is on a regular Windows 10 station, which comes bundled with the curl tool as of OS version 1804.

First you login into the system using the right credentials:

curl -X POST --insecure --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: application/json" -d "username=admin&password=aruba123" "https://10.251.2.2/rest/v10.04/login" -c cx.jar

Notice the following elements:

curl –X POST indicates method used, The -X flag specifies a custom request method to use when communicating with the HTTP server. The GET method is used by default.

-- insecure option explicitly allows curl to perform “insecure” SSL connections by avoiding the digital certificate validation.

- d option includes authentication credentials and specifies the REST resource. In this example it is the login resource.

-c option indicates that the “cookie jar” received will be stored in a specific file. In this example, the file is cx.jar

Note. The “jar” extension is not referring to JAVA but to the “cookie jar” file.

The second step is to make a REST request to receive the VLAN information. Use the following command.

curl -X GET --insecure "https://10.251.2.2/rest/v10.04/system/vlan" -b cx.jar

Notice that the command now includes the GET method and that the “cookie jar” file previously obtained is also included within the request.

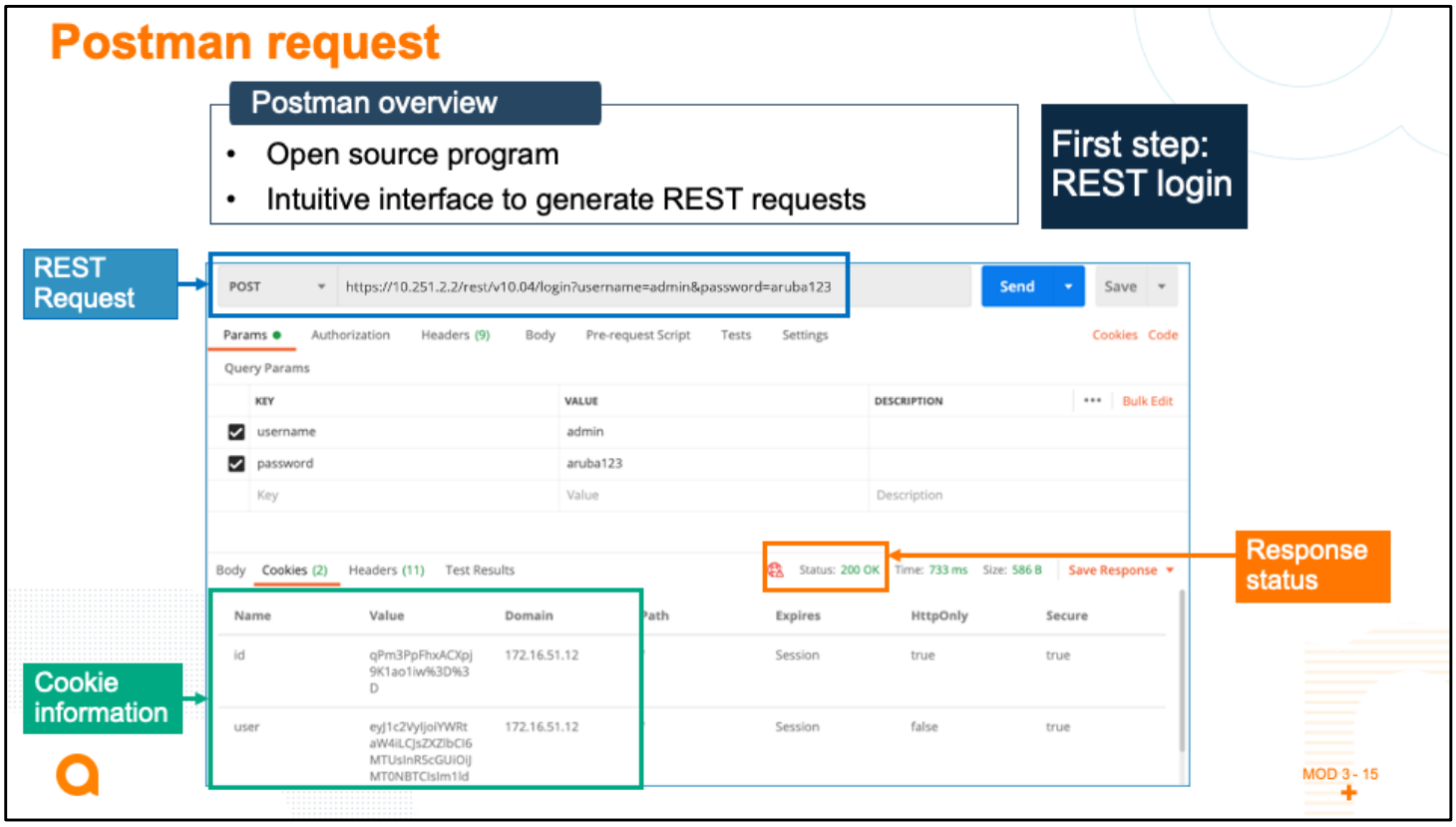
The response is the list of VLANs created in the switch:

["/rest/v10.04/system/vlan/1",
"/rest/v10.04/system/vlan/11",
"/rest/v10.04/system/vlan/12",
"/rest/v10.04/system/vlan/13"]

Note. Remember that the swagger interface will help you to know the command syntax to use for a target resource.

Note. For more information about the cURL command used in AOS-CX you can refer to ArubaOS-CX 10.05 REST v1 API Guide and ArubaOS-CX 10.05 REST v10.04 API

Note. You can learn more about the cURL project in https://curl.haxx.se/



Postman is an open source program used to create and script REST requests. For this task, it offers a more intuitive interface than cURL. Postman stores the cookie automatically, so any subsequent calls within that tab would make use of the session cookie if you did the login prior.

Remember that the first step for a REST API call is to login, In the example, the REST request is used to login into the switch REST interface.

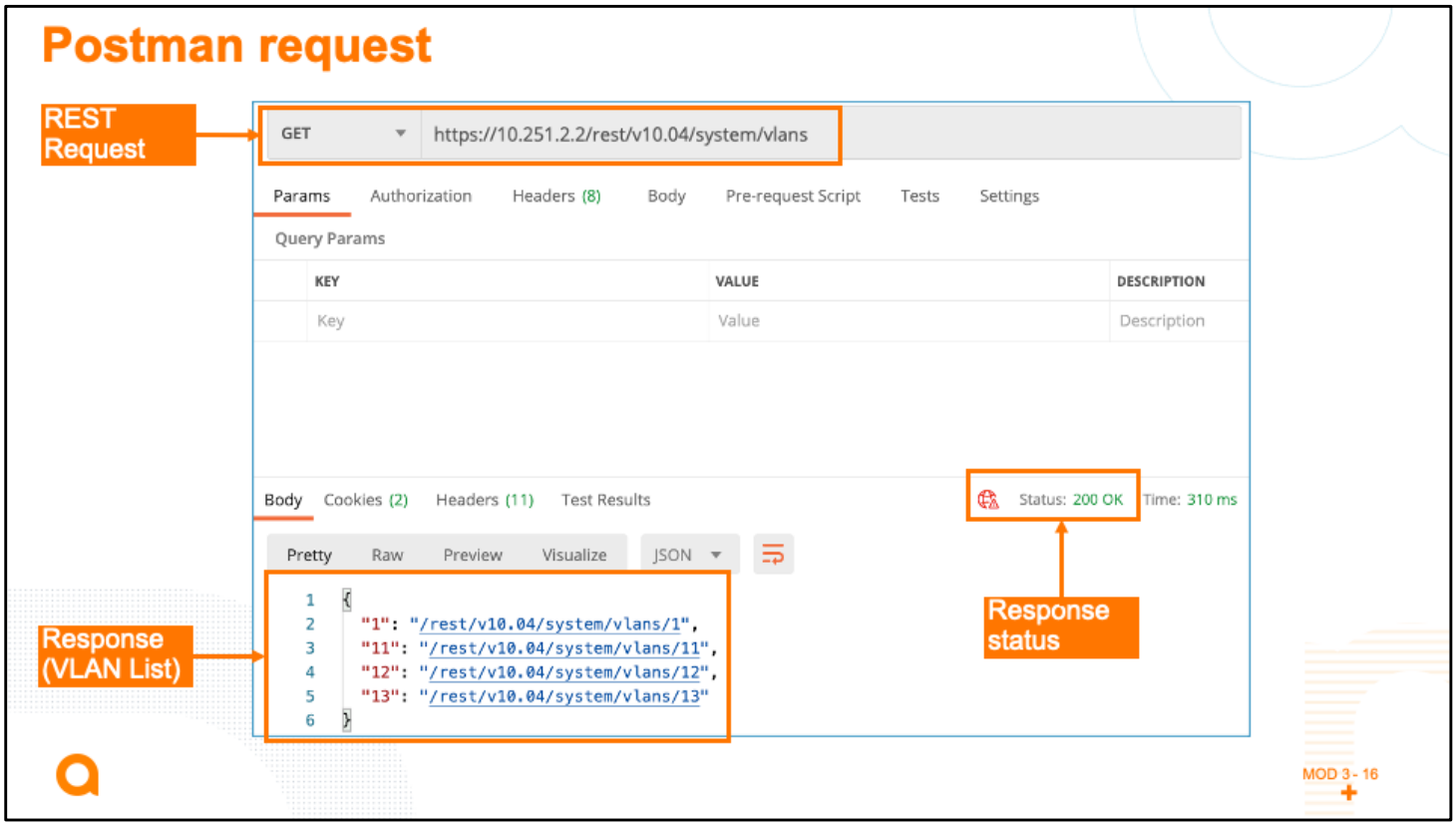
The method is POST

The URL is https://10.251.2.2/rest/v10.04/login?username=admin&password=Aruba123!

The response status is 200 OK

The cookie information includes the id and the user

Note. Postman allows you to skip the SSL certificate verification. Just navigate to File > Settings > General and turn off the SSL cert verification option.



To be consistent with the previous REST methods, the figure shows how to generate the REST request to list all available VLANs.

Let's analyze the information presented:

The method is GET

The URL is https://10.251.2.2/rest/v10.0.4/system/vlans

The response status is 200 OK

The list of VLANs include: 1,11,12 and 13.

Question #1

```
C:\WINDOWS\system32>curl -X GET --insecure "https://10.251.2.2/rest/v10.04/system/vlan" -b cx.jar

401 Authorization Required
```

The network administrator issued the cURL request presented in the output to obtain information about the VLANs in an AOS-CX switch but receives the 401 Authorization Required message, what could cause the problem?

a. The administrator needs to use a different REST API version (v1)

b. The VLAN resource does not exist

c. The administrator needs to login first and save a key in the cookie file `cx.jar` to authenticate future API calls

d. The administrator is using the “insecure” option in the cURL request and no authentication is possible



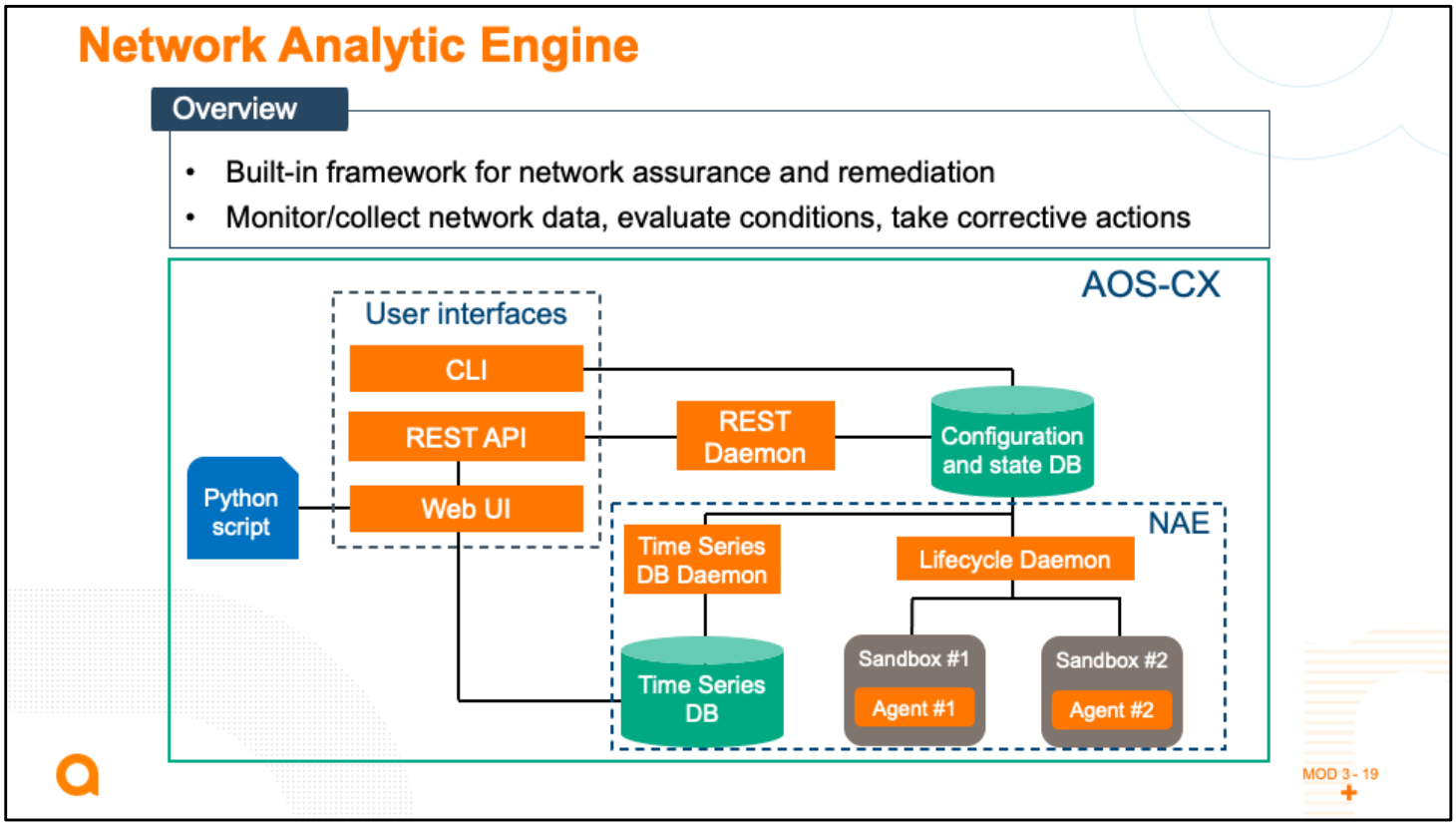
Correct answer:

C

Network Analytic Engine

Architecture, Scripts and Agents

aruba
a Hewlett Packard Enterprise company



NAE is a first-of-its-kind built-in framework for network assurance and remediation. Along with AOS-CX's full automation and deep visibility, this unique framework lets you monitor and collect network data, evaluate conditions, and take corrective actions using simple scripting agents.

Architecture

NAE offers three user interfaces: Command Line Interface, directly through the Web Server, and through the standard Web UI. While the CLI can make changes to the Configuration and State Database, from the NAE script perspective the CLI can only show agent and script status, such as installing, executing, or idle. The CLI is not the primary way to interact with the NAE.

You can use the REST API interface to create and update scripts via REST calls. You can interact with the REST Daemon, which handles the REST calls and interactions with the Configuration and State Database.

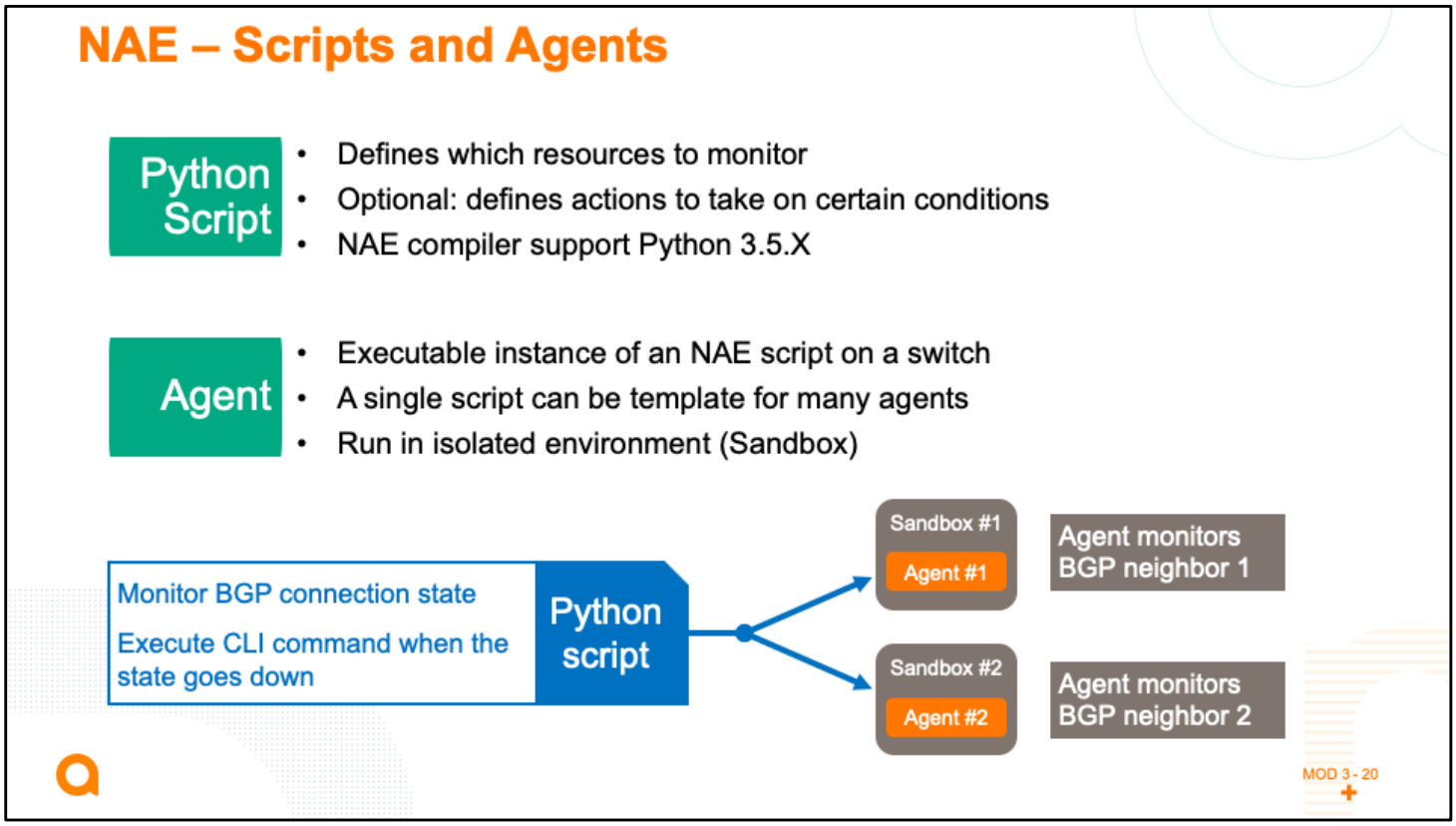
The primary workflow is to upload your script via the Web UI, which uploads it to the REST API interface. The Web Server feeds the script to the REST Daemon which makes the REST call and receives the payload. Then it saves it on the Configuration and State Database as a new entry in the database table NAE Scripts. The Lifecycle Daemon and the Time Series

Database Daemon are subscribed to the tables in the Configuration and State Database and react to changes in the database entries.

The Lifecycle Daemon is responsible for the script and agent lifecycle and for running the python code. The Lifecycle Daemon manages sandboxes loaded with the python runtime environment. A UNIX socket is the only means of communication in and out of the Sandbox - an isolated environment stripped down to very limited resources and features. The benefits of this design are:

- Agents coexist and are prevented from using an excessive amount of CPU resources.
- Agents can benefit from the high-availability features of AOS-CX. During a switch failover event, the daemon that handles the sandbox can recover its state information and continue operations as before.
- Agents are prevented from accessing sensitive information in the switch operating system.

When the NAE Agent table is updated with the new agent, the subscribed Time Series Database Daemon detects the changes from the table and sends that data to the Time Series Database. Time-series data about the resources monitored by agents are automatically collected and presented in graphs in the switch Web UI. The database makes the data seamlessly available to agents that use rules that evaluate network conditions over time.



An Aruba Network Analytics Engine uses a Python script to define which switch resources to monitor. Optionally, it defines actions to take when certain conditions are true. Examples of tasks a script can define include the following:

Monitor average CPU usage and send a system log message when it is greater than 70% for 5 minutes.

Monitor a BGP neighbor connection state and execute a CLI command when the connection state transitions from UP to DOWN.

An Aruba NAE agent is a specifically-configured executable instance of an NAE script on a switch. When the script is enabled, the agent performs script tasks. A single script can be the template for many different agents.

For example, a script that monitors the connection state of a BGP neighbor can define a parameter to specify which BGP neighbor to monitor. From that script, you can do the following:

Create an agent to monitor a specific BGP neighbor by specifying that BGP neighbor as the value of the parameter.

Create multiple agents—one for each BGP neighbor you want to monitor.

When a new BGP neighbor is added, you can create an agent to monitor that neighbor without having to change the script itself.

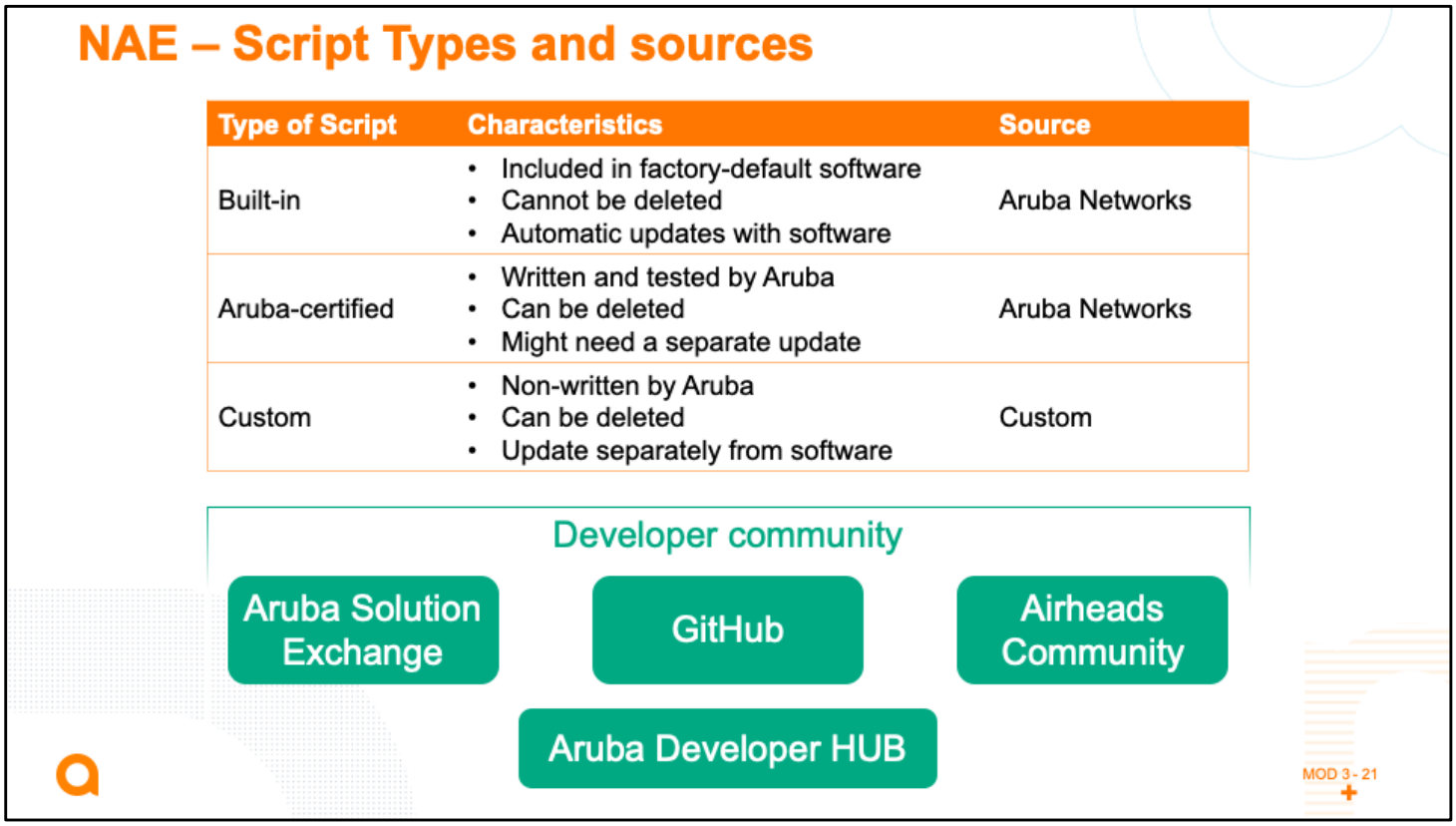
Network Analytics Engine scripts are written in Python 3.5.X. NAE provides a built-in Python interpreter to validate scripts and create agents from scripts.

Note. Python is the go-to language for network engineers because:

Python is high-level and human-readable.

Python is popular with an active development community.

There are many libraries (code written for you that you can use in your programs) available.



Aruba installs built-in scripts and agents before shipping switches from the factory. All scripts and agents include information about their origin. Built-in scripts and agents have an origin of "system" and are marked “System Created” in the Web UI.

Built-in scripts and agents cannot be deleted. You can enable, disable, and change the configuration of built-in agents.

The current software release includes a single built-in script and agent that monitors several system resources:

- Built-in script: system_resource_monitor
- Built-in agent: system_resource_monitor.default

Aruba-certified Network Analytics Engine scripts are written and tested by Aruba, available from the Aruba Solution Exchange. New scripts added to NAE are marked as a user-created agent, which can be deleted. When using these type of scripts you should be careful to verify first if a new version of the script is available in one of the Aruba developing sites which is compatible with the new AOS-CX release that you attempt to use.

Finally, you can create agents on your own with Python script. These agents are managed like the ones that you install from a community source. You can disable, remove and update them separately from the software. This devices will also be marked as a user-created agent.

There are several communities that facilitate the development, use, and sharing of scripts for the Aruba Network Analytics Engine:

Aruba Solution Exchange (ASE). The Aruba Solution Exchange (ASE) is the primary portal for Network Analytic Engine scripts. Aruba- certified public scripts are posted to ASE. Developers can create their own Network Analytics Engine solutions and post them for either private or public use. See the Aruba Solution Exchange at: https://ase.arubanetworks.com/. The AOS-CX web interface includes the link to directly access the ASE web site.

Note. It is important to clarify that the AOS-CX switch does not require access to Internet, the link simply instructs your browser to connect to the remote site.

GitHub. The NAE scripts repository on GitHub includes Aruba-certified Network Analytics Engine (NAE) scripts and examples organized by feature or protocol and then by supported hardware platform. Developers can customize or enhance these scripts for their own use. See the repository at: https://github.com/aruba/nae-scripts

Airheads Community. The Airheads community provides a place for members or participants to search for information, read and post about topics of interest, and learn from each other. Guests (unregistered visitors) can browse or search the community for information. Members (registered users) can post messages or comments, track discussions, and get email notifications on posting activity and other community actions. Within the Airheads community, there is a Developer Community group that is specific to APIs, programming, and automation. The Developer Community is the recommended place to post questions about the Aruba Network Analytics Engine (NAE). See the Developer community at: https://community.arubanetworks.com/t5/Developer-Community/bd-p/DeveloperCommunity See the Airheads community at http://community.arubanetworks.com/

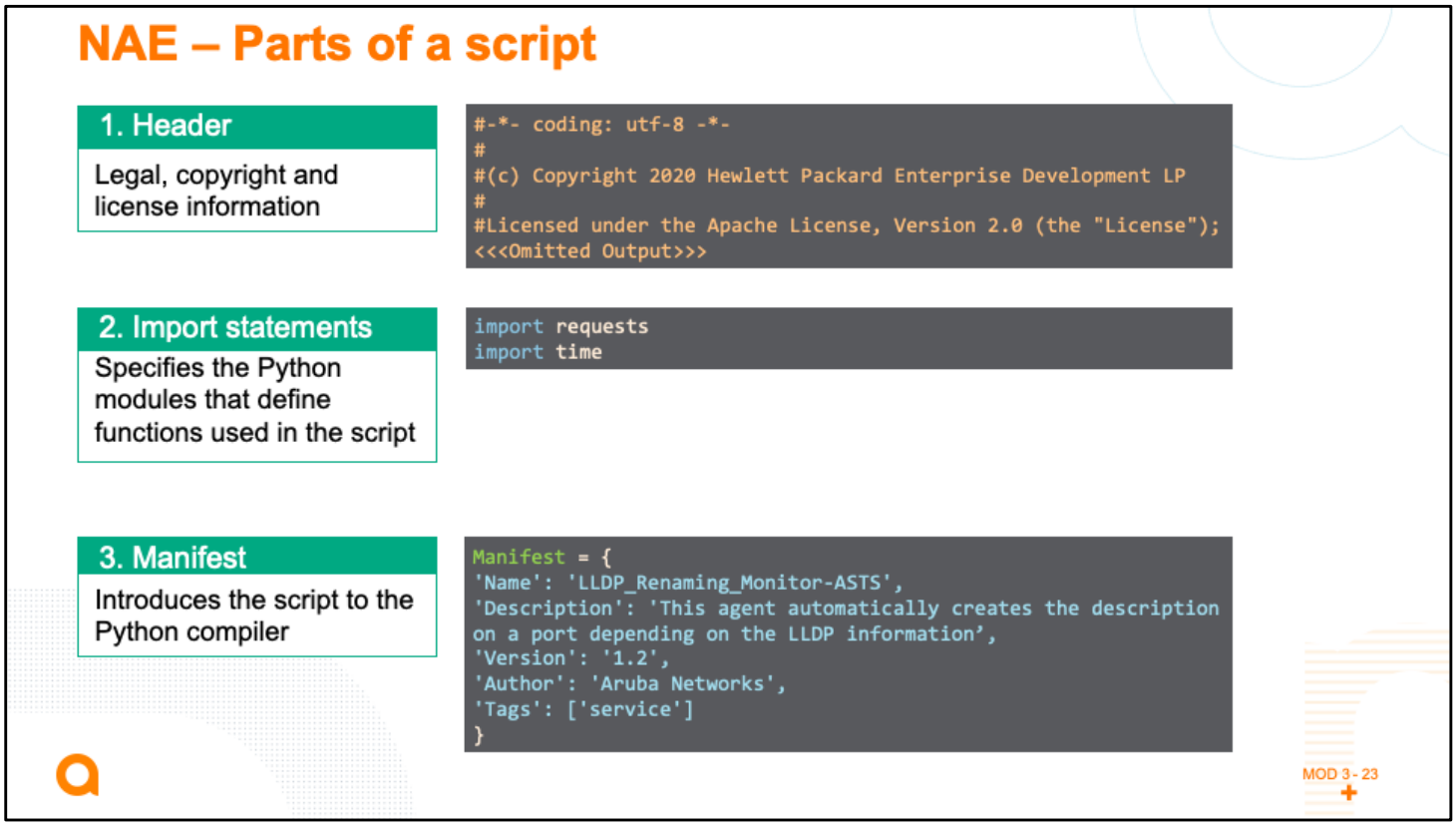
Aruba Developer Hub https://developer.arubanetworks.com/aruba-aoscx/docs/the-aruba-network-analytics-engine

NAE – supported maximums

Switch series	Maximum scripts	Maximum agents	Maximum monitors	Maximum script file size	Number of days of time-series data to store	Storage allocated to time-series data
8400	50	100	300	256KB	400	18GB
8320 and 8325	25	50	150	256KB	400	9GB
6300 and 6400	10	10	130	256KB	45	3.1GB
6200	10	10	130	256KB	15	1GB

MOD 3 - 22

The table shows maximum values related to the NAE framework that are supported in an AOS-CX switch. These values depend on the switch series. Additionally if the maximum amount of storage has been reached, there is a “garbage collector” that would start clearing out the older data to make space for newer data.



The figure shows the parts of a Python script used in NAE, as described below.

1. Header: Includes legal, copyright and license information. Example:

- #-*- coding: utf-8 -*-
  #
  #Copyright (c) 2017 Hewlett Packard Enterprise Development LP
  #
  #Licensed under the Apache License, Version 2.0 (the "License"); #you may not use this file except in compliance with the License. #You may obtain a copy of the License at
  #
- #http://www.apache.org/licenses/LICENSE-2.0
  #
  #Unless required by applicable law or agreed to in writing,
- #software distributed under the License is distributed on an
- #"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
- #KIND, either express or implied. See the License for the
- #specific language governing permissions and limitations #under the License.

2. Import statements: Specify the Python modules that define functions used elsewhere in the

script. If you are using several functions from the same module, consider importing the whole module. Example:

- import requests
- import time

3. Manifest: Introduces the script to the Python compiler. The manifest is a required part. The manifest includes mandatory elements and optional items:

Mandatory items:

Name. Specifies a unique name for the script. The name must be a text string that starts with a letter or number and can contain alphanumeric characters and the special characters – dot, hyphen and underscore – The maximum length including the version must not exceed 80 characters

Description. Describe what the script monitors. Must be a text string.

Version. Specifies version information about the script. The version is appended to the script name. The version must be a text string that starts with a letter or number and can contain alphanumeric characters and the special characters . (dot), - (hyphen), and _ (underscore). The maximum length of the name plus the version must not exceed 80 characters. Version must not be empty.

Author. Specifies the author of the script. Must be a text string.

Optional items:

API Version. Specifies the NAE script API version. This item is informational only. No validation or action is performed for this item.

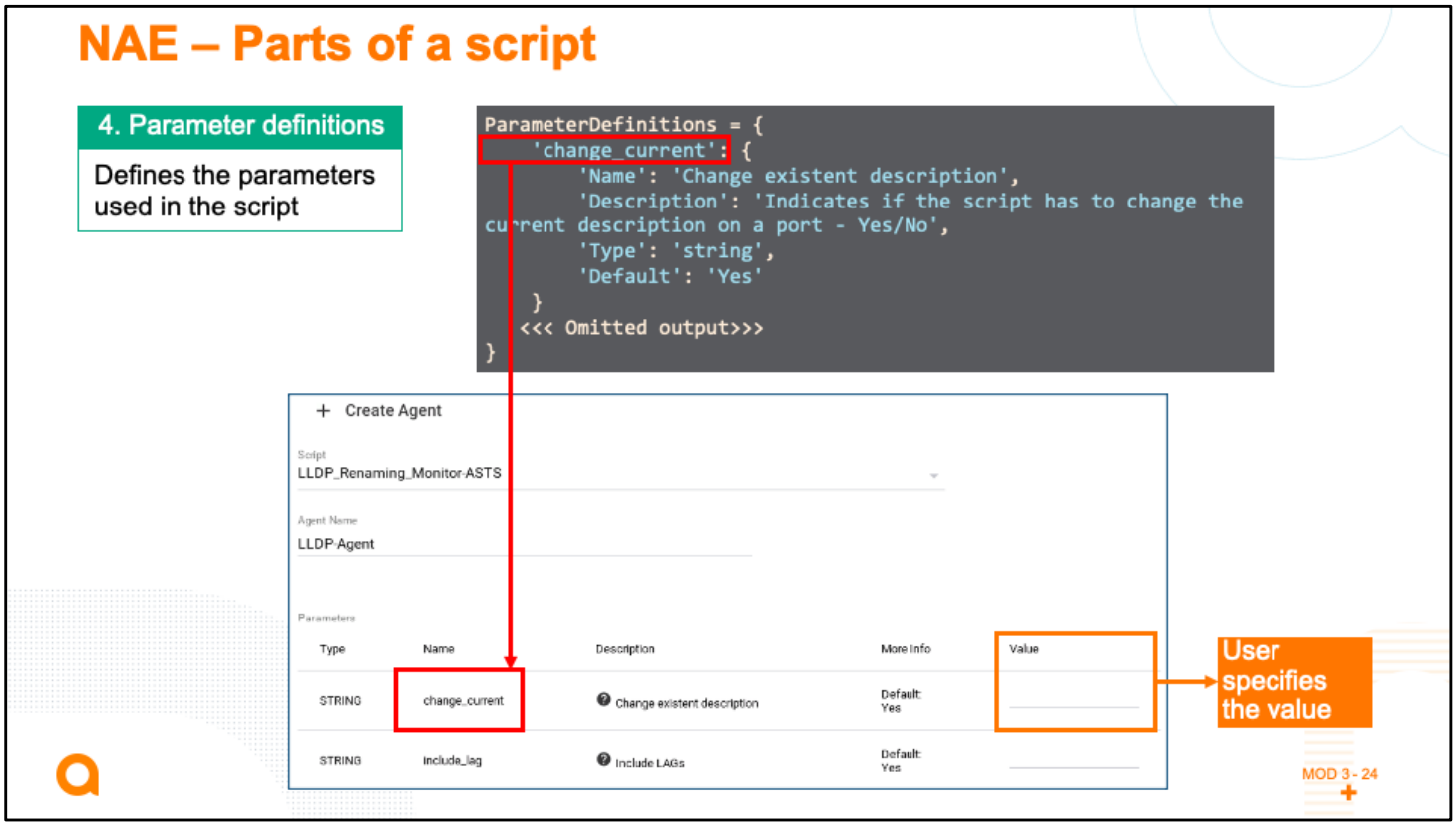
Target Platform. Specifies the switch platform on which the script is designed to

be executed. If this item is not specified, the script is considered to be platform independent. This item is informational only. No validation or action is performed for this item.

Target Software Version. Specifies the switch firmware version or versions on which the script is designed to be executed. If TargetPlatform is not specified, the script is expected to work with any software version. This item is informational only. No validation or action is performed for this item.

Tags. Specifies one or more tags to provide information about the script. The following tags are used by NetEdit to categorize and group the script and related alerts, and to automatically incorporate new NAE scripts into the health ranking for each of the following categories:

- application
- segmentation
- service
- routing
- bridging
- device
- other



4. Parameter definitions. Defines the parameters that are used in the script. When a user creates an agent, the user can specify values for the parameters. The user must specify values for parameters that the script identifies as required parameters.

This parameter definition is a Python dictionary that contains one or more parameters. At the time the agent is created, the default value is assigned to the parameter unless the user specifies a different value.

Each parameter definition could contain the following:

Parameter declaration

Name

Type: which specifies the datatype of parameter (integer or string)

Default: The script can contain a default default parameter which it will be used as part of a monitor or condition.

Encrypted: Specifies whether the user-supplied text is to be encrypted when the text is stored.

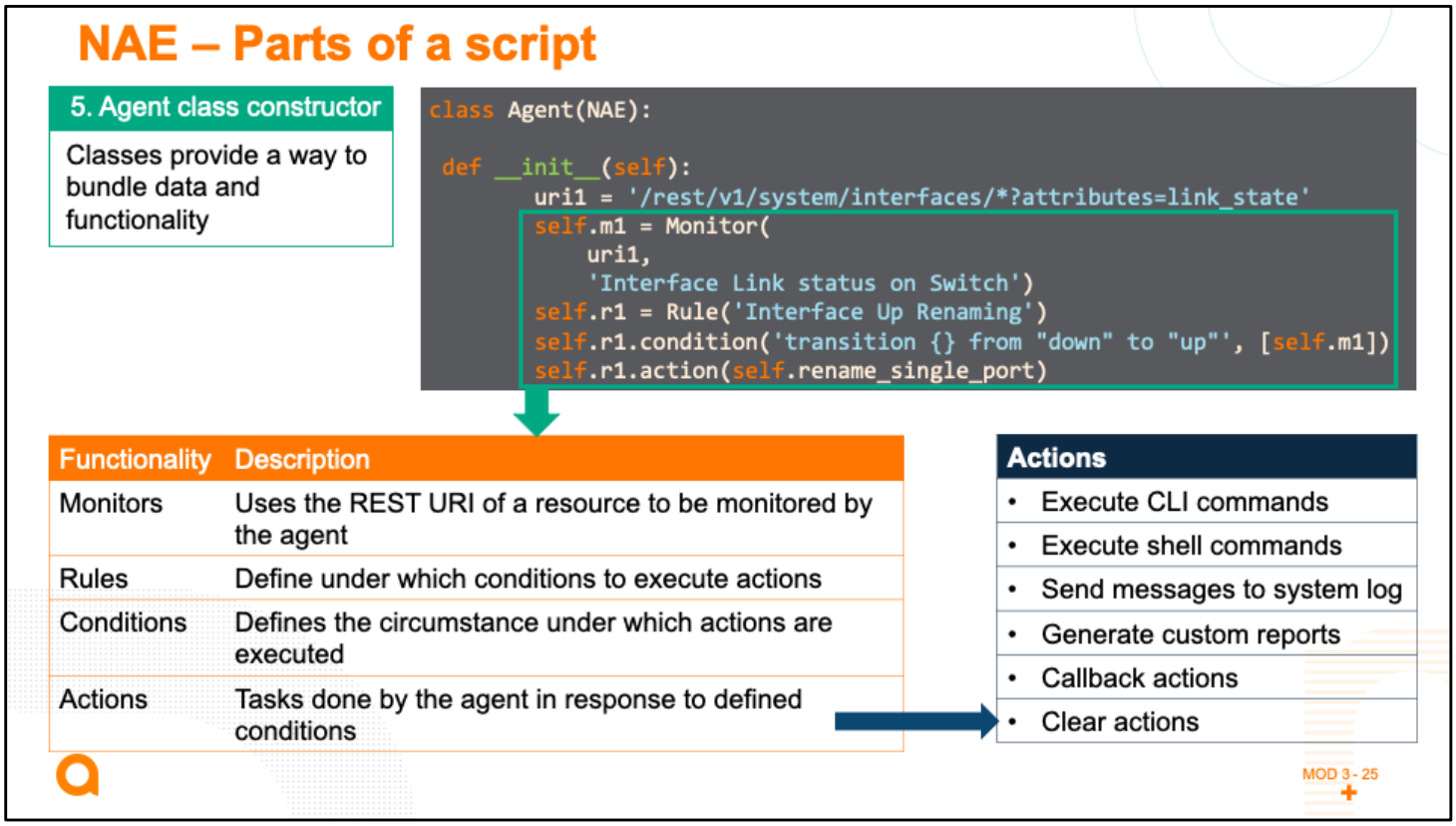
Required: Use this attribute to force the user to provide a value for this parameter.

In the example, the script defines the parameter: change_current

This parameter will change the existing description and the default value is Yes. In this case the user is not forced to enter a value for this parameter.

The slide shows at the bottom an image when an Agent is created. As expected, the parameter section asks the user about the Value for the parameters defined.

Note. You can find more information about Python dictionaries in this URL: https://docs.python.org/3/tutorial/datastructures.html



5. Agent class constructor: Python classes provide a way to bundle data and functionality together. This section of the script can contain the following:

Monitors: Uses the REST URI to define the switch resource to be monitored by the agent.

Rules: Define required conditions to execute actions

Conditions: The condition of a rule defines the circumstance under which actions are executed.

Actions: Actions are the tasks done by the agent in response to defined conditions. Actions are executed depending on the truth value of the condition in a rule. Actions might include things like setting the agent status, executing certain CLI commands, or generating system log messages.

Type of actions include:

Predefined action. Predefined actions are actions that are built into the Aruba Network Analytics Engine framework. These functions enable the agents to do the following:

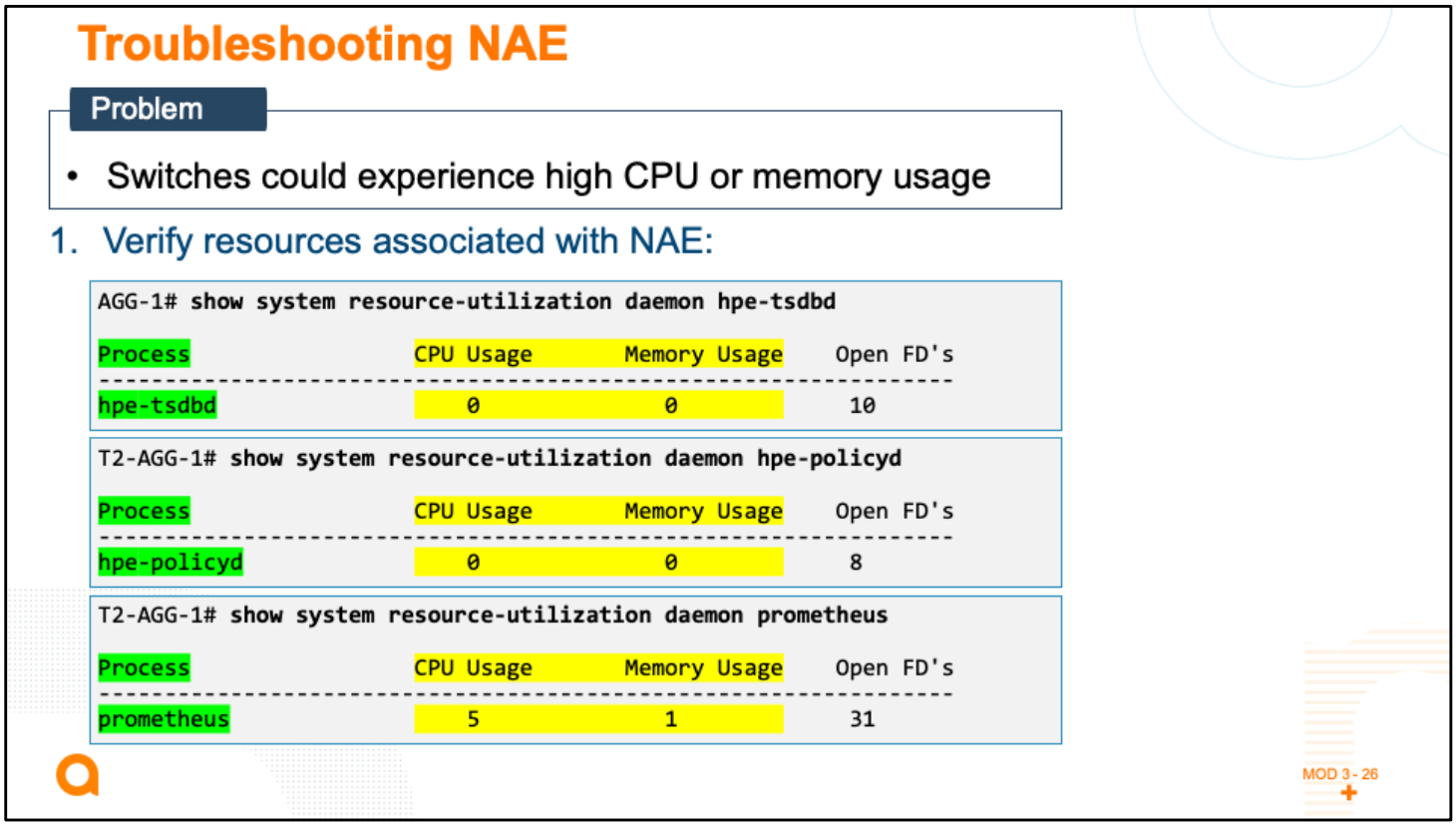
- Execute CLI commands in the AOS-CX network OS
- Execute shell commands in the underlying operating system of the switch.
- Send messages to the system log.
- Generate custom reports that communicate information collected by the agent.

Callback actions. Callback actions are callback functions defined in the script to perform additional processing.

Clear actions. Clear actions are predefined actions or callback actions that are executed only when the clear condition of a rule is active and transitions from false to true.

Note. A callback, also known as a "call-after" function, is any executable code that is passed as an argument to another code; that other code is expected to call back (execute) the argument at a given time.

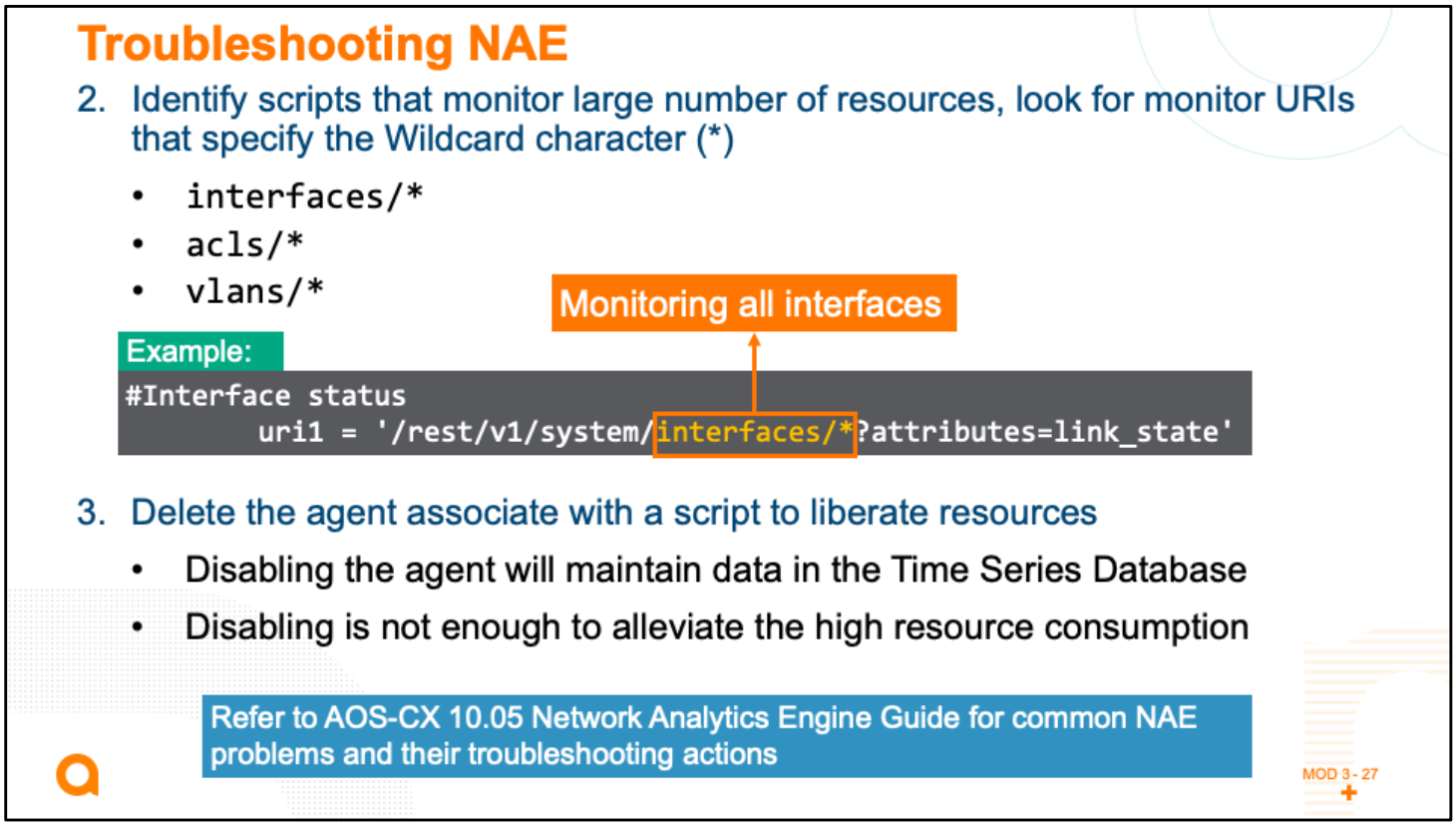
Take a moment to analyze the example shown in the figure.



Suppose that a switch with NAE agents installed experiences high CPU usage, high memory usage, or both. Overall performance is affected. The cause could be that NAE is attempting to monitor too many switch resources. For example, a script uses wildcard characters in a URI to monitor all interfaces, ACLs, or VLANs, and the switch has hundreds or thousands of those items.

Verify resources associated with NAE

- show system resource-utilization daemon hpe-tsdbd
- show system resource-utilization daemon hpe-policyd
- show system resource-utilization daemon Prometheus



Continue with the previous problem, another solution that can help to alleviate the CPU consumption is to identify installed scripts that include wildcard character (*) in the URIs of monitors. Look for monitor URIs that specify the wildcard character for resources that your switch has in large numbers. For example:

- interfaces/*
- acls/*
- vlans/*

After identifying a script that is consuming CPU resources, you decide to replace it with a new one. You must delete the agents associated with the scripts you identified. Disabling the agent is not sufficient because the switch does not delete time series data when an agent is disabled.

For more information about common problems with NAE, is recommended to consult the AOS-CX 10.05 Network Analytics Engine Guide.

Question #2

A network administrator is concerned about the resources consumed by a Python script in the Network Analytic Engine tool, what entity in the AOS-CX architecture creates an isolated environment with limited resources for the script to run?

a. REST Daemon

b. Lifecycle Deamon

c. Agent

d. Sandbox

a

Knowledge Check

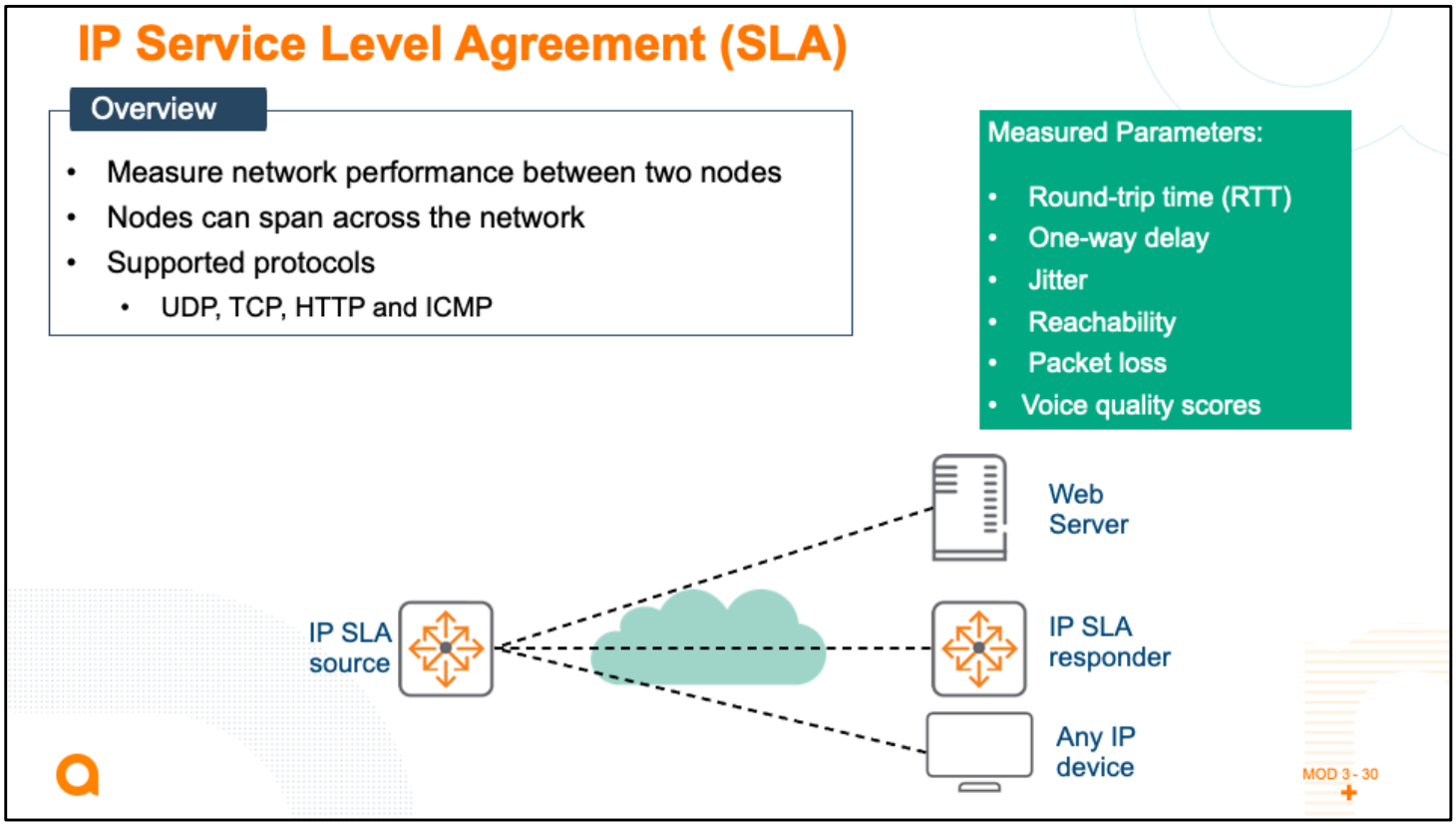
Correct answer:

D

Other monitoring

IP SLA, SNMP and NetEdit

aruba
a Hewlett Packard Enterprise company



The IP Service Level Agreement (IP-SLA) feature measures network performance between two nodes in a network. It measures things like round-trip time (RTT), one-way delay, jitter, reachability, packet loss, and voice quality scores. These two nodes can span across access, distribution or core layers in a building or campus, and across WAN links between cores, or from core to Data Center switches. Thus, you can measure the SLA for different protocols or applications such as UDP echo, UDP jitter (for voice and video), TCP connect, HTTP, and ICMP echo.

From the AOS-CX 10.05 Monitoring Guide, we have the following IP-SLA guideline:

AOS-CX supports only forever tests. On-demand tests are not supported.

NAE supports only syslog.

NAE agents must be triggered for each IP-SLA test on every switch.

If multiple IP addresses are received for a DNS query, DNS works with the first resolved IP.

When the DNS server IP is not configured, the first DNS server in resolve.conf is used.

The source interface/IP option is not applicable for SLAs configured on 'mgmt' VRF, as it has only one interface.

A system time change because of NTP or a manual change causes an incorrect calculation.

Source IP and source port combination must be unique across SLA sessions in a same switch.

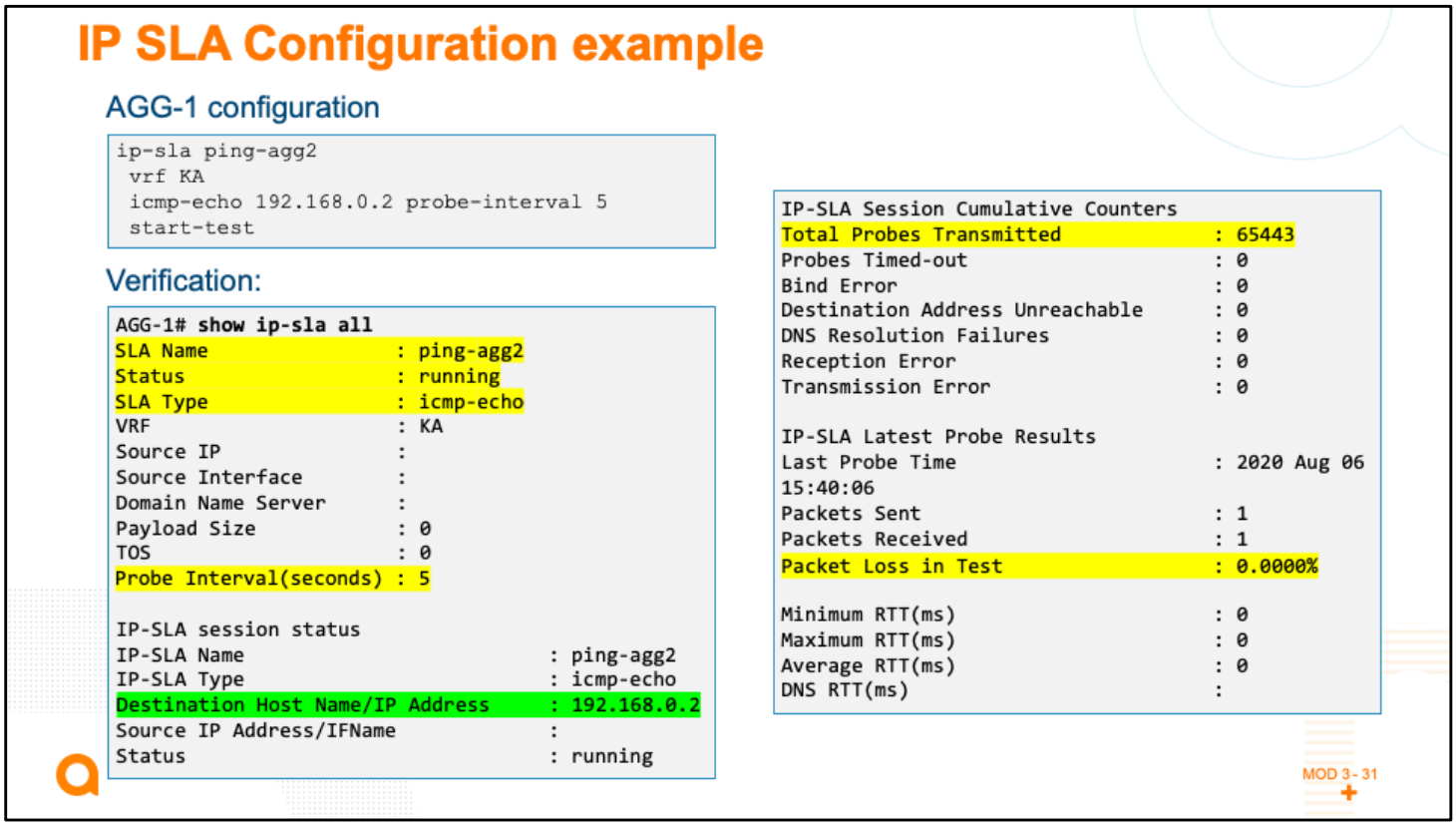
Do not use the same source port across the source and responder sessions in a switch.

NTP synchronization is a must for SLA types involving one-way delay such as UDP jitter VoIP.

It is mandatory to set default CoPP to the max value when UDP jitter SLA is enabled otherwise 100% packet loss can be seen and UDP-Jitter sla probe will result in failure.

Deviations with respect to PVOS results: The packet losses due to internal switch-related issues like interface shutdown or interface flaps will not be considered as 'Probes Timed-out error', as the IP-SLA solution is to measure network performance and anomalies. Rather, this kind of packet loss will be counted in internal counters like 'Destination address unreachable’.

IP SLA is fully supported on 8400, 8320, 8325, 6400 and 6300 switch platforms. The 6200-switch series only supports the IP SLA responder role.



When you create an IP SLA you must use the ip-sla command, the following lines present the available options in 10.05:

ip-sla <IP-SLA-NAME>

http	Configure HTTP as the IP SLA test mechanism.
icmp-echo	Configures ICMP echo as the IP SLA test mechanism
start-test	Starts the IP-SLA probe.
stop-test	Stops the IP-SLA probe.
tcp-connect	Enables responder for tcp-connect probes.
udp-echo	Enables responder for udp-echo probes.
udp-jitter-voip	Enables responder for udp-jitter-voip probes.
vrf	VRF Configuration(Default:default)

This example shows how to generate an IP SLA to a specific destination using ICMP.

The configuration is:

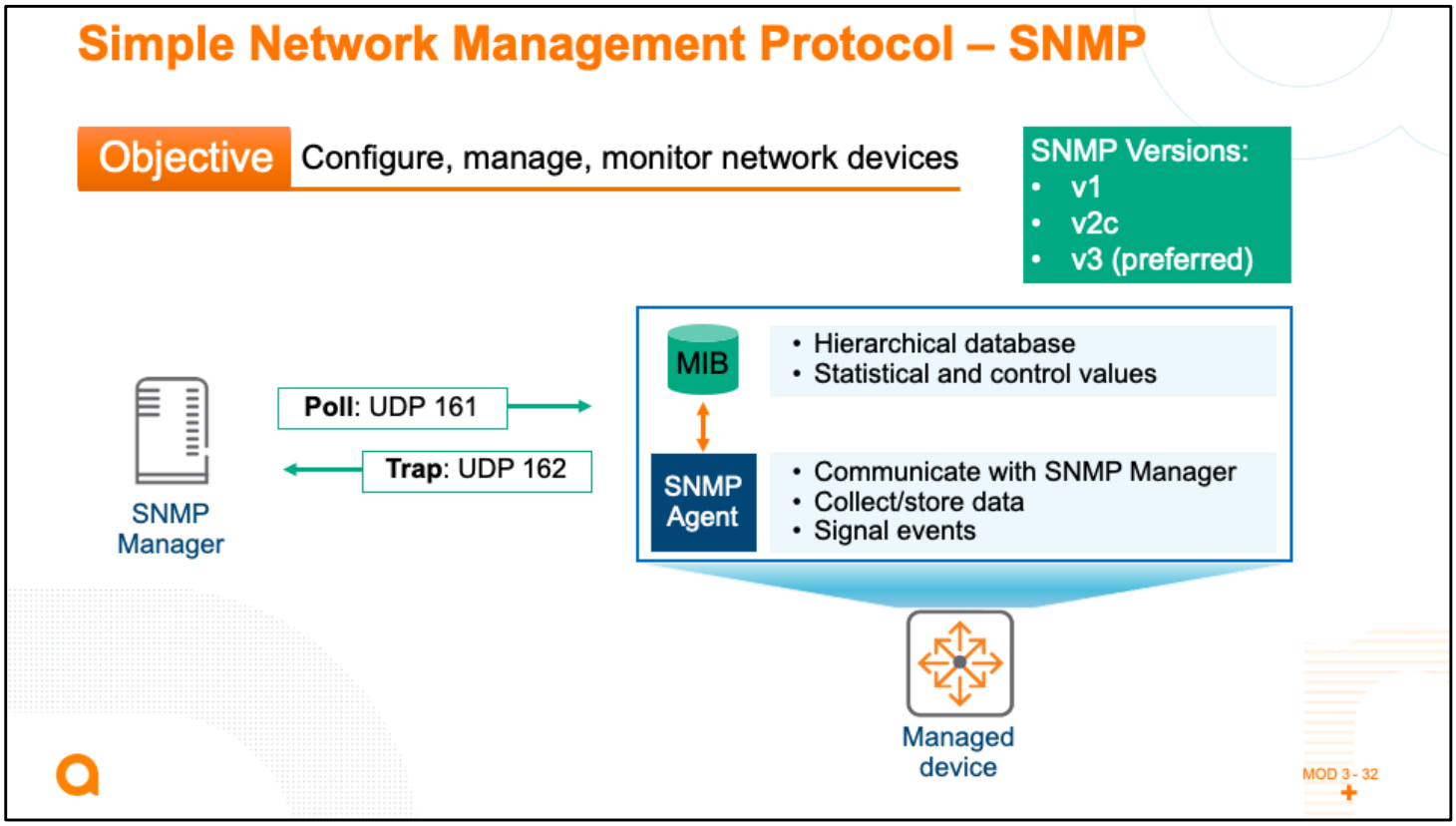
ip-sla ping-agg2

```
vrf KA
icmp-echo 192.168.0.2 probe-interval 5
start-test
```

Verify SLA status with the show ip-sla all command.

Status and Description:

- Running: SLA is fully operational
- Bind Error: Another service is using the same source port
- Interface Down: Interface status is not up
- Dns Resolution Error: Failed to resolve destination hostname
- No Route: No available route to the responder
- Internal Error: Unexpected error prevents SLA session
- Disabled: SLA is disabled
- Configuration Incomplete: Configuration is not complete to enable the SLA



SNMP is an application-layer protocol defined in RFC 1157 for exchanging management information between network devices. This is a popular method to configure, manage, and monitor network elements from any vendor. The figure shows the SNMP architecture, as described below:

SNMP Manager. Is typically a dedicated server that communicates with SNMP agent. It often has a GUI interface, so you can view and change parameters for many network device from a single interface. The manager uses several message types to gather and modify network device parameters, Some of its key functions include:

- Queries agents
- Gets responses from agents
- Sets variables in agents
- Acknowledges asynchronous events from agents

Managed devices. The routers, switches, firewalls, wireless controllers, access points and any other network devices that should be centrally monitored, managed, and configured. Internally, these managed devices have two main SNMP components.

The SNMP Agent collects information and communicates with the SNMP Manager about device configuration and status. It stores and retrieves this information in a Management Information Base (MIB).

The MIB is a hierarchical database that stores device status, statistics, and configuration information. This is the information that is exchanged between the SNMP Manager and the managed device.

Communication and ports

Usually, SNMP Manager to Agent communication is a “pull” mechanism. When this device requires to get information from the managed device it requests the information using simple commands such as GET, GET NEXT, GET BULK. Changes in the device are done using the SET command. This communication uses UDP 161.

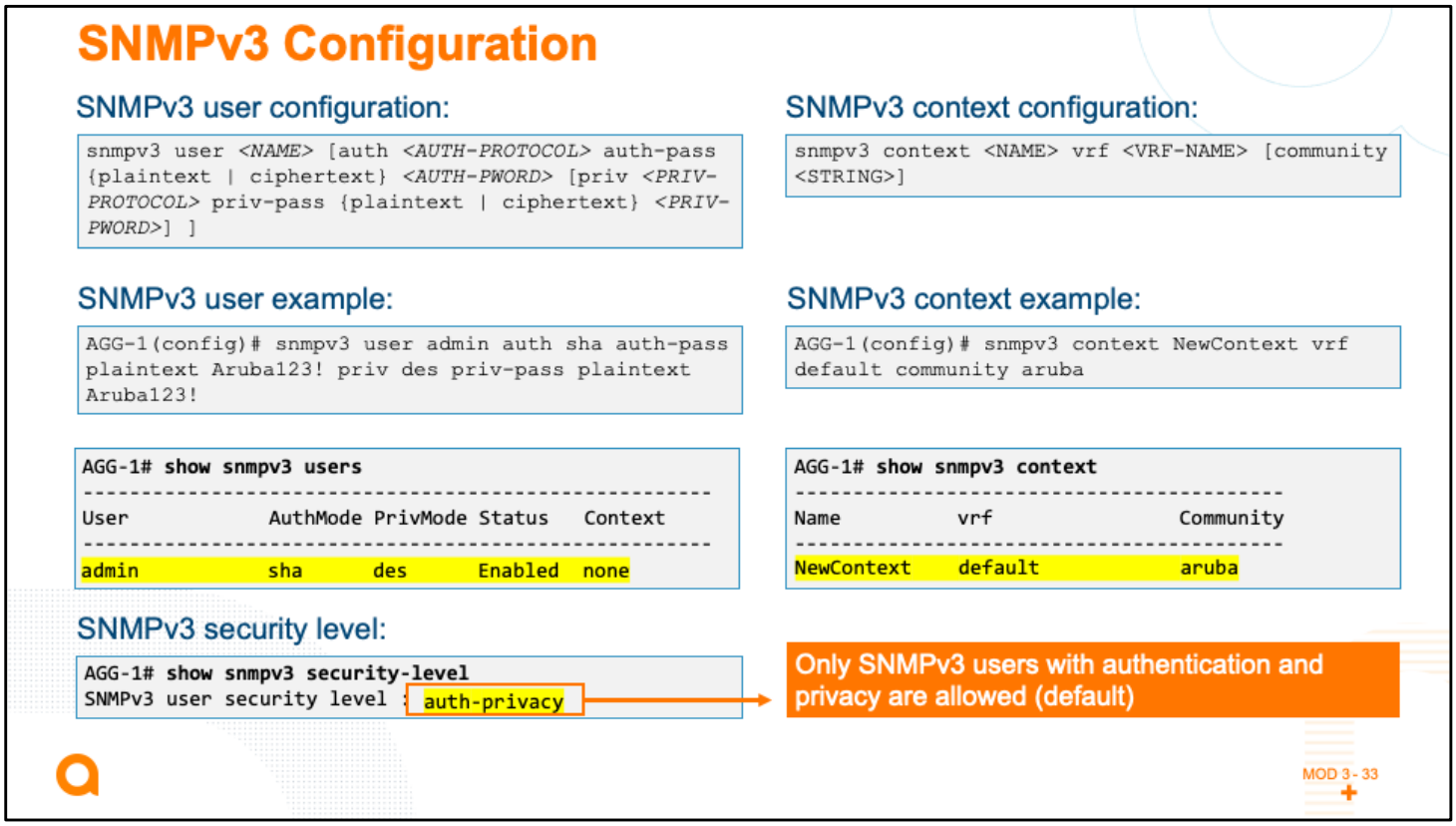
The exception to the “pull” mechanism is when the managed device has important information that cannot wait for the manager’s period polling. “Newsflash!! My CPU utilization is dangerously high”. In cases like this, the managed device “pushes” information to the manager without being asked, using an SNMP Trap. SNMP Traps use UDP 162.

SNMP Versions:

SNMP version 1. First version of the SNMP protocol and uses a community-based security mechanism. The “community string” is like a simple pre-shared passcode. If the Managed device has a community string of public, then any SNMP manager that can reach the device can access the agent’s MIB, as long as community string = public is used.

SNMP version 2c. This is the revised protocol, which includes enhancements of SNMPv1 in the areas of protocol packet types, transport mapping, MIB structure elements. This version also uses community-based security mechanism.

SNMP version 3. Provides a higher level of security, including message integrity, authentication, and encryption. This is the preferred version to be deployed in a real environment.



To create a user for SNMP version 3, use the snmpv3 user command. The parameters are:

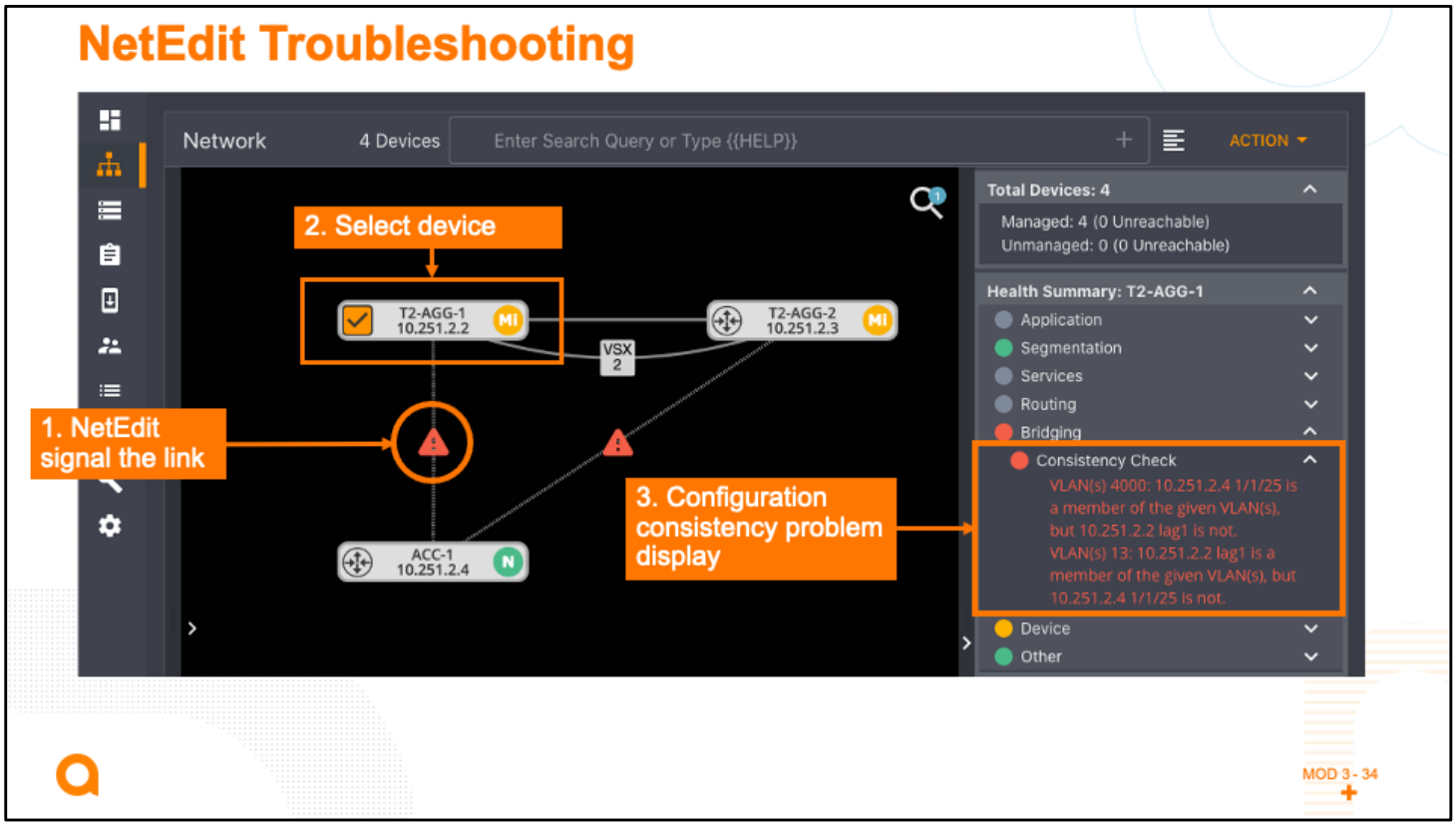
auth <AUTH-PROTOCOL> Specifies the authentication protocol used to validate user logins. Available options are: md5 or sha.

auth-pass {plaintext | ciphertext} <AUTH-PWORD> Specifies the SNMPv3 user password.

priv <PRIV-PROTOCOL> Specifies the SNMPv3 security protocol (encryption method). Available options are: aes or des.

priv-pass {plaintext | ciphertext} <PRIV-PWORD> Specifies the SNMPv3 user privacy passphrase.

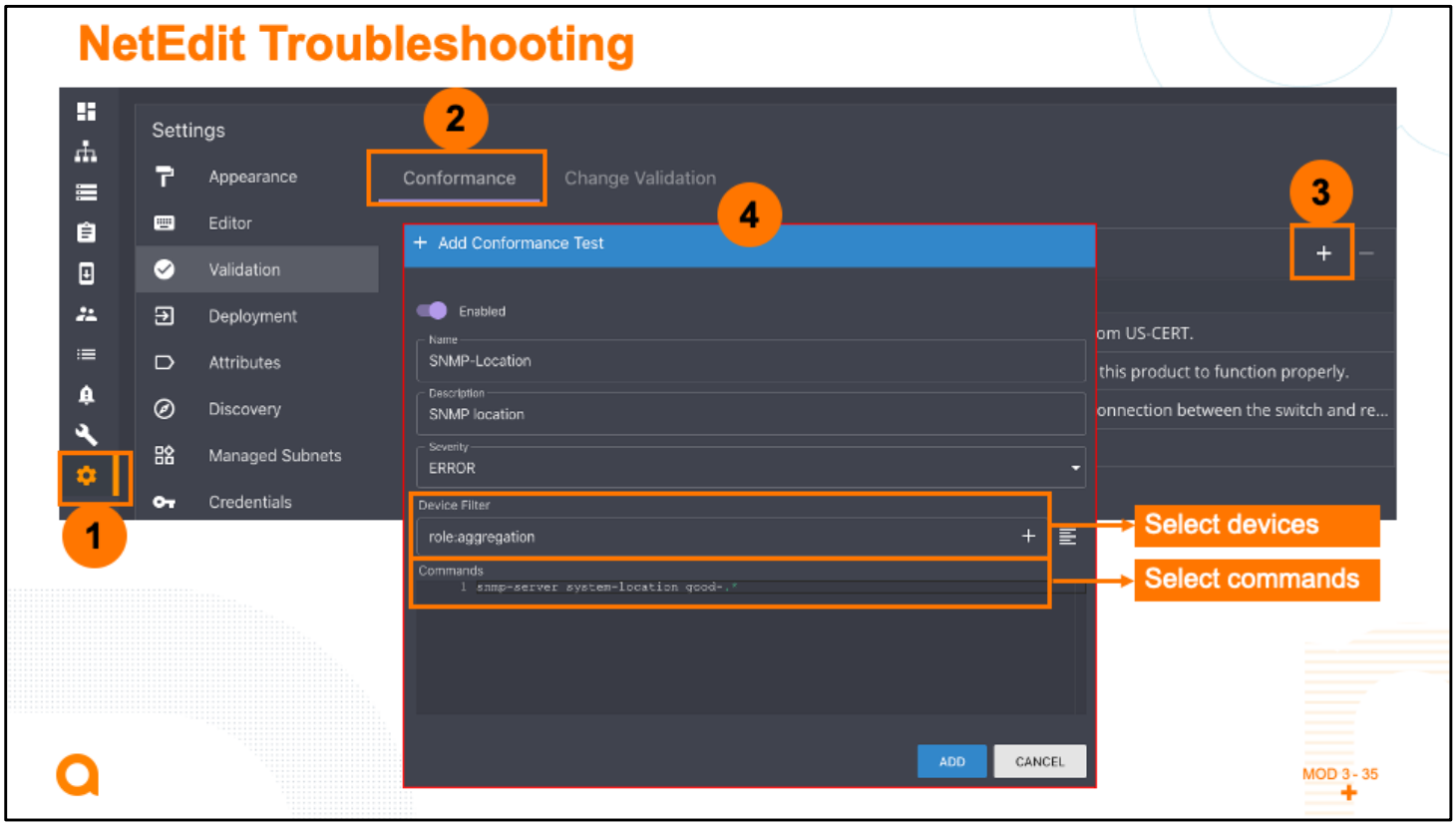
Only SNMPv3 users with both authentication and privacy are allowed. This is the highest level of SNMPv3 security and is the default option.



Aruba NetEdit offers another resource for troubleshooting.

The NetEdit topology view shows all devices currently inventoried by NetEdit . This view has three distinct panels: the properties panel, the selection panel, and the topology panel itself. The properties panel is a context-sensitive panel that shows topology information and the current topology selection. The selection panel provides mechanisms to enable or disable additional data that is layered over the topology view.

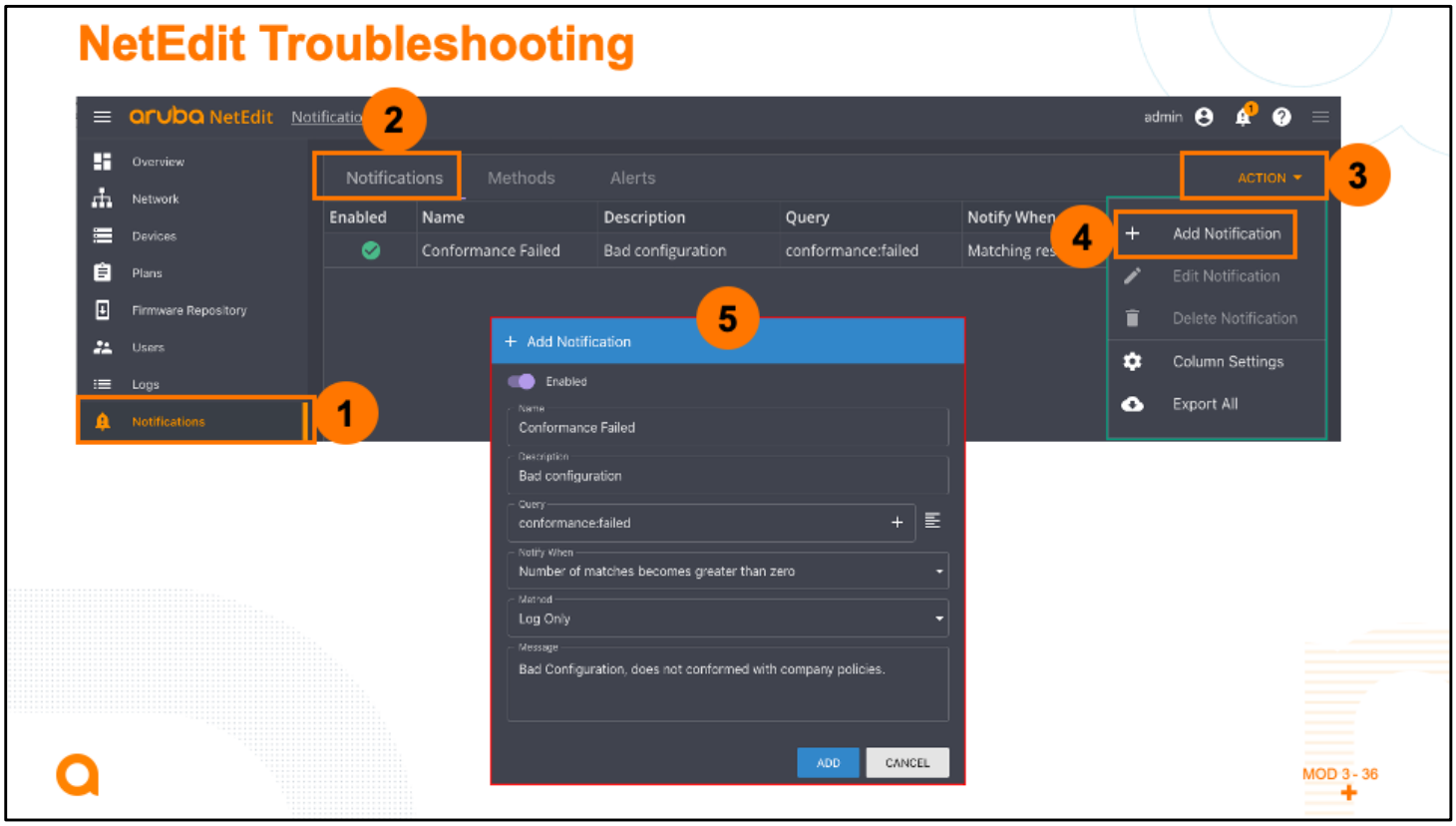
In this example, a VLAN consistency problem between T2-AGG-1 and ACC-1 switches has been signaled by NetEdit (red triangle). To get more information you must select one of the devices involved in the problem, for example T2-AGG-1 switch. When you click on the AOS-CX switch, a description of the problem will automatically appear in the selection panel. In this case it displays the VLAN mismatch in the Trunk link.



Aruba NetEdit can create conformance tests that permit the administrator to detect problems in a proactive manner. A conformance test is composed by a list of commands and arguments; the current configuration of the AOS-CX switch is continuously verified by NetEdit.

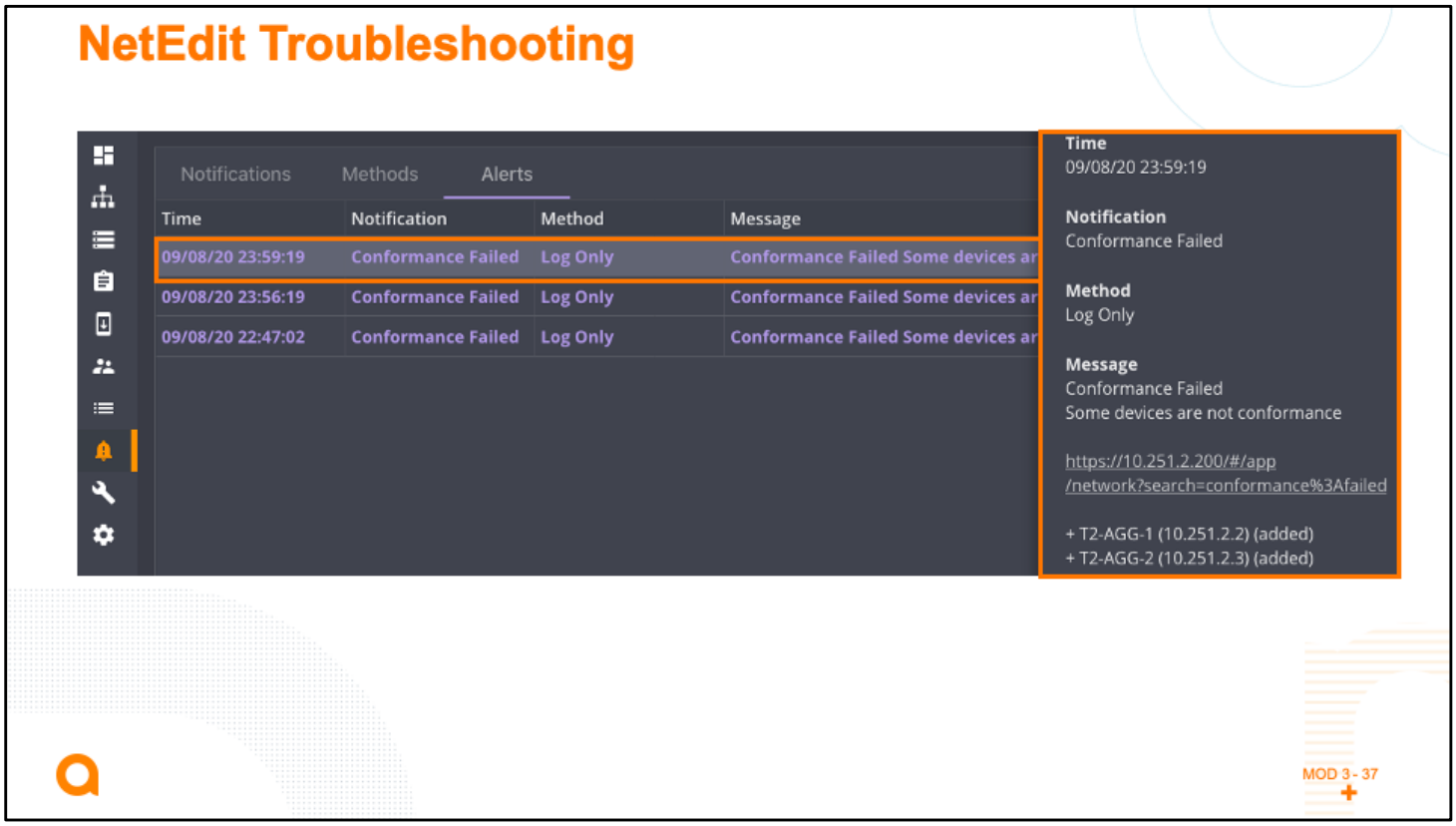
When the administrator uses NetEdit to edit the switch configuration, NetEdit informs the administrator if the command conforms the policy or not. NetEdit is capable to generate a notification when the configuration conformance from a specific AOS-CX switch is not successful.

Note. To reduce the list of commands NetEdit support Regular Expressions (REGEX).



NetEdit also permits to create Alerts which can be trigger on specific circumstances. To do this you must navigate to Notifications > Notifications tab then click on ACTION menu and select Add Notification.

In this example we present a simple notification that is triggered whenever an administrator enters a command that is not consider to be under the Company's conformance policy. Pay special attention on the query that has been used (conformance:failed)



The notification page displays the information about the generated alerts.

In this example you can see that a new alert has been triggered because a conformance failure.

Question #3

AOS-CX supports only SLA forever tests and on-demand tests are not supported.

a. True

b. False



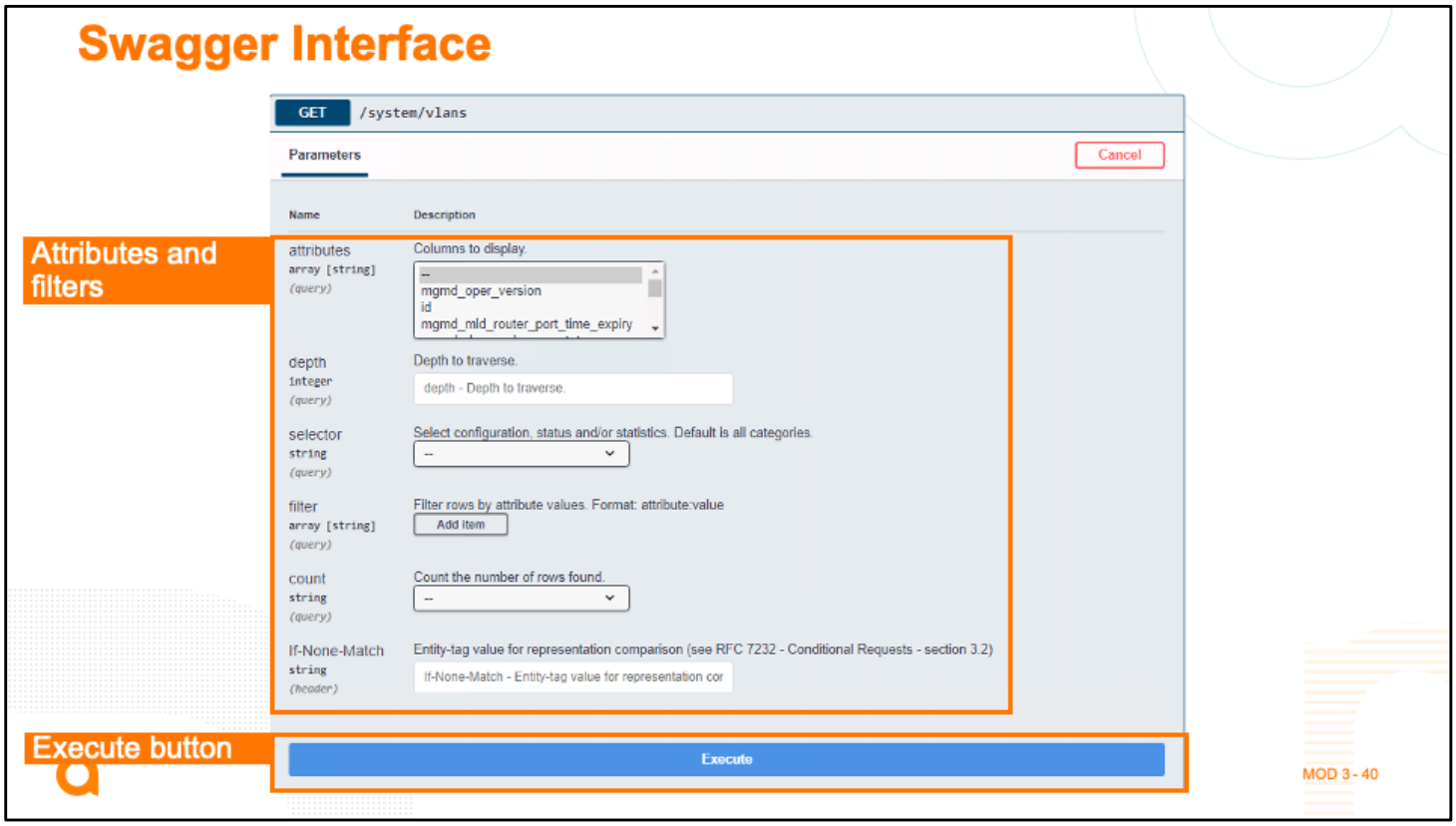
Correct answer:

A

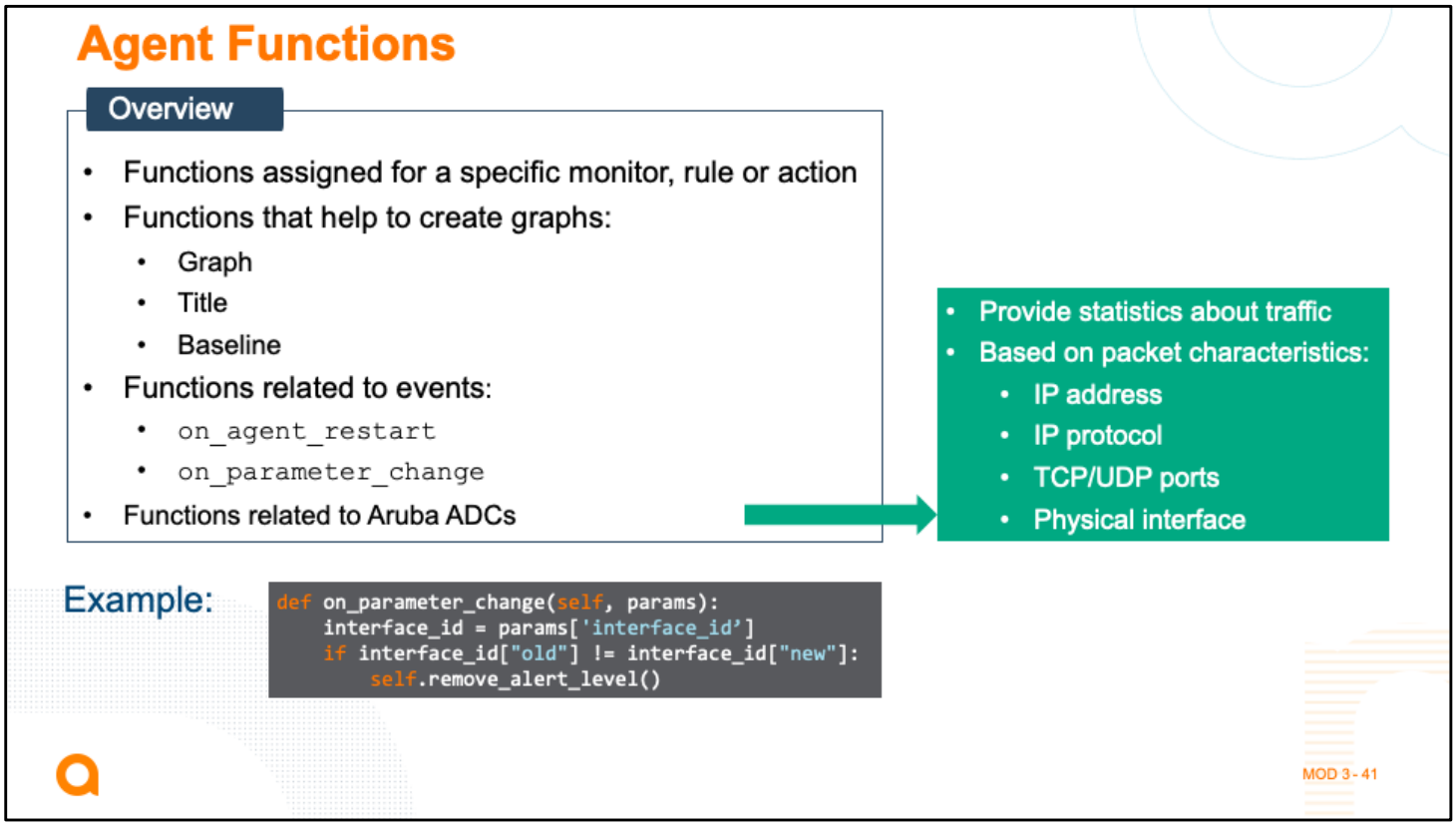
aruba
a Hewlett Packard Enterprise company

Additional Content

Module 03



A new section appears where you can specify attributes and filters to the REST request. When you finish you must click the Execute button that appears below.



The agent constructor can contain functions that are restricted to a specific monitor, rule or action.

For example, for a resource already monitored by the “monitors” function, you can specify the creation of a graph so the data related to the resource can be graphically viewed from the Web UI.

Agent functions available to create graphs are:

Graph. Enables you to select a monitor or group of monitors to display together in a graph on the Agent Details page in the Web UI

Title. Creates a text string to be used as a custom title for a graph or a predefined action

Baseline. Provides a way to specify thresholds that are appropriate to the network conditions on the switch.

Agent functions related to events are:

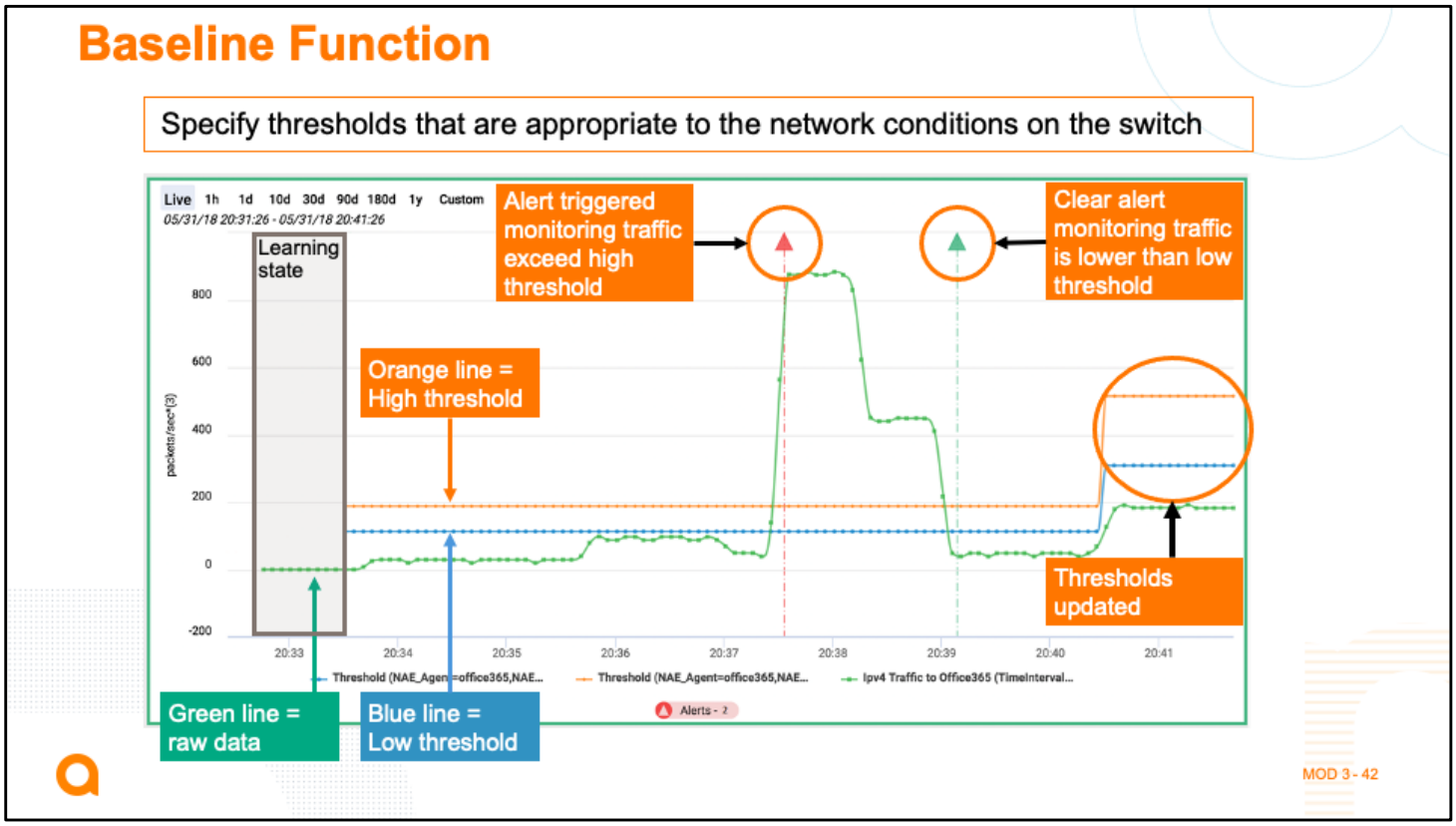
on_agent_restart. Performs specified actions when the NAE agent is restarted.

on_parameter_change. Performs the specified actions when a user changes the value of an agent parameter.

Agent functions related to Aruba Analytics Data Collections (ADCs)

ADCs are script-defined rules (comparable to ACLs) to provide statistics about different types of network traffic passing through a switch. Network traffic statistics can be gathered based on many different frame or packet characteristics, including: Source and destination IP address, IP protocol, Layer 4 application ports and physical switch ports.

For more information about how to use these functions, please refer to the AOS-CX 10.05 Network Analytic Engine Guide.



The Baseline function lets you specify thresholds that are appropriate to the network conditions on the switch. When an agent is created and enabled, it spends a specified amount of time “learning” about the data it is monitoring before it sets thresholds. These thresholds are calculated based on what it learned. The amount of time for most Aruba-certified scripts is set to 1 day.

In addition, these thresholds are dynamic. The agent continues to learn about the data it monitors, and the Baseline function adjusts the thresholds accordingly. For example, if the lower-traffic switch starts to get consistently higher incoming traffic rates, the Baseline function adjust the thresholds to reflect the newly learned rates.

Default thresholds can be used to determine when to set and clear alerts while the agent is in a learning state. Otherwise the agent does not generate alerts while it learns about the data. The methods to determine the baseline from which to calculate the thresholds depend on the algorithm selected in the Baseline function. Currently the only available algorithm is the MaxAlgorithm.

You can find more information about MaxAlgorithm functionality in the AOS-CX 10.05 Network Analytic Engine Guide, Chapter 10 Scripts, Baseline workflow and consideration section.

As an example of the functionality the slide shows a time series graph for a monitor that

includes baselines:

The initial learning time is one minute.

There are no default thresholds.

In this graph:

• The green line is the raw data.

The orange line is the high threshold as calculated by the baseline.

The blue line is the low threshold as calculated by the baseline.

The events in the timeline are as follows:

At 20:32:30, an agent is created and enabled. The baseline enters the learning state. Because the script did not specify default thresholds, there are no thresholds defined. In the graph, just the green line for the raw data is displayed.

At 20:33:30, the baseline exits the learning state and enters the active state:

- The high threshold and the low threshold calculations are completed.
- The graph begins the display of the orange line for the high threshold and the blue line for the low threshold.
- The agent will generate an alert when the monitored traffic rate (in packets per second) exceeds the high threshold.
- The agent will clear the alert when the monitored traffic rate (in packets per second) drops below the low threshold.

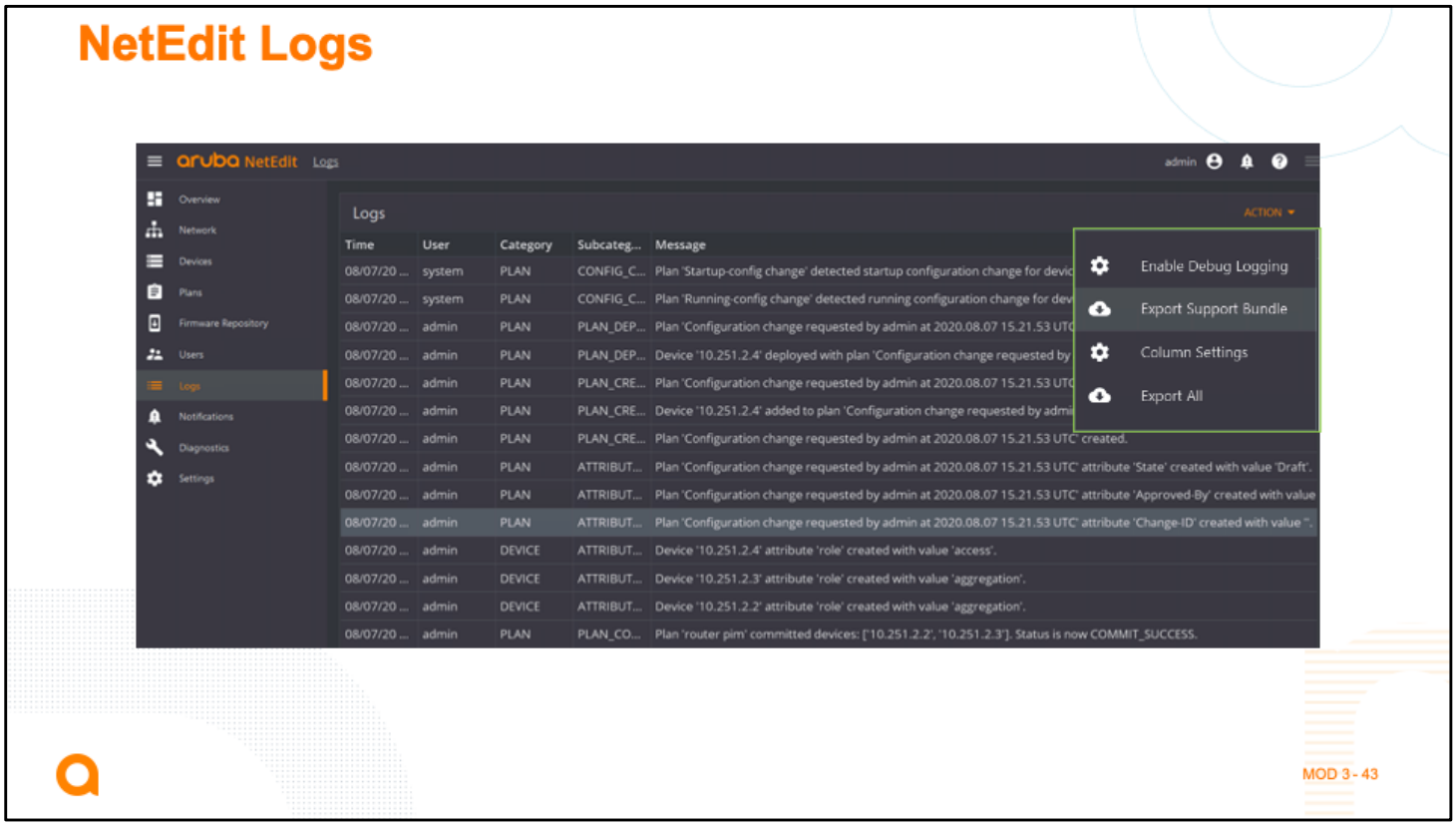
At 20:37:33, an alert is triggered because the monitored traffic rate exceeds the high threshold.

At 20:39:15, the alert is cleared because the monitored traffic rate (in packets per second) is lower than the low threshold.

At 20:40:30, the thresholds are updated. In this case, the thresholds are set significantly higher because the algorithm includes all the data in its continuous learning window, which included the time in which the traffic rate was much higher than the previous threshold.

If the script specified a longer initial learning time, such as one day, the calculations used to create the thresholds can include the typical fluctuations in data that can occur, resulting in more appropriate thresholds and alerts that trigger only for significant anomalies.

You can refer to the AOS-CX 10.05 Network Analytics Engine Guide for additional reference.



The Logs page shows all application logs. Each log entry includes the time stamp, user, category, subcategory, and message. You can change support log's debug log level with the Enable Debug Logging button. Download the support log bundle with the Export Support Bundle button.

Hide or reveal any column with the Column Settings button. Export logs to a .csv file with the Export All button. The logs are also stored on the file system under /opt/netedit/logs/ events.log. The log rotation service runs once a day. The events.log file is rolled at 100 MB to events- [timestamp].gz keeping only the 10 latest files.


Lab Activity
Lab 03


Rev 20.41
aruba
a Hewlett Packard Enterprise company
VSX, VSF, and L2 Technologies
Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)

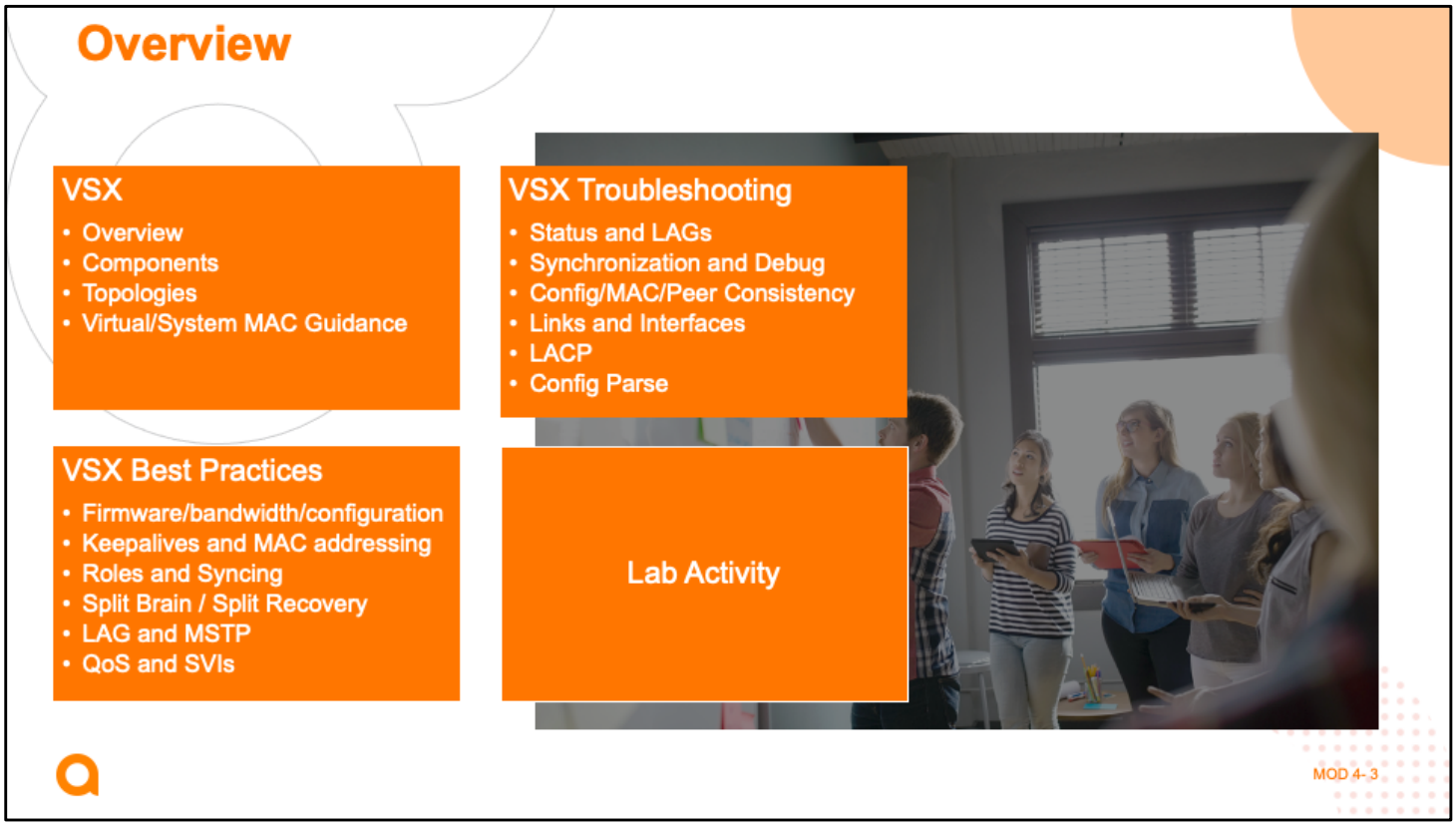
Objectives

- Learning Objectives
- Configure, validate, and troubleshoot Network Virtualization features
- VSF
- VSX



After completing this module, you will be able to configure, validate and troubleshoot network virtualization features.

This includes Aruba Virtual Switching Framework (VSF) and Virtual Switching extension (VSX).



After the physical layer, everything sits atop Layer 2. Resiliency is vital, as is your ability to deploy and troubleshoot these systems. This module helps to enhance your understanding, helping you move toward deployment and diagnostic mastery of VSF, VSX, and their associated L2 protocols and best practices.

First you explore VSF operation and requirements, along with how to mitigate and eliminate so-called “split-brain” conditions. Then you will focus on troubleshooting these systems by validating over system status, along with members, links, and topology.

You move on to VSX, with an overview of operation, components, and topologies. You will also get guidance from Aruba about configuration Virtual and System MAC addressing to ease future operations and diagnostics.

Then you will dive deeper into this study of VSX best practices – related to Firmware, bandwidth, configuration, keepalives, and MAC addressing. You also explore best practices related to roles, synchronization, split recovery, LAG and MSTP, QoS, and SVIs.

Armed with this knowledge of operation and best practices, you are ready to hone your VSX troubleshooting skills. You will use show and debug commands to validate overall system, LAG, and synchronization status, and learn about some key debugging commands and techniques. You will validate VSX peer consistency related to configuration, MAC addressing,

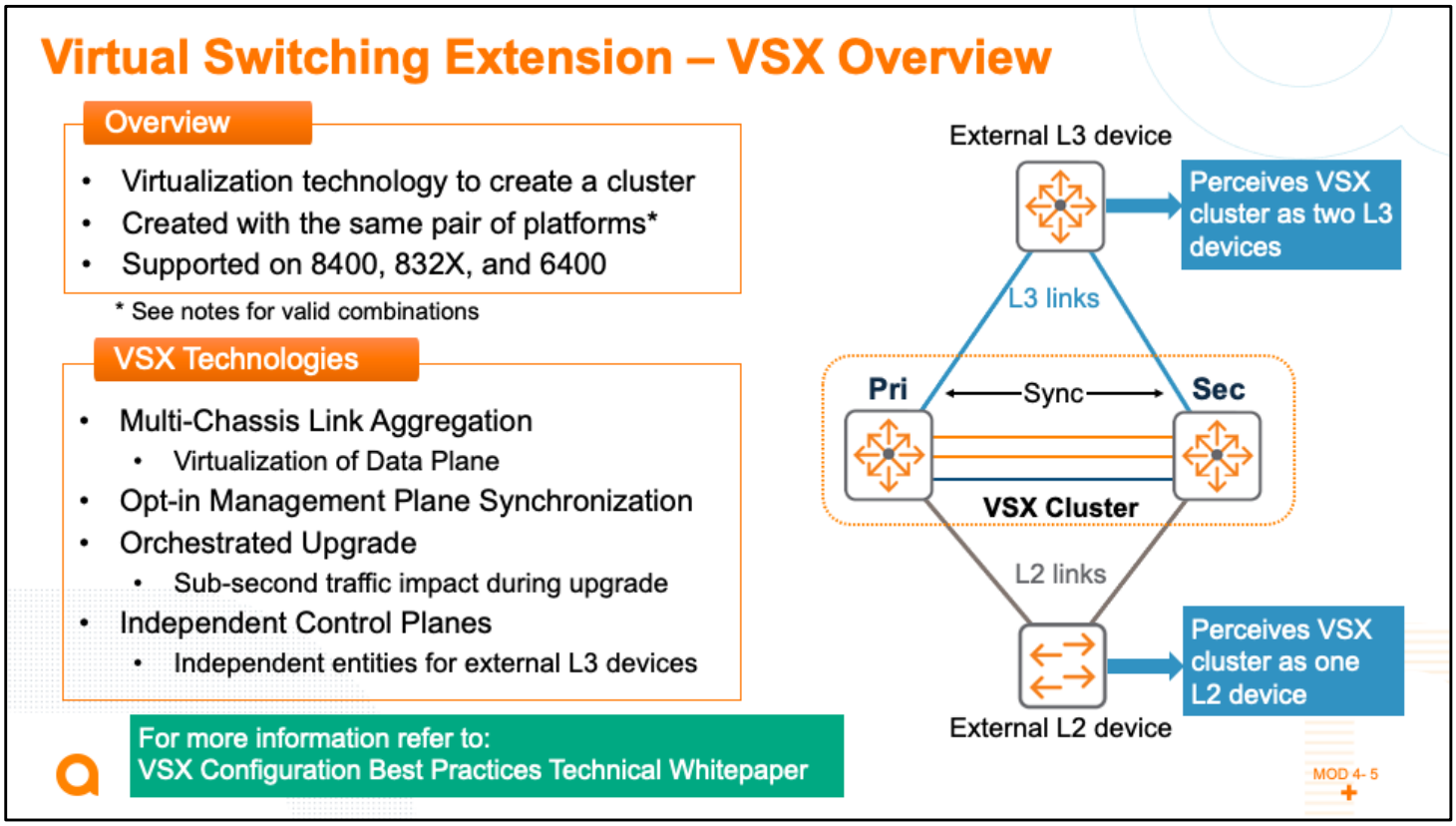
and more. Then you look at links, interfaces, LACP, and the configuration parser – a great tool for doublechecking your deployment.

Finally, you will explore some VSX Split Brain detailed scenarios before moving into some hands-on lab activities.

aruba
a Hewlett Packard Enterprise company

VSX

Overview, components, topologies and virtual MAC and system MAC guidance



Aruba Virtual Switching Extension (VSX) is a virtualization technology to create a cluster of two AOS-CX switches from the same model - 6400, 8320, 8325 or 8400. The feature is not supported on AOS-CX 6300 and 6200. VSX requires consistent platforms - an 8325 switch cannot run VSX with 8320 switch.

VSX relies on multiple technologies:

Multi-Chassis Link Aggregation (MC-LAG). Enables VSX to virtualize the Data Plane, thus reducing the reliance for Spanning-Tree and enables you to use the infrastructure more effectively. VSX peers synchronize the following tables:

- Link Aggregation Control Protocol (LACP)
- Media Access Control (MAC)
- Address Resolution Protocol (ARP)
- Spanning-Tree Protocol (STP)
- Dynamic Host Control Protocol (DHCP)

Opt-in Management Plane Synchronization. You can choose which feature configuration is synchronized from the VSX Primary to the VSX Secondary. Any show command or REST API call can display information of the VSX peer from the local node with a single SSH session or API call.

Orchestrated Upgrade. Also called “VSX Live Upgrade” provides the stacking for sub-second traffic impact for VSX cluster firmware upgrade.

Independent Control Planes. External Layer 3 devices perceive the VSX peers as independent entities that will interact with the VSX Cluster. Thus, VSX nodes can have an Active packet forwarding decision, using standard routing protocols such as OSPF and BGP. This provides industry-leading High Availability.

Note. Here is a list of the supported combinations in AOS-CX 10.05

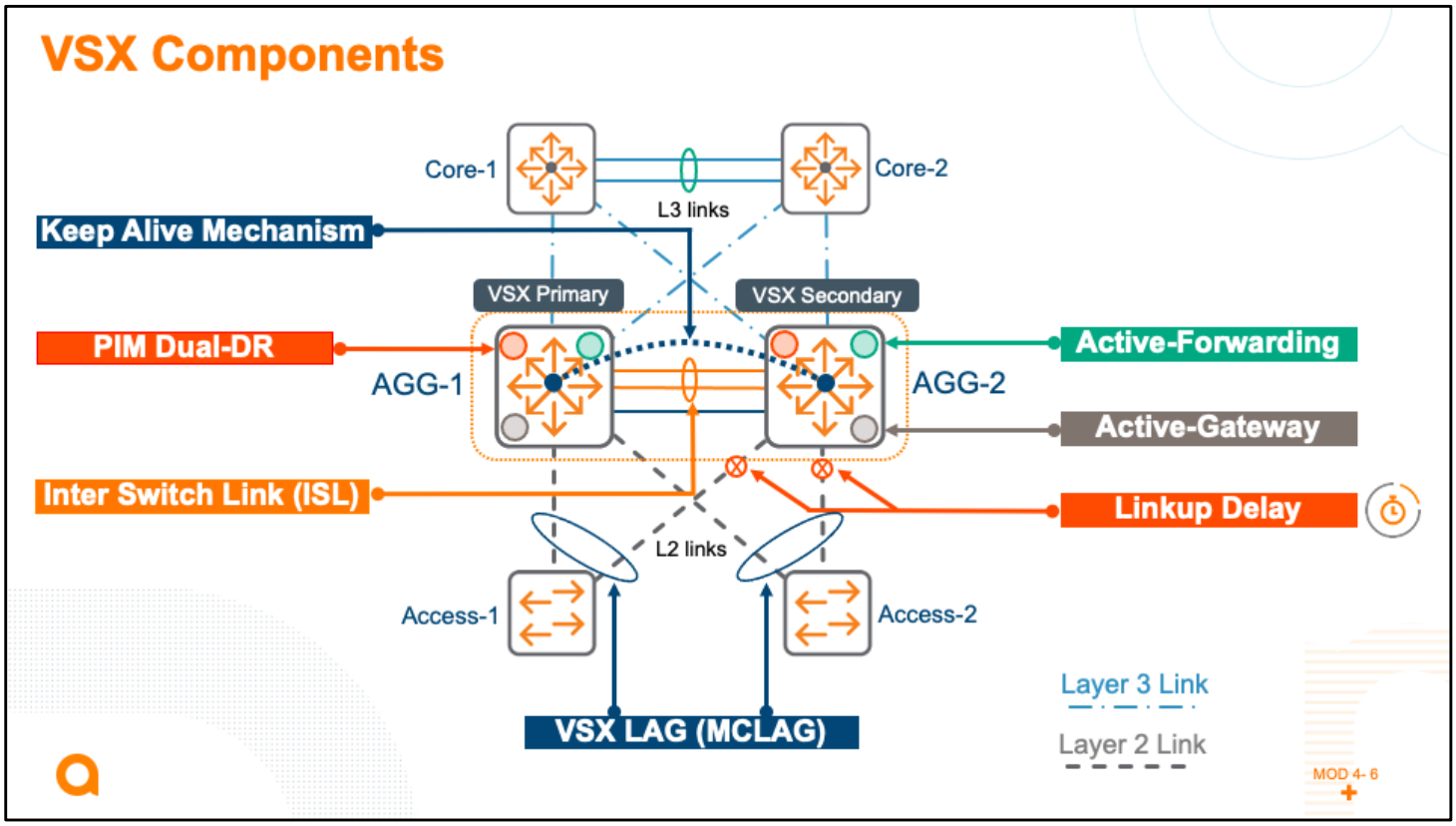
Aruba CX 6400: 6405 + 6405 or 6410 + 6410. The pair 6405 + 6410 is not supported.

Aruba CX 8320: JL479A+JL479A, JL581A+JL581A, JL579A+JL579A, JL479A+JL581A, JL479A+JL579A, JL581A+JL579A

Aruba CX 8325: JL624A/JL625A+JL624A/JL625A, JL626A/JL627A+JL626A/JL627A, JL624A/JL625A+ JL626A/JL627A

Aruba CX 8400: 8400 + 8400

For more information about VSX you can refer to the VSX Configuration Best Practices Technical Whitepaper.



The figure shows VSX components, as described in more detail below.

Inter Switch Link (ISL). VSX nodes interconnect using Inter Switch Link (ISL), using the ISL Protocol (ISLP). Switches use ISLP to synchronize multiple software and hardware tables including:

- LACP states and table
- MAC tables
- ARP tables
- STP states
- DHCP states

ISLP also comprises a framework for synchronization of the management plane from the VSX primary to the VSX secondary. ISL uses its Keepalive mechanism with ISL Hello packet, Dead-interval and Hold-time, keeping default values is recommended. Please do not confuse this ISL Keepalive mechanism with the VSX Keepalive mechanism, which is described below. Here are some key aspects of ISL:

- The ISL physical ports must be directly connected with no intermediate active Layer 2 nodes.
- Physical ports must use the same speed.
- ISL is typically a standard Link Aggregation (LAG) of two to eight physical ports.

2. VSX LAG. This is the AOS-CX name for Multi-Chassis Link Aggregation (MLAG) technology where two or more links across two switches are aggregated together to form a LAG which act as a single logical interface. VSX LAG supports all the standard LAG adjustments:

- Timers
- Layer 2 hashing
- Layer 3 hashing
- LACP fallback
- LACP active mode
- Static LAG
- Only Layer 2, i.e. no routed port, since routed VLANS are supported.

3. VSX Keepalive is a UDP probe on port 7678 by default, but this is configurable. This mechanism sends hello packets between the two VSX nodes to detect a split-brain situation. This probe can be established over a direct link or routed path.

When a split-brain condition is detected, the VSX secondary brings down all its physical ports which are members of any VSX LAG. Use the command show interface brief on the VSX-Secondary to see that its LAG ports are “down by feature”.

Note. AOS-CX version 10.05 does not support the use of VSX Keepalive over the OOBM port.

4. Active-Gateway. This is a default-gateway Virtual IP address (used for endpoints in a subnet) hosted on both VSX primary and VSX secondary. This Virtual IP comes with a Virtual MAC address. Both VSX nodes can reply to ARP request with the same virtual MAC.

The VSX primary periodically sends Gratuitous ARP (GARP) and Broadcast Hello packets sourced from this virtual MAC. The VSX primary relays DHCP request or serves DHCP offers to endpoints. The VSX secondary takes over in case of primary failure.

Warning: If you administratively shut down the SVI hosting the active-gateway on one VSX node, traffic could be dropped. You will get the following warning message:

switch(config-if-vlan)# shutdown

Warning: Active gateway is configured on this interface vlan <id>.

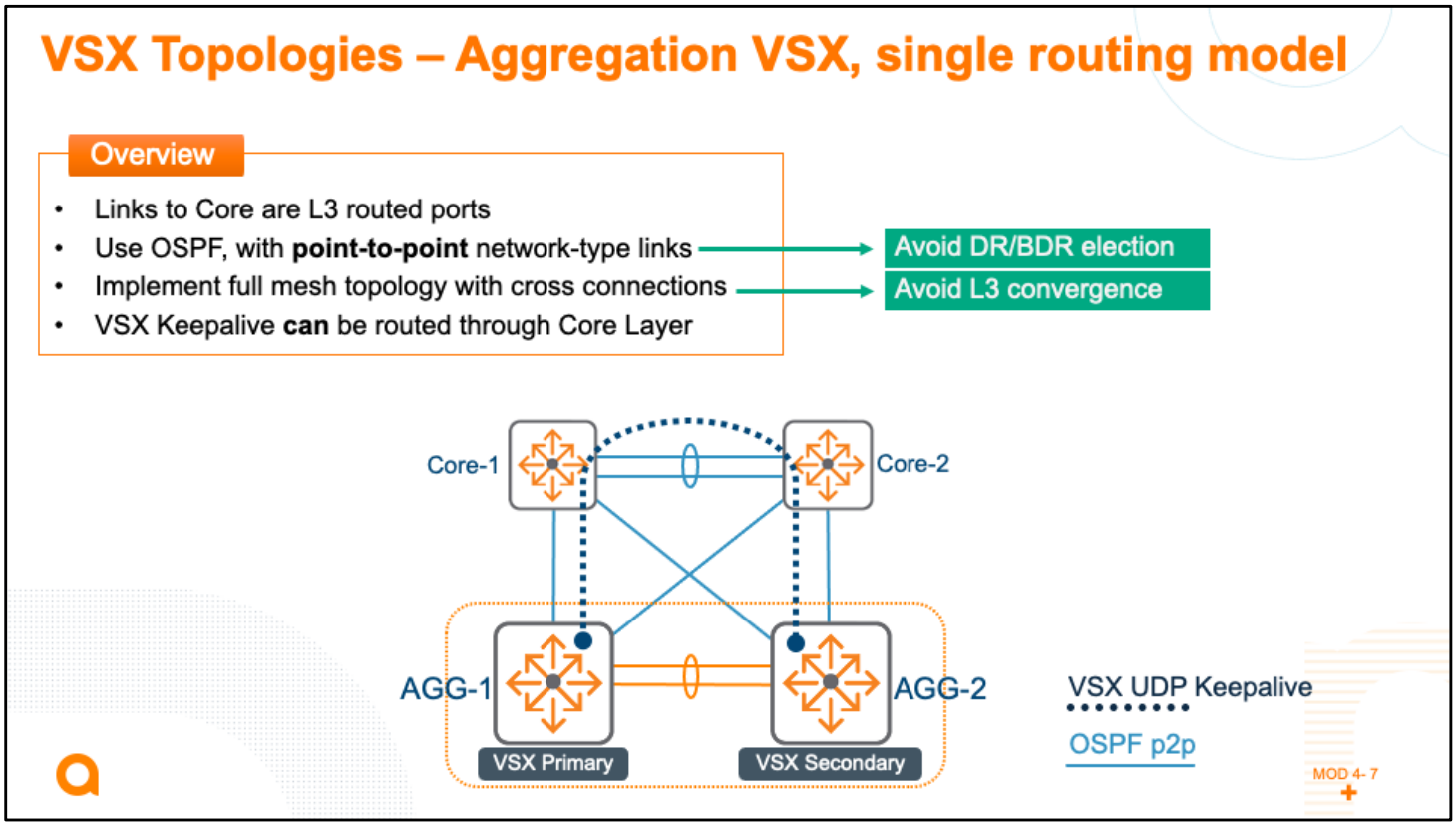
Shutting down the interface may result in traffic loss.

5. Active Forwarding is configured in case VSX LAG is used for upstream connectivity and when North-South traffic can reach one VSX node while the destination MAC is actually the other VSX peer. This situation can happen due to ECMP on upstream routers. To avoid this sub-optimal pathing and unnecessary traffic load on the ISL, configure active-forwarding on the upstream Transit VLAN SVI. When configured, the node receiving packets will process Layer 3 lookup on behalf of the other VSX node and will forward traffic to connected downstream devices without forwarding traffic to ISL (ISL off-load),

Another benefit of VSX active-forwarding is to offer routing high-availability in the event of one VSX node failure. In the context of upstream VSX LAGs and upstream ECMP routing, the remaining VSX node will process L3 lookups on-behalf of its failing peer, masking any routing protocol convergence window and associated issues.

6. PIM Dual-Designated Router. PIM DR and PIM routing is handled by the VSX primary. In order to avoid a long convergence time in case of a VSX primary failure, the VSX secondary can also establish PIM peering, send PIM join messages and build Shortest Path Tree for multicast. This is achieved when PIM active-active is configured.

7. Linkup-Delay. Inside the VSX stack, the switches synchronize their ARP and MAC tables. This happens in two phases: a software exchange and a hardware programming of the ASICs. For the second part, when the VSX node joins the VSX stack it can take several minutes (depending on the network size) to get complete MAC and ARP tables synchronized and configured. During this synchronization time, the VSX LAG ports are kept down to make sure that there will not be any transient traffic loss. This timer is user-configurable and will depend on the network size.

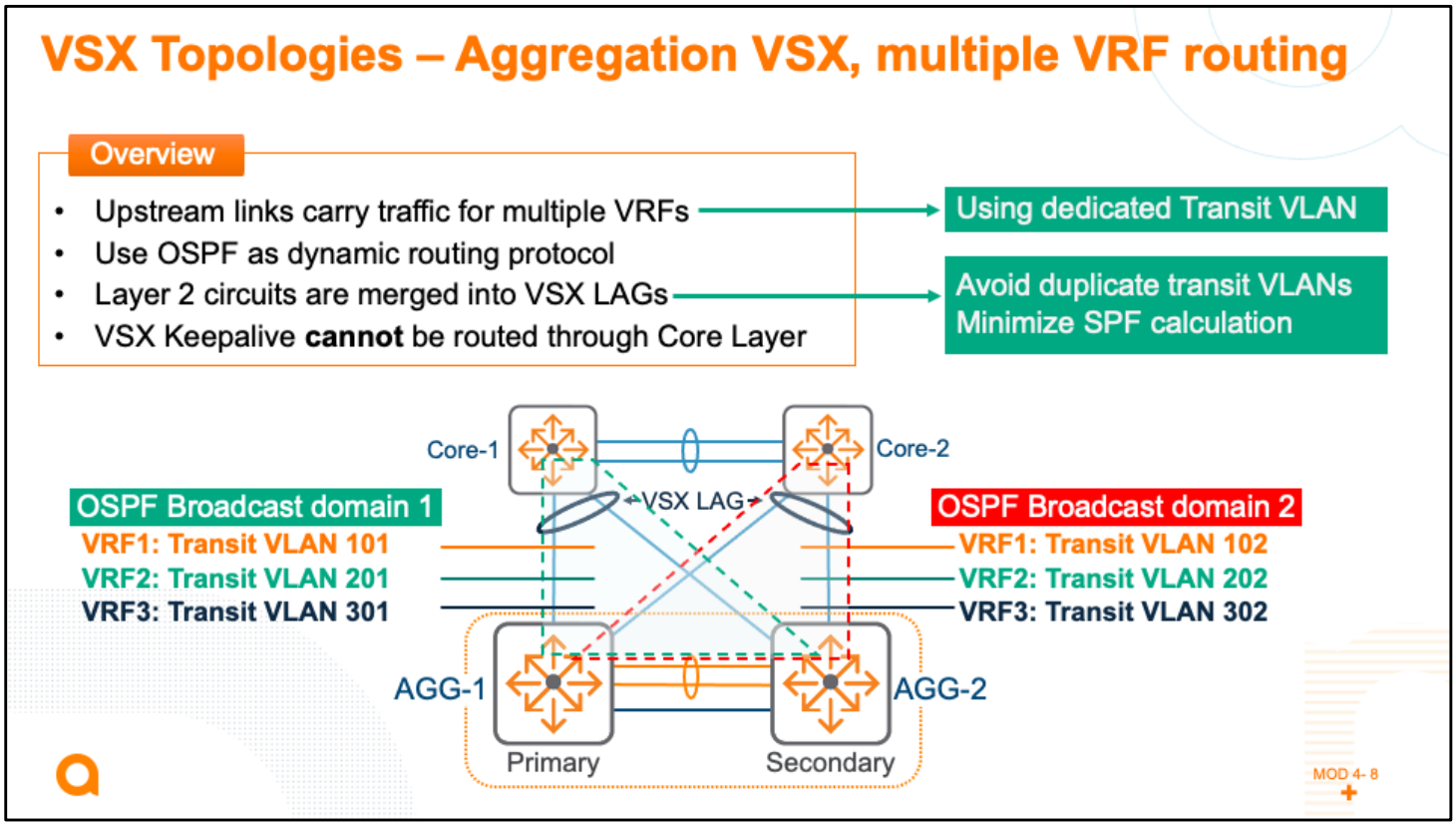


Aggregation VSX with single VRF routing model

The links between the Core Layer and the VSX cluster are Layer 3 circuits using routed ports on each CX switch. You should use OSPF as the dynamic routing protocol, with OSPF links configured as the point-to-point network-type – to avoid DR/BDR election. Also, you should implement a full mesh topology with cross connections - each aggregation switch must have a physical connection to both cores.

Suppose you did not follow this advice, and used a non-meshed topology, such that AGG-1 and AGG-2 each had a single connection to the core layer. If a link fails between the aggregation and Core layers, routing convergence must occur, which can impact traffic flow. Even if OSPF timers can be adjusted to minimize such impact to sub-second, this topology should be restricted to environments where full-mesh fibers is not possible.

VSX UDP keepalive probes can be routed through upstream Core-1 and Core-2 during a split event.

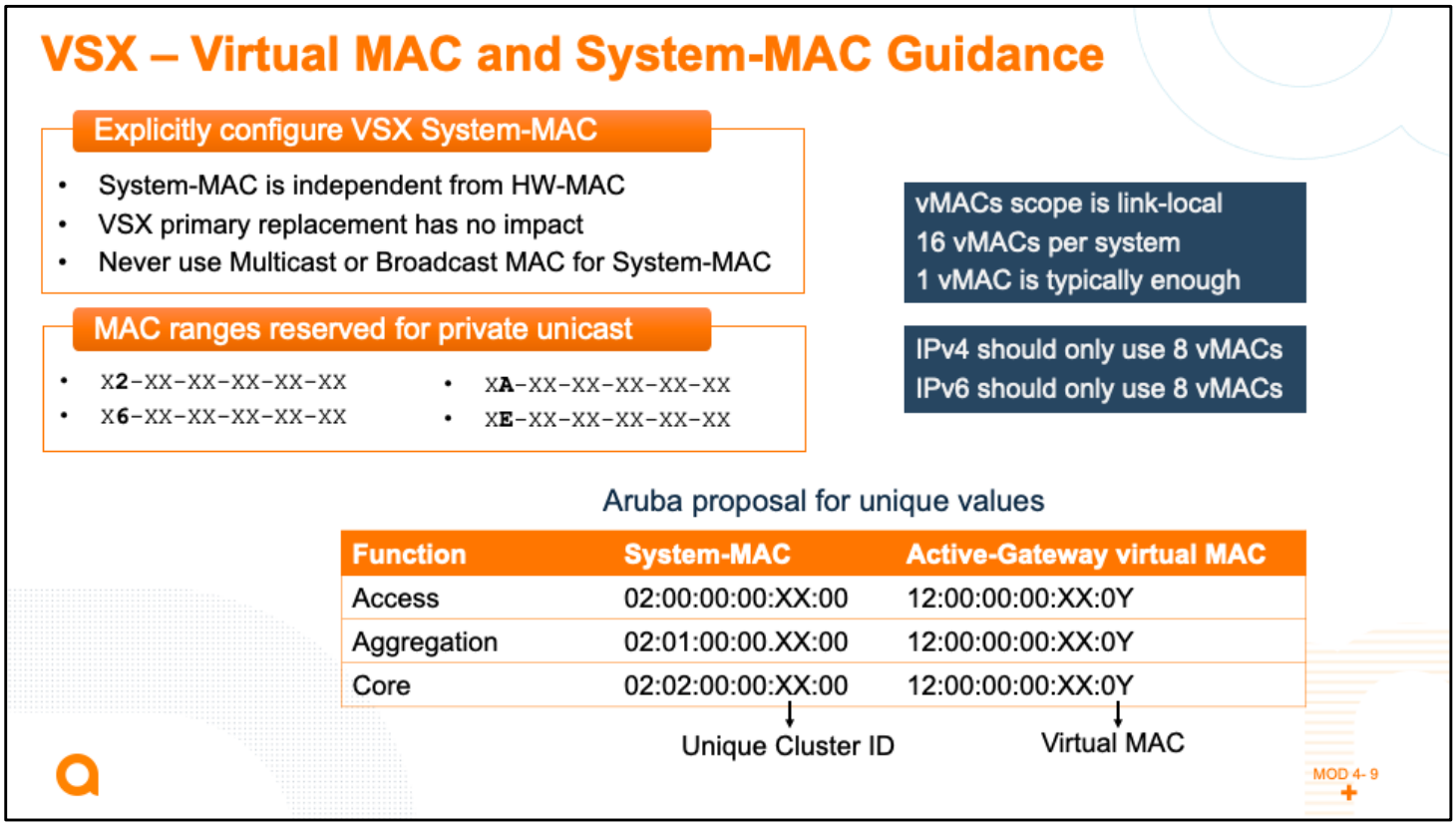


Aggregation VSX with multiple VRF routing model

AOS-CX version 10.05 and earlier does not support sub-interfaces. Therefore upstream links must carry the traffic for various VRFs over dedicated Transit VLANs – to interconnect the VSX cluster routing domain to the upstream Core layer for each VRF. To avoid duplication of multiple Transit VLANs per VRF and to minimize SPF routing calculation, the upstream L2 circuits are merged into two upstream VSX LAGs.

Note. VSX UDP keepalive is NOT possible over the upstream VSX LAG connectivity. In such scenario, the keepalive must be implemented over a dedicated point-to-point L3 circuit that need not be directly connected - some intermediate active equipment can be traversed by the L3 UDP probe only if it does not use a VSX LAG path.

VSX UDP keepalive would have been possible through L2 point-to-point circuits. However due the number of associated Transit VLANs to manage, this would be the very last implementation option.



It is important that you do not leave the VSX system MAC blank, thus allowing the default hardware system MAC to be used. The best practice is to set the VSX system-mac to a specific value. This ensures that the VSX system-mac is independent from the physical hardware MAC address. Thus, VSX primary hardware replacement has no impact on configuration. Simply use the same configuration from the previous hardware - the cluster ID remains unchanged. This ensures that VSX primary HW replacement is hitless for the VSX secondary.

There are four MAC ranges reserved for private use for unicast (with second least significant bit of the first octet of the unicast address set to 1)

x2-xx-xx-xx-xx-xx

x6-xx-xx-xx-xx-xx

xA-xx-xx-xx-xx-xx

xE-xx-xx-xx-xx-xx

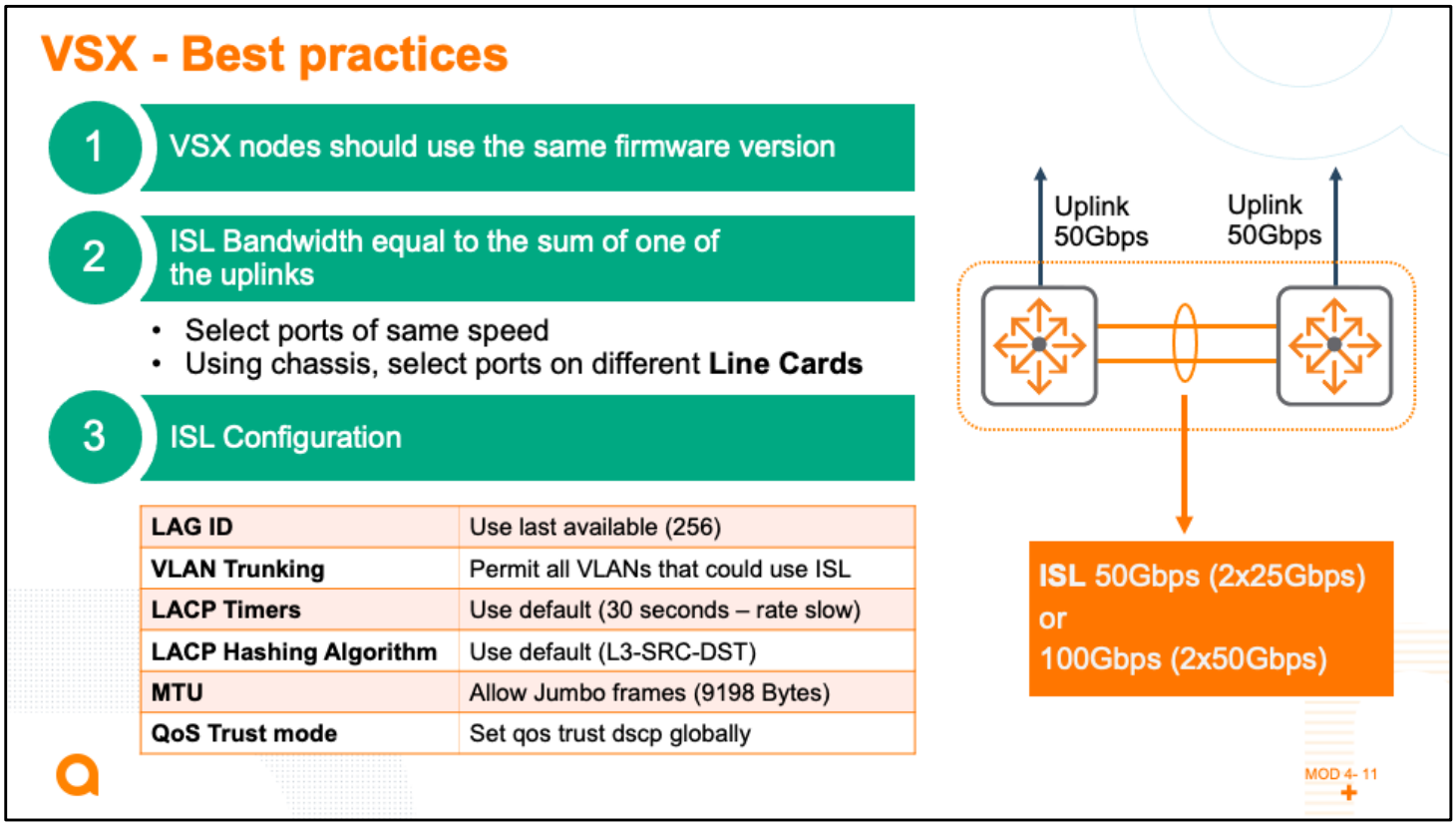
The scope of this vMAC is link-local. Consequently, the same virtual MAC address value can be used on any L3 VLAN interface (SVI)

If some systems are dual-attached to two different SVIs and you would like to see distinct

MAC address for the next-hops over these separate interfaces, then 16 virtual MACs are available. For dual-stack IPv4 and IPv6, 16 vMACs can be used, is a best practice to use only 8 vMACs for IPv4 and 8 different for IPv6.

Note. Multicast or Broadcast MAC addresses must NOT be used for System-mac.


aruba
a Hewlett Packard
Enterprise company
VSX
Best Practices



Follow the best practices when creating a VSX cluster:

Both VSX nodes should run the same Firmware code to avoid incompatibilities. Use the ”show version” command.

ISL link: If uplinks from AGG-1 fail, traffic received on this switch is redirected to AGG-2 over the ISL. So, ensure that your ISL bandwidth is at least equal to the sum of uplinks bandwidth of one VSX node. You should also select at least two ports of the same speed. For a chassis, select ports from different Line Cards. Following the previous best practices a typical ISL bandwidth would be at least 2x40G (QSFP+) or 2x50G (SFP56) or 2x100G (QSFP28).

ISL configuration best practices:

LAG numbering: Use the last available LAG ID for the ISL, in AOS-CX 10.05 this is 256

VLAN Trunking: Permit all VLANs on the ISL LAG for simpler configuration. Specifying a restrictive list of VLAN IDs is a valid option if the administrator needs more control.

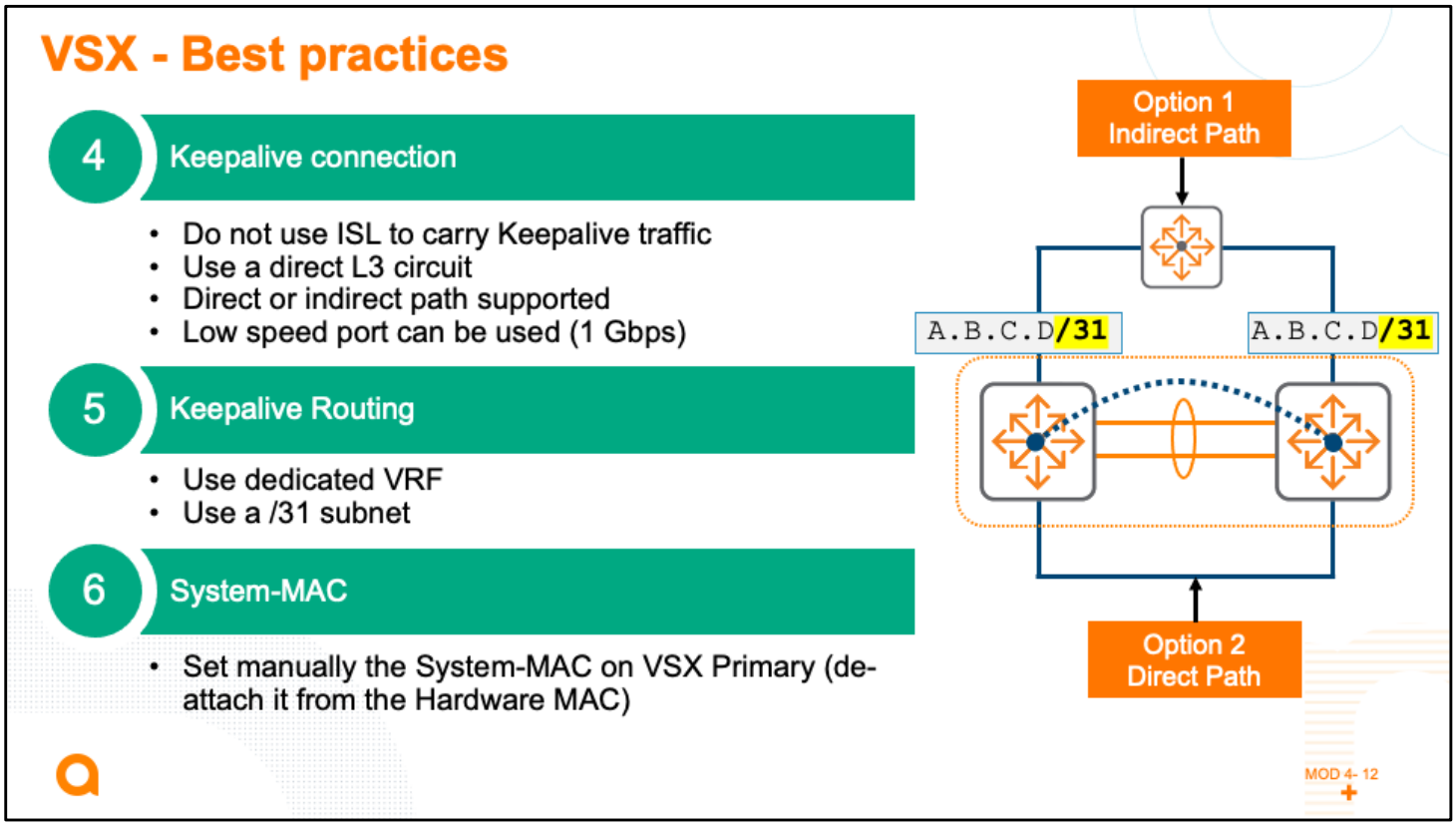
LACP timers: Keep the default long timer (30 seconds for LACP rate slow)

LACP hashing algorithm: Keep default l3-src-dst

MTU: Set the value to 9198 Bytes, this value supports Jumbo frame traffic over ISL.

ACL: Is not recommended to set any access-list on the ISL LAG.

QoS trust mode: The ISL LAG rely on the qos trust dscp command that is globally configured. If not configured globally then this command has to be set directly on the ISL LAG interface.

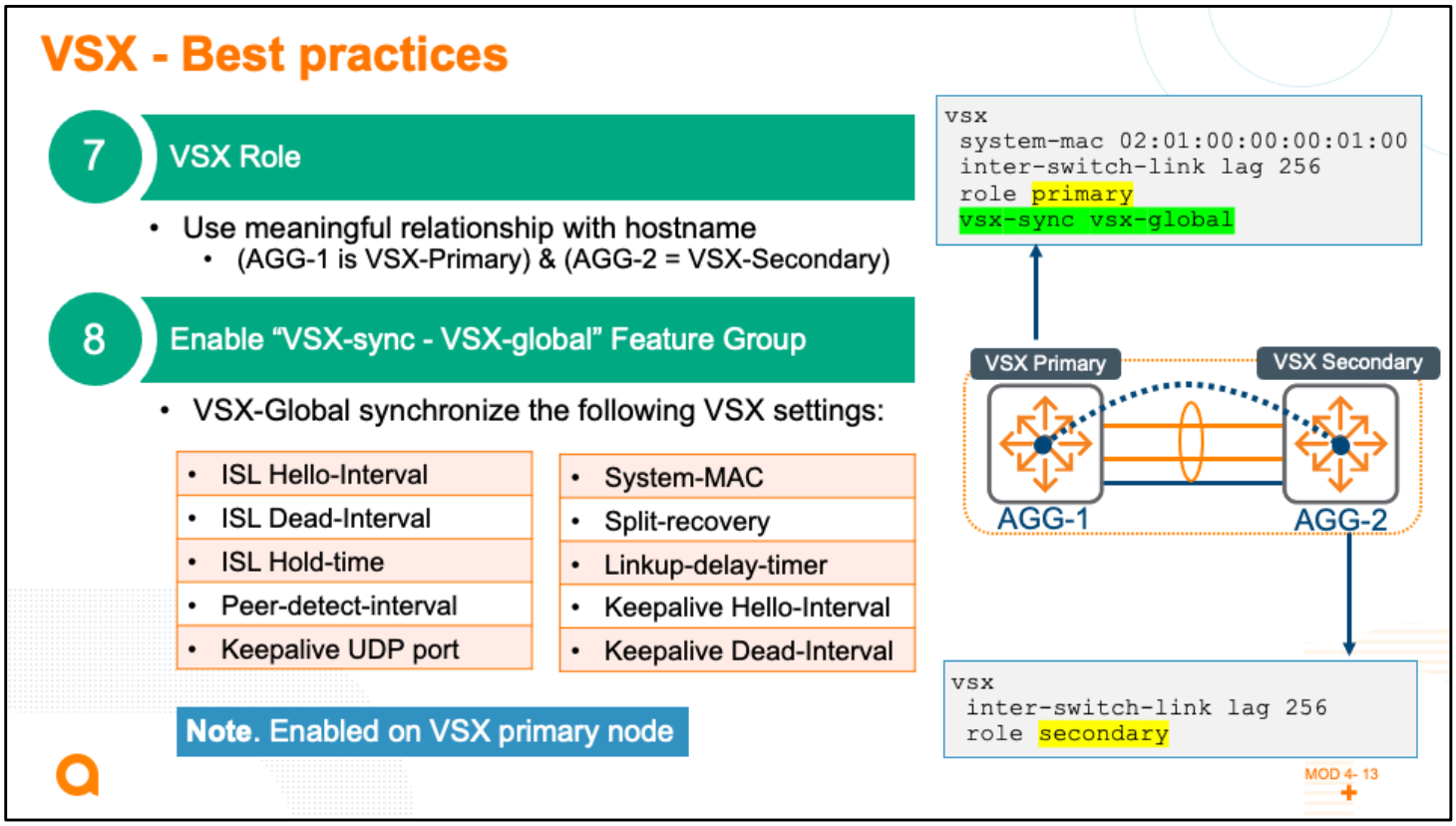


4. Keepalive connection: Use a direct L3 circuit, which can be low speed port, 1Gbps between both VSX nodes. This circuit need not be directly-connected and the path can include active L2 and L3 equipment. In the case of a chassis switch (AOS-CX 6400 or 8400), if possible, you should use a port from a different Line Card than the ones used for the ISL ports.

5. Keepalive routing: Use a dedicated VRF for keepalive simplifies the operations and prevents any impact from routing change on any other VRF.

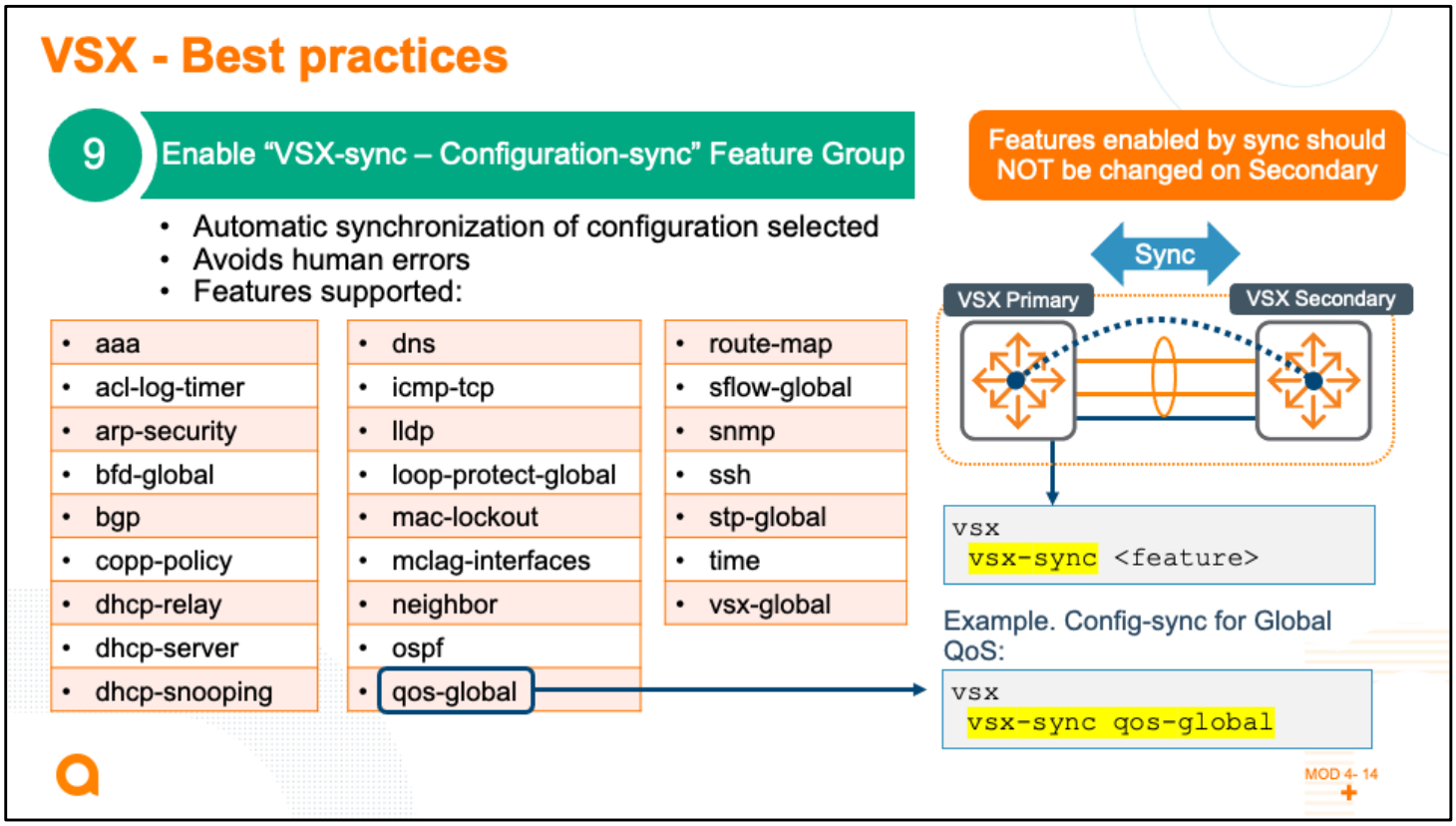
Keepalive subnet: Use a /31 subnet as only 2 nodes will communicate.

6. System-MAC: Set the system-mac explicitly on the VSX primary node. Remember that the main advantage is to be independent from the physical Hardware MAC address. In case of Hardware replacement of the VSX primary, the new switch can be configured exactly with the same configuration that the previous unit and there will be no impact on the secondary node which remain in the same cluster ID. Hardware replacement is hitless for VSX secondary.



7. VSX-Role: The best practice is to have a meaningful relationship with the switch hostname. For example, AGG-1 is the VSX primary and AGG-2 is the VSX secondary.

8.You should syncrhonize the configuration for VSX itself, by using the vsx-sync vsx-global command. The configuration that is syncrhonized includes the list of items showed in the figure.



9. Configuration-sync Feature Group. VSX cluster manages independent management platforms. This implies that you must independently configure the VSX nodes, which can cause human errors related to misconfiguration. To avoid this, enable Configuration-sync as a best practice.

The synchronization process in VSX requires no configuration from the administrator. However, you must explicitly define the Features to sync. The figure shows a list of the current features supported on 10.05 code

To enable config-sync use the following commands:

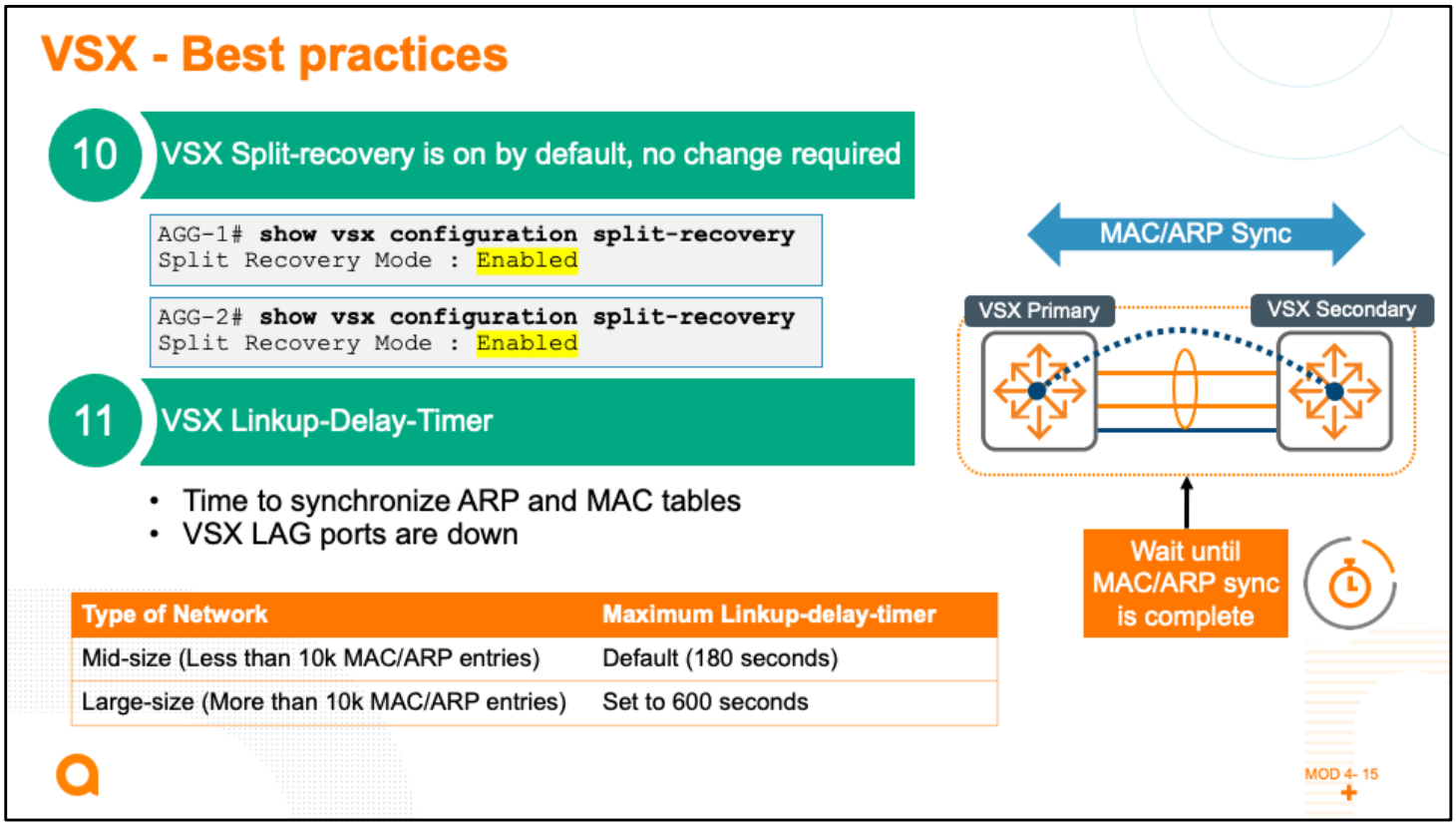
AGG-1(config)# vsx

AGG-1(config-vsx)# sync aaa acl-log-timer arp-security bfd-global bgp copp-policy dhcp-relay dhcp-server dhcp-snooping dns icmp-tcp lldp loop-protect-global mac-lockout mclag-interfaces neighbor ospf qos-global route-map sflow-global snmp ssh stp-global time vsx-global

Note. VSX will synchronize most of the configuration but not everything, specific information must still be entered directly into the VSX Secondary. The sync feature should only be used if the configuration will be the same on both switches; some exceptions might be BGP peer configuration and OSPF with different external connections.

Note. Features that have been enabled for the synchronization should not be changed on the secondary node.

Note. During primary down, no config changes should be made on the secondary node for features that have been enabled for sync, since they will be overwritten by the sync process when the primary joins again.



10. VSX Split recovery allows VSX nodes to recover and re-synchronize from a split when the ISL link is available again. This feature is enabled by default - no action required on your part. Use the show vsx configuration split-recovery command to verify the state of this feature.

11. VSX Link Delay Timer. Inside the VSX stack, the switches synchronize their ARP and MAC tables. This happens in two phases: a software exchange and a hardware programming of the ASICs. In the first phase (Initial phase) the switch learns all the LACP+MAC+ARP+MSTP+DB entries, the timer (initial) in this phase is not configurable. For the second phase (Linkup Delay phase), when the VSX node joins the VSX stack it can take several minutes (depending on the network size) to get complete MAC and ARP tables synchronized and configured. During this synchronization time, the VSX LAG ports are kept down to make sure that there will not be any transient traffic drop. In the second phase the switch waits for:

Installing the downloaded entries to the ASIC

Establishing router adjacencies with core nodes

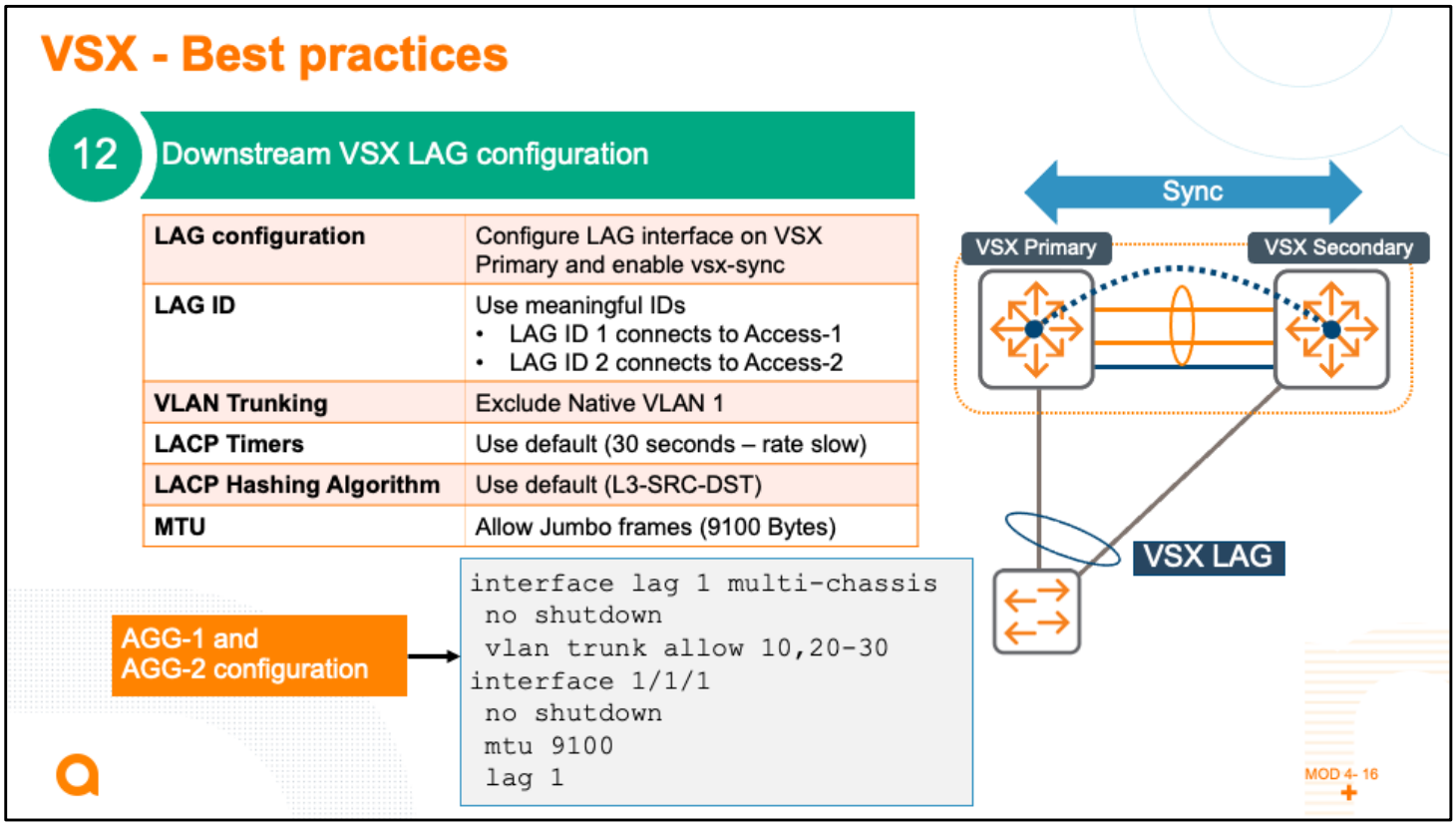
Learning upstream routes

In VSX this period will auto-adapt based on the completion of table exchanges and ASICs readiness, so the time to wait for VSX LAG links being activated is less or equal to the maximum timer being set. Depending on the network size you will use modify this parameter or not.

Mid-size networks with less than 10,000 MAC or ARP entries, you can leave default values (180 seconds).

Large-size networks with more than 10,000 MAC or ARP entries you must setup the maximum linkup-delay-timer to 600 seconds.

Note. In AOS-CX the system is smart, if the tables and the router adjacencies are completed and ready to use, the system starts forwarding traffic, there is no need to wait until the Linkup Delay timer expires.



12. Downstream VSX LAG configuration

VSX LAG: Create multi-chassis LAG interface on the VSX primary with all settings and then create the mirrored lag interface on the VSX secondary. LAG interface settings will be synchronized automatically except for the “no shutdown” command. This command must be entered manually on the VSX secondary. Once the interface LAG is created it can be mapped to the physical ports.

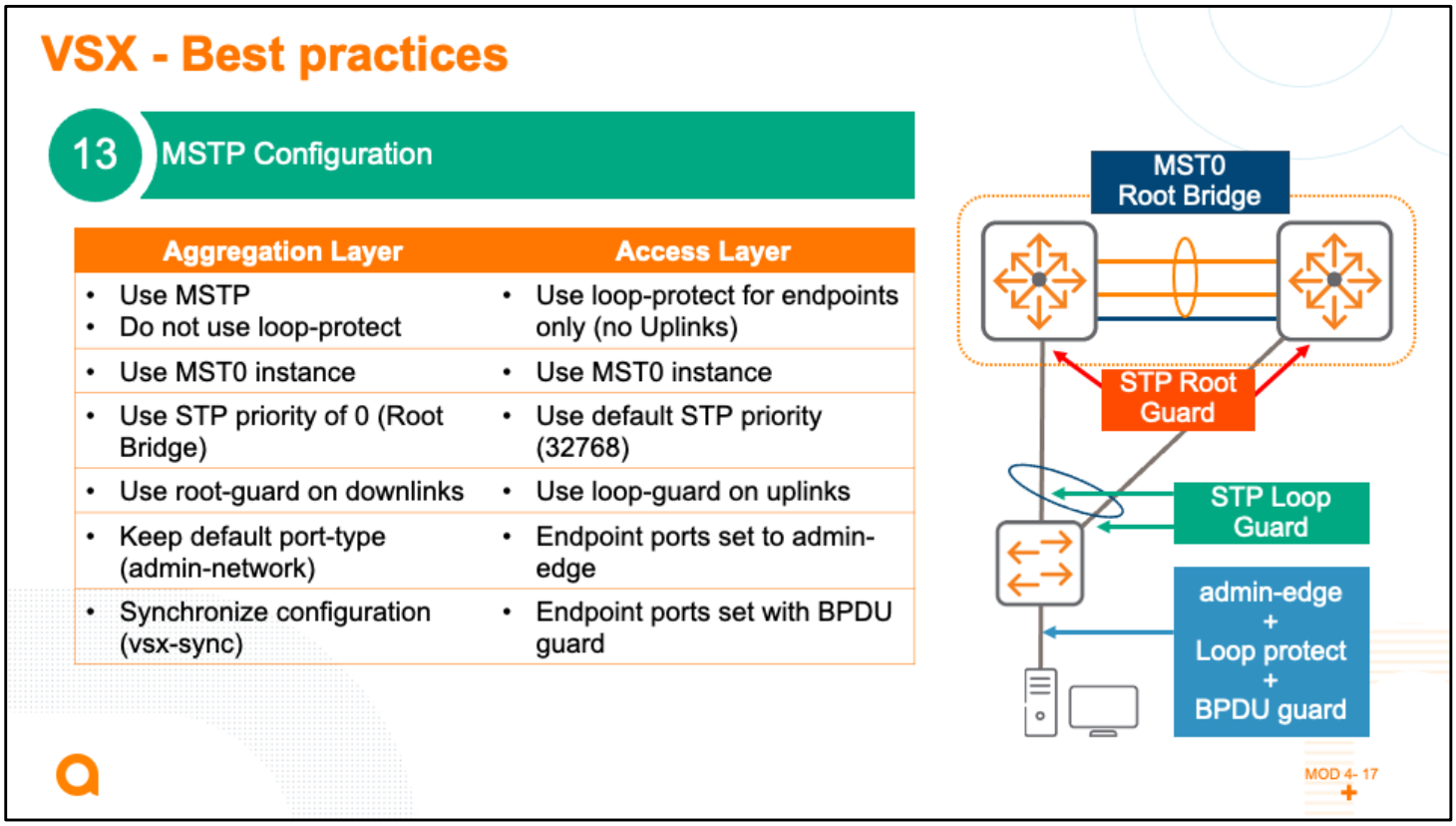
Allowed VLANs: Exclude the native VLAN 1 from being propagated, this method avoids Layer 2 loop preventing broadcast storms. In case of access switch Zero-Touch-Provisioning use-case, this VLAN exclusion on the 802.1Q trunk port is performed after the ZTP process.

LAG numbering: Use LAG IDs that match the connections with the Access switches, i.e. LAG ID = 1 for connecting to the ACC-1 switch on port 1/1/1 and LAG ID = 2 for connecting to the ACC-2 switch on port 1/1/2

LACP Timers: Keep the default link timer (30 seconds = LACP rate slow)

Best practices for MTU: You must take into account the appropriate frame size to support features that exceed the normal MTU values such as Dynamic Segmentation. The IP path from the access devices (Access layer switches and APs) can provide a MTU of at least 1564 Bytes to the Mobility Controllers. Jumbo frames should also be taken in count the recommended value for these frames is 9100 Bytes.

Hashing Algorithm: Keep the default l3-src-dst. This option has an effect only if at least 2 ports per VSX node are members of the same VSX LAG.



13. MSTP Configuration

Best practices on Aggregation layer are:

Use MSTP and no loop-protect.
Use the default common instance 0: MST0
Lower the spanning-tree priority to make VSX aggregation the Root Bridge, preferred value is 1.
Use STP root-guard on all downlinks to prevent any access switch from become Root Bridge.
Keep the default port-type admin-network
Synchronize the configuration using the vsx-sync process.

Example of configuration on AGG-1

AGG-1(config)# spanning-tree
AGG-1(config)# spanning-tree priority 1
AGG-1(config)# spanning-tree trap topology-change instance 0
AGG-1(config)# interface lag 1 multi-chassis
AGG-1(config-lag-if)# spanning-tree root-guard

Note. If STP has been synchronized AGG-2 will have the same lines of configuration.

To verify the previous settings, use the following commands:

show spanning-tree mst 0 interface lag 1 detail

show running-config

Best practice on Access Layer are:

Use loop-protect for all endpoint access ports but never configure it on uplinks. Set the re-enable timer to 60 minutes.

Keep the default common instance 0: MST0

Keep the default spanning-tree priority of 8.

All endpoint access ports are admin-edge

– Should not receive any BPDU (BPDU guard)

– Should not trigger any Topology Change Notification (tcn-guard)

Use loop-protection on all endpoint access ports as an extra-protection mechanism. This is in case of MSTP BPDUs are filtered by insertion of unmanaged switches which create a loop.

Use loop-guard on all uplinks to prevent flood due to failure of BPDU reception.

Example of configuration for Access switch:

ACC-1(config)# spanning-tree

ACC-1(config)# loop-protect re-enable-timer 3600

ACC-1(config)# interface lag 1

ACC-1(config-lag-if)# no shutdown

ACC-1(config-lag-if)# description UPLINK-TO-AGG

ACC-1(config-lag-if)# no routing

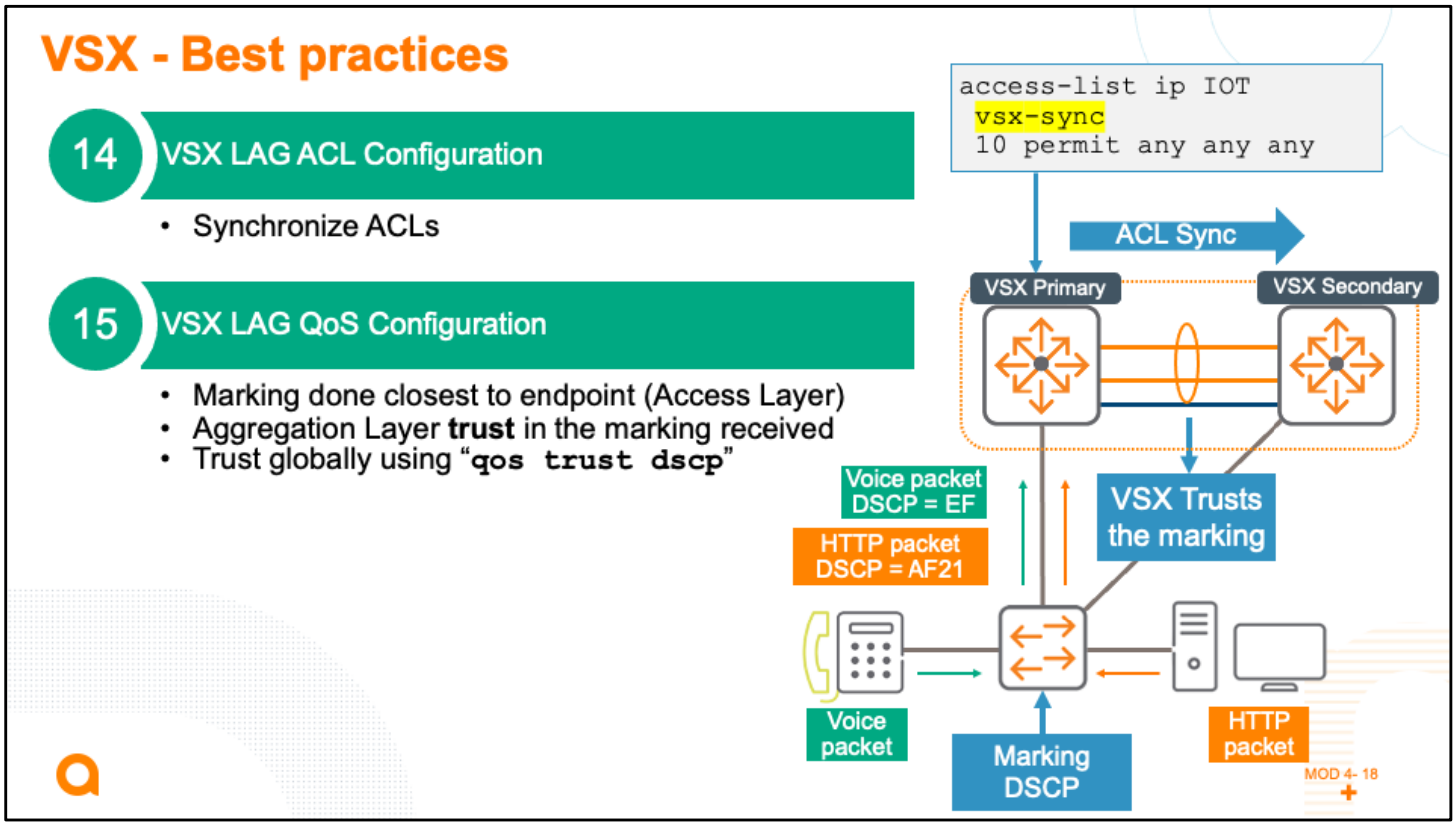
ACC-1(config-lag-if)# vlan trunk native 1

ACC-1(config-lag-if)# vlan trunk allowed 10,20

ACC-1(config-lag-if)# lacp mode active

```
ACC-1(config-lag-if)# spanning-tree loop-guard

ACC-1(config)# interface 1/1/1
ACC-1(config-if)# no shutdown
ACC-1(config-if)# description ENDPOINT-1
ACC-1(config-if)# no routing
ACC-1(config-if)# vlan access 10
ACC-1(config-if)# spanning-tree bpdu-guard
ACC-1(config-if)# spanning-tree port-type admin-edge
ACC-1(config-if)# spanning-tree tcn-guard
ACC-1(config-if)# loop-protect
```

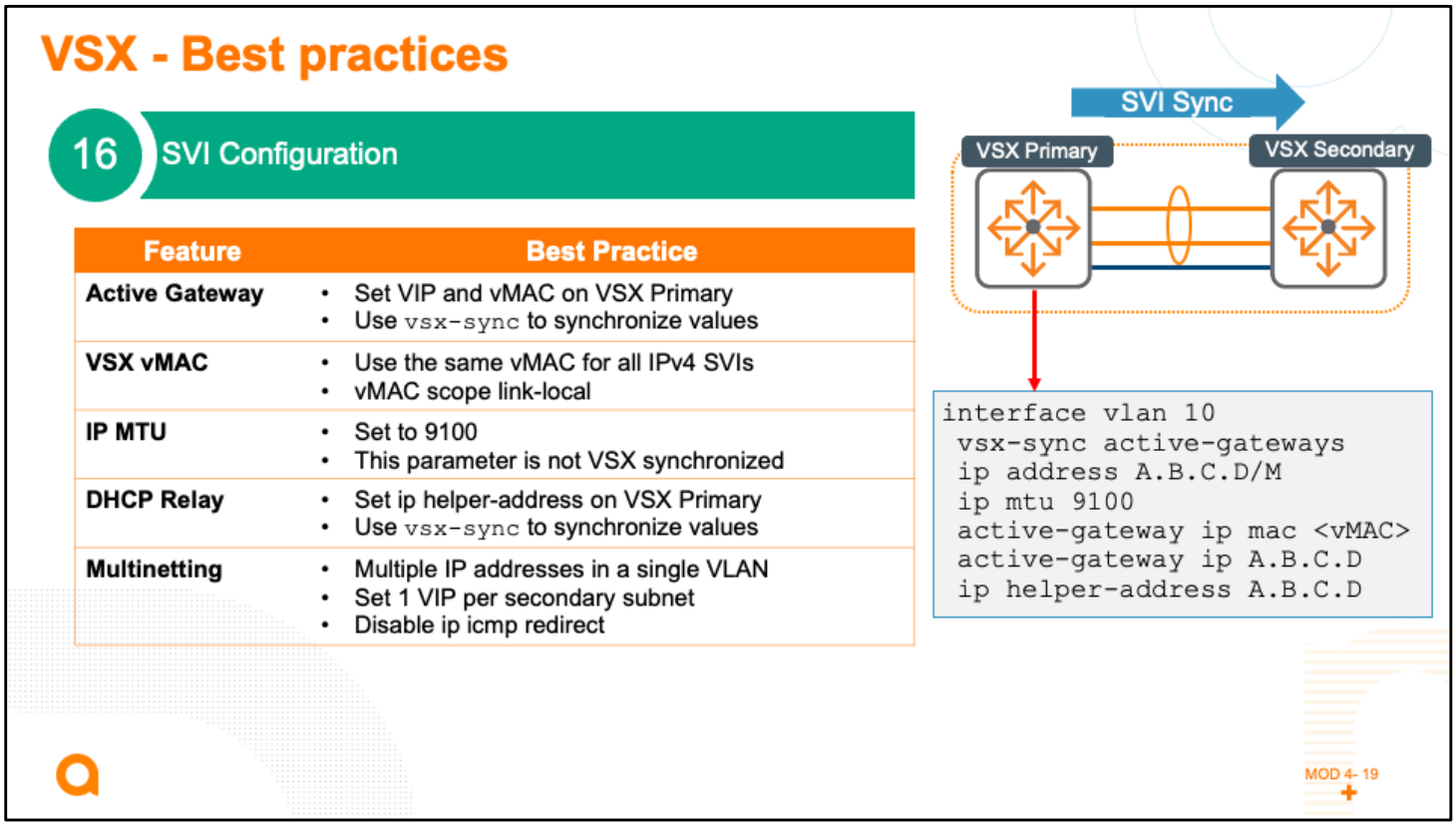


14. VSX LAG ACL Configuration

If an ACL is used, the best practice is to have ACLs synchronized. Any ACL applied on a VSX LAG on the VSX primary will get applied on the VSX secondary as well.

15. VSX LAG QoS Configuration

The marking process should be done as close to the endpoint as possible, in this case the Access Layer is responsible for this task. The aggregation Layer should trust in the marking received, the global configuration qos trust dscp



16. SVI Configuration

SVI active-gateway: Set the active gateway VIP and virtual MAC on the VSX primary node and get the value synchronized using vsx-sync command

Active-gateway virtual MAC: Use the same virtual MAC for all IPv4 SVIs. The scope of this virtual MAC is purely link-local. If some servers or systems have dual-attachment to two different SVIs and the administrator would like to see distinct MAC addresses for the next-hops over these separate interfaces, then 16 virtual MAC are available.

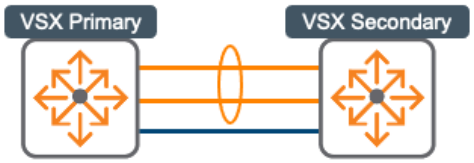
– As a best practice use only 8 virtual MAC for IPv4 and 8 different for IPv6.

Multinetting is the assignment of more than one IP address to a single VLAN. Use this to enable a router to provide default gateway service to different address ranges associated with a single VLAN. If you are using Multinetting set one VIP per secondary subnet and disable ip icmp redirect.

IP MTU: Configure all SVIs to match the size of the L2 MTU: IP MTU recommended value = 9100. This parameter must be identical and manually set on both VSX nodes.

DHCP relay: Configure the ip helper-address on the VSX primary and let vsx-sync configuring the same on the VSX secondary.

Question #2




```
hostname AGG1
interface vlan 11
 ip address 10.1.11.2
 active-gateway ip mac 05:00:00:00:00:11
 active-gateway ip 10.1.11.1
```

```
hostname AGG2
interface vlan 11
 ip address 10.1.11.3
 active-gateway ip mac 05:00:00:00:00:11
 active-gateway ip 10.1.11.1
```

Is the active-gateway MAC address used in the example a valid one?

a. True

b. False

Knowledge Check

Correct answer:

B

The private MAC address should be one of the following options:

X2:XX:XX:XX:XX:XX

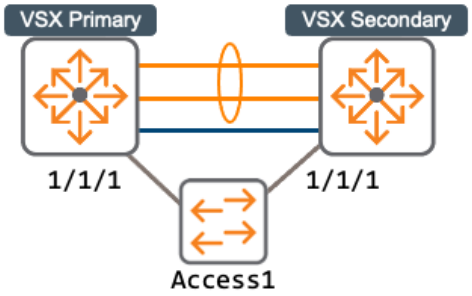
X6:XX:XX:XX:XX:XX

XA:XX:XX:XX:XX:XX

XE:XX:XX:XX:XX:XX

Question #3

```
hostname AGG1 and AGG2
interface lag 1
 no shutdown
 vlan trunk allow 10,20-0
interface 1/1/1
 no shutdown
 lag 1
```



The Link Aggregation between the VSX cluster and Access1 is not functioning, what is causing the problem?

a. The Link Aggregation ID 1 cannot be used in VSX
b. AOS-CX 10.04 does not support Multichassis LAG
c. You must use `multichassis` key word in LAG1 configuration

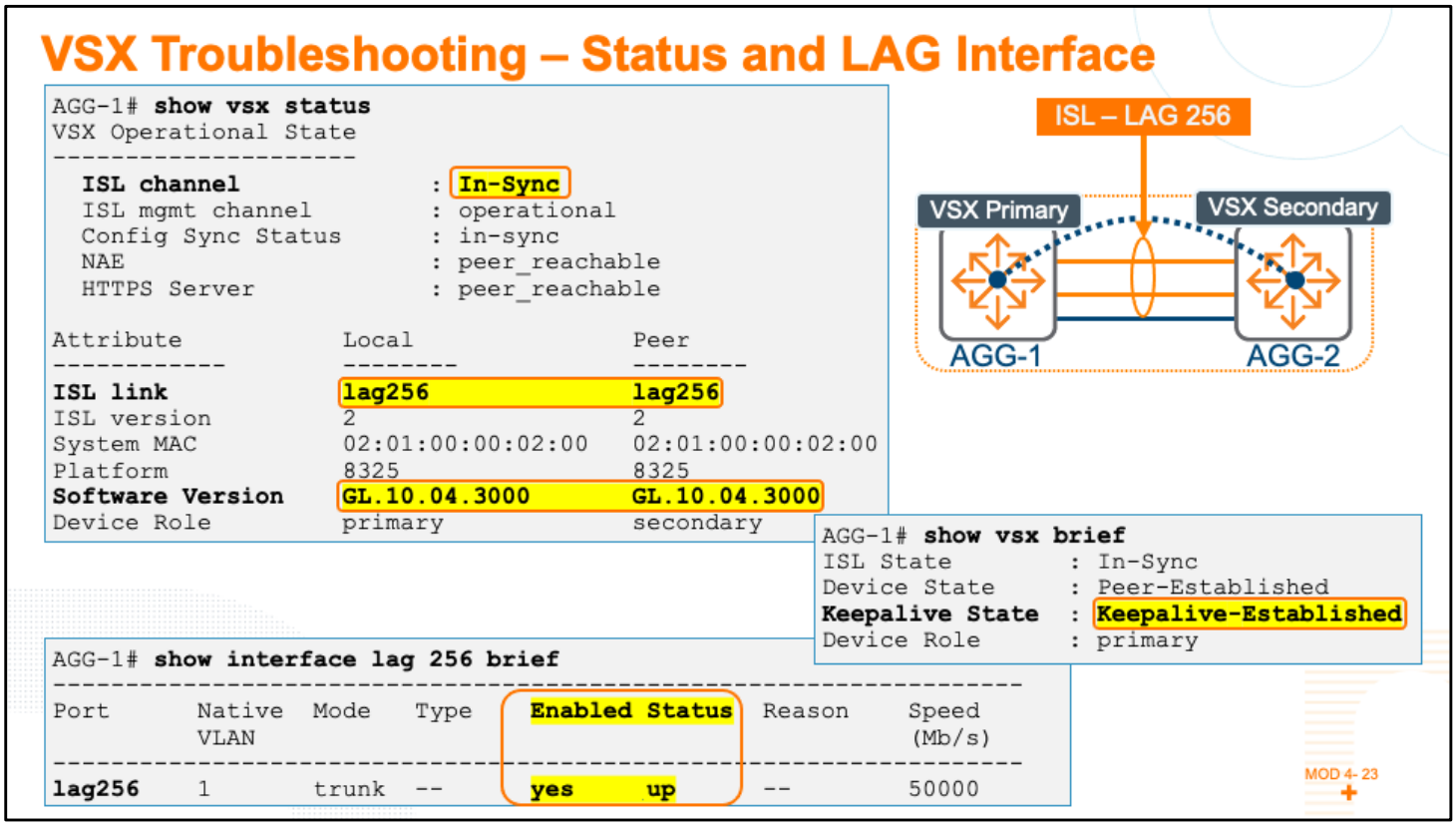
Knowledge Check

Correct answer:
C

VSX Troubleshooting

Show and debug commands

aruba
a Hewlett Packard Enterprise company

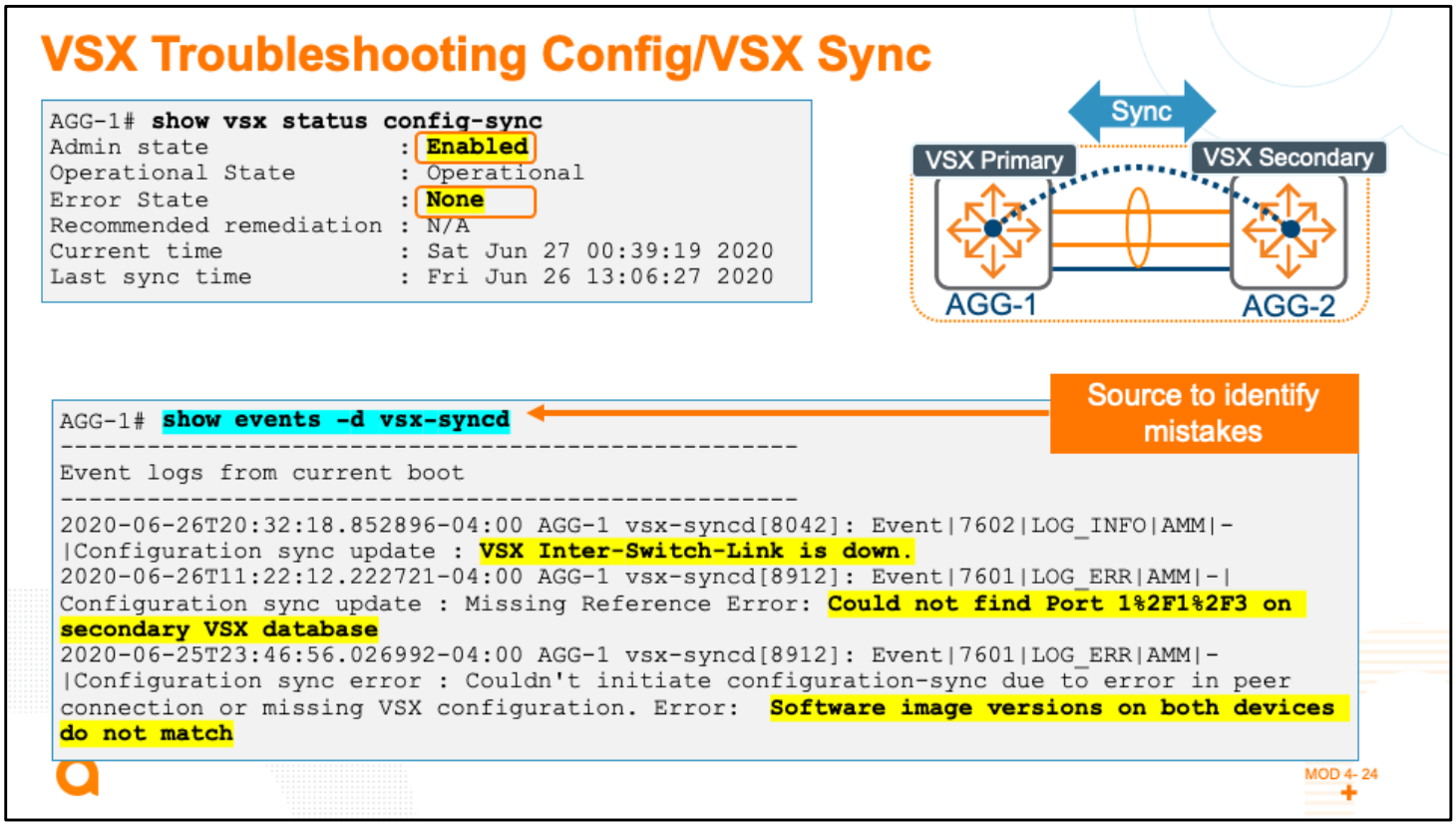


Use the following comands to verify the status of VSX:

show vsx status command displays important information about VSX, for instance you must make sure that both VSX nodes run the same version of AOS-CX code to avoid unexpected behavior. Both VSX nodes should be able to run ISLP between them and synchronize configuration, and the ISL channel should display In-sync.

show vsx brief command displays information about the Keepalive status.

It is always good idea to verify the status of the physical or logical port that is been used as ISL Link. This example highlights output for the show interface lag 256 brief command to verify the state of this interface.



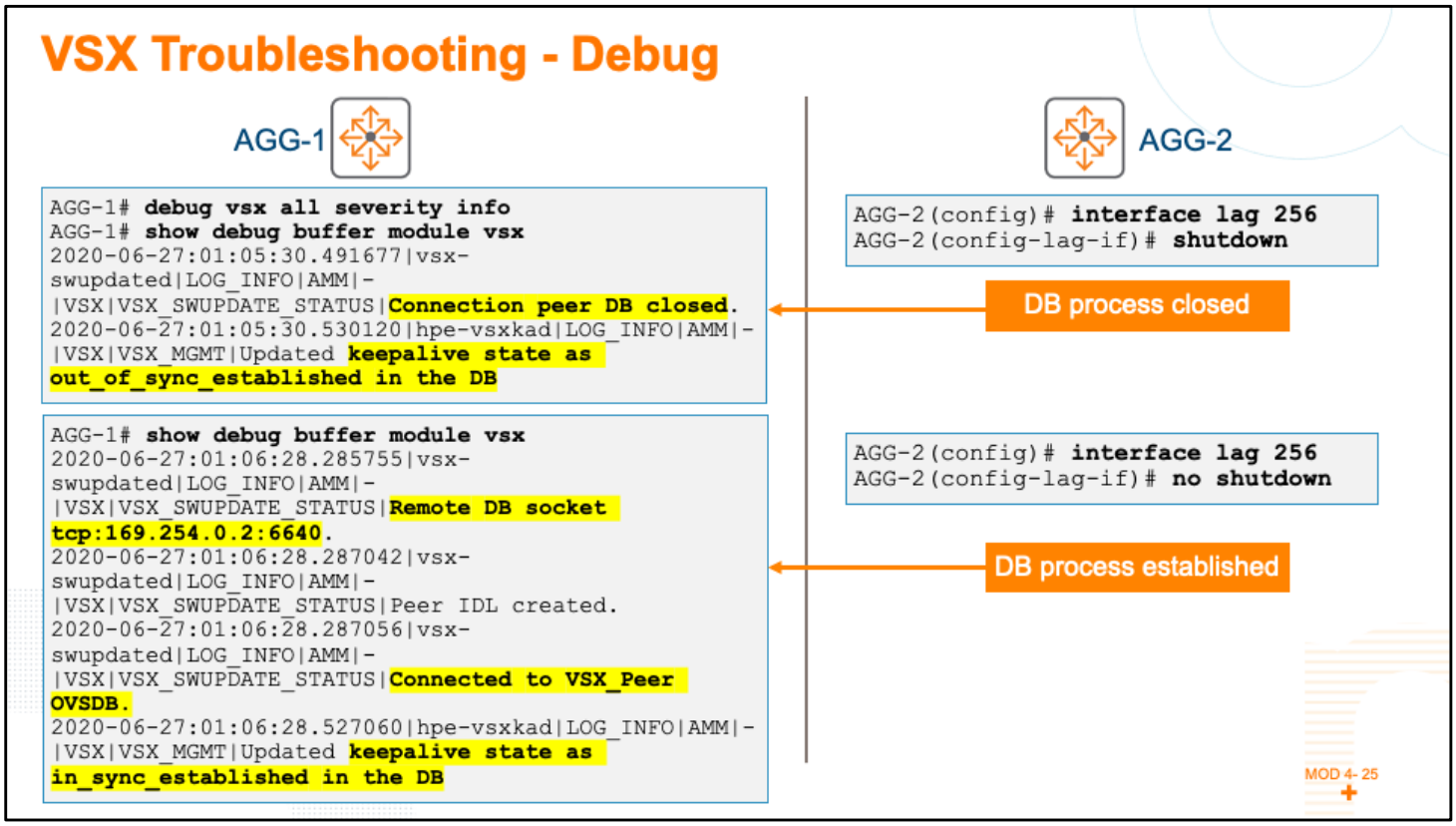
When using Aruba VSX stacking technology, it is important to maintain synchronization for the common configuration required by both VSX nodes. Verify this with the show vsx status config-sync command. Validate that there are no Error States, and the Administrative state should be enabled.

You can verify VSX-generated logs with the command show events –d vsx-syncd. The –d flag displays information from a specific daemon. This example highlights three log message types:

This informational message indicates that ISL link is down. This event has implications on the synchronization process, since new configuration will not going to occur.

The second message indicates that there is a database error, related to synchronization.

The third event refers to an Error, caused by the VSX peers running different Software images. Thus, configuration-sync cannot be initiated.



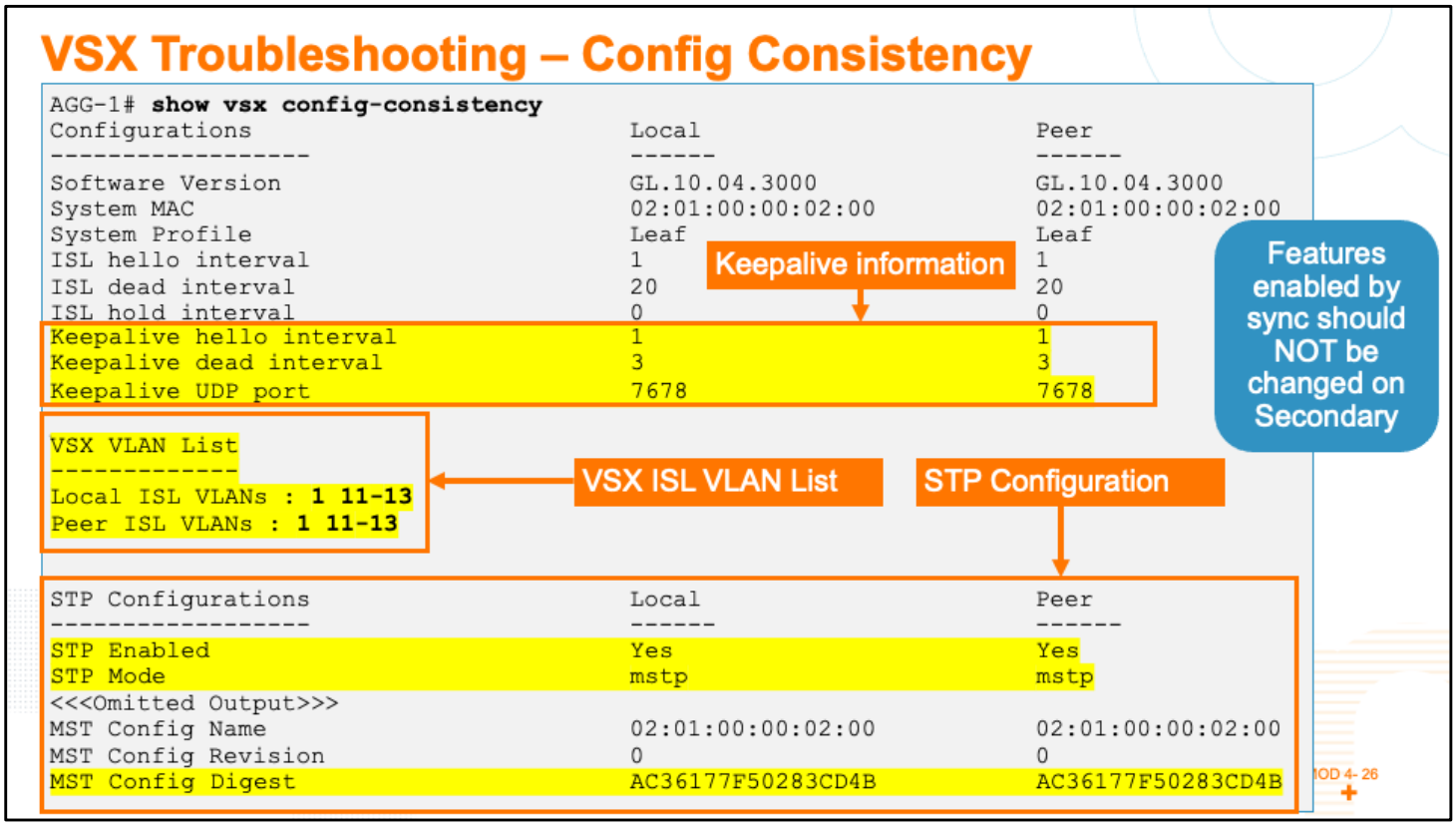
Use debug commands to understand backend VSX processes. For AOS-CX, you must first enable the debugging process for VSX and then you can display the information. Use the following commands:

Switch# debug vsx all severity info

Switch# show debug buffer module vsx

In the top example, the LAG interface on AGG-2 is shutdown, which triggers some events in AGG-1. The debug command shows that the database for VSX is closed and the state for keepalive is out_of_sync_enablished. This means that this VSX node cannot reach its peer Database.

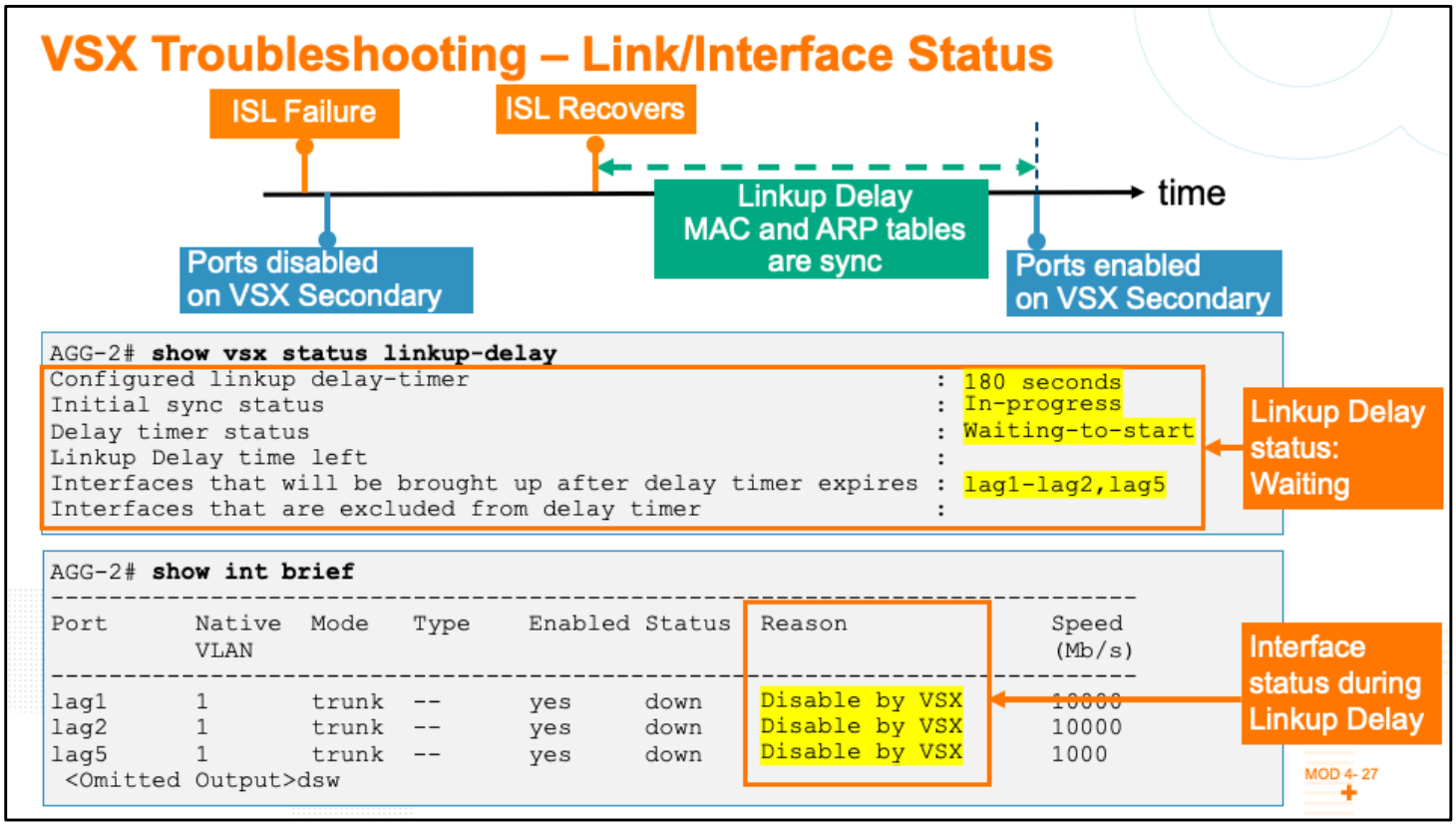
Then the AGG-2 LAG interface is re-enabled, triggering events in AGG-1 - to recover the VSX status. First, a TCP session on port 6640 is established, used to synchronize their database. Then both devices connect to their peer database (OVSDB). Finally, the keepalive state moves to a state of in_sync_established.



Use the show vsx config-consistency command to see information about the Keepalive session that is established between both VSX peers. If you are using a different UDP port to established the Keepalive connection, you will see it here.

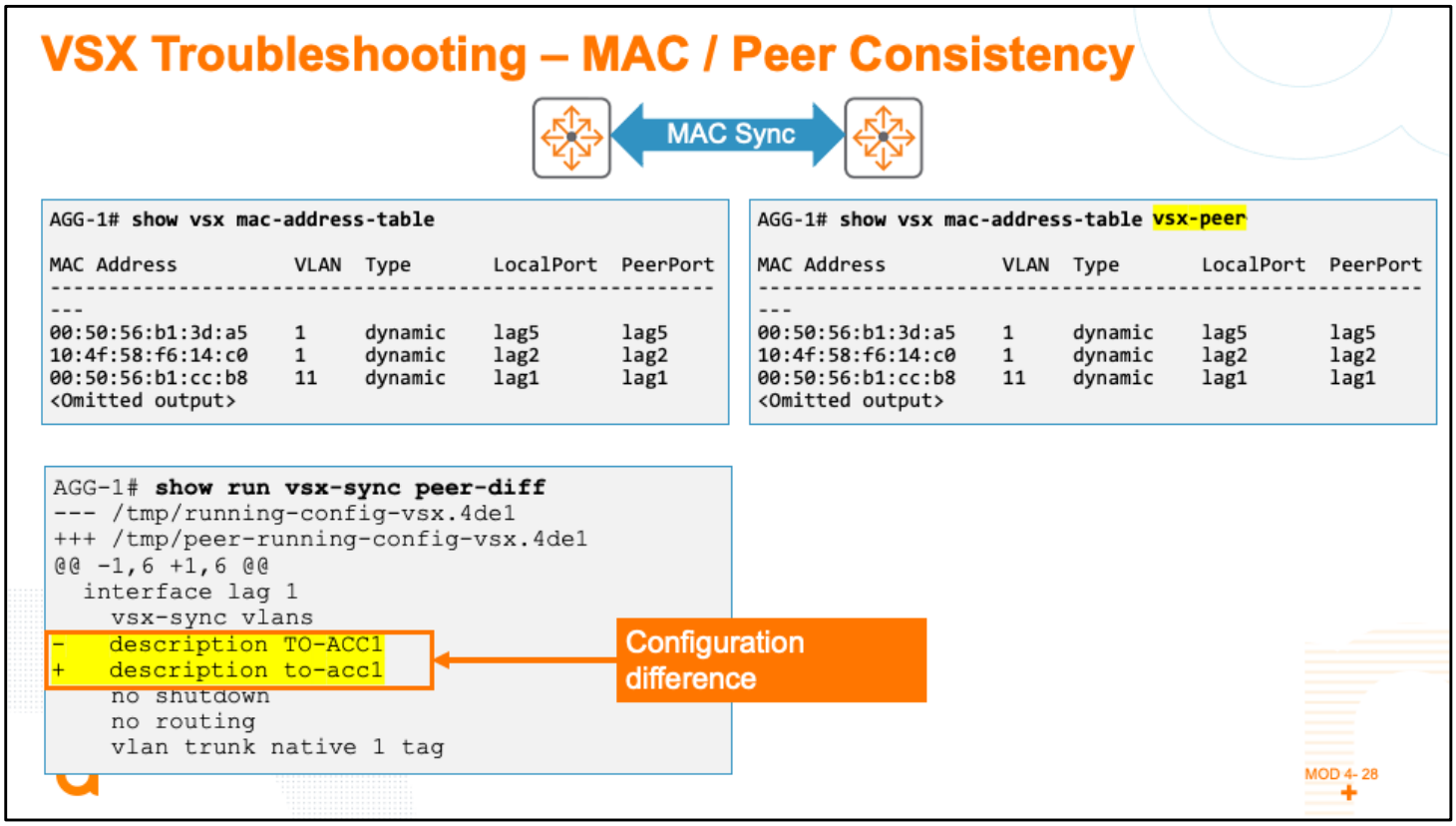
If VSX uses an ISL link to carry traffic, you must permit the appropriate VLANs on this link, which must be consistent on both VSX nodes. You can validate this, as shown in the figure.

Finally, recall that a best practice is to use MSTP to avoid Layer 2 loops, which can also be verified here.



Assume that you properly setup the Keepalive feature. Thus, during an ISL failure, all ports on the VSX Secondary are disabled. The linkup-delay feature helps VSX to avoid any unexpected behavior when the ISL link has been restablished. During this period both VSX nodes syncrhonize MAC and ARP tables and the ports on VSX Secondary remain in a disabled state. By default the delay-timer is set to 180 seconds.

The show vsx status linkup-delay command helps to verify the state of the ports and if they will be brought up after the timer expires. In the example, notice that ports LAG1, 2, and 5 are waiting for the timer to expire.



Verify mac-address synchronization with the show vsx mac-address-table vsx-peer command. Differences can be seen with show run vsx-sync peer-diff command.

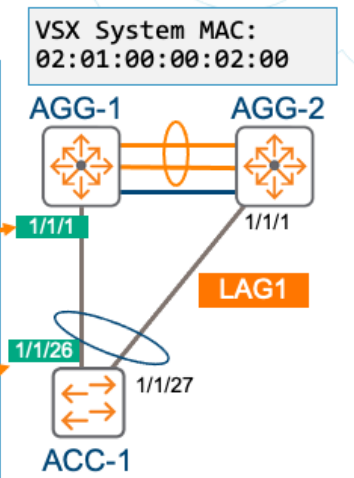
VSX Troubleshooting – LACP Interfaces

```
AGG-1# show lacp interfaces multi-chassis

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Actor details of all interfaces:
-------------------------------------------------------------------------------
Intf    Aggregate  Port    Port     State   System-ID          System   Aggr
        name       id      Priority                            Priority Key
-------------------------------------------------------------------------------
1/1/1   lag1(mc)   1       1        ALFNCD  02:01:00:00:02:00  65534    1
1/1/2   lag2(mc)   2       1        ALFNCD  02:01:00:00:02:00  65534    2
1/1/5   lag5(mc)   5       1        ALFNCD  02:01:00:00:02:00  65534    5


Partner details of all interfaces:
-------------------------------------------------------------------------------
Intf    Aggregate  Partner Port     State   System-ID          System   Aggr
        name       Port-id Priority                            Priority Key
-------------------------------------------------------------------------------
1/1/1   lag1(mc)   26      1        ALFNCD  10:4f:58:f6:01:40  65534    255
1/1/2   lag2(mc)   26      1        ALFNCD  10:4f:58:f6:14:c0  65534    255
1/1/5   lag5(mc)   2       255      ALFNCD  20:4c:03:58:72:ca  32768    2
```



Continue to next slide

MOD 4- 29

Use the show lacp interfaces multi-chassis command to verify the state of these interfaces.

The output first displays the information about the Actor - the VSX node where you issued the command. In this case AGG-1 is the VSX Primary. The first line shows important information about LAG1. Notice that the Port ID is set to 1 – the VSX Primary always uses the first number. The system ID has been manually set to 02:01:00:00:02:00, to conform to best practice.

The next section displays information about the partner, in this case the ACC-1 switch.

VSX Troubleshooting – LACP Interfaces

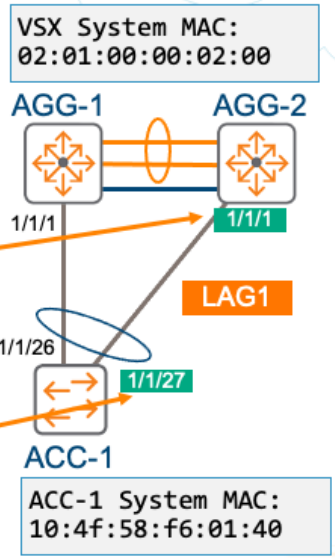
Output continued from previous slide

```
AGG-1# show lacp interfaces multi-chassis

State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state

Remote Actor details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggregate  Port     Port      State   System-ID          System   Aggr
       name       id       Priority                          Priority Key
------------------------------------------------------------------------------
1/1/1  lag1(mc)   1001     1         ALFNCD  02:01:00:00:02:00  65534    1
1/1/2  lag2(mc)   1002     1         ALFNCD  02:01:00:00:02:00  65534    2
1/1/5  lag5(mc)   1005     1         ALFNCD  02:01:00:00:02:00  65534    5


Remote Partner details of all interfaces:
------------------------------------------------------------------------------
Intf   Aggregate  Partner Port      State   System-ID          System   Aggr
       name       Port-id Priority                          Priority Key
------------------------------------------------------------------------------
1/1/1  lag1(mc)   27       1         ALFNCD  10:4f:58:f6:01:40  65534    255
1/1/2  lag2(mc)   27       1         ALFNCD  10:4f:58:f6:14:c0  65534    255
1/1/5  lag5(mc)   3        255       ALFNCD  20:4c:03:58:72:ca  32768    2
```



MOD 4- 30

The next section displays information about the Remote Actor - the VSX-peer node AGG-2. Notice that the port ID for VSX Secondary is always “1000 + id” and the System ID is the same one used for the VSX-peer.

The last section displays information about the Remote Partner from the VSX-peer perspective – switch ACC-1.

VSX Troubleshooting – Config Parser

```
AGG-2# vsx-configmate
==========================================
       VSX FEATURE CONFIG PARSER
==========================================

VSX configuration validation results below:-

Rule-1: The allowed VLAN list for ISL port must match that in config for MCLAG in AGG-2 (Passed)
Rule-1: The allowed VLAN list for ISL port must match that in config for MCLAG in AGG-1 (Passed)
Rule-2: The native VLAN must also be part of allowed VLAN list for a MCLAG in AGG-2 (Passed)
Rule-2: The native VLAN must also be part of allowed VLAN list for a MCLAG in AGG-1 (Passed)
Rule-3: The inter-switch-link Timers should be consistent across VSX peers (Passed)
Rule-4: The Keep-Alive Timers should be consistent across VSX peers (Passed)
Rule-5: Active-Gateway configurations(IP, MAC and VRF) should be consistent across the VSX peers (Passed)
Rule-6: The Keep-Alive UDP port should be consistent across VSX peers (Passed)
Rule-7: The link-up-delay-timer should be consistent across VSX peers (Passed)
Rule-8: The VSX System-MAC should be consistent across VSX peers (Passed)
Rule-9: The VSX device roles should be consistent across VSX peers (Passed)
Rule-10: VSX active-forwarding configurations should be consistent across the VSX peers (Passed)
```



The vsx-configmate command helps to identify possible problems when deploying VSX. A normal output will show Passed on all Rules. Any inconsistencies will show as Skipped.

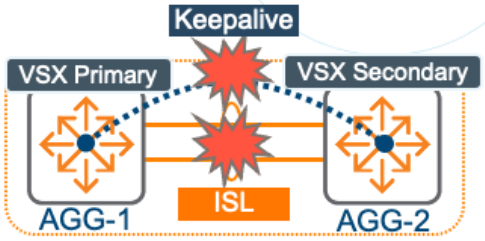
This command is available started in 10.04.3000 switches.


aruba
a Hewlett Packard
Enterprise company
VSX
Split Brain

VSX – Split Brain

Keepalive	ISL	Result
Established	In-Sync	• Forwarding is OK • Protection is OK
Established	Out-of-sync	• ISL is down • Peer is detected and UP • VSX Secondary stops forwarding over its interfaces • VSX tables are out of sync
Not Established	In-Sync	• Forwarding is OK • No Protection (live dangerous)

Not Established	Out-of-Sync	• SPLIT BRAIN
		  • ISL is down
		  • Peer is seen DOWN
		• Both VSX nodes keep forwarding
		• Unexpected behavior and traffic loss



MOD 4- 33

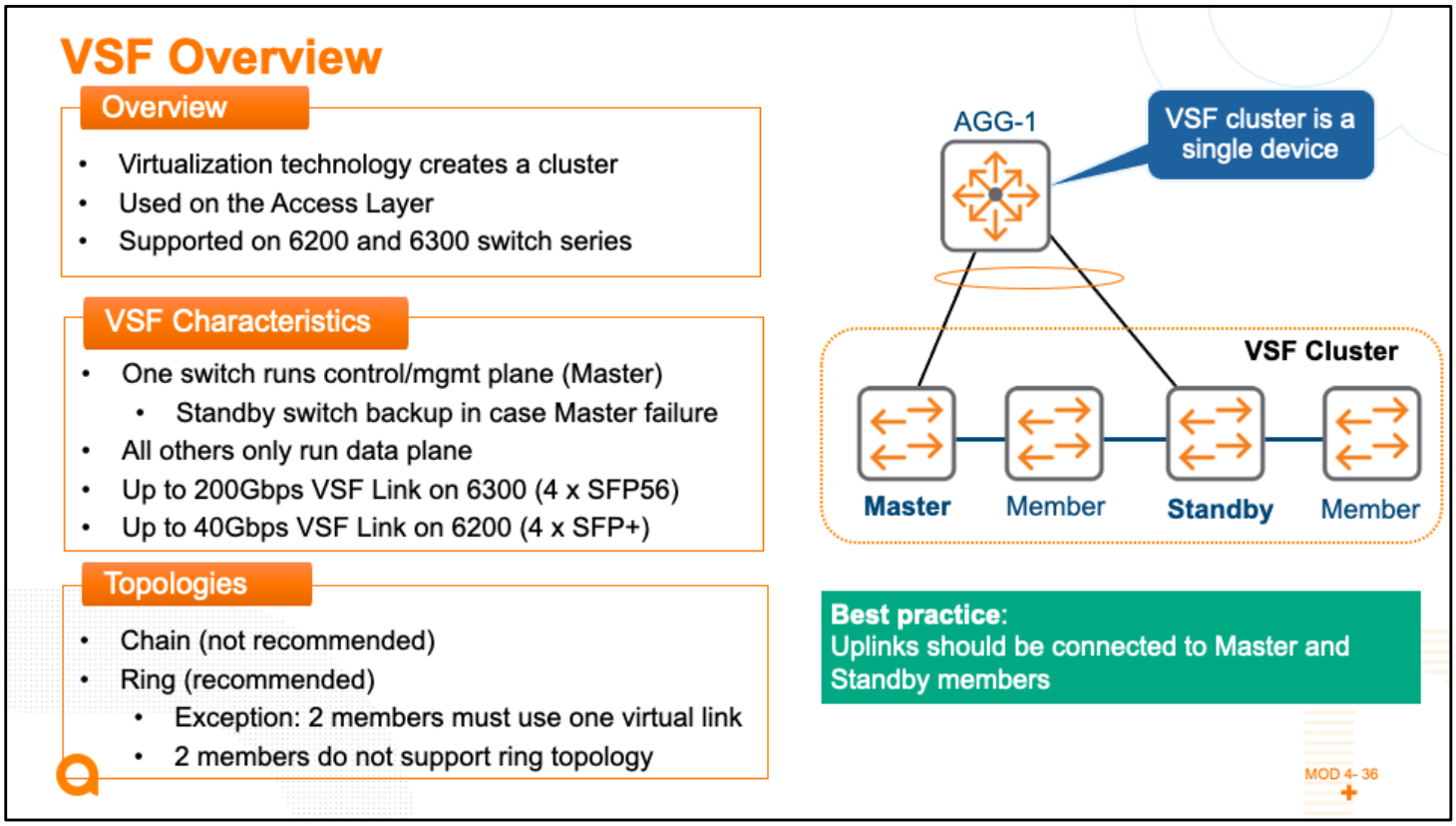
The diagram shows a situation in which the ISL between the two VSX switches is down. Both switches are still active, but the switches cannot exchange information. This causes the switches to be out of sync. This split-brain condition occurs if you fail to enable the keepalive function. This keepalive function prevents split brain issues by providing an out-of-band mechanism to detect if the VSX Peer-to-Peer link is down or if the VSX switches are not working. The keepalive functionality brings down the link between Agg2 and Access Switches. The traffic is forced to go from access switches to Agg1 only.

The table shows what happens with various ISL and keepalive statuses on the VSX pair. If the ISL is up and the peers are in sync, the VSX pair can forward traffic normally. If the keepalive connection is established, the system is protected from a split-brain situation. If the keepalive connection is not established, but the ISL is working correctly, the VSX pair can forward traffic normally, but the system is at risk. If the ISL goes down and the peers are out of sync, but the keepalive is established, the peers realize that the other peer is still up. They know that they are at risk of becoming out of sync, so the secondary switch shuts down its VSX LAGs. If the ISL goes down and the keepalive connection cannot establish, the switch assumes that its peer is down, and that the system is functional but forwarding at 50 percent capacity. It keeps forwarding even if it is the secondary peer. It is important that you form the keepalive connection correctly so that this assumption is correct.

aruba
a Hewlett Packard Enterprise company

Additional Content


aruba
a Hewlett Packard
Enterprise company
VSF
Overview, Requirements and Operation



Aruba VSF is a virtualization technology used to create a cluster or stack. This technology is commonly deployed on the Access Layer and it is supported on 6200 and 6300 switch series. In a VSF cluster, the VSF Master switch runs the control and the management plane for the entire cluster. The rest of the switches in the cluster simply run the Data Plane.

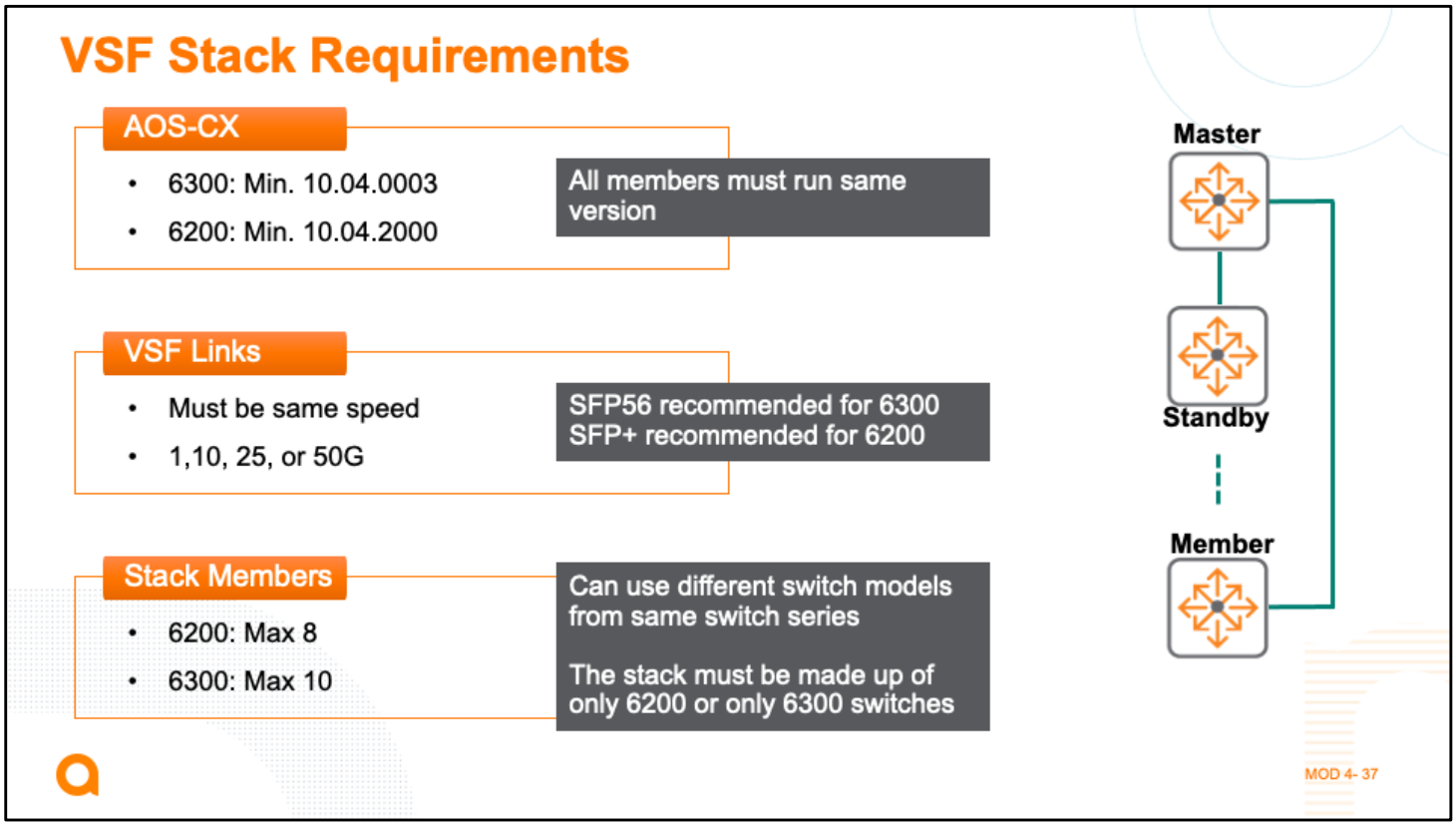
Inter-switch communication within the cluster is via a special VSF Link. You can only use the 4 SFP+/SFP56 ports to create this link. Using 50Gbps connections, you can get up to 200Gbps on the 6300-switch series. The 6200-switch series support up to 4 SFP+, you can get up to 40Gbps on this devices.

VSF supports two topologies - Chain and Ring. Ring is the recommended option, due to superior link failure tolerance.

Note. Neither a 2-member ring nor a full-mesh topology is supported. In other words, a 2-member stack must be configured with one virtual link. Also note that the 6200 does not support SFP56/SFP28, and so therefore does not support 25/50Gbps links.

Note. A virtual Link between two VSF members can be composed by 2 physical links, this allows the solution to be link down tolerant. It is important to clarify that the usage of these two links is not by using Link Aggregation or LACP, Aruba VSF uses proprietary algorithms for this decision.

For resiliency purposes as a best practice Uplinks should be connected to thee Master and Standby devices.



VSF has some requirements that you must meet:

AOS-CX

You must use 10.04.0003 or higher for 6300s switch series, and 10.04.2000 or higher for 6200s switch series. All members in the cluster should run the same AOS-CX version.

VSF links

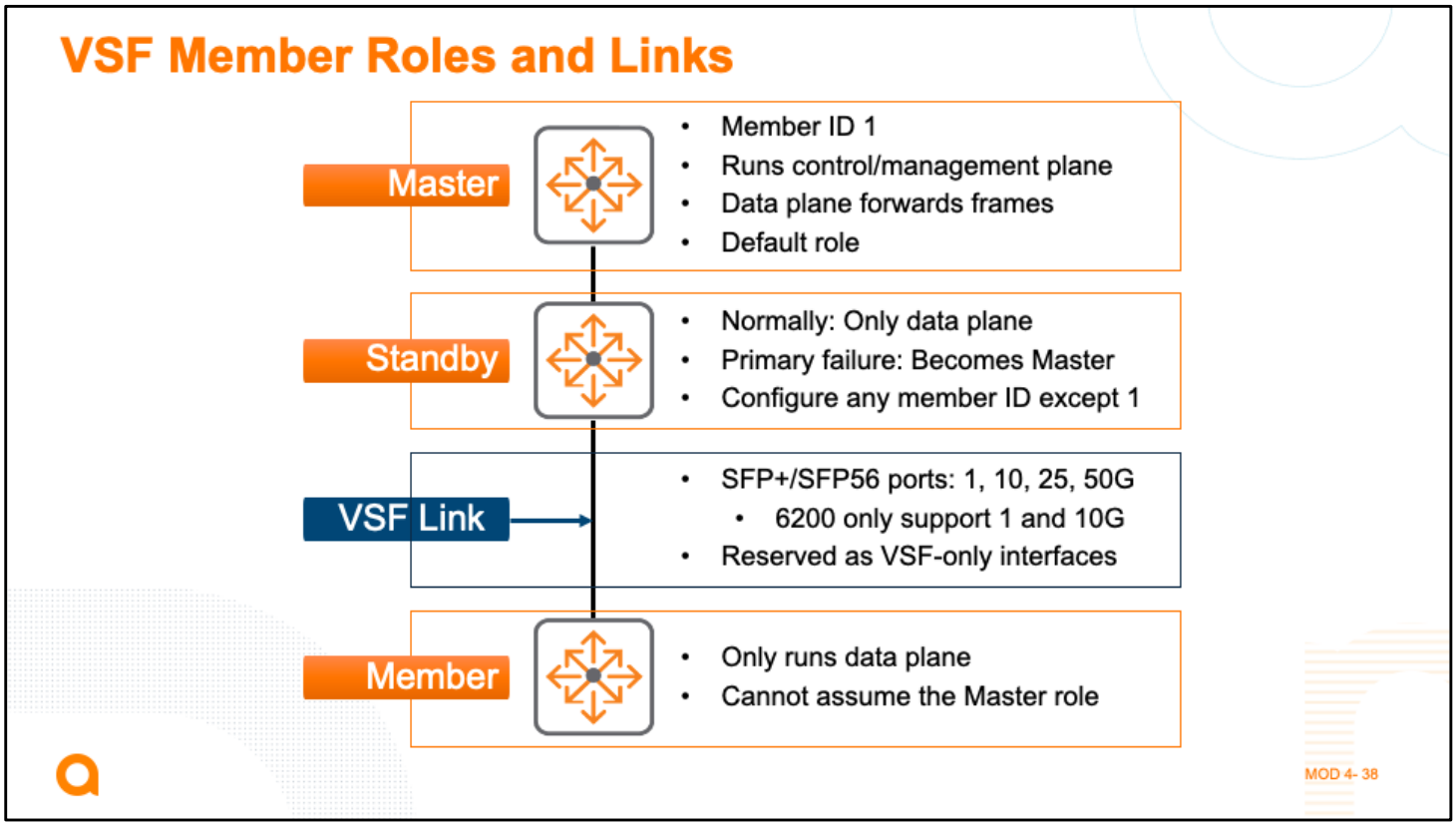
All VSF link ports in a stack must operate at the same speed (1, 10, 25, or 50G). It is a best practice to use Aruba SFP56 when possible and avoid using SFP that runs at 1Gbps as much as possible.

Also, VSF links need not have the same number of ports. For example, you can have 2 SFP+ ports between members 1 and 2, with only one SFP+ port between members 2 and 3.

Stack Members

You can create a VSF stack using different switch models from the same series. The 6300-series supports up to 10 members, while the 6200 series supports up to 8 members.

IMPORTANT: VSF stacking cannot be done with a mixed set of switches. The stack must be made up of only 6200 or only 6300 switches.



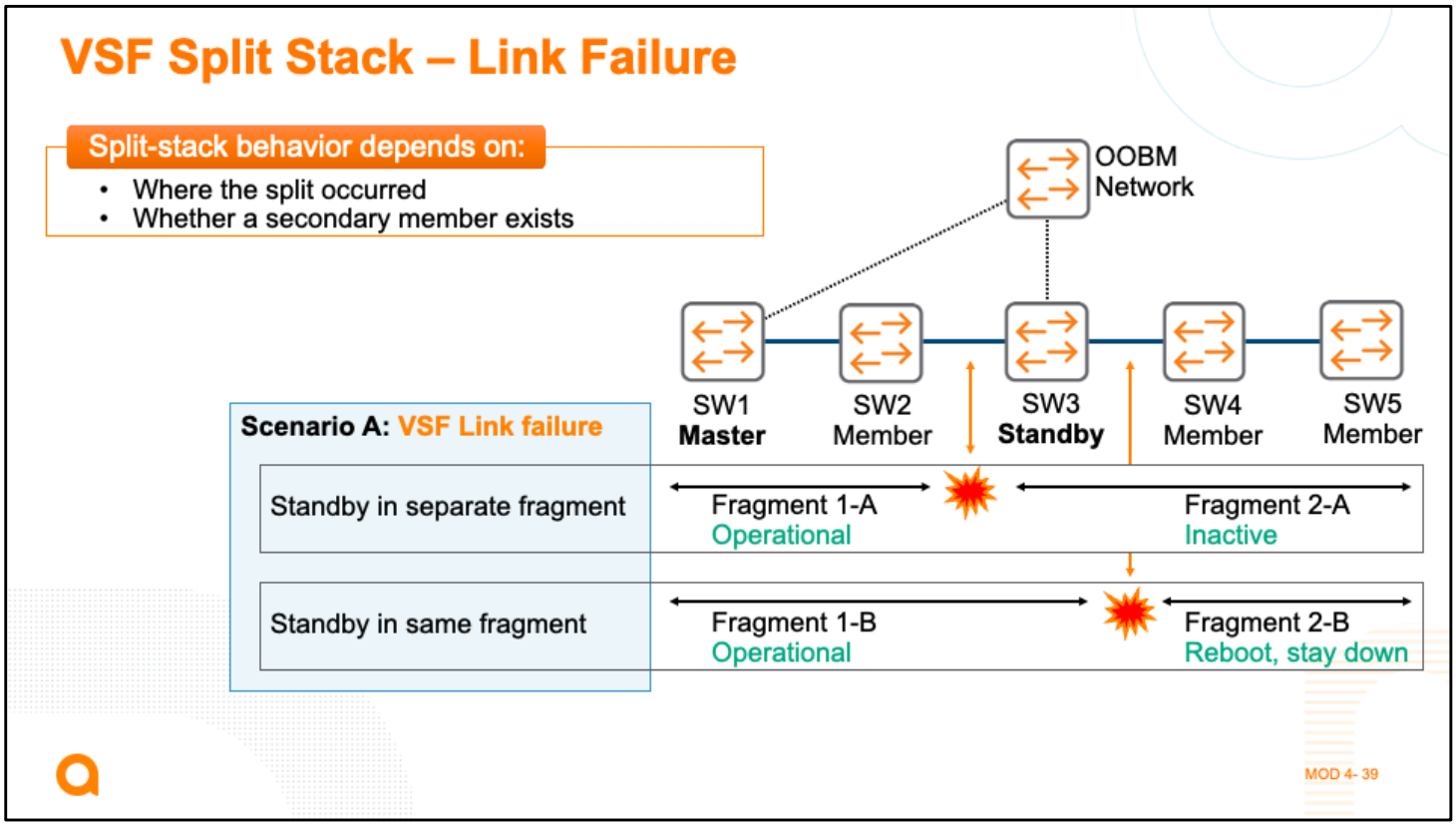
Aruba VSF creates a single control plane that runs on a single VSF member, which always uses member ID 1. This device assumes the Master role in the stack.

Note. A factory-default Aruba 6300 switch boots up as VSF enabled switch with member ID 1. This implies that the switch behaves as the Primary member.

A Standby member provides for high availability in case Master failure. You can choose any member in the stack to take the Standby role, but it must be explicitly defined. Just configure any member ID except 1.

The other devices in the VSF are Members, which only run the data plane, and cannot assume the Master role.

VSF switches are interconnected using the four front-panel SFP+/SFP56 ports, which operate at 1, 10, 25, or 50 Gbps speeds. When you configure a port for VSF it can no longer be used as a Layer 2 or Layer 3 interface. In other words, the port does not belong to the switch's Data plane. Notice that 6200 switches only use 1 or 10G ports.

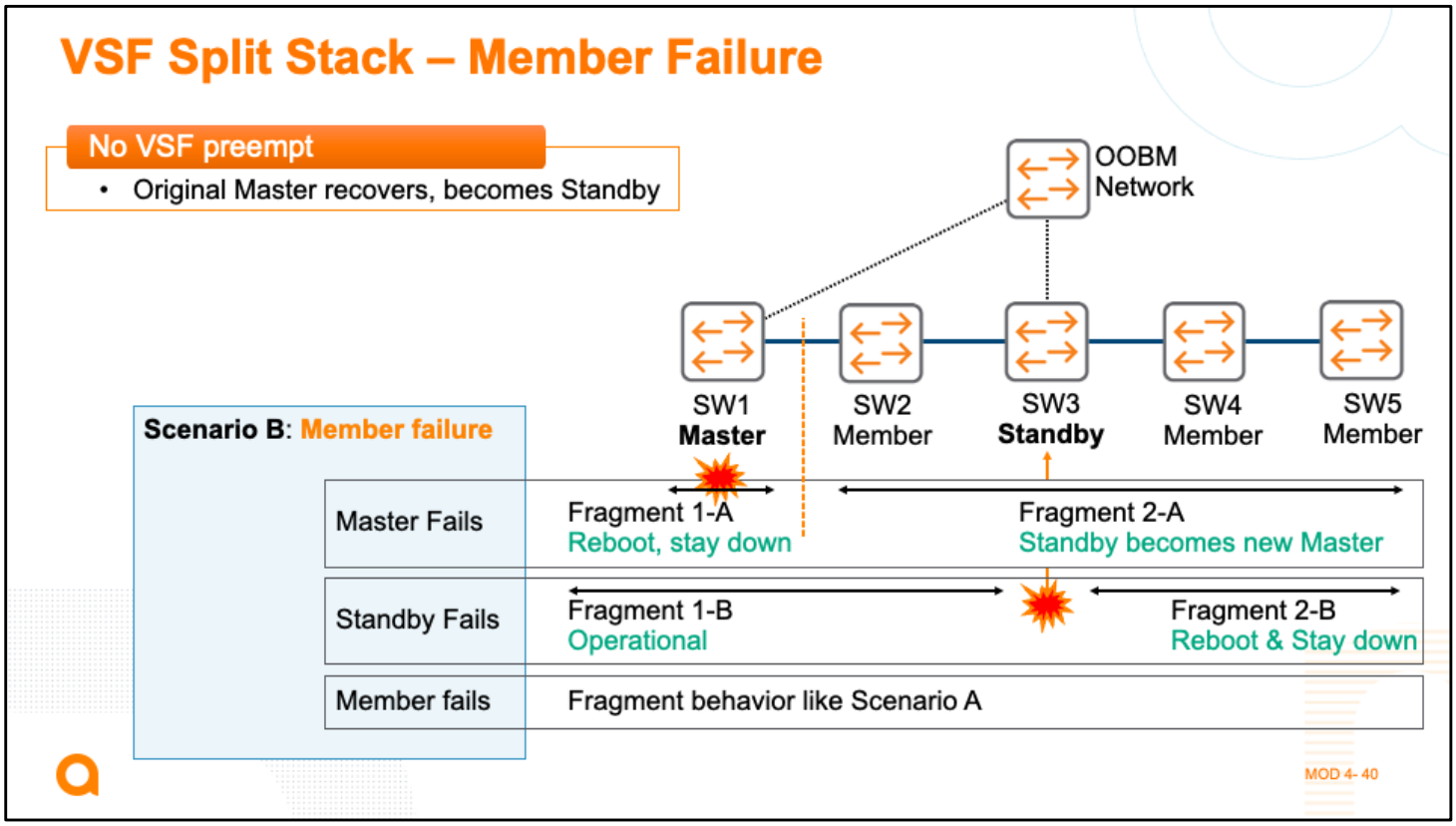


A VSF stack may experience a split stack condition due to the failure of a VSF link or stack member. When a split occurs, the behavior of each resulting stack fragment depends on where the split occurred, whether a secondary member has been designated for the stack, and whether split detection has been enabled.

The behaviors described in the following scenarios presume that VSF split detection has been configured and is functional, and a secondary member has been designated and is operating in the VSF-Standby role.

Scenario A — If a VSF link fails and causes a split, but all stack members remain operational, the following occurs:

The fragment containing the Master remains active.
If the standby is in a separate fragment from the primary, it assumes the VSF-Master role for the fragment, and that fragment is designated inactive (all links except the OOBM port and VSF link ports are disabled)
If the standby is in the same fragment as the primary, all members in the other fragment reboot and stay down until a connection to a VSF-Master is re-established.

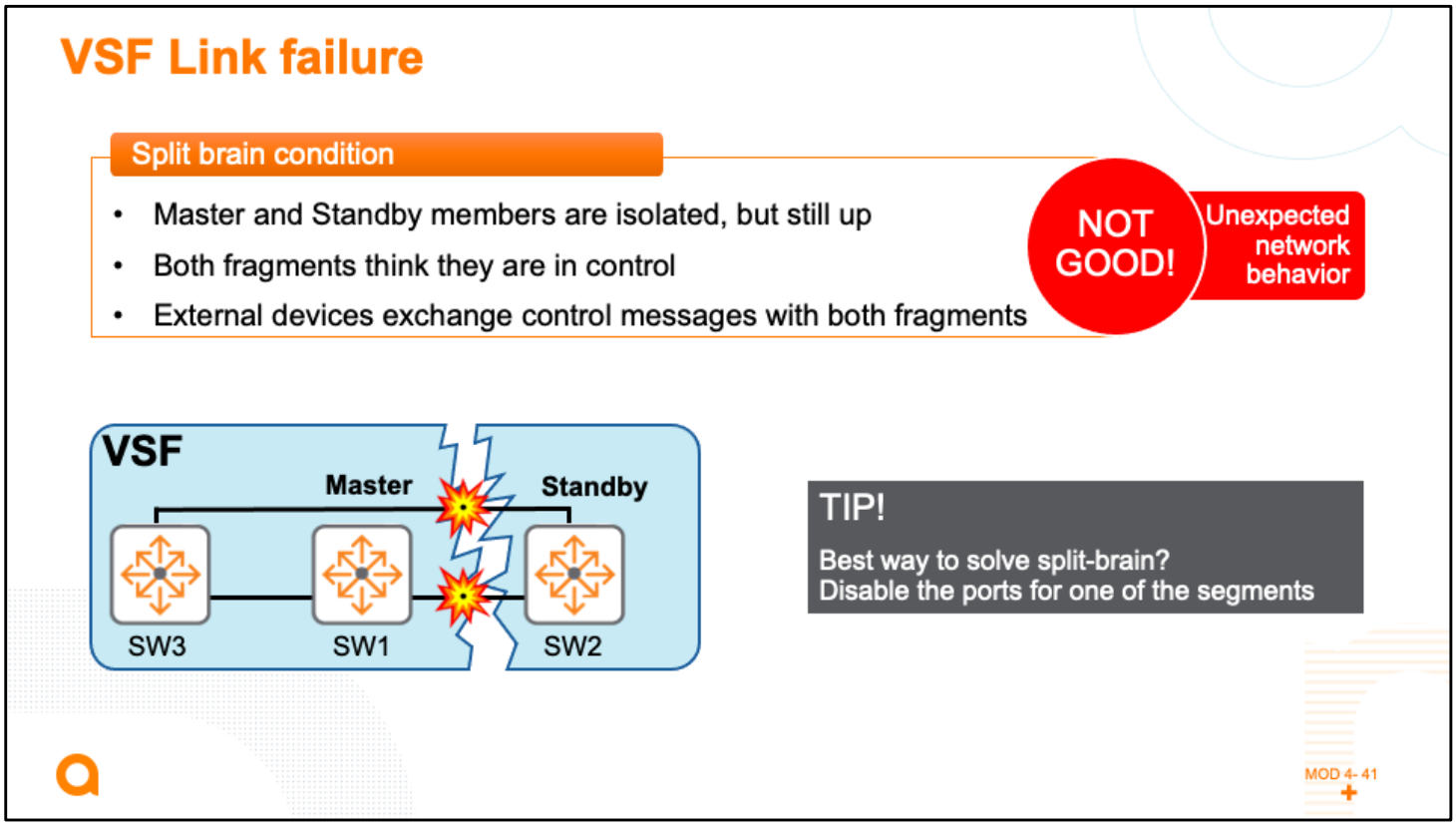


Scenario B — If a VSF member fails and causes a split, the following occurs:

• If the failed member was the Master, the standby assumes the VSF-Master role for its fragment and that fragment remains active; all members in the other fragment reboot and stay down until a connection to a VSF- Master (either Master or Standby) is re-established. When the Master recovers, it will assume the VSF-Standby role until a subsequent failover occurs, and all other stack members in the isolated fragment will recover and rejoin the stack.

• If the failed member was the Standby, the fragment containing the Master remains active. All members in the other fragment reboot and stay down until they establish a connection to a VSF Master. When the Standby recovers, it will reassume the VSF-Standby role, and all other stack members in the isolated fragment will recover and rejoin the stack.

• If the failed member was neither a Master nor standby, stack fragment behavior is the same as for Scenario A: failed VSF link.



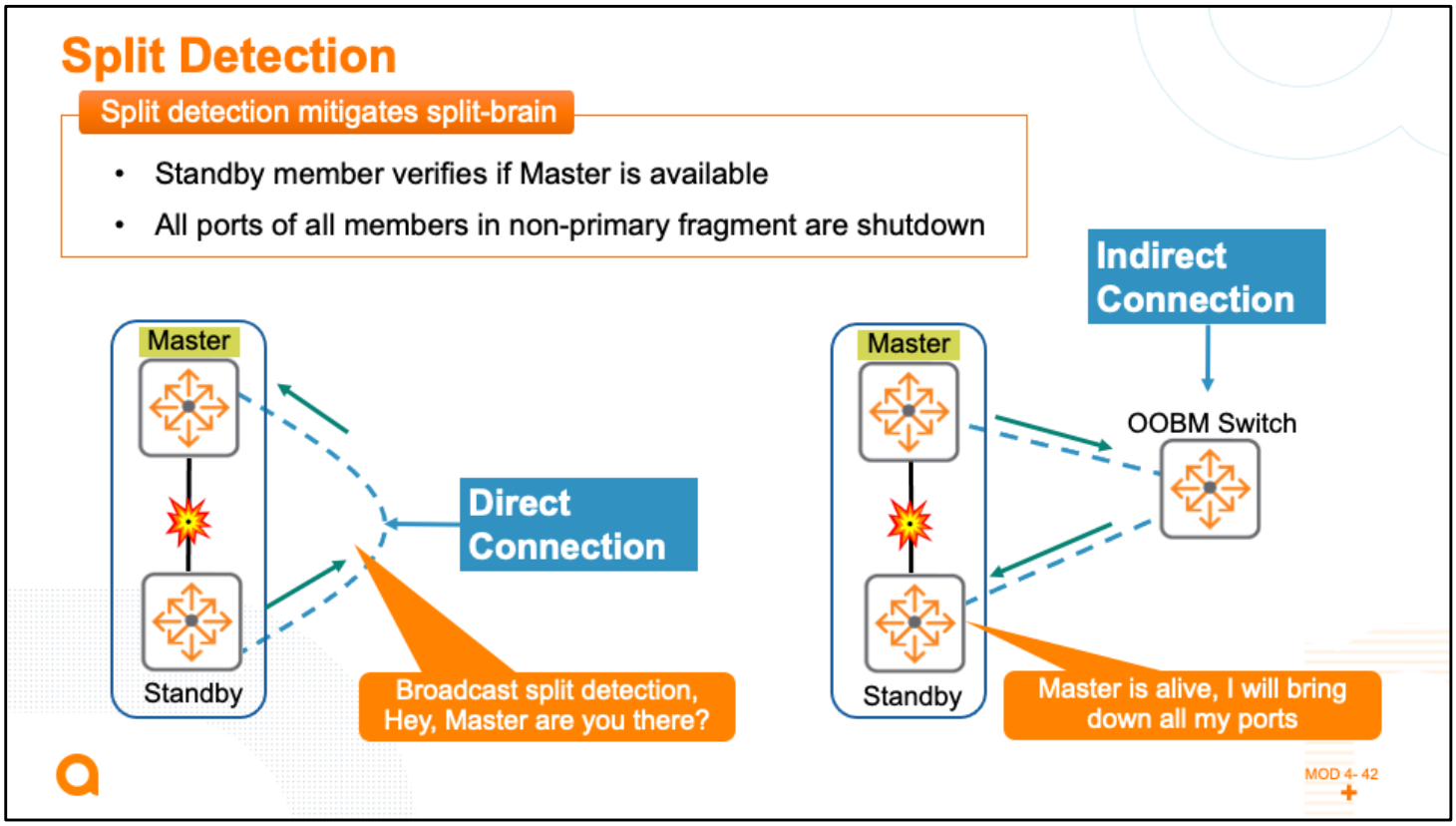
VSF Link failures could cause a fragmented stack.

The figure shows a fragmented VSF stack, due to link failures. Originally, SW1 was Master and SW2 was Standby. After fragmentation, there is no direct connection between the two switches. SW1 continues to think that it is Master. SW2 no longer hears from SW1, and so now thinks that it is the Master. This is known as Split-brain condition - both fragments continue to function although there is no communication between them.

Split-brain can cause unstable, unexpected network behavior. A packet received in one fragment and destined for the other fragment is discarded. Even worse, both segments use the same IP address and the same routing information - external devices could start populating duplicated data. This can cause very strange network behavior.

Note. Split-brain situation could also occur if a VSF member fails in a Chain topology. Therefore a ring topology is preferred over a daisy chain since it eliminates one single point of failure.

The best way to solve a split-brain situation is to disable the ports one of the segments.



You can use Split detection to avoid split-brain situations. Thus, if a VSF link failure occurs, the segment that includes the Standby member verifies Master member status. If the original Primary is up, then all members in the fragment that does not include the Primary member will disable all their ports.

VSF uses the OOBM mechanism to detect and verify the status of the Primary member.

Management interface split detection. This method requires you to connect Out-Of-Band Management (OOBM) interfaces to Master and standby stack members. The connection can be direct using a cable between these two devices or indirect using an external switch, where the OOBM interfaces must be in the same Layer 2 broadcast domain (VLAN). This network is used to identify active stack fragments. Each member broadcast Split Detection Protocol Packets to identify stack fragments that are currently operational. When the standby member detects that Master is alive then automatically it will bring down all its ports. This solution is based on the command vsf split-detect mgmt-interface.

VSF Troubleshooting

Show and debug commands

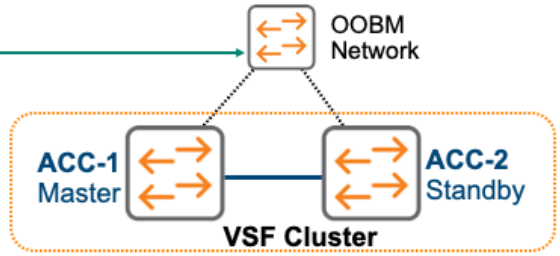
aruba
a Hewlett Packard Enterprise company

VSF Troubleshooting – Status and Members

```
ACC-1# show vsf

MAC Address               : 10:4f:58:f6:01:40
Secondary                 : 2
Topology                  : Chain
Status                    : No Split
Split Detection Method    : mgmt

Mbr Mac Address          type            Status
ID
--- -------------------- --------------- ---------------
1   10:4f:58:f6:01:40    JL666A          Master
2   10:4f:58:f6:14:c0    JL666A          Standby
```




```
ACC-1# show vsf member 2

        MAC Address               : 10:4f:58:f6:14:c0
        Type                      : JL666A
        Model                     : 6300F 24-port 1GbE Class 4 PoE and 4-port SFP56
Switch
        Status                    : Standby
        ROM Version               : FL.01.05.0005
        Serial Number             : SG01KN501G
        Uptime                    : 2 minutes
        CPU Utilization           : 7%
        Memory Utilization        : 9%
        VSF Link 1                : Up, connected to peer member 1, link 1
        VSF Link 2                :
```

MOD 4- 44

show vsf command is the first point of troubleshooting for VSF. It shows:

The number of members that are part of this cluster and its role

The topology that is used is recommended to use Ring (however for lab limitations the output is showing Chain)

The current status No Split - means all VSF links and members work properly

The Split detection method, in this case is by using the OOBM network.

Finally the secondary member, is recommended to set up this role to any member that is not the Master, this increases the reliability of the network.

show vsf member <member-id>, displays detail information about the member selected.

VSF Troubleshooting – Links and Topology

```
ACC-1# show vsf link

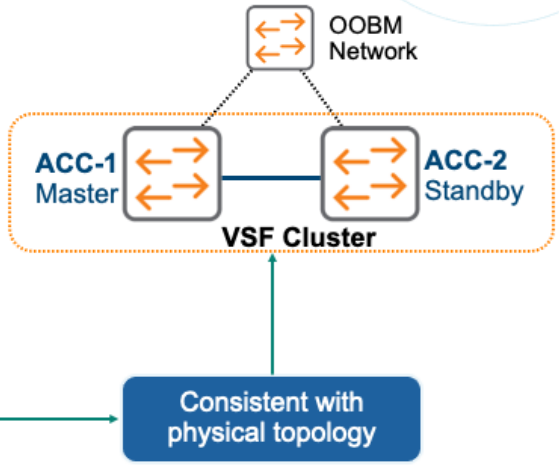
 VSF Member 1

     Link        Peer     Peer
Link State       Member   Link   Interfaces
---- ----------- -------- ------ --------------------------
1    up          2        1      1/1/27


 VSF Member 2

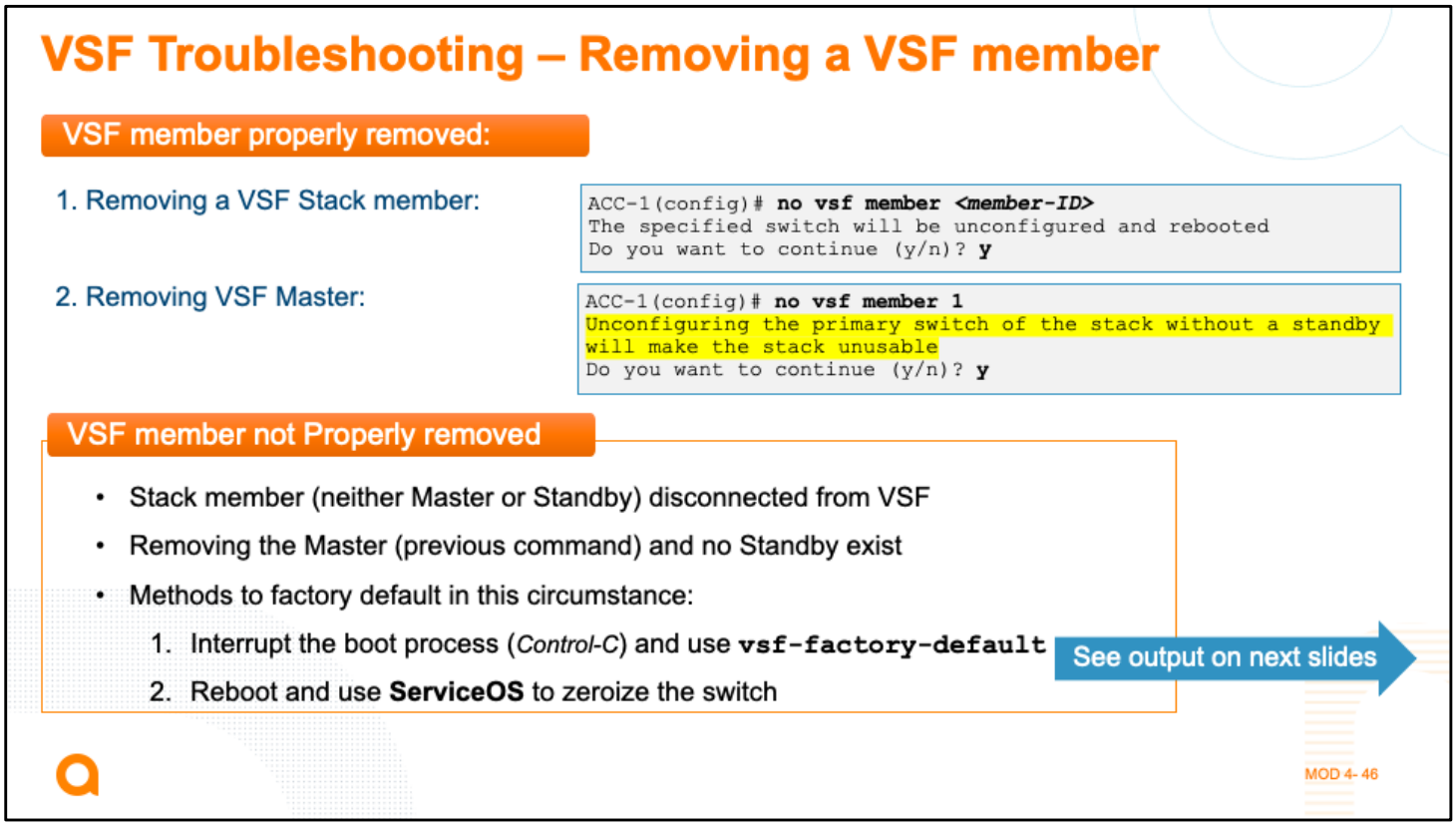
     Link        Peer     Peer
Link State       Member   Link   Interfaces
---- ----------- -------- ------ --------------------------
1    up          1        1      2/1/27

ACC-1# show vsf topology
 Stdby     Mstr
 +---+     +---+
 | 2 |1==1| 1 |
 +---+     +---+
```



show vsf link command displays the information about the VSF Link that are in use by all members.

show vsf topology command displays the physical topology from the VSF process.



To remove a VSF stack member, use the following command (replace the member ID with the desired value):

ACC-1(config)# no vsf member <Member-ID>

This command will remove all stack configuration from the VSF-Master for the specified member. If that member is currently operating as part of the stack, it will be rebooted and restored to factory defaults.

Attempting to remove the stack primary with this command without an operating VSF-Standby will result in the following warning:

– ACC-1(config)# no vsf member 1
  Unconfiguring the primary switch of the stack without a standby will make the stack unusable
  Do you want to continue (y/n)?

If an operating member other than the primary or secondary is disconnected from the stack without being properly removed from the stack, it will reboot itself then stay down as it attempts to reconnect to the VSF-Master. (This will also occur to all stack members with the VSF-Member role if the stack primary is removed using the above command without a secondary member defined and present.)

To restore an isolated VSF-Member switch to factory defaults, you will need to use one of the following two methods via a connection to the USB-C console port on the disconnected stack member:

Method 1: Interrupt the switch boot process using the Control-C key combination; this will cause a redirect to an emergency login prompt. Once you have logged in (using credentials from the stack master the member was last connected to), use the following command to reset the switch to factory defaults:

```
switch# vsf-factory-default
```

This command will immediately reboot the switch, restoring it to a factory default state with a member ID of 1.

Method 2: Reboot the switch via the front panel Reset push button or via a power-cycle and select the ServiceOS option from the boot menu when prompted. Login to the ServiceOS using the username admin; then, from the SVOS prompt, enter the following command:

```
– SVOS> erase zeroize
– ################################WARNING##########################
  #####
– This will securely erase all customer data and reset the switch
– to factory defaults. This will initiate a reboot and render the
– switch unavailable until the zeroization is complete.
  This should take several minutes to one hour to complete.
– ################################WARNING##########################
  #####
– Continue (y/n)? y
```

The switch will reboot for zeroization, and will reboot once more after zeroization is complete, at which point it will boot to a factory default state.

VSF Troubleshooting – Removing a VSF member

Method 1:

```
System is initializing
^CCreated symlink from /etc/systemd/system/bannerd.service to /dev/null.
[FAILED] Failed to start bannerd.service.
See 'systemctl status bannerd.service' for details.
[  OK  ] Stopped bannerd.service.
Warning: bannerd.service changed on disk. Run 'systemctl daemon-reload' to
reload units.
         Starting VSF setup recovery node...
[  OK  ] Started Pluggable Module Daemon.
^C         Starting Bringup OOBM wrapper...
[  OK  ] Started Bringup OOBM wrapper.
[  OK  ] Created slice system-dhclient.slice.
         Starting DHCP client service eth0...
[  OK  ] Started Creates backup for current boot image.

AOS-CX FL.10.04.3000 6300 ttyS0

CONTROL-C

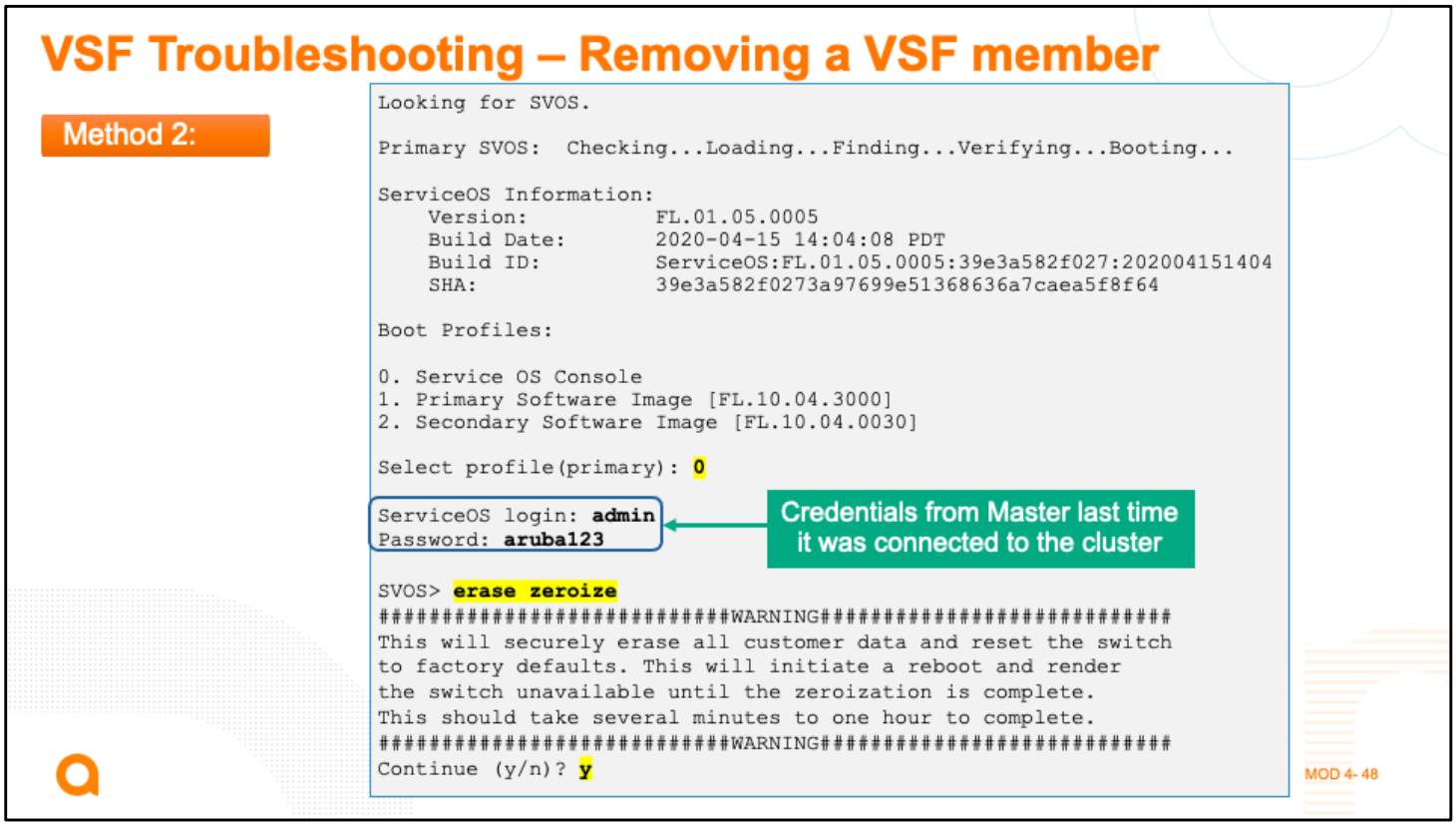
********************************************************************
WARNING! Entering emergency support login mode. This mode is for
support use only and the system will not be fully operational.
The system must be rebooted to exit this mode.
********************************************************************
6300 login: admin
Password: aruba123

switch# vsf-factory-default
```

Stop Boot process

Credentials from Master last time it was connected to the cluster

MOD 4- 47



If you cannot login you are able to login with the zeroize user and follow those instructions.

ServiceOS login: zeroize

This will securely erase all customer data, including passwords, and reset the switch to factory defaults.

This action requires proof of physical access via a USB drive.

* Create a FAT32 formatted USB drive
* Create a file in the root directory of the USB drive named zeroize.txt
* Type the following serial number into the zeroize.txt file: 772632X1830018
* Insert the USB drive into the target module
* Confirm the following prompt to continue

VSX – Split Brain

Failure Scenarios		Split-recover off	Split-recover on (default)
Keepalive is down	ISL is up	No impact (but loss of split detection)	No impact (but loss of split detection)
Keepalive is up	ISL is down	Secondary VSX node tears down its ports	Secondary VSX node tears down its ports
Keepalive is up, but after some time keepalive is down	ISL is down	Secondary VSX node's ports stay down	Secondary VSX node restores its ports (split-brain)
Keepalive is down (at the same time)	ISL is down (at the same time)	All VSX node ports are up	All VSX node ports are up
Keepalive is down (at the same time) but after some time keepalive is up	ISL is down (at the same time)	Secondary VSX node tears down its ports	Secondary VSX node tears down its ports

Best practice: Use **split-recover on** - the probability of a VSX Primary power outage is higher than for all ISL and keepalive paths to be down.

MOD 4- 49

The best practice is to keep the default split-recovery on, as the probability to hit a power outage on VSX primary is higher than for all paths for ISL and keepalive to be down. The figure provides detailed descriptions for the behavior of the VSX Secondary.


Lab Activity
Lab 04


Rev 20.41
aruba
a Hewlett Packard Enterprise company
Layer 3 Routing and OSPF
Aruba Advanced Switching, Troubleshooting and Solutions (ASTS)



After completing this module, you will be able to configure, deploy, and troubleshoot common Layer 3 routing solutions. This includes how to configure and validate static routes, along with various static routing options.

You will be able to configure and validate Policy Based Routing – to override default destination-based routing decisions in the route table.

You will be able to deploy and troubleshoot single-area OSPF solutions and describe some best practices.

You will also be able to deploy and troubleshoot multi-area OSPF solutions and implement various best practices.



As a network engineer, you are expected to have some level of mastery with networks – and that means Layer 3 routing. This module provides an opportunity for you to move closer to that ideal of network mastery.

You begin with an overview of static routing, and explore several static routing options – default routes, floating static routes, BFD, and tags.

Then you learn about Policy Based Routing (PBR), which gives you great power over pathing decisions. You learn how to override the typical destination-based pathing in the route tables to select your own best paths. You learn how to configure and validate PBR before reviewing Virtual Routing and Forwarding (VRF).

The rest of the module is focused on OSPF – arguably the most popular internal routing protocol. You will review reasons for this popularity and how to maximize these benefits – scalability, resiliency, efficiency, and convergence. You learn some fundamental troubleshooting relevant to both single and multi-area deployments. You learn the power of having a well-crafted, accurate, up-to-date network diagram. You see how this one tool can enable simple “compare-and-contrast” or “spot-the-difference” methods of troubleshooting – especially as relates to the OSPF route table, neighbor table, and Link State Database (LSDB). You will also see how the reverse is true – you can use the output of OSPF show commands to validate the accuracy of your network diagram, or to help you create a new diagram.

It cannot be overstated the golden opportunity you have when your network is running

smoothly. This is the time to create good network documentation, to prepare you for network outages. With a well-crafted network diagram, you quickly resolve issues, and become a hero. Without it, maybe not.

So you move into more scalable and complex scenarios with OSPF multi-area solutions. You learn about LSA types and usage, ABR's, and route filtering techniques.

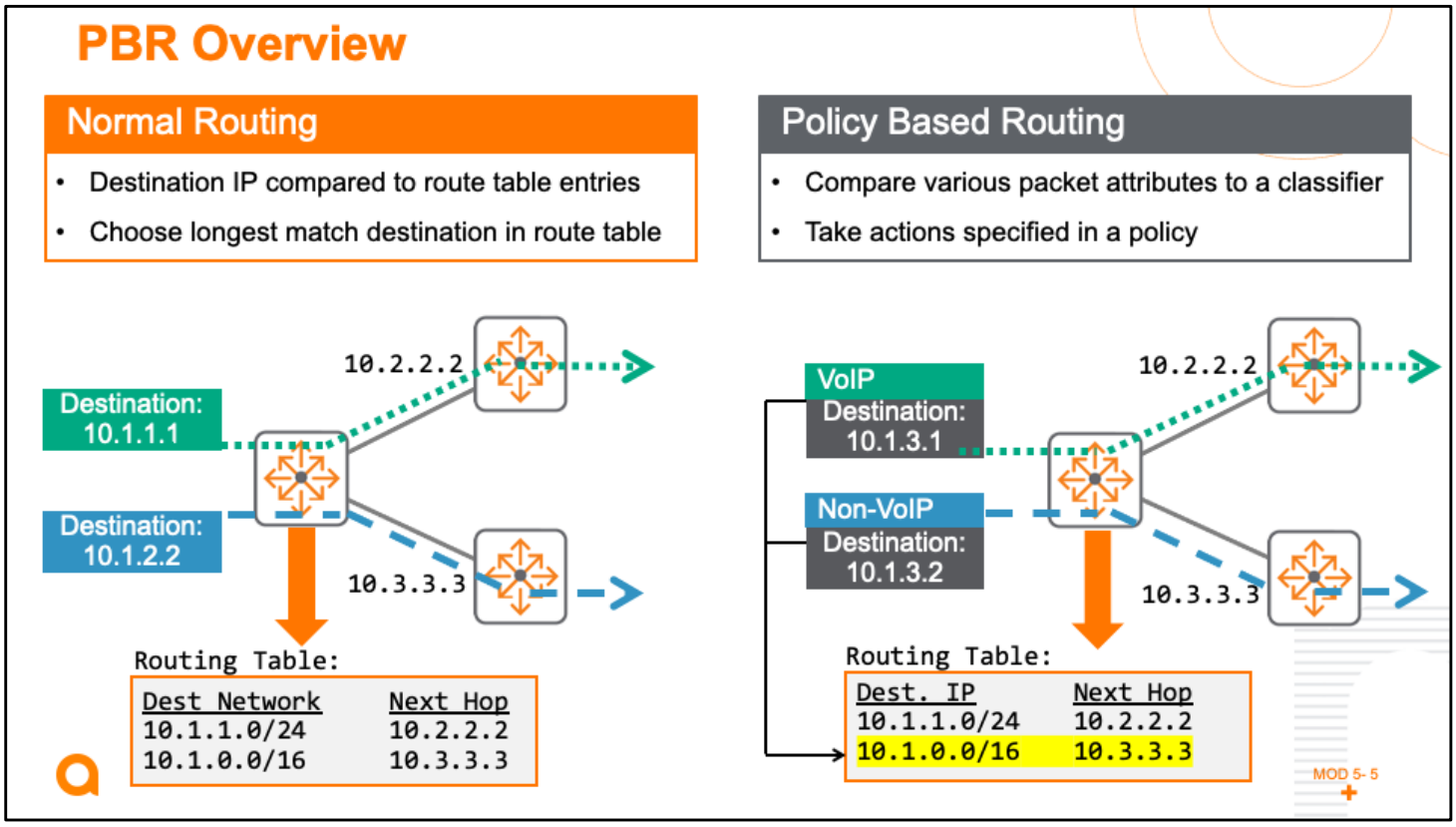
You continue to explore OSPF multi-area solutions by using interface costs to manipulate paths. You learn to use passive interfaces and authentication to improve network efficiency and security. Then you learn how to use BFD to improve convergence time, before reviewing best practices and troubleshooting commands.

Finally, you apply this knowledge with some hands-on lab activities.

Policy Based Routing

Overview, configuration, validation

aruba
a Hewlett Packard Enterprise company



The figure compares normal routing to Policy Based Routing (PBR).

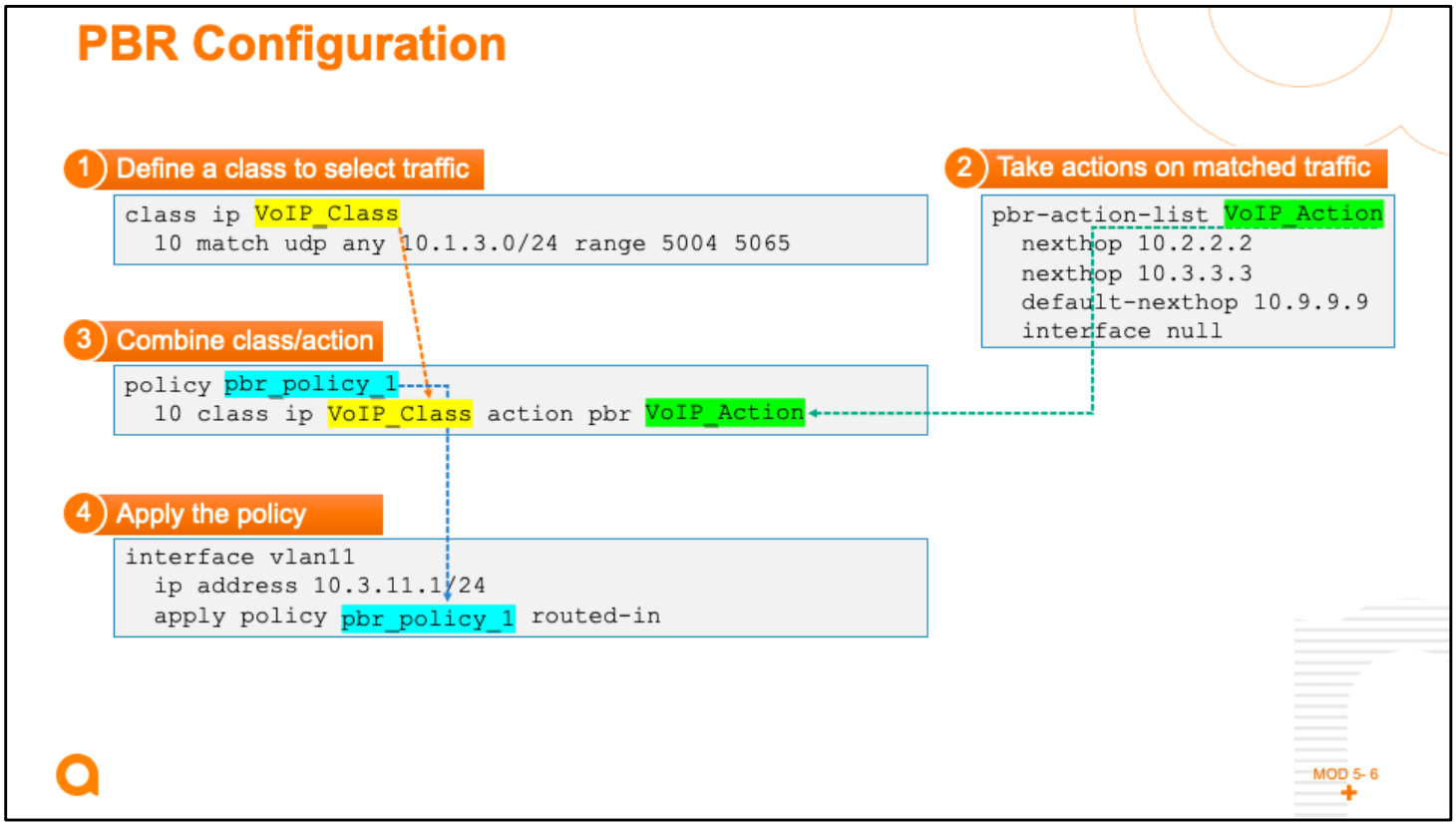
Normal routing is purely destination-based. As each router receives an inbound packet, it compares the packet's destination IP address to the route table, and chooses the entry with the longest match as the best path. In the example, the packet destined to 10.1.1.1 matches both entries in the route table, but only 16 bits match in the top entry, while 24 bits match in the second entry. So the router routes this packet via next hop 10.2.2.2. Next, a packet arrives destined to 10.1.2.2. This destination only matches the top entry, and so 10.3.3.3 is the next hop.

PBR lets you manipulate the path of a packet based on various packet attributes. Packets must be destined to a valid subnet on your network. As each router receives an inbound packet, it compares appropriate header and tag information to a classifier that you configure. The router then takes action on this matching traffic based on a policy. Matching traffic with the same destination can be routed over different paths, to improve traffic handling, performance, and/or security for various traffic types.

In the example, both packets are for the same destination subnet. However, you have created a classifier that matches on the range of UDP ports used for Voice over IP (VoIP) traffic – perhaps UDP ports 5004-5065. This traffic is routed over its own path via destination 10.2.2.2. Note that this action overrides what normal routing would do by default – the ingress router only has next-hop 10.3.3.3 for 10.1.0.0/16. However, 10.2.2.2 is a valid path, and it is directly connected to the ingress router, as required by PBR. In other words, recursive lookups are not

supported.

All other, non-VoIP traffic does not match the classifier, and so is routed as normal, via 10.3.3.3.



The example in the figure provides an overview of PBR configuration.

1. Classifier

First you define a classifier – to select which traffic shall have some special actions applied to it. In the example, you create a class called “VoIP_Class”. This class shall include UDP traffic from any source to destination 10.1.3.0/24, with a destination UDP port between 5004 and 5065 inclusive.

2. Action list

Next, define a PBR action list – to control how some traffic class is to be treated. In the example, you created a list named “VoIP_Action”, which contains four possible actions in an ordered list. You can have a maximum of eight action entries in an action list, 512 total entries across the entire device.

Action List entries are processed top-down, like an ACL. The first entry indicates that the next hop for matched traffic is 10.2.2.2. If that next hop is not available, perhaps due to a link, interface, or router down condition, then the next hop is 10.3.3.3. If that next hop router is also down, then use a default next hop of 10.9.9.9. And finally, if that next hop is down, silently discarding the packet. In other words, forward it out the “null” interface – the proverbial bit bucket.

The example shows three of the four possible actions. The other one is interface tunnel, which

sends traffic over some GRE, 6in4, or 6in6 tunnel. The example does not show the default sequence numbers, which are automatically assigned based on order of entry – in increments of 10. You can configure your own sequence numbers on action list entries if desired.

Understand that next hop actions must point to directly connected IP addresses – PBR does not support recursive lookups to reach non-directly connected next hop addresses.

3. Create a policy to combine classes and actions

Now you create a policy to associate each class with an appropriate action list. The example associates the VoIP_Class with the VoIP_Action. As hinted at by the sequence number 10, each policy can have multiple class-action associations. Just keep in mind where you will apply that policy, to ensure that class-action pairs are appropriate for where you intend to apply the policy, as described below.

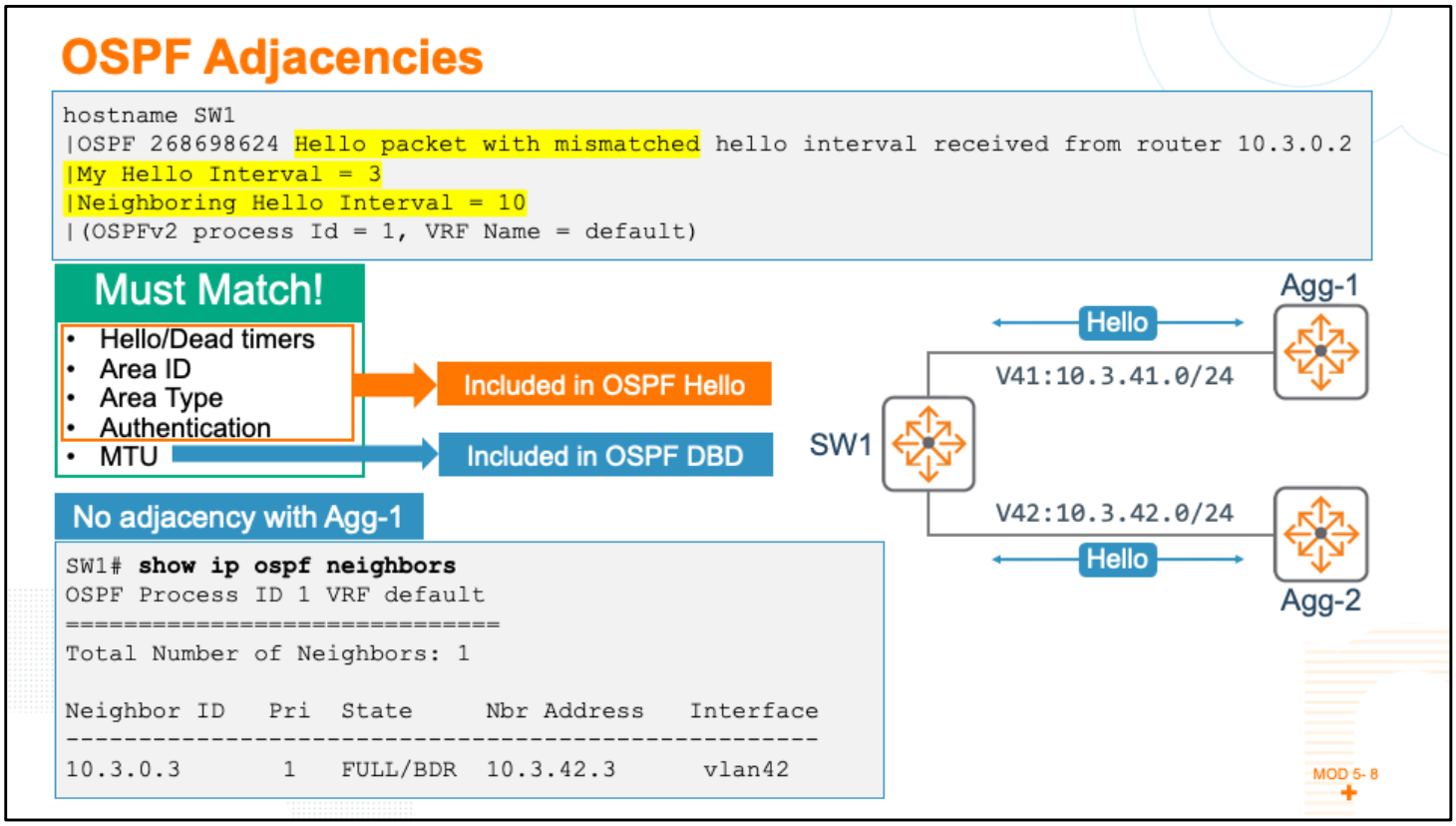
4. Apply the policy

Access config-interface mode on the appropriate interface and apply the policy – always as an inbound policy. In other words, the policy is applied as packets enter the router inbound. This makes sense, correct? The purpose of PBR is to override normal destination-based routing. If you applied the policy outbound on an interface, it would be too late – routing decisions are already made.

OSPF Single Area

Operation, adjacencies, functional assessment

aruba
a Hewlett Packard Enterprise company



Of course, directly-connected routers must form an adjacency before they can share routing information and properly route traffic. They use the hello mechanism you just learned about. Each hello contains several pieces of information, including the following:

OSPF hello and dead interval timers,

Area ID

Area Type (normal, stub, etc.)

Authentication type (none, cleartext, or md5)

Authentication key (must match on both sides if authentication is used)

If the two routers disagree on this information, likely due to misconfiguration, then the adjacency will not form.

MTU value must be the same on the interfaces used to interconnecting the OSPF peers. This value is not included in the OSPF hello message but in the Database Descriptor message.

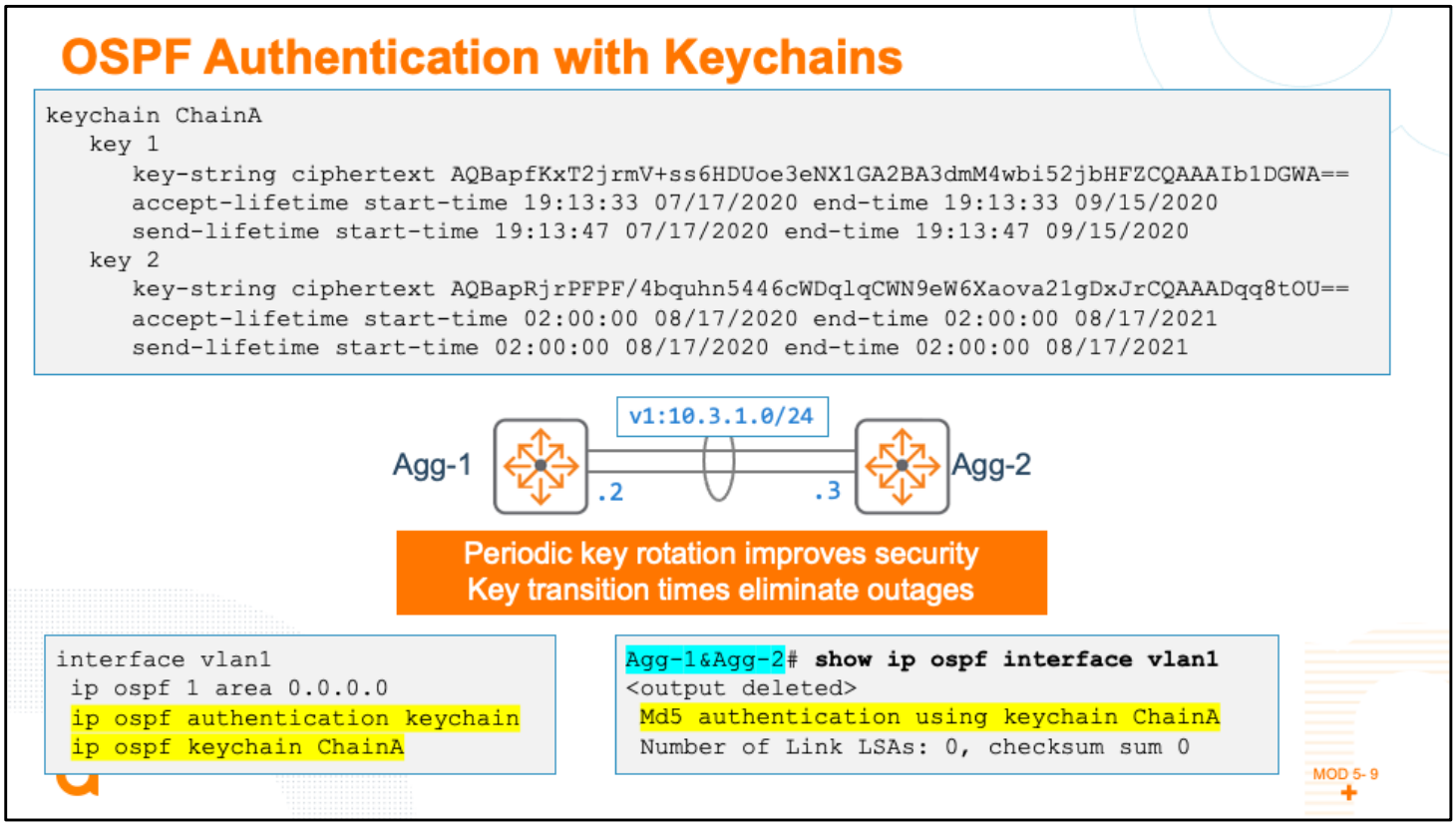
In the example, SW1 has an Adjacency with Agg-2, but not with Agg-1. Why is this?

The debugging output from SW1 reveals the answer. “My Hello Interval = 3”, while “Neighboring Hello Interval = 10” – the problem is a mismatched hello interval.

You could have also found the problem using the “compare and contrast” or “What’s different” method of troubleshooting. Just issue the command show ip ospf interface vlan41 on SW1

and Agg-1 and compare the outputs until you find the difference.

To resolve the issue, you must configure SW1 to match Agg-1, or vice versa.

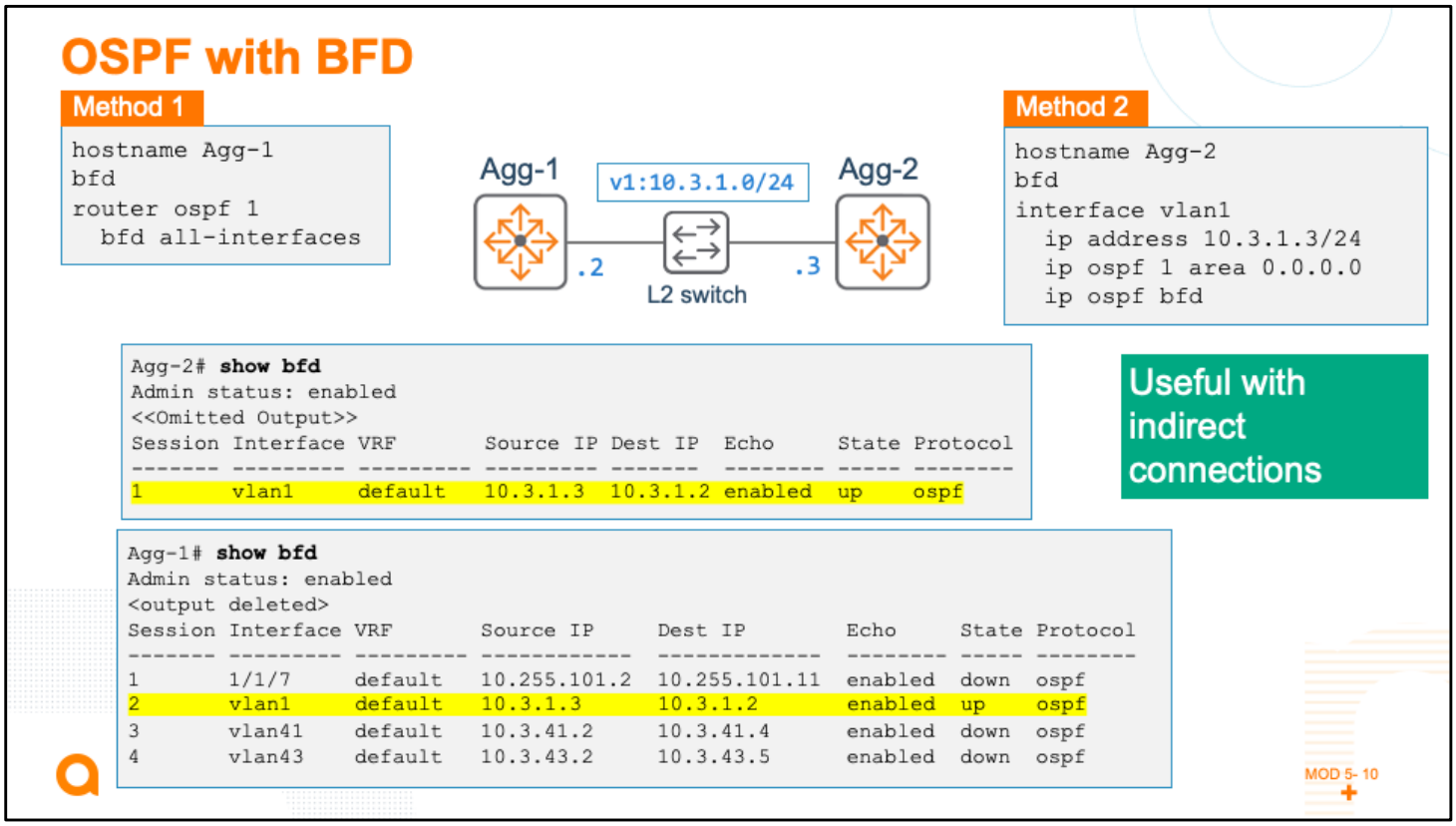


As of ArubaOS-CX 10.05, you can configure OSPF authentication using keychains. You get the same MD5 authentication, but with the added flexibility to change keys while avoiding outages. Imagine that you have a real keychain that holds your car keys and the keys for your new boat. It also holds two keys to your new castle in Scotland. One key is for the current locks, installed by the original owner. This key will work for another week, at which time the locks will be changed. Then you must stop using the old key and start using the new key.

OSPF supports a similar capability. Suppose your corporate policy is to periodically change keys, say every 12 months – or whenever a network engineer leaves the company. You access global configuration mode and configure a keychain named “ChainA”, as shown in the figure. This keychain holds two keys. Key 1 is currently in use but will only be accepted until September 15th, 2020. You configure a second key – key 2. This key will be accepted starting August 17th, until the same date in 2021.

Notice that there is a 1-month period where both keys are accepted. This gives your staff time to ensure that the new key is properly configured and supported on all OSPF routers. After that time, you can remove key 1 from the chain and create a new key for the following year. You can create up to 64 keychains, each with up to 64 keys. You can see this with the command show capacities keychain.

You can view all keys on the keychain with show keychain ChainA.



Enabling BFD Support for OSPF enables OSPF to register with Bidirectional Forwarding Detection (BFD) to receive forwarding path detection failure messages. This is useful when you have indirect (broadcast) connections. You can either configure BFD for OSPF globally on all interfaces or configure it selectively on one or more interfaces. BFD creates a session in asynchronous mode as soon as the switch reaches the 2-Way state with a neighbor. Once the session is established BFD will commence sending echoes for path failure detection.

There are two methods for enabling BFD for OSPF:

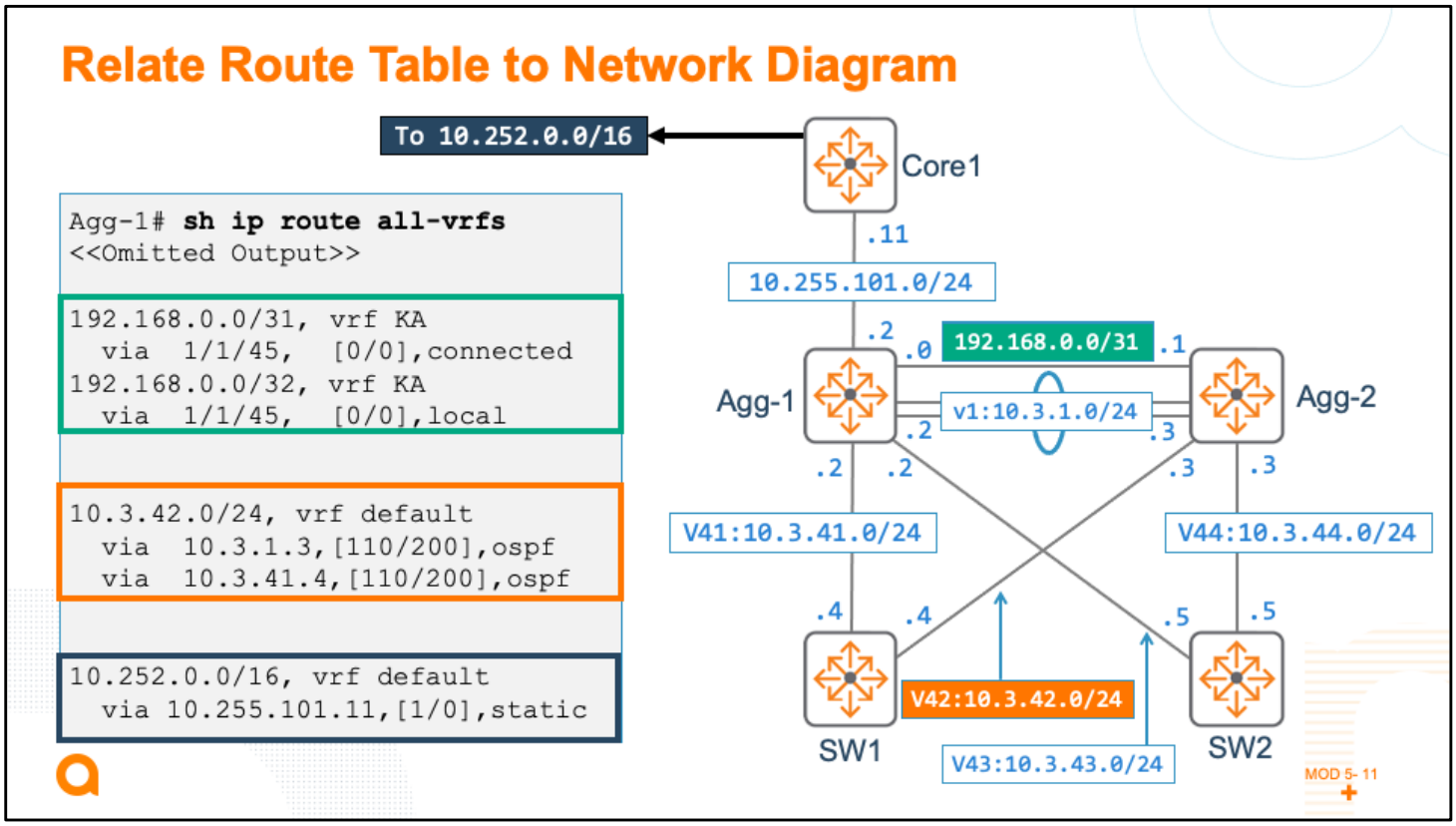
Enable BFD for all interfaces enabled for OSPF with the bfd all-interfaces command in router configuration mode. With this option, you can then disable individual interfaces as desired. Just access the interface and use no ip ospf bfd.

Enable BFD for a subset of interfaces that have OSPF enabled by using the ip ospf bfd command in interface configuration mode.

In the example Agg-1 is configured with method 1, while Agg-2 is configured with method 2. Notice that in both cases, you must first enable bfd globally on the router – from global config mode. If you fail to do this, the admin status from show bfd would be admin-down.

Although not shown in the diagram, router Agg-1 has 3 other OSPF neighbors besides Agg-2. Since Agg-2 is currently the only neighbor with BFD enabled, it is the only neighbor with a BFD status of “up”. The other three will be down until you configure BFD on those routers.

Understand that only BFD is not functional for these other three neighbors, but OSPF itself remains fully functional – you could use show ip ospf neighbors to prove that the neighbors are still up and show ip ospf lsdb and show ip route to prove that all is ok.



Creating diagrams to facilitate network troubleshooting

The figure shows partial output from the command show ip route all-vrfs, along with an associated network diagram. You should be able to relate these two entities, use them for troubleshooting, and draw conclusions from this information. This network diagram appears to have been created by a network engineer, because it is focused on networks – it is specifically created to assist in troubleshooting network-related issues. What does the output of show ip route give you? Networks. What is highlighted in the diagram? Network numbers.

This diagram format also saves “real estate” on the printed page, with less clutter. The network portion of each subnet is listed only once, with the host portion of the address listed next to the appropriate interface. Thus, you can directly compare the output of show ip route to a network diagram for diagnostic purposes, without the clutter of redundant network information. In other words, you have set yourself up for success – before any problems occur.

Relate Route Table output to a well-crafted network diagram

Look at the top two entries in the route table. The first thing you should notice is that they are in a different VRF, named KA – only shown because you used the all-vrfs option with show ip route.

Next, notice that that the top entry (192.168.0.0/31) is a directly connected network on

interface 1/1/45. The next entry shows you the local /32 host address on that network – assigned to interface 1/1/45, which is configured as a Layer 3 routed interface.

The remaining entries are for the default VRF. Destination 10.3.42.0/24 was learned via OSPF, and there are two equal-cost paths. Agg-1 can reach the target via next-hop 10.3.1.3 (Agg-2's VLAN 1 interface) with a cost of 200. This is shown in the second number in brackets [110/200]. The other path is via 10.3.41.4 (SW1's VLAN 41 interface), also with a cost of 200. This means traffic can be load-shared across these two equal-cost paths. The default number of maximum paths for load sharing is 4, but you can access the config-ospf context and use the maximum-paths command to set the value between 1 and 8.

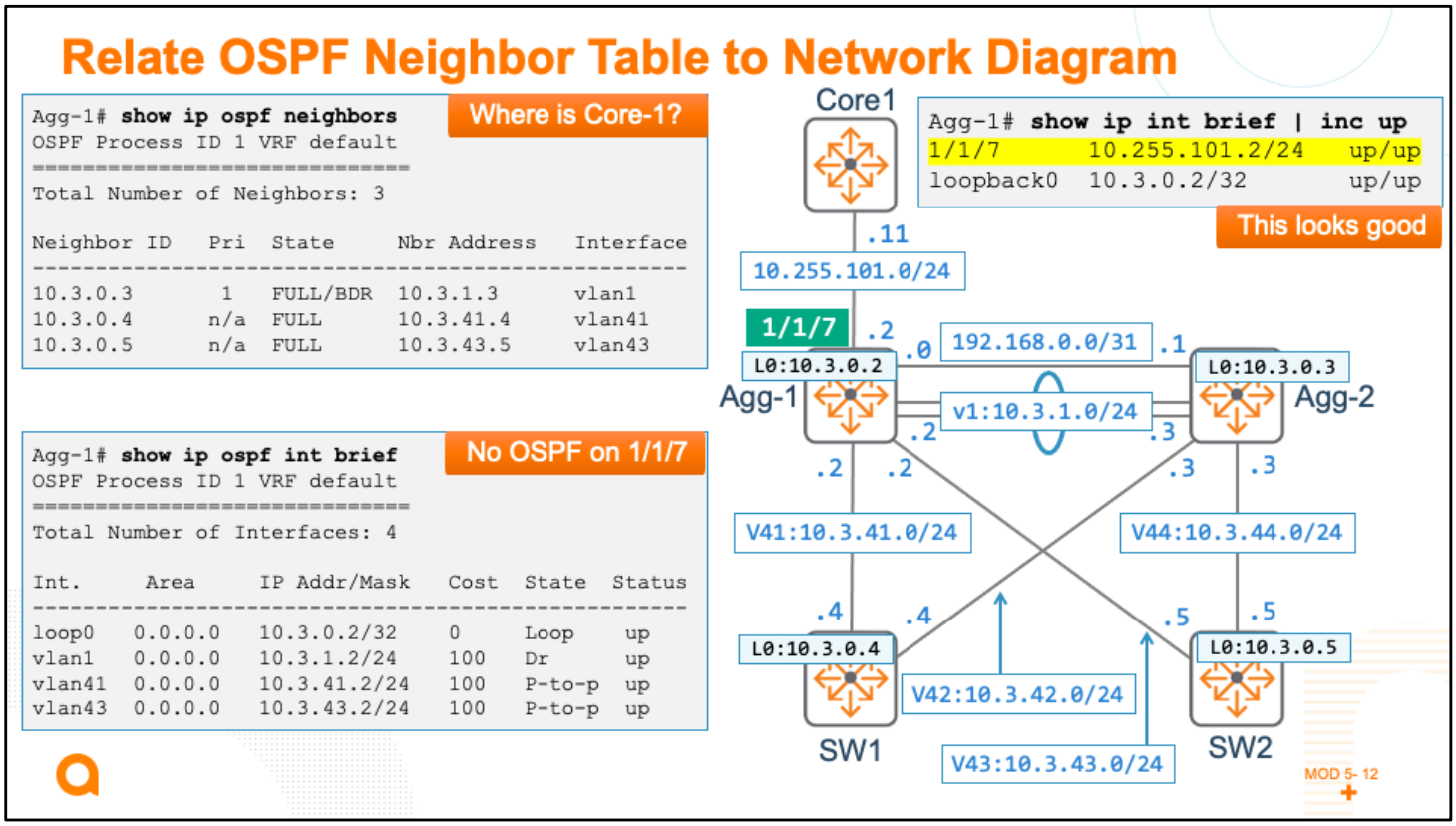
Finally, noticed that a static route is configured. To reach destination 10.252.0.0/16, use next-hop router 10.255.101.11 – an interface on the Core router.

Cost and administrative distance

You know that the second number in brackets is the cost to reach a destination, such as [110/200]. Remember that the route table does not list every path to each destination – only the best paths. So you know that any other paths to a particular destination would have a higher cost. You can see these alternate paths in the routing database. For OSPF, you would use the command show ip ospf lsdb. The cost is used within a single routing protocol, like OSPF, to choose the best path.

Recall that the first number is the administrative distance. This is a kind of "trust rating", used to choose the best path between different routing protocols. If you configured a static route to destination 10.3.42.0/24, it would be placed in the route table, and the two equal-cost OSPF entries would be removed. As you can see in the figure, static routes have a default admin distance of 1, while the OSPF admin distance = 110. In other words, static routes are trusted more than OSPF-learned routes.

You can adjust these default administrative distance settings to tune each router's path decisions. Let's look at an example.



Relate OSPF Neighbor Table output to a network diagram

Recall that directly-connected OSPF neighbors exchange hello messages. They compare the information contained in received hello messages to their own configuration and validate compatibility. If compatible, they move into a FULL or TWO-WAY state (depending on OSPF network type) and begin to exchange routing information. Incompatible routers do not form a full adjacency, do not share routing information, and the network is broken. You must figure out why.

As you learned earlier, a good network diagram facilitates your diagnostic efforts. Prepare your network diagram when all is OK – when things fail, you become known as someone who can quickly diagnose complex issues. You can also use these commands to help you create a network diagram. Having such a diagram enables you to use easier “compare and contrast” or “spot the difference” methods of troubleshooting. This is a bit like preparing the answers to the test before you take it.

Suppose you access the Agg-1 router and use show ip ospf neighbors, as shown in the figure. The diagram shows that Agg-1 is directly connected to four routers, yet only three are listed – Agg-2, SW1, and SW2. Where is Core-1? You use show ip int brief | include up to see if interface 1/1/7 is the problem. No – that interface is up and has the correct IP address assigned. You might also try to use ping 10.255.101.11, which would succeed. The interface is up and fully functional but does not form an OSPF adjacency with Core-1.

Perhaps 1/1/7 is not participating in the OSPF process, which is confirmed with show ip ospf interface brief. In this case, this is intentional – you just saw that a static route is configured over that link, to reach the 10.252.0.0/16 subnet. However, if this was not intentional, you would have found the issue using three quick commands. Depending on the situation, you may have started with show ip ospf interface brief and found the issue with one command. Again, this assumes you have a well-crafted, up-to-date, accurate network diagram.

Analyze the neighbor table

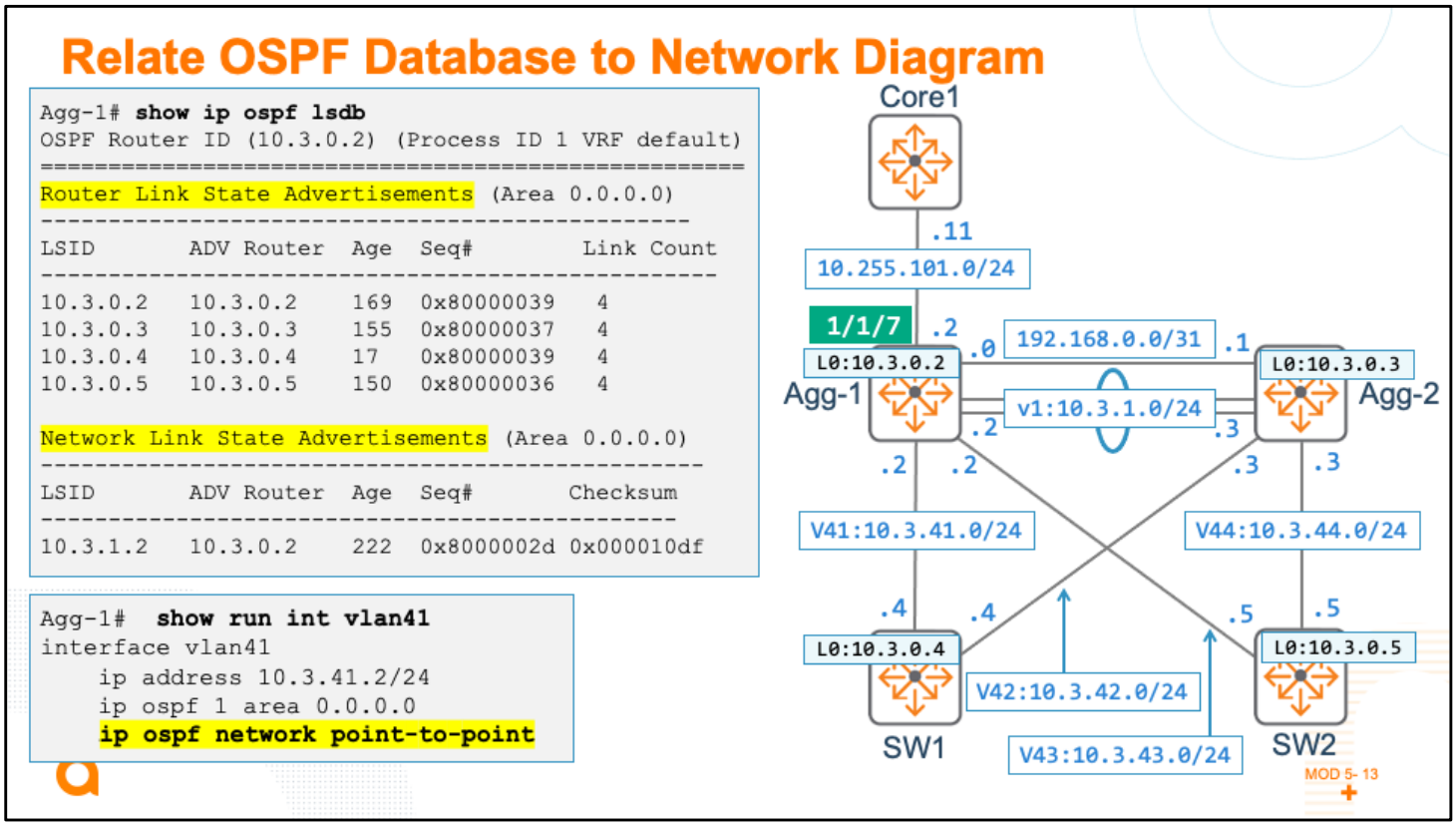
You can deduce and if need be, document more information about this network. Look more closely at the neighbor table. Neighbor 10.3.0.3 (Agg-2 Loopback) has a priority of 1 and has been elected as the Backup Designated Router (BDR). This only occurs on a broadcast-capable network, like Ethernet – not on Point-to-Point networks. So you know that VLAN 1 was not changed from its default OSPF network type of “broadcast”. Since Agg-2 is the BDR, and Agg-1 is the only other OSPF router on this link, you can assume that Agg-1 is therefor the DR. This is confirmed with show ip ospf int brief. The status of Agg-1’s VLAN interface is “DR”. The OSPF interface Priority can be used to control this election. The router with the highest priority wins, with router id used as a tie-breaker. One motivation to control DR election is in large scale environments. It is better to split the load amongst routers than to have one router act as the DR or BDR for 10 or more subnets. Certain high availability solutions might also perform better with judicious DR/BDR selection.

If there were multiple routers on this broadcast network, do you recall what state their interfaces would be in? They would be in a “TWO-WAY” state, and this would be normal and OK. When multiple routers are neighbors on the same broadcast network, the non-DR and BDR routers (DROTHER) only form a FULL adjacency with the DR and BDR, and a TWO-WAY state with all others on that link.

However, a TWO-WAY state in nearly any other case is cause for alarm. It indicates routers with incompatible configuration parameters. You might explore this with show ip ospf interface – without the brief option, and/or show ip ospf neighbor detail.

For now, let's stick with the output shown in the figure. Neighbor 10.3.0.3 has a priority of 1, while the priority value is not applicable (n/a) for the other two neighbors. Why? Again, show ip ospf interface brief tells us why – these subnets were configured as OSPF Point-to-Point links. This is shown in the figure as P-to-P. Since point-to-point networks only have two routers, there is no need to elect a DR/BDR, and so there is no need for a priority value.

Note: the actual output would show as "point-to-point". This and some other output was shortened to improve readability of the figure.



Relate OSPF Database output to a network diagram

The OSPF Link State Database (LSDB) gives you big-picture insight into your routing systems and networks. OSPF routers learn about routes via Link State Advertisements (LSA), of which there are several types. The LSDB is organized based on these LSA types.

For example, LSA Type 1 is also known as a Router LSA – used for a router to advertise itself. This is like showing up to a party and saying, “Hey everyone! I’m here and my name is Rashid!” You see Type 0 Router LSAs in the top section in the figure, under Router Link State Advertisements. The Link State ID (LSID) is WHAT is being advertised, while the advertising (ADV) router indicates WHO is advertising it. In the example, they match a router with ID 10.3.0.2 is advertising itself, “Hello everyone! I am 10.3.0.2”. There are four such entries, and the diagram indicates that this network consists of four OSPF routers. (Recall that Core-1 does not currently run OSPF).

If your network diagram indicates that you should have 215 routers, but only 213 routers appear in the list, you know that 2 routers are either down or disconnected. That is the power of this command. You can issue the command from any router, to validate the basic status of every router. Remember, there is only one reality. The streets in your city are shared by everyone – everyone uses the same map. Only your position in that city is unique from others.

Likewise, there is only one topology, and so all routers in the same area have an identical topological or link state database. Regardless of which router in area 0.0.0.0 you access, the output of show ip ospf lsdb is identical.

LSDB Aging and Sequencing

The age indicates how long this entry has been in the table without going down or needing to be refreshed. Recall that unlike distance-vector routing protocols like RIP, OSPF doesn't use periodic updates – it only updates entries if there is a change in the state of a link (link state routing). However, if no changes occur within 30 minutes (1800 seconds), an update does occur. In other words, an age of greater than 1800 seconds indicates a network that is probably about to time out. That network may be down.

The sequence number increments every time there is a link state change (from up to down, down to up) and every 30 minutes, when there is a refresh. So, a rapidly incrementing sequence number would indicate a flaky network segment – an interface is flapping up-down-up-down. This might also show up as increased CPU utilization and reduced network performance. You can mitigate these types of issues and improve scalability by splitting your OSPF network up into areas.

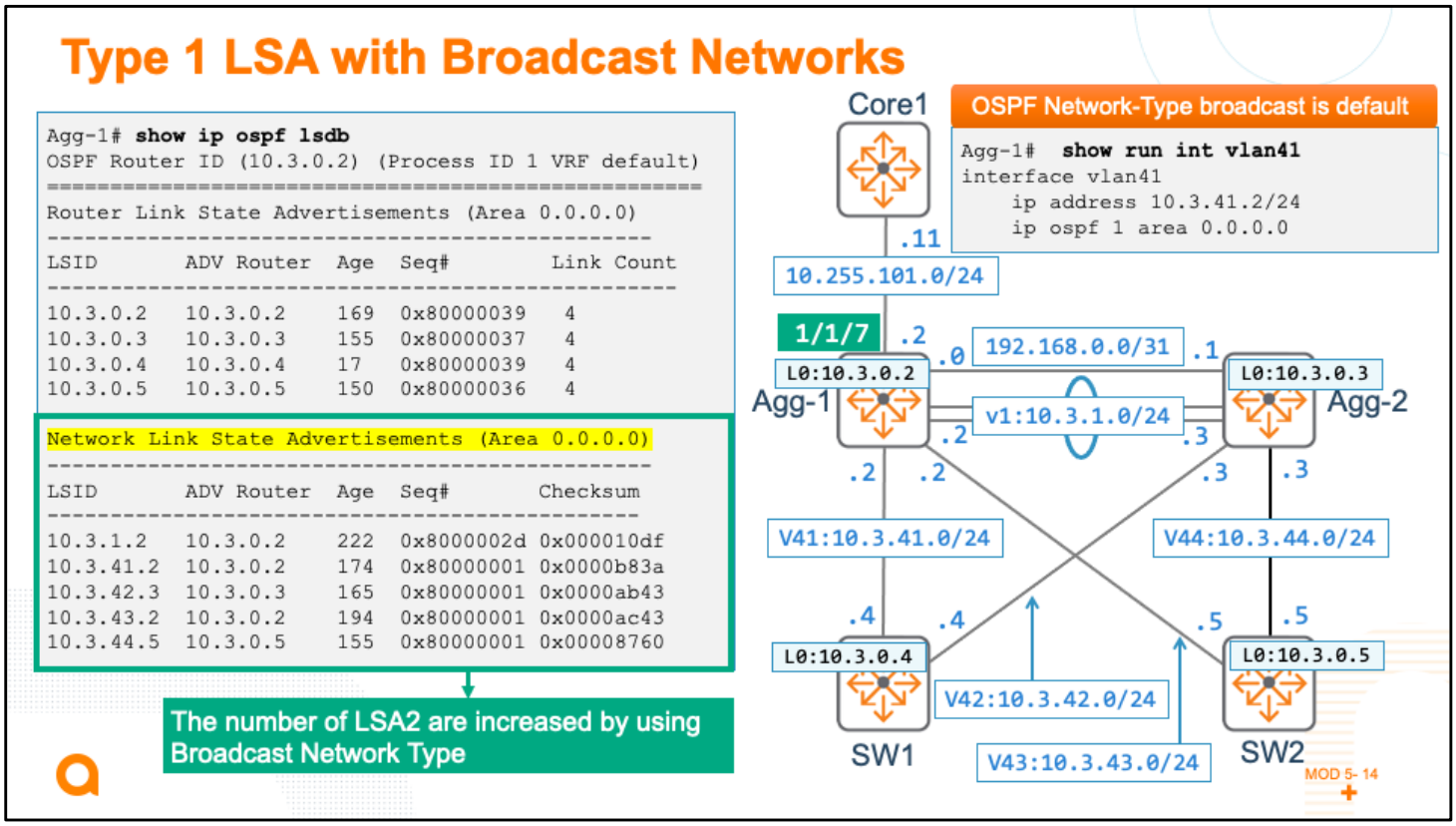
The link count column is a powerful field – as long as you have a good network diagram. You can sit at any router in the area and determine the interface status of every router in the area. If the diagram shows that Router213 has 5 OSPF interfaces, but the Link Count only lists 4, you know that an OSPF interface is down.

Type 2 Network LSA

The next section is Type 2 network LSAs. This is a router advertising the links that it "owns" or is directly connected to and responsible for advertising – the router is a DR or BDR. Again, if you know that you have 1,935 broadcast networks in your system, you should see them all listed here. You previously learned that network 10.3.1.0/24 is a broadcast network, and router 10.3.0.2 is the DR for that network. Thus, 10.3.0.2 shows up as the advertising router for that network – it shows in the output as Agg-1's IP address on that network – 10.3.1.2. Go back and compare the output of show ip ospf neighbor with

show ip ospf lsdb if you like. This is a very good practice. Diagnostic mastery improves with your ability to relate these OSPF commands to each other, and of course, to an accurate network diagram.

But what about the 10.3.4x.0/24 subnets (.41, .42, .43, .44)? Why don't they show up here? They are configured as Point-to-Point networks (See show ip ospf interface brief from the previous slide). Let's see what the show ip ospf lsdb output looks like when these are configured as the default, broadcast network type.



This example is identical to the previous one, with one exception – the 10.3.4x.0/24 subnets were configured as type broadcast – the default for Ethernet networks.

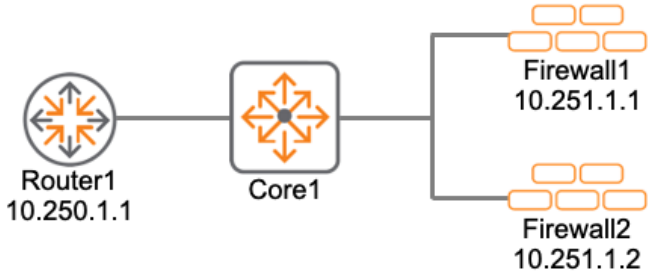
Now these networks have a DR that is responsible for advertising them, and so they appear in the Network LSA section of show ip ospf lsdb.

The intent here is to show you how different configurations affect the LSDB, so you know what to expect when troubleshooting. It is not suggested that OSPF broadcast networks are preferred over point-to-point. In fact, because the VLAN4x links are used in a point-to-point fashion, it is a good idea to configure them as such, to reduce the small but unnecessary overhead of DR/BDR election. The use of point-to-point network types configured on all Layer 3 point-to-point links helps to optimize OSPF and is recommended as a best practice in Aruba's Mobile First Campus VRD.

Note: The Checksum field in the Router Link State Advertisement section was removed to enhance the clarity of the figure, because it has little diagnostic value.

Question #1

```
hostname Core1
pbr-action-list Firewall-PBR
  nexthop 10.251.1.1
  default-nexthop 10.250.1.1
  interface null
```



Using the configuration presented, what would be the next-hop used by Core1 if Firewall1 and Firewall2 are not reachable?

a. 10.251.1.2 – Firewall 2

b. 10.250.1.1 – Router 1

c. Traffic will be sent to the NULL interface

Knowledge Check

Correct answer:

B

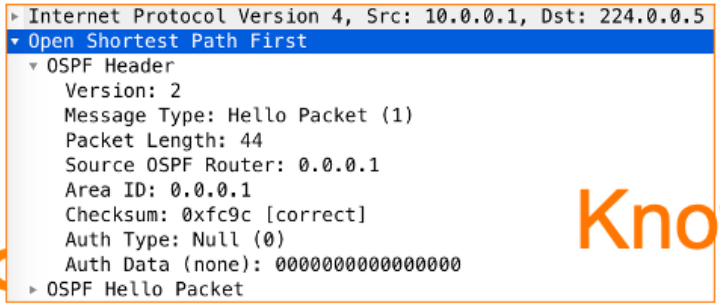
Question #2

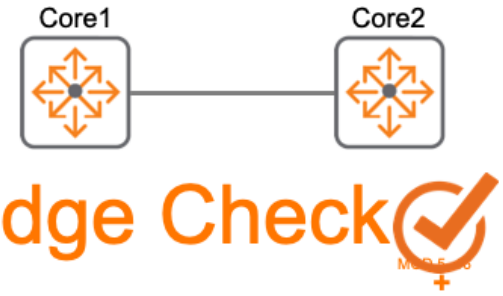
An administrator configures a couple of OSPF devices, but the OSPF neighbor relationship is not coming up, the administrator then decides to capture the traffic, Wireshark is showing that both devices are using the same information in the OSPF header, what could be the problem?

a. Multicast traffic destined to 224.0.0.5 is filtered in one or both devices

b. MTU value does not match in both devices

c. OSPF is not enabled in one or both devices





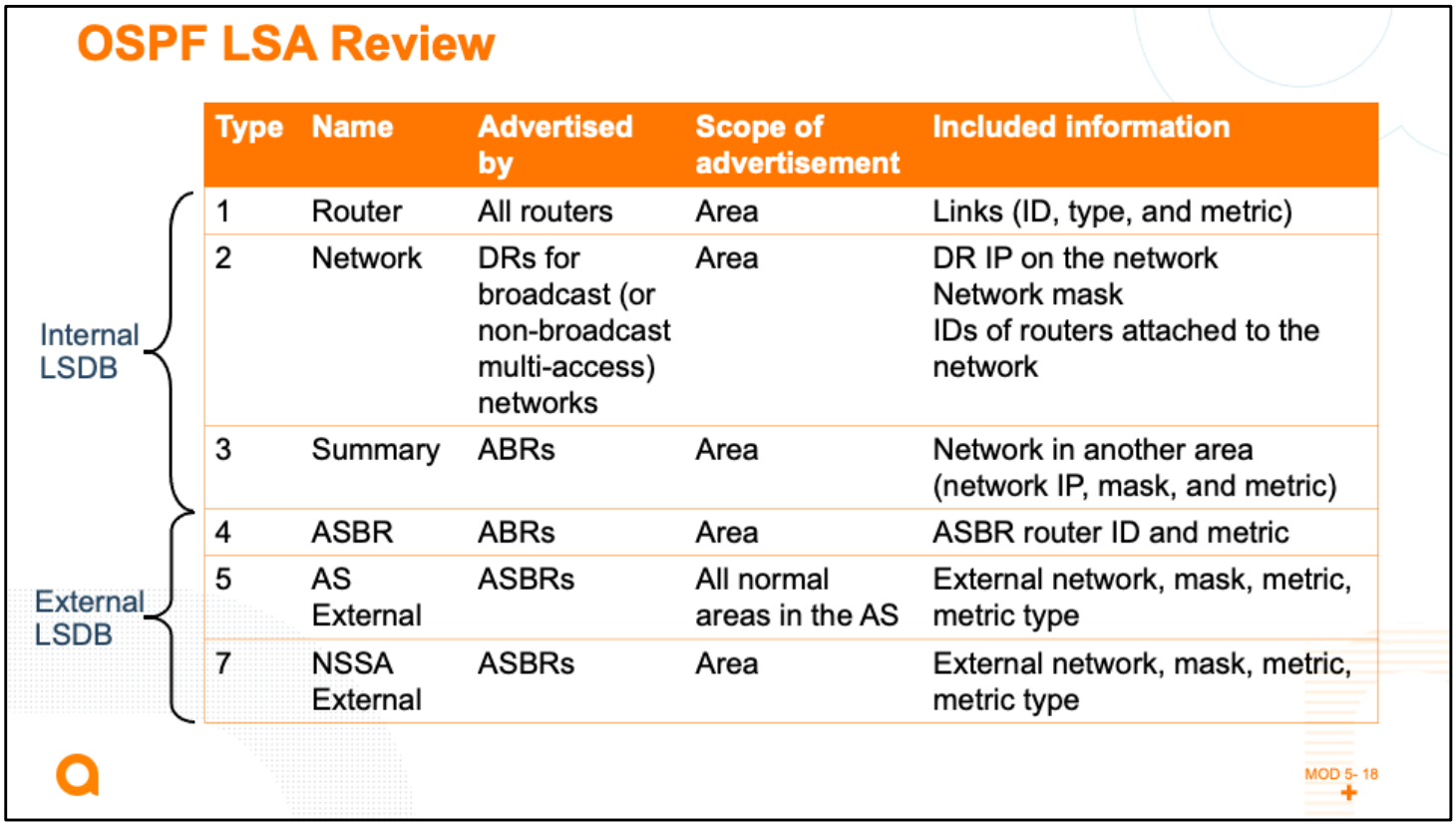
Correct answer:

A and B

OSPF Multi-Area

LSAs, ABRs, Aggregation / filtering / summarization, route tuning

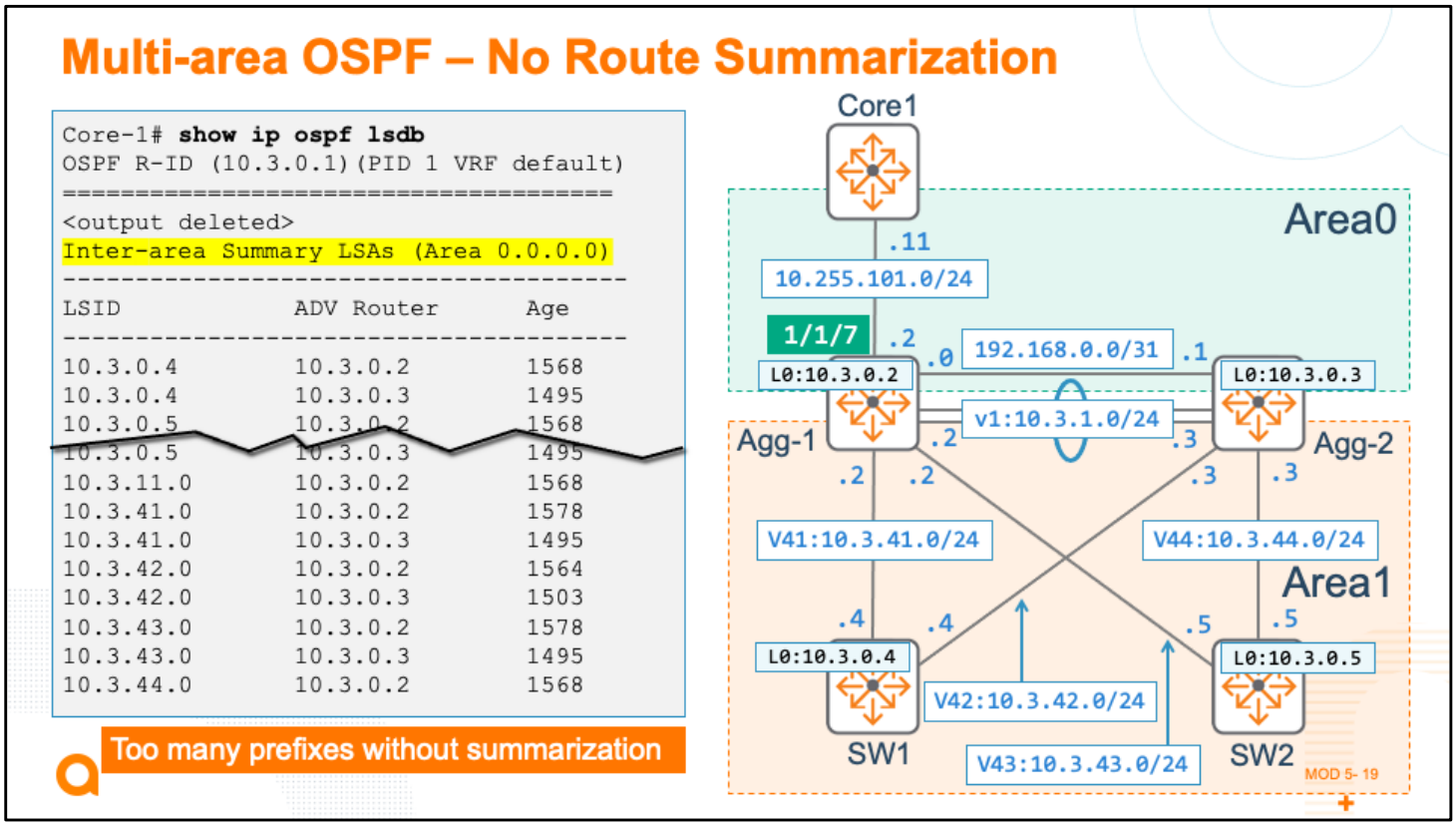
aruba
a Hewlett Packard Enterprise company



Type	Name	Advertised by	Scope of advertisement	Included information
1	Router	All routers	Area	Links (ID, type, and metric)
2	Network	DRs for broadcast (or non-broadcast multi-access) networks	Area	DR IP on the network Network mask IDs of routers attached to the network
3	Summary	ABRs	Area	Network in another area (network IP, mask, and metric)
4	ASBR	ABRs	Area	ASBR router ID and metric
5	AS External	ASBRs	All normal areas in the AS	External network, mask, metric, metric type
7	NSSA External	ASBRs	Area	External network, mask, metric, metric type

The chart summarizes key information for the various LSA types. This table does not show Type 6 LSAs because they advertise multicast groups for Multicast OSPF (MOSPF), a protocol that routers rarely use and that AOS-CX switches do not support.

The table provides information about LSA types. You already learned about Type 1 and 2 LSAs, and you will soon review LSA types 3 to 7.



The figure shows an example of a multi-area OSPF deployment with no route summarization – all individual links show up in the LSDB of router Core-1. You learned that both ABRs (Agg-1 and Agg-2) advertise the Area 1 routes into Area 0 and you see that in the figure. Each LSID is listed twice – once from Advertising router 10.3.0.2, and once from 10.3.0.3.

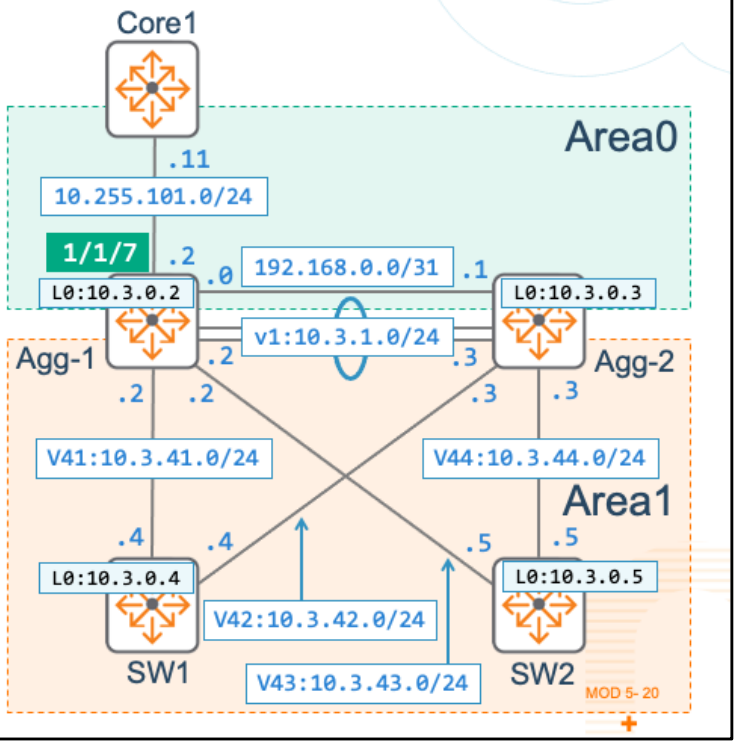
Perhaps you can configure route summarization to reduce the number of LSA, the size of the LSDB and route table.

Multi-area OSPF – Route Summarization

```
hostname Agg-1&Agg-2
router ospf 1
 area 1 range 10.3.8.0/21 type inter-area
 area 1 range 10.3.32.0/19 type inter-area
```

```
Core1# show ip ospf lsdb
OSPF R-ID (10.3.0.1)(PID 1 VRF default)
=======================================
Inter-area Summary LSAs (Area 0.0.0.0)
---------------------------------------
LSID              ADV Router      Age
---------------------------------------
10.3.8.0          10.3.0.2        55
10.3.8.0          10.3.0.3        19
10.3.32.0         10.3.0.2        41
10.3.32.0         10.3.0.3        10
```

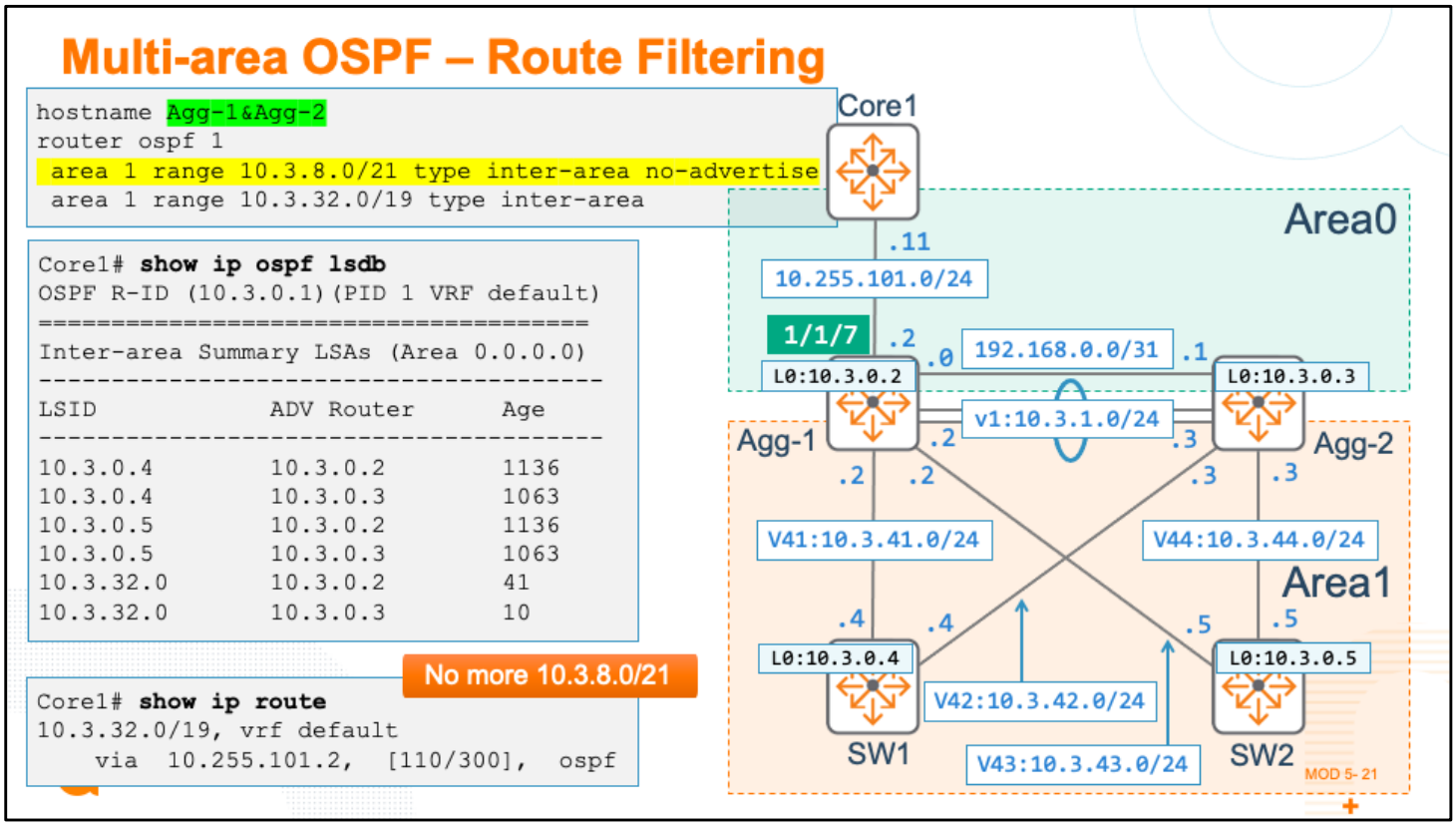
```
Core1# show ip route
10.3.8.0/21, vrf default
    via  10.255.101.2,  [110/300],  ospf
10.3.32.0/19, vrf default
    via  10.255.101.2,  [110/300],  ospf
<<Omitted Output>>
```



Suppose that you have configured route summarization on the two ABRs, as shown in the figure. Now the output of show ip ospf lsdb proves that your summarization was a success. Only your summaries appear in the LSDB, from each ABR.

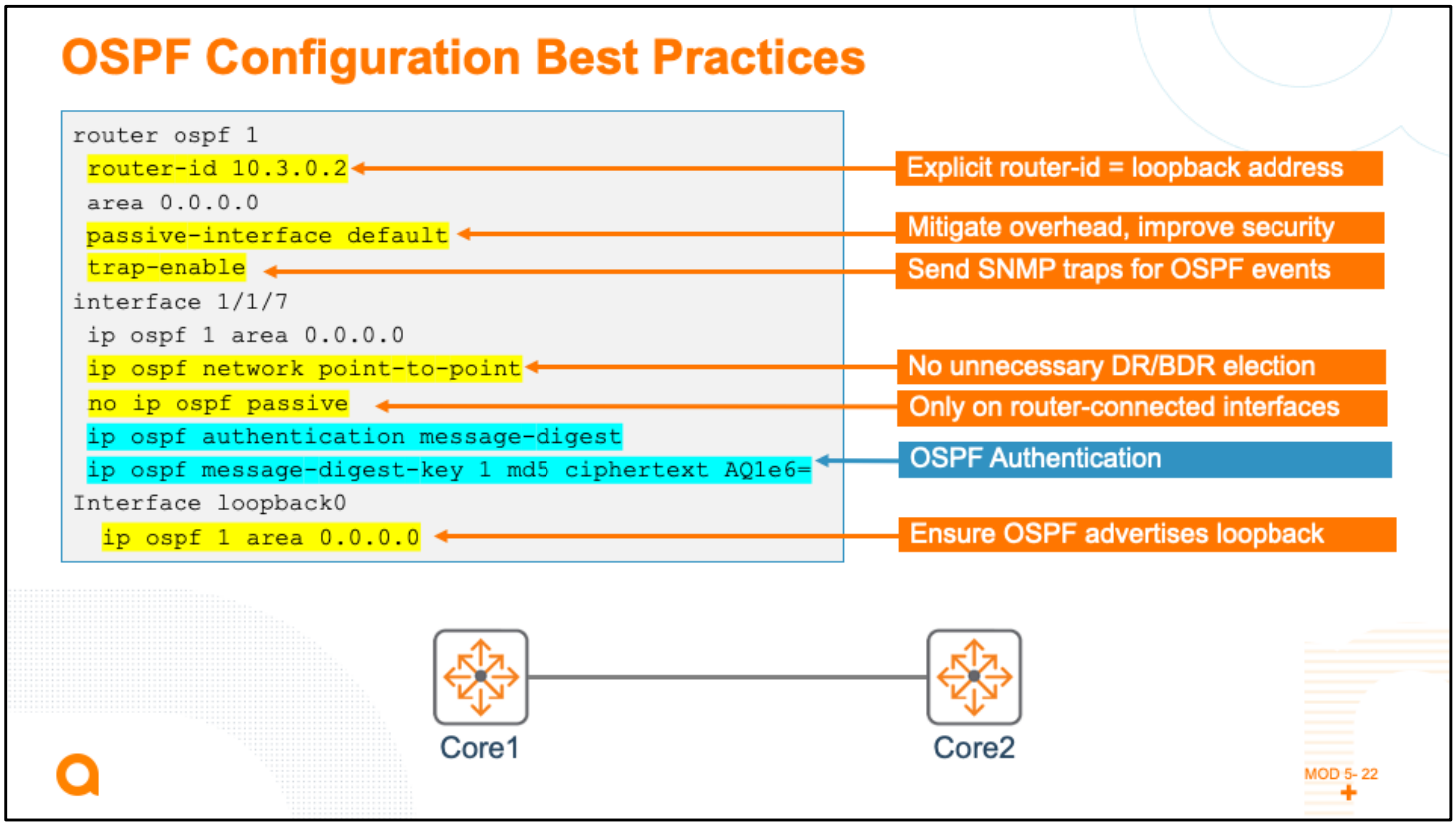
Remember, the LSDB is a list of every path, while the route table only lists best paths. Thus, the router table only shows Agg-1 (10.255.101.2) as the only best path, since Core-1 can only reach Agg-2 via Agg-1.

This looks good, but suppose you no longer want the 10.3.8.0/21 subnet to be advertised into area 0.



One easy method to prevent routes from being advertised is with the area 1 range 10.3.8.0/21 type inter-area no advertise command. The figure shows this command used on both ABRs – Agg-1 and Agg-2.

Then you access router Core1 in Area 0 to validate your efforts and…success! Neither the LSDB nor the routing table has subnet 10.3.8.0/21.

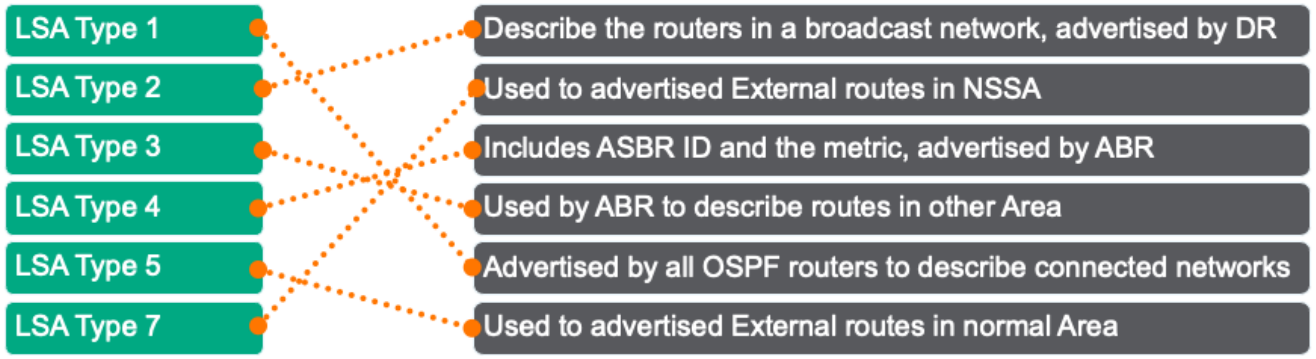


The figure summarizes OSPF configuration best practices, according to the Aruba Mobile First VRD. This is merely for your review, and to assemble these commands at a single place for easy reference. Most of them have already been covered, and so will not be explained again here.

The others are fairly self-explanatory. The trap-enable command sends SNMP traps to your SNMP server, so you have a running log of OSPF events. This can aid in OSPF troubleshooting and forensics.

Question #3 – Match Column 1 with Column 2

OSPF LSAs



Knowledge Check

MOD 5- 23

aruba
a Hewlett Packard Enterprise company

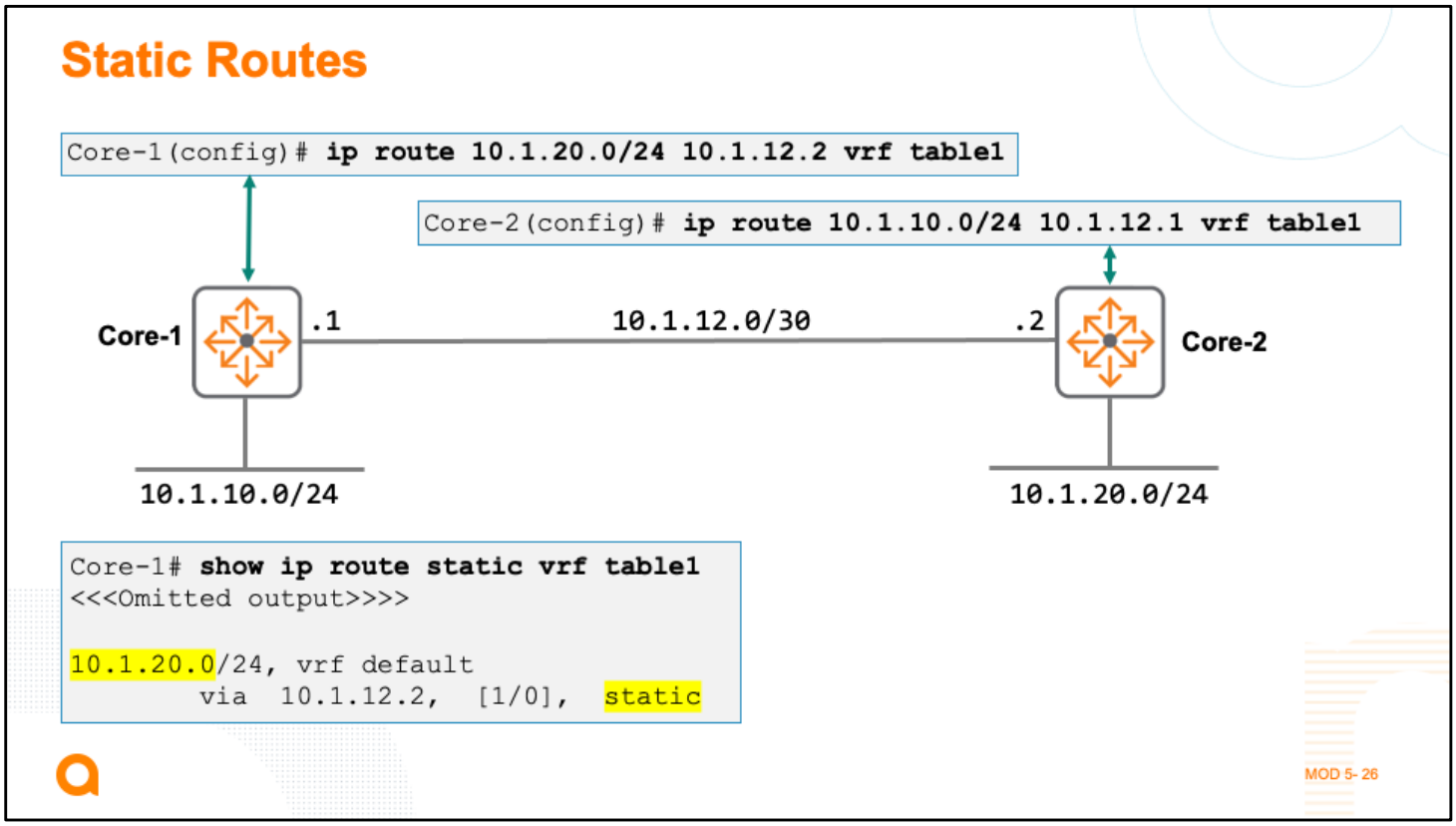
Additional Content

Module 05

aruba
a Hewlett Packard Enterprise company

Static Routing

Default, floating, BFD, tagging



Static routes

You need a static route when the destination network does not have a direct Layer 3 connection to the switch or router, and no routing protocol like OSPF are advertising that route (Remember, L2 devices are transparent to Layer 3 communications). In this case, you manually define the destination network and the next hop in the path. The next hop path must be directly connected to the device where you apply the static route.

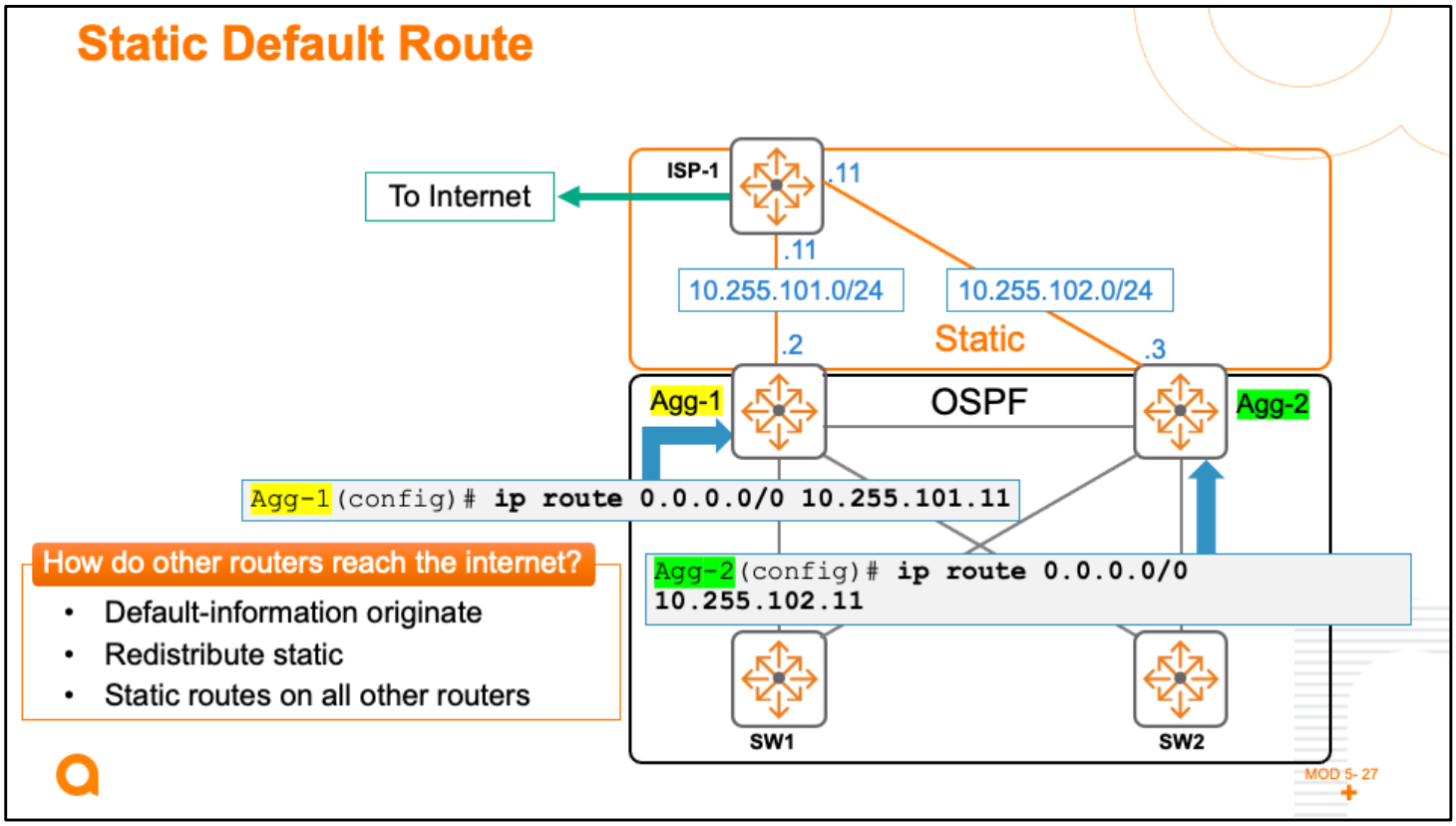
The figure shows a simple example, which configures Core-1 to reach subnet 10.1.20.0/24 via next hop 10.1.12.2, in a VRF named table1.

Core-1(config)# ip route 10.1.20.0/24 10.1.12.2 vrf table1

Core-2 switch also needs a static route to complete the path, from its perspective it only has entries in its routing table for 10.1.20.0/24 and 10.1.12.0/30 networks. You must add a new route:

Core-2(config)# ip route 10.1.10.0/24 10.1.12.1 vrf table1

To verify this entry exists on the routing table issue the show ip route vrf table1 command.

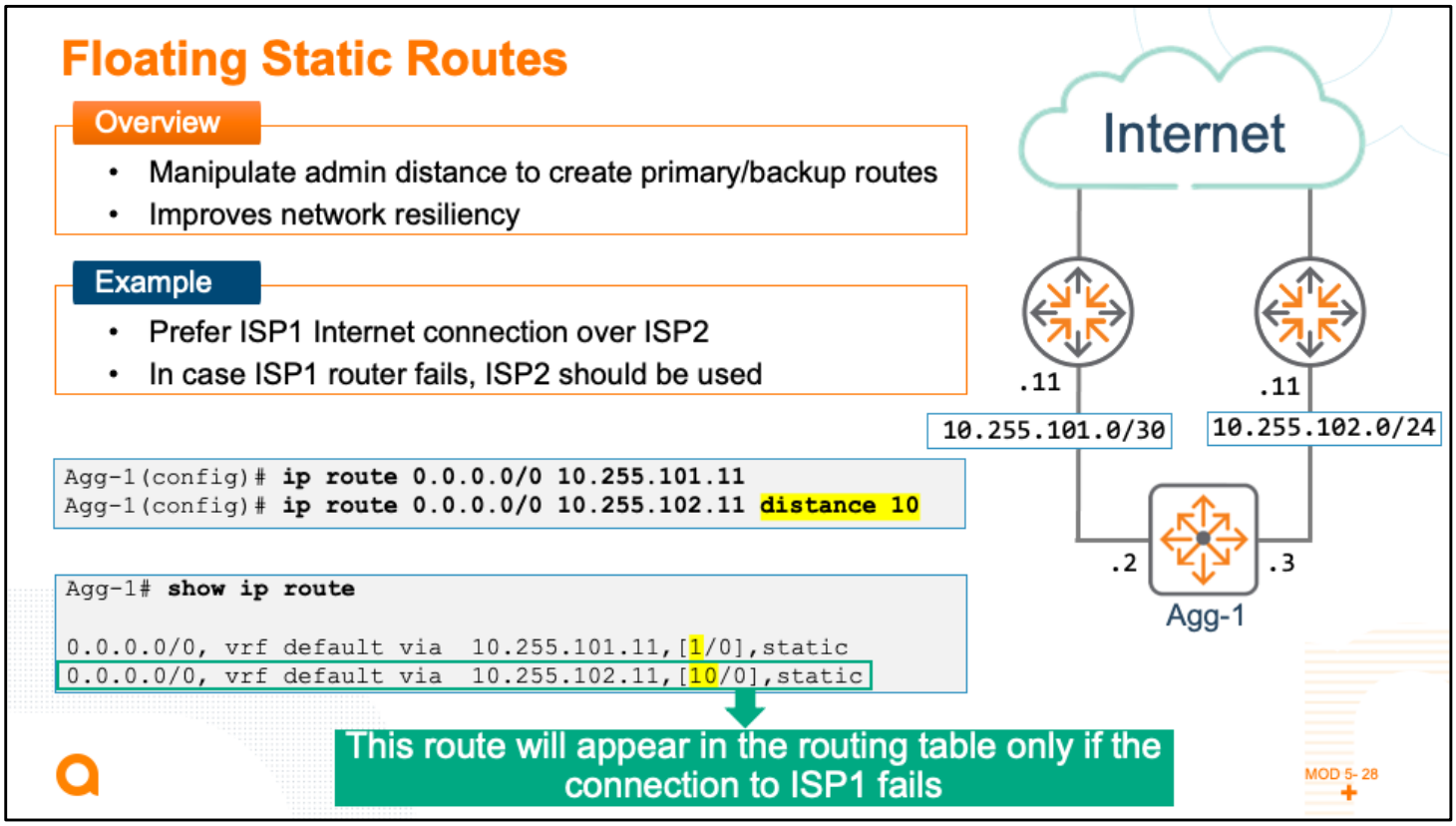


This scenario reviews the use of a static default route. You know that routers look for the longest match when selecting a path. So, Agg-1 receives a packet destined for 2.2.2.0/24, and the route table contains a route to 2.2.0.0/16, next hop 10.1.1.1, and a route to 2.2.2.0/24, next hope 10.2.2.2, the /24 entry is a more specific, longer match – that is the chosen route. The most generic route is 0.0.0.0/0 – If there is no other match, then use this route – the static default route.

In this common example, all routers in the corporation run OSPF, but you do not want to run OSPF to routers external to the corporation, such as your ISP's routers. More complex environments may use BGP to peer with one or more ISP routers, but many environments simply use a static route, as shown in the figure.

Understand that Agg-1 and Agg-2 have static routes to get to everywhere not specifically listed in the route table – the internet. However, this does not help SW1 and SW2. You could manually enter default static routes in those routers, and all other OSPF routers, but this is not scalable. A more common option is to only configure static routes on your edge routers – Agg-1 and Agg-2 in this example, and then configure the OSPF process on Agg-1 and Agg-2 with either default-information originate (always) or redistribute static (route-map).

You will learn about redistribution later in this module.



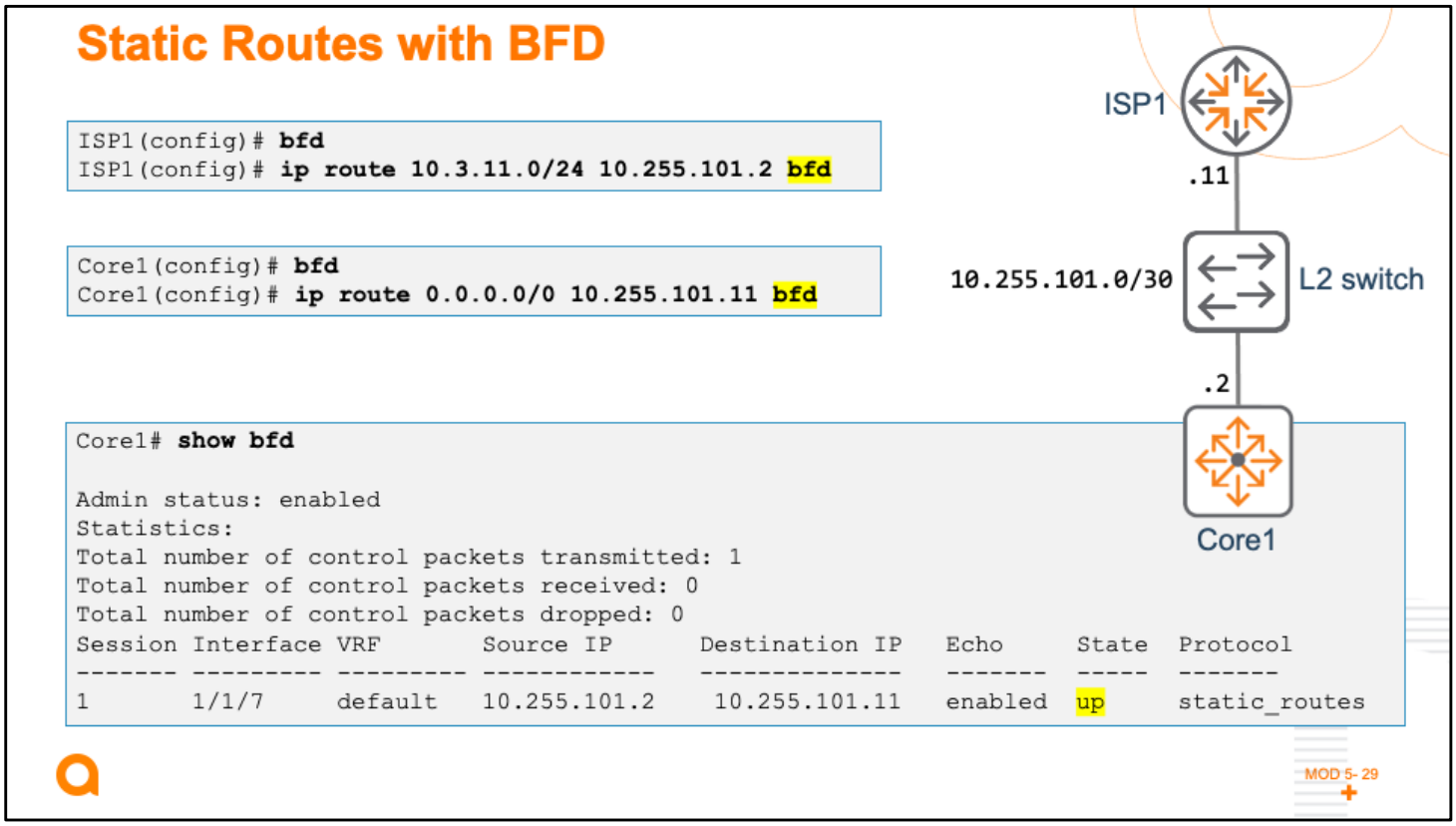
One example of manual route path adjustment is to manipulate administrative distance to create primary and backup routes, to improve network resiliency.

Imagine that you have two Internet connections that use different ISPs. The first connection has a higher bandwidth than the second one and therefore the administrator want to use it as the Primary connection. The secondary ISP connection will be a backup, waiting to be used in case the primary link fails.

To do this, configure two static routes. One to the “all routes” destination of 0.0.0.0 with ISP-1 as the next hop. You let its default administrative distance remain at 0.

Configure a second static route to the “all routes” destination of 0.0.0.0 with ISP-2 as the next hop, changing the administrative distance to 10, as shown in the figure.

Since the ISP-1 route has a lower administrative distance it will be used exclusively. The ISP-2 path shall only be used if the primary link fails. You can verify this behavior using the show ip route command.



Fast link detection is critical to achieve a fast reconvergence for the networking devices, when you have two directly connected devices the link failure is rapidly detected by any of them since the interface transits from Up to Down state. However, in a situation where you have two devices non-directly connected (there is an intermediate device) a link failure detection is not fast, routing protocols such as OSPF or BGP rely on the hello or keepalive mechanisms to verify the reachability to the peer, in some cases it could take some minutes for the reconvergence. A major problem arises with static routes since in this case routing devices do not use any mechanism to detect if the next hop is still reachable, as long as the egress interface is in Up state the routing device will keep forwarding traffic out through it . Fortunately for us Bidirectional Forwarding Detection (BFD) exists.

BFD is a general-purpose, standard, medium- and protocol-independent fast failure detection mechanism. BFD establishes a session between two network devices to detect failures on the bidirectional forwarding paths between the devices and provide services for upper-layer protocols. This option is especially useful when there is some intermediary device between the two network devices.

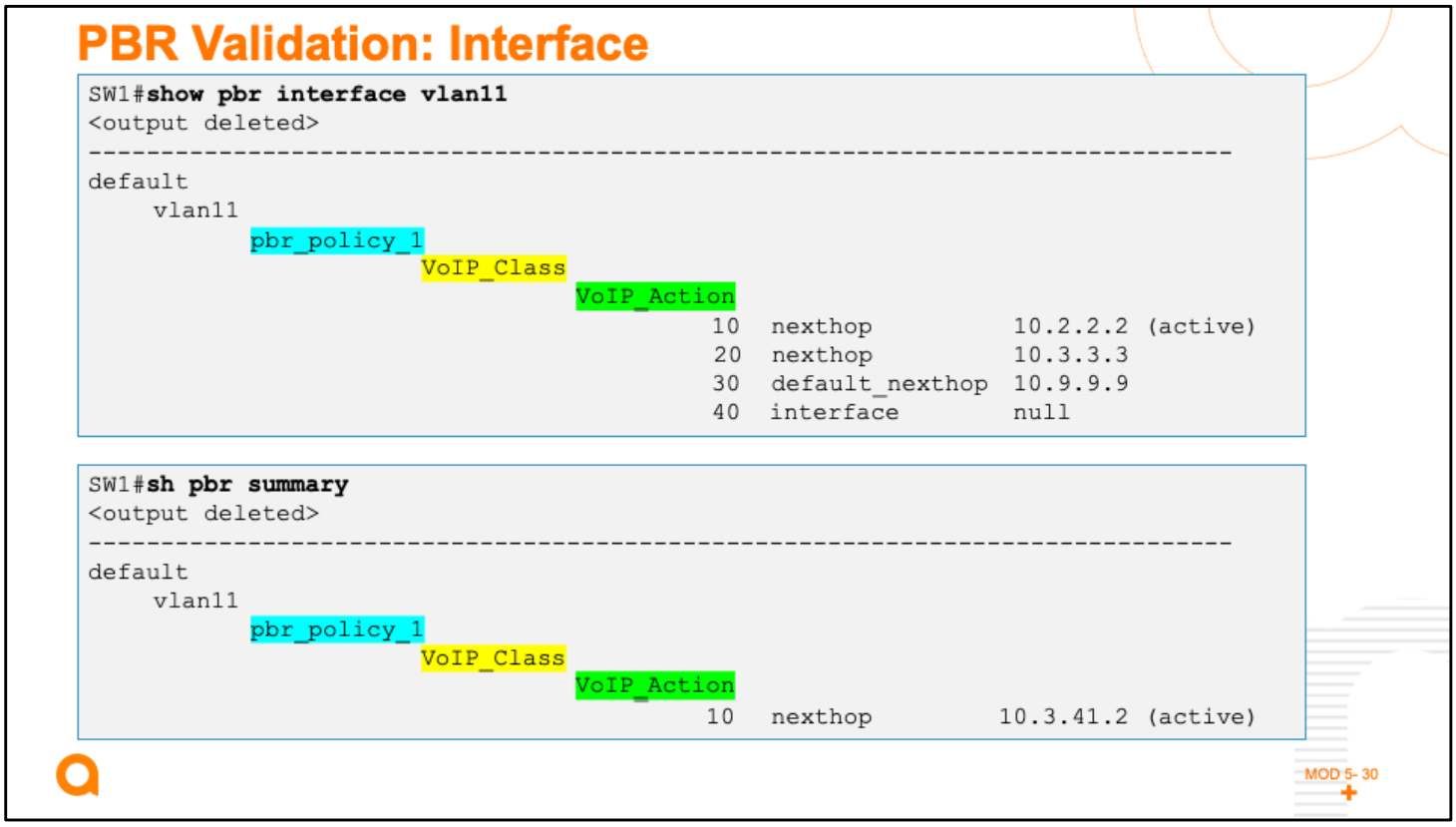
To configure BFD in AOS-CX you must first globally enable it and then you must specify the use of BFD in the static route:

bdf

ip route <destination> <next-hop> brd

Use show bfd to validate that BFD is operating as intended. For more detail, use show bfd interface <interface-id>.

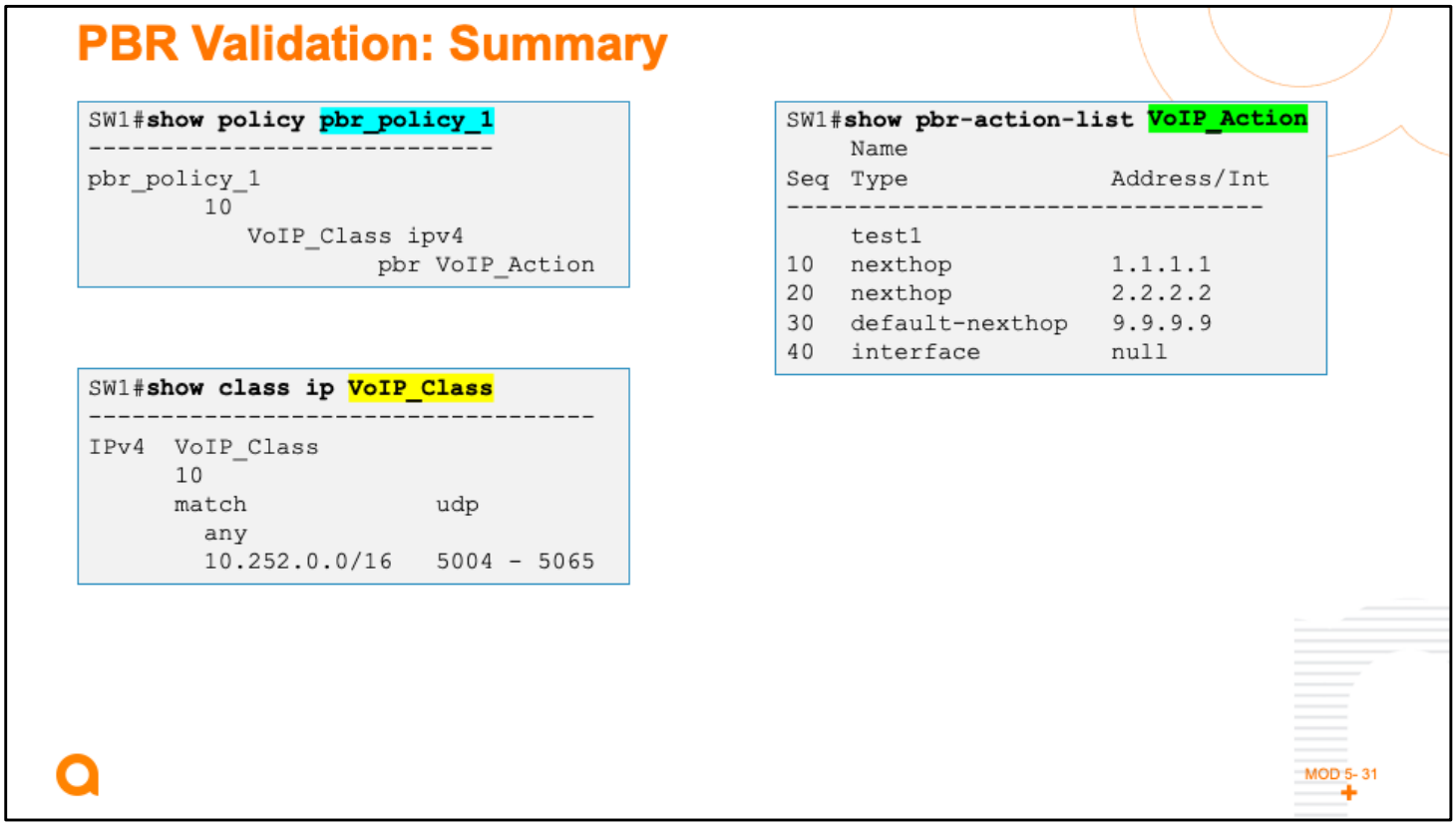
If possible is recommended to enable BFD on both devices, this will ensure a fast link detection on both devices then both devices can start the reconvergence process. In this example we enabled BFD as well in the ISP1 router.



Use show pbr interface vlan11, for example, to see a nice overview of how all PBR pieces fit together in a kind of “configuration chain”. You see that pbr_policy_1 is applied to interface VLAN11. That policy will apply the PBR actions defined in VoIP_Action to all traffic in class VoIP_Class. And you can see the actions listed, and the top action is currently active – it’s a valid route, and so the other actions will not be used.

It can be helpful to compare this output to the appropriate portions of your running configuration, as shown in the figure.

If you were not aware of if or how PBR is configured, show pbr summary can be helpful – it shows all PBR “configuration chains”. In the example switch SW1 only has one configuration chain.



You now know the names of the policy, class, and action. Use this information to enter the appropriate commands to see more detail on each PBR component:

show policy pbr_policy_1

show class ip VoIP_Class

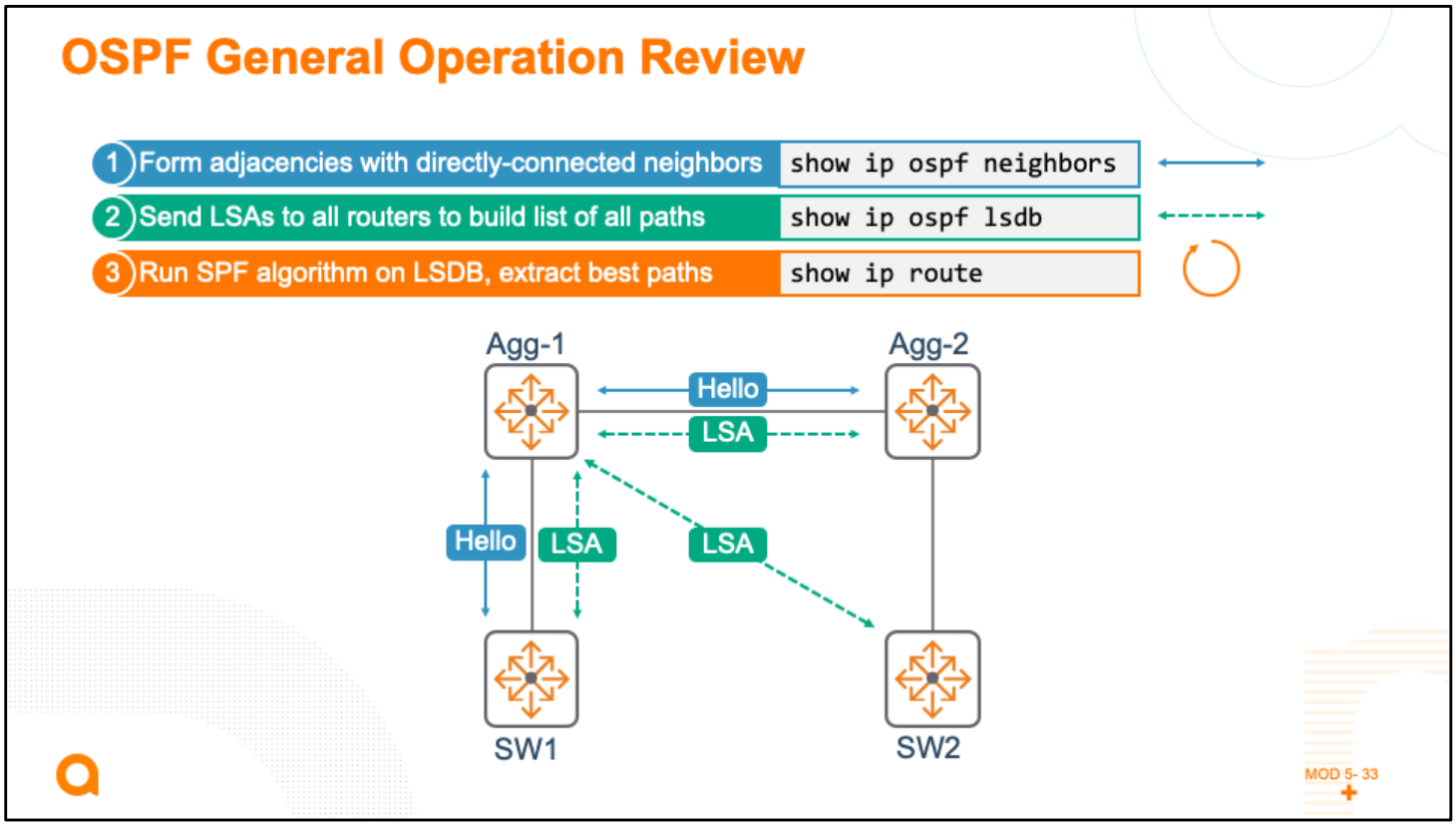
show pbr-action-list VoIP_Action

For more information, see the Aruba 10.xx IP Routing Guide.

OSPF Single Area

Operation, adjacencies, functional assessment

aruba
a Hewlett Packard Enterprise company



Remember that a router's job is to learn about every path, choose the best paths, and then route packets along those paths. To do this, OSPF uses three key processes, to build three important tables. Each table is built, maintained, and stored in local memory on each router. The more deeply you understand each phase and its associated table, the better you will be at designing, deploying, and troubleshooting OSPF networks. These processes are shown in the figure, and listed below:

Phase 1: Form adjacencies with directly connected neighbors

Each OSPF router sends hello packets to its directly-connected OSPF neighbors to establish an adjacency. Neighbors move through several states before they form a full adjacency. These states can be used to help diagnose routing issues.

Down state: interfaces are not functional – perhaps due to being administratively down, or physically disconnected

INIT state: Peer sent a hello but has yet to receive a hello response from its neighbor. If an interface is stuck in the INIT state, its neighbor interface may not be configured for OSPF, and so does not send a response.

2-WAY state: Peer has received a hello from its neighbor – two-way communications are established.

EXSTART state: Peers begin to exchange Database Descriptor packets

EXCHANGE state: peers exchange summary information from their Link State Database (LSDB).

LOADING state: peers request and receive Link State Update (LSU) packets for any learned routes that they do not already have complete information for

FULL state: Once peers have exchanged all LSDB information, and the LSDBs are identical on each device, the routers are in a FULL state.

This information is available in the Neighbor table, which you can view with the command show ip ospf neighbors. The Aruba Mobile First Campus Validated Reference Design (VRD) Guide suggests that an OSPF speaker should have no more than 12 adjacencies. See note below.

Phase 2: Build a topology database - also called a Link State Database (LSDB), OSPF enabled routers exchange network prefixes and parameters. This information is stored in the LSDB table. While Hello packets are only exchanged between directly-connected OSPF neighbors, every router receives Link State Advertisements (LSA) from every other router in the same OSPF area. You can see this topology or link state database with the command show ip ospf lsdb. The LSDB is essentially a list of every destination, and every path to each destination.

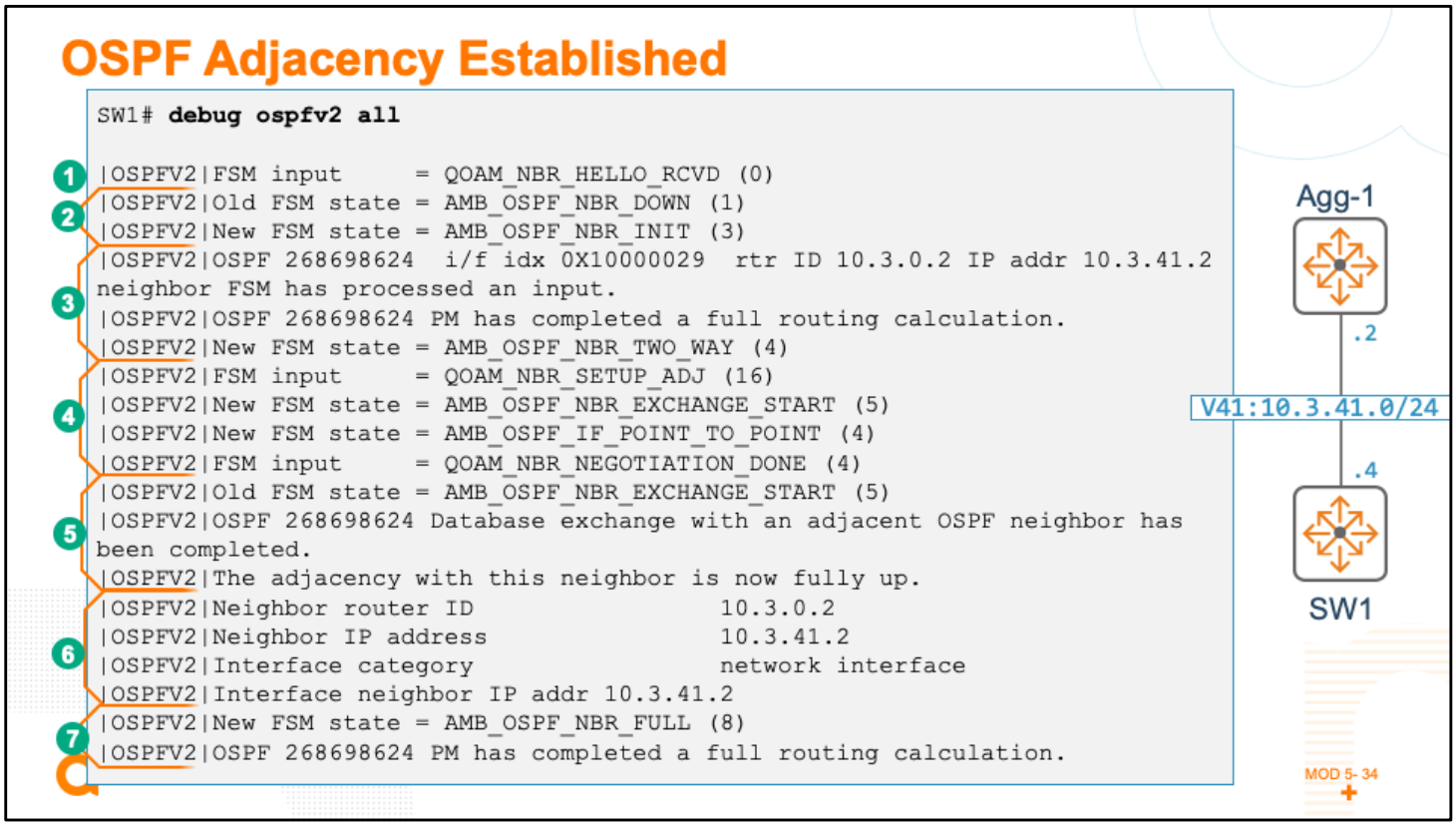
Phase 3: Build the route table – Now that each router in an area has an identical LSDB, they run the Shortest Path First (SPF) algorithm, also known as Dijkstra's Algorithm. This algorithm runs on the list of every path to build a list of best paths – the route table. You can view the route table with the command show ip route.

The following pages help you to explore how to use these commands to troubleshoot OSPF networks.

Note: As it relates to OSPF performance and design best practices, you may find this excerpt from Aruba’s Mobile First VRD guide:

A high-performing OSPF network design would have the fewest adjacencies possible and all OSPF speakers would be of similar performance capabilities. If your network will have OSPF speakers of various performance capabilities, multiple OSPF areas may be required. This may also be a driver to use BGP to build connectivity between OSPF domains. In Campus networks where the links between devices are often more reliable than WAN circuits, it is less

likely to experience the same potential volume and frequency of OSPF events. Thus, optimizing OSPF for a campus design can have somewhat less rigid requirements. In the spirit of ‘simple is best’ it is still a leading practice to work to conform your Campus OSPF design to the same principles and practices as an OSPF WAN design. To that end, we recommend a Campus OSPF speaker have less than 12 adjacencies.



The figure shows partial output from the command debug ospfv2 all, capturing properly deployed neighbors establishing an initial connection. The complete output would occupy over two full pages, and so the highlights are shown here. The output is from the SW1 device.

It is a valuable exercise to use debug commands like this in a lab or test network that is fully functional, for several reasons.

First, you are validating your understanding of OSPF theory of operation. You learned that OSPF neighbors go through several states before becoming fully adjacent, and you see a highlight of these states in the figure.

Second, you become familiar with the commands themselves. You should know which show and debug commands are effective in which situations, and what information they can show you. You are thus better prepared to quickly alleviate outages.

Third, you gain skill in interpreting the output under controlled conditions, which makes it easier to ensure that your interpretations are correct. You follow the output along and learn where to look for the most important information, and when to quickly skim over less vital output to save time.

Let's analyze the partial output in the figure by referencing the numbers next to each section of output.

The interfaces connecting Agg-1 and SW1 are brought up, and so SW1 receives an OSPF hello packet from Agg-1

SW1 informs you that the original state of this OSPF neighbor relationship was “DOWN”, and now it has changed to “INIT”

SW1 now is processing this input from Router ID 10.3.0.2, with physical IP address 10.3.41.2. It completes a route calculation and moves toward a TWO-WAY state.

Now the adjacency (ADJ) is being setup by performing a neighbor exchange. This link is type “Point-to-point”, and so DR/BDRs shall not be elected. The negotiation completes.

Now the neighbors begin to exchange OSPF database information, and the neighbor relationship is more fully established.

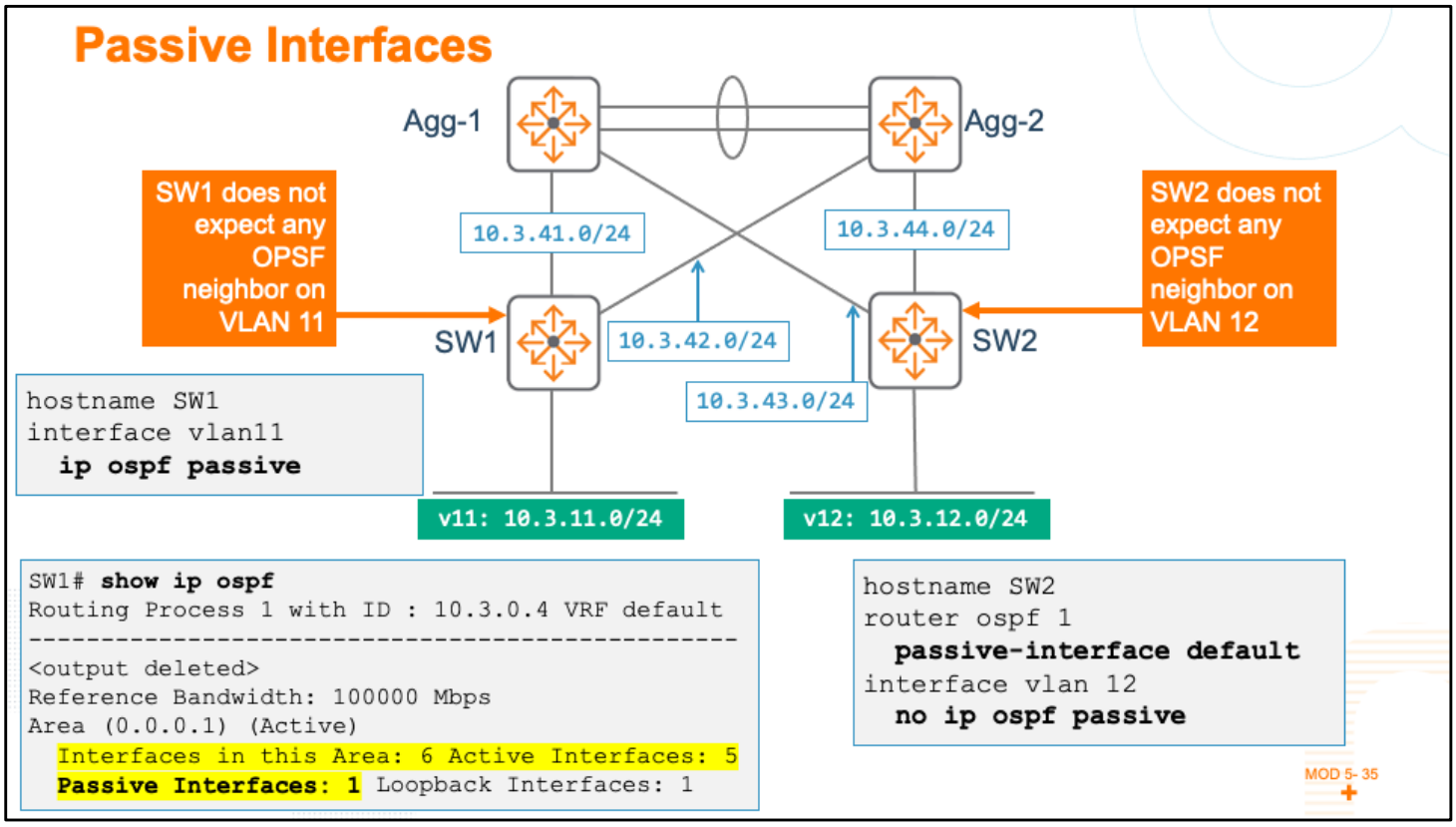
The information for this neighbor is displayed.

The message that the OSPF neighbor is in a FULL state is displayed, and routing calculations are complete.

Again, the output shown here is highly summarized to focus your attention on certain key aspects of this exchange. You are encouraged to repeat experiments like this in a lab setting so you gain familiarity with the entire output.

For example, you could get two neighbors successfully established, perhaps like the example here between Agg-1 and SW1. Then shut down the Agg-1 interface VLAN41.

Then go to SW1 and use clear debug buffer to make it easy to see only the output you are interested in. Then use debug ospfv2 all to enable debugging. Then, access Agg-1 and use no shutdown on interface VLAN41. Back on SW1, use show debug buffer to see the debug output.



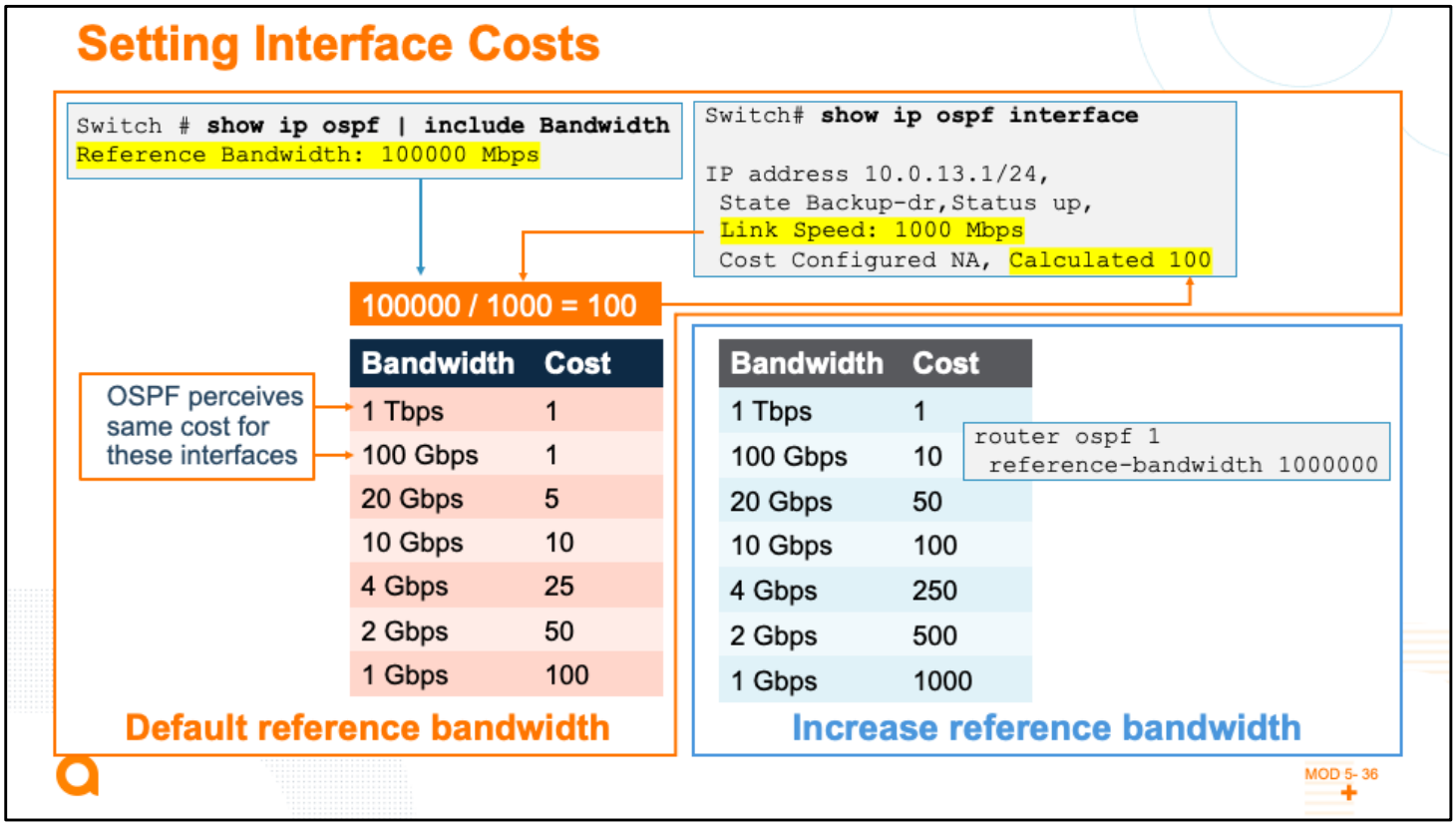
When a specific subnet needs to be advertised into the OSPF domain you must enable the protocol in the interface that is facing that subnet using the ip ospf <process> area <area-id> command. This command as well starts the process to create an OSPF neighbor relationship on this interface by sending Hello messages. This behavior is expected if the interface is connected to another OSPF router like VLAN 41 and 42 for SW1 switch, however for VLAN11 there will be no neighbor relationship. To save a bit of unnecessary overhead, and mainly to improve security, you can configure these interfaces as passive. The routers will still talk about those subnets to other routers, but they will no longer talk on those networks.

There are two methods to configure OSPF passive interfaces, and both methods provide the exact same result. You choose the method that is easiest in your situation.

If a router has ten interfaces, and only one of those interfaces is passive, you simply access config-interface mode for that interface and issue the command ip ospf passive. This scenario is represented with SW1.

If a router has ten interfaces, and all of them should be passive except one or two, you can access router config mode and issue the command passive-interface default. This makes all interfaces passive. Then go to the one or two interfaces that should not be passive and use no ip ospf passive. This scenario is represented with SW2

The Aruba Mobile First VRD best practice is to use the passive-interface default option, simply to ensure that no links waste bandwidth unless you specifically decide to enable it at the interface level. This also improves security – an inside attacker on an end-user segment could not setup an OSPF process and form an adjacency with your routers.



OSPF path selection is based on link cost, which is based on the reference bandwidth, which defaults to 100000. This is shown in the figure, with the output of the show ip ospf command. AOS-CX devices divide this reference bandwidth by interface link speed in Mbps to determine cost. So, the default reference of 100000 Mbps / 1000 Mbps (1 Gbps) = 100. This is proven in the output of show ip ospf interface.

One problem with this default is shown in the figure. The default reference of 100000 Mbps / 100000 Mbps (100 Gbps) = 1. OSPF cannot use fractions, and so all interfaces faster than 100 Gbps will also have a cost of 1. Thus, OSPF will perceive a 100 Gbps path and a 1 Tbps path as being equal cost. If you have no links faster than 100 Gbps, this is not an issue.

If you DO have links faster than 100 Gbps, then you should change the reference bandwidth to a higher value. In the figure, you have changed the reference bandwidth to 1,000,000. Thus, 1 Tbps links have a cost of 1, while 100 Gbps links have a cost of 10, and so on - as shown in the figure. OSPF can now distinguish between these fast links.

Another issue is that by default, AOS-CX VLAN interfaces have a link speed of 1000 Mbps (1 Gbps), and so have a cost of 100. Because every VLAN interface has the same cost, route selection becomes a matter of hop count, rather than true lowest cost path, if you are using layer 3 VLAN interfaces. If you modify the reference bandwidth to 1,000,000 as shown, then all VLAN interfaces would have a cost of 1000.

Changing the reference bandwidth does not mitigate the issue of VLAN interfaces having the same cost. If this is an issue, you can change the individual cost values on a per-interface basis:

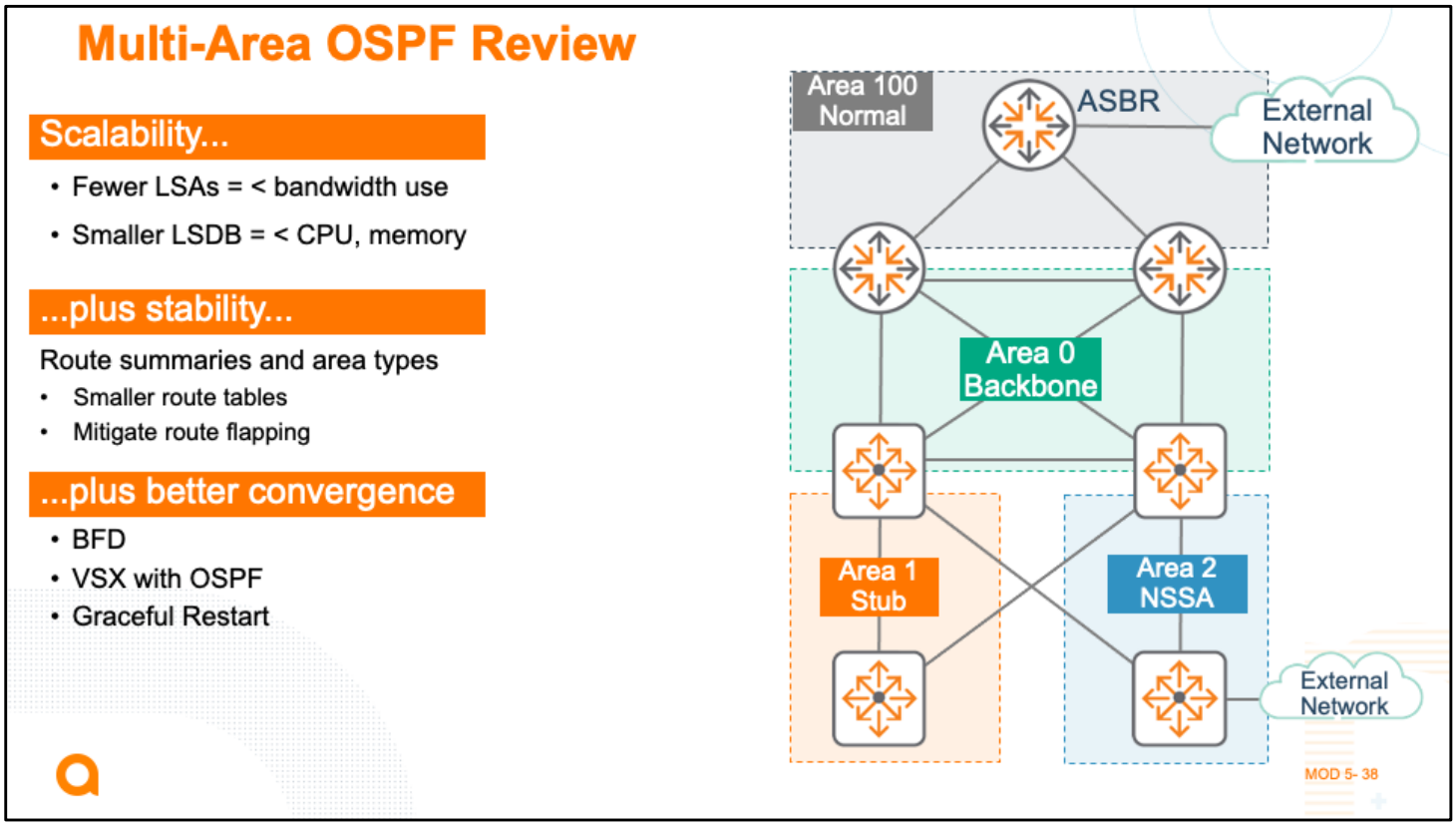
- Interface vlan41
- ip ospf cost <1-65535>

- As you can see in the generic example above, you can set each individual interface to an OSPF cost value of your choice – any value between 1 and 65535 inclusive.

OSPF Multi-Area

LSAs, ABRs, Aggregation / filtering / summarization, route tuning

aruba
a Hewlett Packard Enterprise company



With a single area, every OSPF router must receive LSAs from every other router. Every router's LSDB describes the entire network – every router, every link, every path. This is great for small to medium-sized networks, but large networks may need the scalability benefits of a multi-area OSPF deployment.

Area 0 is the backbone area, and all other areas must connect directly to this backbone area. Most routers have all interfaces in a single area - they are regular OSPF internal routers.

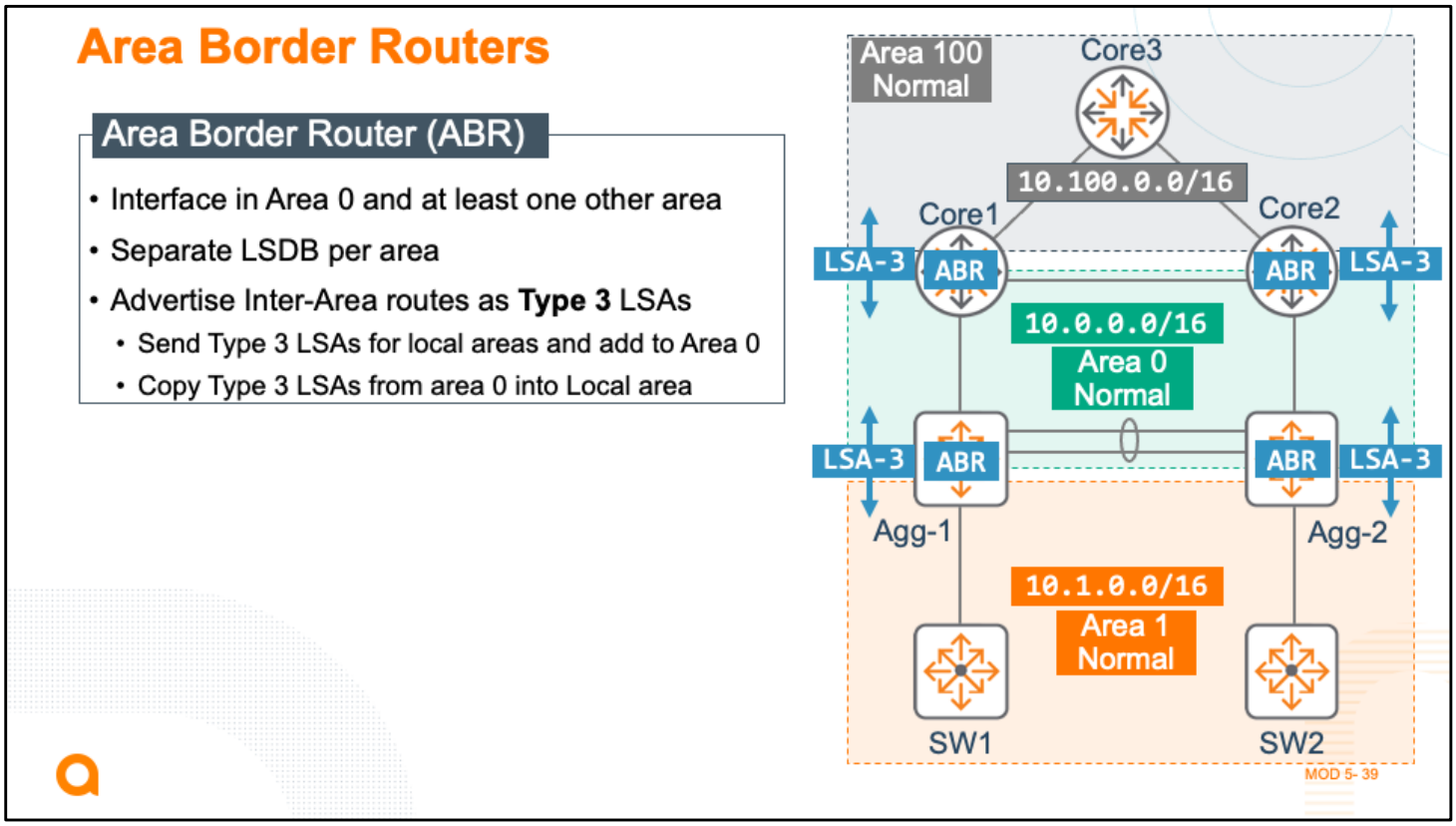
Routers with some interfaces in Area 0 and some interface in another Area are Area Border Routers (ABR).

This sets you up for great scalability. All internal routers must only communicate with other routers in their area. ABRs have a bit more to do. They must communicate with all routers in all connected areas – still much better than having to communicate with all routers. This greatly reduces the size of each router's LSDB, and the number of transmitted LSAs – lower memory, CPU, and bandwidth utilization.

Even better, OSPF ABRs support route summarization. So, in addition to reducing LSDB size, you can also reduce the size of the route table, further lowering resource utilization. You can also configure specialized Area Types, such as a Stub Area, to further reduce route table size and speed convergence. You also improve network stability, because a flapping or unstable segment in Area 1 has no effect on other areas.

These areas and all the routers they contain form your company's Autonomous System (AS). You connect your AS to the internet and other external systems via one or more Autonomous System Boundary Routers (ASBR). You can configure various features on the ASBRs to control what you advertise outside your AS, and what you accept inbound.

You also have features to improve convergence, including Bi-Directional Forwarding Detection (BFD), VSX operations with OSPF, and OSPF graceful restart.



You learned how each OSPF area defines an isolated region beyond which Type 1 and Type 2 LSAs and shortest path topologies do not extend. However, some routing information must flow between the areas. Area Border Routers (ABRs) are responsible for forwarding this information.

When you design a multi-area network, you must understand area 0's special role. This area is also called the backbone area because every other area must connect to it. For example, area 1 reaches area 100 through area 0. Area 0 includes OSPF routers with a special role; these routers are called area border routers (ABRs). Each ABR has interfaces in area 0, which is required, and at least one other area (Area 0 can also have internal routers that are only in area 0.)

As a member of multiple areas, the ABR maintains multiple LSDBs, one for each area. In this figure we have four ABRs, Agg-1, Agg-2, Core1 and Core2.

Type 3 LSAs generated in area 0

The ABR creates a Type 3, or summary LSA, for each network in area 1. It places these Type 3 LSAs in its LSDB for area 0. The ABR can then advertise its routes to area 1 networks to other routers in area 0.

For example, Agg-1 has an inter-area route to 10.1.20.0/24. It learned about this network from SW1's Type 1 LSA, contained in its area 1 LSDB. From that LSDB, Agg-1 was able to generate an intra-area route to 10.1.20.0/24 . Agg-1 generates a Type 3 LSA for 10.1.20.0/24 and places the Type 3 LSA in its area 0 LSDB. Other area 0 routers, such as Core1, receive Agg-1's Type 3 LSA and add it to their area 0 LSDB. They can then calculate an inter-area route to 10.1.20.0/24.

In addition to the network address, the Type 3 LSA includes a metric, which enables routers to select the best inter-area route when multiple ABRs advertise a Type 3 LSA with the same network ID.

Type 3 LSAs generated in non-backbone areas

Agg-1 also generates Type 3 LSAs to advertise into non-backbone areas.

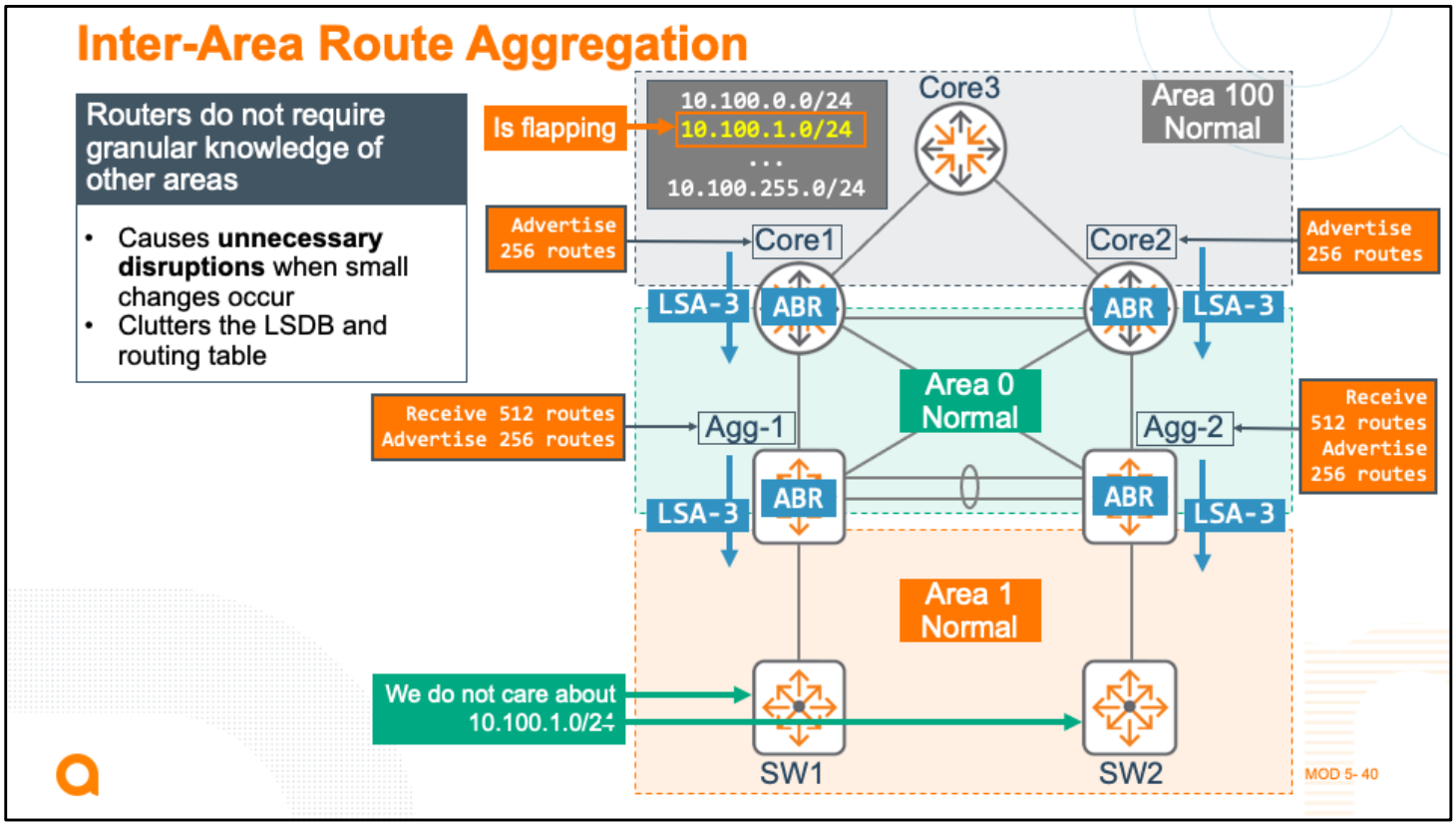
First, just as Agg-1 advertised a Type 3 LSA for area 1 to routers in area 0, the ABR advertises a Type 3 LSA for each area 0 network to routers in area 1. In this example a similar job is performed by Agg-2.

The ABR also advertises a Type 3 LSA for each inter-area route that it has learned from Type 3 LSAs advertised by other ABRs in area 0. For example, Core1 and Core2 have generated Type 3 LSAs for each network in area 100 and advertised them in area 0. Using these LSAs, Agg-1 and Agg-2 have calculated an inter-area route to each area 100 network. Now Agg-1 and Agg-2 generate Type 3 LSAs for these routes and advertises them in area 1, using the metric in the routing table. (It does not advertise these LSAs in area 0 because the inter-area routes' next hops are in area 0. The original ABR, which always resides in area 0, has already advertised these LSAs in area 0.)

If Agg-1 connected to another area, such as area 3, it would place Type 3 LSAs for each area 3 intra-area route in the area 0 LSDB and the area 1 LSDB. In other words, for each area, an ABR creates Type 3 LSAs with routes to other areas to which the ABR connects. Remember, though, that the inter-area routes always pass through area 0 even if the same ABR supports multiple non-backbone areas. That is, you cannot configure an OSPF router with area 1 and area 3 and expect the router to advertise area 3 networks in area 1 and area 1 networks in area 3. The router must have interfaces in

area1, area 3, and area 0.

The figure shows how internal routing switches SW1 and SW2 in area 1 receive the Type 3 LSAs for area 0 and area 100. Note that, although area 1 routers learn routes to area 100, the Type 3 LSAs themselves do not cross area boundaries. Instead Core1, connected to area 100 and area 0, generates Type 3 LSAs for area 100 in area 0. It is the advertising router for these LSAs. Agg-1 and Agg-2 receive these LSAs in area 0 and generates new Type 3 LSAs for advertising these routes into area 1, and it is the advertising router for these new LSAs.

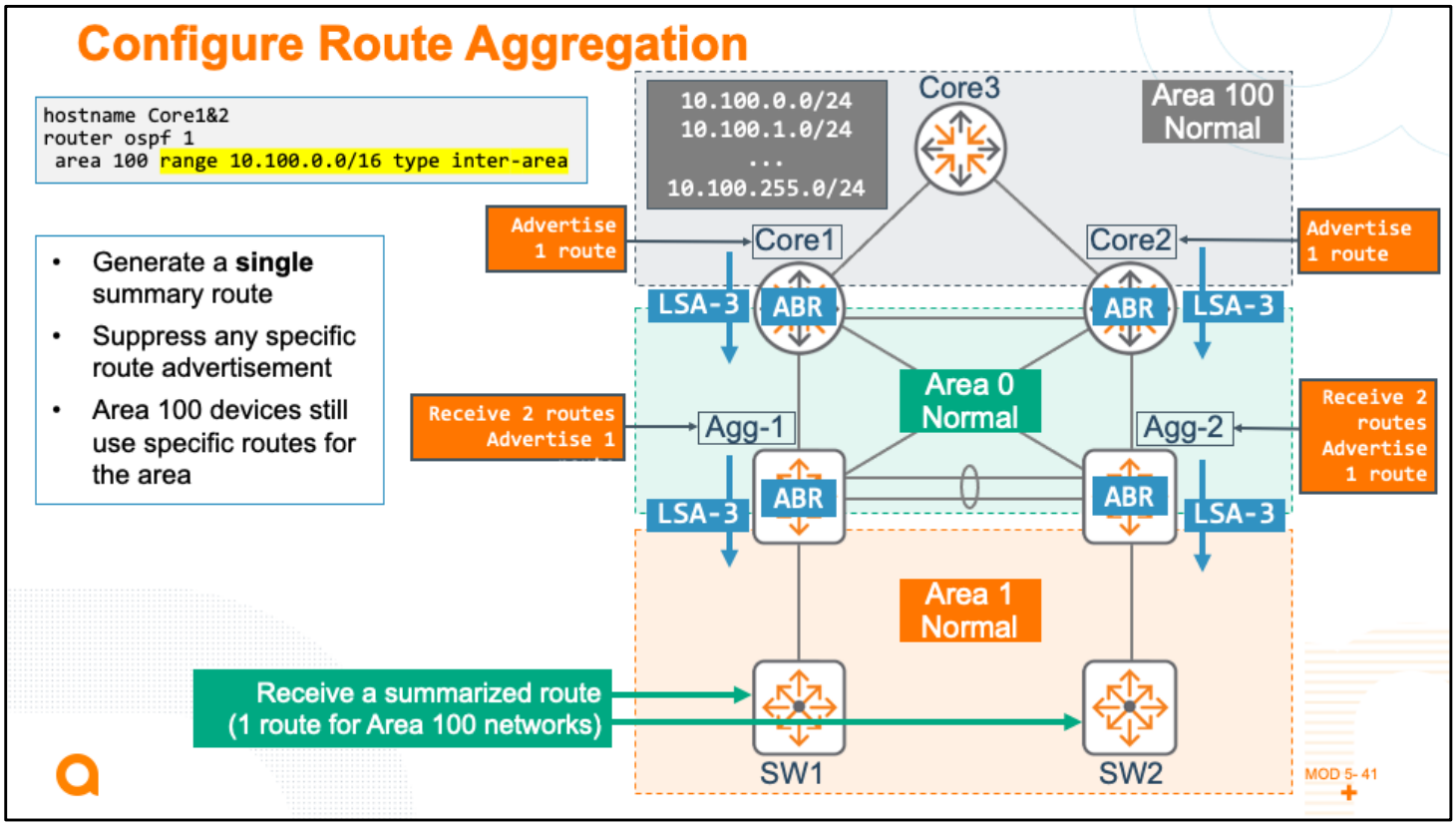


Simply dividing an AS into multiple areas begins to decrease the size of OSPF routers' LSDBs and to reduce the complexity of SPF calculations. However, this does not reduce the size of the routing table. To achieve the true benefits of multiple areas, you should further reduce the number of LSAs advertised between areas.

In the figure, Area 1 has only two paths to reach all routes in areas 0 and 100 – via the ABRs. The same goes for Area 100. It doesn't matter to Area 1 routers if Area 100's internal topology changes. However, by default, all that detail about Area 100 gets flooded into Areas 0 and 1. Further, Area 1 route tables contain 255 route entries for all the Area 2 routes – each /24 subnet is listed individually and uses the same two ABRs for all 255 subnets.

Imagine an unstable or flapping interface in area 2. Link down! Flood LSAs to all Area 100 internal routers and ABRs. ABRs flood to Areas 0 and 1. All routers must respond to this change, run SPF algorithm and forward the LSAs. Link up! Flood LSAs, run SPF. Link down!...and the entire autonomous system has elevated usage and reduced performance. What a waste! The routers in other areas only have two possible paths, via their ABRs, to reach any network in Area 100 – their pathing is not affected by a flapping interface in the corner of Area 100.

It might be better not to advertise Area 100 details to other areas. You need to aggregate the Type 3 LSAs to hide the unnecessary information and to limit network instabilities to a single area. You can do this for all areas – not just our example Area 2. Let's find out how.



You can configure an ABR to aggregate its summary routes. Rather than generate 256 Type 3 LSAs for every /24 network in Area 100, the ABRs collapse those into a single /16 LSA of 10.100.0.0/16. This is an example for Area 100. Of course, you would probably also do this for Area 0 and Area 1. You might need to use a different mask or summary value – perhaps a /20 mask instead of /16.

You configure Core1 and Core2 to summarize Area 100 routes as they are advertised into Area 0. Summarizing Area 100 has no effect on Area 100 – it reduces the number of LSAs and route table size for Areas 0 and 1. This is illustrated in the figure.

Routers in Area 0 receive this single 10.100.0.0/16 entry and advertise it to the ABRs connected to Area 1. Thus, Area 1 routers now have a single entry in their route tables instead of 256 entries for area 100.

To do this, access config-ospf mode and enter the AOS-CX command as shown in the figure, and below:

Switch(config)# router ospf 1

Switch(config-ospf-1)# area 100 range 10.100.0.0/16 type inter-area

When you enter this command, the ABR automatically suppresses the generation of individual Type 3 LSAs for subnets within the specified network. However, it continues to advertise individual Type 3 LSAs for networks that do not fall into the range. Therefore, it is recommended to assign an area contiguous IP addresses within a binary block. Then you can easily aggregate the routes.

Return to the example. The ABRs for area 1 now receive one Type 3 LSA with the aggregated summary from each of the two ABRs that connect to area 100. Each area 1 ABR selects the best LSA for the summary range and creates an inter-area route to 10.100.0.0/16 through the best next-hop.

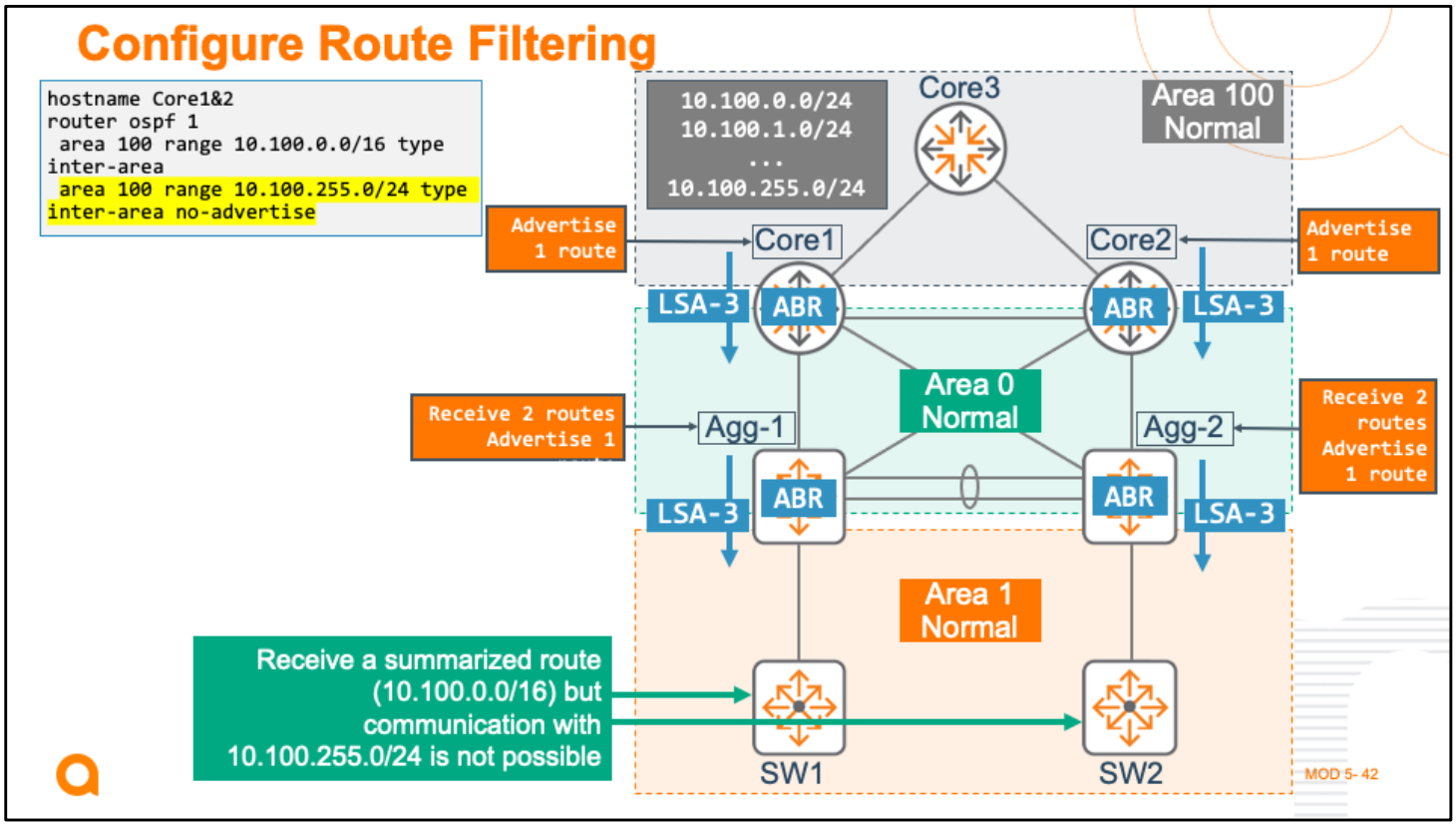
Each area 1 ABR then generates a Type 3 LSA to advertise this inter-area route to routers in area 1. The internal area 1 routers now have one LSAs for all area 100 networks, and they learn routes to only the one aggregated network (If SW1 and SW2 have redundant communication to Agg-1 and Agg-2 they will receive two summary routes if both ABRs (Agg-1 and Agg-2) advertise equal metrics and the internal routers support Equal-Cost Multi-Path [ECMP] traffic will be load balanced using both uplinks).

Note: After you configure area summaries on an area's ABR, you can see the summaries in the LSDB and routing tables for routers not connected to that area.

Usually, you want to create as few area summaries as possible. Sometimes, though, you need to create more summaries. For example, areas might have grown organically and without advance planning, creating non-contiguous ranges or ranges that do not fit neatly within a CIDR subnet. For example, area 100 might include networks in the 10.100.0.0-10.100.255.255 range as well as 10.101.20.0-10.101.23.255. Area 101 includes networks in the 10.101.0.0-10.101.19.255 range and the 10.101.30.0-10.101.39.255 range. You cannot use simply use 10.100.0.0/15 for the area 100 summary and 10.101.0.0/16 for the area 101 summary, due to the non-contiguous and overlapping ranges. Instead, you would need to use several area summaries for area 100 and several summaries for area 101.

Important: Examples throughout this module use the /16 prefix length for summary ranges to illustrate the concept in a simple way. When you create a range for your area, choose an appropriate range. For example, your area

might include 16 /24 subnets, 10.1.0.0/24 to 10.1.15.0/24. This is equal to the 10.1.0.0/20 network, and you should specify that in the area <ID> range command.



You just learned that any routes not included in your summary continue to be individually advertised. To stop ABRs from advertising certain routes, you must configure them with a non-advertised area summary.

For example, a company has a strict security policy for one of the sites that makes up OSPF area 100. The policies allow only local access to high-security subnets 10.100.255.0/24; devices in other areas can reach the rest of the subnets in area 100. Of course, a firewall can filter out prohibited traffic, but, in this case, you want to prevent any routing of traffic from other areas to the restricted networks.

Therefore, you configure two area summaries on the area 100 ABRs, one for 10.100.0.0/16 and one for 10.100.255.0/24. ABRs advertise the first but not the second.

Note that the filtering occurs at the area 100 boundary; you cannot apply the filtering to some areas and not others. For example, you cannot use this configuration to block inter-area routes to the area 100 subnets in area 1 but allow them in area 0.

The commands for configuring this example are:

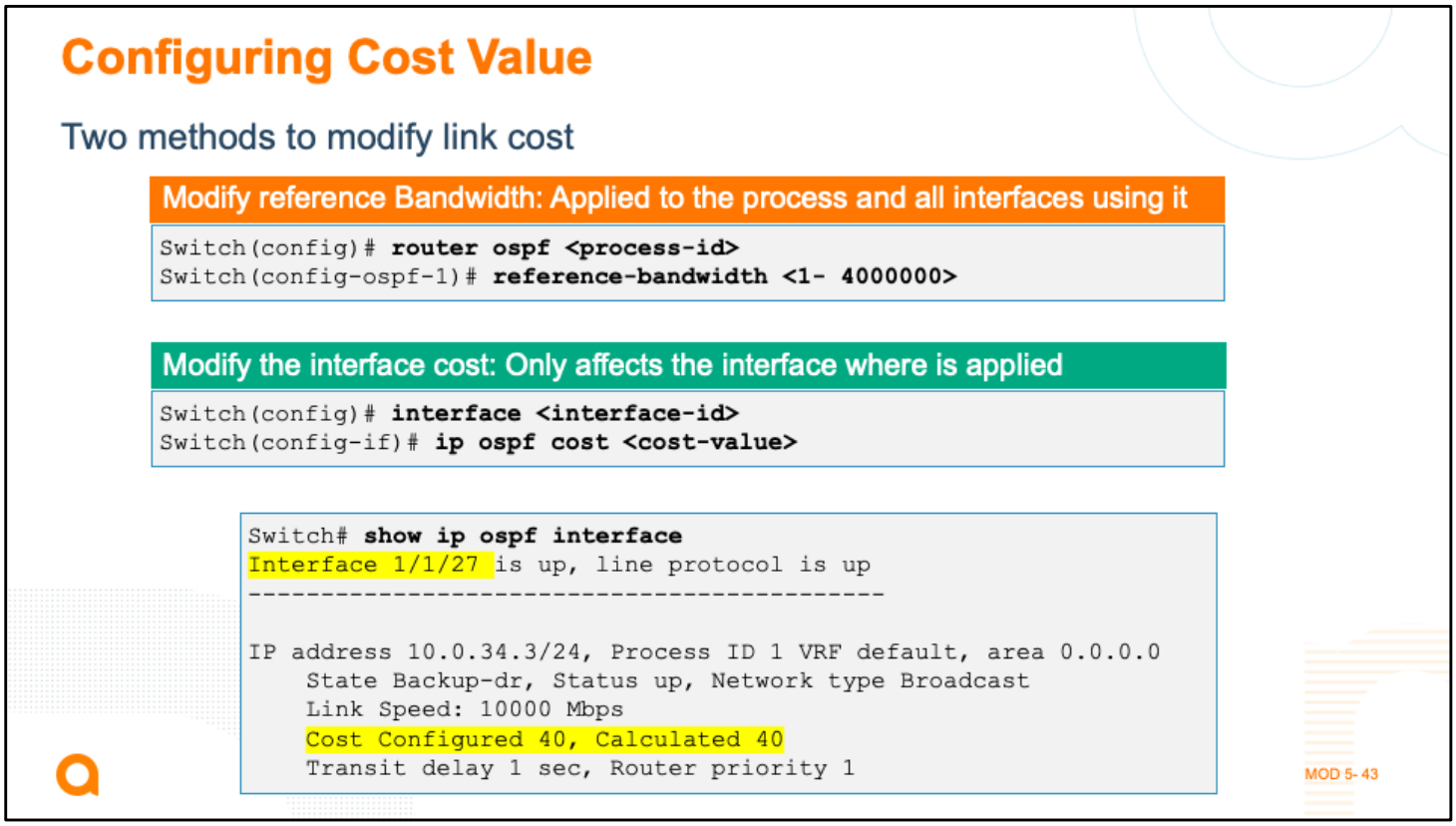
Core1(config)# router ospf 1

Core1(ospf)# area 100 range 10.100.255.0/24 type inter-area no-advertise

To enable the advertisement again, configure the same summary but without the no-advertise option:

Core1(ospf)# area 100 range 10.100.0.0/16 type inter-area

Simply removing the non-advertised summary would enable the advertisement of individual routes without aggregation.



Preferably, you want to set the interface costs globally for the OSPF process by changing the bandwidth reference with the following configuration:

switch(config)# router ospf <process-id>

switch(config-ospf-1)# reference-bandwidth <Mbps>

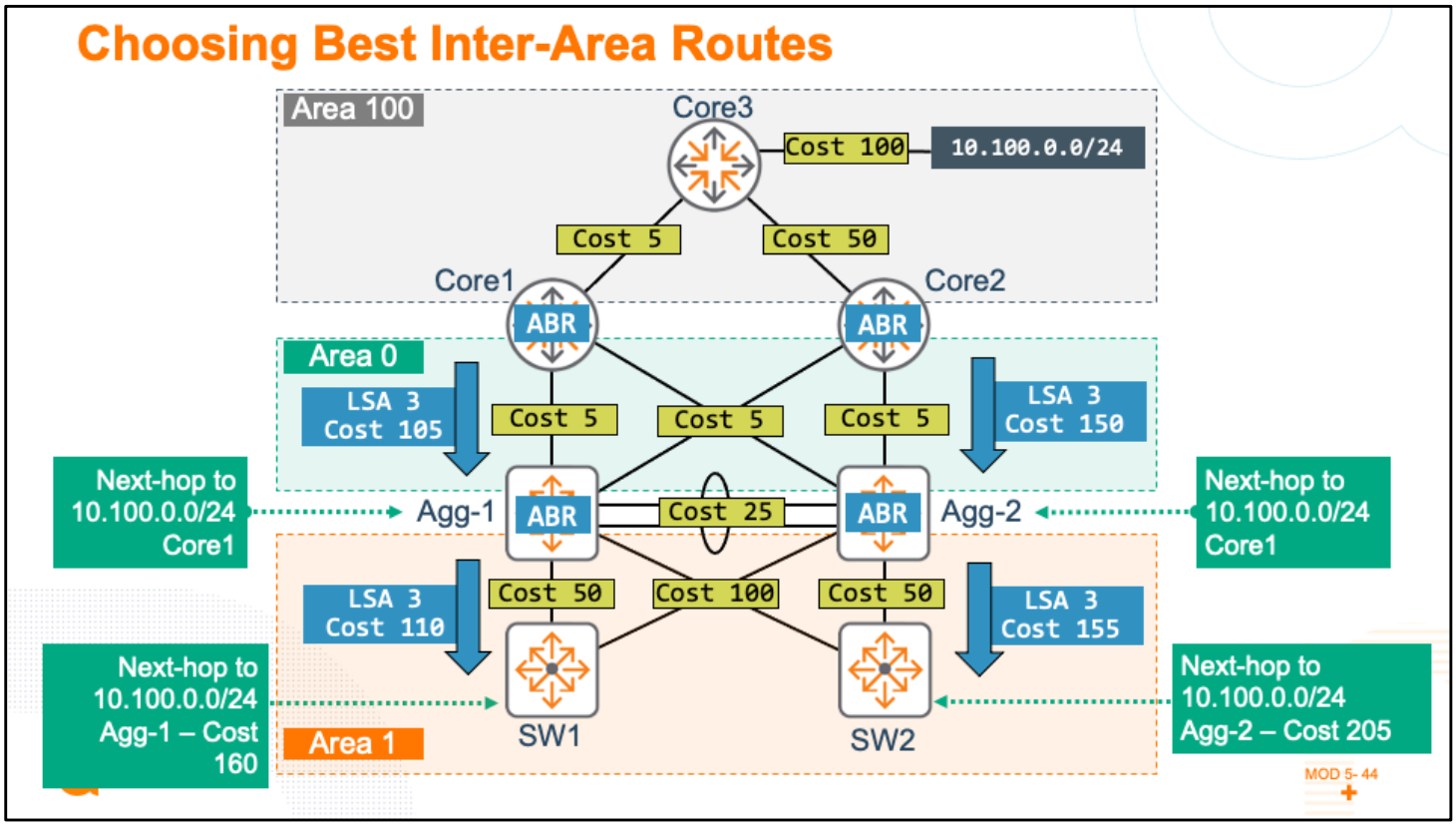
If the OSPFv2 interface cost is not explicitly set, then the cost of all the OSPFv2 interfaces is recalculated based on the reference bandwidth and link speed of the interface. For VLAN interfaces the link speed value is taken as 1 Gbps, if the OSPFv2 interface cost is not explicitly set. The bandwidth range is 1-4000000 Mbps, where the default is 100000 Mbps.

You can use this command to set the cost on a layer 3 interface, which will override the reference calculation:

Switch(config)# interface <interface-ID>

Switch(config-if)# ip ospf cost <cost>

When this command is applied the router is no longer consider the cost formula but instead simply use the manual value that was entered.



There may be times when you want multiple ABRs for an area, making it more important for you to understand inter-area route selection. Recall that the SPF algorithm only operates within areas - routers do not create a topology of the complete AS, then calculate inter-area routes from that. Instead, OSPF routers use the SPF algorithm to calculate a path to each ABR in its area. It considers the ABR as a gateway to the networks that the ABR advertises in Type 3 LSAs.

This discussion assumes that area 1 and area 100 are normal, stub, or NSSA areas, but not Totally stub or Totally stub NSSAs. The Totally stub areas do not receive Type 3 LSAs, except a default route, so the routers in those areas cannot choose best paths for inter-area traffic based on anything except the cost in the default route.

A Type 3 LSA advertises a route to the network. The routers that receive these advertisements do not create a topology of the networks in other areas. Instead, they treat the ABR as a gateway to all the networks which the ABR advertises in Type 3 LSAs. That is, those networks are leaves hanging off the ABR in the shortest path tree.

Each router determines the shortest path to the ABR or ABRs advertising Type 3 LSAs to it. It uses Dijkstra's algorithm to calculate this path as normal for inter-area routes – run for that router's area. The next hop for reaching the ABR becomes the next hop for all inter-area routes advertised by that ABR in Type 3 LSAs.

When the area includes multiple ABRs, internal routers learn multiple inter-area routes for that network. These internal routers use the metric to choose the best path. The metric for each inter-area route consists of:

Metric advertised in the Type 3 LSA, which the ABR copies from its metric for the target route

Metric for reaching the advertising ABR, from the area's shortest path topology

Because each ABR advertises the cost of its own route to the network, cost increments across the complete system. First consider a situation without route aggregation.

Core1 advertises 10.100.0.0/24 as a Type 3 LSA with metric 105 in area 0, and Core2 advertises this route with metric 150. Agg-1 adds 5 to each of these metrics because that is its cost for reaching each of the routers. It selects Core1 as offering the best route and adds an inter-area route with metric 110 to its OSPF table. When Agg-1 generates an LSA for 10.100.0.0/24 in area 1, it advertises metric 110. SW1 adds its metric for reaching Agg-1 to 110, assigning the inter-area route to 10.100.0.0/24 a metric of 160.

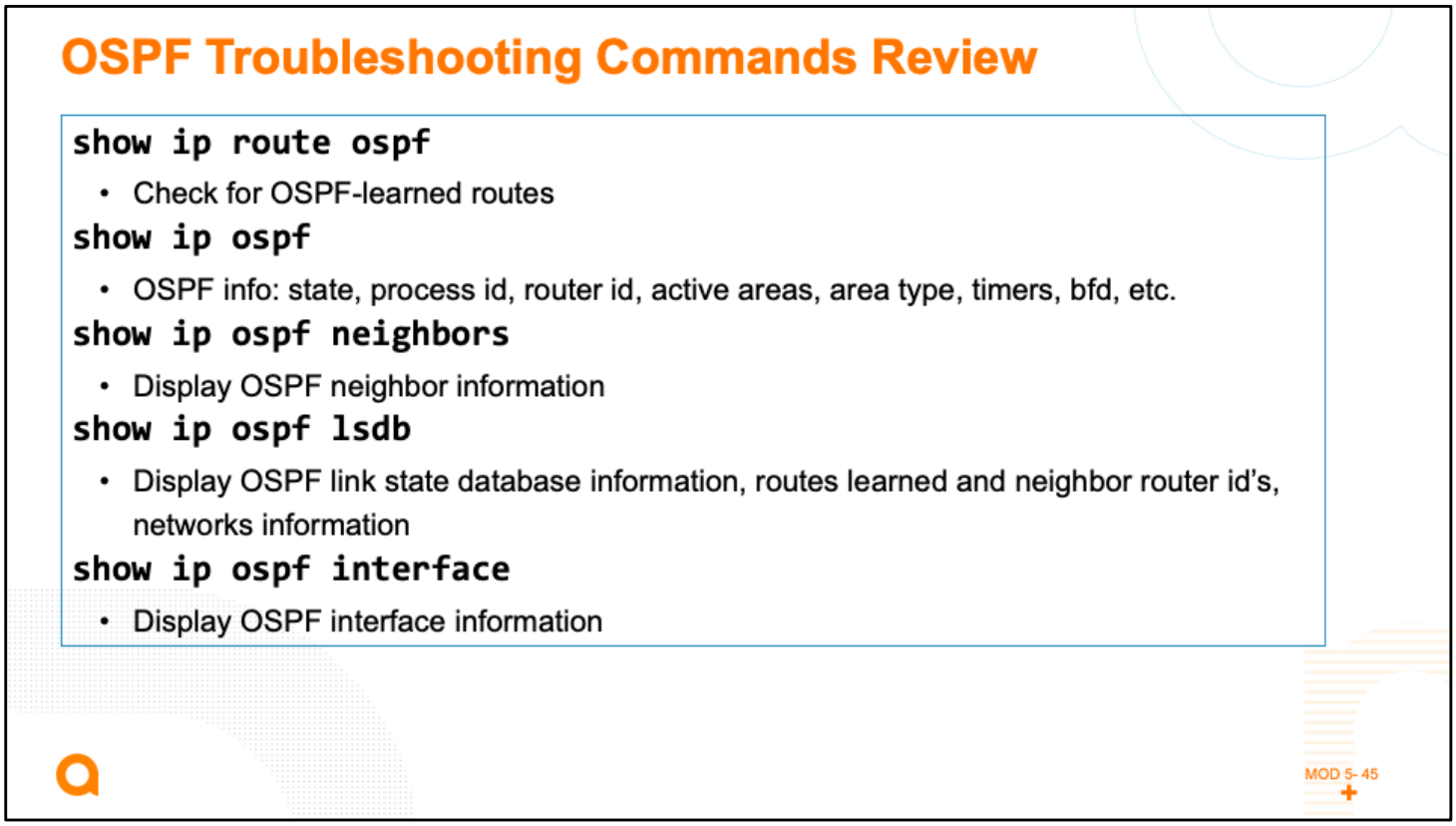
Inter-area routing in OSPF combines the best aspects of link-state and distance-vector routing. SW1, for example, uses link-state routing to determine its shortest path to its two ABRs (Agg-1 and Agg-2). It can then use those same two paths for all routes to other areas without considering the link states of networks in those areas. Even though SW1 only knows a small part of the topology, traffic that it routes toward other areas still receives the benefits of link-state routing across the complete path. For example, SW1 sends traffic to 10.100.0.0/24 on its shortest path to Agg-1. Agg-1, in turn, routes the traffic using the shortest path to Core1, which it has calculated from its detailed area 0 topology. Finally, Core1 routes the traffic using the best path calculated by the detailed area 100 topology. In each area, routers maintain just the local segment of the complete path for inter-area traffic.

Also consider how the area boundaries enhances stability. When an OSPF router receives an update for a Type 3 LSA, it does not run the complete

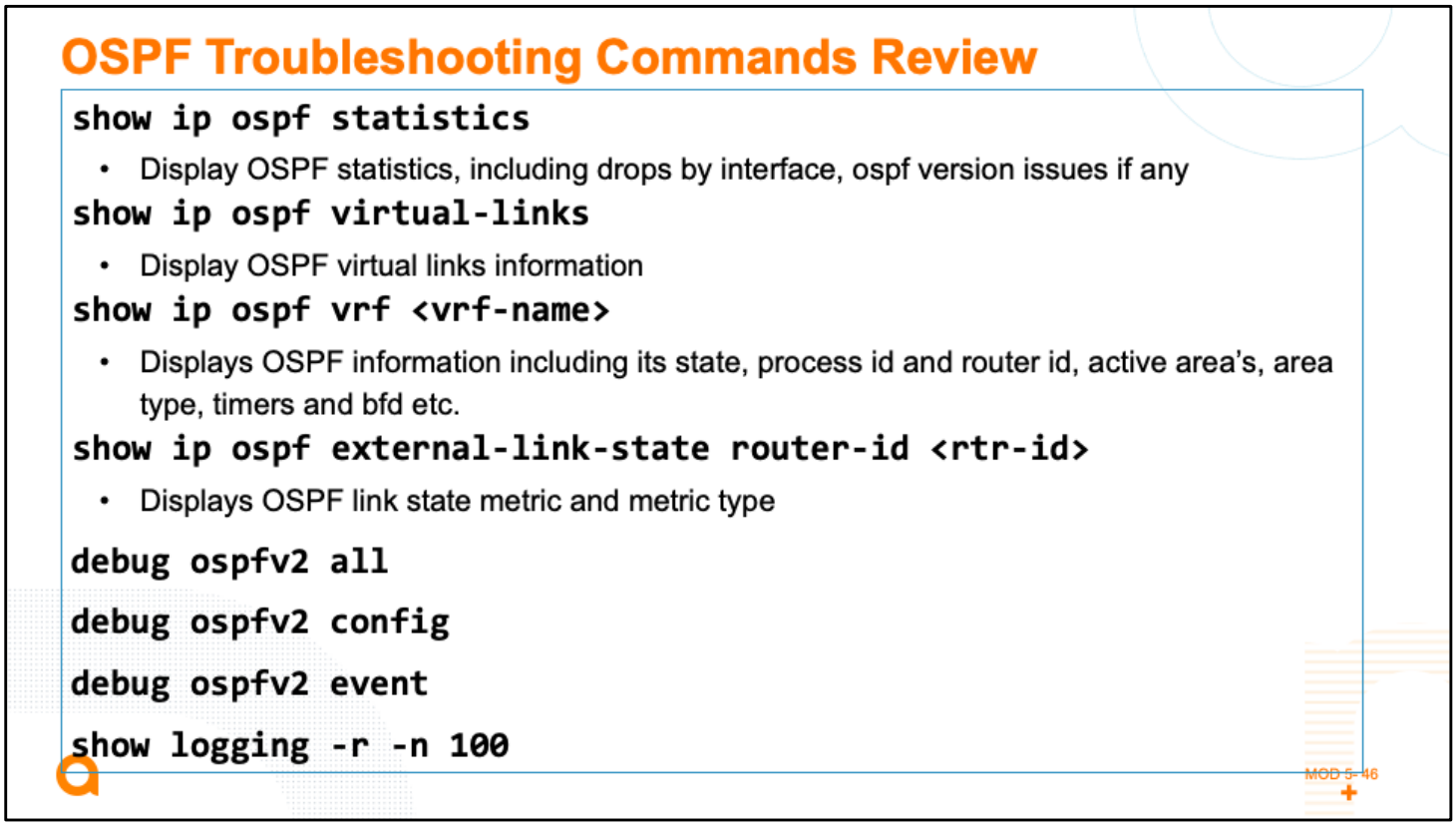
shortest path algorithm. Instead, it adjusts the networks behind each ABR, as indicated by the update, and adds a new inter-area route, removes an inter-area route, or selects a different inter-area route (due to a cost change), as appropriate. The router does not, on the other hand, need to re-calculate the path to the ABR, which is the only segment of the path with which it is concerned.

Some loss of information is introduced by route aggregation at the area boundaries; however, the simplicity is typically worth this loss. For example, in this topology, network administrators might configure an aggregated range of 10.100.0.0/16 for area 100 on Core1 and Core2. They can assign cost 105 to this range on Core1 but cost 150 to the range on Core2. Agg-1 can still choose the route through Core1 as the lowest cost route. In this example, Agg-1 and Agg-2 have the same cost to reach Core1, so they advertise the same cost for the 10.100.0.0/16 route into area 1. However, Agg-2, for example, had a better cost path to Core1, it would advertise a lower cost route, and internal routers in area 1 would direct their area 100 traffic to that ABR.

You could also use multiple area summaries when an area has multiple ABRs, and you want to prefer one ABR for one segment of the area but another ABR for another segment. This approach is more complicated, however. You must also take care to configuring matching summaries on each ABR (although the costs would not match). Otherwise, you can accidentally introduce routing loops.



The figure offers a summary of the commands you have learned about in this module. You can use this a reference for future OSPF troubleshooting scenarios.



The figure offers a summary of the commands you have learned about in this module. You can use this a reference for future OSPF troubleshooting scenarios.


Lab Activity
Lab 05


Rev # 20.41
aruba
a Hewlett Packard Enterprise company
BGP
Aruba Advanced Switching, Troubleshooting and Solutions (ASTS)

Objectives

- Describe BGP use cases and features
- Configure BGP peers
- Describe BGP sessions and states
- Validate BGP peers
- Describe BGP advertisements
- Configure and validate BGP advertisements
- Describe BGP path selection criteria
- Tune and validate BGP path selection
- Describe route filtering
- Configure AS Path lists to filter routes
- Redistribute BGP routes into OSPF
- Configure peer groups and route reflectors



After completing this module, you will have a strong grasp of BGP operation, deployment, validation and troubleshooting. You will be able to describe BGP use cases and features, and to configure BGP peer relationships. You will be able to describe BGP session establishment and state changes and leverage that information for troubleshooting. You will be able to validate various types of BGP peer relationships, and to describe BGP advertisements.

You will be able to configure and validate BGP advertisements.

You will also be able to describe BGP best path selection criteria and leverage this knowledge to tune and validate BGP path selection.

You will be able to describe both inbound and outbound route filtering, and how to use prefix lists and AS path lists to filter routes.

You will be able to inject the proper routes from BGP into OSPF, without overloading the OSPF routing processes.

Finally, you will be able to use peer groups to streamline your configuration efforts and configure route reflectors to improve BGP scalability.



BGP is arguable the most fascinating routing protocol in use today. It seems complex at first, but you quickly realize that you have complete control over BGP best-path selection processes. You can make your routers sing and dance the way you want them to – to conform to your corporate policy requirements.

You begin by exploring BGP use cases and features, before configuring BGP peers. You'll learn about several important concepts and how to apply them for successful internal and external BGP peer relationships. This includes features like eBGP multihop and the iBGP full-mesh requirement. You will learn some effective commands to validate and troubleshoot BGP peers.

You configure peers to tell BGP routers who to talk to, now you learn about BGP advertisements, so you can tell those routers what to say. You will explore several key concepts about BGP advertisements and rules...and how to break those rules for your benefit. This includes using the next-hop-self parameter with iBGP peers and dealing with BGP reachability requirements. You will learn how to control what BGP routers advertise with the network and aggregate-address commands.

Now that your routers are speaking to each other, and advertising subnets, you may want to tune how those subnets are advertised, and how BGP selects the best path. You review BGP path selection criteria, and then quickly move into describing and applying this knowledge to tune router functionality. You will configure and validate route tuning using things like AS path

length, weight, local preference, and the Multi-Exit Discriminator (MED).

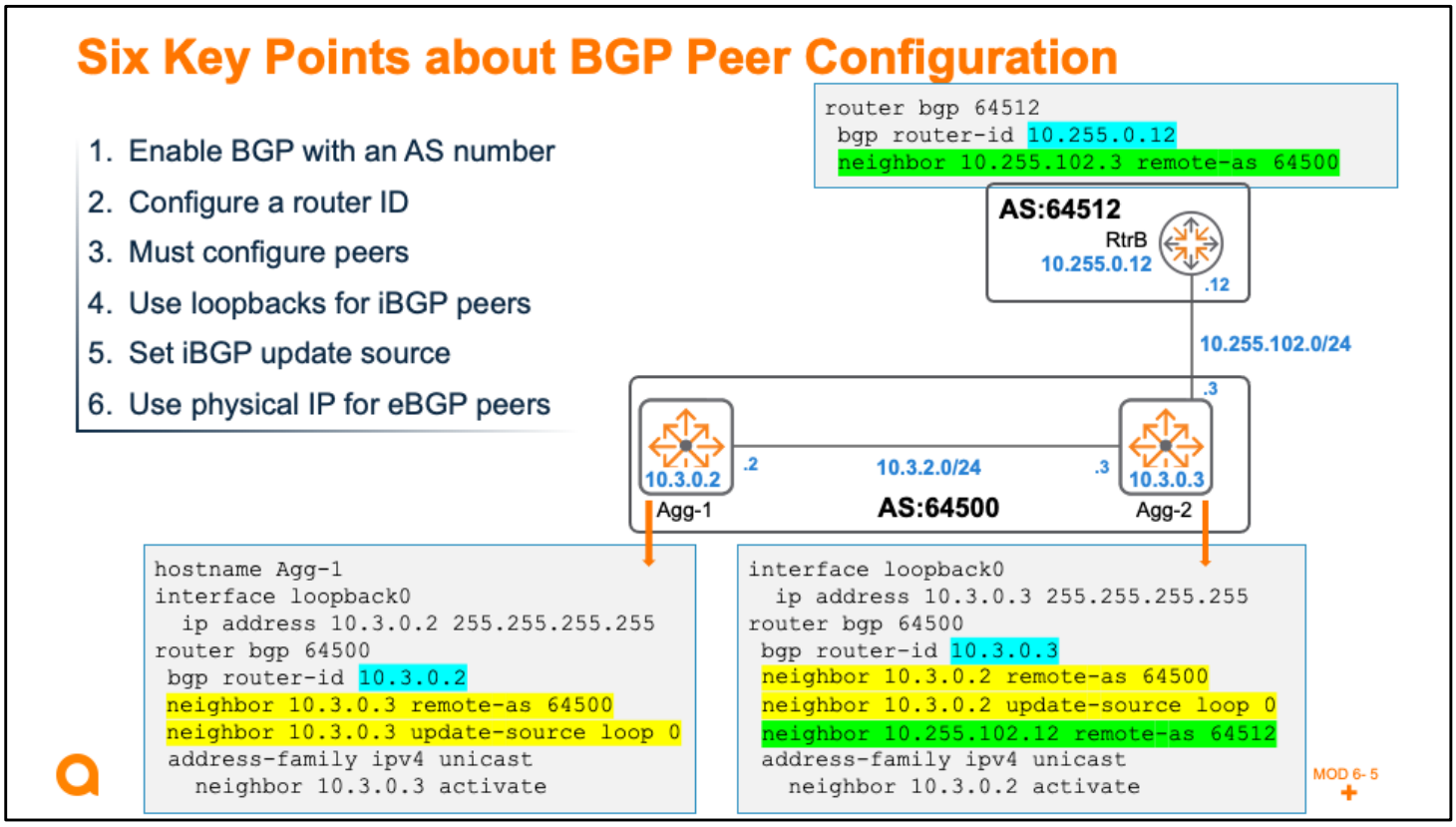
Now you move into route filtering and control. You will learn to use IP prefix lists and AS path lists to avoid becoming a transit AS. You will learn to configure inbound and outbound route filters, and how to inject routes into OSPF. Then you will learn to use peer groups to streamline your configuration efforts, and to use route reflectors to improve BGP scalability.

Finally, you will apply this knowledge in a series of hands-on lab activities.

BGP Concepts and Peers

Use cases, debugging, peer validation

aruba
a Hewlett Packard Enterprise company



The figure shows basic BGP peer configuration, making six key points about BGP peering.

Enable BGP with an AS number

– Start by configuring the BGP process with its AS number, as shown. Remember, the company must obtain this number from a regional authority. You can create only one process and AS on the switch.

Configure a router ID

– Although optional, you are strongly encouraged to assign a BGP router-ID. Like OSPF, the loopback 0 address is typical. If you do not set the router ID manually, the AOS-CX switch uses the highest IP address on a loopback interface, or, if no loopback interfaces exist, the highest IP address on any interface.

You must configure peers

– Unlike IGPs like OSPF and RIP, BGP does not discover peers. You must configure each peer's IP address and AS number. This is how devices know whether they are iBGP or eBGP peers. In the figure, Agg-2 knows that its AS = 64500, and you configured peer 10.3.0.2 (Agg-1) with the same AS. These are iBGP peers. You configured Agg-2 to peer with Router B at 10.255.102.12, AS 64512. Different ASes, so these are eBGP peers.

– Configured peers do not communicate until you activate them. You see this with the neighbor activate command. In the figure only the iBGP peers are currently activated, so the eBGP peers will not come up.

Use loopbacks for iBGP peers

– You should configure iBGP sessions to peer with each other's loopback address, to maximize resiliency. Suppose you configure Agg-1 and Agg-2 to peer via their physical 10.3.2.0 addresses. If that link fails, peering is broken. If you peer as shown in the figure, via loopback addresses, the routers can maintain the peer relationship via some alternate path (not shown).

Set iBGP update source

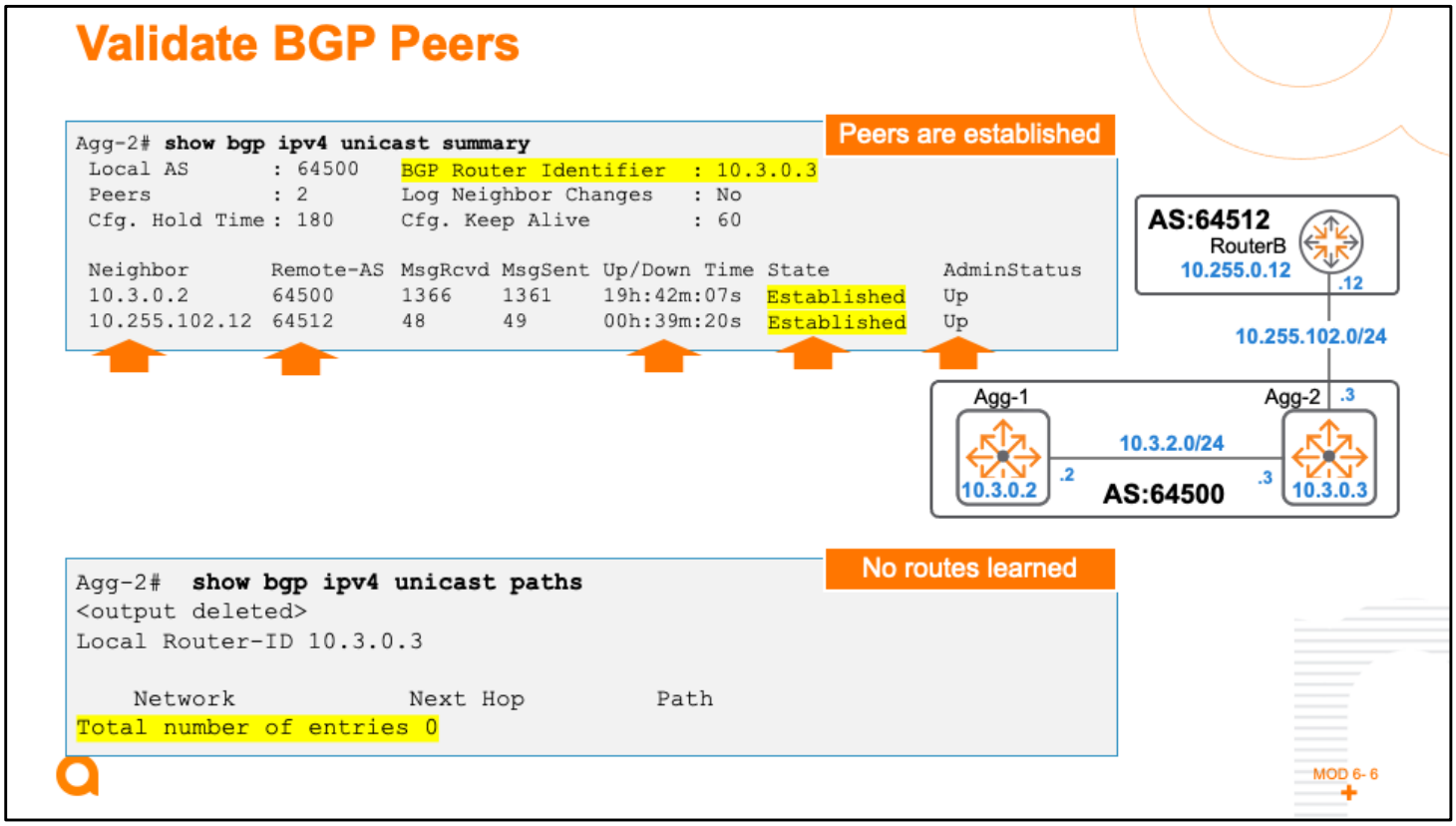
– BGP peers will always use the IP address of their directly connected link to the other peer by default. So Agg-1 will use 10.3.2.2 as the source IP address for all BGP packets sent to Agg-2. Meanwhile, you told Agg-2 to peer with 10.2.0.2, not 10.3.2.2. When Agg-2 receives a packet from 10.3.2.2, it says, "I'm not supposed to talk to you", and refuses to form a peer relationship. So, you tell Agg-1 to use its loopback 0 IP address as the source address. Agg-2 is configured to peer with this address, and so the solution works as intended.

eBGP peers use physical IP

– You often configure eBGP peers using their physical IP addresses. This is because eBGP peers often only have a single link between them, with no alternate paths. Also, unlike iBGP, eBGP peers must be directly connected.

Note: There are many rules that govern BGP operation. However, you will come to realize that almost every rule has at least one exception. In other words, almost anything you say about BGP has a caveat. For example, you just learned the rule that "eBGP routers must be directly connected". There is a way to get around this rule with an eBGP multi-hop parameter. To aid your learning, statements about BGP will often be made plainly – without distracting you with exceptions. When appropriate, the exceptions and caveats will be shown.

Once activated, whether eBGP or iBGP, BGP routers go through a formal process to become peers, a bit like OSPF. You should be familiar with this process.



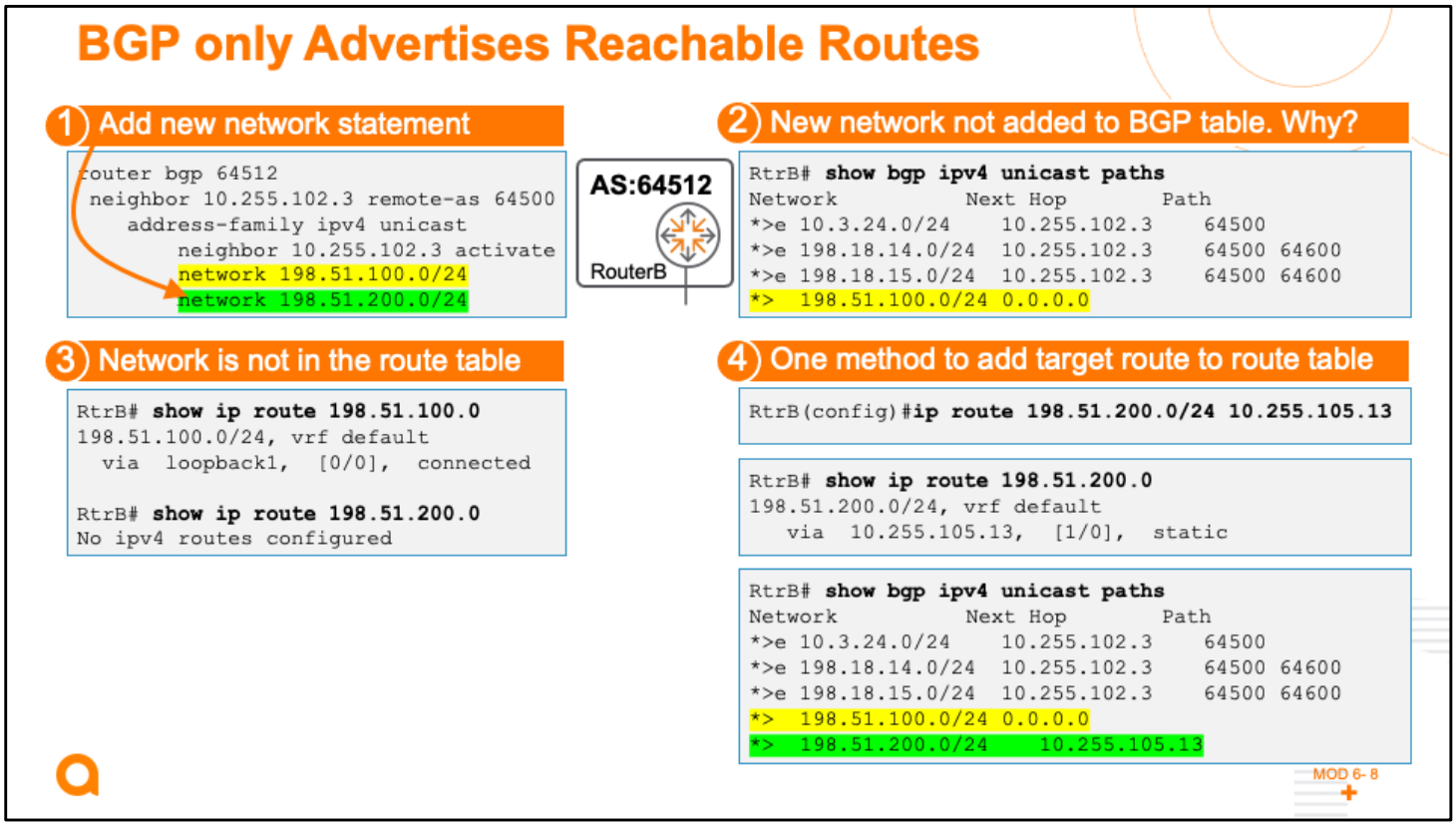
The detail you get with the previous show bgp ipv4 unicast neighbor command is often quite useful. However, you can see a nice summary of this information – often all you really need, with the command show bgp ipv4 unicast summary.

You have successful BGP peering, but the BGP paths command output at the bottom of the figure shows that no routes have been learned. So, you explore BGP advertisements next.

BGP Advertisements

iBGP and eBGP advertisements and table review

aruba
a Hewlett Packard Enterprise company



In this scenario, Router B is configured with yet another network statement. You have told the router to advertise 198.51.200.0/24.

However, unlike the first network, BGP refused to advertise the route for you – it is not added to the BGP table. Why?

You must understand that BGP only advertises routes that it can actually reach. In other words, the route must be in the routing table for BGP to advertise it.

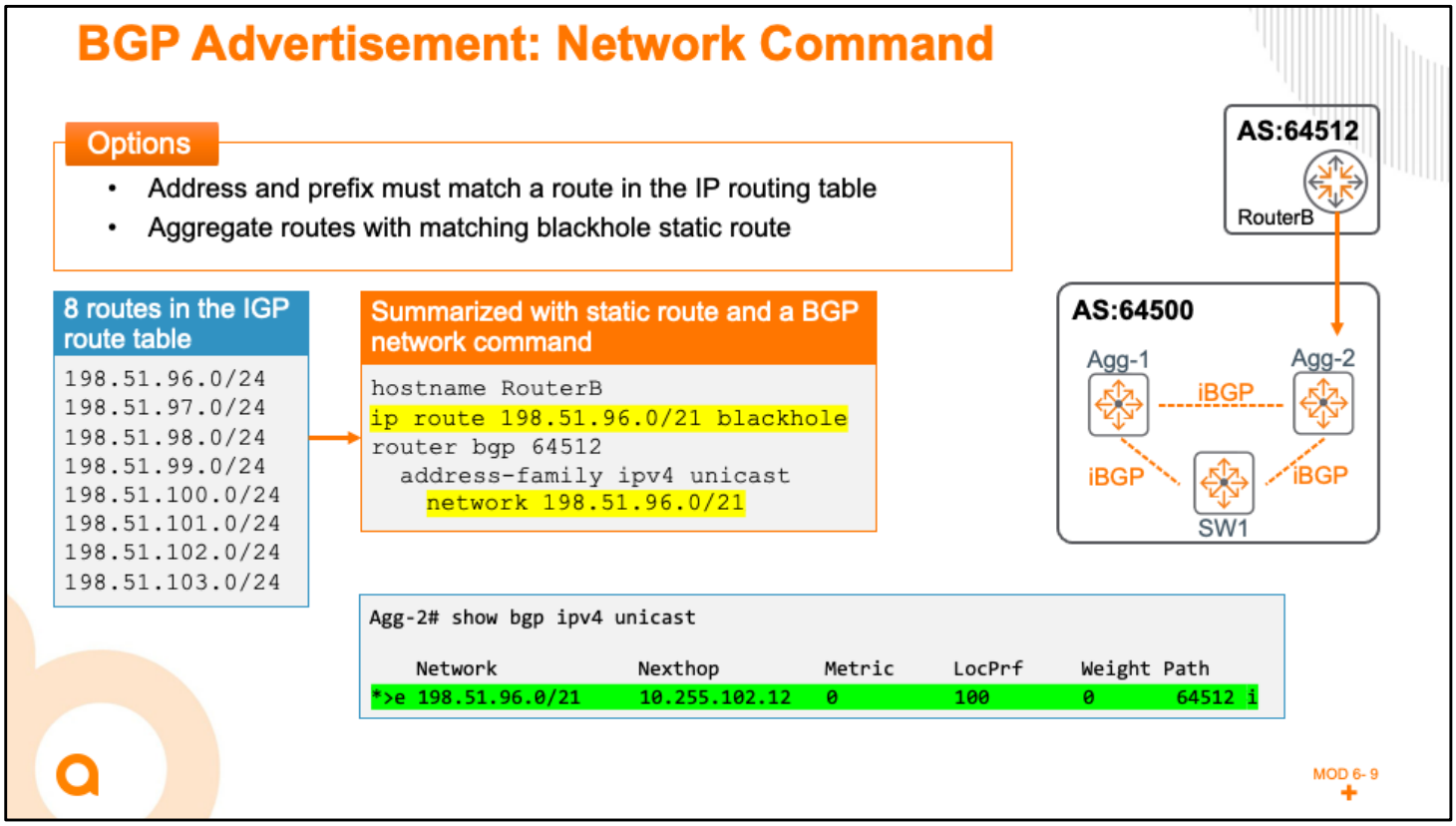
This is shown in step 3 of this scenario. The 198.51.100.0/24 subnet is in the routing table, because it is directly connected. The 198.51.200.0 network is not in the route table, and so BGP refuses to advertise it.

One way to add this target route into the route table is with a static route, as shown in the figure. Since the route is in now in the routing table, it is added to the BGP table, and RouterB will now advertise this route to its peers.

Do you see why using a static route in this scenario is a dangerous technique? If you do this, you must be correct – do not lie to your routers! If that destination route is not truly reachable via the specified next hop, you have essentially created a black hole. BGP advertises a route that it cannot actually reach. Also remember that static routes are static – they do not respond to topology changes. Even if you are accurate with your static route, if some link along the

path fails, the static route is no longer valid. But BGP doesn't know that, and so continues to advertise the route.

More typically, the target route (198.51.200.0/24) would be learned by OSPF or other IGP. If it does not exist in the route table, then there is a problem with OSPF – you should troubleshoot. So, while the static route is a nice, simple way to make this work, it can be a bit dangerous as used here. You might more commonly use this technique as way to get BGP to do route aggregation – to advertise a range of /24 routes as a single /16 group, for example.

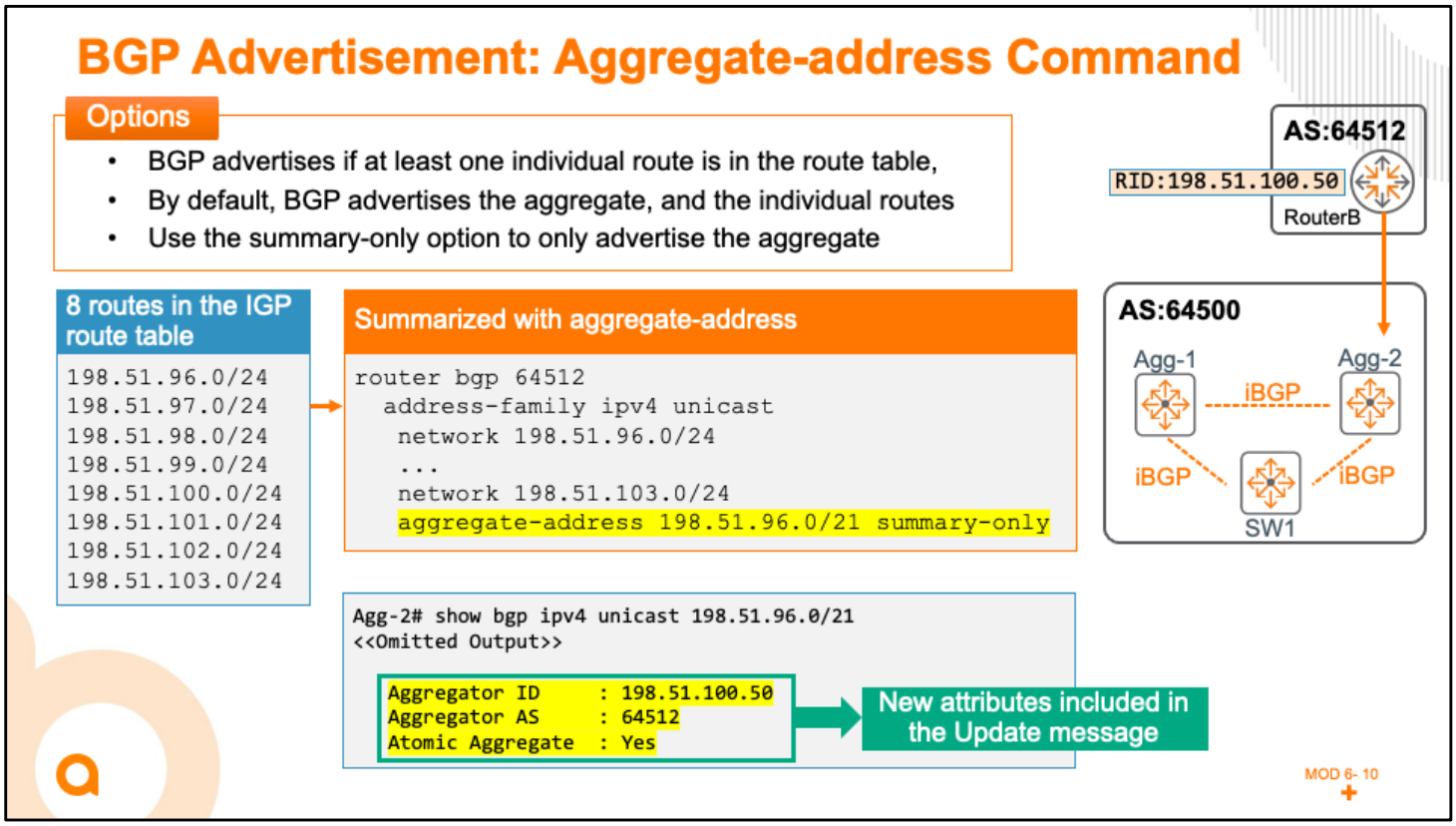


You have learned about one way to tell BGP routers what to say - to advertise routes to local networks, with the network command. You know that your network statement must exactly match an existing prefix/mask entry in the route table. However, you learned to create a blackhole static route to inject an entry into the route table.

Sometimes your BGP router has many IP routes within the public IP block that you want to advertise, and you want to inject a single route that aggregates these routes. You might want to advertise 198.51.96.0/24 through 198.51.103.0/24 as a single aggregate – 198.51.96.0/21 to summarize the shorter subnets within that range. When the aggregated route does not exist in the routing table, you must add a static route to the network through the blackhole. Specifying the blackhole interface ensures that the route always remains in the routing table. It also allows the router to drop any traffic destined to an IP address within this range, if it does not have a more specific route for that address, as it should.

You can then specify the aggregated network in the BGP process. The network matches the blackhole route, so it is added to the BGP table.

You should always view the BGP routing table to check the routes that your router will advertise to its neighbors. When you inject the route into the BGP routing table, the BGP routing process adds a number of BGP-specific attributes, including origin and next hop. The origin of the route is i, indicating that the BGP router has learned this route from one of its IGPs. The next hop is empty, indicating that this router advertises the route.



Another way to get BGP to aggregate several subnets into a single advertisement is with the aggregate-address command, as shown in the figure. As long as the BGP routing table has at least one network in that range, the AOS-CX switch will begin to advertise a route to the aggregated network.

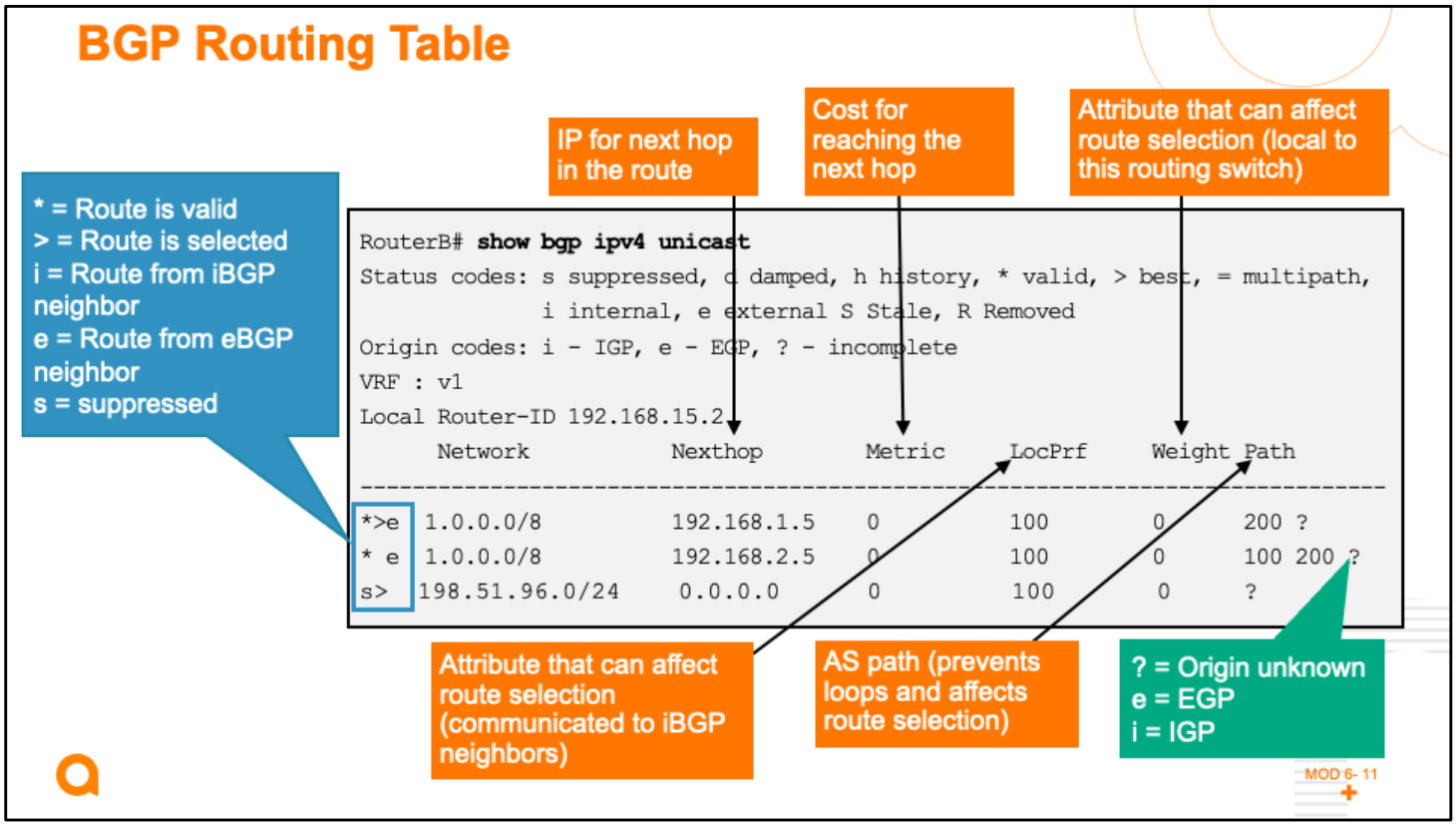
The switch also automatically adds a reject route to the aggregated network in the IP routing table. This allows the router to drop any traffic destined to an IP address within this range, if it does not have a more specific route for that address, which is the proper behavior. Use the summary-only option with the aggregate-address command to tell the switch to suppress advertisement of the individual networks within the range.

You still need to use the network command to inject the individual networks (technically, the network command for only one network is required, but you probably want the switch to continue advertising the aggregated address if it has contact with any of the networks.) | When you add the aggregate-address command, the aggregated route does not show up in the switch's BGP routing table. You will see an "s" at the front of each individual network in the aggregated range because those routes are suppressed. |And you can see the switch advertise the aggregated range to neighbors, and the route appears in their tables.

Another approach for aggregating BGP advertisements exists by manually adding a static reject route to the aggregated network address and then injecting that network into BGP with the network command instead of the individual networks. However, with this static approach,

the BGP router will continue to advertise the aggregated network even if it loses all active routes to the networks within the range. Therefore, the more dynamic aggregate-address approach is preferable.

You can also redistribute from another protocol, such as OSPF, or from static or connected routes. Usually, you should use the injection method, which lets you control precisely which networks the router announces. Redistributing routes can either redistribute more routes than you desire or requires route maps to filter routes.



After BGP neighbors have established a connection, they exchange their entire BGP routing tables. Before you learn more about advertising routes, it can be useful to examine the BGP routing table in more detail.

Each route has several symbols at the beginning. An asterisk (*) indicates that the route is valid, which means that the next hop is reachable. The greater than sign (>) indicates that the router has selected this route as the best one to the network in question. The route's origin is also indicated with (i) for a route received from an iBGP neighbor and (e) for a route received from an eBGP neighbor. If the route has been suppressed, then an (s) will indicate this state.

The route next indicates the destination network address and the next hop. The metric indicates the cost for reaching the next hop; unlike with OSPF, this metric is not the most important criteria for selecting between routes. LocPrf refers to local preference. Both local preference and weight can affect route selection, but local preference is communicated between iBGP routers while weight is locally significant only.

Finally, note the AS path, which is included in every BGP route. When a BGP router first advertises a network in BGP, it puts its AS number in the path. When a BGP router in a different AS receives the route, that BGP router adds its AS to the path, and so on. In this way, the AS path indicates every AS which traffic will traverse to reach the destination.

In this example, the BGP router has two routes to 1.0.0.0/8: one through AS 100 and one through AS 200 and then AS 100. The AS path composes BGP's primary loop elimination technique. If a router receives a route with its own AS in the AS path, it drops the route. The AS path also affects route selection with shorter paths preferred. The third entry indicates a suppressed network, this means that this entry is not advertised and instead we advertise an aggregate route.

Question #1

```
# show debug buffer module bgp
--------------------------------------------------------------------------------
show debug buffer
--------------------------------------------------------------------------------
2020-10-07:13:18:06.180584|hpe-routing|LOG_INFO|AMM|-|BGP|BGP|Remote address = 10.255.101.11
2020-10-07:13:18:06.180594|hpe-routing|LOG_INFO|AMM|-|BGP|BGP|Remote port = 0
2020-10-07:13:18:06.180604|hpe-routing|LOG_INFO|AMM|-|BGP|BGP|Scope ID = 0
2020-10-07:13:18:06.180614|hpe-routing|LOG_INFO|AMM|-|BGP|BGP|Incoming connection? = False
2020-10-07:13:18:06.180624|hpe-routing|LOG_INFO|AMM|-|BGP|BGP|Remote AS number = 64511
2020-10-07:13:18:06.180926|hpe-routing|LOG_ERR|AMM|-|BGP|BGP|A NOTIFICATION message is being sent
to a neighbor due to an unexpected
2020-10-07:13:18:06.181026|hpe-routing|LOG_ERR|AMM|-|BGP|BGP|Error code = OPEN Message Error (2)
```

What could cause a BGP peer to receive an OPEN message?

a. The remote BGP peer is not using ASN = 64511

b. Authentication failed

c. The remote BGP peer is not using the default BGP TCP port (179)

a

Knowledge Check

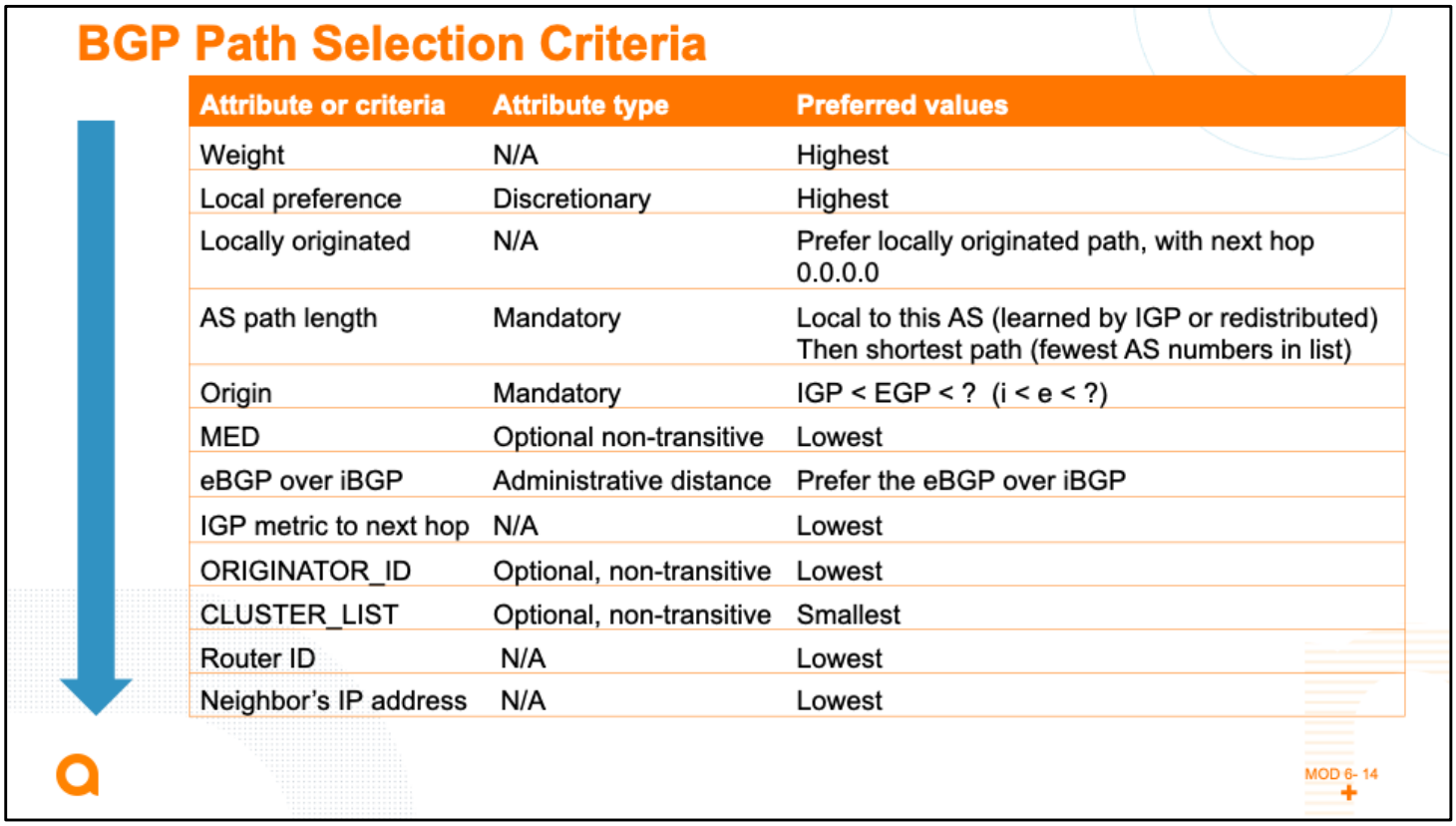
Correct answer:

A

BGP metrics and Tuning

Path selection criteria and tuning techniques

aruba
a Hewlett Packard Enterprise company



When multiple identical BGP routes are valid and have the same administrative distance, BGP uses the criteria shown in the figure to select the best route.

You see the BGP attributes and the values that are preferred for each attribute. Attributes are listed in the order by which the router examines them when selecting the best route. That is, the router selects the route with the better weight first. Only if those values are equal does the router move on to examine the next attribute, local preference. You can set a router's weight for specific routes. This value has meaning only on the router itself – it is not advertised. Local preference is not sent to eBGP peers, but it is one of the attributes sent in an update message to iBGP neighbors – those local to this router's AS.

Without special manipulation of weight and local preference on your part, your router will tend to prefer routes based on the AS path. It will accept the route from the ISP that reaches the networks in question through fewer ISPs. Presumably, the shorter AS path indicates a shorter latency; however, this is not necessarily the case.

For routes with the same number of AS IDs in the AS path attribute, the origin might come into play. Routes that were originally injected into BGP are preferred to routes learned from EGP (deprecated) and redistributed routes. However, often the ISPs will have learned the route from a similar ultimate source, so the origin is identical.

Note: Redistributed routes have an origin code of “?” (unknown) in the BGP table, as you will soon see.

The Multi-Exit Discriminator (MED) would take effect only if the ISP routers have explicitly set this attribute. They might do this if they needed to distinguish between multiple paths into your AS. This is only a consideration when you have multiple connections to the same ISP. Generally, a router only compares MED values between routes advertised from the same AS. However, in some cases, you can request different ISPs to send routes with different MED values. You must enter this BGP command to permit the AOS-CX switch to compare the MED values between the two ISPs:

```
switch(config-bgp)# bgp always-compare-med
```

However, without such special manipulation, if the AS path length is the same, your router will probably accept the route from the router to which it has a lower cost connection. If the metric to both ISPs is the same, the choice becomes nearly arbitrary. If ECMP is enabled, in fact, the AOS-CX routing switch adds multiple routes with the same next hop cost and load balances over them. If ECMP is disabled, the switch selects the route based on age, then router ID, and then the cluster list—but only if these checks are enabled. The final tie breaker is the advertising router ID.

Similarly, assuming no manipulation of attributes, routers throughout the Internet will tend to route traffic to your company through the ISP to which they are closer.

BGP allows you to control several of these attributes, to fine-tune the route selection. A neighbor router can also manipulate attributes and has ultimate authority over its route selection. Therefore, you can dictate which routes your devices choose, but you might not be able to determine the flow for inbound traffic.

The figure also indicates the type of attribute. Attributes without a type (N/A) are not included in the route advertisement – they only affect the local router’s path decisions. A good grasp of attribute types will help you do more advanced manipulation. Mandatory and discretionary attributes are both well-known attributes that all BGP routers understand; mandatory attributes must be included in every route advertisement, while discretionary attributes are

optional. An optional attribute might not be supported by all BGP routers. The “non-transitive” description means that if a BGP router receives this optional attribute and does not support it, it should drop the attribute.

In summary, here are the steps that could be involved in determining which BGP route is preferred:

Do not consider route if the next hop is unreachable

Select the route with the highest weight

Select the route with the highest LOCAL_PREF

Select the route originated by the local router

Select the route with the shortest AS-PATH

Select the IGP, EGP, or Incomplete route in turn , referred to as the “origin” attribute

Select the route with the lowest MED value

Select the route learned from eBGP, confederation, or iBGP in turn

Select the route with the smallest next hop metric

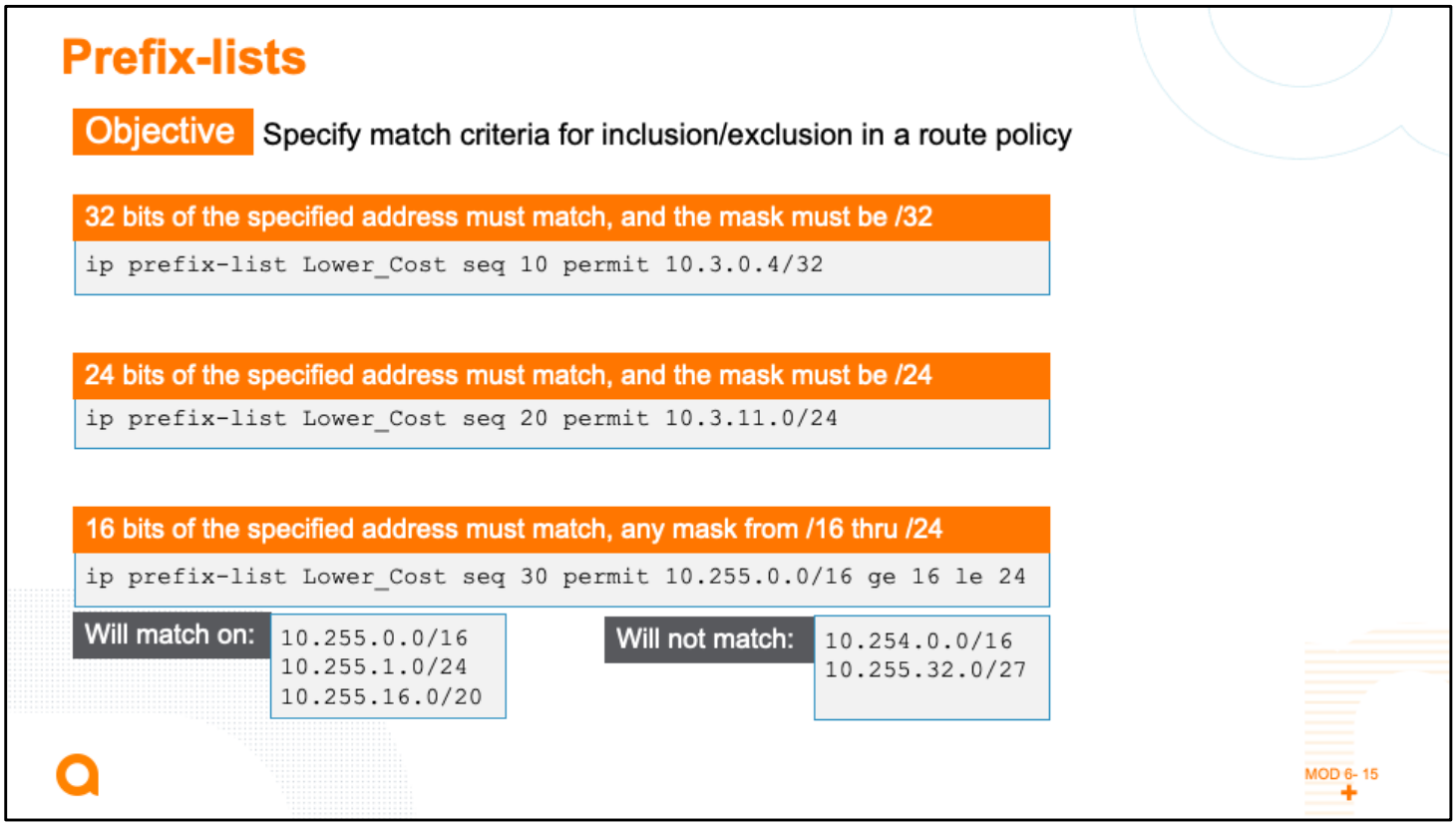
Select the route with the smallest ORIGINATOR_ID

Select the route with the shortest CLUSTER_LIST

Select the route advertised by the router with the smallest Router ID

Select the route advertised by the peer with the lowest IP address

You will often use route maps to manipulate BGP path decisions, sometimes in conjunction with prefix lists.



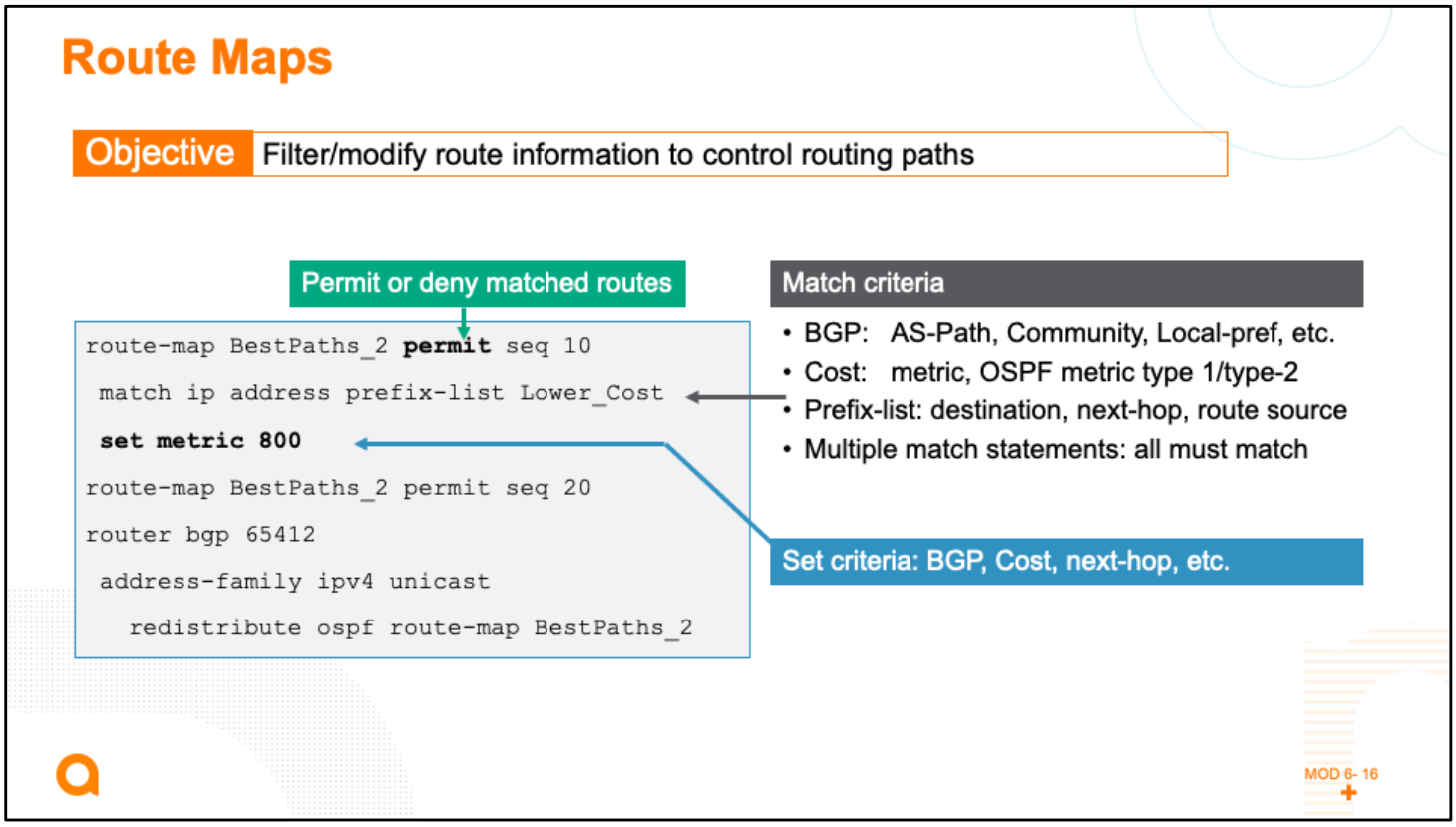
Prefix lists are named lists of route prefixes. They are used to match routes for inclusion in or exclusion from route policies. The sequence number determines the order of matching. The matches are performed starting from the lowest sequence number to the highest sequence number until there is a match. The sequence number is however optional and is autogenerated whenever it is not explicitly mentioned. All prefixes with the same prefix list name are grouped.

The autogenerated sequence number is derived by adding 10 to the highest sequence number available. This technique makes it possible to insert new prefix list sequence number in between.

Use the ge and le parameters to combine prefixes with a range of network mask. For example, 10.255.0.0/16 ge 16 le 24 will match all prefixes within the 10.255.0.0/16 network that have a mask greater than or equal to 16 bits and less than or equal to 24 bits in length. For instance, subnets such as those shown at the left side of the figure would match, because the first 16 bits of each address matches 10.255, and because the mask is between 16 and 24 bits long, inclusive.

The examples on the right would not match, either because the first 16 bits of the address are different, or because the mask is outside of the specified range.

A prefix list is one of several ways to specify match criteria for a route map.



You use route maps to route path selection and availability, by filtering or modifying route information. Routing policies can filter advertised, received, and redistributed routes. They can also modify attributes for specific routes. You use a three-step process to deploy route maps:

Configure filters based on route attributes. You can use prefix-lists to select a destination subnet, next-hop, or route source. The figure lists other criteria you can match on, as you are about to learn.

Create route maps and apply those filters, along with appropriate set criteria as needed

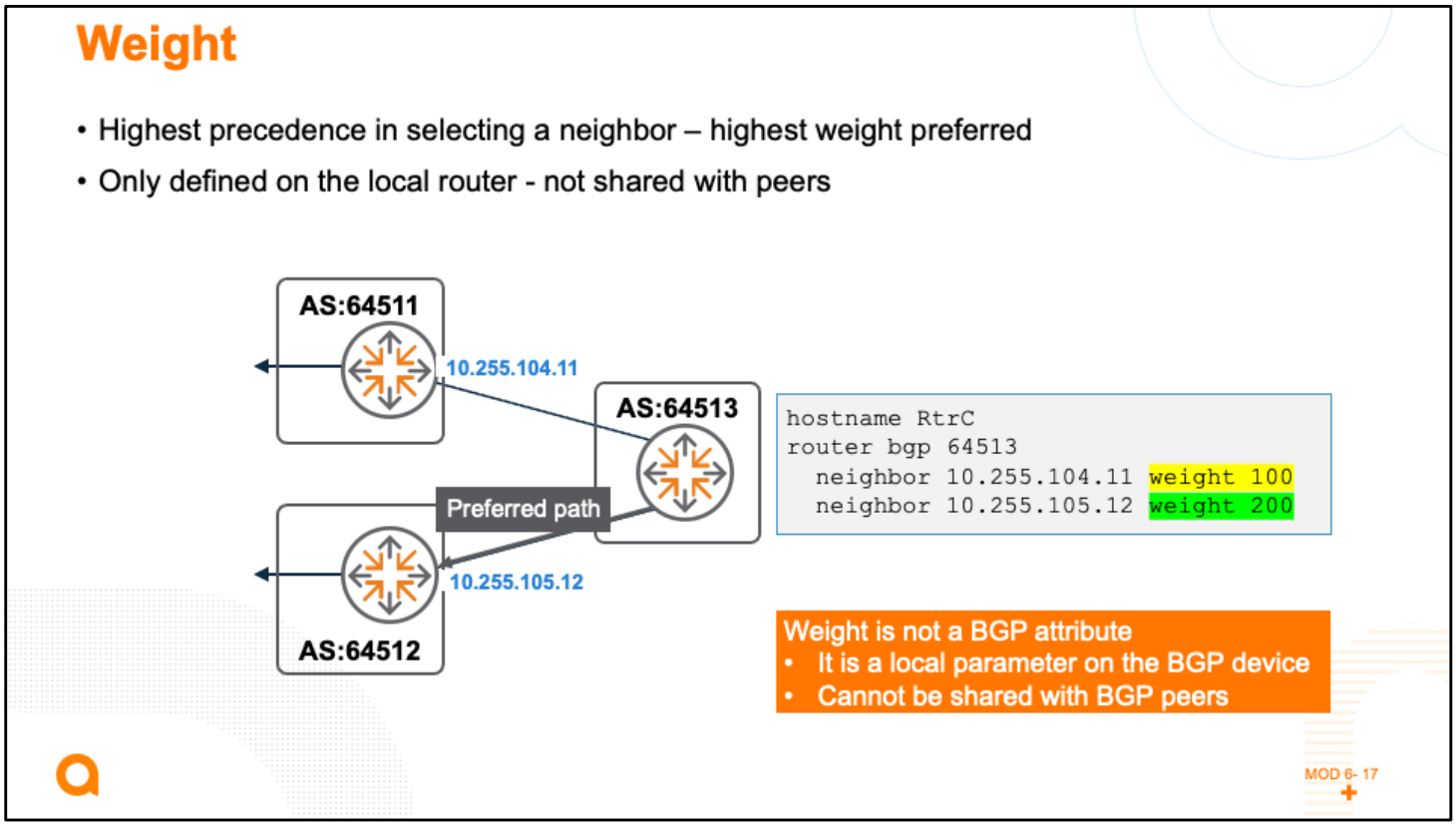
Apply the route map to the redistribution statement

The sequence numbers control top-down processing order, with a permit statement to allow matched routes to be advertised, or a deny statement to prevent their advertisement. Do you see the difference between route maps and ACLs? You often use ACLs to permit or deny user or server data traffic . You use route maps to control the advertisement of routes.

If you specify multiple match criteria within a sequence numbered section, then all criteria must match – it is a logical AND operation. Suppose you added a second match statement under sequence 10 of match ip route-source prefix-list Permitted_Sources. Route advertisements are only permitted and set if they match two criteria – they must be destination subnets that match the prefix-list Lower_Cost, AND they must be sourced (advertised from) ip addresses matched in prefix-list Permitted_Sources.

If both criteria are not met, then there is no match for sequence 10, and sequence 20 is processed. Like an ACL, once there is a match, no more sequences are processed. Also like an ACL, there is an implicit “deny all” at the bottom. If you want unmatched routes to be advertised, simply add a line at the bottom route-map BestPaths_2 permit sequence xx. No need to specify any criteria or set – the statement serves to allow all previously unmatched route advertisements.

Now that you have a better understanding of prefix lists and route maps, you should learn how to used these valuable tools to manipulate BGP path decisions, starting with AS Path Length.

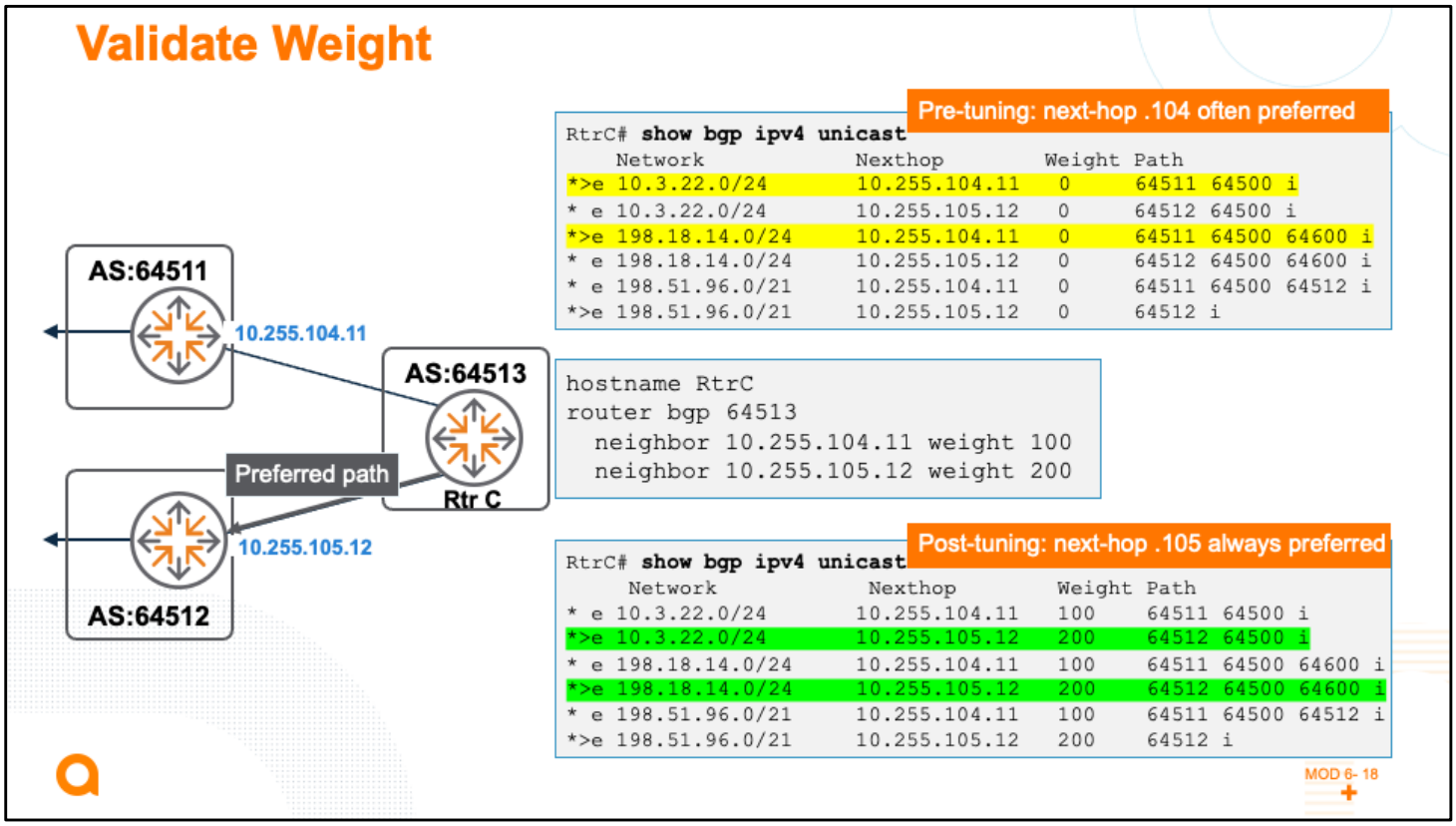


Weight is a metric that is not really part of the BGP standard, since it's not an attribute shared between BGP peers. Instead, the metric allows you to implement a policy on the local router as to which eBGP peer it prefers. The higher the weight, the more preferred the routes from the respective peer.

The figure shows an example of using weight to determine path selection, by simply adding a weight value to the neighbor statements.

You can also use route maps to adjust weight values for only specific routes from a neighbor, instead of all the routes of a neighbor, like shown in the above example. If using a route map, you would use the set weight command to adjust the weight for matching routes.

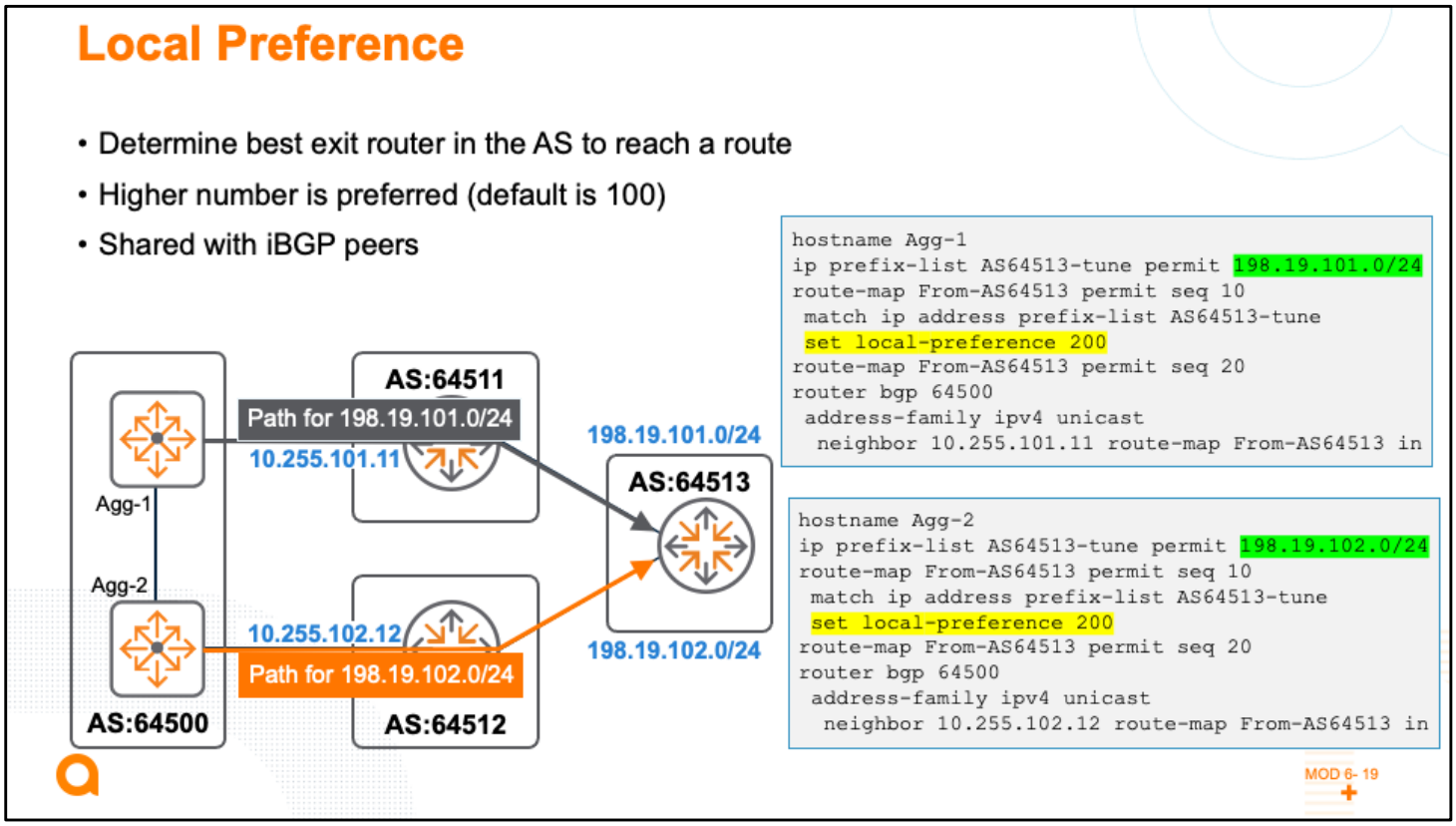
.



The figure shows before and after BGP tables, edited to focus on weight. Before tuning, next-hop 10.255.104.11 is the preferred path for most networks, except for the bottom entry, where the path via 10.255.105.12 has a lower AS Path length.

After tuning and clearing the BGP neighbors, next-hop 10.255.105.12 is ALWAYS preferred – even for paths with a lower AS Path length. Remember, once you change weight, no other metrics are considered for this local router. Also, do not forget that you can use a route map with weight, so only certain paths have a higher weight.

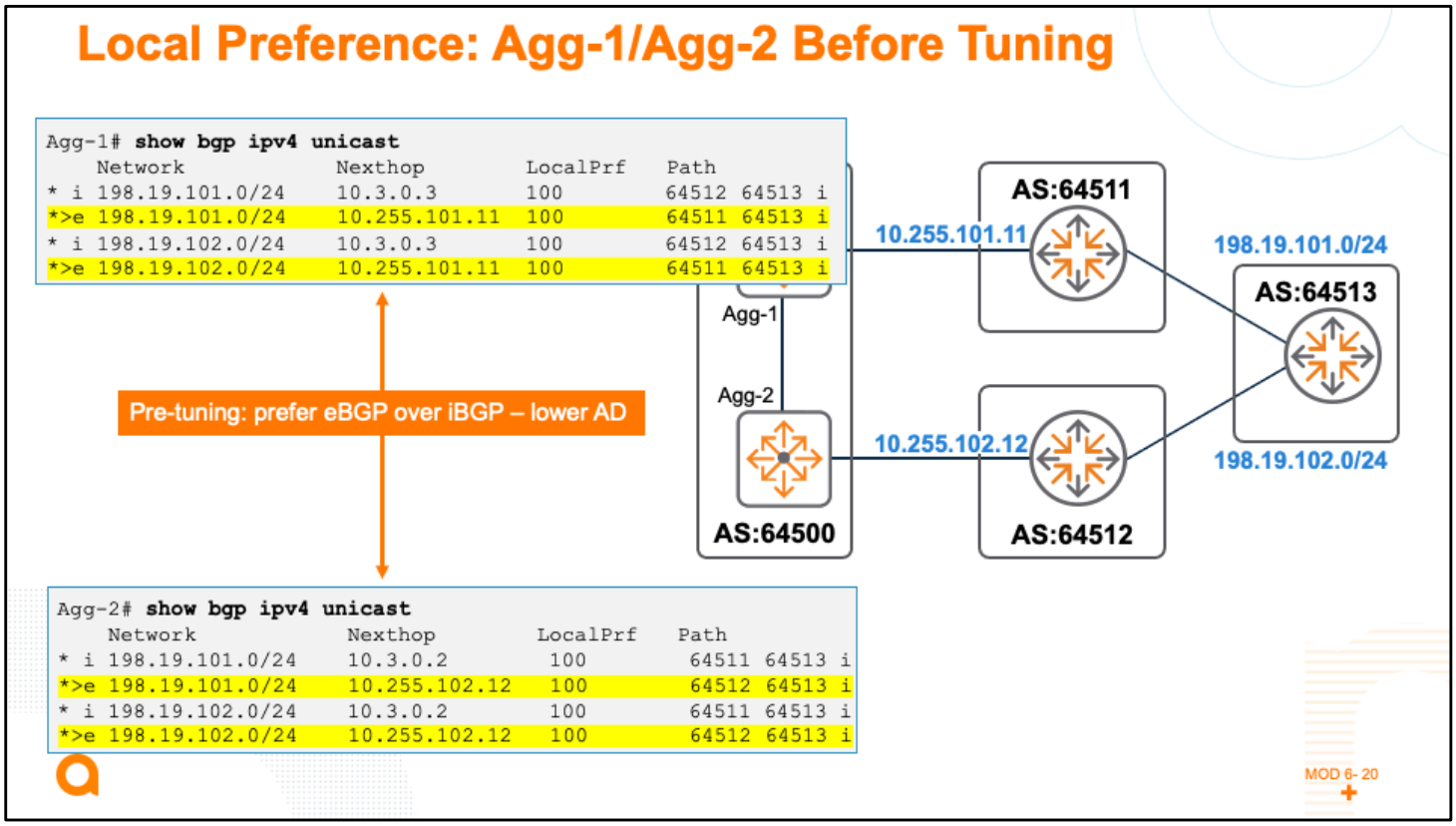
If you were to look at any of Router C's eBGP or iBGP neighbors, there would be no change – no difference in weight. Remember, weight only affects the router on which you configure it. It is not shared with other routers.



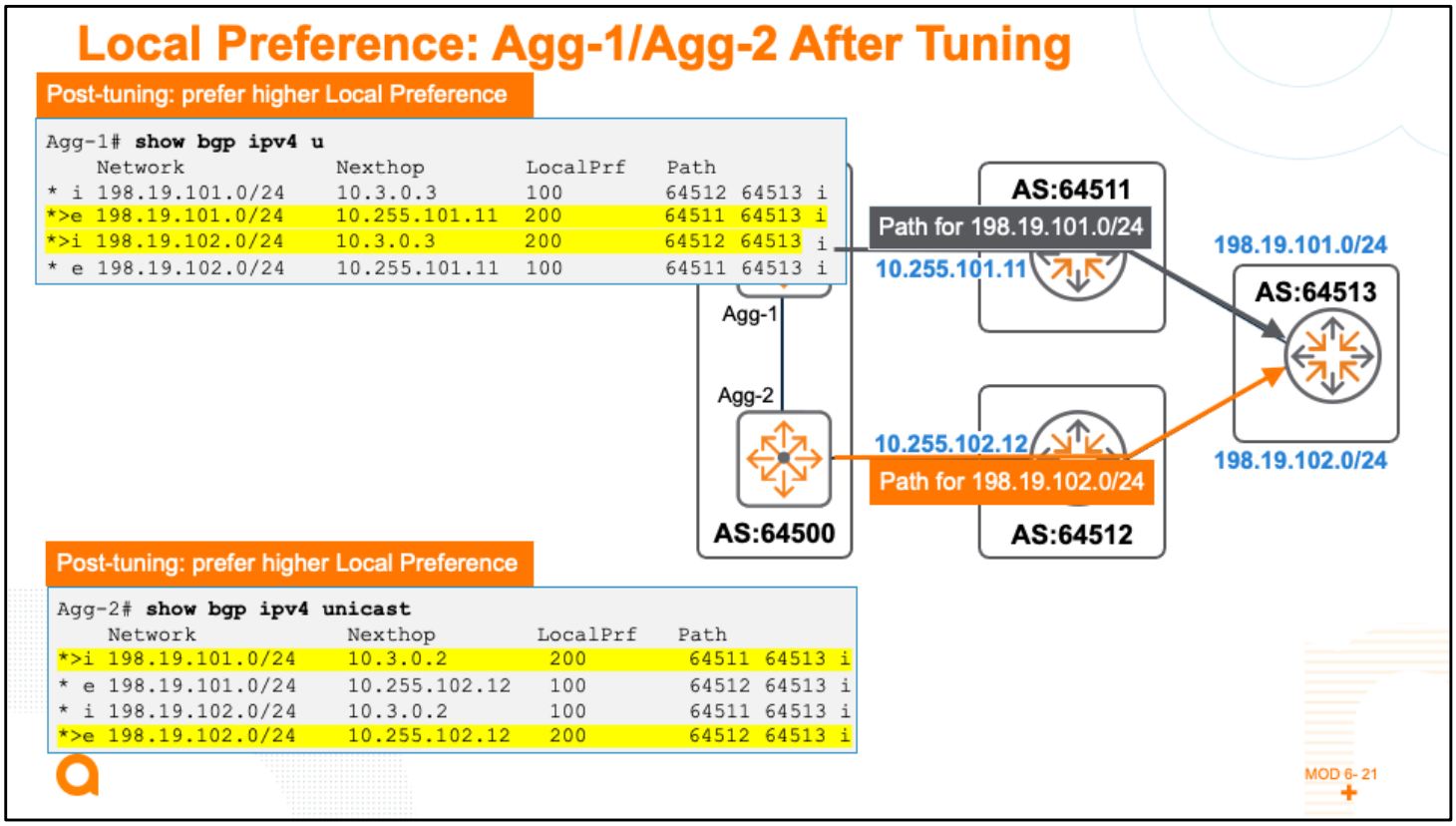
Local preference is a metric used with iBGP to determine the best exit to reach a particular route. The default local preference is 100: higher preference values are preferred.

The figure shows an example of using local preference, such that AS 64500 prefers some paths to AS 64513 via Agg-1, and others via Agg-2.

The route maps for Agg-1 and Agg-2 are the same, except the prefix-list matches on a different route.

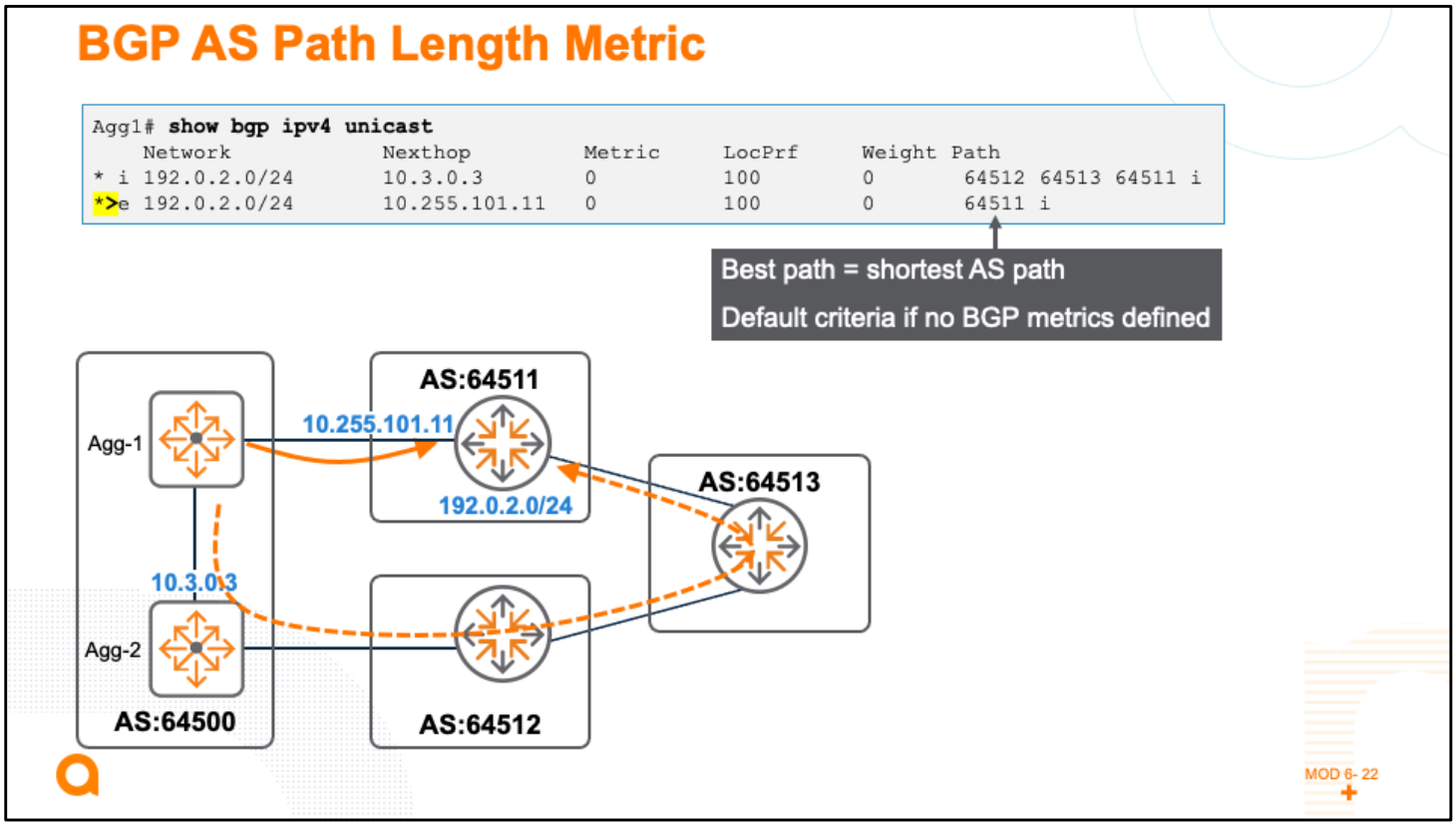


The figure shows current state for the target subnets, before you apply the route map. Agg-1 and Agg-2 reach all AS 65413 routes via their eBGP neighbor. This is because everything else is the same, so eBGP routes are preferred over iBGP routes. This is because eBGP has a lower Administrative Distance (AD).



After you apply the route map, the desired result is achieved. Agg-1 reaches 198.19.101.0/24 via 10.255.101.11, and it reaches 198.19.102.0/24 via Agg-2 at 10.3.0.3.

Agg-2 uses Agg-1 at 10.3.0.2 for 198.19.101.0/24, and it uses 10.255.102.12 for the 198.19.102.0/24 subnet.

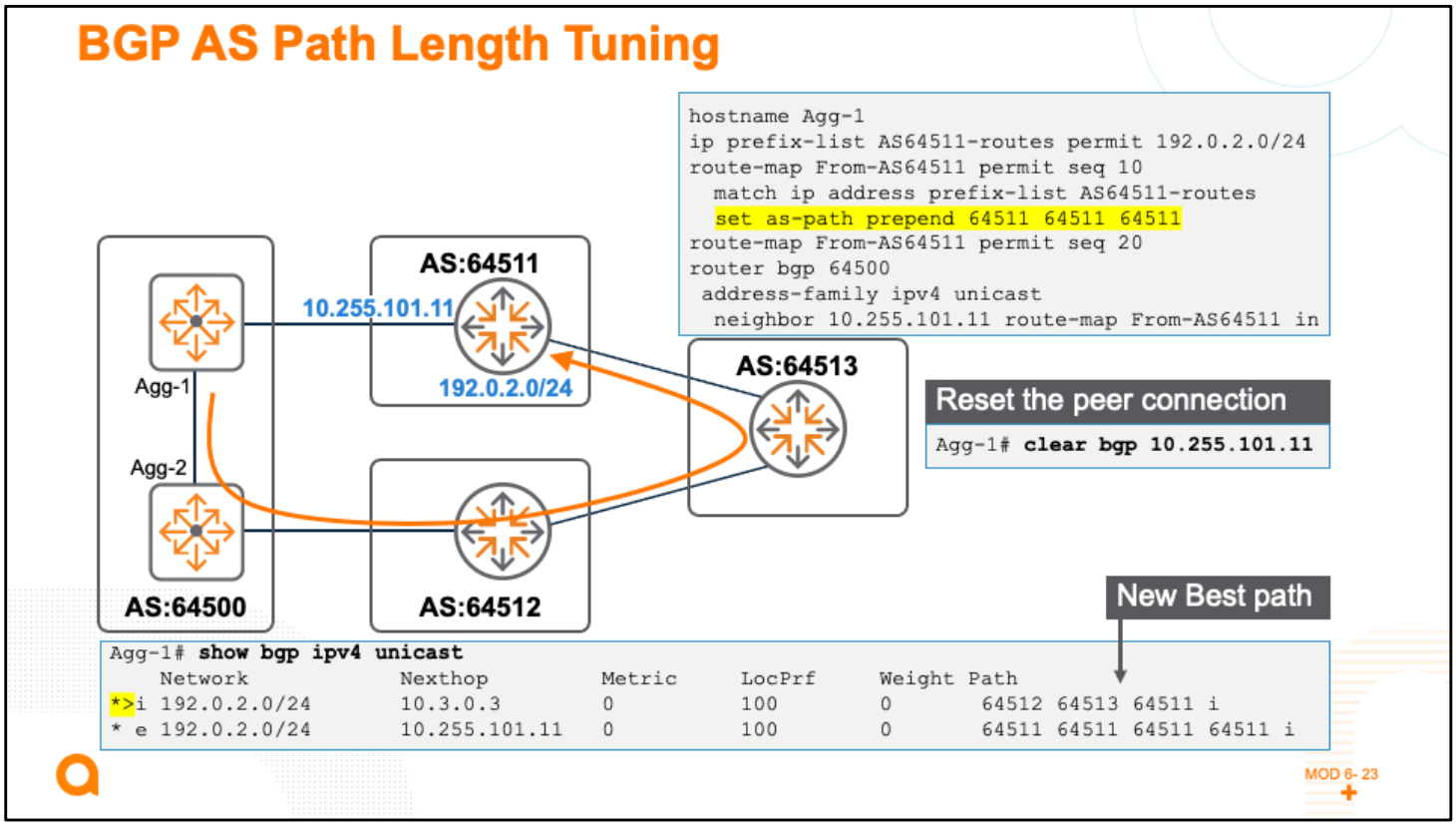


As you saw on the previous page, BGP has many metrics that you can define to manipulate routing policies. If no metrics are defined (weight or local preference), then the AS path length (number of AS hops to reach the destination network) is used. The AS path attribute is also used by BGP to detect routing loops based on path choices.

In the figure, all other things being equal, the best path to the 192.0.2.0/24 network is out Agg-1, directly to AS 64511 – 1 hop away. The path out Agg-2 via AS 64512 is 3 hops or ASes away. A shorter AS path is not necessarily the best path – thus the various metrics you can adjust to influence path choice. Perhaps the link between Agg-1 and AS 64511 is very slow compared to the other paths.

You can see this in the show bgp ipv4 unicast output. Use this command to see all discovered BGP paths, and what BGP believes to be the best paths. The output is edited to only show you the target route in question. The highlighted greater than (>) sign indicates that, of the two possible paths, this one is the best path. Why? The weight and local preference are the same, and so “best” is determined by AS path length. You might tune metrics to get your routes to use the other path.

Note: By default, BGP does not do load sharing: the first “best” path used is what is placed in the routing table unless you enable equal-cost multipath (ECMP) for BGP.



For policy reasons, you may want to use a path other than that with the shortest path length. In the example, you want traffic to 192.0.2.0/24 to go out Agg-2, via AS 64512 > 64513 > 64511. One solution is to implement AS pre-pending, as shown in the figure.

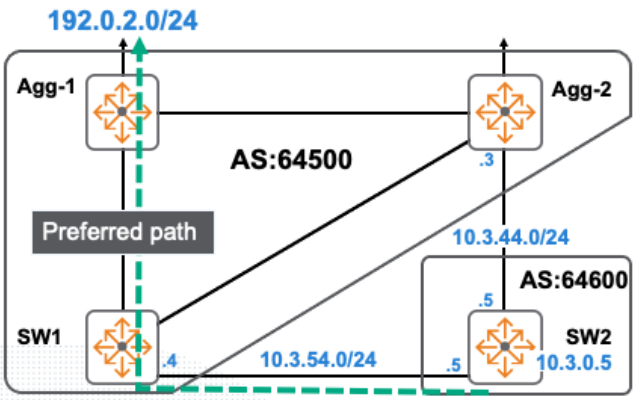
A prefix list is applied to a route map as match criteria, and the AS 64511 is prepended to this route. This route map is applied to the 10.255.101.11 peer inbound. If you were to check the BGP table at this point, you would probably not see a change. New BGP policies are typically applied when the peer is established. So, you must clear the peer connection, which forces it to reform. Now that you know about the need to clear connections, this will not be mentioned in the following tuning discussions. It is now assumed that you will know to do this.

Now when you check the BGP table, you can see that the AS path prepend worked, and the desired best path is now selected.

Do not forget to add the “permit any” statement at the bottom of your route map, as shown in the figure - route-map From-AS64511 permit seq 20. Otherwise, the route will only advertise the routes that are tuned from sequence number 10. The implicit “deny all” at the bottom will filter all other routes from being advertised.

Multiple Exit Discriminator (MED)

- Suggests how a multi-homed eBGP peer should exit their AS, to enter your AS
- Lower value is preferred




```
hostname Agg-2
ip prefix-list AS64500-routes permit 192.0.2.0/24
route-map To-AS64600 permit seq 10
  match ip address prefix-list AS64500-routes
  set metric 200
route-map To-AS64600 permit seq 20
router bgp 64500
  address-family ipv4 unicast
    neighbor 10.3.44.5 route-map To-AS64600 out
```

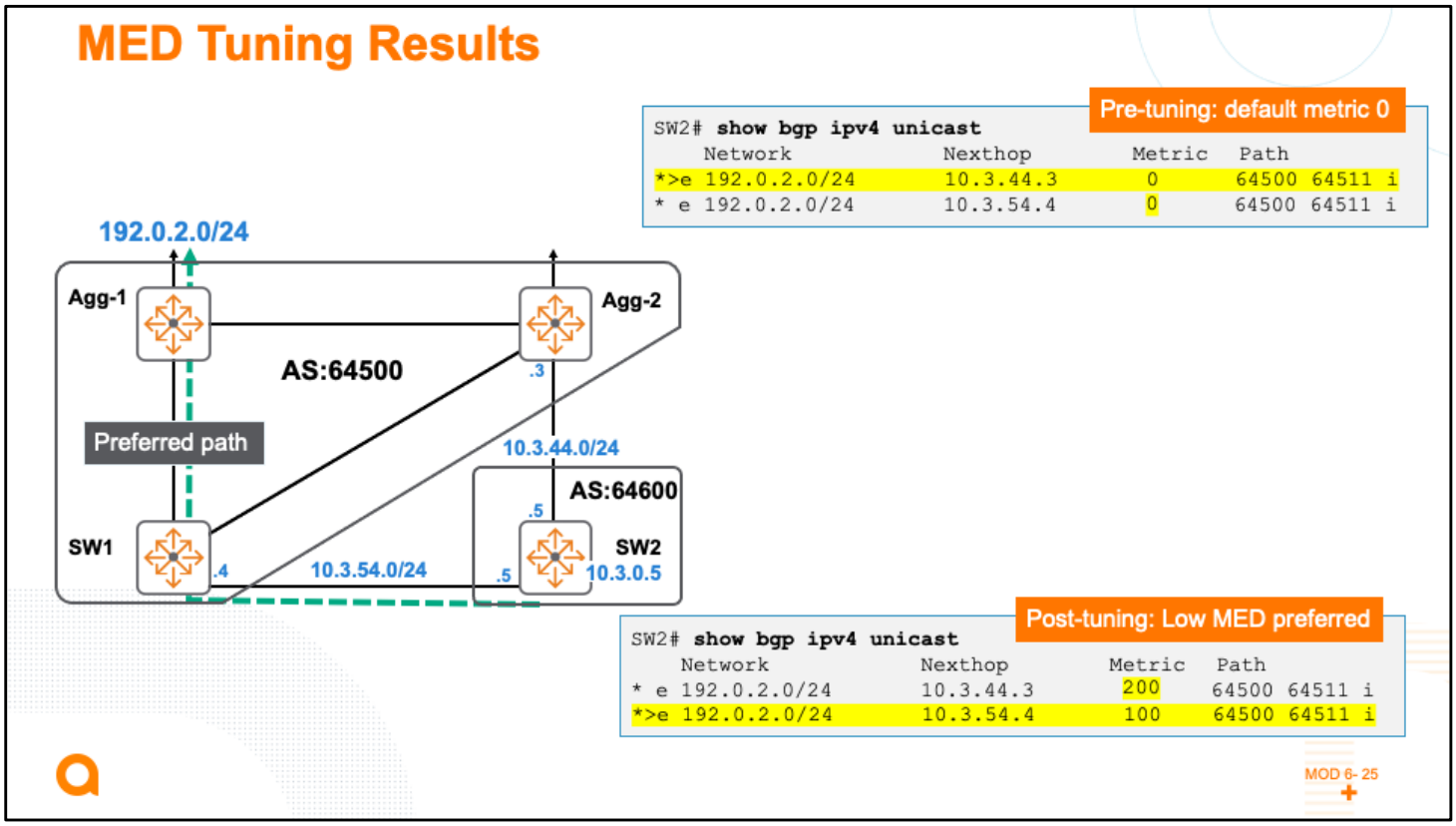
```
hostname SW1
ip prefix-list AS64500-routes permit 192.0.2.0/24
route-map To-AS64600 permit seq 10
  match ip address prefix-list AS64500-routes
  set metric 100
route-map To-AS64600 permit seq 20
router bgp 64500
  address-family ipv4 unicast
    neighbor 10.3.54.5 route-map To-AS64600 out
```

MOD 6- 24

The Multiple Exit Discriminator (MED) attribute is shared with a multi-homed eBGP peer to influence how they will enter your network. The figure shows how AS64500 and 64600 are multi-homed – one connection via the 10.3.54.0 link, and another via 10.3.44.0. This is the use case for MED. You would configure your routers in AS 64500 to send different MED values to the AS 64600 peer. This is a suggestion to these eBGP peers, “When packets exit your network, en route to my network, please use Path A for some routes and use path B for other routes”.

AS 64500 might be a large cloud with thousands of subnets. You know that it is more efficient reach some of your networks via the .44 link, while others should use the .54 link. So you make this suggestion to the external AS to forward traffic accordingly. Remember it is called an Autonomous System because someone has autonomy (control) over it. You cannot force the AS to pay attention to your MED values. You would need to work with them as a partner, tell them of your intentions, and ask that they configure their routers to heed your request.

In the example, it makes more sense for traffic destined to 192.0.2.0/24 to enter AS 64500 via SW1. It can thus go straight to Agg-1, and then 1 more hop to AS 64511 (not shown).If AS 64600 routers use Agg-2 for this route, it might take a more circuitous route. So, you create the configuration shown in the figure, in which SW1 advertises a lower MED for that route than Agg-2.



The figure shows the pre- and post-tuning path selection for router SW2, in AS 64600. After tuning, the router sees the lower MED (aka Metric) value and prefers to reach 192.0.2.0/24 via 10.3.54.4.

Remember that AS 64600 is not compelled to comply with this request from AS64500 about MED. They could adjust the weight on their router and over-ride the MED values.

Question #2

```
# show bgp ipv4 unicast
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, e external S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

VRF : default
Local Router-ID 10.1.0.3

    Network            Nexthop         Metric     LocPrf     Weight Path
*>i 100.100.1.0/24     10.1.0.2        0          100        0       i
s>  100.100.1.0/26     0.0.0.0         0          100        0       i
```

From the output, what would be the next hop used for a packet destined to 100.100.1.145?

a. This packet cannot be routed

b. 0.0.0.0

c. 10.1.0.2



Knowledge Check

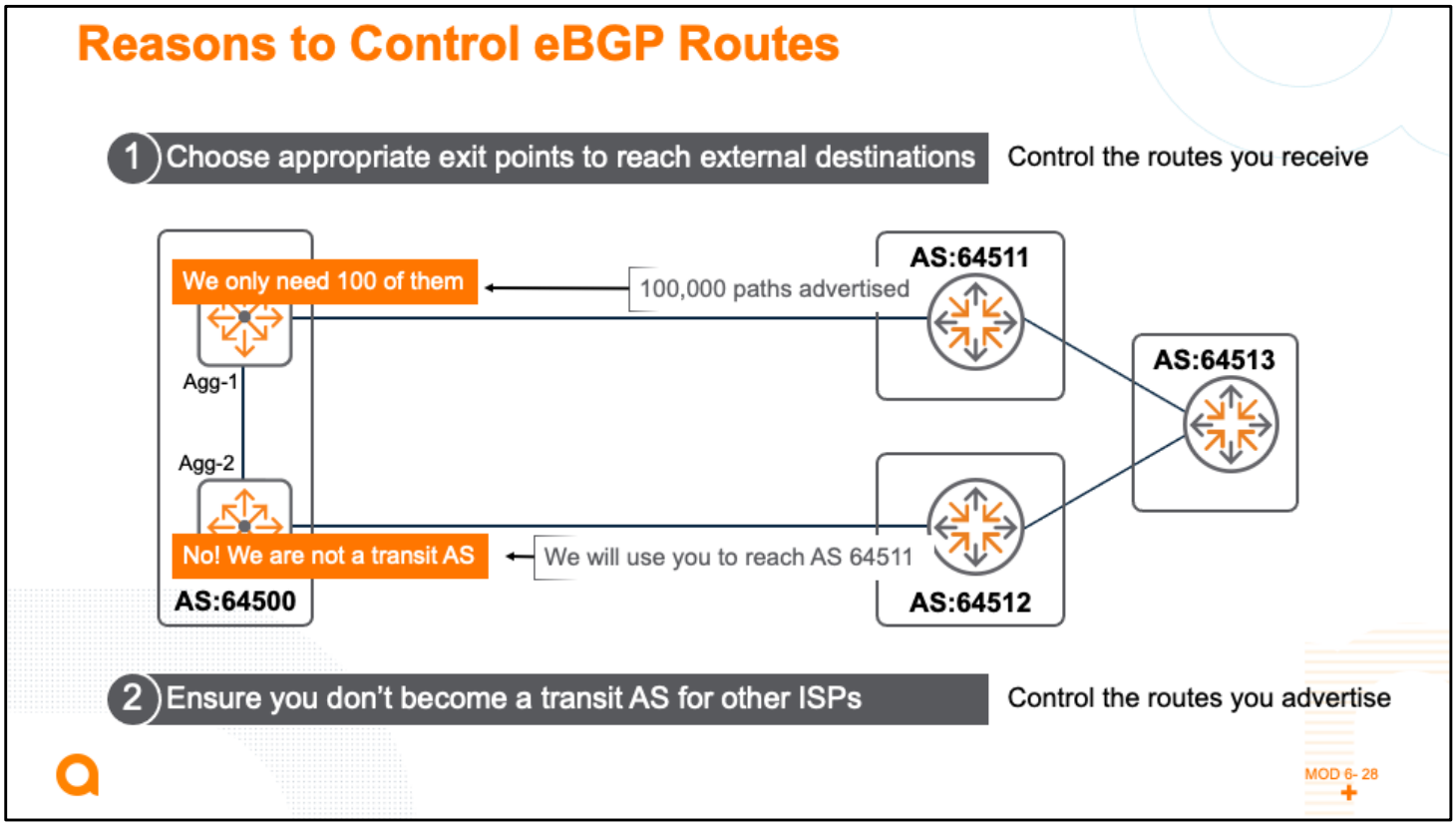
Correct answer:

C

aruba
a Hewlett Packard Enterprise company

Route Control

Techniques, peer groups, route reflectors



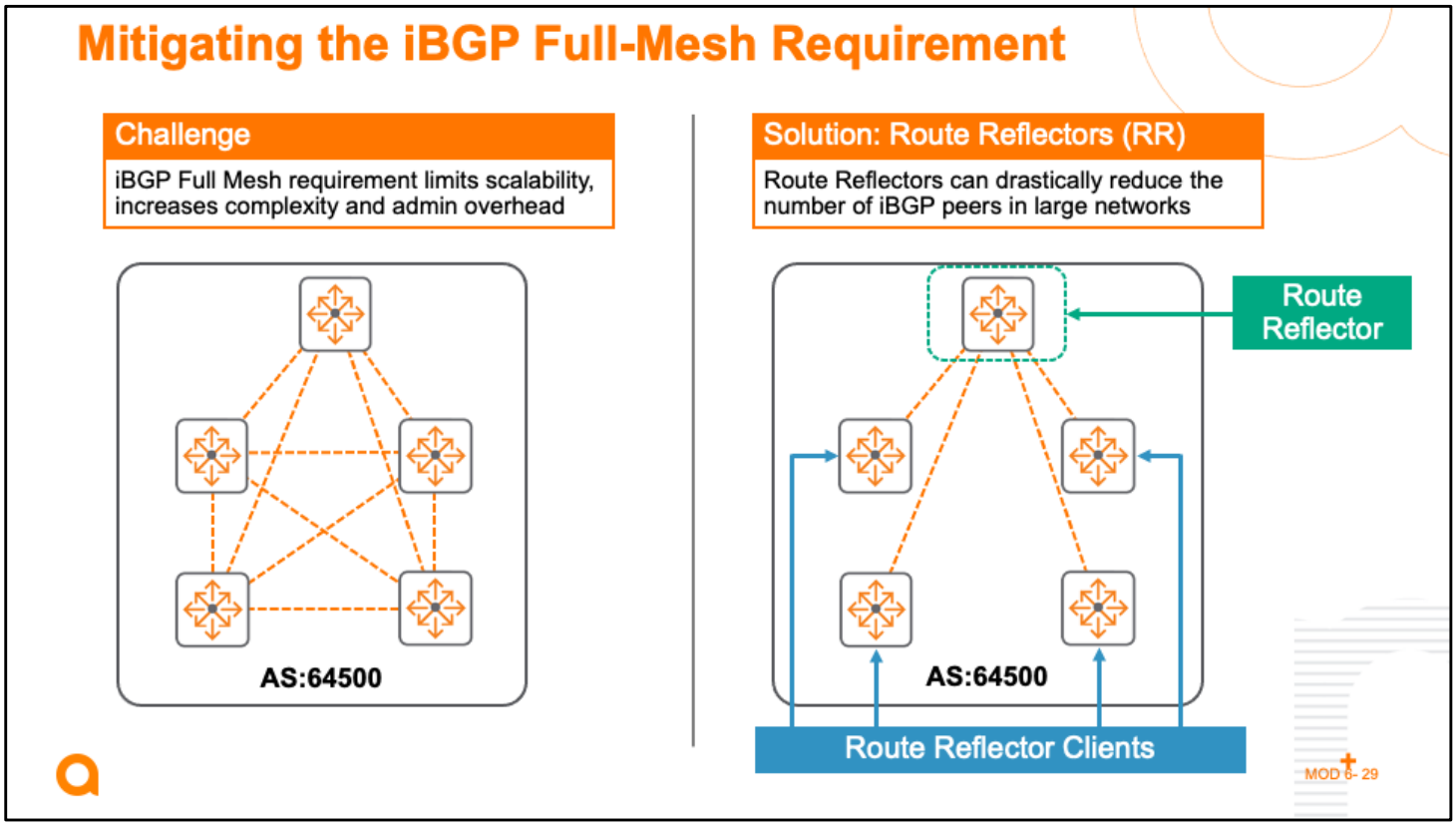
There are many reasons to control external BGP (eBGP) routes. The two most common are:

Choose appropriate exit points to reach external destinations

Ensure you don't become a transit AS for other ISPs

For example, if you have resources in ISP1 that you commonly access, it makes sense to use your ISP1 connection to reach them, as opposed to using another ISP. Second, you don't want ISPs to use your company's bandwidth to reach other ISPs, which is referred to as a transit AS. Based on these two reasons, you will likely need to implement eBGP route control using prefix or AS path lists to control the external BGP routes you receive and/or advertise.

Note: Use Looking Glass to verify the results of your configuration. Find it here: https://www.gin.ntt.net/looking-glass/



Recall that the iBGP full-mesh requirement stems from the fact that iBGP peers do not advertise a prefix learned from other iBGP peers. This leads to scalability issues within an AS. It increases configuration complexity and administrative overhead.

The formula n(n-1)/2 provides the number of sessions required where n represents the number of routers. A full mesh topology of 6 routers requires 15 sessions. A topology of 100 routers requires 4950 peer sessions – this becomes an issue for large networks.

RFC 1966 defines a route reflector solution, such that certain IBGP peers can be configured to reflect routes to other iBGP peers. You chose one or two centrally located routes and configure them as Route Reflectors (RR). All other routes need only peer with the RRs, drastically reducing the number of peer sessions. You do not configure RR clients as such – they are unaware that any route reflection occurs. You simply configure them to peer with the RRs as normal.

Three basic rules apply to route reflectors and route reflection.

1. If an RR receives an NLRI from a non-RR client, the RR advertises the NLRI to an RR client. It does not

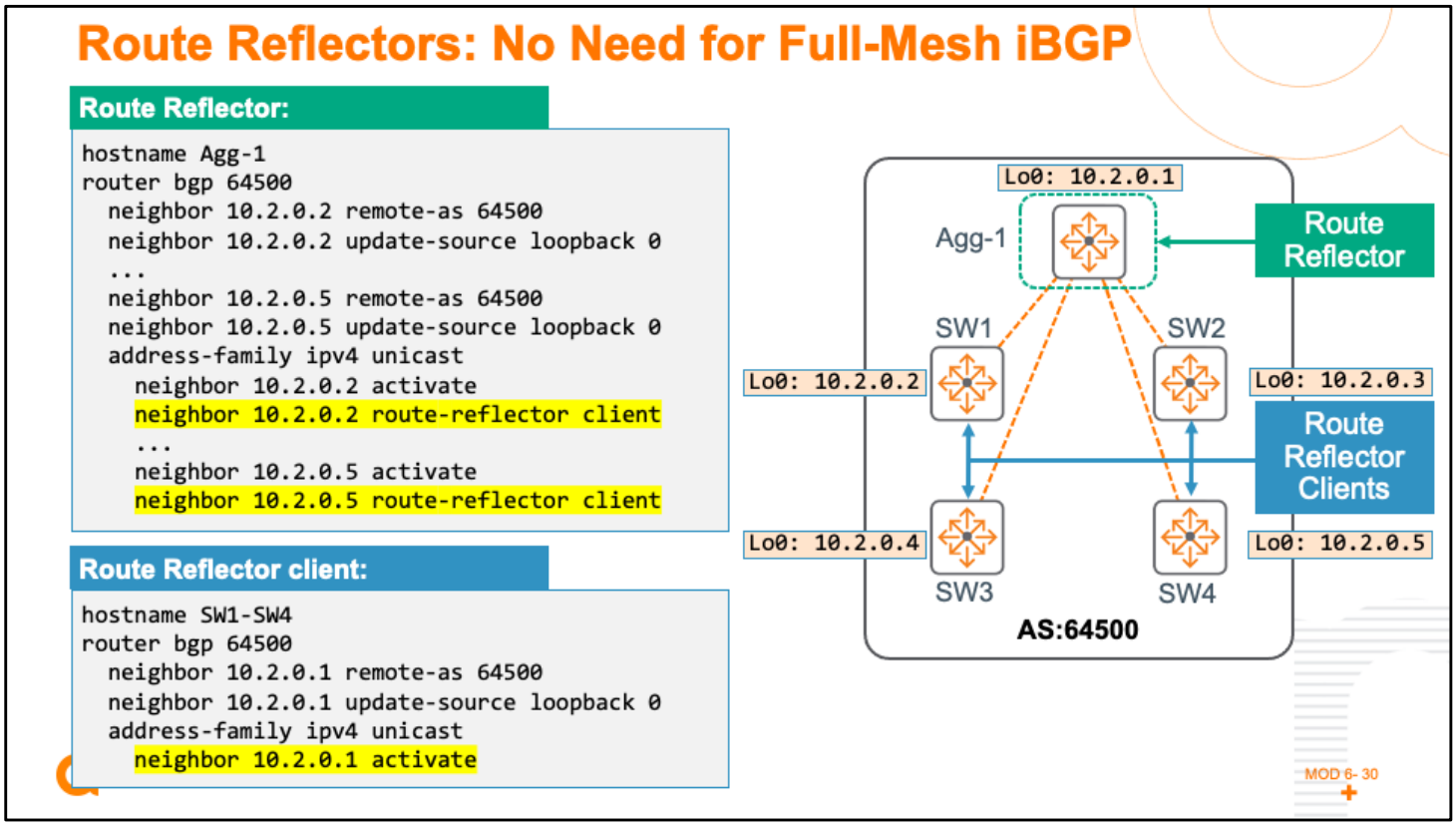
advertise the NLRI to a nonroute reflector client.

2. If an RR receives an NLRI from an RR client, it advertises the NLRI to RR and non-RR clients. Even the RR

client that sent the advertisement receives a copy of the route, but it discards the NLRI.

3. If an RR receives a route from an EBGP peer, it advertises the route to RR and non-RR clients.

Let’s look at a typical Route Reflector configuration.

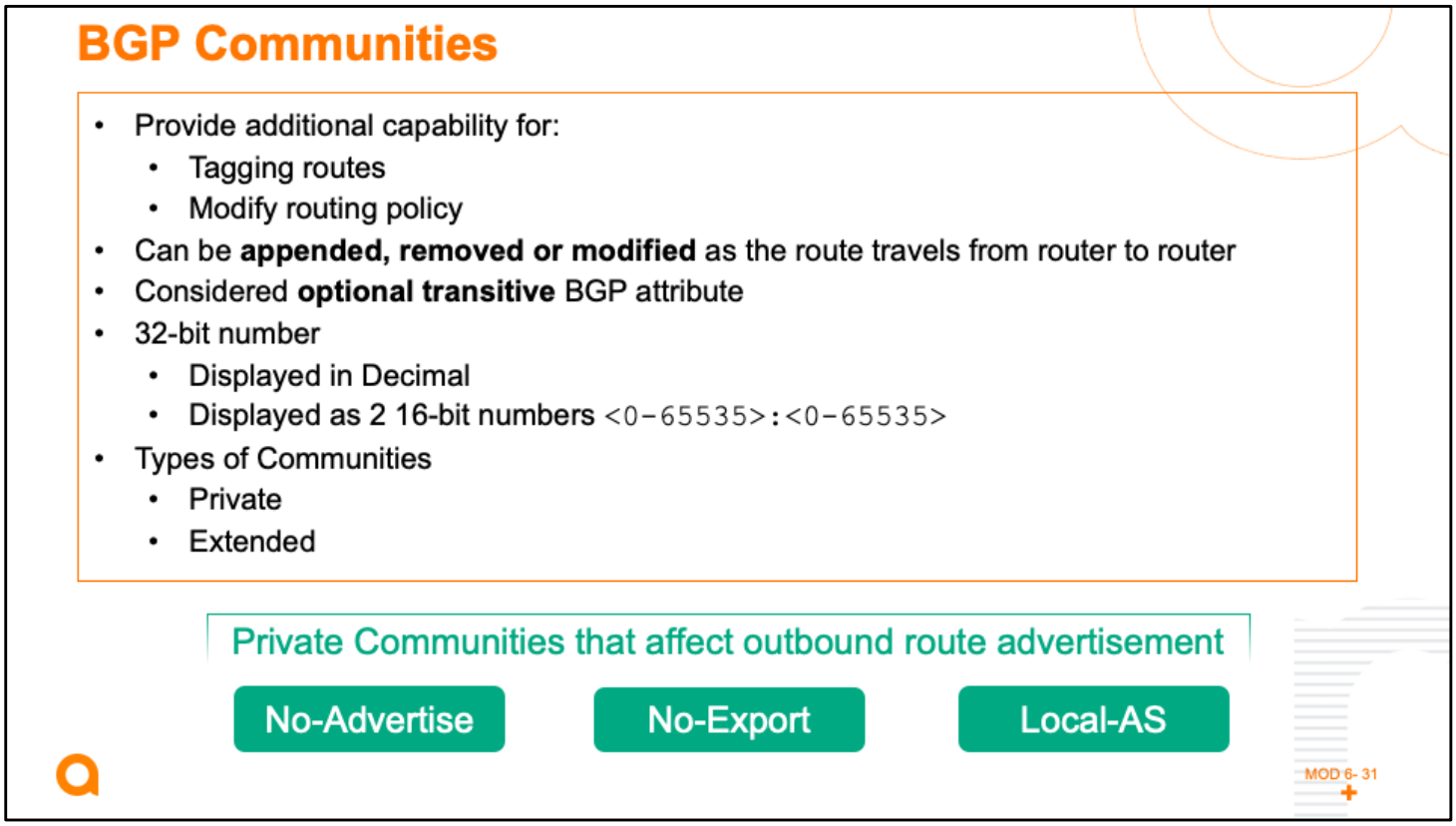


In an AS, a single iBGP router (preferably two for redundancy) can be configured as a route reflector. Other iBGP routers in the AS need only be configured as peers to the route reflector(s), as shown in the figure. Route reflectors reflect or “mirror” iBGP routes to other iBGP peers. In the figure Agg-1 is configured as route reflectors. All other IBGP routers act as route reflector clients.

Use your same show and debug commands to troubleshoot route reflectors as you would a non-reflector deployment. The main difference is that with fewer peers, things are easier to deploy, maintain, and diagnose.

Clusters

A router reflector(s) and its iBGP clients form a cluster. A cluster-ID is associated with every route advertised by the route reflector to its client and/or non-client peers. The cluster-ID is a cumulative, non-transitive BGP attribute and every route reflector must prepend the its local cluster-ID to the cluster-list in order to avoid routing loops. By default, the cluster-ID is the router-ID. The command is shown in the example so you know how to configure it. Route reflectors reduce the number of iBGP peerings to each router and thus reduce processing overhead. To provide scalability, clusters can be interconnected.



BGP communities provide additional capability for tagging routes and modifying BGP routing policy on upstream and downstream routers. They can be appended, removed, or modified on each attribute as the route travels from router to router.

BGP communities are an optional transitive BGP attribute that can traverse from autonomous system to autonomous system but non necessarily has to be recognize by all BGP routers. A BGP community is a 32-bit number that can be included with a route. It can be displayed as two 16-bit numbers (0-65535):(0-65535).

Private BGP communities follow the convention that the first 16-bits represent the AS of the community origination, and the second 16-bits represent a pattern defined by the originating AS. The private BGP community pattern could vary from organization to organization, do not need to be registered, and could signify geographic locations for one AS while signifying a method of route advertisement in another AS.

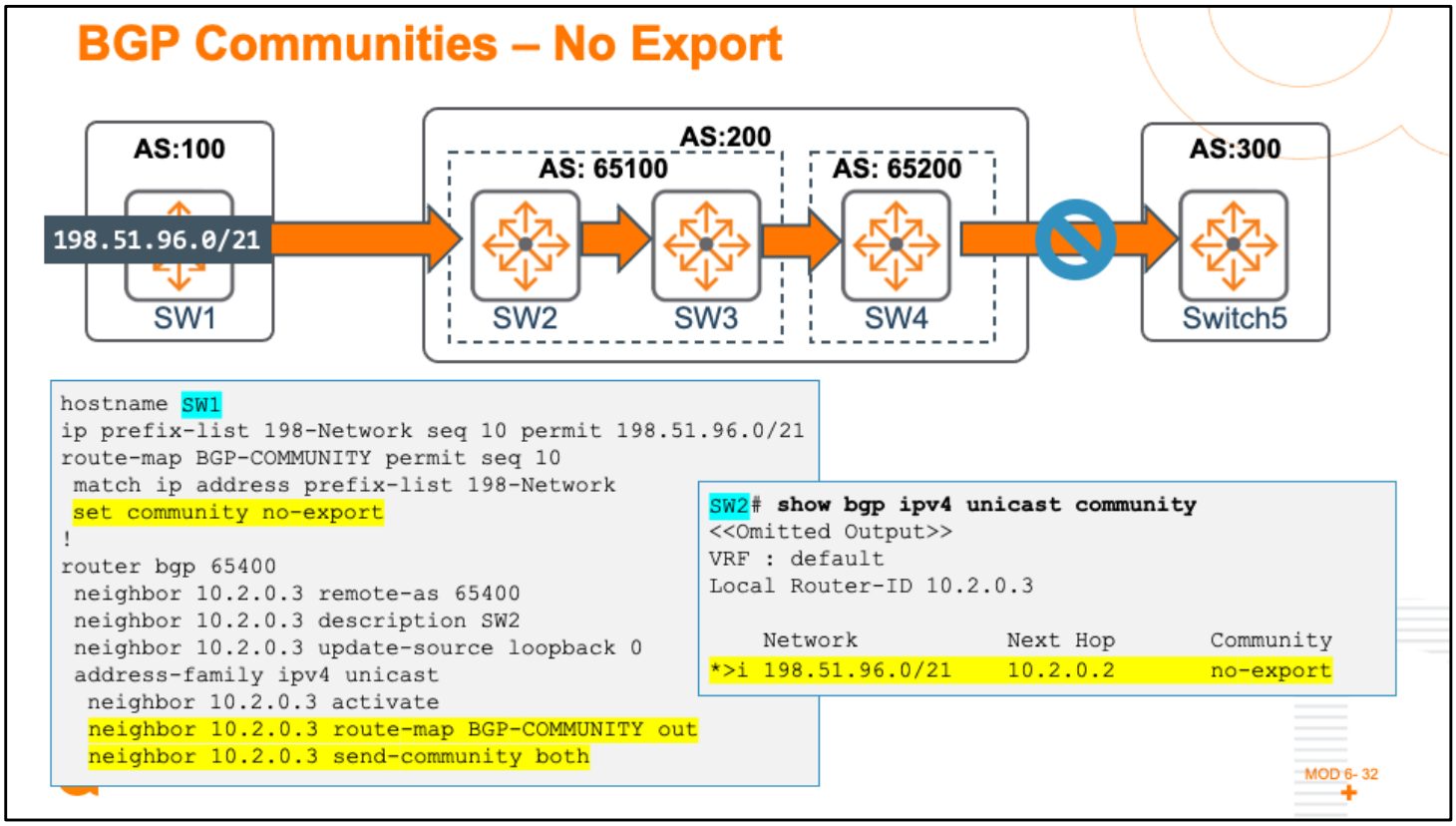
The Extended Community Attribute provides a mechanism for labeling information carried in BGP-4. It provides two important enhancements over the Private BGP Community Attribute:

An extended range, ensuring that communities can be assigned for a plethora of uses, without fear of overlap.

The addition of a Type field provides structure for the community space.

The addition of structure allows the usage of policy based on the application for which the community value will be used. For example, one can filter out all communities of a particular type or allow only certain values for a particular type of community. It also allows one to specify whether a particular community is transitive or non-transitive across an Autonomous System (AS) boundary.

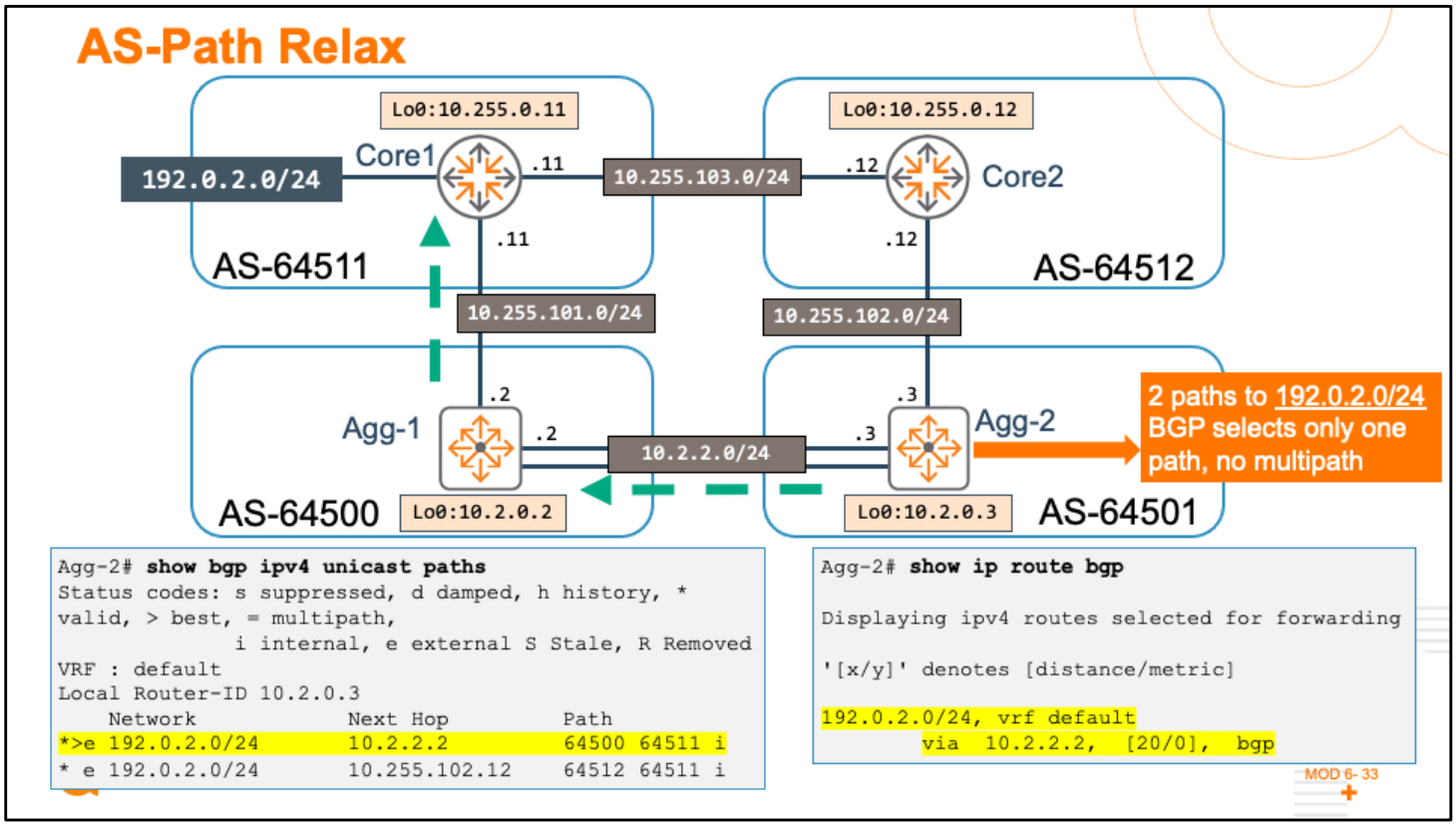
Private BGP communities are used for conditional matching for a router's route policy, which could influence routes during inbound or outbound route-policy processing. There are three well-known communities that affect only outbound route advertisement: No-Advertise, No-Export, and Local-As. Routes that contain these communities can be displayed with the command show bgp ipv4 unicast community {local-as | no-advertise | no-export}.



The No_Export community (0xFFFFFF01 or 4,294,967,041) specifies that when a route is received with this community, the route is not advertised to any EBGP peer. If the router receiving the No-Export route is a confederation member, the route can be advertised to other sub-ASs in the confederation.

The figure demonstrates that SW1 is advertising the 10.1.1.0/24 network to SW2, which advertises to SW3, and then on to SW4. SW4 does not advertise the prefix on to Switch5, so this device does not have the 10.1.1.0/24 network in its BGP table.

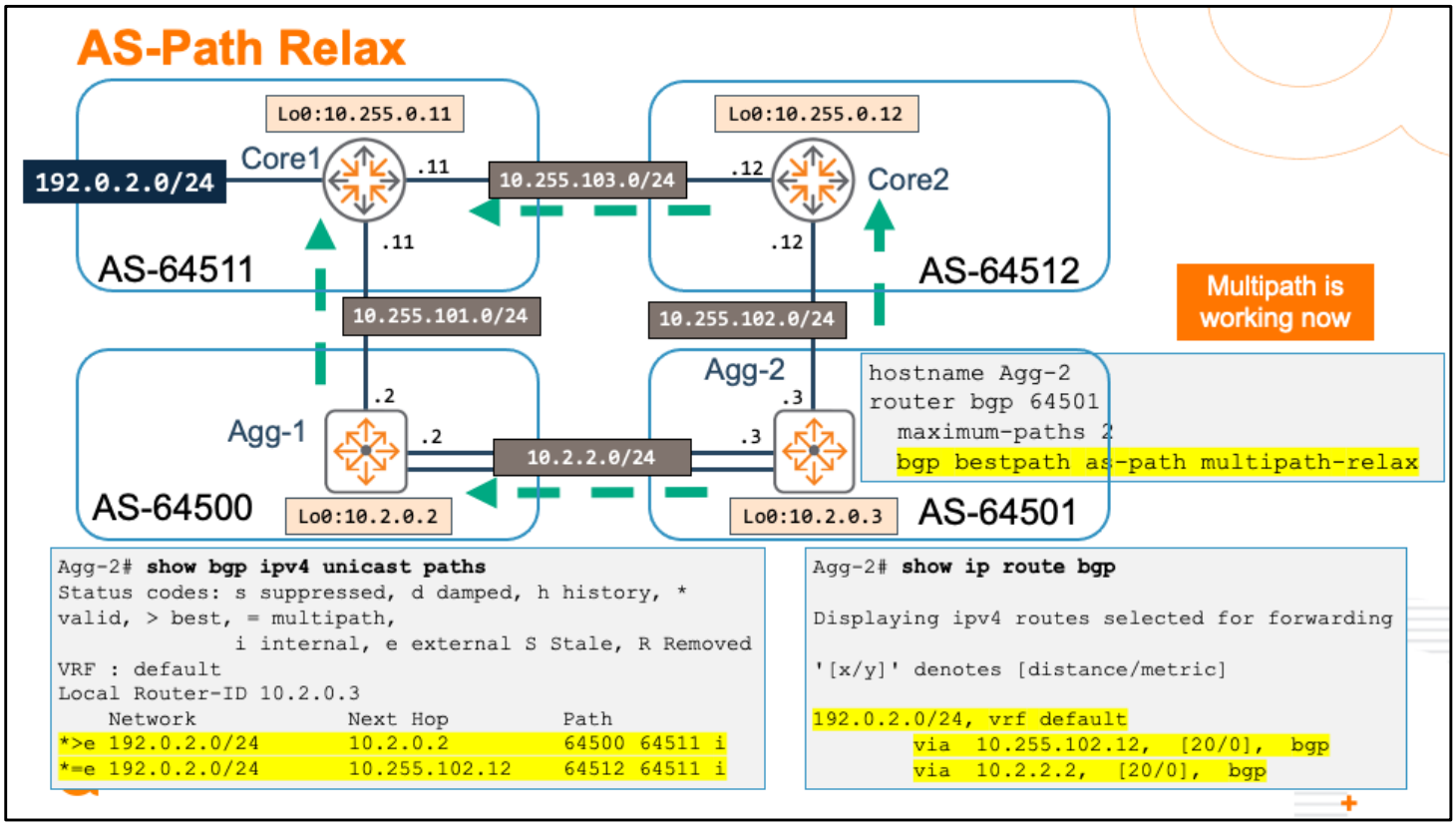
The configuration is similar to the previous example with the difference that we set the community no-export in the route-map. To verify the community received you must issue the show bgp ipv4 unicast community command.



When multiple identical BGP routes are valid and have the same administrative distance, BGP uses the attribute criteria previously described in this module to select a single path for a specific destination.

By running the show bgp ipv4 unicast paths command we can verify that Agg-2 switch has two possible paths to the prefix 192.0.2.0/24, however Agg-2 selects the route through the AS 64500 because the router-id of the neighbor in this AS has a lower value (10.2.0.2) compared with the neighbor in AS 64512 (10.255.0.12), this information can be verified by running the show ip route bgp command

Route load balancing in this situation will not work.



The bgp bestpath as-path multipath-relax command configures Border Gateway Protocol (BGP) to treat two BGP routes as equal cost even if their AS-paths differ, as long as their AS-path lengths and other relevant attributes are the same. This allows routes with different AS-paths to be programmed into the forwarding table as equal cost multipath routes.

Note. When you applied the command, the system will need to restart the BGP sessions, you will be asked if this is what you want.

switch(config-bgp)# bgp bestpath as-path multipath-relax

All current BGP sessions in VRF default will be restarted.

Do you want to continue (y/n)?

In the figure, you can see now how the FIB is using two next-hops for the same prefix (show ip route bgp). When you run the show bgp ipv4 unicast paths command, you can see the flag “=“ in one of the entries which indicates that this device is running multipath for the specified prefix.

Question #3

```
(config)# hostname AGG2
(config)# ip aspath-list bgp-local permit ^$
(config)# route-map bgp-ebgp-out permit
(config-route-map-bgp-ebgp-out-10)# match aspath bgp-local
(config-route-map-bgp-ebgp-out-10)# exit
(config)# router bgp 64500
(config-bgp)# address-family ipv4 unicast
(config-bgp-ipv4-uc)# neighbor 10.255.102.12 route-map bgp-ebgp-out out
```

What is the purpose of the configuration?

a. This configuration removes completely the AS Path attribute when advertise routes to the neighbor 10.255.102.12

b. Advertise to any eBGP neighbor routes originated in the AS 64500

c. Advertise to the eBGP neighbor 10.255.102.12 only routes originated in the AS 64500 preventing a transit AS



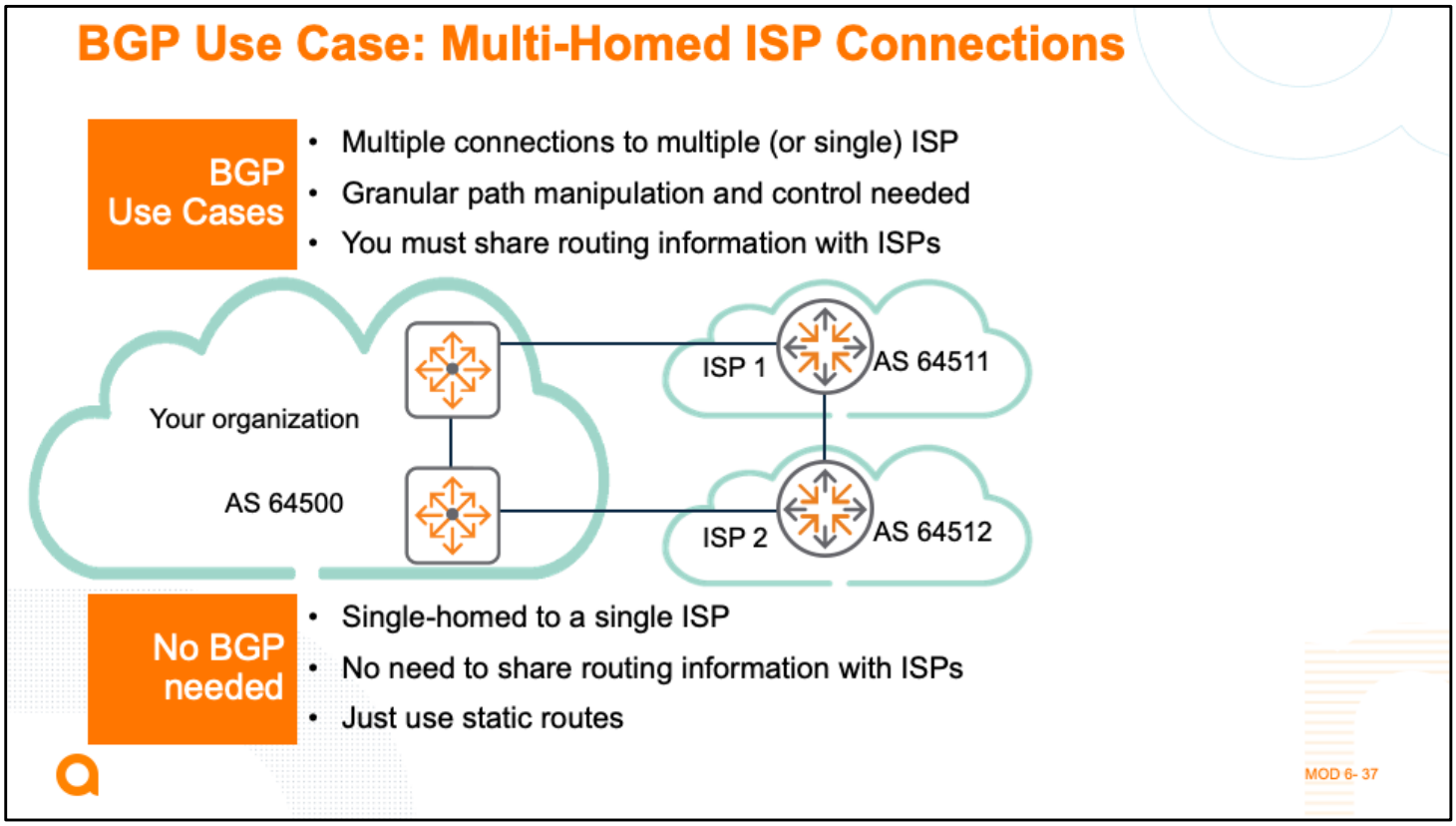
Correct answer:

C

aruba
a Hewlett Packard Enterprise company

Additional Content

Module 06

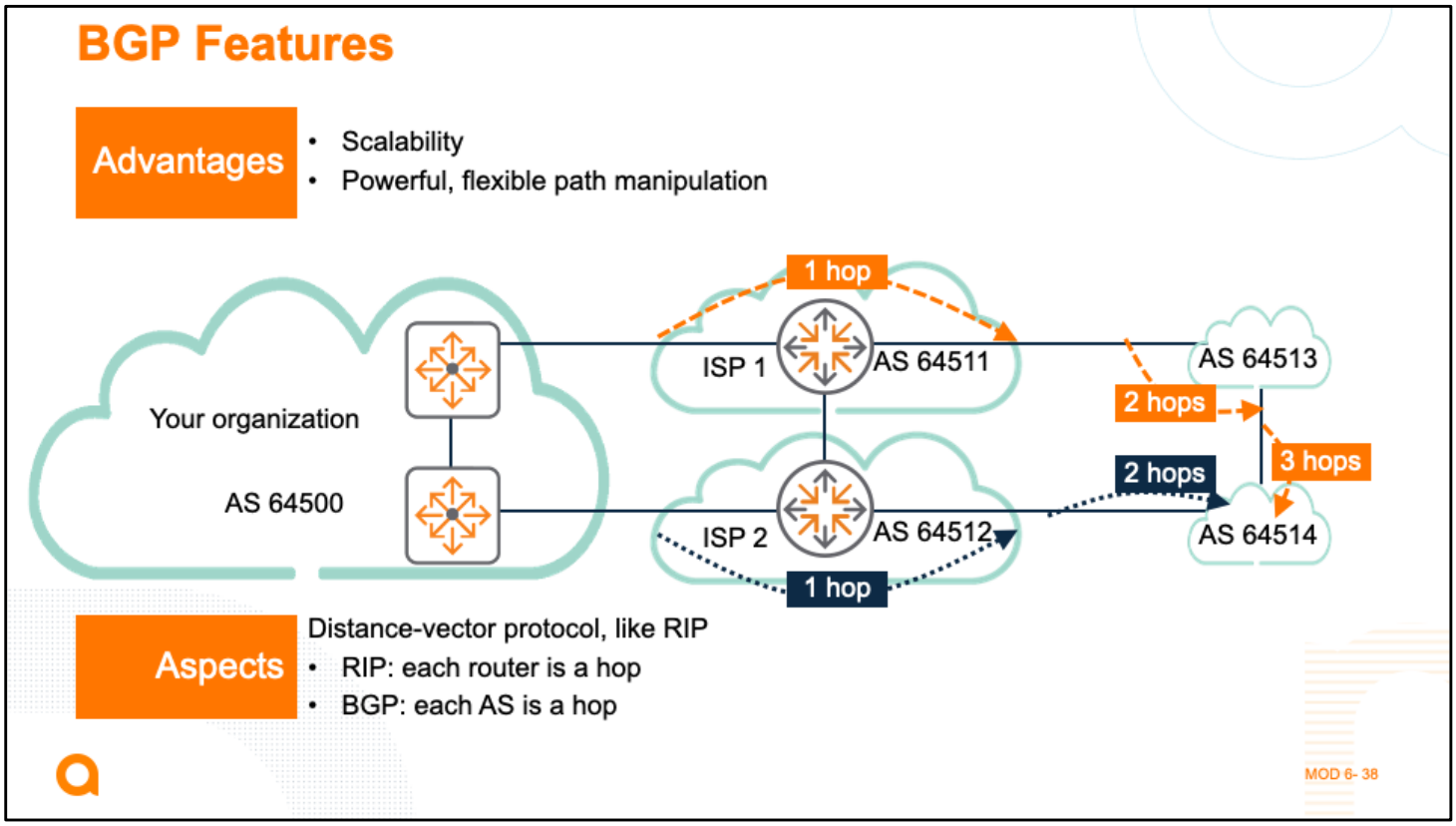


BGP is an exterior gateway protocol (EGP), which means that it communicates routes between different organizations, typically Internet Service Providers (ISPs). The figure illustrates BGP's primary use case - a company must communicate with multiple ISP routers. Another use case is when a large company connects to a single provider, and you need more path manipulation and control than is available with simple static routes. In other words, You only need to do so when the company specifically needs to communicate routing information with the ISP.

You also might need to advertise different IP blocks over different connections. This allows the ISPs to advertise your company's networks to the rest of the world, such that some paths may be best reached via ISP1, while others are best reached via ISP 2.

In many cases, BGP is not required. If you only connect to a single ISP, and no granular path control or manipulation is needed, you can simply use static routes, reaching all internet sites via your ISP routers. The ISP handles announcing the block of public IP addresses assigned to the company.

The figure's dual-ISP example shows a different router connected to each ISP. This improves resiliency over a single connection but can be slightly more complex because it requires two BGP routers, which communicate with each other as well as the ISP routers.



Because BGP is designed for use on the Internet, it can handle millions of routes. The protocol also permits a high level of administrative control for each organization. Organizations can aggregate routes, filter routes, and manipulate attributes in routes.

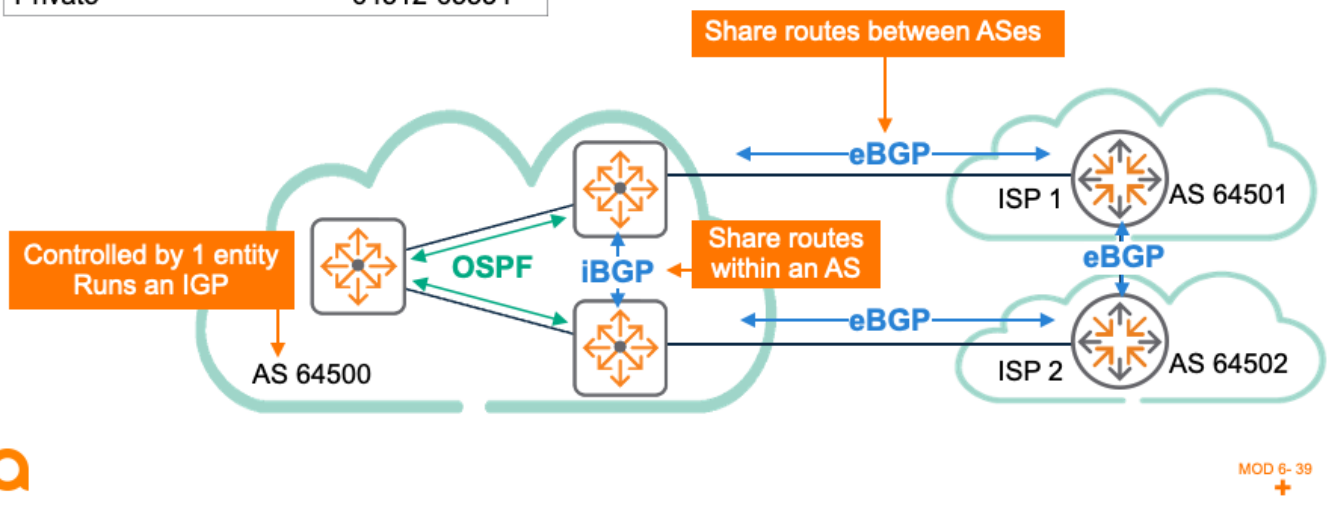
Like RIP, BGP is a kind of distance-vector routing protocol. However, with RIP the “hop count” is based on the number of routers to traverse or hop over. With BGP, it is based on the number of ASs to traverse. As shown in the figure, using ISP 1 your company is three ASs away from any subnet inside AS 64514 (including the target AS). However, your company must only traverse two ASs to those subnets via ISP 2. All other things being equal, the shortest path is considered the best path. You will soon learn to use other criteria to control BGP path decisions – you will soon be the BGP puppet master!

So, if you don’t care about pathing into or out of your AS, you can simply use static routes. Some paths may be sub-optimal, but in many scenarios, it will not affect the user experience, nor your performance or security requirements. However, if you need this level of control, you need BGP.

You can also use VSF to combine two physical AOS-CX switches that support BGP, such as 6300 switches. You can then connect to the two ISPs on different physical chassis, but still retain the simplicity of a single (logical) BGP router for both connections.

BGP AS

Purpose	AS numbers
Reserved	0 and 65535
Public IANA-assigned	1-64495
Documentation/examples	64496-64511
Private	64512-65534



BGP is an external gateway protocol (EGP) used to connect different autonomous systems, between a company and an ISP, multiple ISPs, or even segments within a large private network.

The AS forms the fundamental building block of BGP, and each organization such as a company or ISP generally has its own AS, identified by a unique number. Originally, BGP supported only 16-bit AS numbers, ranging between 0 and 65535. The figure shows how AS numbers are used.

Because 16-bit numbers allow for only 65535 unique numbers, Request for Comments (RFC) 4893 establishes 32-bit AS numbers, increasing the range of AS numbers to 4 billion. (AOS-CX switches supports 32-bit numbers in one of two formats:

1-4294967295: The autonomous system number (ASN) for the domain.

0.1-65535.65535: The autonomous system number (ASN) for the domain in dotted decimal format.

When a company determines that it needs to run BGP in a multihomed scenario, it must obtain a valid public AS from the proper regional authority.

BGP operates in one of two forms depending on the autonomous systems to which the communicating neighbors belong:

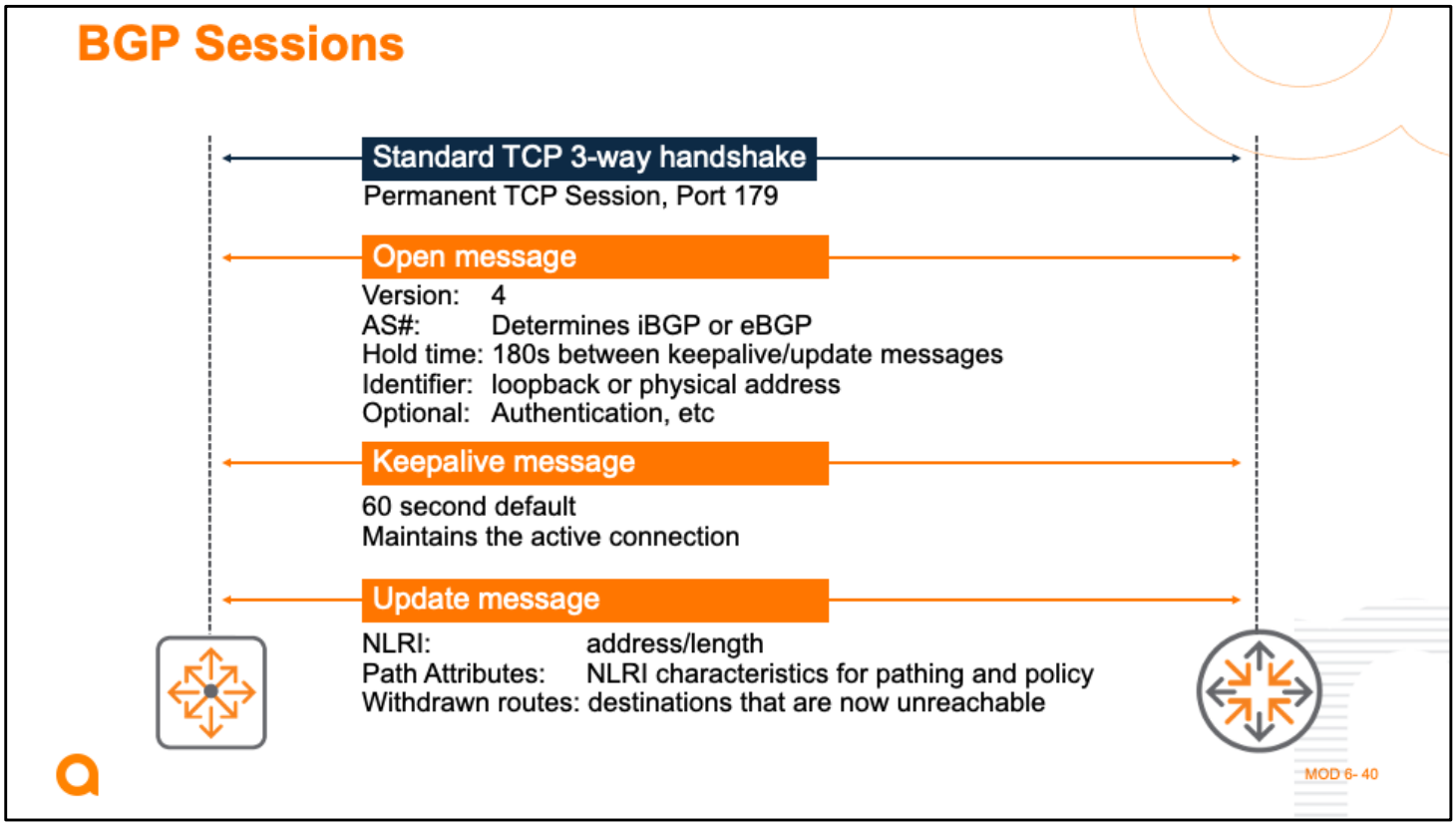
Neighbors in different autonomous systems communicate with external BGP (eBGP).

Neighbors in the same AS communicate with internal BGP (iBGP).

The same router might use eBGP with one neighbor and iBGP with another.

Please understand that iBGP is not the only routing protocol running within the AS. The AS also runs an interior gateway protocol (IGP) such as OSPF or RIP, which is localized to the AS and handles routing the local traffic. Typically, just a few routers in the AS—the routers that connect to other autonomous systems and perhaps some core routers—run BGP in addition to the IGP.

Note: In this course and in the lab material, private AS numbers will be used.



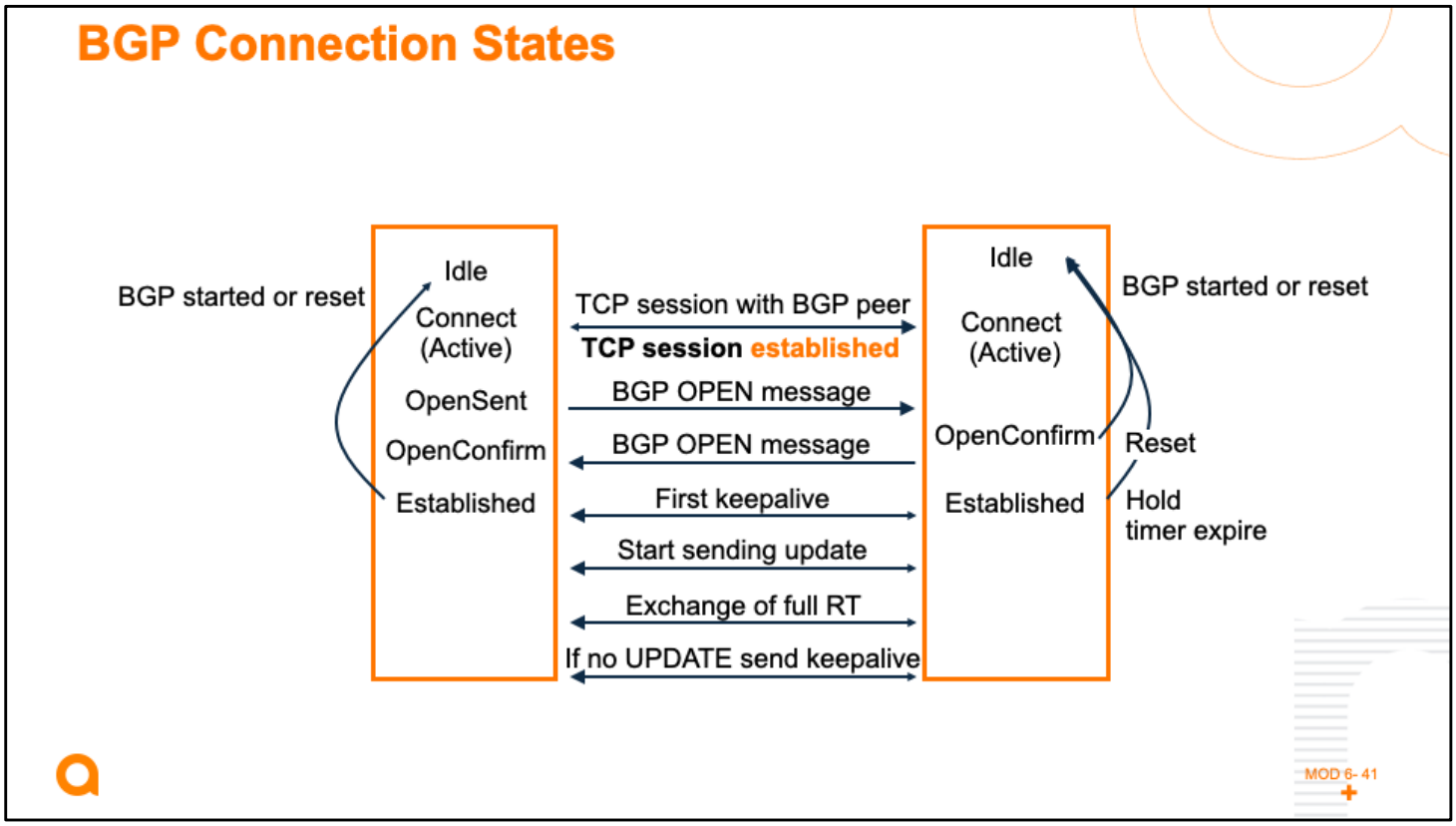
BGP neighbors establish a TCP session on port 179. They then open a BGP connection within the session. This open message contains the version of BGP in use (version 4), along with the AS number. This is how routers know if they are iBGP or eBGP neighbors. The hold time determines the maximum time between keepalive and update messages. This defaults to 180 seconds, or three times the default keepalive time of 60 seconds. If a router stops hearing from a peer, it knows that peer is down. The identifier is the router-ID – typically the loopback 0 IP address. There are also optional parameters to support things like authentication and multi-protocol functions.

To keep the session active, there is a 60-second keepalive message that basically tells the peer, “Hey! I’m still here”.

Once the peers successfully form, routers can exchange their BGP routing tables. BGP terminology often refers to this as Network Layer Reachability Information (NLRI). That’s just a fancy way to say “the prefix and its mask” – 100.10.3.0/24, and 198.198.101.0/24 for example.

For each address/length pair, BGP also advertises a rich set of path attributes – characteristics of each subnet that helps to optimize path decisions and enforce your corporate policy for what should be advertised, and to whom.

Update messages are also used to advertise withdrawn routes – those which have become unavailable. This can be due to a network outage, or because you decided to permanently eliminate or re-address certain subnets.



In the lab, you will see how BGP neighbors pass into Active state and stay there when they encountered issues. You will see that a healthy session should have the Established state.

To establish the BGP connection, the neighbors first open the TCP session on port 179. They then send BGP open messages with a list of parameters for the session and, assuming the parameters match, finally send the first keepalive of the session.

You can trace the states through which a BGP connection might pass:

The BGP connection begins in the Idle state and returns to this state when the session establishment times out without success or an error occurs.

The connection typically transitions immediately to the Connect state, in which the router attempts to contact its neighbor at TCP port 179 and waits for a response.

If the router is set to Passive TCP mode for this neighbor, the session transitions to Active mode, rather than to Connect mode. In this mode, the router waits for the neighbor to initiate the session.

The connection also moves to the Active state if the router does not receive a response to its TCP session initiation. In this state, the router waits for the neighbor's response and also, after a timer expires, might try to initiate the connection again. The router might wait in Active state for a significant period of time if the TCP session fails to establish. If you see a router in Active mode after several minutes, the neighbor might be unreachable.

After the BGP neighbors have established the TCP session, the router sends an open

message, either immediately or after a configurable delay. The connection also transitions to the OpenSent state or directly to the OpenConfirm state depending on whether the router has already received the neighbor's open message.

In the OpenSent state, the router waits for the neighbor's open message. Once it receives that message, it verifies that necessary parameters match (for example, the neighbor's AS matches the remote AS configured on your router). Assuming that the parameters match, the router sends a keepalive and transitions to the OpenConfirm state.

If the parameters do not match, the router sends an error notification and returns to the Idle state. Therefore, if you see a router stuck in the Idle state, you might want to check that you configured the local AS and neighbor AS correctly.

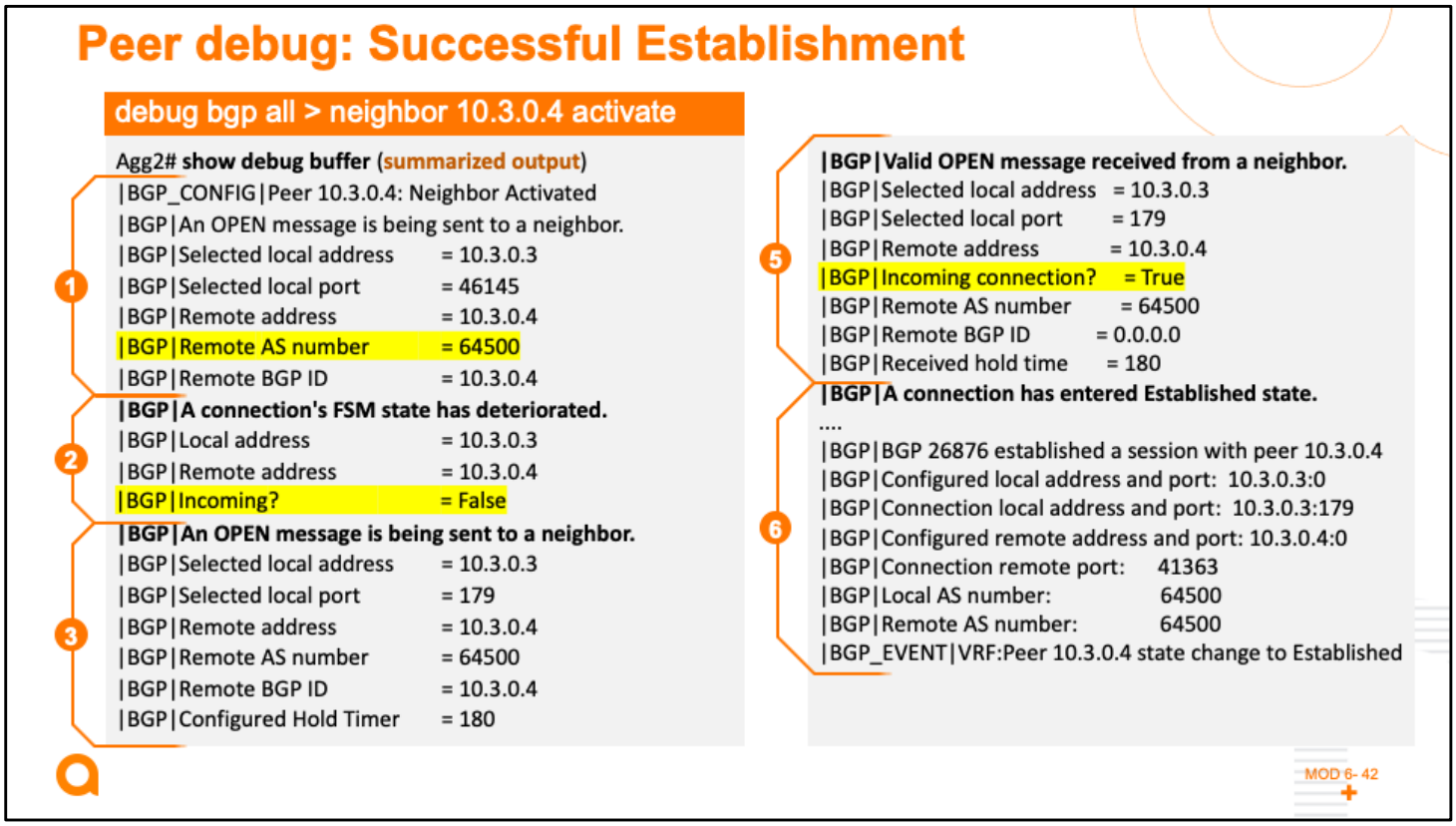
In the OpenConfirm state, the router waits for neighbor's keepalive message and also sends its own keepalive. It also listens for error notifications such as one indicating that the open message parameters do not match.

After receiving the neighbor's keepalive, the connection transitions to the Established state.

The connection remains in this state during normal operation. The neighbors continue to send keepalives. If a BGP router fails to receive a number of keepalives, it ends the connection. Recall that the keepalive period is 60 seconds by default.

New eBGP neighbors exchange their full BGP routing tables. After the eBGP neighbors have exchanged their BGP routing tables, they only exchange updates. For example, a BGP router sends an update if configuration changes affect routing policies or if a topology change has occurred.

Routes remain in the BGP routing table unless an update occurs. A router can also request a route refresh from a neighbor which might be necessary so that the router can apply new policies to routes that it has already received.



The figure shows a summarized output from the command debug bgp all, then Agg-2's neighbor 10.3.0.4 was activated. You can see highlights from the resulting successful neighbor establishment session with the command show debug buffer.

First you see that an open message was sent to the neighbor from Agg-2 (10.3.0.3) to peer SW1 (10.3.0.4). Note the remote AS number, which should match if it is an iBGP peer relationship, as is the case in this example. Of course, it would be different for eBGP peers. In either case, you should make sure that this matches the intended AS number of the peer.

Next you see that the peer's connection state has deteriorated. This is because a hello message has not yet been received from this neighbor, as indicated by the message "Incoming? = False".

You see another OPEN message sent to the neighbor, as before. This time you see the standard TCP port 179 was used as the port, with a default hold timer of 180 seconds.

Next, Agg-2 receives an OPEN message from SW2, thus the "Incoming connection? = True" message.

And so two-way communications and configuration compatibility is verified, so the connection moves to the Established state. Note the typical parameters associated with this communication. This includes the local IP address: port number (10.3.0.3:179), remote IP address: port number (10.3.0.4:41363) and the local and remote AS numbers (both 64500, as expected for iBGP peers).

Using and interpreting this debug output helps to validate your understanding of BGP theory of

operation, which prepares you for more advanced diagnostic situations. This is a useful exercise to perform in a lab environment, under controlled conditions, and sometimes in live troubleshooting situations. Just understand that in many cases, you spend time interpreting debug output for a failed connection, which often leads you to more closely analyze peer configurations using a compare-and-contrast troubleshooting method.

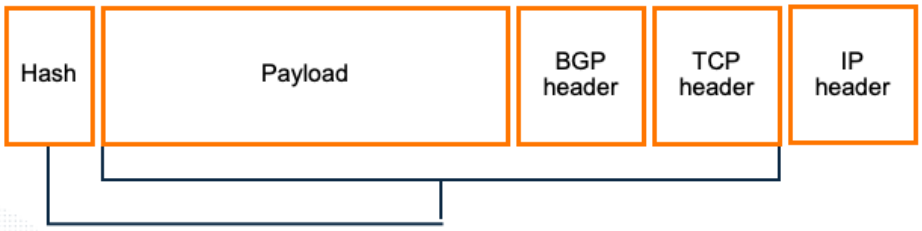
You might save time and eliminate the need for complex debug and packet capture operations by simply doing a simple configuration compare/contrast first, then validating the state of underlying interfaces and IGP routing protocols (for non-directly connected iBGP neighbors). This often resolves the issue and eliminates the need for debug/packet capture. Then, if this doesn't resolve the issue, move forward with using debug techniques like the one summarized here.

MD5 Authentication

–Hashes IP payload

–Ensures that the peer knows the correct password

–Provides data integrity for all BGP messages




```
router bgp 64500
 neighbor 10.0.3.3 password plaintext secret123
```

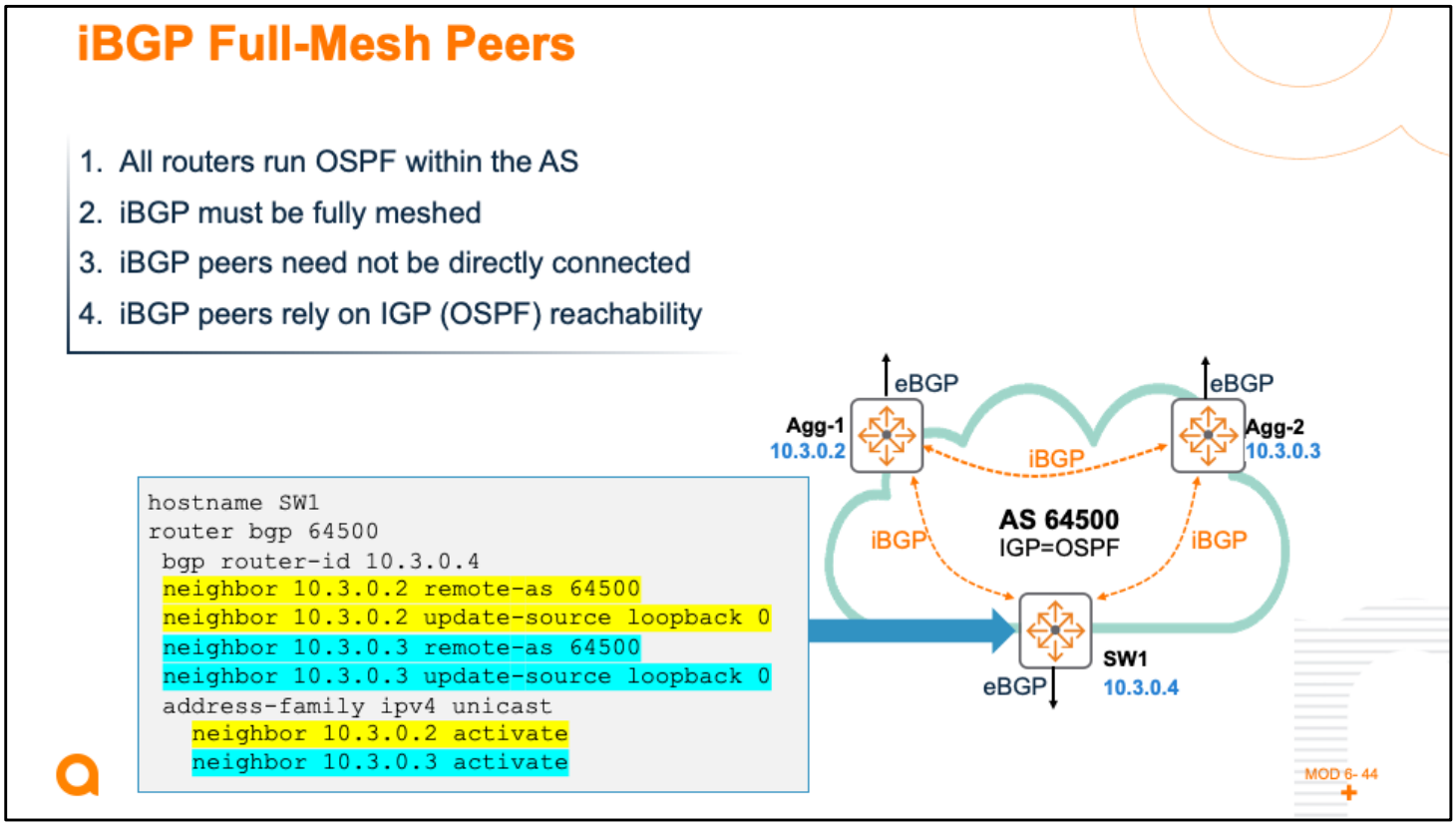
MOD 6- 43

The ISP might require your router to authenticate. You simply add a password when you define the neighbor. The BGP neighbors use this password to hash the TCP payload of packets that carry the BGP messages, appending the hash to the packet. If the neighbors' passwords do not match, the TCP session for the BGP connection fails to establish.

Because the neighbors add the MD5 hash to each packet sent on the TCP session, MD5 authentication provides data integrity for all BGP messages and for underlying TCP session. In other words, each neighbor knows that the proper neighbor sent the message and that no one has altered the message. However, MD5 does not provide protection against eavesdropping.

To assign the password, use the following command:

switch(config-bgp)# neighbor <ipv4-addr> password plaintext <password>



The figure shows what could be a large OSPF-based network, with say 500 routers. Four main points about BGP are highlighted here.

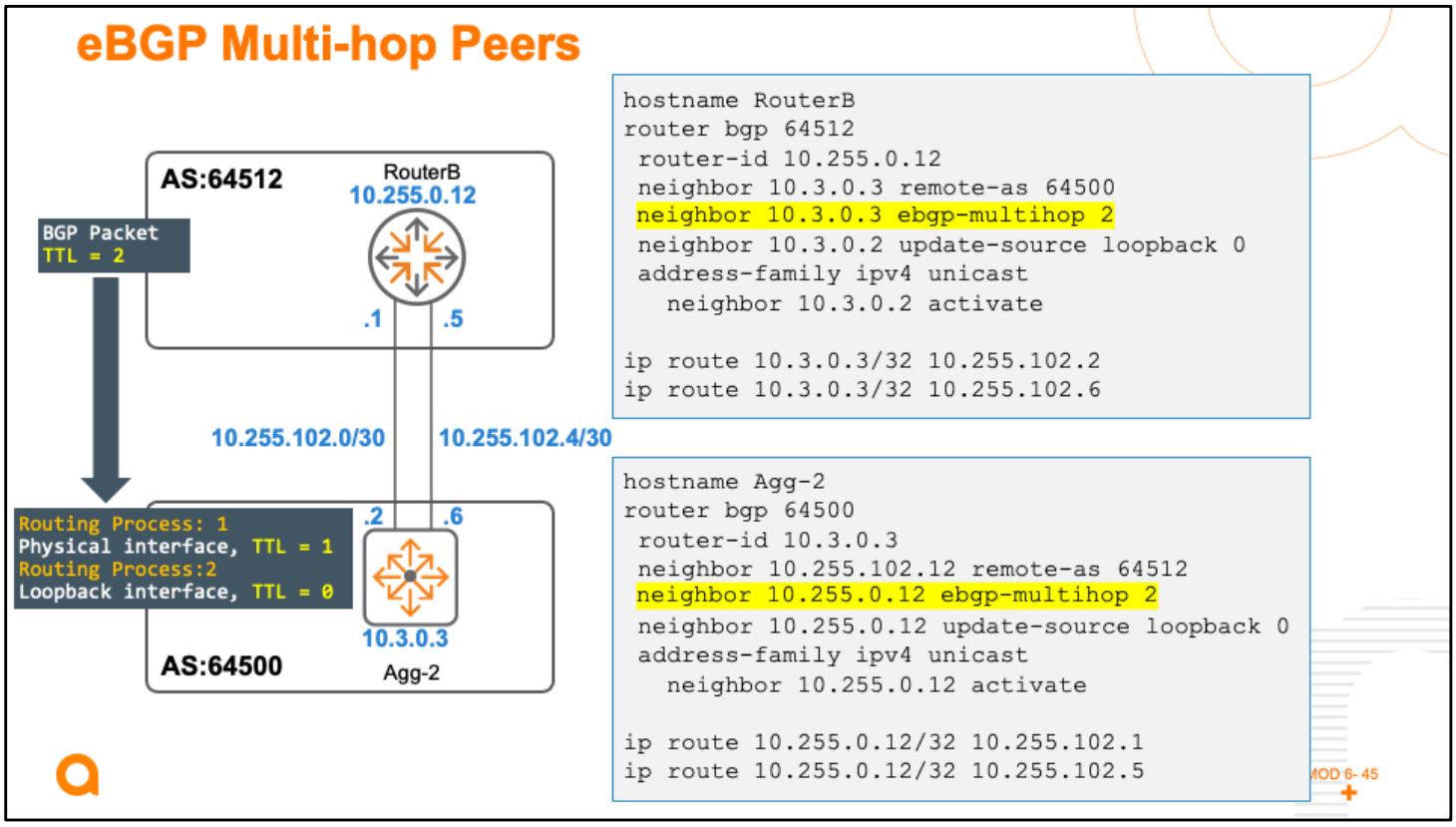
All routers run OSPF, finding the best path from any source to any destination. This includes the best path from the loopback address of Agg-1 to the loopback address of Agg-2 and SW1, and vice versa.

Only the ASBRs run iBGP between each other. All iBGP peers must be fully meshed. Every iBGP router must be configured to peer with every other iBGP router. Many organizations are like the one shown. With only 3 or so ASBRs, you must only configure each router with 2 peers. Some very large organizations may have dozens of iBGP peers or more. This full-mesh requirement could become a burden. The configuration gets long and tedious, and the peering overhead might become an issue. Of course, you will soon learn about an exception to the full-mesh requirement, and how to configure route reflectors.

Unlike eBGP neighbors, iBGP neighbors need not be directly connected. As shown indicated by the cloud in the figure, there could be several routers between Agg-1, Agg-2, and SW1. These routers might be in different campuses, or even different cities. This helps ensure that each region of your organization has an internet connection, and each region can have their own optimized pathing to and from the internet.

iBGP peers can be so separated because they rely on the IGP, OSPF in this case, to route their peer connection. You tell agg-1 to peer with Agg-2's loopback address. Remember, this is a TCP connection, and TCP relies on the underlying routing protocol for pathing. So, OSPF routes BGP hello, keepalive, and update packets between peers. Remember this for troubleshooting! iBGP peers may fail to form because of a faulty OSPF router deployment. For example, if you forget to enable OSPF on the loopback interfaces, then OSPF will not advertise them. They will not be reachable, and no BGP peers will form.

The figure also shows a generic configuration that applies to all three iBGP peers, where x = local loopback address, and y and z represent each of the two iBGP peers.



You just learned that you must configure eBGP peers using their directly-connected physical IP addresses. The figure shows an exception to this rule. If you have multiple physical links between eBGP peers, they will benefit by peering via their loopback addresses – to maximize resiliency.

As shown in the figure, you configure eBGP peers to use their loopback addresses, just like you did with iBGP. The difference is that you must configure the eBGP multihop command highlighted in the figure. The number 2 at the end of the command is a hop count – it sets the TTL field to 2, so the peer can only be 2 router hops away.

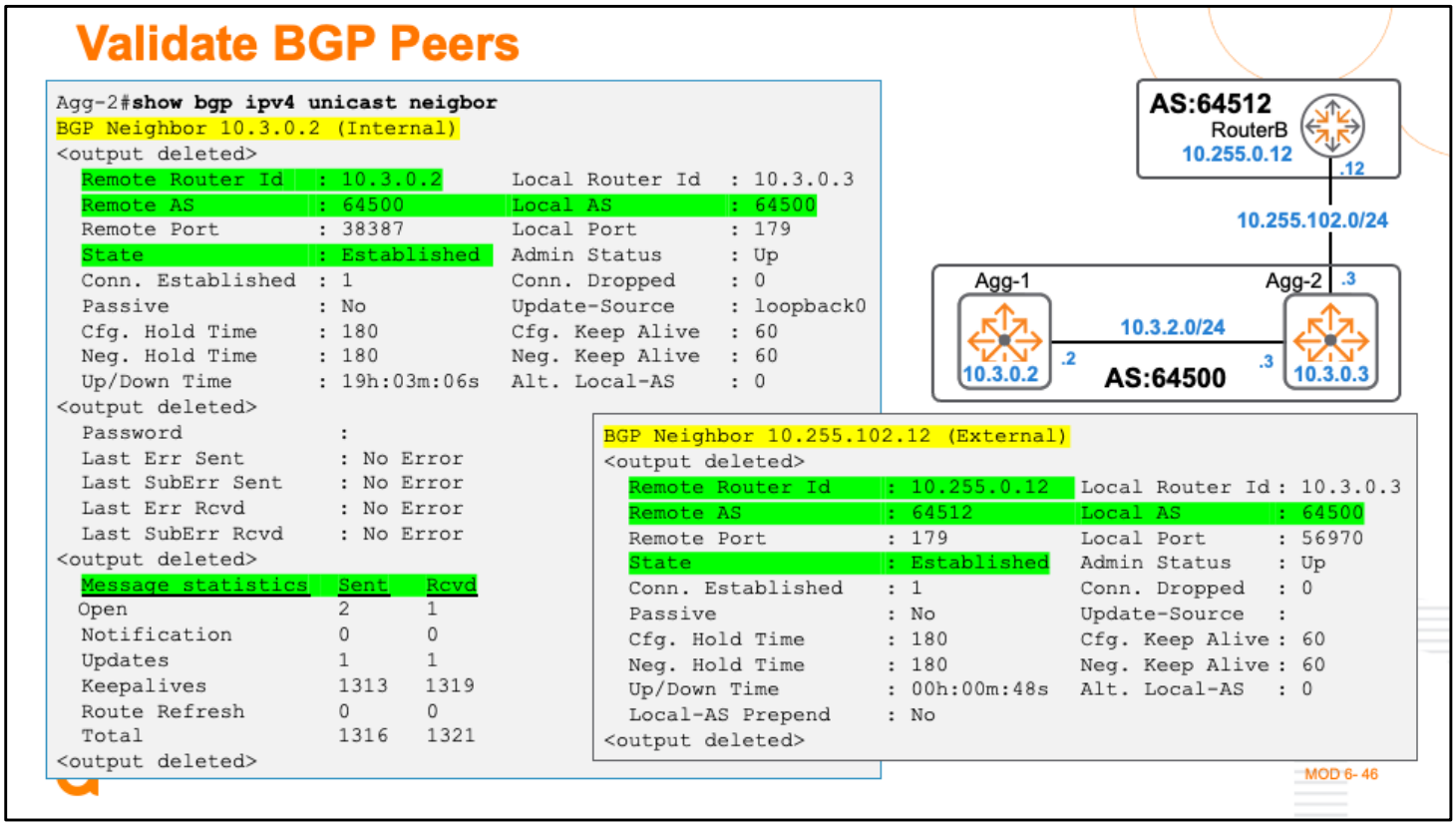
As configured, both links have the same administrative distance. The router will load-share across these links. If desired, you could adjust administrative distance or some other BGP metric, so that one link was the primary link. The other link is only used if the primary link fails.

The static routes are configured so BGP can actually reach the loopback addresses.

Another use case for eBGP multihop is when you must reach the ISP router that runs BGP through another router or routers. To set up this solution, you must set the eBGP multihop to the correct number (2 if one intermediate router intervenes, three if two routers intervene, and so on). Your router also requires a route to the ISP router's IP address, of course. However, it is not recommended to have intervening routers between eBGP neighbors, and you should

avoid this setup whenever possible.

Note: Understand that the multi-hop feature described here is fine to deploy, and quite common. You are only being asked to avoid having routers between eBGP peers.



Now that you can describe and configure BGP peering, you should be able to validate it. The figure shows output for Agg-2's show bgp ipv4 unicast neighbor command, edited to focus our attention. The output is split into two sections – one for each BGP neighbor or peer.

Neighbor 10.3.0.2 is an Internal (iBGP) peer. You see the local and remote router IDs and ASes, and you see that indeed, BGP uses TCP port 179.

The state/status is Established/Up. You can validate that the update source is loopback 0, as well as the Hold and Keepalive timers of 180 and 60 seconds, respectively.

Pay attention to the Up/Down Time – consistently short up times mean that the peers are consistently losing connectivity – you may have a network stability issue.

You can see that there is no password configured, and no error messages received – nice!

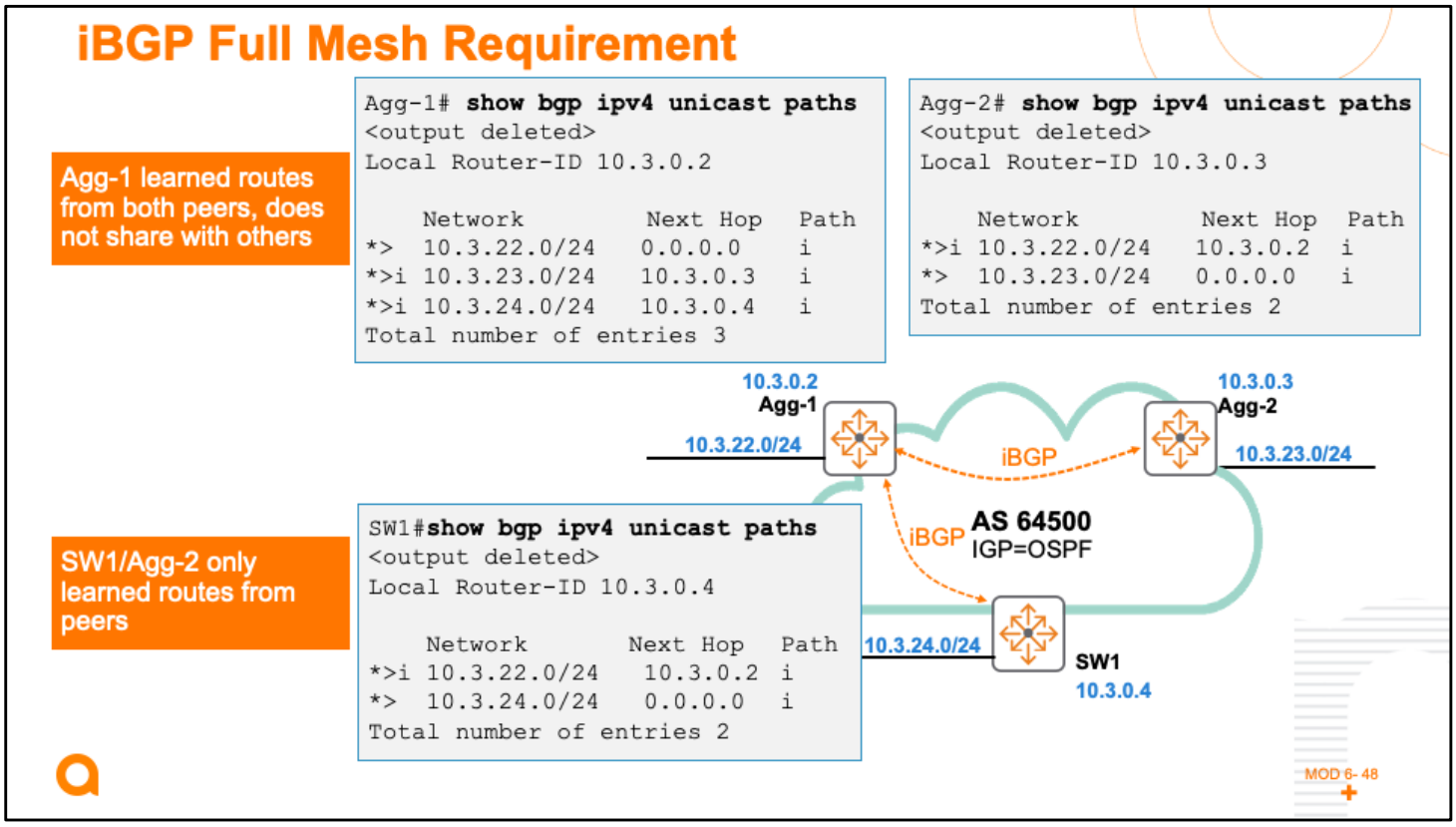
Finally, you see the messages you recently learned about – Open, Notification, updates, keepalives, route refresh.

The next section is for neighbor 10.255.102.12, Agg-2's eBGP peer, which holds the same information as the first section.

BGP Advertisements

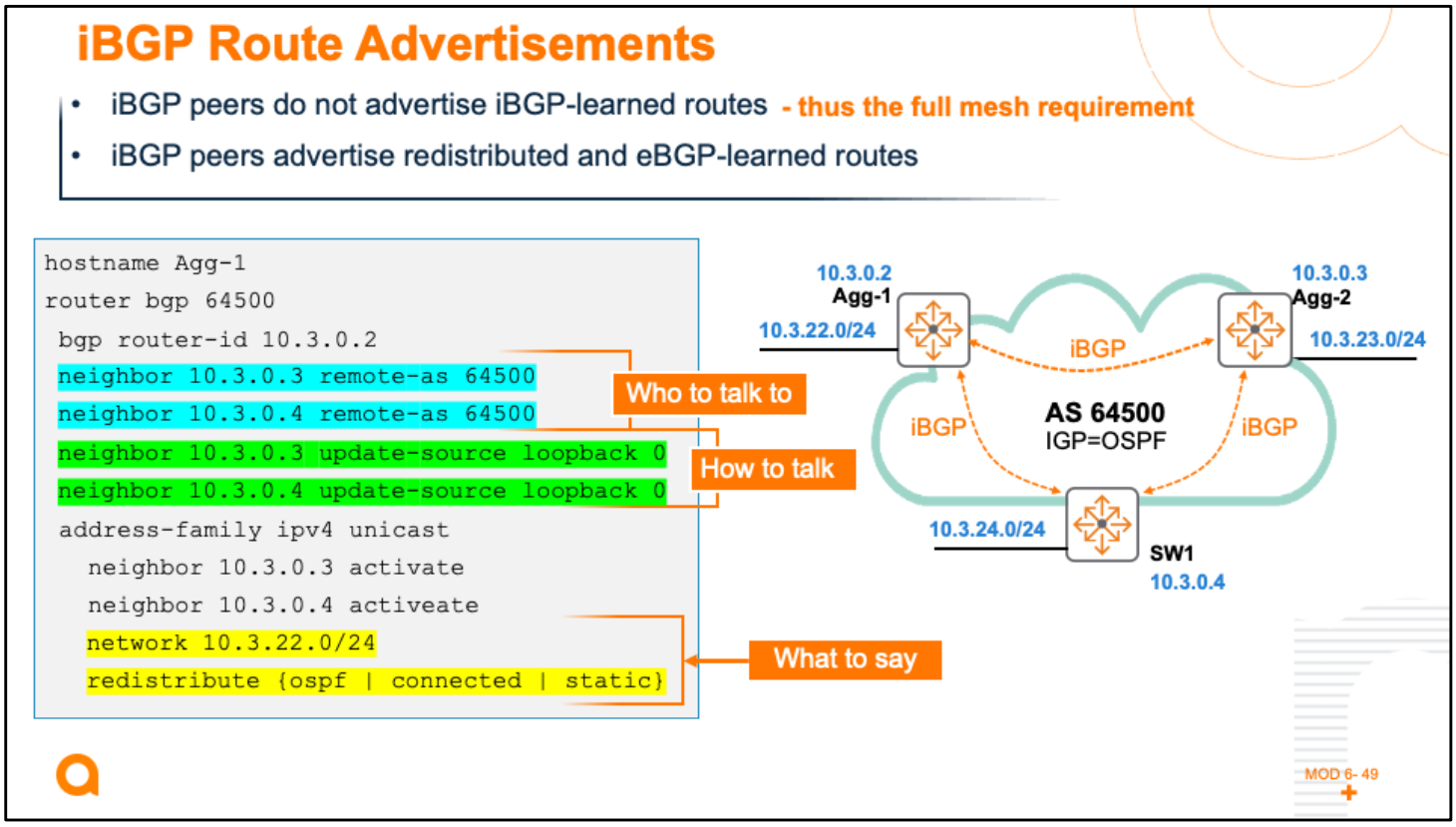
iBGP and eBGP advertisements and table review

aruba
a Hewlett Packard Enterprise company



The figure proves point number 1 about iBGP peers – they do not advertise routes learned from other iBGP peers. You know how to fix this – configure a peer relationship between Agg-2 and SW1. Let's assume that you follow up and do that, using the syntax you now know.

Now you will explore the other key point about iBGP peer advertisements.

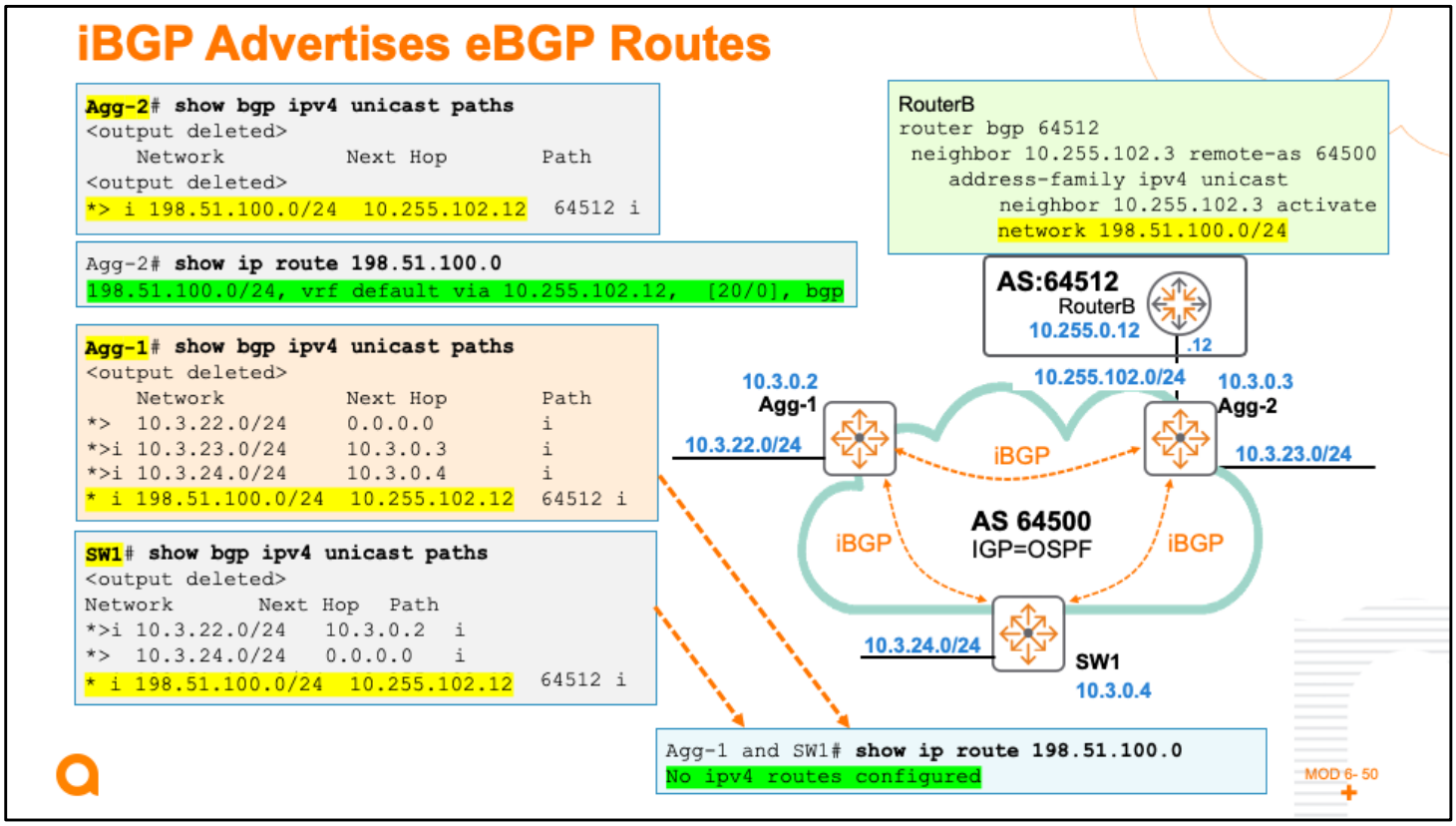


You just learned how to configure BGP peers – you told the routers who to talk to. Now you must ensure they know what to say. What routes will BGP neighbors advertise? How do you get them to advertise routes as desired? This depends on whether they are iBGP or eBGP peers. Let's start with iBGP.

The figure makes three key points about iBGP. The first is that iBGP peers do not advertise routes learned from other iBGP peers – thus the iBGP full mesh requirement. This may seem odd if you are used to OSPF, or any other IGP. OSPF is based on the fundamental idea that OSPF Router A shares what it learns from Router B with all other routers.

The figure shows the BGP configuration for Agg-1, with new syntax – network 10.3.22.0/24. You are telling Agg-1 that it should advertise this route. This makes sense, since this network is directly connected to Agg-1. Similarly, Agg-2 has network 10.3.23.0/24, and SW1 has network 10.3.24.0/24. Agg-1 and Agg-2 are peers. Agg-1 and SW1 are peers. However, suppose that you have neglected to configure Agg-2 and SW1 as peers. What is the result?

Let's explore.



The figure proves the second key point about BGP – iBGP peers advertise redistributed and eBGP-learned routes. Router B in AS 64512 is now configured to originate or inject network 198.51.100.0/24. Router B peers with Agg-2, which learns about this route and advertises this eBGP-learned route to its iBGP peers - Agg-1 and SW1.

The route of course also appears in Agg-2's route table. This is because Agg-2 is directly connected to the next-hop IP address – 10.255.102.12.

However, the route does not appear in Agg-1 nor SW1's route table. Why?

Notice that the next-hop address for Agg-2 is 10.255.102.12 – makes sense. However, the next-hop address in the BGP table for the other routers is also 10.255.102.12. This is because by default, BGP routers do not change the next-hop address when they advertise the route. This is great when you have multiple routers connected to the same broadcast network. If Agg-1, Agg-2, and SW1 were all on the same broadcast link, Agg-2 would learn the route, and advertise it to the other two peers with the same next hop. Thus, they could go directly to Router B, instead of via Agg-2.

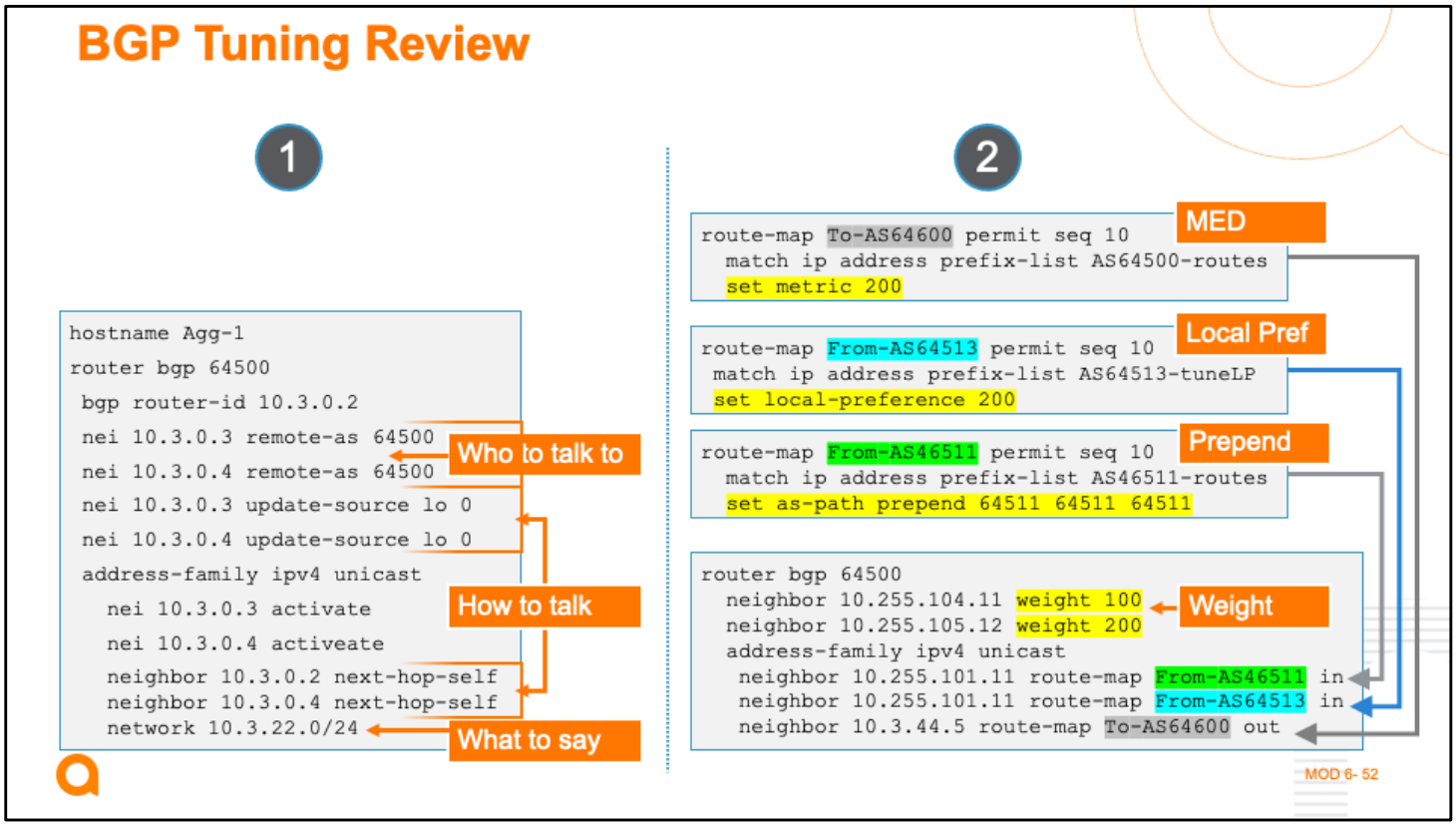
But this is not our situation. Agg-1 and SW1 have the route in their BGP table, but they cannot reach the next hop address, which is an external route that is not redistributed into the AS. This shows the third key point about BGP route advertisements – BGP-learned routes are only added to the route table if the next hop is reachable.

How do you solve this issue?

aruba
a Hewlett Packard Enterprise company

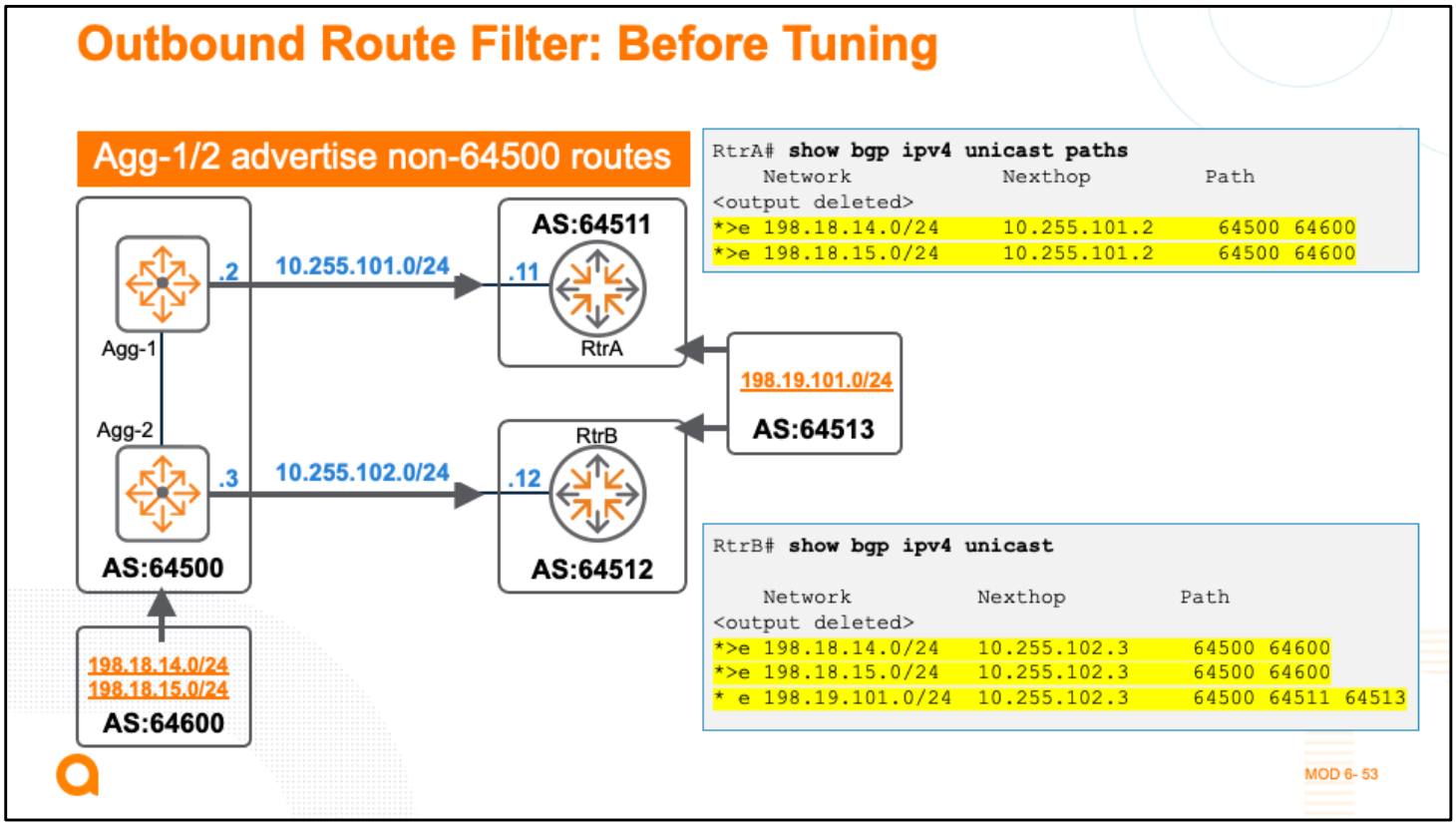
Route Control

Techniques, peer groups, route reflectors

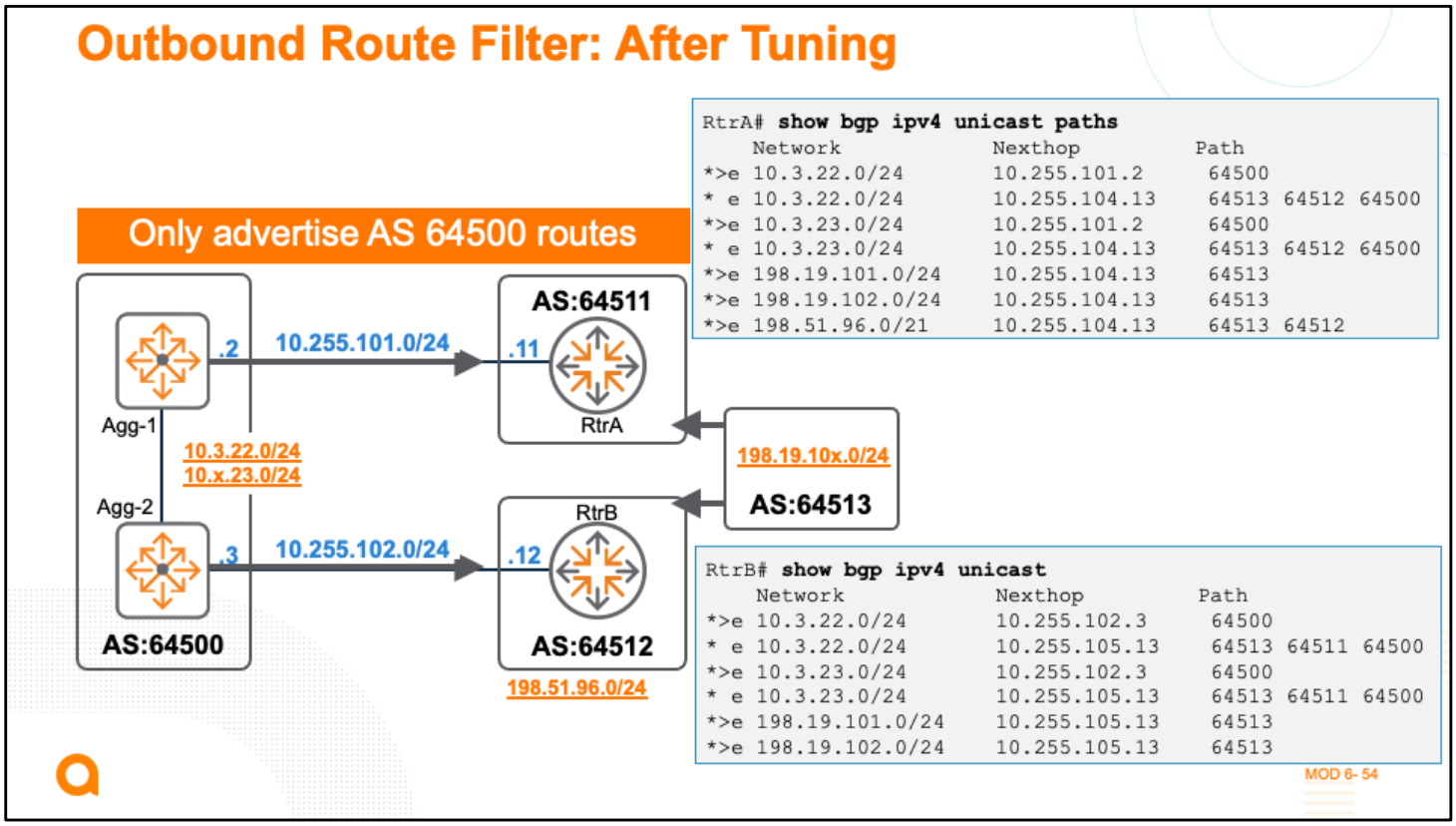


Let's review.

First you learned how to tell the routers who to talk to, how to talk, and what to say. Once they were talking and sharing, you learned how to control best path selection by controlling metrics. This included MED, Local Preference, prepending, and weight.



The figure shows the current system state – before applying the outbound filter. The highlighted entries prove that Agg-1 and Agg-2 are currently advertising routes from other ASes.



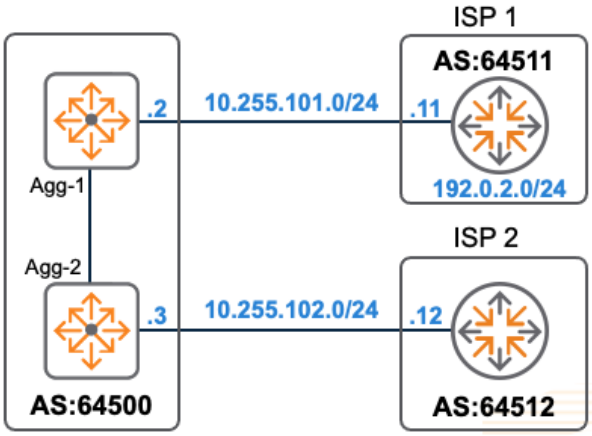
After applying the filter, Agg-1 and Agg-2 are only advertising routes for the local AS 64500.

Control Inbound Routes

Objective

- Use ISP 1 to reach 192.0.2.0/24
- Use existing default routes for other external routes

```
hostname Agg-1
ip prefix-list ISP1resources seq 10 permit 192.0.2.0/24
route-map ISP1 permit seq 10
  match ip address prefix-list ISP1resources
router bgp 64500
  address-family ipv4 unicast
    neighbor 10.255.101.11 route-map ISP1 in
```



MOD 6- 55

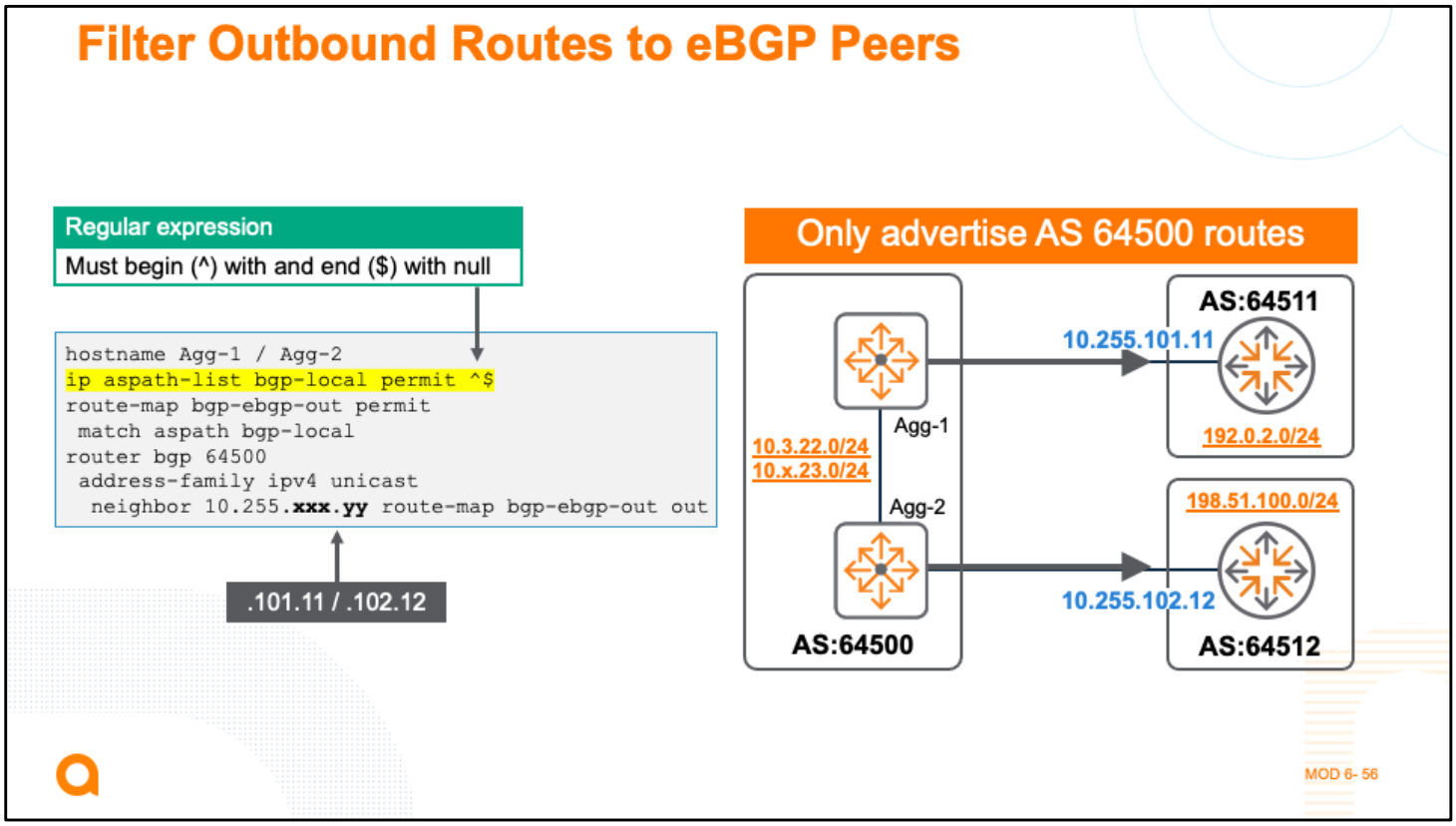
Here is an example route map that filters routes the BGP routers receives from its eBGP neighbor. In this example, you run BGP primarily to announce your public block of IP addresses, and care less about receiving all the BGP routes in the Internet routing tables. You can filter those down in a few different ways. You may choose to just accept routes for larger ranges less than a specific prefix length. Or you might choose to accept just routes that originate in the eBGP neighbor AS because those are the ones where the company will benefit the most by sending the traffic out this specific router. You could also take a combination of these approaches.

In any case, you must ensure that your BGP edge routers have a default route. You could achieve this with the typical “ip route 0.0.0.0/0” entry, or by using OSPF default-information originate, or by ensuring that your ISP advertises a default route, which you would then also need to permit with your route map.

In this example, you want to use ISP1 to access resources on the 192.0.2.0/24 network. For other networks, a default route is typically sufficient. The example configuration uses an IP prefix list to select a specific route to learn from ISP1: 192.0.2.0/24. Please note that you might need to perform similar inbound route filtering to make intelligent choices to reaching resources in ISP2 as well.

Important: Make sure your AS is receiving a default route if you filter inbound routes.

What would happen in your link to ISP1 would go down? You would still be able to access 207.1.1.0/24 via a default route to leave your network via ISP2.



In this example, you are setting up route maps to restrict outbound advertisements to eBGP neighbors. There are a couple of ways that you could solve this issue:

Use an AS Path list where the path list is empty, indicating that the route is internal to your network

Use a prefix list to only allow advertisement of your routes to other ISPs

The ip aspath-list command provides an elegant solution for this use case, because it specifies a specific AS path. AS Path lists are named lists of regular expression rules. They are used to match AS Path attributes in the routes for inclusion in or exclusion from route policies. The sequence number is optional and is autogenerated whenever it is not explicitly mentioned. All AS path list rules with the same name are grouped together. The configuration of IP AS path lists is beyond the scope of this course.

Note: See https://www.regex101.com for regular expression examples and to test your regular expressions.

These regex values are used:

'^' indicates 'must begin with'

'$' indicates 'must end with'

Combined, they result in '^$', which represents an empty string.

All the routes that are originated inside this AS, will have an empty AS-Path attribute, while any route that was received via an external AS (eBGP), will have at least the value of that external AS in the AS-Path. To check the AS-Path of a route, a regular expression can be used. Also notice that the route map is applied to both Agg-1 and Agg-2, when communicating outbound to their respective eBGP peers. You would need to deploy a similar configuration to any other eBGP routers in your network.

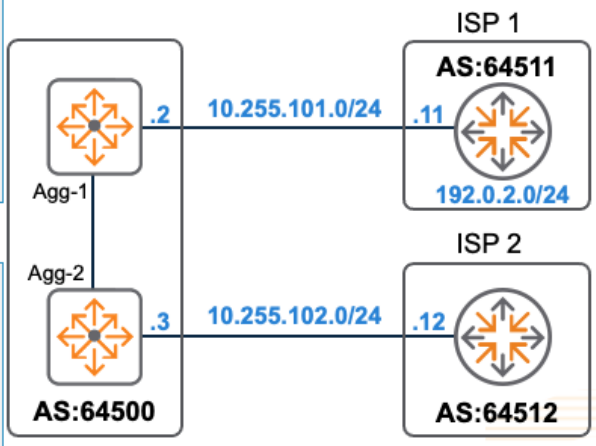
Inbound Route Filter – Before Tuning

Pre-tuning: Several routes from ISP 1

```
Agg1# show bgp ipv4 u paths
    Network          Next Hop         Path
*>  10.3.22.0/24     0.0.0.0          i
*>i 10.3.23.0/24     10.3.0.3         i
*>e 192.0.2.0/24     10.255.101.11    64511 i
* i 198.19.101.0/24  10.3.0.3         64512 64513 i
*>e 198.19.101.0/24  10.255.101.11    64511 64513 i
* i 198.19.102.0/24  10.3.0.3         64512 64513 i
*>e 198.19.102.0/24  10.255.101.11    64511 64513 i
*>i 198.51.96.0/21   10.3.0.3         64512 i
* e 198.51.96.0/21   10.255.101.11    64511 64513 64512 i
```

Post-tuning: only the desired route from ISP 1

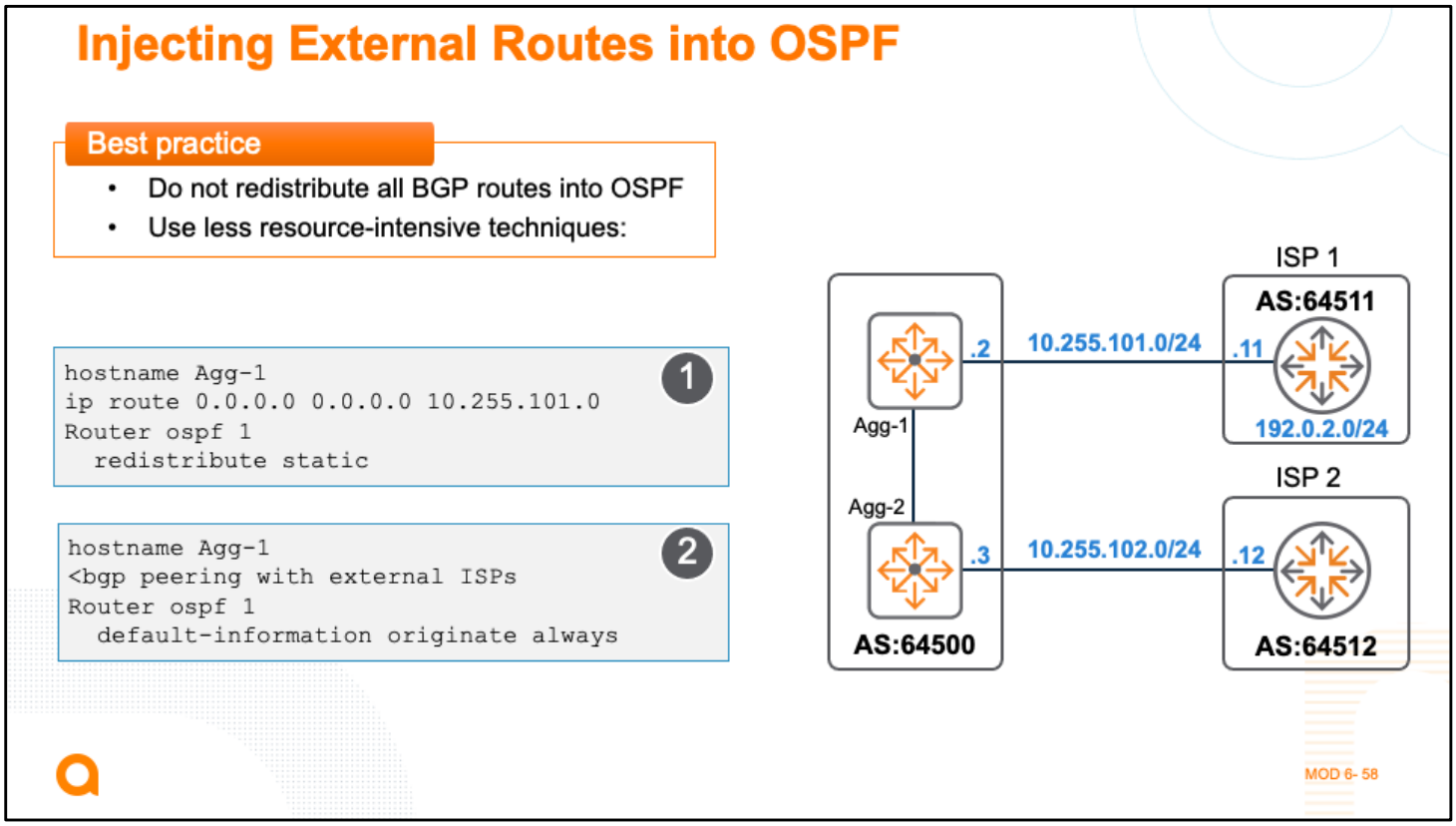
```
Agg1# show bgp ipv4 u paths
    Network          Next Hop         Path
*>  10.3.22.0/24     0.0.0.0          i
*>i 10.3.23.0/24     10.3.0.3         i
*>e 192.0.2.0/24     10.255.101.11    64511 i
*>i 198.19.101.0/24  10.3.0.3         64512 64513 i
*>i 198.19.102.0/24  10.3.0.3         64512 64513 i
*>i 198.51.96.0/21   10.3.0.3         64512 i
*>i 198.51.200.0/24  10.3.0.3         64512 i
```



MOD 6- 57

The figure's top example shows Agg-1's BGP table before tuning. You see that there are several routes learned via ISP 1 – 10.255.101.11.

After applying the route filter, only the desired route exists in Agg-1's BGP table – success!



You should consider the amount of routing information that your BGP routers receive from ISP routers, and how much of that should be available to the company IGP, typically OSPF. Although you can redistribute BGP routes into OSPF, it is not typically recommended. The large BGP routing table might overwhelm the OSPF processes on the local router and on other routers throughout the AS.

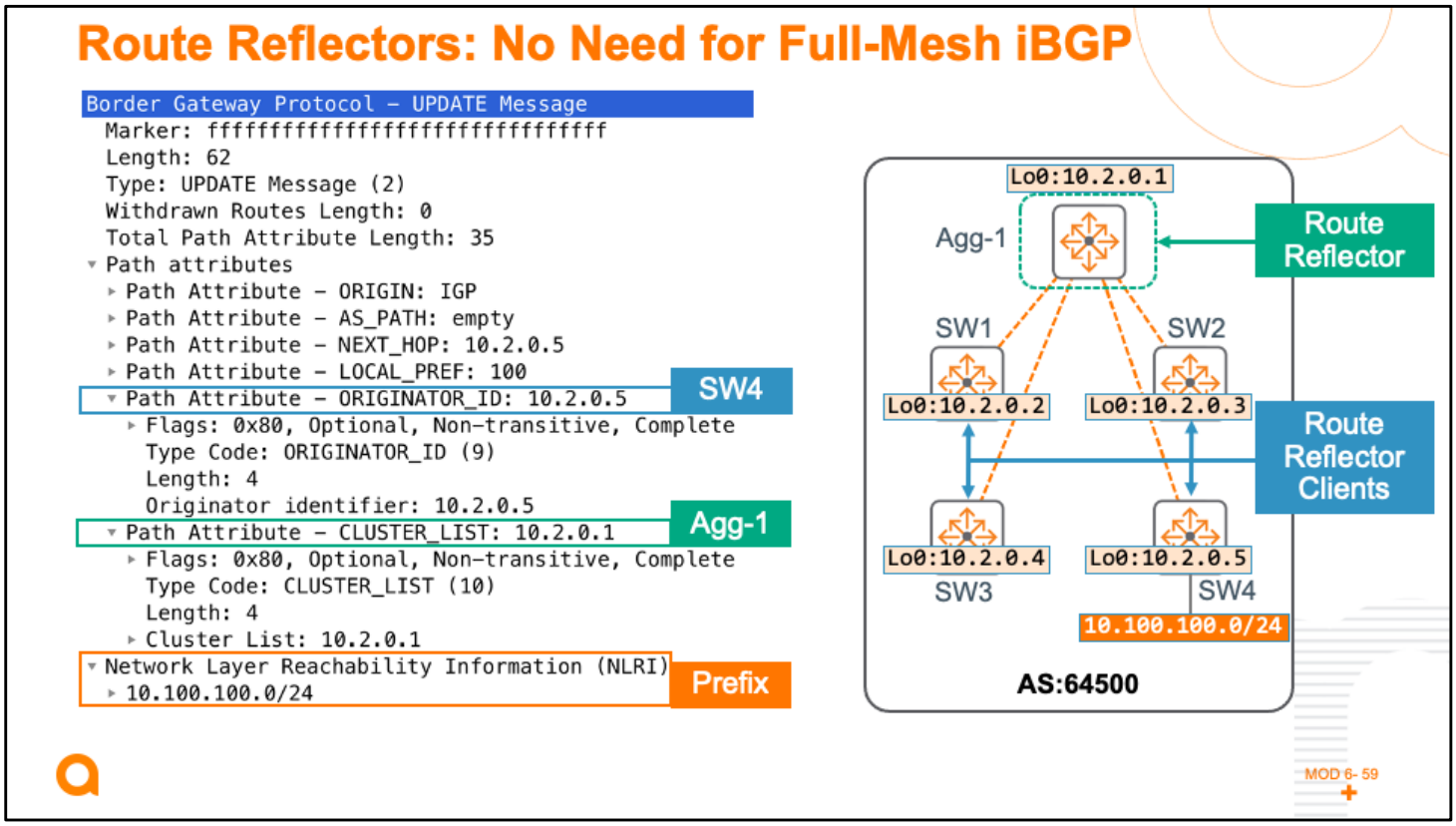
Often, you can simply have your BGP routers advertise default routes in OSPF. If they are already acting as ABRs for stub areas, they already advertise default routes in those areas. You might need to have the BGP routers advertise default routes in the OSPF backbone, though. A simple approach is to configure a static route through its ISP connection on the BGP router. Then redistribute static routes into OSPF. This is shown in the figure as option 1.

If you are going to redistribute BGP into OSPF, it is highly recommended that you use a route-map to qualify exactly which route(s) you want to redistribute into OSPF. Use the techniques you learned earlier in the redistribution module.

The figure also shows a second option, using the OSPF default-information originate command, with the always option. Without this option, OSPF only advertises default information (gateway of last resort – 0.0.0.0/0) if it actually has a valid path to 0.0.0.0. So, if the interface to 10.255.101.0 goes does down, the static route goes down, so it will not be in OSPF.

If OSPF should have default route, independent of the default route existence in the routing table, use the always.

Note: Your ISP might advertise a default route to your BGP router. In that case, instead of redistributing static routes, you could redistribute just the BGP default route. When you redistribute BGP into OSPF, you apply a route map that permits only that route.

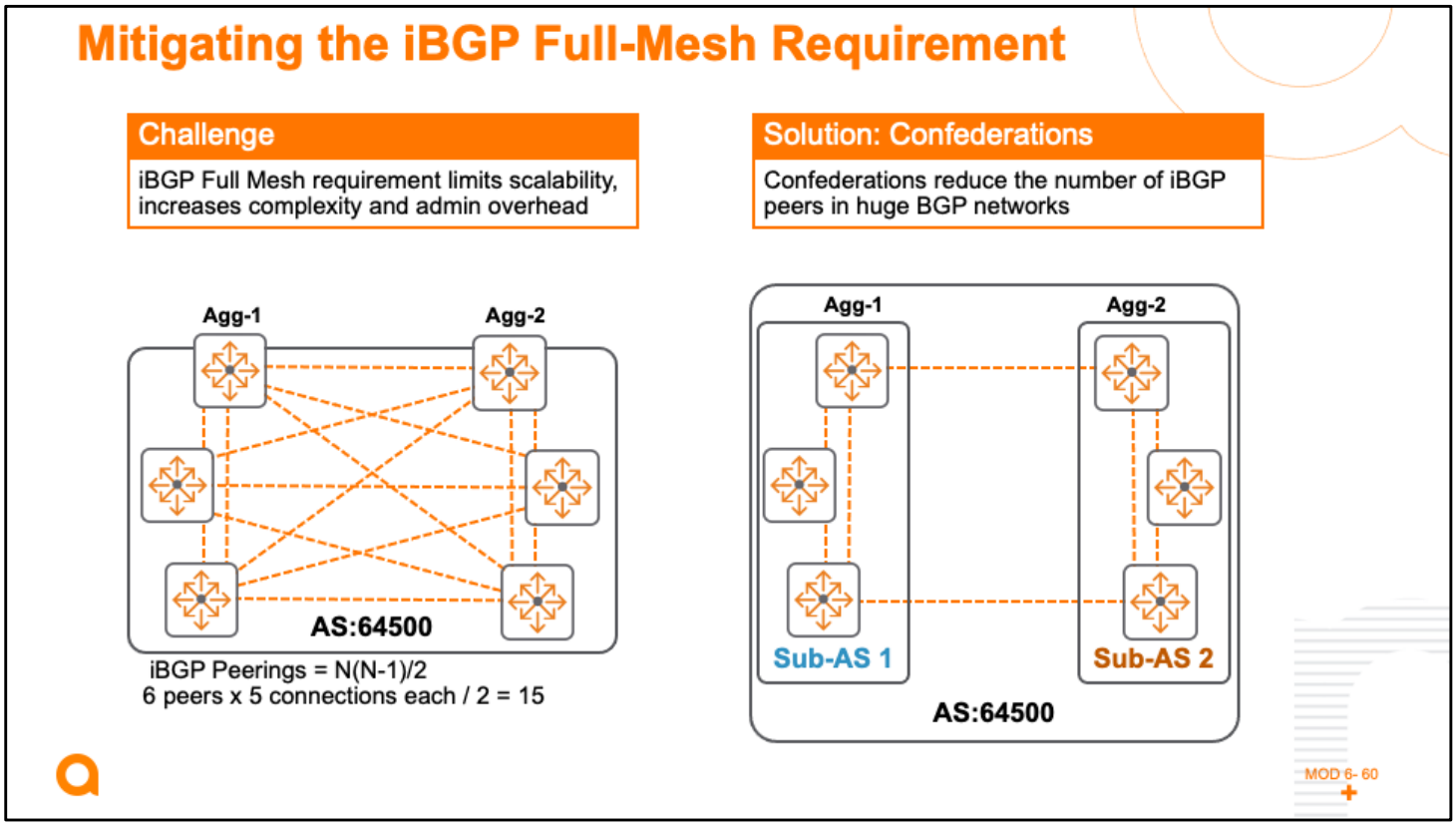


Removing the full mesh requirement in an iBGP topology introduces the potential for routing loops. RFC 1966 introduces two other BGP route reflector-specific attributes to prevent loops.

ORIGINATOR_ID is an optional non-transitive BGP attribute created by the first route reflector. This attribute sets the value to the router-id of the router that injected (advertised) the route into the Autonomous System (AS). If a router receives an NLRI with its router-id in the Originator attribute the NLRI is discarded.

CLUSTER_LIST is an optional non-transitive BGP attribute updated by the route reflector. This attribute is appended by the route reflector with its cluster-id. By default, this ID is the BGP router-id. The cluster-id can be set with BGP configuration command bgp cluster-id <cluster-id>. If a router receives an NLRI with its cluster-id in the cluster list attribute, the NLRI is discarded.

In the example, SW4 a route-reflector client advertises the 10.100.100.0/24 prefix into the iBGP domain. The Wireshark capture shows that the ORIGINATOR_ID is the router-id used by SW4 (10.2.0.5) and the CLUSTER_LIST is the router-id used by the route-reflector in this example Agg-1 (10.2.0.1).

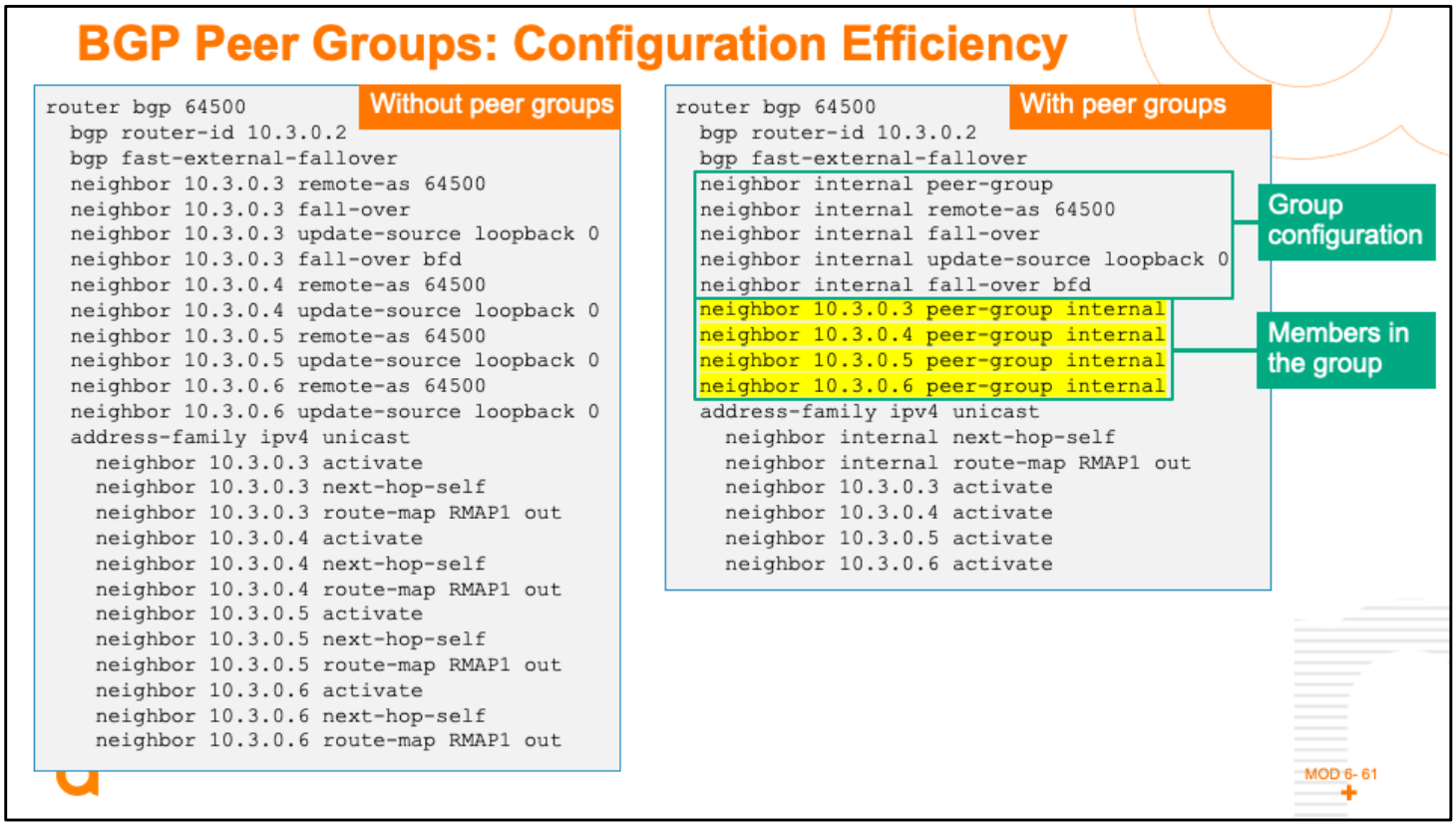


Another solution to reduce the number of required iBGP peers and thus improve scalability is to configure confederations. So, confederations serves the same purpose as Route Reflectors, but in a different way. With confederations, you split your AS up into “sub-ASes”.

Each sub-AS has a full-mesh requirement, but you can have many sub-ASes, so each sub-AS has few routers. You can configure route reflectors within each sub-AS to further reduce the number of peers.

Confederations was actually the first solution devised to reduce iBGP peer requirements. It works for very large BGP networks, larger than many organizations will ever have. It is also a bit more complex to deploy than route reflectors. This was one of the motivations for creating the relatively newer route reflector solution. Route Reflector deployments are very simply, and scale enough for all but the very largest of BGP deployments.

Unless you work for a very large ISP, you will probably be better off using a Route Reflector solution. For this reason, we will not discuss the configuration of BGP confederations in this course.



You have learned to configure BGP with many options, including setting the update source, using next-hop-self, applying inbound and outbound route maps, perhaps with prefix-lists and AS-Path lists, and more.

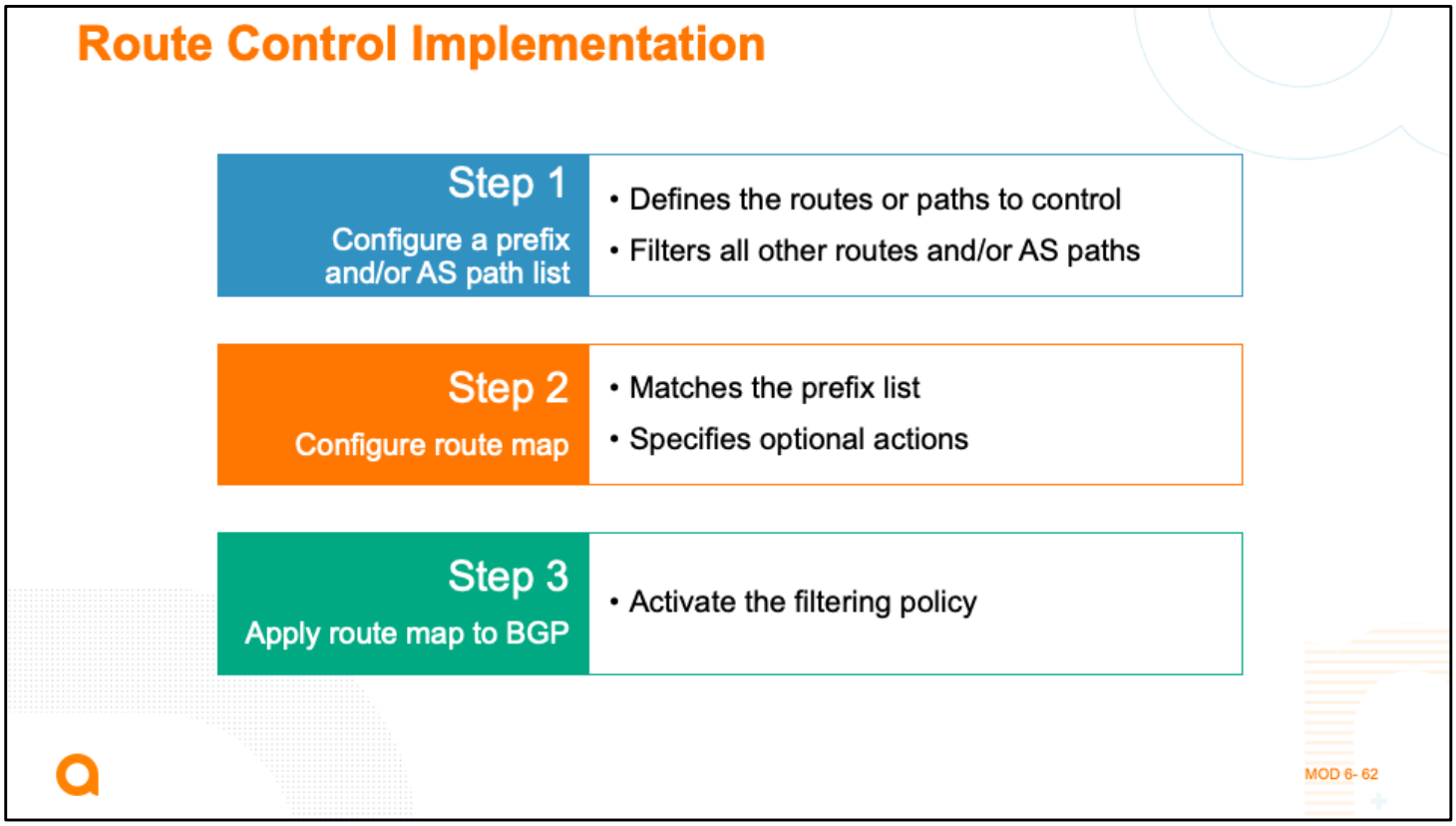
For those running large networks with many dozens of BGP peers or more, the use of peer groups can streamline your configuration efforts. The idea is to group BGP peers with identical or at least very similar needs and configure them as a group. This alleviates from individually configuring all peers with repetitive syntax. You might create a group for eBGP peers and another for iBGP peers. Or you might have a few groups for iBGP peers, each group with identical or near-identical needs.

The example on the left shows a router with only four iBGP peers. Each peer must be configured with a remote-AS, update-source, next-hop-self, and a route map.

The example on the right accomplished the same thing with less typing. You create a peer group named “internal” and identify group members with the command neighbor 10.3.0.x peer-group internal. Now you simply apply all of your configuration parameters one time – to the peer group. Of course, this configuration applies to all group members.

The advantage may seem slight in this example, but the more peers you have, the more repetitive typing you save with peer groups.

Note: A BGP peer can only belong to one peer group.



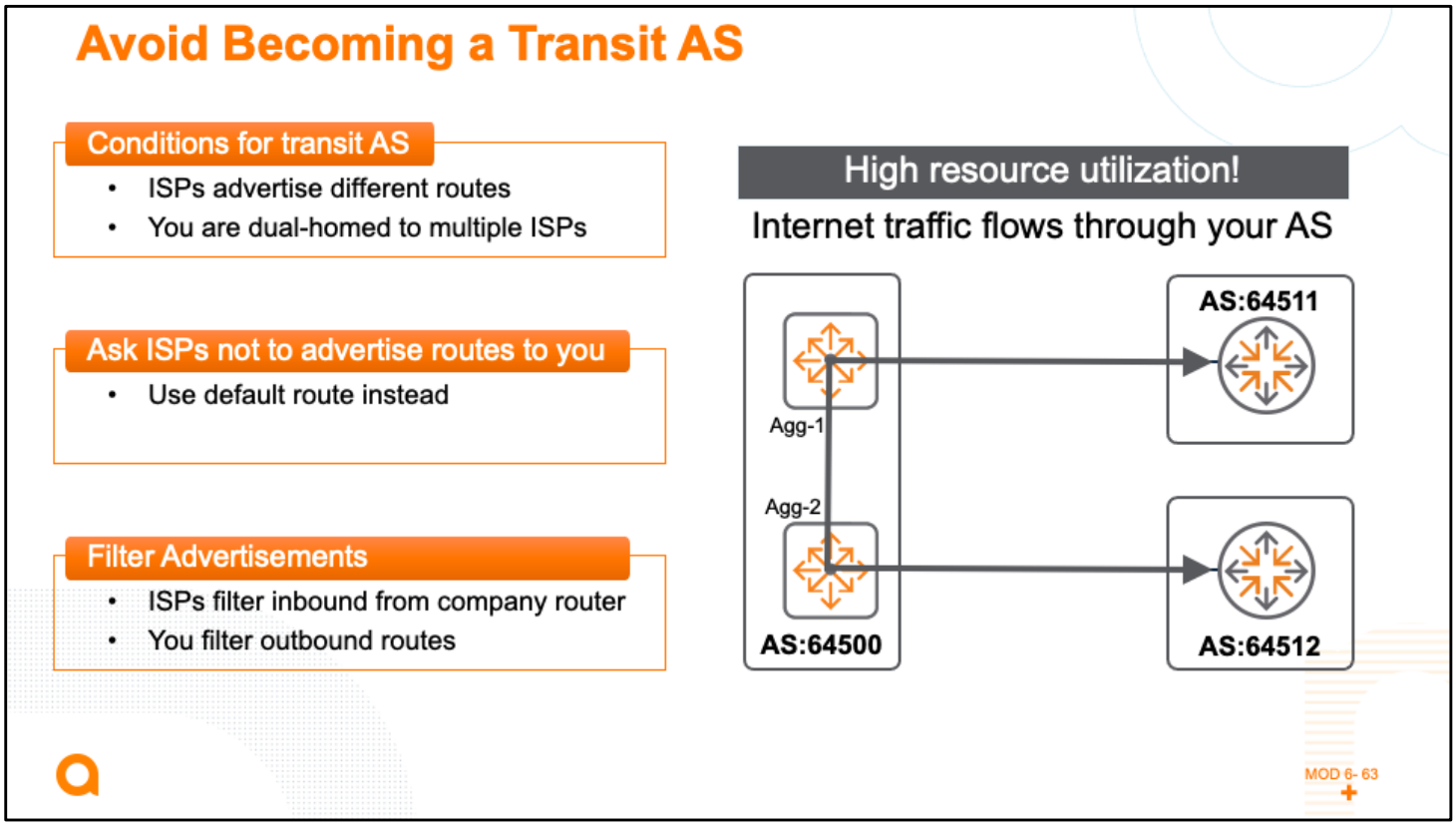
You can manipulate route preference using route maps, which you have already used extensively in this training. So far, you have used route maps to tune routes that already exist in the BGP table. Earlier you used them to control OSPF route redistribution. Now you will use them to control prefixes advertised out to other ASes or accepted inbound from other ASes.

Recall that deploying a route policy is three-step process:

Configure filters based on route attributes, such as destination address and the advertising router address.

Create a route map and apply filters to it.

Apply the route map to BGP



In the multi-homed ISP scenario, you should only advertise your own public IP addresses to the ISPs. Do not announce routes that your AS learns from ISP 1 to ISP 2, nor vice versa. This can cause ISP routers to use your AS for internet traffic. Your router will quickly become overburdened by external Internet traffic.

Several scenarios could cause your AS to become a transit AS:

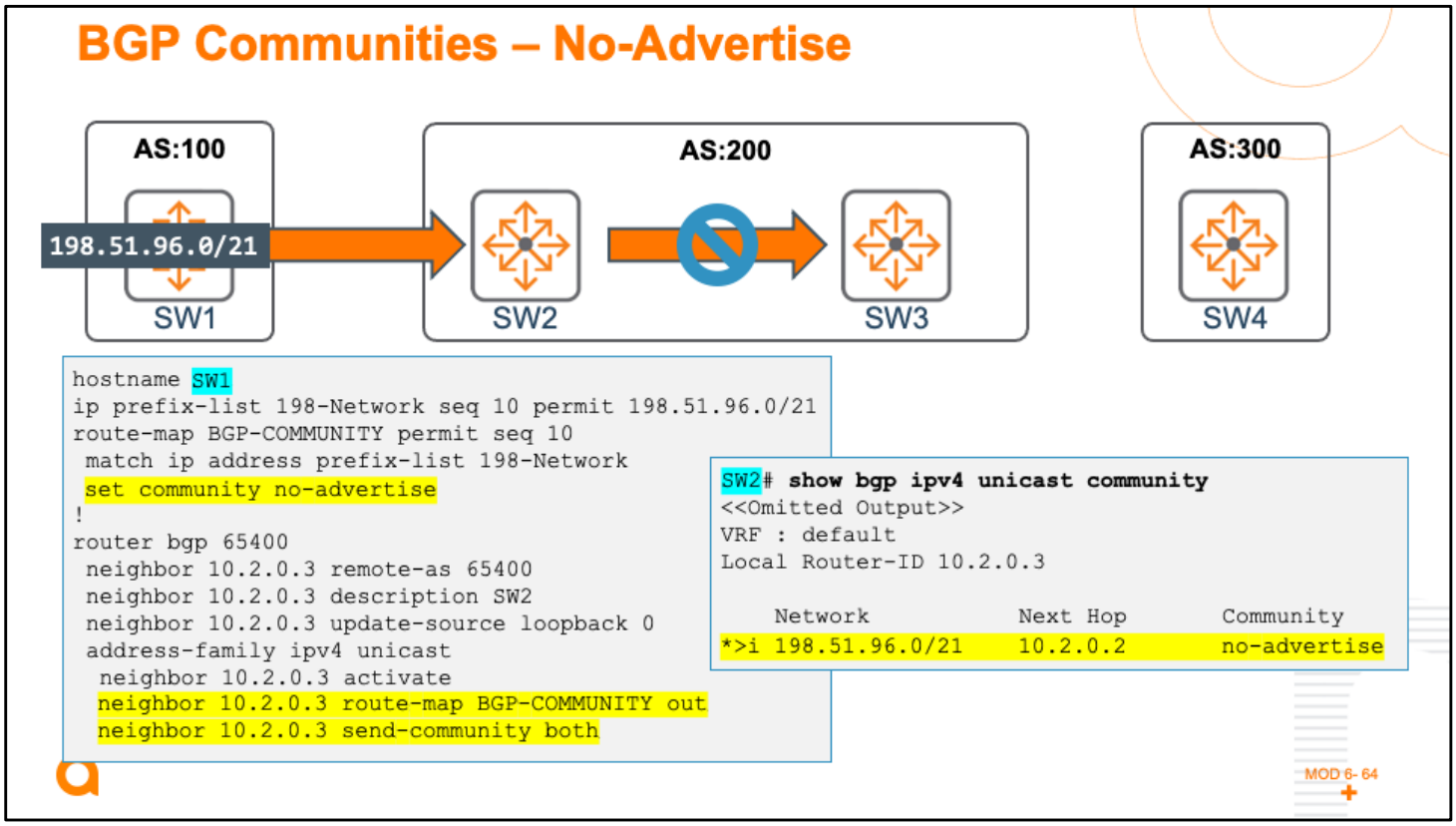
The ISPs advertise Internet routes differently with different aggregation. Any route that your BGP routers receive from only one ISP, they could begin advertising to the other ISP.

You connect a single AOS-CX switch or VSF fabric to both ISPs. In this case, the switch has two eBGP neighbors, and it will advertise best routes received from one to the other.

You should communicate with your ISPs to prevent issues in any of these scenarios.

Sometimes you are using BGP primarily to announce your public IP block. For traffic outbound to the Internet, you do not care about finding the best path. You might want to use the connection to one ISP as the primary connection and the other as a standby only if the first fails, or you might want to load balance across both. In cases such as these, you might ask your ISPs not to send any route advertisements to you. You can then create static default routes on the BGP routers. Or in some cases, you might want the ISP to send only a default route in BGP.

If you want to receive routing advertisements from the ISPs, you must ensure that your router does not pass them on to the other ISP. You can ask the ISP router to filter inbound advertisements from you. But it is best practice to filter outbound advertisements yourself.



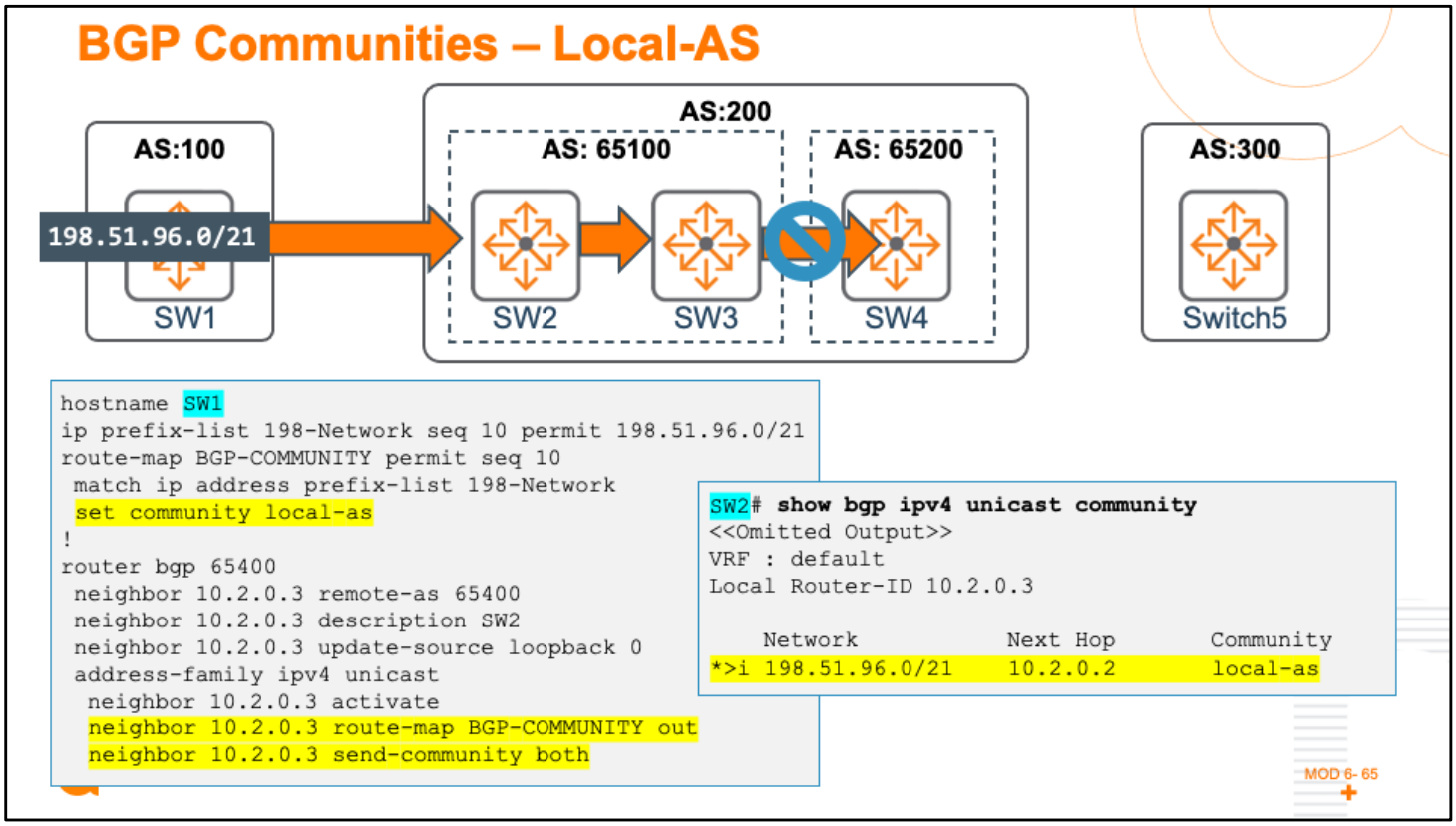
The No-Advertise community (0xFFFFFF02 or 4,294,967,042) specifies that routes with this community should not be advertised to any BGP peer. The No-Advertise BGP community can be advertised from an upstream BGP peer or locally with an inbound BGP policy. In either method, the No-Advertise community is set in the BGP FIB table that affects outbound route advertisement.

SW3 cannot see the route in its FIB table because the route was never advertised to it from SW2. The configuration of SW1 is:

hostname SW1
ip prefix-list 198-Network seq 10 permit 198.51.96.0/21
route-map BGP-COMMUNITY permit seq 10
 match ip address prefix-list 198-Network
 set community no-advertise
!
router bgp 65400
 neighbor 10.2.0.3 remote-as 65400
 neighbor 10.2.0.3 description Agg2
 neighbor 10.2.0.3 update-source loopback 0
 address-family ipv4 unicast

```
 neighbor 10.2.0.3 activate
 neighbor 10.2.0.3 route-map BGP-COMMUNITY out
 neighbor 10.2.0.3 send-community both
 aggregate-address 198.51.96.0/21 summary-only
exit-address-family
```

From SW2, you can run the show bgp ipv4 unicast community command to verify that this device is receiving the community.



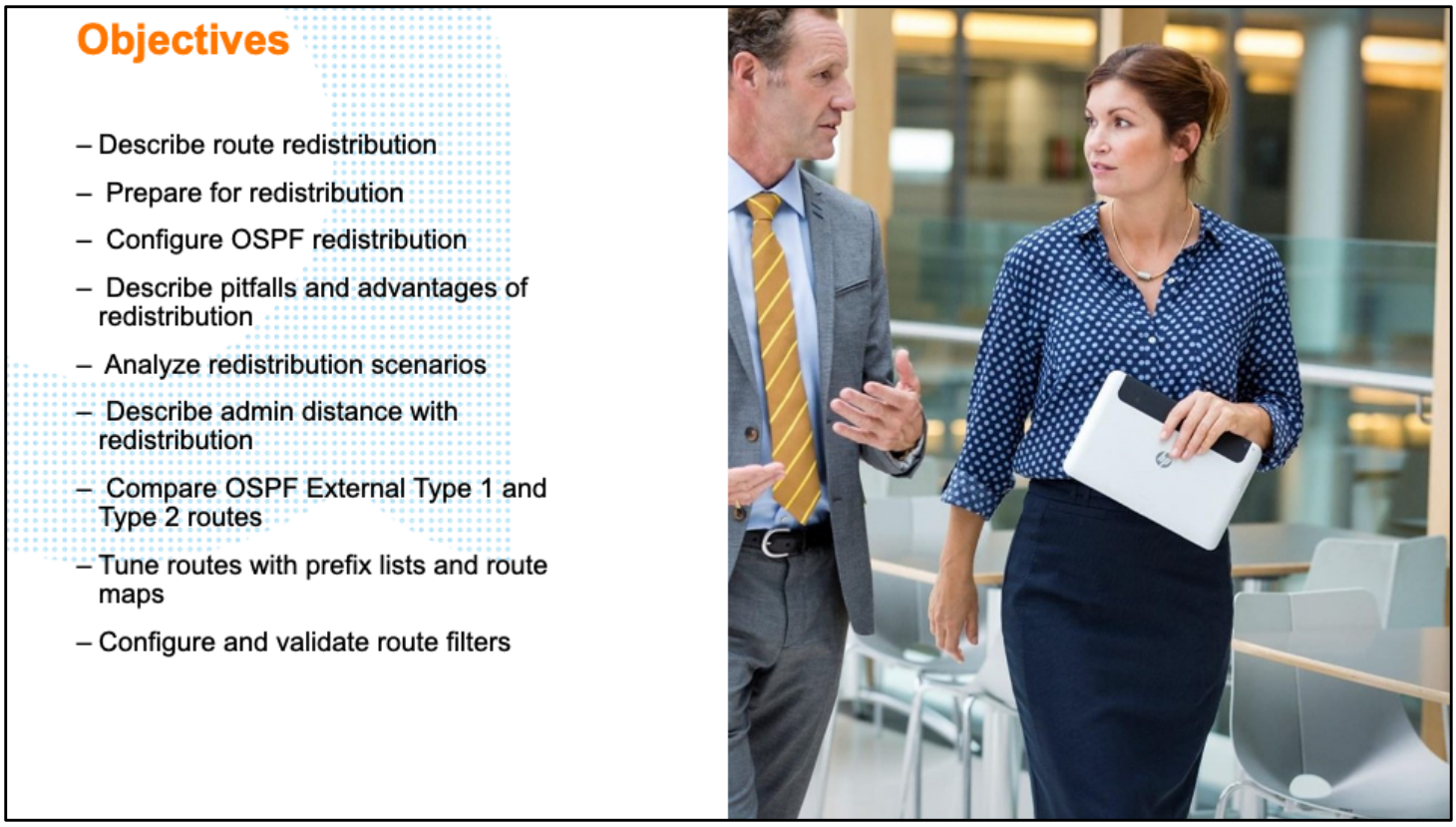
The No_Export_SubConfed community (0xFFFFFF03 or 4,294,967,043) known as the Local-AS community specifies that a route with this community is not advertised outside of the local AS. If the router receiving a route with the Local-AS community is a confederation member, the route can be advertised only within the sub-AS (Member-AS) and is not advertised between Member-ASs.

The figure demonstrates that SW1 is advertising the 10.1.1.0/24 network to SW2, which advertises to SW3. SW3 does not advertise the prefix on to SW4, so this device does not have the 10.1.1.0/24 network in its BGP table.

The configuration is similar to the previous example with the difference that we set the community local-as in the route-map. To verify the community received you must issue the show bgp ipv4 unicast community command.


Lab Activity
Lab 06


Rev 20.41
aruba
a Hewlett Packard Enterprise company
Route Redistribution
Aruba Advanced Switching, Troubleshooting and Solutions (ASTS)



After completing this module, you will have a very good understanding of route redistribution and related topics. You will be able to describe route redistribution, and how to be methodical in your approach to prepare for and configure route redistribution, with a focus on OSPF and static routes.

You will be able to describe common pitfalls or challenges, as well as the many advantages of route redistribution.

You will be able to analyze redistribution scenarios with various show commands, and to compare this output to a network diagram to facilitate troubleshooting and validation.

You will learn to describe how redistribution works in conjunction with administrative distance, and the difference between OSPF Type 1 and Type 2 external routes.

You will be able to tune routes for more optimal path selection using prefix lists and route maps.

Finally, you will learn to configure and validate route filters, to improve the security and efficiency of your network.

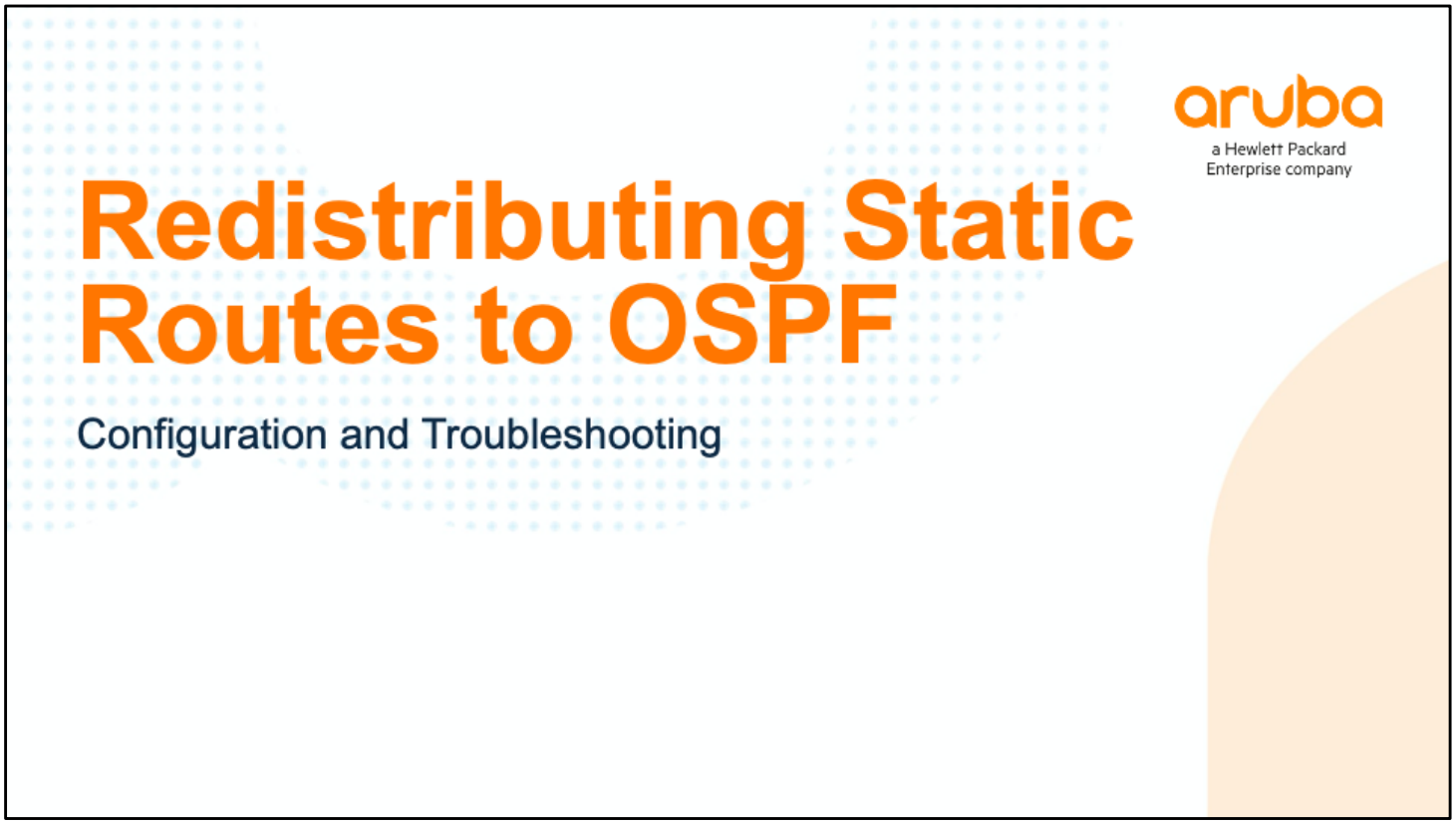
Overview

Intro to Redistribution

Advanced Redistribution

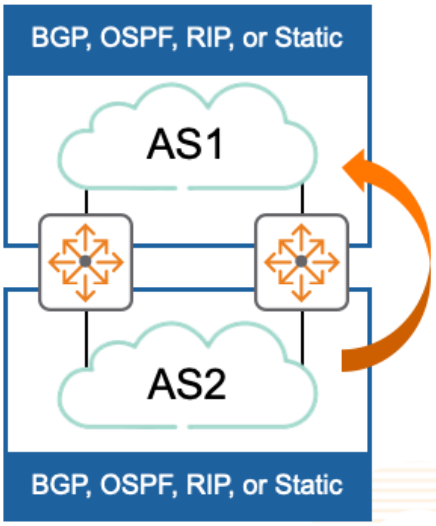


MOD 7- 3


aruba
a Hewlett Packard
Enterprise company
Redistributing Static Routes to OSPF
Configuration and Troubleshooting

Route Redistribution Overview

Objective	Inject routes from a foreign routing protocol or AS
Adds Complexity	Homogenous networks are simple and easy.
Use Cases	• Support legacy systems • Complete a migration • Merge organizations • Take advantage of certain protocol characteristics



MOD 7- 5

Route redistribution is used to inject routes from one routing protocol or AS into another routing protocol or AS. In the figure the routers might take the OSPF routes in AS 2, convert them into a BGP-compatible format, and inject them into AS 1's BGP cloud. You might also configure the edge routers to take some or all of the BGP routes in AS 1 and inject them into the OSPF AS 2.

This is merely an example. You can redistribute from any routing protocol (BGP, OSPF, RIPv2, or static) into any other routing protocol. But why do this? Dealing with two different routing protocols, redistribution, and inevitable tuning increases the complexity of deployment, management, and troubleshooting. Running a single, homogenous network is simple and easy. You don't really want to use redistribution, but you might have to. What are the use cases?

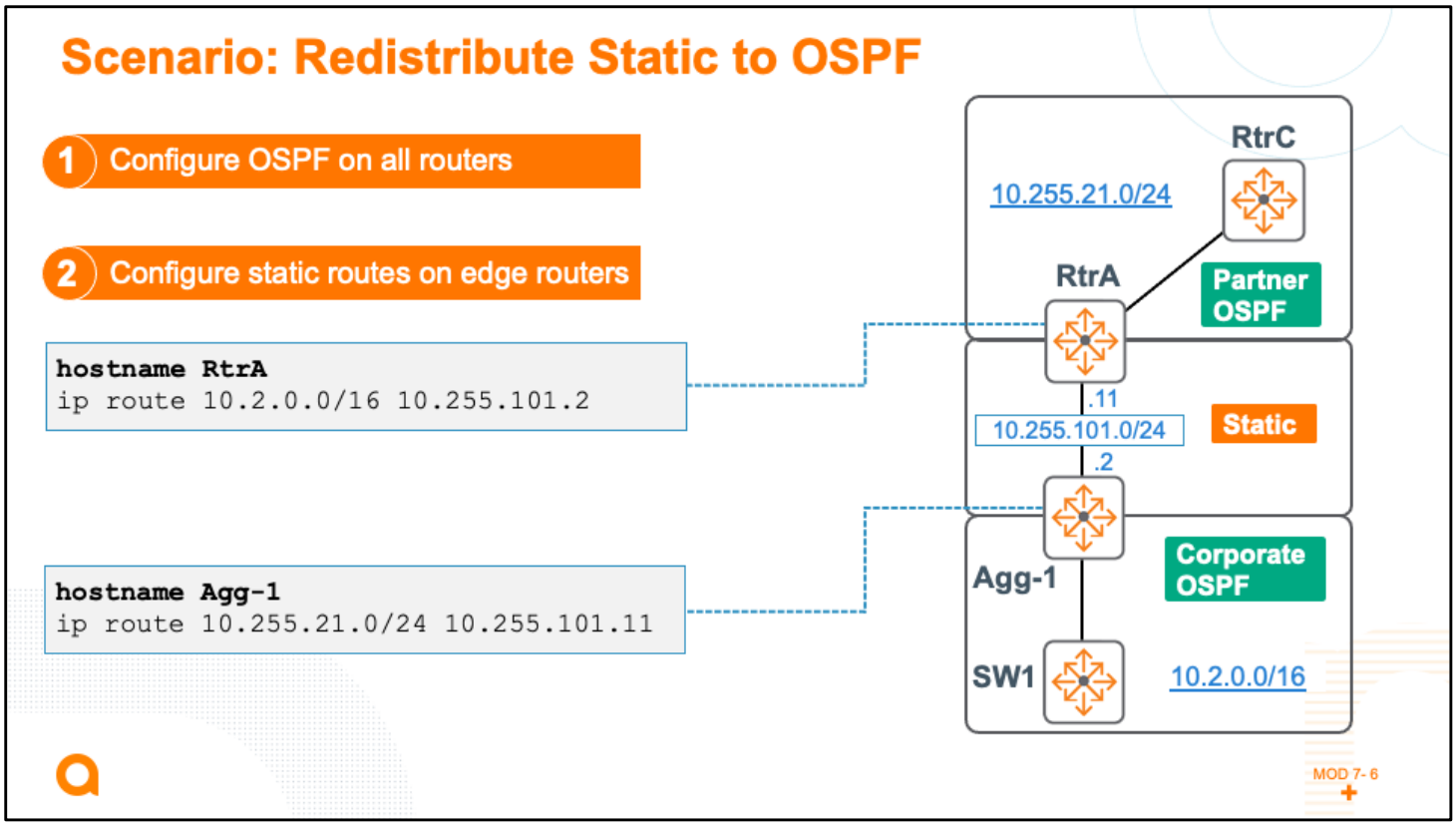
You might have to support a legacy system. Perhaps nearly all of your network runs OSPF, but some older portion only supports RIPv2, and it is not in the budget to upgrade yet. You redistribute those RIP-learned routes into the OSPF route tables. Now all OSPF routers know how to reach those RIP subnets. By definition, those OSPF routers doing the redistribution are ASBRs – they connect to some foreign routing protocol or AS. The other routers are blissfully unaware of any RIP subnets – their route tables contain only OSPF-learned routes. Now, those original RIP routes do show up as "External" OSPF routes, so there is some distinction, as you will soon learn.

Perhaps your company has merged with another. They run RIPv2, and you run OSPF. Since OSPF is more scalable, the long-term plan is probably to migrate everything to OSPF. But doing that properly might require some design time, some IP address adjustments, and so on. But the CEO has mandated that you “get us connected by the end of the week”. You might run route redistribution to get everyone connected, thus giving your team more time for planning a well-designed, rock-solid new architecture.

You might intentionally run a combination of two routing protocols – OSPF and BGP come to mind. In fact, you should study Aruba’s Mobile First Campus Validated Reference Design (VRD). This document recommends such a combination in certain situations, to strike a nice balance between the simplicity of OSPF running in each campus, and the scalability of BGP used to connect all campus locations into one cohesive system.

For smaller systems, OSPF and static routes is another common combination. You might want to redistribute the static routes into your OSPF or BGP domain, or use some alternate techniques, such as OSPF’s **default-information originate** option.

Let us now look at some basic route redistribution scenarios.



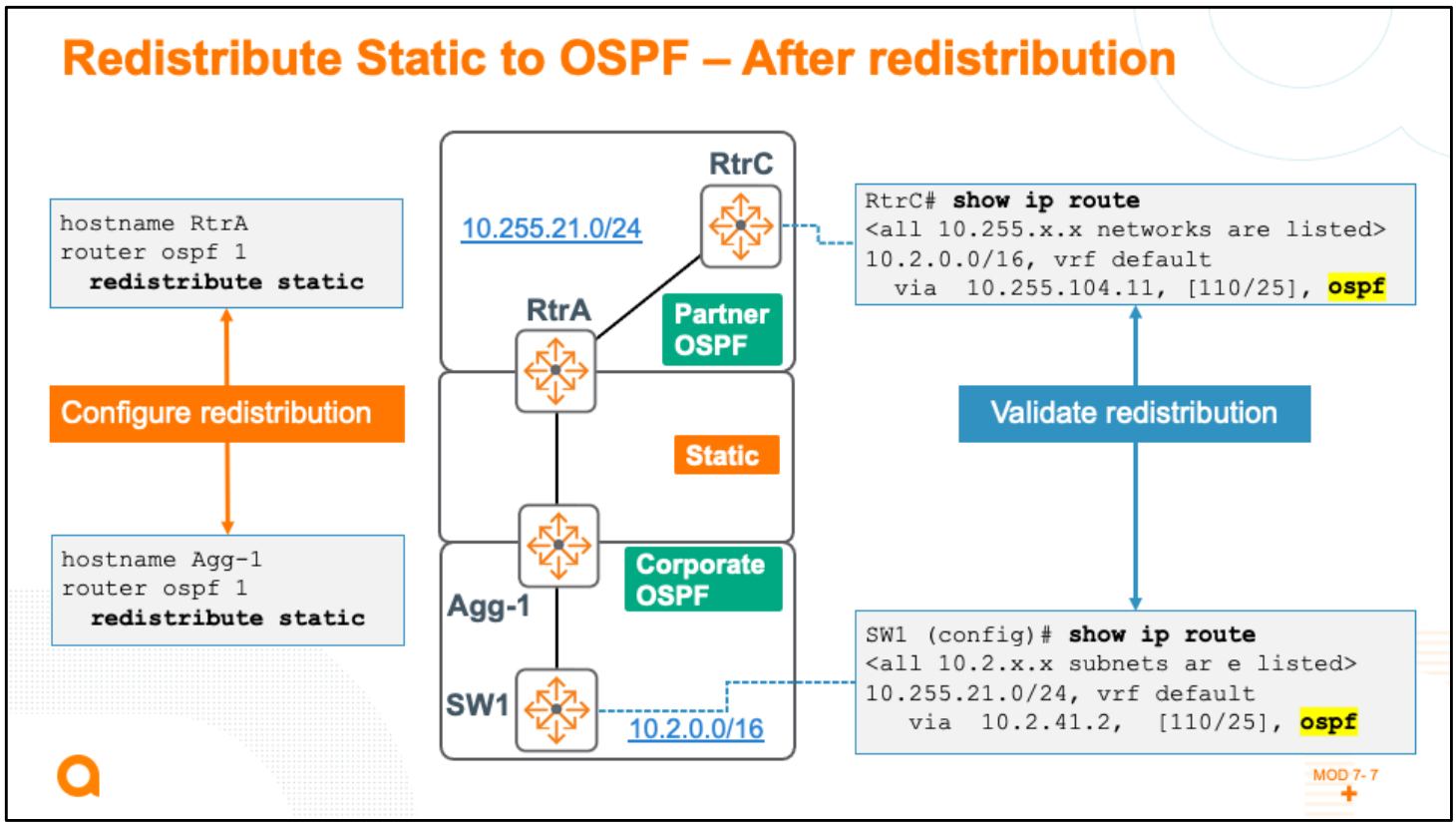
Scenario

Your company and a primary partner run large OSPF networks, which you have decided to connect using static routes and route redistribution. Only small portions of these large networks are shown in the figure. Your partner will allow access to three key subnets, as shown in the figure. You will allow the partner to access your 10.2.0.0/16 subnets.

Initial routing protocol configuration – no redistribution

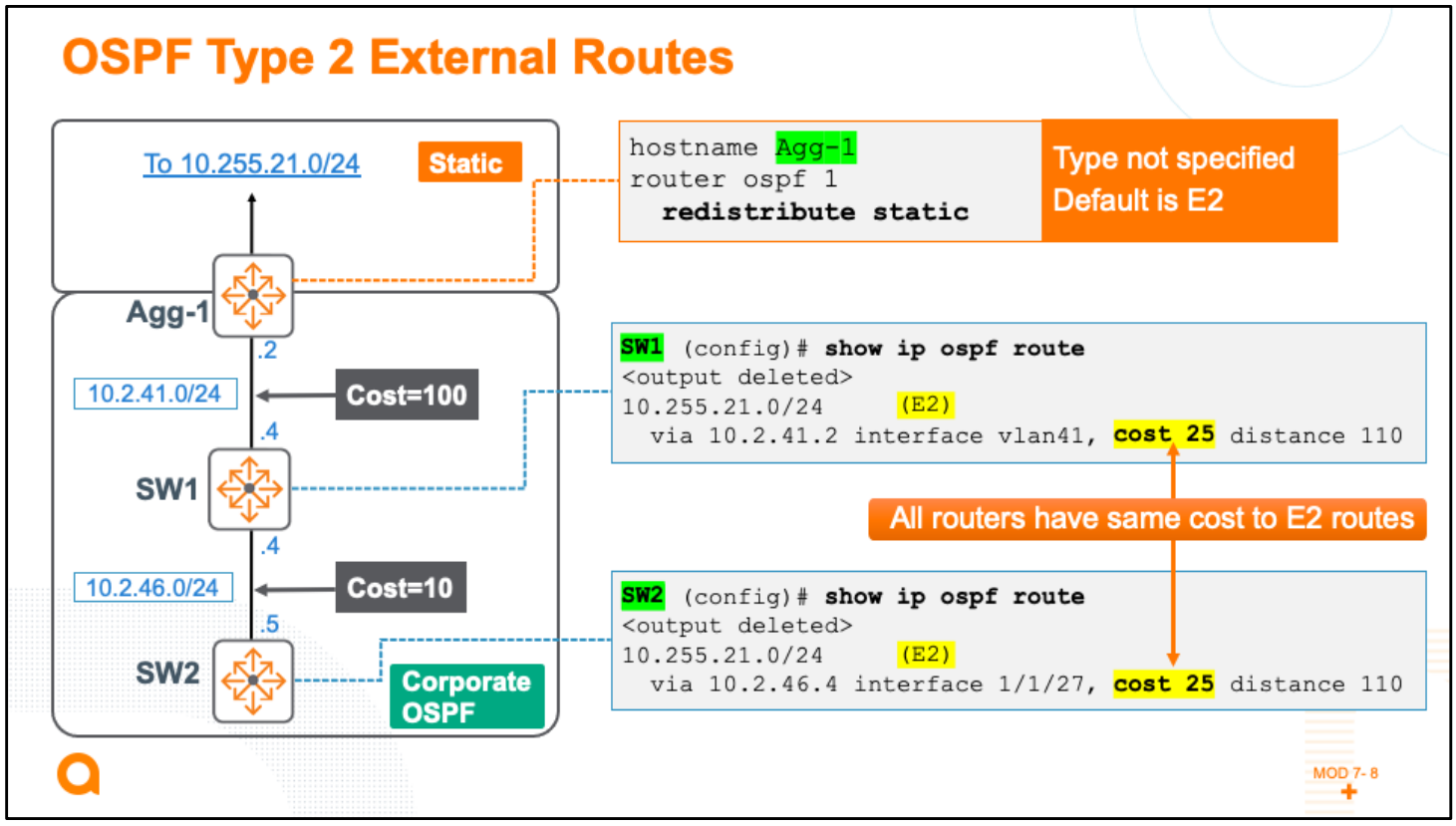
Configuring route redistribution is like general network troubleshooting, in that a methodical approach is best. Do not add the complexity of route redistribution until you have verified that fundamental routing operation is good. First, you and the partner configure OSPF on all routers in your respective networks, and validate that neighbors are established, and route tables are populated as expected.
Then you both configure static routes on your edge routers, as shown in the figure.

Now stop and verify. What do you expect to see?



The figure shows the edge routers are configured to redistribute static routes into OSPF. This means they find the static routes in their route tables, convert them into an OSPF format, and advertise them to their OSPF peers. This new configuration has no affect on edge router route tables – only on the OSPF peers that receive these redistributed route advertisements.

So, you check the route tables on the internal routers to validate that the edge static routes appear as OSPF routes. The figure shows that this is true, with the **show ip route** command. You see the routes are learned via OSPF, with an admin distance of 110 (the default for OSPF) and a cost of 25 (the default cost for static routes redistributed to OSPF).

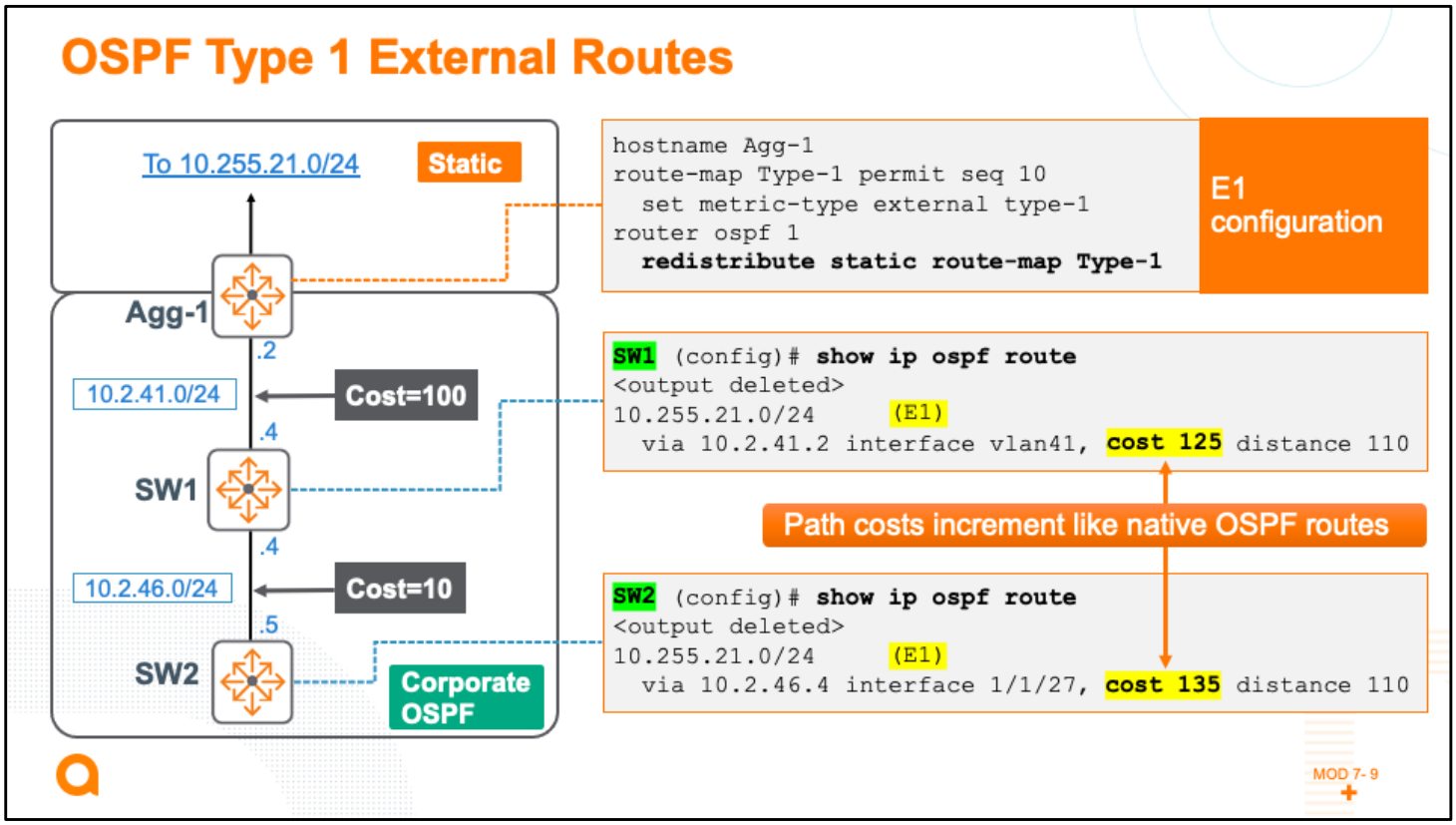


This example is focuses on one of the three redistributed routes (the other two have the same route type and cost). You can tell the routes shown are redistributed because they are labeled “E2”. They are redistributed as OSPF Type 2 external routes by default.

When foreign subnets are redistributed as OSPF Type 2 routes, all routers in the AS will have the same costs to reach those networks. In other words, internal path costs are not considered. This fact is proven in the **show ip ospf route** output for SW1 and SW2, as shown in the figure.

Agg-1 injected these subnets into the OSPF network as E2 routes with a cost of 25. Router SW1 in turn advertised these routes to SW2 with a cost of 25. Even though the link between SW1 and SW2 has a cost of 10, this cost is ignored – the cost for SW2 remains at 25. This can cause sub-optimal path decisions.

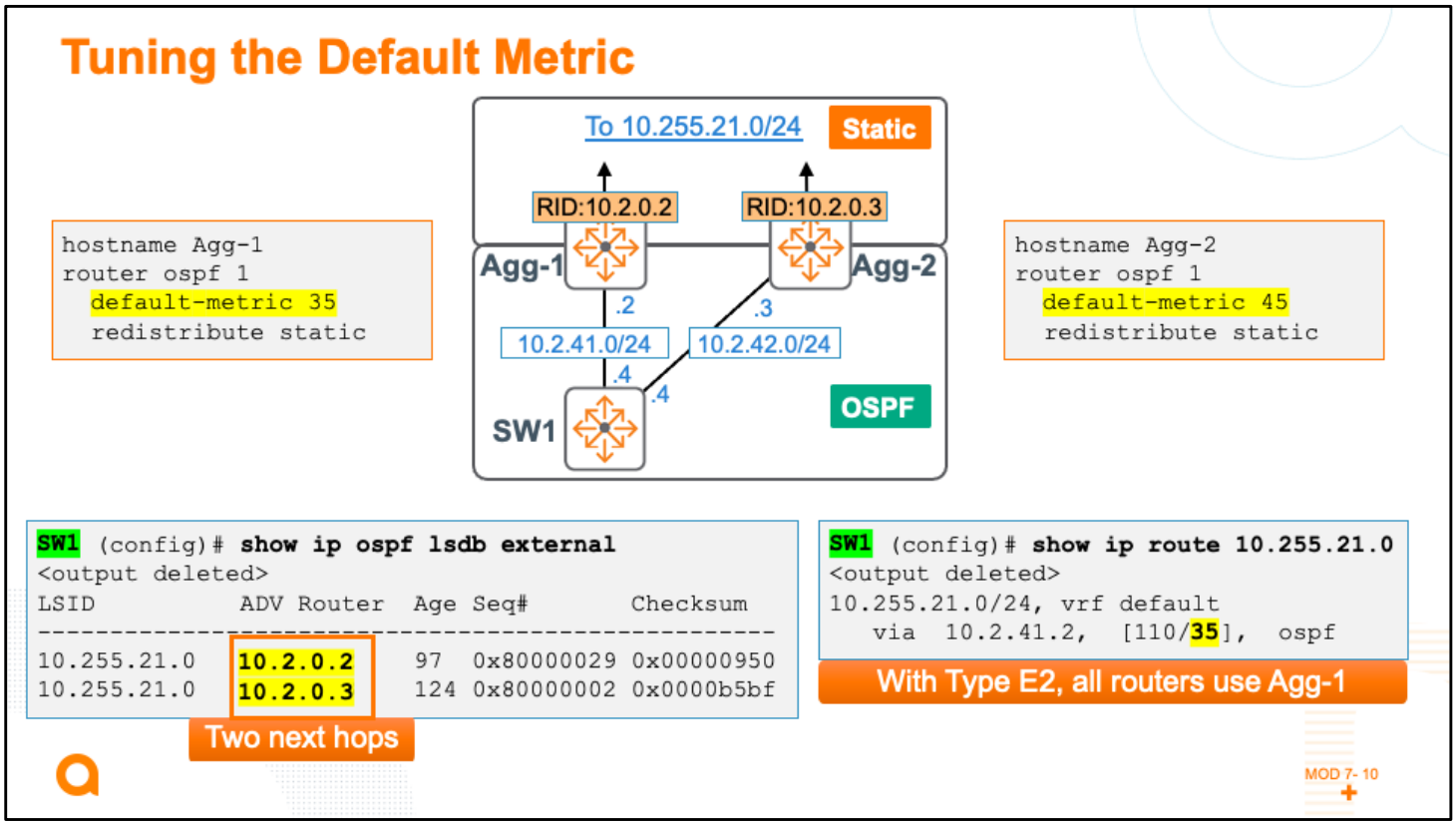
E2 can be used for a deterministic exit point in the network, independent of the internal cost, to deal with edge firewalls that require symmetric routing. For other scenarios, you can tell your routers to redistribute routes as OSPF Type 1 routes.



This example shows how to configure all redistributed static routes as external Type 1 routes. Recall that for Type 1 routes internal path costs are considered. Now you see that internal path costs are considered, which can help to mitigate the risk of sub-optimal path decisions.

In simpler deployments, other variables may still enable optimal path selection with E2 routes. However, in more complex redistribution scenarios involving OSPF, many experienced engineers prefer to redistribute routes into OSPF as E1 routes. You get more consistent, predictable scenarios with less risk of sub-optimal paths.

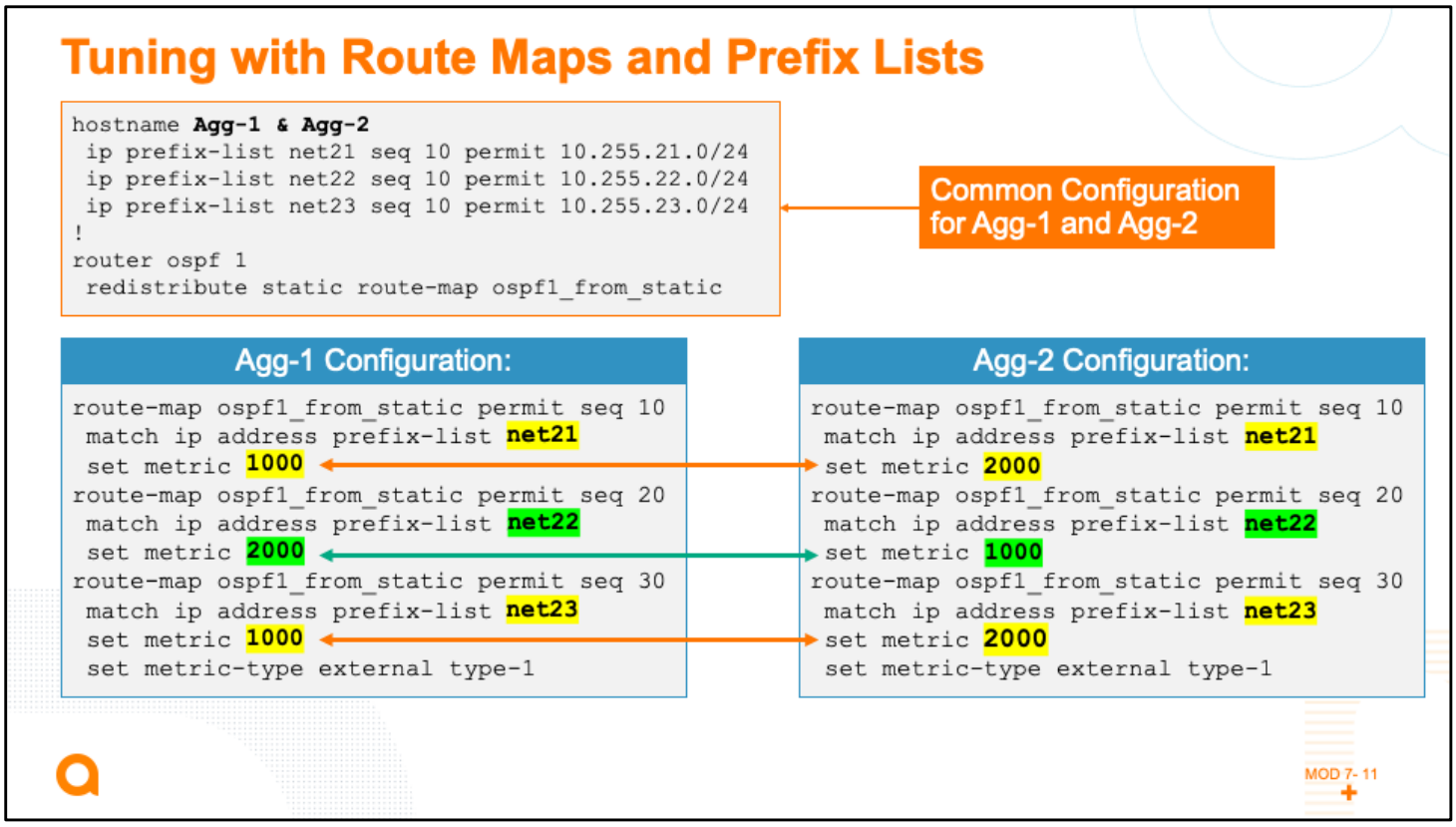
You have seen static-to-OSPF route redistribution with a single router, and a bit of route tuning. Now let's look at route redistribution with two edge routers into your AS, along with some additional tuning techniques.



This example shows the scenario, but with the default metric changed on both devices. Agg-1 redistributes all routes with a default metric of 35, while Agg-2 uses a default metric of 45.

This means that all internal routers will prefer to use Agg-1 (10.2.42.2) to reach the external destinations. Since you are redistributing routes as Type E2, ALL routers in your AS will use Agg-1 as the preferred exit point. Of course, both routes still show up in the LSDB, because both are VALID paths. However, the path via Agg-1 is, of course, perceived as the best path – the one added to the route table.

If you used this technique while redistributing these routes as type E1, then internal path costs would be considered. So, your internal routers would load share between Agg-1 and Agg-2 if their internal cost to reach Agg-2 that was 10 less than its cost to reach Agg-1. (45 - 35 = 10). If it was at least 11 less, then Agg-2 would be the best path. If you left the Agg-1 default metric at 25, and changed the Agg-2 metric to 125, then internal routers would only use Agg-2 if the cost was at least 101 less than Agg-1.



Spend a few moments to analyze the configurations shown in the figure for Agg-1 and Agg-2. They both have the same three prefix lists defined, and nearly the same route map is configured and applied to the OSPF **redistribute static** command.

The difference is the cost (metric) applied to each match criteria in the route map.

You should expect your router SW1 to prefer Agg-1 to reach the 10.255.21.0 and .23.0 subnets, and to prefer Agg-2 to reach 10.255.22.0.

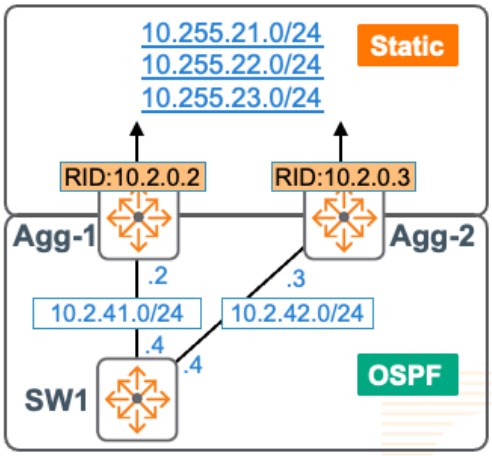
Validate Prefix List Tuning

```
SW1 (config)# show ip ospf lsdb external
<output deleted>
LSID            ADV Router   Age    Seq#       Checksum
-------------------------------------------------------
10.255.21.0     10.2.0.2     933    0x8000002e 0x0000d3b7
10.255.21.0     10.2.0.3     891    0x80000007 0x00004f76
10.255.22.0     10.2.0.2     933    0x8000002e 0x0000fba2
10.255.22.0     10.2.0.3     891    0x80000007 0x0000119f
10.255.23.0     10.2.0.2     933    0x8000002e 0x00003acf
10.255.23.0     10.2.0.3     891    0x80000007 0x0000b58e
```

External routes learned, Agg-1 and Agg-2 are the next-hops

```
SW1 (config)# show ip route
<output deleted>
10.255.21.0/24, vrf default
  via  10.2.41.2,  [110/1000], ospf
10.255.22.0/24, vrf default
  via  10.2.42.3,  [110/1000], ospf
10.255.23.0/24, vrf default
  via  10.2.41.2,  [110/1100], ospf
```

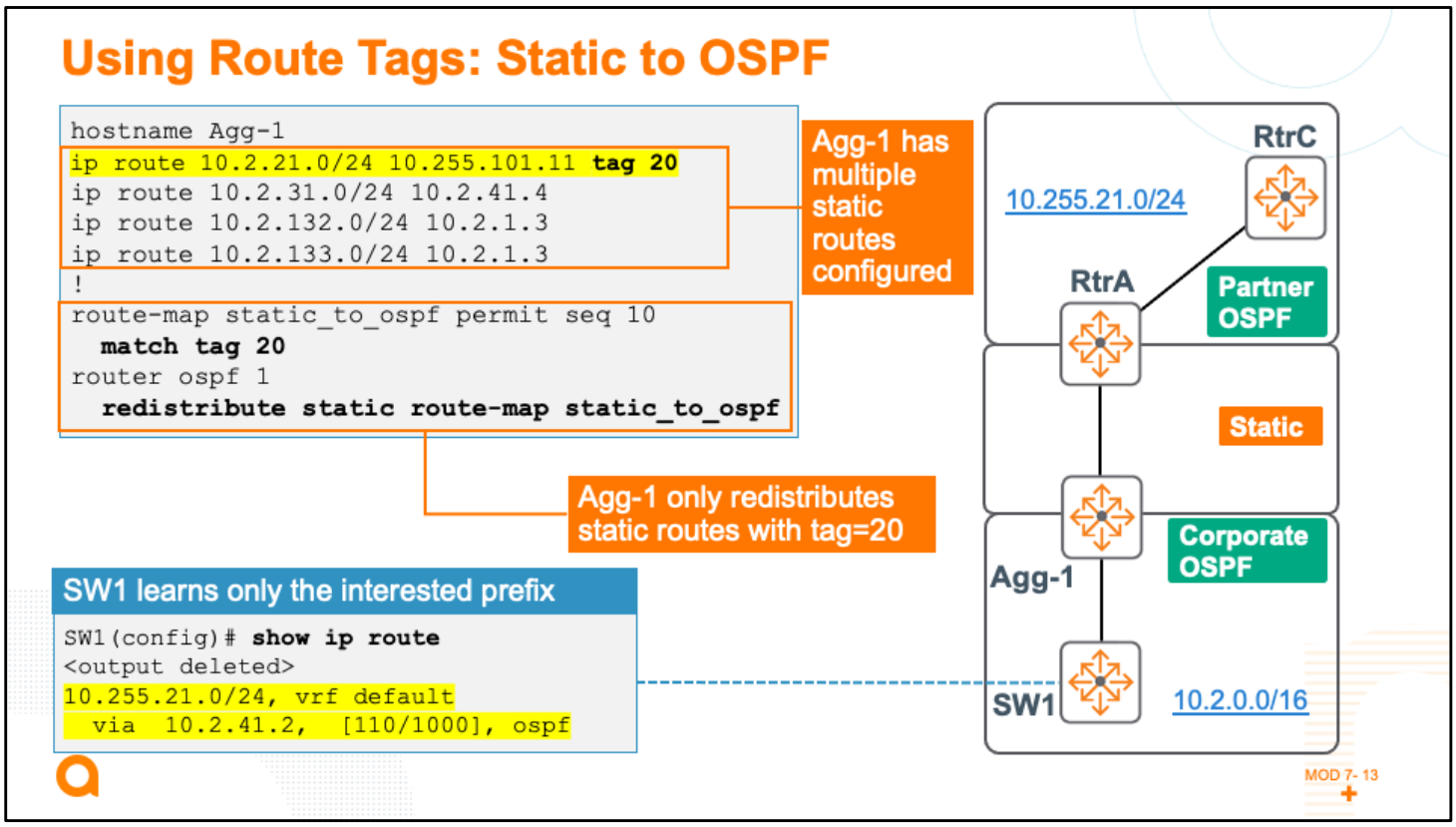
Tunning after applying the route-map



MOD 7- 12

The solution works as expected. Both Agg-1 and Agg-2 are listed in SW1's LSDB as next hops for all three external routes. However, router SW1 prefers Agg-1 to reach 10.255.21.0 and .23.0. It uses Agg-2 to reach 10.255.22.0.

Since both next hops are listed in the LSDB, you know that if the primary path fails, the alternate path will be added to the route table.



You can manipulate route paths and redistribution using route tags. A route tag is an arbitrary number that you assign to route or group of routes. Then downstream routers can make decisions based on those tags. For example, you might start by trying to use a prefix list but realize that the subnets you must permit are not contiguous. The prefix list might get long and complex. Instead, you tag certain routes with, say 20. Other routes are tagged with a 30. The number 20 has no inherent “priority” than 30. The tag merely provides another way for routers to distinguish between routes.

The figure shows how Agg-1 is configured with multiple static routes, redistributing all of them to Corporate OSPF does not make any sense, you must filter them. To accomplish this task the static route to 10.255.21.0/24 (the interested subnet) has been configured with tag = 20. Then you configure a route map that only permits routes with tag = 20 and apply it to the OSPF **redistribute static** command.

On SW1, now only the interesting subnet appears in its routing table.

Question #1

```
Access1# show ip ospf routes 10.255.21.0/24
Total Number of Routes : 2

10.255.21.0/24     (E2)
     via 10.1.41.2 interface vlan41, cost 25 distance 110
10.255.21.0/24     (E2)
     via 10.1.42.3 interface vlan42, cost 25 distance 110
```



From the output, Access1 is using ECMP to reach the 10.255.21.0/24 prefix, the administrator wishes to use Agg2 and not Agg1, which script can be used for this task?

a.
```
(config)# hostname Agg1
(config)# ip prefix-list net21 seq 10 permit 10.255.21.0/24
(config)# route-map static_to_ospf1 permit seq 10
(config-route-map-static_to_ospf1-10)# match ip address prefix-list net21
(config-route-map-static_to_ospf1-10)# set metric 1000
```

b.
```
(config)# hostname Agg2
(config)# router ospf1
(config-ospf-1)# default-metric 1000
```

Knowledge Check

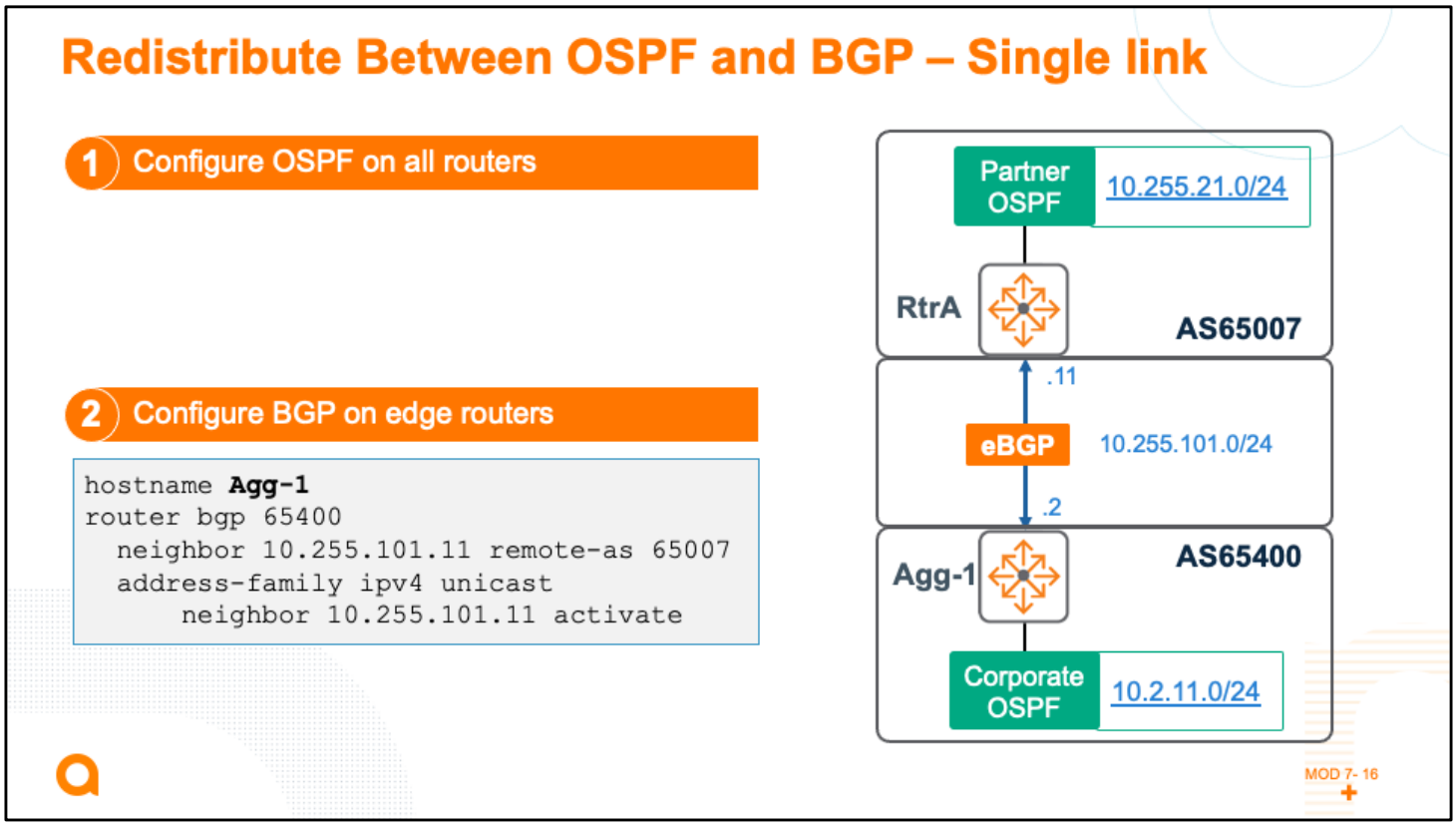
Correct answer:

A

Redistribute OSPF to BGP

Configuration and Troubleshooting

aruba
a Hewlett Packard Enterprise company



Scenario

Your company and a primary partner run large OSPF networks, which you have decided to connect using BGP and route redistribution. Only small portions of these large networks are shown in the figure. Your partner will allow access to one key subnet, as shown in the figure. You will allow the partner to access your 10.2.11.0/24 subnet.

Initial routing protocol configuration – no redistribution

Configuring route redistribution is like general network troubleshooting, in that a methodical approach is best. Do not add the complexity of route redistribution until you have verified that fundamental routing operation is good. First, you and the partner configure OSPF on all routers in your respective networks, and validate that neighbors are established, and route tables are populated as expected.

Then you configure BGP your edge router. The figure shows a typical configuration – that of Agg-1. The Partner BGP router is configured similarly.

Now stop and verify. What do you expect to see?

You use **show bgp ipv4 unicast summary** to validate BGP peers.

Agg1# **show bgp ipv4 unicast summary**

VRF : default

BGP Summary

Local AS : 64500 BGP Router Identifier : 10.2.0.2

Peers : 1 Log Neighbor Changes : No

Cfg. Hold Time : 180 Cfg. Keep Alive : 60

Neighbor Remote-AS MsgRcvd MsgSent Up/Down Time State AdminStatus

10.255.101.11 65007 2884 2881 01d:17h:48m **Established** Up

Due to the current configuration in AS65007, the partner AS 65007 is not advertising BGP routes to your edge router – Agg-1. Similarly, you are not advertising BGP routes to the partner router RtrA.

Agg1# **show bgp ipv4 unicast neighbors 10.255.101.11 advertised-routes**

Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,

i internal, e external S Stale, R Removed

Origin codes: i - IGP, e - EGP, ? - incomplete

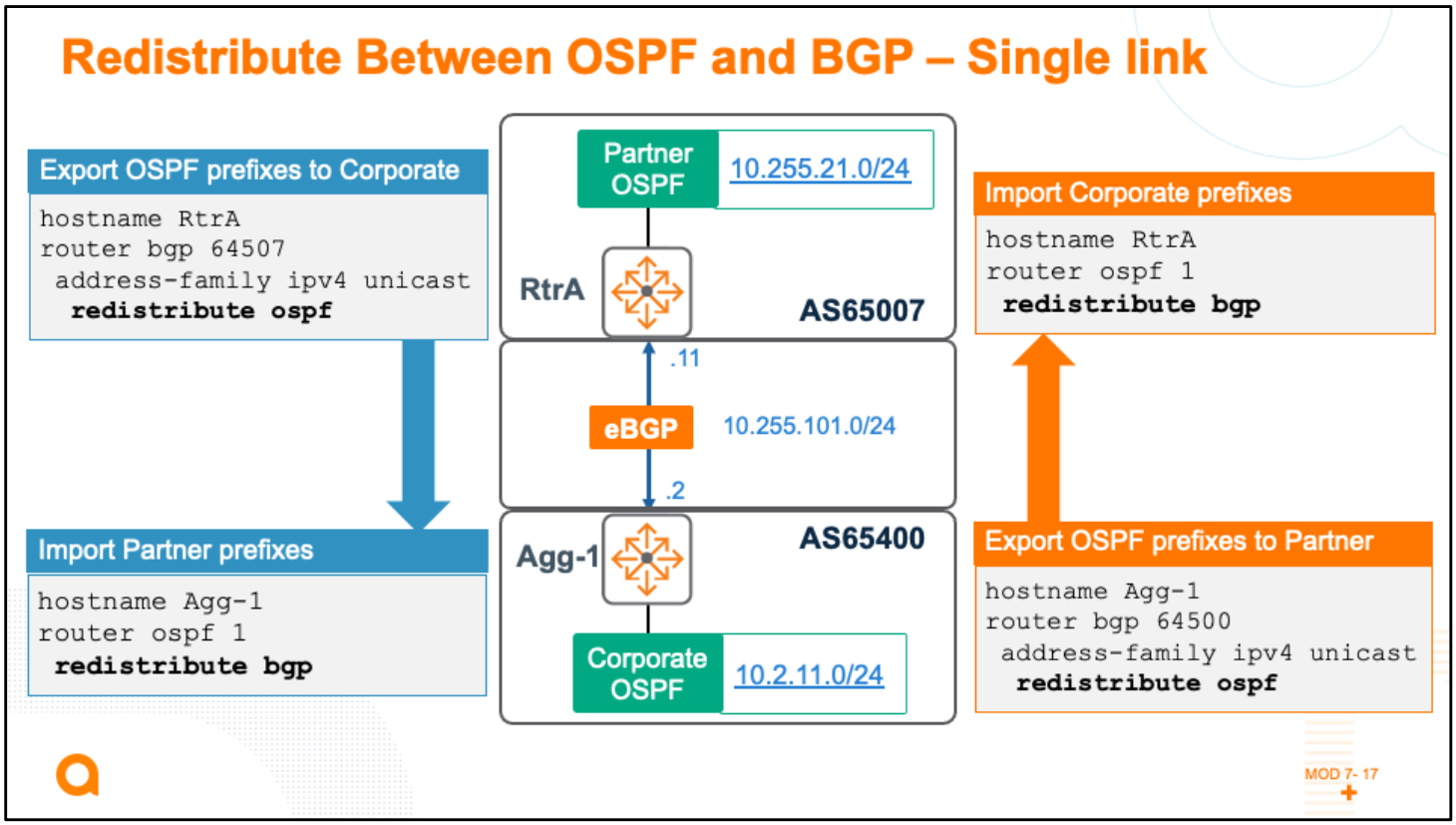
VRF : default

Local Router-ID 10.2.0.2

Network Nexthop Metric LocPrf Weight Path

Total number of entries 0

One way to get eBGP peers to advertise routes is by configuring route redistribution.

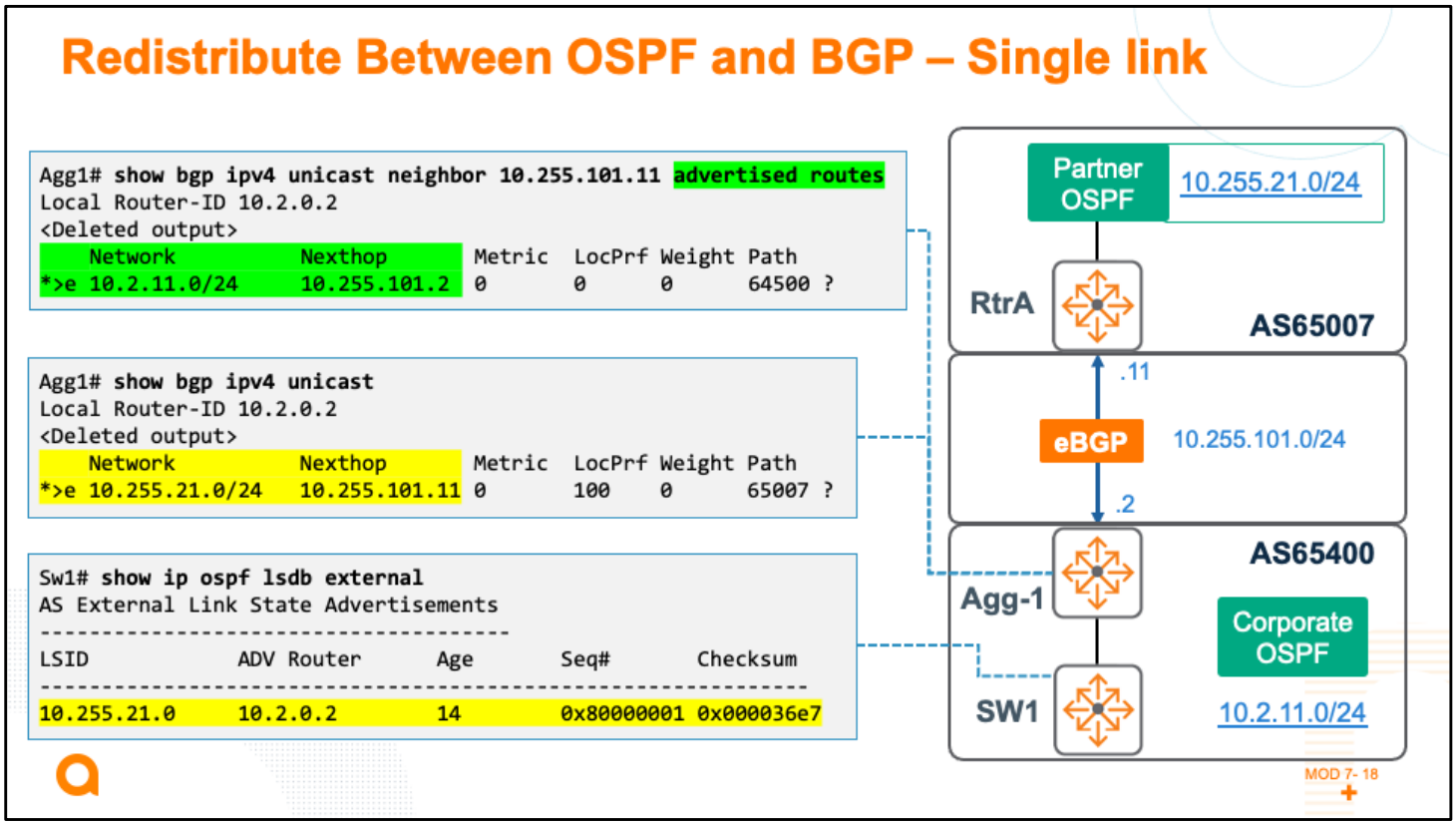


The redistribution process is a simple mechanism, in this scenario to export the internal OSPF prefixes to the other organization the administrator simply redistributes these prefixes into the BGP process.

router bgp <as-number>
address-family ipv4 unicast
redistribute ospf

To import the prefixes from the peer organization, you must use the inverse logic, prefixes are learned via eBGP and you must redistribute those prefixes into the OSPF process, this will make them available to internal routers.

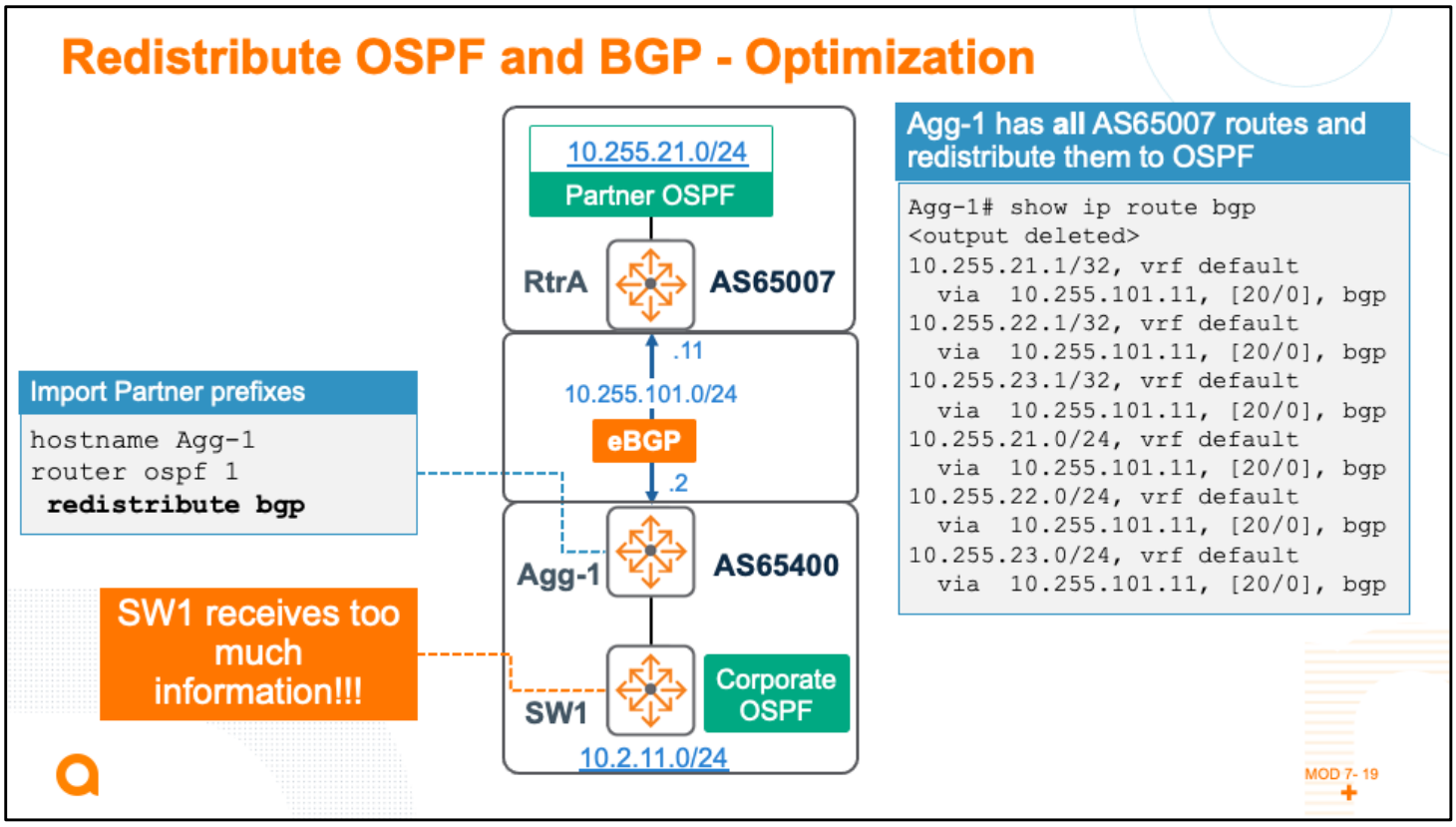
router ospf <process-id>
redistribute bgp



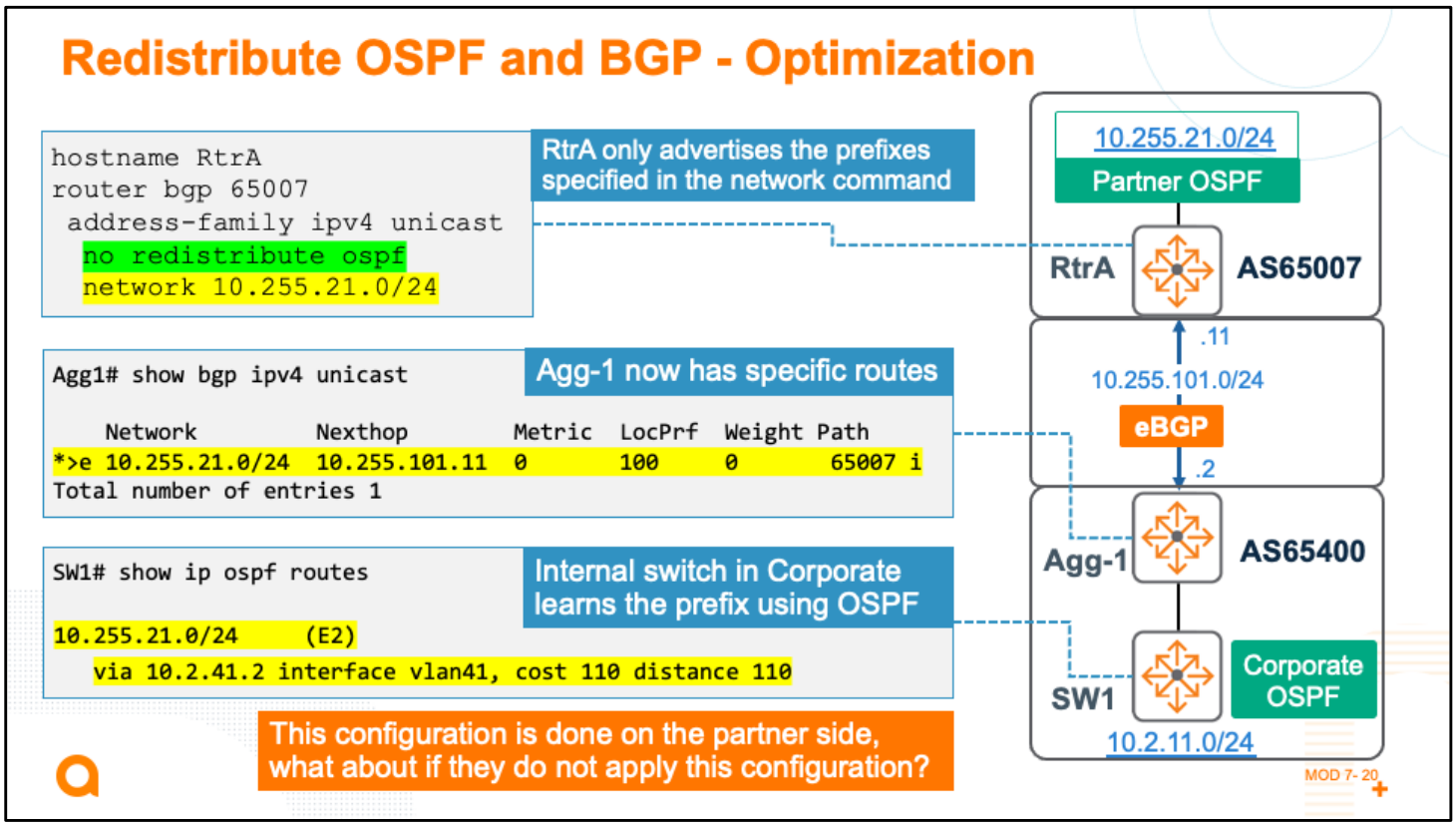
Using the **show bgp ipv4 unicast neighbor <neighbor-ip> advertised routes** command we can verify the prefixes that we are advertising to our BGP peer. In this example Agg-1 advertises the prefix 10.2.11.0/24 successfully to RtrA.

In a similar way we can verify the information that we receive from the BGP peer, this time you should use the **show bgp ipv4 unicast** command.

The output at bottom shows how SW1 which only run OSPF receive properly an External route from 10.2.0.2 (Agg-1) to 10.255.21.0/24 (the Partner prefix).



As we discussed, the redistribution process to import external prefixes is very simple. You only need to use the redistribution command in the correct routing process. The problem that we present in this slide is that potentially hundreds or thousands of subnets could be learned by the redistribution process. You end up wasting router and bandwidth resources and increasing your security risk by over-sharing. Corporate should only learn about the 10.255.21.0/24 prefix and the partner should only learn about 10.2.11.0/24 subnet.



In the past activity, you used route redistribution to get the partner prefix of 10.255.21.0/24, but this resulted in too many advertised routes. You might recall that you could apply a route map to filter these routes. However, an easier method might be to simply not advertise so much in the first place.

So, you remove all redistribution commands and configure some network statements, as shown in the figure. RouterA is configured to advertise the target 10.255.21.0/24 subnet.

Looking at the route table, you see that Agg-1 has a single eBGP route that was configured on RouterA, instead of all routes as in the like the previous example. Internal in the corporate will learn about 10.255.21.0/24 since Agg-1 is redistributing the external prefix into OSPF. We can verify this in SW1.

SW1# show ip ospf routes

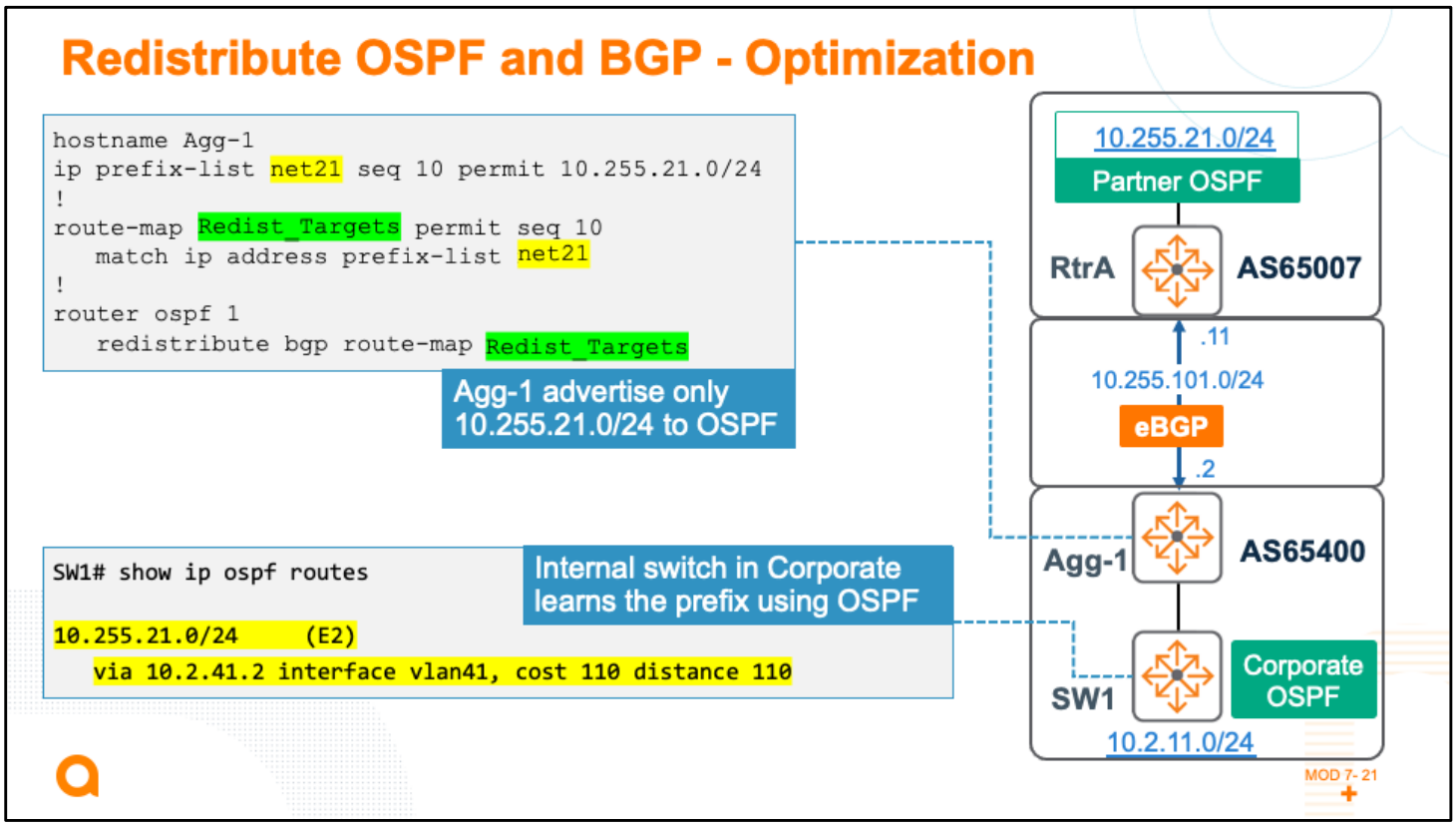
10.255.21.0/24 (E2)

via 10.2.41.2 interface vlan41, cost 110 distance 110

In a similar way, Agg-1 can only advertise specific prefixes to the Partner, for example 10.211.0/24. The configuration is presented in the following lines:

```
hostname Agg-1
router bgp 64500
  address-family ipv4 unicast
    no redistribute ospf
    network 10.2.11.0/24
```

The problem with this approach is that you depend on the advertising router to filter prefixes and select the ones you are interest on. In real scenarios, this is not always the case.



Your edge router (Agg-1) has the target subnets, learned via BGP. To get them into your internal routers, you decide to redistribute BGP into OSPF. You must do this with precision, so you create a prefix list as match criteria for a route map, then apply that route map to the redistribution.

Look at the prefix list in the figure, it is matching the target subnet 10.255.21.0/24. You apply this to the route map, which in turn is applied to the redistribution command.

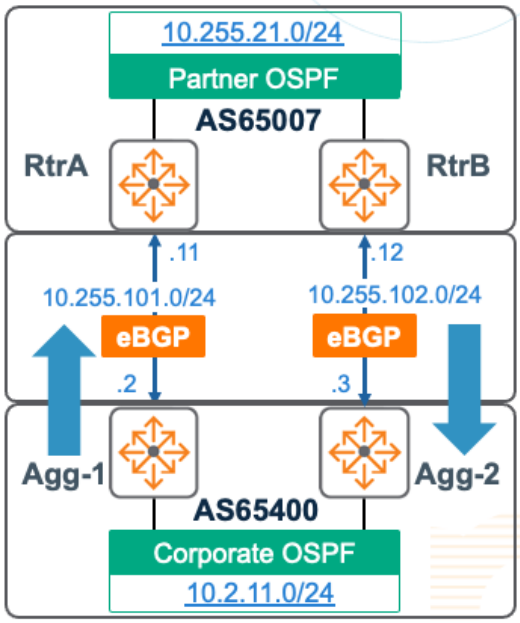
Looking at the SW1 route table, you see that only the target subnet has been redistributed from BGP into OSPF.

Redistribute Between OSPF and BGP – Dual links

1. RtrA and Agg-1 successfully redistribute prefixes
2. Add a second link between both organizations
   - Assume that the Partner has properly configured the second link
3. Redistribution BGP into OSPF in Agg-2

```
hostname Agg-2
router ospf 1
 redistribute bgp
```

4. Prefix 10.255.21.0/24 is advertised to RtrA
   - Agg-1 learns the prefix from OSPF and advertises it to RtrA



MOD 7- 22

A single link between both organizations creates a single point of failure, on situations where communication is critical for business, a second link is necessary.

On the slide you can see that a new connection has been established between RouterB and Agg-2. Let's analyze what the effect is of adding a new connection to the redistribution process.

1. Consider that redistribution was properly working between both organizations using the link between Agg-1 and RouterA. As we discussed previously this process (redistribution) helps to announce the internal prefixes to the partner and besides make available external prefixes.

2. The second link is added. To simplify this explanation, assume that the partner has properly configured the second link (they have already considered the route feedback situation – the situation that is explained in this example).

3. Using the same redistribution logic applied previously (on Agg-1), you import the partner prefix into the corporate OSPF in the second border router (Agg-2). This implies that the prefix 10.255.21.0/24 known by eBGP is redistributed into the ospf process 1. This prefix now is seen by all the internal OSPF routers as an External subnet.

hostname Agg-2

```
router ospf 1
  redistribute bgp
```

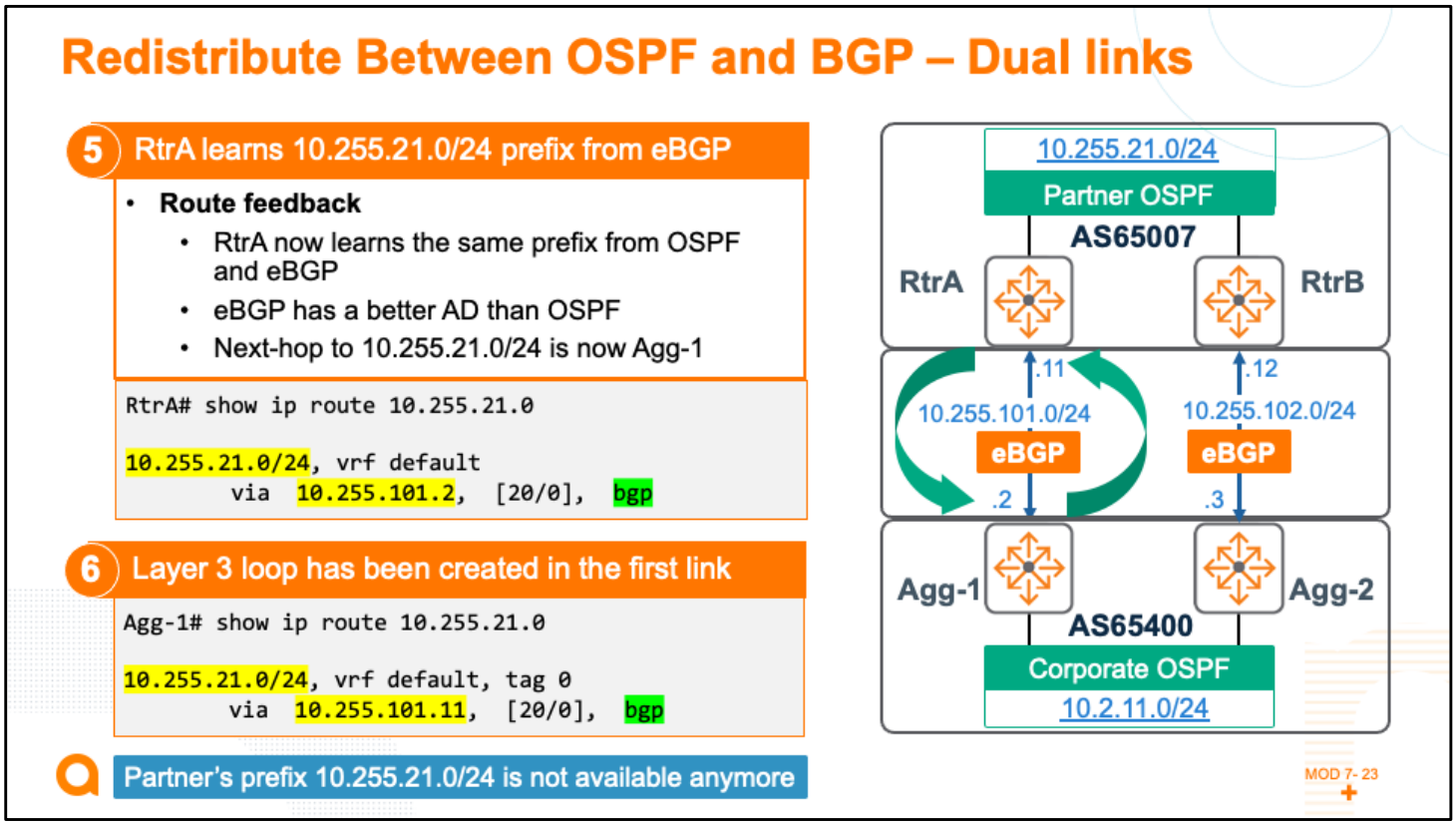
4. Agg-1 which participates in the OSPF domain learns that this prefix is reachable via Agg-2. Agg-1 will not use Agg-2 as next-hop since the same prefix is advertised by RouterA using eBGP. The lower administrative distance associated to eBGP will make Agg-1 discard the route announced by Agg-2. According to the current configuration in Agg-1 all the prefixes known by OSPF are advertised to RouterA. The 10.255.21.0/24 prefix is now advertised back to the RouterA via Agg-1.

We can confirm this by using the **show bgp ipv4 unicast neighbors 10.255.101.11 advertise-routes** command.

Agg1# **show bgp ipv4 unicast neighbors 10.255.101.11 advertised-routes**
<<Omitted Output>>

Network	Nexthop	Metric	LocPrf	Weight	Path
*>e 10.2.11.0/24	10.255.101.2	0	0	0	64500 ?
***>e 10.255.22.0/24**	**10.255.101.2**	**0**	**0**	**0**	**64500 ?**

Total number of entries 2



5. RouterA is experiencing a route feedback, from its point of view the prefix 10.255.21.0/24 is now reachable via Agg-1. The internal routing process in RouterA will conclude that the next-hop to use is Agg-1 because eBGP has a better Administrative Distance than OSPF. This conclusion is catastrophic for the communication, RouterA cannot longer reach the 10.255.21.0/24 subnet because is pointing to the wrong next-hop.

6. A layer 3 loop has been created, Agg-1 reaches the 10.255.21.0/24 via RouterA, and RouterA reaches the prefix via Agg-1.

Pings running from RouterA and SW1 (internal device in the AS 64500) demonstrate that communication is not possible with the current configuration.

RtrA# ping 10.255.21.1

PING 10.255.21.1 (10.255.21.1) 100(128) bytes of data.

From 10.255.101.2: icmp_seq=1 Redirect Host(New nexthop: 10.255.101.11)

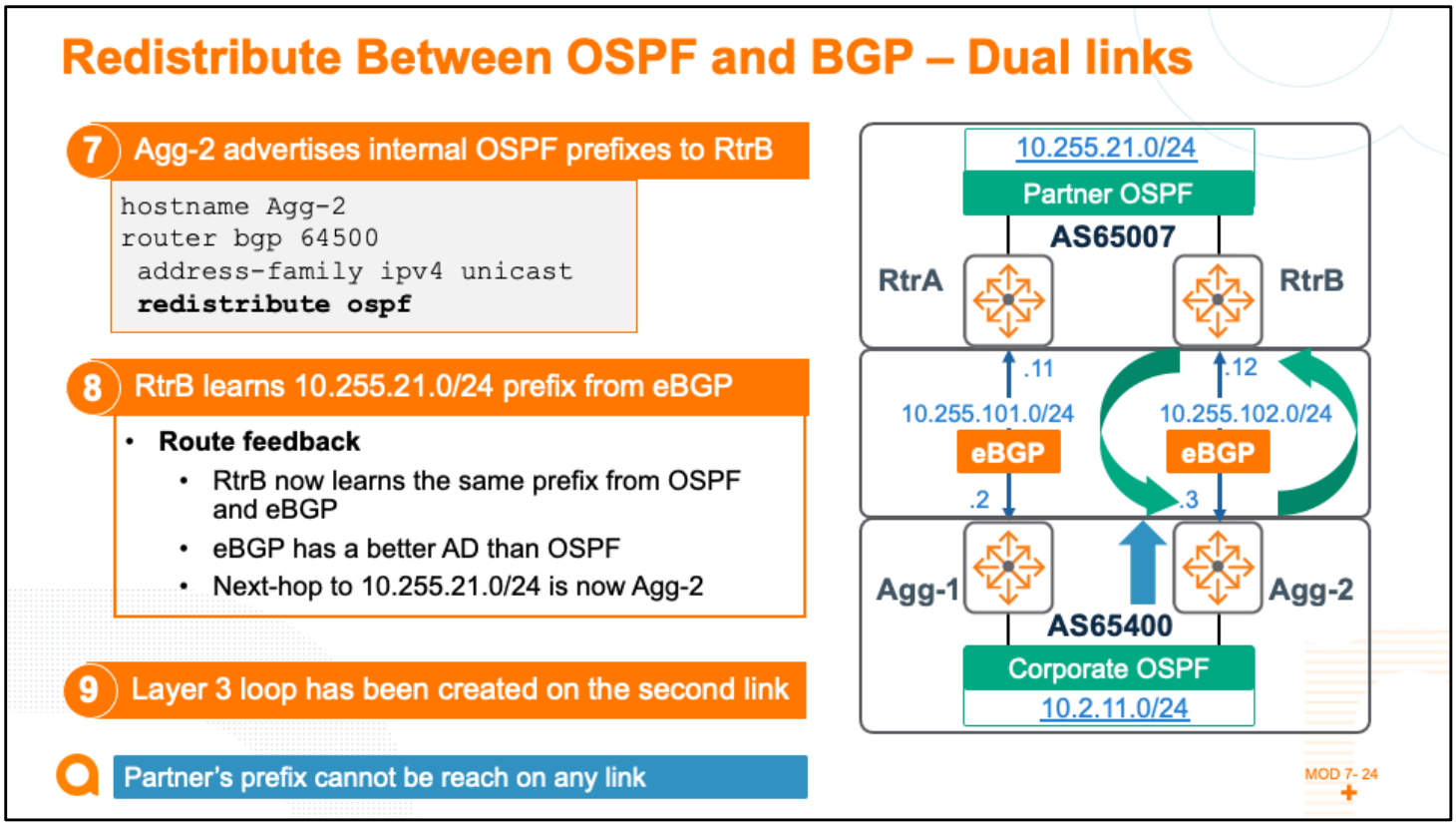
From 10.255.101.2: icmp_seq=1 Redirect Host(New nexthop: 10.255.101.11)

From 10.255.101.2: icmp_seq=1 Redirect Host(New nexthop: 10.255.101.11)

SW1# ping 10.255.21.1

PING 10.255.21.1 (10.255.21.1) 100(128) bytes of data.

```
From 10.255.101.11 icmp_seq=1 Time to live exceeded
From 10.255.101.11 icmp_seq=1 Time to live exceeded
From 10.255.101.11 icmp_seq=3 Time to live exceeded
```



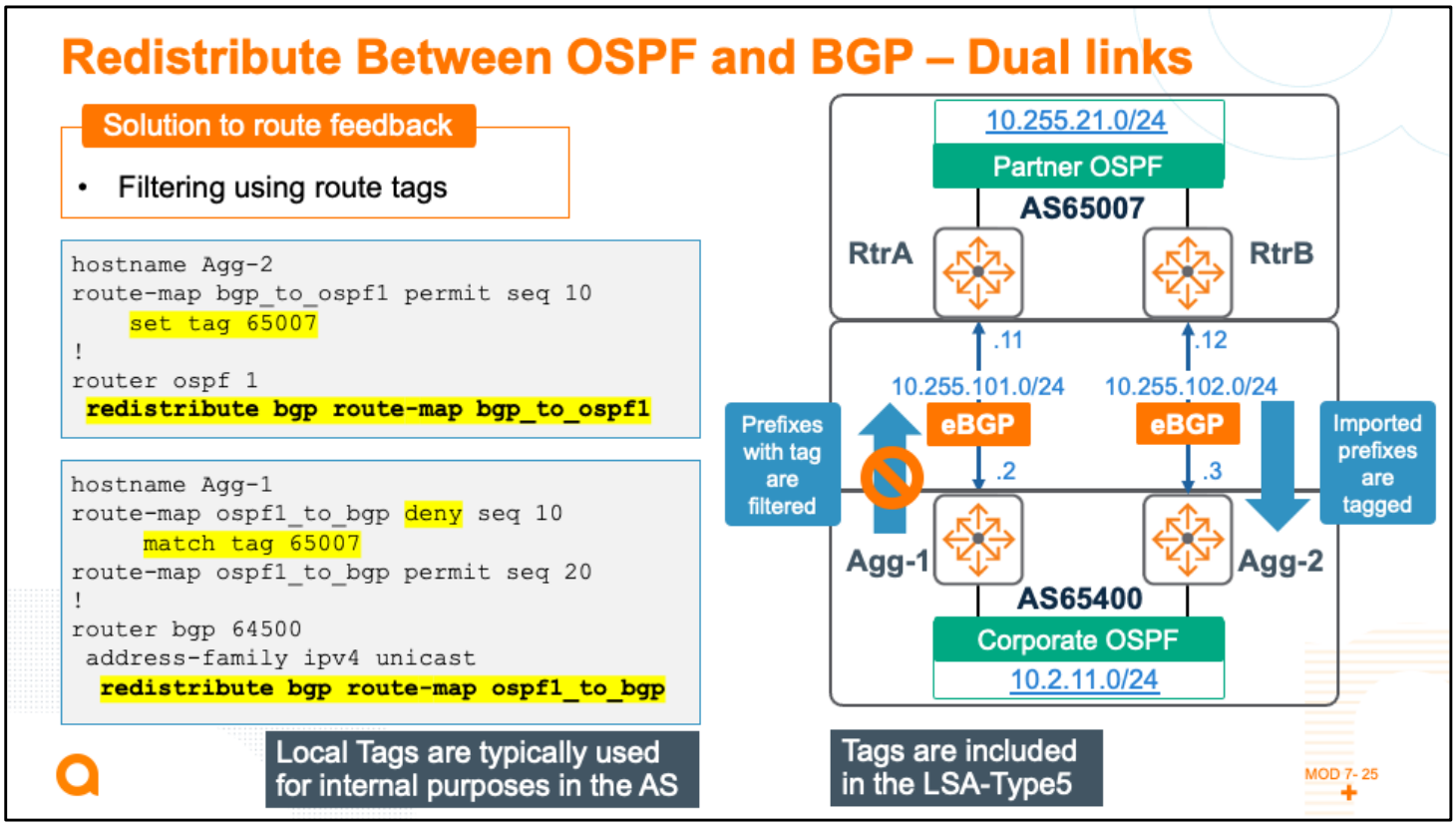
The communication using the link between Agg-1 and RouterA is not possible as we explained, let's analyze the second link.

7. In order to use the second link, we need to advertise the internal subnets to the partner using Agg-2. This task can be done by redistributing OSPF into BGP.

```
hostname Agg-2
router bgp 64500
 address-family ipv4 unicast
  redistribute ospf
```

8. The same route feedback situation will occur on the second link. The prefix 10.255.21.0/24 is imported from BGP to OSPF via Agg-1 and then it is advertised to RouterB via Agg-2 using BGP. RouterB will prefer the entry announced by Agg-2 due eBGP has a lower Administrative Distance compared to OSPF. RouterB cannot longer reach the internal prefix.

9. The layer 3 loop will occur also on the second link and the communication to the external prefix 10.255.21.0/24 will fail.



The solution to route feedback is filtering, you simply need to make sure the Partner prefix is not advertised back to that organization. To accomplish this task we will use route tags.

The first step is to mark the imported prefixes using a route tag. This task can be done by using a route-map and apply it on the redistribution of BGP into OSPF. The following lines present the configuration applied to Agg-2. The route tag used in this example is 65007.

```
hostname Agg-2
route-map bgp_to_ospf1 permit seq 10
    set tag 65007
!
router ospf 1
 redistribute bgp route-map bgp_to_ospf1
```

Note. The tag is included in OSPF Type 5 (external) LSA announcements. Thus, all OSPF routers that have the external route in their LSDB can see the tag. Also know that each route entry may only have one tag. You cannot add multiple tags to a route. Tags are typically used internally in an AS for your own use, remote BGP peers might strip the tags and thus they should not be heavily relied on.

The second step is to filter the partner prefixes, in this case the border router Agg-1 only announce the prefixes that does not have the route tag 65007. The configuration need it is:

```
hostname Agg-1
route-map ospf1_to_bgp deny seq 10
     match tag 65007
route-map ospf1_to_bgp permit seq 20
!
router bgp 64500
 address-family ipv4 unicast
  redistribute bgp route-map ospf1_to_bgp
```

In a similar way you must perform a similar filtering on the second link, the following lines present the configuration required for this task.

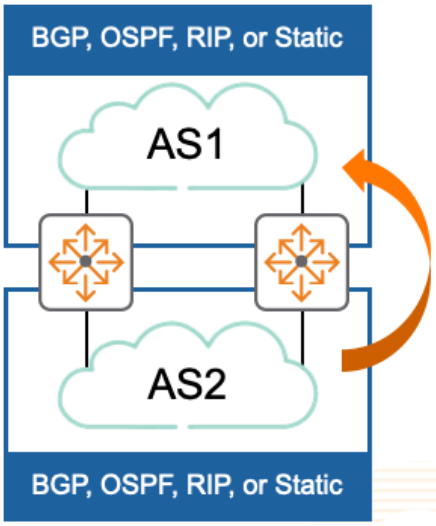
```
hostname Agg-1
route-map bgp_to_ospf1 permit seq 10
    set tag 65007
!
router ospf 1
 redistribute bgp route-map bgp_to_ospf1
---------------------------
 hostname Agg-2
 route-map ospf1_to_bgp deny seq 10
      match tag 65007
 route-map ospf1_to_bgp permit seq 20
 !
 router bgp 64500
  address-family ipv4 unicast
   redistribute bgp route-map ospf1_to_bgp
```

Notice that your partner should deployed a similar configuration to avoid route feedback.

Route Redistribution Tips

If possible
- Redistribute in one direction only – from edge to core
- Use default routes for the edge

If mutual redistribution is needed
- Beware of re-redistribution
- Use consistent metrics at every redistribution point
- Expect sub-optimal routing or route loops
- Triple-check your basic configuration before tuning
- OSPF: Same router is ASBR and DR



MOD 7- 26

If you must use route redistribution, you should make it as easy as possible for yourself and others. This means only redistributing in one direction whenever possible – from the "edge" protocol into the "core" protocol. Perhaps most of your entire network runs OSPF, that is your so-called "core" routing protocol. Perhaps only a few routers near the edge are still running RIPv2 or static routes. In cases like this, you might only need to redistribute the RIPv2 or static routes into OSPF. Once redistributed, all other OSPF routers will know those routes, and the best path to reach them.

However, do you really want to redistribute 15,000 OSPF routes into those tired, old RIPv2 routers? Probably not, especially when the RIPv2 network only has two possible paths to reach them – via the two (typically) ASBRs connected to both systems. Just configure default routes on the RIPv2 routers. You not only save resources on the routers, and bandwidth on the RIPv2 side of the house, you also avoid the complexity of mutual route redistribution, and the likelihood of sub-optimal paths, route loops, and the tuning required to mitigate these issues. Keep it simple if you can.

However, some scenarios require you to configure mutual redistribution. Perhaps you have two large but separate OSPF networks to connect - you must redistribute routes from AS 1's OSPF process into AS 2's OSPF process and redistribute AS 2's OSPF routes into AS 1. Both AS's require more granular path control and optimization so default routes simply will not suffice. Mutual redistribution carries with it the risk of creating route loops. However, you now understand the various techniques to eliminate these loops so you can always make it work.

Just be consistent, methodical, and follow a few simple rules:

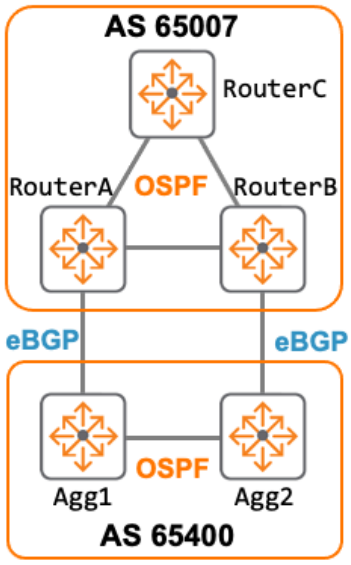
- **Beware of re-redistribution**. Route loops and/or sub-optimal paths are almost inevitable with mutual redistribution. You have learned how to use route maps, filters, and tagging to mitigate this risk.
- **Use consistent metrics at every redistribution point**. It is often best to have at least two routers doing mutual redistribution, for redundancy. It is best to start with both redistributing routers to inject the foreign routing protocol with the same cost. This consistency will make it easier for you to figure out why you have sub-optimal paths or route loops, because there is one less variable or difference to worry about. You might end up tuning things later, to have routers use different costs for all or part of the redistributed routes, but it is usually best to start with a common “seed metric” or cost for all redistributed routes.
- **Expect sub-optimal routing or route loops**. As stated before, sub-optimal routes and/or route loops is almost inevitable with mutual redistribution using two or more redistributing routers. Be methodical in your approach.
    - First make sure that both protocols (say BGP and OSPF for example) are flawless BEFORE you redistribute. Use all your show commands and verify that all expected routes are learned and optimal on a wide selection of routers – as many as feasible. If you add the complexity of redistribution to an already malfunctioning or misconfigured system, you are likely to get entangled in an excessively complicated structure that does not function. You may be convinced the problem is due to redistribution, simply because you didn’t notice the issue until after you did redistribution. This can cause you to misdirect your diagnostic efforts.
- **Triple-check your basic configuration before tuning**. This is simply another reminder to be methodical – make sure all routers are properly configured and have truly chosen optimal paths before you add the complexity of redistribution.

Another tip is specific to OSPF. It is best to make your OSPF DRs the same router as the ASBRs (the routers doing redistribution). This helps to reduce the amount of flooding. This is because the ASBR is the router most likely to detect changes, and so it directly advertises those changes to all other routers. If another router was the DR, the ASBR must first tell the DR, which would then flood the new information to others. This would make for a less efficient deployment.

Question #2

In the scenario, the administrator has redistributed OSPF into BGP and BGP into OSPF in Agg1 and RouterA only. With this configuration routes from the neighbor AS are reachable and the communication is successful. Now the administrator attempts the same redistribution process between Agg2 and RouterB, What is the potential problem with this configuration?

a. ECMP cannot implemented
b. Route feedback
c. Suboptimal behavior
d. AOS-CX does not support more than one redistribution process in the same switch



Knowledge Check

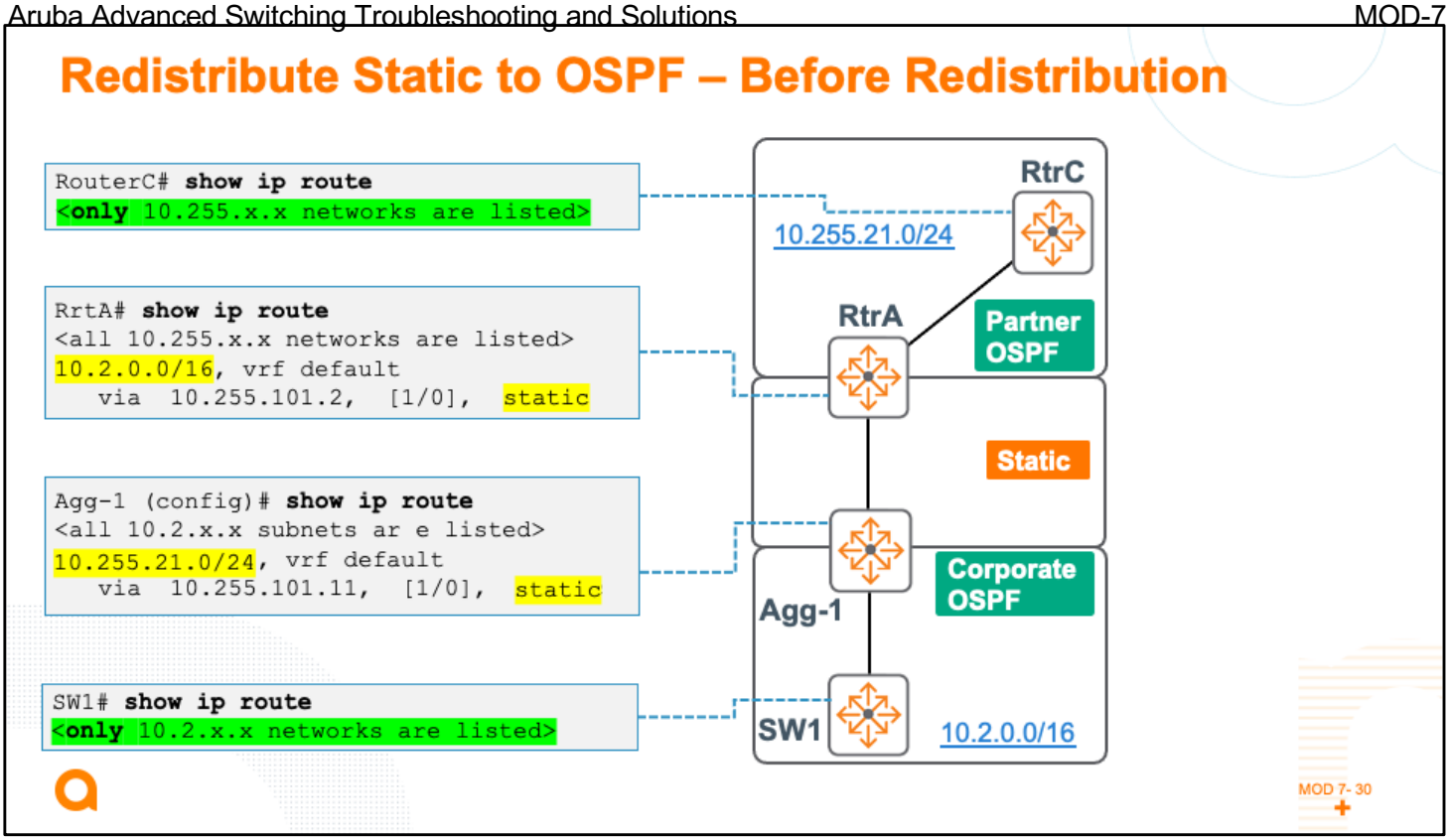
Correct answer:
B,C


Lab Activity
Lab 07

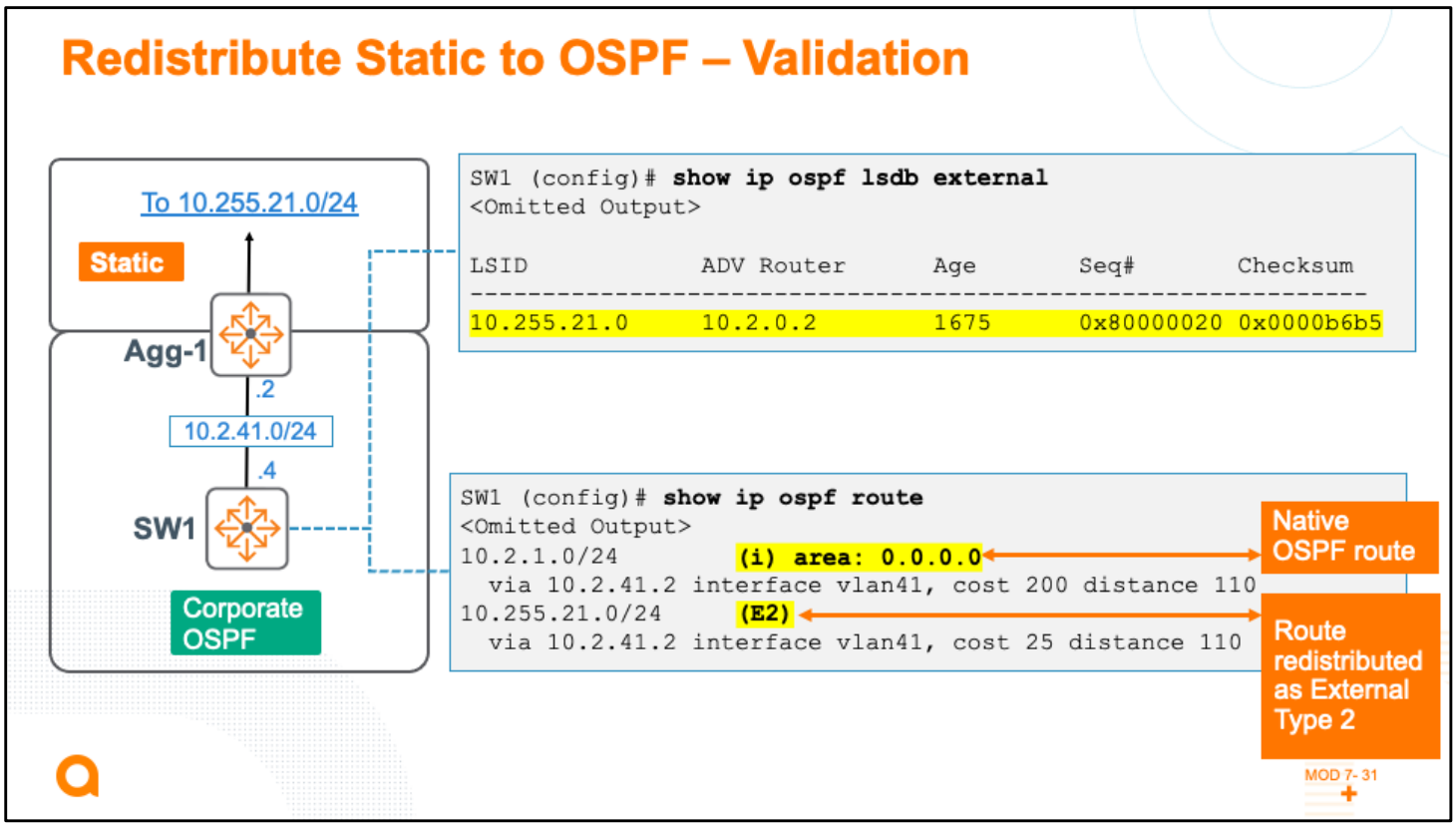
aruba
a Hewlett Packard Enterprise company

Additional Content

Module 07

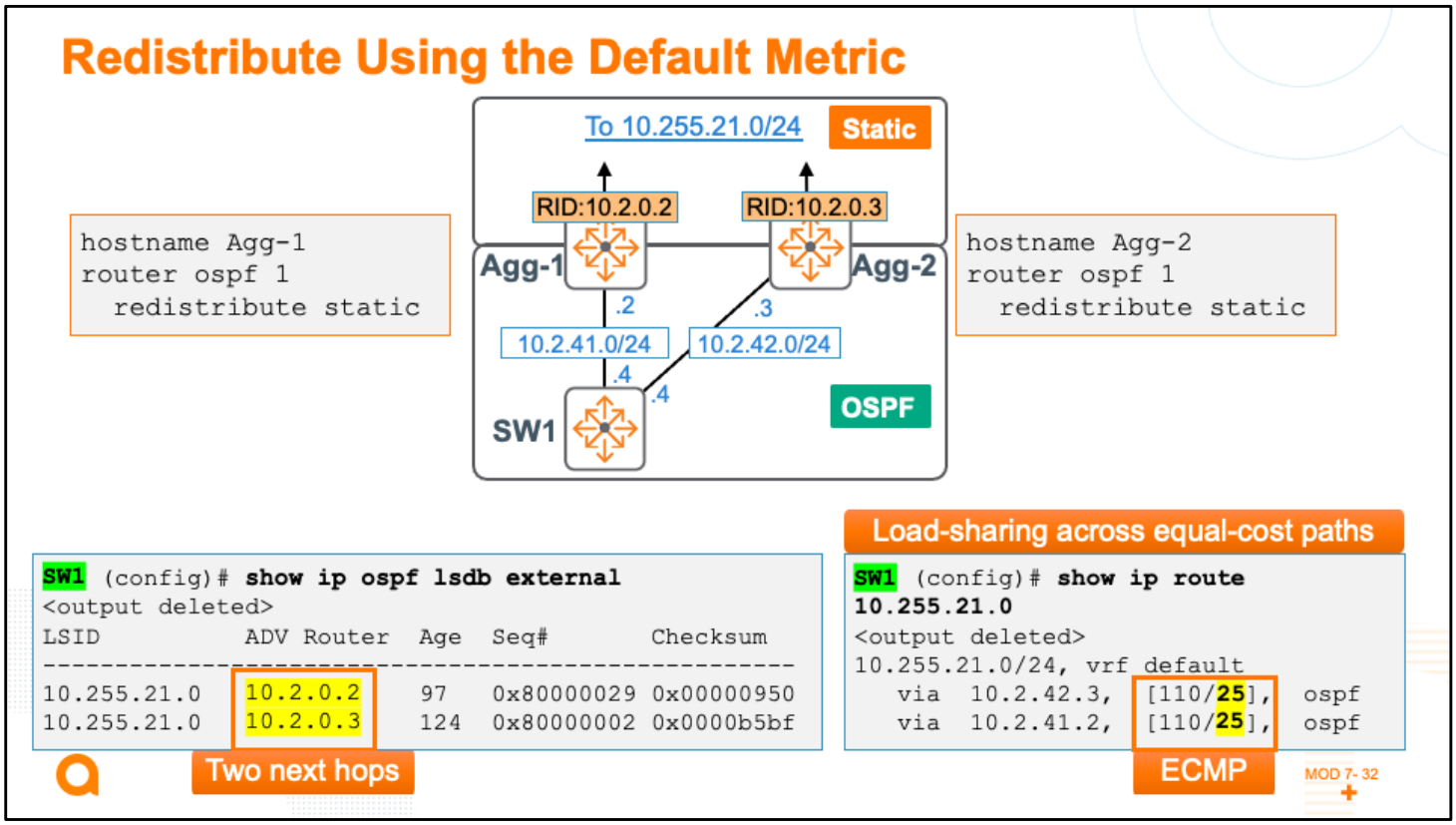


The figures shows the expected results of the previous configuration. The edge router tables have their respective subnets, learned via OSPF. Plus, they have the static routes, as you just configured. The internal routers only have their respective internal subnets. Of course, a ping attempt from the host device fails, due to the lack of redistribution.



You can also use the **show ip ospf lsdb external** command, which is more tightly focused on external (redistributed) routes, as shown in the figure. You can see these LSDB entries show up as OSPF's best-path selections with **show ip ospf route**. Notice that native OSPF routes are flagged as "(i)" for internal. The redistributed routes have "(E2)" for External Type 2 routes – the default type for OSPF redistributed routes.

Note. Remember that the output displayed in the **show ip ospf route** command is the OSPF Routing table is not the FIB (routing table).



In this scenario, we add another edge router to your corporate network, and revert to redistributing default E2 routes, with the default static route redistribution metric of 25. Both Agg-1 and Agg-2 have the appropriate static routes to the 10.255.2x.0/24 subnets.

Thus, router SW1 perceives two equal-cost paths to the destination. Both paths appear in the route table, and so SW1 load-shares among the two paths.

This is a great solution if you have no knowledge or concern about the routes external to your AS. In your estimation, it doesn't matter whether your routers use Agg-1 or Agg-2 to reach the destinations.

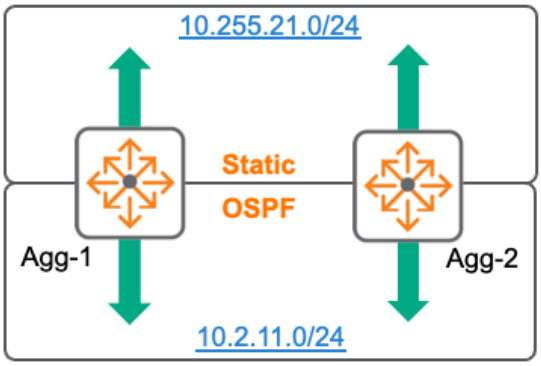
Compare Tuning Techniques

Scenario 1
Agg-1 path to targets is better than Agg-2

- **Tune the default metric with Type E2**
- This ensures that ALL internal routers will prefer one edge router over the other
- Agg-2 is only used if Agg-1 fails

Scenario 2
Agg-1 & Agg2 path to targets is the same

- **Redistribute external routes as Type E1**
- This allows each router to use internal path costs to determine the best exit path
- Agg-1/Agg-2 share the load while providing redundancy

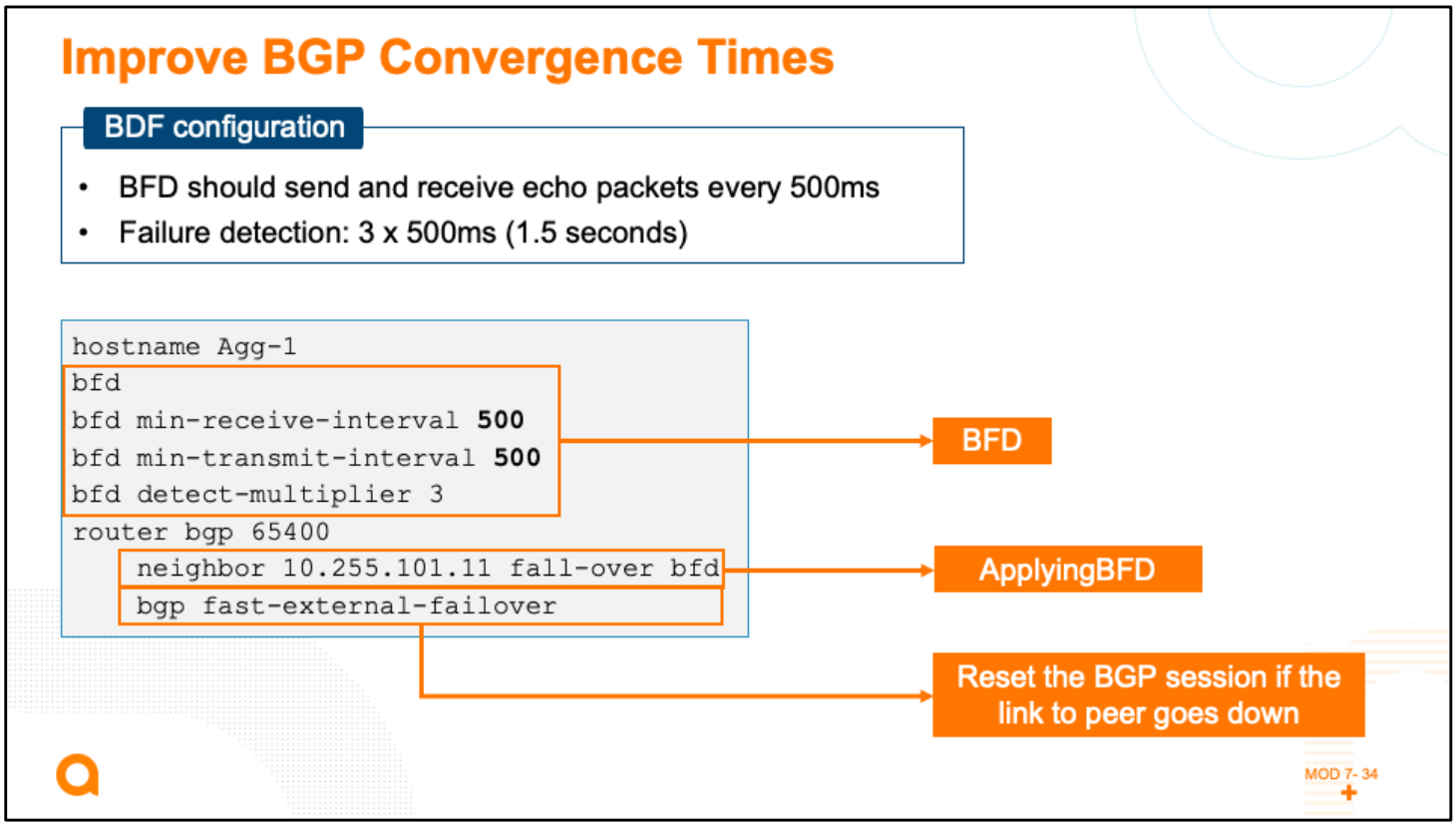


MOD 7- 33

Tuning the default metric using E2 routes is a good solution if you have some knowledge about how Agg-1 and Agg-2 reach the external targets. If Agg-1 his a far better path than Agg-2, you would want all routers to use Agg-1, regardless of internal path costs. Agg-2 is only there to provide redundancy, in case router Agg-1 fails.

However, perhaps Agg-1 and Agg-2 have about the same cost to reach the targets. In this scenario, it might be better to leave the default metric the same and redistribute the routes as type E1. Thus, each internal router uses internal path costs to determine its preferred route to external targets. Agg-1 and Agg-2 will share the load while providing redundancy.

These are great solutions when you want all external routes to be treated the same. However, many situations call for more granular control of individual routes. Let’s explore.



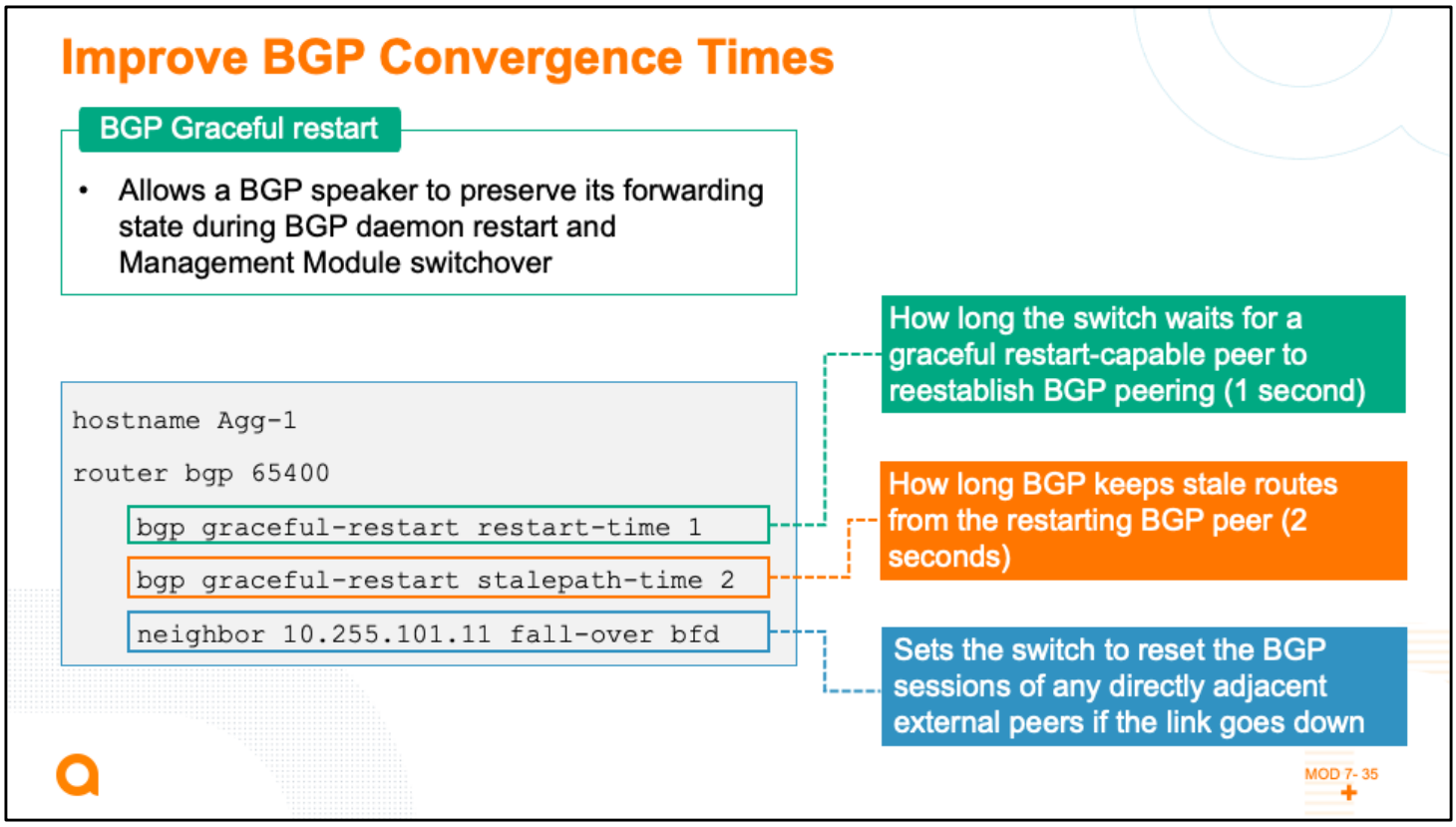
This example shows how you can improve BGP convergence times with Bidirectional Forwarding Detection (BFD) – a lightweight protocol to quickly detect route/link outages with sub-second convergence time.

As shown in the figure, first the BFD feature is enabled globally, and timers are set. This configuration example sets a 500ms timer for sending and receiving BFD echo packets and sets the detection multiplier to 3. This means, 3 x 500ms = 1.5 seconds for detecting an outage. You could reduce the multiplier to improve detection and convergence, but you risk getting false positives that would cause spurious flaps, which could reduce network stability.

The command **neighbor 10.255.101.11 fall-over bfd** means that the BGP process monitors the RIB. If the route to reach a peer is not in the route table, it immediately deactivates the peer session without waiting for the holddown timer. This speeds convergence. The feature is disabled by default.

The command **bgp fast-external-failover** sets the switch to reset the BGP sessions of any directly adjacent external peers if the link used to reach them goes down.

Note. BFD timers only apply if the BGP router can initially establish a BFD connection to a remote BGP router. If the initial BFD connection fails, then BFD is ignored. In other words, the BGP session is independent from the BFD session.



The BGP's graceful restart feature allows a BGP speaker to preserve its forwarding state during BGP daemon restart and Management module switchover. The feature is enabled by default. The command **bgp graceful-restart restart-time 1** sets the graceful restart timer to 1 second. This determines how long the switch waits for a graceful restart-capable peer to reestablish BGP peering. The command **bgp graceful-restart stalepath-time 2** determines how long BGP keeps stale routes from the restarting BGP peer – in this case, 2 seconds.

Rev #20.41

VRF and Route Leaking

Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)



aruba
a Hewlett Packard Enterprise company



After completing this course, you will be able to describe the operation and use cases for the Virtual Routing and Forwarding (VRF) feature.

You will be able to configure and validate VRF with both static routing and OSPF routing.

You will also be able to describe VRF route leaking, and how to configure and validate both static and dynamic route leaking – to help you optimize connectivity.



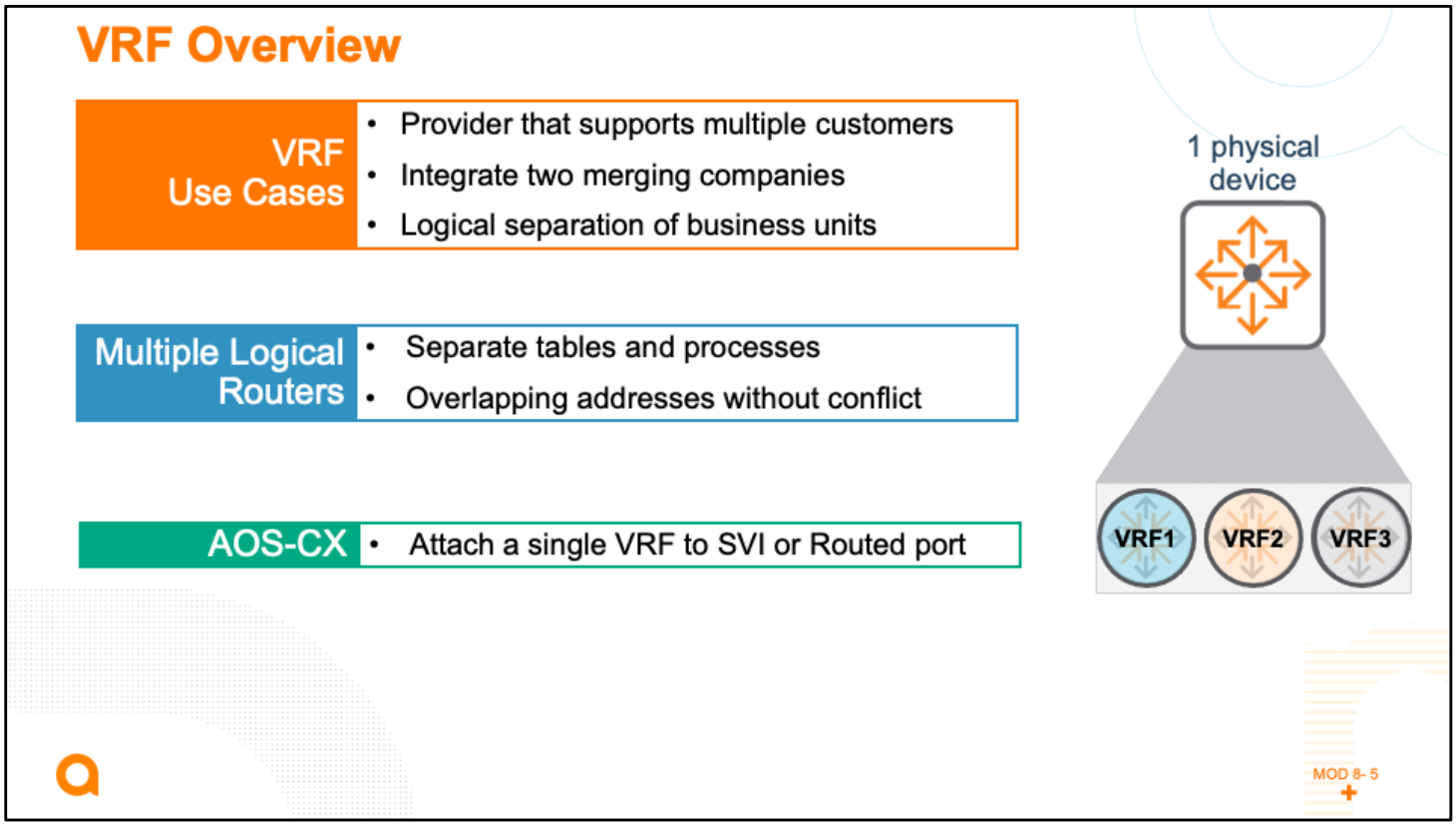
VRF provides problem-solving options for you that would otherwise either be unavailable or prohibitively expensive. Your understanding of VRF operation and configuration can help you to easily solve problems that might not be feasible otherwise.

You begin with an overview of VRF operation and use cases, before diving into VRF configuration – for both static routing and OSPF routing. You already know that a single Layer 2 switch can maintain separate logical VLANs, and a single Access Point (AP) can maintain separate logical Wi-Fi connections or SSIDs. Now you see how you can use one router or set of routers to maintain separate routing processes.

Then you learn that it would be nice if some of the routes from one VRF could be shared with or “leaked to” another VRF, to help you easily accommodate routing requirements.

So you explore VRF route leaking. You learn how to leak static routes from one VRF into another VRF – static VRF route leaking and verification. Then you learn how to do the same with dynamic routing protocols like OSPF or BGP.


aruba
a Hewlett Packard
Enterprise company
VRF
Overview and Configuration



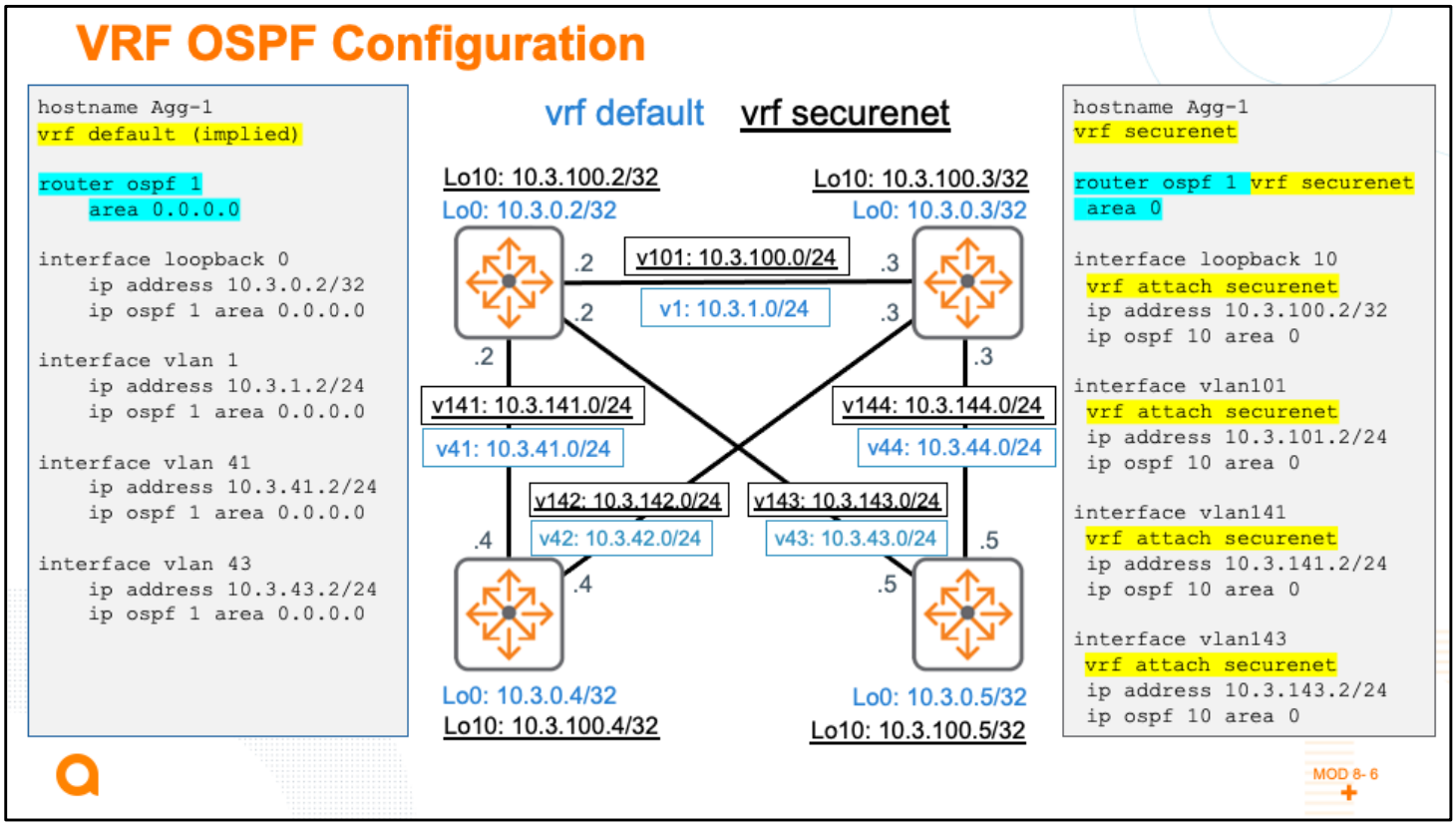
Virtual Routing and Forwarding (VRF) allows a single physical device to host multiple logical routers, with complete L3 isolation. Because the device maintains separate route processes and tables for each VRF, they can have overlapping address space. Network functionality is improved because network paths can be segmented without requiring multiple routers.

Perhaps the scenario shown in the figure represents a single service provider, hosting separate environments for three customers. Customers can use any address space they choose, without fear of conflicting with other customers.

Another use case is when you are merging with another organization. You can physically connect the two organizations without having to worry about addressing conflicts. As you normalize the address space between the two entities, you can eliminate the VRF configuration as appropriate.

Another common use case is to maintain logical separation of business units or divisions. For example, a banking company may keep separation between their retail, lending/mortgages, investments, and internal applications.

In AOS-CX you can attach a single VRF to a SVI or Routed Port.



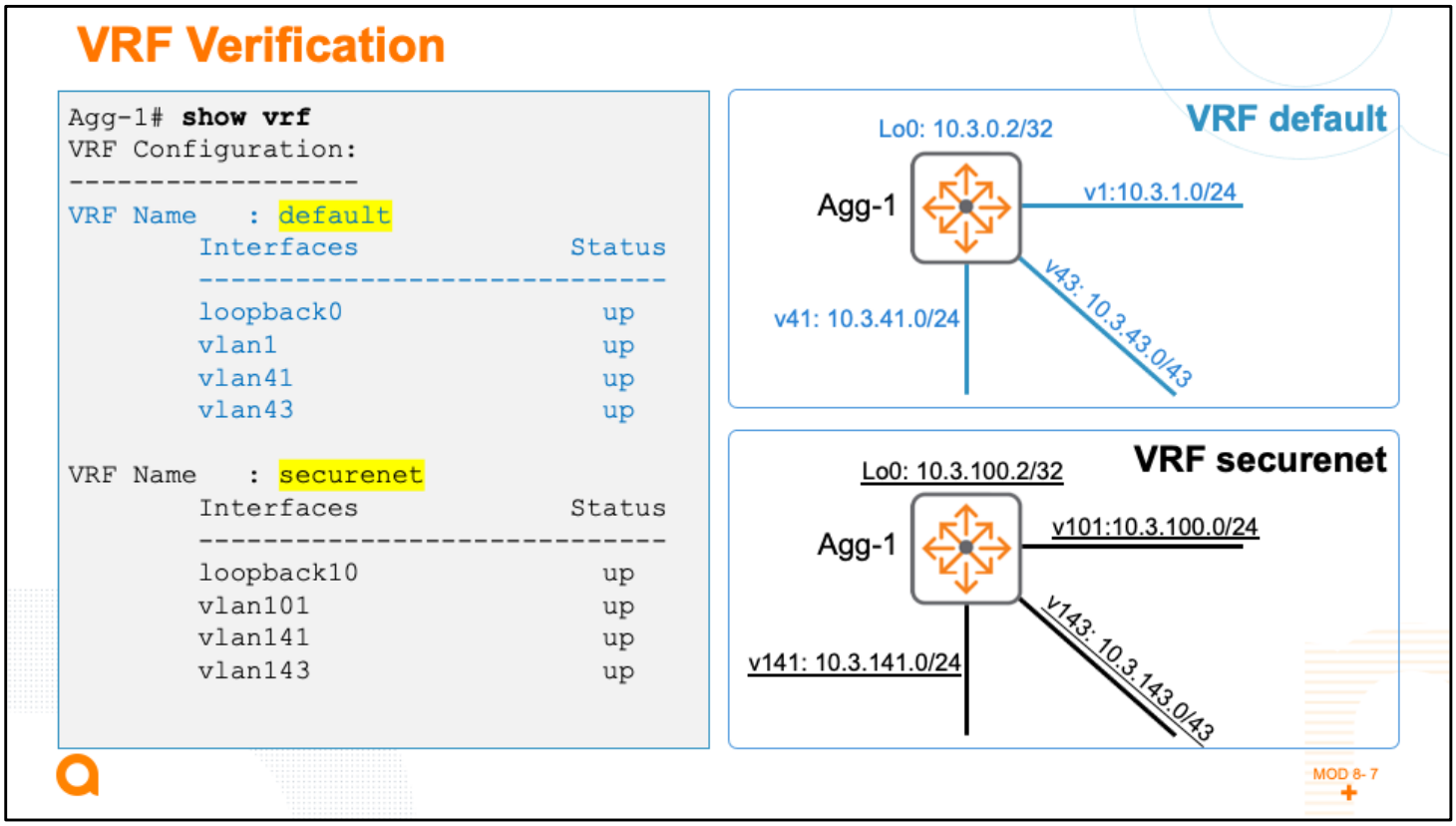
The figure shows an example of one physical infrastructure that supports two separate OSPF networks. The configuration to the right is for VRF “securenet”. This configuration is reflected in the network diagram – just focus on the black, underlined addresses, and ignore the securenet, non-underlined addresses.

Notice that each interface is configured with vrf attach securenet. This is how you tell the route which interfaces belong to which VRF. If you enter the command no vrf attach securenet, the interface is returned to the default VRF, and all interface configuration is removed. This is true anytime you move an interface to a different VRF.

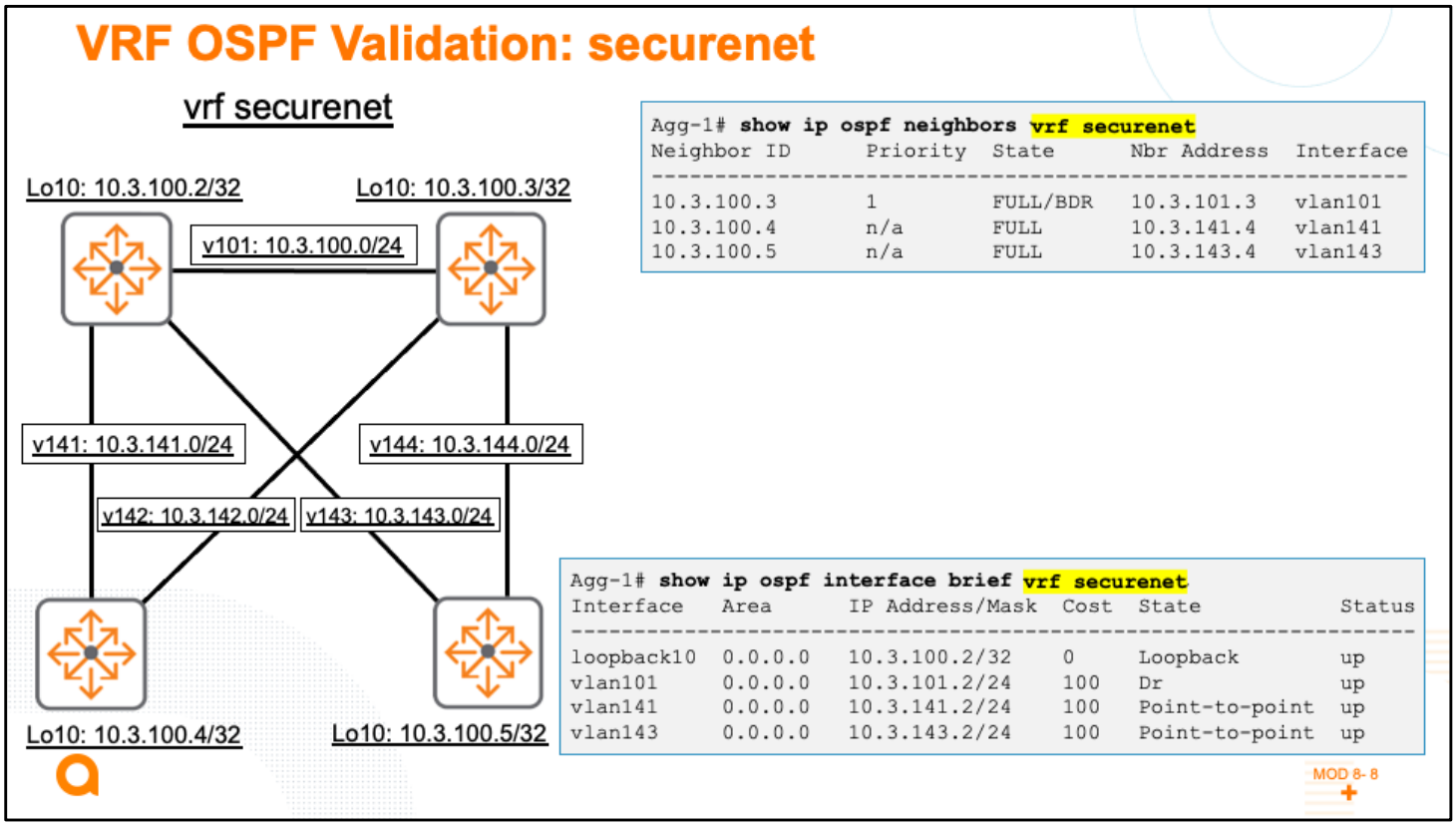
The configuration to the left is for VRF default, also reflected in the network diagram. Just focus on the securenet, non-underlined addresses, and ignore the grey, underlined addresses.

In AOS-CX 10.05 you can reuse the same OSPF process ID in multiple VRFs however a unique OSPF process ID is supported in a VRF. For example:

```
router ospf 1
    router-id 1.1.1.1
    area 0.0.0.0
router ospf 1 vrf securenet
    router-id 2.2.2.2
    area 0.0.0.0
```



The figure shows how to validate which VRFs exist, and which interfaces are associated with each VRF. To look at each VRF separately, you could use show vrf default and show vrf securenet.



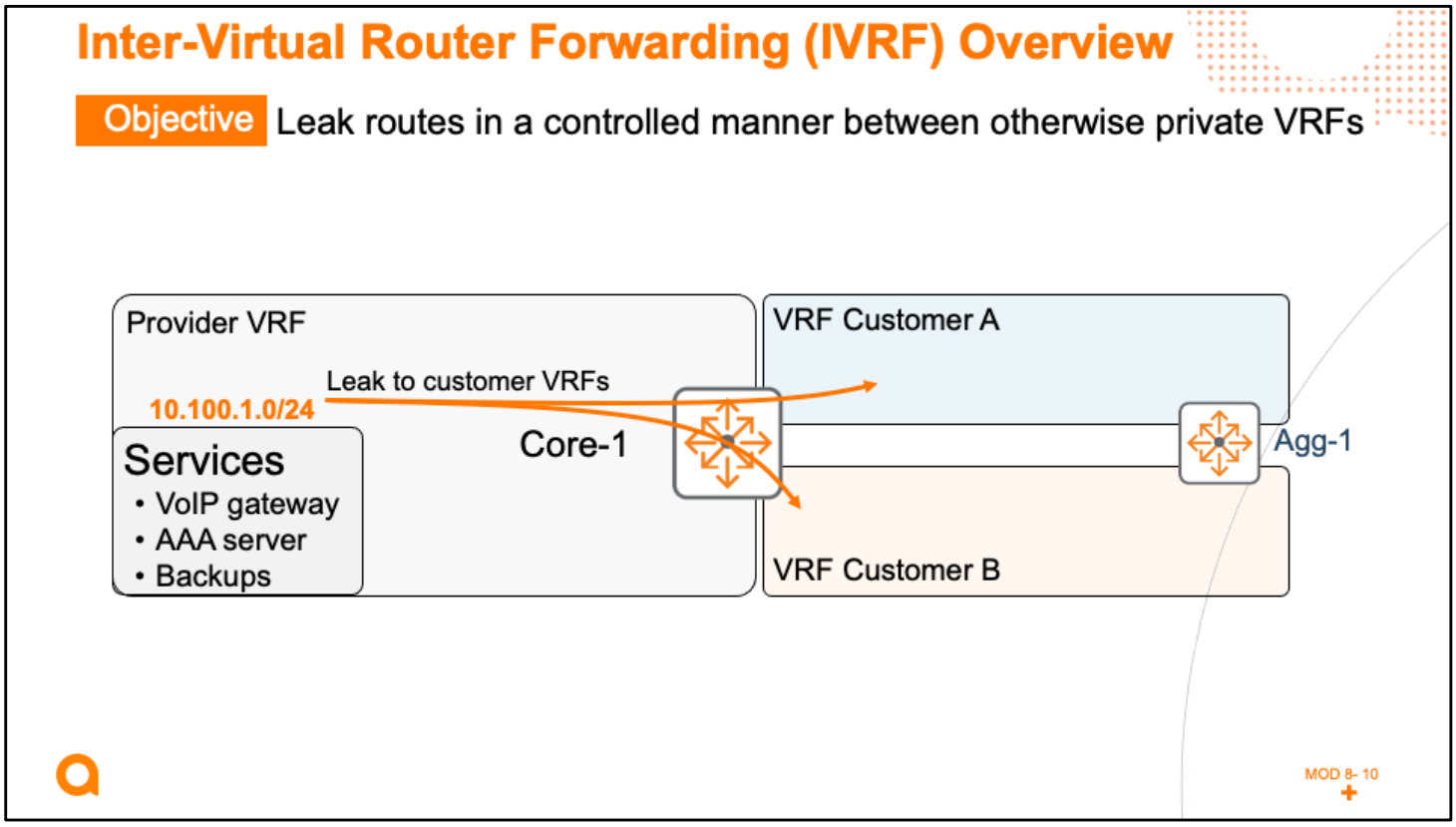
The figure shows how to verify OSPF for your VRF named securenet. The syntax is identical to what you already know – just add “vrf securenet” at the end. Although the figure only shows the commands show ip ospf interface brief and show ip ospf neighbor, you can of course use any other OSPF command to validate this VRF. You might also want to verify OSPF on the other routes.

You could see all VRFs in a single output by using the all-vrfs parameter. For example, show ip ospf interface brief all-vrfs.

aruba
a Hewlett Packard Enterprise company

VRF Route Leaking

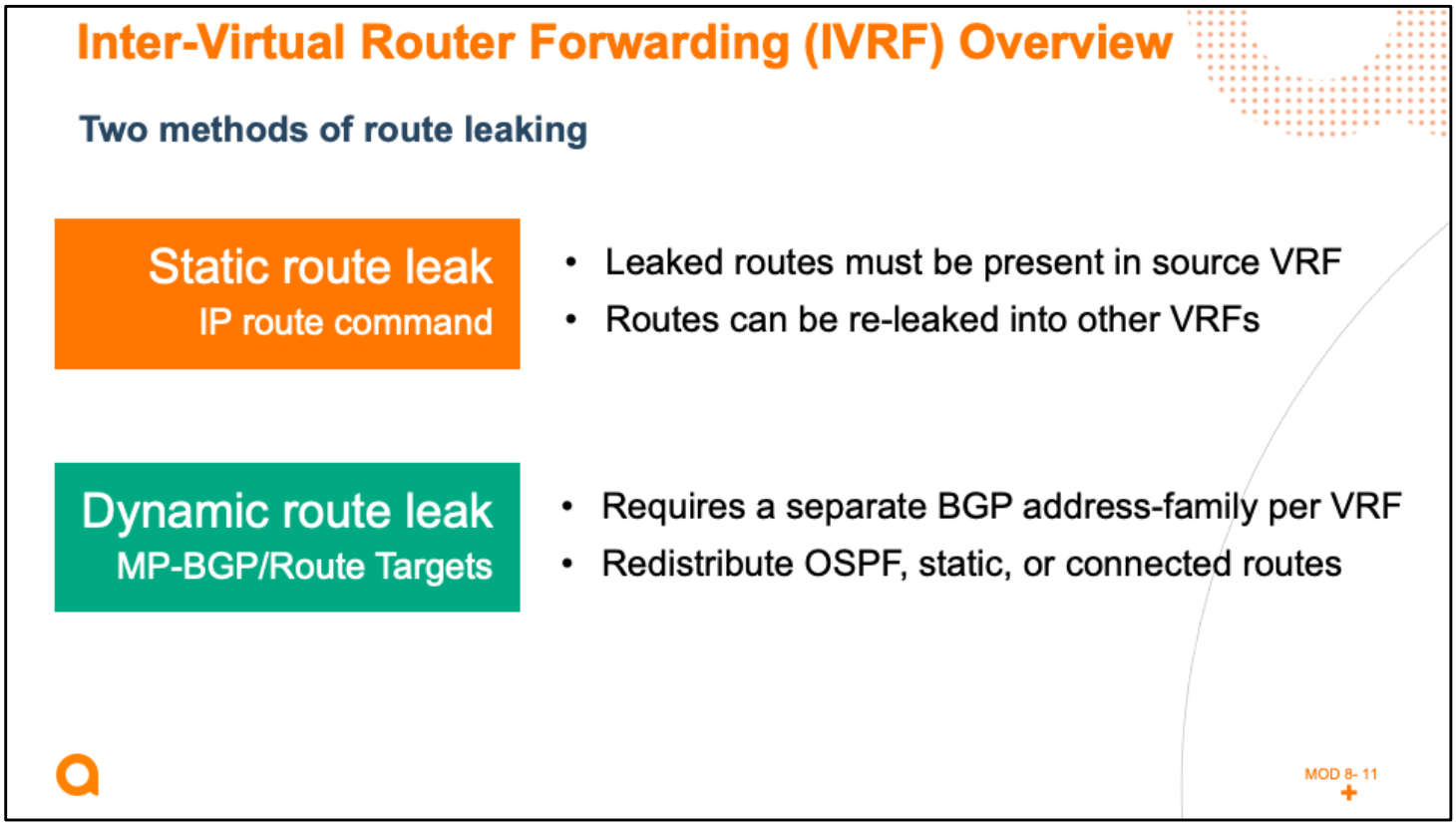
Configuration and Troubleshooting



Inter-Virtual Router Forwarding (IVRF) is also known as route leaking, because it allows you to "leak" routes in a controlled manner between otherwise private VRFs.

Consider a multitenant use case. You know that VRF allows customers A and B to share the same infrastructure while enjoying logically separate, private networks, including separation from the host provider's VRF. However, suppose the provider would like to make optional services available to their clients – a Voice-over IP (VoIP) gateway, AAA services, or a backup service.

These services exist on provider VRF subnet 10.100.1.0/24, which you configure to leak into the customer VRFs that pay for those additional services. Mechanisms in the services themselves maintain customer isolation.



There are two methods to leak routes from one VRF to another.

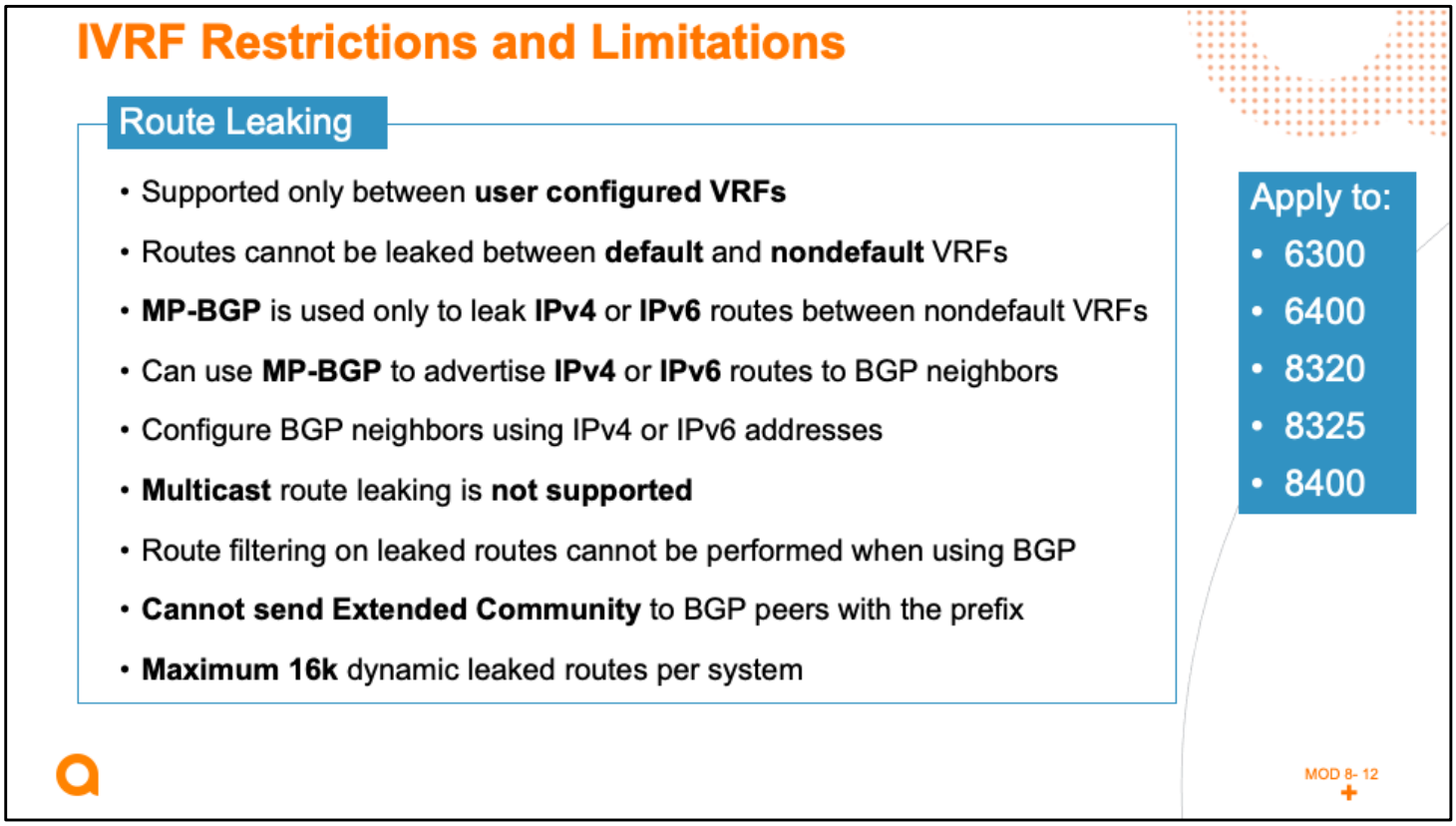
1. Static VRF route leak

You use an IP route configuration command to do static VRF route leaking between two VRFs. Understand the following:

- The routes being leaked to other VRFs must be present in the source VRF.
- When a route leak has been accomplished and is therefore present in the source VRF, that route can then be leaked to other VRFs.

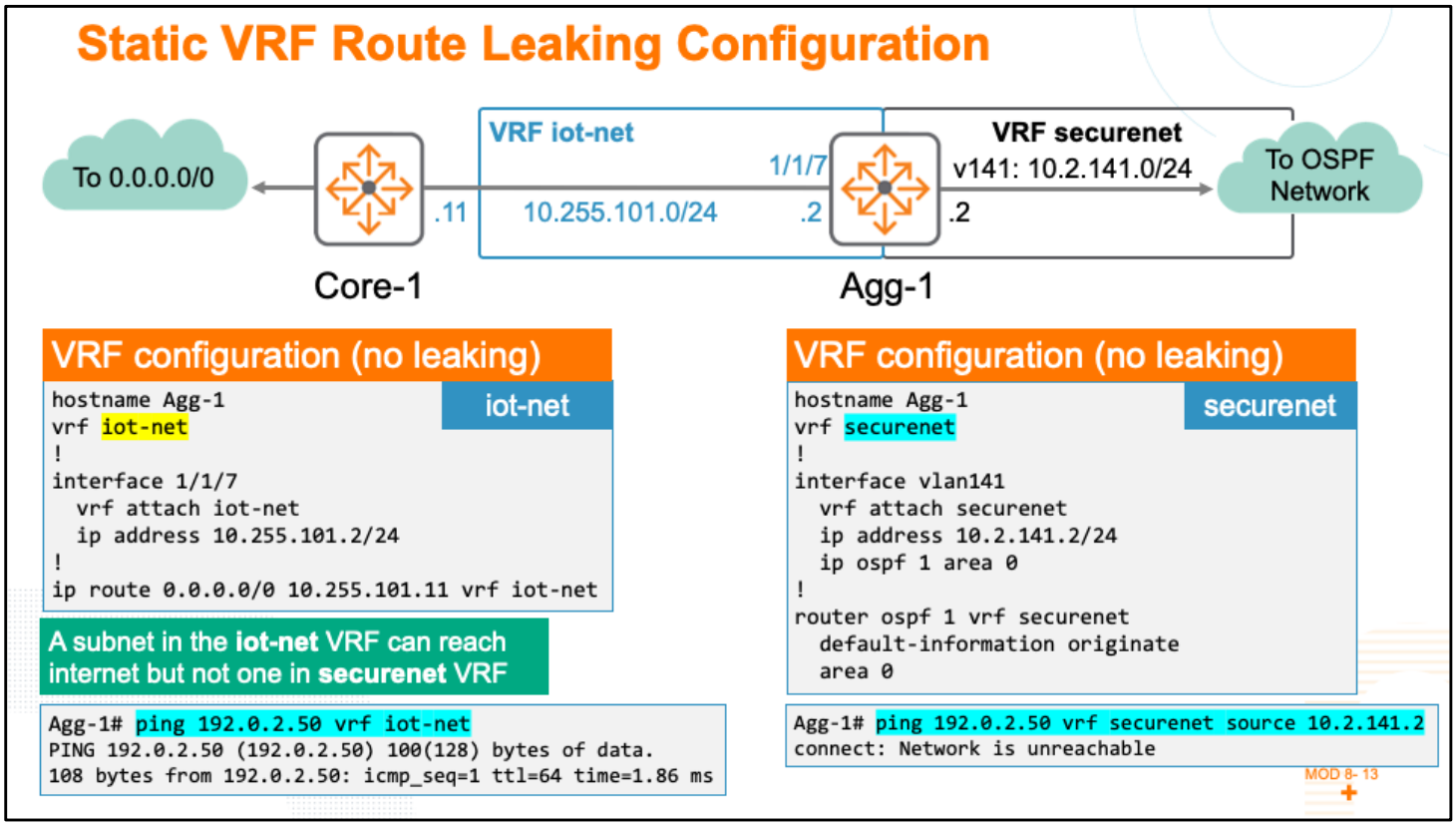
2. Dynamic VRF route leak

You use Multi-Protocol BGP (MP-BGP) to leak routes between dynamic VRFs, which is configured using export and import route-targets. To configure MP-BGP, a separate address-family must be configured within BGP for each VRF. This is used to redistributes routes within that VRF for the routes to be leaked. You can redistribute OSPF, static, or connected routes.



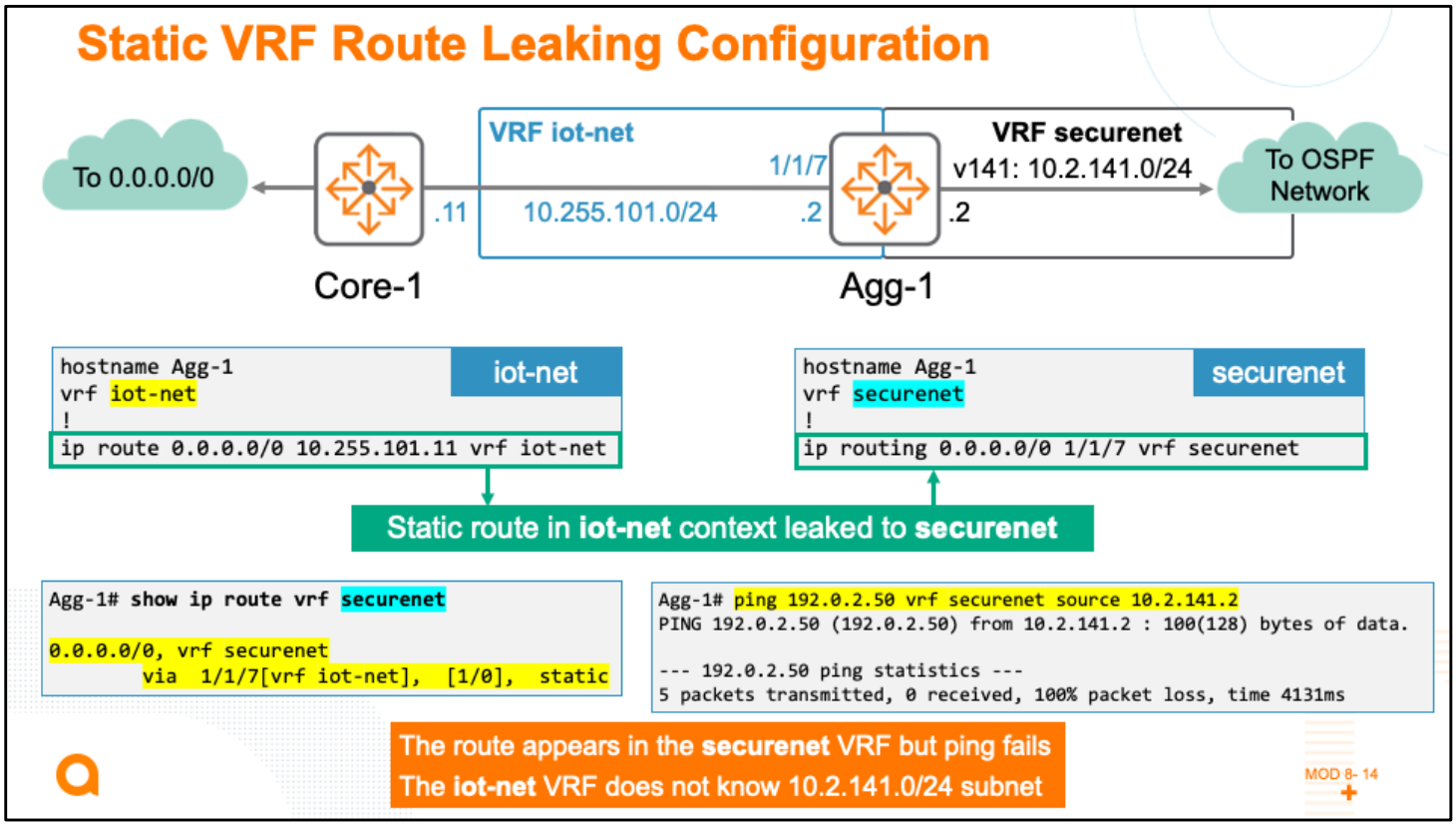
The ArubaOS-CX 10.05 IP Routing Guide document specifies important restrictions and limitations related to route leaking (IVRF).

These limitations apply for 6300, 6400, 8320, 8325 and 8400 switch series.



The figure shows that Agg-1 has a static route to 0.0.0.0/0 via 10.255.101.11. This next hop address connects to interface 1/1/7, which is attached to the iot-net VRF. This means that VRF securenet cannot use this interface directly. The bottom of the figure shows that ping to public IP 192.0.2.50 is successful from the iot-net VRF

Note. The default-information originate command configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors if it is present in the routing table. The default-information originate always command configures OSPF to advertise the default route (0.0.0.0/0) to its neighbors, regardless of whether it is present in the routing table.



This route leaking is shown in the right-side configuration – ip route 0.0.0.0/0 1/1/7 vrf securenet. This enables switch Agg-1 to reach these destinations. To ensure all other securenet routers also have a gateway of last resort, you configure Agg-1's OSPF process with default-information originate.

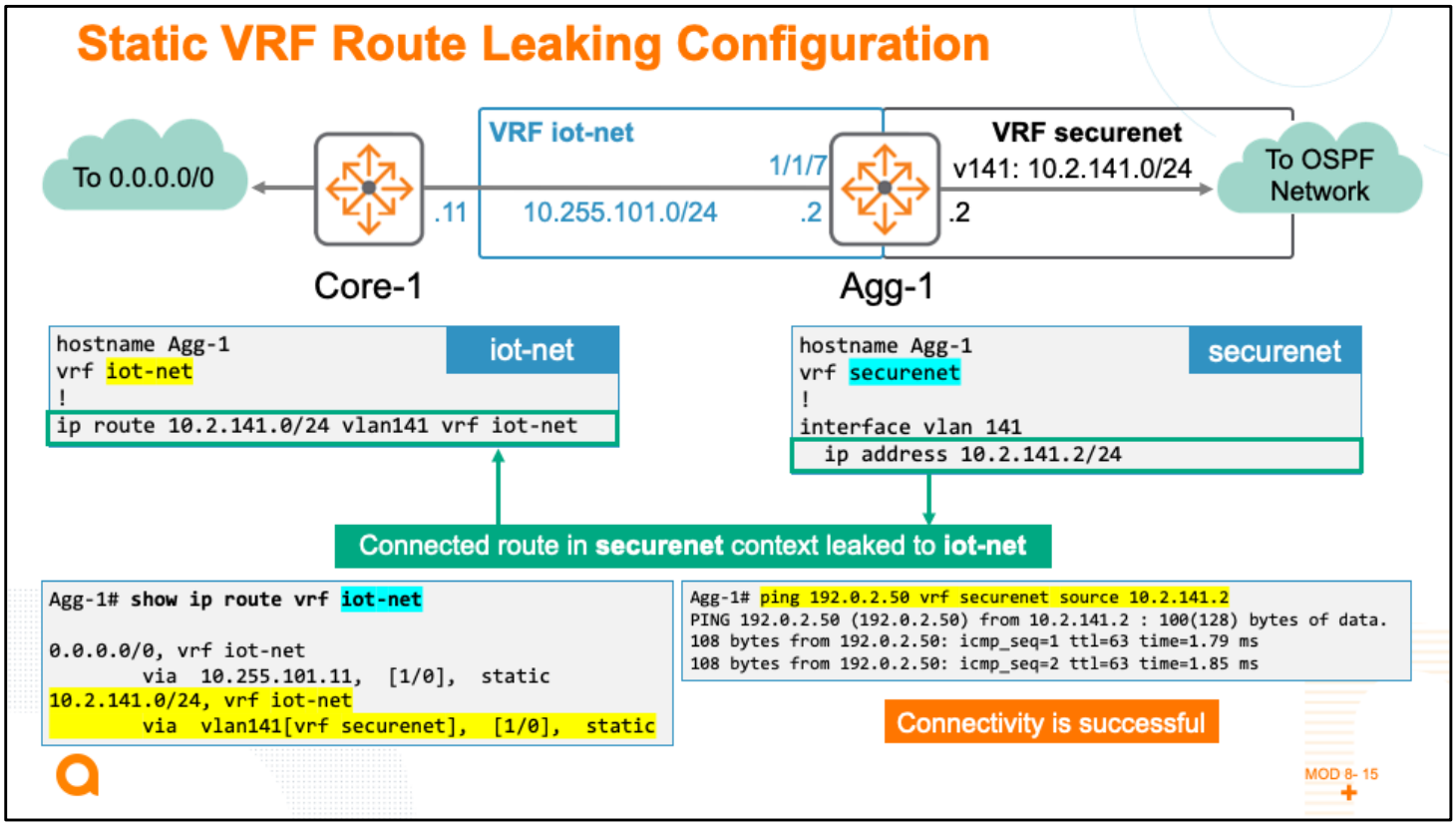
This configuration allows devices in securenet to reach their gateway of last resort. However, the ping connectivity test fails. This is because next hop 10.255.101.11 communicates to Agg-1 using the iot-net VRF. From this VRF, subnet 10.2.141.0/24 does not have a specific entry in the iot-net routing table, so packets are sent back one more time to the default gateway (10.255.101.11), which creates a Layer 3 loop.

Agg-1# show ip route 10.2.141.2 vrf iot-net

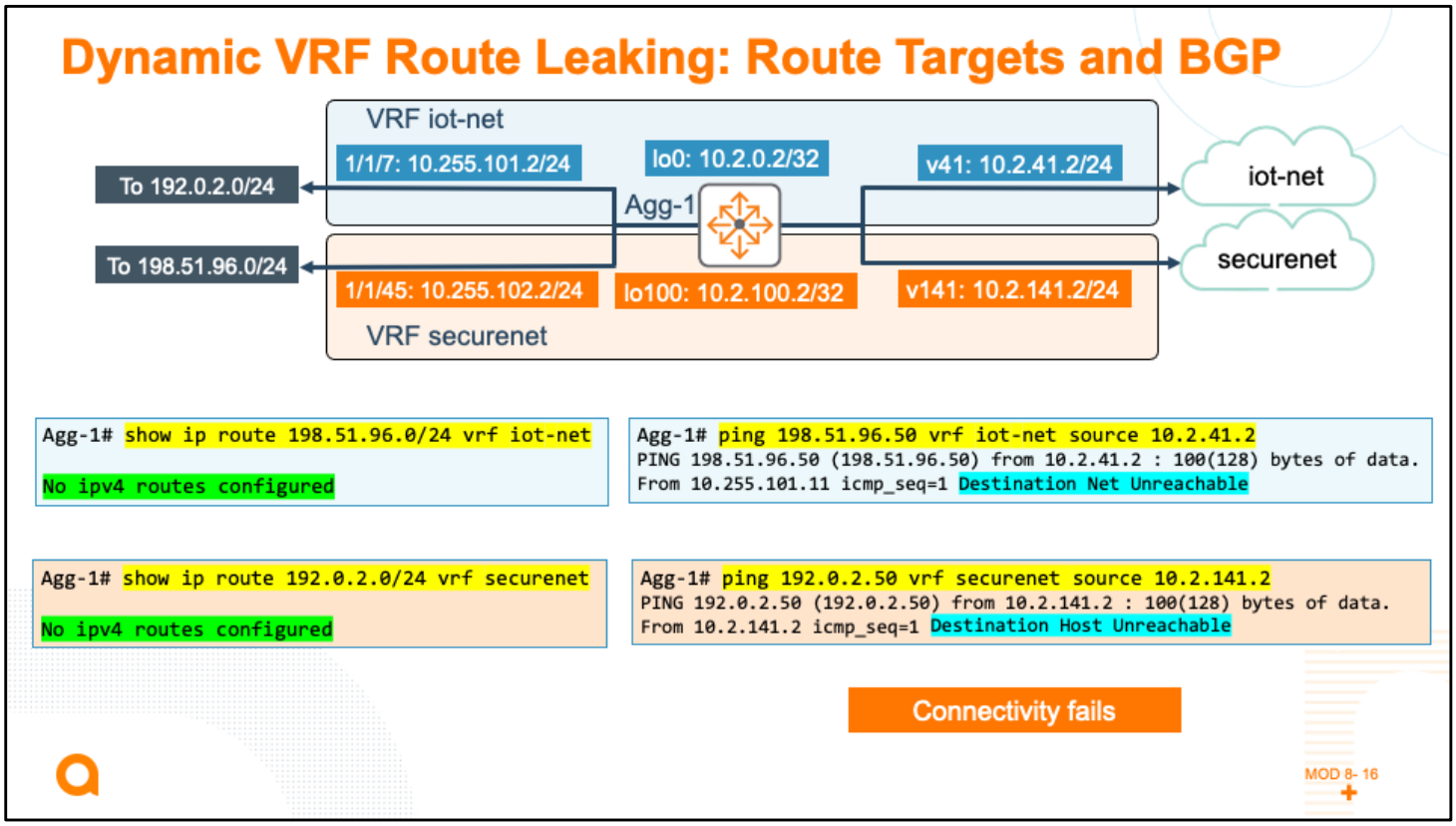
Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

0.0.0.0/0, vrf iot-net
via 10.255.101.11, [1/0], static



To solve the previous problem, VRF iot-net must know how to reach prefix 10.2.141.0/24. To do this, you leak VRF securenet's directly connected route on interface vlan 141 into the iot-net VRF. Use the command ip route 10.2.141.0/24 vlan41 vrf iot-net. This time the ping is successful.



The figure shows how Agg-1 is configured with two VRFs - securenet and iot-net. The output at bottom shows how an internal device in securenet can successfully communicate with the prefix 192.0.2.0/24. However, it cannot communicate with 198.51.96.0/24. In the same way, an internal device in iot-net can communicate with 198.51.96.0/24 but not with 192.0.2.0/24.

You use route leaking to modify this behavior, so both external subnets (198.51.96.0/24 and 192.0.2.0/24) are available for both internal networks (securenet and iot-net).

You can examine the configuration prior route leaking below:

For VRF iot-net

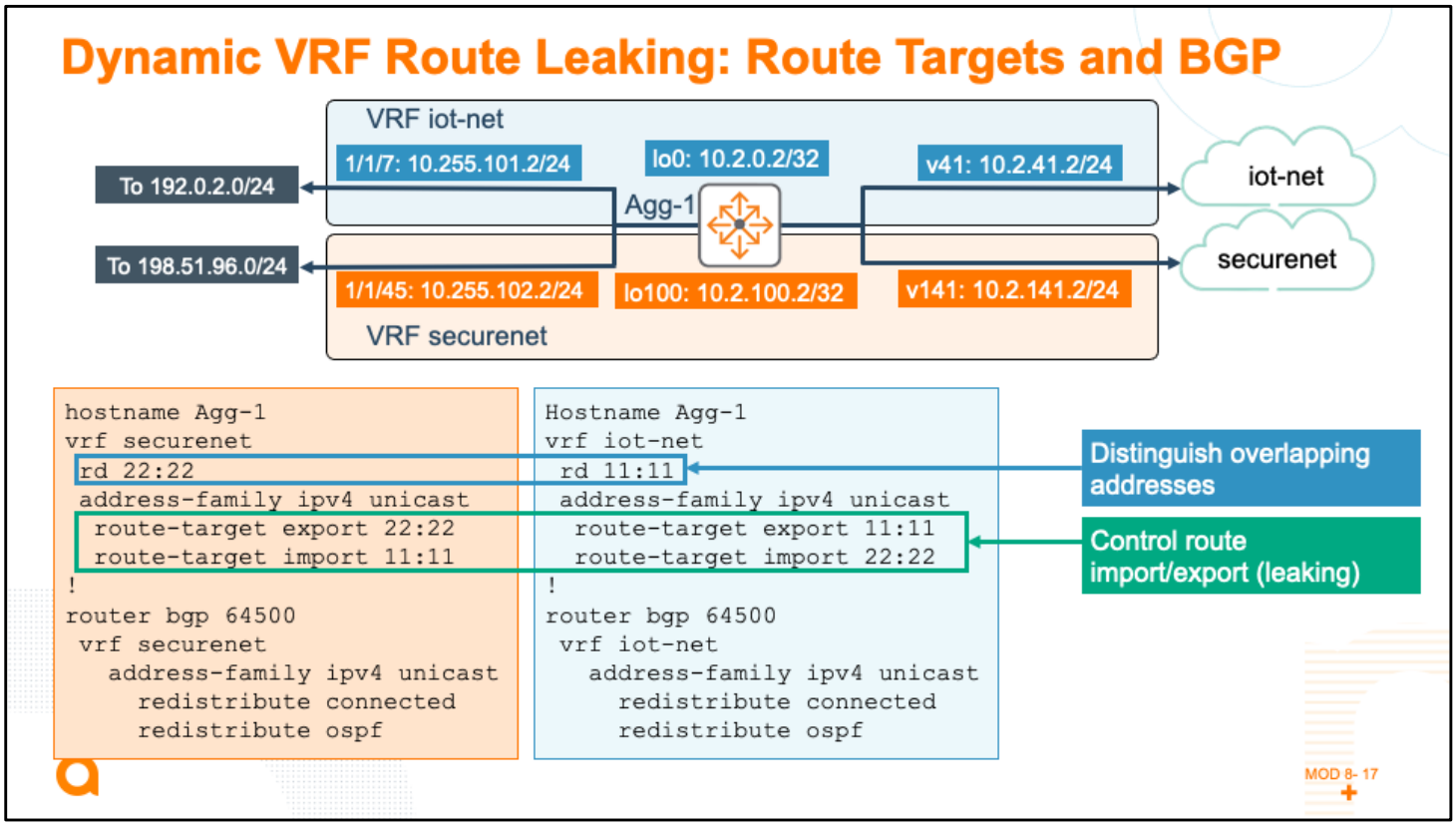
```
hostname Agg-1
vrf iot-net
!
!
interface 1/1/7
    no shutdown
    vrf attach iot-net
```

```
    ip address 10.255.101.2/24
interface loopback 1
    vrf attach iot-net
    ip address 10.2.0.2/32
    ip ospf 1 area 0.0.0.0
interface vlan 41
    vrf attach iot-net
    ip address 10.2.41.2/24
    ip ospf 1 area 0.0.0.0
ip route 0.0.0.0/0 10.255.101.11 vrf iot-net
!
router ospf 1 vrf iot-net
    router-id 10.2.0.2
    default-information originate
    area 0.0.0.0
router bgp 64500
!
    vrf iot-net
        neighbor 10.255.101.11 remote-as 64511
        address-family ipv4 unicast
            neighbor 10.255.101.11 activate
            redistribute connected
            redistribute ospf
        exit-address-family
```

For VRF securenet:

```
hostname Agg-1
vrf securenet
!
!
interface 1/1/45
    no shutdown
```

```
    vrf attach securenet
    ip address 10.255.102.2/24
interface loopback 101
    vrf attach securenet
    ip address 10.2.101.2/32
    ip ospf 1 area 0.0.0.0
interface vlan 141
    vrf attach securenet
    ip address 10.2.141.2/24
    ip ospf 1 area 0.0.0.0
    ip ospf network point-to-point
ip route 0.0.0.0/0 10.255.102.12 vrf securenet
!
router ospf 1 vrf securenet
    router-id 10.2.0.2
    default-information originate
    area 0.0.0.0
router bgp 64500
!
    vrf securenet
        neighbor 10.255.102.3 remote-as 64512
        address-family ipv4 unicast
            neighbor 10.255.102.3 activate
            redistribute connected
            redistribute ospf
        exit-address-family
```

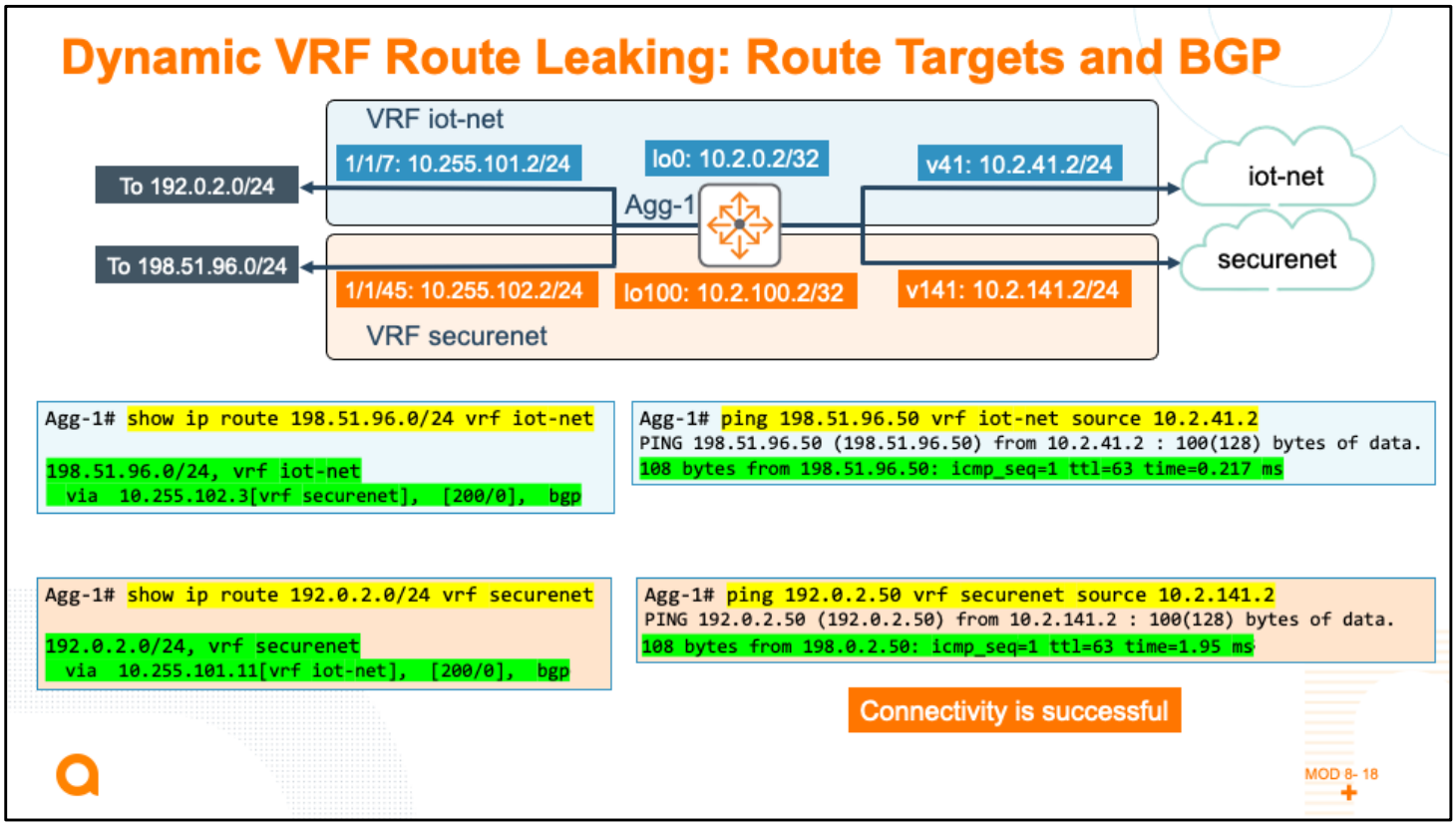


To enable route leaking using BGP, you assign Route Distinguisher (RD) 22:22 to VRF securenet, and RD 11:11 to VRF iot-net. As the name implies, this helps routers to distinguish which routes belong to which VRF - especially helpful when the VRFs have overlapping address space.

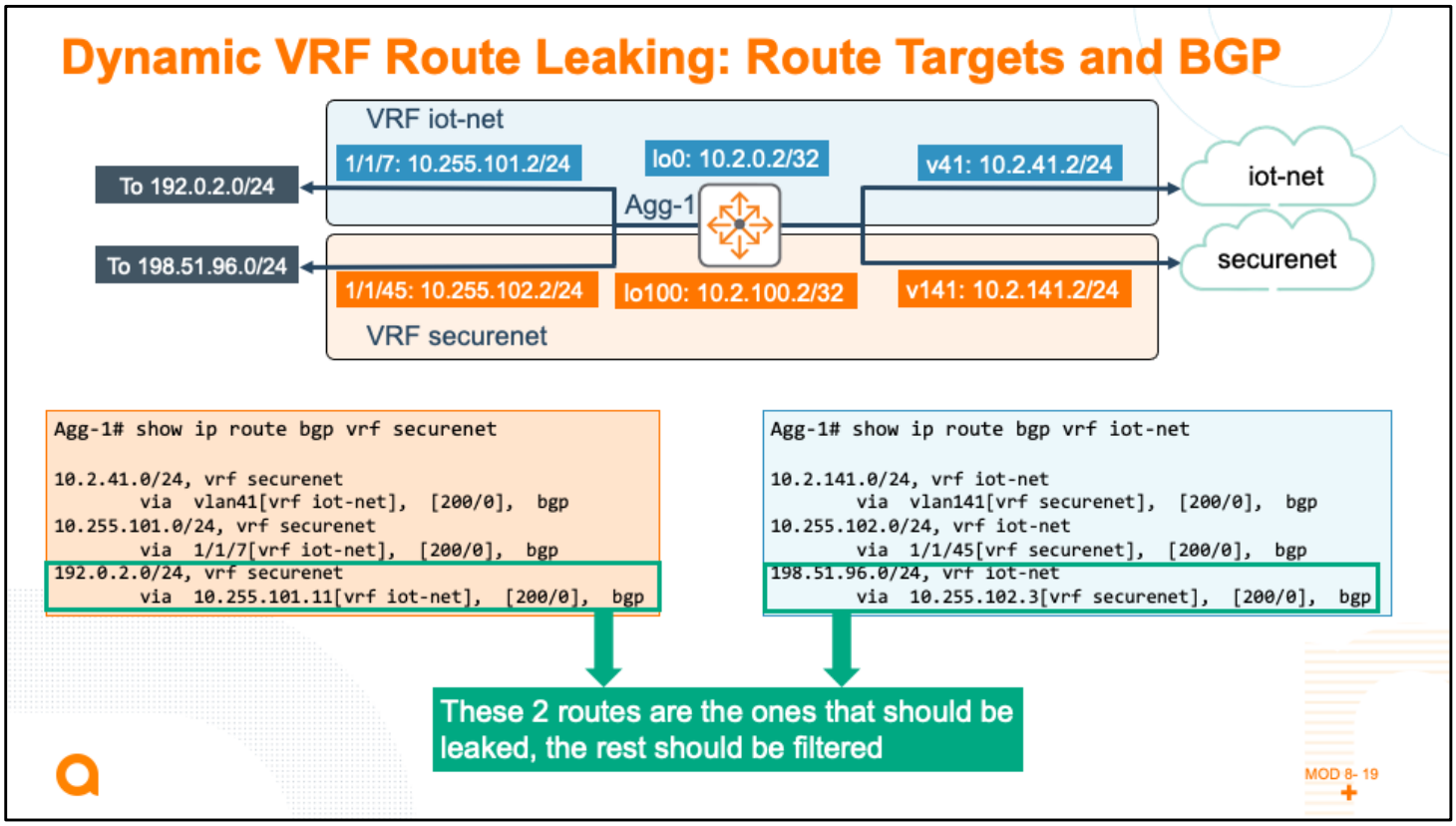
Suppose that both VRFs contain several unique prefixes, and both contain prefix 10.2.200.0/24. Both VRFs advertise their prefixes to peers, tagged with the appropriate RD. Thus, receiving peers can determine which advertisements are for which VRF. Essentially, we humans see VRFs securenet and iot-net. The routers see RDs 22:22 and 11:11.

Each VRF's address family is then configured for which Route Targets (RT) to accept in (import) and which routes to send out (export). In other words, RTs control route leaking.

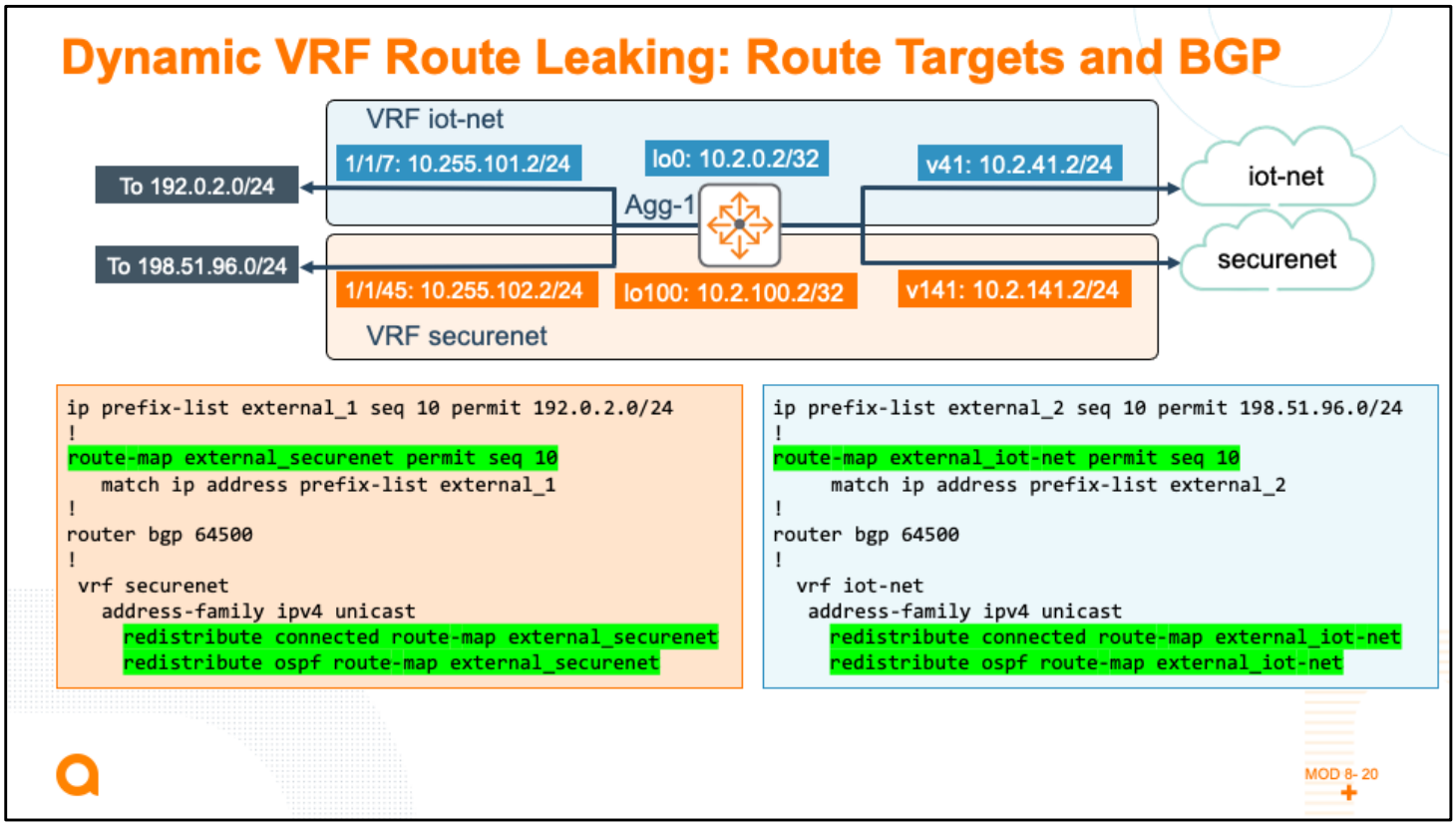
In the example, VRF securenet (22:22) wants to import its own routes into its route table, and it will export its routes to others. It will also import VRF iot-net (11:11) routes. The iot-net VRF does the same.



Now you look at the route tables. This proves that iot-net routes are successfully exported by VRF iot-net and imported by VRF securenet. You also see that securenet routes are imported into VRF iot-net, as expected. Connectivity works now.



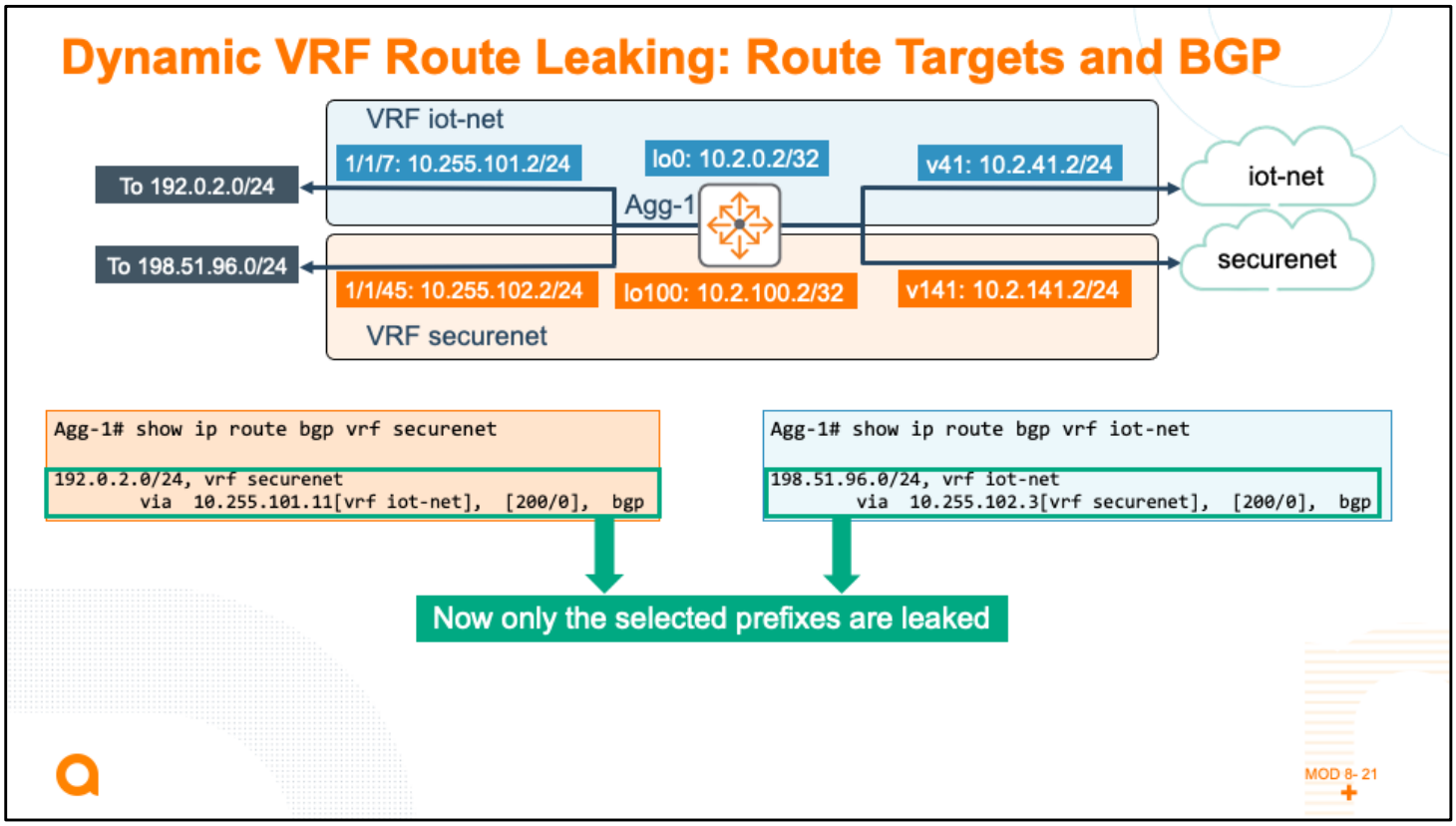
Using the previous method (Route Targets and BGP) route leaking is possible, but it is not optimal. The slide shows the output from the show ip route bgp vrf securenet and show ip route bgp vrf iot-net. Notice that the entire routing table on one VRF has been leaked into the other. This default behavior is not optimal, fortunately for us route-maps can help to filter the information leaked.



To select specific routes to be leaked, first select the prefixes by using a prefix-list. In this example you configure prefix-lists external_1 to match 192.0.2.0/24, and external_2. to match 198.51.96.0/24

Now from the securenet VRF point of view a route map is configured - in this case matching prefix-list external_1, since the 192.0.2.0/24 prefix is to be leaked into the securenet VRF. Next apply the route-map when you redistribute connected routes into BGP and when you redistribute OSPF routes into BGP. A similar process is used for iot-net VRF, which uses prefix-list external_2.

Note. Notice that all prefixes shown are directly connected to Agg-1 and there is no OSPF routes displayed.



Using the previous commands, now you can see that a single prefix has been leaked from one VRF into the other.

Question #1

```
Sw2(config)# ip route 10.1.61.0/24 vlan61 vrf branch
Sw2# show ip route 10.1.61.0/24 vrf branch
Displaying ipv4 routes selected for forwarding

'[x/y]' denotes [distance/metric]

10.1.61.0/24, vrf branch
        via  vlan61[vrf iot],  [1/0],  static
```

An administrator wishes to redistribute routes between VRF branch and VRF iot, and configures a static route for this purpose as shown in the output, Would a ping work between the Sw2 VRF branch and VRF iot at this point?

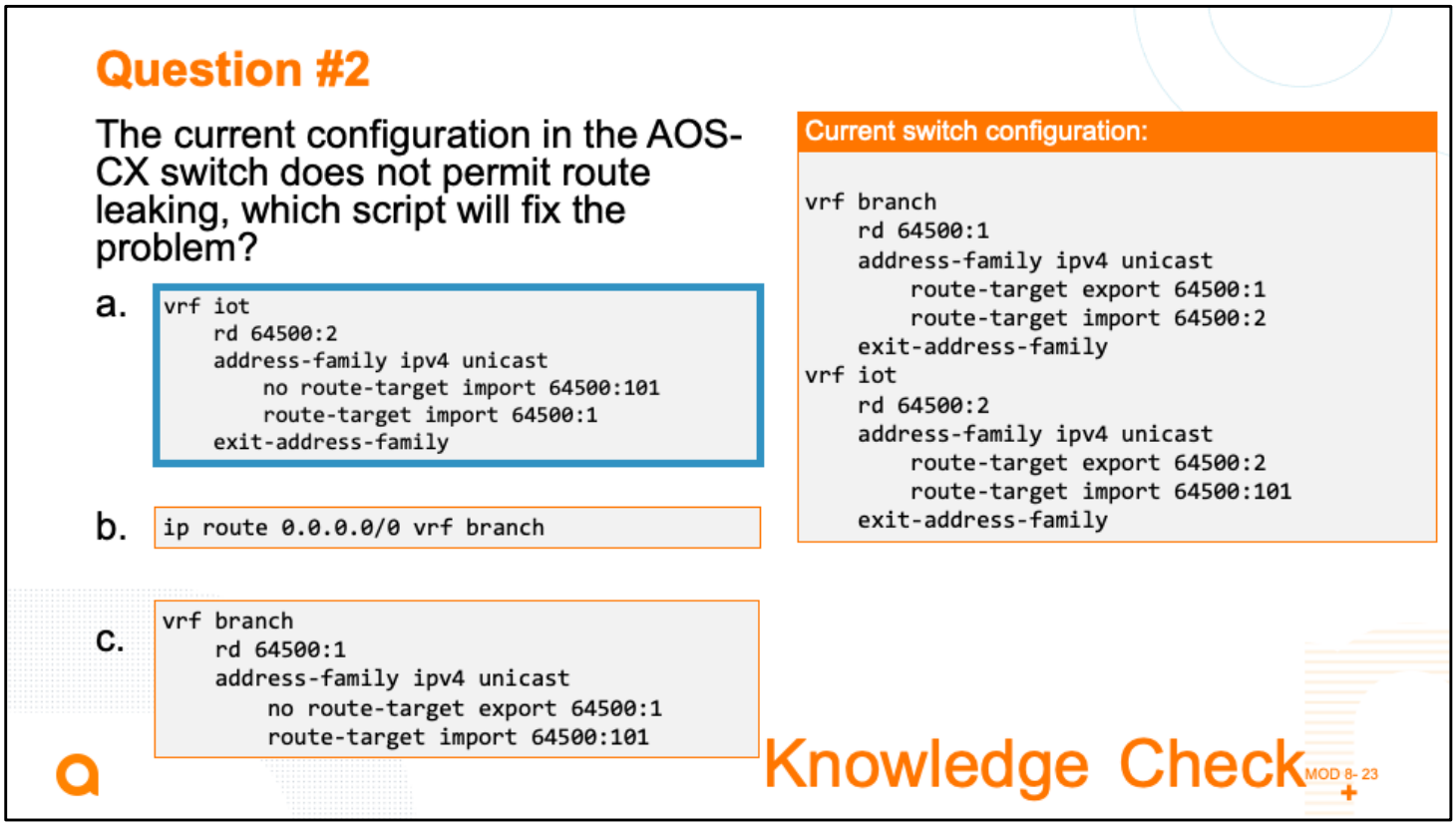
a. No, static VRF route leaking requires bi-directional routes.

b. Yes, AOS-CX only uses a single static route for VRF route leaking

Knowledge Check

MOD 8- 22

Correct answer: A



Correct answer:

A

aruba
a Hewlett Packard Enterprise company

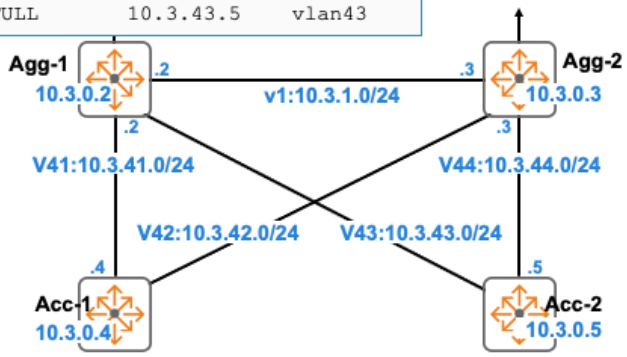
Additional Content

Module 08

VRF OSPF Validation: default

```
Agg-1# show ip ospf int brief
Interface     Area        IP Address/Mask     Cost    State      Status
-------------------------------------------------------------------------
loopback0     0.0.0.0     10.3.0.2/32         0       Loopback         up
vlan1         0.0.0.0     10.3.1.2/24         100     Dr               up
vlan41        0.0.0.0     10.3.41.2/24        100     Point-to-point   up
vlan43        0.0.0.0     10.3.43.2/24        100     Point-to-point   up
```

```
Agg-1# show ip ospf neighbor
Neighbor ID      Priority  State      Nbr Address  Interface
-----------------------------------------------------------
10.3.23.3        1         FULL/BDR   10.3.1.3     vlan1
10.3.0.4         n/a       FULL       10.3.41.4    vlan41
10.3.0.5         n/a       FULL       10.3.43.5    vlan43
```




```
hostname Agg-1
vrf default (implied)

router ospf 1
    area 0.0.0.0

interface loopback 0
    ip address 10.3.0.2/32
    ip ospf 1 area 0.0.0.0

interface vlan 1
    ip address 10.3.1.2/24
    ip ospf 1 area 0.0.0.0

interface vlan 41
    ip address 10.3.41.2/24
    ip ospf 1 area 0.0.0.0

interface vlan 43
    ip address 10.3.43.2/24
    ip ospf 1 area 0.0.0.0
```

MOD 8- 25

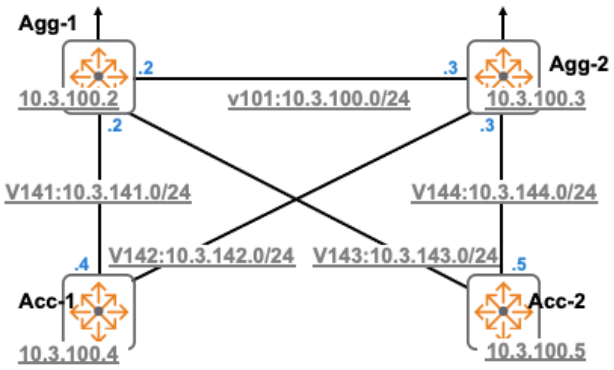
The figure shows the same validation commands for the default VRF, for comparison purposes.

VRF Static Route Configuration

```
Hostname Agg-1
ip route 10.252.0.0/16 10.255.101.11 vrf securenet
```

```
Agg-1#  show ip route static vrf securenet
Displaying ipv4 routes selected for forwarding
'[x/y]' denotes [distance/metric]

10.252.0.0/16, vrf securenet
        via  10.255.101.11,  [1/0],  static
```

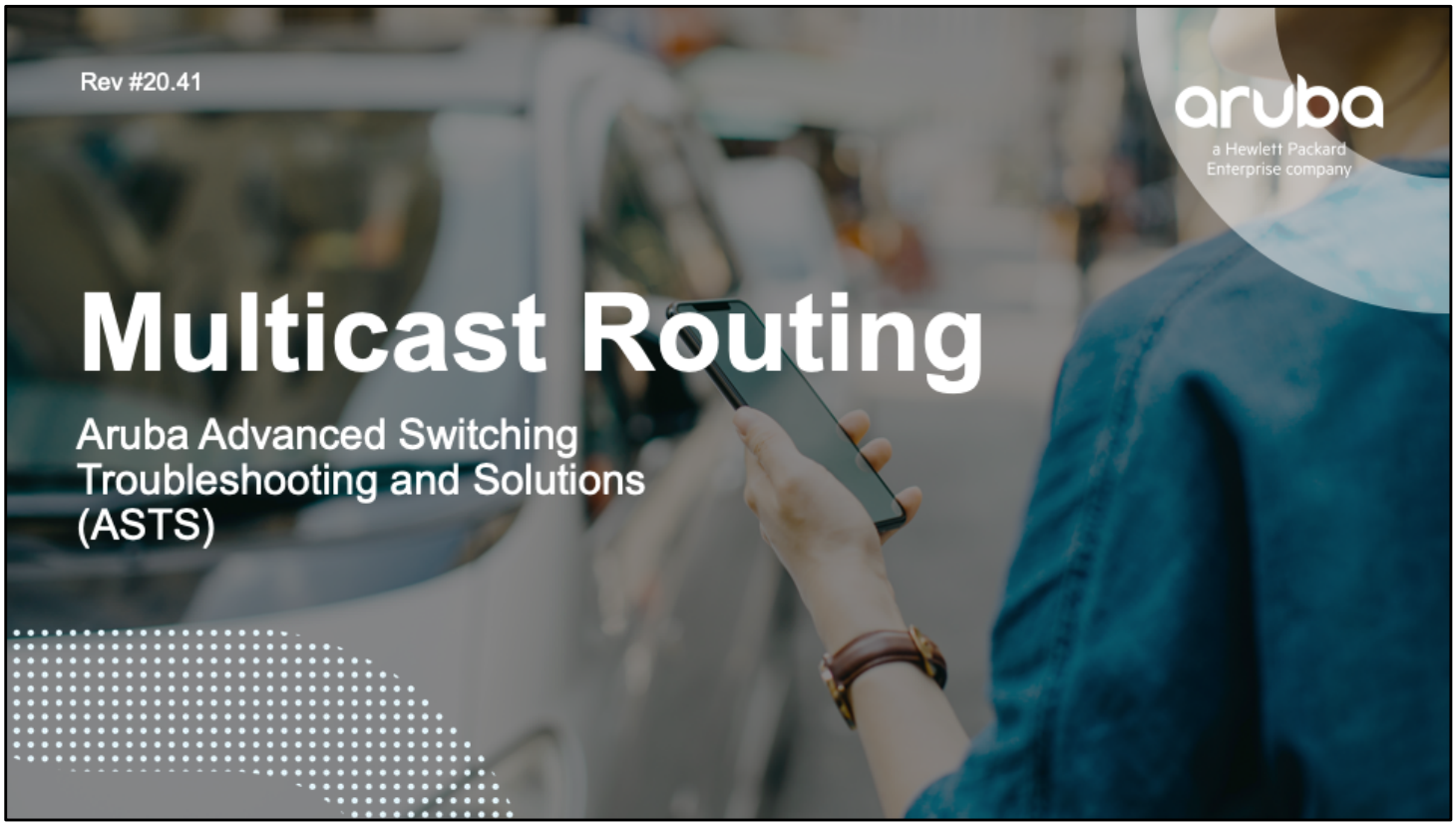


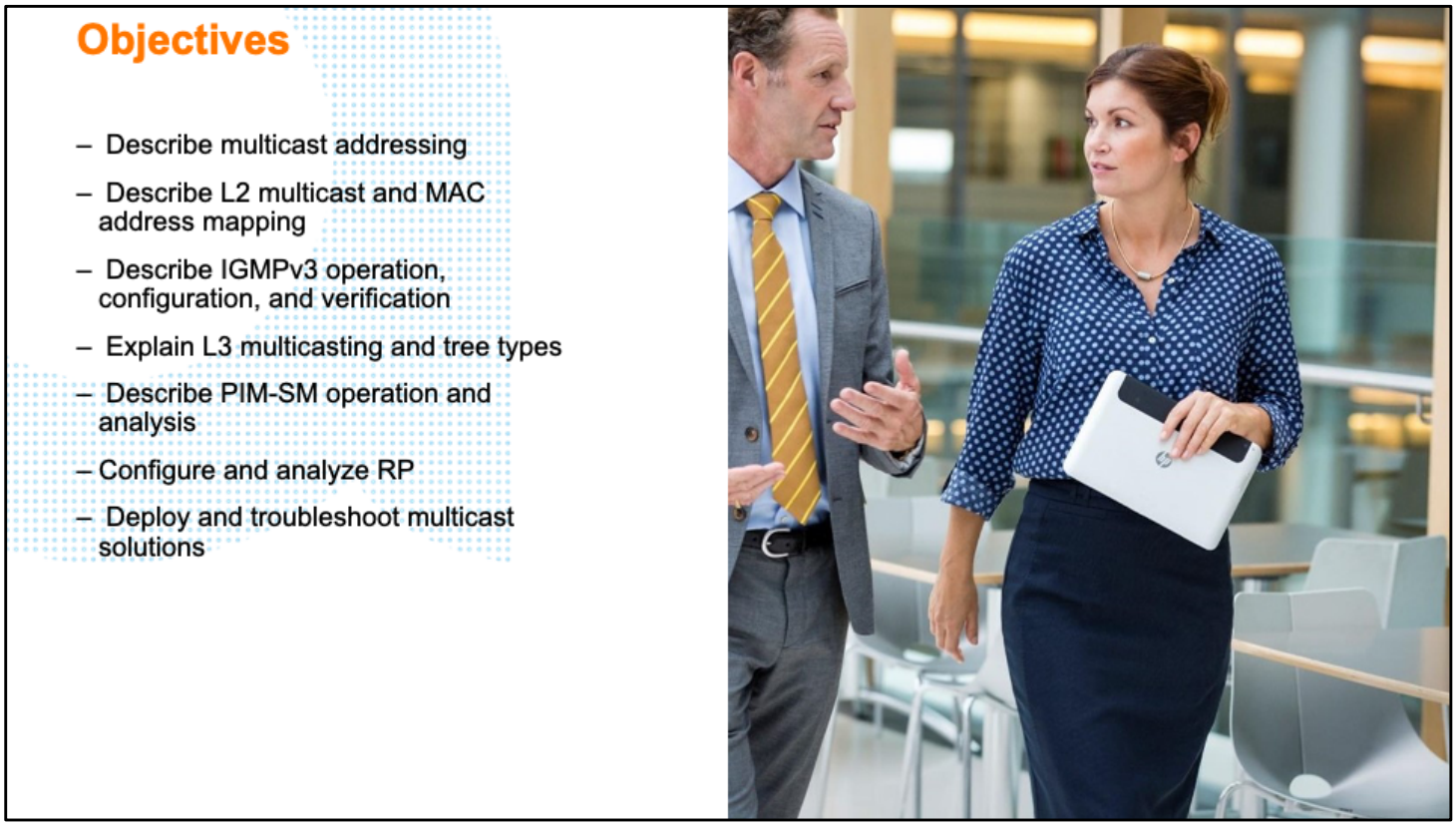
MOD 8- 26

Of course, you can also configure and validate static routes with VRF, as shown in the figure. You may recall that this same route statement has been configured on the Agg-1 router for previous OSPF configurations. To issue the static route shown, you must first make the associated 10.255.101.2/24 interface a part of the securenet VRF. It is therefore no longer a part of the default VRF.

Here's the problem. You need a way to leave that original static route as a part of the default VRF but let this route "leak" into the securenet VRF. Thus, both VRFs can use the same static route to reach external destinations. You will explore this in the next section.


Lab Activity
Lab 08


Rev #20.41
aruba
a Hewlett Packard Enterprise company
Multicast Routing
Aruba Advanced Switching Troubleshooting and Solutions (ASTS)



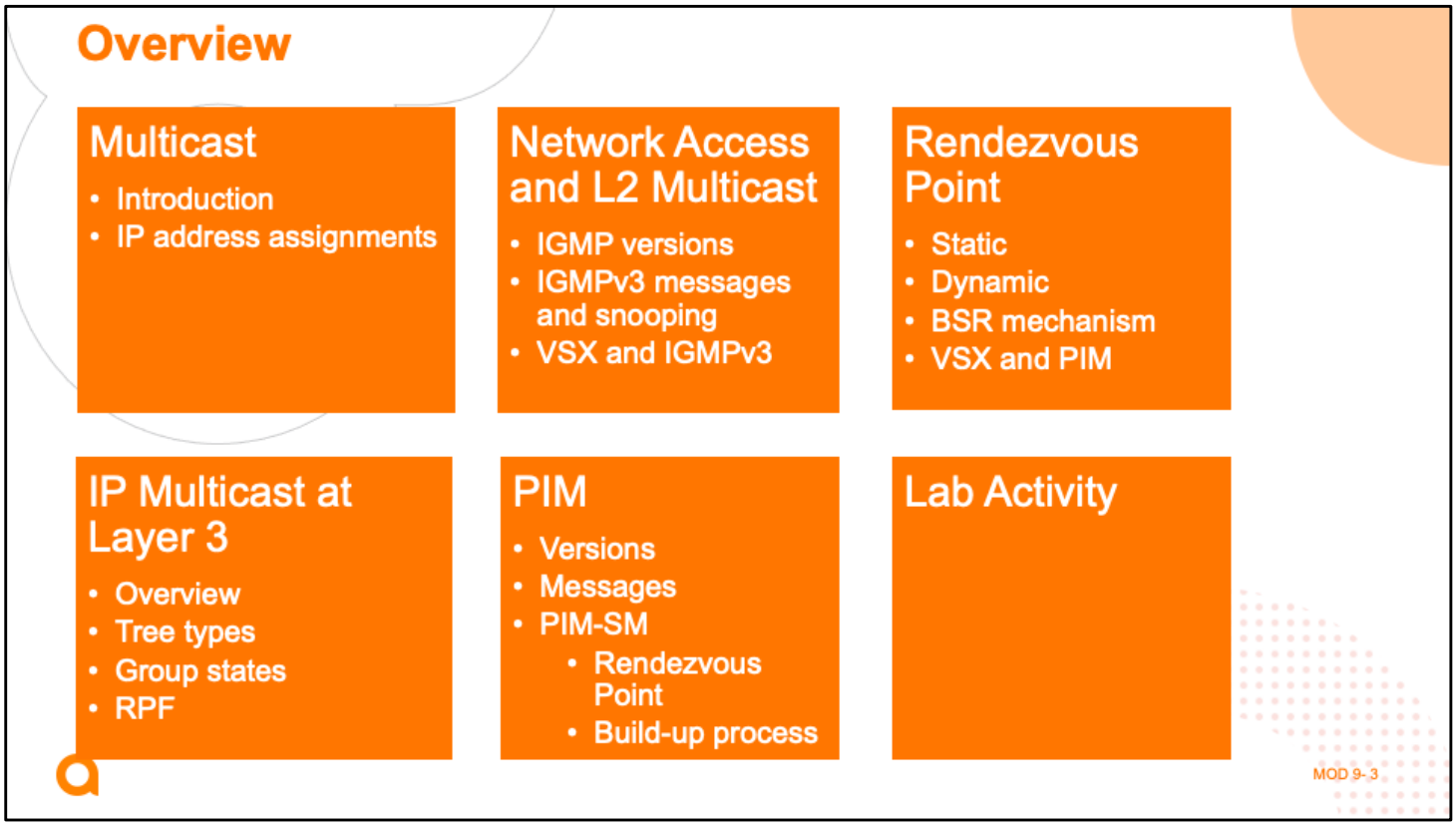
After completing this module, you will be able to describe multicast addressing in all its various forms. This includes Layer 2 multicasting, and how L3 multicast IP addresses are mapped to special Layer 2 multicast MAC addresses.

You will be able to describe the Internet Group Message Protocol (IGMP) and its various versions, with a particular focus on IGMPv3 operation, configuration, and verification..

You will be able to explain layer 3 multicast routing and the use of multicast trees – both shared trees and source trees, along with group states and Reverse Path Forwarding (RPF).

You will be able to describe the Protocol Independent Multicast – Sparse Mode (PIM-SM) operation and analysis, and to configure static and dynamic Rendezvous Points (RP).

Finally, you will be able to deploy and troubleshoot multicast solutions.



Effective use of multicasting can have quite a positive effect on your network – more efficient communications with lower link utilization.

You begin with an introduction to multicasting and multicast IP address assignments. Then you learn about Layer 2 multicasting – how multicast frames are switched, and how to Layer 3 IP addresses are mapped to special Layer 2 multicast MAC addresses. You will explore IGMPv3 operation and messages, along with a great feature that saves both bandwidth and end-system processing overhead – IGMP snooping. You will also learn how IGMPv3 works in conjunction with VSX.

Now it is time to move on to Layer 3 multicasting, where you explore both shared trees with a Rendezvous Point (RP) and Source or “Shortest Path First” trees. You will learn to analyze group states and the ever-important Reverse Path Forwarding (RPF) process.

Next, you dive into the most important protocol for routing multicast packets – Protocol Independent Multicasting (PIM). You will review PIM versions and message types.

Now you dive into the specifics of PIM-Sparse Mode (PIM-SM) – Designated Routers (DR), Rendezvous Points (RP), and the process of how PIM-SM multicast forwarding trees are built, along with some Wireshark analysis.

Next, you learn about RP configuration. This includes both static and dynamic RP configuration, and the BootStrap Router (BSR) mechanism. You will analyze some Wireshark captures, and explore how PIM operates with VSX.

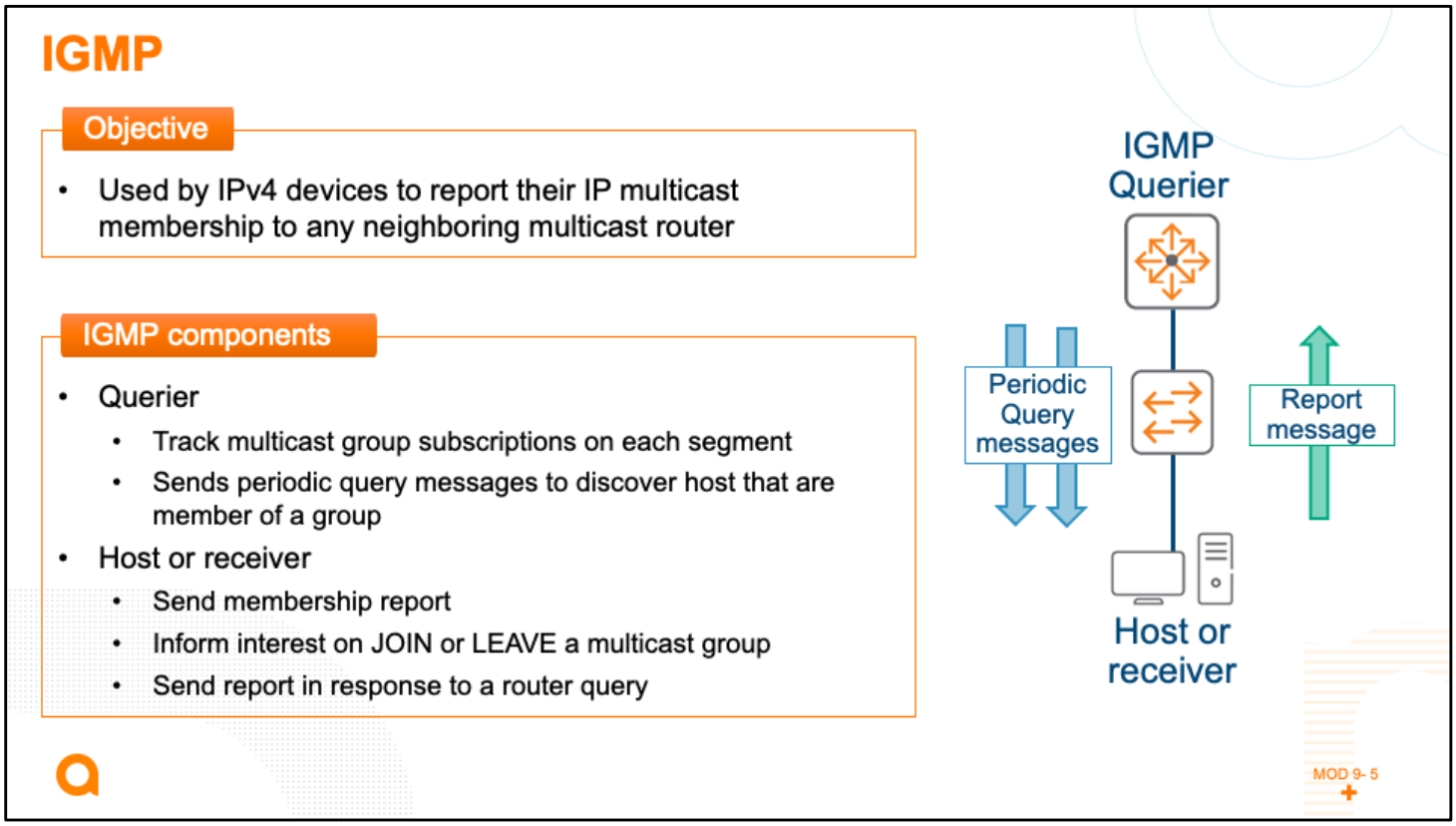
This all prepares you for multicast deployment and troubleshooting. You will look at PIM configuration and methodical method you can use to troubleshoot multicast issues.

You will leverage all this knowledge during some hands-on lab activities.

aruba
a Hewlett Packard Enterprise company

Network Access and Layer 2 Multicast

IGMP

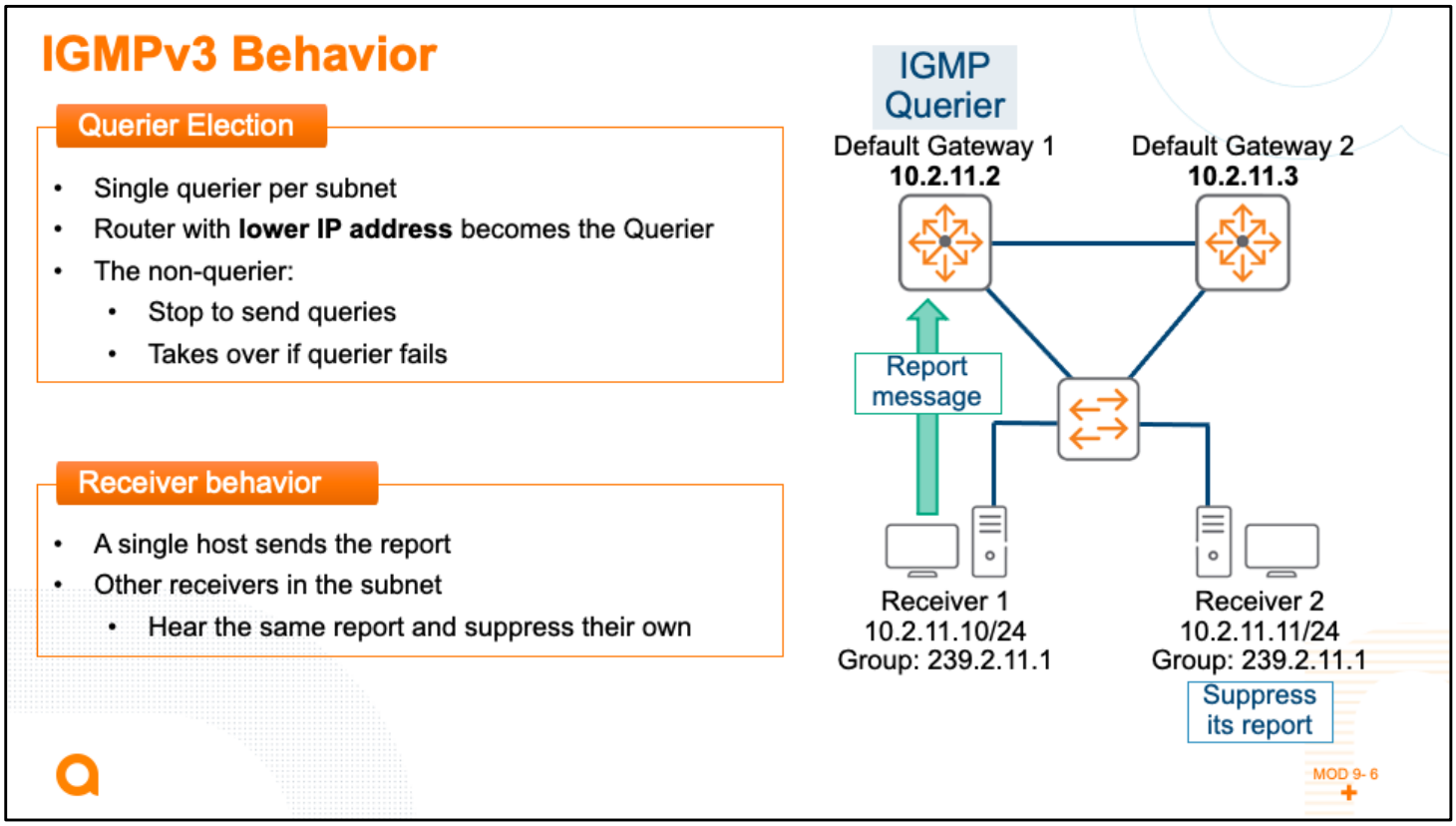


IPv4 systems (hosts and routers) use Internet Group Management Protocol (IGMP) to report their IP multicast group memberships to any neighboring multicast routers. IGMP messages are encapsulated in IP datagrams, with IP protocol number 2.

IGMP establishes two device types - Querier and Hosts or receivers

A querier is a Layer 3 switch that runs IGMP. It sends periodic queries, asking hosts in the subnet which multicast group traffic they wish to receive. A multicast group is commonly referred to as a multicast stream - the application information, like a video stream.

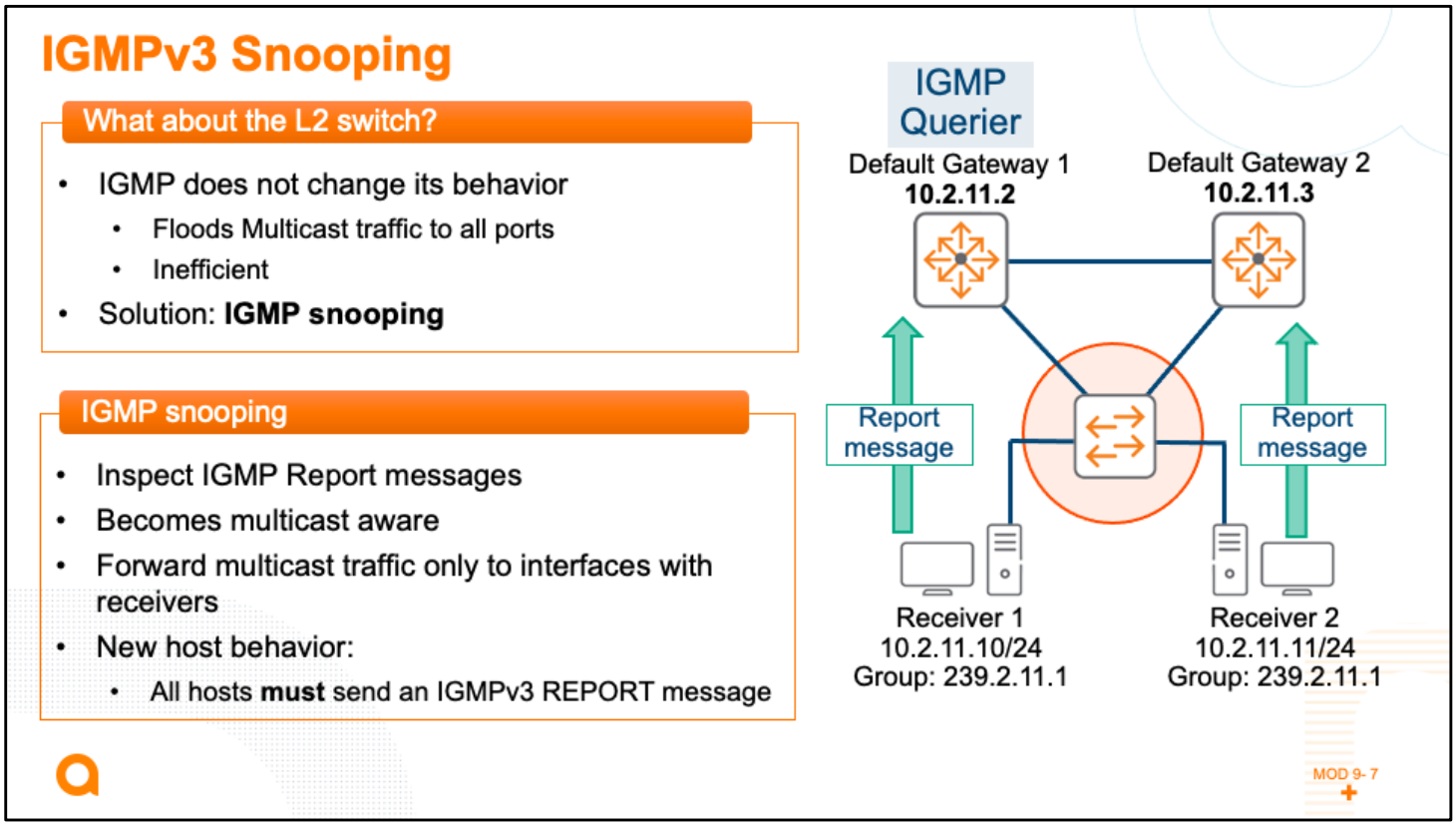
A multicast receiver is a host that wishes to receive a multicast stream or streams, like a video conference unit or a user's desktop.



Just as you should consider redundancy for the default gateway, you should also provide redundancy for the IGMP querier role. IGMPv3 elects a single querier per subnet. Other querier-capable switches become IGMP listeners. A listener continues to use IGMP to learn the multicast traffic that endpoints desire in the interface, just like the querier. It simply refrains from sending queries itself. The listener also takes over as querier if it does not hear a query within two times the querier interval.

IGMPv3 elects a single querier per subnet this decision is based on the IP address. When a router receives a query with a lower IP address, it sets the Other-Querier- Present timer to Other Querier Present Interval and ceases to send queries on the network if it was the previously elected querier. After its Other-Querier Present timer expires, it should begin sending General Queries.

When a host receives an IGMP query packet, it kicks off a timer that begins with a random value that is less than the Maximum Response Time. If no other host responds with a membership report before this random timer expires, the host will then reply with a report. This decreases the number of total IGMP reports needed to maintain the group state. This preserves local bandwidth, because the host suppresses its own reports unless absolutely necessary.



Recall that IGMP only cares about the Querier and the Receivers. But what about the intermediate Layer 2 switches between these two? IGMP does not change its default behavior, which means that traffic is flooded out all ports – wasting bandwidth and reducing efficiency. The solution is IGMP snooping.

With IGMP snooping, Layer 2 switches monitor frames from devices and listen specifically for IGMP messages. Thus, switches can maintain a map of which ports need which IP multicast stream. Thus, switches only forward multicasts packets out interfaces connected to interested receivers.

After receiving a host's IGMP report, the snooping switch forwards the report through all the querier ports in the VLAN. It also looks up the forwarding table for a matching entry as follows:

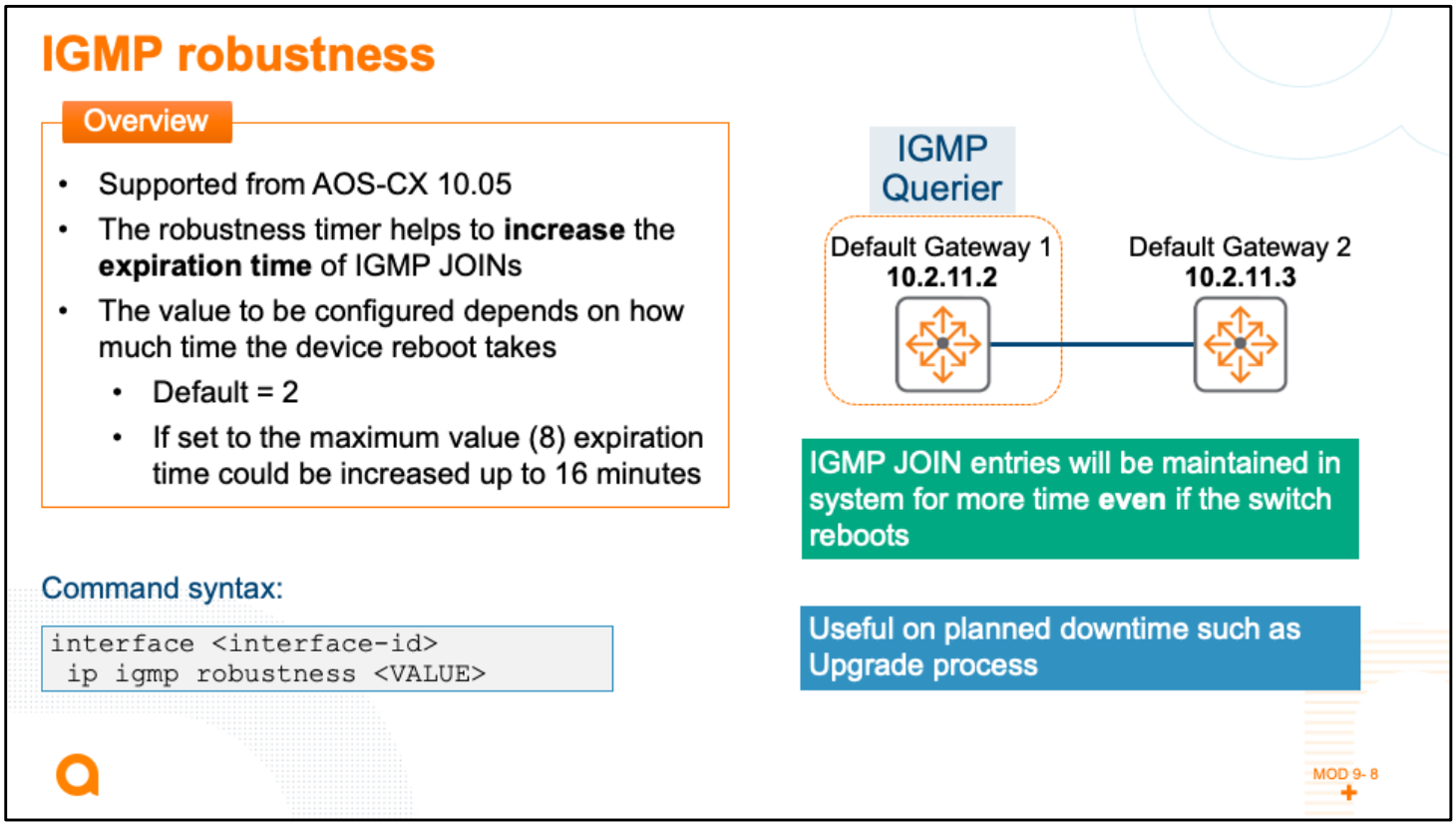
If no match is found, the snooping switch creates a forwarding entry with the receiving port as an outgoing interface. It also starts group membership expiry timer for the port to track the amount of time that must pass before a multicast router decides there are no more members of a group on a network.

If a match is found but the matching forwarding entry does not contain the receiving port, the snooping switch adds the receiving port to the outgoing interface list. It also starts group membership expiry timer for the port.

If a match is found and the matching forwarding entry contains the receiving port, the

snooping switch restarts the group membership expiry timer for the port.

IGMP snooping modifies the behavior of a host, all hosts must send their IGMPv3 Report membership message to the querier.



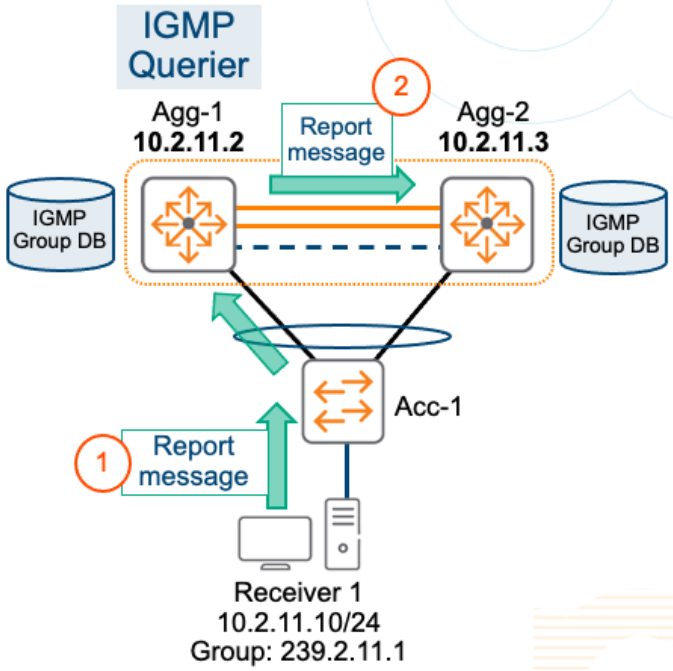
IGMP robustness is a feature starting from AOS-CX 10.05 that can be used on planned downtime periods of the Querier such as the upgrade process. The feature will maintain the IGMP JOIN entries in the system for more time even if the switch reboots. For example, when the robustness value is set to the maximum value (8) the expiration timer could be increased up to 16 minutes giving enough time for the switch to reboot.

To configure the robustness feature, use the ip igmp robustness command on the interface level.

VSX and IGMPv3

Behavior

- Both members build their IGMP group DB
- Synchronization process
  - Any VSX peer receives an IGMP membership (LAG hash decision)
  - Uses ISL to flood info with its VSX peer
- Forwarding/Pruning locally computed on each VSX member
- Querier election process remains unchanged
- If a VSX switch reboots:
  - Learn IGMP entries by itself (sent out general queries)
  - To prevent disruption it floods multicast traffic for 120 seconds



MOD 9- 9

Each VSX peer builds its own IGMP group database independently. The synchronization process is described as follows:

Any VSX node could receive an IGMP membership report (Join or Leave message) from an access switch (LAG hash algorithm determines the link to use). This VSX node learns from the IGMP message and an entry is added on its IGMP group database.

The receiving VSX node sends a copy using the ISL link. The VSX peer learns from the IGMP membership report and the entry is added on its database.

In this example the Acc-1 switch chooses the link to Agg-1 for forwarding the IGMP membership report (Join), and Agg-1 learns the group on the VSX LAG. The IGMP report crosses the ISL to Agg-2. The VSX IGMP process translates the IGMP JOIN learned on the ISL into an IGMP JOIN message learned on the VSX LAG, just as if the JOIN had arrived on the link from Acc-1. Therefore, Agg-2 also learns the group on the VSX LAG to SW1.

In this way, each VSX switch can determine which interfaces to forward or prune multicast traffic for an IGMP group, based on its own IGMP group database. Because the IGMP database construction relies on this data plane-based process, Inter Switch Link Protocol (ISLP) does not synchronize IGMP groups between VSX peers.

If a VSX switch reboots, it needs to re-learn all the IGMP groups. For about 120 seconds after

rebooting, the VSX switch floods multicast traffic within VLANs that have active, forwarding physical ports, which prevents disruption of multicast forwarding during the learning process. The VSX switch also sends a General Query message. Clients send membership report messages in response, so the VSX switch can re-learn the groups and re-create the IGMP group database.

Notice that querier election process is not changed because of VSX cluster, a single device in the VSX cluster becomes the querier.

Question #1

The administrator has applied the script to enable IGMPv3 in VLANs 11 and 212, why is IGMP snooping not enabled?

a. AOS-CX does not support IGMP and IGMP snooping to be running in the same device

b. IGMP snooping is a separate process and needs to be enabled in the VLAN level

c. There is an ACL applied in VLANs 11 and 212 that avoids IGMP snooping to be enabled

```
(config)# interface vlan 11
(config-if-vlan)# ip igmp enable
(config-if-vlan)# interface vlan 212
(config-if-vlan)# ip igmp enable

# show ip igmp all-vrfs

VRF Name   : default
Interface  : vlan11
IGMP Configured Version  : 3
IGMP Operating Version   : 3
Querier State              : Non-Querier
Querier IP                 : 10.1.11.2
Querier Uptime             : 13m 58s
Querier Expiration Time    : 2m 53s
IGMP Snoop Enabled on VLAN : False
```



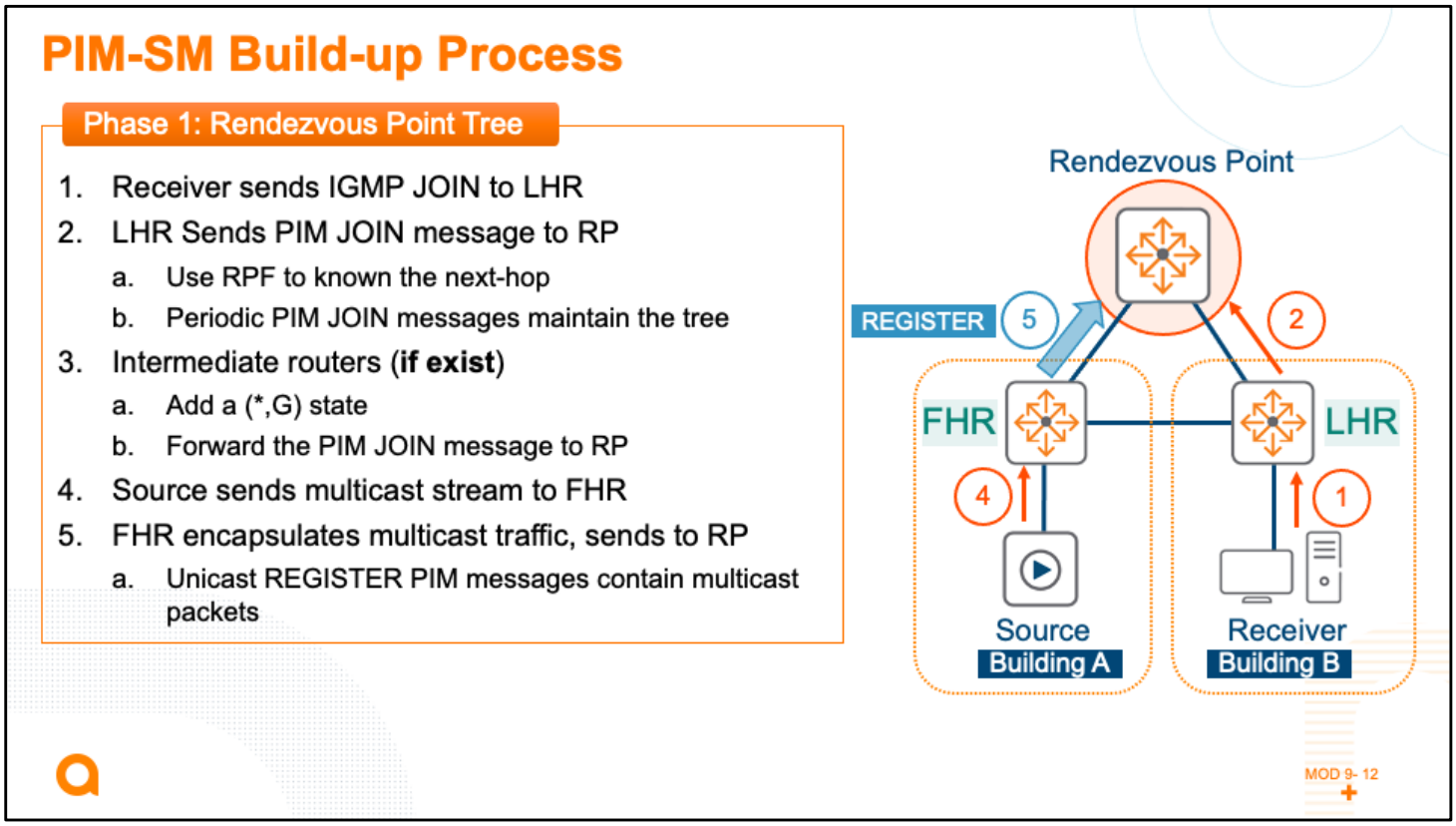
Correct answer:

B

aruba
a Hewlett Packard Enterprise company

PIM-SM

Designated Routers, Rendezvous Point, Build-up Process



RFC 4601 defines three phases for the build-up process of PIM-SM.

Phase 1: Rendezvous Point Tree

Receiver endpoint sends an IGMP JOIN request asking for specific multicast group to the Last-Hop Router (LHR). This message includes only the Multicast group, therefore multicast state (*,G) is created on the FHR's multicast routing table. This action also adds the receiver-facing interface to the Outgoing Interface List.

LHR Sends a JOIN PIM message to the RP:

– LHR uses Reverse Path First (RPF) to look in the Unicast routing table and determine the next-hop to reach the RP.

– PIM Join messages are sent periodically to maintain the multicast state.

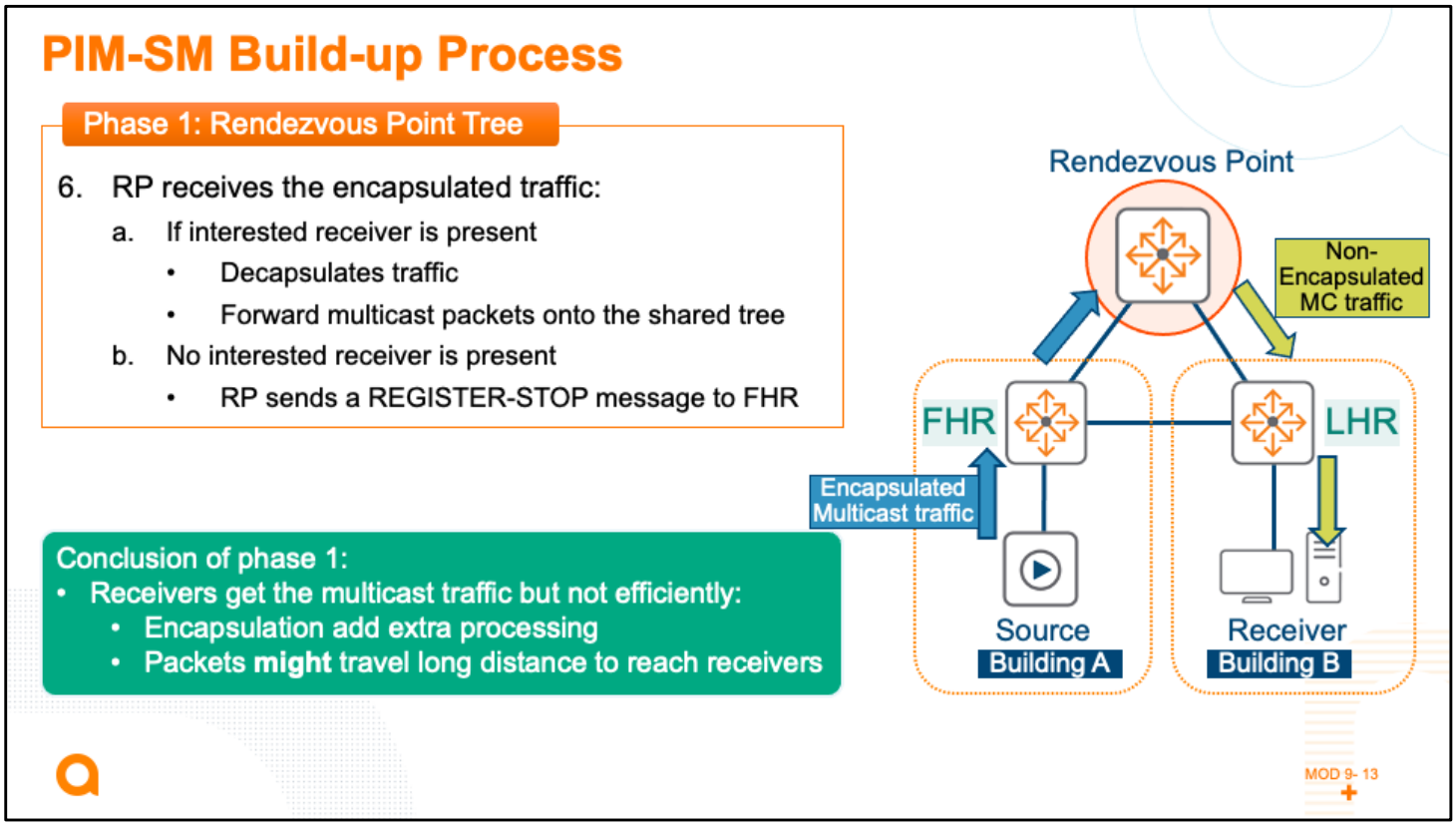
If intermediate routers exist between the LHR and RP, those devices add a (*,G) state on their multicast routing table and forward the PIM JOIN message to the RP. In the example this step is not needed.

Eventually the packet reaches the RP, which adds a (*,G) state in its multicast routing table and adds the receiver-facing interface in the OIL.

Source sends the multicast stream to the First Hop Router (FHR), which adds a state (S,G) to

its multicast routing table.

The FHR must send the multicast traffic to the RP, but there is a problem. Intermediate routers (between the FHR and the RP path) and the RP do not have a (S,G) state in their multicast routing table. This means that routers will drop the multicast traffic coming from the FHR. To resolve this problem, the router closest to the source must encapsulate the multicast packet inside of a unicast packet with the IP address of the rendezvous point as the IP destination. In other words, the multicast stream must be tunneled from the router at the source to the RP. The encapsulation packets are known as PIM register packets.



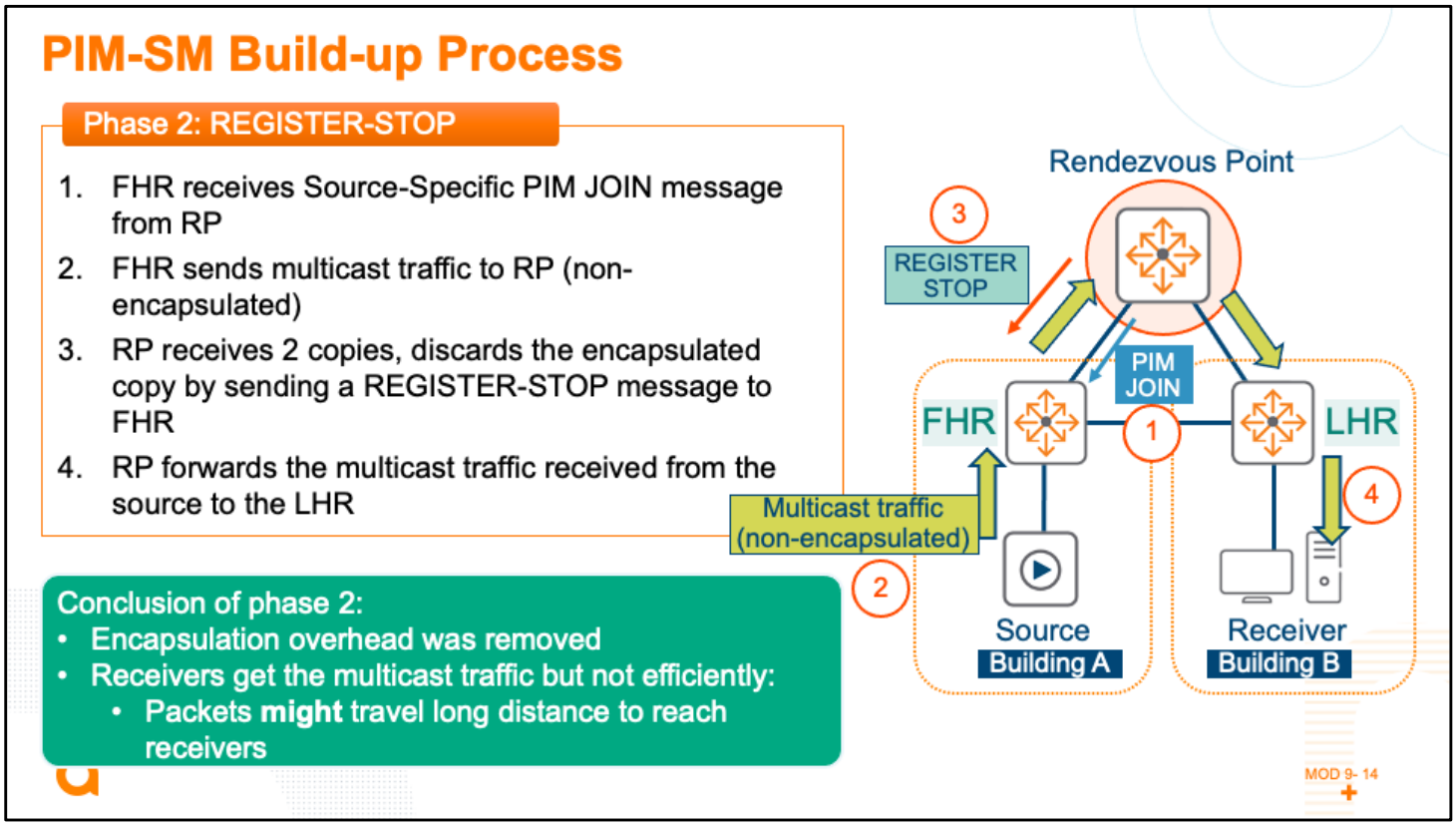
Phase 1: Rendezvous Point Tree

6. The RP receives the PIM REGISTER message (encapsulated traffic) and checks for interested receivers present (*,G) entry in its multicast routing table.

– If there is a receiver present, then it decapsulates the traffic and forward the multicast packets onto the shared tree.

– If there is no reciver present, then the RP sends a REGISTER-STOP message to the FHR. Each router (between RP and source) maintains a register suppression timer that is initiated by the REGISTER STOP message. Upon expiration of this timer, the FHR again sends a register packet to the RP to verify if there are any active receivers for that group.

At the end of phase one, multicast traffic flows encapsulated to the RP, and then natively over the RP tree to the multicast receivers. During this time, the flow is not efficient.



Phase 2: REGISTER-STOP

The FHR receives a source-specific PIM JOIN message from the RP. This adds a (S,G) entry to the FHR multicast routing table and to the intermediate routers in the path. This state permits all devices to process multicast information that is received from the source, this means that the encapsulation process is no longer required.

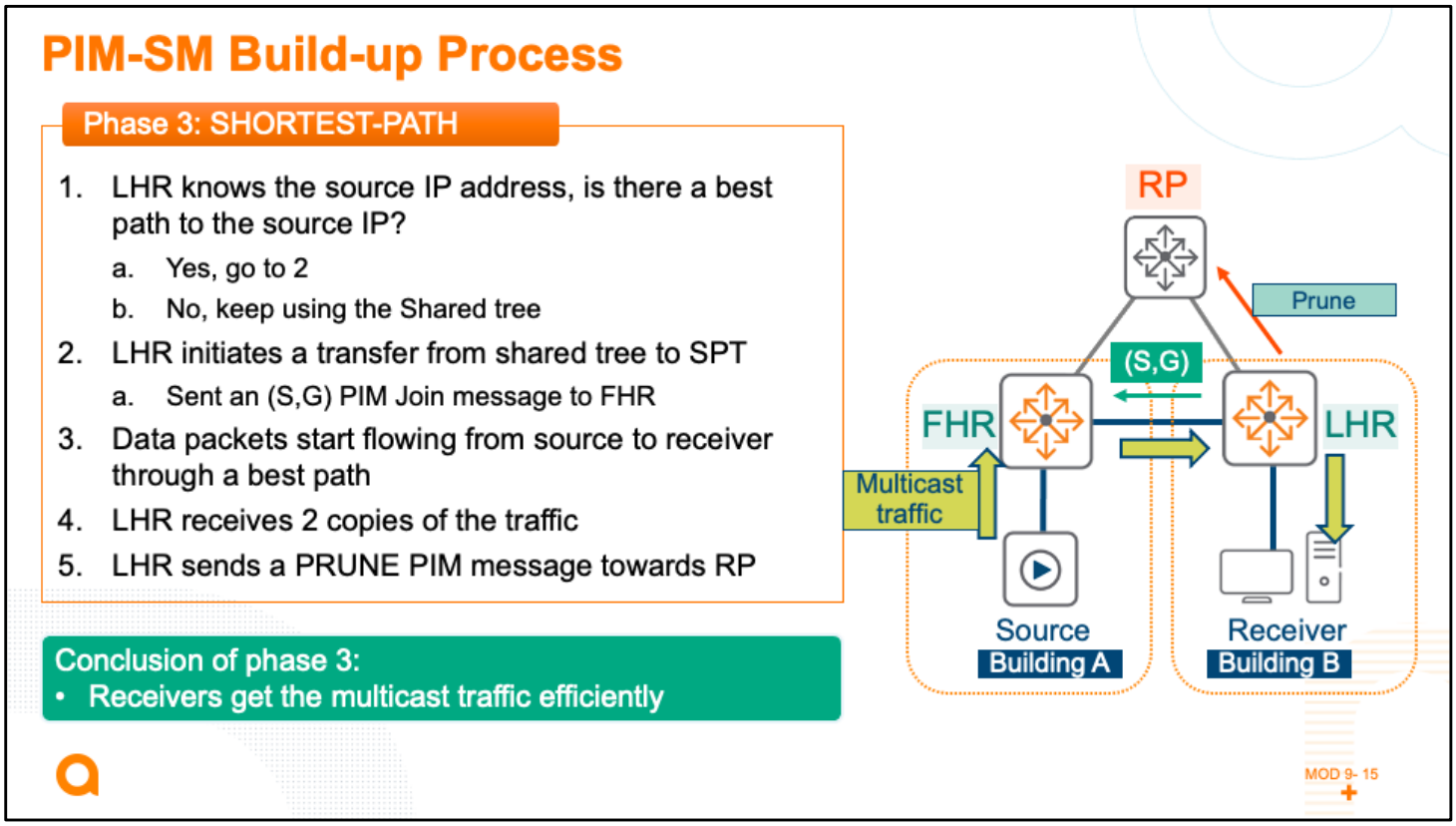
The FHR sends the multicast traffic natively (non-encapsulated) to the RP. All intermediate devices in the path can route the traffic.

At this moment, the RP receives two copies of the same traffic, one encapsulated in the REGISTER message and other natively received. The RP discards the encapsulated copy.

The RP sends a PIM Register-Stop messsage to FHR.

RP receives the multicast traffic from the source with no encapsulation and forwards this traffic to the LHR, this action creates a new state (S,G) in the multicast routing table on all routers between the RP and the LHR.

At the end of phase 2, encapsulation overhead has been removed but the multicast traffic might travel long distance to reach receivers.



Phase 3. Shortest Path

At this moment LHR knows about the source of the Multicast traffic, now the LHR asks itself if is there a best path to the source IP? As you might imagine this process is done by running RPF process. If there is no new best path available, then the LHR keeps using the shared tree path. On the other hand, if there is a new best path, then a process to optimize the path must be done and the process continues with step 2.

The LHR initiates a transfer from the shared tree to the Shortest Path Tree, this means that the LHR will send a PIM Join message (S,G) to the FHR.

When the FHR receives the PIM message it starts flowing the multicast packets to the LHR through the best possible path.

LHR receives two copies of the traffic, one from the RP using the shared tree and other from the Source using the SPT.

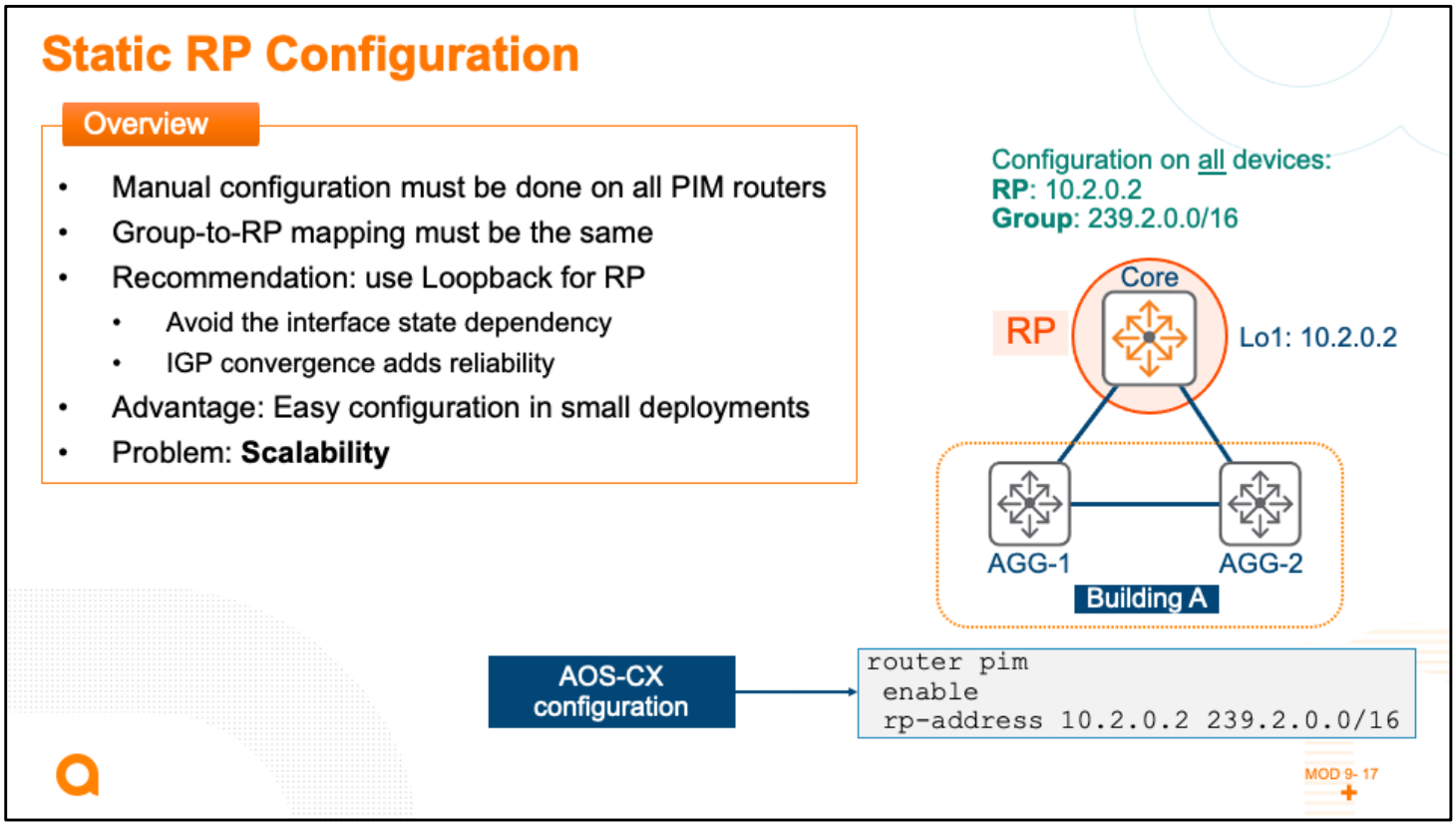
LHR sends a PIM Prune message towards RP.

At the end of phase 3, receivers get the multicast traffic efficiently.

aruba
a Hewlett Packard Enterprise company

RP Configuration

Static and BSR Mechanism



Enabling PIM sparse-mode using a static RP is perhaps the simplest way to configure a multicast domain. A static RP is defined through manual configuration on the RP itself and at every downstream router. The configuration, with multicast scoping (that is, multicast RP configuration with the multicast group definition), must be defined at every downstream router in exactly the same way. Otherwise, the domain will be incomplete. This configuration option can be suitable in small deployments with few multicast group mappings.

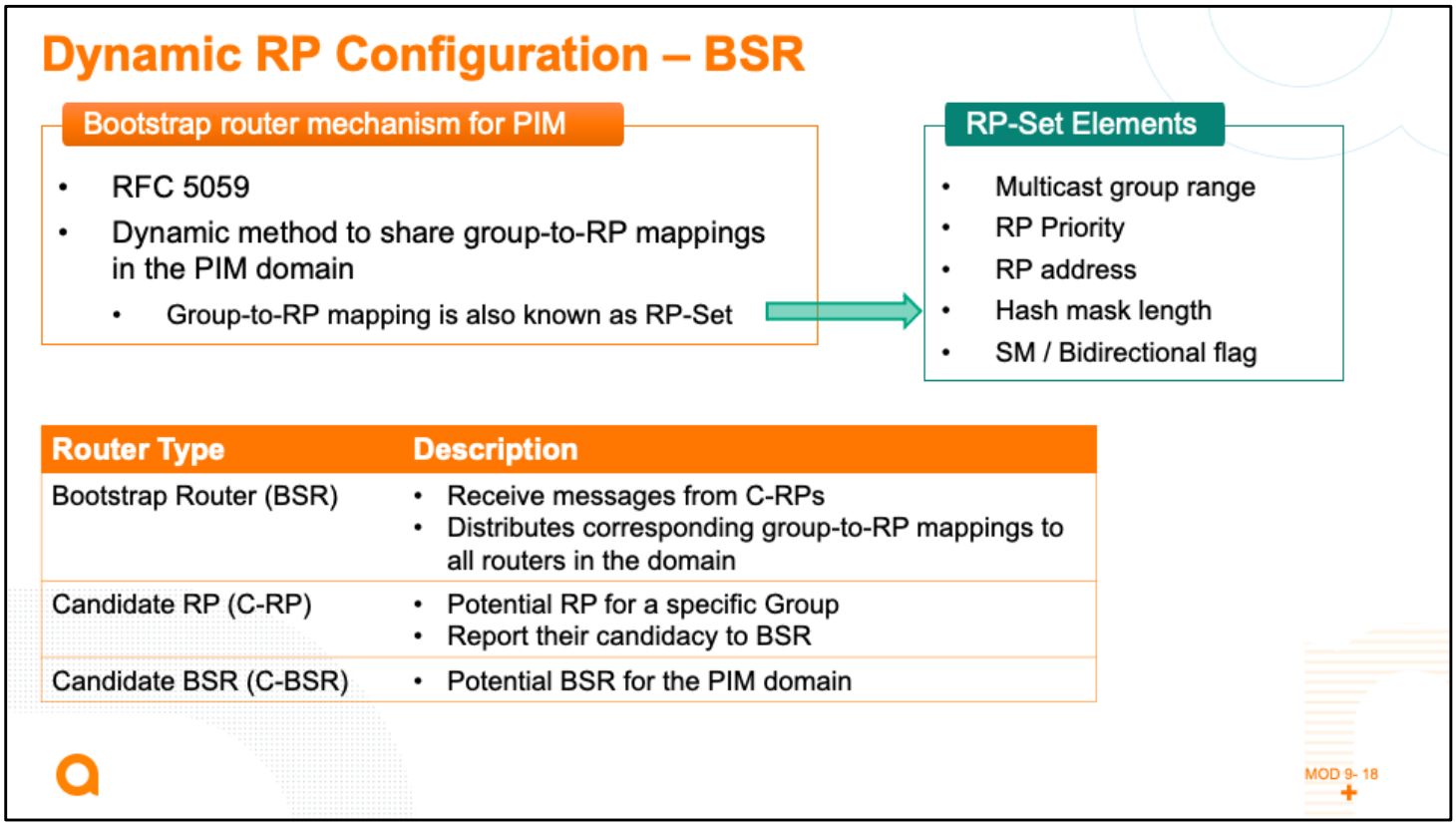
If a network is very large, has many overlapping domains or many multicast applications and many rendezvous points a consistent group mapping through static commands may become extremely difficult to manage.

The configuration in AOS-CX includes a single command:

router pim
 enable
 rp-address <RP-address> <multicast-group/length>

Statically configure the router as the RP for a specified multicast group or range of multicast groups. This must be configured on all PIM-SM routers in the domain. If group address is not specified, it applies to all IPv4 multicast addresses (224.0.0.0 - 239.255.255.255). PIM-SM supports a maximum of 8 static RPs per VRF.

Why is it important to configure the main network loopback interface with sparse-mode PIM? This is a recommended practice for any multicast overlay. The reason for this recommendation is that the router can then fully participate in the multicast domain, even if errors are occurring on leaf facing interfaces. It also allows the loopback interfaces to be used as a RP addresses or mapping agents in dynamic RP propagation, making them more reliable.



The BootStrap Router (BSR) mechanism for PIM is defined in RFC 5059.

BSR is a RP high-availability mechanism that provides active/standby functionality and automatic downstream RP information propagation. This mechanism dynamically establish a method to share group-to-RP mappings (also known as RP-Set) to all PIM routers in the domain. This mechanism avoids wrong mappings that can cause unexpected problems.

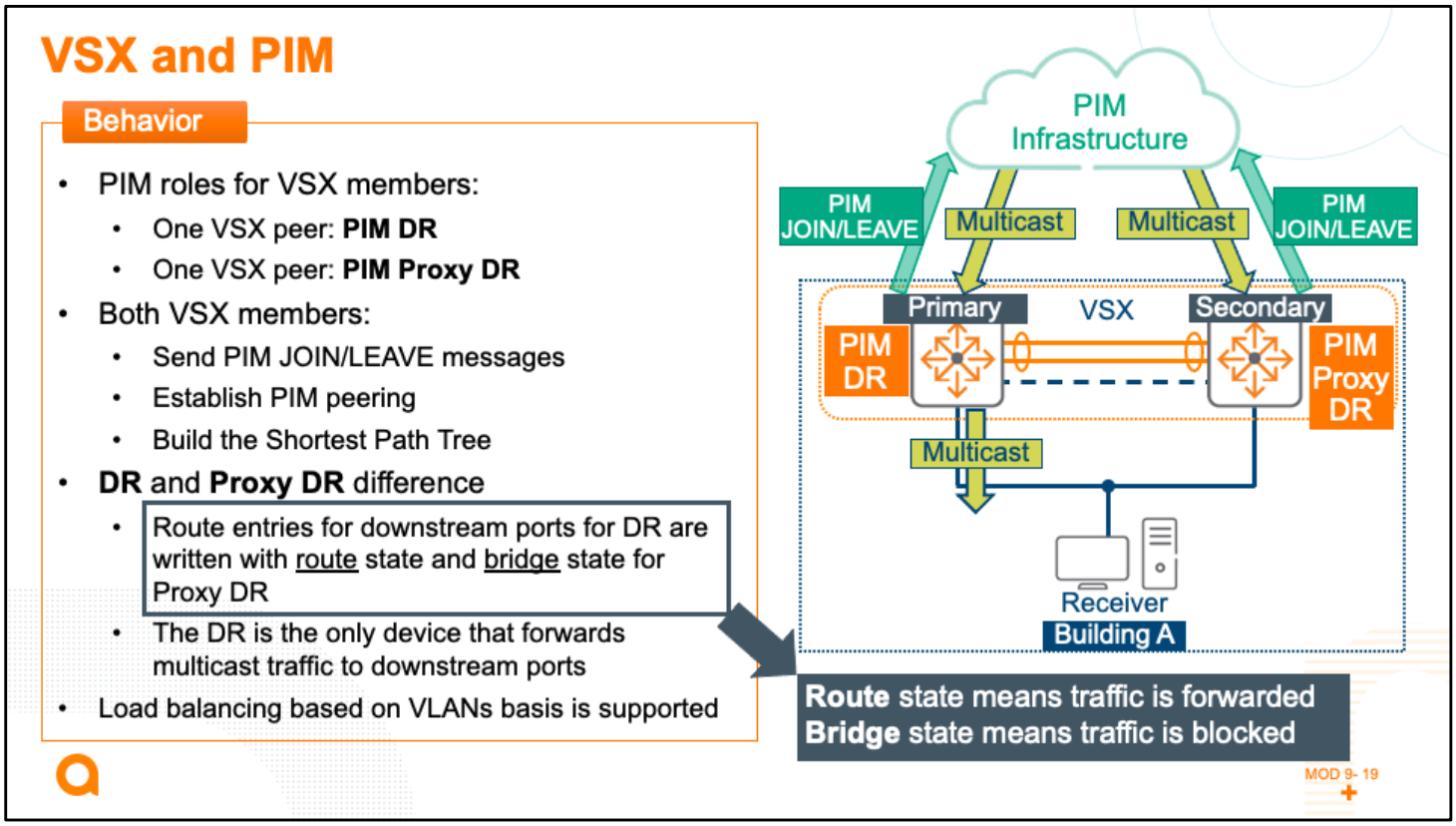
The general idea behind the BSR mechanism is that some of the PIM routers within a PIM domain are configured to be potential RPs for the domain. These are known as Candidate-RPs (C-RPs). A subset of the C-RPs will eventually be used as the actual RPs for the domain. In addition, some of the PIM routers in the domain are configured to be candidate bootstrap routers, or Candidate-BSRs (C-BSRs). One of these C-BSRs will be elected to be the bootstrap router (BSR) for the domain, and all the PIM routers in the domain will learn the result of this election through Bootstrap messages. The C-RPs will then report their candidacy to the elected BSR, which chooses a subset of these C-RPs and distributes corresponding group-to-RP mappings to all the routers in the domain through Bootstrap messages.

RP-Set elements shared by Bootstrap are:

Multicast group range, expressed as an address and prefix length

RP Priority

RP address
Hash mask length
SM / BIDIR flag

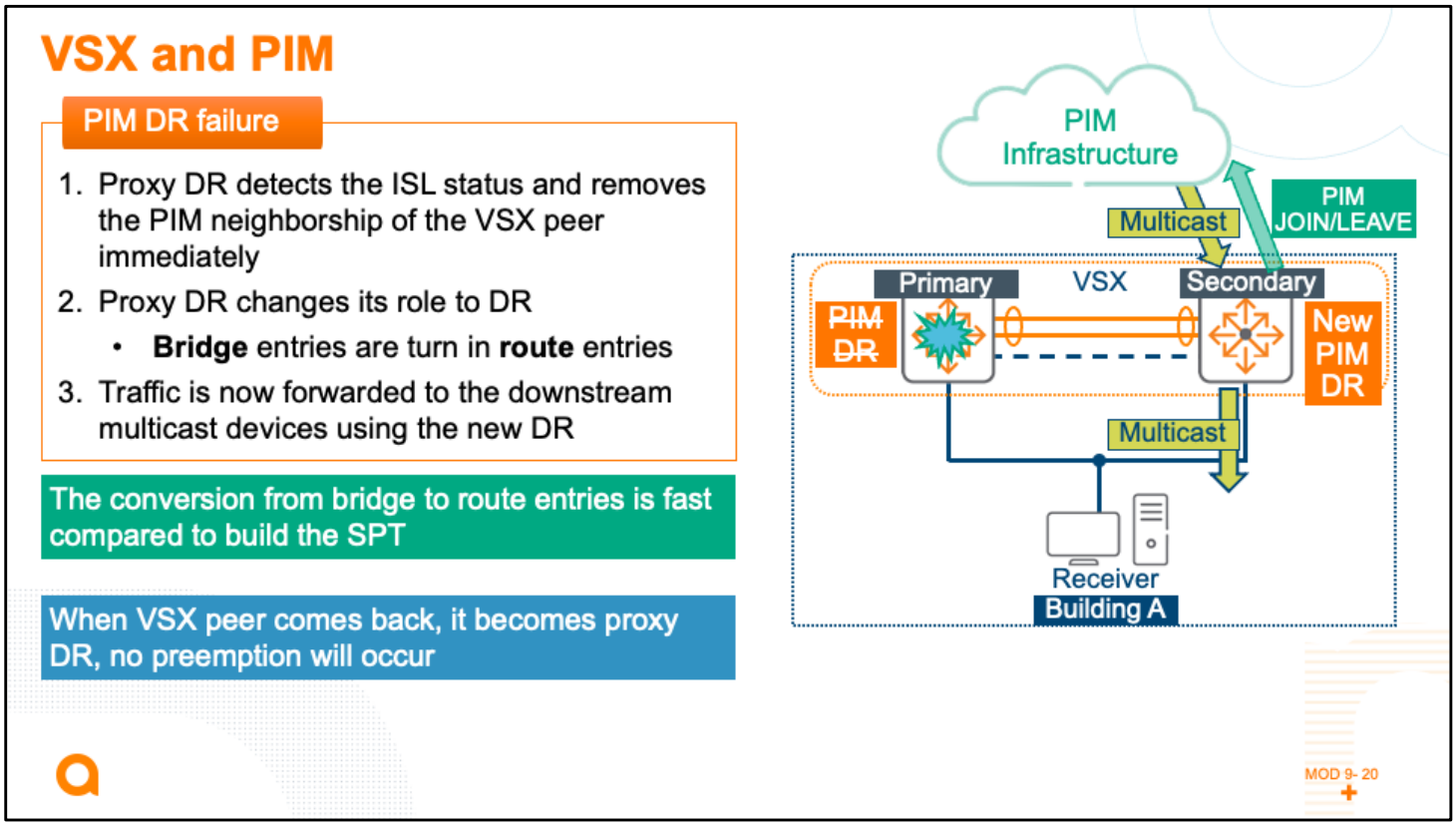


AOS-CX supports Dual designated router (DR) functionality for VSX. One VSX member is elected as PIM DR for each IP subnet, using the typical PIM election process. The other VSX member becomes a PIM Proxy DR.

Both VSX peers have the same Control Plane information, this means that both members will be able to establish PIM neighborships, send PIM JOIN/LEAVE messages to the RP, populate (S,G) entries on their multicast routing table and Build a Shortest Path Tree (SPT). However, multicast traffic (data plane) is only routed from the VSX peer that acts as the PIM DR. Database route entries for downstream ports in DR and Proxy DR are always in sync but are written different on each device, multicast entries in the DR are written as route entries and the entries in the Proxy DR as bridge entries.

You can implement VLAN-based load balancing so that for each VSX pair, one switch can be the DR for a group of VLANs and the other switch the DR for the other group of the VLANs.

Note. This solution works for IPv4 and IPv6.



The mechanism to have a pre-established Control Plane on both VSX peers permits the VSX cluster to achieve a fast-failover in case the PIM DR fails.

Once the VSX peer that is acting as DR does down, the Proxy DR detects the ISL status and removes the PIM neighborship of the VSX peer immediately.

The Proxy DR changes its role to DR, this action turn the bridge entries to downstream VSX ports to route entries.

Traffic is now forwarded to the downstream multicast devices using the new DR.

Note. The conversion from bridge to route entries is done in the Configuration and State Database and the change would propagate to Hardware. This approach take much less time compared to forming the SPT for the multicast traffic.

When the VSX peer is back online again, it will become Proxy-DR and will start to learn about the Multicast routes (this implies building PIM neighbor relationships, populate (S,G) entries on their multicast routing table and Build a Shortest Path Tree (SPT)). To avoid the new Proxy-DR becomes DR again after the reboot (because of higher IP address), the current DR sends Hello packets with highest DR priority (4294967295) any user-configured DR priority will be ignored.

Recommendation:

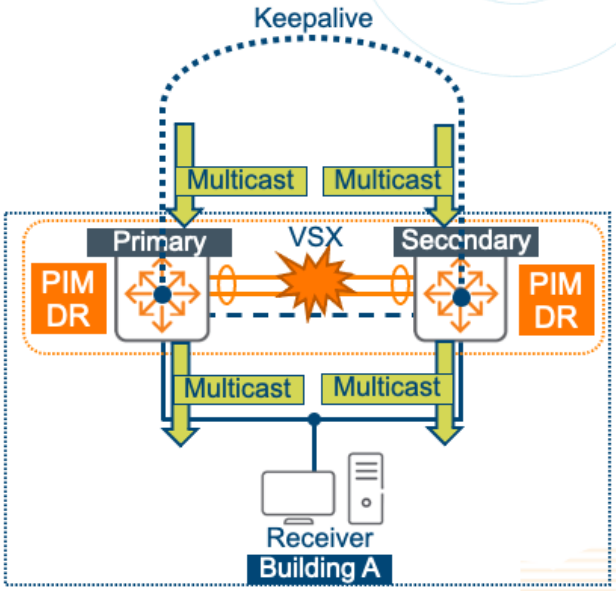
Keep the PIM hello-interval timer to default value: 30 seconds; this timer no longer has effect to improve VSX PIM fast-failover.

Consider configuring PIM-Sparse DR-Priority in a way that each VSX node has an equal VLANs load distribution regarding the actual DR role. This will help in further minimizing the impact during fast-failover.

VSX and PIM

ISL Failure

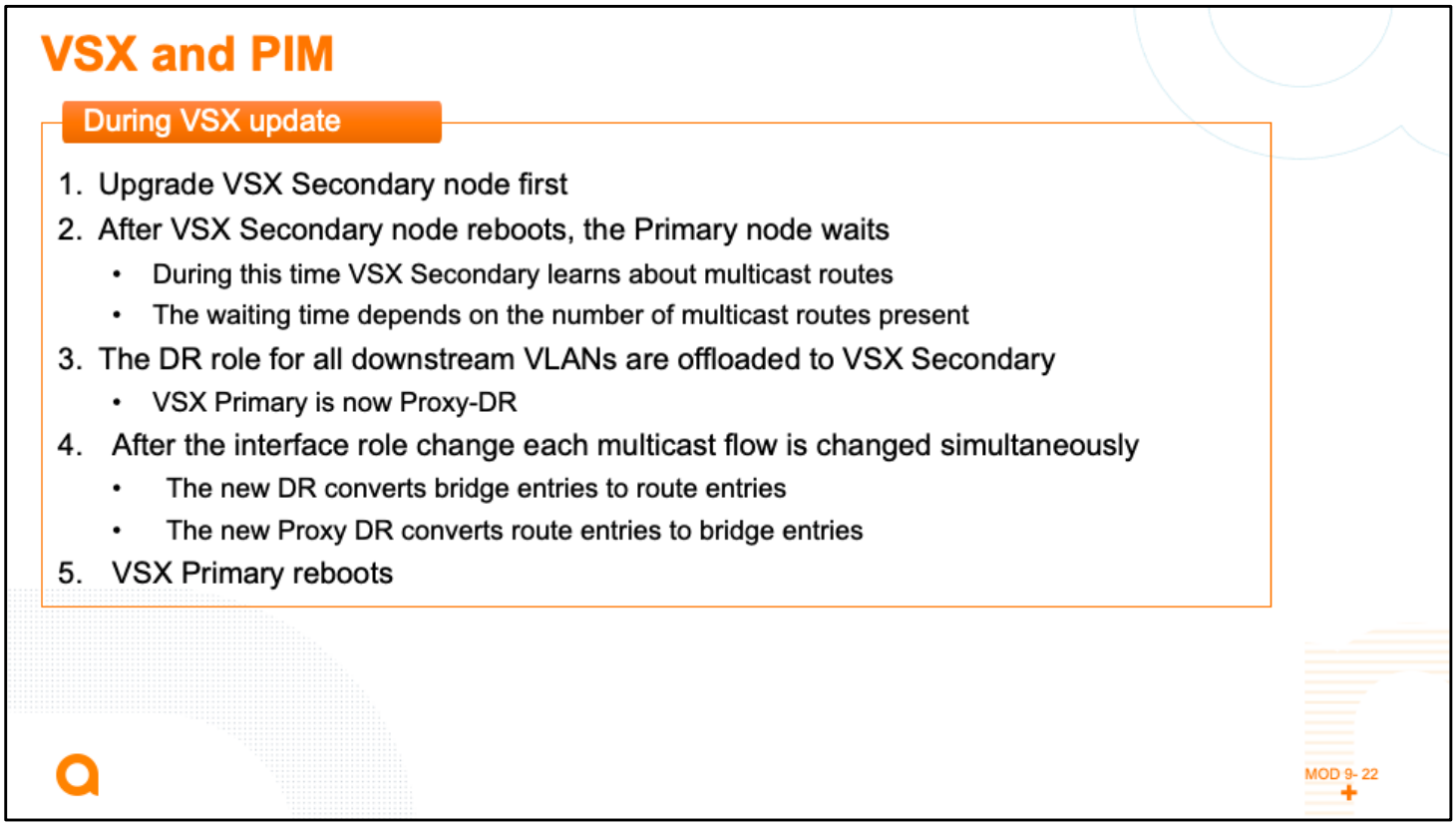
- ISL goes down with VSX keepalive
  - VSX Primary stays as **DR**
  - VSX Secondary becomes **Proxy DR** after at least 5 seconds
  - Duplicated traffic is expected during this time
- ISL goes down without VSX keepalive
  - Both VSX peers act independently
  - Both VSX peers will forward multicast traffic



MOD 9- 21

If the ISL goes down with VSX keepalive present and both VSX peers are up, then the VSX primary stays as DR, the VSX Secondary becomes the Proxy DR when its downstream links gets cut by VSX infrastructure. This process takes at least 5 seconds. During this time duplicated traffic is expected as both of the VSX peers act as Designated Routers.

If ISL goes down when keepalive is not configured, both the VSX peers would act independently and the downstream VSX ports would not be brought down. This results in duplicate traffic as both of the VSX peers start acting as Designated Routers.



During VSX update:

1. The VSX Secondary node must be upgraded first, this action does not represent any problem to the current multicast streams that are using the network since the VSX Secondary node acting as DR Proxy does not handle the data plane.

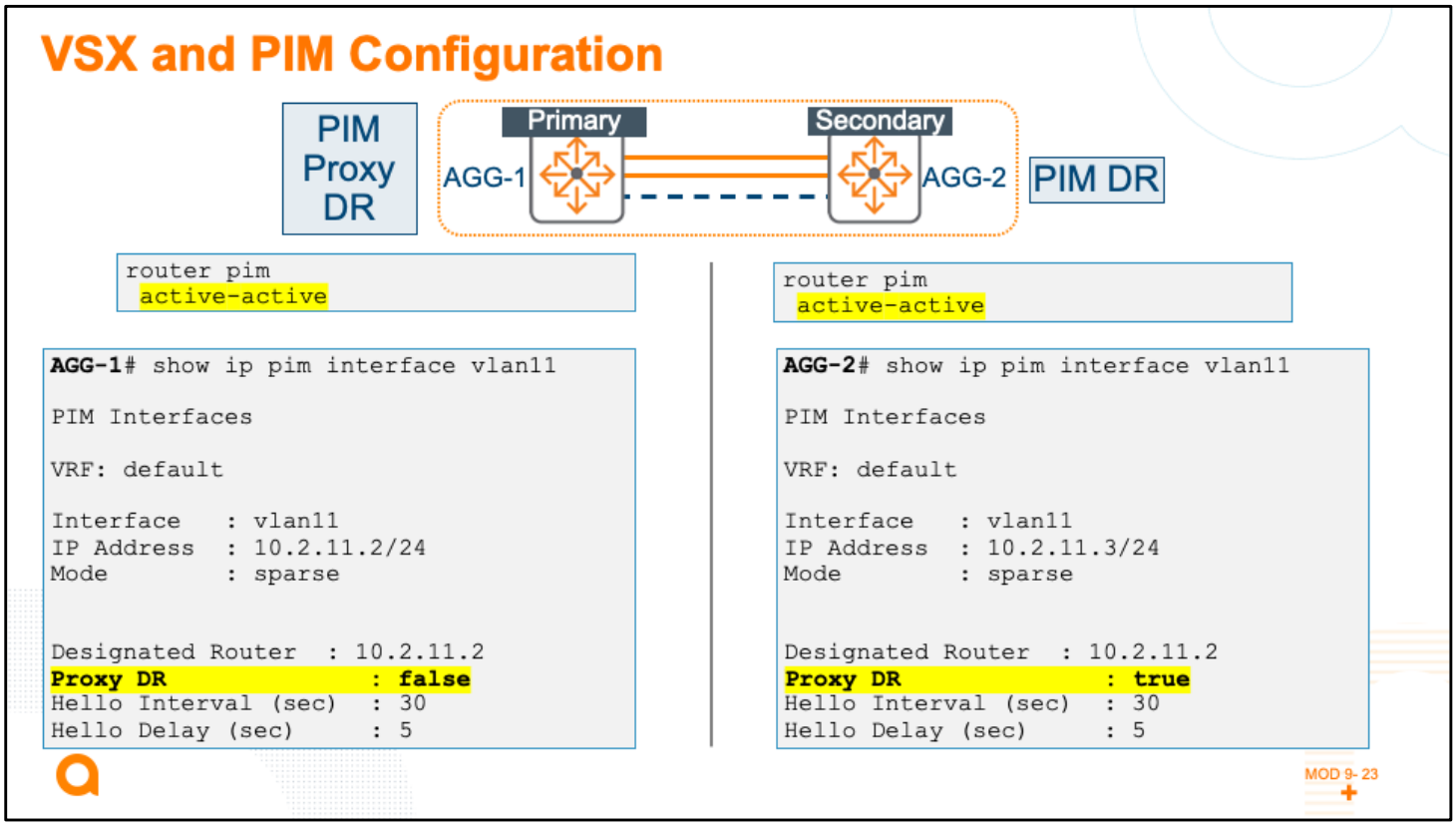
2. After the VSX Secondary node reboots using the new firmware the VSX Primary node waits for few minutes for the Secondary VSX node to learn all Multicast routes, the amount of time that the Primary node must wait depends on the number of multicast routes present.

3. The DR role for all downstream VLANs are offloaded to VSX Secondary, VSX Primary is now the new Proxy-DR.

4. After the interface role change each multicast flow is changed simultaneously: The new DR (VSX Secondary) converts bridge entries to route entries and the new Proxy DR (VSX Primary) converts route entries to bridge entries.

5. VSX Primary reboots.

The advantage of using VSX is there is no multicast traffic loss during the upgrade traffic, however some duplicates are expected to be seen for 1-3 seconds for each multicast traffic.



VSX Configuration

router pim
 enable
 active-active

active-active command

This command enables the PIM active-active mechanism per VRF on VSX. The default is disabled. This command keeps the multicast forwarding state synchronized on both VSX peer devices. Synchronization is achieved by electing the VSX peer that has the highest IP address as a designated router (DR) and the other as Proxy-DR. If the user wants the multicast traffic to flow through VSX primary, assign higher IP addresses to the interfaces in VSX primary. When the VSX peer that is acting as the DR goes down, traffic is recovered faster since the multicast routes are synchronized.

Recommendations:

Do not configure the DR priority of interfaces when active-active is enabled. The DR priority will be set to high on DR and default on Proxy-DR and any user-configured DR priority will be

ignored.

Always configure keepalive between VSX peers. If the ISL goes down when keepalive is not configured, both VSX peers start acting independently as DRs, resulting in duplicate traffic.

Do not have IGMP joins on transit VLANS.

RP redundancy is not supported on the active-active feature. If one of the VSX peers is configured as RP and it goes down, the new flows will not be converged until the RP is elected. For a static RP, new flows will never be converged until the VSX peer is back up.

Do not configure active-active on a non-VSX setup.

Question #2

```
VSX-node# show ip mroute brief vrf b2
-------------------------------------------------------------------------------
            VRF : b2                          Total number of entries : 1
-------------------------------------------------------------------------------
Group Address      Source Address     Neighbor      Interface      Uptime          State
                                                                  (HH:MM:SS)
---------------    ---------------    ----------    -----------    ------------    ------
239.1.212.1        10.1.212.205       10.1.212.3    vlan212        00:00:06        bridge
```

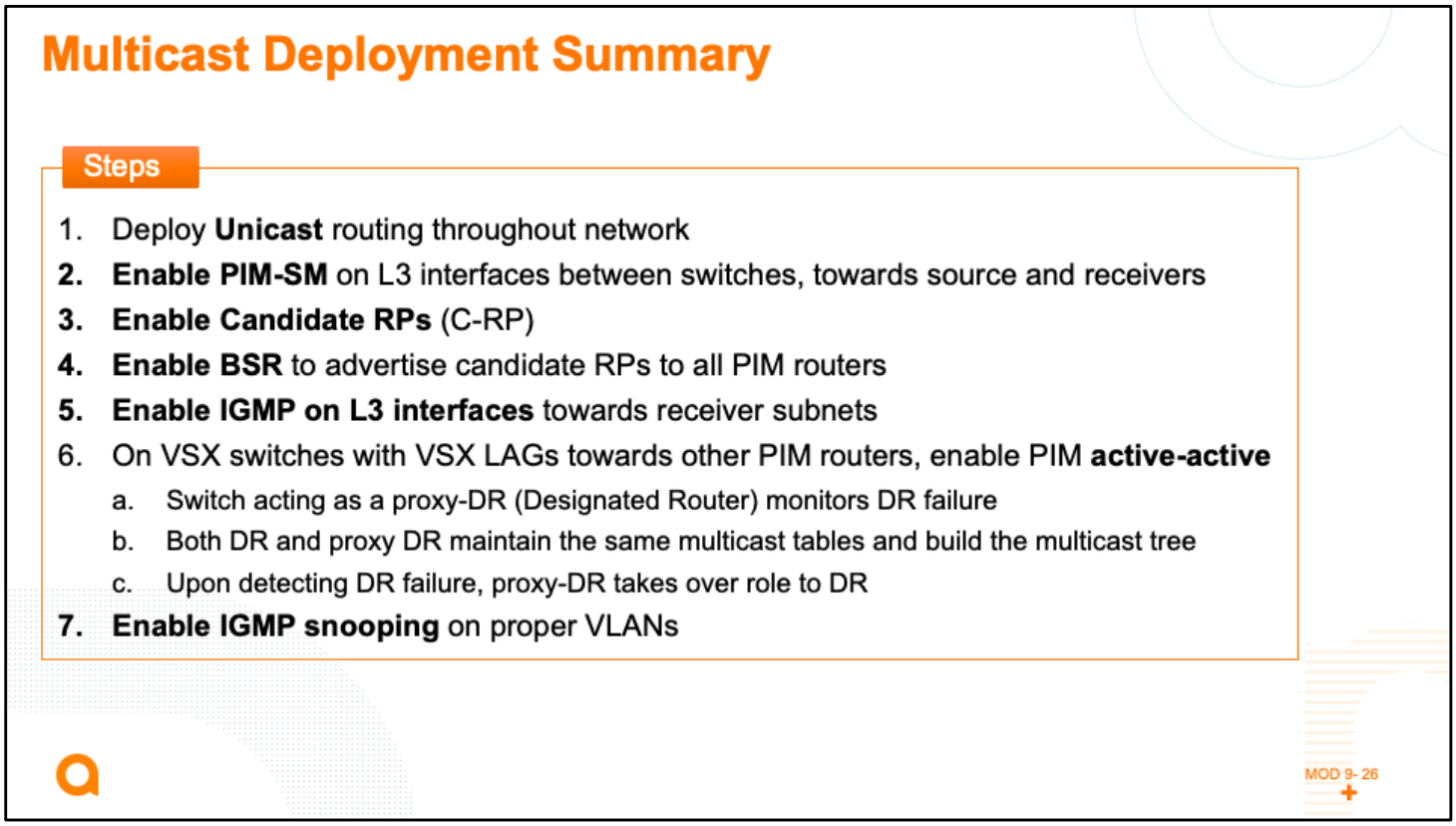
From the output, What does the State **bridge** mean in a VSX node?

a. This device is the DR and multicast traffic is forwarded
b. This device is the Proxy DR and multicast traffic is forwarded
c. This device is the DR and multicast traffic is not forwarded
d. This device is the Proxy DR and multicast traffic is not forwarded

Knowledge Check

Correct answer:
D


aruba
a Hewlett Packard
Enterprise company
Multicast
Deployment
Configuration and Troubleshooting



Multicast deployment summary:

Deploy unicast routing (OSPF/BGP) throughout network

Enable PIM-SM on L3 interfaces between switches, towards source and receivers

Enable Candidate RPs (C-RP)

Enable BSR to advertise candidate RPs to all PIM routers

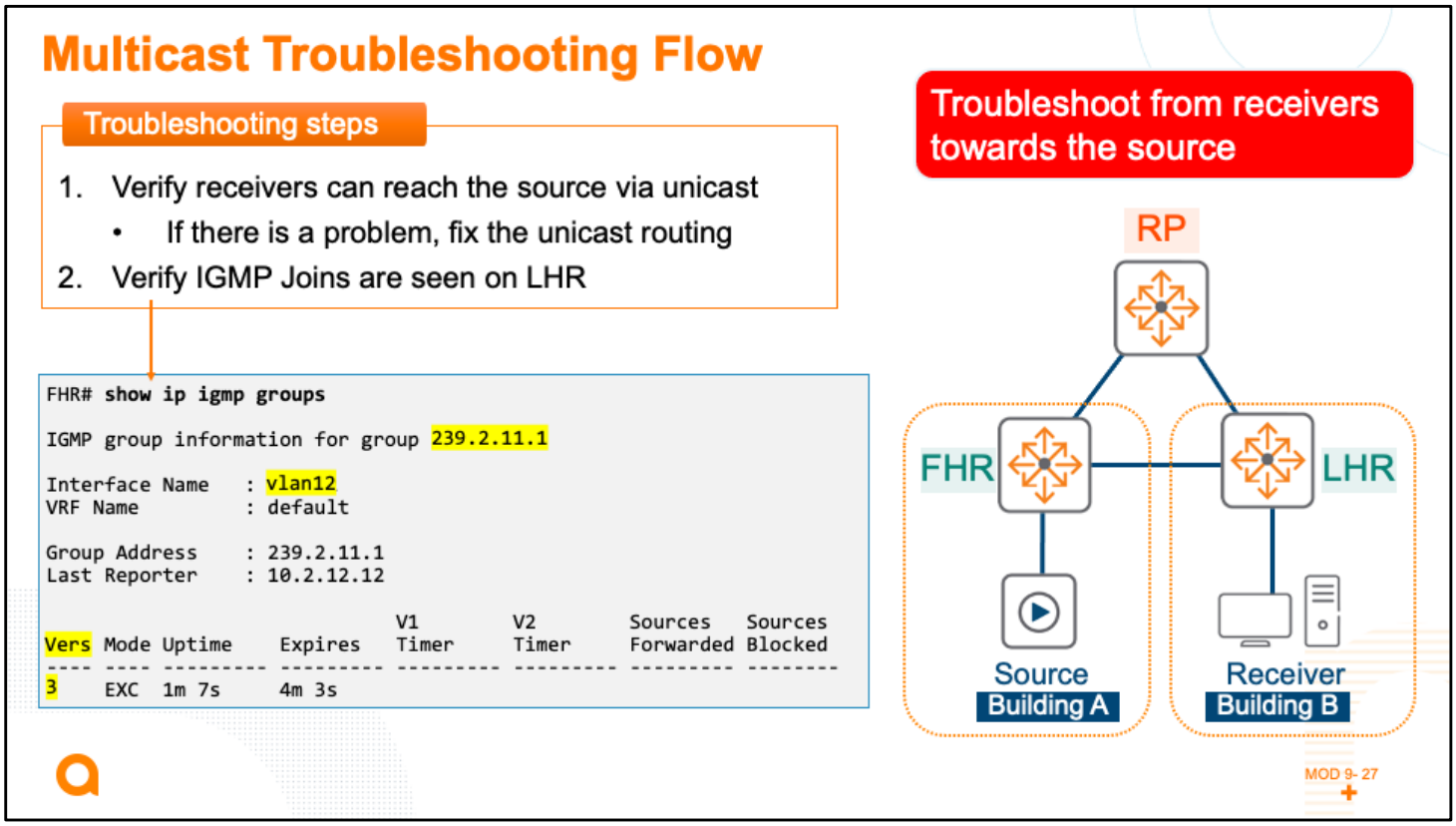
Enable IGMP on L3 interfaces towards receiver subnets

On VSX switches with VSX LAGs towards other PIM routers, enable PIM active-active to speed up switch failover

- Switch acting as a proxy-DR (Designated Router) monitors DR failure
- Both DR and proxy DR maintain the same multicast tables and build the multicast tree
- Upon detecting DR failure, proxy-DR switches role to DR, forwards traffic downstream to PIM router on VSX LAG

Enable IGMP/MLD snooping on proper VLANs

Note. Step 7 is only needed if the switch does not have a Layer 3 SVI interface where IGMP has not been enabled.



To simplify your trobleshooting process you should always start from the receivers and move towards the source.

Verify the unicast routing. Remember that multicast routing overlays the unicast routing, this means that your first point to toubleshoot a multicast network is to verify if that unicast routing is working properly. Ping and traceroute can be used in this phase, verify that communication from any receiver to its Designated Router is complete; communication between LHR and FHR to the RP must be successful as well.

Verify IGMP Joins are seen on LHR. IGMP Join and Prune messages should be correctly received by the Querier. Use the show ip igmp groups command to verify this information.

Multicast Troubleshooting Flow

Troubleshooting steps

3. Verify IGMP and PIM are enabled on LHR
   - `show ip igmp interface <interface-id>`
   - `show ip pim interface <interface-id>`

```
LHR# show ip igmp interface vlan12

IGMP Configured Version   : 3
IGMP Operating Version    : 3
Querier State             : Querier
Querier IP [this switch]  : 10.2.12.1
Querier Uptime            : 12m 39s
Querier Expiration Time   : 1m 1s
IGMP Snoop Enabled on VLAN : False

Active Group Address   Vers Mode Uptime    Expires
---------------------- ---- ---- --------- ---------
239.2.11.1             3    EXC  0m 34s    3m 46s
```

```
LHR# show ip pim interface vlan12

PIM Interfaces

VRF: default

Interface  : vlan12
Neighbor count : 0
IP Address : 10.2.12.1/24
Mode       : sparse

Designated Router : 10.2.12.1
Proxy DR                : false
Hello Interval (sec)  : 30
Hello Delay (sec)     : 5

Override Interval (msec)  : 2500     Lan Prune Delay     : Yes
Propagation Delay (msec)  : 500      DR Priority         : 1
Neighbor Timeout          : 0
```

MOD 9- 28

Verify IGMP and PIM are enabled on LHR. Use the show ip igmp interface <interface-id> command to verify the state of IGMP. Use show ip pim interface <interface-id> to can verify the state of PIM. Remember that PIM must be enabled on all the interfaces where the multicast traffic is going to be flooded.

Multicast Troubleshooting Flow

Troubleshooting steps

4. Verify IGMP snooping is active on expected ports
   - `show ip igmp snooping vlan <vlan-id> group <group-id>`

```
LHR# sh ip igmp snooping vlan 12 group 239.2.11.1

IGMP ports and group information for group 239.2.11.1

VLAN ID   : 12
VLAN Name : VLAN12

Group Address : 239.2.11.1
Last Reporter : 10.2.12.12
Group Type    : Filter

                                                    V1        V2        Sources   Sources
Port                    Vers Mode Uptime    Expires   Timer     Timer     Forwarded Blocked
----------------------- ---- ---- --------- --------- --------- --------- --------- --------
1/1/4                   3    EXC  1d 17h 34m 3m 47s                       0         0
```



MOD 9- 29

Verify IGMP snooping is active on expected ports. In Layer 2 devices, this step is important for efficient multicast traffic flow. To verify whether the receiver's port has been enabled for multicast traffic, use show ip igmp snooping vlan <vlan-id> group <group-id>.

Multicast Troubleshooting Flow

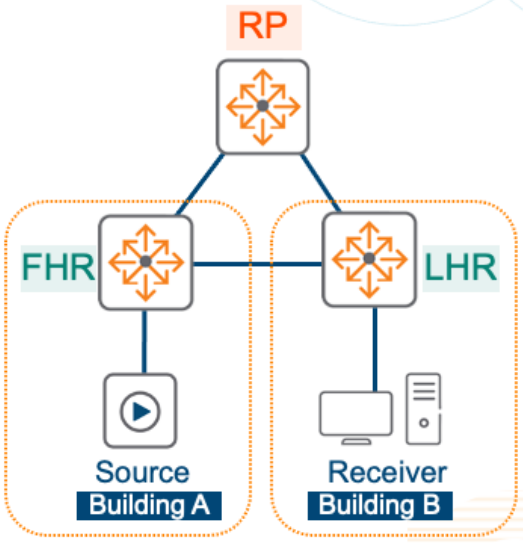
Troubleshooting steps

5. Verify multicast routes are seen on LHR
   - `show ip mroute`

```
LHR#  show ip mroute
IP Multicast Route Entries

VRF : default
Total number of entries : 1

Group Address            : 239.2.11.1
Source Address           : 10.2.11.10
Neighbor                 : 10.2.40.4
Incoming interface       : vlan40
Outgoing Interface List :
Interface       State
-----------     ----------
vlan12          forwarding
```



MOD 9- 30

To verify that multicast routes are seen on LHR, use the show ip mroute command.

Multicast Troubleshooting Flow

Troubleshooting steps

6. Verify PIM is enabled on LHR Uplinks
   - `show ip pim interface`
7. Verify all PIM neighbors are up as expected
   - `show ip pim neighbors`

```
LHR# show ip pim interface

<<Omitted output>>

Interface  : vlan44
Neighbor count : 1
IP Address : 10.2.44.5/24
Mode       : sparse

Designated Router : 10.2.44.5
Proxy DR              : false
Hello Interval (sec)  : 30
Hello Delay (sec)     : 5

Override Interval (msec)  : 2500     Lan Prune Delay     : Yes
Propagation Delay (msec)  : 500      DR Priority         : 1
Neighbor Timeout          : 95
```

```
LHR# show ip pim neighbor

PIM Neighbor

VRF                          : default
Total number of neighbors : 2

IP Address                   : 10.2.44.3
Interface                    : vlan44
Up Time (HH:MM:SS)           : 1 days 14:19:46
14:19:46
Expire Time (HH:MM:SS)       : 00:01:18
DR Priority                  : 1
Hold Time (HH:MM:SS)         : 00:01:45

IP Address                   : 10.2.40.4
Interface                    : vlan40
Up Time (HH:MM:SS)           : 1 days 14:19:46
14:19:46
Expire Time (HH:MM:SS)       : 00:01:41
DR Priority                  : 1
Hold Time (HH:MM:SS)         : 00:01:45
```

MOD 9- 31

– Verify PIM is enabled on LHR Uplinks. Remember that PIM Join and Prune messages should be sent to the RP therefore the uplink must properly establish PIM relationships. You can use the show ip pim interface command.

– Verify all PIM neighbors are up as expected. All intermediate routers between LHR and RP should be running properly PIM show ip pim neighbors command can be issued to verify the neighbor state.

Multicast Troubleshooting Flow

Troubleshooting steps

8. Verify RP is known
   - `show ip pim rp-set`
9. Verify BSR/C-RP if RP is known
   - `show ip pim rp-set`

```
LHR# show ip pim rp-set

VRF: default

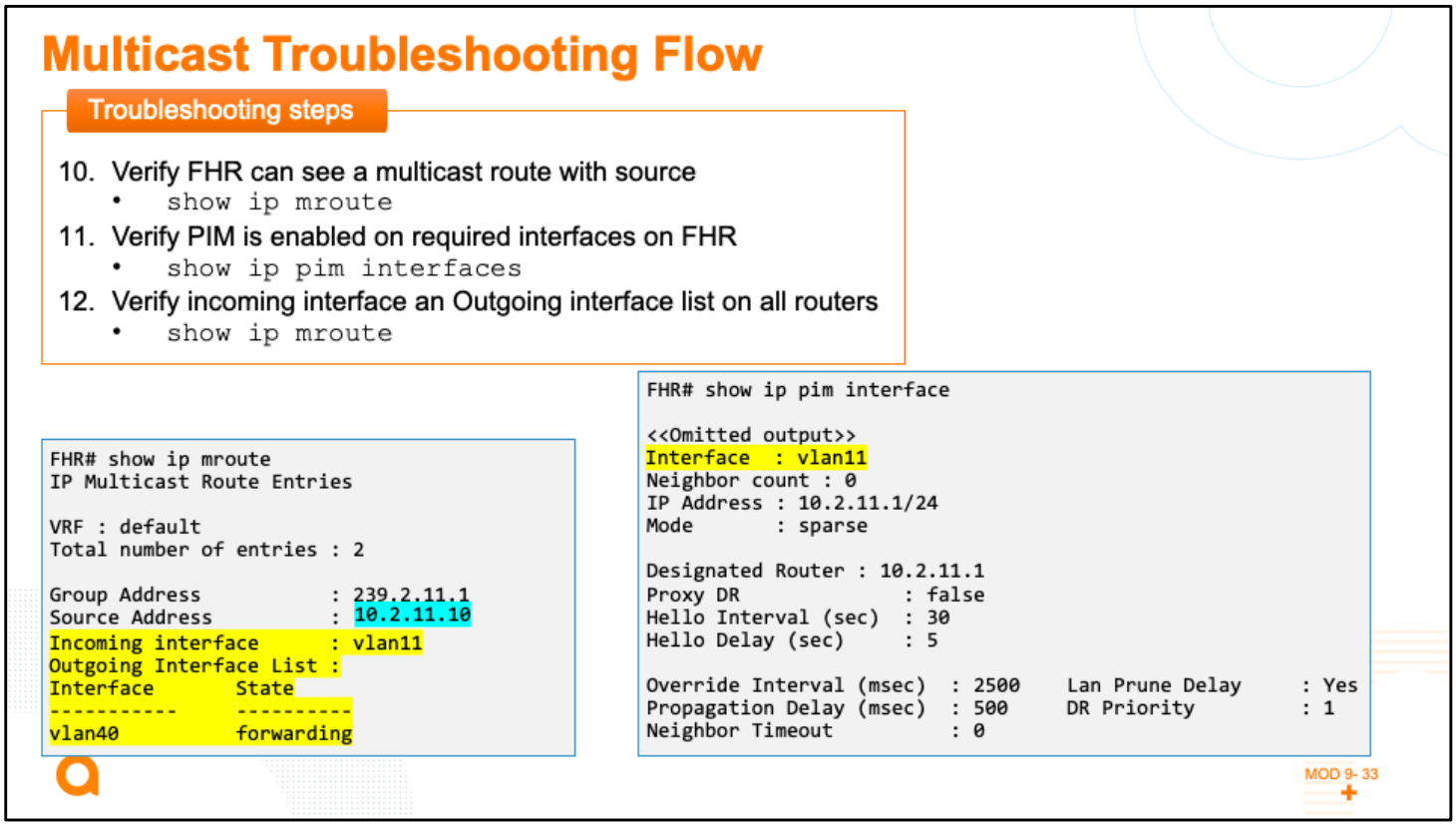
Status and Counters - PIM-SM Static RP-Set Information
Group Address    Group Mask       RP Address       Override
---------------  ---------------  ---------------  --------
224.0.0.0        240.0.0.0        10.2.0.2         No
239.2.0.0        255.255.0.0      10.2.0.2         No

Status and Counters - PIM-SM Learned RP-Set Information
Group Address    Group Mask       RP Address       Hold Time  Expire Time
---------------  ---------------  ---------------  ---------  -----------
224.0.0.0        240.0.0.0        10.2.0.4         150        110
239.2.0.0        255.255.0.0      10.2.0.4         150        110
```

MOD 9- 32

– Verify that the RP is known. You have learned that for PIM-SM it is critical that all PIM routers in the domain have the same RP-Set (Group-to-RP mapping). Verify this with the show ip pim rp-set command.

– If you are running the Bootstrap mechanism, then the BSR and RP must be available to all devices. Verify this with the show ip rp-set, show ip pim rp-set and show ip pim bsr commands.



– Verify if FHR can see a multicast route with source. The FHR must have a (S,G) entry, issue the show ip mroute command.
– Verify if PIM is enabled on required interface on FHR. Issue the show ip pim interfaces command.
– Verify incoming interface and Outgoing interface list on all PIM routers. The show ip mroute command can help on this task.

Another useful commands include:

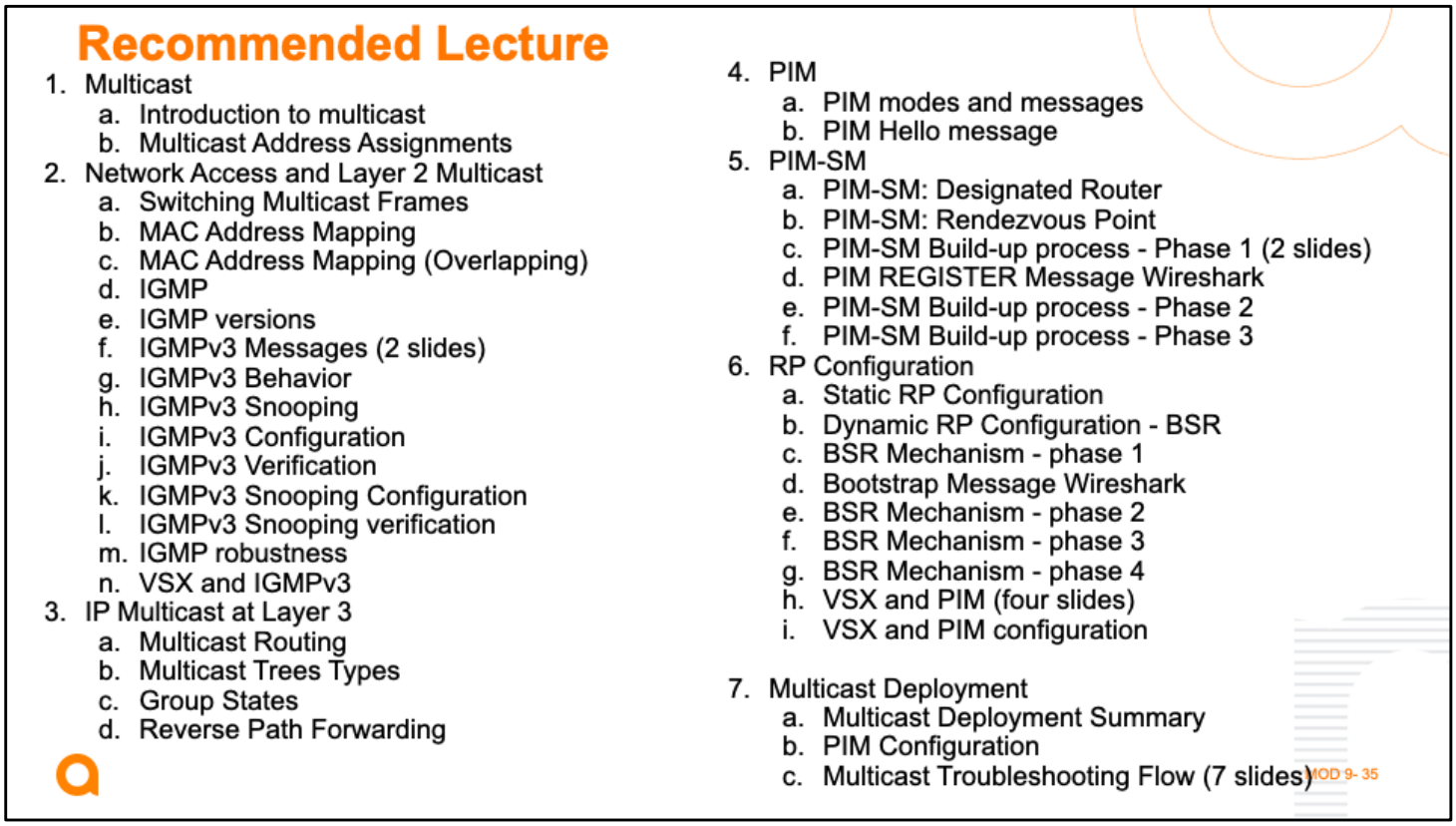
diagnostics
diag-dump pim basic
diag-dump igmp basic
diag-dump mcast-mtm basic

debug pim config
debug pim mroute
debug pim neighbor
debug pim packet
debug pim route

aruba
a Hewlett Packard Enterprise company

Additional Content

Module 09

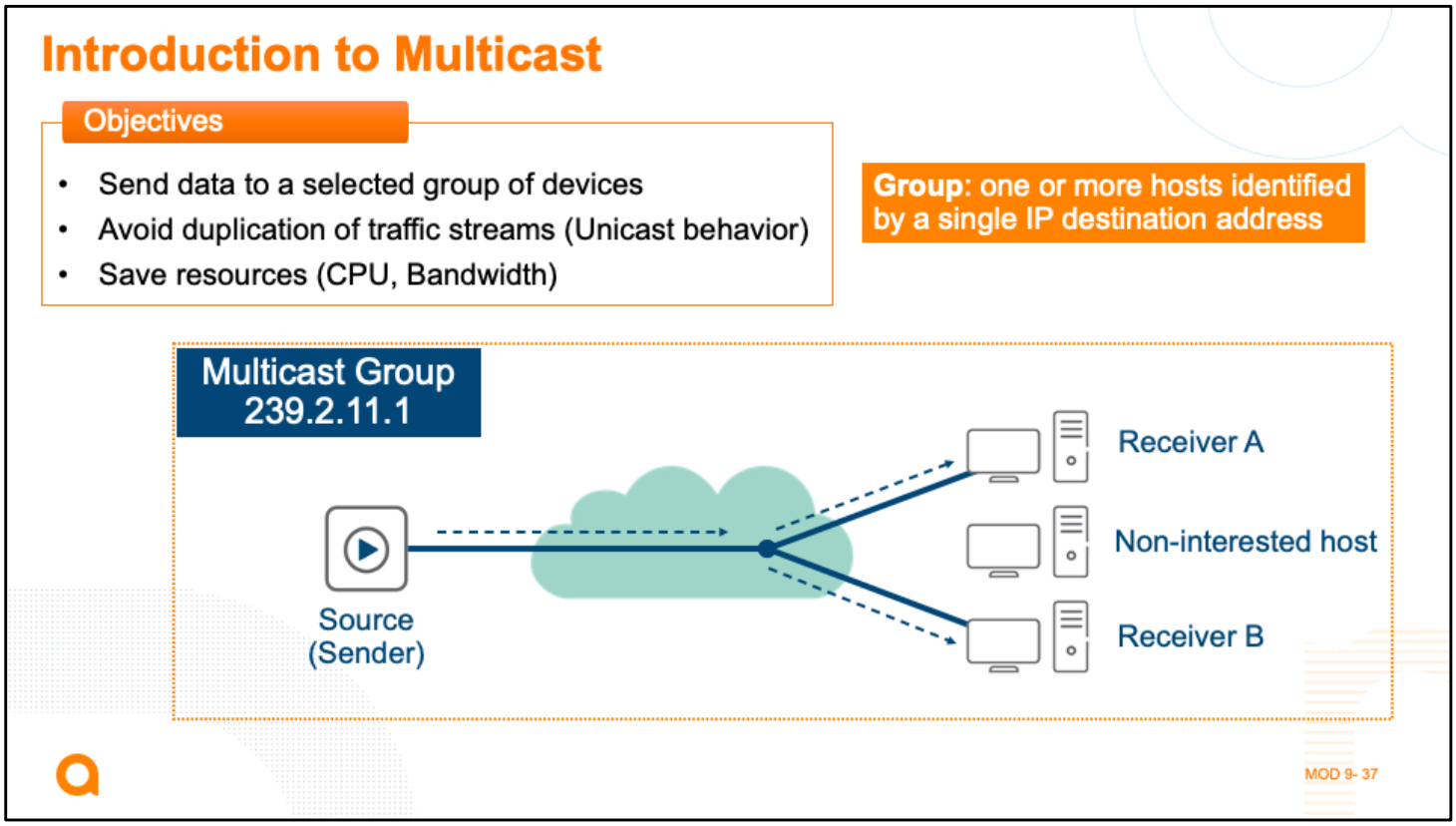


The additional material section includes information that will help you to understand AOS-CX. It is recommended to read this in the order presented in the slide.

aruba
a Hewlett Packard Enterprise company

Multicast

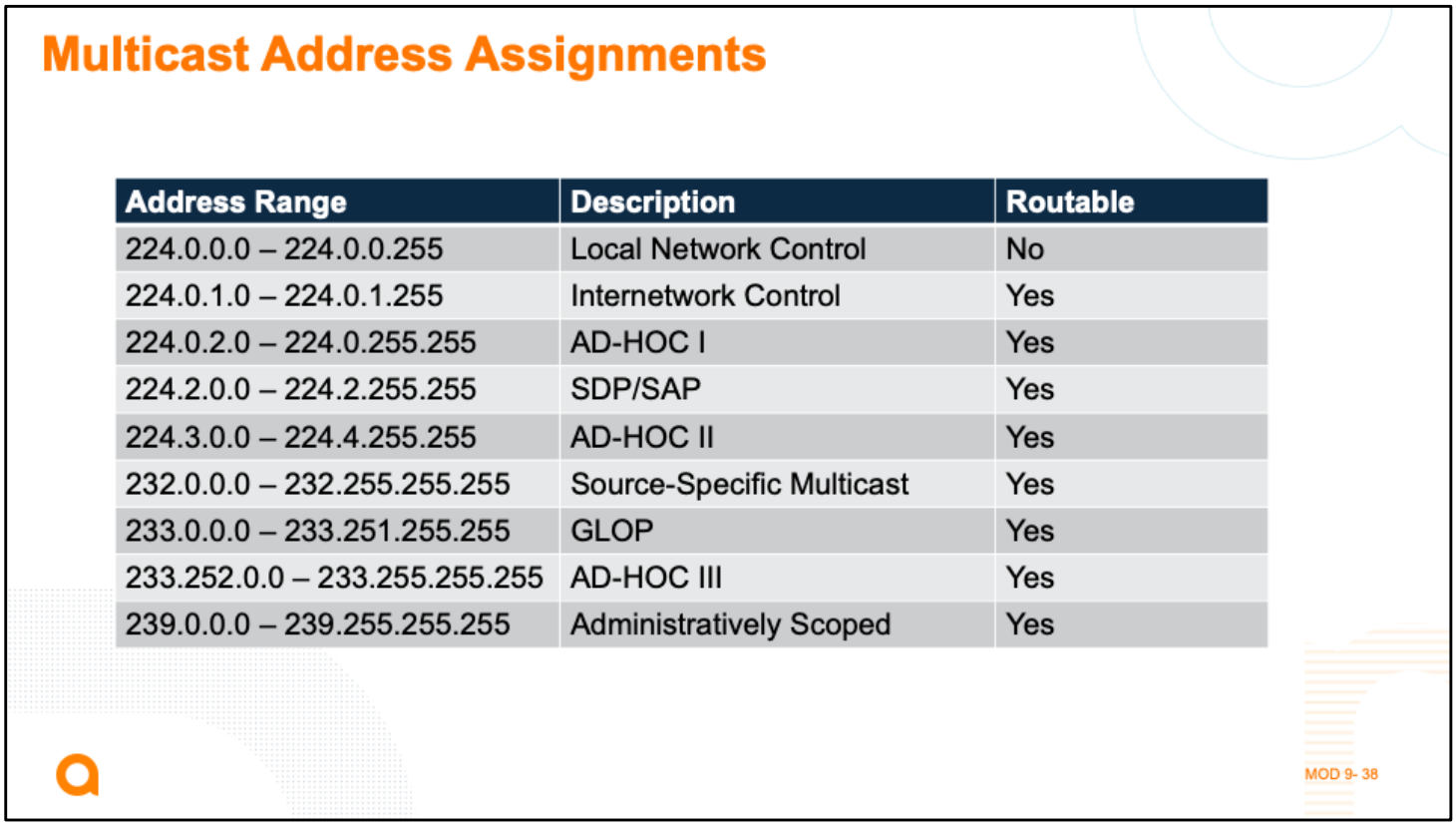
Introduction, IP address assignments



The purpose of multicast is to send information to a select group of devices and to avoid duplication of traffic streams. This saves network resources such as bandwidth.

A sender device generates information and advertises it to a group. Receivers join that group and listen to the information. A multicast group is defined as a set of zero or more hosts identified by a single IP destination address. The figure shows a multicast group represented by IP address 239.2.11.1.

IP multicast messages are transmitted from one-to-many, from many-to-many, and/or from many-to-one over an IP infrastructure that may span Layer 3 boundaries. The destination nodes (receivers) send join and leave messages that create an on-demand community of devices interested in receiving the multicast stream. Multicast optimally uses network resources by requiring the source to send a single packet stream, even though large numbers of receivers might need to receive the messages. The replication of messages takes place at the network node(s) or Layer 3 device(s) and diverges the stream to the receivers.



The group address simply identifies a group of nodes interested in a flow. The group address combined with the source IP address identifies the multicast flow.

The table identifies the Multicast Address Assignment:

Local Network Control block (224.0.0/24): The local control block is used for specific protocol control traffic. Router interfaces listen to but do not forward local control multicasts; for example, OSPF “all routers” (224.0.0.5). Assignments in this block are publicly controlled by IANA. You can find a complete list of Local Network Control address assignments at the IANA website (www.iana.org).

Internetwork Control block (224.0.1/24): The Internetwork Control block is for protocol control traffic that router interfaces may forward through the Autonomous System Number (ASN) or through the Internet. Examples include 224.0.1.1 Network Time Protocol (NTP), defined in RFC 4330, and 224.0.1.68 mdhcpdiscover, defined in RFC 2730. Internetwork Control group assignments are also publicly controlled by IANA.

AD-HOC blocks (I: 224.0.2.0–224.0.255.255, II: 224.3.0.0–224.4.255.255, and III:233.252.0.0–233.255.255.255): Traditionally assigned to applications that do not fit in either the Local or Internetwork Control blocks. Router interfaces may forward AD-HOC packets globally. Most applications using AD-HOC blocks require few group addresses (such as, for example, less than a /24 space). IANA controls any public AD-HOC Block assignments.

SDP/SAP block (224.2.0.0/16): The Session Description Protocol/Session Announcement

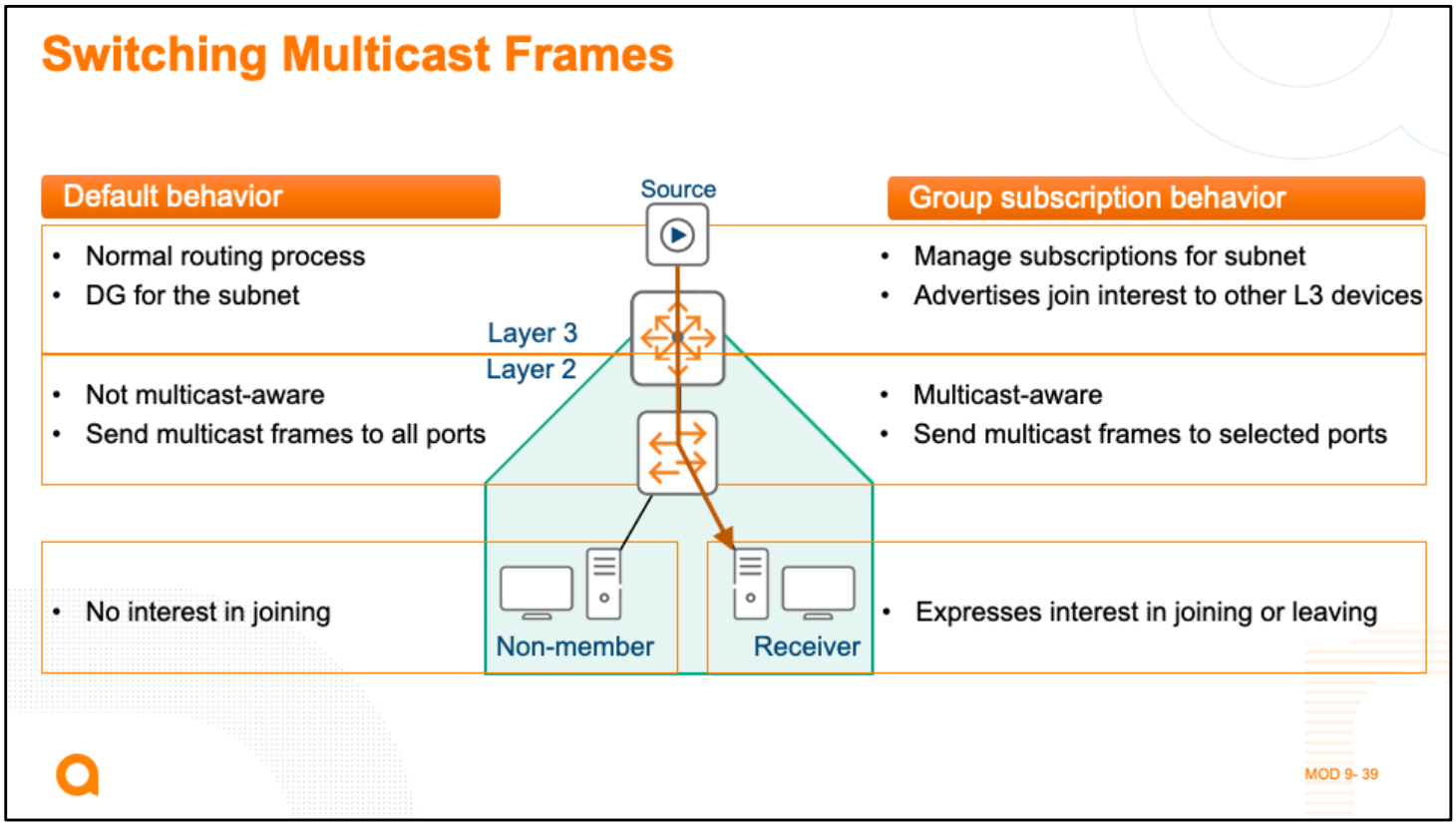
Protocol (SDP/SAP) block is assigned to applications that receive addresses through the SAP as described in RFC 2974.

Source-Specific Multicast block (232.0.0.0/8): SSM addressing is defined by RFC 4607. SSM is a group model of IP Multicast in which multicast traffic is forwarded to receivers from only those multicast sources for which the receivers have explicitly expressed interest. SSM is mostly used in one-to-many applications. No official assignment from IANA is required to use the SSM block because the application is local to the host; however, according to IANA policy, the block is explicitly reserved for SSM applications and must not be used for any other purpose.

GLOP block (233.0.0.0/8): These addresses are statically assigned with a global scope. Each GLOP static assignment corresponds to a domain with a public 16-bit autonomous system number (ASN), which is issued by IANA. The ASN is inserted in dotted-decimal into the middle two octets of the multicast group address (X.Y). An example GLOP assignment for an ASN of X.Y would be 233.X.Y.0/24. Domains using an assigned 32-bit ASN should apply for group assignments from the AD-HOC III Block. Another alternative is to use IPv6 multicast group addressing. Because the ASN is public, IANA does not need to assign the actual GLOP groups. The GLOP Block is intended for use by public content, network, and Internet service providers. IANA considers GLOP addressing to be experimental, and 233.252–255.0.0 is reserved.

Administratively Scoped block (239.0.0.0/8): Administratively Scoped addresses are intended for local use within a private domain as described by RFC 2365. These group addresses serve a similar function as RFC 1918 private IP address block (such as, for example, 10.0.0.0/8 or 172.16.0.0/16 blocks). Network architects can create an address schema using this block that best suits the needs of the private domain and can further split scoping into specific geographies, applications, or networks.

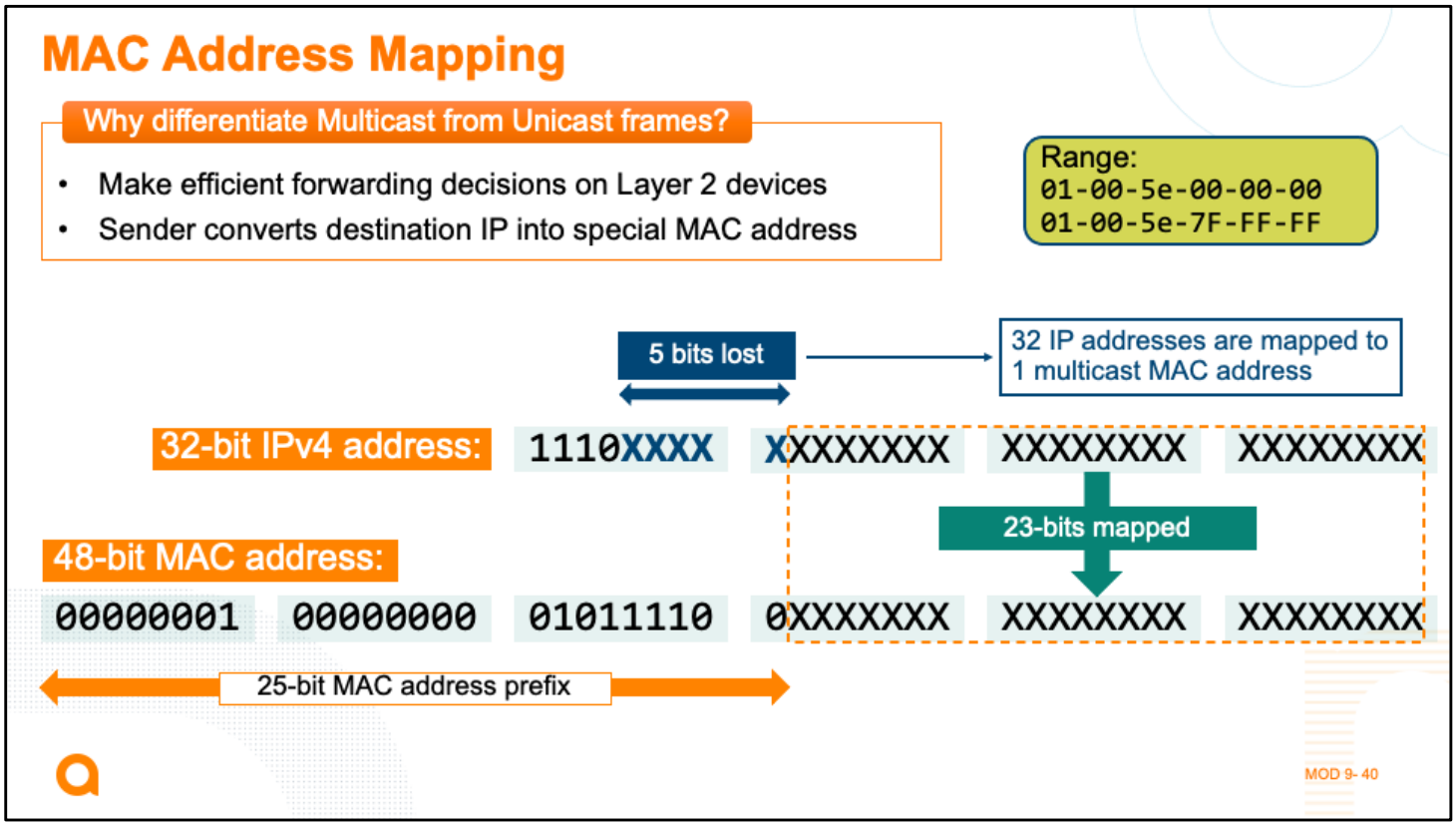
Note: “GLOP”, surprisingly, is not an acronym. It does not stand for anything. The authors of the RFC needed a better way to refer to “that addressing method where you put your AS in the middle two octets”. David Meyers started calling it “GLOP” and the name stuck.



The major difference between a unicast and multicast network is that, in a unicast network all traffic is fully permitted to the host, assuming no filter is applied. In a multicast network, the host must advertise its intention to join the network.

Without multicast-aware Layer 2 protocols, all hosts on a given Layer 2 segment will receive multicast packets for any groups joined by a host on that segment. This is not efficient, since non-interested endpoints must process each frame, only to drop it. When Layer 2 devices are multicast-aware they properly send multicast traffic to only the ports where we find interested hosts.

The gateway is the network demarcation between Layers 2 and 3 and is the most appropriate device to manage host group membership for the larger network. This device receives these management messages and adds host segment interfaces to the local multicast table. The gateway communicates group interest for a multicast group to other Layer 3 devices using multicast routing (PIM).



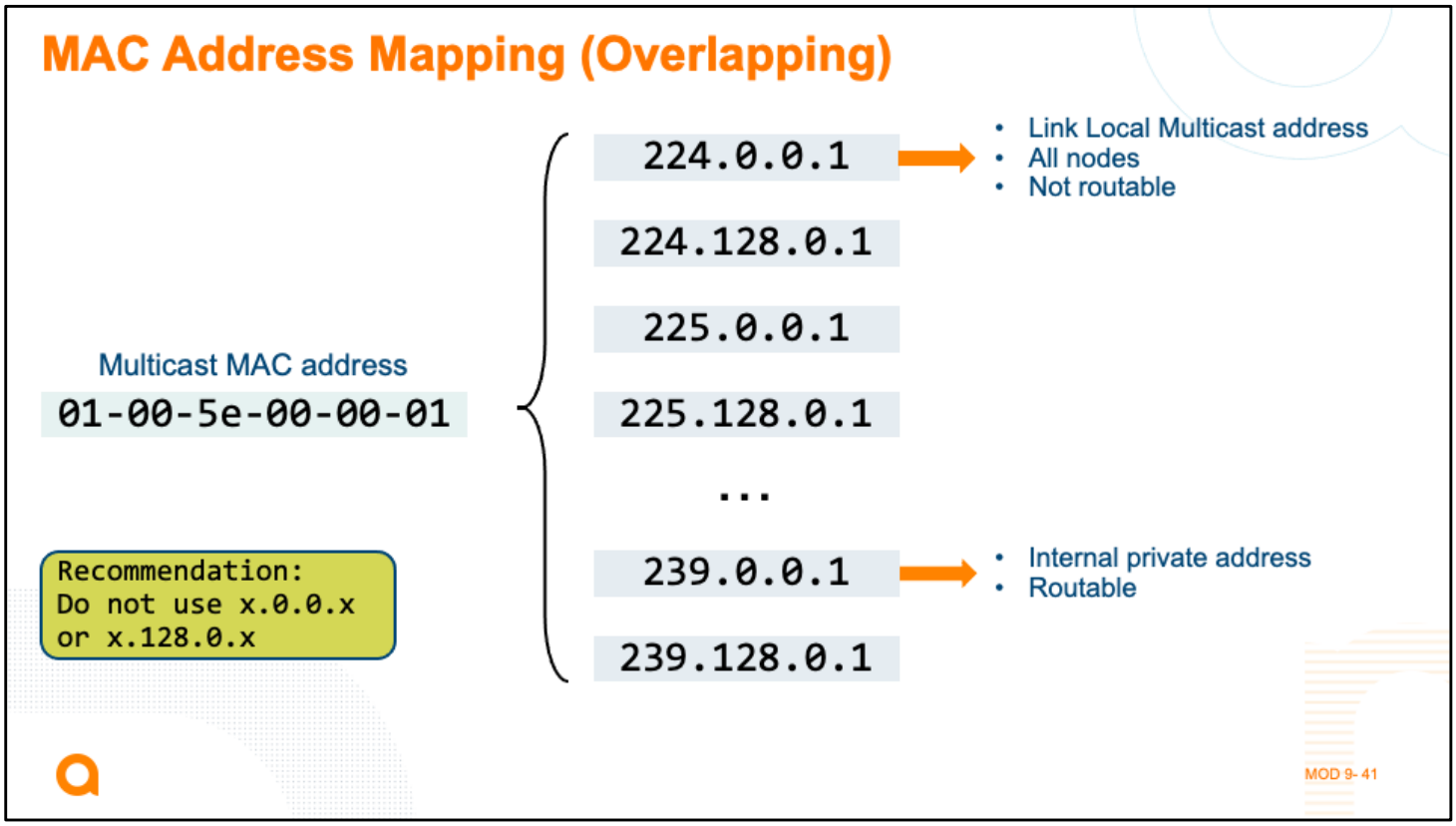
To optimize network resources, an Ethernet switch must understand multicast addressing. This is where the magic happens. The sending device must convert the destination IP multicast address into a special MAC address as follows:

The high-order 25 bits is the official reserved multicast MAC address range from 0100.5E00.0000 to 0100.5E7F.FFFF (RFC-1112). These bits are part of the organizational unit identifiers (OUI).

The lower-order 23 bits of the destination IP multicast address are mapped to the lower-order 23 bits of the MAC address.

A switch can use this calculated multicast MAC address to distinguish a frame as a multicast and make efficient forwarding decisions.

We miss 5 bits of mapping information: 25 = 32. This means we will map 32 multicast IP addresses to 1 multicast MAC address.

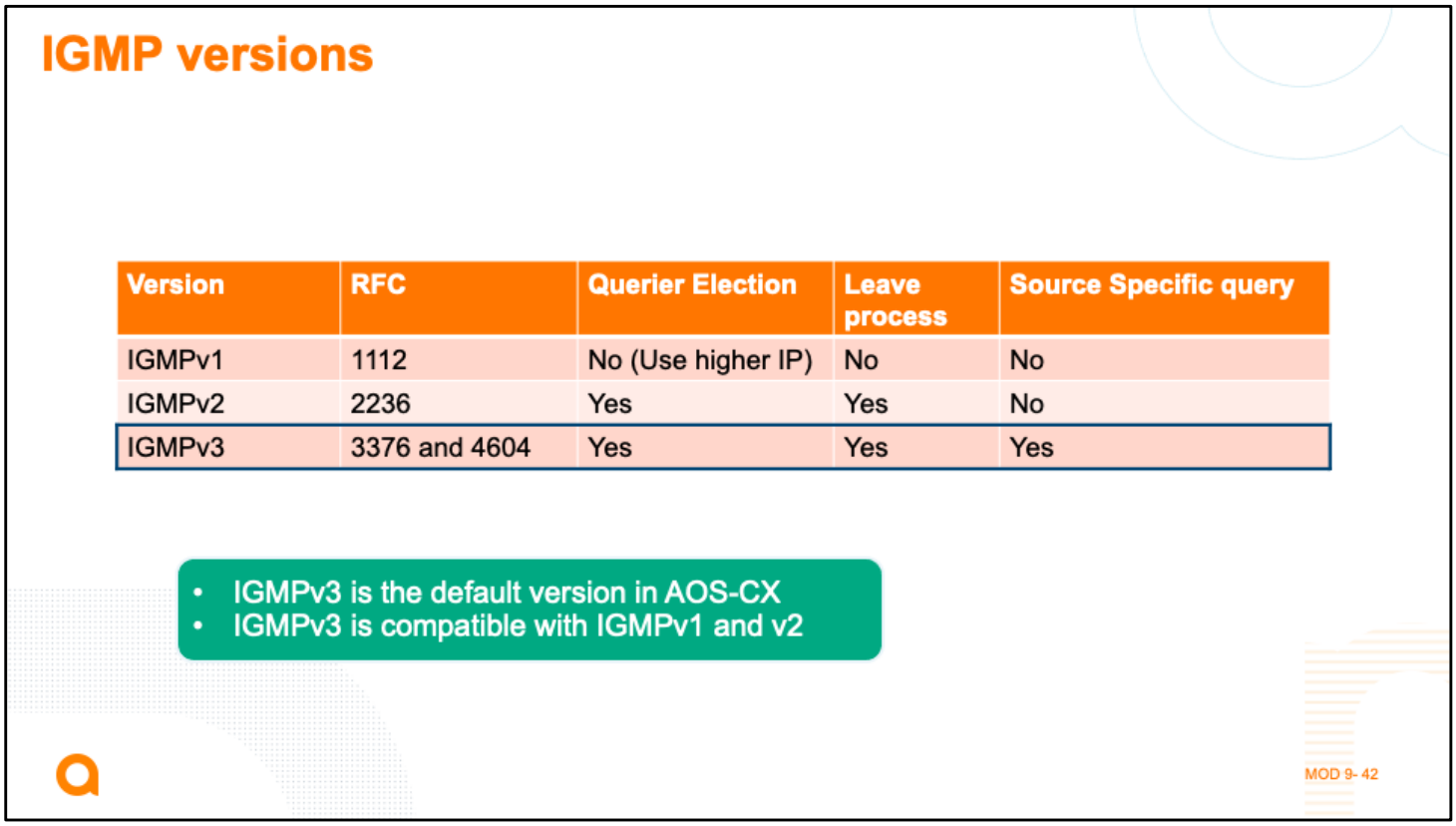


The multicast IP addresses above all map to the same multicast MAC address (01-00-5E-00-00-01).

This can cause some network problems. For example, a host that listens to the 239.0.0.1 multicast IP address will configure its network card to listen to MAC address 01-00-5E-00-00-01. At the same time, any device could generate traffic to 224.0.0.1 (all-nodes in the link) which is mapped to the same MAC address hosts will have to look at the IP address (Process Layer 3 header) of the received frame to see if it's for 239.0.0.1 and discard frames that are meant for 224.0.0.1. This process is not efficient.

As a recommendation do not use x.0.0.x or x.128.0.x, since these address will overlay with the Link-Local Multicast address scope.

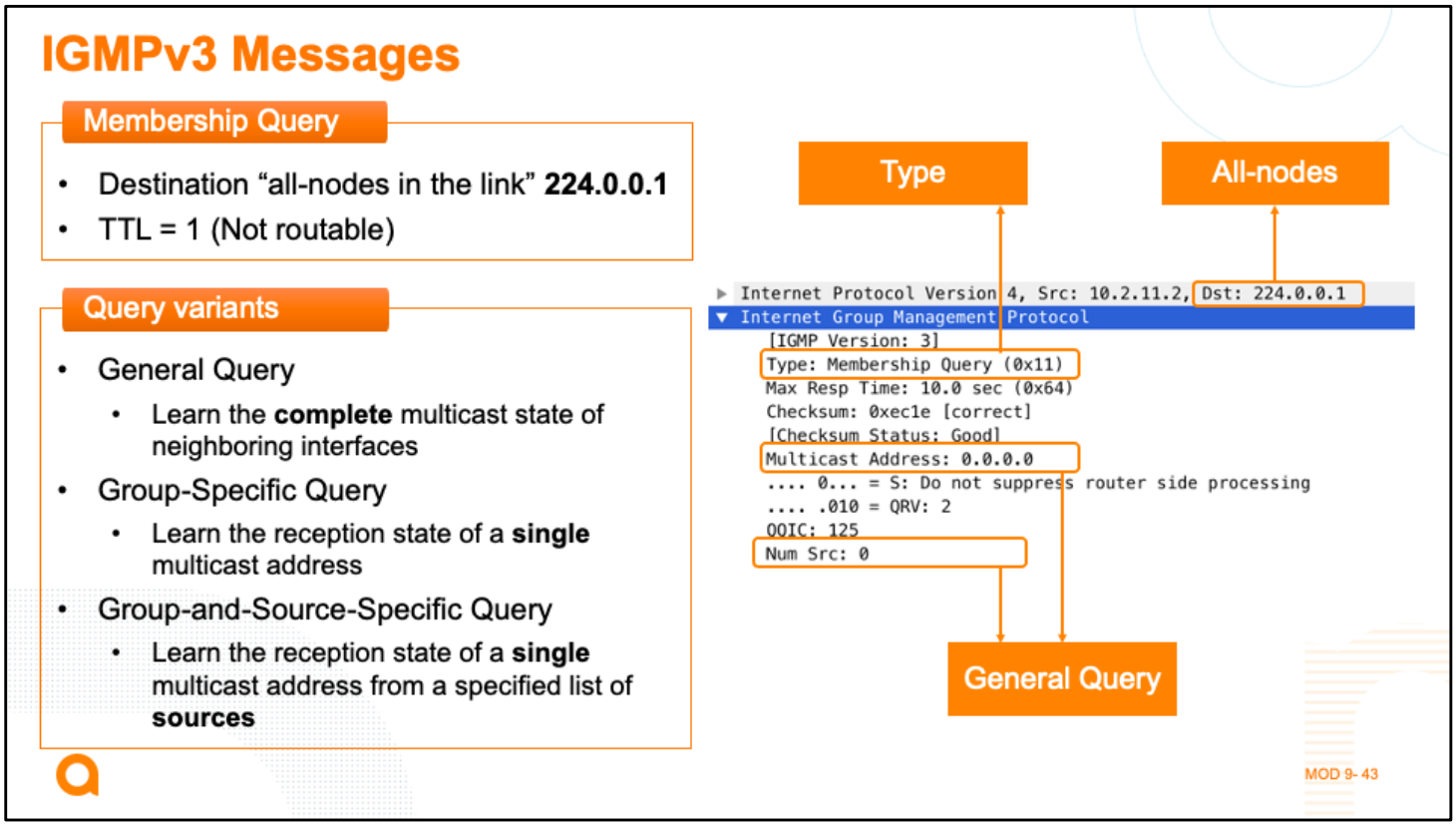
Note. Some network devices does IGMP filtering based on MAC address. This implies that you should be careful when picking your multicast groups that you do not overlay the mac-addresses as to avoid odd interactions at layer-2.



IGMPv1: Defined in RFC 1112, IGMPv1 offers a basic query-and-response mechanism to determine which multicast streams should be sent to a particular network segment. IGMPv1 has no mechanism for a host to signal that it wants to leave a group. When a host using IGMPv1 leaves a group, the router will continue to send the multicast stream until the group times out. As you can imagine, this can create a large amount of multicast traffic on an interface if a host joins groups very quickly. IGMPv1 also does not elect a querier. If there are multiple queriers (routers) on the subnet, a designated router (DR) is elected using PIM to avoid sending duplicate multicast packets. The elected querier is the router with the highest IP address. IGMPv1 is rarely used in modern networks.

IGMPv2: Defined in RFC 2236, this newer version made improvements over IGMPv1. One of the most significant changes was the addition of a leave process. A host using IGMPv2 can send a leave-group message to the querier indicating that it is no longer interested in receiving a particular multicast stream. IGMPv2 also added group-specific-queries. This feature allows the querier to send a message to the host(s) belonging to a specific multicast group. The querier election process determines the querier based on a priority.

IGMPv3: Defined in RFC 3376 and 4604, this version 3 adds support for "source filtering“ - the ability for a system to report interest in receiving packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. Multicast routing protocols can use this information to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.



To determine group membership, the querier (router) sends a message to every host on the subnet. The querier must maintain a list of hosts in the subnet interested in multicast flows. To do this, the query message is multicast to the “all-hosts” multicast address of 224.0.0.1. This message is created with a TTL of 1 which indicates its non-routable nature.

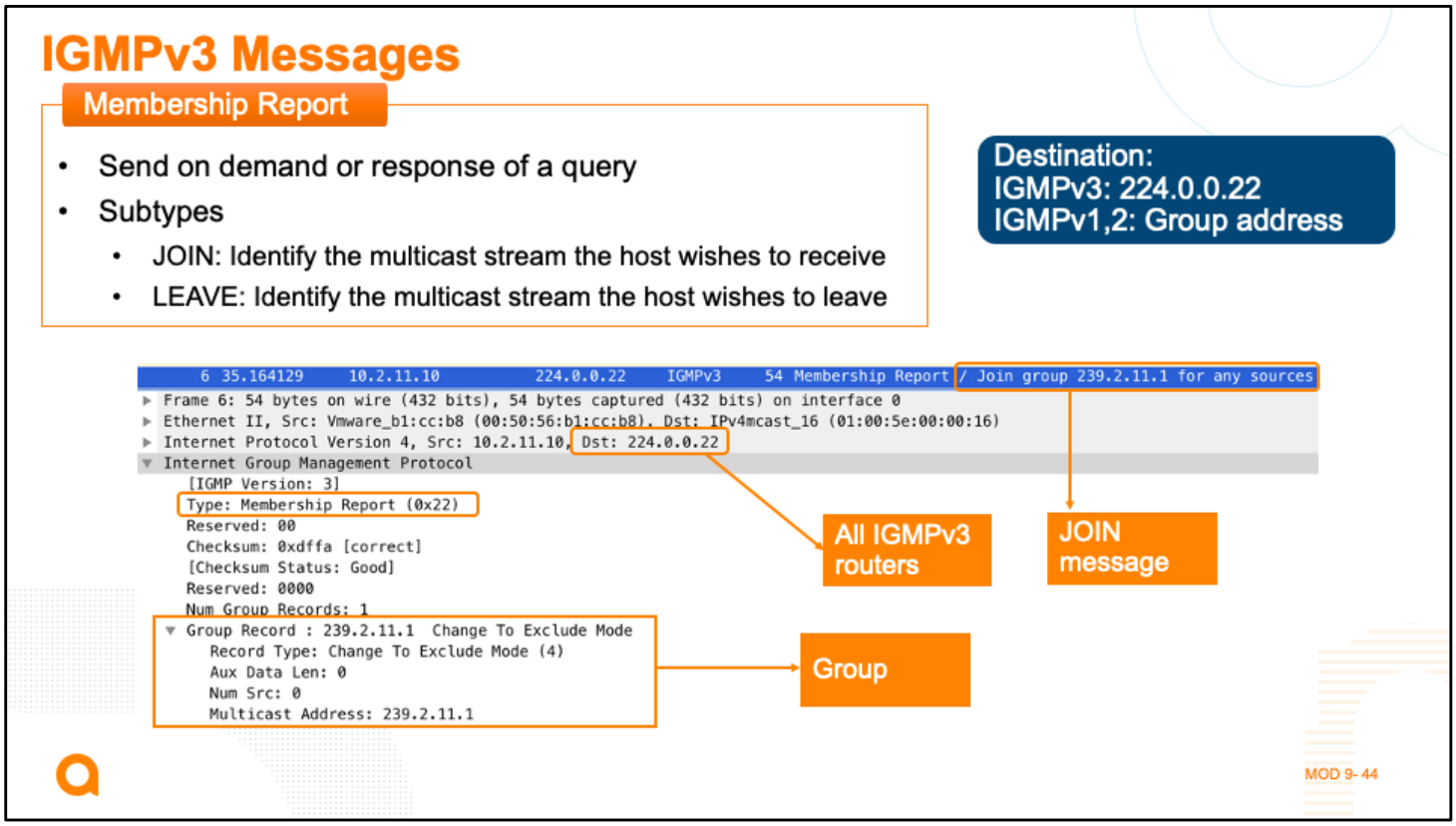
The membership Query in IGMPv3 has the following variants:

General Query: Sent by Multicast router to learn the complete multicast reception state of the neighboring interfaces. In a General Query, both the Group Address field and the Number of Sources (N) field are zero.

Group-Specific Query: Sent by Multicast router to learn the reception state, with respect to a single multicast address of the neighboring interfaces. In a Group-Specific Query, the Group Address field contains the multicast address of interest, and the Number of Sources (N) field contains zero.

Group-and-Source-Specific Query: Sent to learn if any neighboring interface desires reception of packets sent to a specified multicast address, from any of specified list of sources. In a Group-and-Source-Specific Query, the Group Address field contains the multicast address of interest, and the Source Address [i] fields contain the source address(es) of interest.

In the Wireshark capture you can see a general query, identifiable as such because the multicast address is set to 0.0.0.0 and the number of sources is 0.



Membership report messages are sent by IP endpoints to report (to neighboring routers) interest in joining or leaving a multicast stream. These messages are sent either in response to a membership query message or when the multicast application in the endpoint starts.

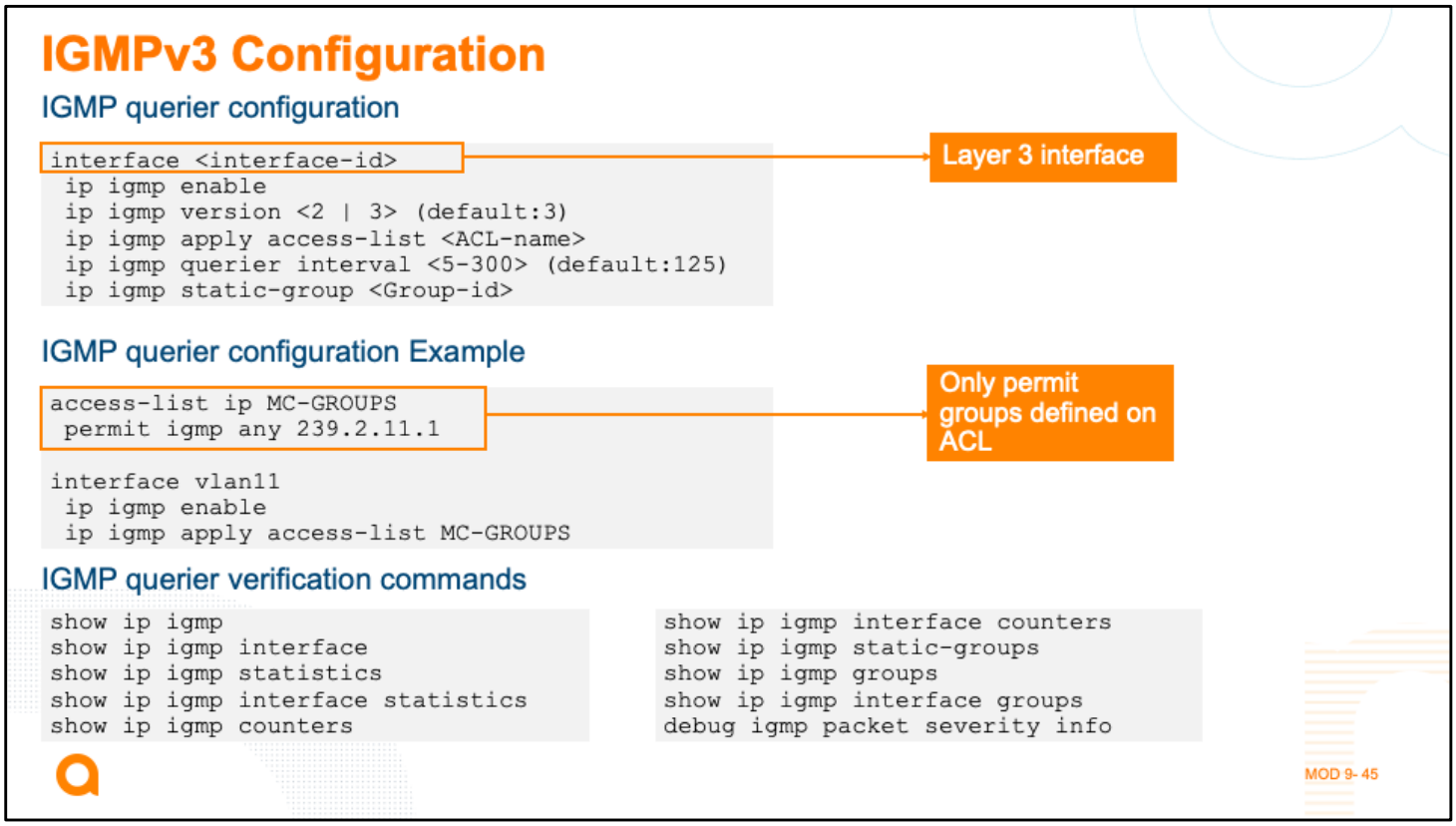
Version 3 Reports are sent with an IP destination address of 224.0.0.22, to which all IGMPv3-capable multicast routers listen. A system that operates in version 1 or version 2 compatibility mode sends version 1 or version 2 Reports to the multicast group specified in the Group Address field of the Report.

There are two subtypes of membership messages:

JOIN: Sent from host to querier to identify the multicast stream the host wishes to receive.

LEAVE: Sent from host to querier to indicate that the host shall longer be a member of a Multicast group.

The Wireshark capture shows a Membership Report where the endpoint (10.2.11.10) includes a JOIN message to receive packets from the group 239.2.11.1.



In AOS-CX IGMP snooping is disabled by default and version 3 is used by default.

IGMP querier commands are:

interface <interface-id>
 ip igmp enable
 ip igmp version <2 | 3> (default:3)
 ip igmp apply access-list <ACL-name>
 ip igmp querier interval <5-300> (default:125)
 ip igmp static-group <Group-id>

Notice that all IGMP commands are applied on the interface level.

IGMP querier filter configuration example:

access-list ip MC-GROUPS
 permit igmp any 239.2.11.1

```
interface vlan11
 ip igmp enable
 ip igmp apply access-list MC-GROUPS
```

You can see how only the group 239.2.11.1 is processed by the querier. Existing classifier commands are used to configure the ACL. If an IGMPv3 packet with multiple group addresses is received, it will only process the permitted group addresses based on the ACL rule set, and any existing joins will time out. If there is no match or if there is a deny rule match, the packet is dropped.

To view IGMP information, use the following commands.

To view IGMP configuration details and status, use: show ip igmp or use show ip igmp interface.

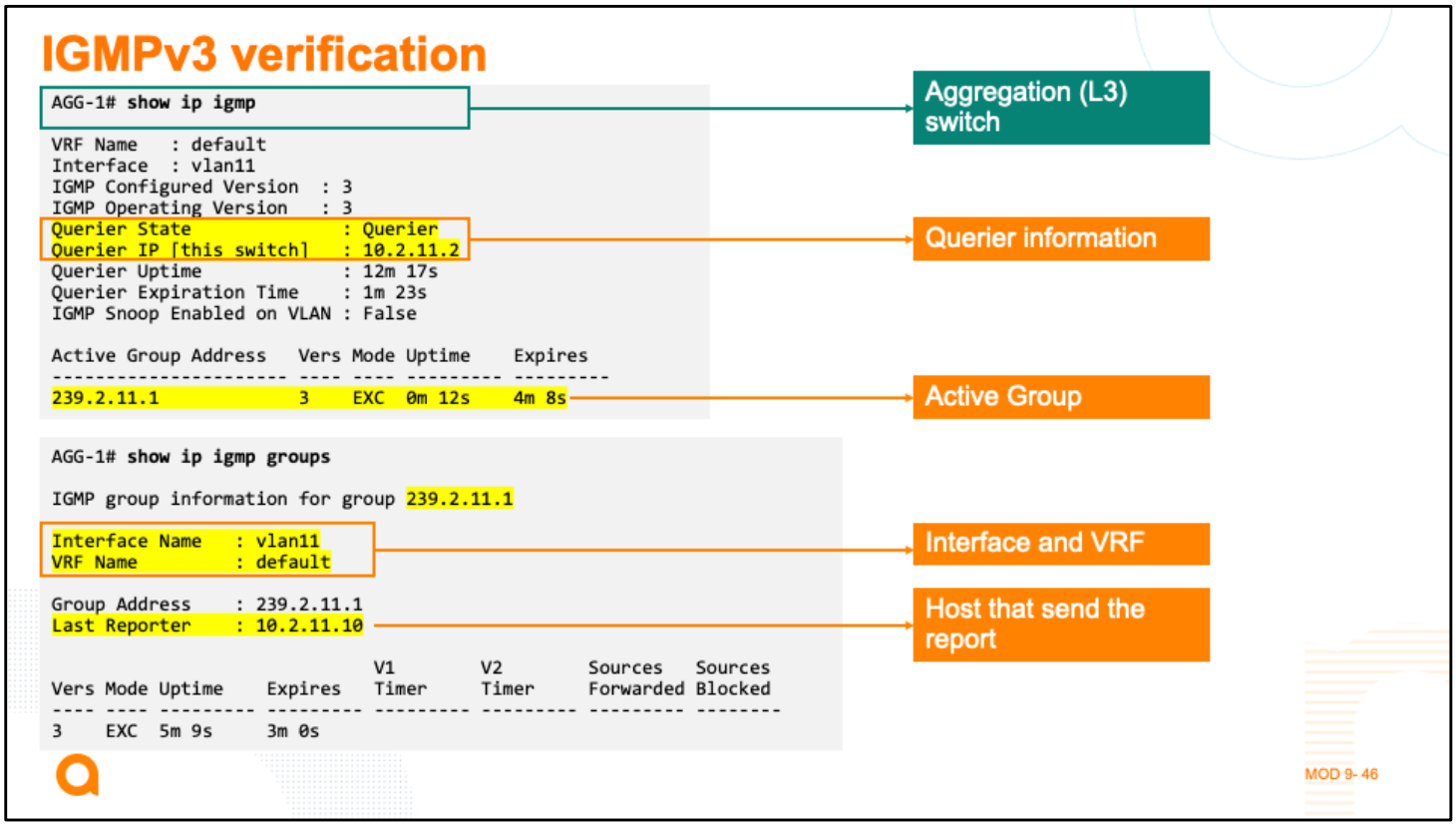
To view IGMP statistics and groups joined, use: show ip igmp statistics or use show ip igmp interface statistics.

To view IGMP counters, use: show ip igmp counters or use show ip igmp interface counters.

To view IGMP static groups, use: show ip igmp static-groups.

To view IGMP group information, use: show ip igmp groups or use show ip igmp interface groups.

To view IGMP group details for a specific group and source, use: show ip igmp group or use show ip igmp interface group. Optionally you can also display joined group details by VRF.



The show ip igmp command displays information about the querier. Notice that the output includes which device is running the querier role for the segment. This command includes information such as:

VRF Name

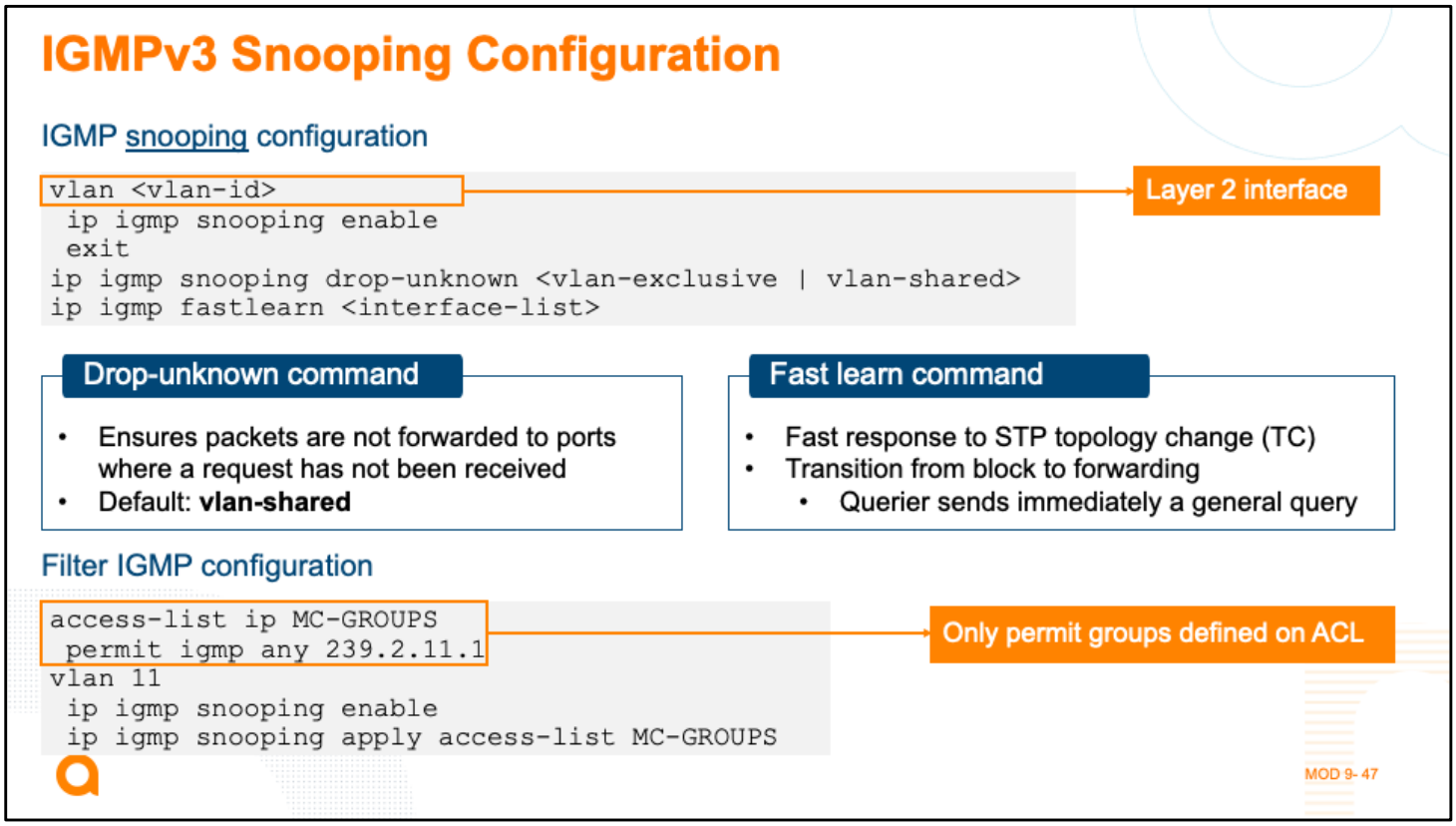
Layer 3 Interface

IGMP Configured Version (v3 by default)

Querier State

Querier IP address

The show ip igmp groups comand display information about the multicast groups including the interface and the VRF from where the switch receive these packets.



IGMP snooping configuration is done at the VLAN level, not at the VLAN interface.

To enable this feature, use the following commands:

vlan <vlan-id>

ip igmp snooping enable

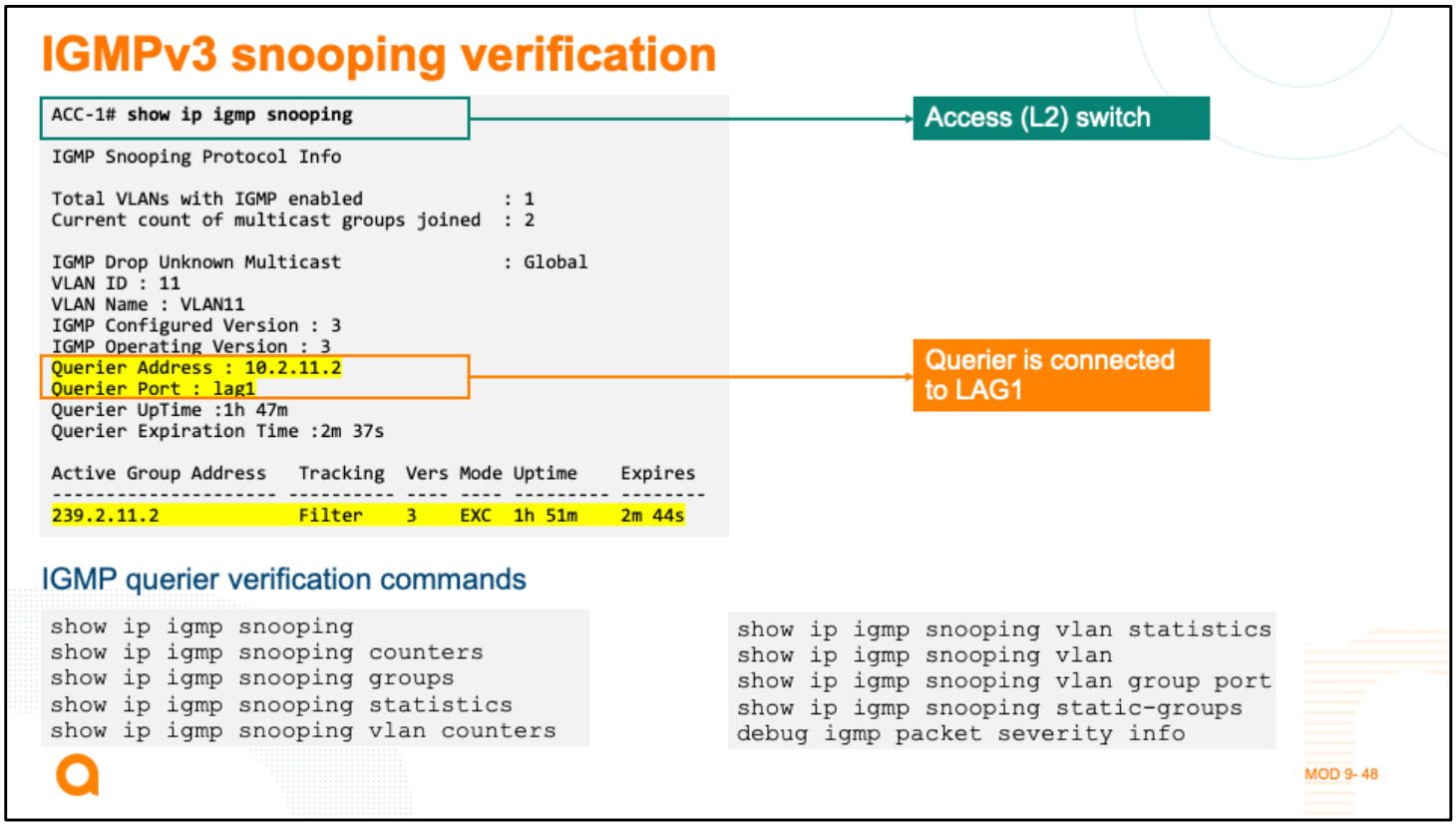
There are two global IGMP snooping commands that modify the global behavior of the switch:

Drop-unknown: Configures drop-unknown mode. While IGMP snooping is enabled, traffic is forwarded only to ports that made an IGMP request for the multicast. Drop unknown filters ensure that packets are not forwarded to ports that did not make a request for the traffic stream. This could either be a filter across all VLANs (vlan-shared) or per VLAN (vlan-exclusive). The default is vlan-shared.

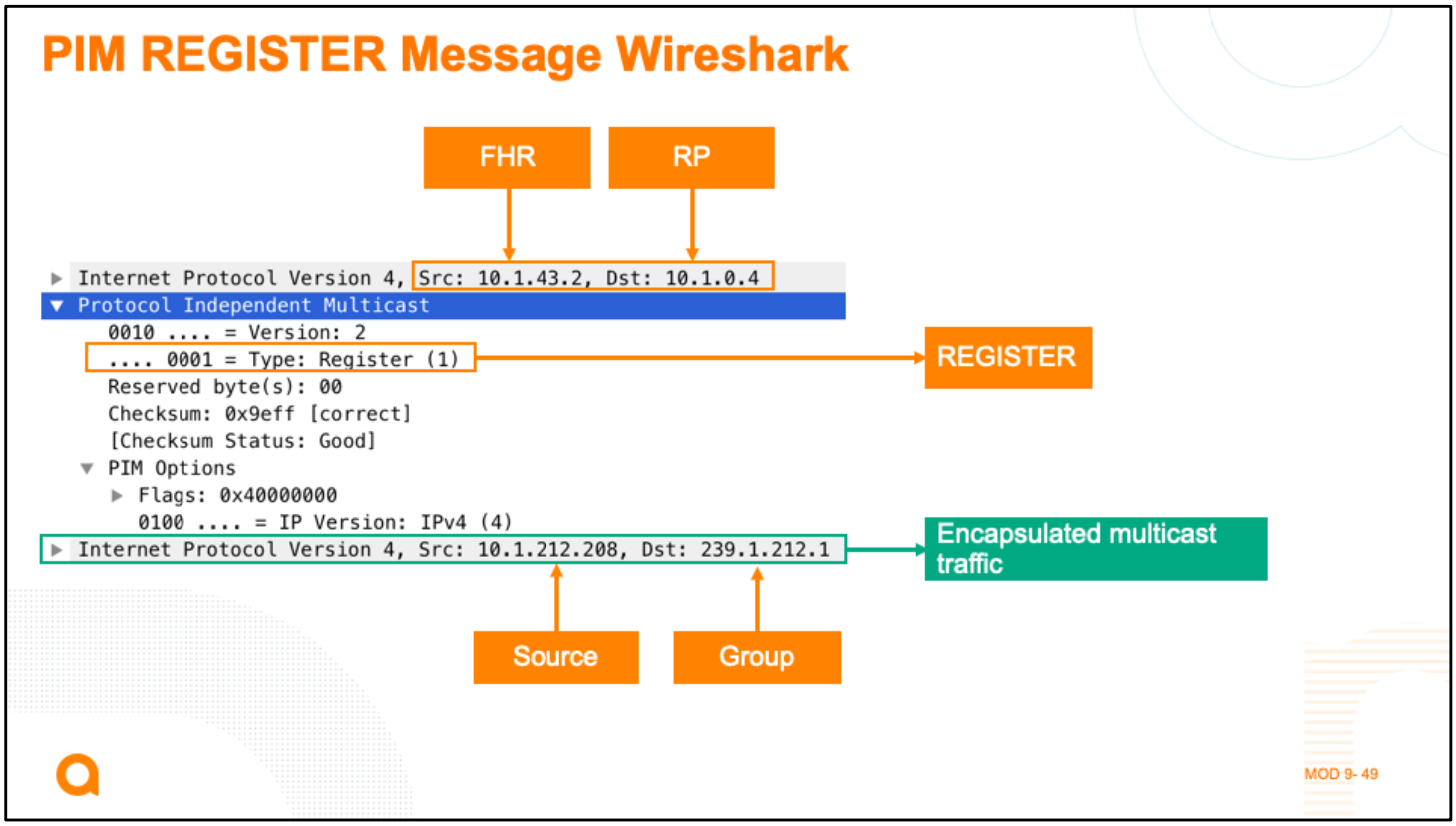
Fast learn: Enables the port to learn group information when receiving a STP topology change (TC) notification transitioning from block to forwarding STP state. By default, fast learn is not enabled on ports.

AOS-CX also allows filtering on the IGMP snooping process, the following example shows how only IGMP Join messages containing the group 239.2.11.1 are processed.

```
access-list ip MC-GROUPS
 permit igmp any 239.2.11.1
vlan 11
 ip igmp snooping enable
 ip igmp snooping apply access-list MC-GROUPS
```



The show ip igmp snooping command displays the information analyzed by the Layer 2 device. The example shows that the querier is connected to LAG1 from the ACC-1 switch perspective. This command also reveals information about the Active multicast group.

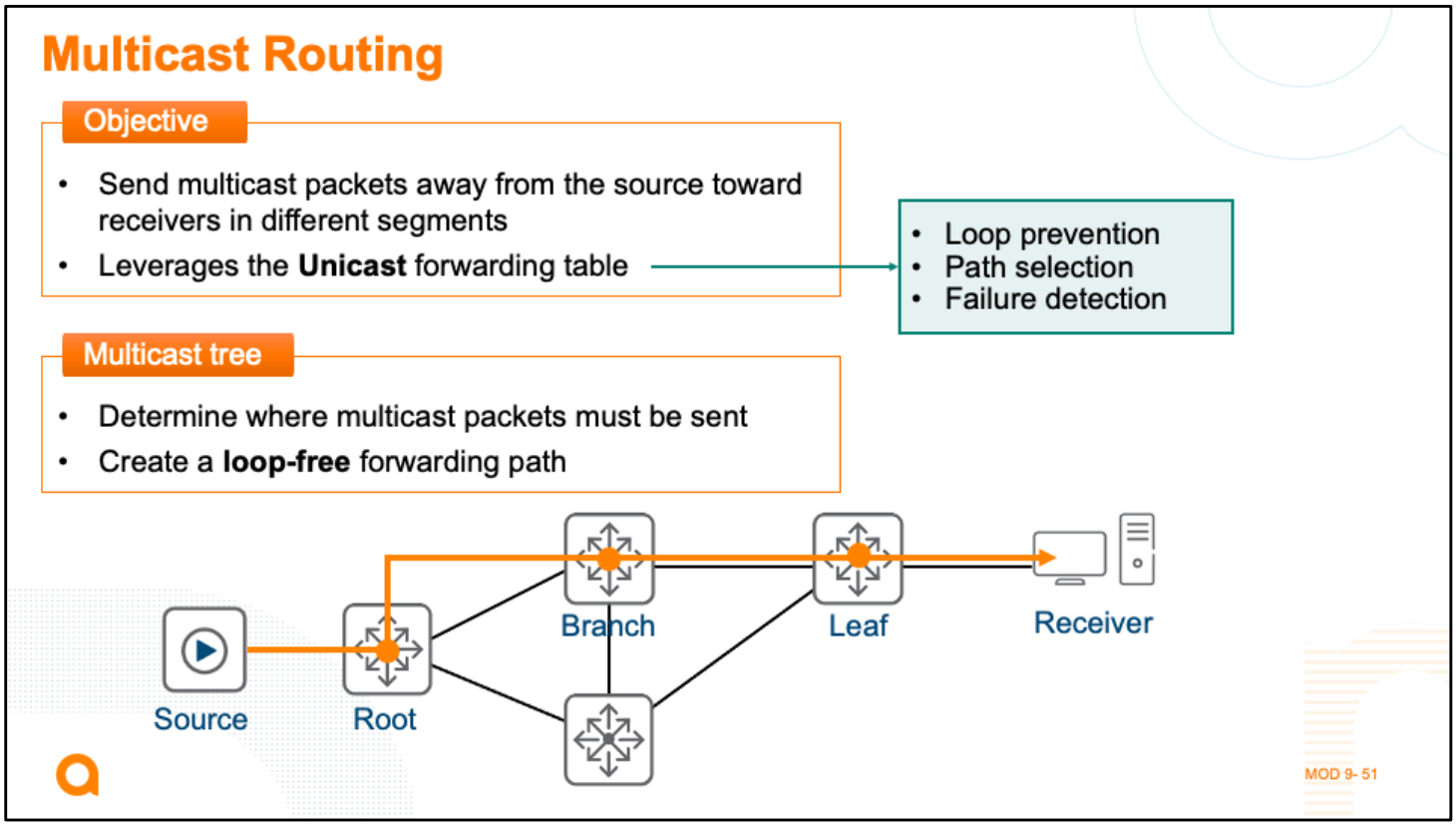


The packet capture in the figure shows a PIM REGISTER message. Pay special attention to the highlighted areas - the source and destination addresses at Layer 3, the PIM message type (Register) and at the bottom the encapsulated traffic.

aruba
a Hewlett Packard Enterprise company

IP Multicast at Layer 3

Multicast Routing, Multicast Tree Types, Group States and Reverse Path Forwarding



You know that regular unicast routing sends packets toward a destination. The objective of multicast routing is to send multicast messages away from the source toward all branches or segments of the network interested in receiving those messages. Multicast packets must always flow away from the source, and never back on a segment from where the transmission originated. This means that rather than tracking only destinations, multicast routers must also track the location of sources, the inverse of unicast routing. Even though multicast uses the exact inverse logic of unicast routing protocols, you can leverage the information obtained by those protocols (Unicast routing – OSPF and BGP for example) for multicast forwarding.

This allows a network to avoid using another full routing protocol, minimizing the use of memory space. All the functionality that is built into the unicast routing protocol, including loop prevention, path selection, failure detection and so on, can be used for multicast. Modern IP multicast routing uses Protocol Independent Multicast (PIM), which leverages unicast routing information, to forward messages to receivers. The process of determining where the multicast messages are to be sent is referred to as building the tree.

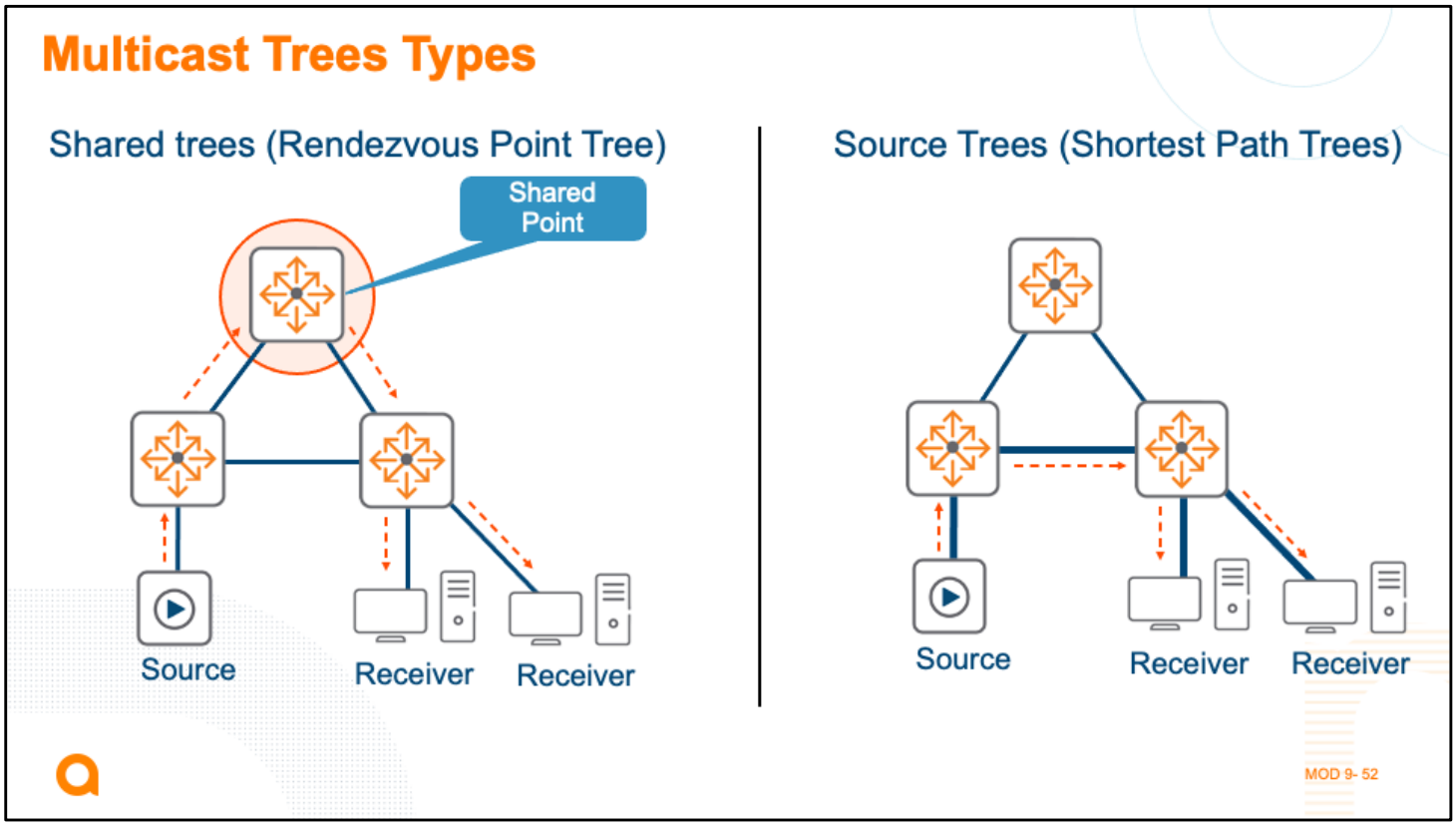
Multicast tree

A network tree is essentially a graph, or directed graph (digraph), that represents a hierarchical logical structure. The primary purpose of a multicast tree is to build an efficient loop-free forwarding path from the source of the multicast stream toward all the receivers. A multicast tree is composed by the following components:

Root. Router closest to the source.

Leaf. Router with an attach receiver.

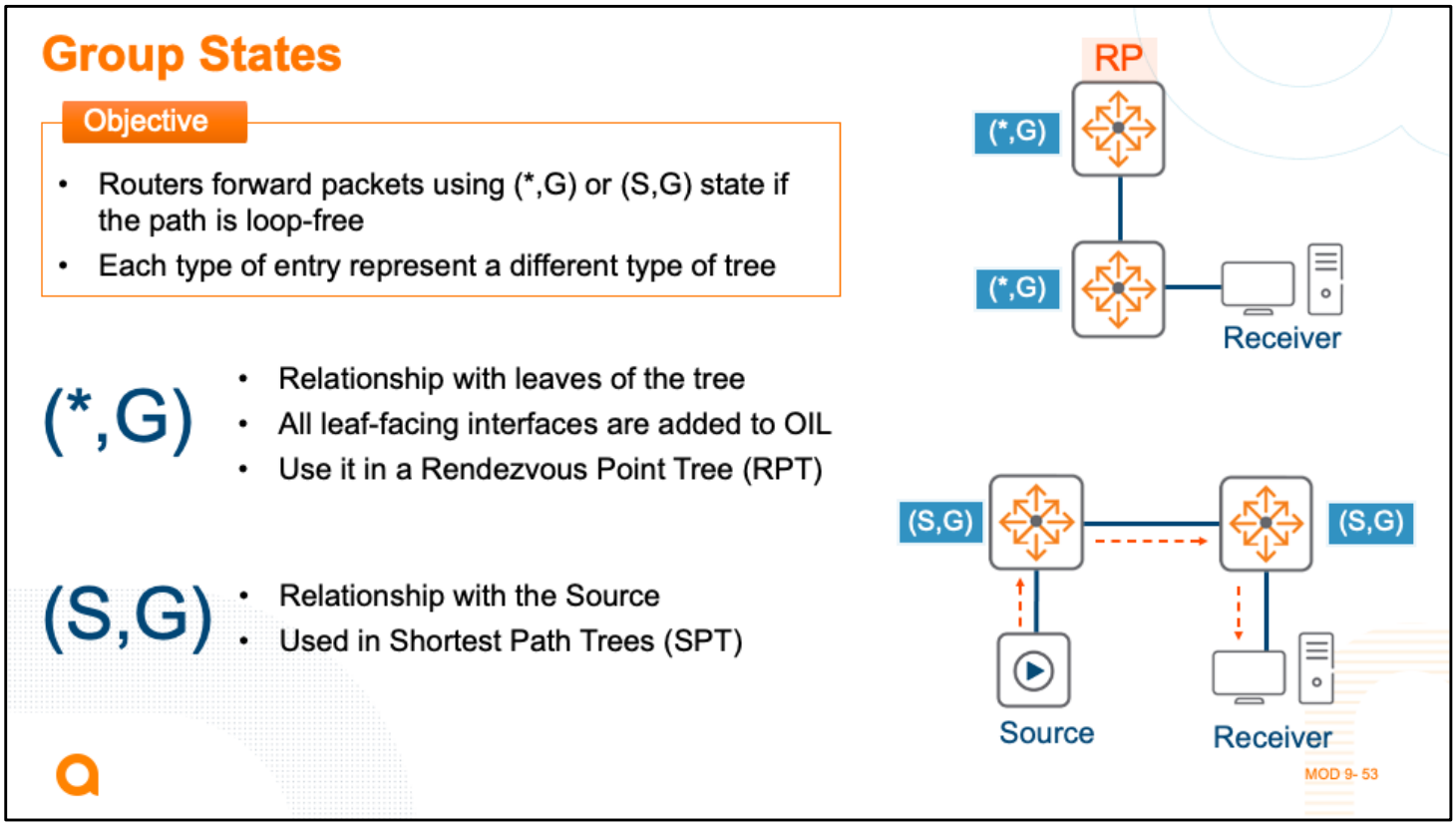
Branch. Intermediate router that perform replication to connect the root to the leaves.



The two types of trees used in multicast networks to control the distribution of traffic are source trees and shared trees.

Source trees (Shortest Path Trees). This is the most common type of multicast tree. From a network perspective, the root of the source tree is the router closest to the source. The tree extends from the root router in a direct path toward all the group members. This creates a tree in which the path might look like the one shown in the figure. Each router in the network must calculate the tree independently based on the information it has received, whether by dynamic protocol updates (PIM), or by configuration. Using Reverse Path Forwarding (RPF), the router checks for loops and creates the Outgoing Interface List (OIL) from the best (sometimes called shortest) paths from the unicast routing table. For this reason, a source tree is also called a shortest path tree.

Shared trees (Rendezvous Point Tree - RPT). The shared tree uses a shared point in the network from which to build the distribution tree. This location is called a Rendezvous Point(RP) and is therefore the root of the tree. Each router still uses Reverse Path Forwarding (RPF) checks to keep the topology loop-free, but the RPF check is in the direction of the RP, not the source. Notice the RP placement has no relation to the IP multicast source. The RP may or may not be in the direct path (shortest path) between the group source and the group client(s). In fact, it can be placed outside of the path entirely. For this reason, a shared tree is also sometimes called a Rendezvous Point Tree (RPT). The advantage to using this type of tree is that all non-RP routers can conserve control- plane resources during stream establishment or during host maintenance. In addition, the network path is pre-determined and predictable, which leads to consistent network operations.



Entries in the multicast table have the form (source,group).

A (*,G) entry in an mroute table represents a router's relationship to the leaves of a tree. All the leaf-facing interfaces are added to an outgoing interface list (OIL). This type of entry is used in a Rendezvous Point Tree (RPT) because routers will have a preconfigured meeting point (RP). This means that routers need only to get one interested receiver for a particular multicast group to build the tree - the source is not really needed. As you will learn later in this module, the RP is in charge of asking this information from the source.

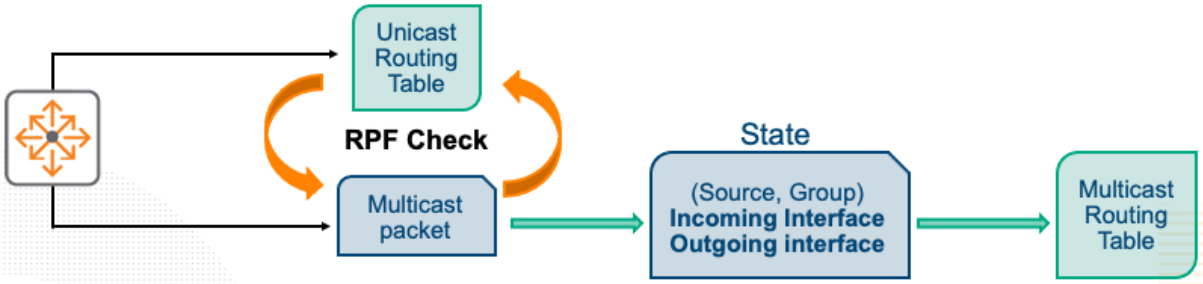
The (S,G) entry in the mroute table represents the router's relationship to the source of the multicast stream. These entries are used in Shortest Path Trees, which indicate both the source IP address (S) and the multicast group (G).

Routers can forward packets using either the (*,G) or the (S,G) entries, as long as the paths are determined to be loop-free.

Reverse Path Forwarding

Objective

- Routers forward packets based on a lookup where the **source** is located:
  - (*,G) entries: the root is the "Rendezvous Point"
  - (S,G) entries: the root is the router where the source is connected
- Routers ensure loop-free forwarding over the existing unicast IP network



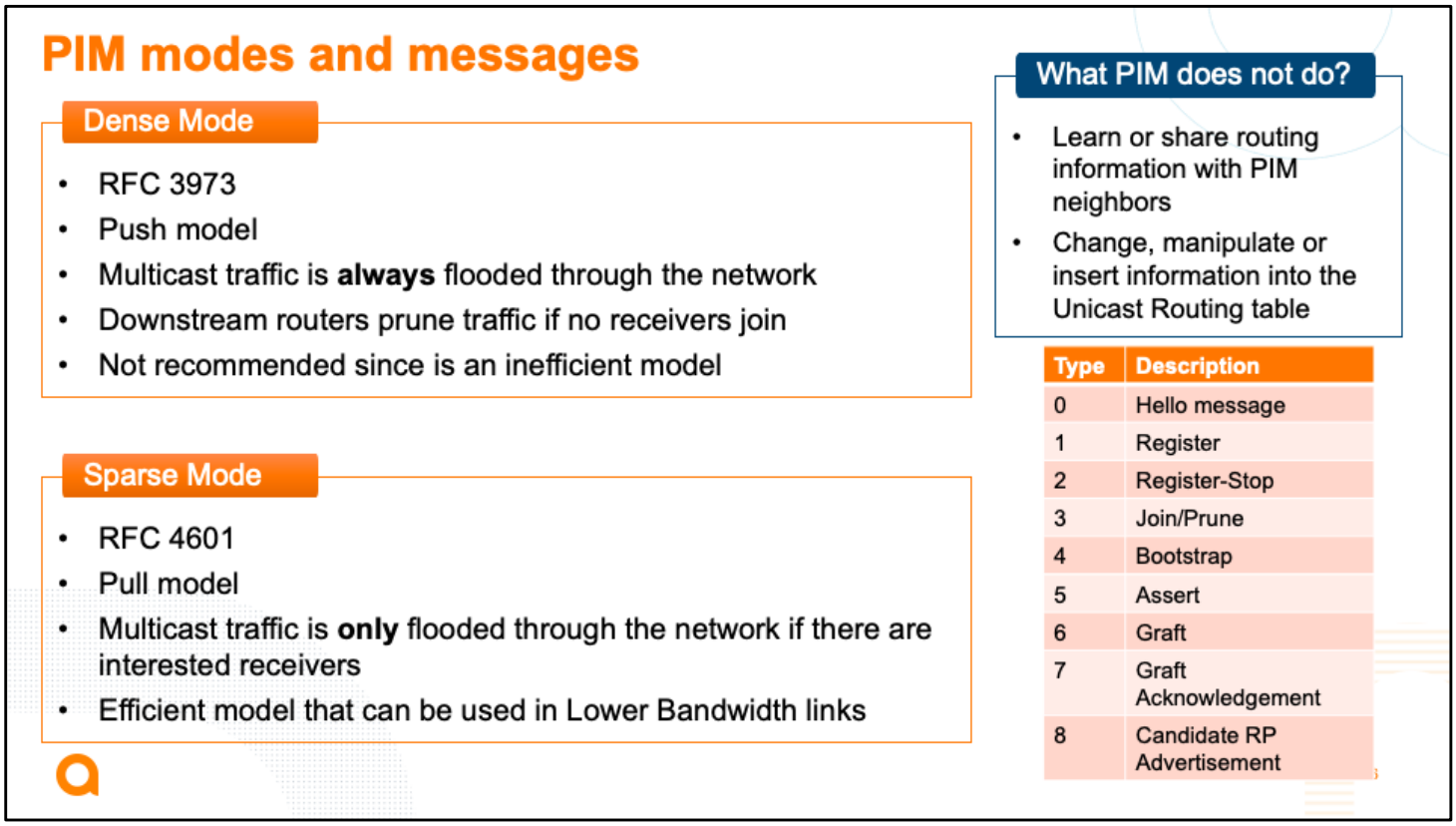
MOD 9- 54

Using IP multicast, the router forwards the packet based on a lookup of where the source is, the exact reverse of the unicast forwarding methodology. Consequently, the multicast network is a true overlay of the unicast network and even uses the unicast routing table to ensure a loop-free multicast forwarding tree. This process is known as Reverse Path Forwarding (RPF). For (*,G) entries, the root of the distribution tree is the RP. For the (S,G) entries, the root for the distribution tree is the router that connects to the multicast source. In short, if RPF check passes then packets flow from incoming interface to all interfaces in the OIL.

aruba
a Hewlett Packard Enterprise company

PIM

Versions and messages



Layer 3 devices use Protocol Independent Multicast (PIM) to build multicast trees. It is “protocol independent” because although relies on he underlying unicast routing to provide reverse-path information for multicast tree building, it is not dependent on any particular unicast routing protocol.

Two of the most important PIM versions are Dense and Sparse mode.

Dense mode: This protocol is defined on RFC 3973. In this mode multicast messages are flooded throughout the entire DM network. When a downstream router receives the multicast packet but does not have a downstream receiver interested in the multicast stream, it responds with a prune message. The IETF implementation of PIM dense-mode (PIM-DM) is often referred to as a push model for tree building. The name is derived from the fact that all dense-mode nodes assume that all other PIM neighbors have an interest in each source tree. The PIM-DM router floods (pushes) (S,G) path updates for each PIM neighbor, regardless of expressed interest. PIM-DM routers may have to process and maintain unnecessary PIM updates, making PIM dense-mode very inefficient for most multicast implementations.

Sparse mode: This protocol is defined on RFC 4601. PIM sparse-mode is an implementation of Any-Source Multicast (ASM). A router receives explicit join/prune messages from those neighboring routers that have downstream group members. The router then forwards data packets addressed to a multicast group (G) only onto those interfaces on which explicit joins have been received. This PIM mode uses the pull model. This model is preferred since it

establishes an efficient way to handle the multicast traffic, this permit to be implemented in lower bandwidth links.

PIM is the most widely deployed multicast routing protocol; however, the term multicast routing protocol confuses many engineers. PIM does not learn and share routing information. PIM does not change, manipulate, or insert information into the unicast Routing table. The primary concern of the multicast routing protocol is to ensure loop-free forwarding over the existing IP network, acting as a control-plane overlay.

PIM uses different messages:

Hello message sent to all PIM neighbors and they help to establish PIM neighbor relationships.

A register message is sent by a DR or a PIM multicast border router (PMBR) to the RP, indicating that a source is sending traffic to a particular multicast group.

The register-stop message is a unicast message sent by the RP to the DR after the SPT is created.

Join/prune messages are sent by routers to the upstream sources and RPs. As the name indicates, join messages are sent to join a stream and prune messages to leave a stream.

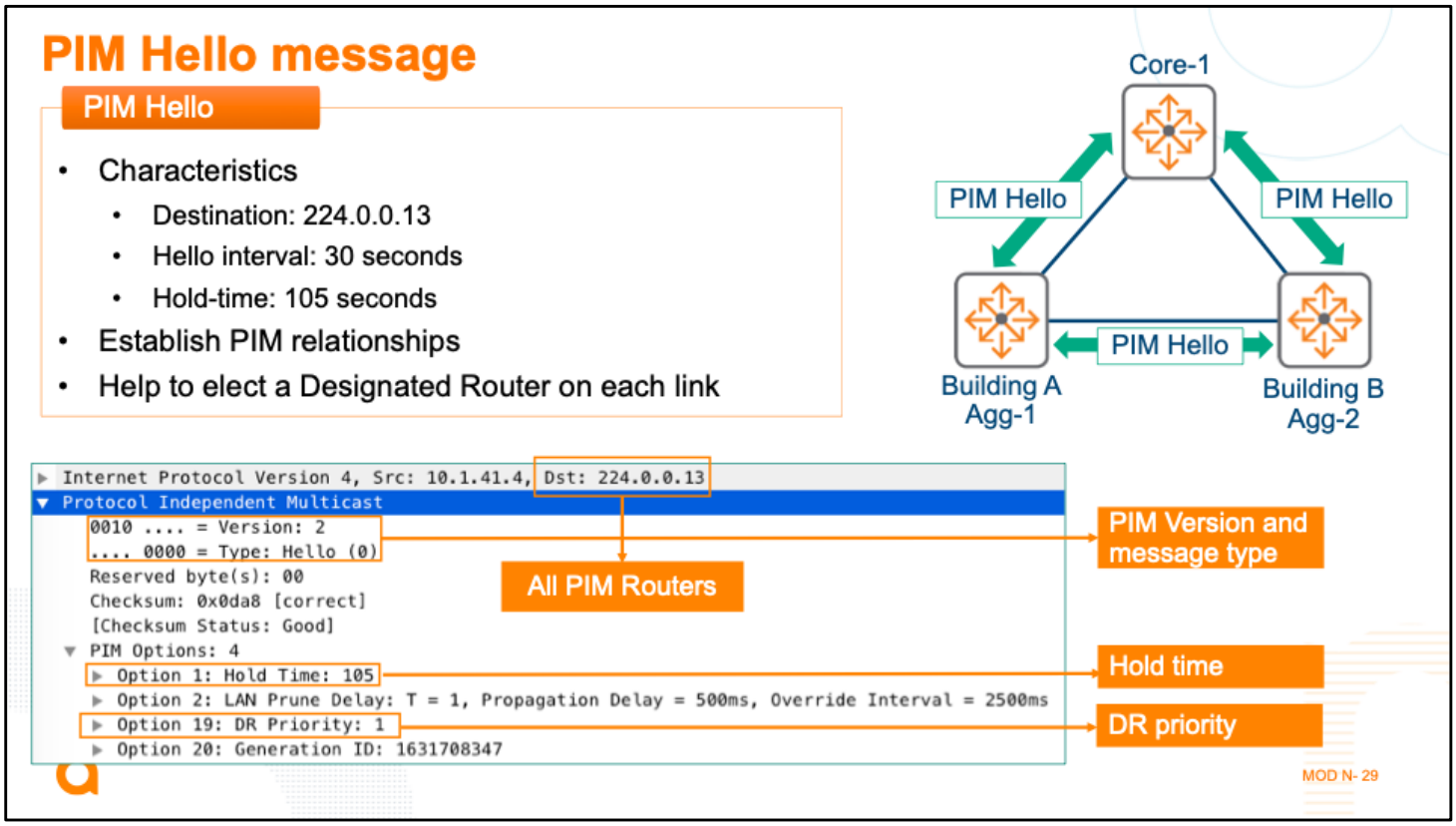
Bootstrap messages contain several entries for the election of the bootstrap router (BSR).

Assert messages are used to elect the PIM forwarder.

Graft messages are used in a PIM-DM environment to indicate the desire to receive a multicast stream.

A graft acknowledge or graft-ack is sent to the graft originator upon receipt of an (S,G) entry.

A candidate RP advertisement is used to send advertisements to the BSR.

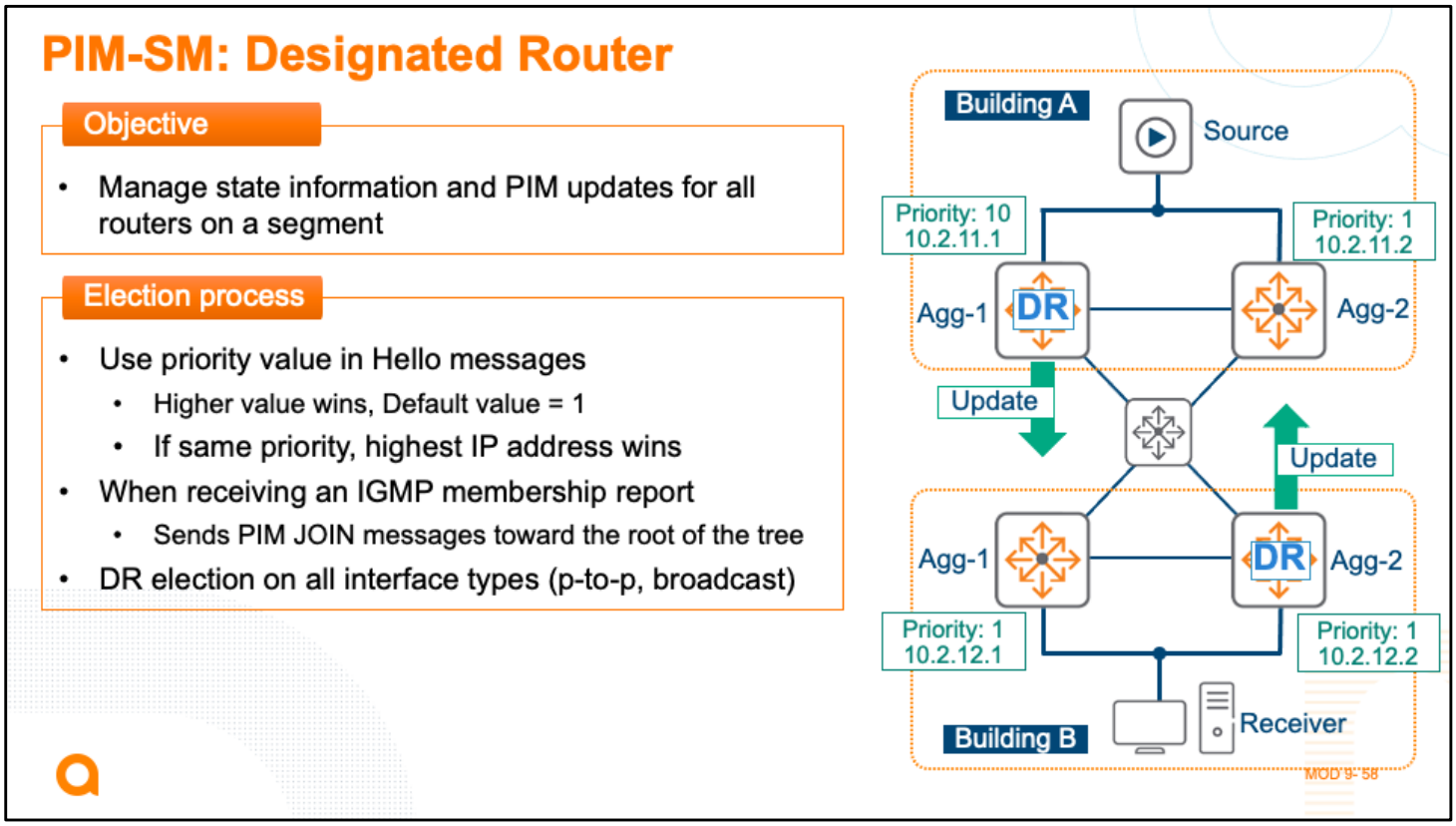


Routers and L3 switches running PIM send periodic hello messages on PIM-enabled interfaces to discover neighbors. These messages are sourced with the IP address of the outgoing interface (In AOS-CX, you can specify the interface that should be used to source those packets) and are sent to all PIM routers with the multicast destination address of 224.0.0.13. They contain information such as the hello period time, hello hold-time, LAN prune delay time and capabilities. The hold-time signifies how long the neighbor will keep the neighbor relationship without receiving an update. The LAN prune delay is used on multi-access networks for propagation delay.

The packet capture in the figure shows a hello message. Pay special attention to the highlighted areas - the destination multicast addresses at Layer 3, the PIM version (v2) and the PIM message type (Hello).

The PIM process begins when two or more routers establish a PIM neighbor adjacency on a given segment. To begin the adjacency process, any interfaces participating in multicast routing must be configured for PIM communications. Routers discover PIM neighbors by sending PIM hello messages to the link-local multicast address 224.0.0.13 out each PIM-enabled interface.

Hello messages are used to elect a Designated Router - a single PIM router that acts on behalf of directly connected hosts in a shared media, like Ethernet. Hello messages are also the way that option negotiation takes place in PIM, so that additional functionality can be enabled, or parameters tuned.



A single network interface and IP address may connect to a multipoint or broadcast network, such as Ethernet. If the network is PIM-enabled, many neighbor adjacencies can occur, one for each PIM-enabled IP neighbor. If every PIM router on the segment were to attempt to manage the tree state, confusion and excess traffic could occur. Much like an OSPF DR, the PIM DR manages state information and PIM updates for all the routers on the segment.

The DR selection process occurs when all neighbors on a segment have replied to PIM hello messages. Each router on a segment selects one router to function as the DR. The DR router is chosen using the PIM DR priority parameter, where the highest priority router is selected as the DR. The DR priority is configurable and sent in the PIM hello message. If the DR priority value is not supplied by all routers, or the priorities match, the highest IP address is used to elect the DR, which is the IP address of the interface sending the PIM message. By default, all interfaces have a PIM DR priority of 1. All routers on the segment should select the same DR router to avoid conflicts.

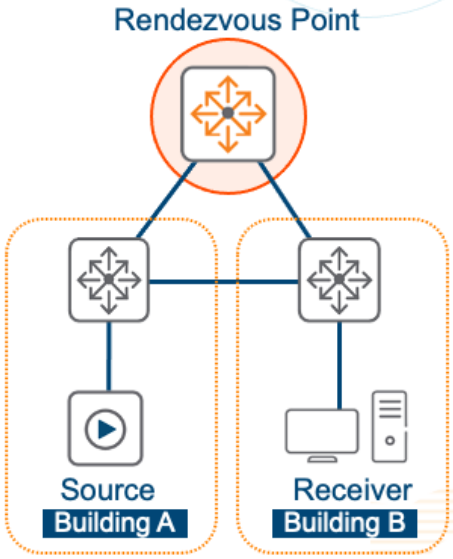
When the DR receives an IGMP membership report message from a receiver for a new group or source, the DR creates a tree to connect the receiver to the source by sending a PIM join message out the interface toward the rendezvous point, when using Any-Source Multicast (ASM) or to the source when using Source-Specific Multicast (SSM). When the DR determines that the last host has left a group or source, it sends a PIM prune message to remove the path from the distribution tree.

From the RFC 4601 section 4.3, DR election will happen on all type of interfaces (point-to-point or broadcast): ”Because the distinction between LANs and point-to-point interfaces can sometimes be blurred, and because routers may also have multicast host functionality, the PIM-SM specification makes no distinction between the two. Thus, DR election will happen on all interfaces, LAN or otherwise.”

PIM-SM: Rendezvous Point

Definition

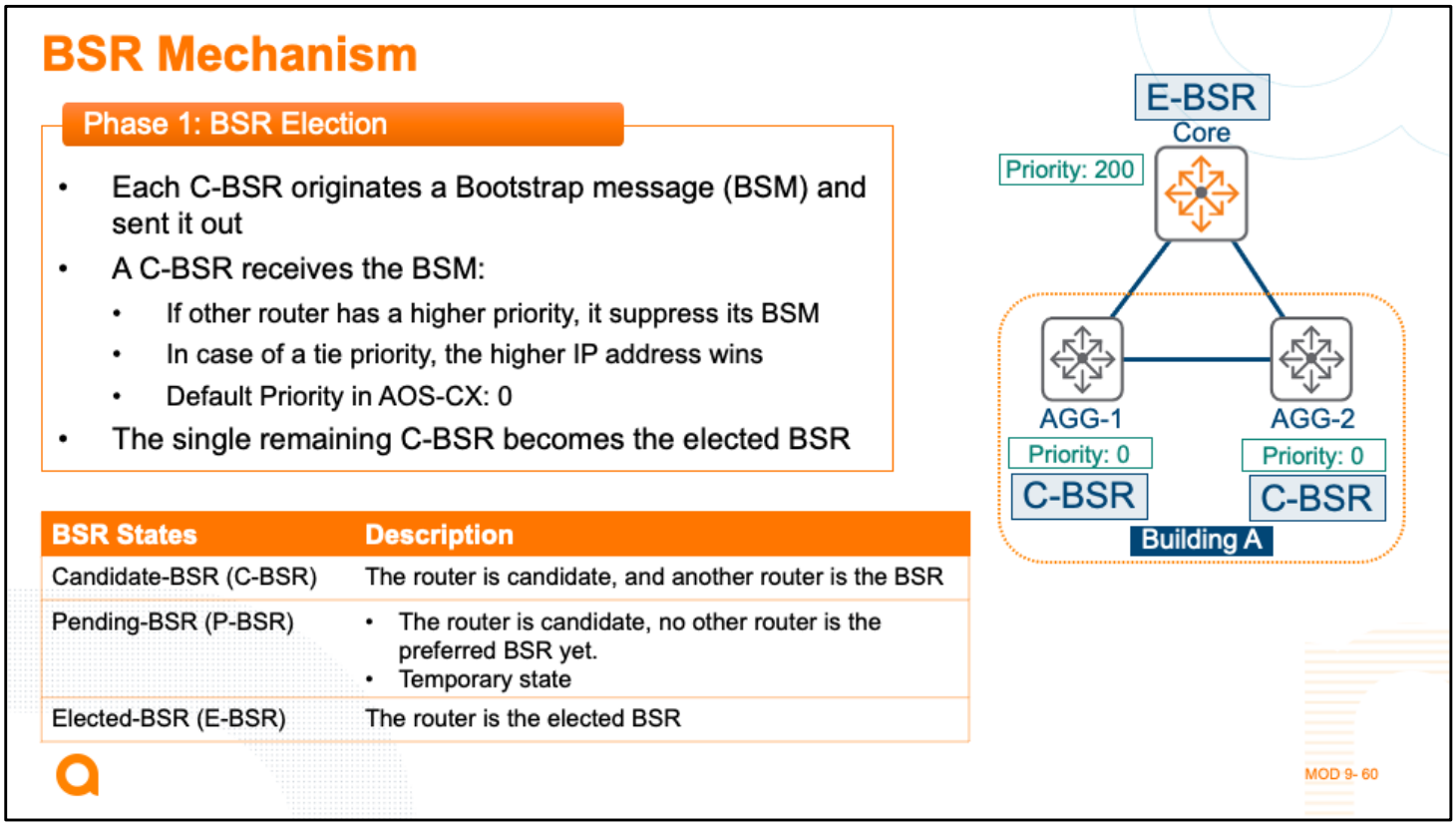
- Root of a Shared tree
- Meeting Point for a Multicast Group
  - Traffic from senders
  - Request from receivers
- Placement
  - Not necessarily on the path between Senders and Receivers
  - Prefer a location closer to sources (if possible)
- One multicast group can have Multiple RPs



MOD 9- 59

A Rendezvous Point (RP) is a router that has been configured to be used as the root of the non-source-specific distribution tree (Shared tree) for a multicast group. Join messages from receivers for a group are sent towards the RP, and data from senders is sent to the RP so that receivers can discover who the senders are and start to receive traffic destined for the group. The RP router is not a packet source but simply a location from which to root and calculate a loop-free topology. The RP can also offload the processing of network joins and leaves during the initial setup of a multicast stream, saving network resources by concentrating them in a single location.

You should place the RP close to the source if possible. This is only applicable to a few sources that are of key importance to the business. Enterprise-wide deployments for multicast normally use RPs in the data center for the enterprise scope. The RP might not be in the path between sender and receivers. PIM-SM uses the RP to build a shared tree as an initial state, but it can move to an optimized shortest path tree, this means that traffic will not be sent to RP all the time and therefore RP location has minimal impact on network performance.



The bootstrap process has four phases as defined in RFC 5059.

Phase 1 - BSR Election:

The function of the BSR is to collect and broadcast the RP set to all routers in the domain. Each Candidate-BSR originates BootStrap Messages (BSMs) announcing their capacity to become a BSR. Every BSM contains a BSR Priority field. Routers within the domain flood the BSMs throughout the domain. A Candidate or C-BSR that hears about a higher-priority C-BSR than itself stops sending BSMs for some period of time. The single remaining C-BSR becomes the elected BSR, and its BSMs inform all the other routers in the domain that it is the elected BSR. If the priority value is the same, the router with the highest IP address is elected.

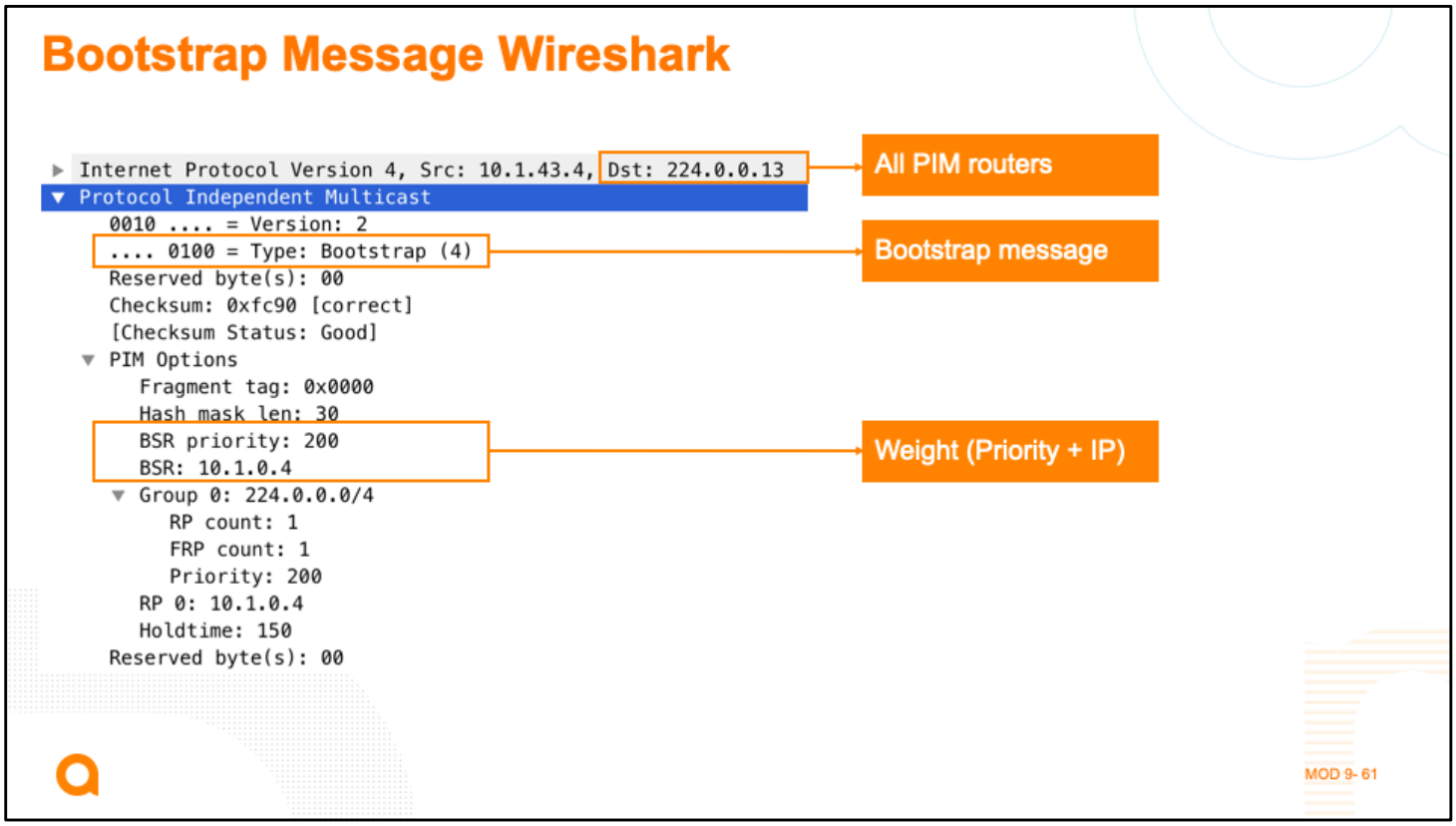
There are three states during the BSR election:

Candidate-BSR (C-BSR). The router is a candidate to be the BSR for the scope zone, but currently another router is the preferred BSR.

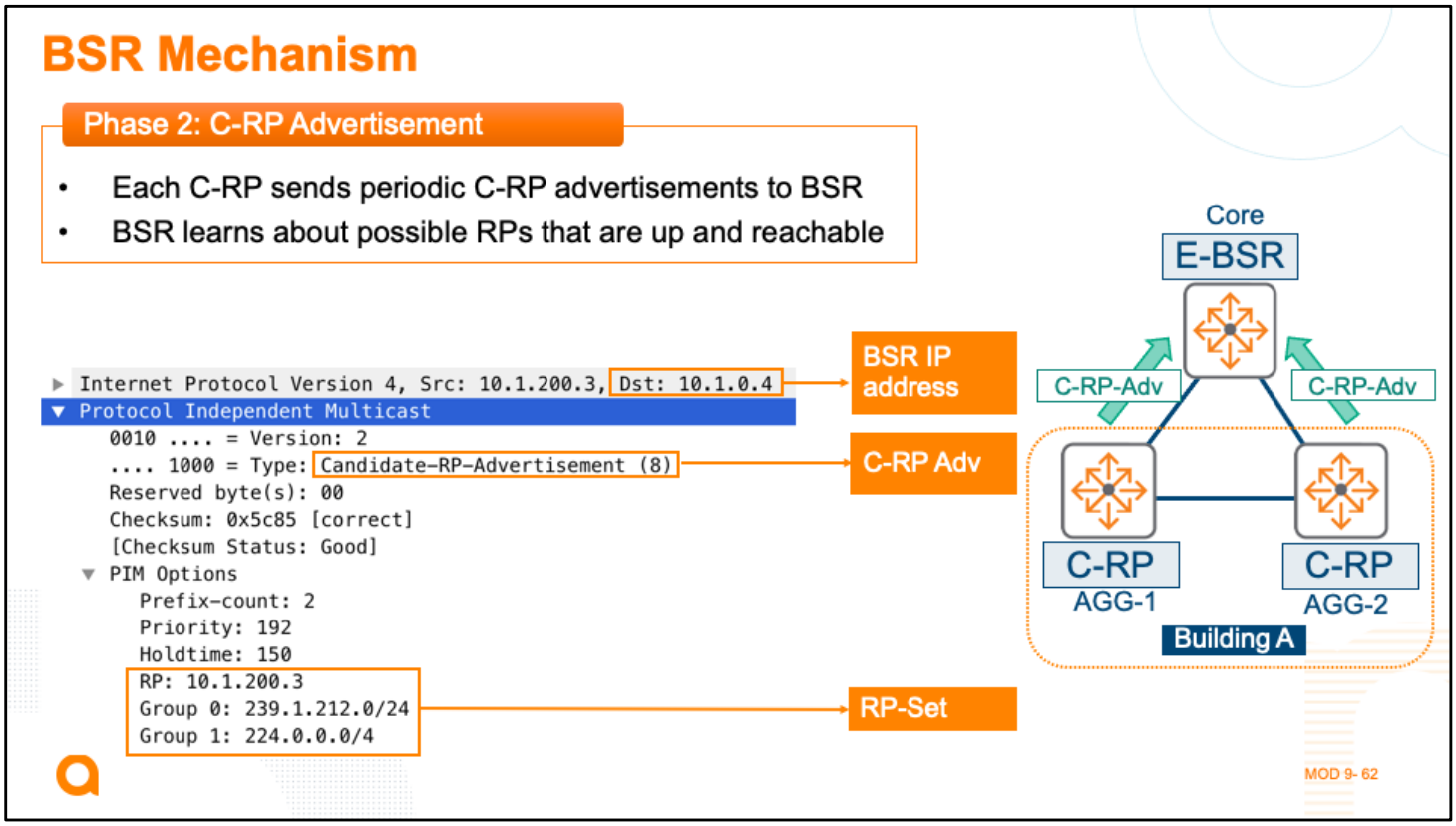
Pending-BSR (P-BSR) The router is a candidate to be the BSR for the scope zone. Currently, no other router is the preferred BSR, but this router is not yet the elected BSR. This is a temporary state that prevents rapid thrashing of the choice of BSR during BSR election.

Elected-BSR (E-BSR) The router is the elected BSR for the scope zone and it must perform all the BSR functions.

Note. A key part of the election mechanism is that we associate a weight with each BSR. The weight of a BSR is defined to be the concatenation in fixed-precision unsigned arithmetic of the BSR Priority field from the Bootstrap message and the IP address of the BSR from the Bootstrap message (with the BSR Priority taking the most-significant bits and the IP address taking the least-significant bits).

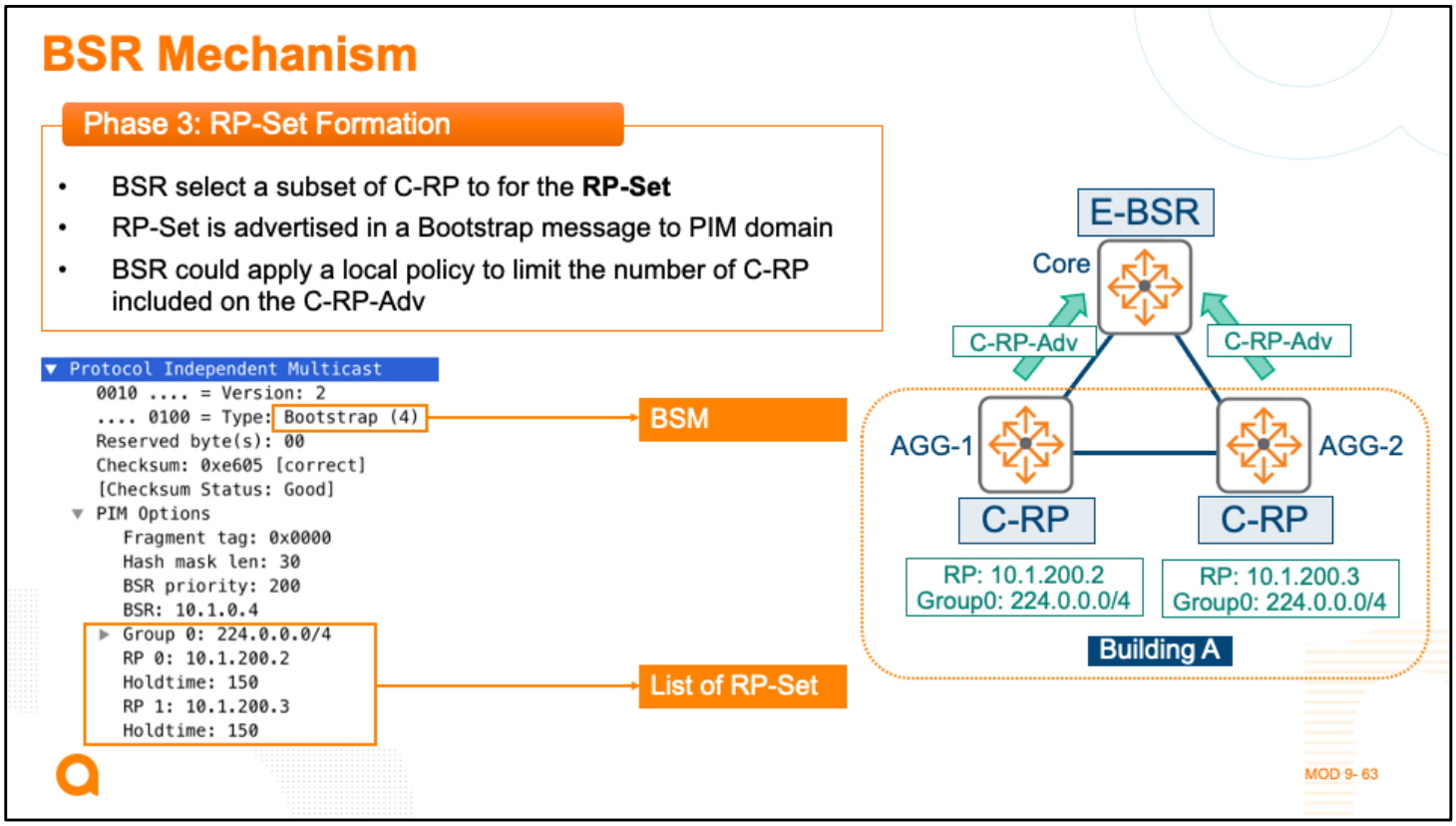


The packet capture in the slide demonstrates a Bootstrap PIM message. Pay special attention to the highlighted areas - the destination addresses at Layer 3 is the All PIM routers address - 224.0.0.13. Also note the PIM message type (Bootstrap), the Weight that is composed by the concatenation of the BSR priority and IP address.



Phase 2 – C-RP Advertisment:

Each Candidate-RP within a domain sends periodic Candidate-RP-Advertisement (C-RP-Adv) messages to the elected BSR. A C-RP-Adv message includes the priority of the advertising C-RP, as well as a list of group ranges for which the candidacy is advertised. In this way, the BSR learns about possible RPs that are currently up and reachable.



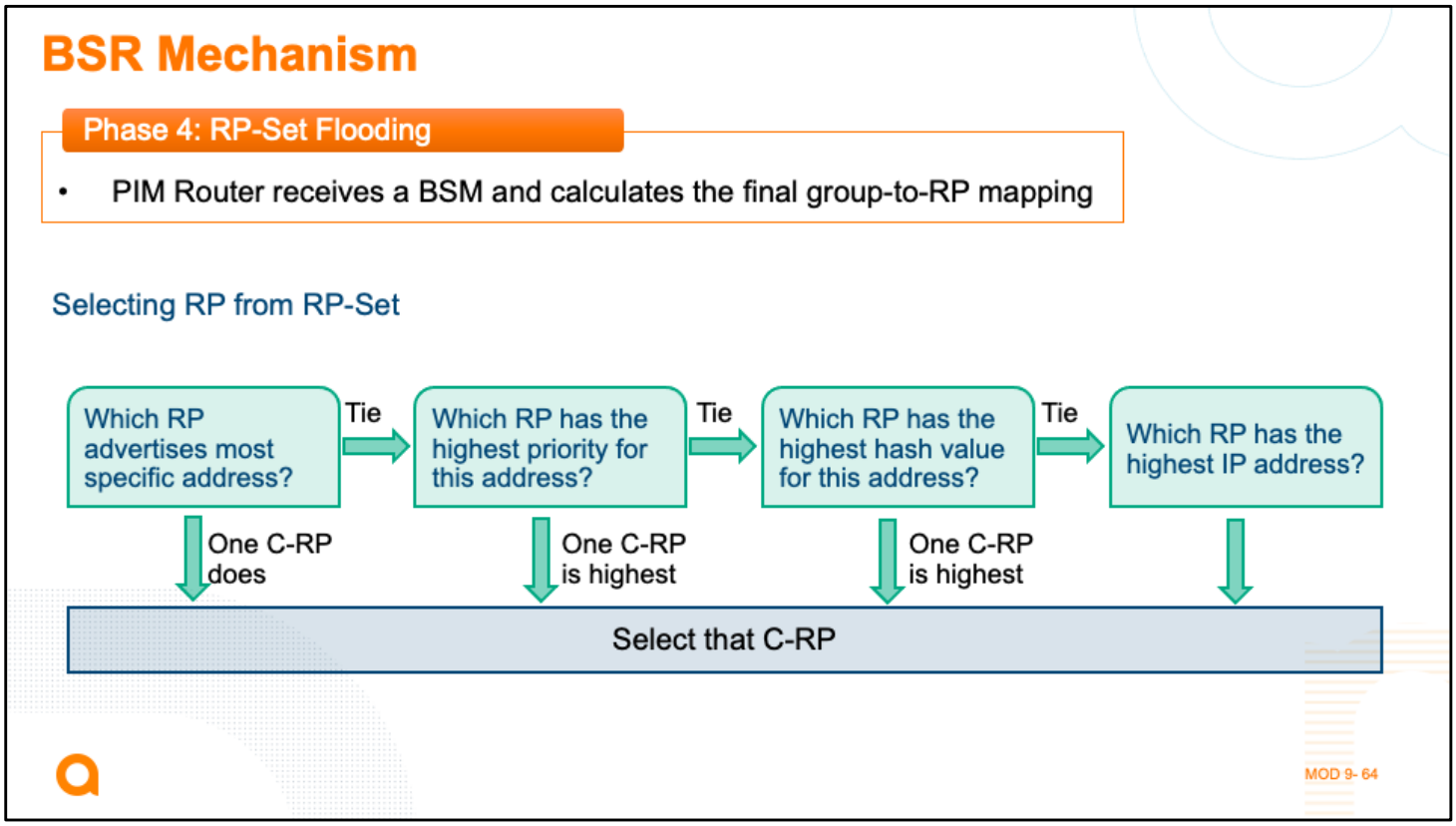
Phase 3 – RP-Set Formation:

To form the RP-Set, the BSR selects a subset of the C-RPs from which it has received C-RP-Adv messages. In general, it should do this in such a way that the RP-Set is neither so large that all the routers in the domain cannot be informed about it, nor so small that the load is overly concentrated on some RPs. It should also attempt to produce an RP-Set that does not change frequently.

The packet capture shows a Bootstrap PIM message originated by the BSR. Pay special attention to the highlighted areas - the RP-Set at the bottom includes the Candidates-RP 10.1.200.3 and 10.1.200.2 that can be used for the Group range 224.0.0.0/4. Even that this bootstrap is only showing one RP-Set it is possible to gave multiple advertised in a single Bootstrap message.

Note: For security reasons, it is recommended that there should be a way to restrict which IP addresses the BSR accepts C-RP-Adv messages from – such as an access lists. For use of scoped BSR, it may also be useful to specify which group ranges should be accepted.

Note. If a group or range of groups has more than one RP the router selects the RP for each group deterministically, so all routers of the same domain use the same RP.



Phase 4 – RP-Set Flooding:

When a PIM router receives a Bootstrap message, it adds the group-to-RP mappings contained therein to its pool of mappings obtained from other sources (e.g., static configuration). It calculates the final mappings of group addresses to RP addresses using the following criteria:

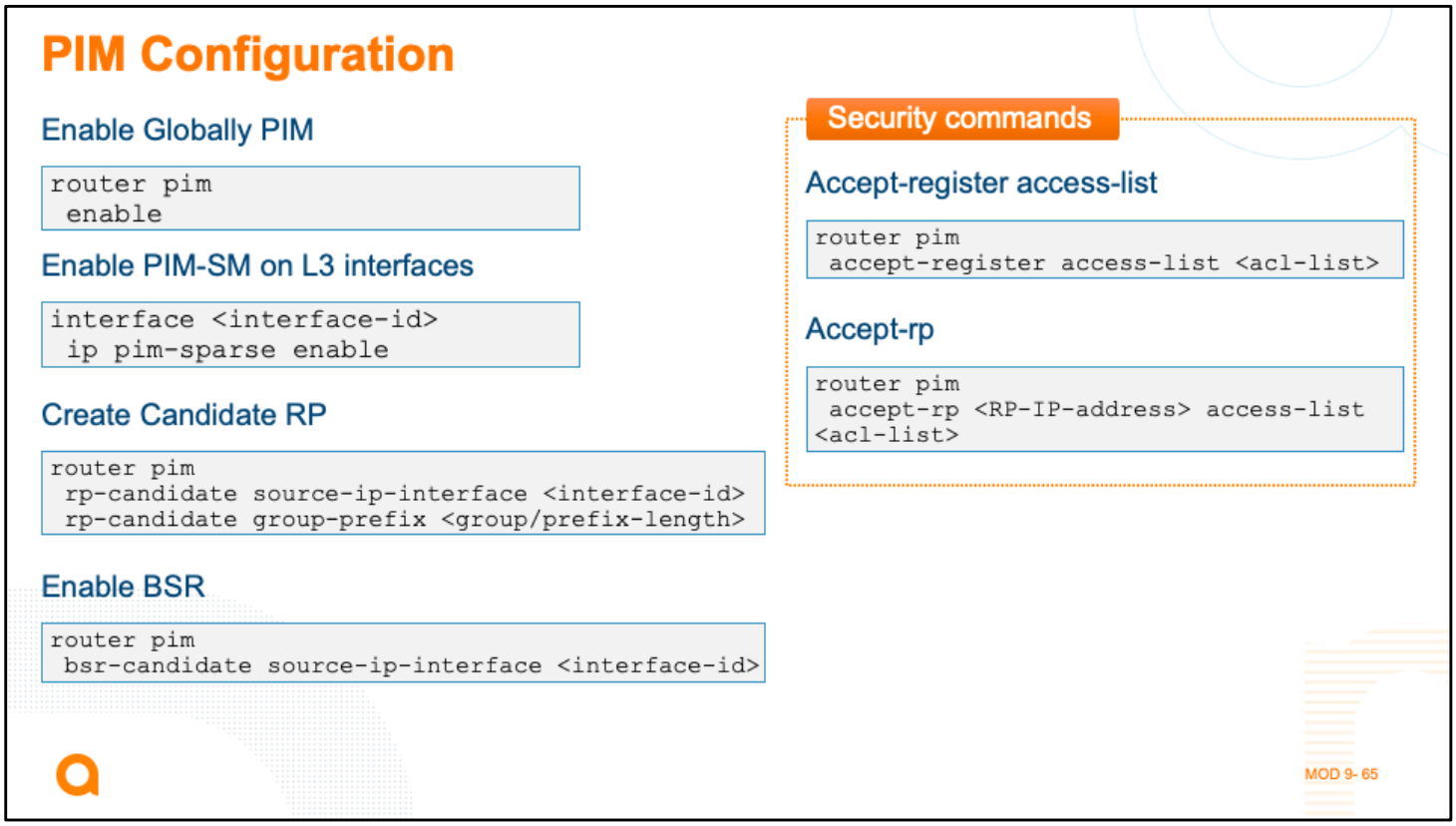
Select the RP that advertises the most specific range of multicast addresses that includes the desired address.

If more than one C-RP advertises a range of the same specificity, select the C-RP configured with the highest priority.

If more than one C-RP is configured with the highest priority, a hash function generates a series of mask length values that are individually assigned to the set of eligible C-RPs. If the hash function matches a single RP candidate to a longer mask length than the other candidates, that candidate is selected to support the group.

If the hash function matches the longest mask length to multiple RP candidates, the C-RP that has the highest IP address is selected to support the group.

In future Bootstrap messages, the BSR includes the RP-Set information. Bootstrap messages are flooded through the domain, which ensures that the RP-Set rapidly reaches all the routers in the domain. BSMs are originated periodically to ensure consistency after failure restoration.



Globally enable PIM

```
router pim
 enable
```

Enable PIM-SM on Layer 3 interfaces

```
interface <interface-id>
 ip pim-sparse enable
```

Create Candidate RP

```
router pim
 rp-candidate source-ip-interface <interface-id>
 rp-candidate group-prefix <group/prefix-length>
```

Enable BSR

router pim

bsr-candidate source-ip-interface <interface-id>

This configures the router to advertise itself as a candidate PIM-SM BSR on the interface specified and enables BSR candidate operation. The result makes the router eligible to be elected as the BSR for the PIM-SM domain in which it operates. One BSR candidate interface is allowed per-router.

Security commands:

accept-register access-list command

This configures ACL on RP to filter PIM Register packets from unauthorized sources. The ACL specified will contain the (S,G) traffic in register packets to permitted or denied.

When register ACL is associated with a PIM Router, PIM protocol will store the source and destination address details along with the action (permit or deny). If there are any existing flows, the user will need to disable and enable PIM on the interface to apply the ACL.

Upon receiving the register messages, a look up is made to check if the S and G in the packet is in the permitted list. If there is no match or if there is a deny rule match, a register stop message is immediately sent and the packet is dropped, and no further action is taken. Permitted packets will go through the normal flow.

Example:

```
switch(config)# access-list ip pim_reg_acl
switch(config-acl-ip)# 10 permit any 20.1.1.1 225.1.1.2
switch(config-acl-ip)# 20 deny any 30.1.1.1 225.1.1.3
switch(config)# router pim
switch(config-pim)# accept-register access-list pim_reg_acl
```

accept-rp command

Enables PIM router to filter PIM join/prune messages destined for a specific RP and specific groups. The ACL specifies the group addresses which are allowed or denied. Up to 8 RP addresses and group ACL can be associated with the PIM router.

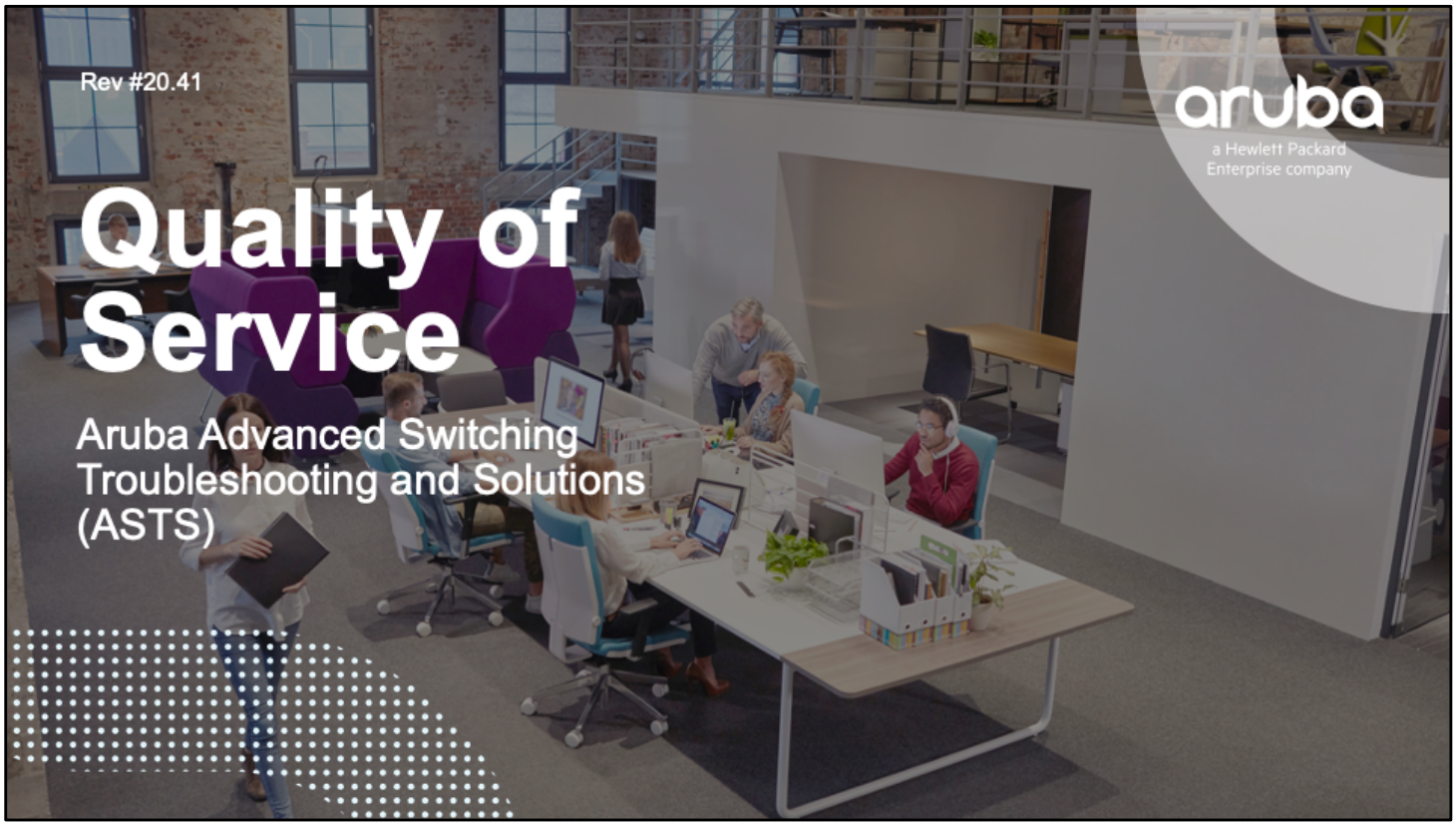
PIM will store the accepted RP address and the associated group ACL. When a join or prune message is received, a RP look up is made for the packet. If the RP is in the configured list and if the group in the join/ prune packet is allowed in the ACL, the packet is allowed. Otherwise, the packet is dropped.

To allow join/prune message from any groups, group address in the ACL can be wild-carded. In this case, only RP address check is performed.

This command impacts only (*,G) join/prune messages. If there are any existing flows, the user will need to disable and enable PIM on the interface to apply the ACL.

```
switch(config)# access-list ip pim_rp_grp_acl
switch(config-acl-ip)# 10 permit any any 225.1.1.2/255.255.255.0
switch(config-acl-ip)# 20 permit any any 239.1.1.2/255.255.255.0
switch(config)-acl-ip# router pim
switch(config-pim)# accept-rp 30.1.1.1 access-list pim_rp_grp_acl
```


Lab Activity
Lab 09


Rev #20.41
aruba
a Hewlett Packard Enterprise company
Quality of Service
Aruba Advanced Switching Troubleshooting and Solutions (ASTS)

Objectives

- Describe QoS operation, traffic characterization.
- Explain the Differentiated Services model and AOS-CX QoS processing
- Describe rate limiting operation, configuration, and verification
- Describe the prioritization stage of QoS
- Configure and verify classifiers and marking
- Describe and configure queuing mechanisms
- Configure and verify QoS remarking

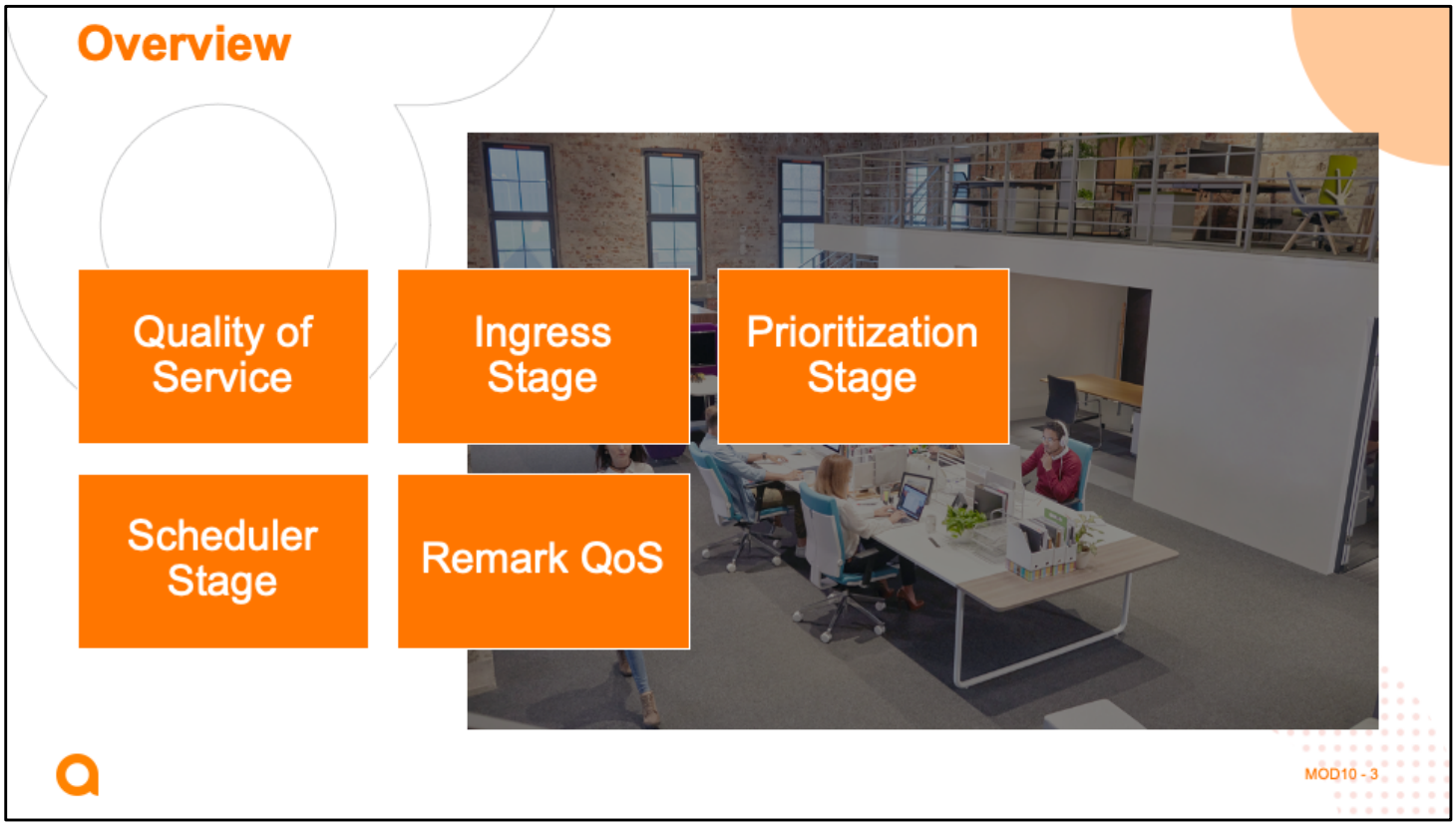


After completing this module, you will be able to describe QoS operation, traffic characterization, and explain the Differentiated Services model and AOS-CX QoS processing.

You will be able to describe rate limiting operation, configuration, and verification, and describe the prioritization stage of QoS.

You will also be able to configure and verify classifiers and marking, and to describe and configure queuing mechanisms.

Finally, you will be able to configure and verify QoS remarking.



Quality of Service (QoS) mechanisms enable you to customize network traffic handling characteristics, to ensure that specific traffic types receive the service they need, and that the traffic most important to your corporation is treated preferentially.

You begin by reviewing QoS concepts and traffic characterization techniques, with a special focus on the Differentiated Services (Diff-Serv) model of QoS.

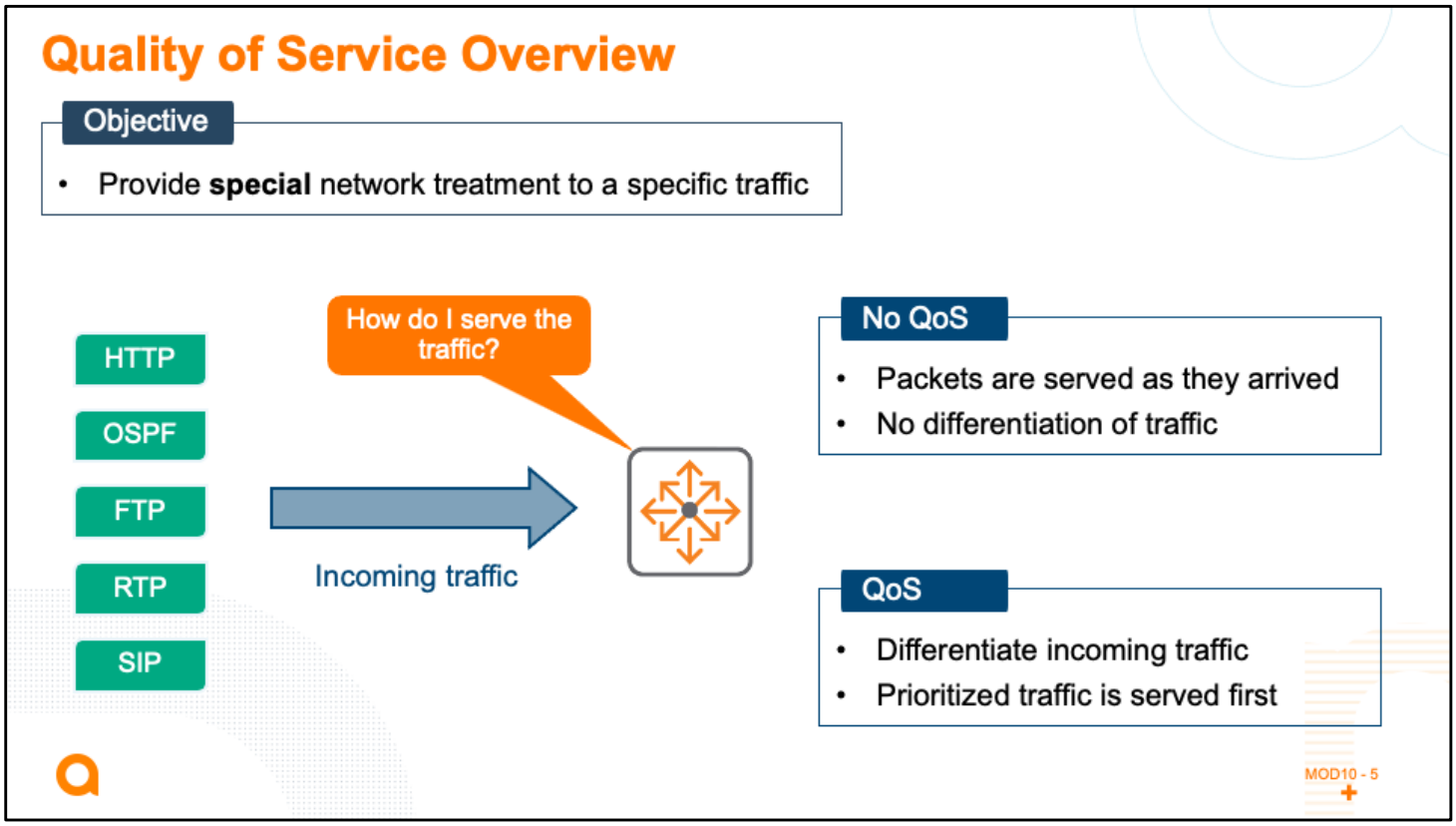
Then you will explore each key phase of the QoS process and related configuration and verification, including the Ingress, Prioritization, Scheduler, and remarking processes.

Quality of Service

Overview, Traffic characterization, Differentiated Services and Per Hop Behavior



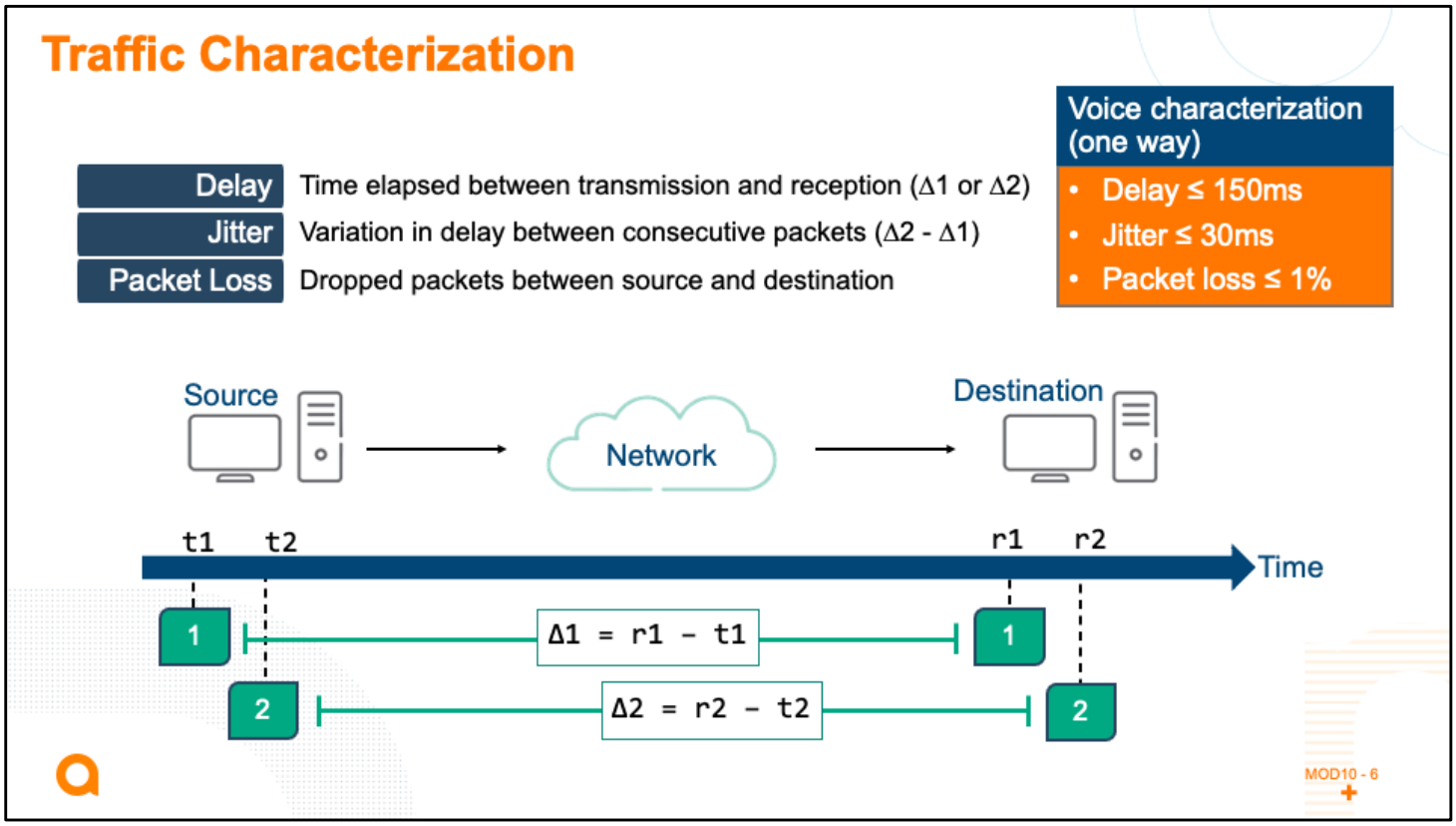
MOD10 - 4



Quality of Service (QoS) enables networks to provide better or “special” service to a set of users/applications to the disadvantage of other users/applications. There is no win-win situation when you implement Quality of Service.

In a network, different traffic types compete for the same network resources such as bandwidth and fast processing. Allowing fair and even competition by having no traffic differentiation does not work because different types of traffic have different requirements, just like an ambulance and a truck on the same road have different needs. QoS is all about not being fair when dividing the network resources, but rather selecting discriminately who is favored and who gets penalized. QoS does not make the road wider, it just decides who goes first and consequently who must wait.

Is important to clarify that QoS is not a standalone service or product, but rather a concept that supports the attributes of a network by spanning horizontally across it.



You need the network to apply different QoS behaviors to treat different traffic types. You need to characterize these behavioral requirements, for which there are three commonly used parameters: delay, jitter, and packet loss.

Delay (also commonly called latency) is defined as the time elapsed between the transmission of the packet by the source and the receipt of the same packet by the destination. In the figure for packet 1 delay is the difference between the values r1 and t1, represented by the symbol Δ1, and is usually measured in milliseconds.

Jitter represents the variation in delay between consecutive packets. Thus, if packet 1 takes Δ1 to transit the network, while packet 2 takes Δ2, then the jitter between packets 1 and 2 is the difference between Δ1 and Δ2 (also measured in milliseconds).

Packet loss, this parameter represents how many packets are not received compared with the total number of packets transmitted and is usually measured as a percentage.

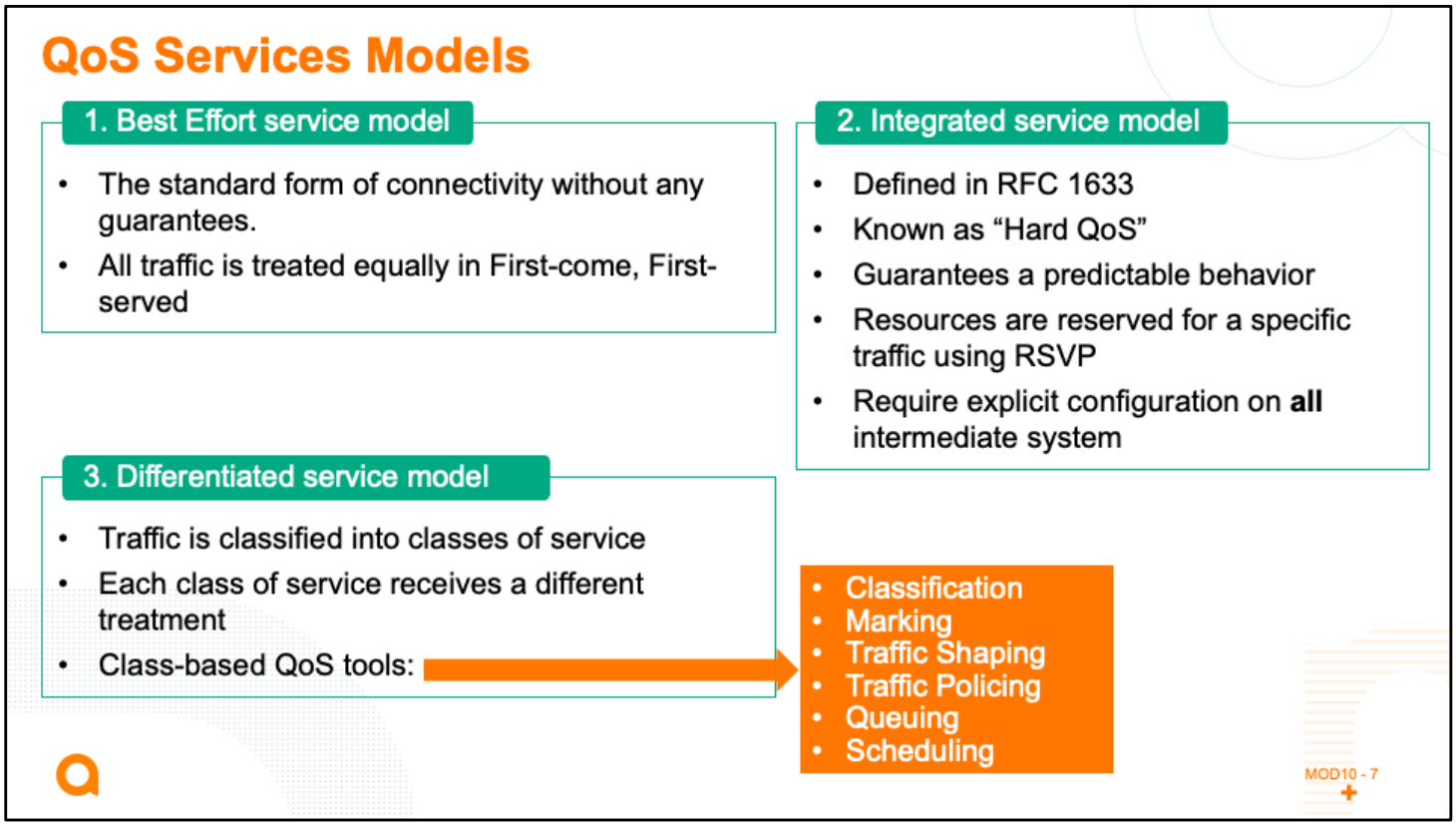
The tolerance to these parameters is highly dependent on whether the traffic is real time or not, because for real-time traffic the time gap in which the packet is considered useful for the destination is typically much narrower.

For example, a typical characterization of voice traffic in one way is as follows:

Delay less or equal to 150ms

Jitter less or equal to 30ms

Packet loss less or equal to 1%



Best effort service model

This is the simplest service type. All traffic is treated equally in a first-come, first-served manner. Using this model, the traffic does not have any guarantee of arriving to its destination. If the traffic load is low in relation to the capacity of the network links, then there is no need for the administrative complexity and costs of maintaining a more complex end-to-end policy. This is sometimes called over-provisioning, as all link speeds are much higher than peak loads on the network.

Integrated service model

IntServ, also known as hard QoS, is an absolute reservation of services. In other words, the IntServ model implies that traffic flows are reserved explicitly by all intermediate systems and resources.

The IntServ model, defined in RFC 1633, guarantees a predictable behavior of the network for applications. IntServ provides multiple services that accommodate multiple QoS requirements. IntServ is implemented through the use of the Resource Reservation Protocol (RSVP), enabled at both the endpoints and the network devices between. The RSVP-enabled application requests a specific kind of service from the RSVP-enabled network before it sends data. Explicit signaling via RSVP makes the request, and the application informs the network

of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. The application is expected to send data only after it gets a confirmation from the network. The network also expects the application to send data that lies within its described traffic profile.

The RSVP-enabled network performs admission control based on information from the requesting host application and available network resources. It either admits or rejects the application's request for bandwidth.

Differentiated services (DiffServ) model

This model was developed based on a class scheme, in which traffic is classified into classes of service. With DiffServ each class of service receives a different treatment from the network device. This model includes Class-based QoS tools or techniques which include:

Classification

Marking on Layer 2 and Layer headers (CoS, IP Precedence and DSCP)

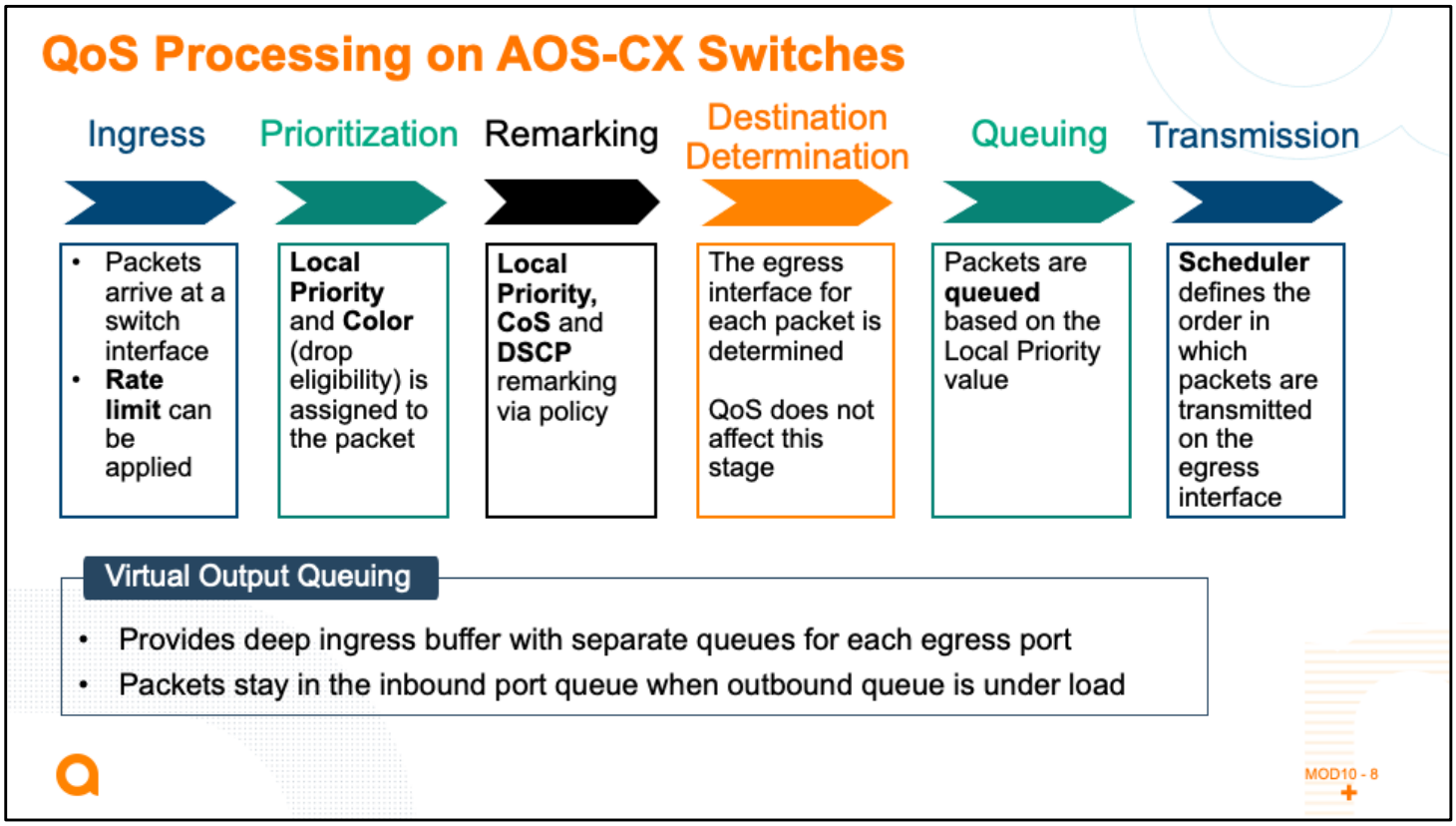
Traffic shaping

Traffic policing

Queuing

Scheduling

The DiffServ model provides multiple levels of service that satisfy differing QoS requirements. However, unlike with the IntServ model, an application that uses DiffServ does not explicitly signal the network devices before sending data.



AOS-CX switches use a special process to locally apply QoS features. This process has five stages:

Ingress. Packets that arrive at a switch interface could be filtered by the rate limit QoS feature. This QoS feature can limit Broadcast, Multicast and Unknown Unicast (BUM) traffic, to protect the switch from unnecessary processing.

Prioritization. The switch applies the QoS classifier tool to determine the correct priority for a specific traffic type. AOS-CX defines two values: Local Priority and Color. The Local Priority helps to determine the correct egress priority queue for the traffic. The color determines the traffic's drop eligibility. If queues become congested, more drop-eligible traffic will be dropped first.

Remarking. By using a policy, the administrator can modify the Local Priority, Class of Service or DSCP value.

Destination Determination. The egress interface for each packet is determined based on the switching and routing logic. Internally the switch moves the traffic to the correct egress interface. QoS tools in this stage are not used and therefore do not affect this stage.

Queuing. The switch assigns each egress packet to an interface queue, based on the packet's Local Priority value. AOS-CX supports up to 8 queues. Packets wait in the queue for their turn to be transmitted.

Transmission. In this stage a Scheduler QoS tool defines the order in which packets from different queues are transmitted out the egress interface.

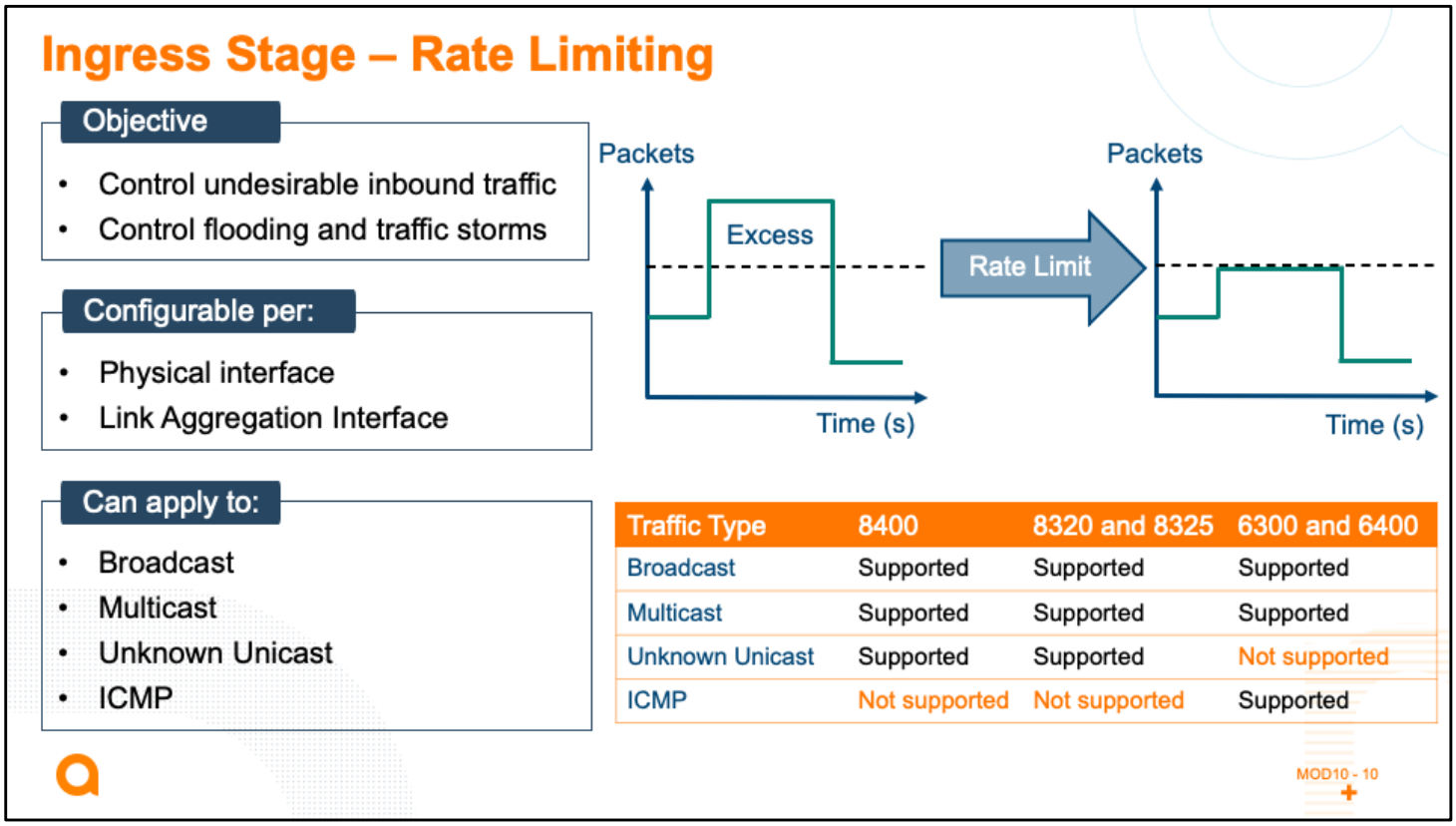
AOS-CX switches use an intra-switch queuing method called virtual output queuing (VoQ). If the ingress buffer used a single queue, head of line (HOL) blocking could delay the traffic. If the packet at the front of the queue is destined out a congested port, it delays all packets behind it, even though those that are destined to non-congested ports. VoQ prevents this problem by providing deep ingress buffers with separate queues for each egress port. Physically the ingress buffer consists of internal and external DRAM banks with most traffic typically being buffered in the internal DRAM. This means that packets will stay in the inbound port queue when the outbound queue is under load. Using this technique packets are never dropped in the outbound interface but in the inbound.

This feature is embedded in the Operating System and is not configurable.

Ingress Stage

Rate Limiting, CIR, Configuration and Verification





Port rate limiting helps control undesirable traffic. The goal is to allow only enough BUM traffic for the network to function properly, while preventing flooding and traffic storms.

A certain amount of BUM traffic is required for normal network operation. Broadcast packets may include ARP and DHCP traffic, for instance. Video streams, and certain types of network protocol packets, are multicasts. Unknown-unicast packets may be intended for devices whose addresses have temporarily aged out of network-forwarding caches. Configuring rate limits can help provide the balance between necessary traffic and flooded traffic.

Rate Limiting is a traffic policer QoS tool. It ensures that traffic conforms to a defined rate – a bandwidth limit. An important characteristic of rate limiting is that it does not add delay to traffic to be forwarded - that which is under the threshold and is accepted by the switch.

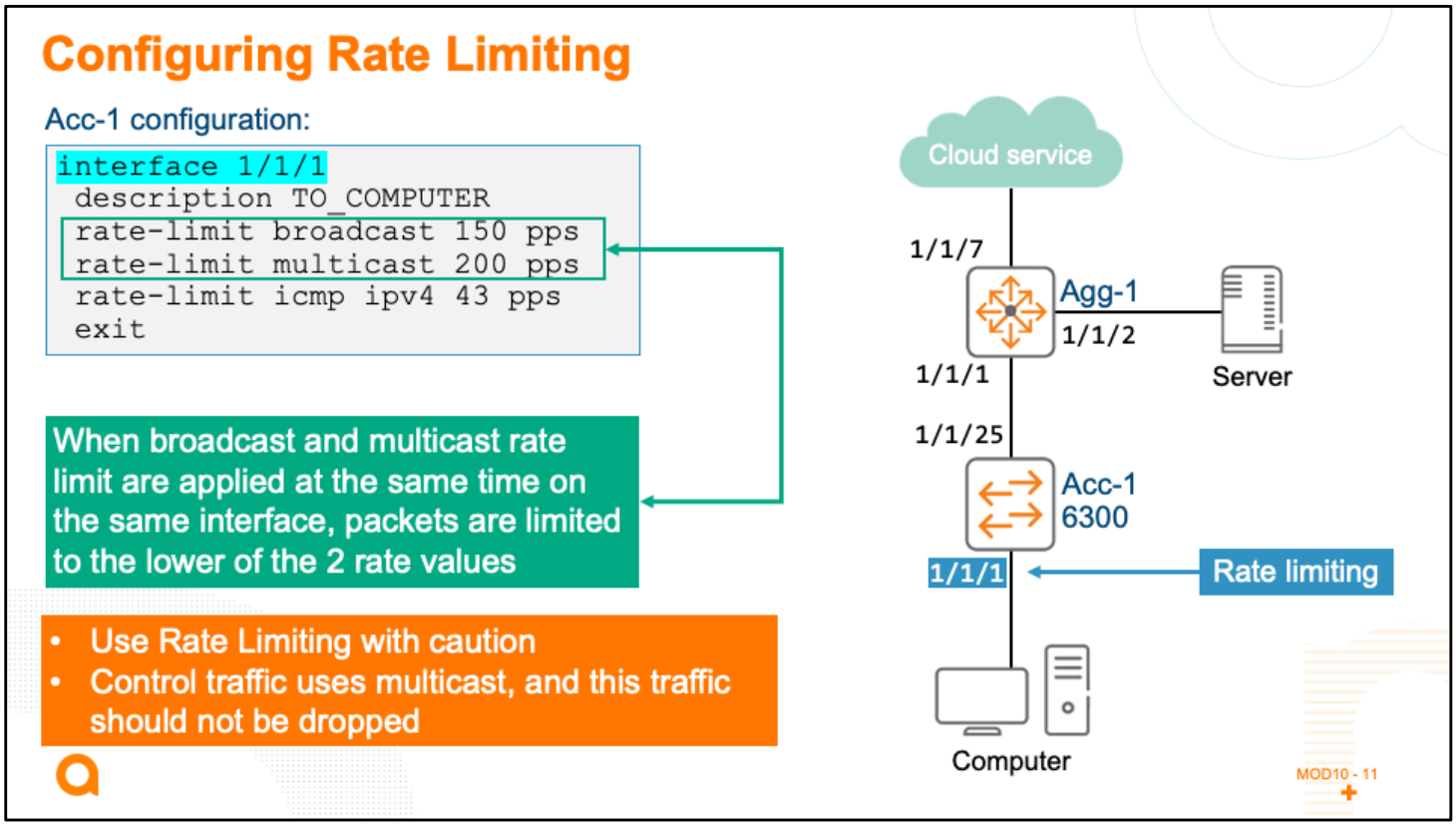
The rate limiting QoS feature can be implemented on all AOS-CX platforms. However, supported parameters are platform dependent:

Broadcast: Supported on all AOS-CX platforms, limit only the broadcast traffic.

Multicast: Supported on all AOS-CX platforms, limit only the multicast traffic.

Unknown Unicast: Supported only on 8400, 8325 and 8320 switch platforms, limit only the unknown unicast traffic.

ICMP: Supported only on 6400 and 6300 switch platforms, the ICMP rate limit can be configured to apply to IPv4, IPv6, or all IP traffic. Only one ICMP rate-limit can be configured at a time. Applying a new ICMP rate-limit replaces any previous ICMP rate-limit.



In this example you can see the Rate limiting configuration applied on port 1/1/1.

Note: When Multicast and broadcast rate limit are applied at the same time on the same interface, broadcast packets are limited to the lower of the two rate values.

Rate Limiting should be use with caution specially when it is limiting Multicast traffic. Since many of the control traffic uses Multicast the stability of the network can be compromised if control packets are dropped by this feature.

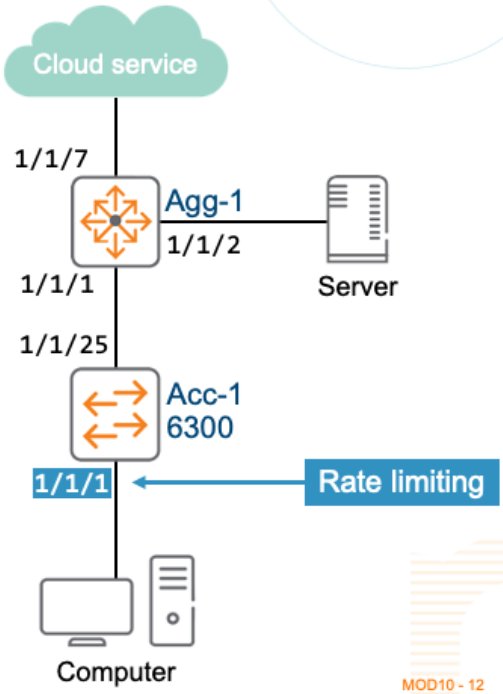
Verifying Rate Limiting

```
AGG-1# show interface 1/1/1 qos
Interface 1/1/1 is up
 Admin state is up
 qos trust none (global)
 qos queue-profile factory-default (global)
 qos schedule-profile factory-default (global)
 rate-limit broadcast 150 pps (150 actual)
 rate-limit multicast 200 pps (200 actual)
 rate-limit icmp ipv4 43 pps (43 actual)
```

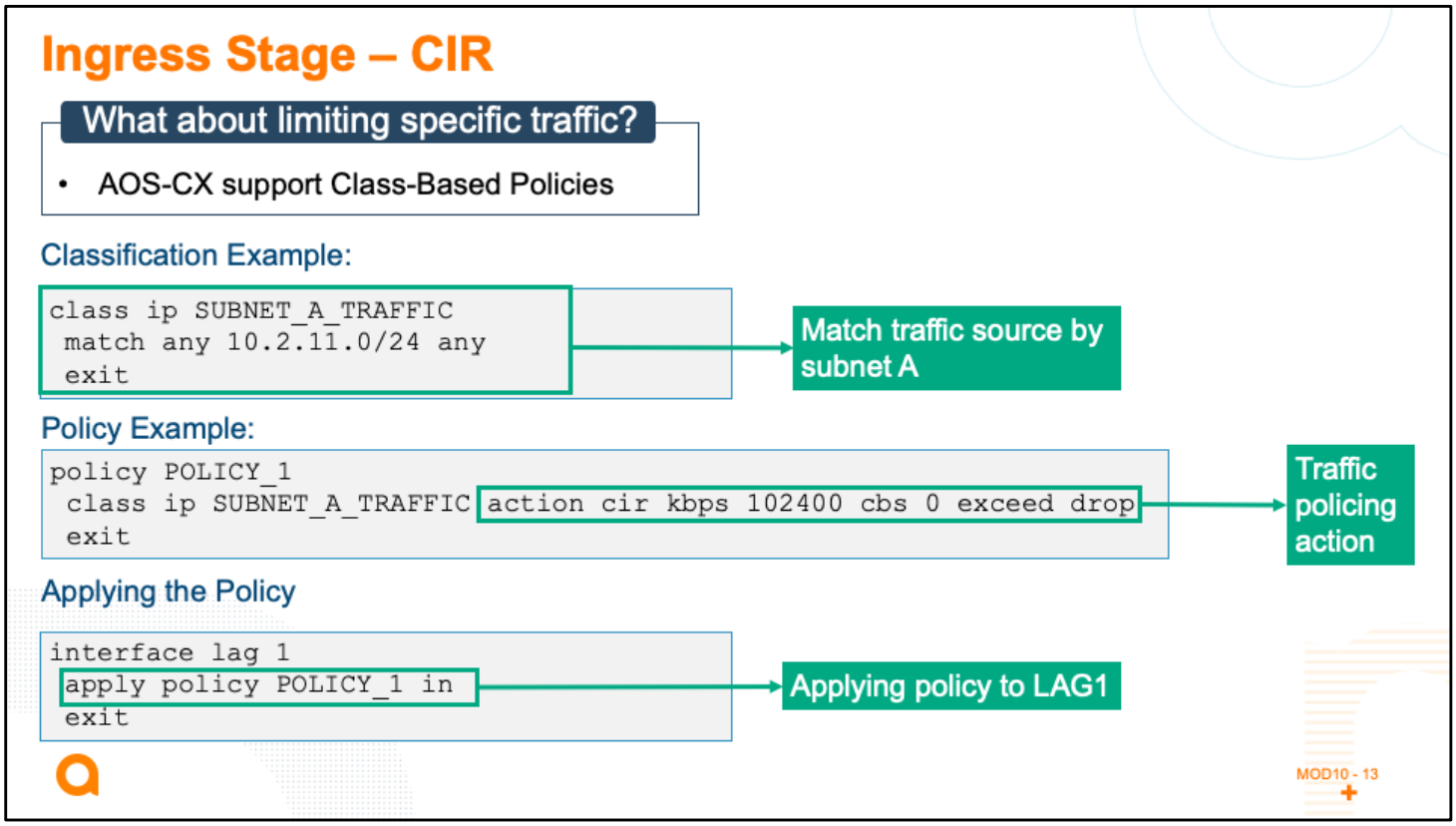
```
AGG-1# show interface 1/1/1

Interface 1/1/1 is up
 Admin state is up
 Link transitions: 1
 Description: TO_COMPUTER
 Hardware: Ethernet, MAC Address: 10:4f:58:f6:01:67
 MTU 1500
 Type 1GbT
 Full-duplex
 qos trust none
 rate-limit broadcast 150 pps (150 actual)
 rate-limit multicast 200 pps (200 actual)
 rate-limit icmp ipv4 43 pps (43 actual)
 Speed 1000 Mb/s
 Auto-negotiation is on

 <<Omitted output>>
```



The show interface <interface-id> qos and show interface <interface-id> comands are useful to verify the rate-limiting feature implemented on the interface.



Classifier policies let a network administrator define sets of rules based on network traffic addressing or other header content and use these rules to restrict or alter the transit of traffic through the switch.

Choosing the rule criteria is called Classification, in AOS-CX this is achieved by creating a traffic class. The three types of classes (MAC, IPv4, and IPv6) are each focused on relevant frame/packet characteristics. Classes can be configured to match or ignore almost any frame or packet header field. Network traffic passing through a switch can be classified based on many different frame/packet characteristics including, but not limited to:

Frame ingress VLAN ID

Source and/or destination Ethernet MAC, IPv4 or IPv6 address

Layer 2 (EtherType) and Layer 3 (IP Protocol)

Layer 4 application ports

The example presents a couple of traffic classes created to match the traffic destined to a Windows server (10.254.1.21) and other that matches the traffic generated from the subnet A (10.2.11.0/24).

class ip SUBNET_A_TRAFFIC

match any 10.2.11.0/24 any
exit

A policy contains one or more policy entries, which are listed according to priority by sequence number. A single policy entry contains a class and corresponding policy action. Policy action is taken on traffic matched by its corresponding class. A policy can be applied to an individual front plane port, a Link Aggregation Group (LAG) interface, or a VLAN.

In the example POLICY_1 is created, the actions of this policy are:

Limit the Bandwidth sourced by the Subnet A to 102400 kbps, exceed traffic is dropped without accepting any burst size.

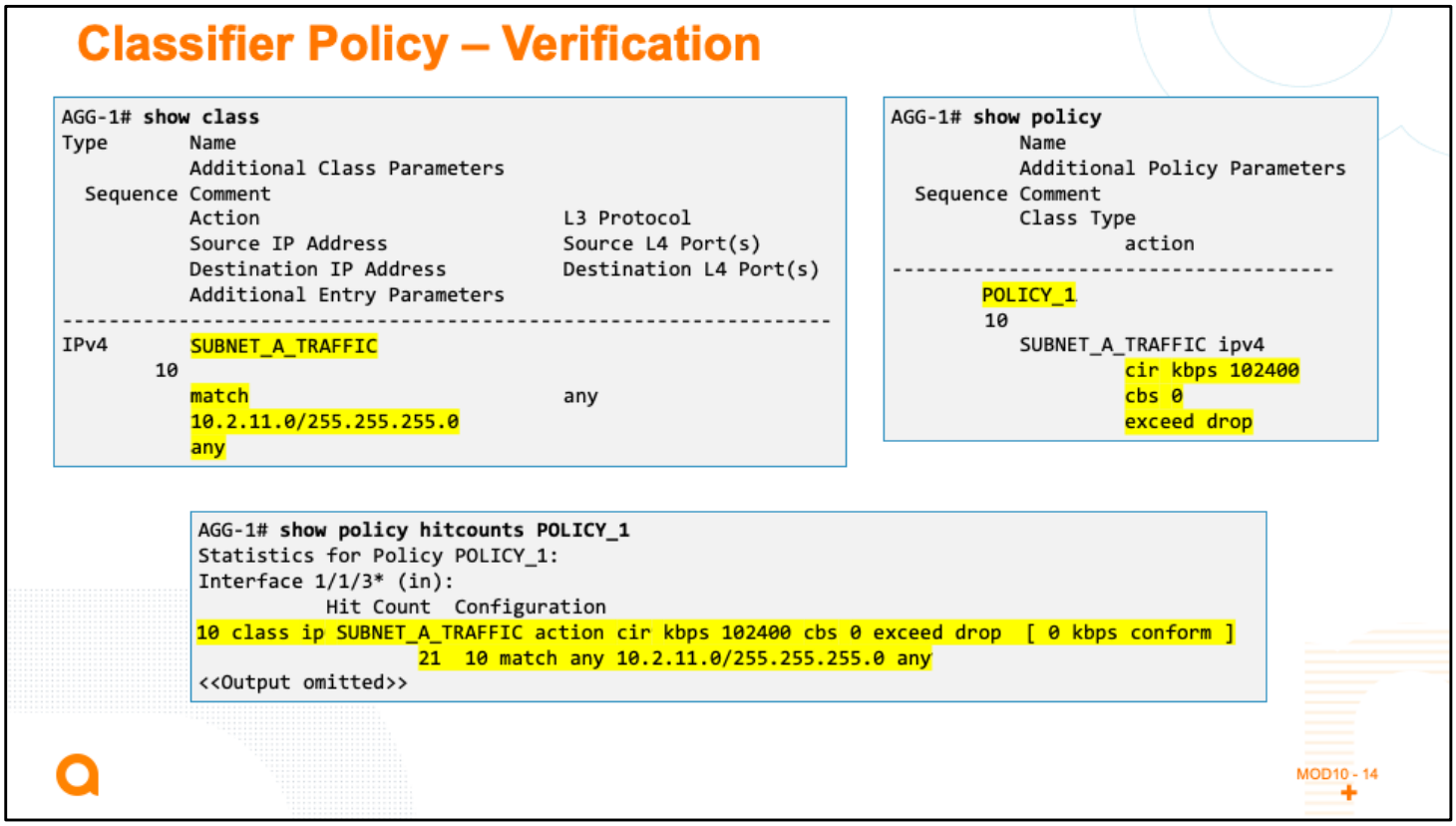
How policy matching works?

A single policy entry matches on one or more characteristics of the particular traffic type and has a configured action to continue through the switch.

This matching occurs by beginning with the entry with the lowest sequence number. The entry is then compared against the incoming frame to its particular match characteristics. If there is a match, the action is taken.

If there is no match, the match characteristics of the next sequence are compared to the relevant frame/ packet details. If there is a match, the specified actions are taken.

This process continues until a match is found; otherwise, the packet is permitted to flow through the switch unaltered. The "implicit permit" behavior of policy matching differs from the "implicit deny" behavior of ACL matching.

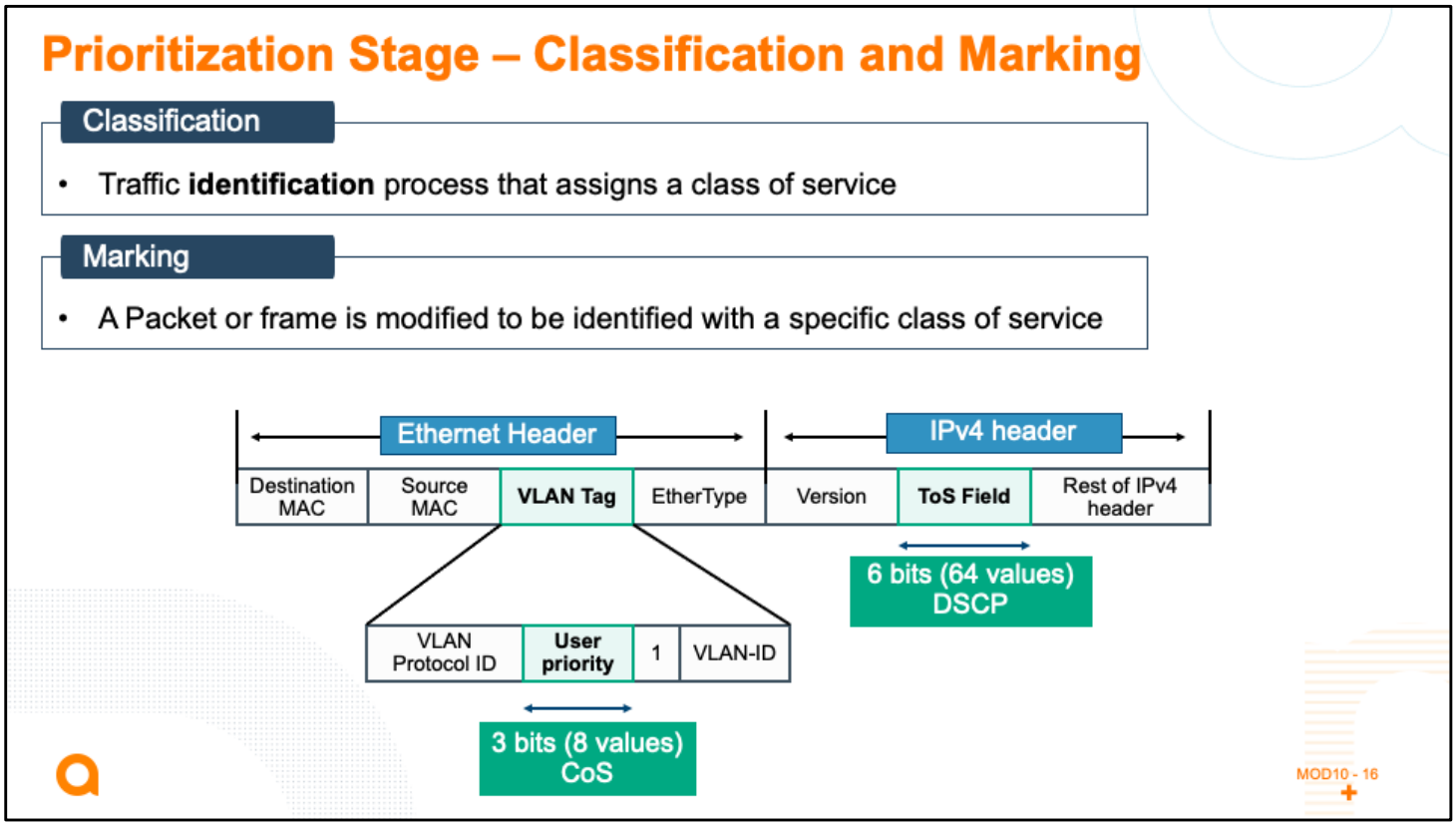


show class and show policy commands can be used to verify the configuration of the classifier policy. The show policy hitcounts command is useful to verify the number of hits on a particular policy.

Prioritization Stage

Classifier and Class of service, Marking, Trust, Configuration and Verification





Classification performs a kind of identification process. This process inspects incoming traffic, identifies it, and then assigns it to a class of service. A classifier has one input, the incoming packet, and it has N possible outputs, where N is the number of possible classes of service.

All traffic belonging to a specific class of service has the same behavioral requirements in terms of the PHB (per-hop-behavior) that should be applied.

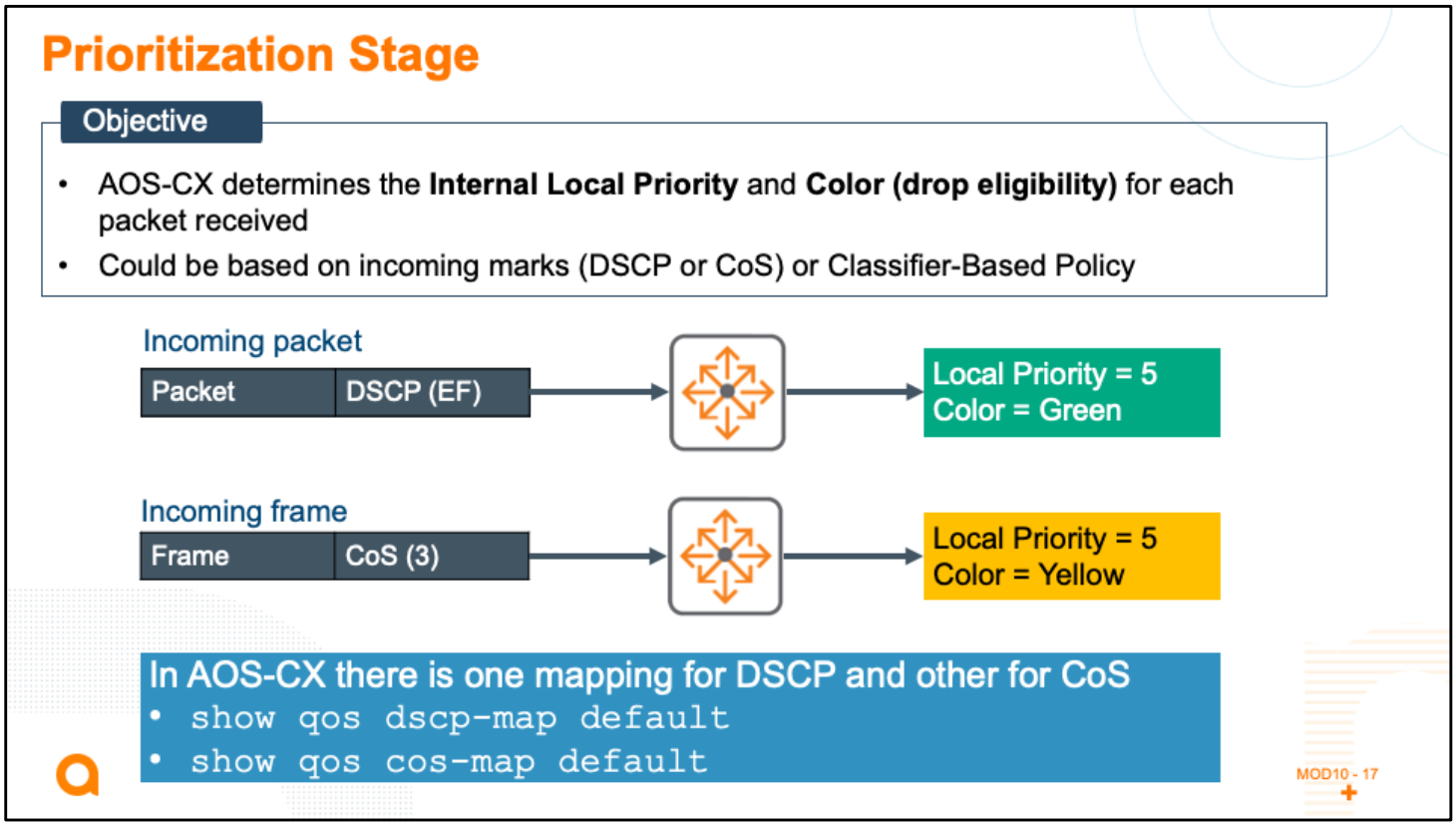
Assigning traffic to different classes of service provides the necessary differentiation. A packet's assigned class of service determines how the router treats that packet – the router's Per-Hob Behavior. The concept of traffic differentiation is present in every QoS tool, so classes of service are present across the entire QOS design.

Marking. This means that a packet or frame has been "modified" to be identified with a specific class of service. This modification can be done in:

User priority field inside the VLAN Tag in an Ethernet header.

ToS field inside the IPv4 header

Traffic Class field inside the IPv6 header (not shown in the figure)



The objective of the prioritization stage is to determine the Internal Local priority value and the Color (drop eligibility) for a specific traffic. These parameters are used by the AOS-CX switch to specify the behavior or the class of service used.

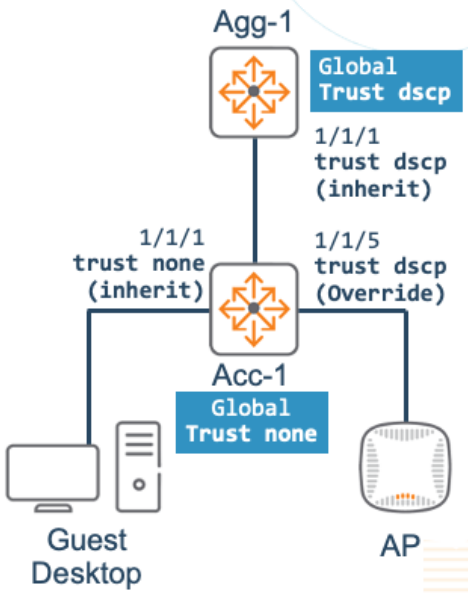
The decision to assign a Local Priority and Color value could be based on the DSCP or CoS marks in ingress packets. Each switch maintains a mapping table that helps with the decision. The default mapping table can be displayed using the show qos dscp-map default command or show qos cos-map default command. AOS-CX supports the creation of custom mapping. The other possibility is to use a classifier-based policy where the interesting traffic is selected based on specific characteristics. In this case the marking may or may not be present.

Trust is an important setting for QoS in AOS-CX. When it is set on the incoming interface the switch can use the incoming marks to determine the Local Priority. However, if the trust is not set (or set to none) then incoming marking cannot be used and all traffic is placed in the Default Best Effort behavior, represented by the Local Priority 1. Notice that the assignment based on the classifier-based policy is not dependent on the trust status and overrides default and custom mapping. The table in the slide summaries this discussion.

Prioritization Stage – Trust

Overview

- Configured on global or interface level
- **Interface** level **overrides global** level configuration
- Possible settings:
  - Trust CoS
  - Trust DSCP
  - None
- An interface supports only one setting at a time
- Default values:
  - Global level: **Trust none**
  - Interface level: **No trust setting**



MOD10 - 18

When traffic arrives at the switch, the switch can trust existing markings or not. The term trust refers to whether the packet markings are valid input for the classifier, which decides each packet's service class.

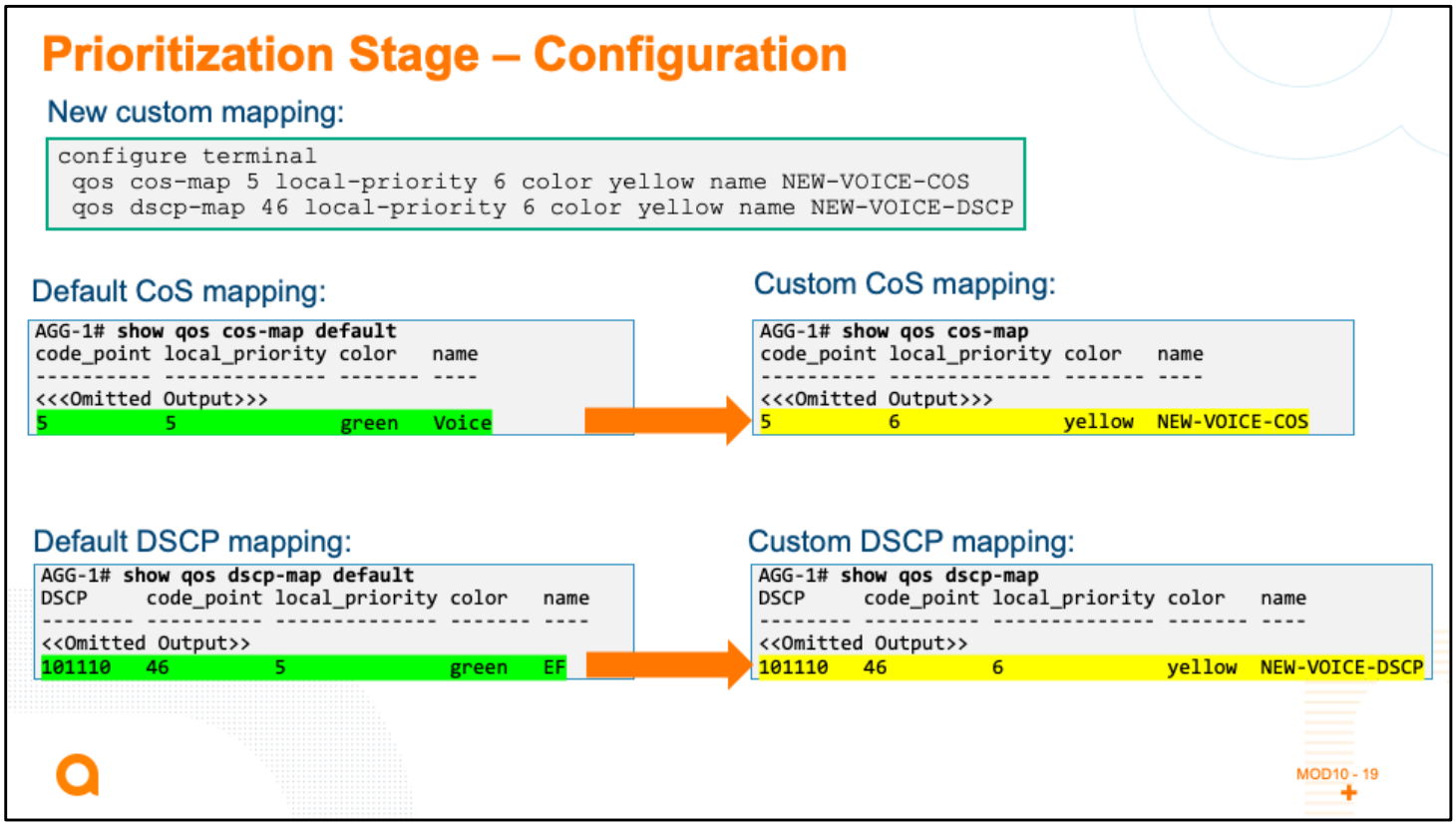
You know that trust configuration plays an important role in assigning Local Priority and Color values. In AOS-CX, you can configure trust at the global or Interface level. When the trust level is set at both levels, the interface level configuration takes precedence.

AOS-CX supports three possible configurations: Trust CoS, Trust DSCP and Trust none. The first two options act as the name suggests - the switch accepts the incoming marking and uses it to determine class of service. In AOS-CX 10.5 an interface can only support one trust setting at the time. The means that you can set DSCP or CoS, but not both, at the same time on the same interface.

It is not recommended to trust ingress markings, especially if traffic is received from non-trusted devices. In these situations, you can protect the network by ignoring the incoming marking setting the trust setting to none.

By default, the global setting is "none," and all ports have no trust setting.

Note. It is important to clarify that a port not having a trust configuration is different from the port having a configuration of none. If a port has no setting, the port takes its setting from the global setting. Otherwise, the port setting, including "none," overrides the global setting.



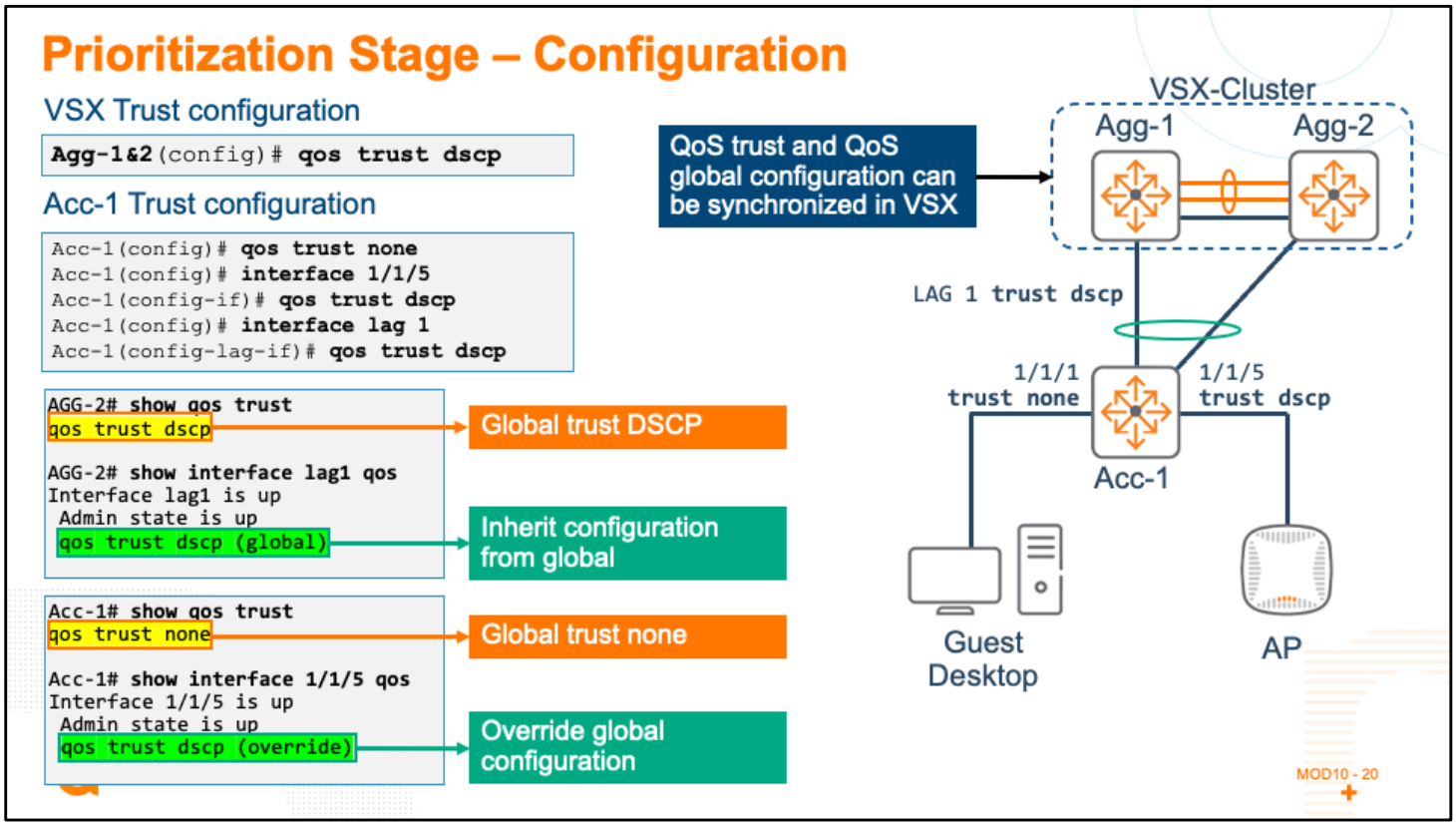
Use the show qos cos-map default command to verify the default mapping for incoming CoS marks. Use show qos dscp-map default to display the default mapping for incoming DSCP marks.

AOS-CX allows you to customize these default mappings, in this example, suppose you want to modify the default mapping for the voice marks using the following commands:

qos cos-map 5 local-priority 6 color yellow

qos dscp-map 46 local-priority 6 color yellow

This action overrides the default mapping, which you can verify with the show qos cos-map or show qos dscp-map commands.



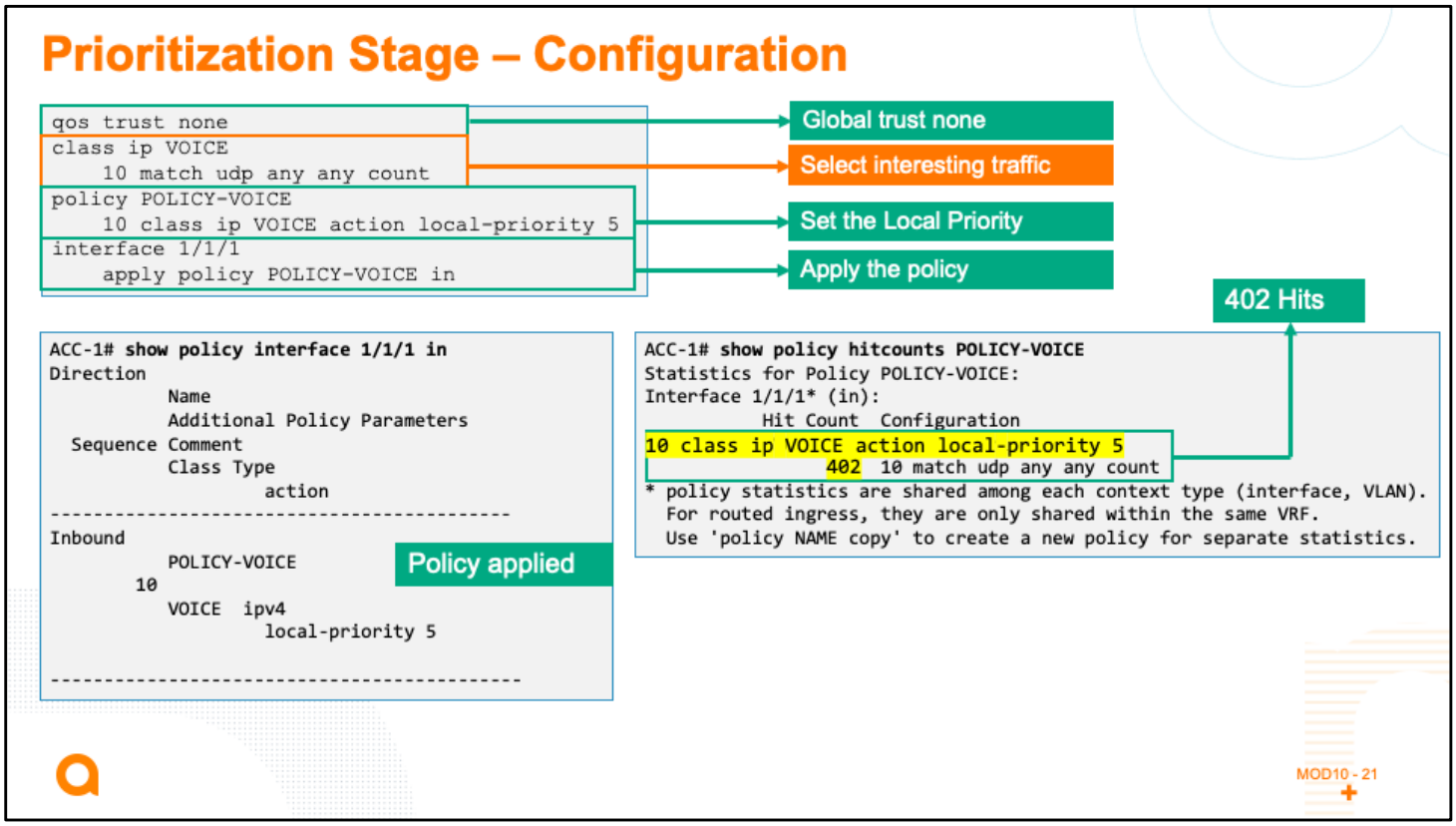
This example shows the configuration of a VSX cluster and an access switch. Because the VSX cluster faces trusted devices, it can use ingress traffic markings. Therefore, a global configuration trusting DSCP markings is more than enough.

The Access-1 switch is set globally to not trust ingress markings. This helps to secure the network by avoiding unwanted traffic prioritization. Then you decide which interfaces to configure to override this global markings. In the figure, you have configured interface 1/1/5 to trust ingress DSCP settings. This port connects to an Access Point (AP) which you control, so you trust its ingress markings. The administration team needs to define the trust boundary to ports that the received marking can be processed.

You can use the show qos trust command to verify the global configuration and show interface <interface-id> qos command to verify the configuration on each individual interface.

Note. The qos trust level can be synchronized with the VSX config sync as well 'qos-global'.

Note. For deployments using port authentication, the QOS trust level can be included in the Aruba user role that is applied to the authenticated device. This ensures dynamic configuration of the access port trust level.



This example shows how you can use a classifier-based policy configuration to assign the Local Priority.

```
! set the global trust configuration to none
qos trust none
! select the interesting traffic
class ip VOICE
    10 match udp any any count
! set the local priority 5 to the interesting traffic
policy POLICY-VOICE
    10 class ip VOICE action local-priority 5
! apply the policy to the incoming traffic received on interface 1/1/1
interface 1/1/1
    apply policy POLICY-VOICE in
```

Use the show policy interface <interface-id> in command to verify if the policy was correctly set on the interface. The command show policy hitcounts <Policy-name> displays the number of hits counts that the policy receive.

Question #1

What is the objective of Virtual Output Queuing (VOQ)?

a. AOS-CX uses this technique to buffer traffic when inbound port congestion reaches 75%

b. Provides deep ingress buffer with separate queues for each egress port

c. Performance is improved by using more than 8 virtual queues in the outbound port

d. Packets stay in the inbound port queue when outbound queue is under load

Knowledge Check

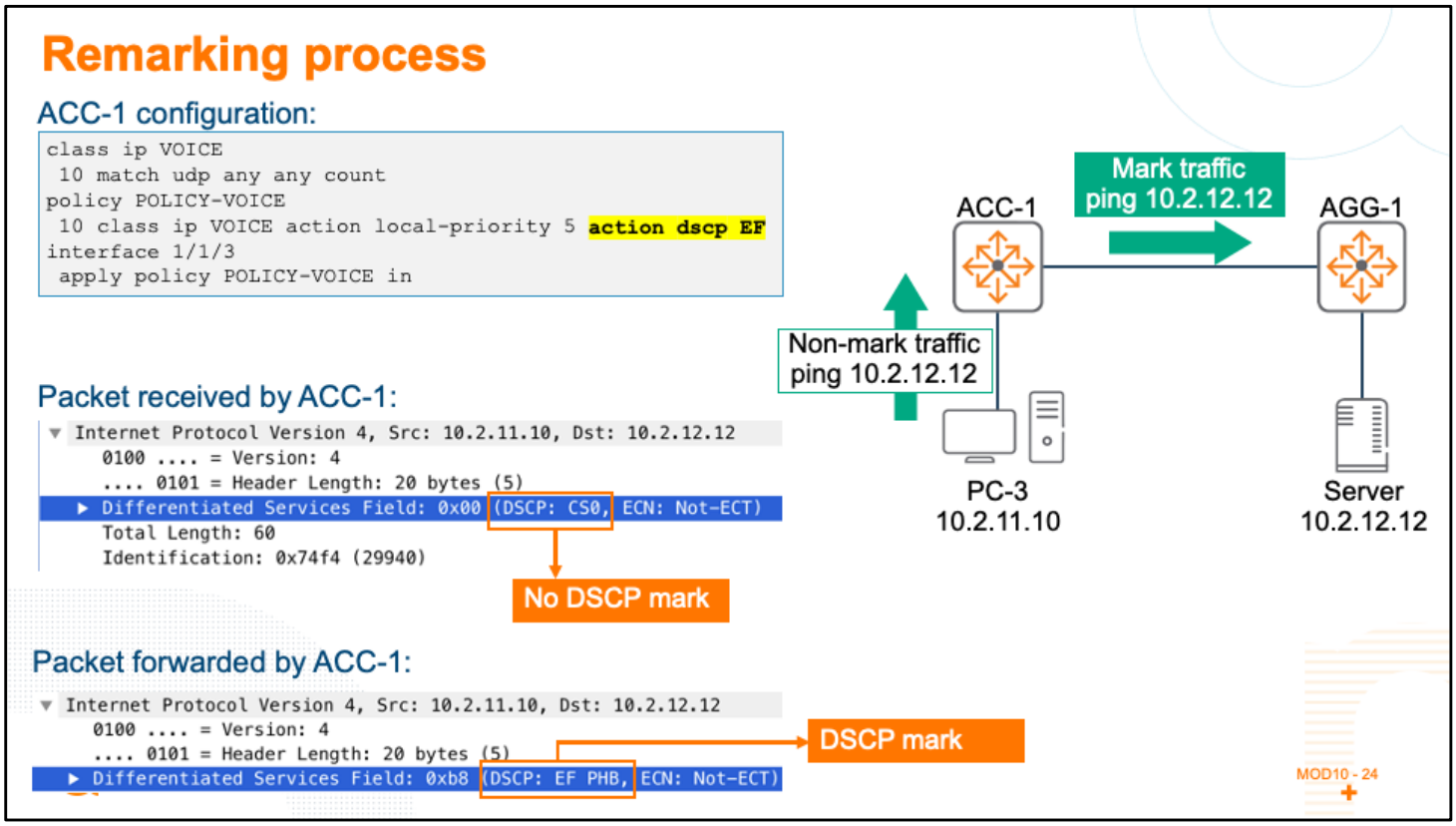
Correct answer:

B and D

Remark QoS

Configuration and Verification





As we discussed previously the PHB concept share information among network devices changing the packet's or frame's headers by using the rewrite QoS tool (marking process). In this way QoS does not need to define a special protocol to advertise to the next-hop how a packet should be treated.

This process can be done in AOS-CX using the Classifier-based Policy tool. In the example you can see a basic configuration of QoS using a Classifier-based Policy the difference now is that we add a new action in this case marking the traffic as EF. By applying this change, we can now properly mark the traffic that is sent out.

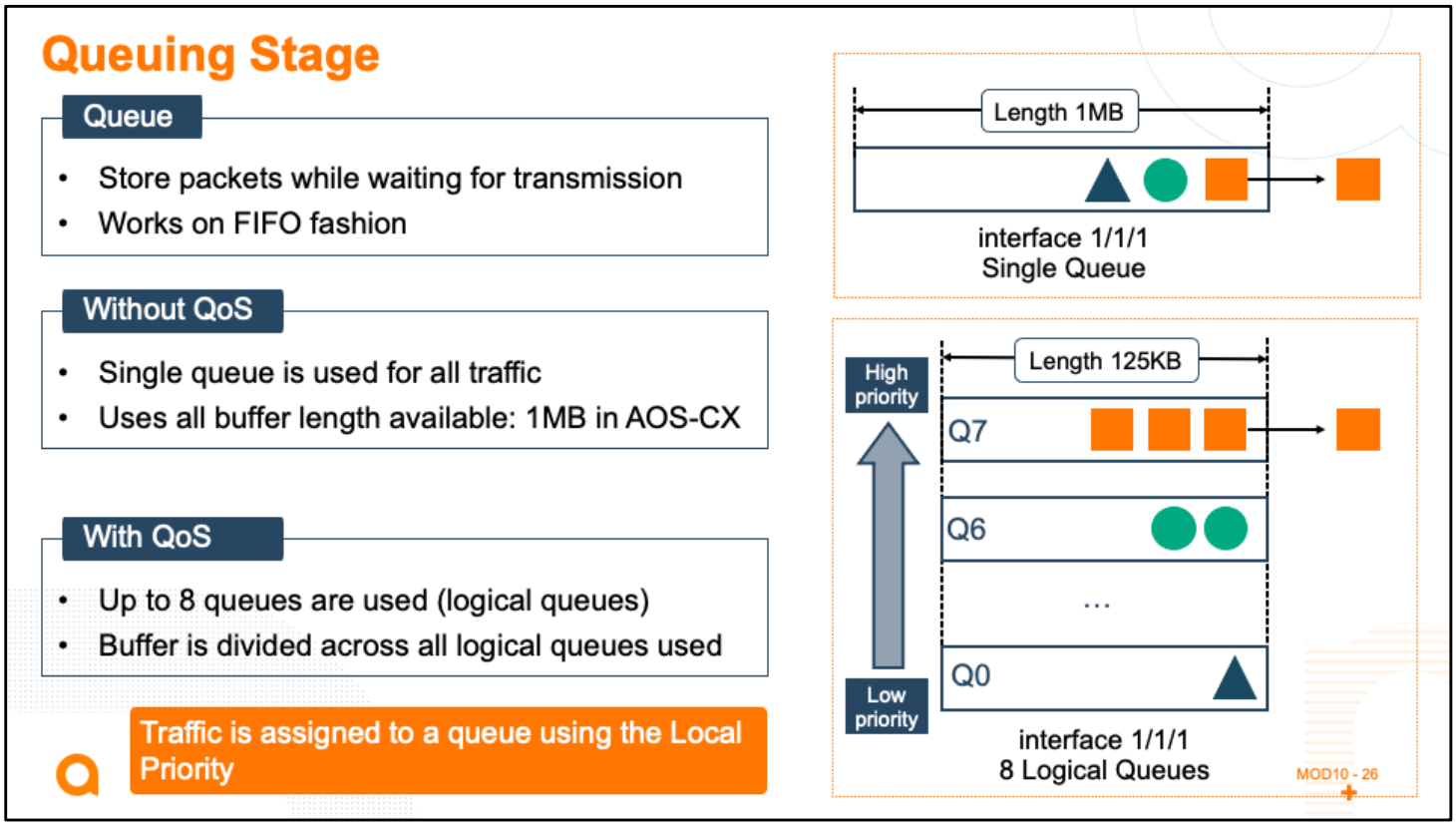
The Wireshark captures confirm the remarking process.

Note. Remark DSCP/COS will only change the mark, not the internal local priority.

Queuing Stage

Configuration and Verification

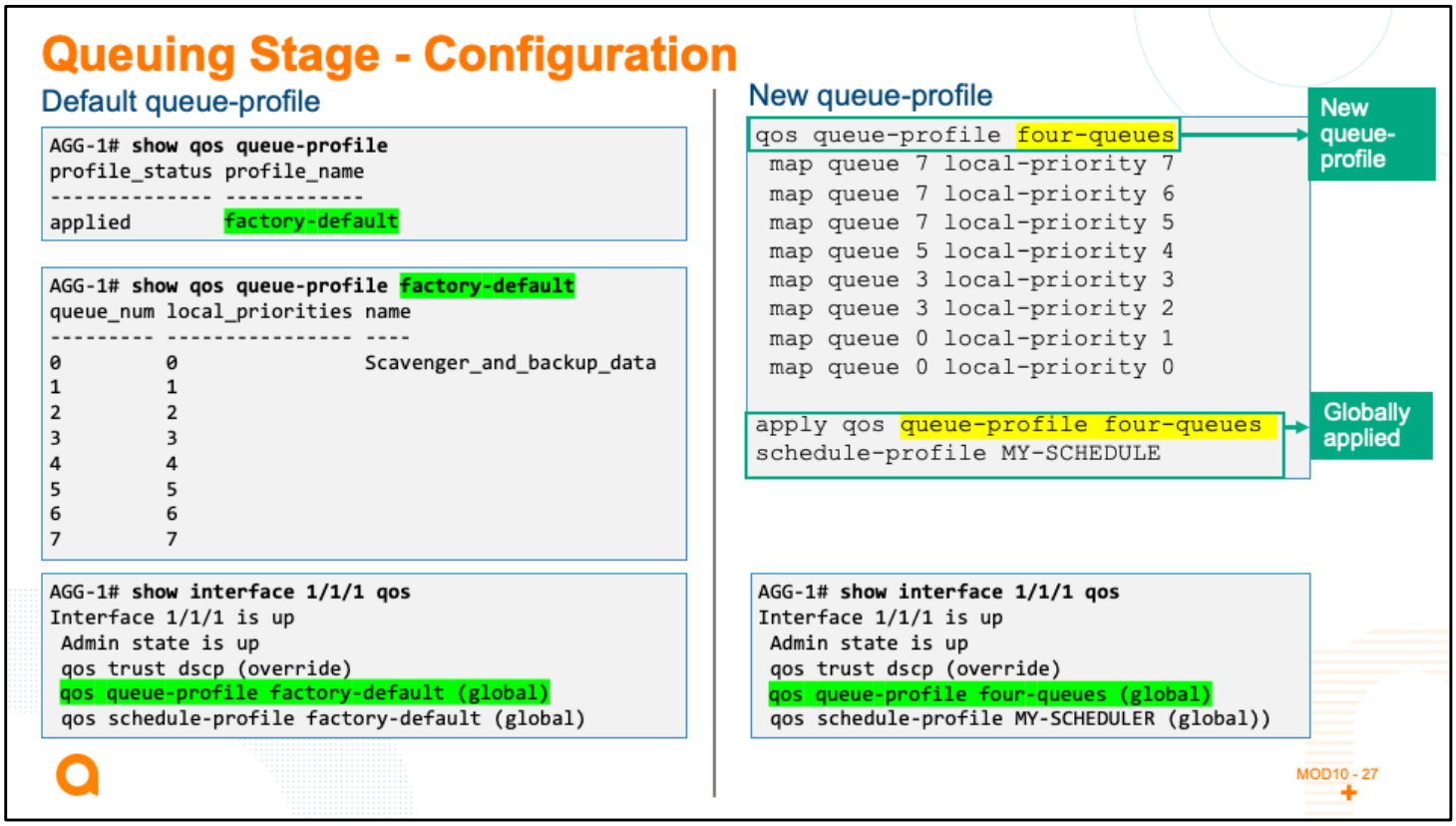




When a switch determines the egress interface, the traffic is not sent out immediately. The switch places all egress packets in a queue, where they briefly wait to be transmitted. The queue works in a First In First Out (FIFO) fashion. This means that the switch sends packets in the same order they were received. The main parameter that defines the queue is its length, which is how many packets it can store.

Normally, without QoS, all traffic is placed in a single queue. In AOS-CX each interface has an egress buffer up to 1MB. As you might imagine, using a single queue does not offer any advantage to sensitive traffic.

When QoS is deployed the egress queue is divided into logical queues. Differentiated traffic is then placed in a specific logical queue based on its priority. In AOS-CX you can use up to 8 logical queues and traffic is assigned to each using the Local Priority. Most important or sensitive traffic is placed in higher queues while the less important or delay tolerant traffic is placed in lower queues. The decision regarding the order in which multiple queues are served is the objective of the scheduler.



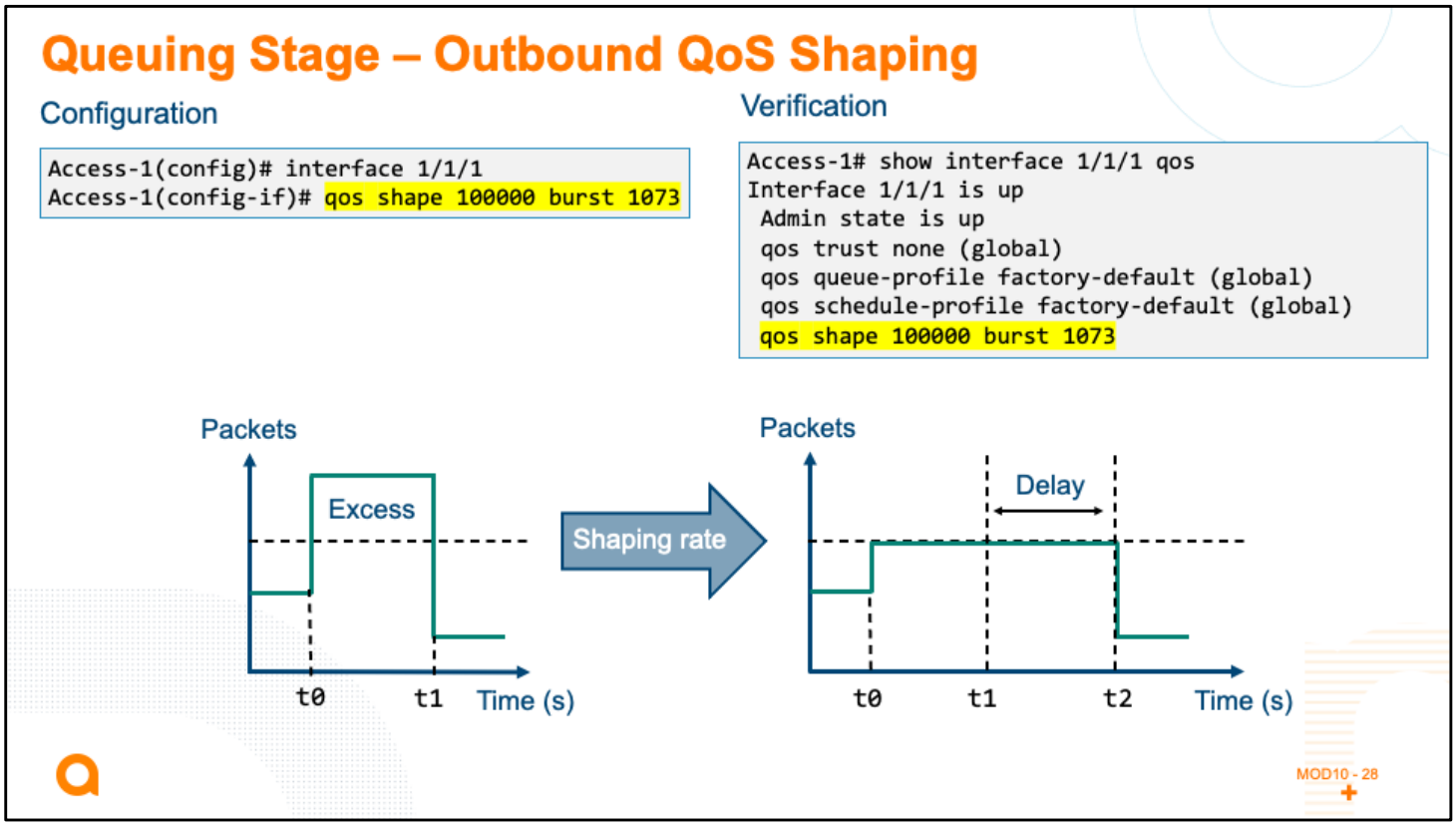
The queue-profile contains the mapping between the Local Priority and the queues. AOS-CX uses a default queue-profile named “factory-default”, which uses a direct mapping (1:1). Use the show qos queue-profiles command to display the profiles available. To display the mapping, use the show qos queue-profile <profile-name> command. To verify the queue-profile used on each interface, use the show interface <interface-id> qos command.

AOS-CX allows you to create a new queue-profile with the qos queue-profile command.

As an example, we create a queue-profile with only four queues:

qos queue-profile four-queues
map queue 7 local-priority 7
map queue 7 local-priority 6
map queue 7 local-priority 5
map queue 5 local-priority 4
map queue 3 local-priority 3
map queue 3 local-priority 2
map queue 0 local-priority 1
map queue 0 local-priority 0

To apply the new queue-profile, use the command apply qos queue-profile <queue-profile-name> schedule-profile <schedule-profile-name>. You can execute this command globally or at the interface level. When applied globally, the specified schedule profile is configured only on Ethernet interfaces and LAGs that do not already have their own schedule profile.



This example shows how to apply egress shaping to an interface to limit the rate of egress traffic. To be effective, the egress rate must be less than the line rate of the egress interface. If the configured egress rate exceeds the line rate of the interface, then egress shaping is effectively disabled.

The following example configures an egress rate of 100 Mbps and a burst of 70 KB.

switch(config)# interface 1/1/1

switch(config-if)# qos shape 100000 burst 1073

The shaper feature causes a traffic flow to conform to a bandwidth value referred to as the shaping rate. Excess traffic beyond the shaping rate is stored inside the shaper and transmitted only when doing so does not violate the defined shaping rate.

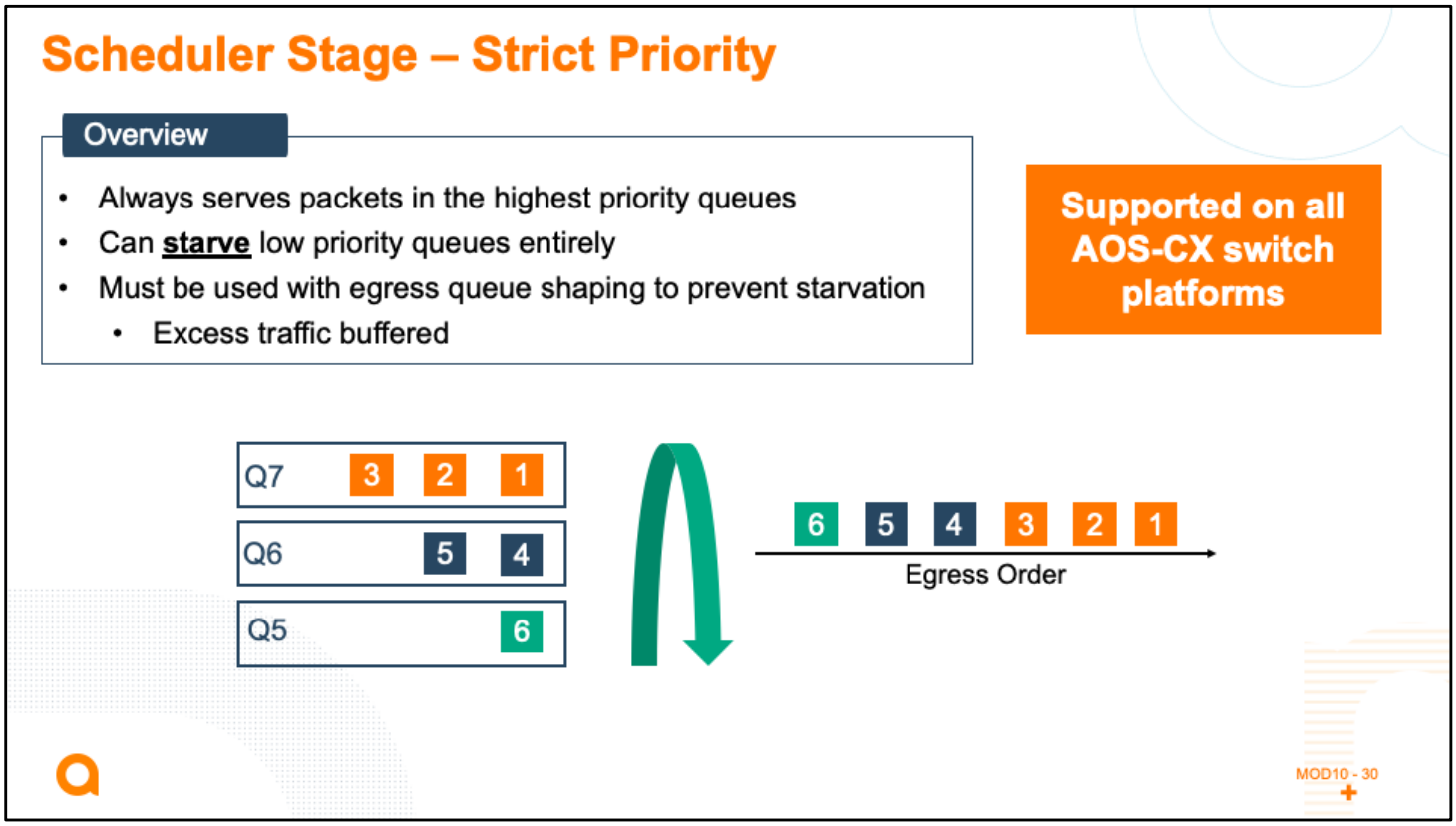
Comparing the input and output traffic flow graphs, the most striking difference is the height of each. The height of the output traffic flow is lower, which is a consequence of complying with the shaping rate. The other difference between the two is the width. The output traffic flow is wider, because the traffic flows for a longer time, because of how the shaper deals with excess traffic.

The use of shaping to store excess traffic instead of dropping it has earned it the nickname of “TCP friendly,” because for connection-oriented protocols such as TCP it is usually better to delay than to drop packets.

Scheduler Stage

Strict Priority, DWRR, WFQ, Configuration and Verification





Strict Priority scheduling always forwards the highest priority traffic first. That is, the port forwards all packets in queue 7. When that queue is empty, the port forwards all packets in queue 6. If that queue does not have any packets queued, the port forwards all packets in queue 5, and so forth.

Under Strict Priority, the port will never forward a packet in a lower priority queue if a higher priority queue includes even one packet. This behavior could potentially starve low priority queues. Packets in low queues could experience a major latency, potentially resulting in packet aging. Strict Priority works best when the network uses the higher priority queues for only a few applications, which take a relatively low amount of bandwidth. Common examples are network control traffic and voice over IP (VoIP).

Strict priority benefits

The SP algorithm provides a simple method of supporting differentiated service classes, in which queues are serviced according to the priority assigned. The buffering and scheduling computational requirements are low and not very complex, even when implementing SP on network equipment with high-speed links.

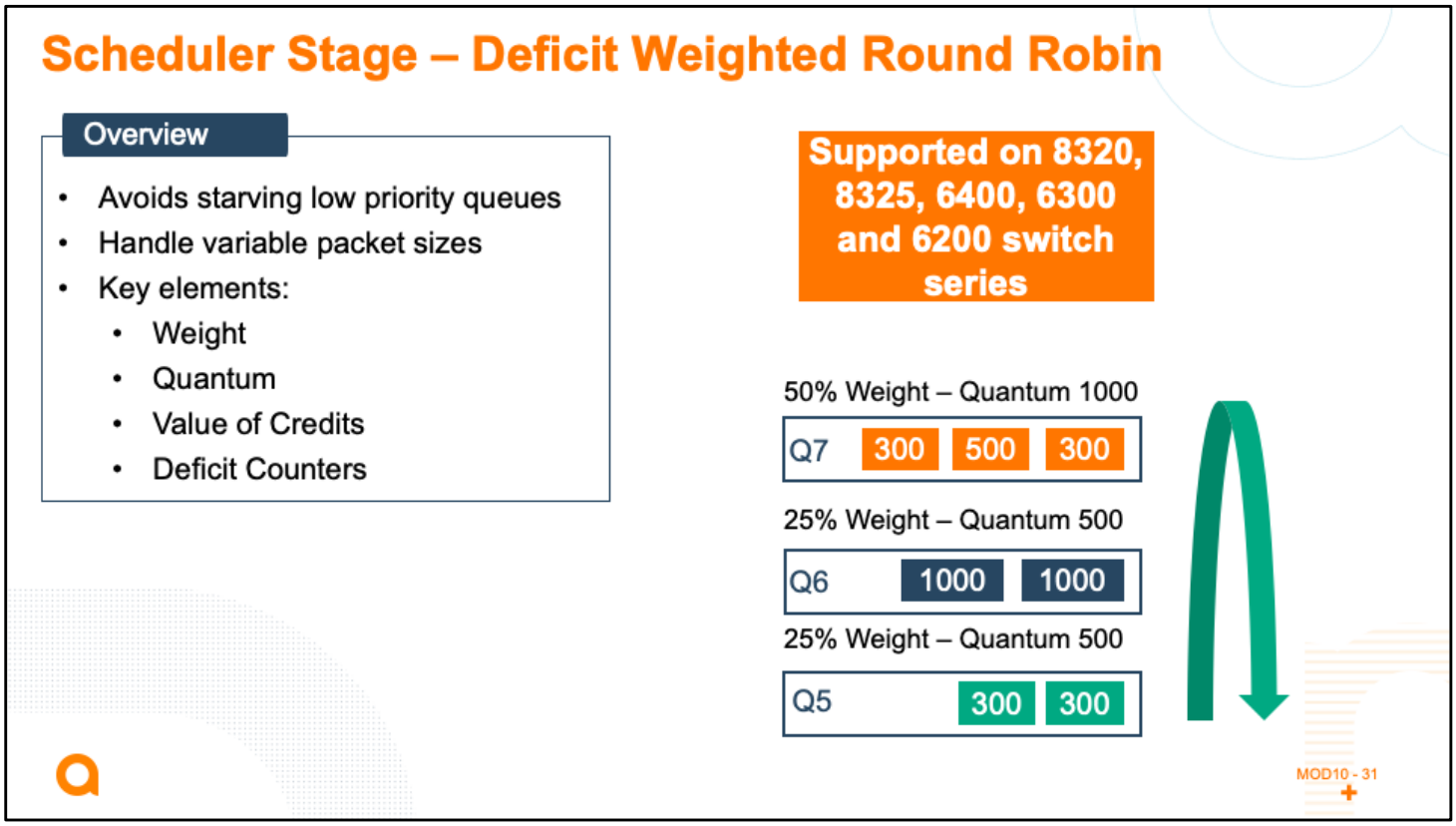
Low-delay and loss-sensitive applications such as real-time traffic can be effectively protected from other greedy traffic flows such as TCP sessions, and low delay and jitter can be maintained for the traffic classified as high priority.

Strict priority limitations

Because of its absolute allocation of resource and scheduling services to the priority-classified traffic, the SP algorithm can result in network malfunctioning because low-priority traffic becomes stalled, in essence this can introduce similar behavior as head of line blocking. The cost of giving certain traffic a higher priority comes at the expense of penalizing lower-priority traffic.

High-priority traffic must be very well provisioned and rate-controlled. Otherwise, service breakdowns can result for all traffic that is not high priority.

With AOS-CX, you can apply egress queue shaping to the high priority queues to prevent starvation of low priority queues. You can apply a maximum bandwidth to a priority queue, and a burst size. The port buffers excess traffic up to the burst size and sends the buffered traffic at the max rate, smoothing out bursts while also preventing the high priority queue from exceeding its maximum rate and starving out lower priority queues.



Deficit algorithms can handle packets of variable size without knowing the mean size. A maximum packet size number is subtracted from the packet length, and packets that exceed that number are held back until the next scheduling turn.

Deficit Weighted Round Robin (DWRR) serves packets at the head of every non-empty queue whose deficit counter is greater than the size of the packet at the head of the queue. If the deficit counter is lower, the queue is skipped and its credit value, called a quantum, is increased. The increased value is used to calculate the deficit counter the next time around when the scheduler examines the queue for service. If the queue is served, the credit is decremented by the size of packet being served.

There are four key elements that affect how DWRR schedules queue service:

Weight. Reflects a proportion of the bandwidth on the outgoing interface.

Quantum. Translates the weight value into bytes. With each scheduling turn, a value of quantum is added in proportion to the weight. Thus, the quantum is the throttle mechanism that allows the scheduling to be aware of the bandwidth.

The value of credits can be positive or negative. Positive credits accrue when, at the end of a scheduler turn, there are leftover bytes that were not used in the queue's scheduling turn. This value is deferred to the queue's next scheduling turn. Negative credits accrue when the queue has transmitted more than its bandwidth value in a scheduling turn, and thus the queue is in

debt when the next scheduling turn comes around.

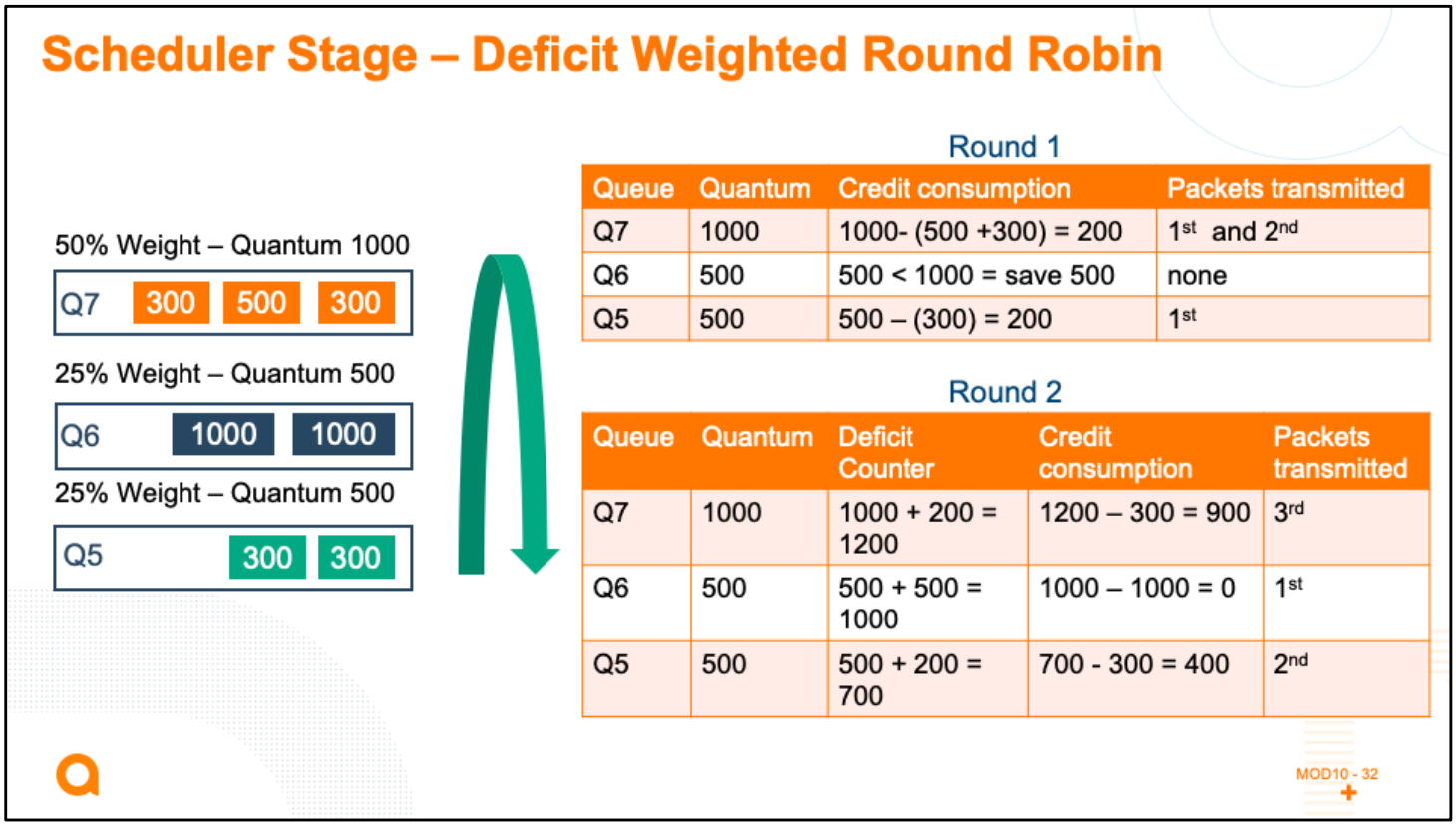
The deficit counter, which provides bandwidth fairness, is the sum of the quantum and the credits. The scheduler removes packets until the deficit counter reaches a value of zero or until the size of the packets is larger than the remaining deficits. But if the queue does not have enough deficits to schedule a packet, the value of the deficit is retained until the next scheduling round, and the scheduler moves to the next queue.

Benefits:

The DWRR algorithms limit the shortcomings with WRR because they are a more modern form of Weighted Fair Queuing (WFQ) that incorporates scheduling aware of bits and packets.

Limitation:

Services that have very strict demand on delay and jitter can be affected by other queues by the scheduling order. DWRR has no way to prioritize its scheduling.



Consider the scenario shown in the figure, with three queues with different weights that reflect the proportion of bandwidth of outgoing interface. Q7 has 50% of the bandwidth, and Q6 and Q5 each have 25%. To simplify the example, all queues have an equal length. The queues have variable-sized packets and different numbers of packets. At each scheduling turn, credits are added to the quantum number of each queue that reflect the weight as a quota of the bandwidth.

First round

Q7 has three packets, two that are 300 bytes and one that is 500 bytes. Each scheduling turn, a quantum value of 1000 is added to Q7. In this turn, then, the deficit counter is 1000, which means that 1000 bytes can be subtracted from the queue. The 300-byte and 500- byte packets can be transmitted because they consume 800 credits from the 1000-credit bucket. But the last 300-byte packet cannot be transmitted in this turn, because 300 is more than the remaining 200 credits (1000 − 800 = 200). The result is that the scheduling jumps, in order, to the next queue with packets inside, and the 200 credits for Q7 are saved for next scheduling turn.

Q6 contains two 1000-byte packets, but the quantum per scheduling turn is 500 credits. This means that no packets can be transmitted in this scheduling cycle because not enough credits exist (500 < 1000). The result is that the deficit counter for Q1 is incremented with 500 credits for the next scheduling round. This example illustrates the power of deficit-style algorithms compared with plain round-robin systems. With DWRR, Q6 is punished because the available

quantum and credit values are smaller than the actual packet sizes. This situation is proof that DWRR scheduling is aware of variable packet sizes, not just the number of packets to be queued.

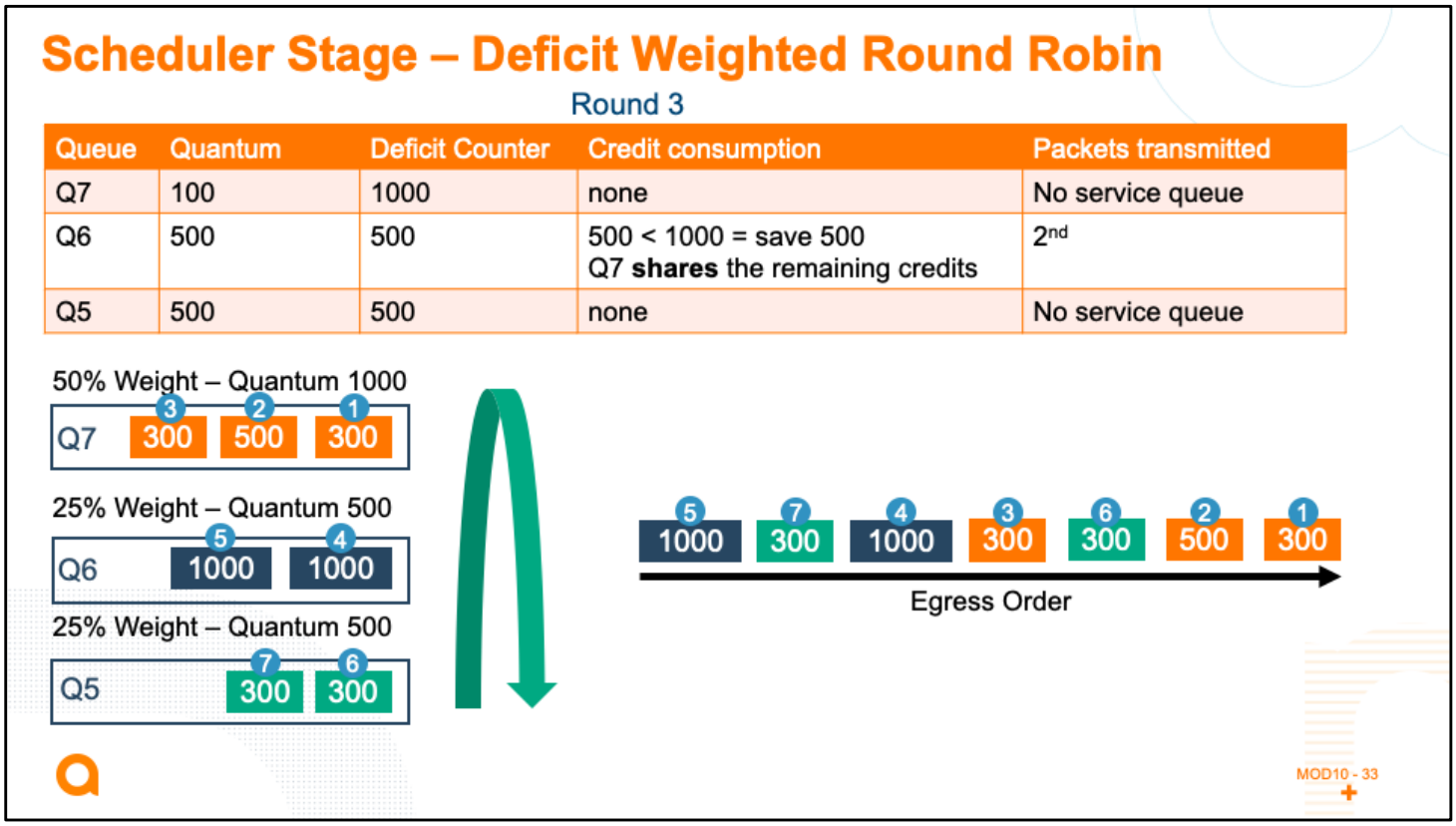
Q5 has a quantum value of 500, and because the packet at the head of the queue is 300 bytes, it is transmitted. The value of 300 is subtracted from the 500, resulting in 200 credits that are saved for the next scheduling turn. The scheduling wheel has now completed one full turn.

Second round

The Q7 deficit counter is now 1200. The 300-byte packet is transmitted, and its value is subtracted from the deficit counter, resulting in a new value of 900 credits. While implementations can vary, a common behavior is to set the deficit counter to zero if no packets are in the queue. Queues that behave this way are honored by the algorithm, but they receive no extra points for no work. You cannot get fame and glory unless the queue is filled with packets. Now the scheduler returns to the troubled Q6, from which no packets were transmitted in turn 1.

Finally, there is some good news for Q6. The updated quantum value is 500, and with the 500 credits saved from cycle 1, for a total of 1000 credits, Q6 can transmit one of the 1000-byte packets.

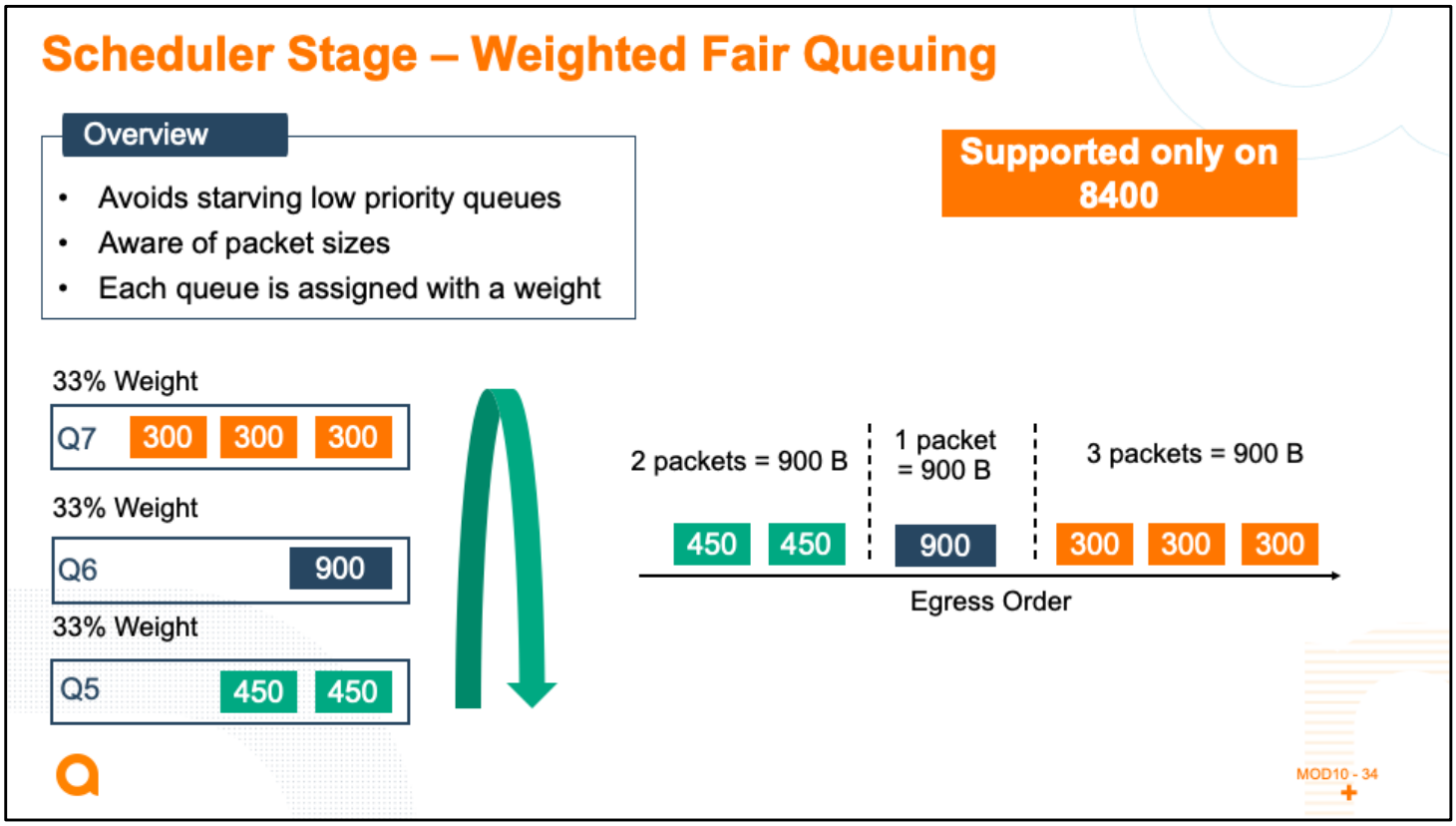
Q5 the deficit counter is far bigger than the number bytes in the queue, so the last packet is transmitted



Third round

Let us look at a third scheduling turn, which illustrates an interesting DWRR scenario. Q7 has no packets to schedule, so it is not serviced, and the scheduler moves to Q6. This queue has the same problem, with big packets eating up the deficit counter, and Q6 cannot transmit its packet. Scheduling turns, that is, bandwidth that has not been used by one queue can be shared with other queues. So for poor Q6, this is good news. The extra available quantum of credits from Q7 is shared with other queues that have packets to be transmitted, with the leftover quantum divided between the queues based on their weight: the more weight, the more of the extra quantum credits are allocated. Because Q5, had no packets in its buffer, the entire leftover quantum for Q7 is given to Q6.

The image in the slide shows the Egress order of each packet.



Weighted fair queuing (WFQ) is commonly referred as “bit-by-bit round robin,” because it implements a queuing and scheduling mechanism in which queue servicing is based on bits instead of packets.

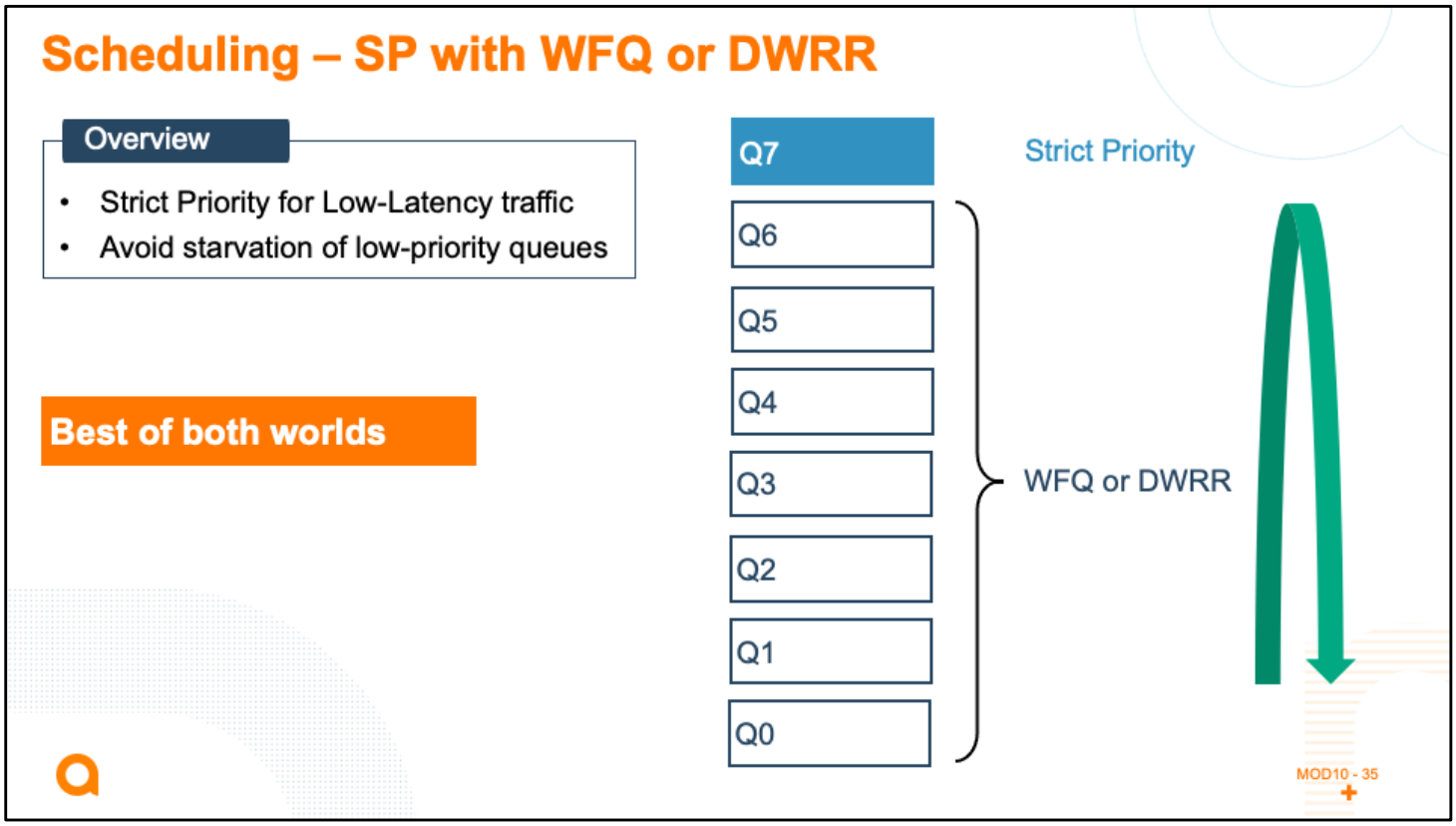
In WFQ, each queue or flow is allocated a weight that is a proportion of the interface rate or the shaping rate. WFQ is aware of packet sizes and can thus support variable-sized packets. The benefit is that sessions with big packets do not get more scheduling time than sessions with smaller packets, because effectively the focus of WFQ is bits and not packets. So there is no unfairness in the scheduling for sessions with smaller packet sizes. With WFQ, each queue is scheduled based on a computation performed on the bits of each packet at the head of the queue.

WFQ brings queue weight into the calculation of the virtual finish time. You can approximate how this process works by thinking of each queue as having its own bandwidth, which is its share of the port bandwidth relative to its weight. Based on this weighted bandwidth, queues with higher weight will have relatively sooner finish times for their packets than queues with lower weights.

In this example, all queues have the same weight, and thus the number of bytes scheduled in each scheduling turn is the same for all queues and reflects the weight value. Q7 removes three packets with a total value of 900 bytes, Q6 removes one packet with 900 bytes, and Q5 removes two packets with 900 bytes. The weight factor is effectively an allowance for how

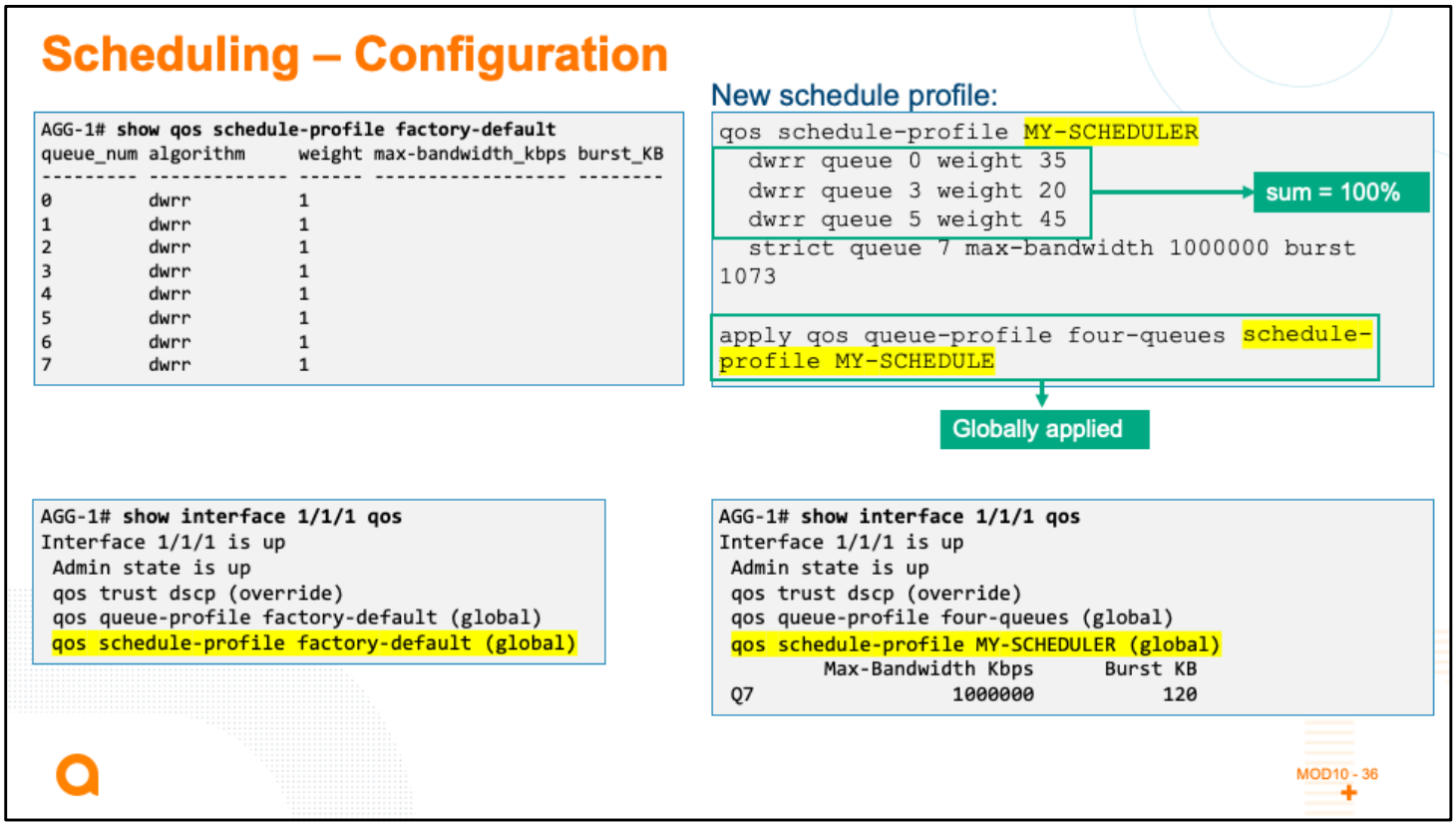
many resources can be assigned and used.

Note: Queuing does not fragment frames. If a queue could transmit 450 more bytes, but there is a 900-byte frame in the queue, it is not transmitted.



If the network has traffic that requires very low latency and jitter, such as VoIP traffic, you may want to use a Strict Priority (SP) queue for that traffic. This is because SP does give high priority traffic the lowest latency and jitter. However, you may not want to use SP for medium high priority queues due to the risk of starving out other traffic.

AOS-CX switches give you the best of both worlds. You can create a schedule profile that combines either DWRR or WFQ, depending on the switch model, with Strict Priority. In a hybrid profile, only the highest priority queue can use SP. This is queue 7 if the profile uses all queues or the highest queue used if the profile does not. You should assign traffic like VoIP to this queue; network control traffic is also automatically assigned to it. The switch forwards traffic from this queue as long as the queue has traffic. When the queue is empty, the port uses the remaining bandwidth to serve other queues. It allocates this remaining bandwidth using DWRR or WFQ. Because VoIP and network control traffic typically have low bandwidth requirements, the risks of starving the other traffic is minimal.



AOS-CX uses a default schedule profile named “factory-default”, to display this profile you can use the show qos schedule-profile factory-default command. You can use the show interface <interface-id> qos command to verify the schedule profile used at the interface level.

You can setup a new schedule profile, you can use the qos schedule-command command for this task.

As an example, we create a schedule-profile with only four queues:

qos schedule-profile MY-SCHEDULER

dwrr queue 0 weight 35

dwrr queue 3 weight 20

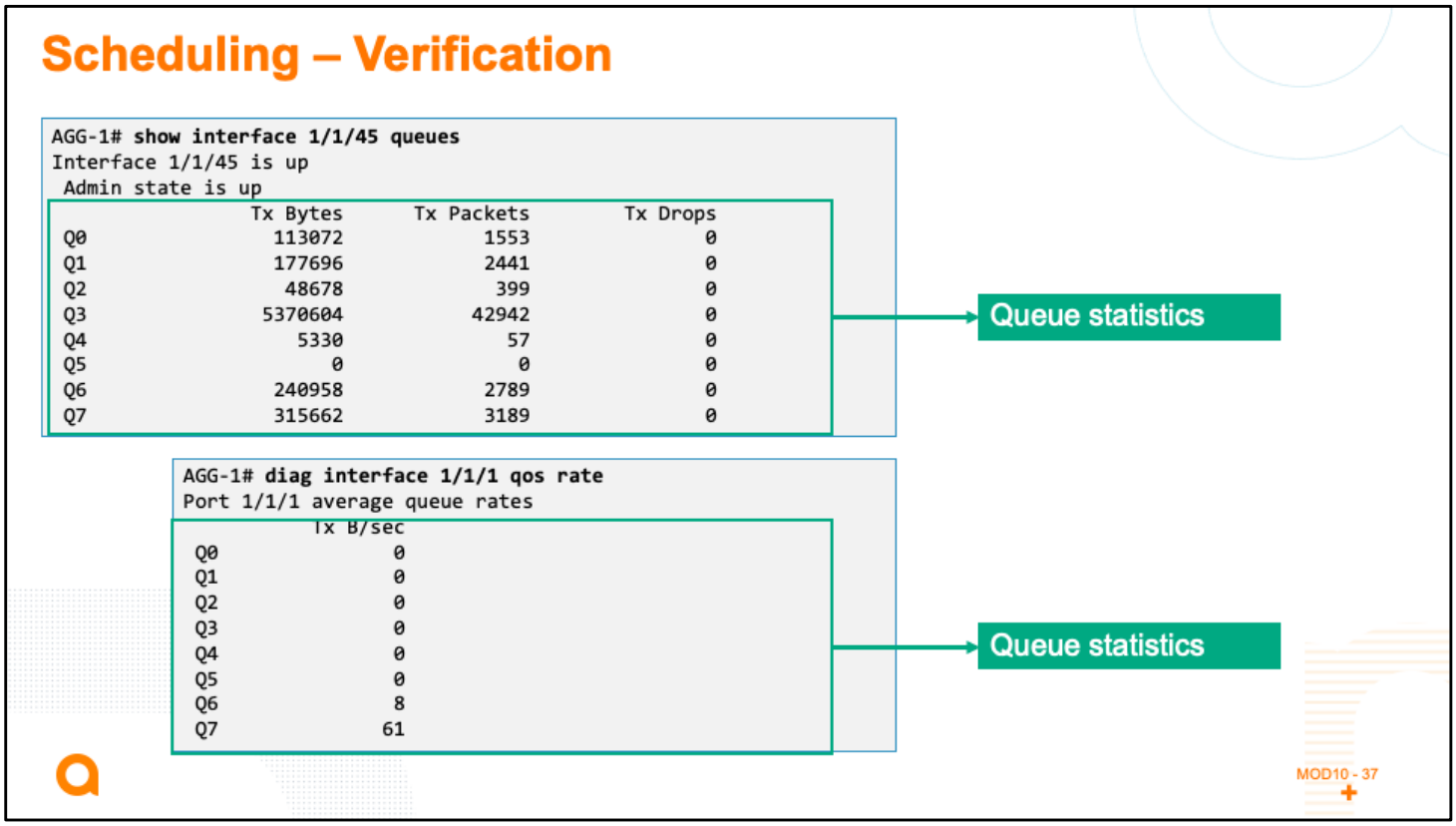
dwrr queue 5 weight 45

strict queue 7 max-bandwidth 1000000 burst 1073

There are two important points to mention about this configuration. First the sum of the weight values in the configuration is 100. When this happens the weight represents a percentage of the entire available bandwidth. The second point relates to the maximum bandwidth that queue 7 can use, in this case limited to 1000 Mbps with a burst size of 1073 kilobytes.

To apply the new queue-profile use the command apply qos queue-profile <queue-profile-name> schedule-profile <schedule-profile-name>. You can execute this command globally or per interface level.

Note. The burst option for the strict queue command is not available in 6300 and 6400 switch series.



For final QoS verification, use the show interface <interface-id> queues command, which verifies the queuing statistics on a specific interface. You can also use diag interface <interface-id> qos rate command.

Question #2

```
(config)# class ip bulk-data
(config-class-ip)# 10 match tcp any any eq 8000 count
(config-class-ip)# exit
(config)# policy asts-qos
(config-policy)# 10 class ip bulk-data action dscp CS1 action pcp 2
```

From the previous configuration, what would be the Marking used in the Layer 2 header for a packet using the TCP with destination port 8000?

a. No marking will be used
b. DSCP CS1
c. CoS value 2
d. DSCP 0



Correct answer:
C

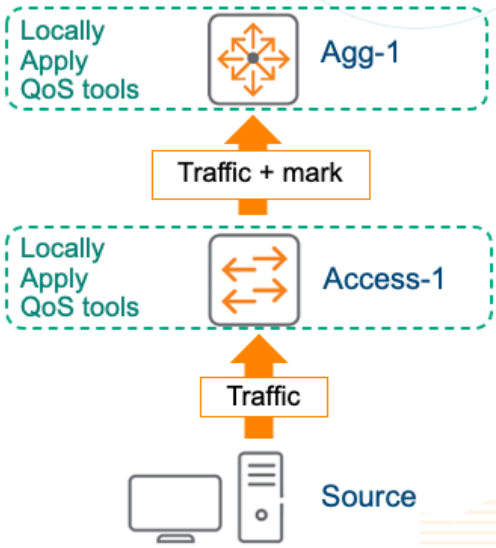
aruba
a Hewlett Packard Enterprise company

Additional Content

Differentiated Services (DiffServ) Model

Overview

- Defined on RFC 2475
- Network traffic identified by class
- Benefits:
  - Flexibility
  - Ease of implementation
  - Scalability
- Define a **Per Hop Behavior** (PHB)
  - Each network device define a local configuration, but devices act as **team**.
  - No special protocol between network devices, instead packets and frames are marked



MOD10 - 40

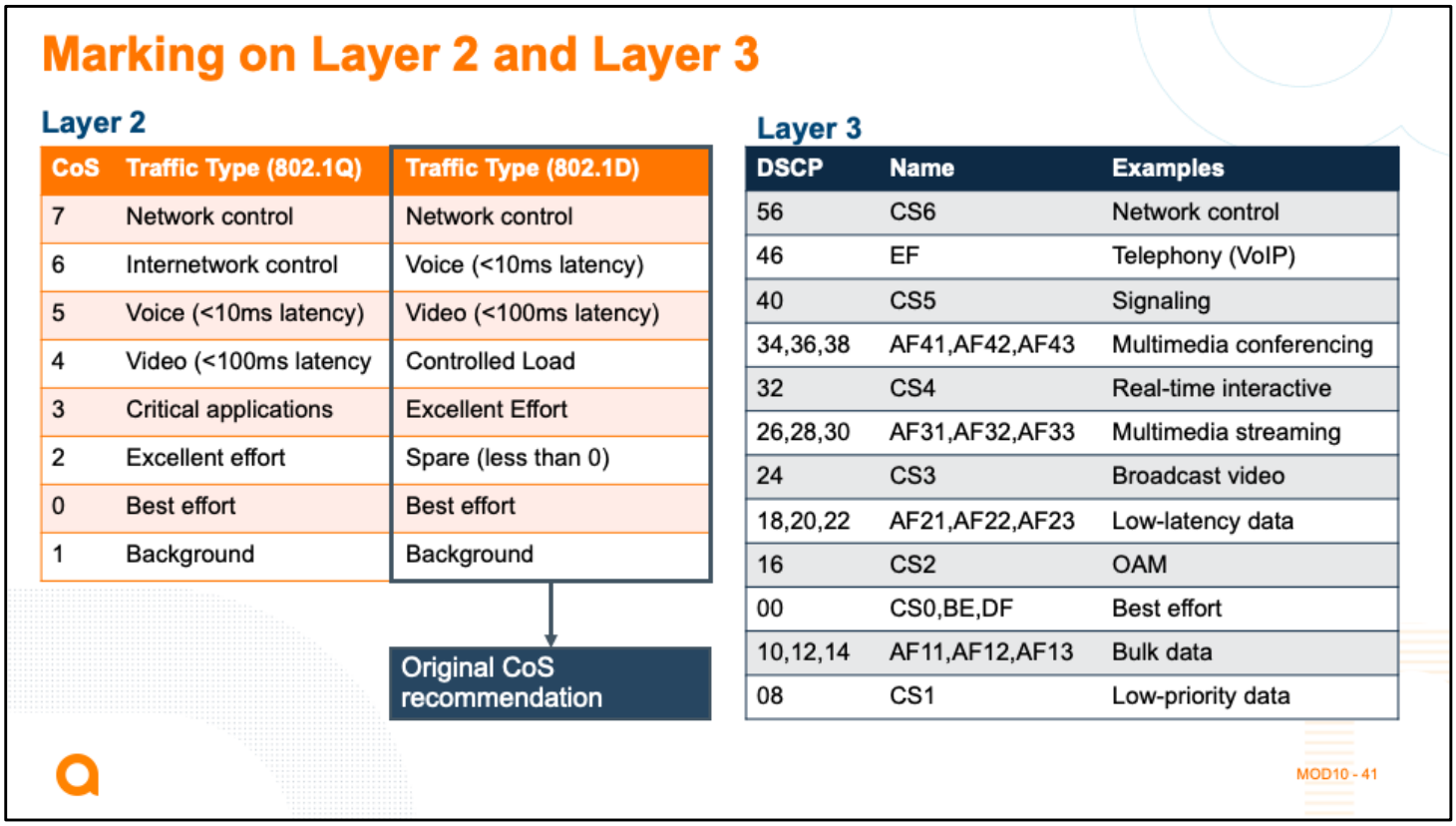
The Differentiated Services (DiffServ) is defined in RFC 2475. This model was developed based on a class scheme, in which traffic is classified into classes of service. With DiffServ each router effectively works in a standalone fashion locally differentiating between various types of traffic by applying a classification process. This standalone nature defines an inherent Per-Hop Behavior (PHB).

With PHB, there is no signaling between neighbors or end to end, and the QoS behavior is defined at each network device. This operation raises two obvious concerns. The first is how to achieve coherence of the behavior applied to traffic that crosses multiple routers, and the second is how to propagate information among routers.

Coherence is achieved by assuring that network devices act as a team – you must deploy consistent configurations across all participating devices. Thus, each device matches the same criteria and applies the same QoS behaviors. If just one device in the communication chain is not properly configured, then QoS is sub-par at best, or broken at worst. Consistent configuration is vital.

The second PHB concern relates to the lack of signaling between QoS peers. How is information shared? The QoS process can change packet or frame headers – they can “mark” packets. Thus, QoS does not need a special protocol to advertise to the next-hop how a packet should be treated.

Although it does pose some challenges, the Diffserv/PHB model has proved to be highly popular. In fact, it is so heavily deployed that it has become the de facto standard. The reasons for this are its flexibility, ease of implementation, and scalability.



Devices use standard QoS markings to communicate a packet's desired class of service. These values reside within frames or packet headers, and so travel with the packet, throughout its journey. There are two types of QoS marks: Class of Service (CoS) (sometimes called 802.1p) and Differentiated Services Code Points (DSCPs).

CoS uses a three-bit field in the 802.1Q VLAN tag to indicate priority. It defines eight values between 0 and 7. Note that because the VLAN tag carries the priority value, untagged traffic cannot contain a CoS value.

This standard uses higher CoS values for the most important traffic and lower values for traffic that is either tolerant to delay or is less important. It is recommended that the two highest values (7 and 6) are reserved for Control and Management Plane traffic. For example, AOS-CX automatically marks control traffic with the highest priority value (7). End user traffic uses the remaining 6 CoS values, where typically voice traffic uses CoS value 5, due to its sensitivity to delay and jitter. A frame that does not contain any CoS value (0) gets a Best Effort treatment which is default.

802.1D recommendations are based on the original standard that defines CoS. Some network infrastructure still use these recommendations, so it is important for you to be aware of them. In this scheme, network control traffic has just one value, 7 which shifts voice and video traffic up one priority level. Controlled load (rather than critical) has the next highest level and then excellent effort. One of the most crucial differences in the 802.1d standard is that it assigns

value 2 to "spare" traffic and is intended for traffic with a priority less than best effort. It is crucial that you understand whether other devices in the network consider value 2 as higher than 0 or less than 0.

RFC 2474 redefined the TOS field in the IP header as a 6-bit Differentiated Services Code Point (DSCP) field. The DSCP field boosts the possible distinct QOS markings from eight in the IP Precedence realm to sixty-four in the IP DSCP.

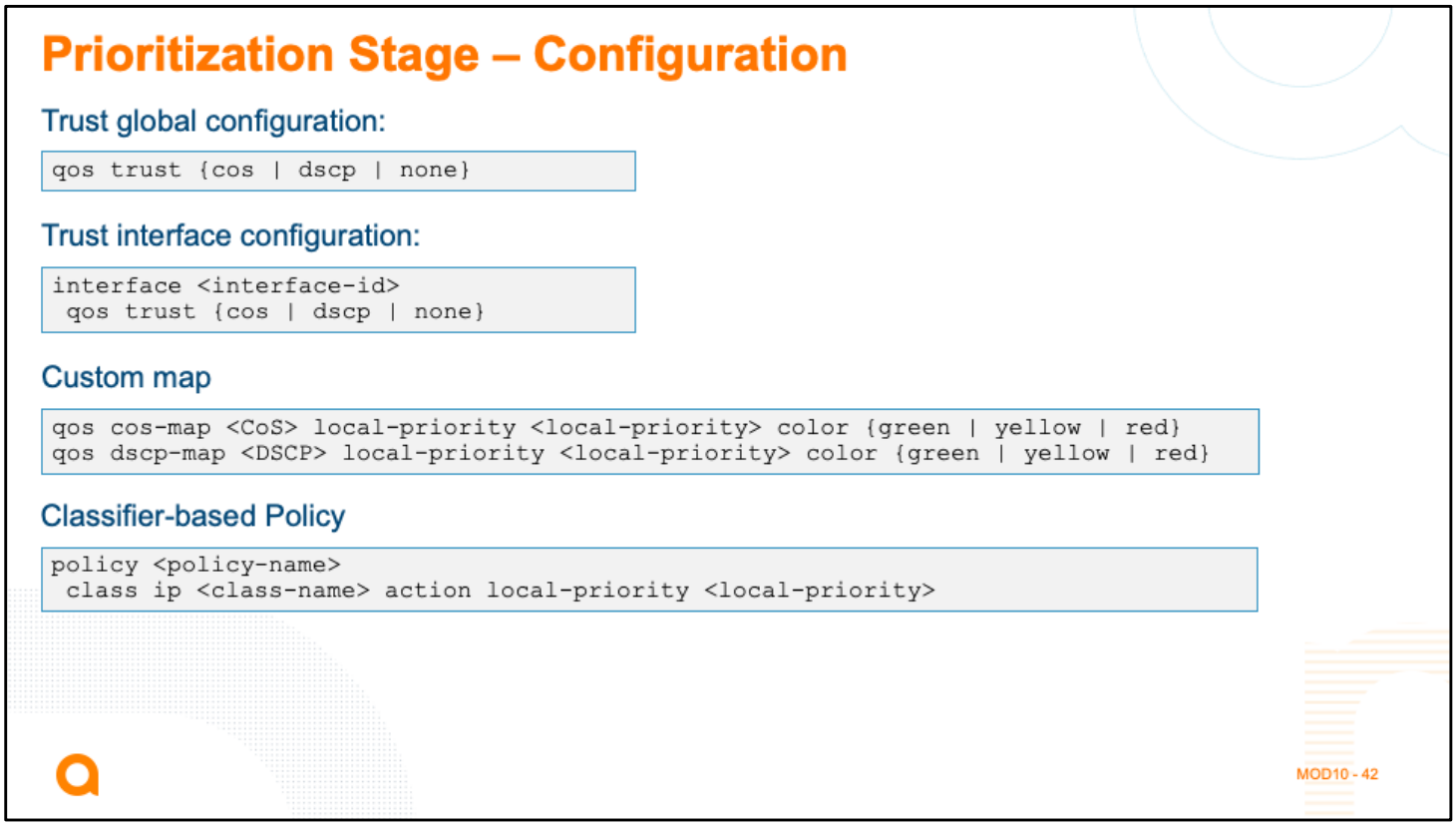
There are several standard definitions for DSCPs.

Expedited Forwarding (EF). Defined in RFC 3246 this DSCP value is intended to produce the lowest levels of jitter and delay.

Class Selector (CS). Defined in RFC 2472 these DSCP values have compatibility to the legacy IP Precedence values. The table suggests the types of applications that you could assign to each of these values.

Assured Forwarding. Defined in RFC 2597, establish four classes for queuing purposes, along with three levels of drop probability inside each queue. To mark packets and distinguish into which of four queues a packet should be placed, along with one of three drop priorities inside each queue, the AF PHB defines 12 DSCP values and their meanings. The names of the AF DSCPs conform to the following format, AFxy, where x implies one of four queues (values 1 through 4) and y implies one of three drop priorities (values 1 through 3). The AF PHB suggests that the higher the value of x in the DSCP name AFxy, the better the queuing treatment a packet should get.

Note. In IP precedence the first 3 bits in the ToS byte are used to define a precedence. The higher the value, the more important the IP packet is. During times of congestion the router would drop the low priority packets first. The ToS bits are used to assign the kind of delay, throughput, and reliability you want. In 1992, RFC 1349 was created to change the definition of the TOS byte therefore IP precedence is considered an old reference.



You can configure the prioritization stage with the following commands:

Global configuration:

configure terminal
qos trust {cos | dscp | none}

Interface configuration:

configure terminal
interface <interface-id>
qos trust {cos | dscp | none}

Custom map has two options:

1. CoS

configure terminal
qos cos-map <CODE-POINT> local-priority <PRIORITY-NUMBER> [color <COLOR>] [name <DESCRIPTION>]

Defines the local priority assigned to incoming packets for a specific 802.1 VLAN priority code point (CoS) value. The CoS map values are used to mark incoming packets when QoS trust mode is set to cos.

2. DSCP

configure terminal

qos dscp-map <CODE-POINT> local-priority <PRIORITY-NUMBER> [color <COLOR>] [cos COS-CODE-POINT] [name <DESCRIPTION>]

Defines the local priority assigned to incoming packets for a specific IP differentiated services code point (DSCP) value. The DSCP map values are used to prioritize incoming packets when QoS trust mode is set to dscp.

Classifier-based Policy:

This configuration is a three-step process:

Select the interesting traffic using a class

configure terminal

class ip <class-name>

{match | ignore} <protocol> <source> <destination>

Configure the Local-Priority

configure terminal

policy <policy-name>

class ip <class-name> action local-priority <local-priority>

Apply the configuration to a port

configure terminal

interface <interface-id>

apply policy <policy-name> in


Lab Activity
Lab 10

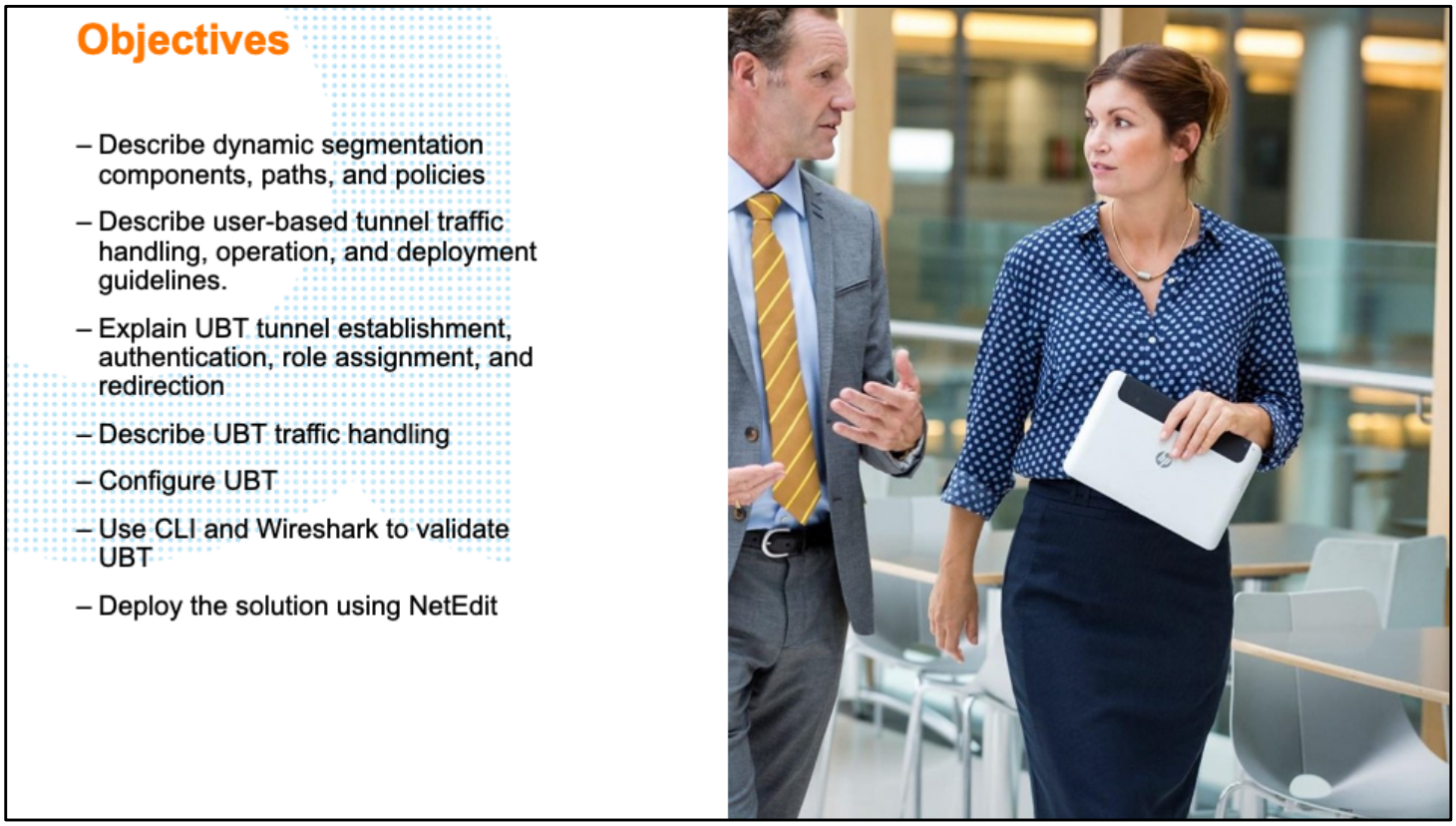
Rev #20.41

Dynamic Segmentation

Aruba Advanced Switching Troubleshooting and Solutions (ASTS)



aruba
a Hewlett Packard Enterprise company

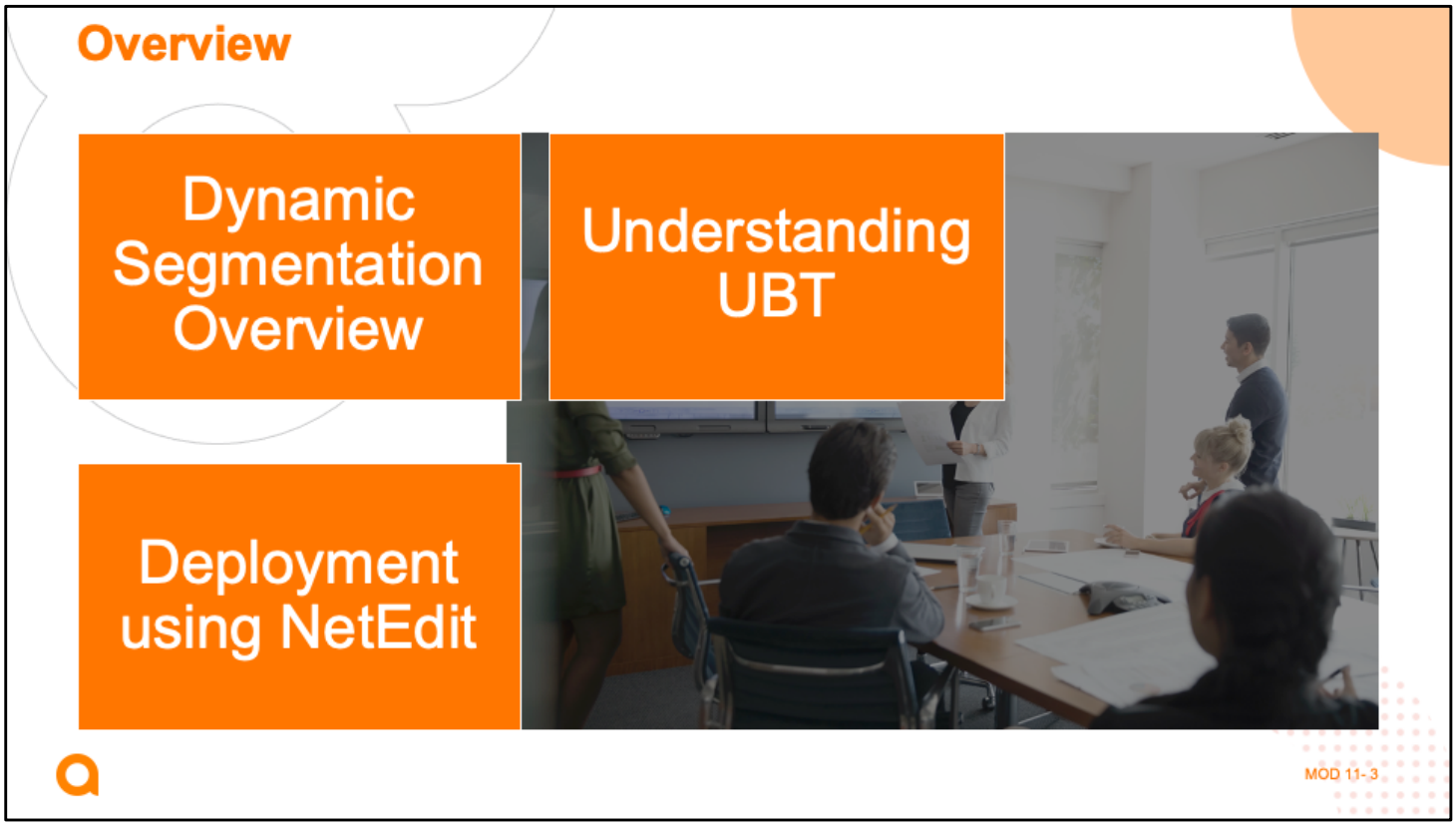


After completing this course, you will be able to describe dynamic segmentation components, paths, and policies.

You will be able to describe User Based Tunneling (UBT) traffic handling, operation, and deployment guidelines, and to explain UBT tunnel establishment, authentication, role assignment, and redirection

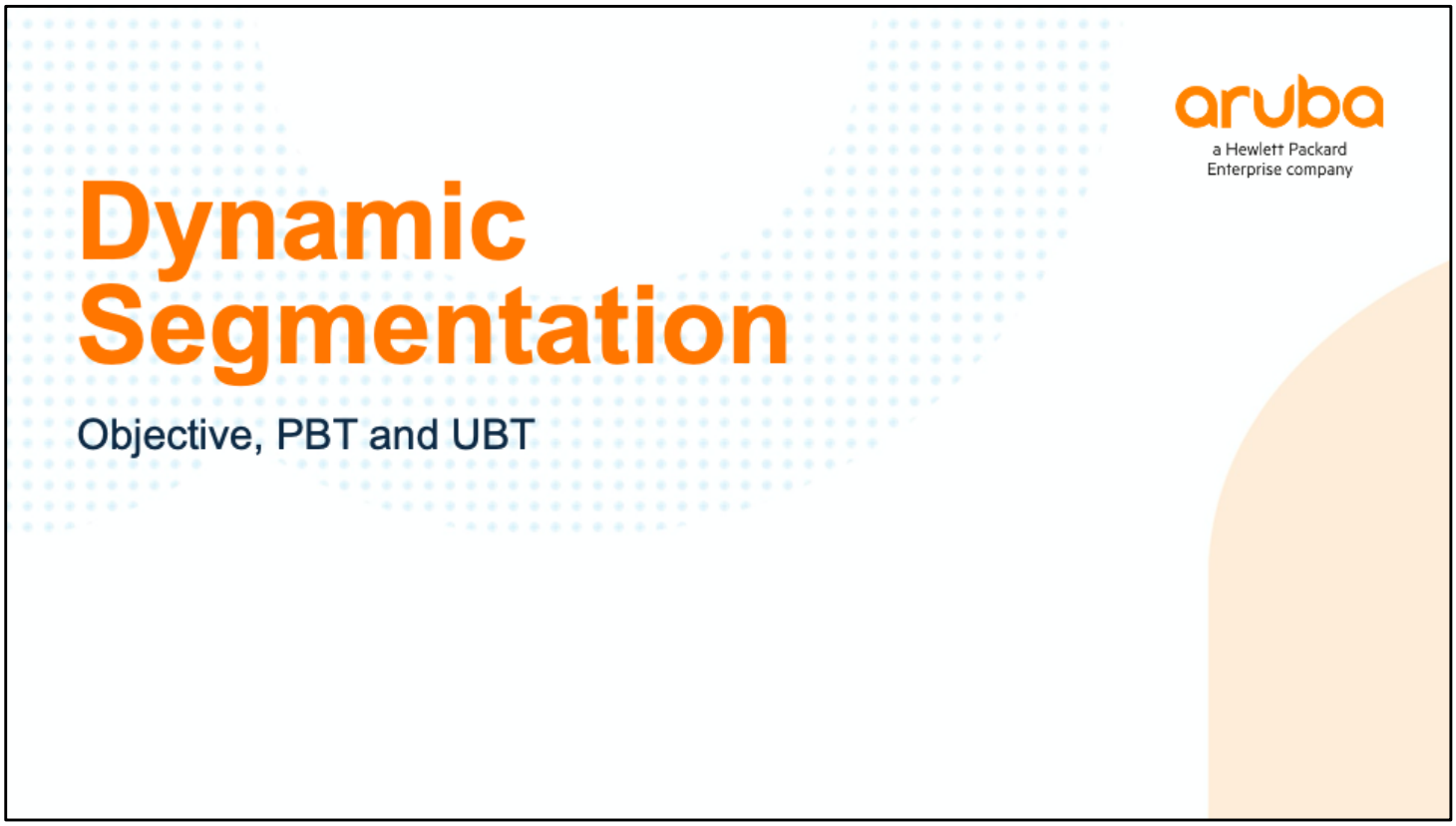
You will also be able to describe UBT traffic handling, and to configure UBT.

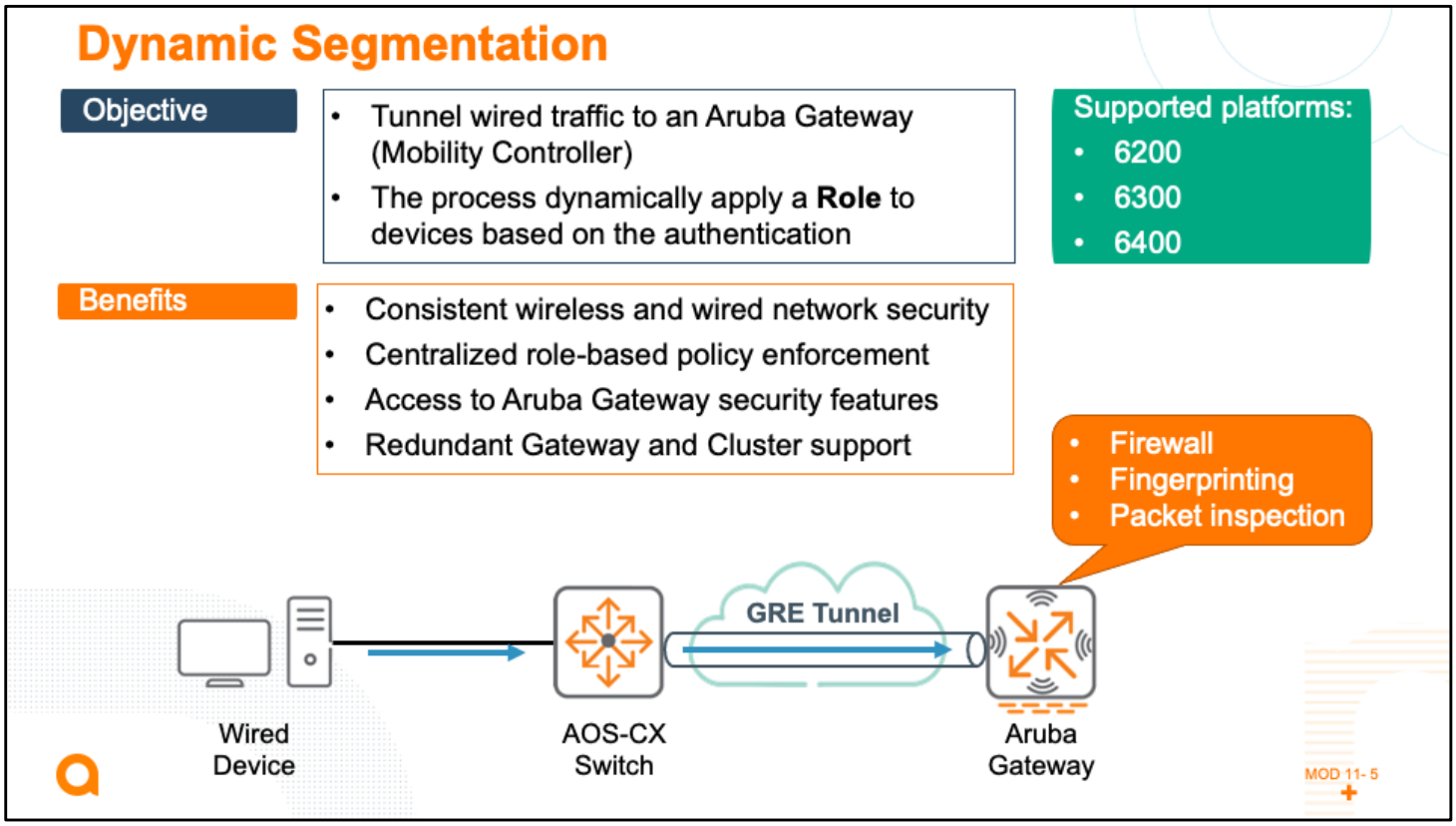
You will be able to use the CLI and Wireshark to validate and troubleshoot UBT, and to deploy the solution using NetEdit.



Dynamic segmentation is a powerful deployment technique – potentially saving huge amounts of staff overhead related to switch configuration, while centralizing and improving security and policy conformance, and providing for a consistent experience among wired and wireless users.

You begin this module with an overview of dynamic segmentation. You explore components, paths, and policies, before diving into various User Based Tunneling (UBT) operational characteristics – tunnel establishment, authentication, role assignment, and redirection. Then you move on to configuring and troubleshooting Dynamic Segmentation and UBT, using CLI tools, Wireshark, and NetEdit.


aruba
a Hewlett Packard
Enterprise company
Dynamic
Segmentation
Objective, PBT and UBT



The Aruba Dynamic Segmentation solution enables you to tunnel traffic on a user or device basis, sending the traffic to an Aruba Gateway (previously known as Mobility Controller). Technically Dynamic Segmentation is a process of dynamically applying policies (roles) to a switch port based on a user's credentials (and possibly fingerprinting), where one of those policies is local switching or tunneling of the user's traffic.

The benefits of Dynamic Segmentation are:

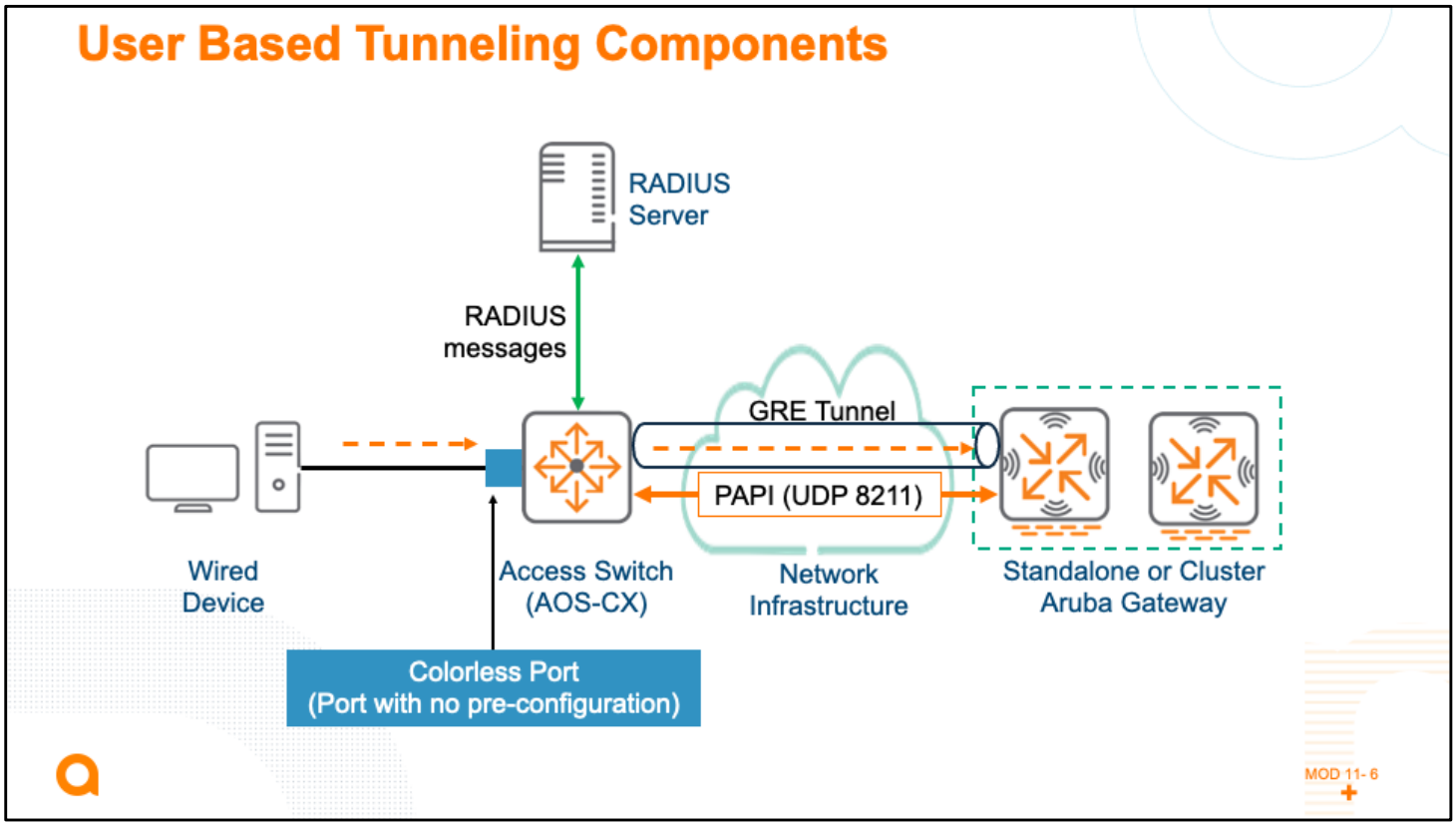
Consistency between wired and wireless network security. The Aruba Gateway enforces firewall policies on received traffic - either coming from an Access Point or a switch.

Centralized role-based policy enforcement. All firewall policies are setup in the Aruba Gateway. This dramatically simplifies network management by eliminating complex and static network configuration on each individual switch.

Access to Aruba Gateway security features. All traffic can be sent to the Gateway for consistent firewalling, fingerprinting, and packet inspection.

Redundant Gateway and Mobility Cluster support. Tunneling traffic to a unique Aruba Gateway creates a single point of failure. The Dynamic Segmentation solution is supported with redundant controllers (in standalone deployment) or with Multiple Gateways under a cluster. This cluster solution also helps to increase network performance.

Dynamic Segmentation is supported in the following switch platforms: 6200, 6300, 6400.



The User Based Tunneling solution is composed of five different devices:

Wired Devices. These are the endpoints that connect to the network.

Access Switch. The access switch acts as the 802.1X authenticator device, this means that the switch is responsible to send the RADIUS Access Request message to the RADIUS server when the wired device connects to the network. If the authentication is successful, the access switch assigns a User-Role to the Wired Device. The access switch builds a GRE tunnel to the Aruba Gateway.

Network infrastructure. The infrastructure must of course route this traffic between switches, Gateways, and RADIUS server. Any access lists or firewalls must permit the end user traffic, 802.1X/RADIUS communications, and GRE/PAPI tunnel traffic.

Aruba Gateway. This network device has many built-in security and application capabilities tailored to wireless traffic but extended to wired traffic. Tunnel traffic takes advantage of the Aruba Gateway's firewall, fingerprinting and packet inspection capabilities. The implementation supports Standalone and Cluster.

RADIUS server. This device is used to authenticate the credentials of the User or Device.

Note. The User-Role in the Access switch (which runs AOS-CX) is a collection of internal policies and port attributes assigned to the device and the port where it is connected.

Note. When Dynamic Segmentation is used with an AOS-CX switch, the RADIUS server will

need the Aruba Vendor Specific Attribute (VSA) Dictionary installed on it.

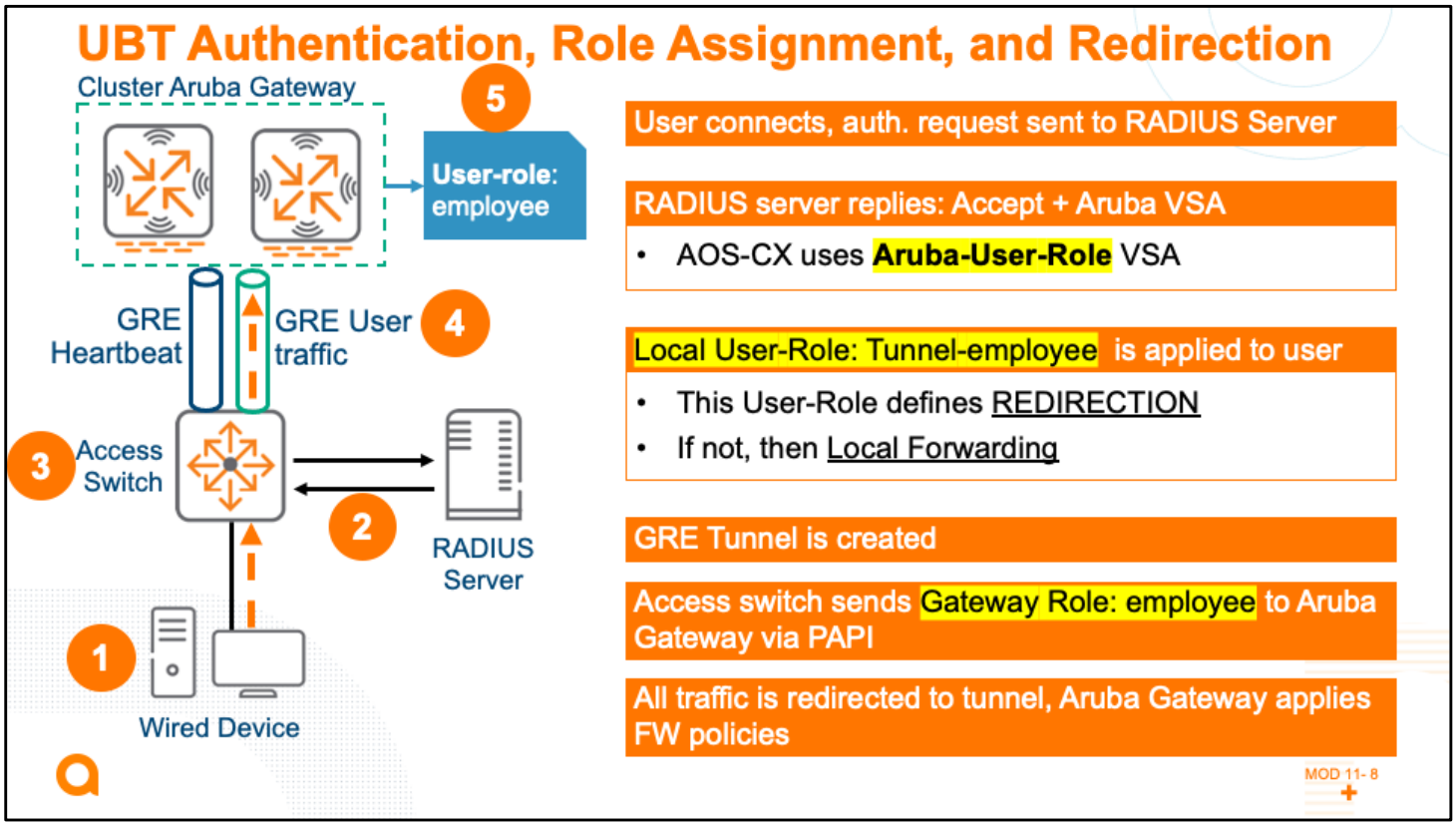
Note. The port where the wired device is connected does not need any previous configuration on it. This type of port is known as colorless port.

Note. Aruba ClearPass as the RADIUS server adds extra functionalities to the Dynamic Segmentation solution such as Downloadable User Role and Profiling.

Understanding User Based Tunneling

General Functionality, Multicast and Broadcast traffic, Configuration

aruba
a Hewlett Packard Enterprise company



As a user connects to a secure port, the authentication subsystem on the AOS-CX switch sends a RADIUS request to a RADIUS server. Often, the Aruba ClearPass Policy Manager provides this RADIUS service, which authenticates the user and returns a User Role contained in an Aruba Vendor Specific Attribute (VSA) to the switch.

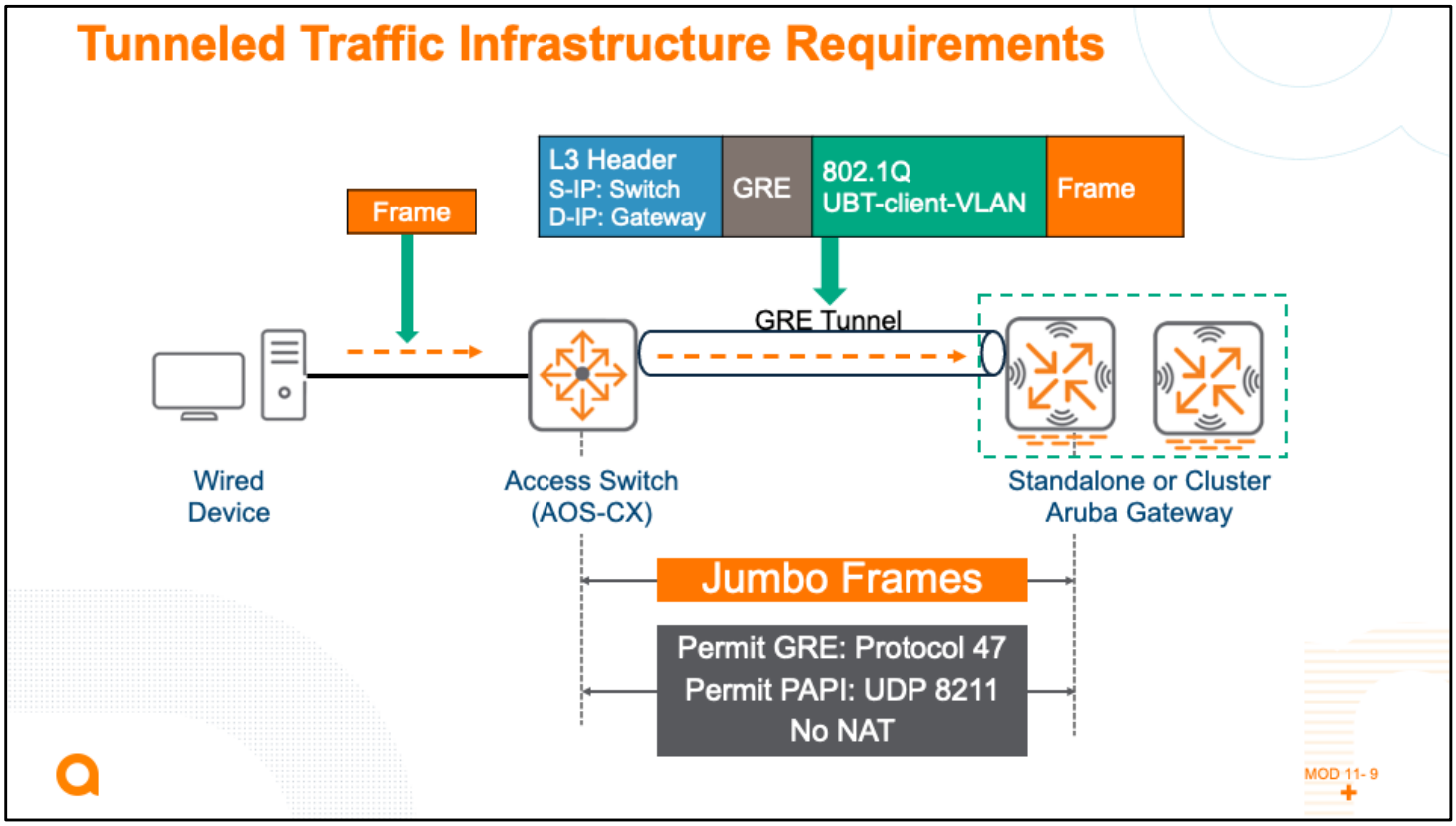
Once the switch receives the VSA, it applies the local User Role to the user. If the role assigned to the device is explicitly configured to redirect traffic to an Aruba Gateway, the authentication subsystem tells the switch's tunneling subsystem to create a GRE tunnel. In this example the Local User Role has been named Tunnel-employee.

The switch then sends the Gateway role to the Aruba Gateway via PAPI where firewall and security policies are applied to wired users. This Gateway role can also be the same role used for wireless users. In this example the Gateway user-role is employee.

When the tunnel is formed to a controller cluster, the AOS-CX switch uses the bucket map to determine the Active and Standby User Anchor Controller. To perform this election, the switch applies a hash algorithm on the client's MAC address. The result is a numerical value, used to lookup into the bucket map to determine the UAC.

Once user tunnels are established to the UAC, a PAPI-based keepalive packet is exchanged with the Aruba Gateways that have users anchored to them, maintaining tunnel integrity.

Note. If the user-role does not contain an attribute to redirect traffic to an Aruba Gateway, the user or device is placed into the specified VLAN on the switch and traffic is locally forwarded. VoIP and other sensitive traffic can benefit from this behavior, since no extra delay is added.



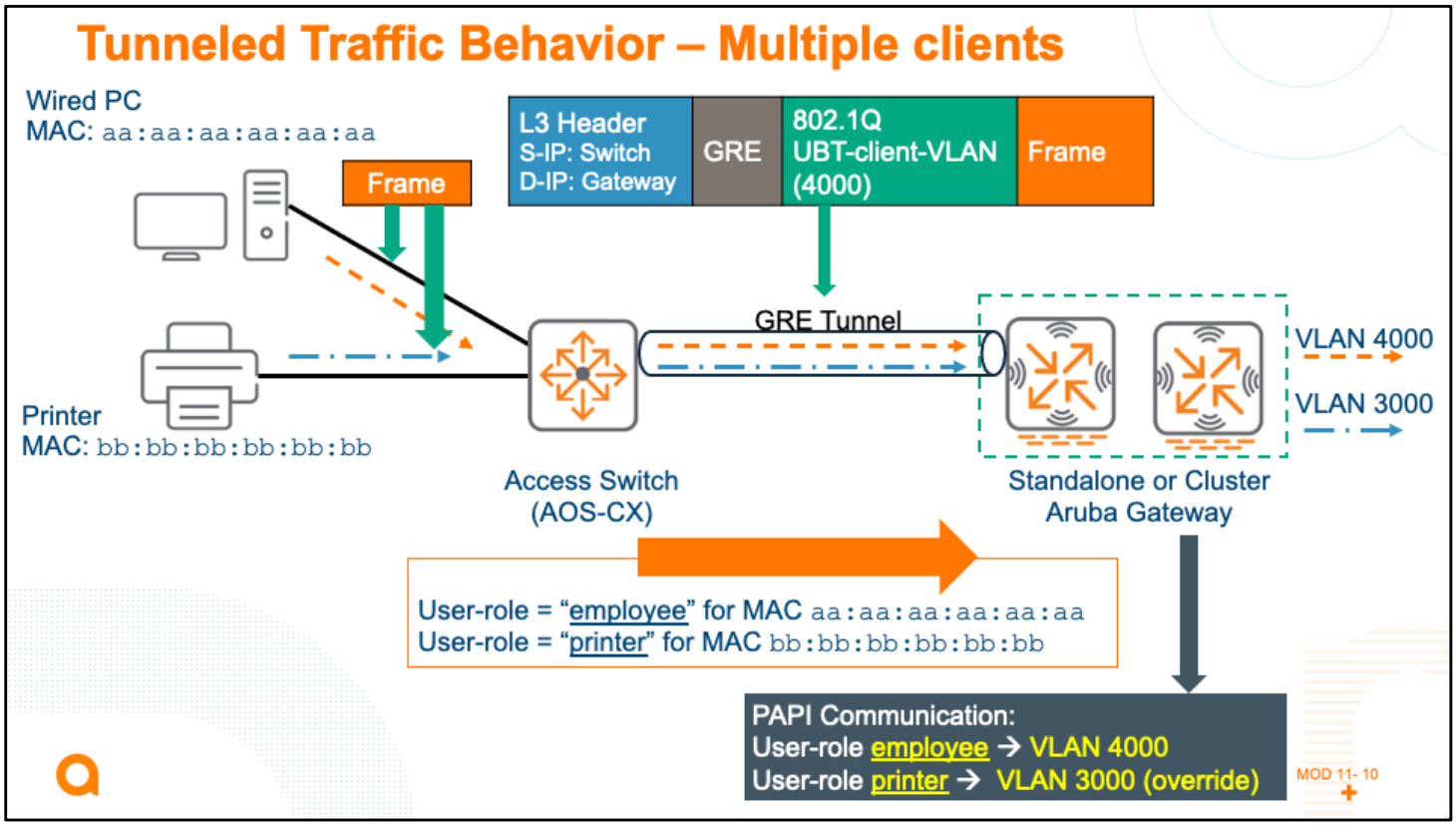
You should check the network infrastructure between the AOS-CX switches that implement Dynamic Segmentation and the Aruba Gateways.

Any firewalls or network infrastructure devices with ACLs must allow GRE (protocol 47) and PAPI traffic on UDP port 8211. Also, NAT must not be applied between the switch and the controller.

Also know that GRE-encapsulated frames can exceed the standard MTU. Aruba recommends that you enable jumbo frames across the path from the switch to the Aruba Gateway. On AOS-CX switches, you enable jumbo frames on a port basis. Make sure that every port that carries traffic between the switch and the Aruba Gateway supports jumbo frames.

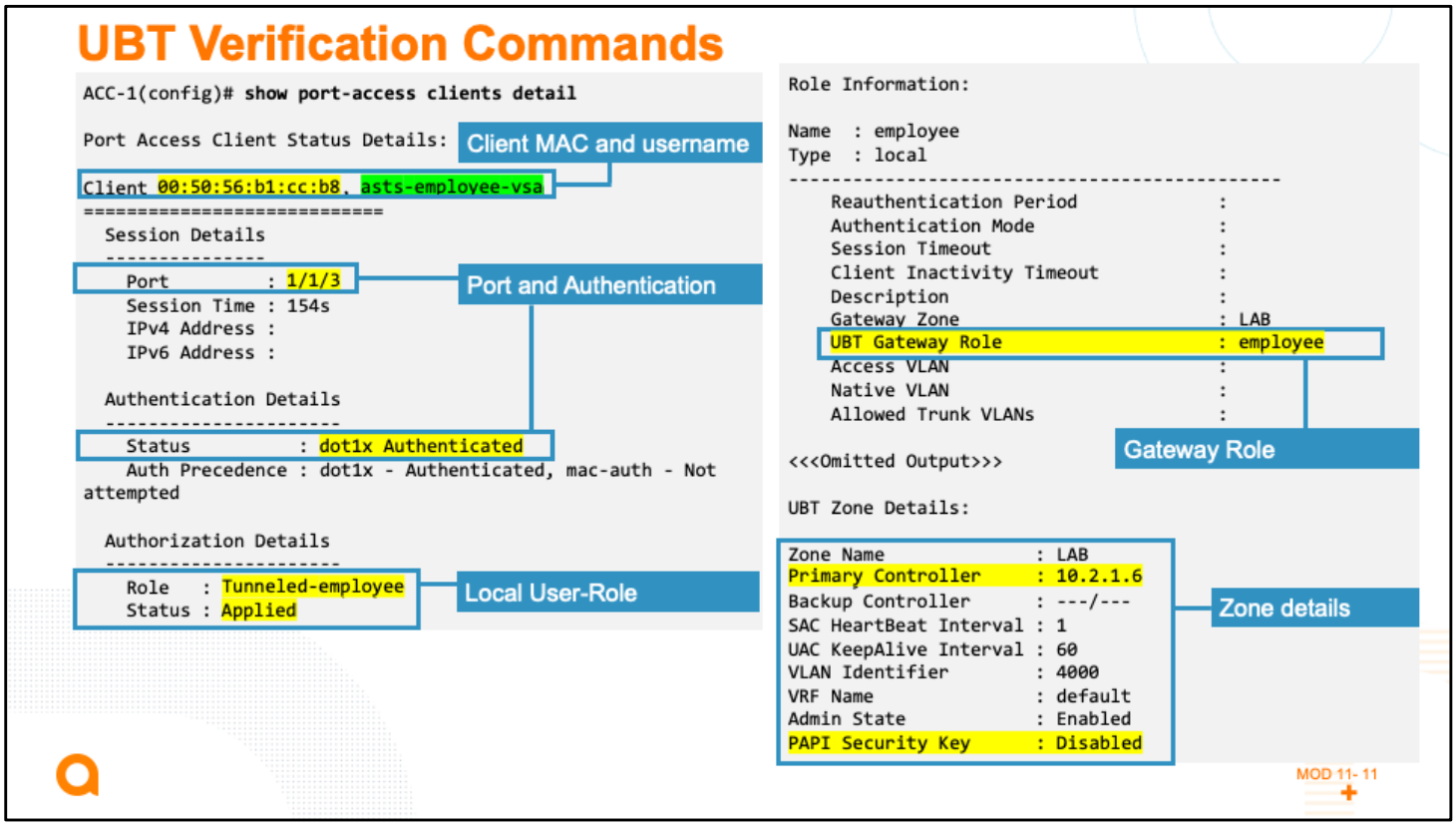
AOS-CX allows you to define the source IP and the VRF to use in UBT.

The slide shows also the content of the Headers, noticed that the switch encapsulates the original frame inside the GRE tunnel, the switch can add an 802.1Q header with the VLAN-ID defined in the ubt-client-vlan command. This indicates to the Aruba Gateway which VLAN-ID is used to map the client's traffic.



When you configure Dynamic Segmentation, AOS-CX switches allow you to set a single UBT-client VLAN. The Aruba Gateway places all wired client traffic on this VLAN. In real scenarios this is not the expected behavior. To solve this problem, the AOS-CX switch sends to the Aruba Gateway the user-role that the client has been assigned, the Aruba Gateway can then use this information to assign a new VLAN, which overrides the 802.1Q Tag included in the inner frame header.

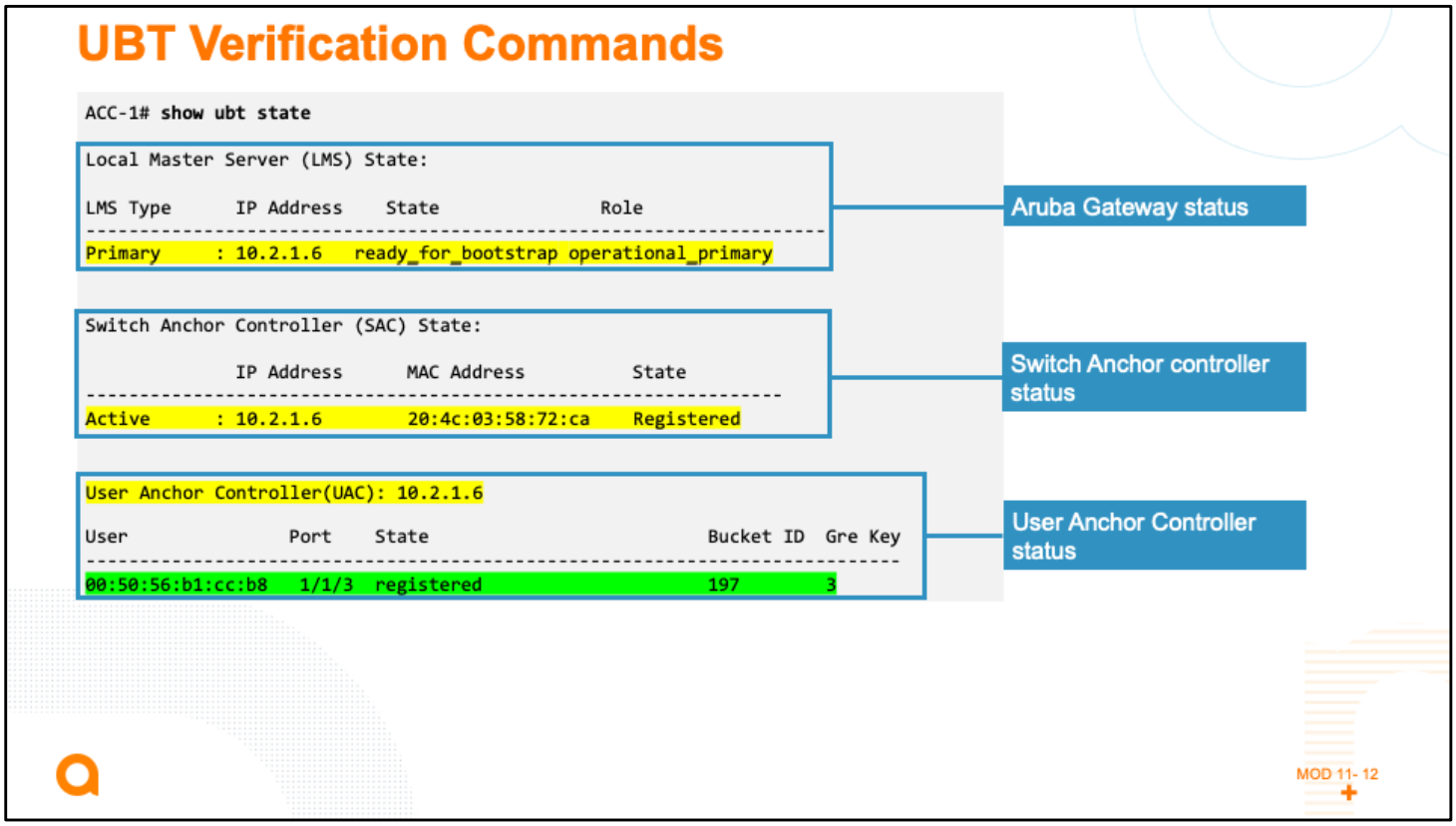
As an example, the administrator wants to place the traffic from Wired PC which has a MAC aa:aa:aa:aa:aa:aa on VLAN 4000 and the traffic from the Printer (MAC bb:bb:bb:bb:bb:bb) on VLAN 3000. The switch sends to the Aruba Gateway the correspondent user role for each MAC. The Aruba Gateway then must be configured with the proper VLAN for each user role, in the case of the "printer" role VLAN 3000 and for employee VLAN 4000.



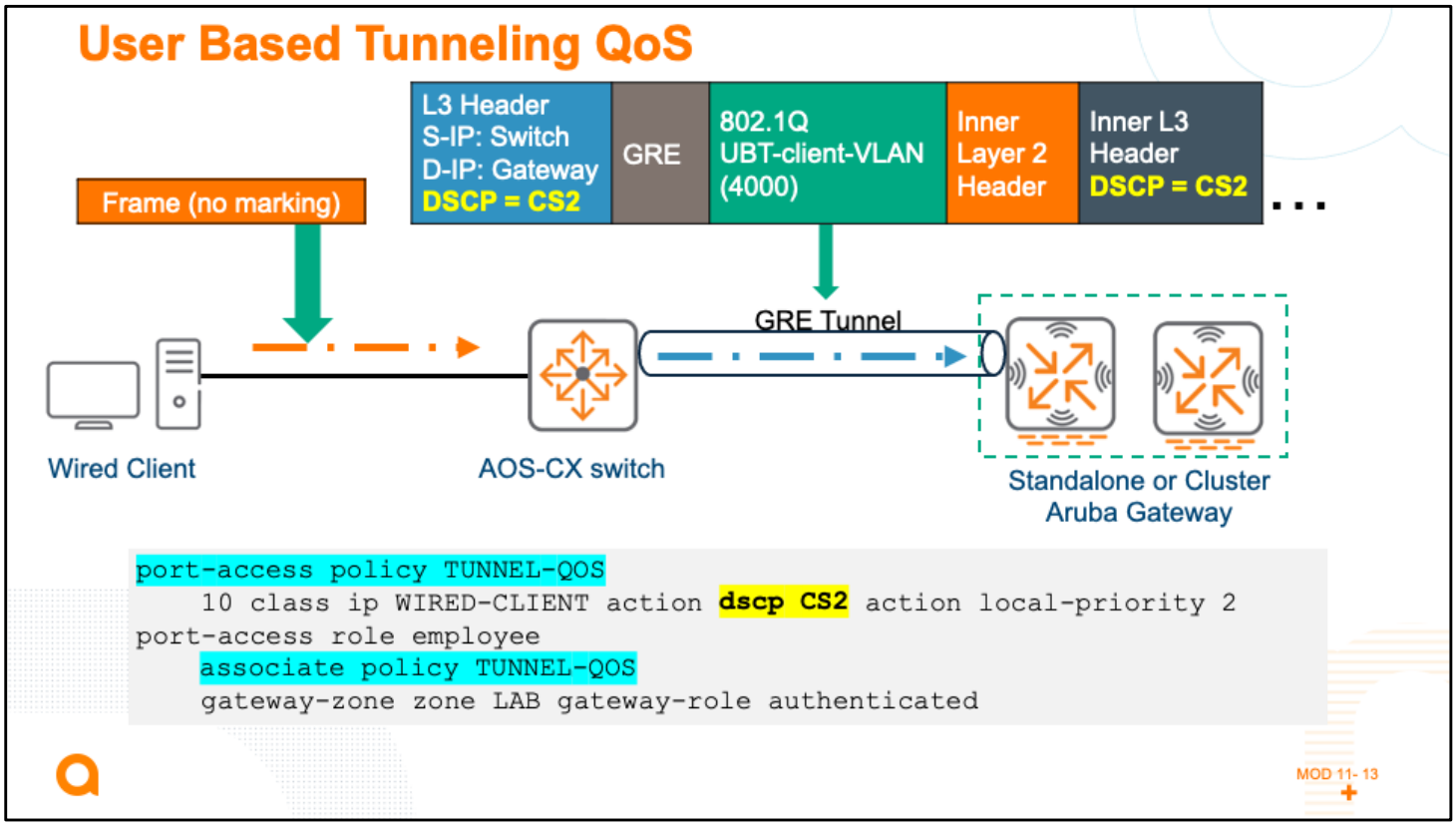
The show port-access clients detail command shows information about key elements related to UBT.

The output shows how the client with MAC address 00:50:56:b1:cc:b8 connected to port 1/1/3 performs 802.1X authentication using the username asts-employee-vsa. The authentication is successful, and the VSA Aruba-User-Role value received (Tunneled-employee) from the RADIUS server is applied to the client.

The output also shows the Gateway role used defined in AOS-CX as UBT Gateway Role. The value used in this example is “employee”. Finally, the last section is referring to the UBT zone where you can see the Aruba Gateway communication configuration. The output shows that the IP address of the Aruba Gateway is 10.2.16 and we are not using PAPI Security Key.



The show ubt state command displays the state with the Aruba Gateway, Switch Anchor Controller, and User Anchor Controller.



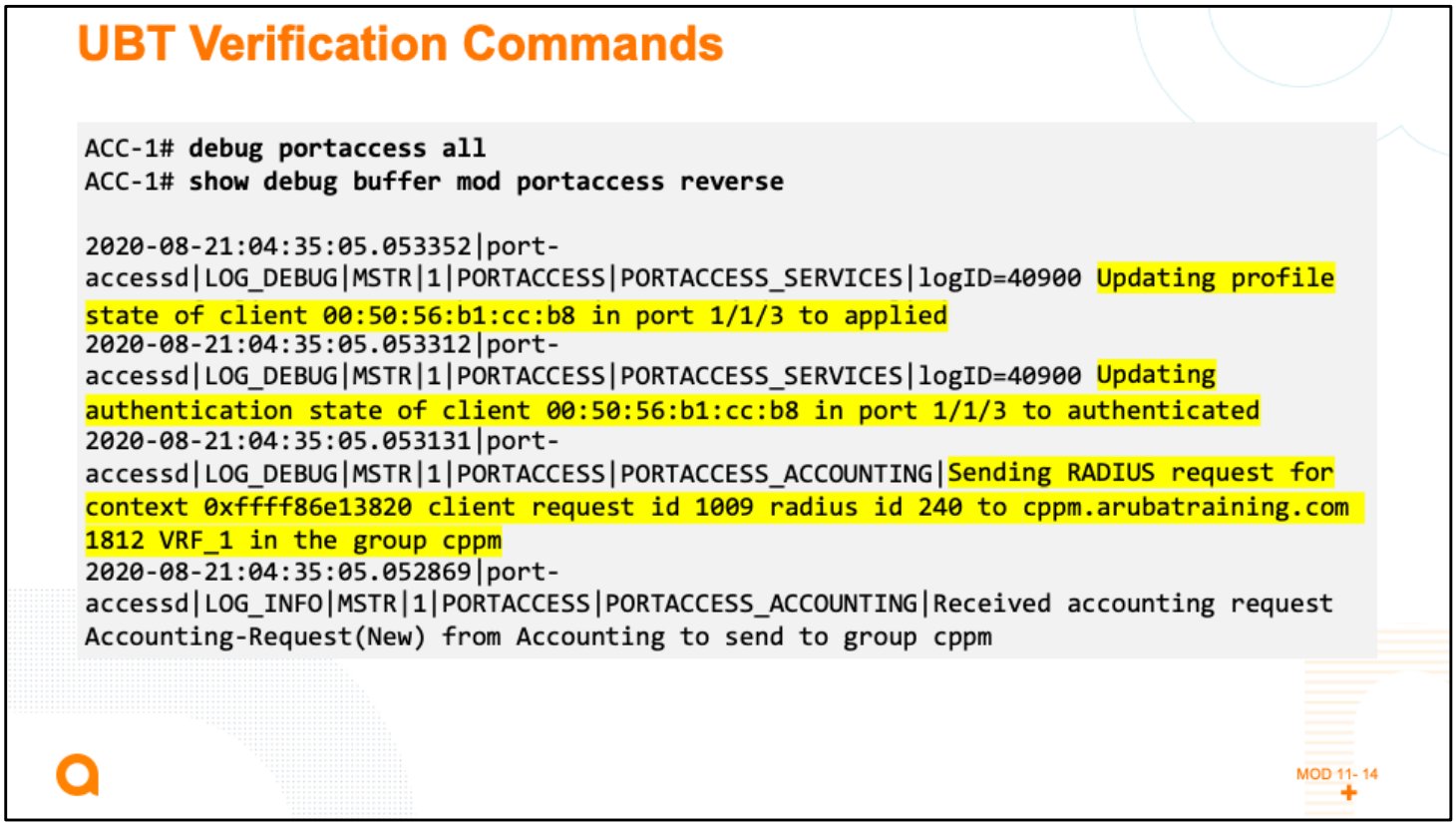
Often, you want to apply QoS marking to device traffic destined for an Aruba Gateway.

The AOS-CX switch can still perform local inspection and control of the traffic that is sent over the tunnel. This allows the administrator to apply QOS policies on the user traffic. The switch will automatically push the inner DSCP markings to the outer GRE tunnel IP DSCP field. This ensures that the transport network between the switch and the Aruba Gateway can also respect the QOS of the user traffic, even when the traffic is inside a GRE tunnel.

In the diagram you can see that a DSCP marking (DSCP = CS2) has been configured in the AOS-CX switch, this marking affects the inner and outer headers.

To accomplish this task in the AOS-CX switch you must associate a port-access policy to the role.

```
class ip WIRED-TRAFFIC
    10 match any any any count
port-access policy TUNNEL-QOS
    10 class ip WIRED-CLIENT action dscp CS2 action local-priority 2
port-access role employee
    associate policy TUNNEL-QOS
    gateway-zone zone LAB gateway-role authenticated
```



The output displays the output for the debug portaccess all command.

In the example, the debug command gives you specific information about the dynamic segmentation process.

It gives you information about the RADIUS request sending to cppm.arubatraining.com RADIUS server.

Once the switch receives the RADIUS access accept, you can see how the switch updates the state of the client to authenticated.

Finally updating the profile to the client indicates the correct application of the User-Role.

Question #1

```
#show debug buffer severity info
#show debug buffer
2020-10-15:14:53:46.673681|port-
accessd|LOG_INFO|MSTR|1|PORTACCESS|PORTACCESS_SERVICES|logID=5701 Secure client state
changed for client 00:50:56:b1:95:30 in port 1/1/4 to Authenticated but
failed to apply role.
```

According to the output, what could be the problem?

a. The RADIUS server is configured with an AVP that is not supported by the AOS-CX switch

b. The RADIUS server is using a not valid Aruba VSA

c. The RADIUS server indicates a role that does not exist in the switch

d. The AOS-CX switch does not support the Aruba VSA sent by the RADIUS server

Knowledge Check

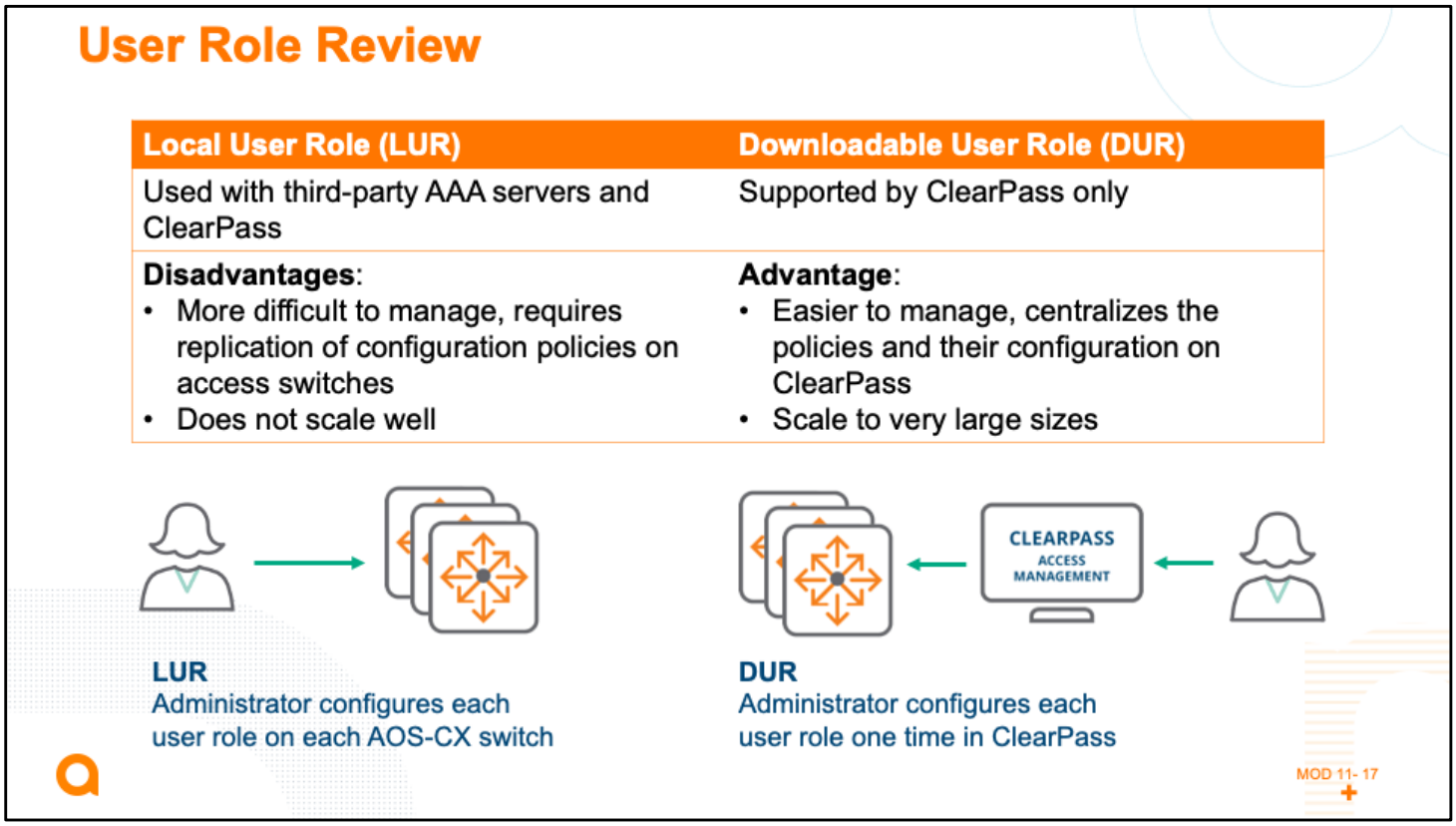
Correct answer:

C

aruba
a Hewlett Packard Enterprise company

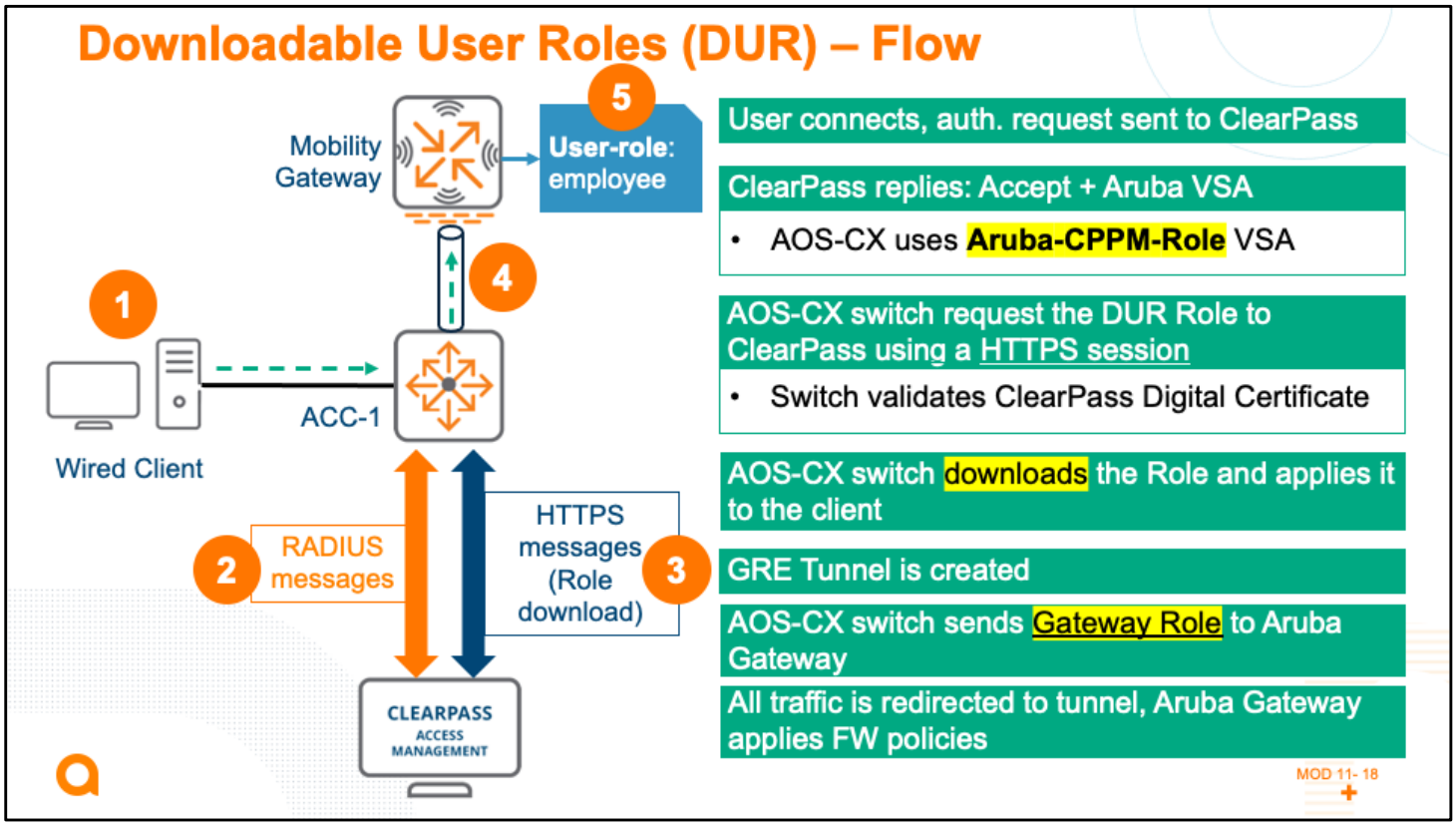
Downloadable User Roles

Configuration and verification



A Local User Role (LUR) is where the role is defined locally on the switch, and DUR is where the role is defined on the ClearPass server and downloaded to the switch (the role does not pre-exist on the switch).

With a Downloadable User Role (DUR), instead of having the user role configured on the edge switch, the user role is configured on ClearPass and downloaded to the edge switch after successful authentication. The method for sending the user role is through REST API using an SSL connection. This means that a HTTPS certificate must be installed on the edge switch. In addition, a username/password must be configured to authenticate the switch with ClearPass.



When Downloadable User Role is implemented, the process will have a new flow described below:

The device is connected to a switch port, this port has been configured with MAC or 802.1X authentication.

The authentication process starts and the AOS-CX switch sends a RADIUS Authentication Request to the ClearPass server.

ClearPass validates the device credentials and sends a RADIUS Access Accept message containing the VSA Aruba-CPPM-Role

The AOS-CX switch receives the RADIUS Accept message and initiates the User Role downloading process by setting up a HTTPS session.

– The AOS-CX switch validates the digital certificate received by ClearPass.

– The AOS-CX switch receives the User Role.

The switch applies the User Role to the device.

The GRE tunnel is created.

The AOS-CX switch sends “Gateway” role to Aruba Gateway.

The traffic is redirected to the Aruba Gateway.

Important: UBT/local switching and user roles are two different, but related concepts. They

are both used to implement dynamic segmentation. With Aruba AOS-CX switches, some devices could be locally switched with a pre-define local role, perhaps for VoIP phones; but for other devices, their role information could be downloaded from ClearPass and the traffic tunneled to an Aruba MC. It's flexible in how you want to set it up. However, Aruba recommends that if you are using ClearPass and have either AOS-CX or the older AOS switches, that you implement DUR because of the scalability and ease of centralized policy management that it provides.

Question #2

```
# show debug buffer severity err
2020-10-15:19:11:24.032831|port-
accessd|LOG_ERR|MSTR|1|PORTACCESS|PORTACCESS_DOT1X_PROTOCOL|Certificate subject name
doesn't match the expected peer's hostname.
2020-10-15:19:11:24.033786|port-
accessd|LOG_ERR|MSTR|1|PORTACCESS|PORTACCESS_ROLE|logID=25297 Error SSL peer
certificate or SSH remote key was not OK while downloading role
asts_aruba_dur_employee-3118-6
```

According to the output, the SSL certificate is not valid, what the administrator should do in order to solve this problem?

a. Generate a CSR for the AOS-CX switch

b. Install the CA certificate that the ClearPass server is using as a trusted CA in the AOS-CX switch

c. Validate the DNS server can resolve the Subject Name of the ClearPass server certificate

Knowledge Check

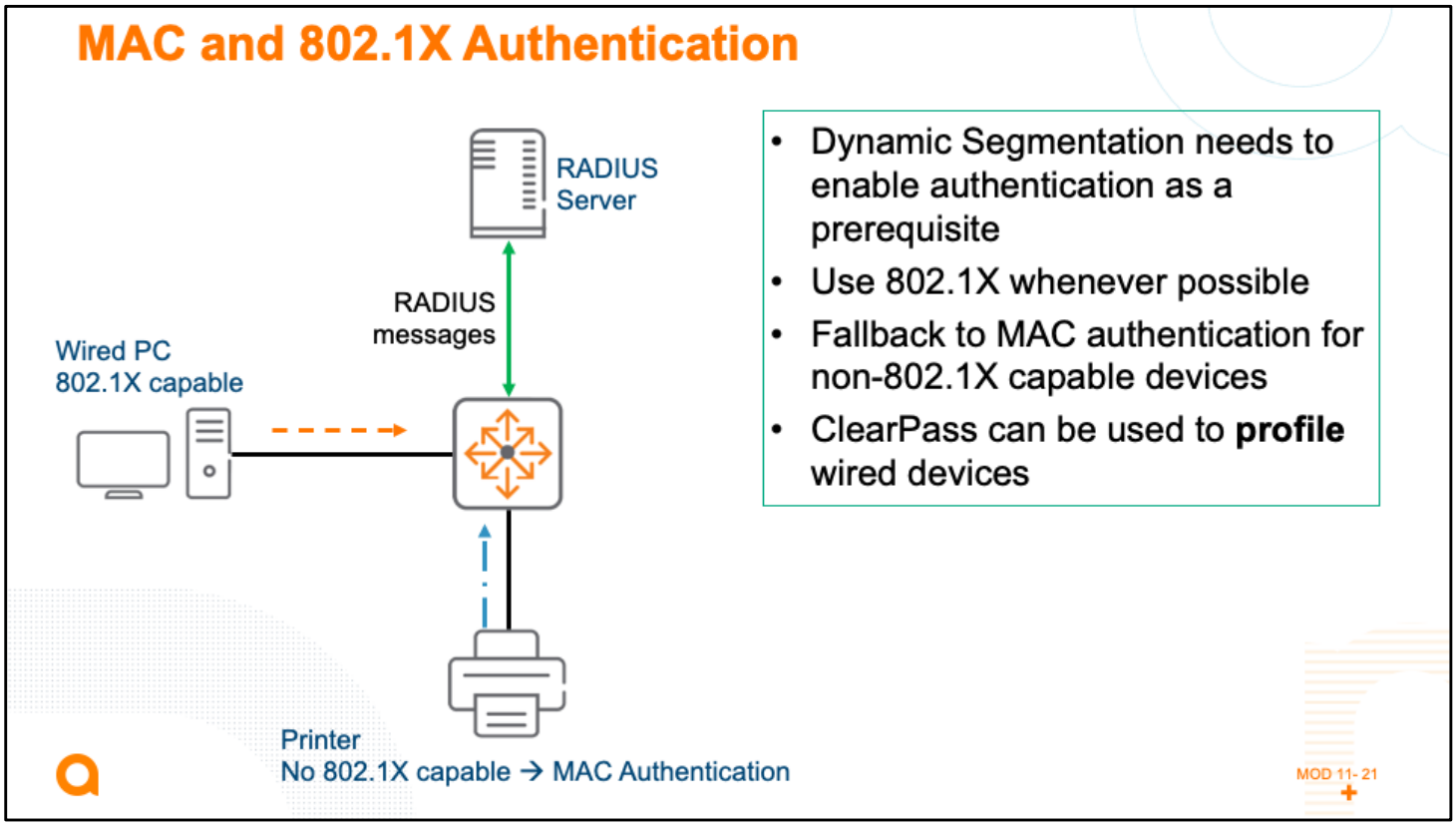
Correct answer:

C

aruba
a Hewlett Packard Enterprise company

Additional Content

Module 11



Dynamic segmentation needs as a prerequisite to enable either MAC or 802.1X authentication in order to function. AOS-CX supports to enable both authentication on the same port, the first authentication type to be used is 802.1X but if the device connected does not support 802.1X then fallback to MAC authentication is possible.

MAC Authentication

MAC authentication is a rudimentary form of authentication. The network access device (NAD) reads the client MAC address from the physical hardware NIC and then stores it in the software registry in Windows or other OS, making it susceptible to MAC spoofing. Additionally, the MAC address is not encrypted so it is easy to capture it and spoof to gain access to the network.

The main characteristics about this authentication are:

Simple to deploy

Client does not need any special installation or configuration

Easy to be spoofed

Combined with Aruba ClearPass MAC authentication becomes a more reliable authentication

type. ClearPass can profile the device that is connected, in case some attacker spoofs the MAC address of a valid device and connects a new one. ClearPass can detect this change by profiling the new device connected. This action triggers a conflict flag in ClearPass which when it is set it can instruct the switch to disconnect the client or apply any other action the administrator configures.

802.1X Authentication

The Extensible Authentication Protocol (EAP) is a common way of authenticating users to the network using a RADIUS server such as Active Directory or ClearPass. There are many flavors of EAP, including PEAP, EAP-TLS, EAP-TTLS and others. EAP uses 802.1X to transport the user credentials between the user device and the Network Access Device – NAD (Switch). 802.1X happens at Layer 2 where users will be authenticated before an IP address is assigned. This is independent of the Layer 3 protocol. Most NAD solutions will then forward the EAP information onward to a centralized AAA server using the RADIUS protocol.

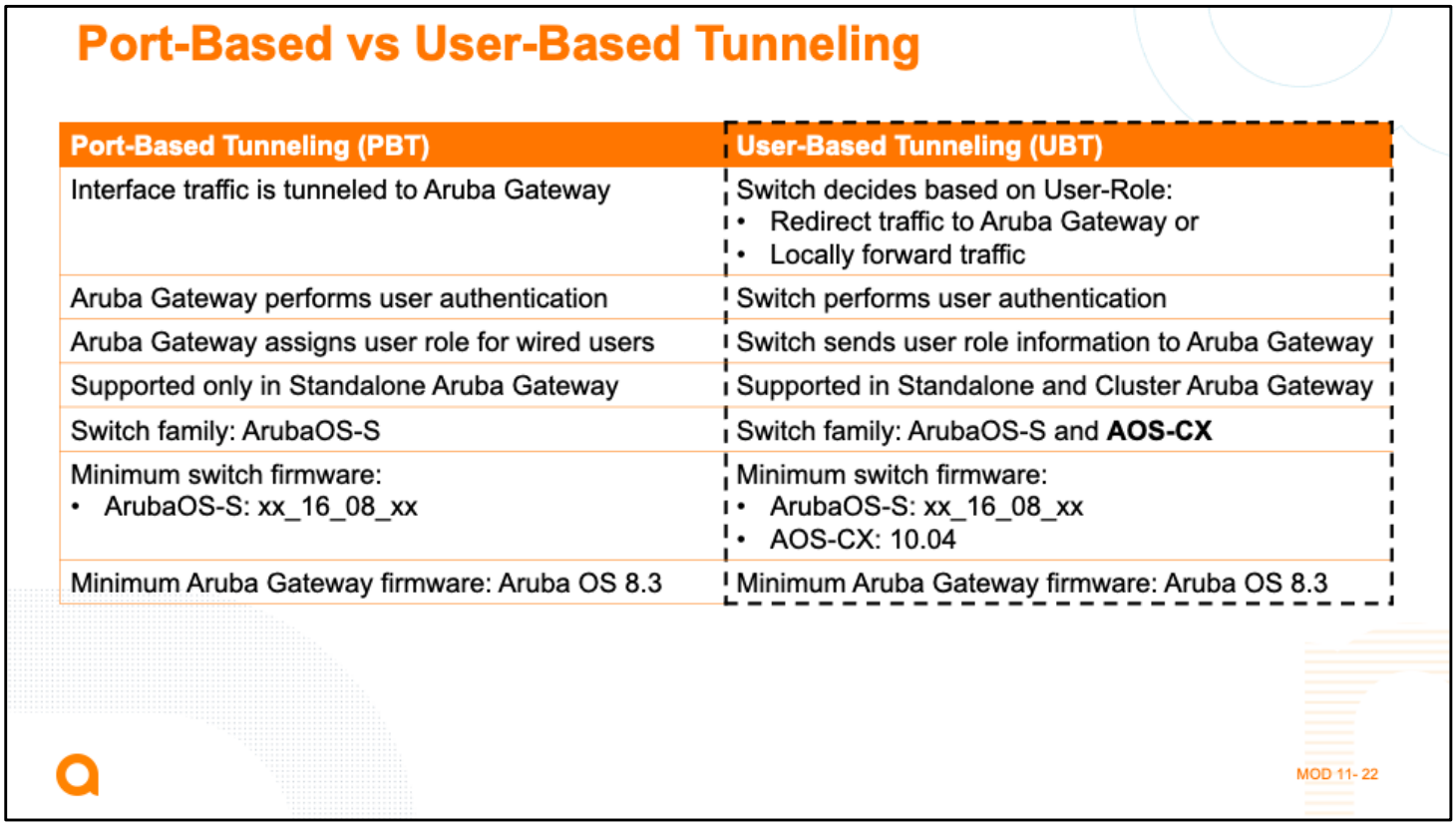
The main characteristics about this authentication are:

Securest authentication

Supplicant must support the authentication type; this means that a software must install a proper software and be configured properly

Bidirectional authentication is possible

Commonly deployed for employees



The Dynamic Segmentation solution has two methods to be deployed: Port-Based Tunneling and User-Based Tunneling.

Port-Based Tunneling (PBT)

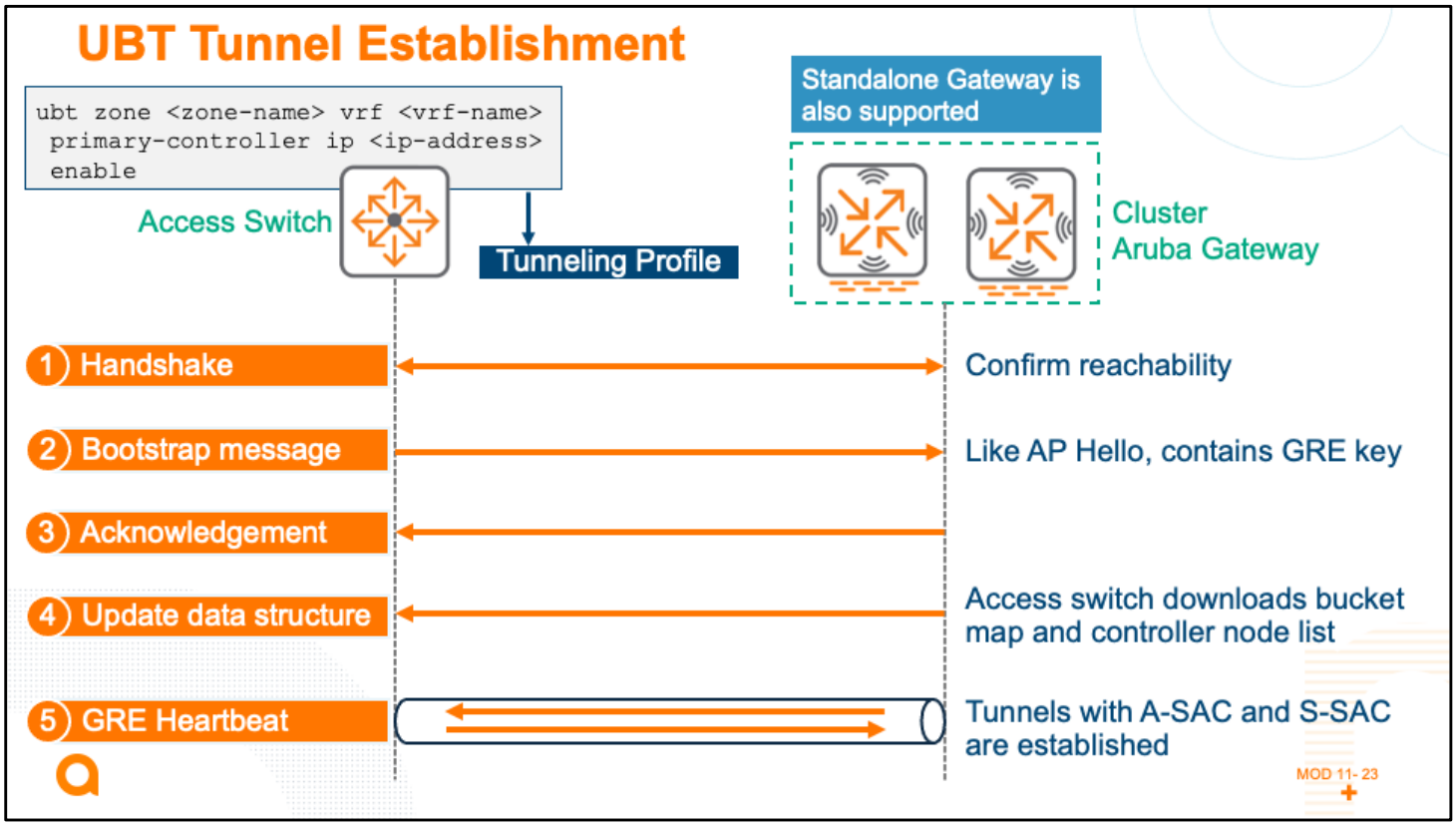
In this method, the traffic received in the interface is unconditionally tunneled to the Aruba Gateway. In this case the Aruba Gateway is in charge to perform the user authentication and assign the user role based on the response of the RADIUS server. This method is only supported in ArubaOS-S switch family (Aruba 5400, 3810, 2930M/F, 2530 and 2520).

User-Based Tunneling (UBT)

In this method, the switch is in charge of performing the user authentication. Based on the authentication result the switch assigns a User-Role (Primary) which contains instructions to locally forward the device's traffic or redirect it to the Aruba Gateway. If the last option is true, the switch sends the user role (Gateway role) information to the Aruba Gateway. This method supports Standalone and Cluster Aruba Gateway deployments.

AOS-CX only supports User Based Tunneling (UBT) configuration method.

Note. For clarification, In the Dynamic Segmentation context the User Role that is mapped to the device on the switch is called a Primary role. The Gateway role, as the name implies, exists on the Aruba Gateway and is applied to tunneled clients.



The AOS-CX switch is configured with a Tunneling Profile which includes the Aruba Gateway IP address. The AOS-CX switch uses this information to do a handshake with the Aruba Gateway to determine its reachability and discover version information.

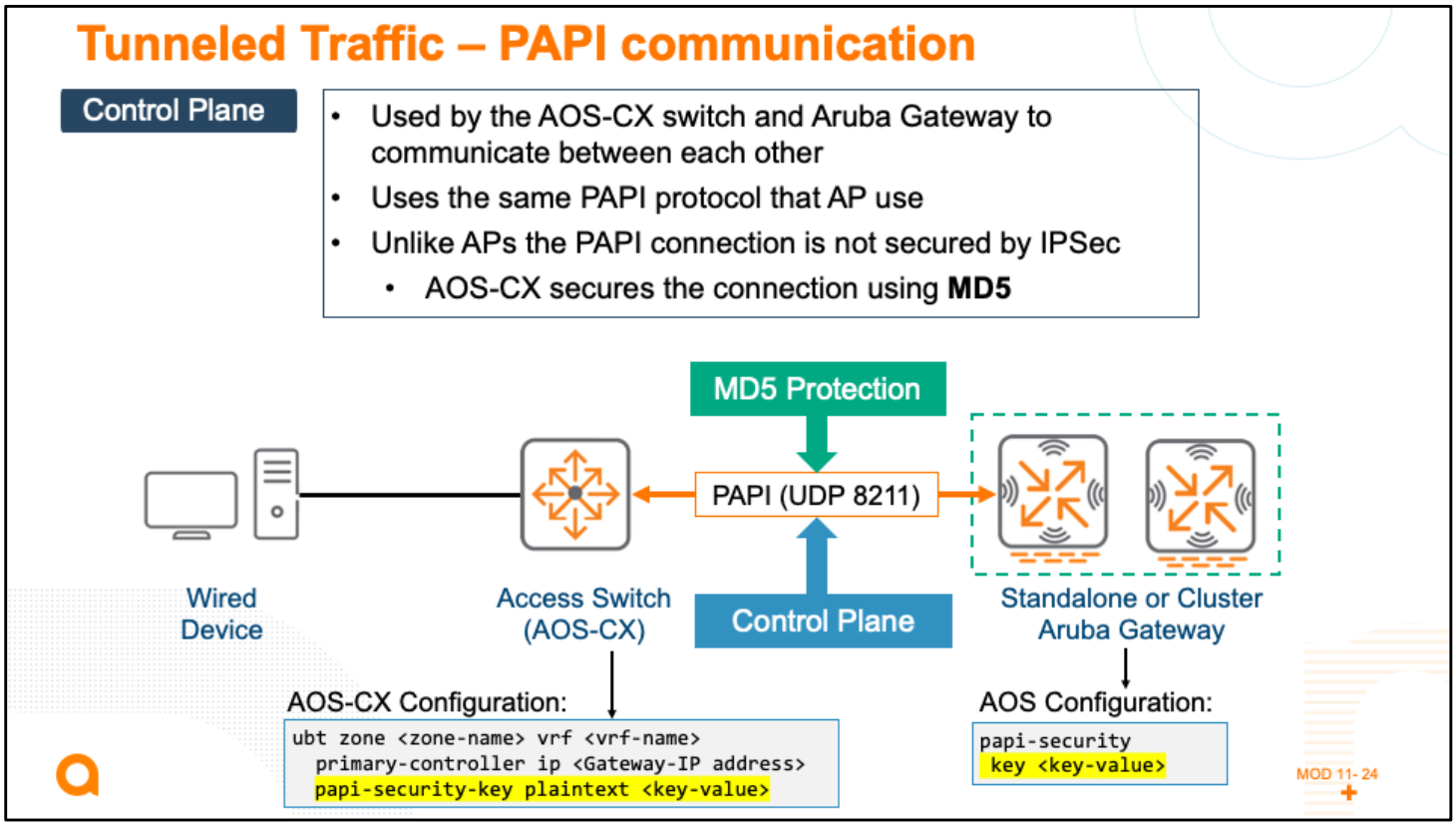
When reachability is confirmed, the switch sends a bootstrap message to the Aruba Gateway. This is like an “AP Hello” between AP and Aruba Gateway. This message contains the GRE key. Once the Aruba Gateway receives the message, it replies with an acknowledge message.

If the Aruba Gateway is part of a Cluster1, then the AOS-CX switch updates its local data structure with a bucket map and controller node list. This switch uses this information to know which member of the Mobility Cluster is used to terminate the GRE tunnel for wired users’ traffic. The cluster provides high availability and the possibility of load balancing the traffic from several wired devices to multiple Aruba Gateways.

After the bucket map is downloaded, a GRE heartbeat starts between the AOS-CX switch and the Aruba Gateway that was elected as the Switch Anchor Controller (SAC). This is the primary-controller IP in the ubt zone command. A secondary heartbeat is established with a standby controller, acting as a secondary switch anchor controller (S-SAC).

Note. If the Aruba Gateway is deployed in standalone mode, then the Access switch skips step number 4 and starts a GRE heartbeat against the standalone Gateway.

1Refer to the User Guide for ArubaOS 8.6.0.0 for additional information about Mobility Clustering.
https://support.hpe.com/hpesc/public/docDisplay?docId=a00092459en_us



The switch uses two protocols to connect to an Aruba Gateway. PAPI which is used as the protocol that runs the control plane and GRE which is used to encapsulate the user's traffic (data plane). PAPI runs on UDP 8211 and this protocol is the same used by AP-to-Aruba Gateway communications. However, where APs use IPSec to protect the PAPI connection between the AP and Aruba Gateway, AOS-CX switches do not support this protection. Instead, you can optionally implement an MD5 HMAC function to protect PAPI between the AOS-CX switches and MCs.

If PAPI is not protected, this communication creates a potential security risk. Rogue devices could start sending PAPI messages to the APs and switches and disrupt functionality. They could also try to insert spoofed PAPI messages on the controller to disrupt communications or obtain information about the network. PAPI enhanced security minimizes this risk by using a MD5 and user-configurable key to hash the message. If a rogue device tries to spoof or tamper with a message, the hash will not check out because the rogue does not know the correct key to create the hash.

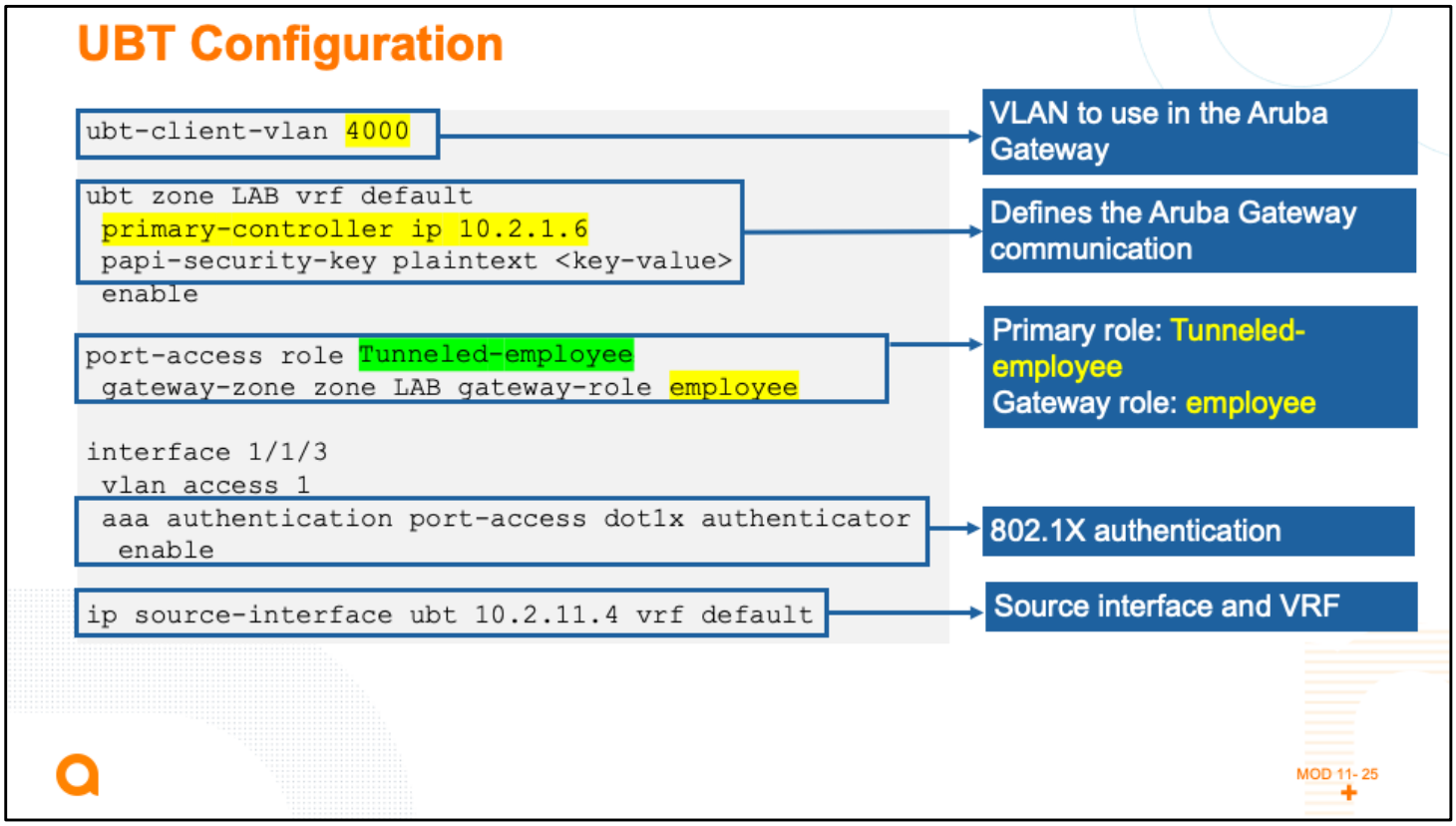
If administrators have enabled PAPI enhanced security on the MC, you must set a matching PAPI key on the switch. The key must be between 10 and 64 characters. The switch then no longer accepts PAPI messages unless they have a valid hash created with this key. It is also able to send its PAPI messages with the correct key. Enhanced security mode is disabled by default.

PAPI Security configuration on Aruba Gateway (AOS 8.5.0.4):

(Aruba Gateway) [mynode] (config) # papi-security
(Aruba Gateway) ^[mynode] (PAPI Security Profile) #key <key-value>

PAPI Security configuration on AOS-CX 10.05:

ubt zone <zone-name> vrf <vrf-name>
primary-controller ip <Gateway-IP address>
papi-security-key plaintext <key-value>



The figure shows a UBT configuration example, as described below.

1. The ubt zone <zone-name> vrf <default> command defines the Aruba Gateways used for this deployment. AOS-CX supports the definition of primary and secondary Aruba Gateway IP addresses. The PAPI security is defined under the ubt zone command.

2. The port-access role <role> command defines the Local User-Role that is assigned to the client. This User-Role can contain changes that can be applied to the received traffic. This includes policies to filter the traffic or QoS configuration. In this case the User-Role defines a redirection action to the Aruba Gateway, as defined in the command gateway-zone zone <zone-name> gateway-role <gateway-role>. Notice that the last part of this last command includes the gateway-role, the one that the Aruba Gateway must use.

Note. The VSA received by the RADIUS server must match the string in the Local User-Role, so the switch can properly use it.

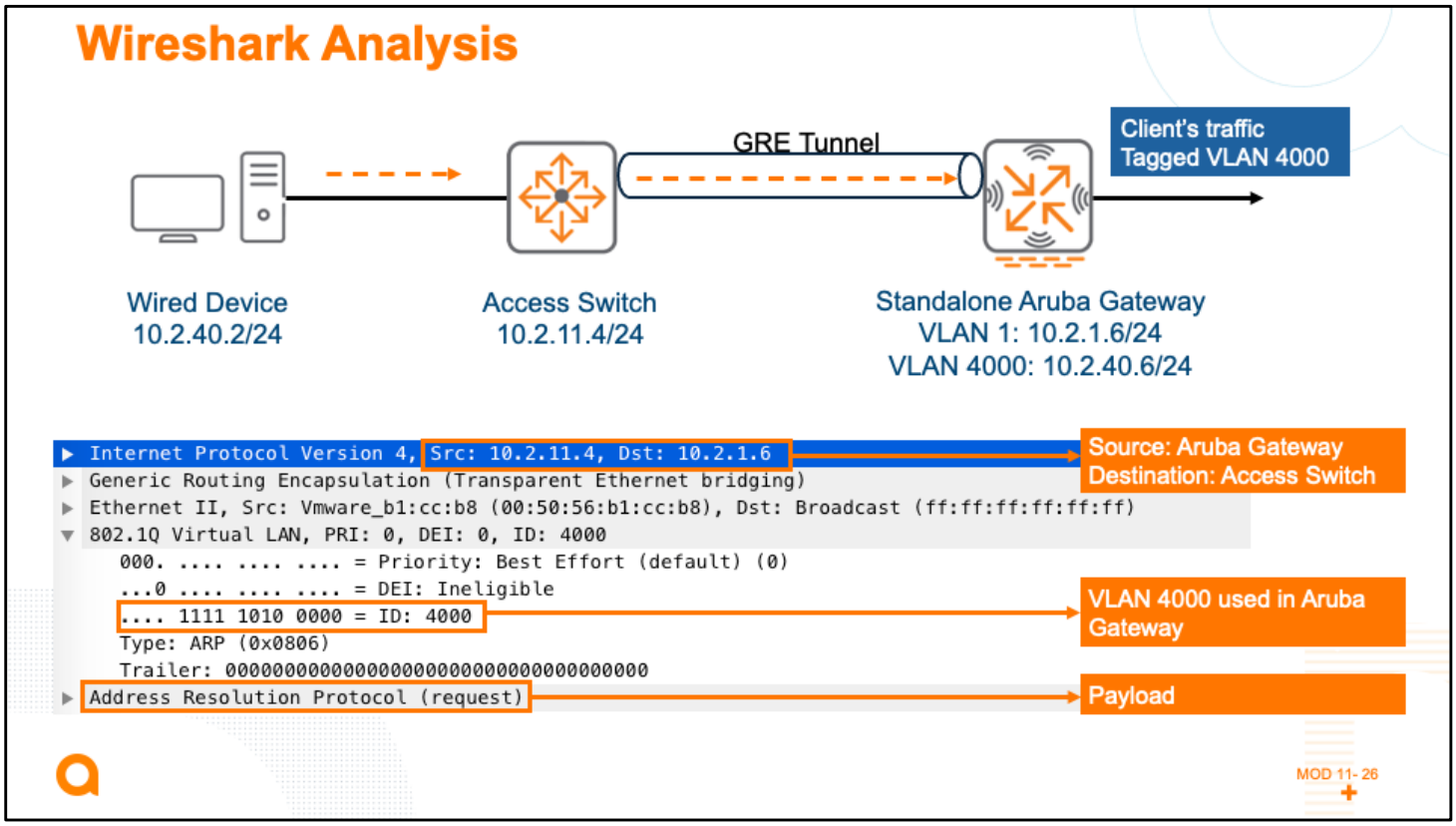
3. The interface where you want to connect the wired device must be set up with authentication. In this case you setup an 802.1X authentication, but you can also configure MAC authentication.

4. The source-interface ubt command defines the IP address and the VRF to use to connect

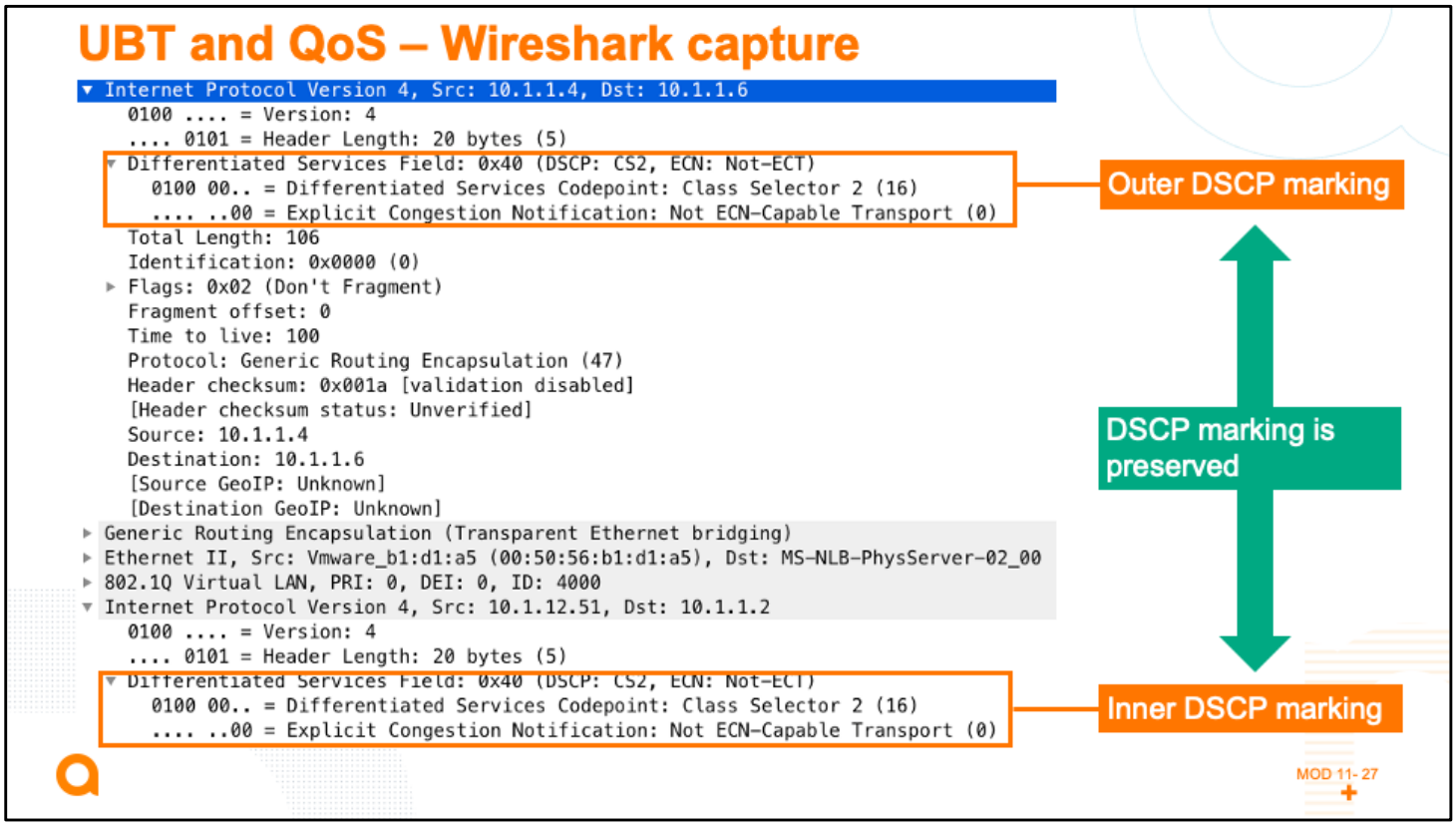
with the Aruba Gateway.

5. The ubt-client-vlan <vlan-id> command tells the Aruba Gateway which VLAN to use for this client. This VLAN must exist in the Aruba Gateway.

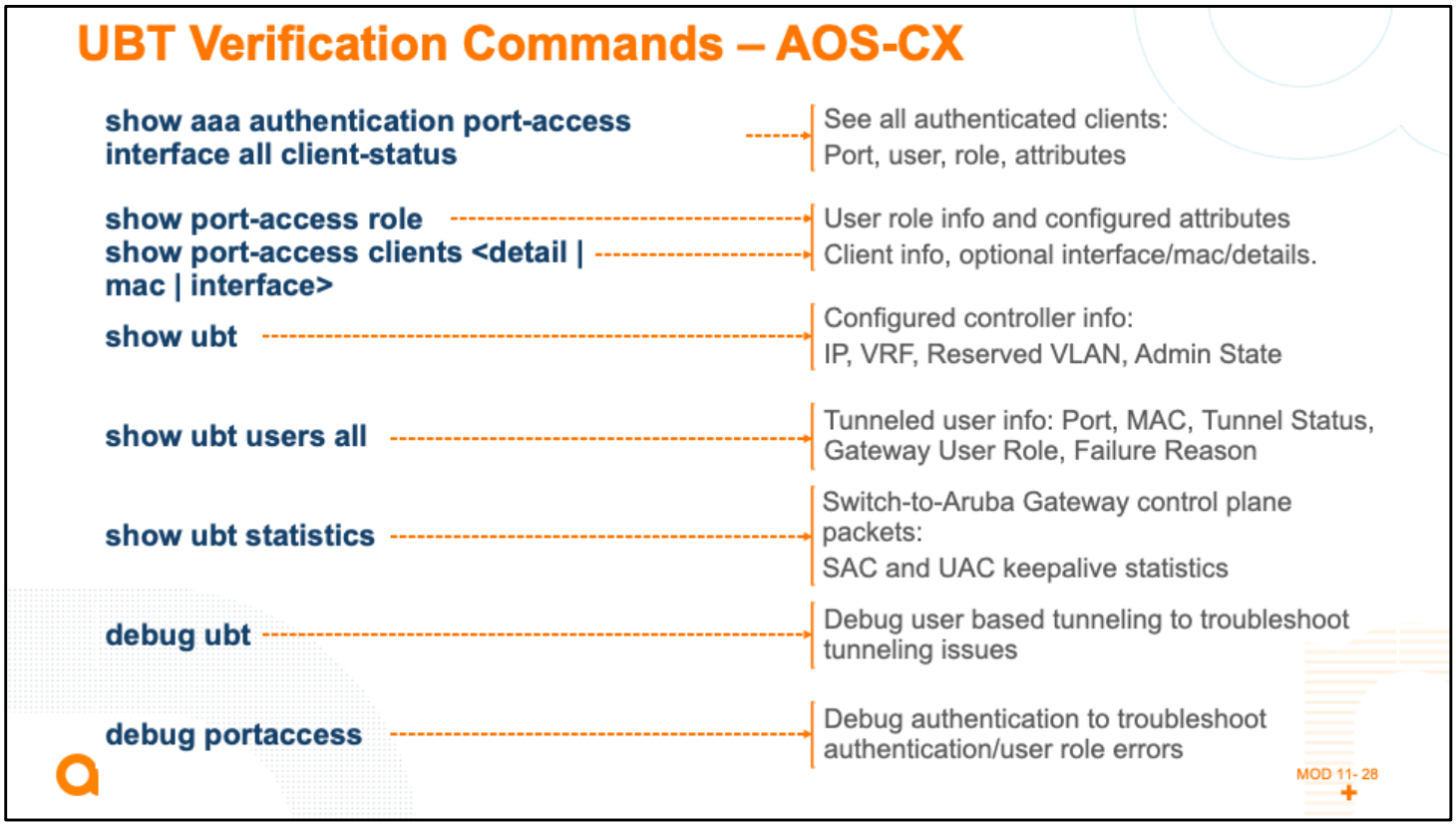
Note. The port-access role command has the capability to apply different settings to the traffic that has been mapped to it. This includes the following options:



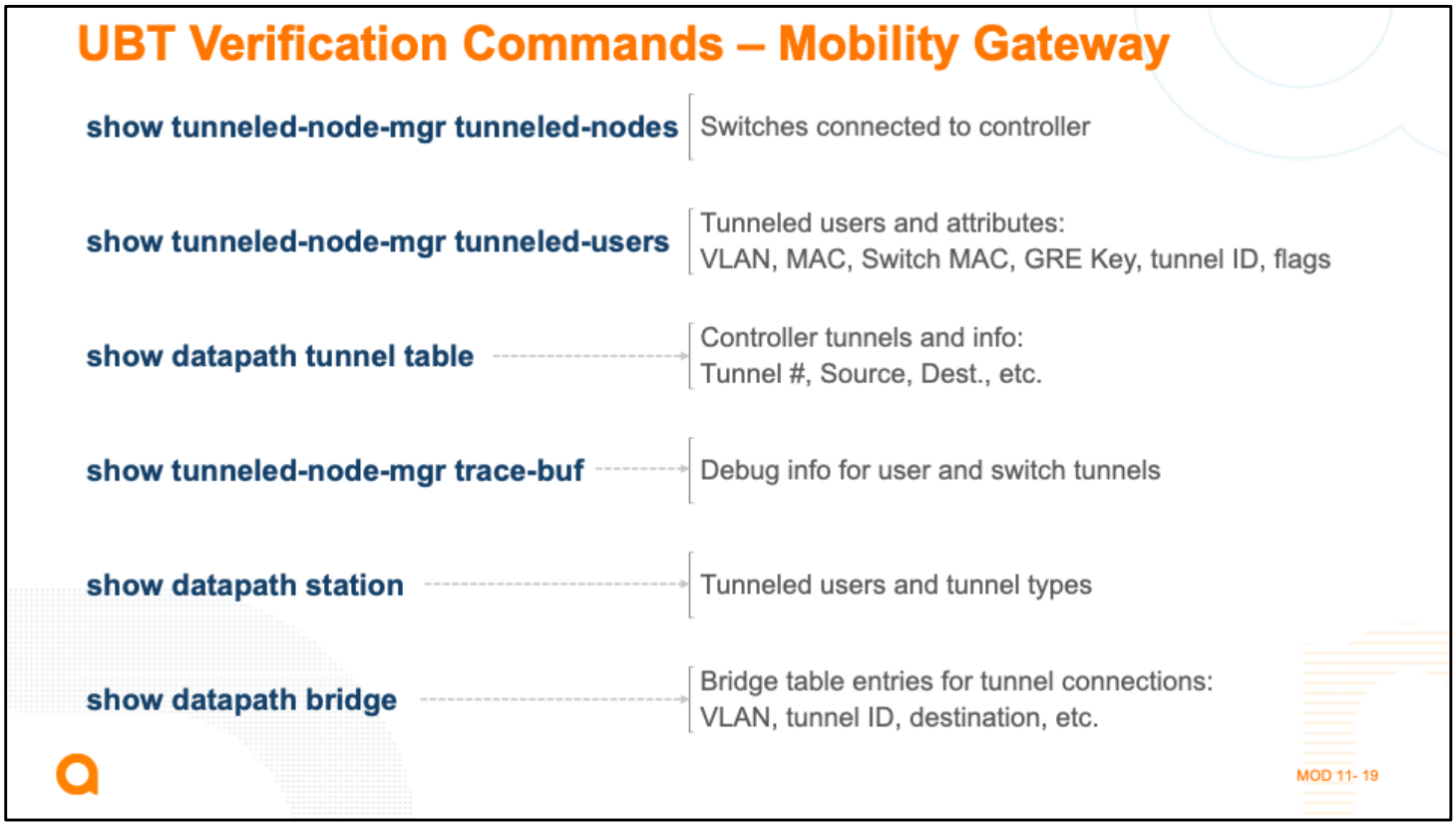
This Wireshark capture shows traffic between the Access switch and the standalone Aruba Gateway. Notice the external Source and Destination IP address. The switch adds the 802.1Q Header, including the VLAN-ID 4000.



The Wireshark capture shows how the inner traffic is properly marked with DSCP: CS2 as well as the outer IPv4 header. As we discussed, this give consistency on the treatment of the traffic by all devices in the path.



These are the various commands useful for troubleshooting user-based tunneling for the switch in AOS-CX. These commands show authenticated clients, tunneling statistics, tunneled users, and debug output for both authentication and tunneling.

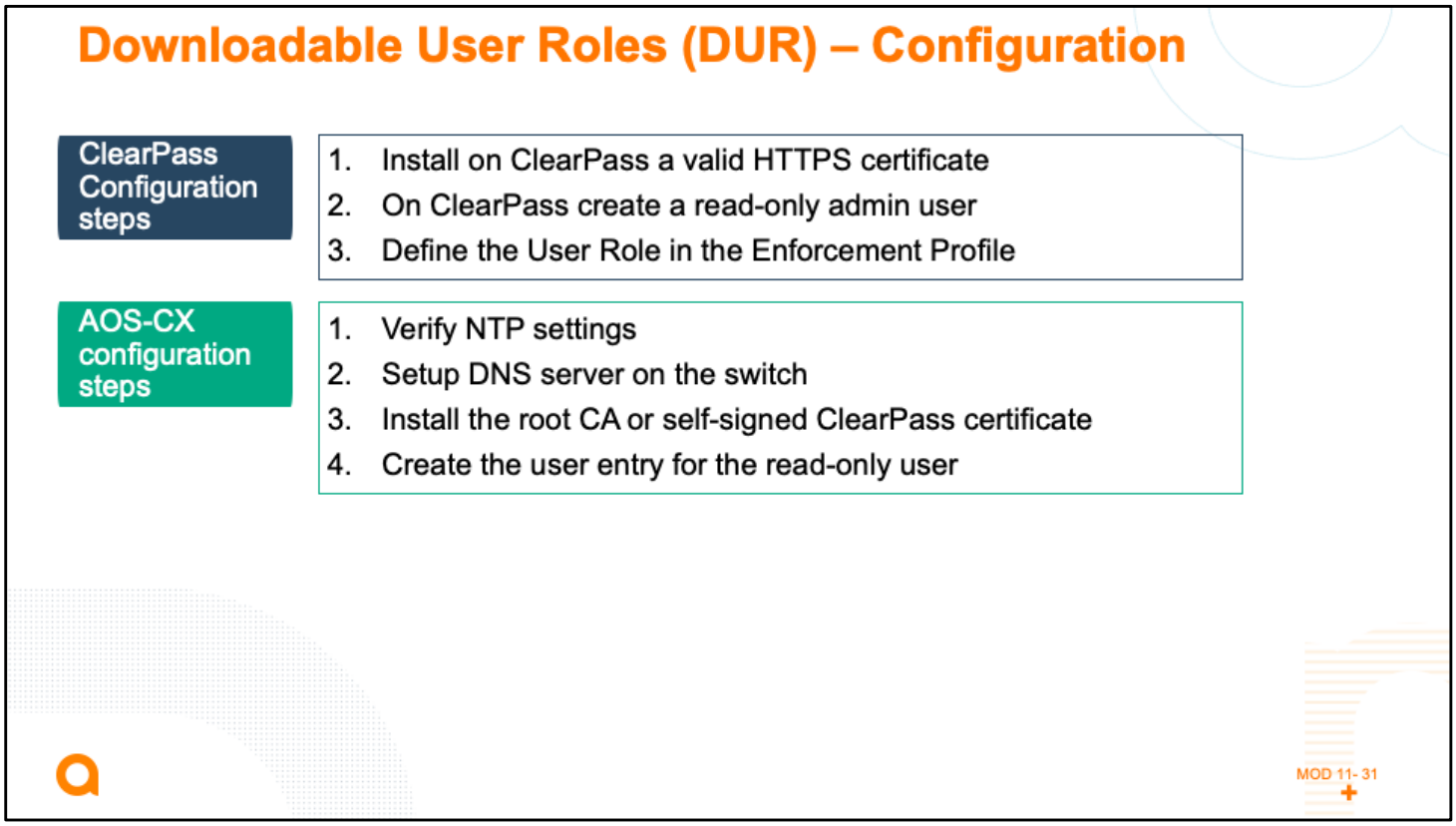


The commands listed here are for troubleshooting and displaying output for tunneled users on the Aruba Gateway. This is useful if, at the switch, the feature seems to be working correctly, but at the Aruba Gateway there may be issues.

Downloadable User Roles

Configuration and verification

aruba
a Hewlett Packard Enterprise company



DUR ClearPass configuration steps:

Install on ClearPass a valid HTTPS certificate

Create a read only admin user that will be used by the switch for authentication in order to download the actual role information

Define the User Role and the commands in an Enforcement Profile

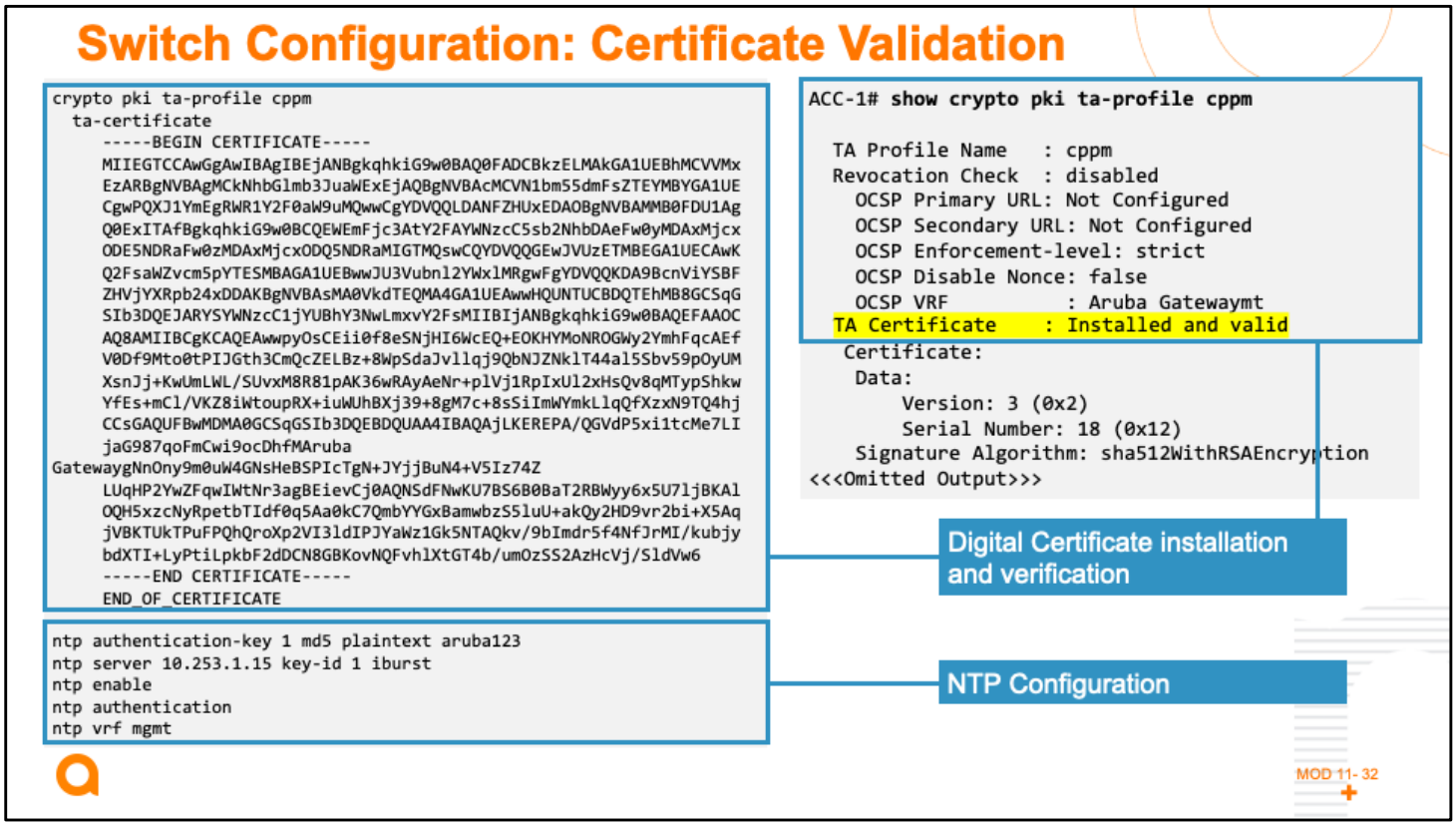
DUR AOS-CX configuration steps:

Verify the NTP settings for the HTTPS certificate validation

Setup DNS server on the switch

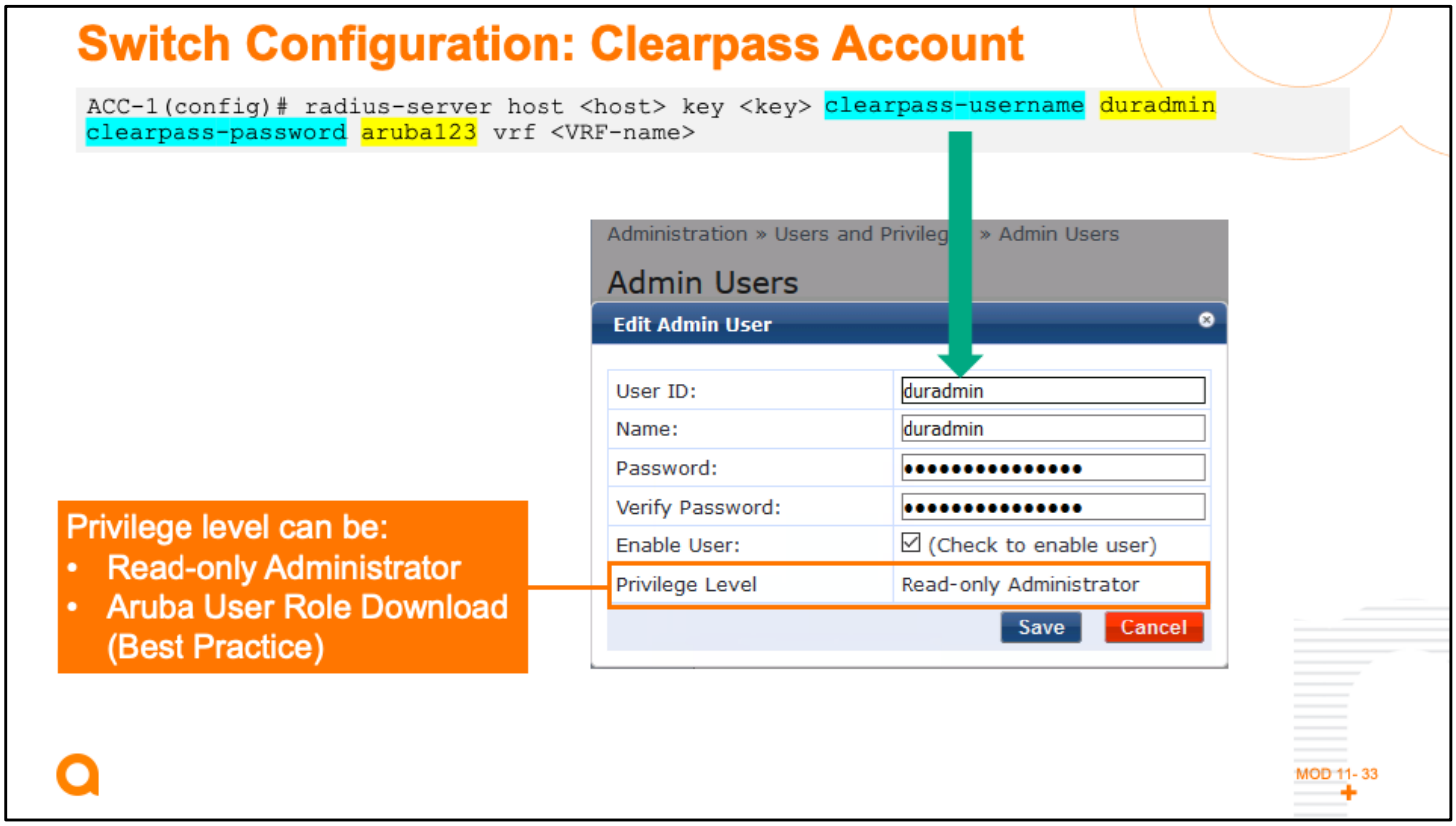
Install the root CA for the HTTPS certificate

Create the user entry for the read only user and enable downloadable user roles



ClearPass will send a user role to the edge switch after successful authentication. The method for sending the user role is through the REST API using an SSL connection. This means that a valid HTTPS certificate has to be installed on the edge switch: one from a trusted Certificate Authority (CA) of both the switch and the ClearPass server.

You will also need to configure NTP to ensure the certificate for the switch is valid based on the current date and time and the time range defined on the certificate. If the switch's time falls outside the range of the time on the certificate, the certificate is automatically invalidated.

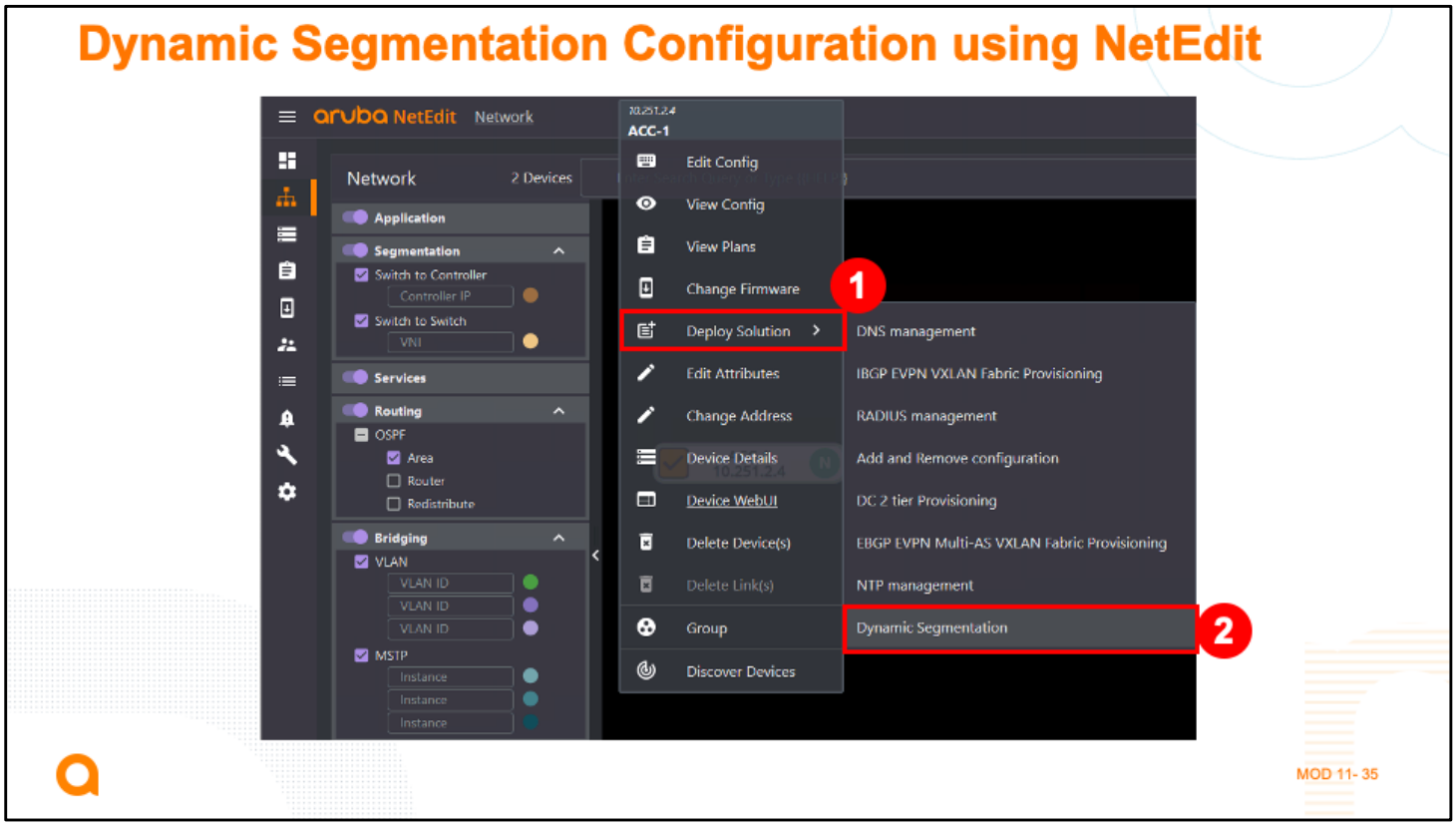


In addition, a username and password must be configured to authenticate the switch with the ClearPass server. This is defined on the ClearPass server but referenced on the switch. The privilege level can be “Aruba User Role Download” or “Read-Only Administrator”; The best practice is to use “Aruba User Role Download” privilege.

Deployment Using NetEdit

Configuration and verification

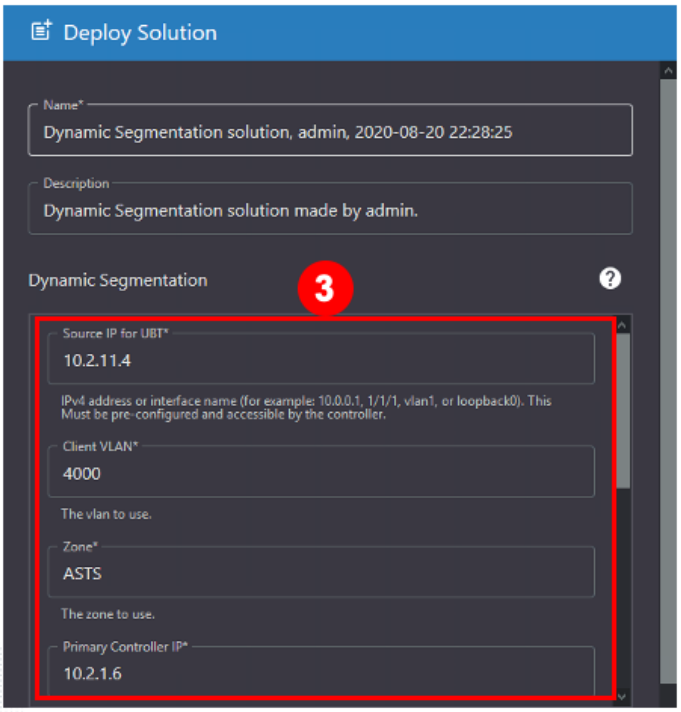
aruba
a Hewlett Packard Enterprise company



NetEdit can help to deploy Dynamic Segmentation in AOS-CX switches, using as simple wizard.

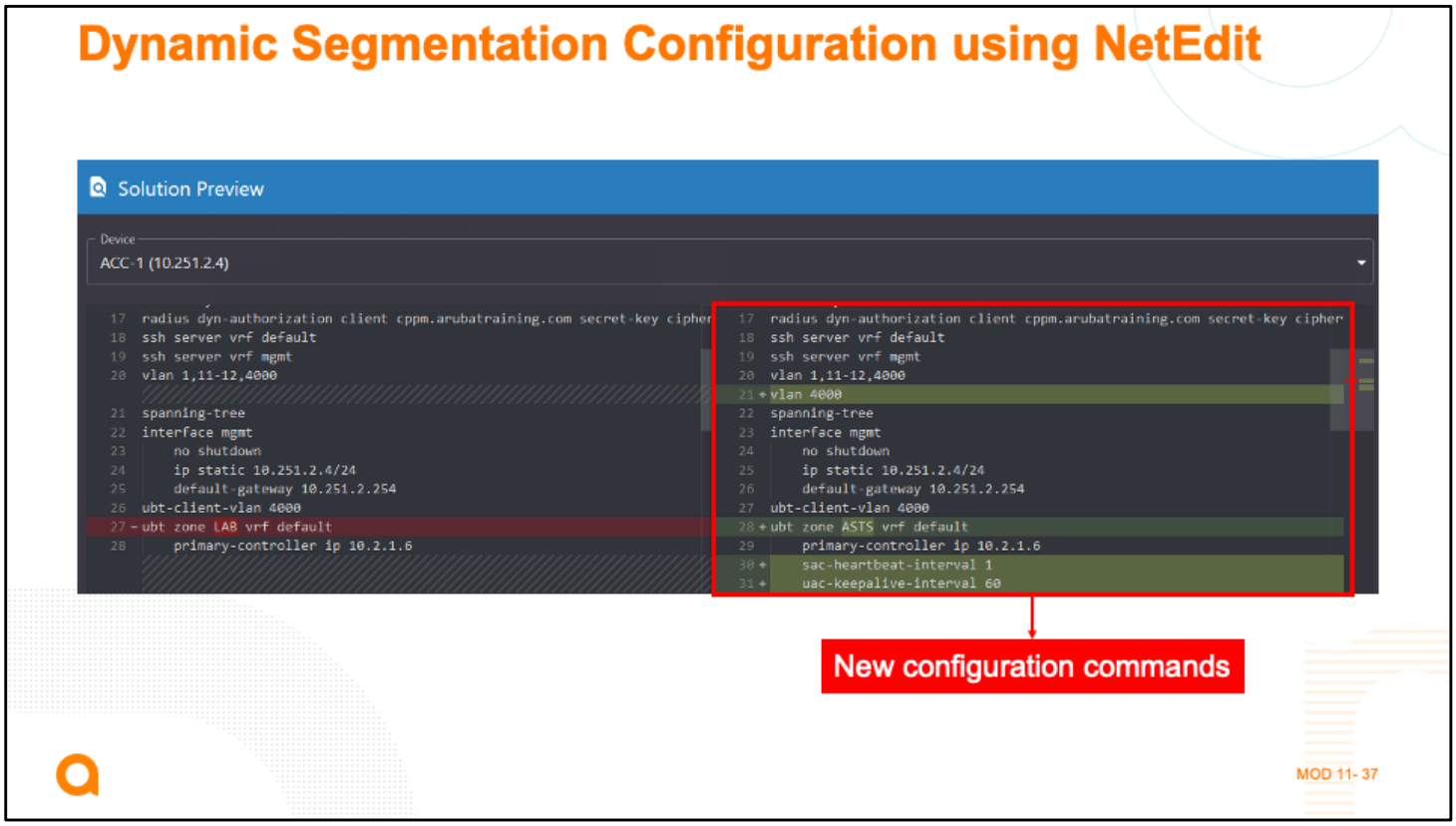
Move to the Network section, right click on the Switch access icon, then select Deploy Solution > Dynamic Segmentation.

Dynamic Segmentation Configuration using NetEdit

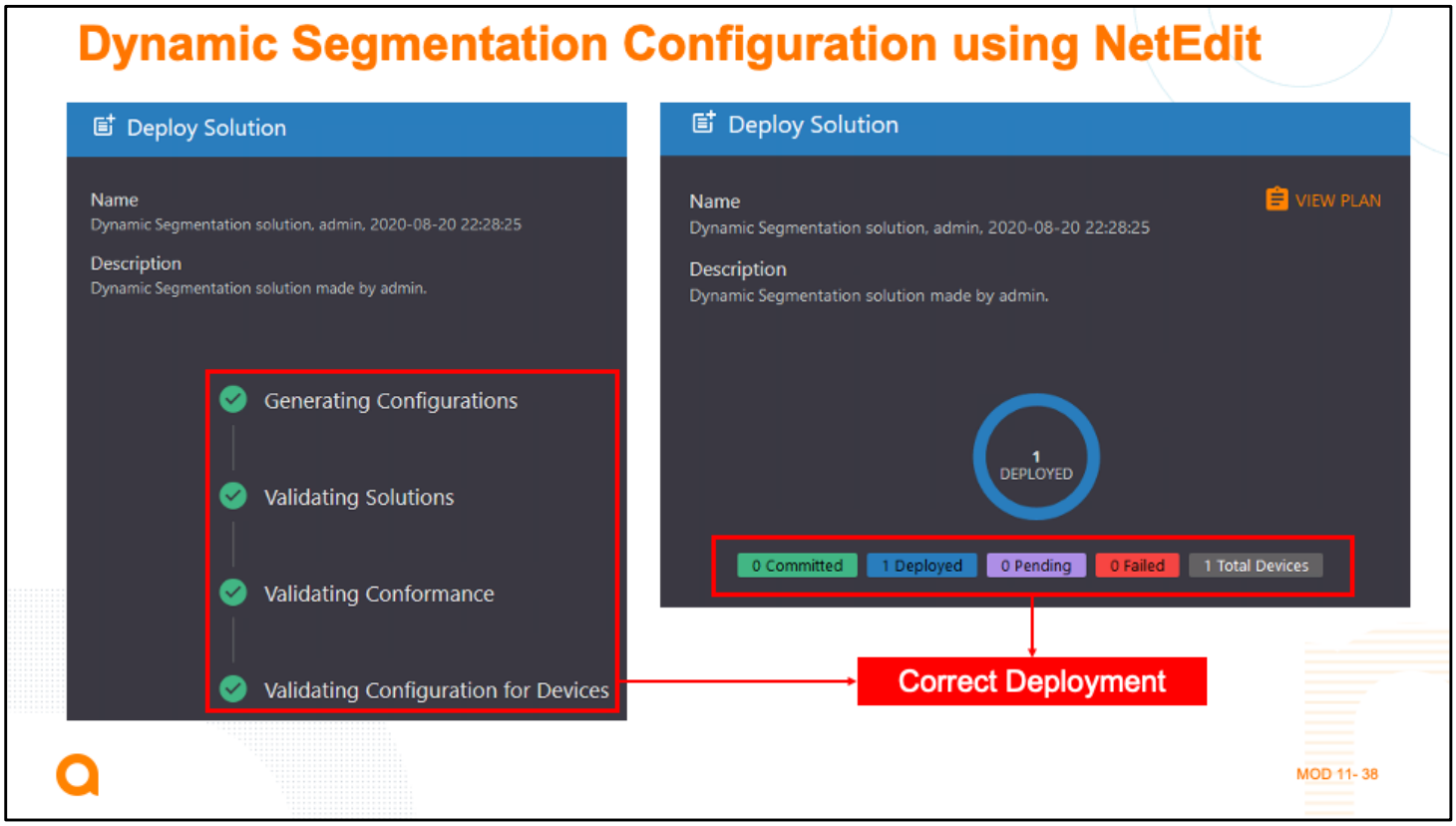


MOD 11- 36

The next step is to simply complete the wizard



NetEdit gives lets you see the commands that will be added prior the deployment.



The last step is to verify the correct solution deployment.


Lab Activity
Lab 11


Rev #20.41
aruba
a Hewlett Packard Enterprise company
Network Security
Aruba Switching Troubleshooting and Solutions (ASTS)

Objectives

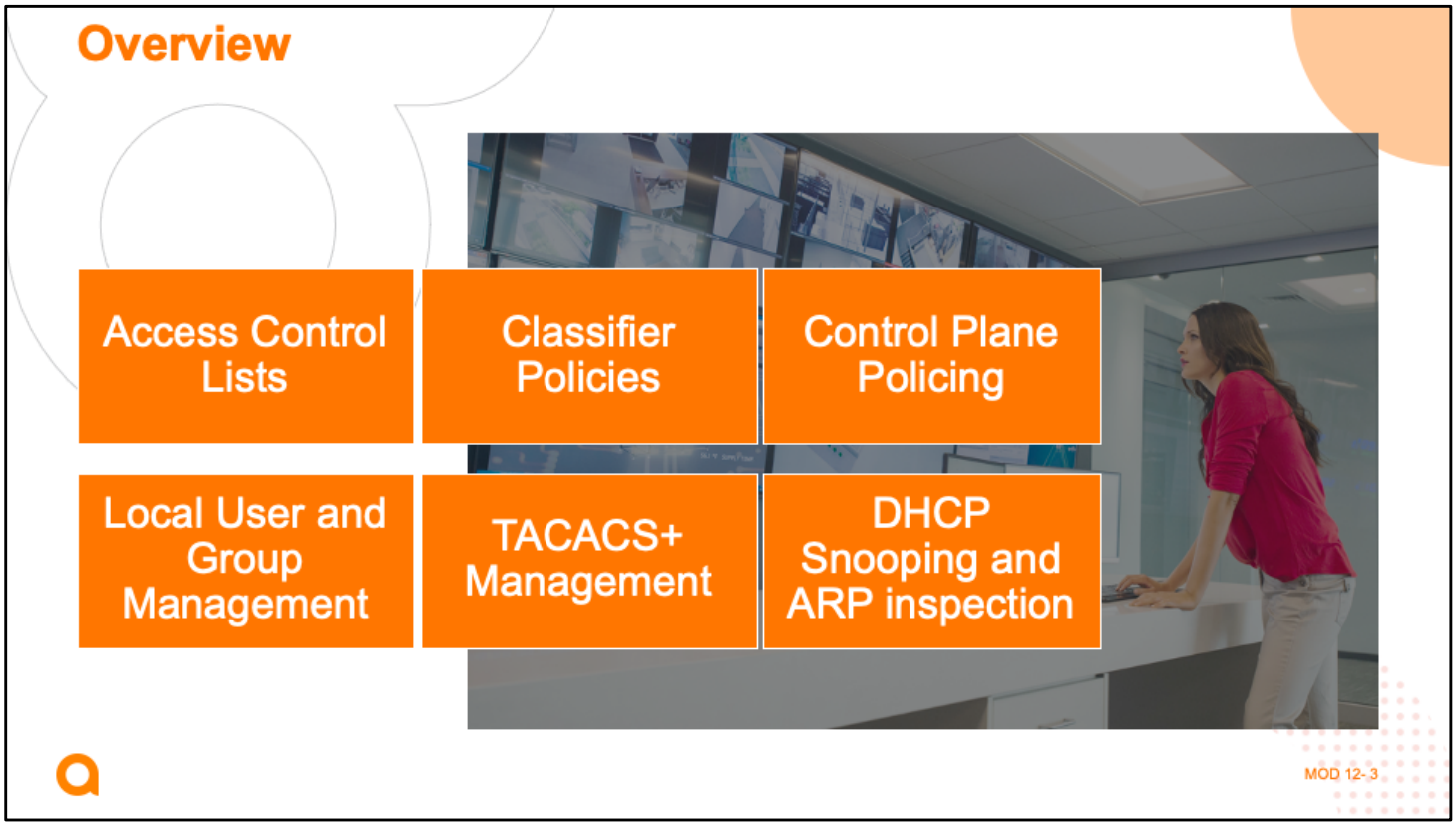
- Describe the operation and application of Access Lists (ACL) and object groups
- Configure and verify ACLs with counting and logging features
- Describe, configure, and verify classifier policies and Control Plane Policing (CoPP)
- Describe, configure, and verify local and TACACS+ based user management
- Describe, configure, and verify DHCP snooping and ARP inspection



After completing this module, you will be able to describe, configure, and verify several network security-related features. This includes ACLs and object groups, with counting and logging features.

This also includes control plane security with Control Plane Policing (CoPP), and the management plane security with user management – both local management configuration on the switch, and TACACS+ based management.

Finally, you will be able to describe, configure, and verify additional security features - DHCP snooping and ARP inspection.



The most important network capabilities include reliability, reachability, resiliency, scalability, manageability – and security. This module will help you maintain operational security and situational awareness as relates to network security, starting with Access Lists (ACL). You will configure ACLs with various features – object groups, counting, and logging.

Now, think about security based on network architecture planes – data, control, and management.

For the data plane, you learn about Classifier Policies - rules based on traffic attributes to control traffic through a switch.

For the control plane, you deploy Control Plane Policing (CoPP) to mitigate Denial of Service attacks and more.

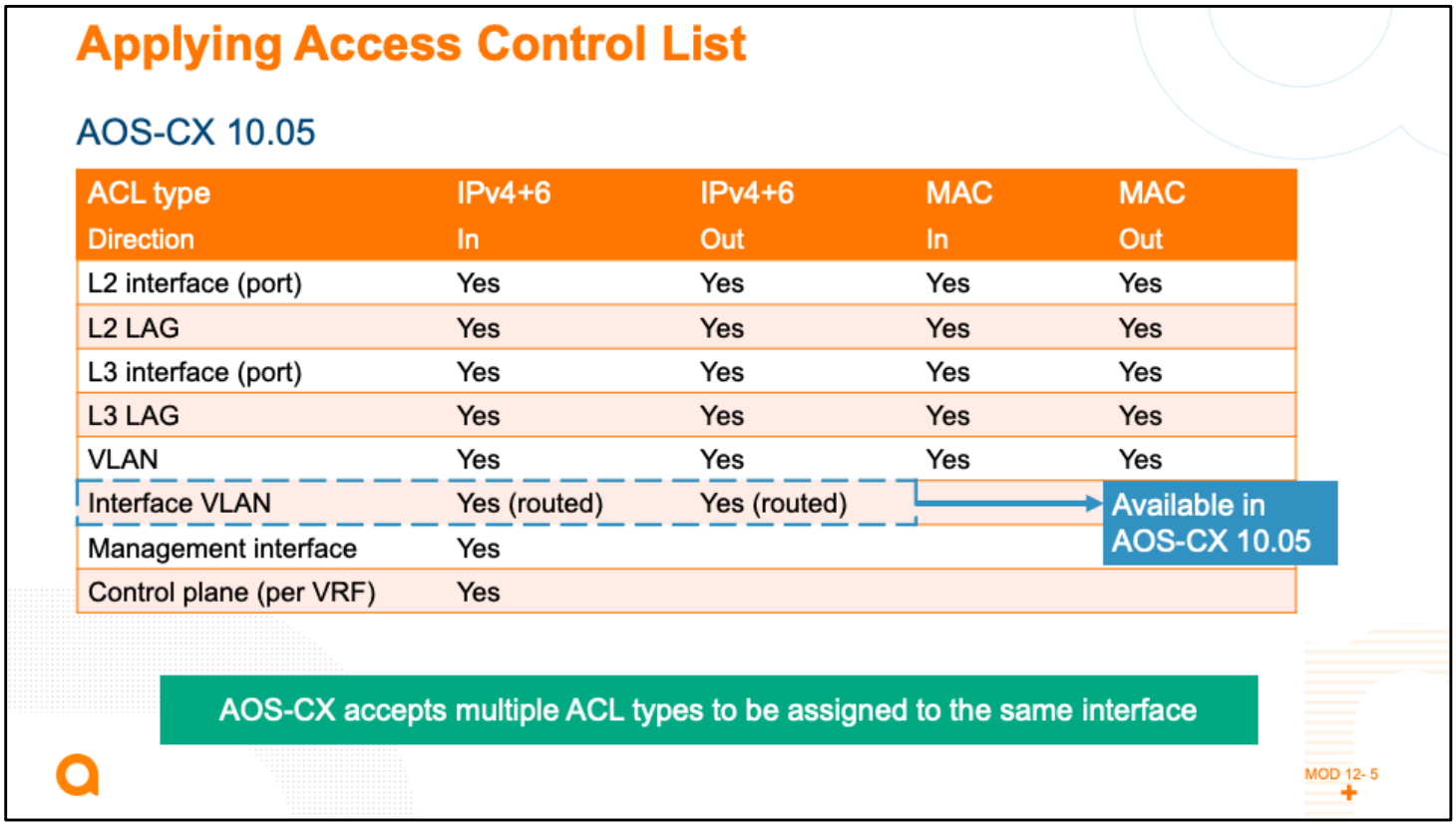
Next, you secure the management plane with local configuration on each switch, or with switch integration with an external TACACS+ server for a more centralized solution.

You finish up by locking things down a little tighter with DHCP snooping and ARP inspection.

Access Control List

Configuration and Validation

aruba
a Hewlett Packard Enterprise company



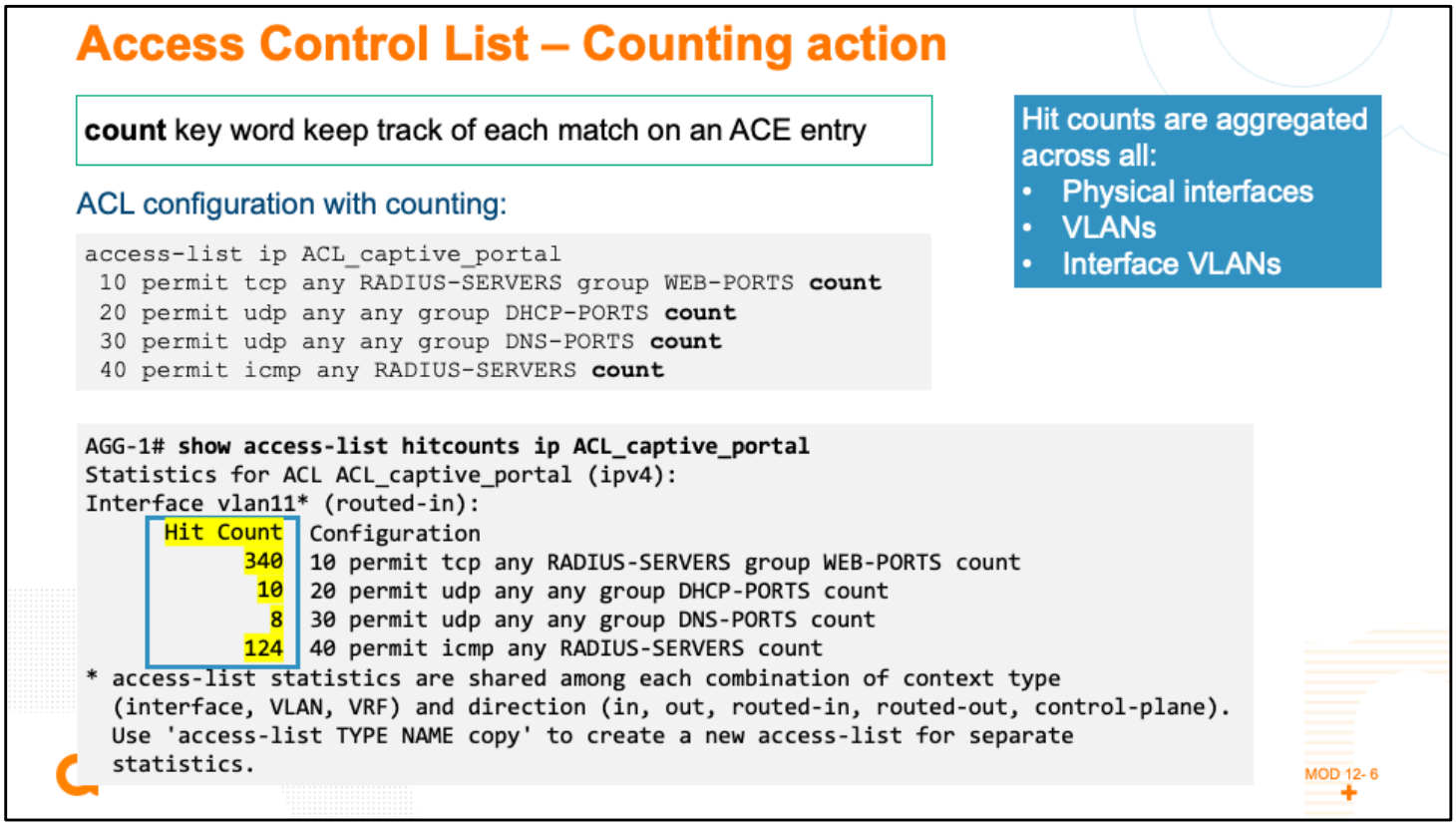
ACLs must be applied (using an apply access-list command) to take effect. ACLs can be applied to interfaces (including LAGs), VLANs, or the Control Plane. ACLs can be applied to an interface in two directions (inbound and outbond). The table in the slide shows the Different options that exist to apply an ACL.

Notice that AOS-CX accepts multiple ACL types to be assigned to the same interface as follows:

- One MAC ACL inbound
- One MAC ACL outbound
- One IPv4 ACL inbound
- One IPv4 ACL outbound
- One IPv6 ACL inbound
- One IPv6 ACL outbound

Note. If you apply an ACL of a type, in a direction that is already in use, the switch replaces the current ACL with the new ACL.

Note. In AOS-CX 10.05 the operating system allows you to assign an Access-List to the SVI (Interface VLAN), previous releases did not support it. The only possibility to have a similar behavior on those releases was using a Class-Based Policy.



The count key word will keep track of each match on an ACE entry.

The show access-list hitcounts command is used to display the counting of each ACE.

Use the command clear access-list hitcounts command to reset the hit counts. This command can be applied to all ACLs or to a specific ACL or to a specific interface.

Note. The data displayed in the show access-list hitcounts command is data that has been aggregated if the same access-list has been applied to multiple interfaces:

ACL hit counts are aggregated across all:

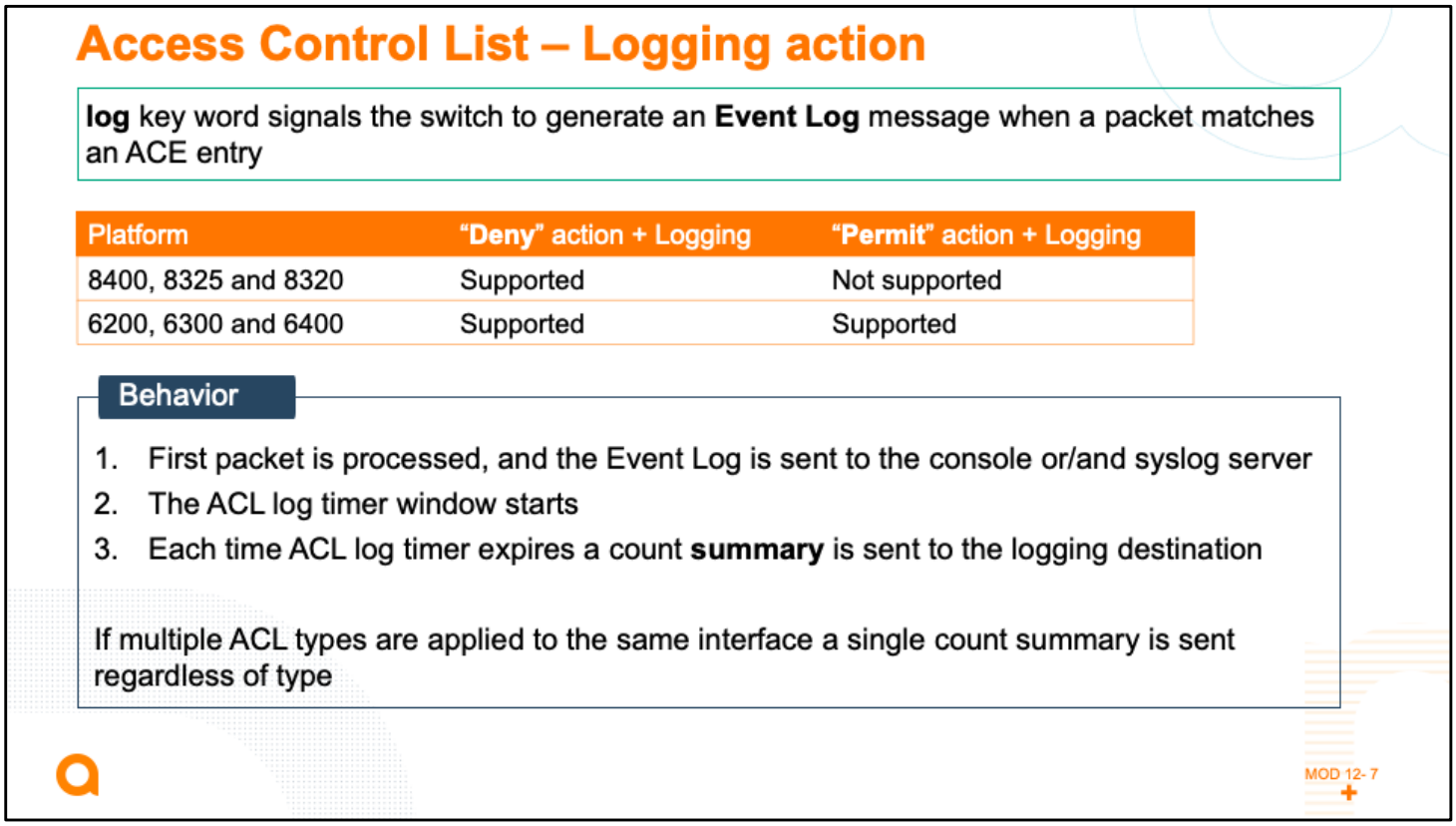
- Physical interfaces to which the ACL is applied to on ingress,
- Physical interfaces to which the ACL is applied to on egress,
- VLANs to which the ACL is applied to on ingress.
- VLANs to which the ACL is applied to on egress.
- Interface VLANs to which the IPv4 or IPv6 ACL is applied on routed ingress
- Interface VLANs to which the IPv4 or IPv6 ACL is applied on routed egress

If an ACL with an ACE with the count keyword is applied to multiple physical interfaces or VLANs, the hit counts are aggregated. There is one aggregation for physical interfaces and

another for VLANs.

Accumulated hit counts for an applied ACL are cleared upon any modification of the ACL.

In case you want to review specific hit counts to an interface, you must create a unique access-list and apply it to the interface. To simplify this task ArubaOS-CX allows you to copy the access list with the access-list <type> <name> copy command.



The log option signals the switch to generate an Event Log message when a packet matches an ACE entry. Depending on the platform the log option can be applied with an action:

8400, 8325 and 8320 switch series support only deny actions to be used with logging.

6200, 6300 and 6400 switch series support both permit and deny actions to be used with logging.

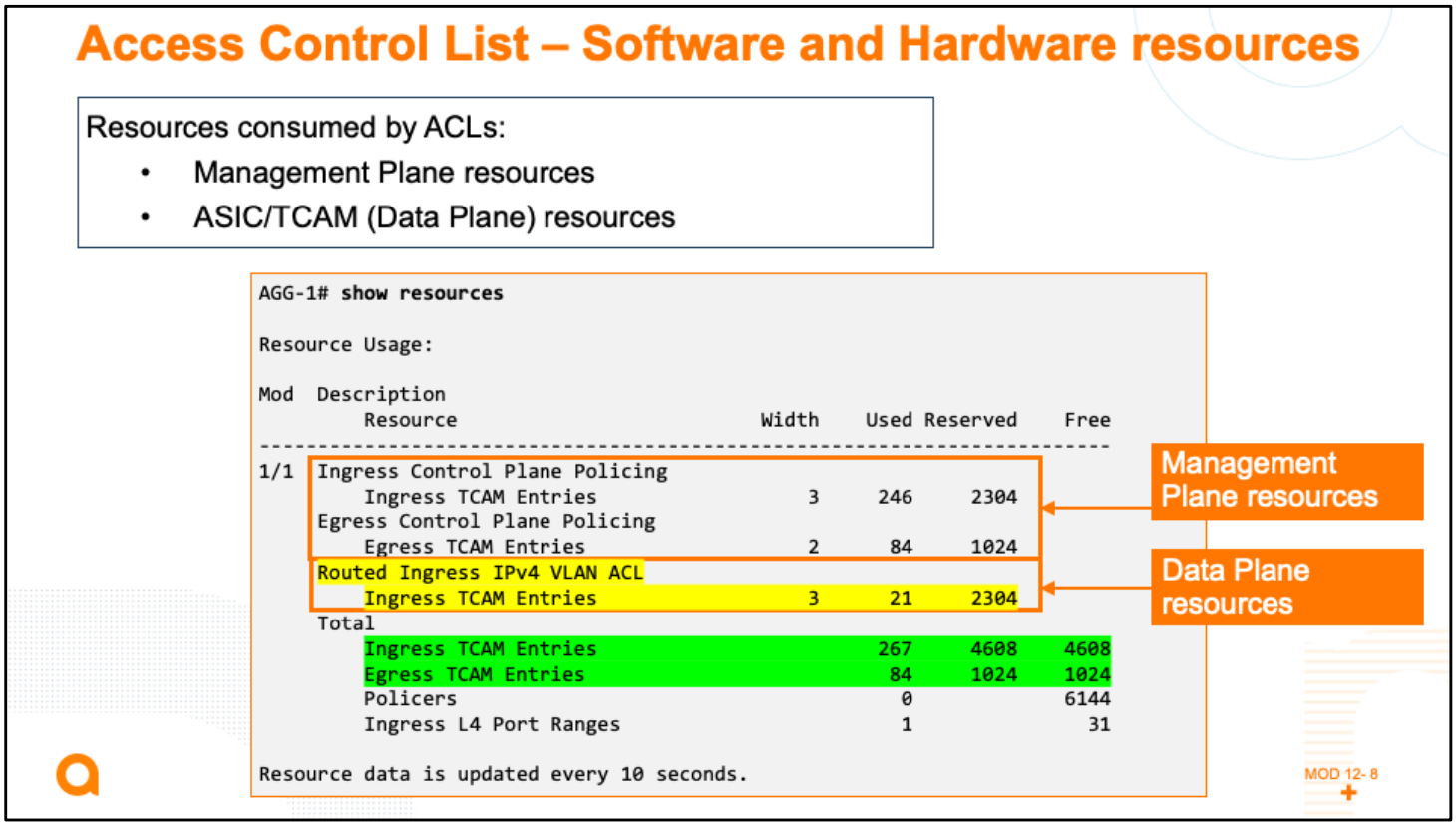
Behavior:

The first packet that matches an ACE with the log parameter within an ACL log timer window (configured with the access-list log-timer command) has its header contents extracted and sent to the configured logging destination, such as the console and syslog server. Each time the ACL log timer expires, a summary of all ACEs with log configured are sent to the logging destination. This capability allows throttling of logging ACL hits.

If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to log as soon as new match occurs.

Behavior using multiple ACL types with logging on the same interface:

The first packet that matches an ACE with the log option is logged. Any packets, matching other ACL types, do not create a log until the log-timer wait-period is over. At the end of the wait-period, a summary log is made of all the ACLs that were matched, regardless of type.



Switches have finite hardware resources used in the application of ACLs. When applying ACLs, the following hardware resources are consumed to enforce these rules on the data plane:

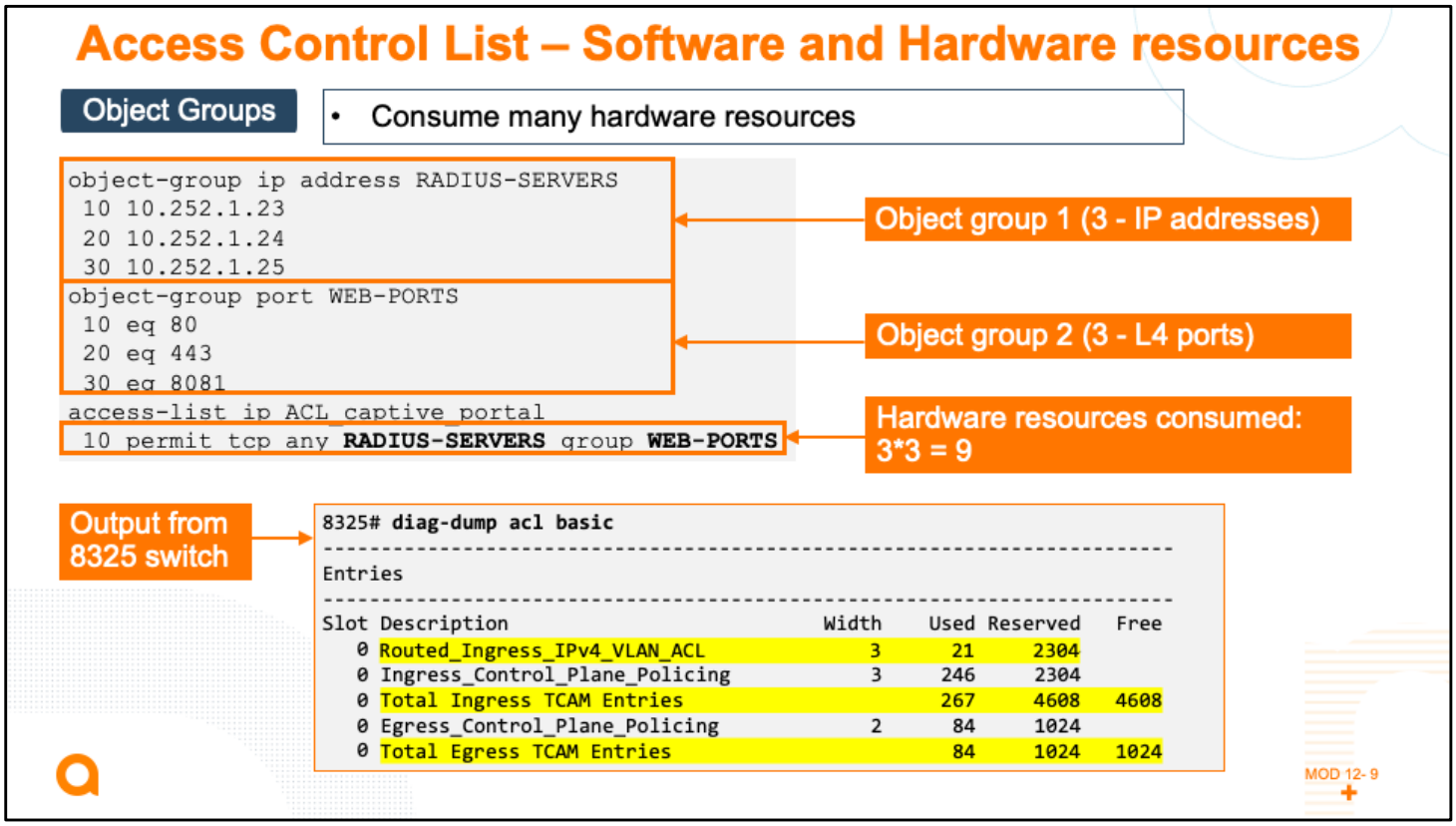
1. Management Plane Resources.
2. ASIC/TCAM (data-plane) Resources.

Note. Switching ASICs have different types of tables where they keep MAC addresses, ARP and ND entries, IPv4 and IPv6 routes, access control lists and policies. The resource used to perform lookup at hardware speed and to provide index corresponding to a given search-key is called a TCAM: Ternary Content Addressable Memory. All AOS-CX switches have “internal TCAMs” which are responsible for look-ups inside the ASIC of the switch or of the line-card.

You can track the available hardware resources in the switch using the following commands:

show resources
show capacities-status classifier
diag-dump acl basic

In the example shown in the slide you can see how the ACL that we have defined previously (ACL_captive_portal) is consuming some resources in the system, pay special attention to the TCAM consumption.



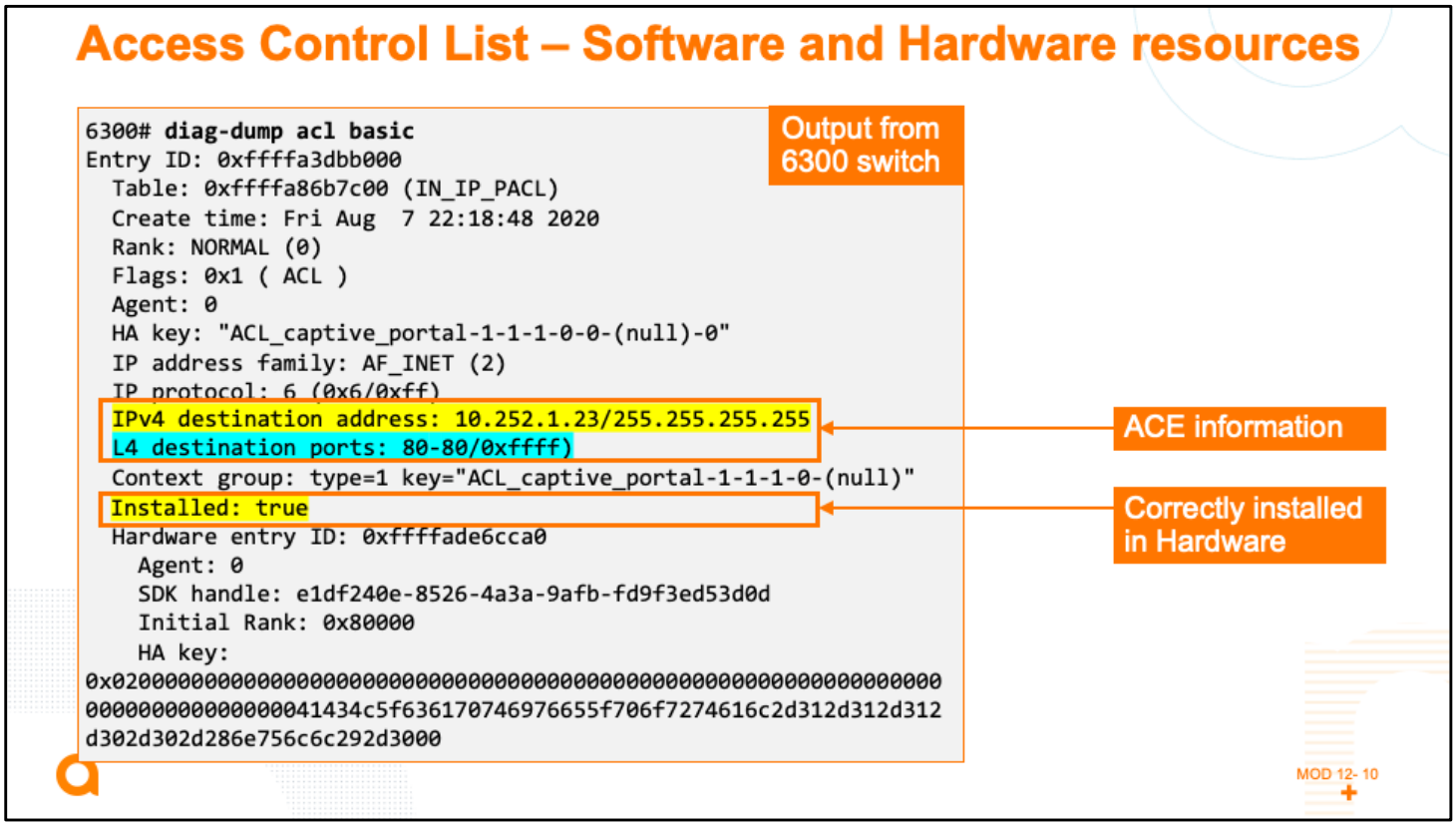
Keep in mind that it is possible to consume many hardware resource entries when using the object group commands. For example, in a typical situation, an ACE that uses object groups with three destination addresses and three destination Layer-4 ports, a total of 9 hardware entries are consumed (3 * 3= 9).

In a situation that you have an ACE that uses object groups with three destination addresses, three source addresses, three source Layer-4 ports and three destination Layer-4 ports a total of 81 hardware entries are consumed (3*3*3*3 = 81).

You can track the available hardware resources in the switch using the following commands:

The output of diag-dump acl basic command provides information about the usage of the resources in the platform consumed by ACLs.

Note. The output presented in the slide for diag-dump acl basic command was taken from an 8325 AOS-CX switch series.



The output presented in a 6300-switch series for the same command (diag-dump acl basic) is different because it presents detailed information about each instance configured in the ACL and how this instance is installed in hardware.

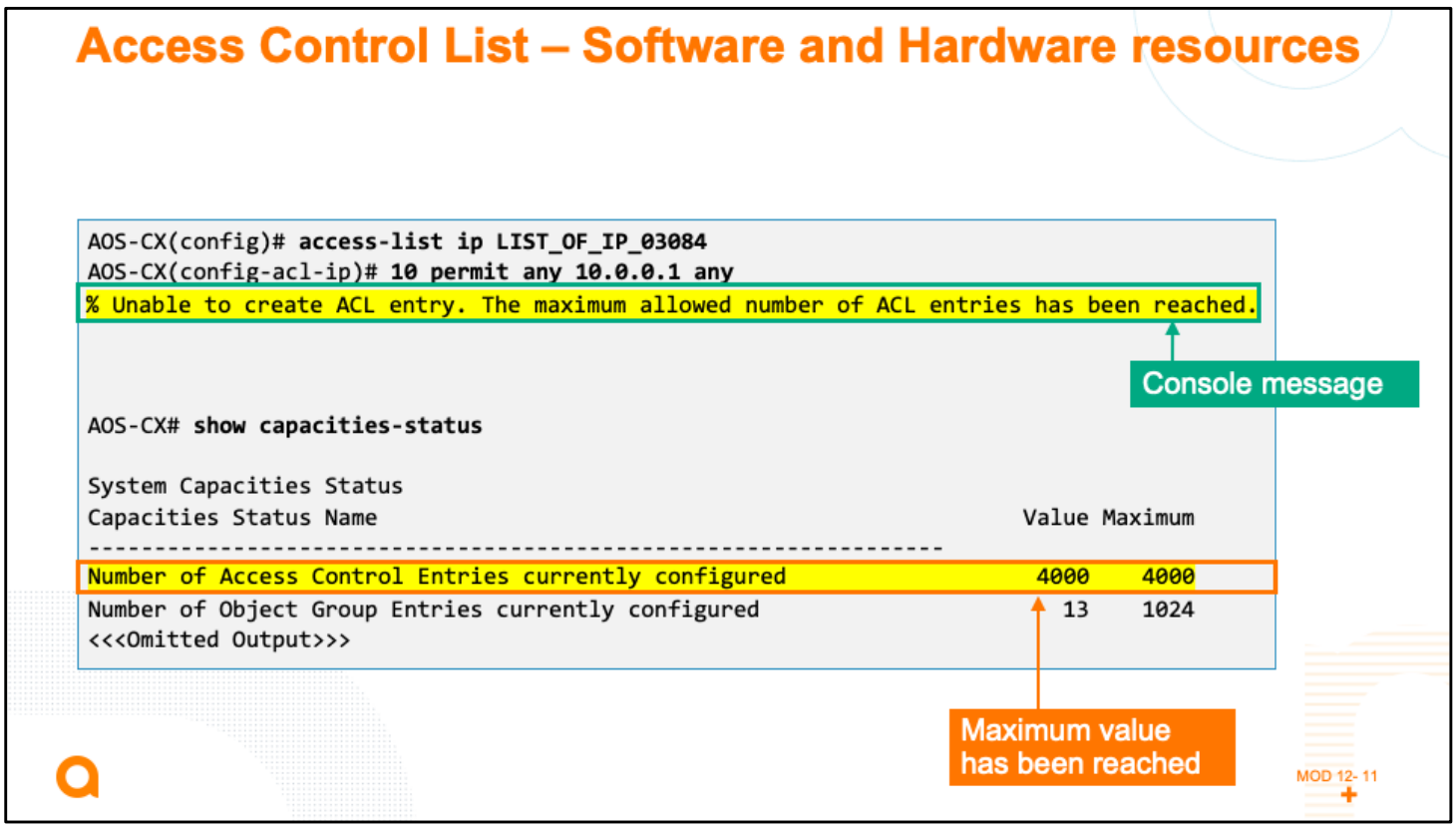
The installation of the ACE in hardware is crucial for a correct operation of the system. When the installation of the ACE is not correct the system will display a message indicating the user configuration does not match the active configuration:

AOS-CX(config)# do show access-list commands

! access-list ip TEST_ACL user configuration does not match active configuration.

! run 'access-list TYPE NAME reset' to reset access-list ip TEST_ACL access-list to match active configuration.

As the second line suggest you must use the access-list reset command to fix this issue. This command changes the user-specified ACL configuration to match the active ACL configuration. You must use this command when a discrepancy exists between what the user configured and what is active and accepted by the system.



In Module 2 - Troubleshooting, we presented an output where the system has reached the maximum value of the ACEs permitted on the system, when this situations happen AOS-CX will show a message, specifying the reason.

Use show capacities-status to determine if any processes or configuration parameters are at or near maximum values. This could prevent you from configuring desired features or entities.

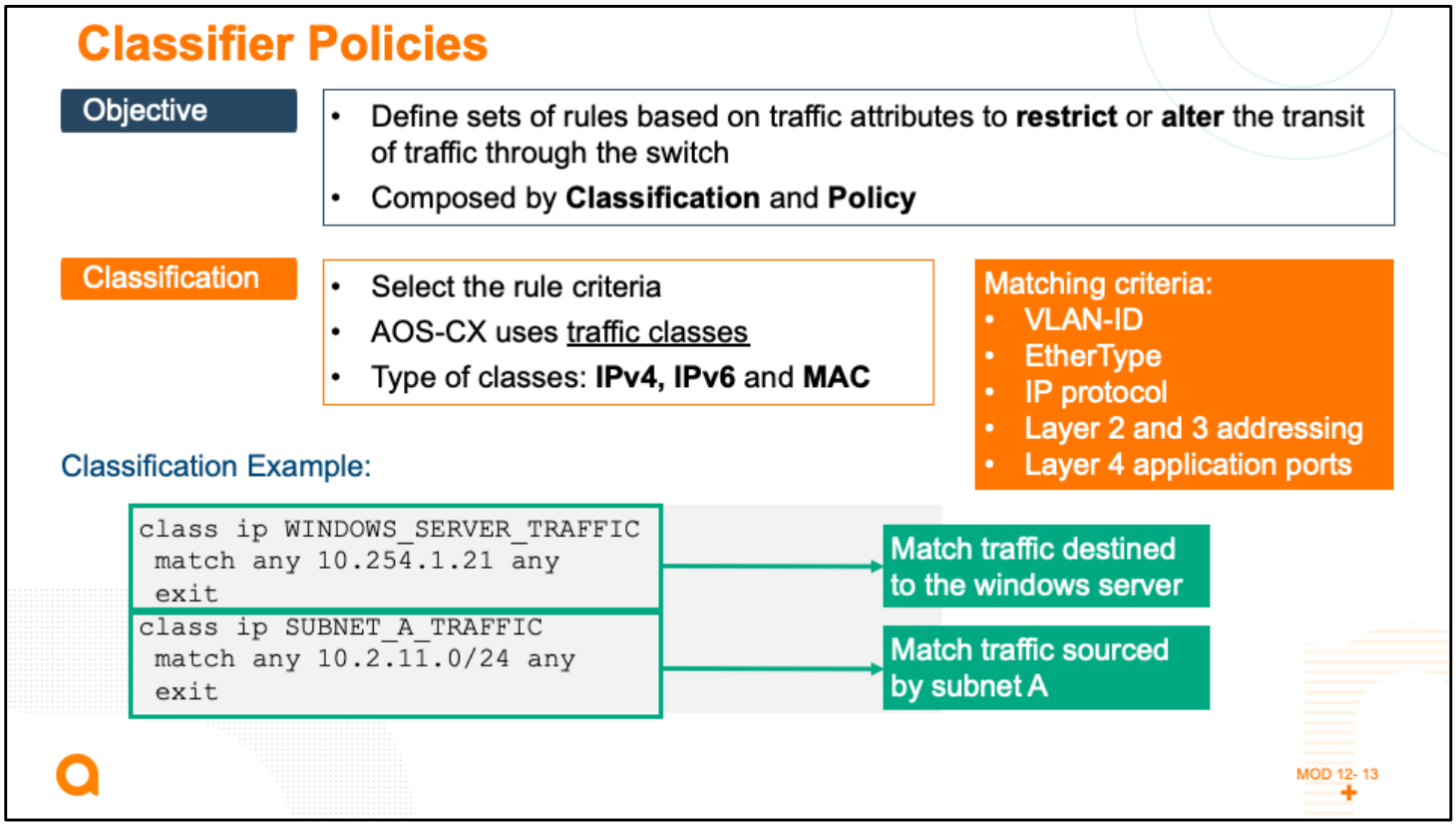
In this example the administrator wishes to create a new access-list. However, a message appears when the Access Control Entry is defined. This message indicates the system is unable to create the entry due to a capacity problem.

Often, this command may help you realize that some “clean-up” may be in order. For example, you may have VLANs or Access Control Entries that are no longer used and can be deleted to make room for more current requirements.

Classifier Policy

Configuration and validation

aruba
a Hewlett Packard Enterprise company



Classifier policies let a network administrator define sets of rules based on network traffic addressing or other header content and use these rules to restrict or alter the transit of traffic through the switch.

Choosing the rule criteria is called Classification, in AOS-CX this is achieved by creating a traffic class. The three types of classes (MAC, IPv4, and IPv6) are each focused on relevant frame/packet characteristics. Classes can be configured to match or ignore almost any frame or packet header field. Network traffic passing through a switch can be classified based on many different frame/packet characteristics including, but not limited to:

Frame ingress VLAN ID

Source and/or destination Ethernet MAC, IPv4 or IPv6 address

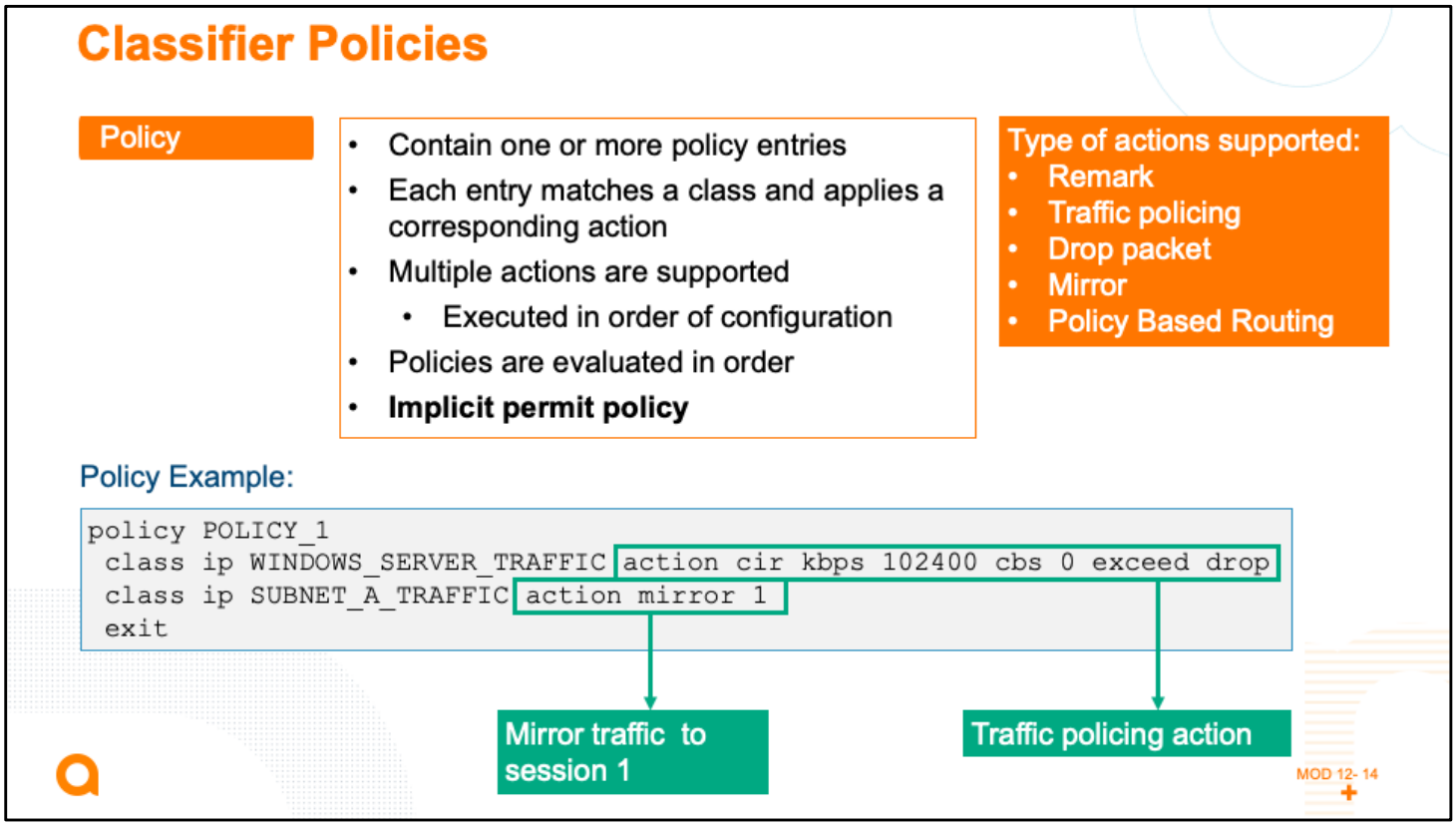
Layer 2 (EtherType) and Layer 3 (IP Protocol)

Layer 4 application ports

The example presents a couple of traffic classes created to match the traffic destined to a Windows server (10.254.1.21) and another that matches the traffic generated from the subnet A (10.2.11.0/24).

class ip WINDOWS_SERVER_TRAFFIC

```
 match any 10.254.1.21 any
 exit
class ip SUBNET_A_TRAFFIC
 match any 10.2.11.0/24 any
 exit
```



A policy contains one or more policy entries, which are listed according to priority by sequence number. A single policy entry contains a class and corresponding policy action. Policy action is taken on traffic matched by its corresponding class. A policy can be applied to an individual front plane port, a Link Aggregation Group (LAG) interface, or a VLAN.

Types of policy actions

The policy actions are broadly classified in the following categories:

Remark

Traffic policing

Drop the packet

Mirror

Policy Based routing

Each policy entry can have a combination of policy actions from these multiple categories, which are executed in order of configuration.

In the example, POLICY_1 is created. The actions of this policy are:

Limit the Bandwidth destined to the Windows Server to 102400 kbps, exceed traffic is dropped without accepting any burst size.

Mirror the traffic generated by the subnet A to session 1.

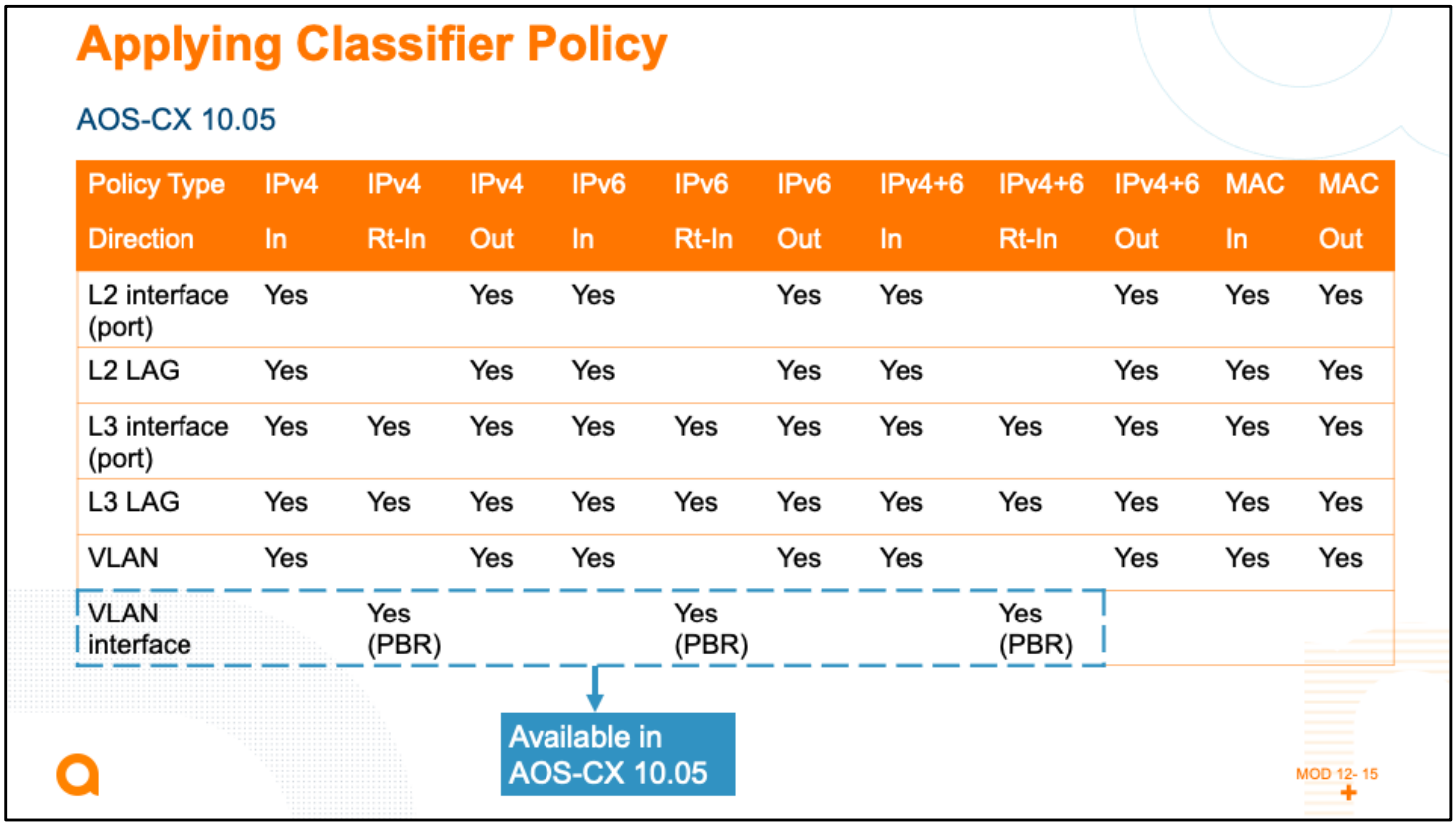
How policy matching works?

A single policy entry matches on one or more characteristics of the particular traffic type and has a configured action to continue through the switch.

This matching occurs by beginning with the entry with the lowest sequence number. The entry is then compared against the incoming frame to its particular match characteristics. If there is a match, the action is taken.

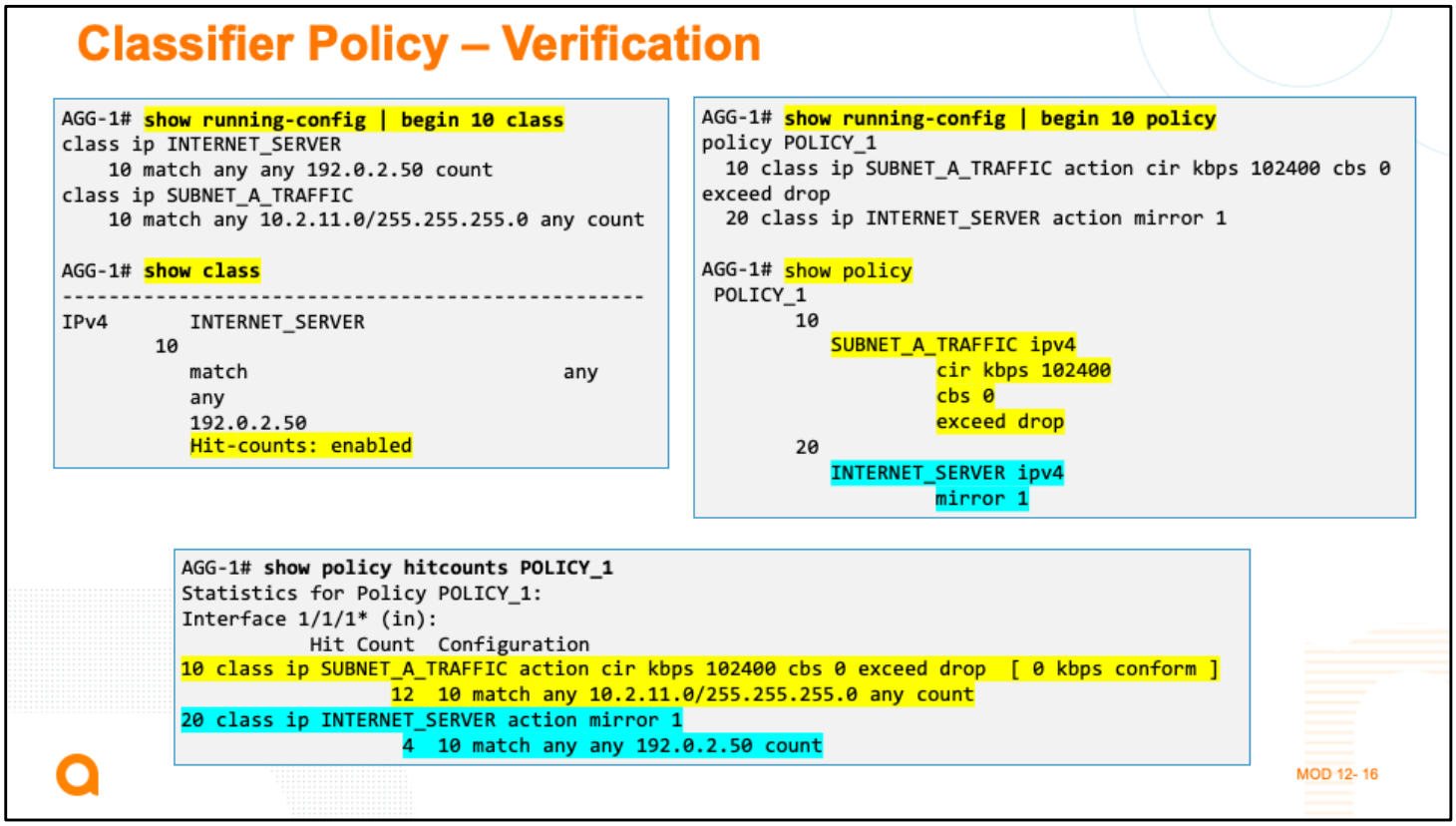
If there is no match, the match characteristics of the next sequence are compared to the relevant frame/ packet details. If there is a match, the specified actions are taken.

This process continues until a match is found. Otherwise, the packet is permitted to flow through the switch unaltered. The "implicit permit" behavior of policy matching differs from the "implicit deny" behavior of ACL matching.



Policy Type / Direction	IPv4 In	IPv4 Rt-In	IPv4 Out	IPv6 In	IPv6 Rt-In	IPv6 Out	IPv4+6 In	IPv4+6 Rt-In	IPv4+6 Out	MAC In	MAC Out
L2 interface (port)	Yes		Yes	Yes		Yes	Yes		Yes	Yes	Yes
L2 LAG	Yes		Yes	Yes		Yes	Yes		Yes	Yes	Yes
L3 interface (port)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
L3 LAG	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VLAN	Yes		Yes	Yes		Yes	Yes		Yes	Yes	Yes
VLAN interface		Yes (PBR)			Yes (PBR)			Yes (PBR)			

A policy must be applied to an interface to take effect, the policy can affect the traffic in inbound and outbound direction. The table presented displays the possible options to apply a policy in AOS-CX 10.05.



show class and show policy commands can be used to verify the configuration of the classifier policy. The show policy hitcounts command is useful to verify the number of hits on a particular policy.

To verify the hardware consumption of the policy you can use diag-dump policy basic command:

8325# diag-dump policy basic

==========

==

[Start] Feature policy Time : Tue Sep 15 03:00:29 2020

==========

==

[Start] Daemon ops-switchd

Entries

Slot Description Width Used Reserved Free

```
  0 Ingress_IPv4_Port_ACL                3      30     2304
  0 Ingress_MAC+IPv4_Port_Policy         3       6     2304
  0 Ingress_Control_Plane_Policing       3     246     2304
  0 Total Ingress TCAM Entries         282    6912   2304

  0 Egress_Control_Plane_Policing        2      84     1024
  0 Total Egress TCAM Entries           84    1024    1024
```

* This feature shares reserved resources with preceding feature

```
----------------------
Policers
----------------------
Slot    Used    Free
----------------------
  0       1    6143

----------------------
L4 Port Range Checkers
----------------------
Slot    Used    Free
----------------------
  0       0      32
```

The same command issued from the 6300-switch series has a different output, which includes detailed information about each entry created in the hardware:

```
6300# diag-dump policy basic
================================================================
==============
[Start] Feature policy Time : Tue Sep 15 03:03:16 2020
================================================================
```

```
==============
------------------------------------------------------------------------
[Start] Daemon ops-switchd
------------------------------------------------------------------------
Entry ID: 0xffffb19c1000
  Table: 0xffffb52b9800 (POLPI)
  Create time: Tue Sep 15 03:03:06 2020
  Rank: NORMAL (0)
  Flags: 0x8 ( POLICY )
  Agent: 0
  HA key: "POLICY_1-4-1-1-0-0-(null)-0"
  IP address family: AF_INET (2)
  IPv4 source address: 10.254.1.21/255.255.255.255
  Context group: type=4 key="POLICY_1-4-1-1-0-(null)"
  Policer: ID=0xffffbaa408b0
  Installed: true
  Hardware entry ID: 0xffffbaa69780
    Agent: 0
    SDK handle: 8f83ebb0-e86e-4490-9c40-c0af0b3a7a45
    Initial Rank: 0x80000
    HA key:
0x0400000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000504f4c4943595f312d342d312
d312d302d302d286e756c6c292d3000
```

Question #1

According to the output shown, how many packets destined to 10.252.1.23 has the switch detected?

a. 9 in total

b. 18 in total, 9 on interface 1/1/1 and 9 on interface 1/1/2

Counter is not possible on per interface basis

```
# show running-config
class ip CPPM
    10 match any any 10.254.1.23 count
policy counter1
    10 class ip CPPM action drop
!
interface 1/1/1
   apply policy counter1
interface 1/1/2
   apply policy counter1

# show policy hitcounts counter1
Statistics for Policy counter1:
Interface 1/1/1* (in):
           Hit Count  Configuration
10 class ip cppm
                   9  10 match any any 10.252.1.23 count
Interface 1/1/2* (in):
           Hit Count  Configuration
10 class ip cppm
                   9  10 match any any 10.252.1.23 count
```

Knowledge Check MOD 12- 17

Correct answer:

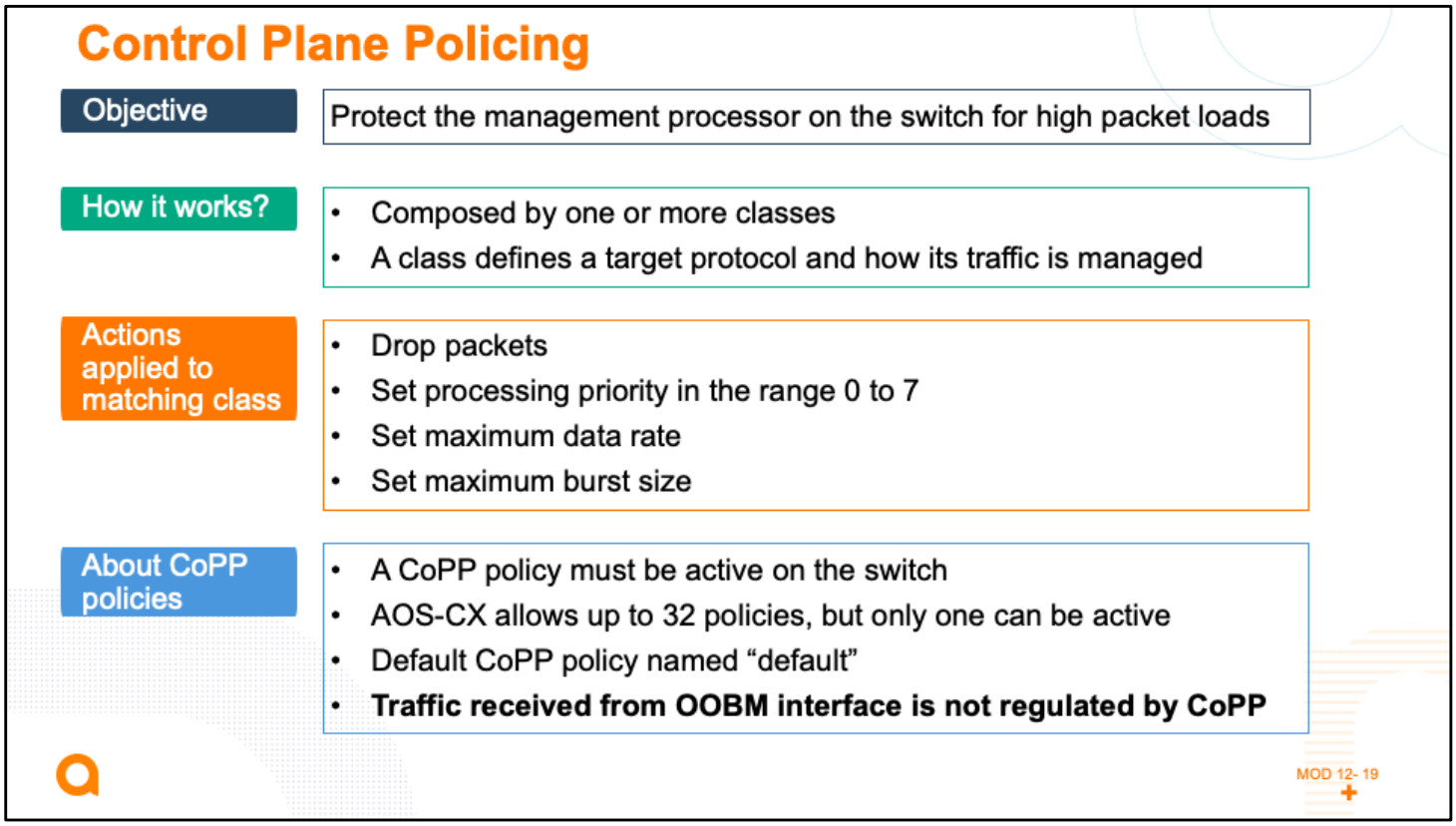
A

The output of the show policy hitcounts command displays the total number of hitcounts detected on all interfaces where the policy has been applied. In this case there are nine matches that the switch has detected but there is no visibility on which interface they were received. In other words, counting is not possible per interface basis.

Control Plane Policing

Configuration and validation

aruba
a Hewlett Packard Enterprise company



Control Plane Policing (CoPP) provides a way for administrators to protect the management processor on the switch from high packet loads (generated by malicious or non-malicious sources) that might interfere with its ability to keep data plane traffic flowing. For example, a denial-of-service attack can result in excessive traffic that would slow down the management processor and negatively affect switch throughput.

A CoPP policy is composed of one or more classes. Each class defines a target protocol and how its traffic is managed. Every policy also has a default class to regulate packets that do not match any other class. The following actions can be applied for all packets matching a class:

Drop the packets. (Excluding the default class.)

Set the processing priority in the range 0 to 7.

Set the maximum data rate in packets per second (pps) at which each line module can send packets to the management processor.

Set the maximum burst size in packets at which each line module can send packets to the management processor.

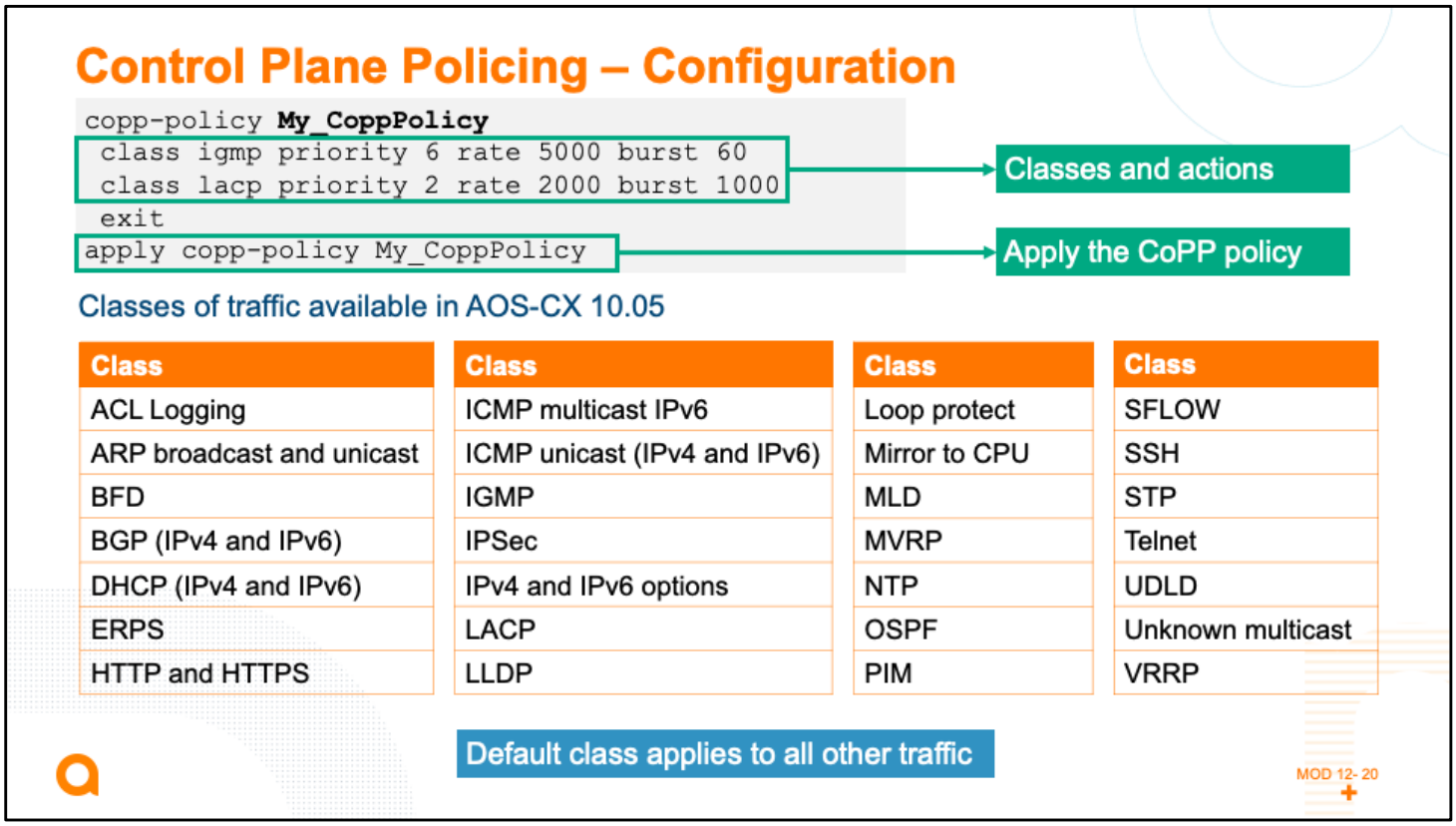
Up to 32 CoPP policies can be defined, but only one can be active on the switch at a time.

A CoPP policy must always be active on the switch. By default, the switch has a CoPP policy named default which is automatically applied at first boot.

When the switch is rebooted, the CoPP policy that was actively applied to the switch before the reboot occurred will be applied, if it was saved to the startup configuration with the copy running-config startup-config command.

For GRE tunneled traffic, CoPP policies match on the payload.

CoPP policies do not regulate traffic received from the Out-of-Band-Management (OOBM) Ethernet port.



To configure a Control Plane Policy follow the steps:

Configure the default CoPP policy, edit an existing policy, or create a policy with the command copp- policy.

Add, edit, or remove classes in the policy with the command class.

If the policy is not the active policy on the switch, apply it with the command apply copp-policy. (Changes made to an active policy take effect immediately and do not need to be applied.)

Review the CoPP policy configuration settings with the command show copp-policy.

The different classes of traffic that can be individually configured are:

acl-logging: Access Control List logging packets.

arp-broadcast: Address Resolution Protocol packets with a broadcast destination MAC address.

arp-unicast: Address Resolution Protocol packets with a switch system destination MAC address.

bfd: Bidirectional Forwarding Detection (BFD) packets with a destination IP address owned by the switch.

bgp-ipv4: Border Gateway Protocol packets with a destination IPv4 address owned by the

switch.

bgp-ipv6: Border Gateway Protocol packets with a destination IPv6 address owned by the switch.

dhcp-ipv4: Dynamic Host Configuration Protocol packets with a destination IPv4 address.

dhcp-ipv6: Dynamic Host Configuration Protocol packets with a destination IPv6 address.

erps: Ethernet Ring Protection Switching control packets with the destination MAC address 01:19:a7:00:00:XX, where XX can be any value.

hypertext: Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) packets. icmp-broadcast-ipv4: Internet Control Message Protocol packets with a broadcast or multicast destination IPv4 address.

icmp-multicast-ipv6: Internet Control Message Protocol packets with a well-known multicast destination IPv6 address.

icmp-unicast-ipv4: Internet Control Message Protocol packets with a destination IPv4 address owned by the switch.

icmp-unicast-ipv6: Internet Control Message Protocol packets with a destination IPv6 address owned by the switch.

igmp: Internet Group Management Protocol packets.

ip-exceptions: Routable packets that would exceed the MTU for the egress interface, packets that trigger ICMP redirects, and packets with TTL/hop_limit=1 that are discarded when routing through the switch.

ipsec: Internet Protocol Security IPv4 or IPv6, unicast or configured multicast. All IPsec traffic received by the CPU will be regulated by the ipsec class regardless of the encapsulated protocol.

ipv4-options: Unicast IPv4 packets including option headers.

ipv6-options: Unicast IPv6 packets including option headers.

lacp: Link Aggregation Control Protocol packets with the destination MAC address 01:80:c2:00:00:02.

lldp: Link Layer Discovery Protocol packets with the destination MAC address 01:80:c2:00:00:0e.

loop-protect: Loop Protection packets with the destination MAC address 09:00:09:09:13:a6.

mirror-to-cpu: Packets from mirroring session configured to deliver to the console.

mld: Multicast Listener Discovery packets of type V1 or V2 with an IPv6

address of FF00::/8, FF02::16 or FF02::2.

mvrp: Multiple VLAN Registration Protocol packets with the destination MAC address 01:80:c2:00:00:20 or 01:80:c2:00:00:21.

ntp: Network Time Protocol packets with a destination IP address owned by the switch.

ospf-multicast-ipv4: Open Shortest Path First packets with the multicast destination IPv4 address 224.0.0.5 or 224.0.0.6. The 8325 switch has VxLAN packets with VNI 0.

ospf-multicast-ipv6: Open Shortest Path First packets with the multicast destination IPv6 address FF02::5 or FF02::6.

ospf-unicast-ipv4: Open Shortest Path First packets with a destination IPv4 address owned by the switch.

ospf-unicast-ipv6: Open Shortest Path First packets with a destination IPv6 address owned by the switch.

pim: Protocol Independent Multicast packets with the destination IPv4 address 224.0.0.13 or IPv6 address FF02::D, or with a destination IP address owned by the switch.

sflow: Packet headers sampled by the switch that will be sent to the sFlow collector.

ssh: Secure Shell (SSH) or Secure File Transfer Protocol (SFTP) packets. Dropping ssh packets will result in the connection to the CLI being lost.

stp: Spanning Tree Protocol (STP) packets with the destination MAC address 01:80:c2:00:00:00 or Per- VLAN Spanning Tree (PVST) packets with the destination MAC address 01:00:0c:cc:cc:cd.

telnet: Secure Telnet packets.

udld: Unidirectional Link Detection packets with the destination MAC address 01:00:0c:cc:cc:cc or 00:e0:52:00:00:00, or Cisco Discovery Protocol packets with the destination MAC address 01:00:0c:cc:cc:cc.

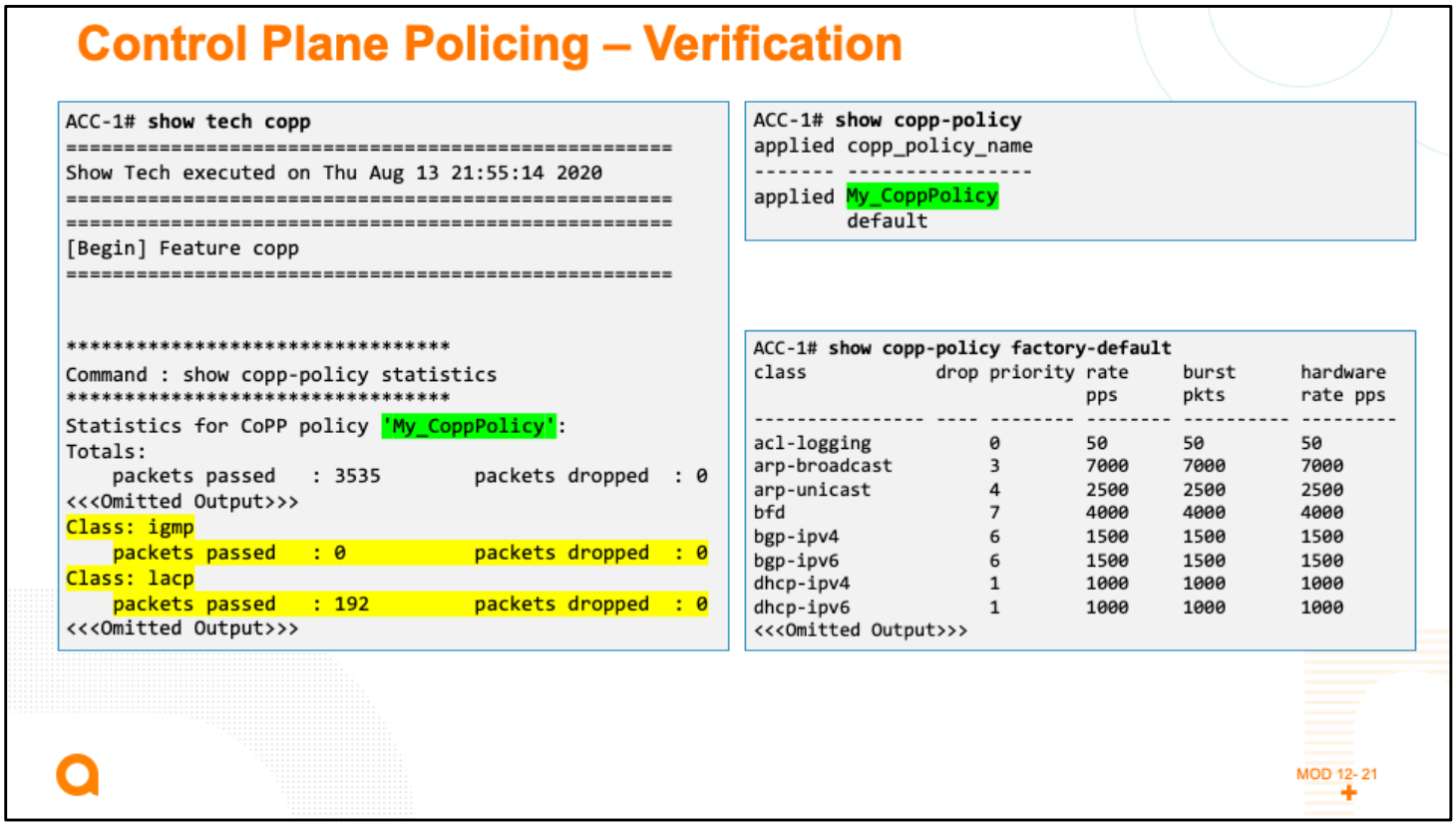
unknown-multicast: Packets with an unknown multicast destination IP address.

unresolved-ip-unicast: Packets to be software forwarded by the management processor.

vrrp-ipv4: Virtual Router Redundancy Protocol packets with the destination IPv4 address 224.0.0.18 or VSX-Keepalive.

vrrp-ipv6: Virtual Router Redundancy Protocol packets with the destination IPv6 address FF02::12.

Note. In AOS-CX there is a default class, which applies to all other traffic. Traffic generated by the IP SLA feature for instance, will be regulated by this default class, so limiting the default class may affect the IP SLA operation.



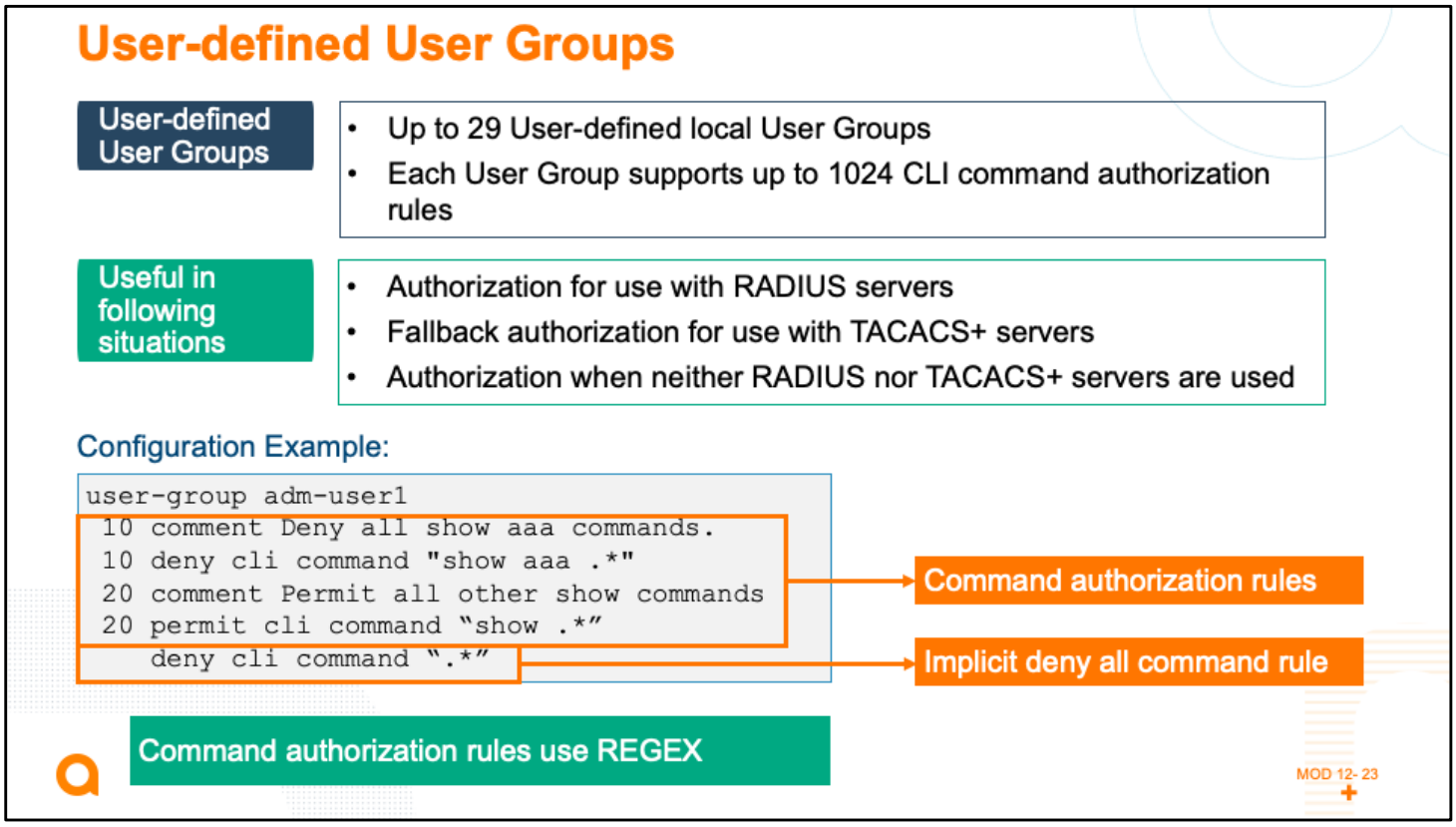
You can verify the Control Plane Policing using the following commands:

show tech copp
show copp-policy statistics
show copp-policy factory-default
show copp-policy

Local user and group management

Configuration and validation

aruba
a Hewlett Packard Enterprise company



The switch enables you to create up to 29 user-defined local user groups for configuring local authorization. Each of the 29 user-defined groups support up to 1024 CLI command authorization rules that define what CLI commands can be executed by members of the group.

The local user groups with their command execution rules are useful for the following:

Providing authorization for use with RADIUS servers.
Providing fallback authorization for use with TACACS+ servers
Providing authorization when neither RADIUS nor TACACS+ servers are used.

Configuration

The configuration of a User Group is comparable to an Access Control List. The administrator must define the command authorization rules including the permit or deny action for the defined command. The User Group includes an implicit “deny .*” rule, this means that members of a user-defined group without any permit rules have no CLI command privileges.

The rule evaluation proceeds from lowest to highest sequence number until the first successful match. Therefore, rules for related CLI commands must be defined in most

restrictive to least restrictive order.

In the example you can see how the User Group created denies all show aaa commands while allowing the rest of the show commands.

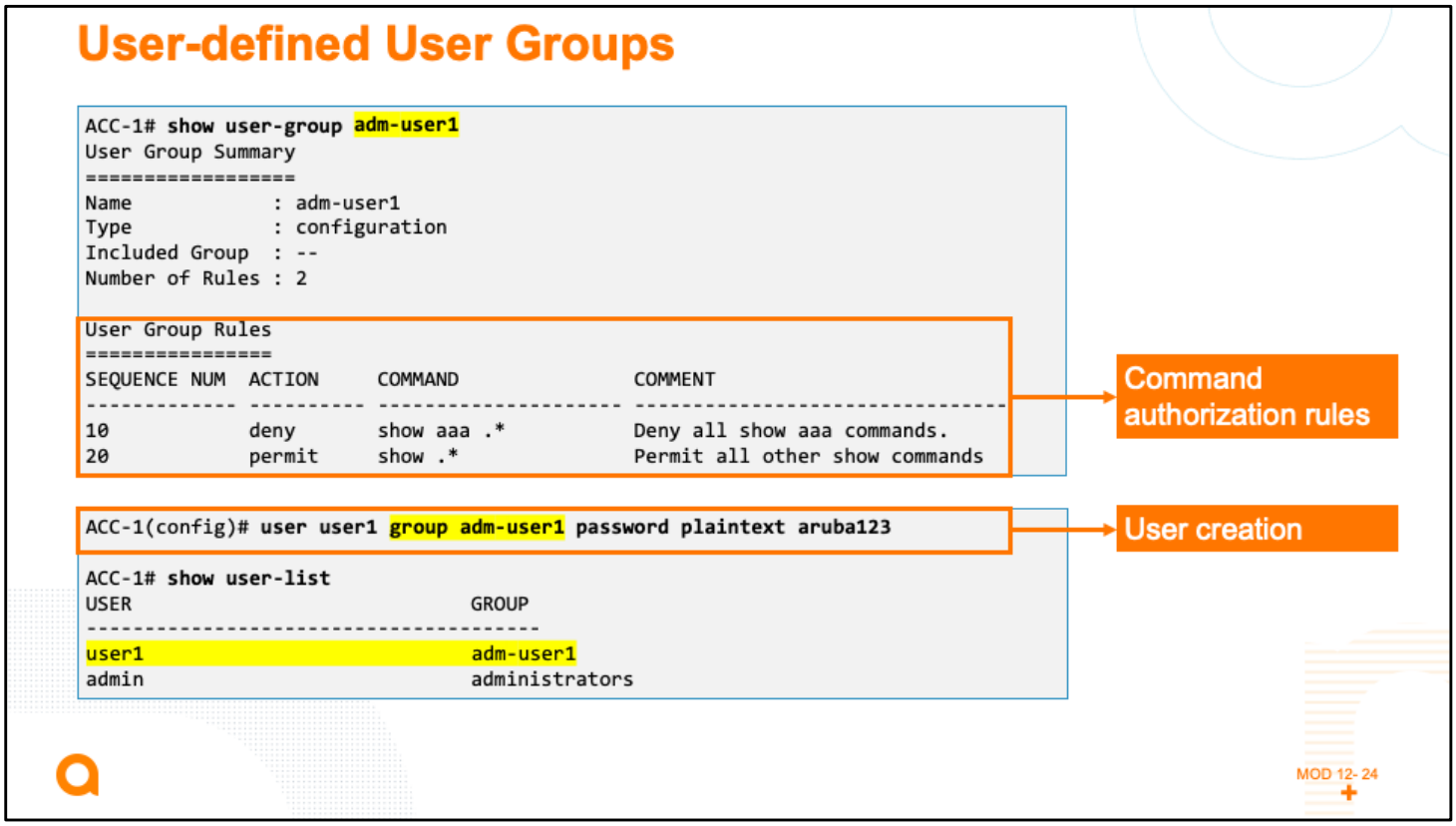
The definition of the rule accepts Regular Expression (Regex) strings.

Note. The implicit deny command is shown in the slide for academy purposes.

Note. To learn more about regular expressions you can consult these external references:

https://regexr.com/

https://en.wikipedia.org/wiki/Regular_expression



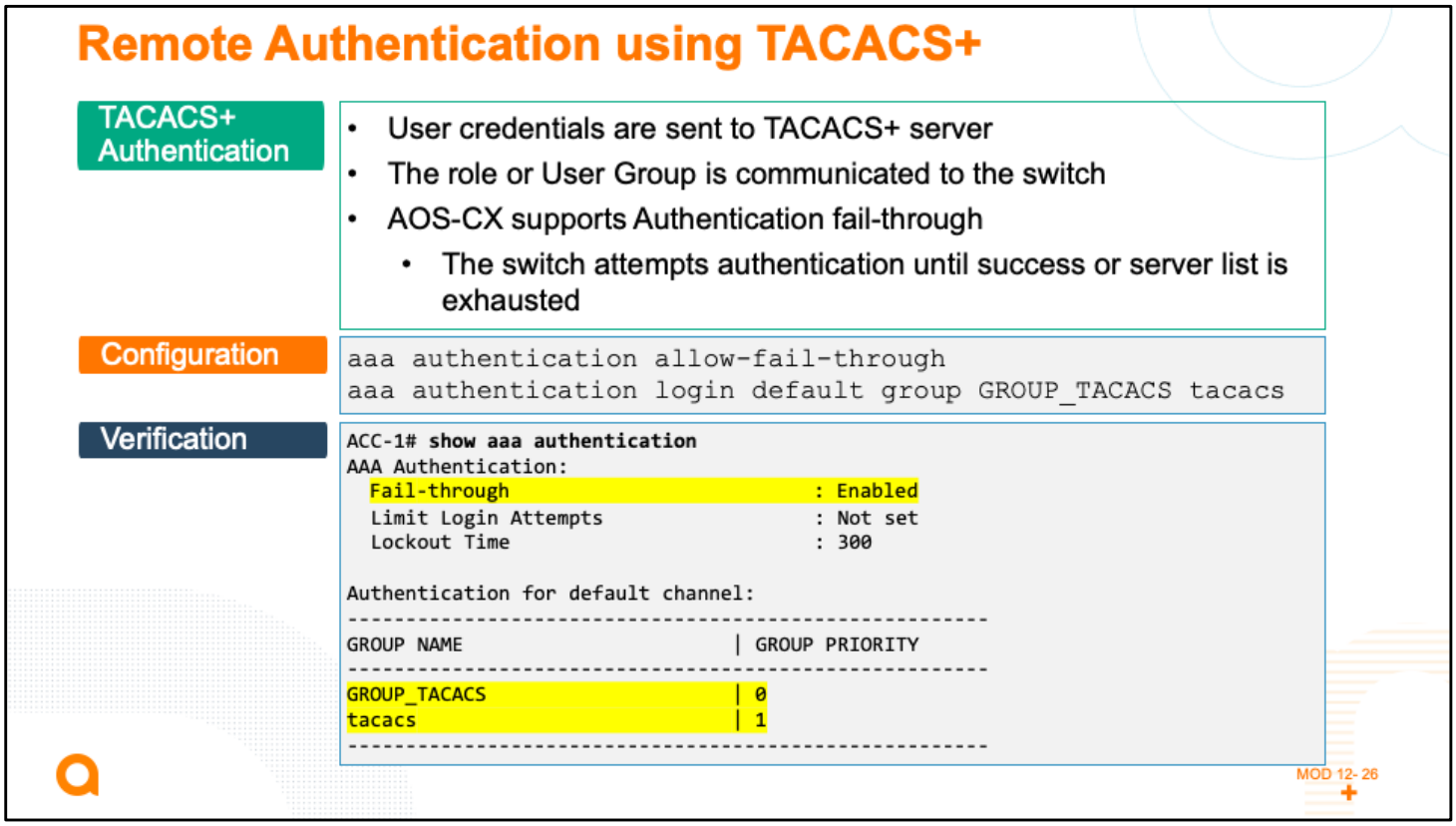
The show user-group command displays the command authorization rules defined.

The show user-list command list the local users and the Group that is used.

Management access using TACACS+

Configuration and validation

aruba
a Hewlett Packard Enterprise company



TACACS+ authentication

TACACS+ authentication occurs as follows:

User credentials are sent from the switch to TACACS+ server using the PAP or CHAP authentication protocol.

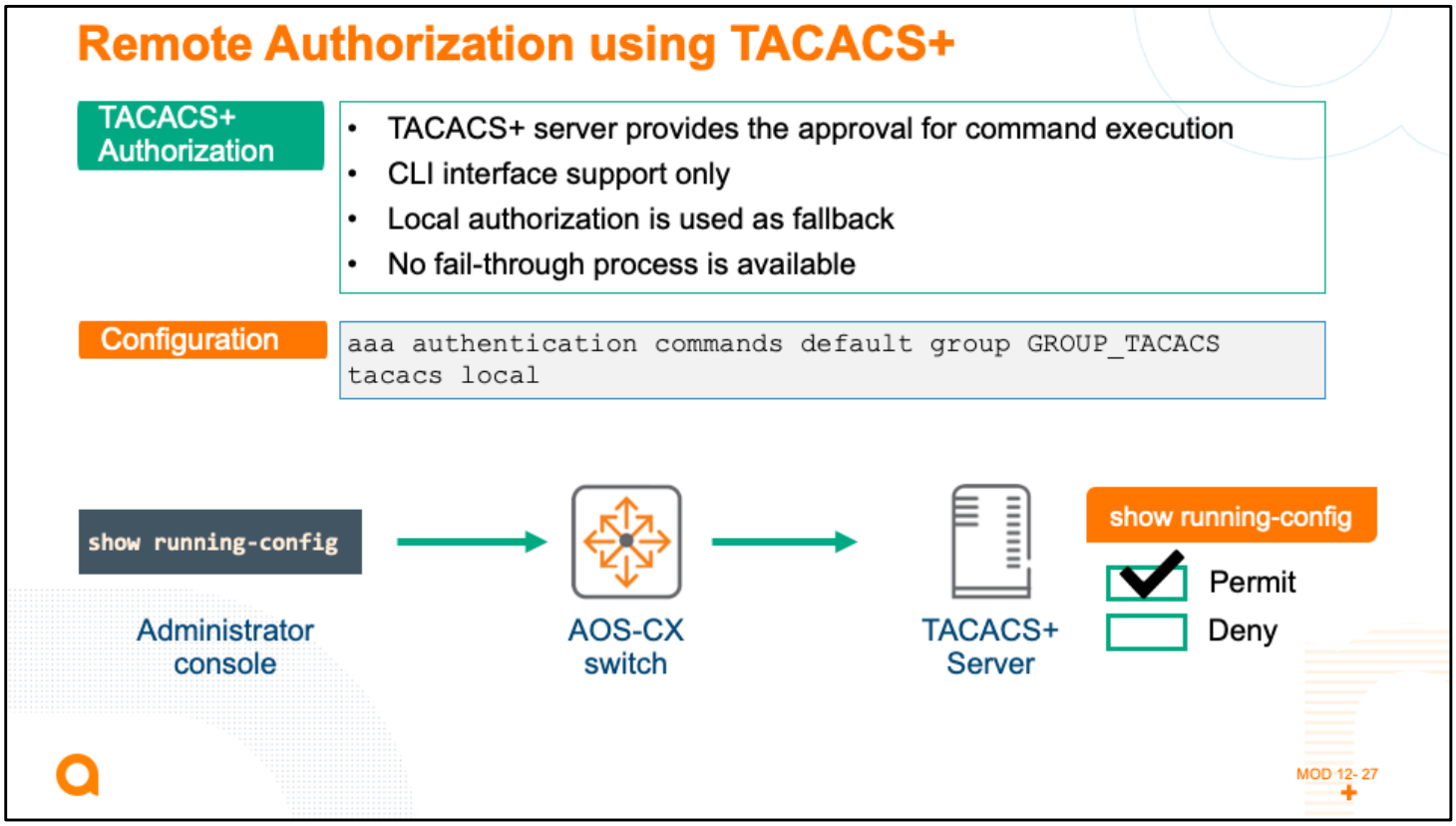
If a user is authenticated, their role is communicated to the switch as Administrator, Operator, or Auditor.

An unknown user or a user who entered an invalid password is identified as such to the switch, which then rejects user login.

Authentication fail-through

Normally, authentication is performed by the first AAA server reached. A rarely needed feature named "Authentication fail-through" is available. If Authentication fail-through is enabled and authentication fails on the first reachable AAA server, authentication is attempted on the second AAA server, and so on, until successful authentication or the server list is exhausted.

Enabling Authentication fail-through is typically unnecessary because the user credential databases should be consistent across all AAA servers. Authentication fail-through might be helpful if your AAA user credential databases are not quickly synchronized across all AAA servers.



TACACS+ authorization

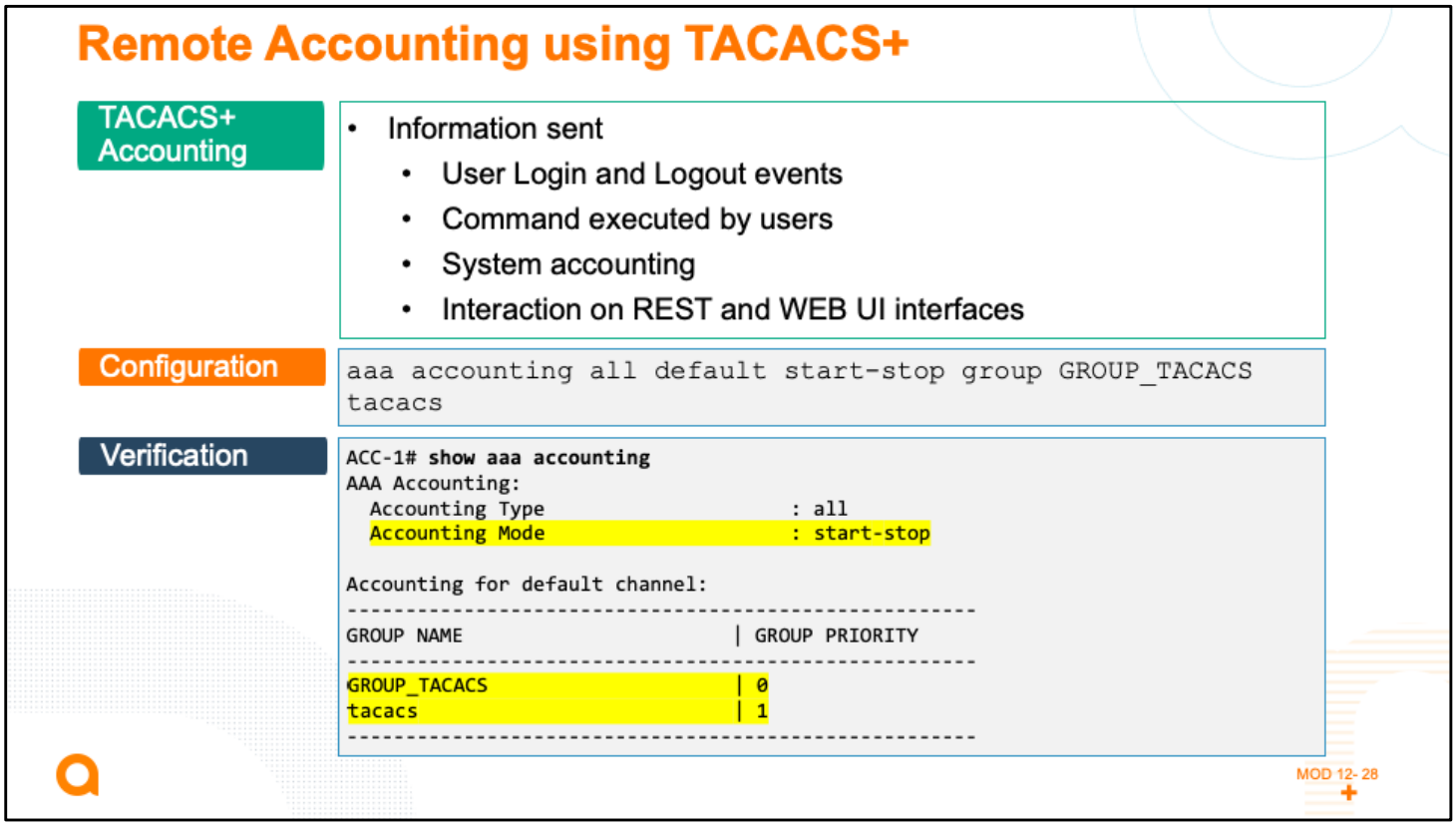
Upon successful user authentication, the user is assigned their role by the TACACS+ server. TACACS+ authorization provides command filtering to allow/disallow individual command or command set execution. Each command is sent to the TACACS+ server for approval, and the switch then allows/disallows command execution according to the server response.

Using local authorization as fallback from TACACS+ authorization

Local authorization can be used for the situation in which communication is lost with all TACACS+ servers after a successful authentication. Users that are members of the built-in local user groups (administrators, operators, or auditors) are authorized according to the fixed roles and privilege levels of those groups. Optionally, local user-defined user groups can be configured with specific command execution rules per group. Users that are members of such groups, are authorized according to the command execution rules of the group to which they belong.

For authorization, there is no equivalent of the authentication fail-through feature. Therefore, if the first reachable TACACS+ server responds with "Authorization Denied," no additional TACACS+ servers are interrogated.

Note: Rare potential out-of-synchronization situation could occur when using authentication fail-through. Successful authentication on one server can be followed by authorization denial on another. The user is known on the server doing the authentication but unknown on the server attempting the authorization. This situation typically arises only during brief periods in which user credential databases are not synchronized across all TACACS+ servers.



TACACS+ accounting

This accounting information is captured and made available for sending to remote accounting servers:

- Exec Accounting: user login/logout events.
- Command accounting: commands executed by users.
- System accounting: remote accounting On/Off events.
- CLI show commands.
- Interactions on the non-CLI interfaces: REST and WEB UI

The following is not captured or made available as accounting information:

CLI command s that reboot the switch.
Interactions in the bash shell

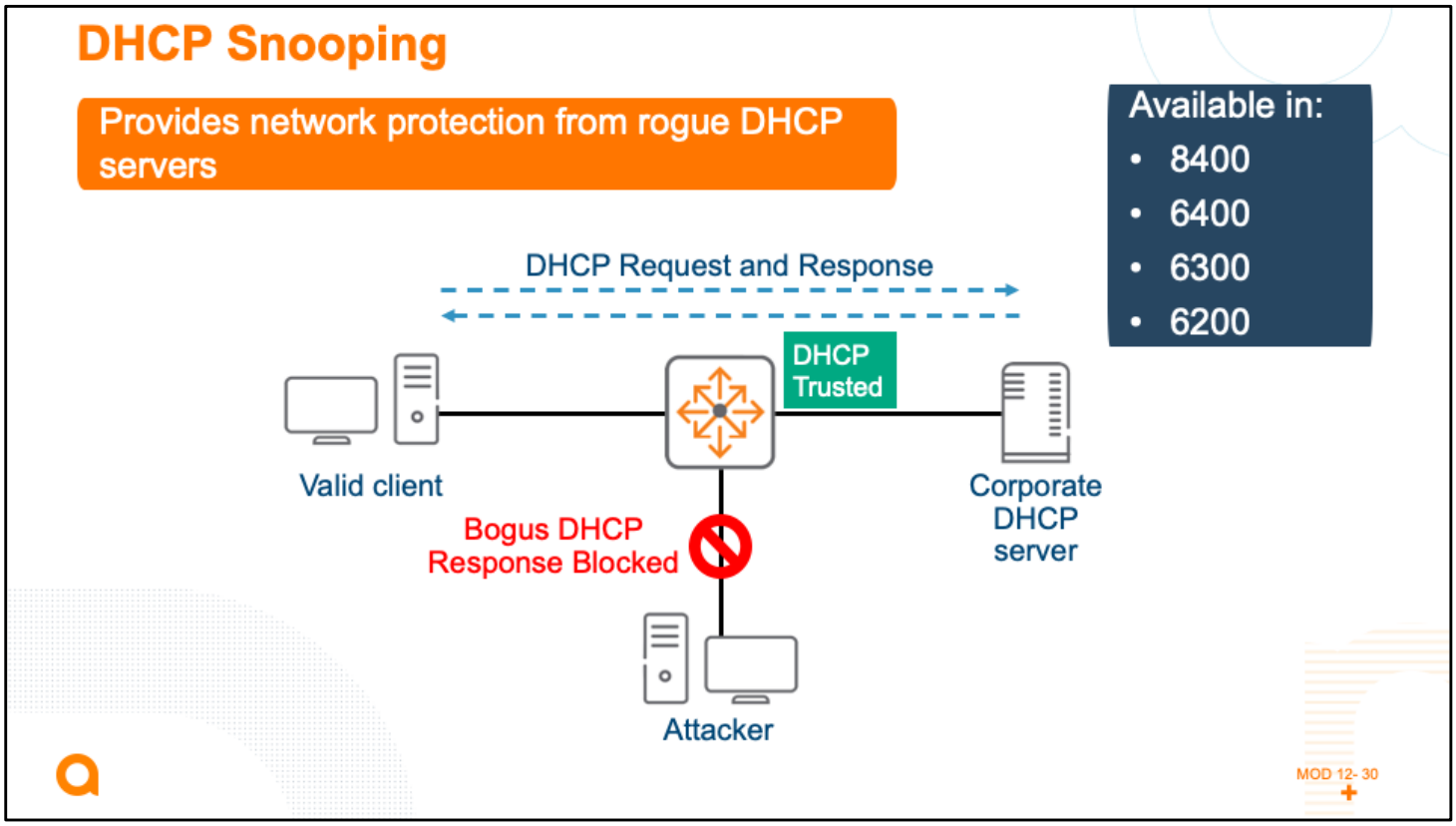
Note. Local accounting (always enabled) must be functioning properly for remote Accounting to work.

Note. The accounting information is sent to the first remote TACACS+ AAA server (configured for remote accounting). If no remote TACACS+ server is reachable, local accounting remains available.

DHCP Snooping and ARP Inspection

Configuration and validation

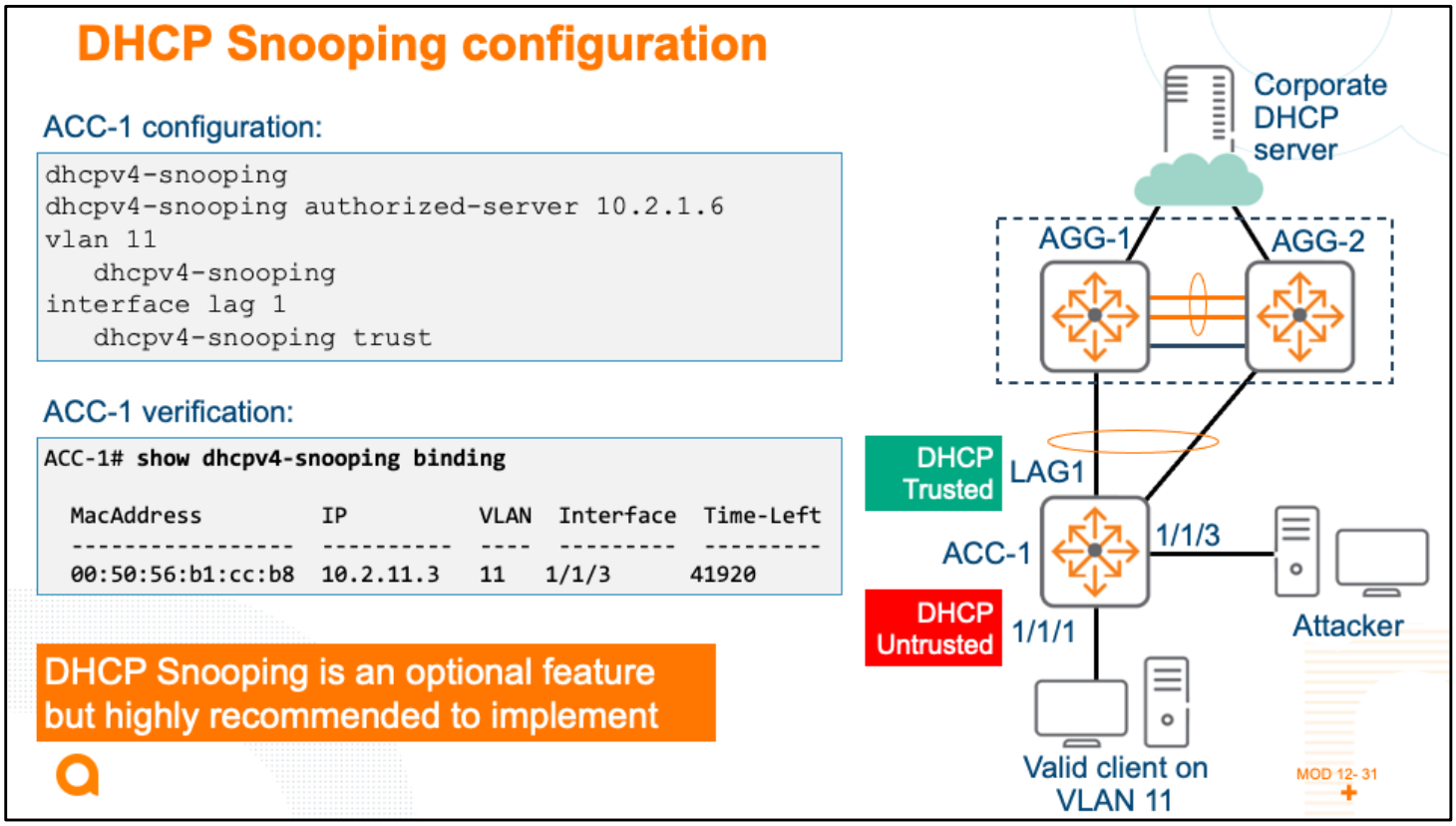
aruba
a Hewlett Packard Enterprise company



The DHCP Snooping feature provides network protection from rogue DHCP servers. It creates a logical firewall between untrusted hosts and DHCP servers. The switch builds and maintains a DHCP snooping table (also called DHCP binding database). In addition, the switch uses this table to identify and filter untrusted messages from the network. The switch maintains a DHCP binding database that keeps track of DHCP addresses that are assigned to ports, as well as filtering DHCP messages from untrusted ports. For incoming packets received on untrusted ports, packets are dropped if the source MAC address does not match MAC in the binding table entry.

The slide illustrates the DHCP Snooping feature in action, showing how the intruder is blocked on the untrusted port when it tries to intervene by injecting a bogus DHCP response packet to a legitimate conversation between the DHCP client and server.

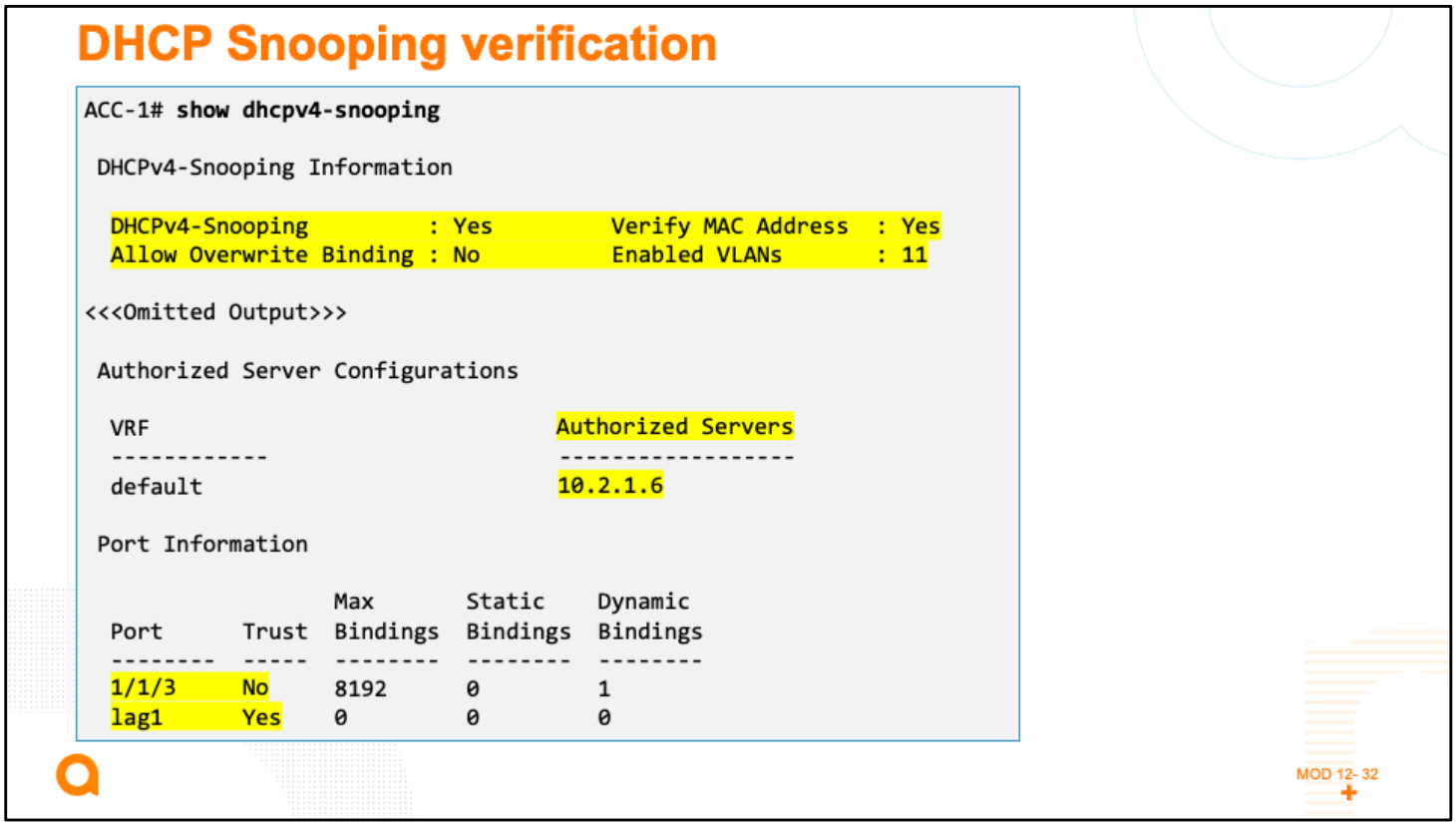
In AOS-CX the DHCP Snooping feature is available in 6200, 6300, 6400 and 8400 switch series.



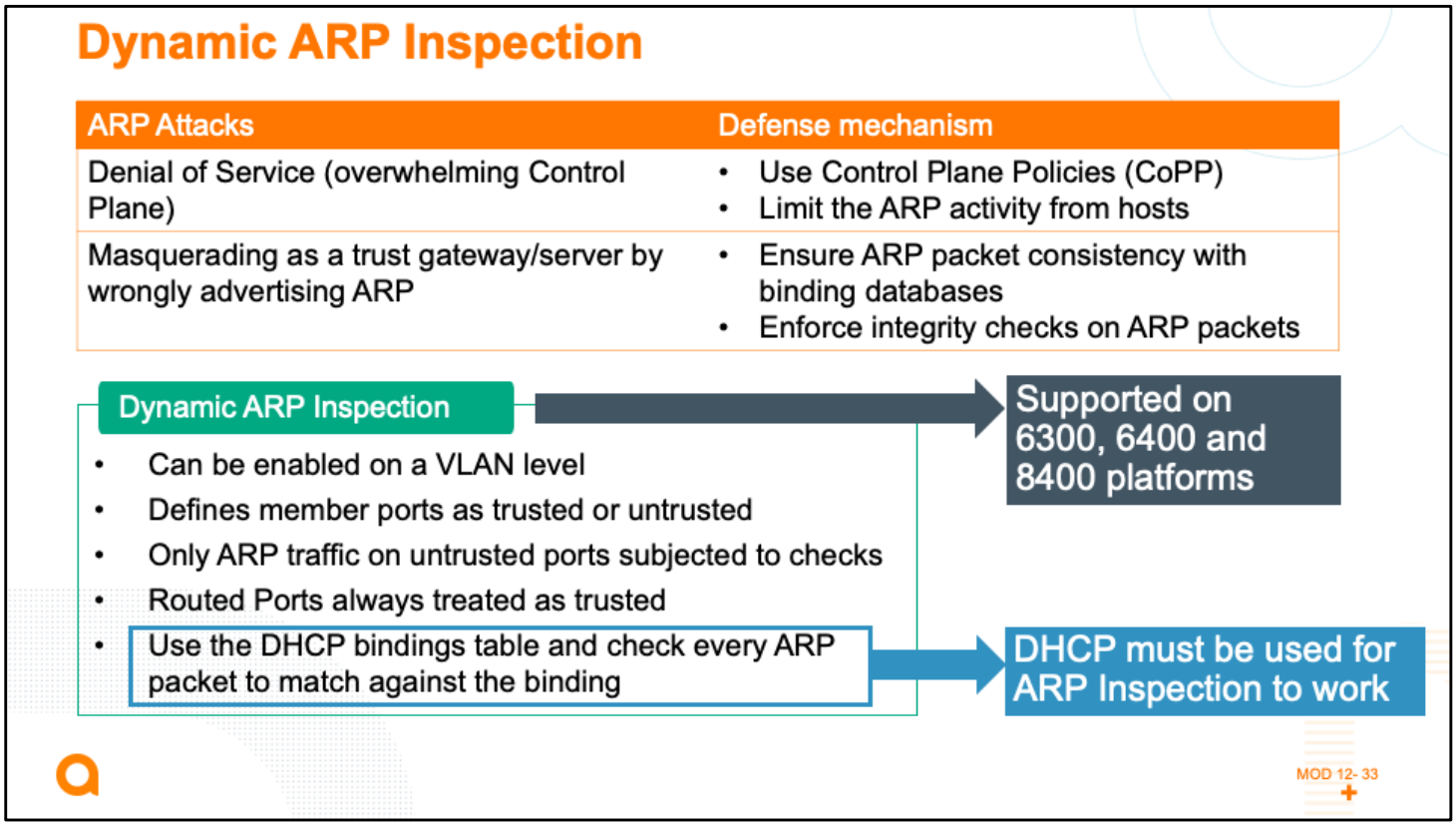
The first step is to globaly enable DHCP snooping feature using the dhcpv4-snooping and dhcpv6-snooping commands. In AOS-CX you must specify the Authorized servers and enable the feature in the VLAN level.

For DHCP Snooping to function correctly, all DHCP servers connected to the switch must be configured as trusted interfaces. A trusted interface can be configured by using the dhcpv4-snooping trust or dhcpv6-snooping trust interface configuration commands. All other DHCP clients connected to the switch are considered untrusted. In AOS-CX all ports are considered untrusted by default.

You can use the show dhcpv4-snooping binding and show dhcpv6-snooping binding commands to verify the statistics.



You can use the show dhcpv4-snooping and show dhcpv6-snooping command to verify the state.



ARP resolves MAC addresses from IP addresses on broadcast networks like Ethernet. It was originally defined by Internet Standard RFC 826. ARP does not support any inherent security mechanism and as such depends on simple datagram exchanges for the resolution, with many of these being broadcast.

Because it is non-secure protocol, ARP is vulnerable to attacks. Some attacks may be targeted toward the networks whereas other attacks may be targeted toward the switch itself. The attacks primarily intend to create denial of service (DoS) for the other entities present in the network.

Most of the attacks are carried out in one of the following three forms:

Overwhelming the switch control plane with too many ARP packets.

Overwhelming the switch control plane with too many unresolved data packets.

Masquerading as a trusted gateway/server by wrongly advertising ARPs.

Several defense mechanisms can be put in place on a switch to protect against attacks:

Protect the control plane by using CoPP policies.

Limit the amount of ARP activity allowed from a host or on a port.

Ensure that all ARP packets are consistent with one or more binding databases.

Enforce integrity checks on the ARP packets to check against different MAC or IP addresses in the Ethernet or IP header and ARP header.

AOS-CX 10.05 implements Dynamic ARP Inspection to enforce DHCP snooping binding on all ARP packets and is supported on the 6300, 6400, and 8400 platforms.

Only the following is supported:

Enabling and disabling of Dynamic ARP Inspection on a VLAN level (it does not have to be SVI).

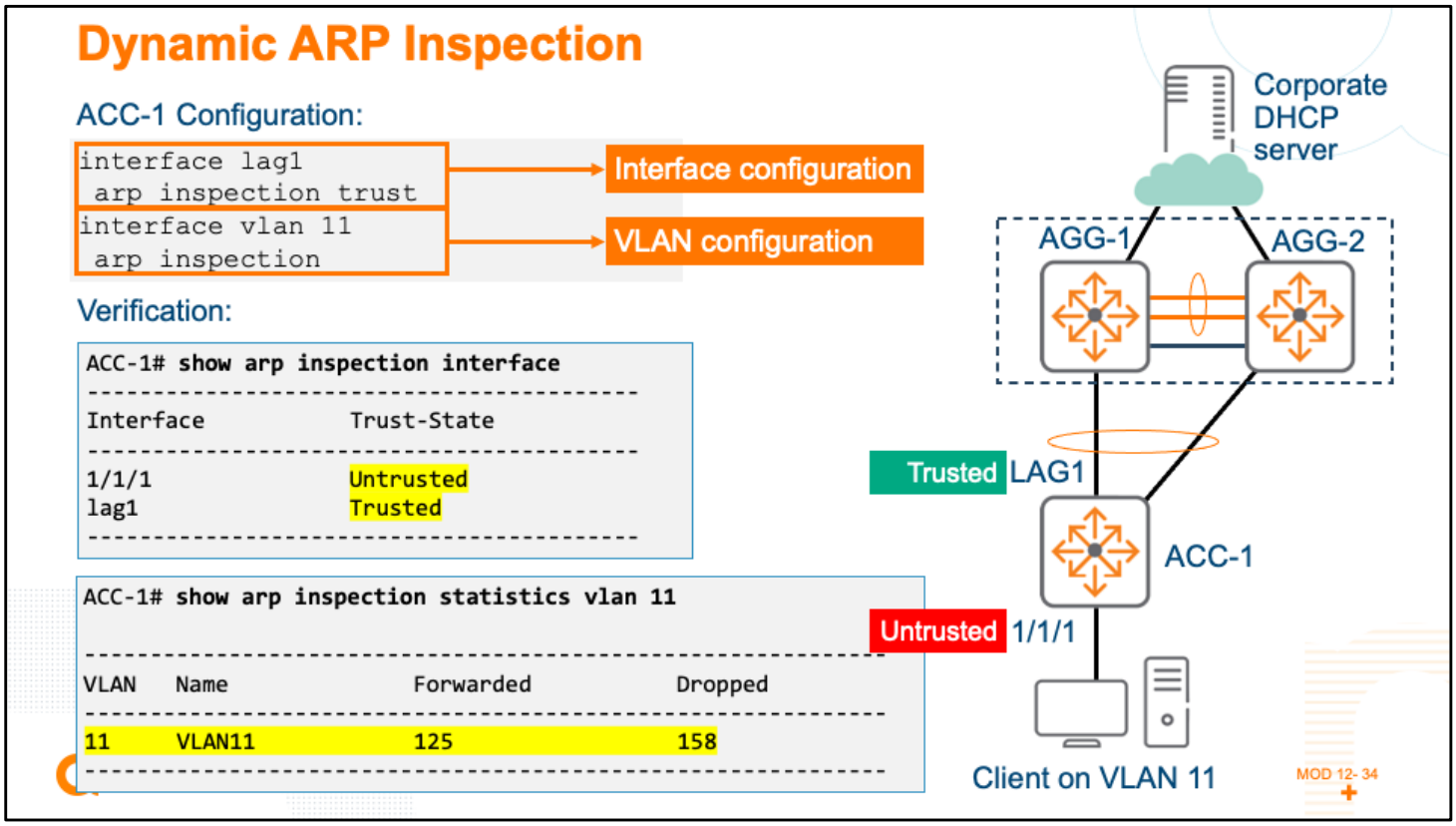
Defining the member ports of a VLAN as either trusted or untrusted.

Only ARP traffic on untrusted ports subjected to checks.

Routed ports (RoPs) always treated as trusted.

Use the DHCP Bindings table and check every ARP packet to match against the binding.

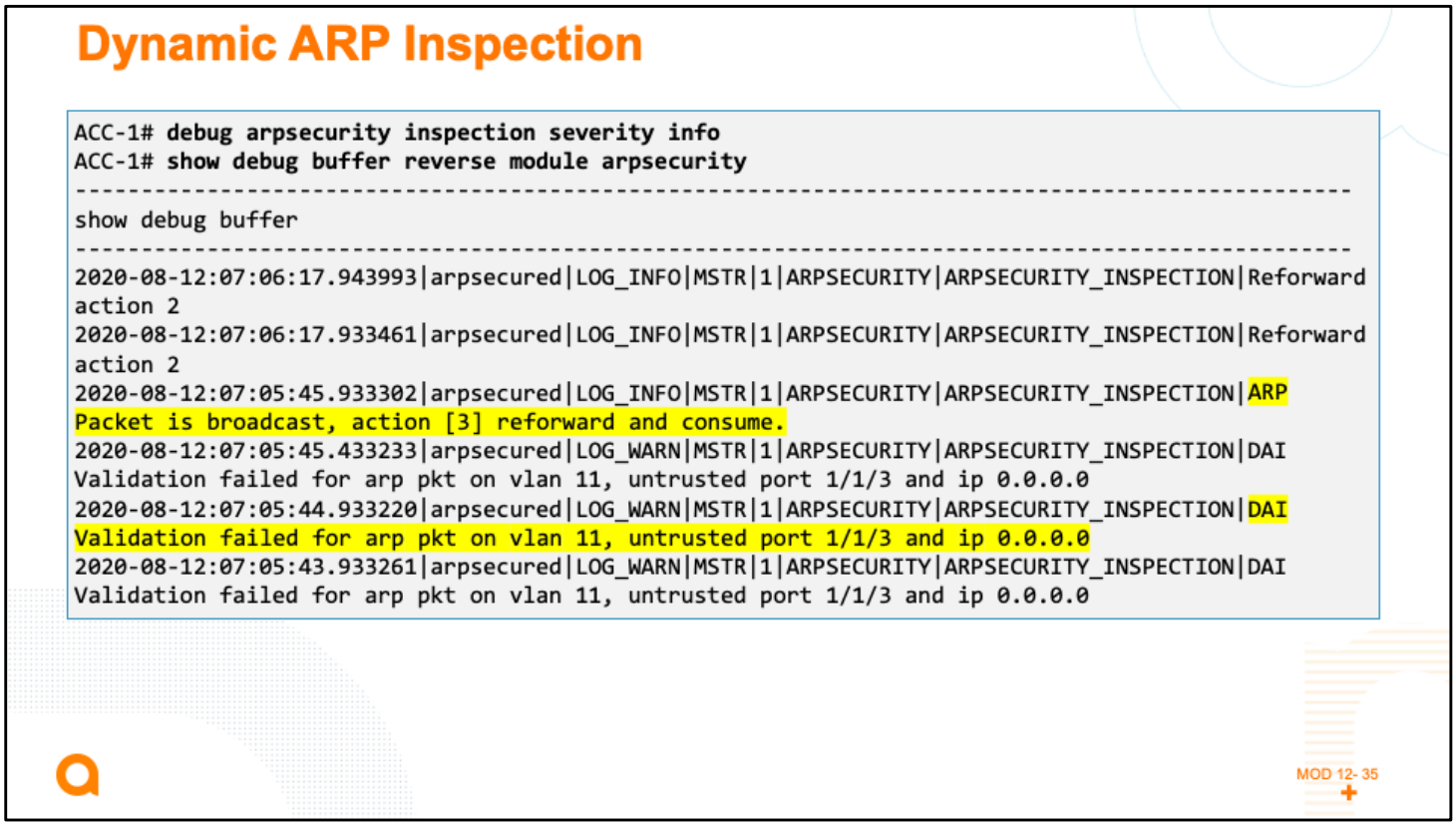
ARP ACLs are not supported in this release and the DHCP snooping table will be the only source of binding.



Take in consideration that Dynamic ARP inspection requires DHCP snooping to be configured first.

In the example, LAG 1 from ACC-1 switch perspective becomes the trusted interface. This means that ARP inspection will not apply to ARP packets received from this interface. On the other hand, traffic received from VLAN 11 is inspected.

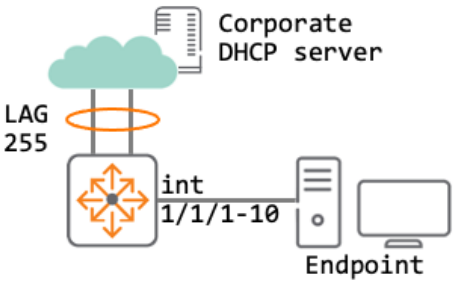
You can use show arp inspection interface command to display the Trust-state on each interface. AOS-CX can display statistics only when ARP inspection is applied to a VLAN, to visualize this information use the show arp inspection statistics vlan command.



The debug arpsecurity inspection severity info command helps to visualize the ARP inspection.

Question #2

```
# show dhcpv4-snooping statistics
Packet-Type  Action   Reason                          Count
-----------  -------  ------------------------------  ---------
server       forward  from trusted port               0
client       forward  to trusted port                 0
client       drop     destination on untrusted port   5
```



An administrator has detected that endpoints are not receiving IP parameters from DHCP server after deploying DHCP snooping. A close look into the DHCP-snooping statistics revels the AOS-CX switch has dropped packets, which script can solve this problem?

a.
```
interface lag 255
   dhcpv4-snooping trust
```

b.
```
interface 1/1/1-1/1/10
   dhcpv4-snooping trust
```

c.
```
dhcpv4-snooping trust all ports
```

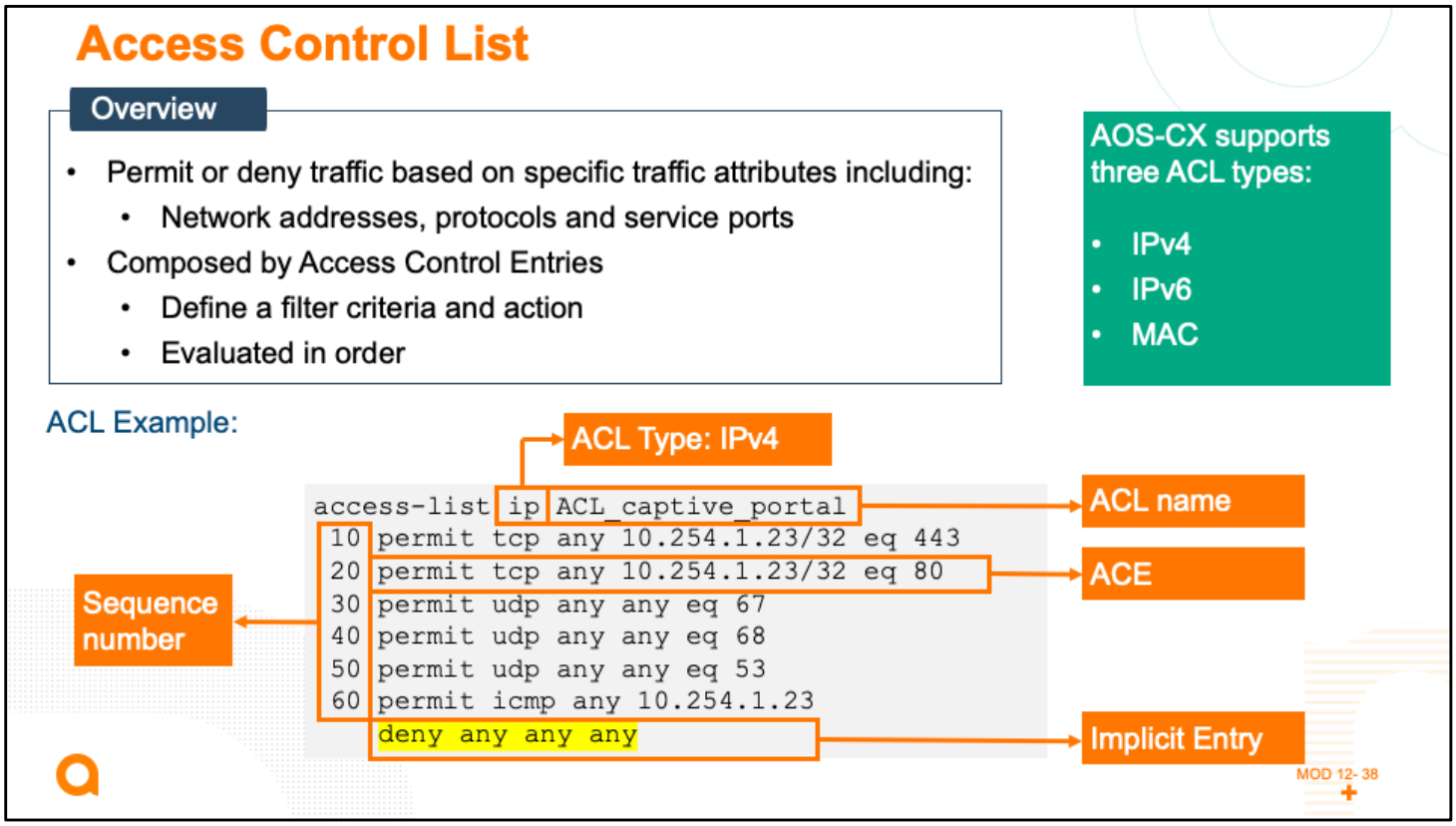
Knowledge Check

Correct answer:

A

aruba
a Hewlett Packard Enterprise company

Additional Content



Access Control Lists (ACLs) let you permit or deny traffic based on network addresses, protocols, service ports, and other packet attributes. ACLs are composed of one or more Access Control Entries (ACE). Each ACE defines a filter criteria and an action, either permit or deny. If the traffic matches the filter criteria, the specified action is taken. The permit action permits the traffic to continue through the switch. The deny action causes the traffic to be discarded (dropped). ACEs can also log or count matching traffic.

AOS-CX supports three ACL types, to support the unique frame or packet characteristics of three types of traffic - IPv4, IPv6, and Layer 2 MAC.

How do ACLs work?

Access Control Entries (ACEs) are listed according to priority by sequence number and processed from lowest to highest sequence number. Each ACE attempts to match on one or more attributes of the traffic type. Attempted ACE matching ceases upon the first successful match. For a successful match, a packet must match all ACE criteria, qualifiers, and attributes. Processing stops with a match. So, higher-numbered ACEs are only processed if no lower-numbered ACE matches. If the traffic matches no ACE in the entire ACL, the default deny action is taken, causing the traffic to be discarded (dropped).

Defining ACLs

When defining an ACE, if you omit the sequence number, the ACE is auto-assigned a new sequence number in increments of 10. If you choose to include the ACE sequence numbers, you can use any number you like, however it is suggested that you follow the practice of entering them as 10, 20, 30, and so on. Regardless of the order in which ACEs are entered, they are stored in low-to-high sequence number order. If you enter three ACEs numbered 10, 30, 20, when creating an ACL, the ACEs are stored in the ACL as 10, 20, 30. If no sequence number is specified, the software appends new ACEs to the end of the ACL with a sequence number equal to the highest ACE currently in the list plus 10.

Editing ACLs

When using the access-list ip or access-list ipv6 commands, if you enter an existing ACL-NAME, the existing ACL is modified as follows:

Any ACE entered with a new sequence-number creates an additional ACE.

Any ACE entered with an existing sequence-number replaces the existing ACE.

If you modify an ACL that has already been applied, it is possible that packets, blocked by the previous ACL, will briefly pass through the switch during the ACL reconfiguration.

NOTE: In a highly secure environment, it is safest to first bring down interfaces and VLANs to which an ACL has been applied before modifying the ACL. Then bring the targets of ACL application back up after completing the ACL modification. Respecting this recommendation ensures that an ACL is never partially programmed while traffic is passing through the switch.

ACL Example:

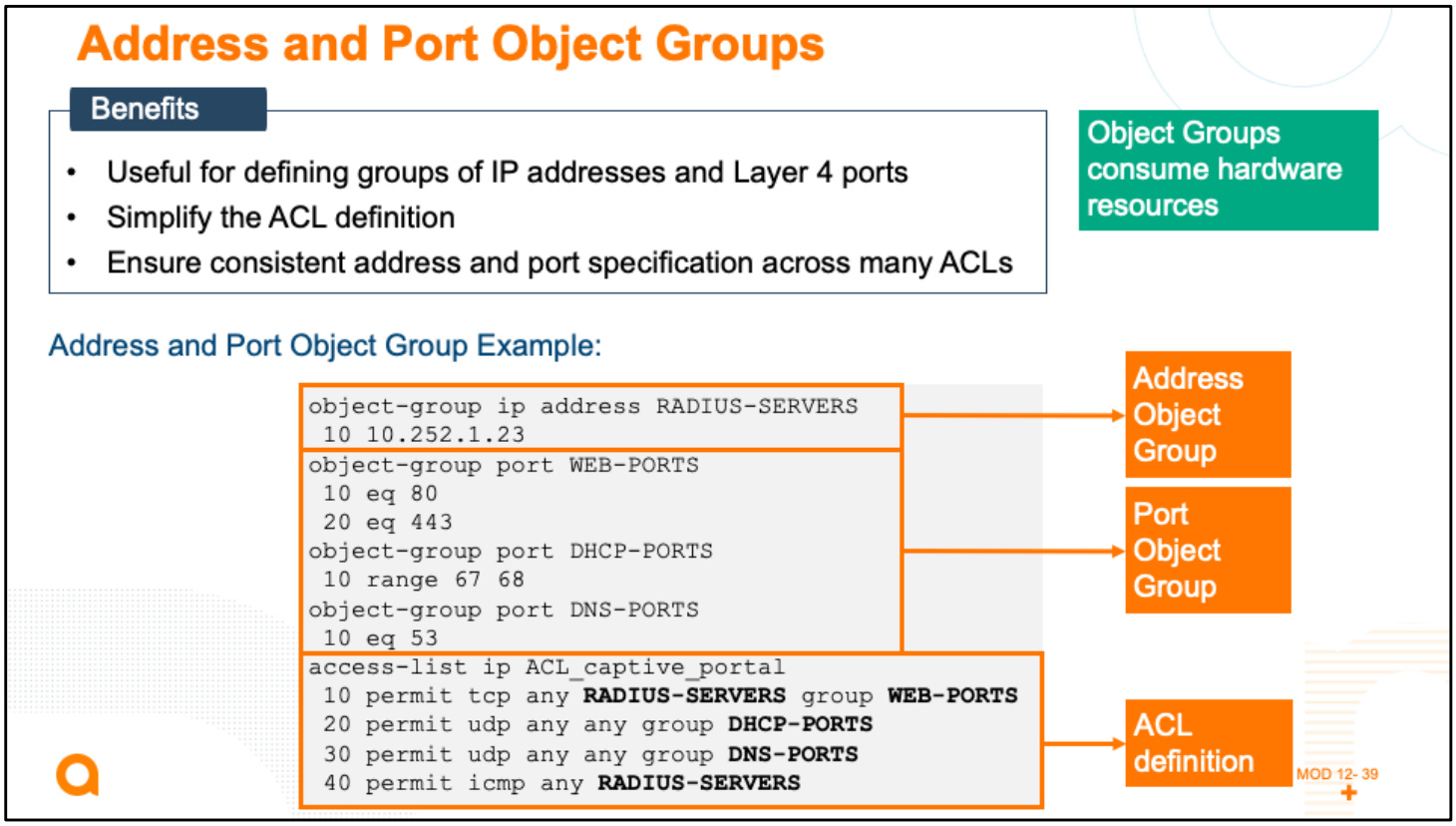
The following ACL permits the connection to a web captive portal for authentication purposes. This ACL allows the communication to the web server (10.254.1.23) using ports 443 and 80; it also permits DHCP and DNS traffic and for troubleshooting purposes it also allows the ICMP

communication.

access-list ip ACL_captive_portal
 10 permit tcp any 10.254.1.23/32 eq 443
 20 permit tcp any 10.254.1.23/32 eq 80
 30 permit udp any any eq 67
 40 permit udp any any eq 68
 50 permit udp any any eq 53
 60 permit icmp any 10.254.1.23

At the end of the ACL there is an implicit deny any any any ACE that blocks the rest of the traffic.

Note. This implicit ACE is not shown in the configuration and is displayed in the slide for academic purposes.



Object groups are useful for defining groups of IP addresses and Layer 4 ports for use exclusively in the two ACL-defining commands access-list ip and access-list ipv6.

Often, common groups of addresses and ports or port ranges are used repeatedly in many ACL definitions. Without address and port object groups, the same addresses and ports must be repeated in each ACL definition that uses them.

With address and port object groups, the IP addresses and ports can be defined once, using any of these commands:

object-group ip address

object-group ipv6 address

object-group port

Once an object group is defined, the group is available for inclusion by name as the <ADDRESS-GROUP> and <PORT-GROUP> parameters in the access-list ip and access-list ipv6 ACL-definition commands.

Object groups simplify the ACL definition process and help ensure consistent address and port specification across many ACLs.

Object groups are self-documenting. When you're setting up an ACE entry using an IP address or network number, its use may not be obvious. Using object groups with descriptive names makes the use more obvious.

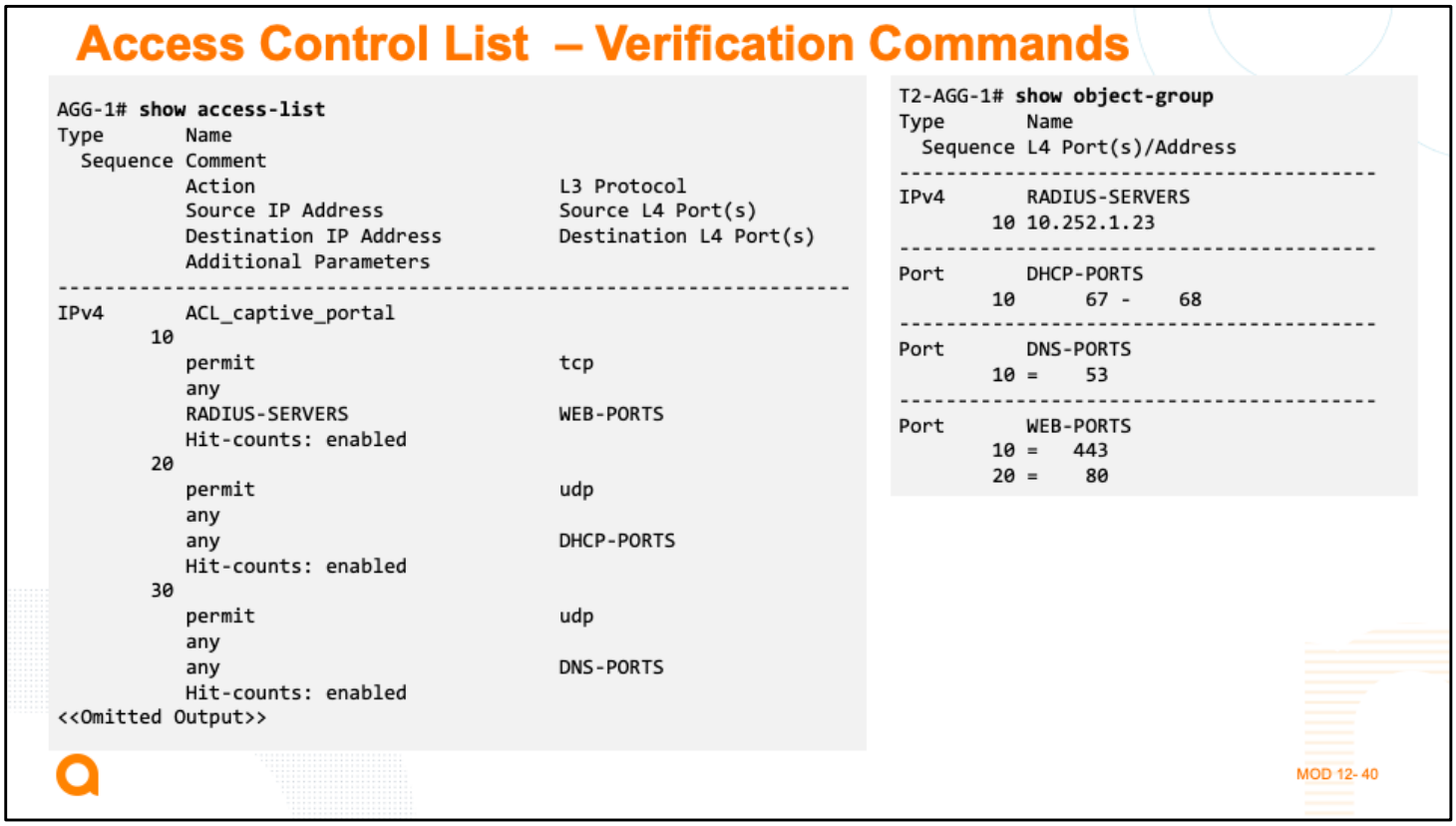
Note: Keep in mind that it is possible to consume many hardware resource entries when using the object group commands. For example, in a typical situation, an ACE that uses object groups with 3 source addresses, 3 source Layer-4 ports, 3 destination addresses, and 3 destination Layer-4 ports, a total of 81 hardware entries are consumed (3 * 3 * 3 * 3 = 81). Hardware resources are discussed in the next section.

In the example, we create the following object groups

One Address Object Group "RADIUS-SERVERS" that contains a single server (10.252.1.23)

Three Port Object Group that list Layer 4 ports used for WEB, DHCP and DNS services respectively

As you can see the ACL definition is simpler and easy to understand compared to the ACL presented in the previous slides.

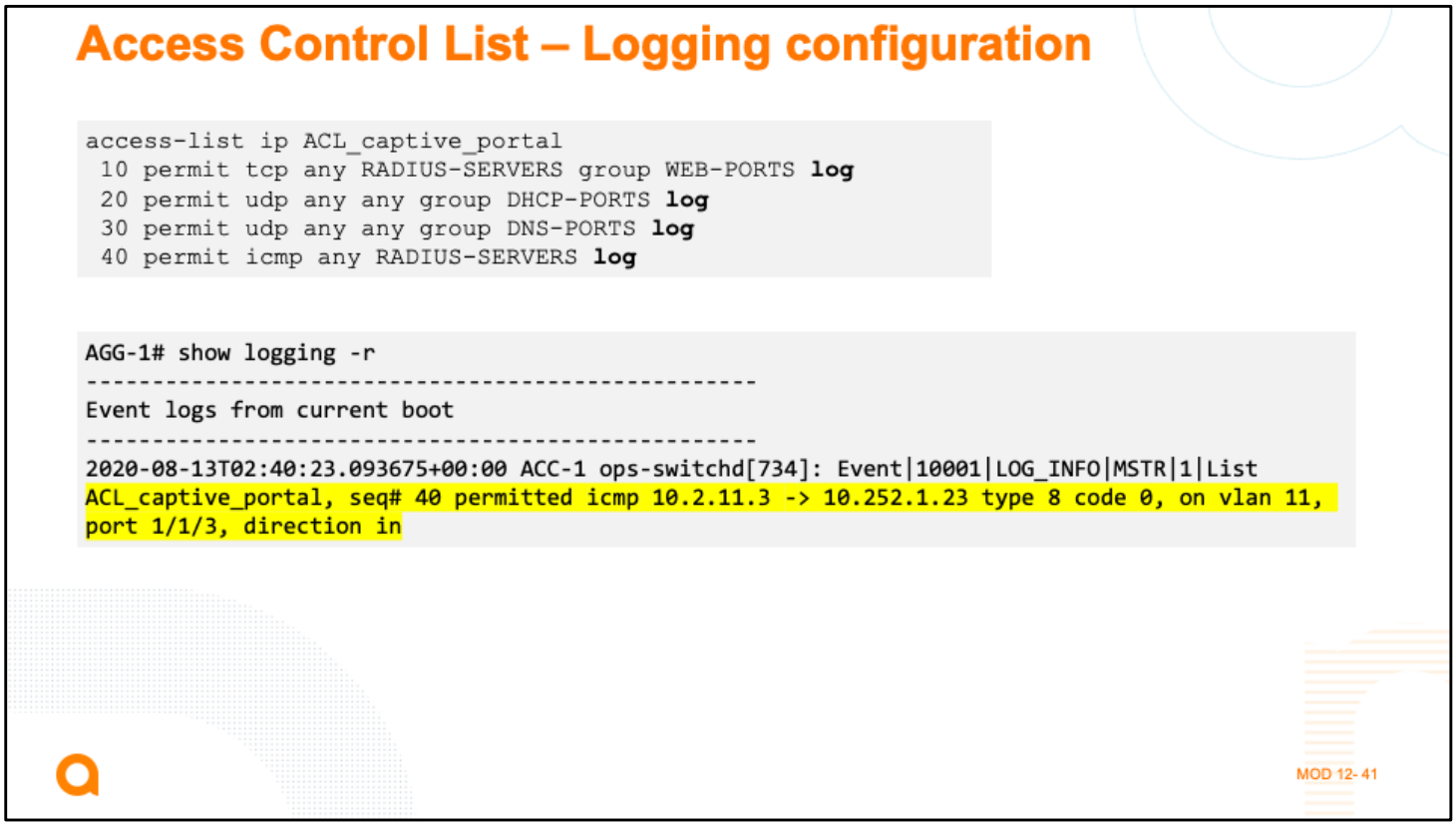


The simpler method to verify the access list is by using the show access-list command. This command shows information about your defined ACLs and where they have been applied. When show access-list is entered without parameters, information for all ACLs is shown. The parameters filter the list of ACLs for which information is shown.

Available filtering includes:

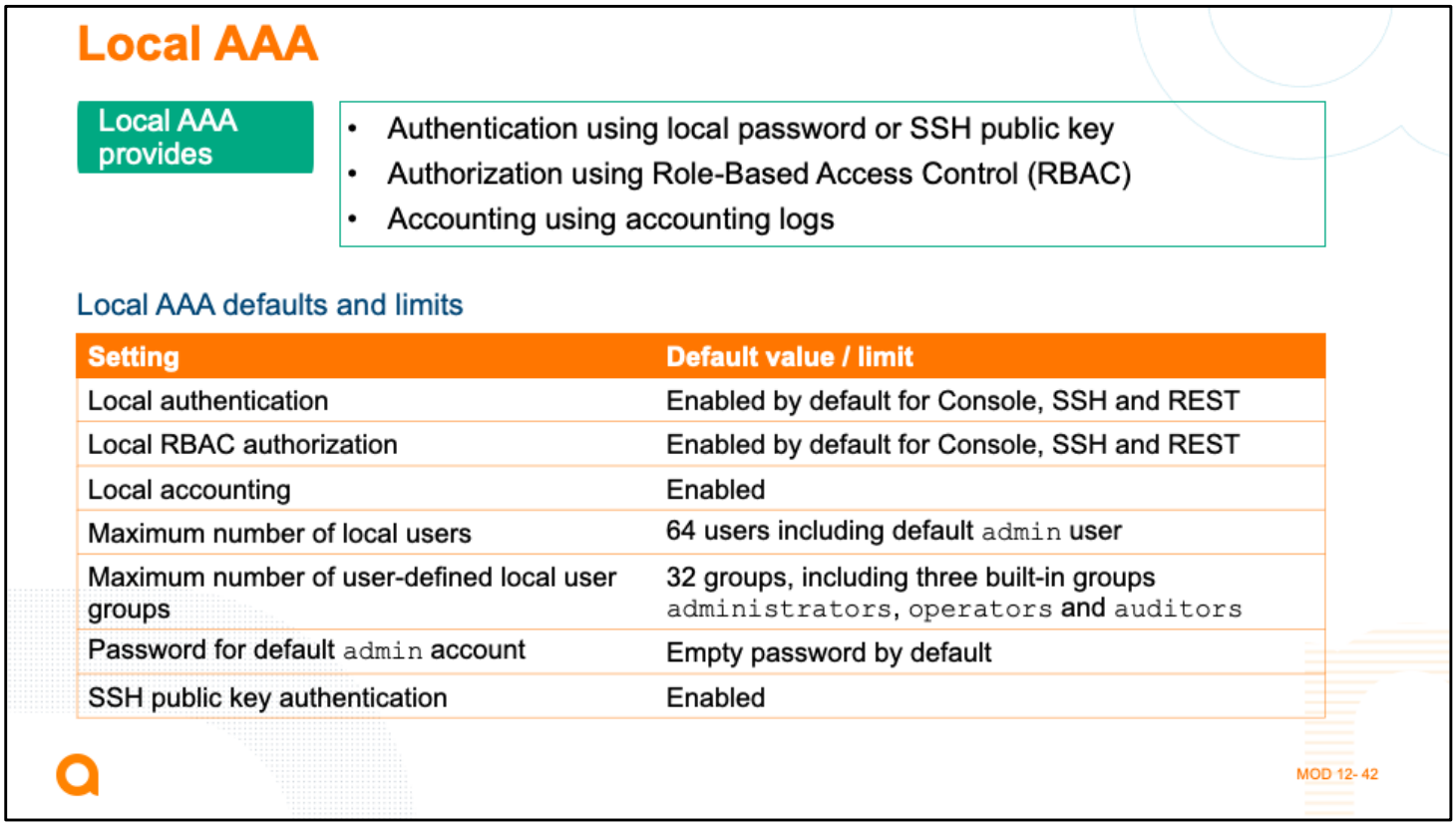
- The content of a specific ACL.
- All ACLs of a specific type.
- All ACLs applied to a specific interface (port or LAG).
- All ACLs applied to a specific VLAN.
- All ACLs applied in a particular direction.
- All ACLs applied to control plane (per VRF)

The show object-group command displays information about your defined object groups. When show object-group is entered without parameters, information for all object groups is shown. The parameters filter the list of object groups for which information is shown.



The log key word is used to configure the Logging action.

In the console the log can be displayed using the show logging – r command or show events – r command.



Local AAA on your Aruba switch provides:

Authentication using local password or SSH public key.

Authorization using role-based access control (RBAC), and optionally, using user-defined local user groups with command authorization rules defined per group.

Accounting of user activity on the switch using accounting logs.

The slide displays the Local AAA defaults and limits that applied to all AOS-CX platforms.

Local Authentication

Authentication identifies users, validates their credentials, and grants switch access. Local authentication is either password-based or SSH public key-based.

Password-based local authentication

Validates users with local username and password credentials

Is supported on all interfaces/channels (SSH, WebUI, Console, REST)

Is enabled by default but can be superseded by remote authentication or with SSH client

using SSH public key authentication

SSH public key-based local authentication

- Validates users identified with SSH public keys stored in the local user database
- Is supported on the SSH interface/channel with SSH client
- Takes precedence over password-based authentication whether local or remote
- Is enabled by default (also requires key configuration to work)

Local Authorization

Authorization controls authenticated users command execution and switch interaction privileges. Local authorization uses role-based access control (RBAC) to provide role-based privilege levels plus optional user-defined local user groups with command execution rules. Authorization occurs only after successful authentication.

- Administrators have full command execution and switch interaction privilege
- Operators are limited to the use of several nonsensitive show commands
- Auditors are limited to a few auditing-related commands

Optional per-command authorization is available through configuration of user-defined local user groups with command authorization rules applied to respective group members.

Local Accounting

Local accounting is always active. It cannot be turned off.

This accounting information is captured and made available locally (using show accounting log) and, if desired, for sending to remote AAA servers:

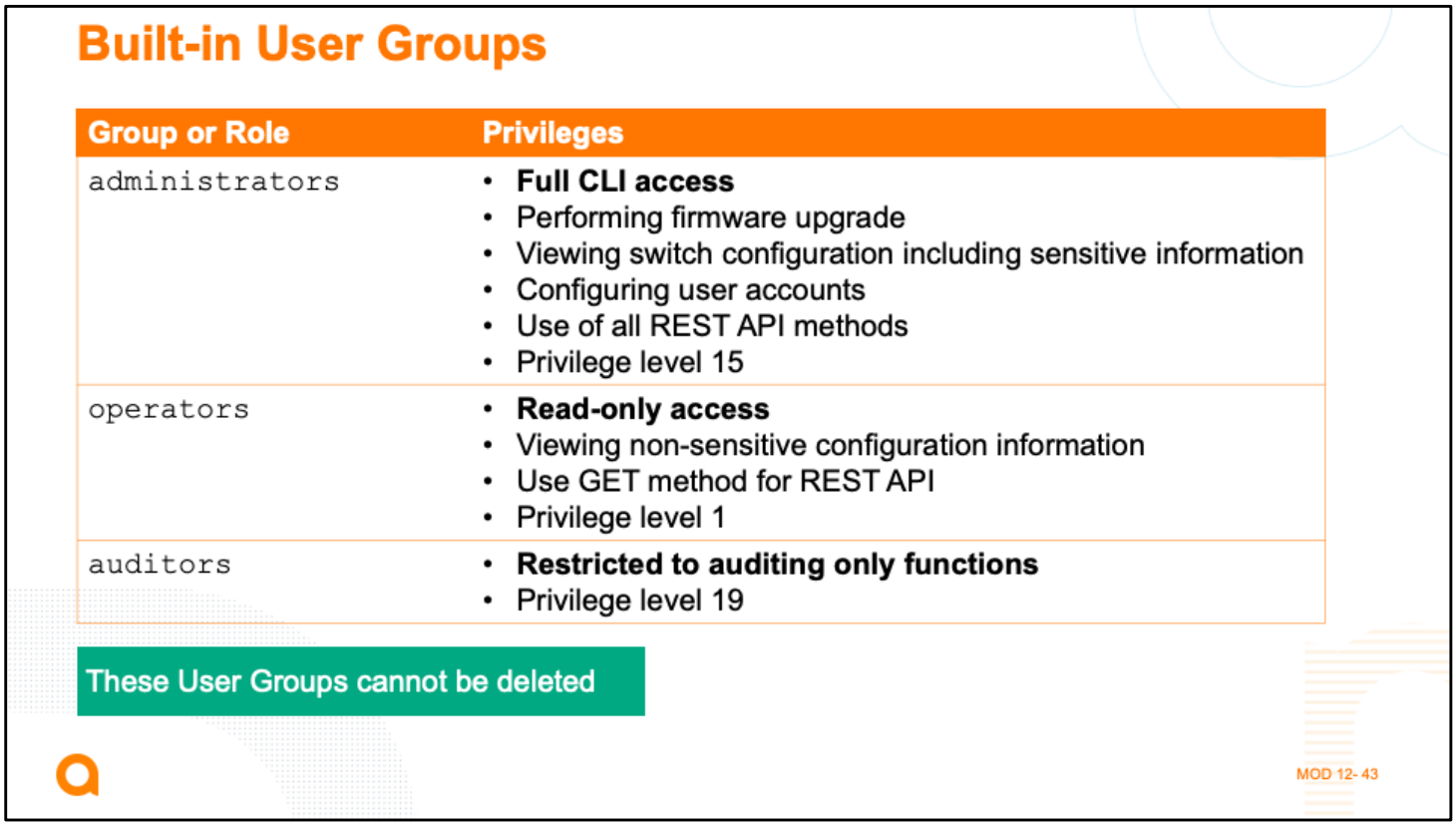
Exec Accounting: user login/logout events
Command accounting: commands executed by users
System accounting: remote accounting On/Off events
CLI show commands
Interactions on the non-CLI interfaces: REST and WebUI

The following is not captured or made available as accounting information:

CLI commands that reboot the switch
Interactions in the bash shell

The following commands are used to display the configuration of the AAA in the switch:

show aaa accounting
show aaa authentication
show aaa authorization



Built-in User Groups

administrators

Full CLI access

Performing firmware upgrades

Viewing switch configuration information, including sensitive information such as passwords which are displayed as ciphertext

Performing switch configuration

Adding/Removing user accounts

Configuring user accounts, including passwords. Once set, a password cannot be deleted or set to empty

REST API: All methods and switch resources are available

Privilege level is 15

operators

Operators have no switch configuration privilege.

Basic display-only CLI access

Viewing of non-sensitive configuration information

REST API: Other than the login and logout resources, only the GET method is available

Privilege level is 1

auditors

Restricted to functions related to auditing only

CLI: Access to commands in the auditor context only (auditor>)

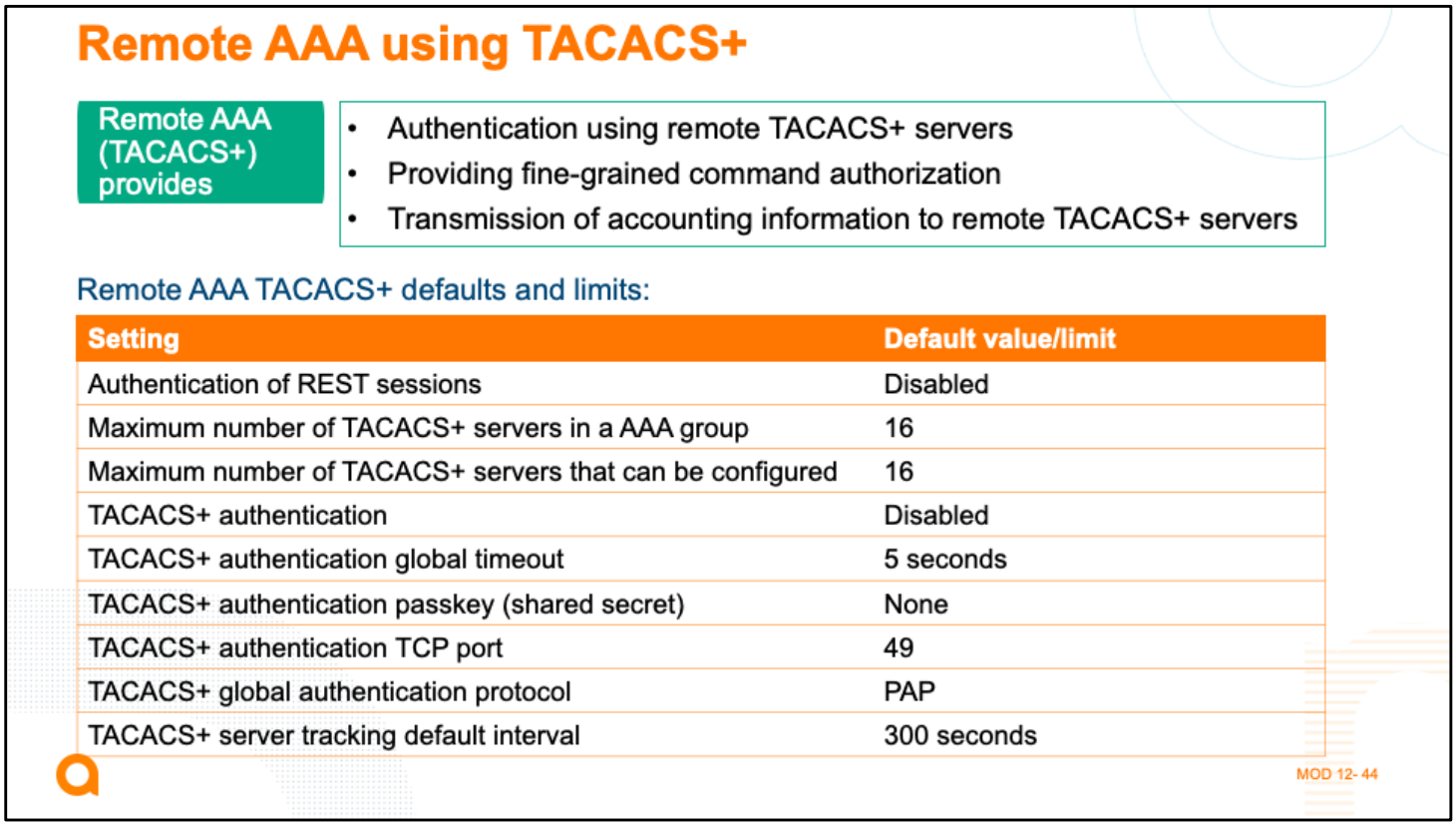
Web UI: Access to the System > Log page only

REST API: POST method available for the login and logout resources. GET method available for the following resources only:

– Audit log

– Event log

Privilege level is 19



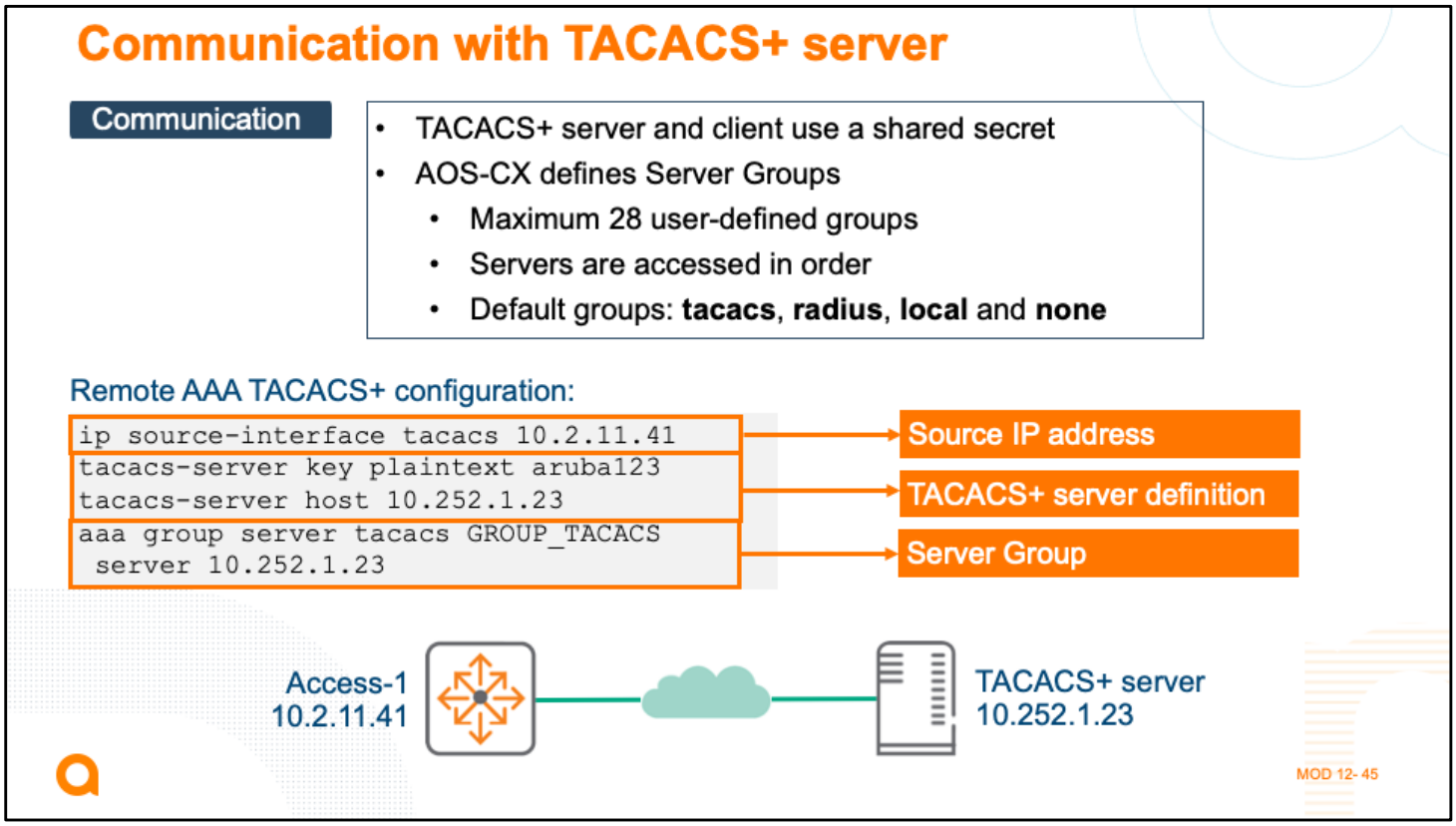
Remote AAA provides the following for your Aruba switch:

Authentication using remote TACACS+ AAA servers

Authorization using remote TACACS+ AAA servers, providing fine-grained command authorization. Optional user-defined local user groups with configured command authorization rules can be used to provide authorization fallback protection for when TACACS+ servers become temporarily unavailable.

Transmission of locally collected accounting information to remote TACACS+ servers

The slide displays the Remote AAA TACACS+ defaults and limits that applied to all AOS-CX platforms.



To communicate with a TACACS+ AAA server, the switch must have a passkey (shared secret) configured that matches what is configured on the server. Use one of these commands to achieve the desired configuration:

For a global passkey common to every TACACS+ server, use tacacs-server key comand

For a per-TACACS+ server passkey, use tacacs-server host command with the key parameter

Default Server Groups

The switch always has these four default groups:

tacacs: for remote AAA, always contains every configured TACACS+ server

radius: for remote AAA, always contains every configured RADIUS server

local: for local authentication

none: for local (RBAC) authorization

User-defined AAA servers are always added to the matching default group, either tacacs or

radius. Optionally, each server can also be added to exactly one additional user-defined (custom) group. A maximum of 28 user-defined groups can be created.

The order in which servers are added to a group is important. The server added first is accessed first, and if necessary, the second server is accessed second, and so on.

You can use the show aaa server-groups command to verify the correct creation of the server group.

Source IP address

If your topology allows the AAA server to be reached through multiple paths, the server interprets the incoming packets to be from different switches even though they are all coming from the same switch. Having a switch associated with multiple IP addresses makes it more difficult to interpret system logs and accounting data.

To ensure that all traffic sent from the switch to the AAA server uses the same source IP address, use ip source-interface tacacs or ipv6 source-interface tacacs.


Lab Activity
Lab 12

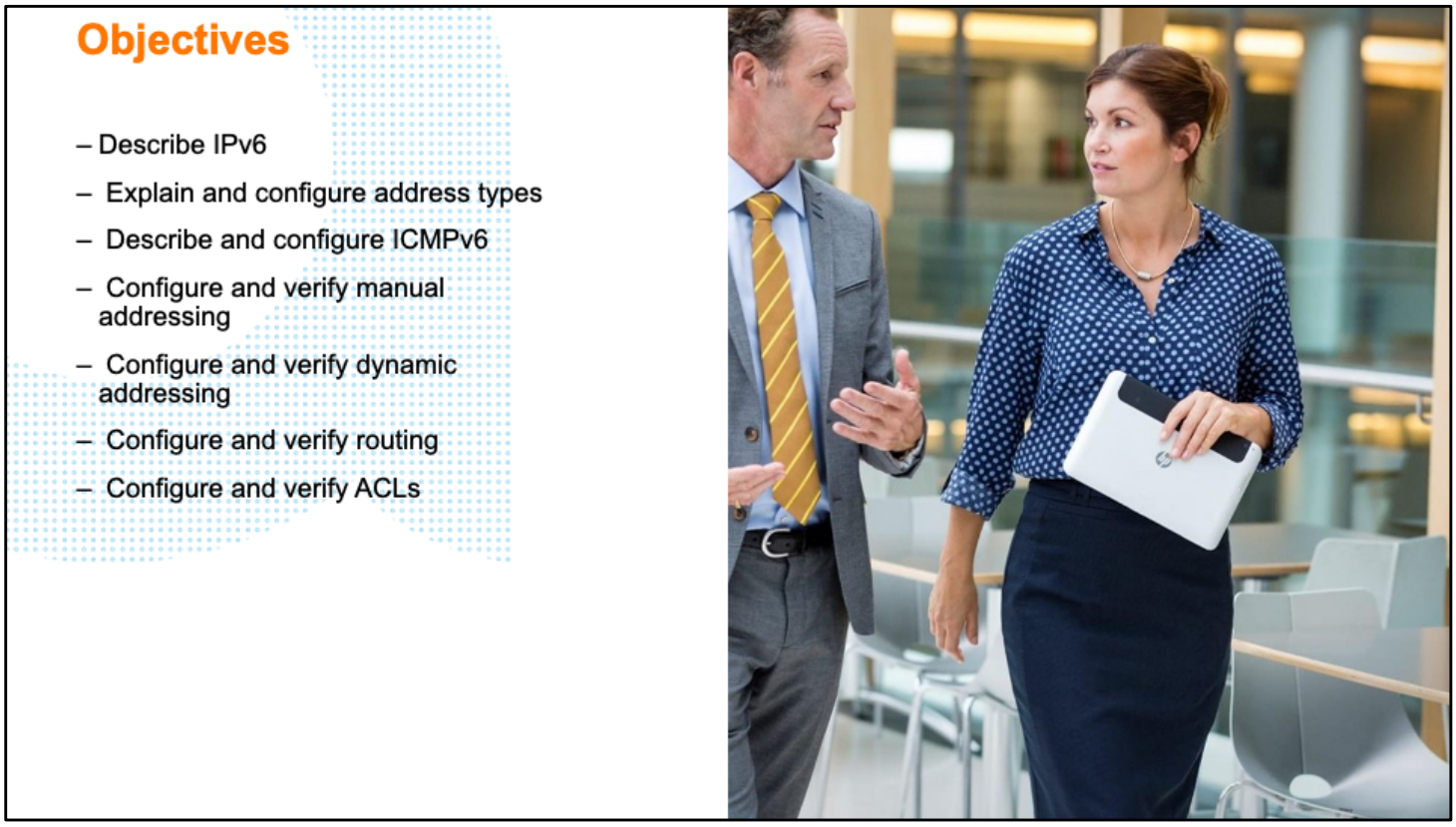
Rev #20.41

IPv6

Aruba Advanced Switching, Troubleshooting, and Solutions (ASTS)



aruba
a Hewlett Packard Enterprise company

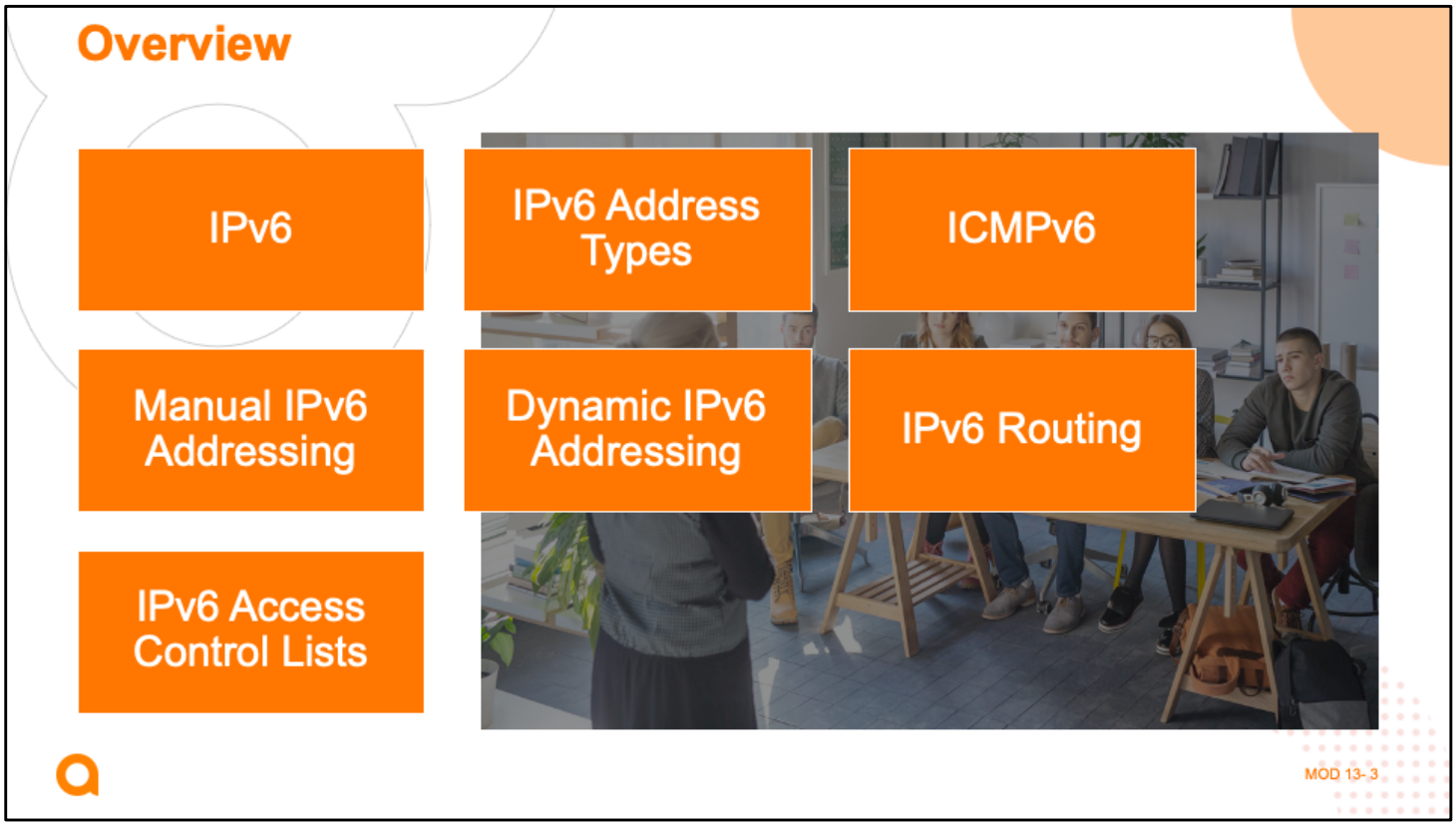


After completing this module you will be able to describe, configure, validate, and diagnose key aspects of IPv6 addressing, routing, and security.

You will be able to explain and configure various IPv6 address types, as well as to describe and configure ICMPv6.

You will configure and verify both manual and dynamic IPv6 addressing.

You will then leverage all these newly-gained capabilities to configure and verify IPv6 static routing and OSPFv3 routing, before moving on to the configuration and verification of IPv6 Access Lists.



As IPv6 continues to gain traction in the industry, we realize that its not just about having a much larger, more accommodating address space. There are significant efficiency and capability improvements inherent to IPv6, which you will learn to describe, configure, and validate.

You begin with an overview of IPv6 features, headers, and the extension headers - which make IPv6 inherently extensible.

You explore IPv6 address types. The various unicast addresses include Global, Link Local, Loopback, Embedded, and more. The multicast types include well-known and solicited node, and then there are the Anycast addresses.

Next you explore ICMPv6, with a focus on Neighbor Discovery and Router Advertisements. This prepares you for learning to describe, configure, and verify manual IPv6 addressing, including Duplicate Address Detection (DAD).

From manual addressing, you move to Dynamic IPv6 addressing – enhancing your understanding of Router Advertisements and their role in StateLess Address AutoConfiguration (SLAAC), along with DHCPv6. You will explore Interface-ID generation, the privacy extension, and address states, along with configuration and verification. This gets interesting as you learn how to confirm and apply your understanding with some Wireshark analysis. This is a great way to solidify your knowledge, in preparation for real-world deployment and diagnostic scenarios.

Then you will learn about IPv6 routing – both static and dynamic, with an emphasis on OSPFv3 – the IPv6-capable version of OSPF. You dive into interface numbering, packet

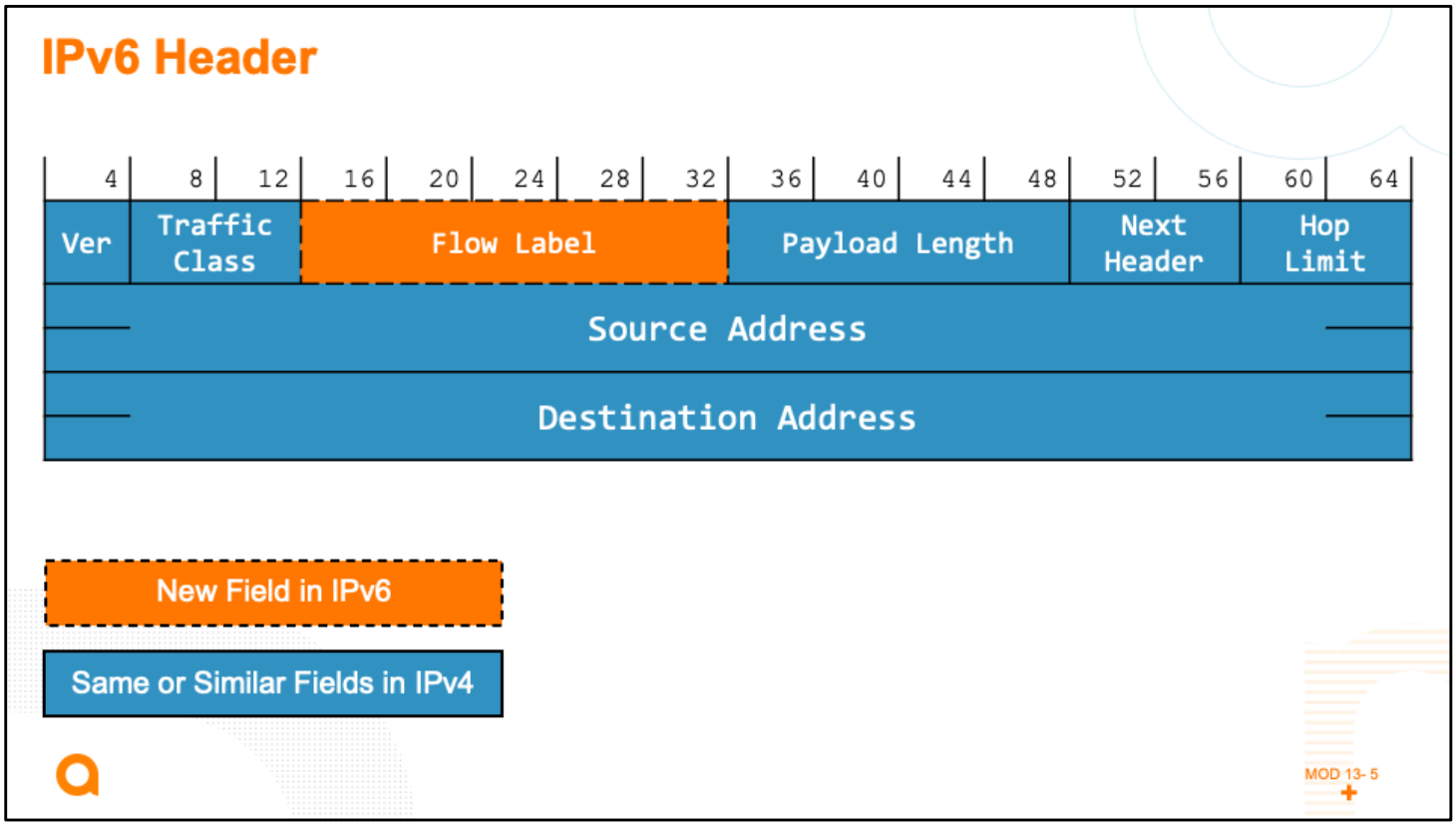
headers, and various packet types – such as hello and Link State Advertisements (LSA). You explore options and header fields, before getting into configuration and verification.

You then review IPv6 Access Control List (ACL) configuration and verification, before doing some hands-on lab activities to practice your new-found understanding.

aruba
a Hewlett Packard Enterprise company

IPv6

Features, Header, Extension Headers, Address representation



The structure of the IPv6 header was defined on RFC 2460.

IPv6 fields:

Version. The expected number is 6, this field is the same as the one used in IPv4.

Traffic Class. This is like the IPv4 Type of Service (ToS) field. It tells the router how to treat each received packet – used for Quality of Service (QoS) features to offer different levels of service or precedence. IPv6 uses this same Differentiated Services technique. Using 6 bits for the DSCP (Differentiated Services Code Point) allows for 64 unique markings, for 64 possible service levels.

Flow Label. This is a new field, used to tag a sequence or flow of IPv6 packets sent from a source to one or more destinations. This flow can be used by a source to label a sequence of packets for which it requests special handling by the IPv6 routers. Flow label usage is described in RFC 6437.

Payload Length. Indicates the payload length in bytes, not including the IPv6 header. If the packet has one or more extension headers, they are considered part of the payload. IPv6 fragmentation is only performed by the packet's source. So, if a router receives a larger packet than expected, it discards the packet and sends an "ICMPv6 – Packet too big" message to the source, indicating that the frame was dropped. To avoid this issue, set a minimum MTU (1280 Bytes) on all the links in the path.

Next Header. Indicates the type of header expected after the main IPv6 header. This field is

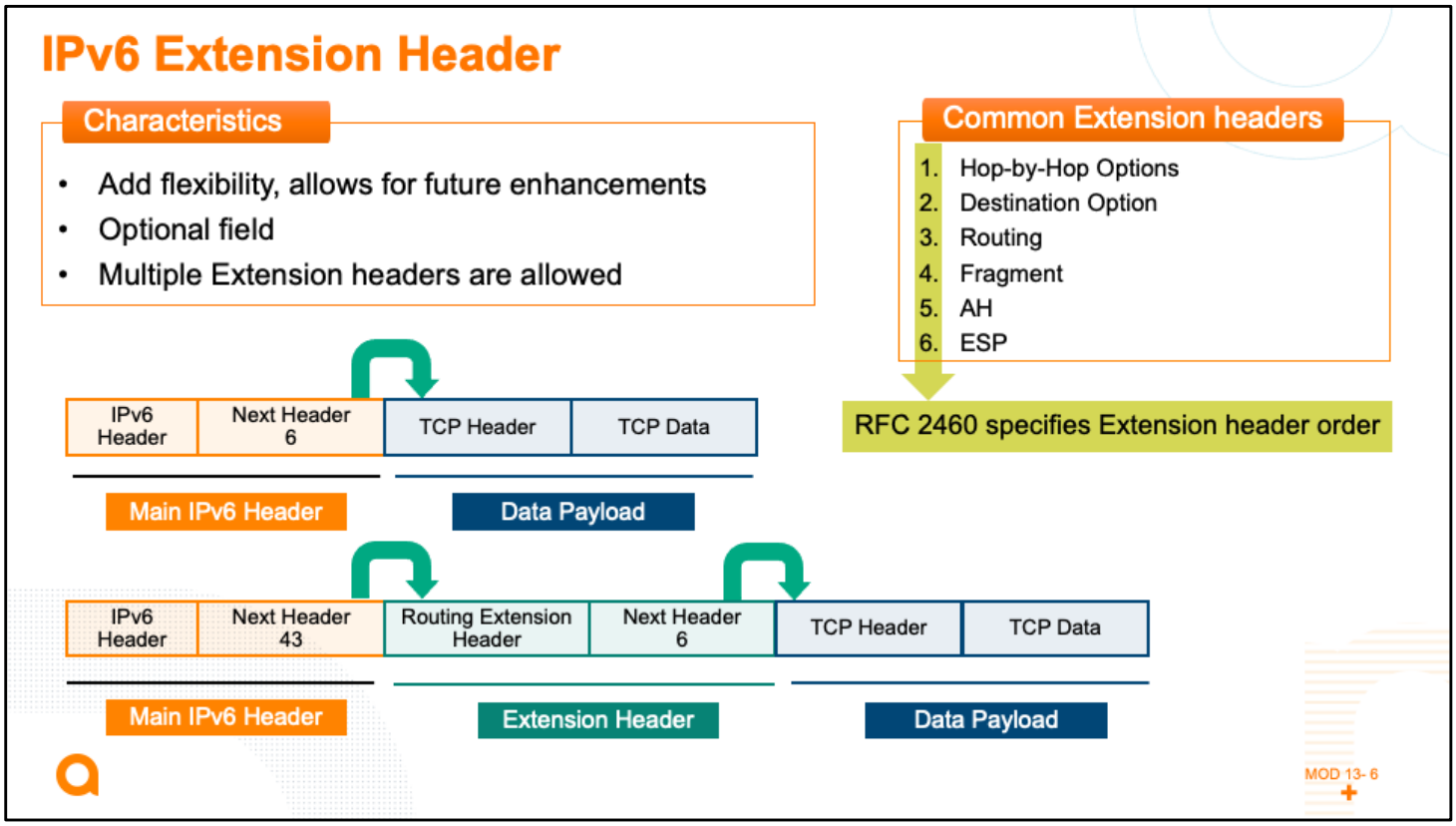
comparable to the IPv4 Protocol field.

Hop Limit. Like the IPv4 Time-to-Live (TTL) field, this value is decremented by 1 each time a router process a packet. This helps to mitigate Layer 3 route loops by providing a finite number of possible hops.

Source address. This is the source IPv6 address.

Destination address. This is the destination IPv6 address.

Note. The IPv6 header is byte aligned (64 bits) which eases processing and logic.



Extension headers are an important addition to IPv6. They are optional headers that add flexibility and allow future IPv6 enhancements. As shown in the figure, extension headers come after the main IPv6 header. Packets can have multiple extension headers.

RFC 2460 defines the order of the extension headers:

Hop-by-hop options header. Used to carry optional information that should be examined by every router along the path of the packet. Options provide flexibility, allowing IPv6 packets to be supplemented with sets of values or TLV (Type-Length-Value) triplets. This header can reference a Jumbo Payload option.

Destination options header. Used to carry optional information that is only examined by a packet's destination node(s). Described on RFC 6275.

Routing header. Allows the source of the packet to specify the path to the destination.

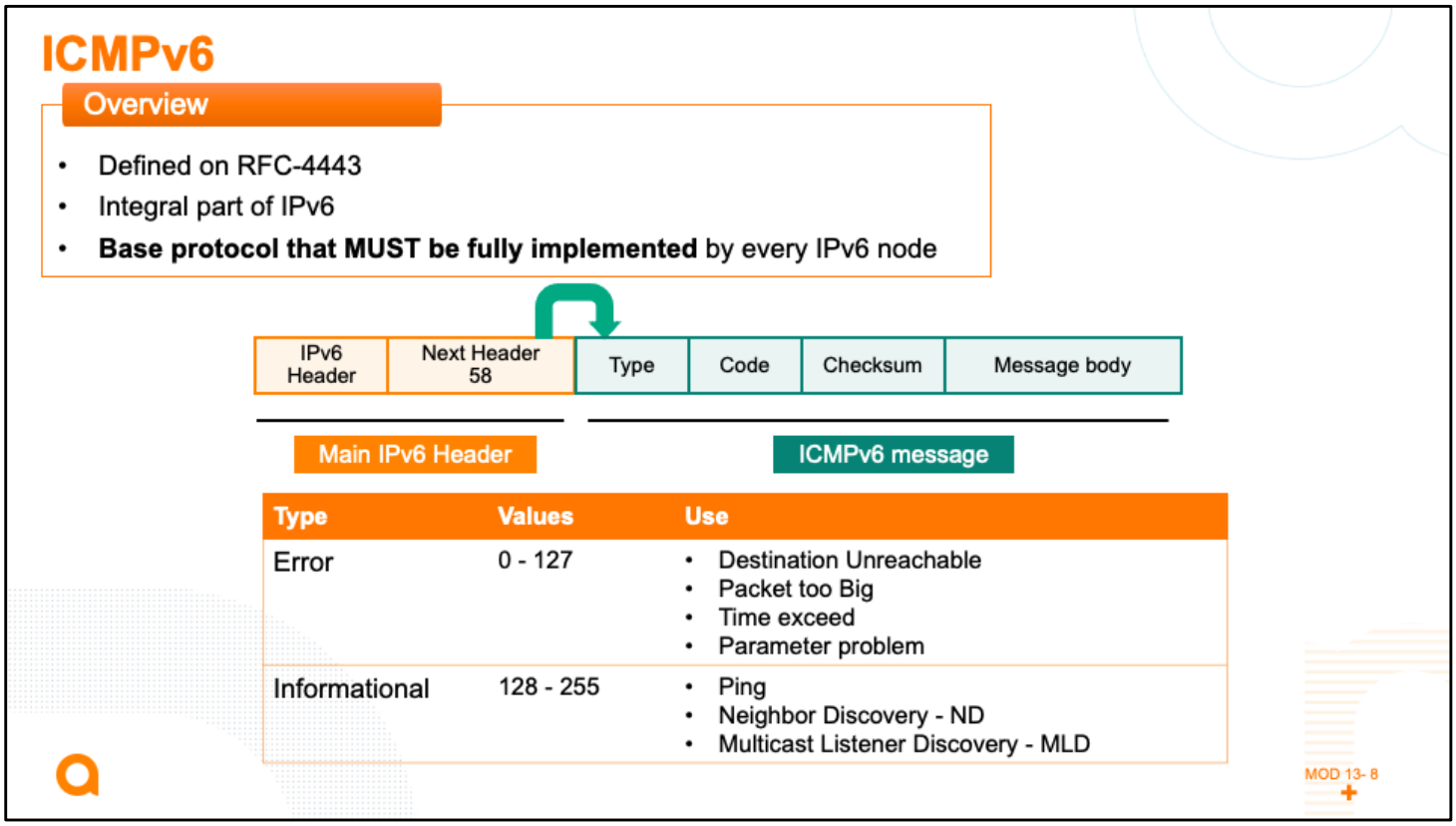
Fragment header. When the source must fragment the packet, it generates a unique identification value for each piece. This ensures that fragments are properly reassembled. Remember that in IPv6 intermediate nodes cannot fragment the packet.

Authentication header. Provides authentication and integrity to ensure the participants are who they say they are and that any alteration in the packet's content is detected. So, unlike IPv4, IPsec capabilities are integrated directly into IPv6.

Encapsulating Security Payload header. Provides authentication, integrity and encryption services, as with IPv4 IPsec.

IPv6 Neighbor Discovery

aruba
a Hewlett Packard Enterprise company



Defined on RFC 4443, ICMPv6 is used by IPv6 nodes to report errors and to perform other functions such as diagnostics (ICMPv6 ping). ICMPv6 is an integral part of IPv6 and all nodes MUST be fully implemented.

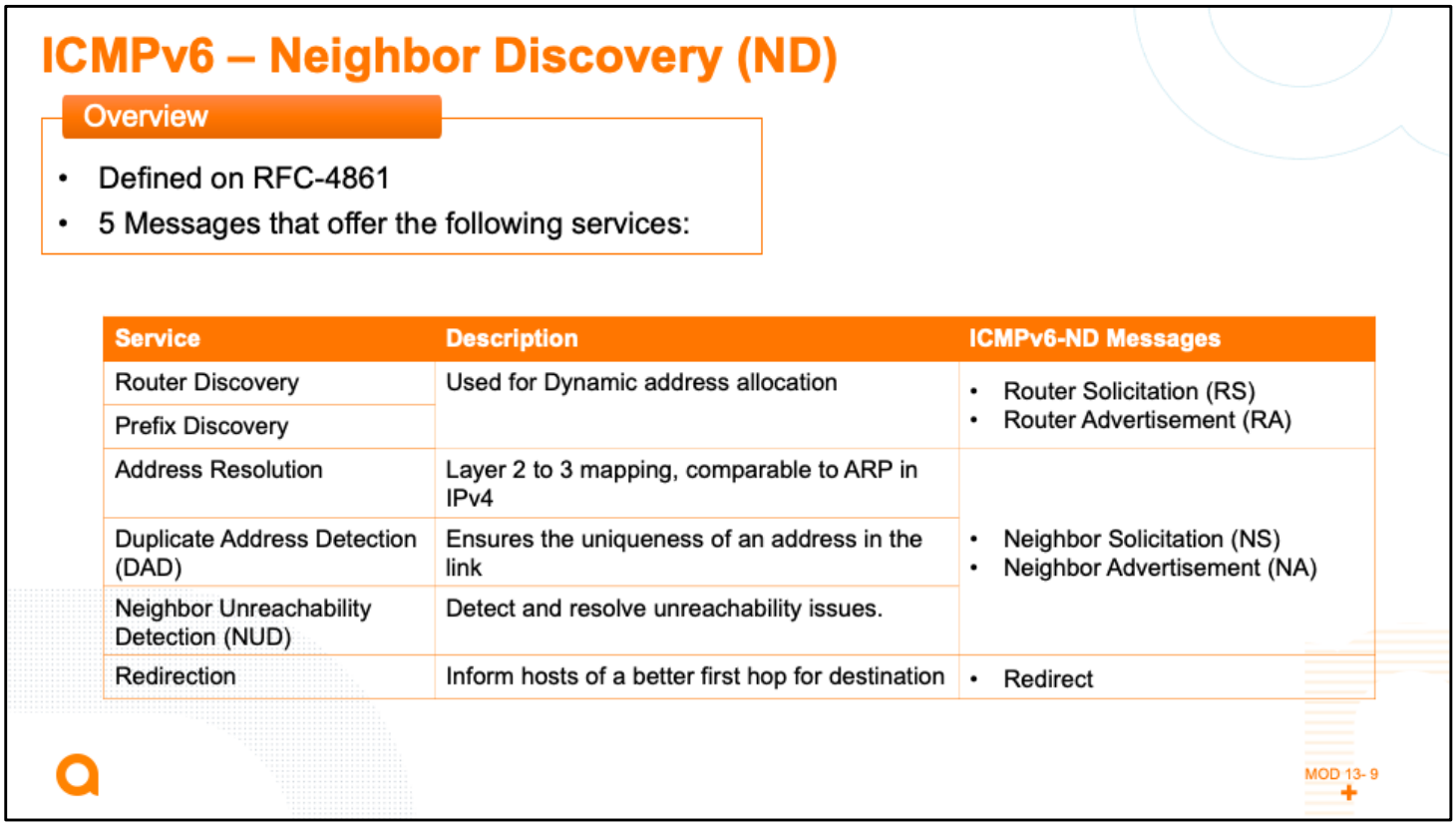
ICMPv6 is an extra header identified by the number 58 in the main IPv6 header. As the figure shows ICMPv6 packets are divided in 2 types:

Error. These messages have the Type value set from 0 to 127. The following are the common usage:

- Destination Unreachable. This message is sent when a packet cannot be delivered to its destination for reasons other than congestion. For example, when there is no route to destination, or the communication is administratively prohibited by an ACL.
- Packet too big. The packet received is larger than the MTU of the egress interface configuration. The router drops the packet and sends an ICMPv6 packet too big message to the source, including the MTU size so that the source can modify the packet size.
- Time exceed. Indicates either a routing loop or a hop limit value that is too small. Another use is when fragmentation is used. The destination expects packets to arrive within a specific amout of time. If this does not occur and the time expires an ICMPv6 Time exceed message is sent to the source.
- Parameter problem. Generated when a device processing a packet finds a problem with a field in the main IPv6 header or an extension header.

Informational. These messages have the Type value set from 128 to 255. The following are common usage:

- Ping. Same as IPv4, echo request and echo reply messages help to verify the connectivity between nodes.
- Neighbor Discovery. A set of 5 messages that offer important services such as Address resolution or redirection.
- Multicast Listener Discovery. Messages that help nodes to handle the join or leave process for nodes that participate in a specific Multicast Group-ID (comparable to IGMPv2,3 in IPv4).



Neighbor Discovery is probably the main functionality that ICMPv6 provides to IPv6. This protocol is defined on RFC-4861 and uses the following 5 messages:

Router Solicitation: When an interface becomes enabled, hosts may send out Router Solicitations that request routers to generate Router Advertisements immediately rather than at their next scheduled time.

Router Advertisement: Routers advertise their presence together with various link and Internet parameters either periodically, or in response to a Router Solicitation message. Router Advertisements contain prefixes that are used for determining whether another address shares the same link (on-link determination) and/or address configuration, a suggested hop limit value, etc.

Neighbor Solicitation: Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Neighbor Solicitations are also used for Duplicate Address Detection.

Neighbor Advertisement: A response to a Neighbor Solicitation message. A node may also send unsolicited Neighbor Advertisements to announce a link-layer address change.

Redirect: Used by routers to inform hosts of a better first hop for a destination.

The services that are provided by Neighbor Discovery (ND) are:

Router Discovery. A node can request the assistance of a router, which responds with specific

information, such as prefix and prefix length. This information can be used to generate a valid Global Unicast address.

Prefix Discovery. Layer 3 devices advertise the prefix to devices in the subnet, those devices use the information to autoconfigure a Global Unicast address.

Address Resolution. Process on which the Layer 2 and Layer 3 information is mapped, this processes is comparable to ARP in IPv4.

Duplicated Address Detection. Process that ensures the uniqueness of an address in the subnet. This process happens regardless of the configuration method (manual or dynamic).

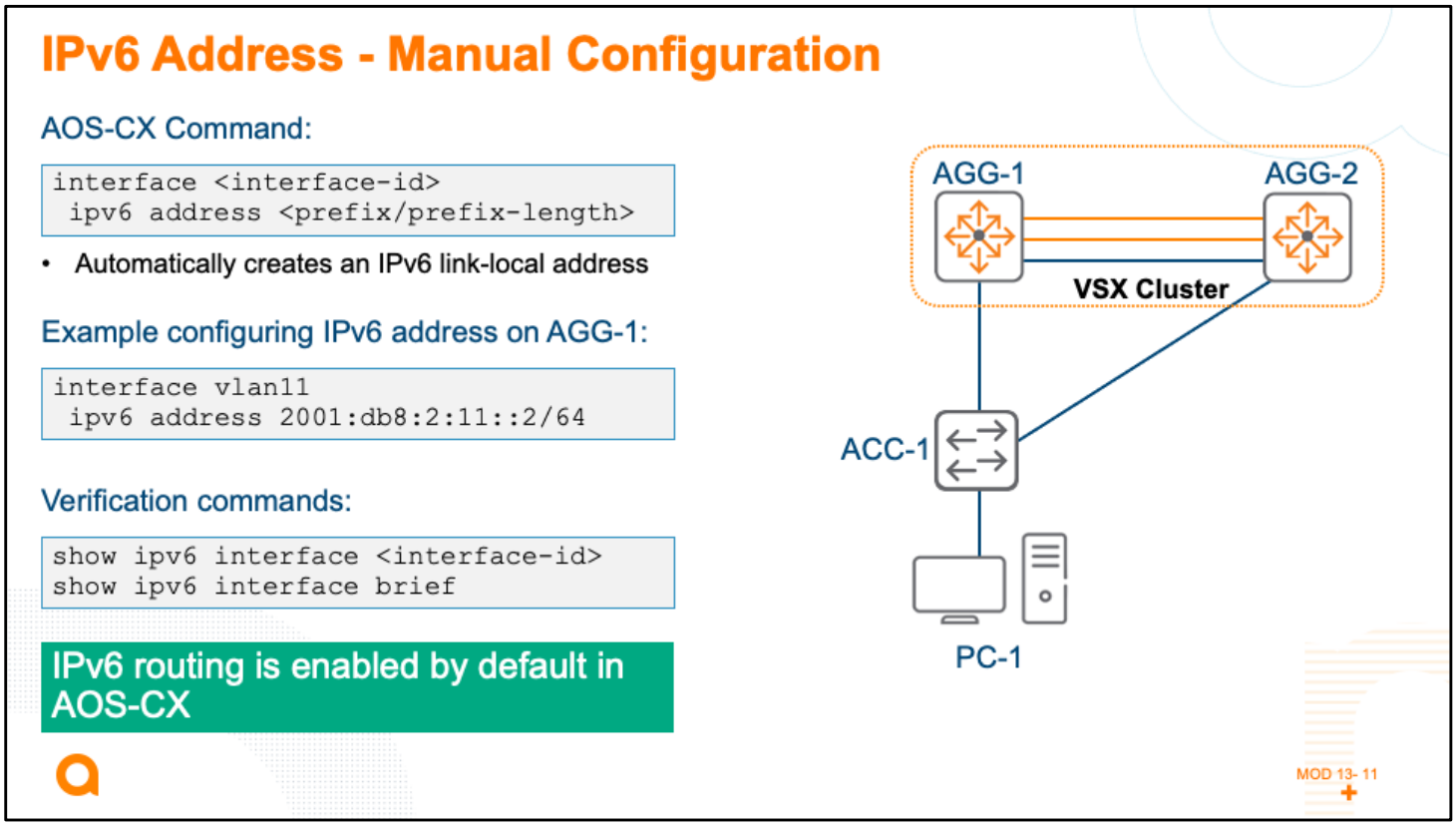
Neighbor Unreachability Detection. Determine whether a neighbor is reachable from the perspective of the device.

Redirection. Routers send Redirect messages to redirect the sender of a packet to a better first-hop or next-hop router. Redirect messages also inform the sender that the destination is on its same subnet and the packet can be sent directly to the destination.

aruba
a Hewlett Packard Enterprise company

Manual IPv6 Addressing

Configuration, Verification and DAD



The configuration of an IPv6 address manually in AOS-CX is straight forward. Just use the ipv6 address command from config-interface mode. This command automatically creates an IPv6 link-local address on the interface. However, it does not add the command with the link-local address to the running configuration. If you remove the IPv6 address, the link-local address is also removed. To maintain the link-local address, you must manually enter the command.

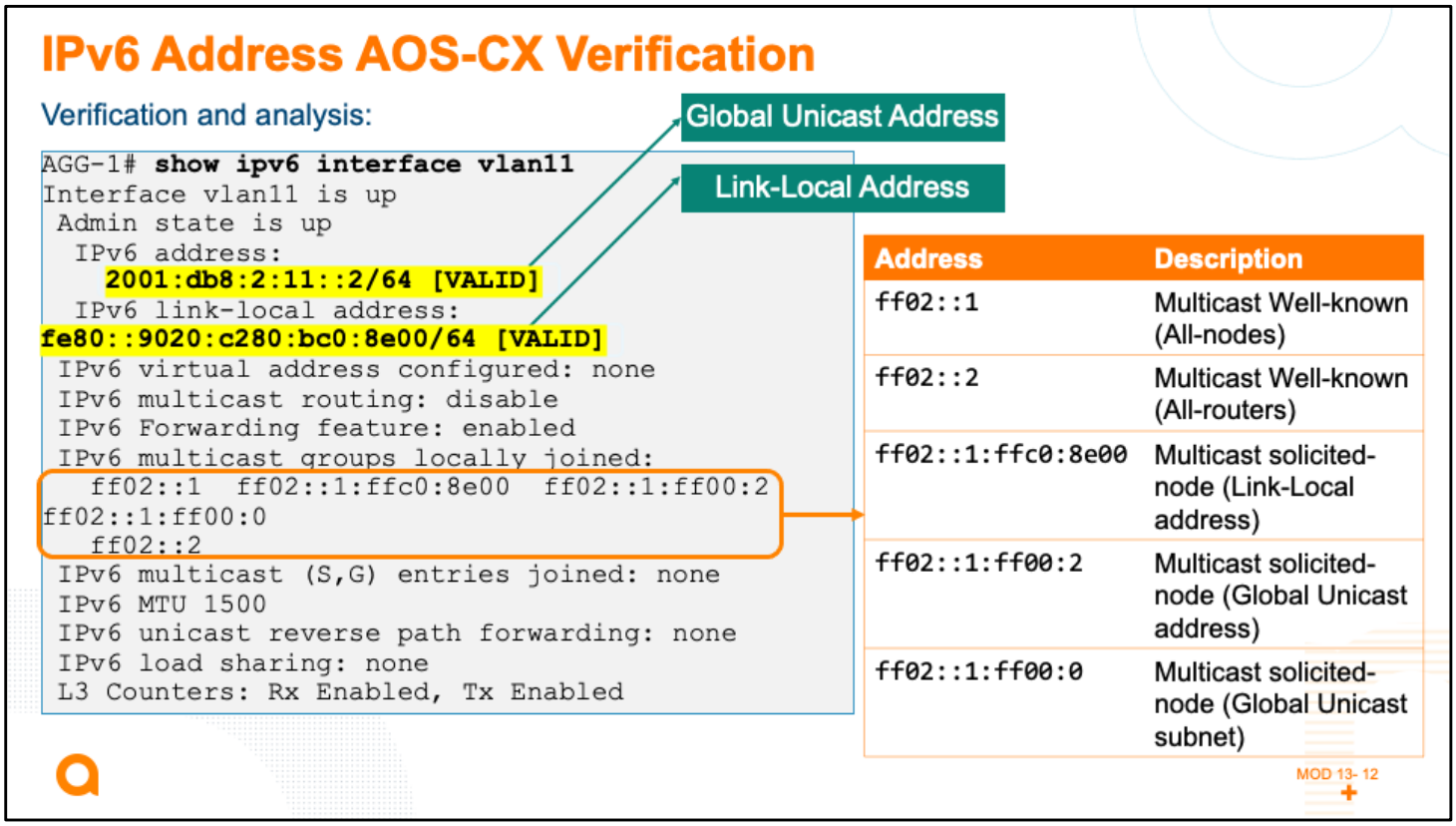
IPv6 routing is enabled by default with AOS-CX.

The figure shows and example configuration on the AGG-1 switch:

interface vlan 11

ipv6 address 2001:db8:2:11::2/64

To verify the IPv6 address use the show ipv6 interface and show ipv6 interface brief commands. Let's analyze that first command.



This output helps you to verify the IP address value assigned to the interface VLAN11. Notice that after the IP address, the keyword VALID indicates that this address is unique in the link - the DAD process has already run.

As stated before, when a Global Unicast address is defined on an interface, a Link Local address is automatically generated. It is important to clarify that a unique Link-Local address will be defined for each interface, but the same interface can be defined with more than one Global Unicast address.

Since Broadcast is no longer used in IPv6, communication to multiple devices must be through Multicast. The IPv6 address automatically creates a mapping to pre-defined Multicast groups:

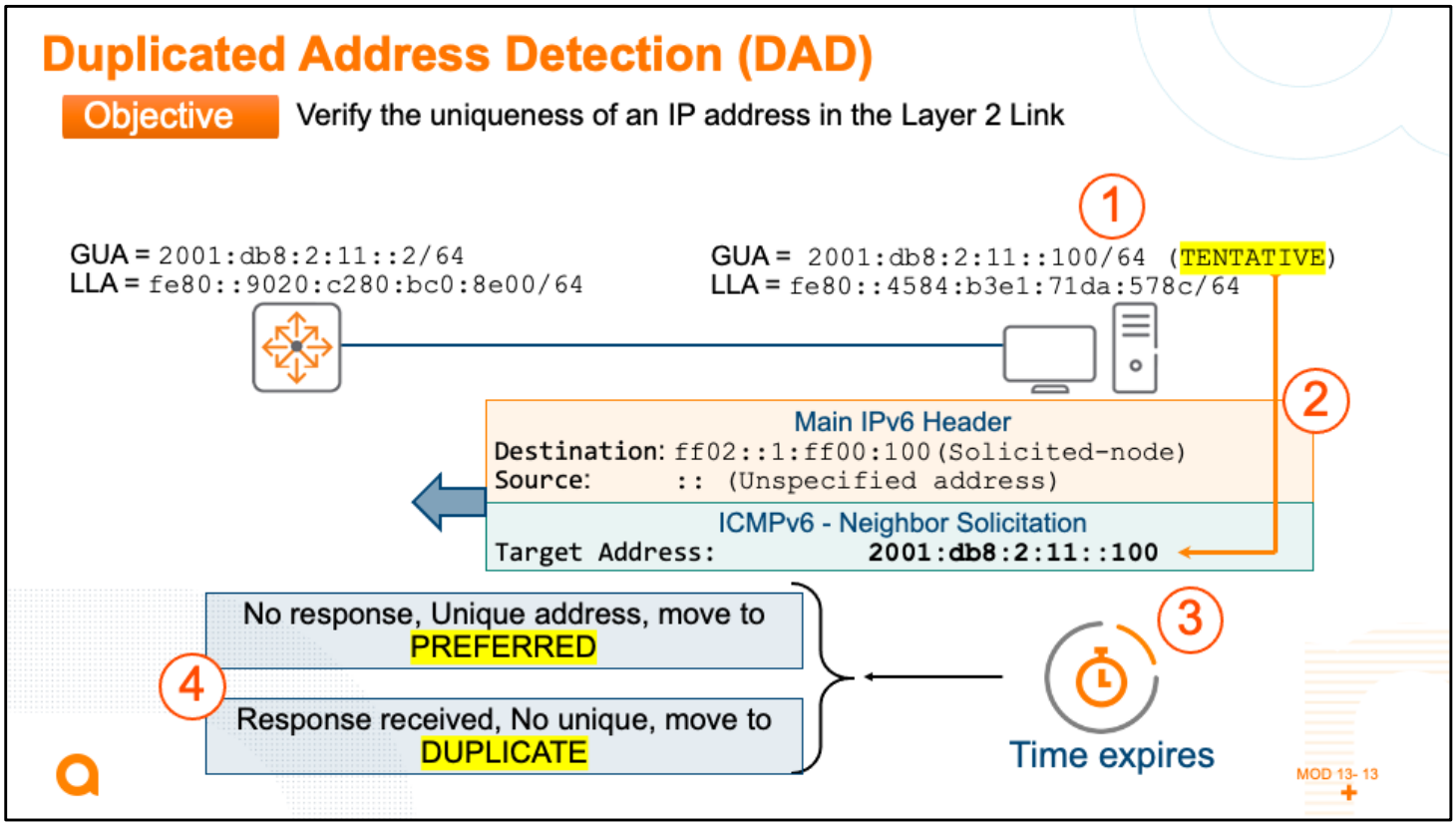
ff02::1, All-nodes multicast group with Link-Local scope

ff02::2, All-routers multicast group with Link-Local scope

ff02::1:ffc0:8e00, Multicast solicited-node for the Link-Local address

ff02::1:ff00:2, Multicast solicited-node for the Global address

ff02::1:ff00:0, Multicast solicited-node for the Global address subnet



Duplicate Address Detection (DAD) is used to verify the uniqueness of an IPv6 address. This process uses “ICMPv6-ND” Neighbor Solicitation and Neighbor Advertisement messages. The process is described as follows:

The station generates its own Global Unicast Address (manually or dynamically) on its Ethernet interface. The station assigns a TENTATIVE state to the address (no traffic can be generated using this IPv6 address yet).

The station sends out a Neighbor Solicitation message to see whether anyone else on the link is already using this link-local address. This message has the following characteristics:

- Source MAC address (not shown in the slide): MAC address of the Ethernet interface
- Destination MAC address (not shown in the slide): Multicast address 33:33:ff:00:01:00
- Source IPv6 address: Unspecified address ‘::’
- Destination IPv6 address: Solicited-node address ‘ff02::1:ff00:100’
- ICMPv6 Target address: 2001:db8:2:11:100 (Tentative address)

The station sets a timer to give an opportunity for another device using this tentative Global Unicast address to respond with a Neighbor Advertisement message.

If no message is received after this time expires, the Global Unicast address transitions from TENTATIVE to PREFERRED. If another device on the link has the same Global address, it responds with a Neighbor Advertisement message. The receipt of the NA message tells station that this address is already in use and cannot be assigned. The address is then moved to the DUPLICATE state, this address cannot be used to generate any traffic.

Question #1

```
# show ipv6 interface vlan1
Interface vlan1 is up
 Admin state is up
  IPv6 address:
    2001:db8:1:1::2/64 [VALID]
  IPv6 link-local address: fe80::b8d4:e780:141:3800/64 [VALID]
 IPv6 virtual address configured: none
 IPv6 multicast routing: disable
 IPv6 Forwarding feature: enabled
```

According to the output shown, What does the [VALID] flag means?

a. DAD process has not run yet but the system considers a Valid IPv6 address

b. There is no need to run the DAD process, the IPv6 address is VALID

c. DAD process has verified the uniqueness of the IPv6 address in the segment and has assigned the VALID flag

Knowledge Check

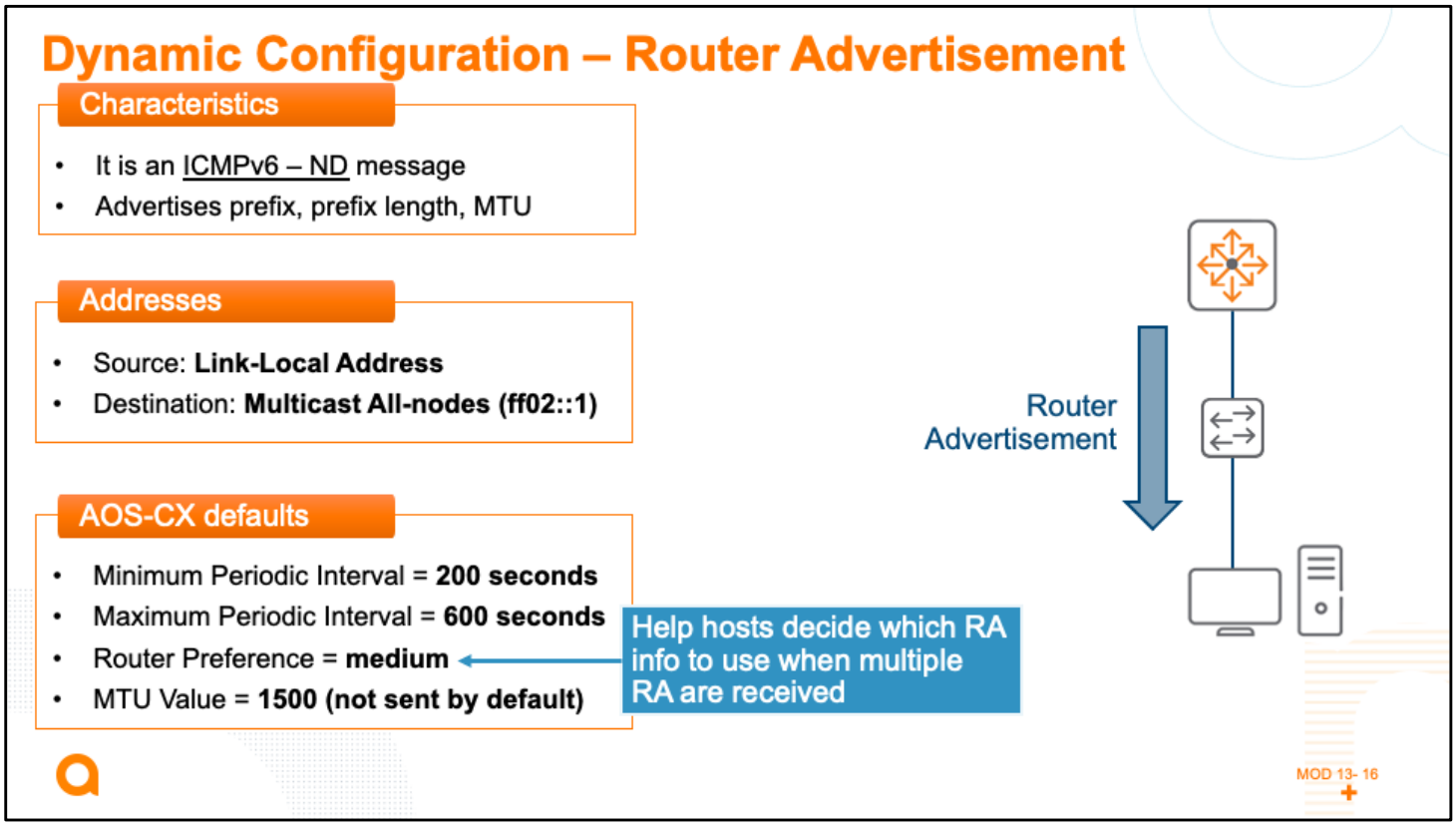
Correct answer:

C

aruba
a Hewlett Packard Enterprise company

Dynamic IPv6 Addressing

SLAAC, DHCPv6 Stateless and Stateful



The router advertisement message (RA) is key to understand how an IPv6 station will dynamically obtain or generate its IPv6 address.

Characteristics

The RA message is part of the ICMPv6 – Neighbor Discovery framework defined on RFC-4861. As the name indicates it is generated only on Routers or Multilayer switches. The RA message advertises specific parameters, like prefix, prefix length, and MTU.

Addressing

The source of the packet has the Link-Local address of the router interface and destination is ff02::1, which indicates Multicast All-nodes.

AOS-CX defaults

A router advertisement is sent every 200 seconds by default and it could take up to 600 seconds. To modify these parameters, use the commands ipv6 nd ra min-interval <3-1350> and ipv6 nd ra max-interval <4-1800>.

Router preference is medium. This parameter helps stations to decide which RA to use in environments where multiple Routers or Multilayer switches are present and sending Routing Advertisements. Values for this parameter are Low, Medium and High. To modify this

parameter, use the command ipv6 nd router-preference <low | medium | high>.

The MTU value is set to 1500, but the advertisement of this parameter in the RA is not included by default. To start advertising this parameter you can issue the command no ipv6 nd suppress-ra mtu.

To modify the MTU use the command ipv6 nd mtu <1280-65535>.

Dynamic Configuration – RA Address Allocation

Flags in RA suggest devices how to create their Global Unicast Address

RA Address Allocation Method	A - Flag	O - Flag	M - Flag
Method 1: SLAAC	1 (set)	0 (not set)	0 (not set)
Method 2: SLAAC and stateless DHCPv6	1 (set)	1 (set)	0 (not set)
Method 3: Stateful DHCPv6	0 (not set)	N/A	1 (set)

MOD 13- 17

The most important characteristic of a Router Advertisement packet is the Flag section. Flags helps a station to decide which of the three available methods shall be used to obtain an IP address. This is shown in the figure and described below.

Method1: For Stateless Address Autoconfiguration the A flag (Autoconfiguration flag) is set. Thus, a station can obtain its GUA address, prefix-length, and default-gateway. Other IP parameters cannot be obtained from this method.

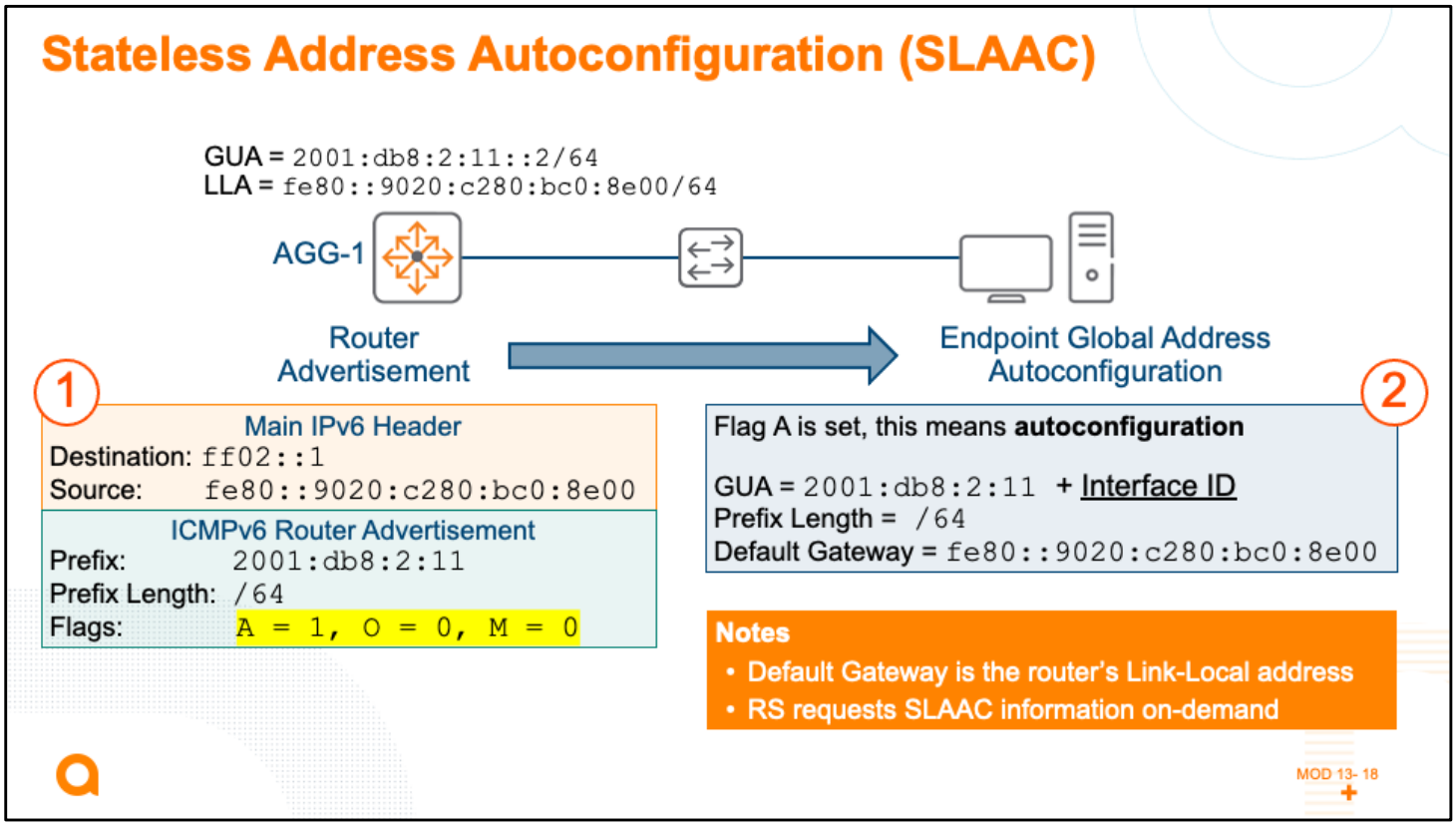
Method2: SLAAC and stateless. The A flag and the O flag (Other flag) are set. Thus, a station can obtain its GUA address, prefix-length, and default-gateway from the SLAAC process. Other parameters are obtained from a DHCPv6 server, which complements any missing information, such as a DNS, Domain, or NTP server.

Method3: With Stateful DHCPv6, the M flag (managed-config flag) is set. A DHCPv6 server delivers all information needed to the station and keeps a state (record) for the address.

You might wonder what happens if all three flags are set to 1. When both the O and M flags are set to 1, the O flag becomes irrelevant. The device then uses stateful DHCPv6 for all its configuration information (except for the default gateway). If the A flag is also set to 1, many operating systems (such as Windows) configure an additional address using SLAAC. The result is that the device may have an address it created using SLAAC and another address

obtained from a stateful DHCPv6 server.

Note. Stateful DHCP is a service that keeps the state of the leases. It can also provide Global Unicast Addresses. However, Stateless DHCP servers do not maintain lease states. It is only used to offer extra parameters (Not Global Unicast Addresses) to endpoints such as Domain, NTP or TFTP. You will learn more about these concepts in the following pages.



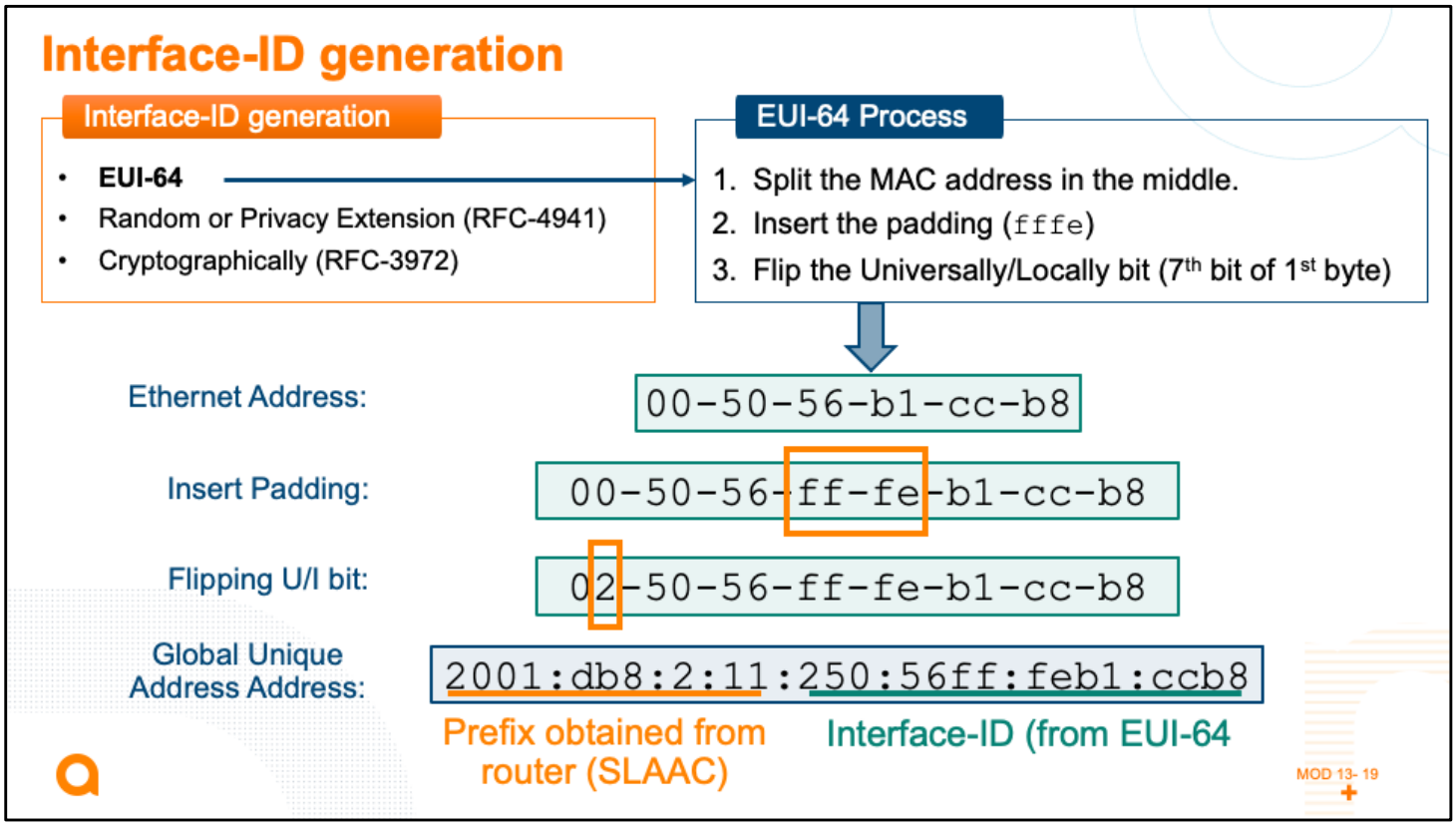
The slide shows the SLAAC process:

The Router Advertisement RA message is encapsulated in an IPv6 header with a destination IPv6 address of ff02::1 - the all-IPv6 devices multicast. The source IPv6 address is AGG-1's link-local address - fe80::9020:c280:bc0:8e00. The link-local address was automatically configured on AGG-1. The IPv6 Next Header field contains the value 58 in decimal. This indicates that the encapsulated data is an ICMPv6 message. The A flag is set to 1, while the O and M flags are set to 0. The A flag set to 1 suggests that receiving devices should use the prefix in the RA message to generate a global unicast address (SLAAC). Notice that this message also includes the prefix (2001:db8:2:11) and the length (/64).

The station receives the RA message and uses the source IPv6 address of the RA, fe80::9020:c280:bc0:8e00, as its default gateway address. The A flag is set to 1, which indicates that the station should use the prefix to create a global unicast address:

- Prefix = 2001:db8:2:11
- Interface ID = <depends on the Operating system and method used (see next section)>

Note. As previously mentioned, RA messages are sent periodically – every 200 seconds with AOS-CX. This reduces overhead, but it would not be good for users to wait 200 seconds for an address. To speed address assignment, stations can send a Router Solicitation (RS) message to request SLAAC information on demand. Routers respond with an RA message when they receive an RS.



What about the Interface-ID? How is this information is generated? This important piece of an IPv6 address is generated using one of the following methods:

EUI-64: Uses the MAC address to generate 64 bits of the Interface-ID.

Random or Privacy extension (RFC-4941): Uses MD5 hash algorithm to randomly create 64-bits. This is the default method used in Windows.

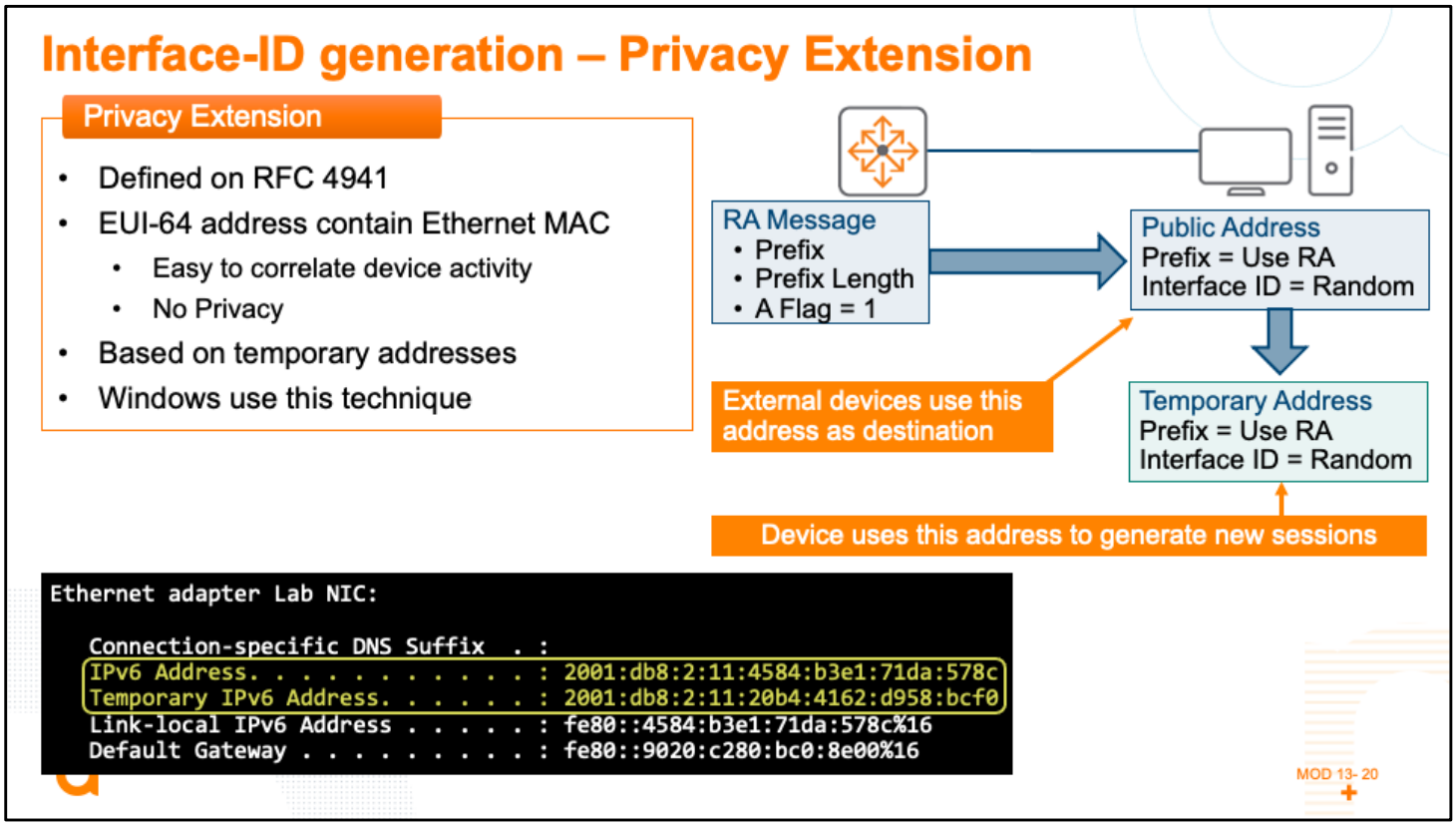
Cryptographically (RFC-3972): Uses a couple of hash values to generate the 64-bits of the Interface-ID and it has a security parameter (Sec) that determines its strength against brute-force attacks. This is the default method used in MAC-OSX.

In this course you will focus on the first method, EUI-64. The modified EUI-64 is a concatenation of the 24-bit OUI with the U/L OUI bit flipped, a 16-bit value of FFFE, and the 24-bit device identifier. This process is accomplished in three steps:

After converting the MAC address to binary, the MAC address is split in the middle, with the 24-bit OUI on the left and the 24-bit device identifier on the right.

FFFE is inserted between the OUI and the device identifier. FFFE is an IEEE-reserved value which indicates that the EUI-64 address was generated from a 48-bit MAC address.

Flip the Universally/Locally (U/L) bit, also known as the Local/Global bit - the seventh bit of the first byte. This is used to determine whether the address is universally or locally administered.



Addresses generated using EUI-64 method contain an embedded interface identifier, which remains constant over time. When a fixed identifier is used in multiple contexts, it is possible to correlate seemingly unrelated activity using this identifier. The correlation can be performed by:

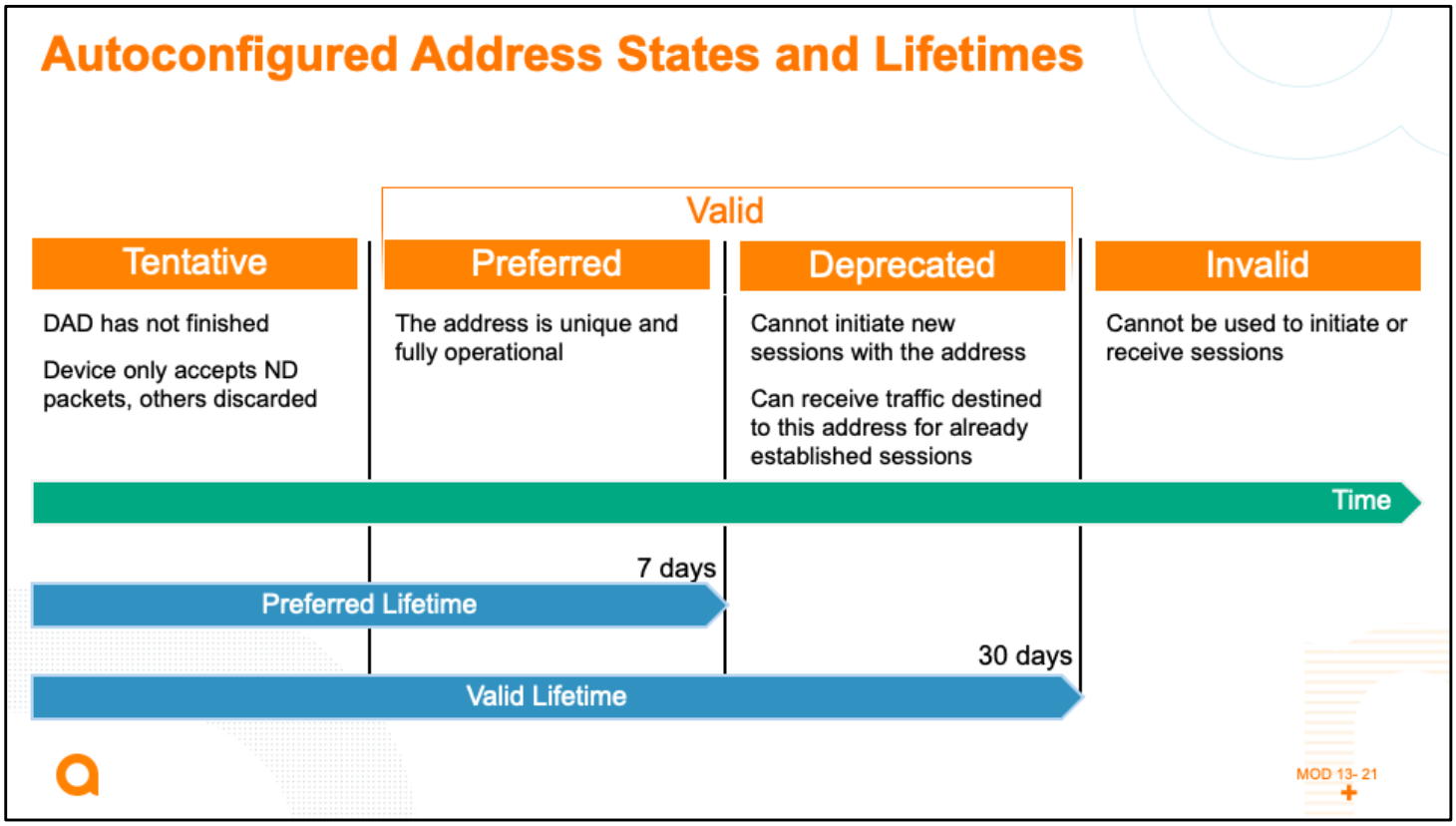
An attacker who is in the path between the node in question and the peer(s) to which it is communicating, and who can view the IPv6 addresses present in the datagrams.

An attacker who can access the communication logs of the peers with which the node has communicated. Since the identifier is embedded within the IPv6 address, which is a fundamental requirement of communication, it cannot be easily hidden.

Privacy Extension, defined on RFC-4941, offers a solution to randomly generate the Interface Identifier using MD5 hash algorithm to randomly create 64-bits. This technique will generate two IPv6 addresses:

Public address: Used by external devices to communicate with the host. Typically, the duration of this address is 30 days.

Temporary address: This address has a short Lifetime and it is used as the source address when originating connections. Typically, the duration of this address is 7 days.



Tentative address: The uniqueness of the address is in the process of being verified. A tentative address is not considered to be assigned to an interface. An interface discards received packets addressed to a tentative address but accepts Neighbor Discovery packets related to Duplicate Address Detection for the tentative address.

Valid address: The address is a preferred or deprecated address. A valid address can be the source or destination address of a packet. RA messages include the amount of time that an address remains in the valid and preferred states. The Valid Lifetime must be greater than or equal to the Preferred Lifetime.

Preferred address: The interface address has been verified as unique. The device can send and receive traffic using this address. New connections can be initiated using a preferred address as the source address. RA messages include the time period that an address can remain in the preferred state.

Deprecated address: The address assigned to an interface is still valid, but usage is discouraged. A deprecated address should no longer be used as a source address in new communications, but packets sent from or to deprecated addresses are delivered as expected. A deprecated address can continue to be used as a source address in existing communications where changing to a preferred address might cause a problem with specific upper-layer activity, such as an existing TCP connection.

Invalid address: A valid address becomes invalid when its Valid Lifetime expires. Invalid addresses should not appear as the destination or source address of a packet.

The amount of time addresses remain valid depends on 2 timers:

Preferred Lifetime: The length of time a valid address is preferred until it becomes deprecated. When the Preferred Lifetime expires, the address becomes deprecated.

Valid Lifetime: The length of time an address remains in the valid state. The Valid Lifetime must be greater than or equal to the Preferred Lifetime. When the Valid Lifetime expires, the address becomes invalid.

SLAAC Configuration

AOS-CX Command:

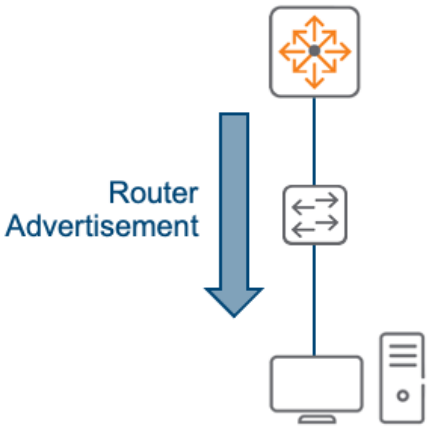
```
interface <interface-id>
 ipv6 nd prefix <prefix/prefix-length>
 no ipv6 nd suppress-ra
```

Example:

```
interface vlan11
 ipv6 address 2001:db8:2:11::2/64
 ipv6 nd prefix 2001:db8:2:11/64
 no ipv6 nd suppress-ra
```

Verification commands:

```
show ipv6 nd interface <interface-id>
show ipv6 nd interface <interface-id> prefix
show ipv6 nd global traffic
debug ipv6ra {all | packet | timer}
```



MOD 13- 22

In AOS-CX the configuration of SLAAC can be accomplished using the commands shown in the figure. When you configure an address with the ipv6 address command, a Link Local address is automatically created.

After you configure the ipv6 nd prefix command, the RA includes the specified prefix.

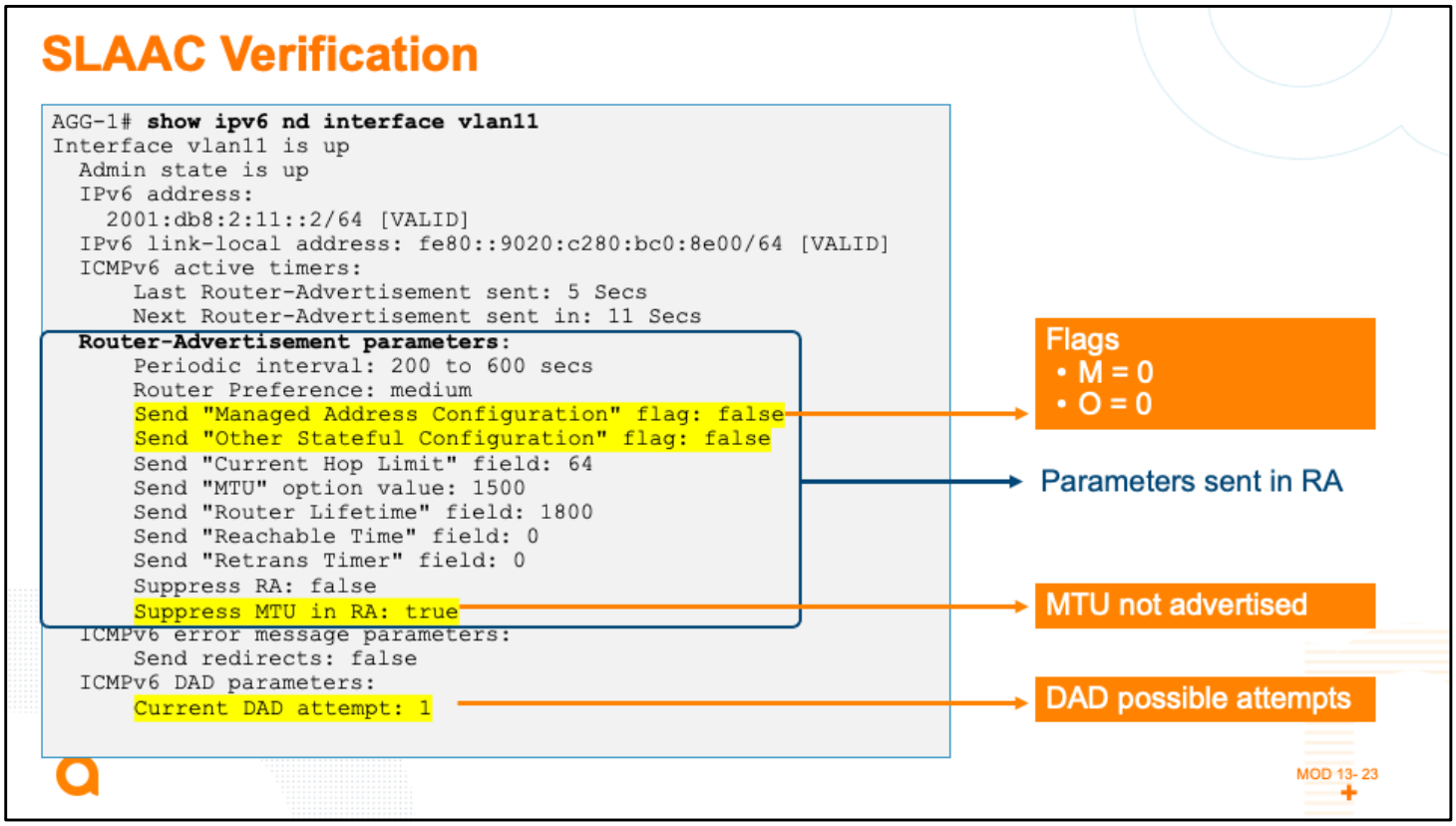
Use no ipv6 nd suppress-ra to allow RA advertisements, which are disabled by default in AOS-CX.

To verify the configuration, you can use the following commands:

show ipv6 nd interface <interface-id> prefix

show ipv6 nd global traffic

debug ipv6ra {all | packet | timer}

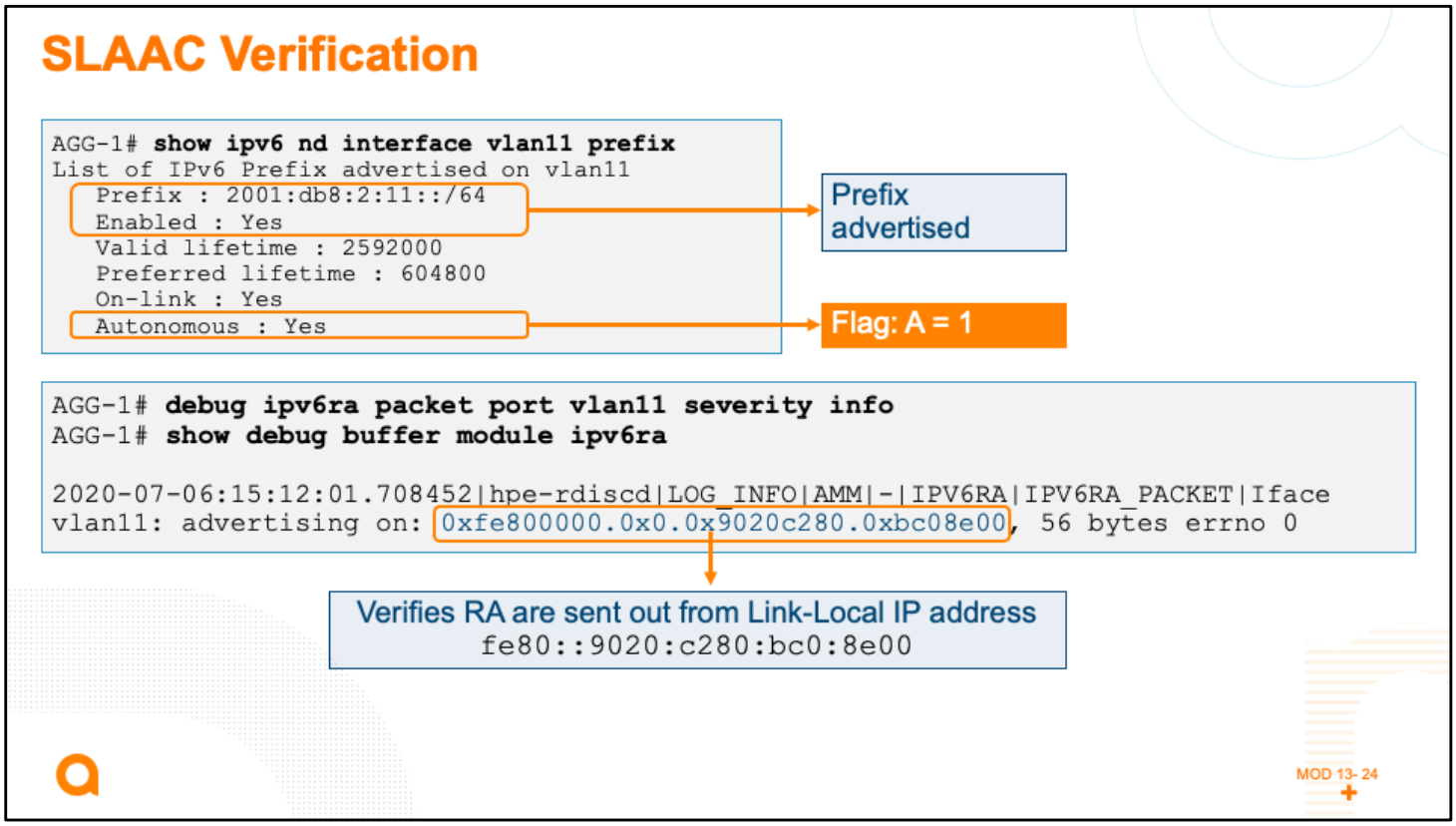


The command show ipv6 nd interface displays important information about the RA message:

Flags M = 0 and 0 = 0

MTU is not advertised by default in AOS-CX. This does not represent any problem if the packets generated by the sources are 1500 bytes or smaller. If not, the router discards the packet and sends an ICMPv6 “packet too big” message to the source.

DAD can only run one time; this avoids unnecessary processing.



The command show ipv6 nd interface <interface-id> prefix displays the following information:

Prefix: This is the /64 prefix advertised by the router.

Valid lifetime: This is the length of time an address remains in the valid state. The Valid Lifetime must be greater than or equal to the Preferred Lifetime. When the Valid Lifetime expires, the address becomes invalid.

Preferred lifetime: This is the length of time a valid address is preferred until it becomes deprecated. When the Preferred Lifetime expires, the address becomes deprecated.

On-Link: Defined on L flag. When set to 1, this indicates that the prefix can be used for on-link determination; when it is not set, the advertisement makes no statement about on-link or off-link properties of the prefix. The default is 1.

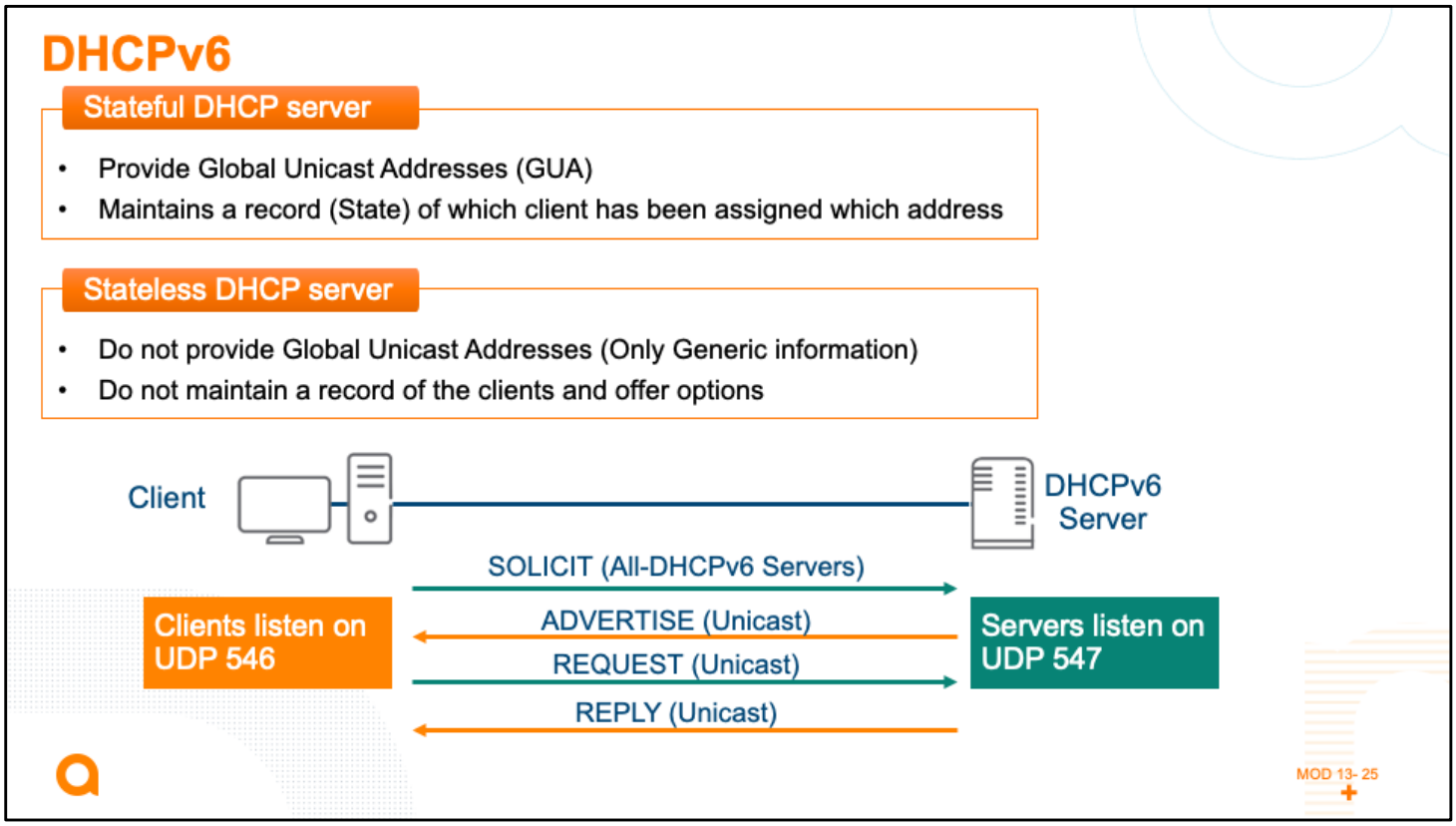
Autonomous: Indicates the A flag.

Note. On-link means that a packet can be sent directly to a device in the same subnet without being forwarded through a router. This may seem like an odd flag – it seems obvious that if another device is on the same subnet, a device can communicate directly to it.

The most common use case of using the L flag for operational advantages is with Wi-Fi deployments. In a wireless configuration using a central controller, you likely want client

sessions to always go through the controller, especially for guest networks. There is no reason for a wireless client to talk to another wireless client on the same Layer 2 domain, so telling the host to send all traffic through the router is not a bad strategy. Reducing the amount of multicast to broadcast traffic is important in wireless – to limit interruptions for all hosts on a given network. This conserves the radio spectrum and increases overall performance of the Wi-Fi network. In this example the L flag must not be set (0).

The second output refers to a debug command. In this case you can verify that the RA is periodically sent out from the VLAN11. The output also specifies the Link-Local address used.



DHCPv6 is a valid solution to offer IP information to endpoints. There are two types of DHCPv6 servers (stateful and stateless).

First, know that a “state” is simply a record that the DHCPv6 server creates when it offers an IPv6 address to an endpoint. DHCPv6 servers that maintain a state are of course referred to as stateful servers. If not, they are considered stateless.

Characteristics of a DHCPv6 stateful server:

Can offer Global Unicast Addresses to endpoints and maintain a state

Can offer other parameters (options) to the endpoint such as TFTP, NTP or Domain.

A DHCPv6 server will not provide a Default-Gateway to the endpoint - that parameter is gleaned from the RA message

An AOS-CX switch can be a DHCPv6 stateful server

Characteristics of a DHCPv6 stateless server:

It cannot provide or offer Global Unicast Addresses to endpoints, so no record or state can be created.

Offers only other parameters (options) rather than Default-Gateway, IPv6 address and prefix.

An AOS-CX switch can be a DHCPv6 stateless server offering only DNS Server and DNS Suffixes.

There are many DHCP-related messages, but regardless of the DHCPv6 server type, four primary messages are used:

SOLICIT: DHCPv6 clients use a SOLICIT message to locate servers.

ADVERTISE: A server sends an ADVERTISE message in response to a client's SOLICIT message to indicate that it is available for DHCPv6 service.

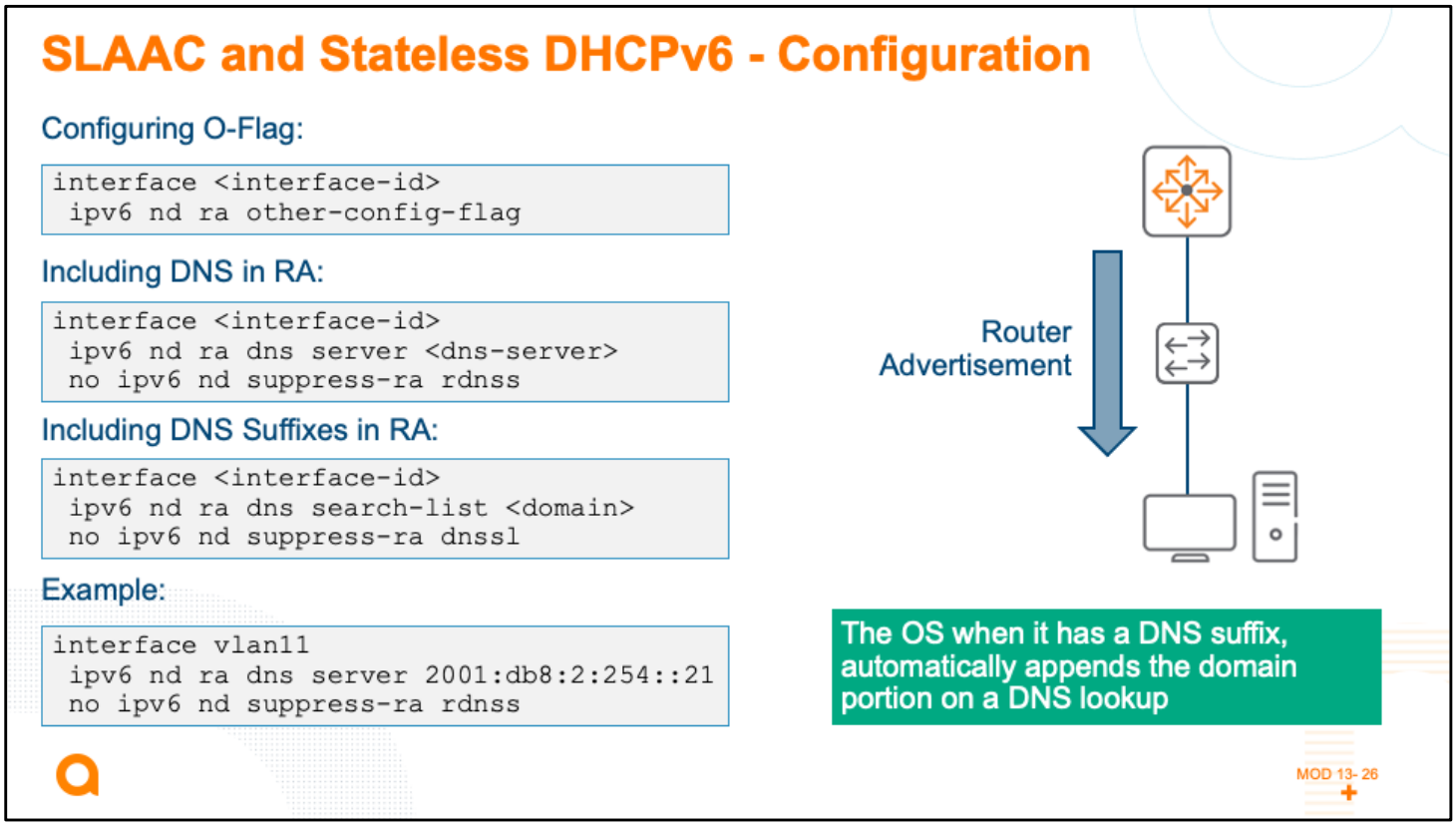
REQUEST: A client sends a REQUEST message to request configuration parameters, including IPv6 addresses, from a specific DHCPv6 server.

REPLY: A server sends a REPLY message containing assigned addresses and configuration parameters in response to a client's REQUEST message. A server also sends a REPLY message in response to a SOLICIT message when the Rapid Commit Option is used. A server can also send a REPLY message in response to an INFORMATION-REQUEST, CONFIRM, RELEASE, or DECLINE message.

UDP port 546: DHCPv6 servers and relays send messages to clients using UDP destination port 546. Clients listen for DHCPv6 messages on UDP port 546.

UDP port 547: DHCPv6 clients send messages to servers and relay agents using UDP destination port 547. Servers and relay agents listen for DHCPv6 messages on UDP port 547.

Note. By default, a DHCPv6 client-server exchange requires four messages (SOLICIT, ADVERTISE, REQUEST, and REPLY). The rapid-commit option reduces this from four to two messages (SOLICIT and REPLY). The client sends a SOLICIT message to the all-DHCP relay agents and the server multicast address, to request an address and other information. This message includes an indication that the client is willing to accept an immediate REPLY message from the server and commit to the addresses provided. This is done by adding the rapid-commit parameter in the ipv6 server dhcp server interface command.



In AOS-CX the configuration of SLAAC and Stateless DHCPv6 can be accomplished using the following commands:

interface <interface-id>

ipv6 nd ra other-config-flag → Set the O-Flag to 1

ipv6 nd ra nds server <dns-server> → DNS server is advertised in the RA

no ipv6 nd suppress-ra rdnss → By default the RA will not include the DNS server, even if it is configured.

ipv6 nd ra dns search-list <domain> → DNS suffix is advertised in the RA

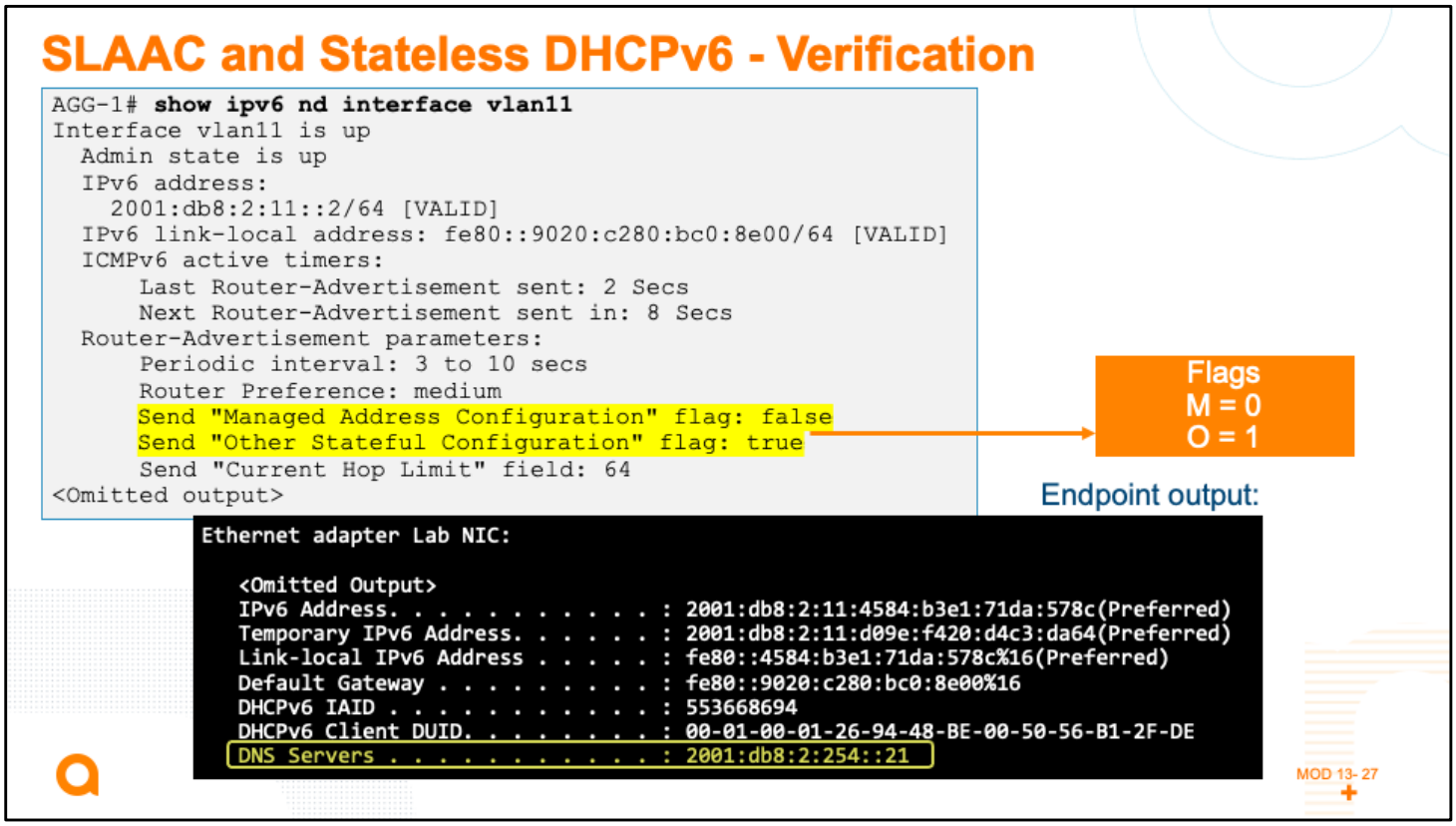
no ipv6 nd supress-ra dnssl → Enable the advertising of the DNS Suffix in the RA.

Example of configuration:

interface vlan11

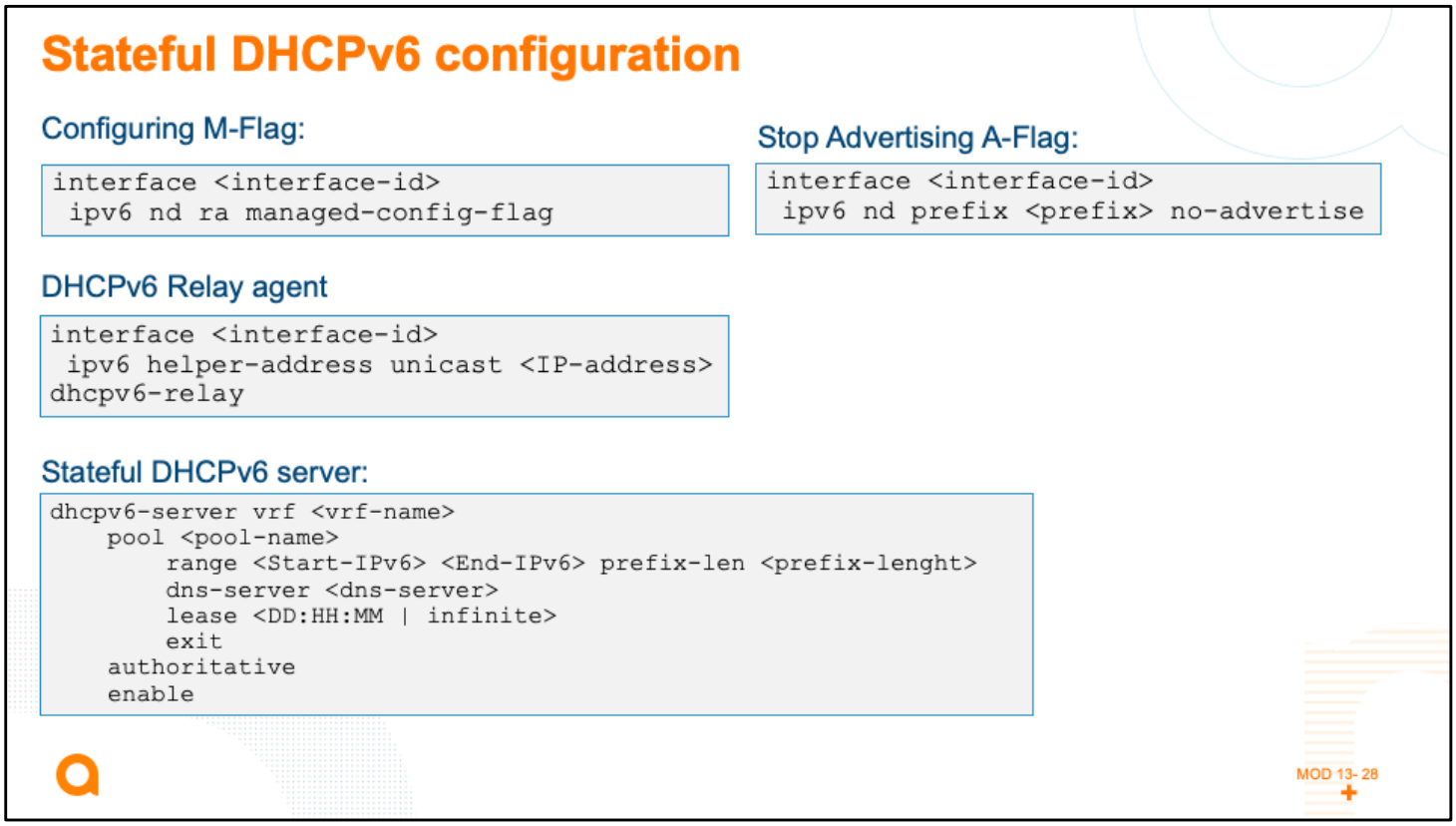
ipv6 nd ra dns server 2001:db8:2:254::21

no ipv6 nd suppress-ra rdnss



The command show ipv6 nd interface displays important information about the RA message. In this example, Flags M = 0 and 0 = 1. This indicates that other information can be found from a Stateless DHCPv6.

The output from the client shows a DNS server.



In AOS-CX the configuration of DHCPv6 server can be accomplished using the following commands:

interface <interface-id>
ipv6 nd ra managed-config-flag → Set the M-Flag to 1
ipv6 nd prefix <prefix> no-advertise → Stop advertising A-Flag

In case you want to use the Switch as a relay agent use the following command:

interface <interface-id>
ipv6 helper-address unicast <IP-address
dhcpv6-relay

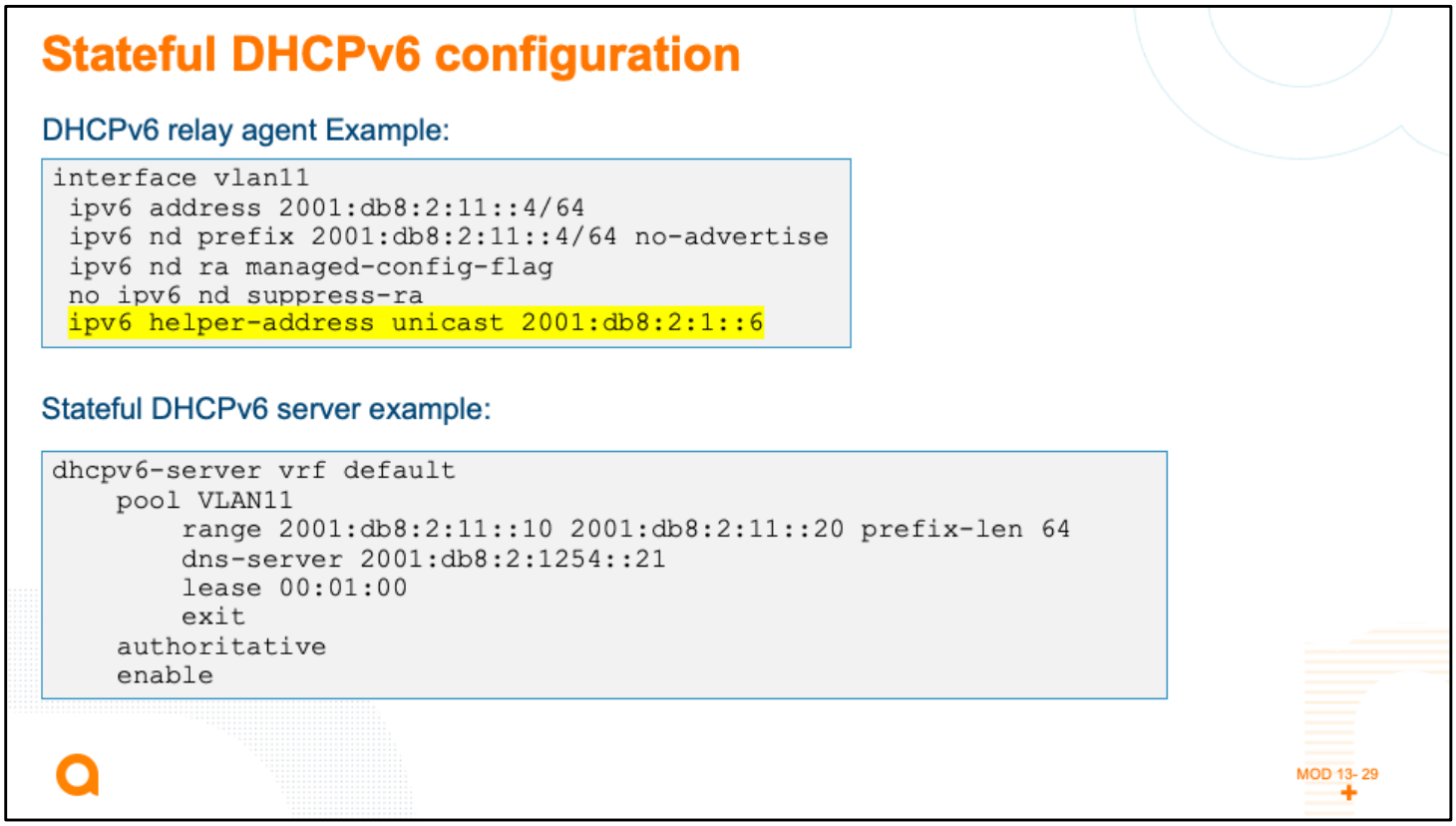
The switch can act as a Stateful DHCPv6 server, you can accomplish this task by setting the DHCPv6 pool using the following commands:

dhcpv6-server vrf <vrf-name>. → Defines the DHCP server and attach it to a specific VRF
pool <pool-name> → Defines the Pool

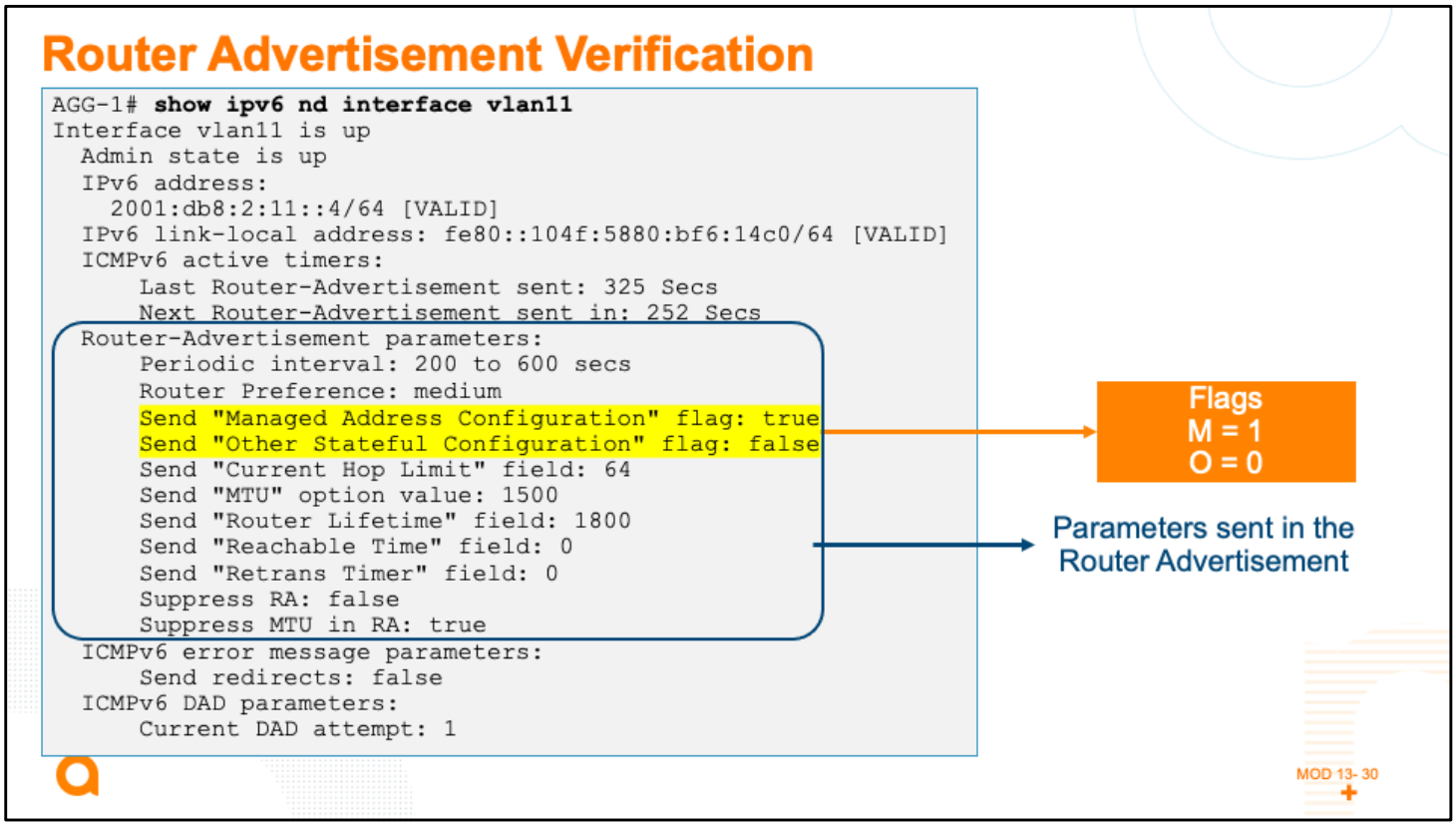
range <Start-IPv6> <End-IPv6> prefix-len <prefix-lenght> → Defines the range of IPv6 address and the length of the prefix

dns-server <dns-server>	→ Set a DNS server
lease <DD:HH:MM | infinite>	→ Set the lease time
authoritative for issuing addresses.	→ Makes the server the sole authority
enable	→ Enable the feature

Note. If you configure the DHCPv6 server as authoritative on the current VRF, then it is the sole authority for the network on the VRF. It responds to client solicit messages with advertise messages having a priority/preference value set to 255 (the maximum), instead of 0 (the minimum). Clients always choose the DHCPv6 server with the highest priority/preference value. If two DHCPv6 servers send an advertise message with the same priority/preference value, then the client picks one and discards the other.

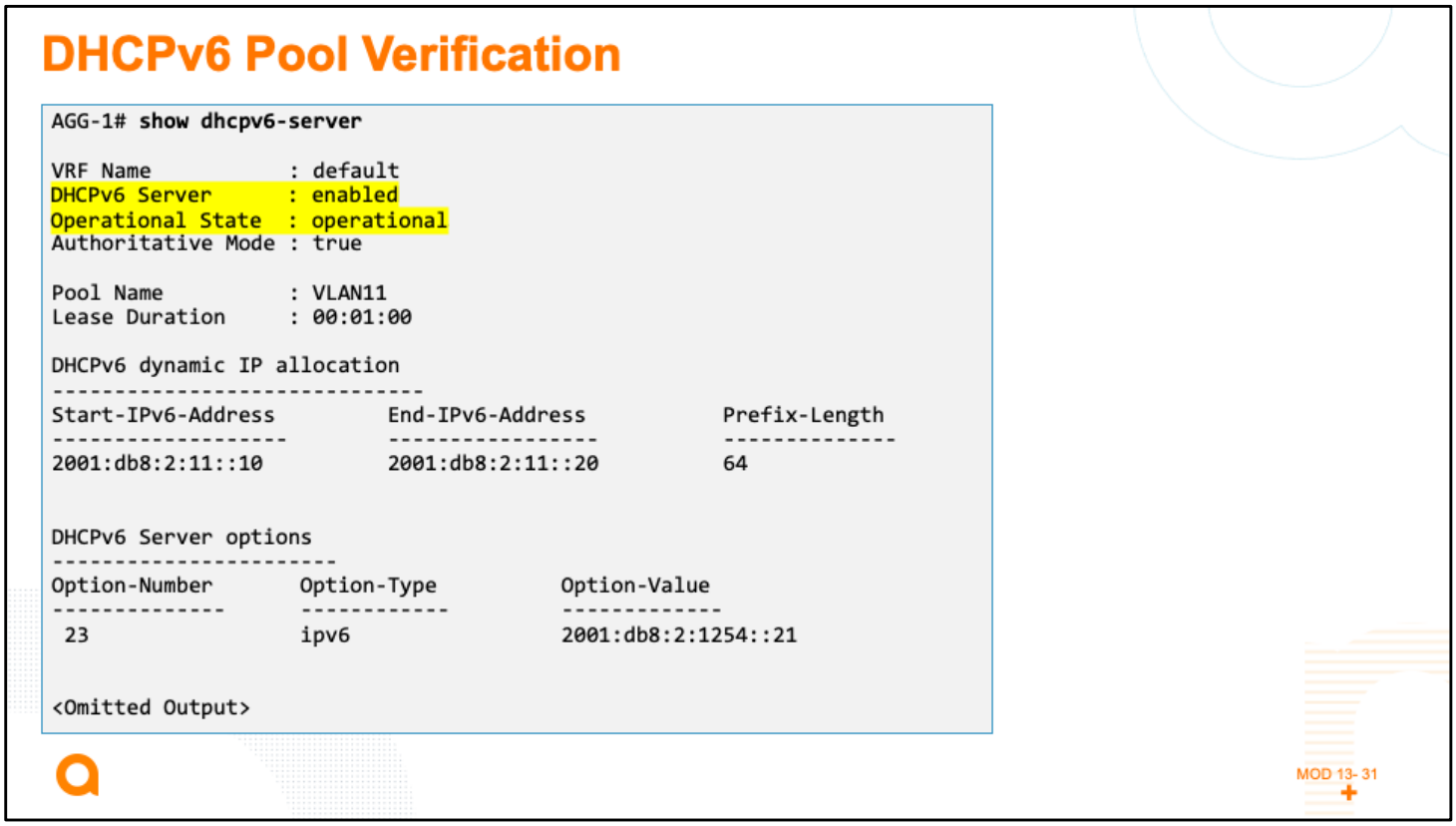


The figure shows examples of configuring a relay agent and Stateful DHCPv6 server.

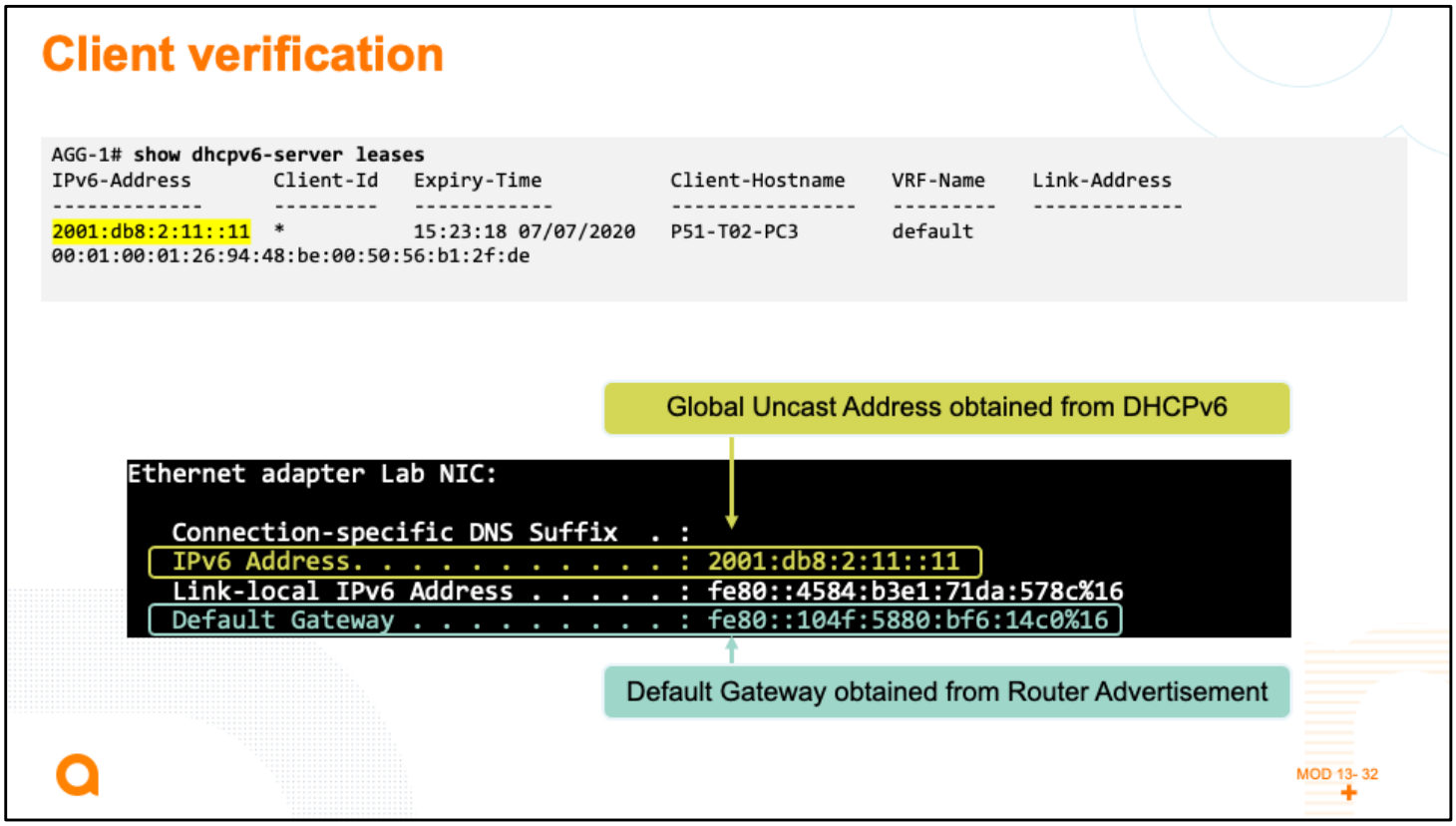


The command show ipv6 nd interface displays important information about the RA message:

Flags M = 1 and 0 = 0, this indicates that all information will be obtained from DHCPv6 server.



The show dhcpv6-server command displays the information and state of the created address pool.



The show dhcpv6-server leases command displays the leases offer to clients.

The output from the endpoint shows that it correctly receives the IP address from the DHCPv6 server. Remember that the Default-Gateway is obtained from the Router Advertisement.

Question #2

In an IPv6 address, what does the DEPRECATED flag means?

a. The IPv6 address is Valid but no new sessions can be generated using this IPv6 address as source

b. The IPv6 address is not Valid but new sessions can be generated using this IPv6 address as source

c. Traffic sent to this IPv6 address can still be received

d. Traffic sent to this IPv6 address cannot still be received



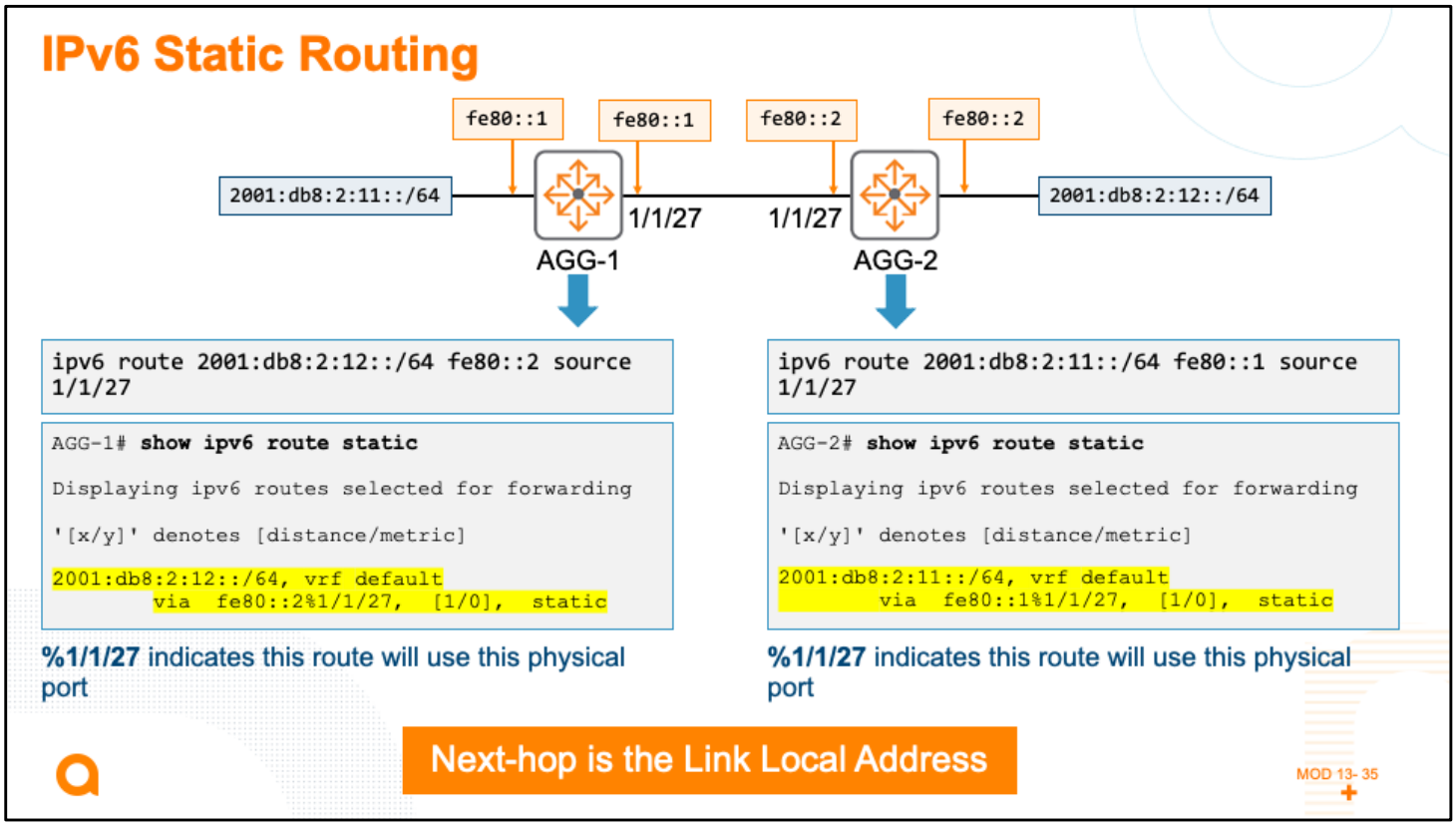
Knowledge Check



Correct answer:

A and C

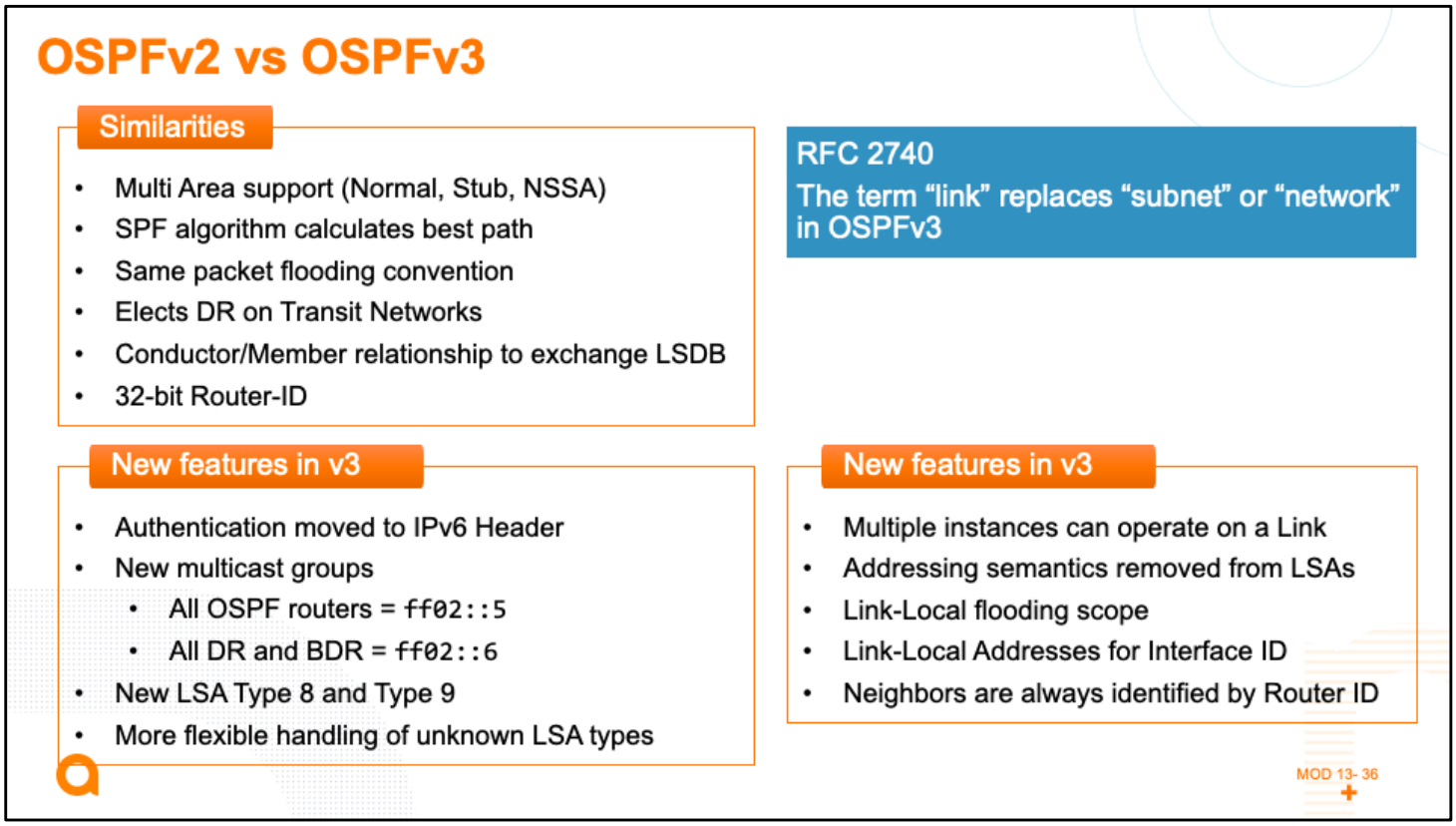

aruba
a Hewlett Packard
Enterprise company
IPv6 Routing
Static and OSPFv3



Static routing in IPv6 follows the same general rules as IPv4 - you must specify the target prefix/prefix-length and Next Hop.

In this example, notice that both interfaces on AGG-1 use the same Link-Local address (fe80::1) this is possible since the address has link scope. This implies that there is a possibility that the Next-Hop (fe80::2) exits in both links. To avoid confusion in IPv6, it is recommended to specify the interface from which you will reach the Next-Hop.

From the show ipv6 route static command output, you can see that the interface was considered as part of the routing decision. This information is represented by (%1/1/27).



The Open Shortest Path First (OSPF) routing protocol has been updated to version 3 to accommodate IPv6 addressing conventions. The basic operation of OSPF and structure of OSPF has not changed. A backbone area connects to different areas within the AS. Other unchanged aspects include the convention of flooding packets, the process of designated router election, the conductor/member relationship, the Shortest Path First (SPF) algorithm. The size of the router, area, and link-state IDs is still 32 bits.

Version 3 differs in the following aspects:

It operates on a per-link basis rather than per-subnet because IPv6 links can contain more than one subnet. An interface to a link can have more than one IPv6 address. In fact, a single link can belong to multiple subnets, and two interfaces attach to the same link but belonging to different IPv6 subnets can still communicate.

More than one instance of version 3 can operate on a link at the same time. There are applications in which multiple OSPF routers can be attached to a single broadcast link but should not form a single adjacency among them. For example, suppose four OSPF routers are attached to an Ethernet link. Routers 1 and 2 belong to one OSPF domain and routers 3 and 4 belong to a different OSPF domain. There should be adjacencies between 1 and 2 and between 3 and 4, but not, for instance, between 1 and 3. You can accomplish this kind of separation of adjacencies with OSPFv2 by manipulating authentication, but it is not ideal. OSPFv3 allows for multiple instances per link by adding an Instance ID to the OSPF packet header to distinguish instances. An interface assigned to a given Instance ID will drop OSPF

packets whose Instance ID does not match.

The addressing semantics have been removed from most packets and Link-State Advertisements (LSAs). OSPFv3 Router and Network LSAs do not carry IP addresses. A new LSA is defined for that purpose. This has some scaling advantages.

A link-local flooding scope has been added. OSPFv3 retains the domain or AS and area flooding scopes of OSPFv2 but adds a link-local flooding scope. A new LSA – the Link LSA – has been added, for carrying information that is only relevant to neighbors on a single link. This LSA has link-local flooding scope meaning it cannot be flooded beyond any attach router.

IPv6 link-local addresses (fe80::/10) are used as the interface ID. OSPFv3 uses a router's link-local IPv6 as the source address, and as next-hop addresses.

New LSAs have been created to increase efficiency and improve scalability.

The authentication functions have been moved to the IPv6 header packet. IPv6 has, using the authentication extension header, a standard authentication procedure. Because of this, OSPFv3 has no need for its own authentication of OSPFv3 packets; it just uses IPv6 authentication.

Neighbors are always identified by Router ID. OSPFv2 neighbors on broadcast links are identified by their interface addresses, whereas neighbors on other types of links are identified by RID. OSPFv3 removes this inconsistency, so all neighbors on all links types are identified by RID.

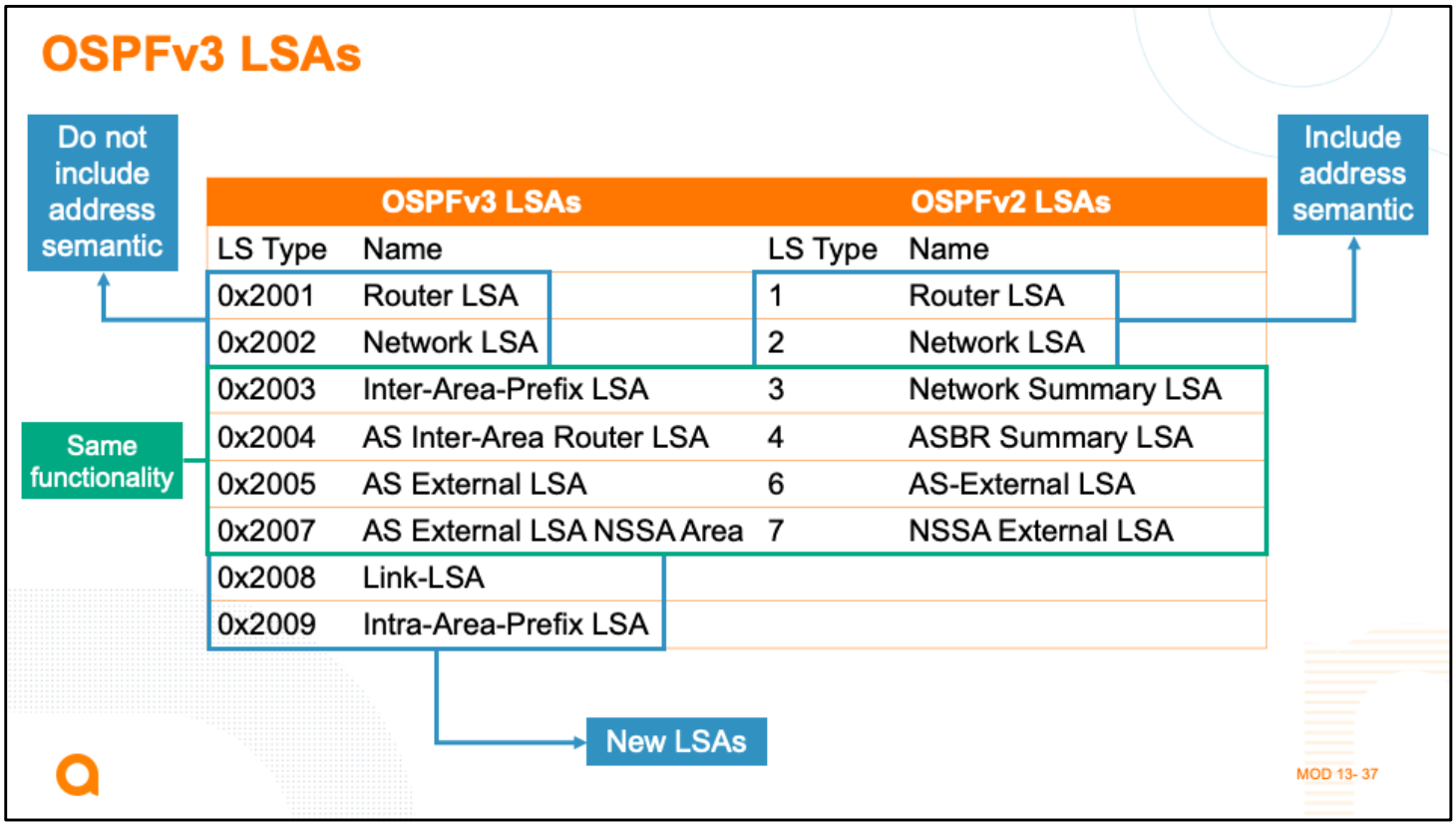
More flexible handling of unknown LSA types. Where OSPFv2 always discards unknown LSA types, OSPFv3 can either treat them as having link-local flooding scope or store and flood them as if there are understood, while ignoring them in their own SPF algorithms. This can result in easier network changes and easier integration of new capabilities in OSPFv3 than in OSPFv2

All routers receive packets from the Multicast group ff02::5.

Designated Routers and their backups receive packets from the multicast group ff02::6

Note. IPv6 uses the term "link" to indicate "a communication facility or medium over which nodes can communicate at the link layer". Interfaces connect to links. Multiple IP subnets can be assigned to a single link, and two nodes can talk directly over a single link, even if they do not share a common IP subnet (IPv6 prefix). For this reason, OSPF for IPv6 runs per-link instead of the IPv4 behavior of per-IP-subnet. The terms "network" and "subnet" used in the IPv4 OSPF specification should be replaced by link. Likewise, an OSPF

now connects to a link instead of an IP subnet.



OSPFv3 LSAs		OSPFv2 LSAs	
LS Type	Name	LS Type	Name
0x2001	Router LSA	1	Router LSA
0x2002	Network LSA	2	Network LSA
0x2003	Inter-Area-Prefix LSA	3	Network Summary LSA
0x2004	AS Inter-Area Router LSA	4	ASBR Summary LSA
0x2005	AS External LSA	6	AS-External LSA
0x2007	AS External LSA NSSA Area	7	NSSA External LSA
0x2008	Link-LSA		
0x2009	Intra-Area-Prefix LSA		

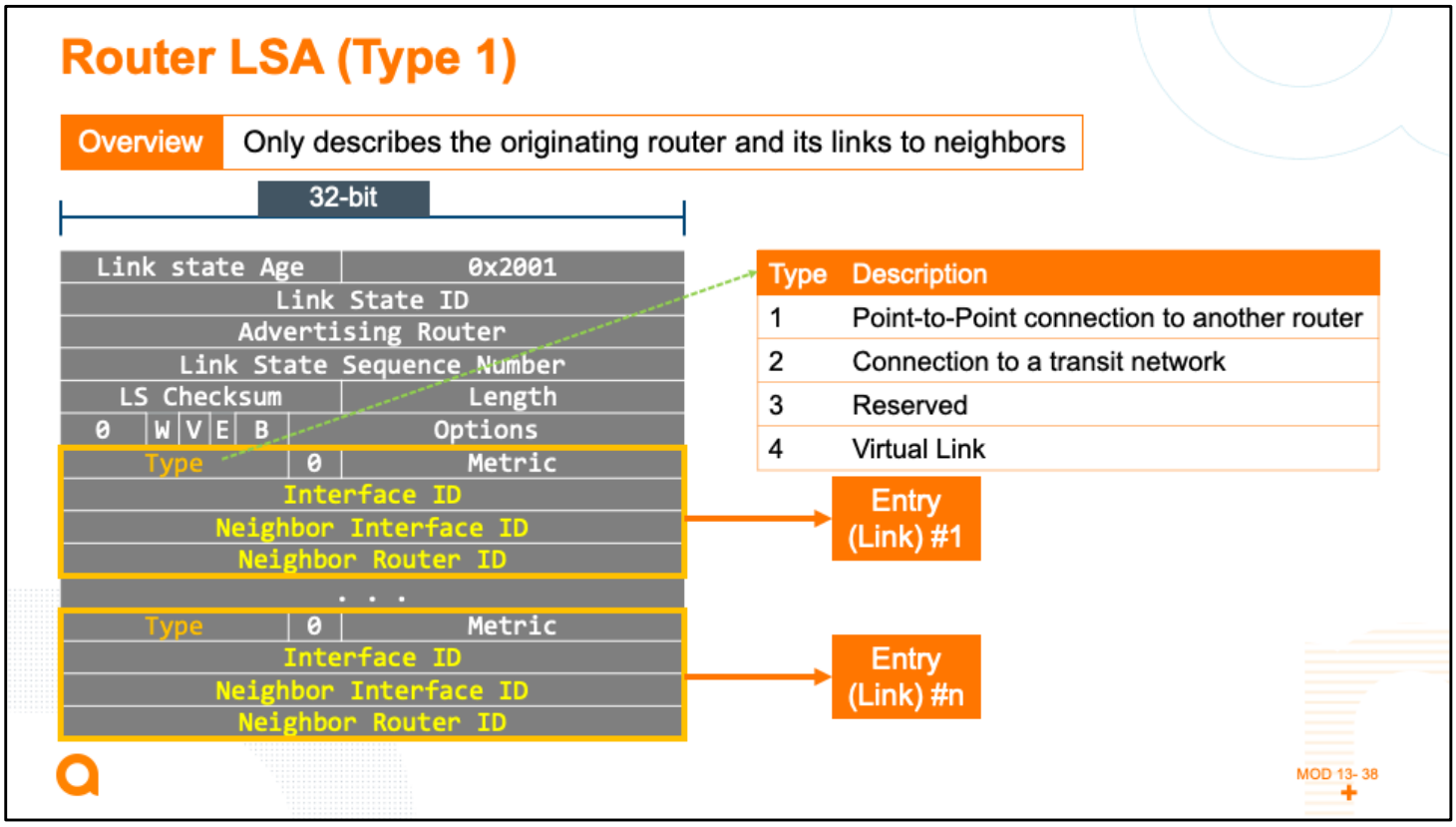
Although Router and Network LSAs have the same names, there is a significant difference in how OSPFv3 and OSPFv2 Router and Network LSAs function. Specifically, OSPFv3 Router and Network LSAs do not advertise prefixes. This is an important improvement in the scaling of the protocol. These LSAs are primarily to represent the router as node on the SPF tree. When a Router or Network LSA is flooded, there is an assumption that a topological change has taken place and all routers in the area, on receipt of the LSA, rerun SPF. But because OSPFv2 routers also use these LSAs to advertise their connected subnets, if a subnet changes, the LSA must also be flooded to advertise the change. Even though something like an address change does not affect the SPF topology, the reception of a Router or Network LSA triggers an SPF run anyway. This can be particularly problematic for edge or access routers that have many sub links that change regularly.

OSPFv3 removes the prefix advertisement function from Router and Network LSAs and puts it in the new Intra-Area Prefix LSA. Now Router and Network LSAs only represent the router's node information for SPF and are only flooded if information pertinent to the SPF algorithm changes. If a prefix changes, or the state of a stub link changes, that information is flooded in an Intra-Area Prefix LSA that does not trigger an SPF run.

Another difference between Router and Network LSAs between OSPFv2 and OSPFv3 is in the exchange of some information that is only relevant to directly connected neighbors. OSPFv2 puts this information in Router and Network LSAs; and although only the directly connected neighbors care about the information, it is flooded with the LSAs throughout the

area. OSPFv3 take this neighbor-specific information and put it in the new Link LSA, which has link-local flooding scope. Although minor, the introduction of the Link LSA does add an efficiency improvement over OSPFv2.

Inter-Area Prefix, Inter-Area Router and Type-7 LSAs, although their names have changed, have the same function as OSPFv2 Network Summary, ASBR Summary and NSSA LSAs, respectively. AS-External and Group Membership LSAs have both the same names and the same functions in both OSPFv2 and OSPFv3.



As discussed in the previous section, prefix information is not included in the Router LSA as it is in its OSPFv2 counterpart. The OSPFv3 Router LSA only describes the originating router and its links to neighbors, for use in SPF calculations. Prefix information is carried in the Intra-Area Prefix LSA.

Options, although in a different position within the LSA format and a longer length (24 bits) than the OSPFv2 Options field, performs the same function of identifying optional capabilities.

Following the Options field is a set of fields that can appear multiple times, to describe each of the attached interfaces:

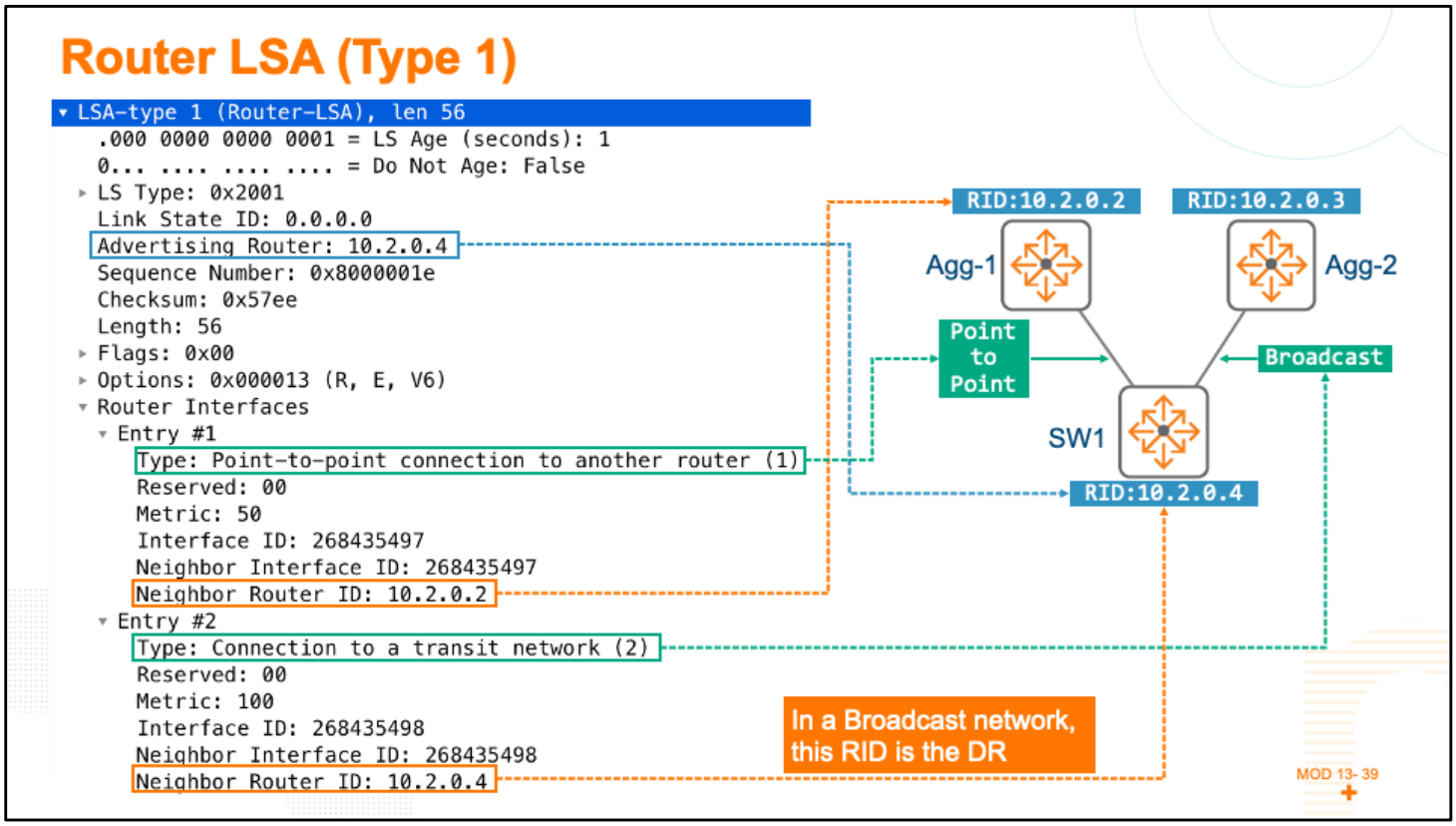
Type. Specifies the interface type. The table lists the possible interface type values.

Metric. Specifies the outbound cost of the interface.

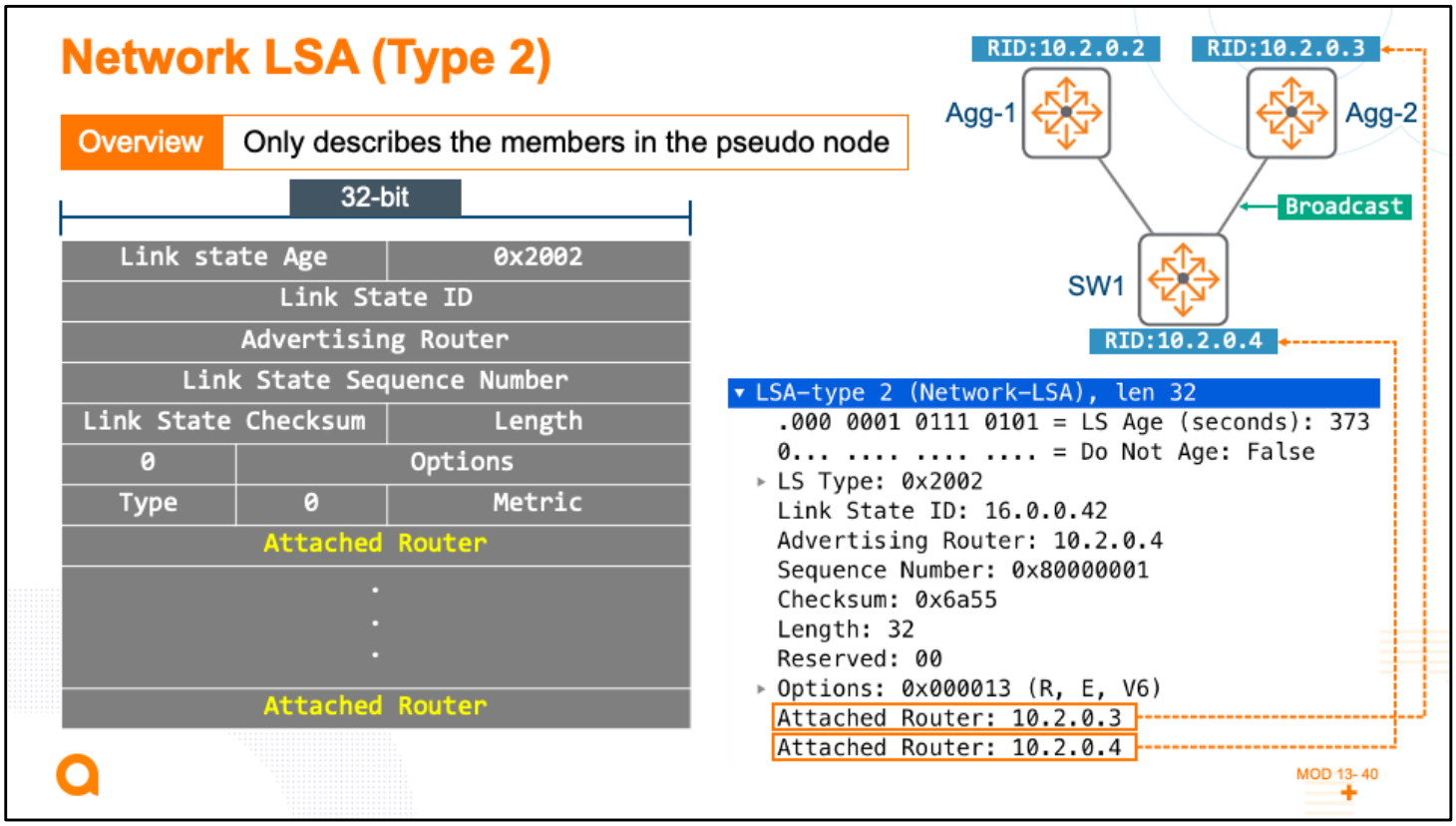
Interface ID. Is a 32-bit value distinguishing the interface from other interfaces on the originating router.

Neighbor Interface ID. Is the Interface ID advertised by neighbors on the link in their Hellos or in the case of Type 2 links, the interface ID of the Link's DR.

Neighbor Router ID. Is the neighbor's RID or in the case of type 2 links, the DR's RID.

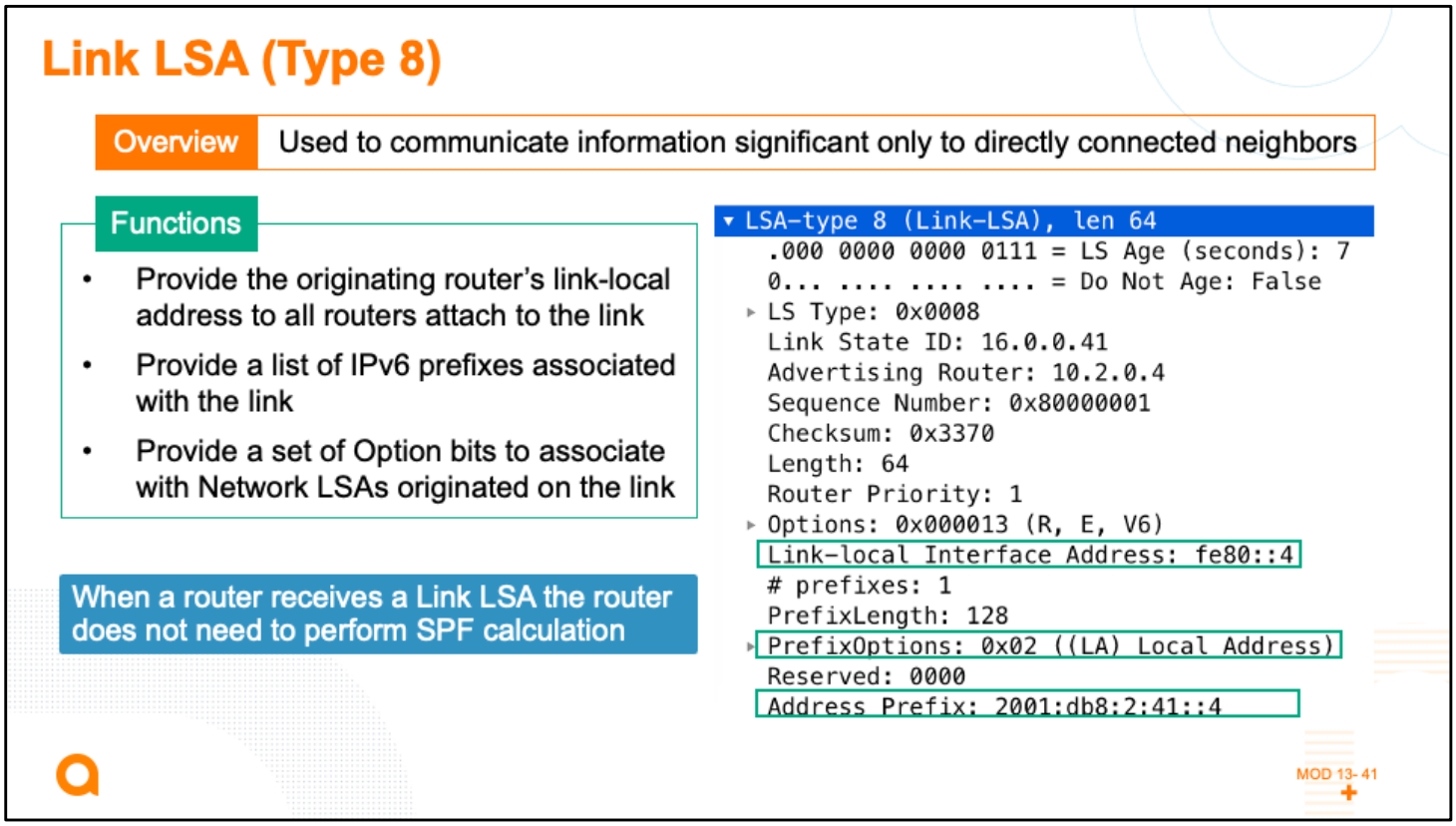


The wireshark capture shows a Router LSAs generated by SW1 which has been configured with the RID 10.2.0.4. This LSA describes the two connections to Agg-1 and Agg-2 respectively. The first connection is considered a Point-to-Point and then properly marked as Type 1, the LSA indicates that the Neighbor RID is 10.2.0.2, the RID used by Agg-1. The second connection is considered Broadcast or Transit network and is marked as Type 2, in this case the Neighbor RID is poiting to the DR that has been elected in the network, which in this case is SW1.



The OSPFv3 Network LSA is shown in the figure. Functionally it is identical to the OSPFv2 Network LSA (originated by the DR to represent a pseudo node). The only significant differences are the location of the Options field and the elimination of the Network Mask field, which has no meaning in IPv6. The main purpose of the Network LSA is to describe all the members that are located in the pseudo node.

In the Wireshark capture you can see that SW1 and Agg-2 are the two devices that are members of the pseudo node.



The Link LSA is used for communicating information that is significant only to two directly connected neighbors. The format of the Link LSA is shown in the next figure. A separate Link LSA is originated on each of a router's attached links belonging to an OSPFv3 domain, and a receiving router, because of the link-local flooding scope, never forwards the LSA to any other link.

The LSA performs three functions:

It provides the originating router's link-local address to all other routers attached to the link.

It provides a list of IPv6 prefixes associated with the link.

It provides a set of Option bits to associate with Network LSAs originated on the link.

Router Priority specifies the router priority assigned to the interface of the originating router.

Options specifies the options bits that the originating router would like to set in the Network LSA that will be originated for the link. This is the same 24-bit Option field carried in OSPFv3 Hello and Database Description (DD) packets, in a number of OSPFv3 LSAs.

Link-Local Prefix Address specifies the 128-bit link-local prefix of the originating router's interface attached to the link.

Number of Prefixes specifies the number of IPv6 prefixes contained in the LSA, as described by the following Prefix Length, Prefix Options and Address Prefix fields.

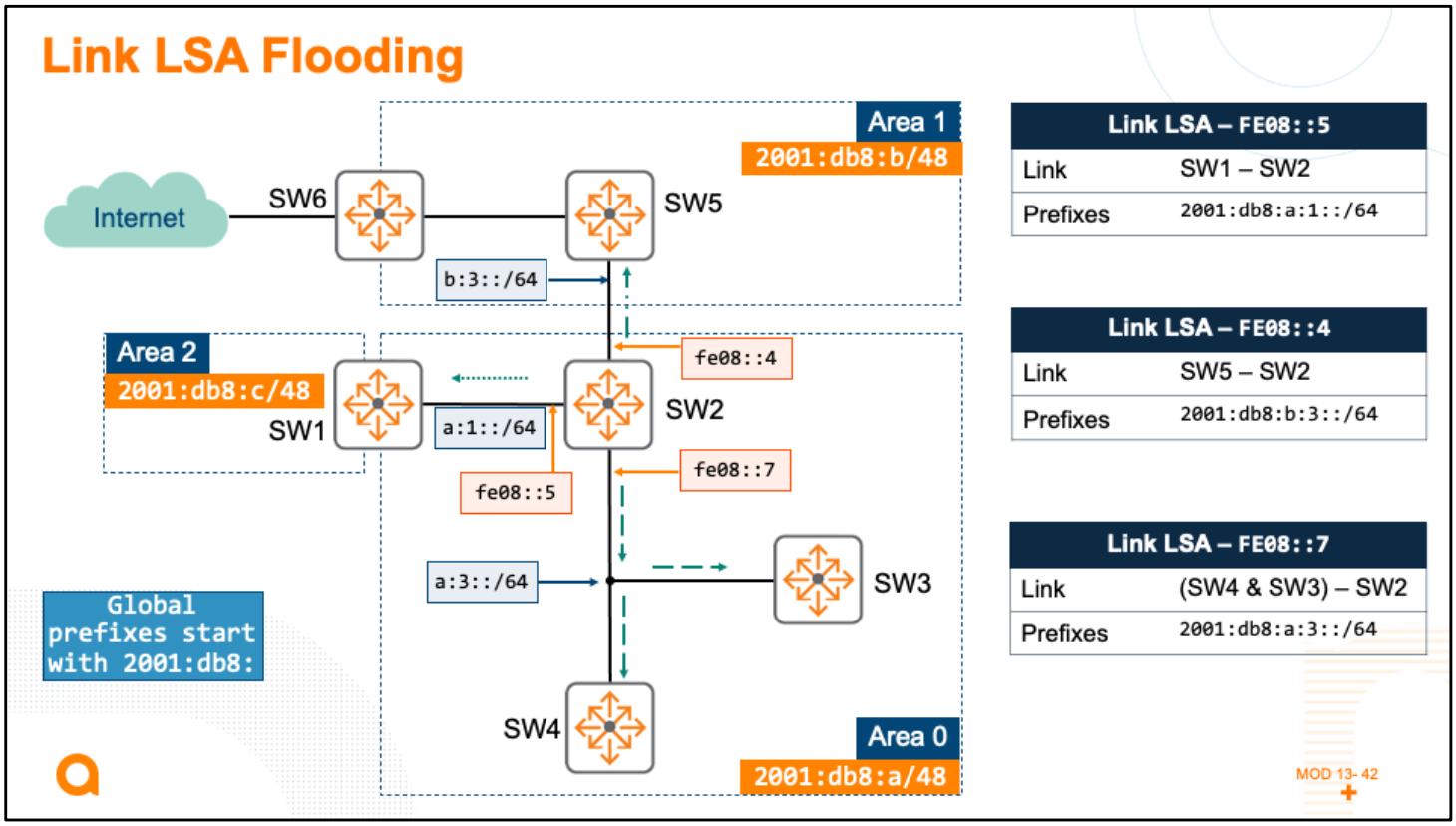
Prefix Length, Prefix Options and Address Prefix describe one or more IPv6 prefixes associated with the link by the originating router. This set of fields is used not only in the Link LSA, but also the Intra-Area-Prefix, Inter-Area Prefix and AS-External LSAs. The advertised prefix can be any length between 0 and 128. When the prefix is not an even multiple of 32-bits, it is padded out with zeros to fit the 32-bit boundaries of the Address Prefix field. The Prefix Length field specifies the length of the unpadded prefix, in bits. The Prefix Options field, shown in the table, specifies handling of the prefix during routing calculations.

The Propagate (P) bit is set on NSSA area prefixes that should be re-advertised at the NSSA area border.

The Multicast (MC) bit, when set, specifies that the prefix should be included in multicast routing calculations.

The Local Address (LA) bit, when set, specifies that the prefix is an interface address of the advertising router.

The No Unicast (NU) bit, when set, specifies that the prefix should be excluded from unicast route calculations.



The figure shows an OSPFv3 network with the standard OSPF areas: Area 0 (backbone) and Areas 1 and 2.

The figure shows the three “Link LSAs” that router SW2 sends, one for each of its interfaces. The “Link LSA” for the interface fe80::4, for example, includes a list of the network prefixes associated to that interface in this case: 2001:db8:b:3::/64

Each packet is flooded only on the local link. For example, the Link LSA for fe80::7 is sent through that interface. Switches SW3 and SW4 receive that Link LSA. The Link LSAs are sent by all routers regardless its OSPF role.

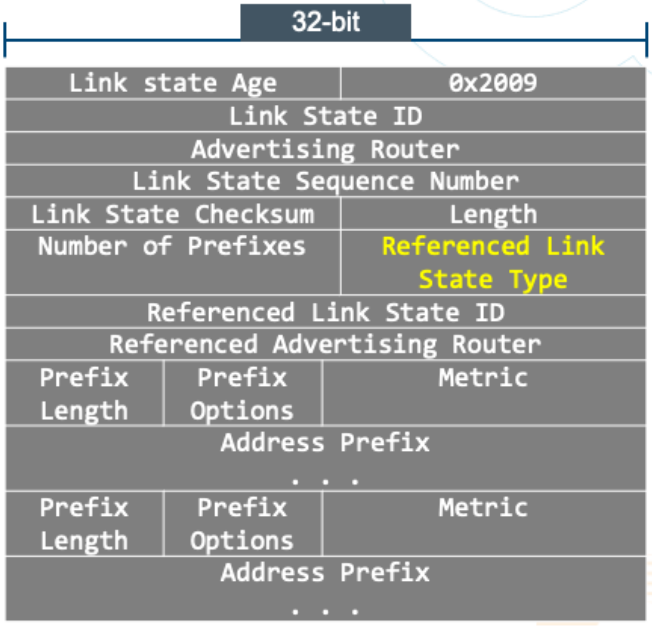
Notice that this LSA does not contain information about network topology, such as the metric (cost of link). Therefore, when the routers receive the Link LSA, it is not necessary to perform the SPF calculation.

Intra-Area Prefix LSA (Type 9)

Overview

- Advertise subnet address changes
  - In a stub network
  - In a transit network
- Receipt does not trigger SPF calculation
- Contains prefix info, such as Metric
- Area flooding scope

Referenced Link State Type	Associated LSA	Referenced Link State ID
1	Router LSA	Router ID of originating router
2	Network LSA	Interface ID of the link's DR



MOD 13- 43

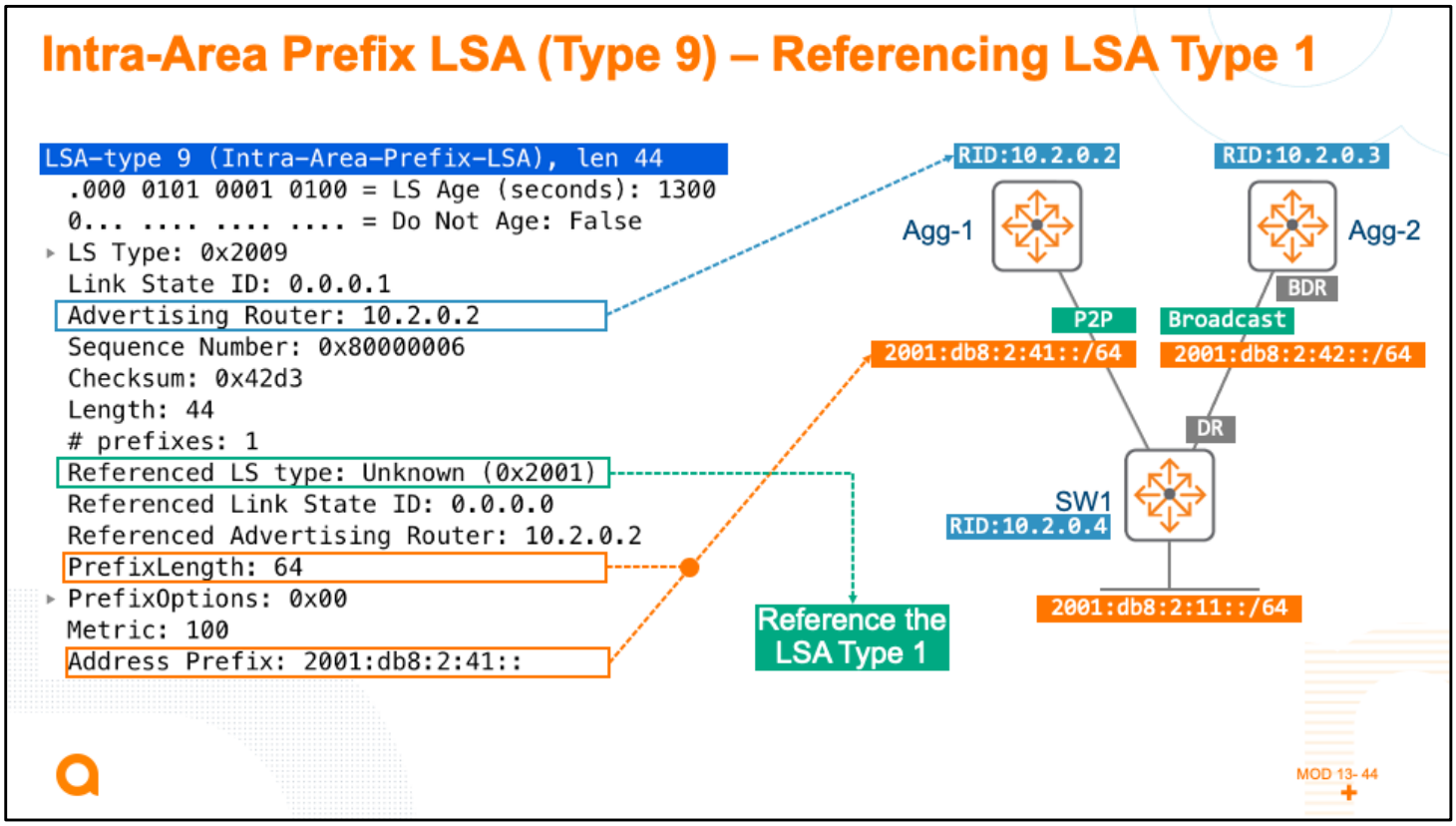
The Intra-Area Prefix LSA is shown in the figure. Attached prefixes, which in OSPFv2 are carried in Router and Network LSAs in OSPFv3 are carried in Intra-Area Prefix LSAs. Prefix changes – including additions and deletions – do not influence the branches of the SPF tree in any way. But OSPFv2 advertises prefixes in Router and Network LSAs, causing an SPF calculation in all routers in the area whenever a prefix change occurs. With OSPFv3, however, when a link or its prefix changes, the connected router originates an Intra-Area Prefix LSA to flood the information throughout the area. This LSA does not trigger an SPF calculation; the receiving routers simply associate the new prefix information with the originating router. Router and Network LSAs, in OSPFv3, serve only to provide topological information. As a result, this new LSA should make OSPFv3 significantly more scalable for networks with large numbers of frequently changing prefixes.

Number of Prefixes. Specifies the number of prefixes contained in the LSAs.

Referenced Link State Type, Referenced Link State ID, and Referenced Advertising Router. Identify the Router or Network LSA with which the contained prefixes should be associated.

If the prefixes are associated with a Router LSA, the Referenced Link State Type is 1, the Referenced Link State ID is 0 and the Referenced Advertising Router is the Router ID (RID) of the originating router. If the prefixes should be associated with a network LSA, the Referenced Link State Type is 2, the Referenced Link State ID is the Interface ID of the link's DR and the Referenced Advertising Router is the RID of the DR.

Each Prefix is then represented by a Prefix Length, Prefix Options, and Address Prefix field, as described previously for the Link LSA. Added to these three fields is the Metric field which is the cost of the prefix.



The Wireshark capture shows how the LSA-Type 9 originated by Agg-1 (10.2.0.2) includes the prefix 2001:db8:2:41::/64. When an OSPFv3 router receives this LSA it knows that the information contained must be mapped to the LSA Type 1 (Router LSA) that was generated by the same device (Agg-1).

SW1 should generate an Intra-Area Prefix LSA referencing the Router LSA (Type 1) describing the prefixes used in the stub network (2001:db8:2:11:/64) and the link to Agg-1 using the prefix 2001:db8:2:41::/64. The Wireshark capture in the next lines confirms this event:

LSA-type 9 (Intra-Area-Prefix-LSA), len 68

.000 0000 0000 0111 = LS Age (seconds): 7

0... .... = Do Not Age: False

LS Type: 0x2009

Link State ID: 0.0.0.1

Advertising Router: 10.2.0.4

Sequence Number: 0x80000016

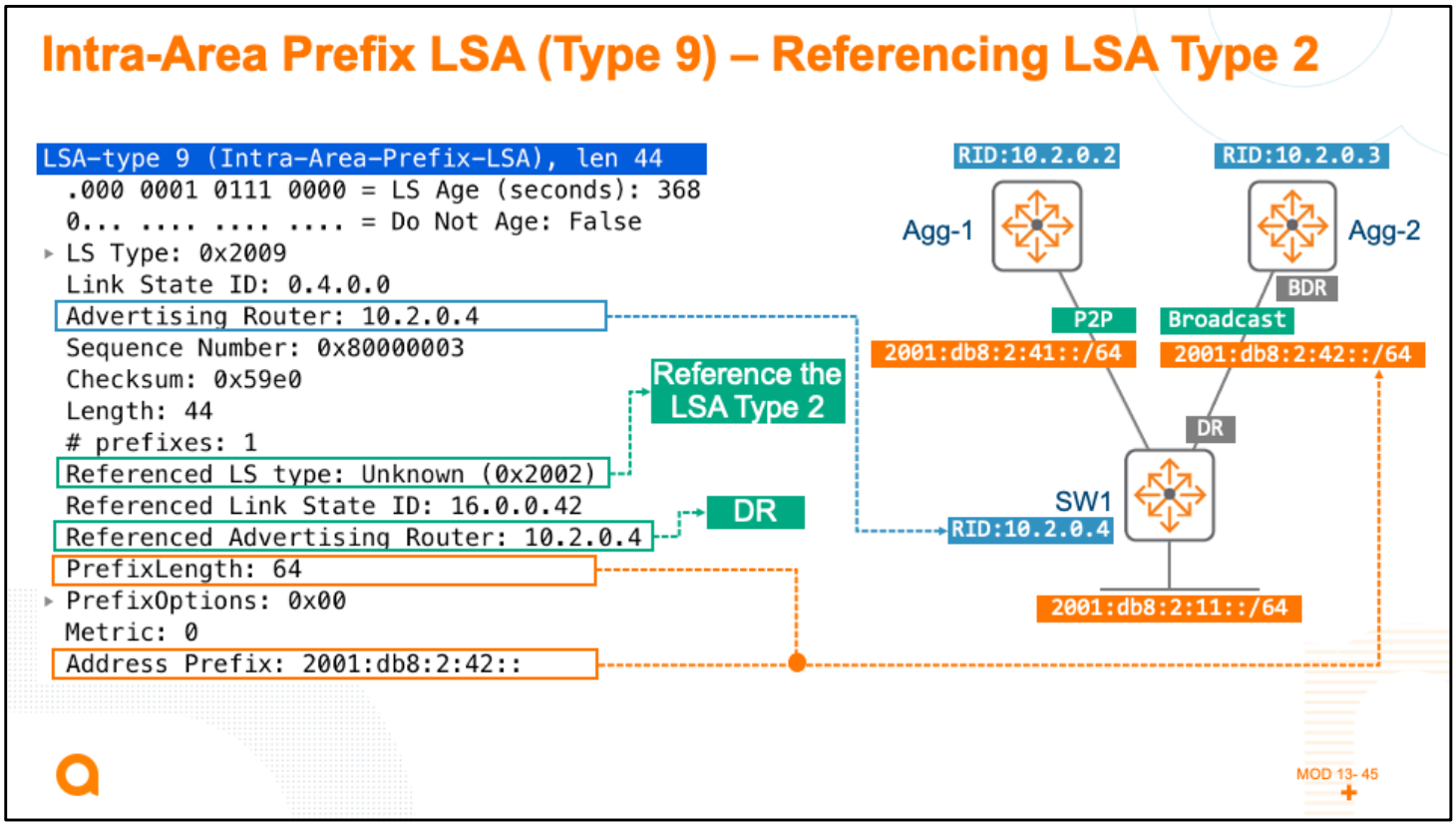
Checksum: 0x7566

Length: 68

prefixes: 2

Referenced LS type: Unknown (0x2001)
Referenced Link State ID: 0.0.0.0
Referenced Advertising Router: 10.2.0.4
PrefixLength: 64
PrefixOptions: 0x00
Metric: 100
Address Prefix: 2001:db8:2:11::
PrefixLength: 64
PrefixOptions: 0x00
Metric: 50
Address Prefix: 2001:db8:2:41::

The reason why the link to Agg-2 is not included is because this connection uses a network type broadcast, and the Intra-Area Prefix should reference the Network LSA (Type 2).



The Wireshark capture shows how the LSA-Type 9 originated by SW1 (10.2.0.4) includes the prefix 2001:db8:2:42::/64. When an OSPFv3 router receives this LSA it knows that the information contained must be mapped to the LSA Type 2 (Network LSA) and the DR is the information contained in the Referenced Advertising Router field (10.2.0.4 – SW1).

This capture probes how OSPFv3 has separated the semantic addressing from LSA Type 2.

Even though the figure does not show it, a similar LSA should be generated by SW1 describing the prefix used to connect to the stub network 2001:db8:2:11::/64. The following capture lines confirm this:

LSA-type 9 (Intra-Area-Prefix-LSA), len 44

.000 0100 1000 1001 = LS Age (seconds): 1161

0... .... = Do Not Age: False

LS Type: 0x2009

Link State ID: 0.2.0.0

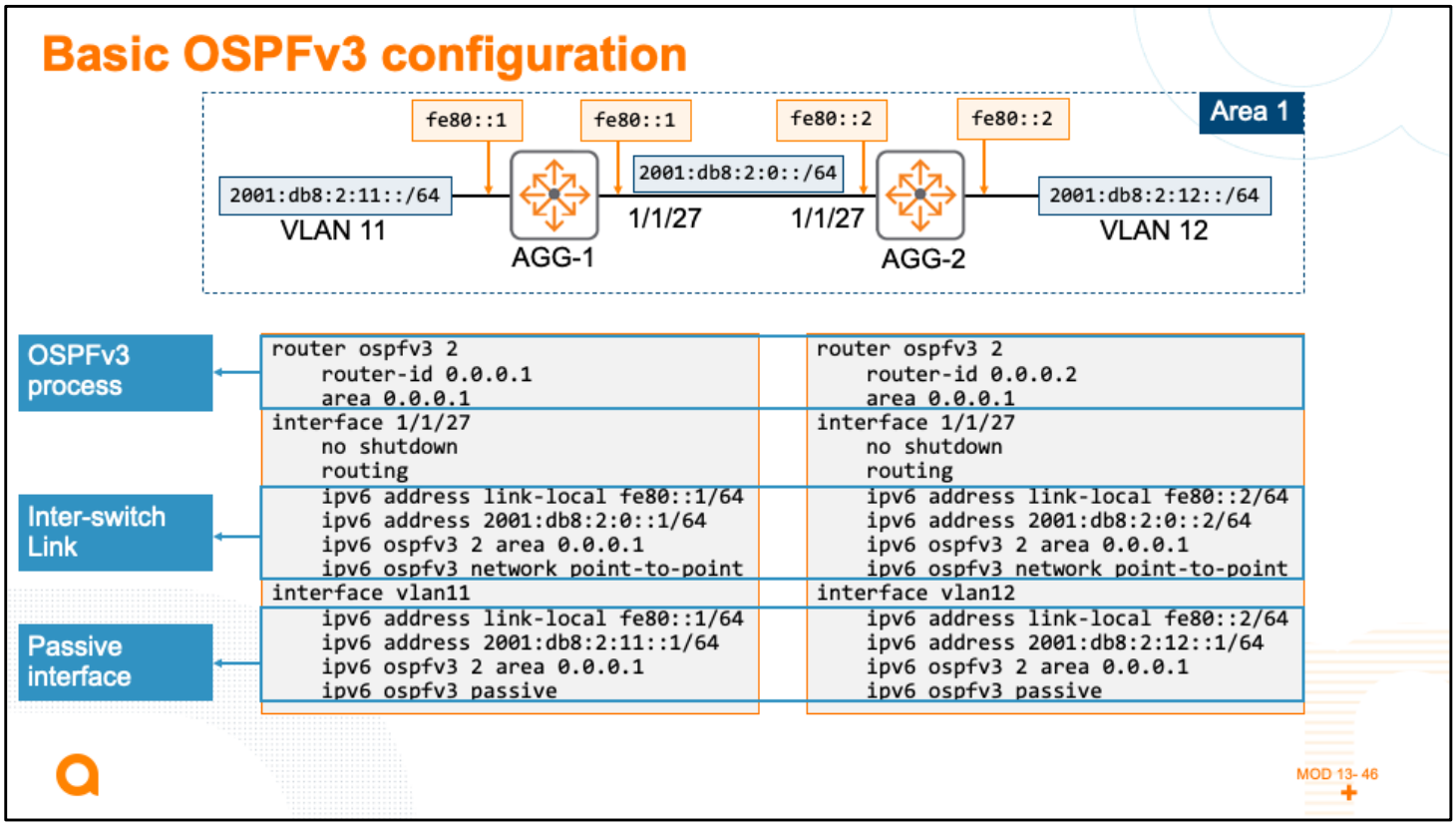
Advertising Router: 10.2.0.4

Sequence Number: 0x80000002

Checksum: 0x3d50

Length: 44

prefixes: 1
Referenced LS type: Unknown (0x2002)
Referenced Link State ID: 16.0.0.11
Referenced Advertising Router: 10.2.0.4
PrefixLength: 64
PrefixOptions: 0x00
Metric: 0
Address Prefix: 2001:db8:2:11::



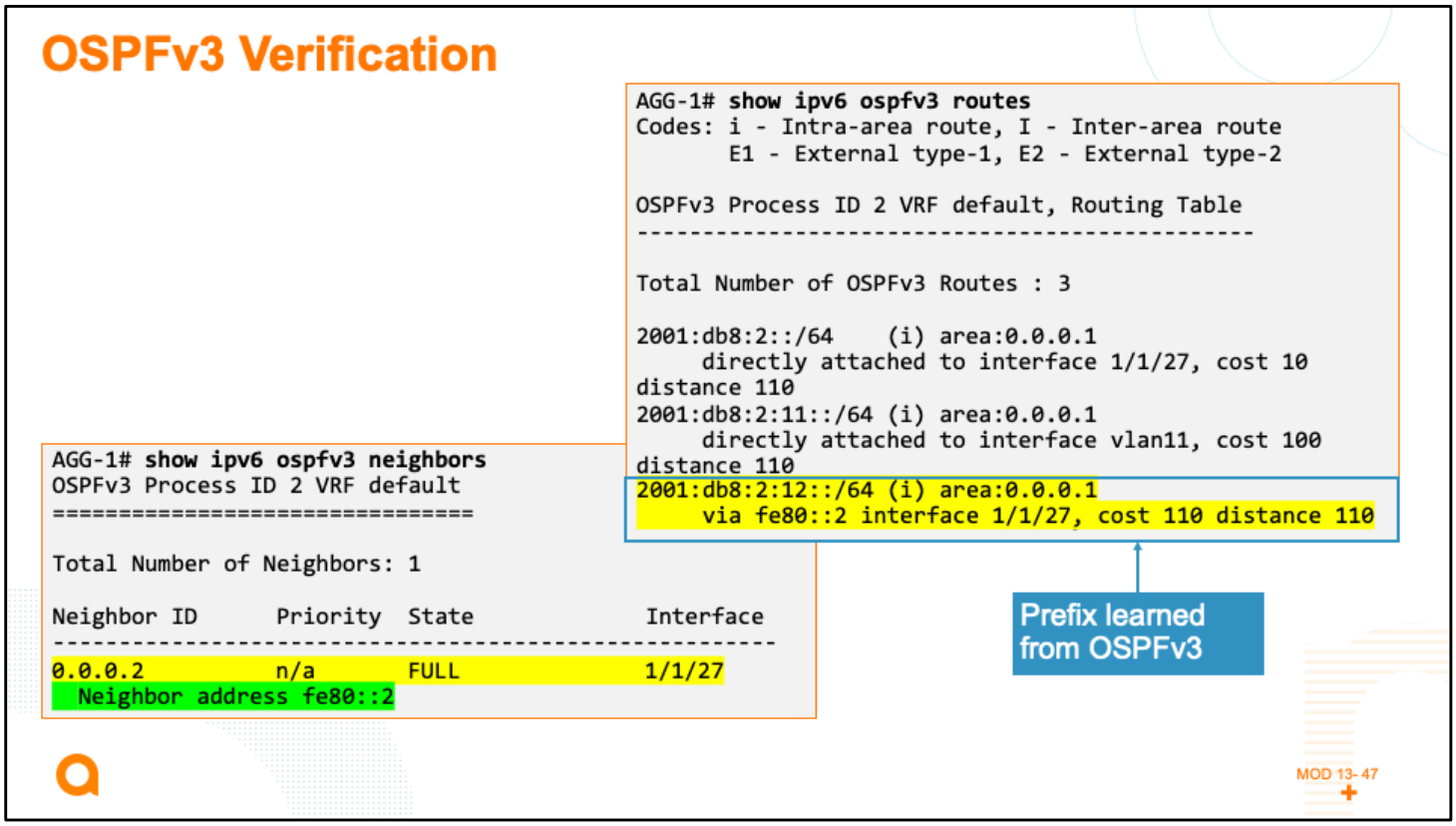
The figure shows a basic configuration between two switches running OSPFv3 in a single Area.

First, OSPFv3 maintains the concept of the Router ID which uniquely identifies each device in the OSPF domain. Notice that the Router ID is a 32-bit length value and can be represented using the dot notation.

An area is represented in OSPFv3 by either a 32-bit dotted-decimal address or a number. Area types include: Backbone, Normal, Not-so-stubby (NSSA), and Stub.

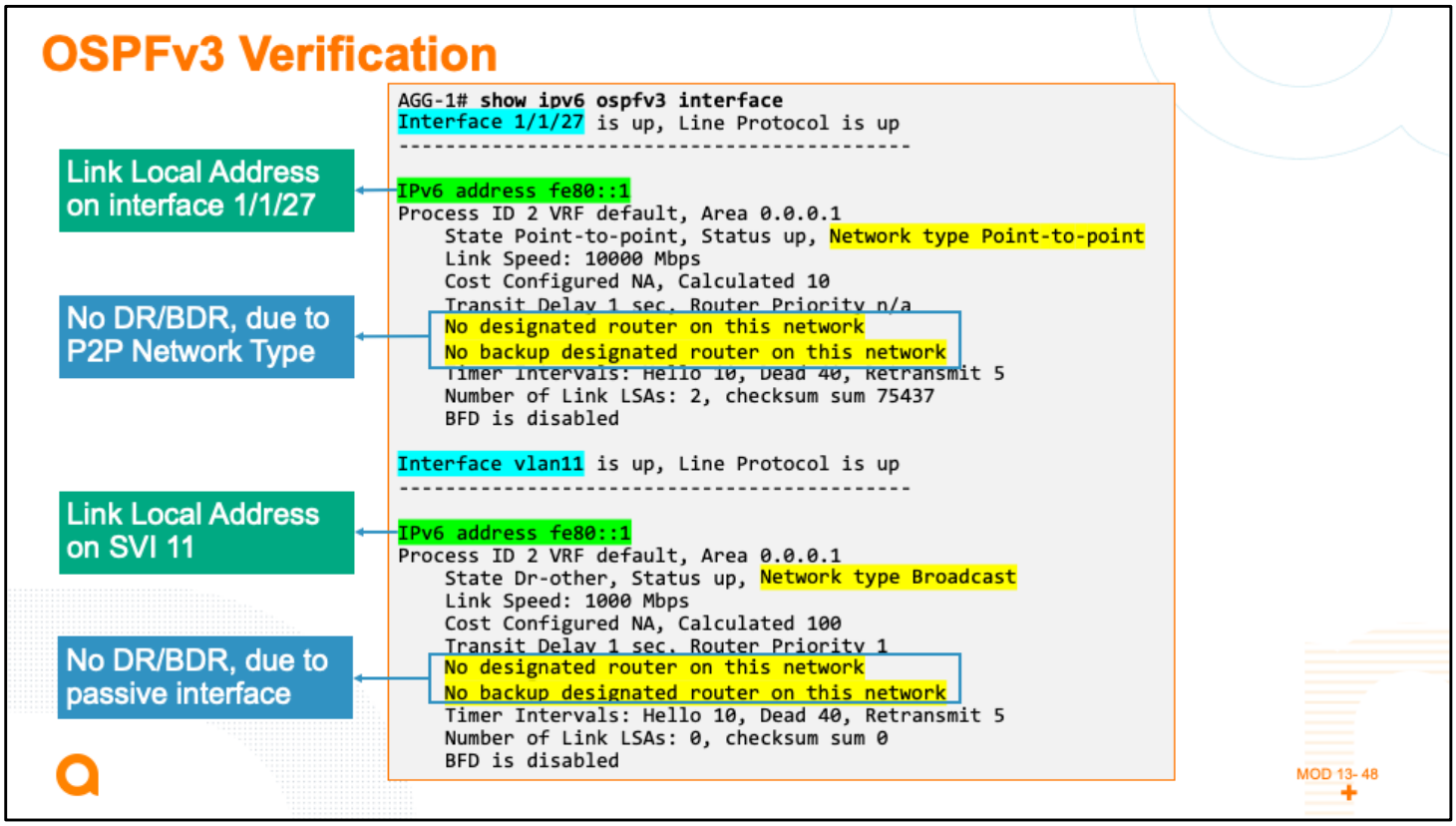
For easy management, the administrator has configured all the Link Local addresses for all interfaces with the same value on the same device, in this case Agg-1 uses the Link Local address fe80::1 and Agg-2 uses fe80::2. The Global Unicast prefixes are configured accordingly to the diagram.

To enable OSPFv3 in the correct interface you must enter the ipv6 ospfv3 <process-id> area <area-id> command, this step has not changed from OSPFv2. Noticed that in the Inter-switch link the administrator is using the Point-to-Point network type to avoid the election of the DR and BDR on the link. On the stub networks (VLAN 11 on AGG-1 and VLAN 12 on AGG-2) is recommended to use the passive interface if you are not expecting any OSPFv3 neighbor relationship on the link.



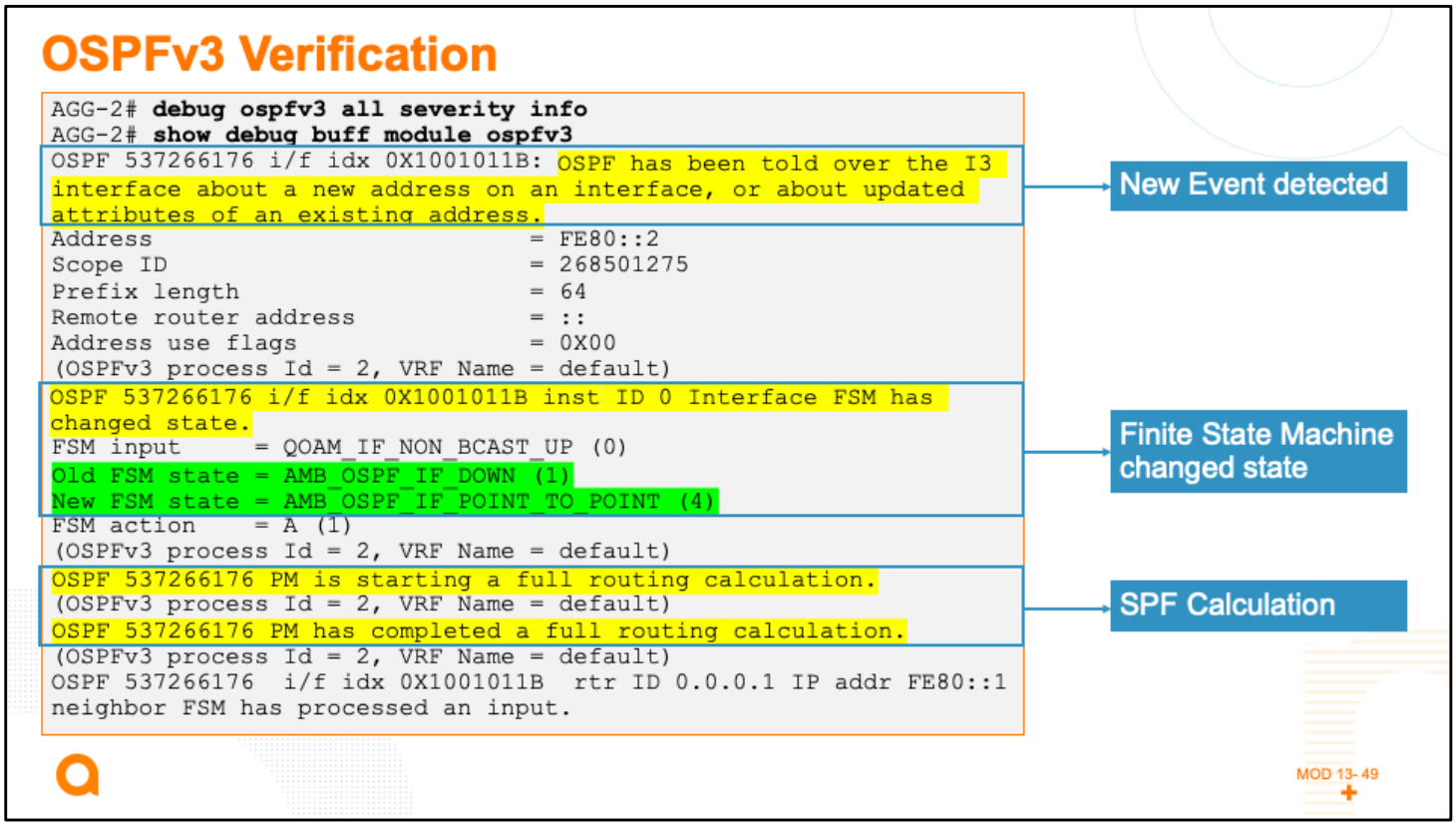
To verify neighbor relationships, use the show ipv6 ospfv3 neighbor command. In this example we have a FULL state with the peer 0.0.0.2, and since we are using Point-to-Point network type there is no DR/BDR election. Notice that the command displays at the bottom of the output what is the Link Local address of the neighbor, in this case fe80::2.

The show ipv6 ospfv3 routes displays at the bottom the Global Unicast Prefix learned from OSPFv3 in this case 2001:db8:2:12::/64 via fe80::2. Notice the Administrative Distance is the same as OSPFv2, 110. The cost of the route in this example has been calculated as 110.



The show ipv6 ospfv3 interface command provides important information about the routing protocol enabled in the interfaces. For example, you can verify the Link Local address used on each interface, the Status of the protocol in the interface, Network Type and state of the DR and BDR.

In the example you can see that on both interfaces (1/1/27 and VLAN 11) the is no DR nor BDR. The first interface does not need one since the Network Type has been set to Point to Point and the second because the interface was set to Passive interface.



In this example you can see how an AOS-CX switch process an OSPFv3 event. In this case a new address has been configured in the interface 1/1/27, this triggers a change in state of the Finite State machine which moves from DOWN to POINT_TO_POINT state. At the bottom you can see how the SPF calculation takes place.

Question #3

A couple of routers are running OSPFv3, the administrator adds a new prefix on one of them, does the OSPFv3 peer need to run the SPF algorithm for this change?

a. Yes, all the routes updates in OSPFv3 trigger the SPF algorithm

b. No, OSPFv3 does not include the metric in the update therefore there is no need to run the SPF algorithm again



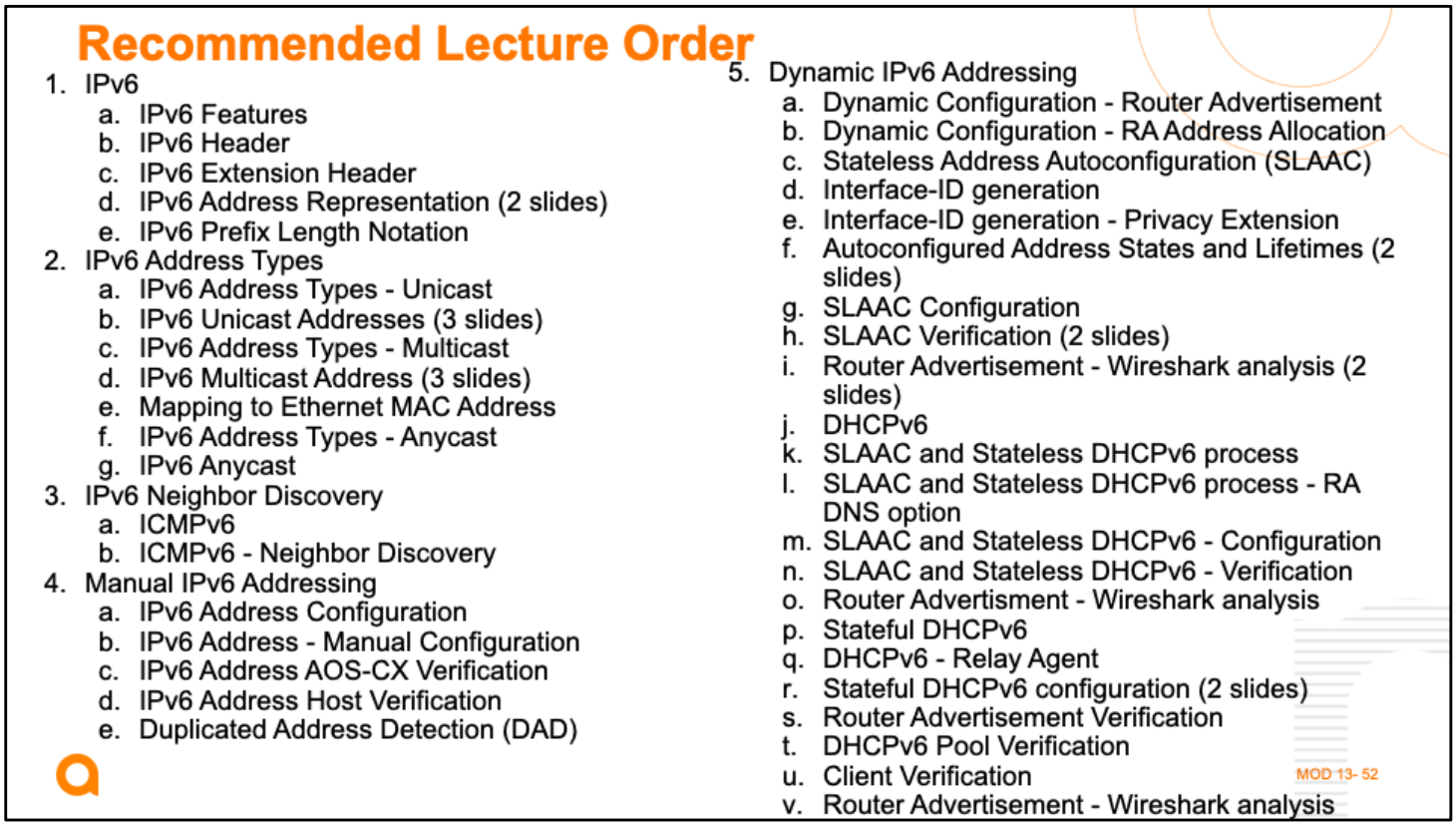


Correct answer:

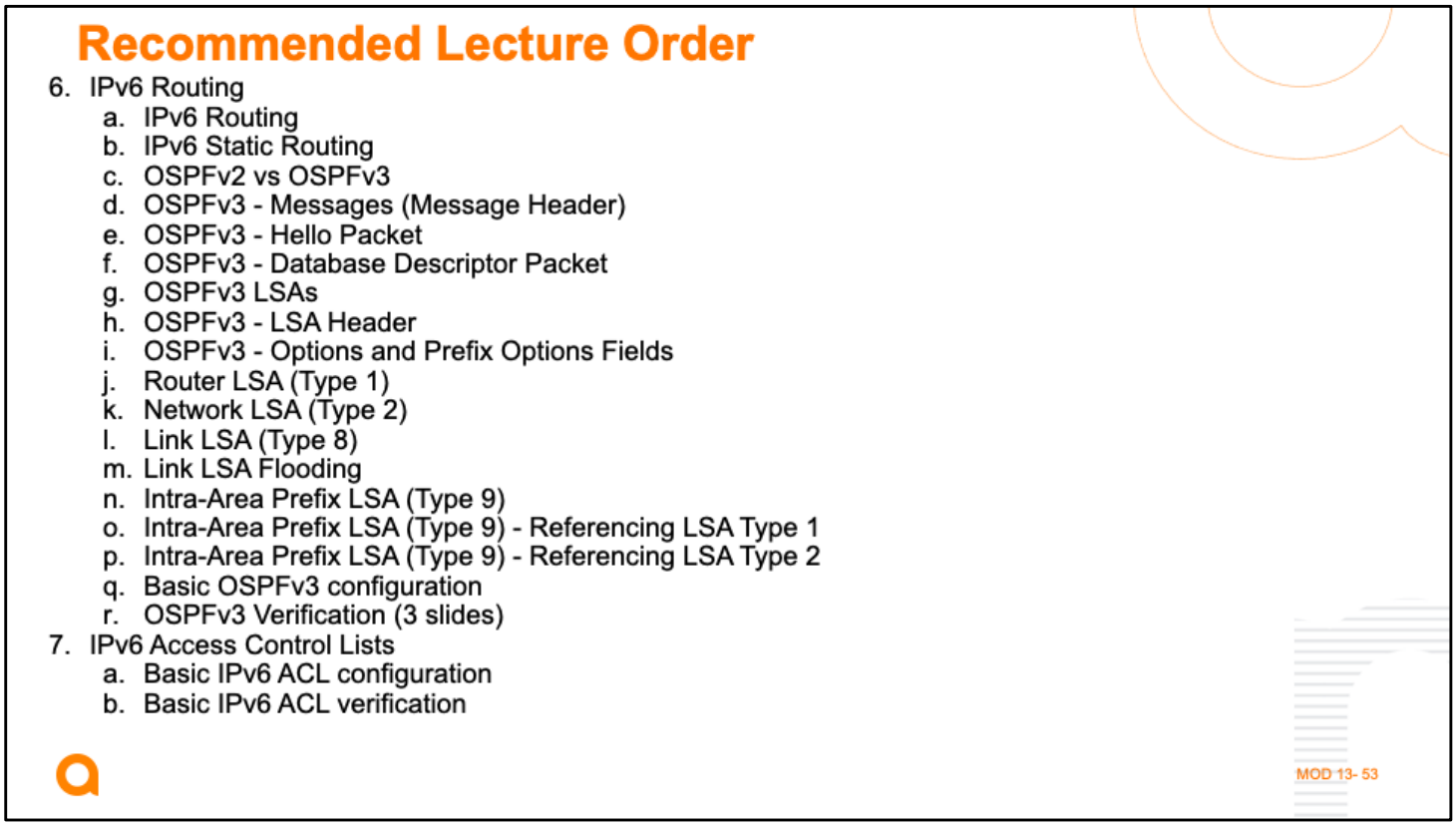
B

aruba
a Hewlett Packard Enterprise company

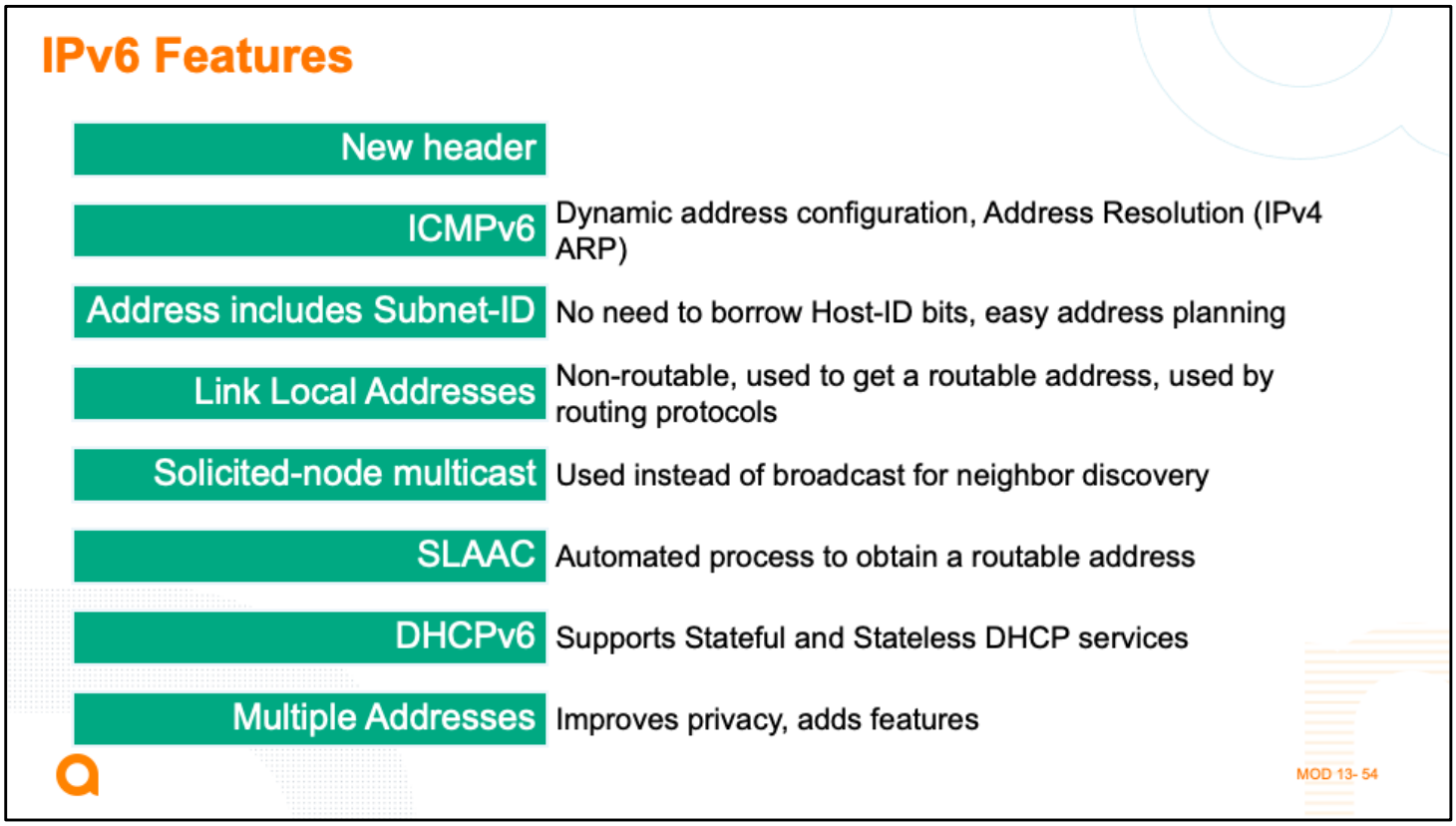
Additional Content



The additional material section includes information that will help you to understand AOS-CX. It is recommended to read this information in the order presented in the slide.



The additional material section includes information that will help you to understand AOS-CX. It is recommended to read this information in the order presented in the slide.



IPv6 offers the following features:

A new header facilitates more efficient packet processing.

ICMPv6 is an integration of IPv6, offering services such as Dynamic Auto-configuration, Address resolution (comparable to ARP in IPv4), and Duplicated Address Detection – to verify that addresses are unique.

The address length includes the Subnet-ID, so there is no need to borrow bits from the Host-ID. Network engineers will find planning the IP addressing an easy task.

The Link-Local address is a non-routable address - it only has local significance on the Layer 2 interface. This address is used for hosts to get a routable address. Network devices such as Routers or Multilayer switches use these addresses to create neighbor relationships.

A solicited-node multicast address is used instead of broadcast to make neighbor discovery more efficient.

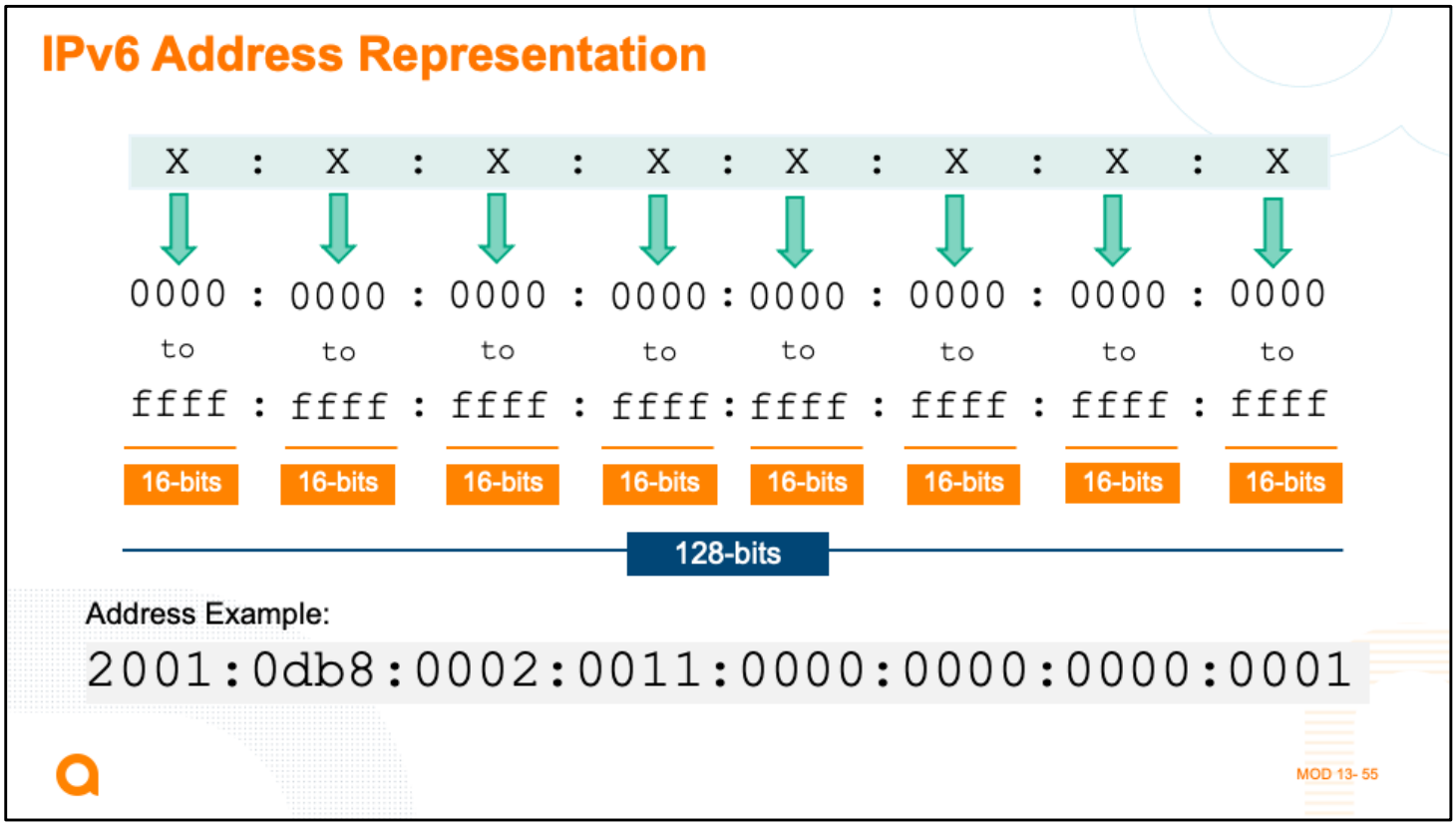
Stateless Address Autoconfiguration (SLAAC) allows devices to obtain an IPv6 address without the services of a DHCPv6 server.

DHCPv6 is another service option for hosts to obtain an IP address. DHCPv6 services can be classified in Stateless and Stateful.

Multiple addresses can be assigned to the same interface. This can be used to improve privacy and to add features.

In simple words, IPv6 is more than longer addresses!

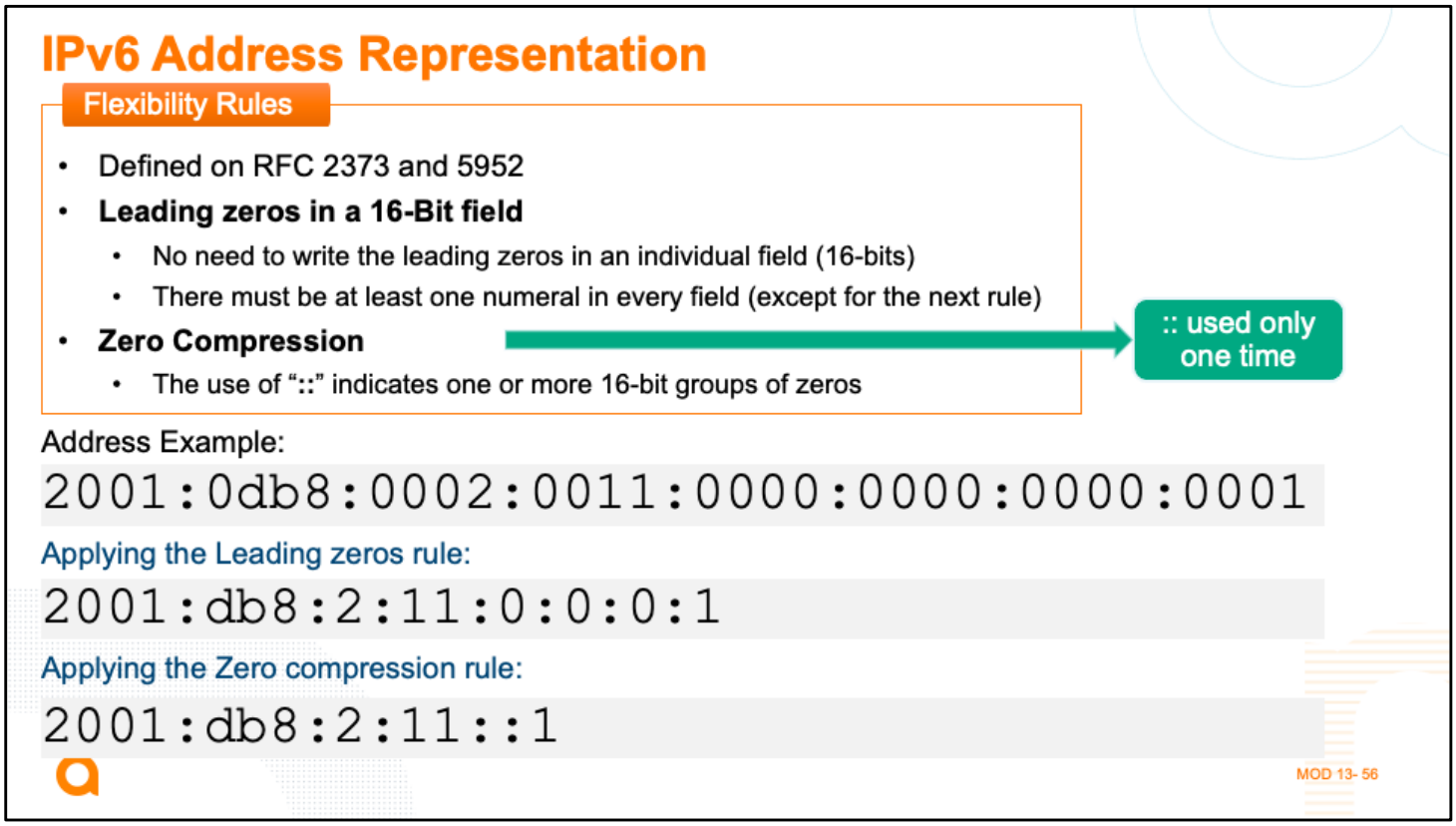
Note. Stateful DHCP is a service that keeps the state of the leases. It can also provide Global Unicast Addresses. However, Stateless DHCP servers do not maintain lease states. It is only used to offer extra parameters (Not Global Unicast Addresses) to endpoints such as Domain, NTP or TFTP. You will learn more about these concepts in the following slides.



An IPv6 address is 128-bits long, commonly written as a string of hexadecimal digits. This reduces the representation to 32 hexadecimal characters.

As described in RFC 4291, the preferred form is x:x:x:x:x:x:x:x, where the 'x's are one to four hexadecimal digits of the eight 16-bit pieces of the address. Each 16-bit ‘x’ value is separated by colons.

For example: 2001:0db8:0002:0011:0000:0000:0000:0001 is a valid IPv6 address.



Writing down 32 hexadecimal characters can seem tedious and is not very practical. This is why the IPv6 protocol defines a couple useful rules:

Omit leading zeros in a 16-bit field. There is no need to write the leading zeros but there must be at least one numeral in every field (except for the case described in rule 2).

For example: 2001:0db8:0002:0011:0000:0000:0000:0001 address can be written 2001:db8:2:11:0:0:0:1

Zero compression. Due to some methods of allocating certain styles of IPv6 addresses, it will be common for addresses to contain long strings of zero bits. In order to make writing addresses containing zero bits easier a special syntax is available to compress the zeros. The use of "::" indicates multiple 16-bit groups of zeros. The "::" can only appear once in an address. The "::" can also be used to compress the leading and/or trailing zeros in an address.

For example: 2001:db8:2:11:0:0:0:1 address can be written 2001:db8:2:11::1

Other examples:

1080:0:0:0:8:800:200C:417A a unicast address may be represented as:

1080::8:800:200C:417A

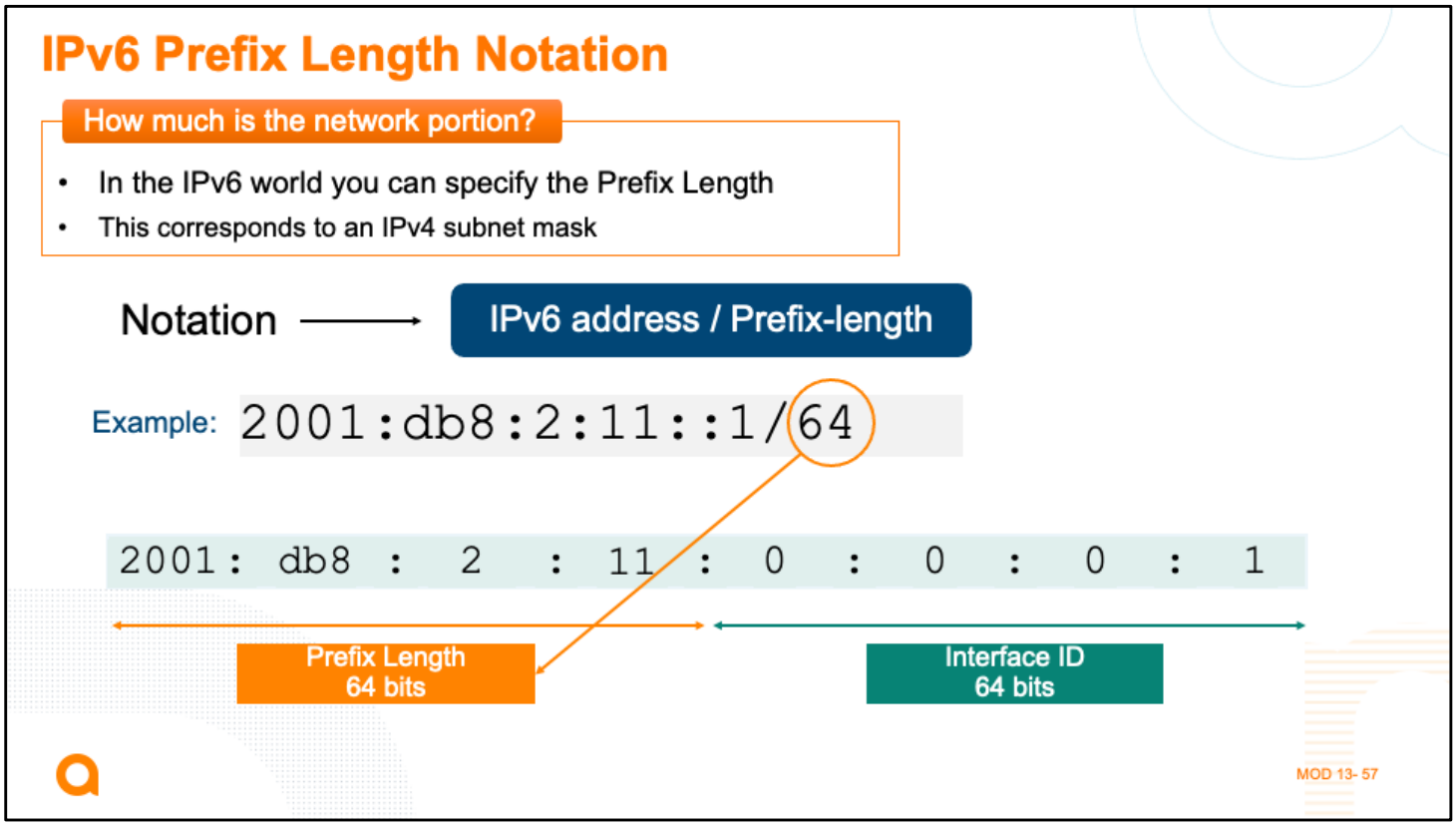
FF01:0:0:0:0:0:0:101 a multicast address may be represented as: FF01::101

0:0:0:0:0:0:0:1 the loopback address may be represented as: ::1

0:0:0:0:0:0:0:0 the unspecified addresses may be represented as ::

What happens if you have an address with more than one contiguous string of all-0s, for example, 2001:0db8:0000:0000:abcd:0000:0000:1234? In that case, where should you use the single double colon (::)? RFC 5952 answers these questions.

If the strings are of equal length, the first string should use the double colon (::) notation. Therefore, 2001:0db8:0000:0000:abcd:0000:0000:1234 would be written 2001:0db8:: abcd:0000:0000:1234. Applying both Rules 1 and 2, the address would be written 2001:db8::abcd:0:0:1234.



An IPv6 address prefix (the network portion of the address) is represented using the following format:

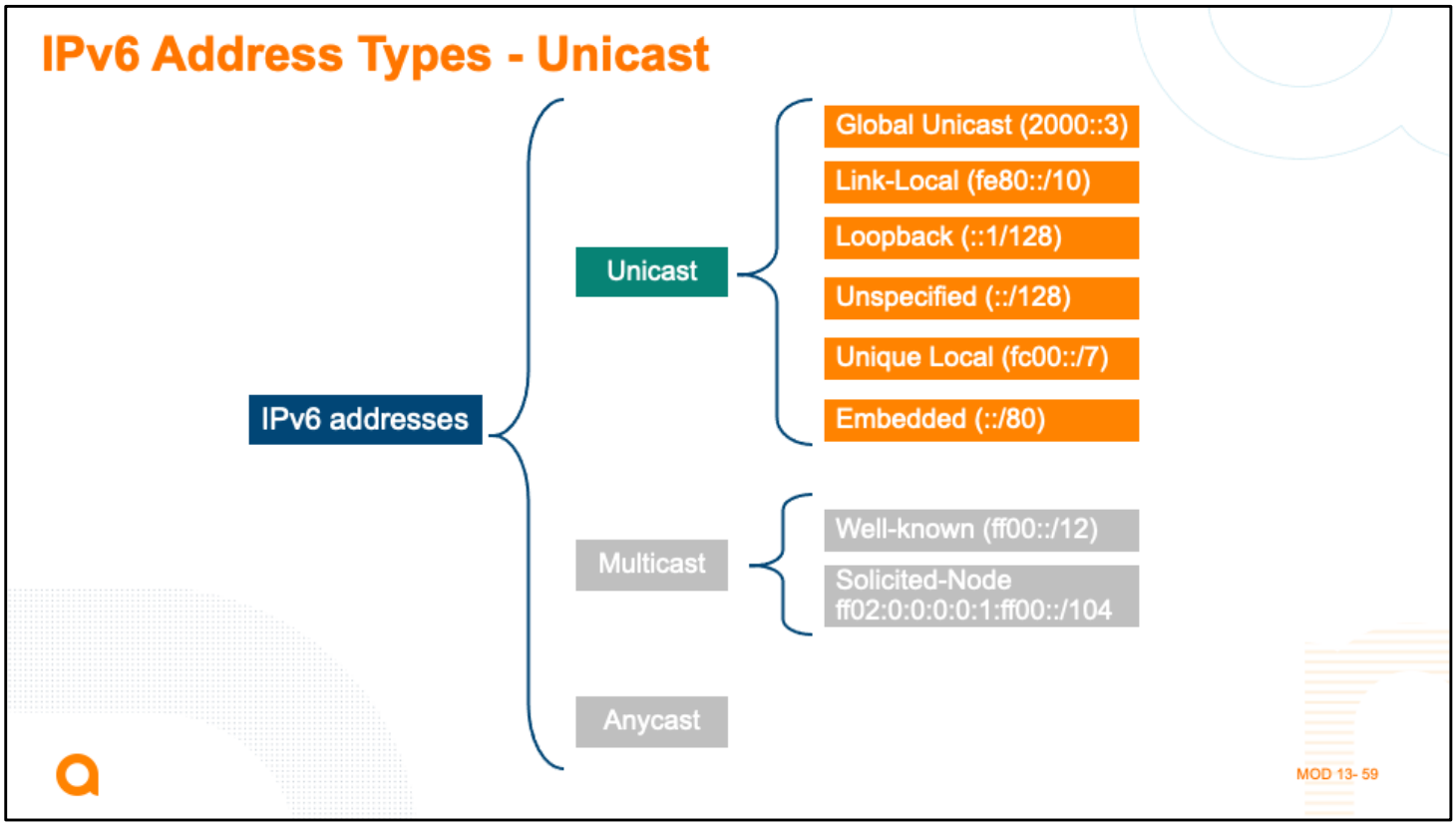
ipv6address / prefix-length

The prefix-length is a decimal value indicating the number of leftmost contiguous bits of the address. It identifies the prefix (that is, the network portion) of the address. It is also used with unicast addresses to separate the prefix portion of the address from the Interface ID.

aruba
a Hewlett Packard Enterprise company

IPv6 Address Types

Unicast, Multicast, Anycast



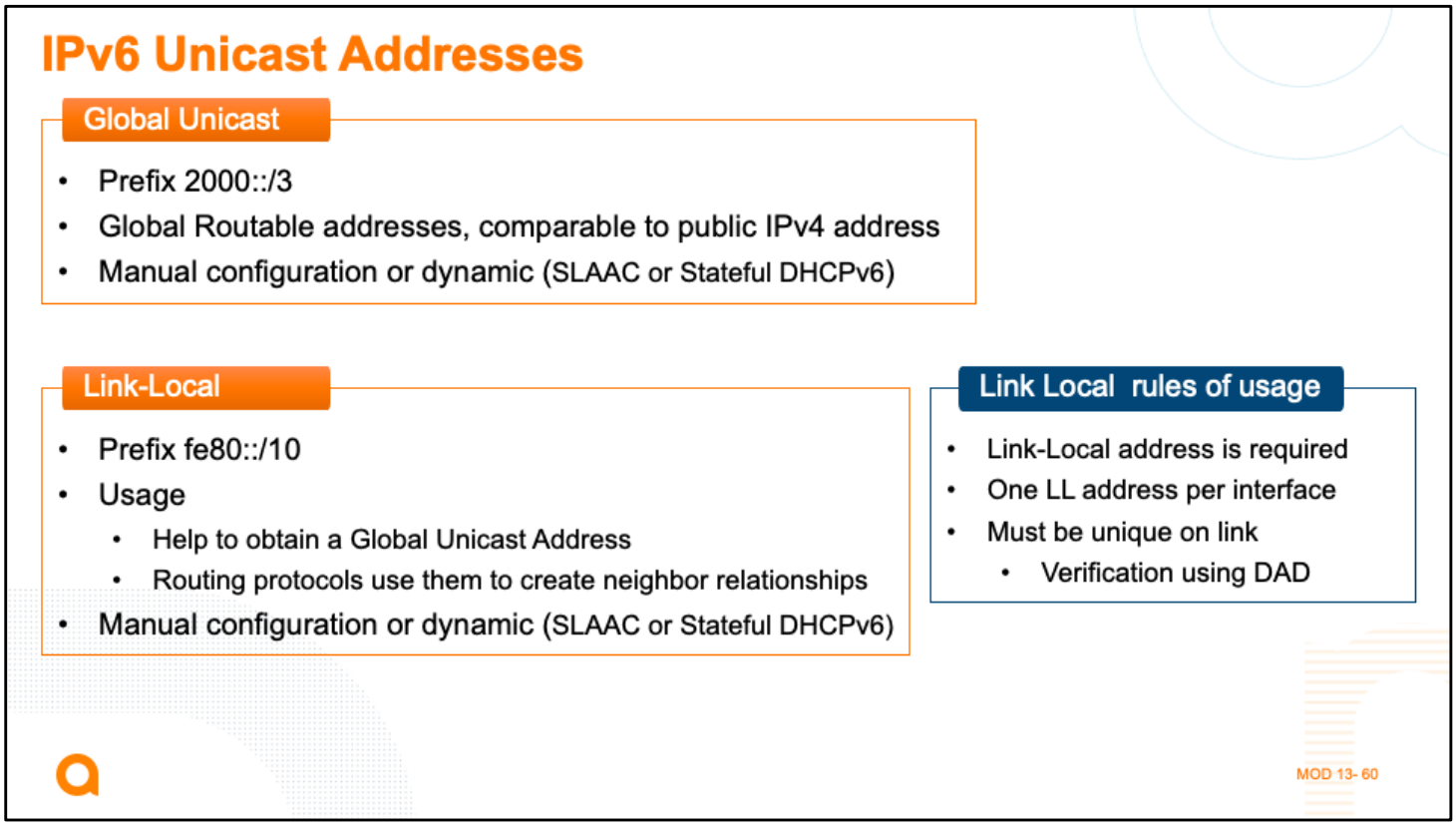
RFC 4291 defines the following IPv6 address types:

Unicast
- Global Unicast address (GUA)
- Link Local address (LLA)
- Loopback address
- Unspecified address
- Unique Local address
- Embedded address

Multicast
- Well-known address
- Solicited-node address

Anycast

This outline above acts as a reference for the topics to be covered next. You will progress through the various Unicast, Multicast, and Anycast addresses.



Global Unicast Addresses: Also known as aggregatable global unicast addresses, these are globally routable and reachable in the IPv6 internet. They are equivalent to public IPv4 addresses. These addresses use the Prefix 2000::/3 and can be manually configured or dynamically using SLAAC or Stateful DHCPv6. The structure of these addresses is 48 bits for Global Routing Prefix, 16 bits for Subnet ID and 64 bits for Interface ID.

Link-Local addresses: These addresses are only used in the same link (subnet or segment). In the IPv6 world these addresses play an important role since they can be automatically created to carry packets and messages that help the nodes to obtain a routable address such as Global Unicast. Another use is related to the routing process - routers always use a Link-Local address as the next-hop for an IPv6 route. Routing protocols like OSPFv3 use Link-Local address to establish neighbor relationships. These addresses use the prefix fe80::/10.

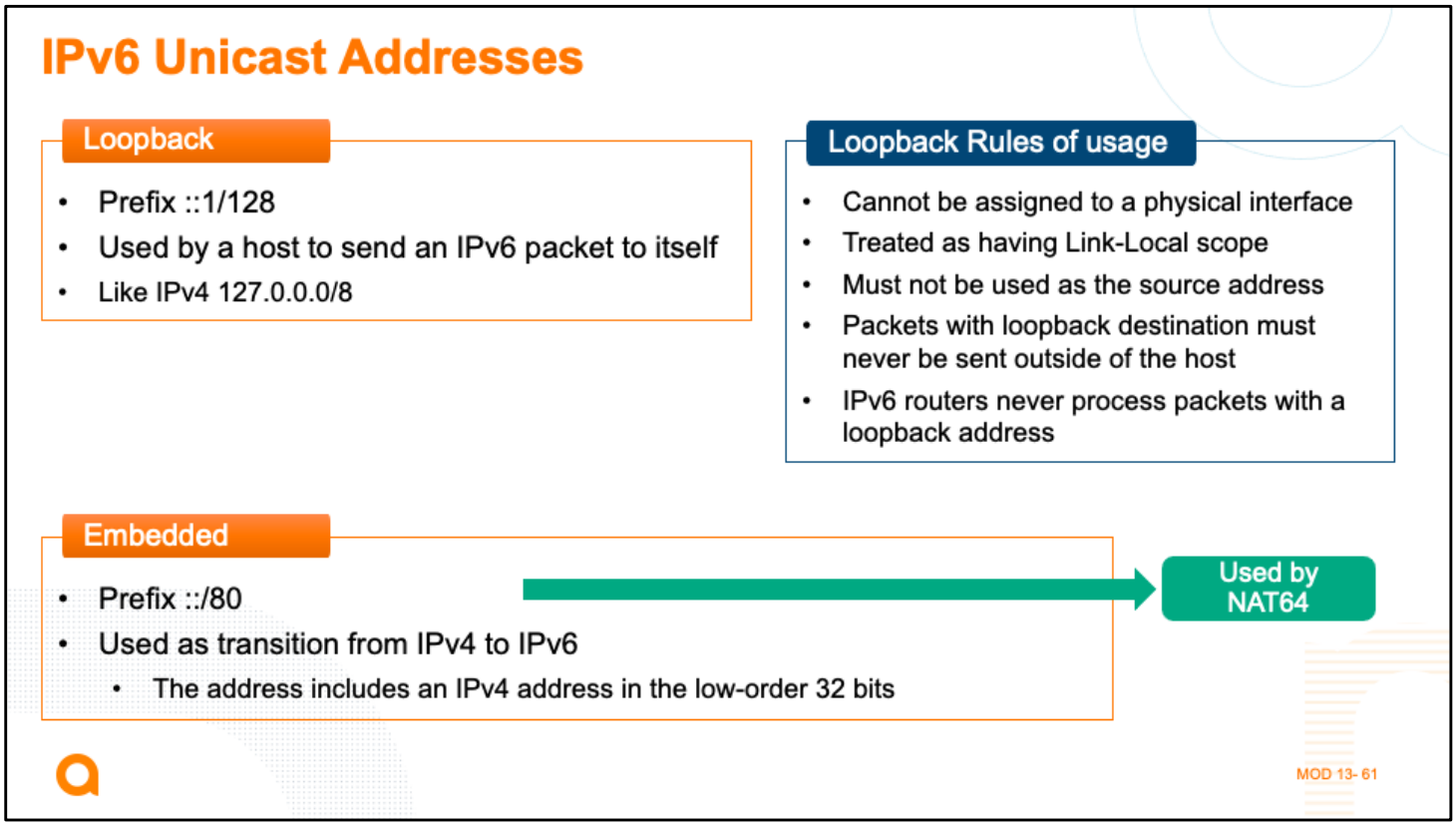
Link-Local addresses have the following rules:

All nodes running IPv6 must have a Link-Local address.

A single Link-Local address per interface can be used.

A Link-Local address must be unique on the link. IPv6 nodes use the Duplicated Address Detection (DAD) process to verify the uniqueness of the address. You will learn more about this later in this training.

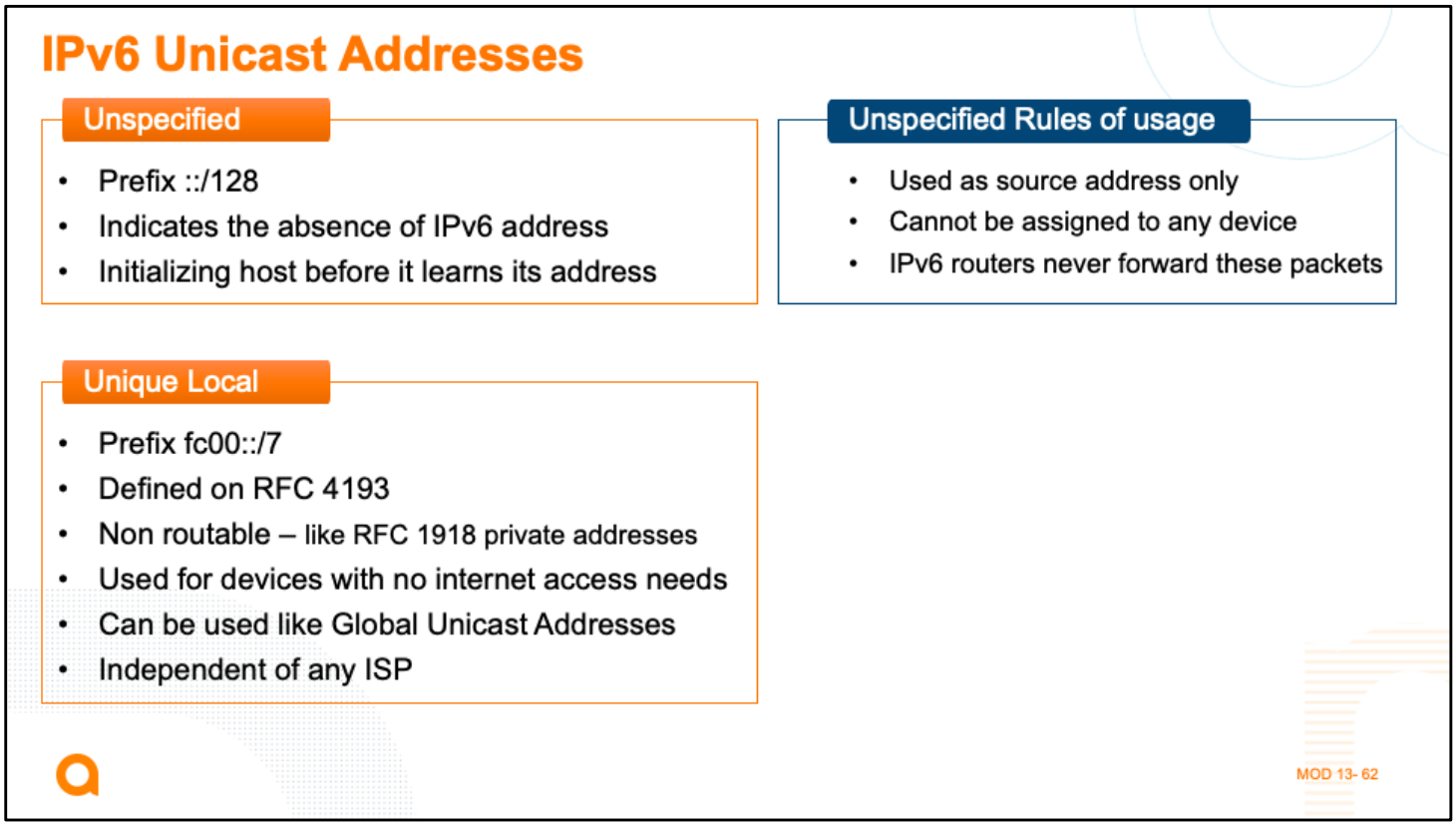
Note. The prefix 2001:db8::/32 has been assigned for documentation purposes (RFC 3849). This book uses this prefix for the examples.



Loopback address: The unicast address ::1 is called the loopback address. It may be used by a node to send an IPv6 packet to itself, but it shall never be assigned to any physical interface. It is associated with a virtual interface - the loopback interface. The loopback address must not be used as the source address in IPv6 packets that are sent outside of a single node. An IPv6 packet with a loopback destination address must never be sent outside of a single node and must never be forwarded by IPv6 routers. It is equivalent to the IPv4 address block 127.0.0.0/8, most commonly the 127.0.0.1 loopback address.

Embedded address: IPv4 embedded addresses are IPv6 addresses used to aid the transition from IPv4 to IPv6. IPv4 embedded addresses carry an IPv4 address in the low-order 32 bits. These addresses are used to represent an IPv4 address inside an IPv6 address. The prefix used for these addresses is ::/80.

Note. Defined on RFC-6146, NAT64 is a mechanism for IPv4-to-IPv6 transition and IPv4/IPv6 coexistence. The primary purpose of NAT64 is to allow an IPv6-only client to initiate communications to an IPv4-only server. NAT64 can also be used for IPv4-only clients initiating communications with IPv6-only servers using static or manual bindings. This technique uses embedded addresses.



Unspecified Unicast Address: This is an all-0s address, used as a source address to indicate the absence of an address. It cannot be assigned to an interface.

One example use case is as a source address in ICMPv6 Duplicate Address Detection (DAD). Devices use DAD to ensure that their unicast address is unique on the local link (network).

Unspecified addresses have the following characteristics:

They indicate the absence of an address.

They cannot be assigned to a physical interface.

They cannot be used as a destination address.

A router will never forward a packet that has an unspecified source address.

Unique Local Address: Comparable to IPv4 RFC 1918 private addressing, these addresses cannot be routed over the IPv6 Internet.

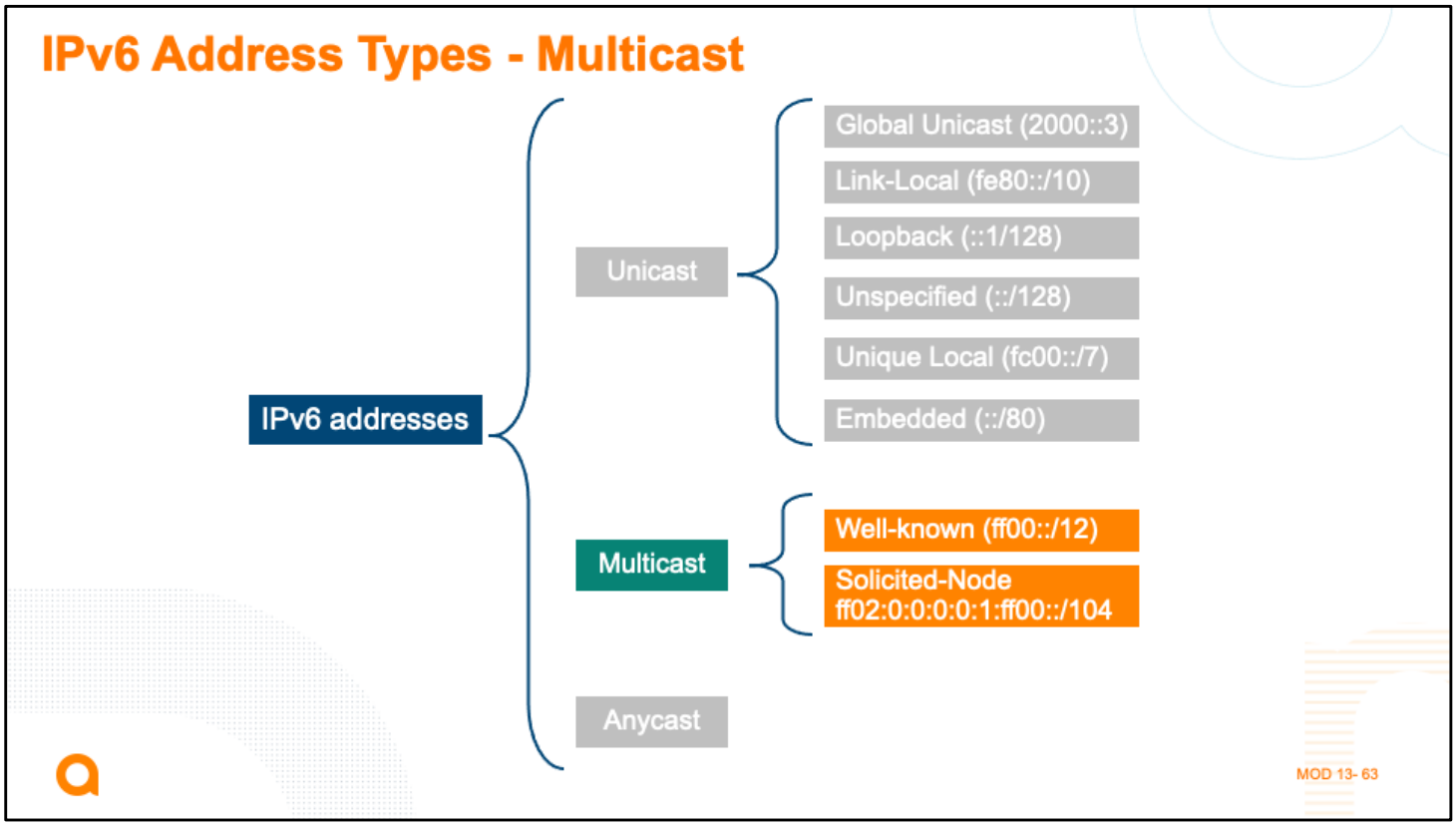
Unique local addresses have the following characteristics:

They can be used just like global unicast addresses.

They can be used for devices that never need access to or from the global Internet.

They allow sites to be combined or privately interconnected without address conflicts and without requiring addressing renumbering. (Address conflicts are highly unlikely due to the large address space.)

They are independent of any ISP and can be used within a site even without having Internet connectivity.

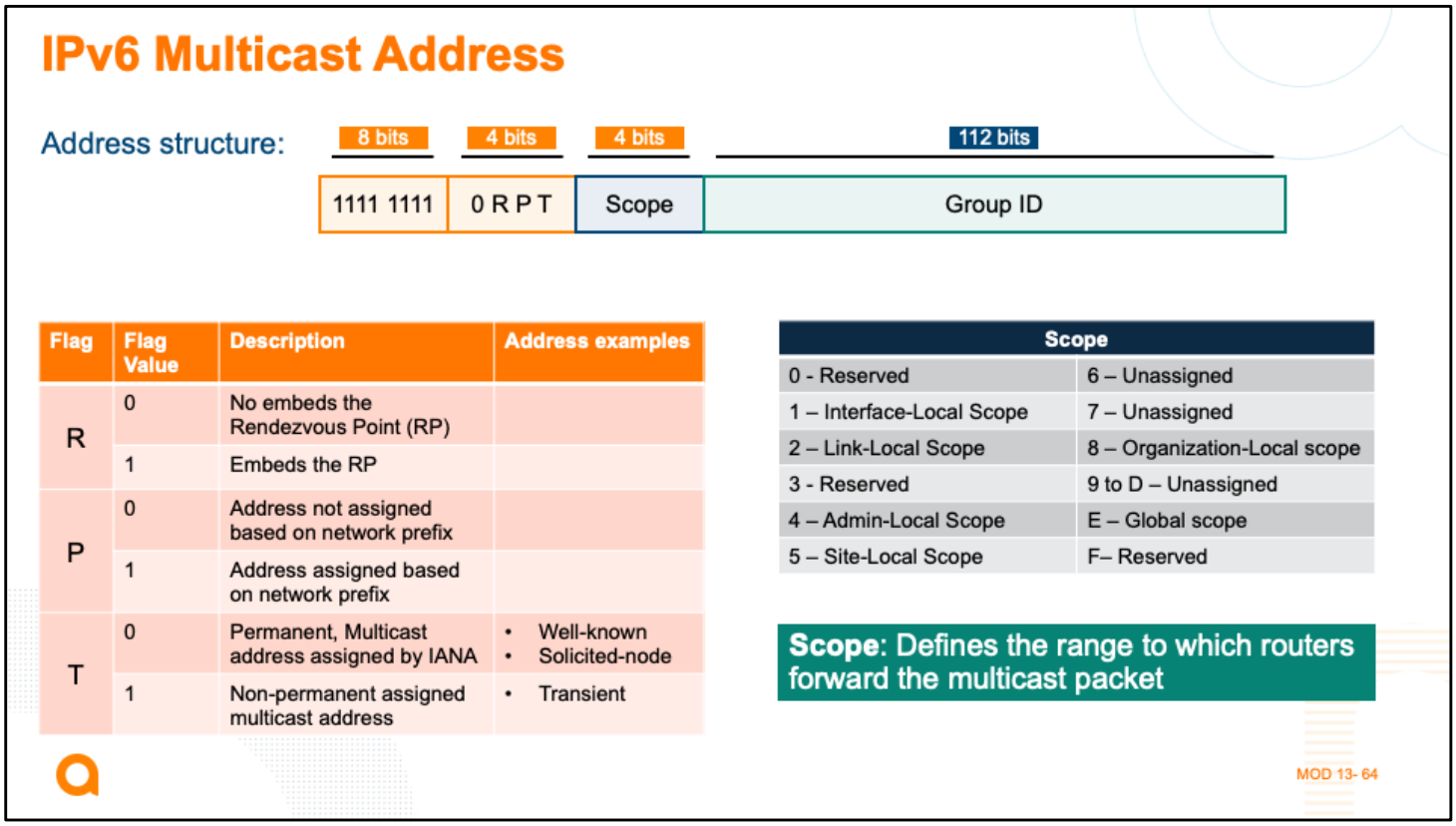


As previously mentioned, RFC 4291 defines the IPv6 address types listed below. You have learned about the unicast addresses, and so now you will explore multicast addressing.

Unicast
- Global Unicast address (GUA)
- Link Local address (LLA)
- Loopback address
- Unspecified address
- Unique Local address
- Embedded address

Multicast
- Well-known address
- Solicited-node address

Anycast



An IPv6 multicast address identifies a group of interfaces, typically on different nodes. An interface may belong to any number of multicast groups. IPv6 multicast addresses use the prefix ff00::/8, which is equivalent to the IPv4 multicast address 224.0.0.0/4. A packet sent to a multicast group always has a unicast source address. A multicast address can never be the source address.

The figure shows the structure of a Multicast address:

R-Flag: Defined in RFC 3956 – Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address

R = 0, it indicates that this multicast address does not embed the address of the RP.

R = 1, it indicates a multicast address that embeds the address on the RP.

P-Flag: Defined on RFC-3306 – Unicast-Prefix-based IPv6 Multicast Address

P = 0, it indicates a multicast address that is not assigned based on the “network prefix”. This indicates a multicast address defined in “IPv6 Address Architecture” - RFC-2373.

P = 1, it indicates a multicast address assigned based on the “network prefix”.

T-Flag: Defined on RFC-4291 – Transient flag

T = 0 indicates a permanently-assigned ("well-known") multicast address, assigned by the

Internet Assigned Numbers Authority (IANA).

T = 1 indicates a non-permanently-assigned ("transient" or "dynamically" assigned) multicast address.

Scope is a 4-bit multicast scope value used to limit the scope of the multicast group. The values are as follows:

0 reserved

1 Interface-Local scope

2 Link-Local scope

3 reserved

4 Admin-Local scope

5 Site-Local scope

6 (unassigned)

7 (unassigned)

8 Organization-Local scope

9 (unassigned)

A (unassigned)

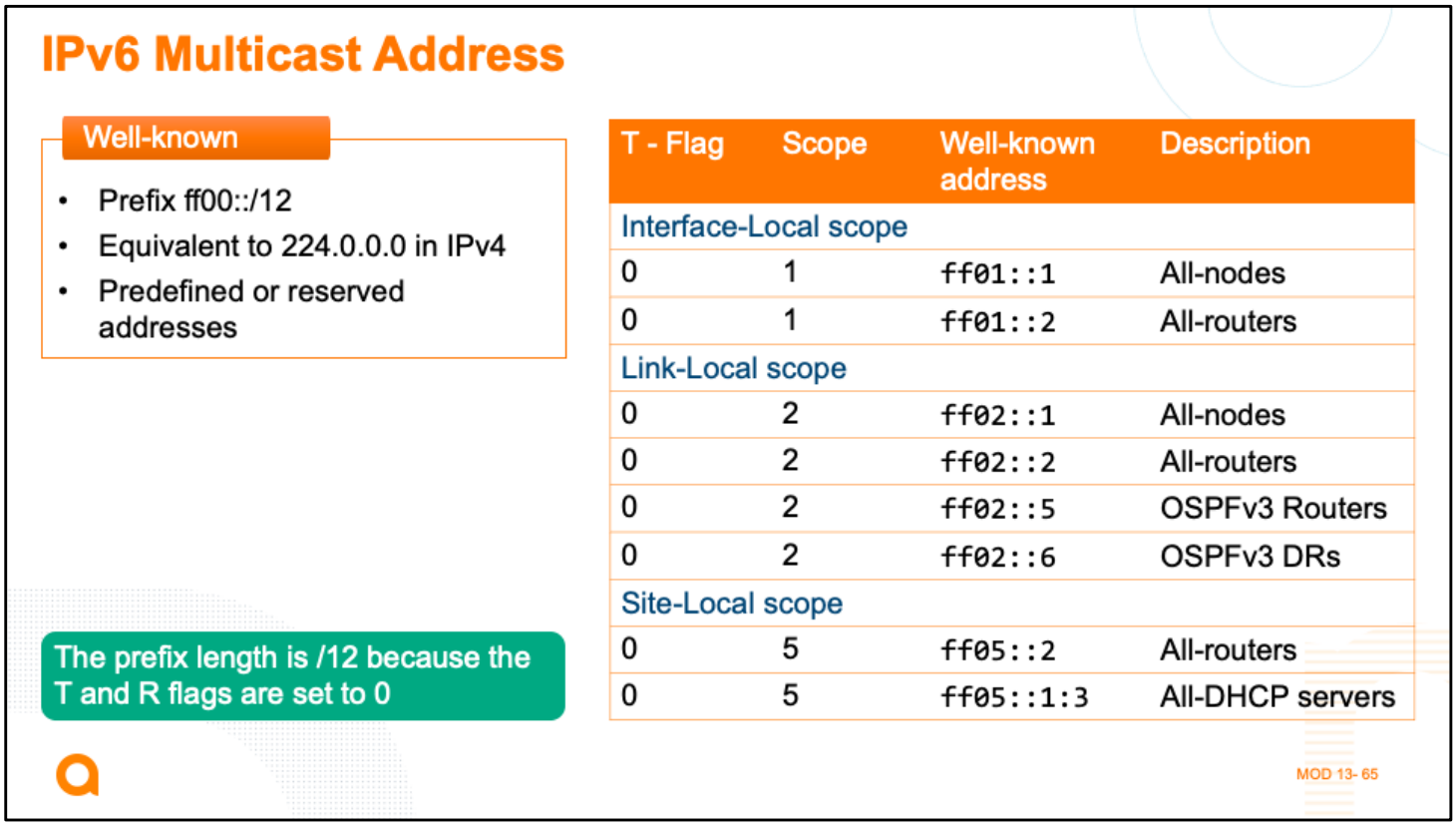
B (unassigned)

C (unassigned)

D (unassigned)

E Global scope

F reserved



T - Flag	Scope	Well-known address	Description
Interface-Local scope			
0	1	ff01::1	All-nodes
0	1	ff01::2	All-routers
Link-Local scope			
0	2	ff02::1	All-nodes
0	2	ff02::2	All-routers
0	2	ff02::5	OSPFv3 Routers
0	2	ff02::6	OSPFv3 DRs
Site-Local scope			
0	5	ff05::2	All-routers
0	5	ff05::1:3	All-DHCP servers

Well-known multicast addresses have the prefix ff00::/12, notice that the T flag is set to 0, which means that these addresses have been defined by IANA.

All Nodes Addresses:

FF01:0:0:0:0:0:0:1

FF02:0:0:0:0:0:0:1

The above multicast addresses identify the group of all IPv6 nodes, within scope 1 (interface-local) or 2 (link-local).

All Routers Addresses:

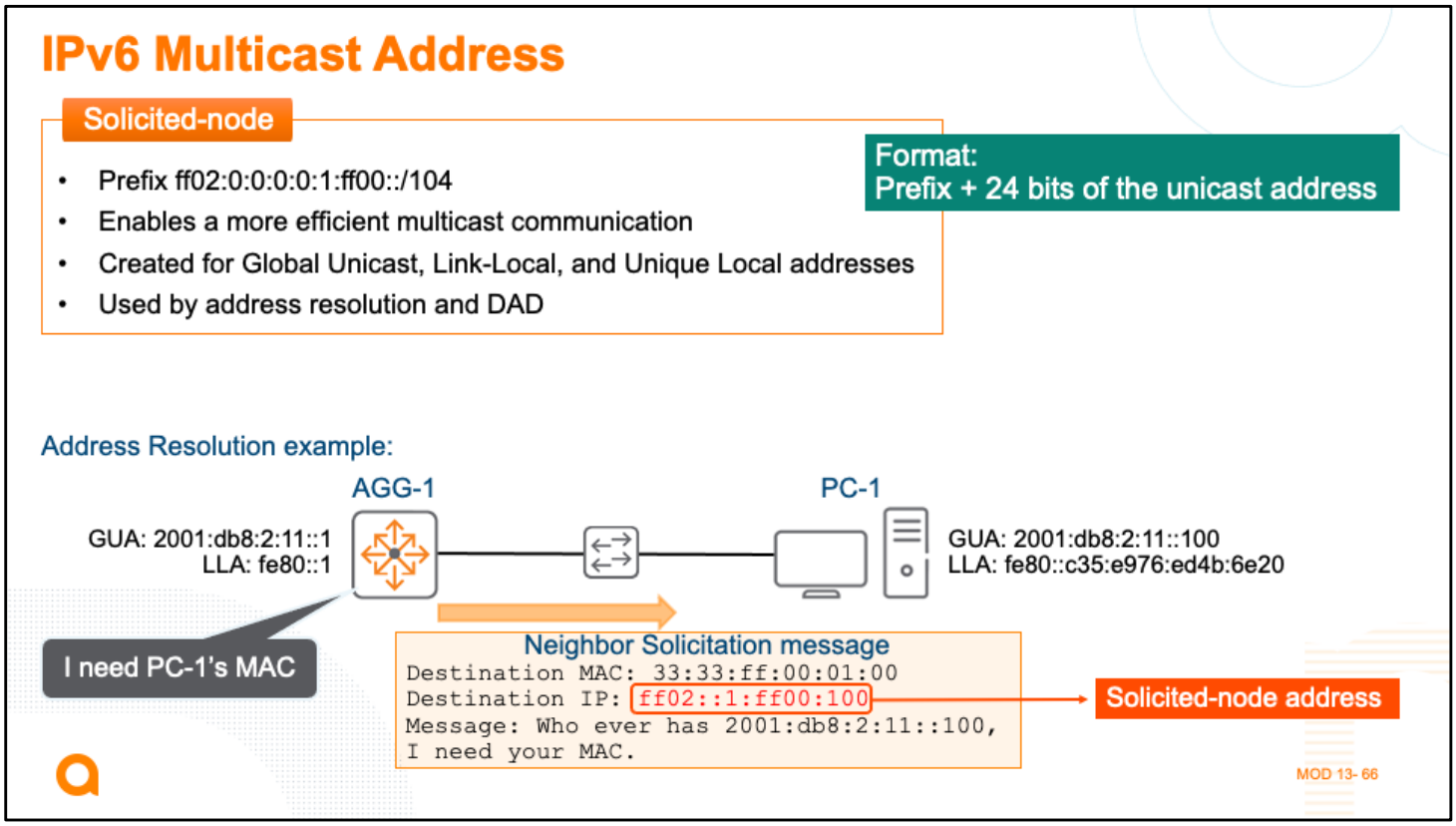
FF01:0:0:0:0:0:0:2

FF02:0:0:0:0:0:0:2

FF05:0:0:0:0:0:0:2

The above multicast addresses identify the group of all IPv6 routers, within scope 1 (interface-local), 2 (link-local), or 5 (site-local).

Note. The Use of /12 prefix length is because the T and R flag must be set to 0 for well-known Multicast Addresses.



A solicited-node multicast address is automatically created and assigned to an interface for every global unicast address, unique local address, and link-local address on that interface. A Solicited-Node multicast address is formed by taking the low-order 24 bits of an address (unicast or anycast) and appending those bits to the prefix FF02:0:0:0:0:1:FF00::/104 resulting in a multicast address in the range.

For example, the Solicited-Node multicast address corresponding to the GUA IPv6 address 2001:db8:51:11::100 is ff02::1:ff00:100. IPv6 addresses that differ only in the high-order bits (e.g., due to multiple high-order prefixes associated with different aggregations) will map to the same Solicited-Node address, thereby reducing the number of multicast addresses a node must join. A node is required to compute and join (on the appropriate interface) the associated Solicited-Node multicast addresses for all unicast and anycast addresses that have been configured for the node's interfaces (manually or automatically).

One of the benefits of using a Layer 3 multicast address is that it is mapped to a Layer 2 Ethernet MAC address. This allows the frame to be filtered by the switch1. This means these packets will only be forwarded out ports where there are devices that are members of that multicast group.

IPv6's solicited-node multicast address provides a more efficient solution for processes such as address resolution (ARP). The mapping of a solicited-node multicast address to an Ethernet MAC address allows the NIC to determine whether it is the intended recipient,

without sending to an upper-layer protocol for processing.

The figure shows the process for address resolution. An IPv6 device sends a Neighbor Solicitation message to a solicited-node multicast address to learn the link layer (typically Ethernet) address of a device on the same link. The device knows the IPv6 address of the destination on that link but needs to know its data-link (Ethernet) address. Please notice that the switch does not send a broadcast message as destination - it uses a Multicast solicited-node address.

Because a solicited-node multicast address is created using only the low-order 24 bits of its respective unicast address, and not the entire 64-bit Interface ID (in a /64 network), there is the potential for multiple devices to share the same solicited-node multicast address. This is not a problem. The upper-layer protocol, such as an ICMPv6 Neighbor Solicitation message, contains the target IPv6 address. The intended target, the device whose IPv6 address matches the target address, continues to process the packet and the other devices (not targets) drop the packet.

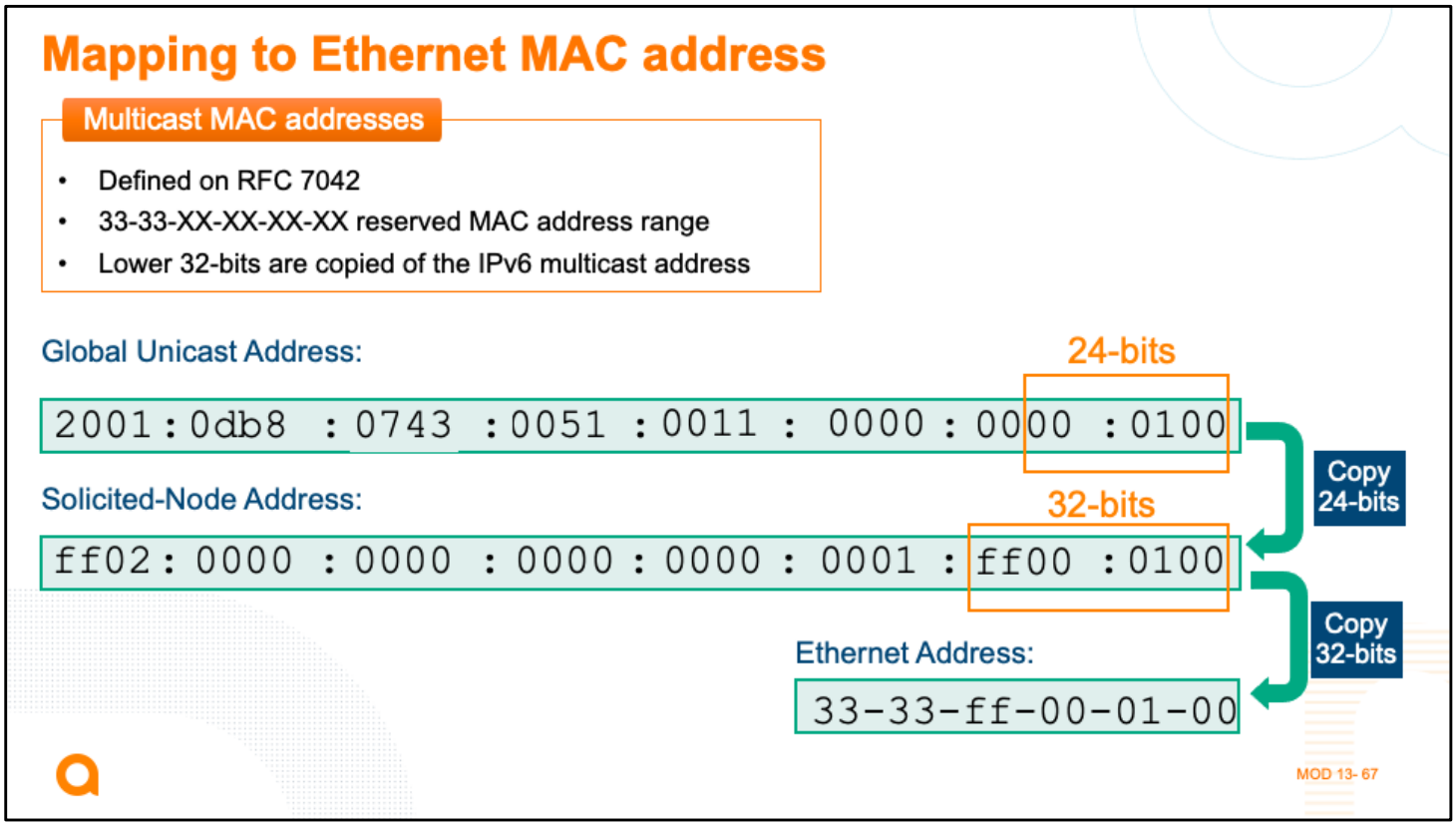
It is possible and even common for a device to have the same solicited-node multicast address for both its global unicast and link-local unicast addresses. For example:

Link-Local Address: fe80::1

Global Unicast Address: 2001:db8:2:1::1

This information will result in a common solicited-node address: ff02::1:ff00:1

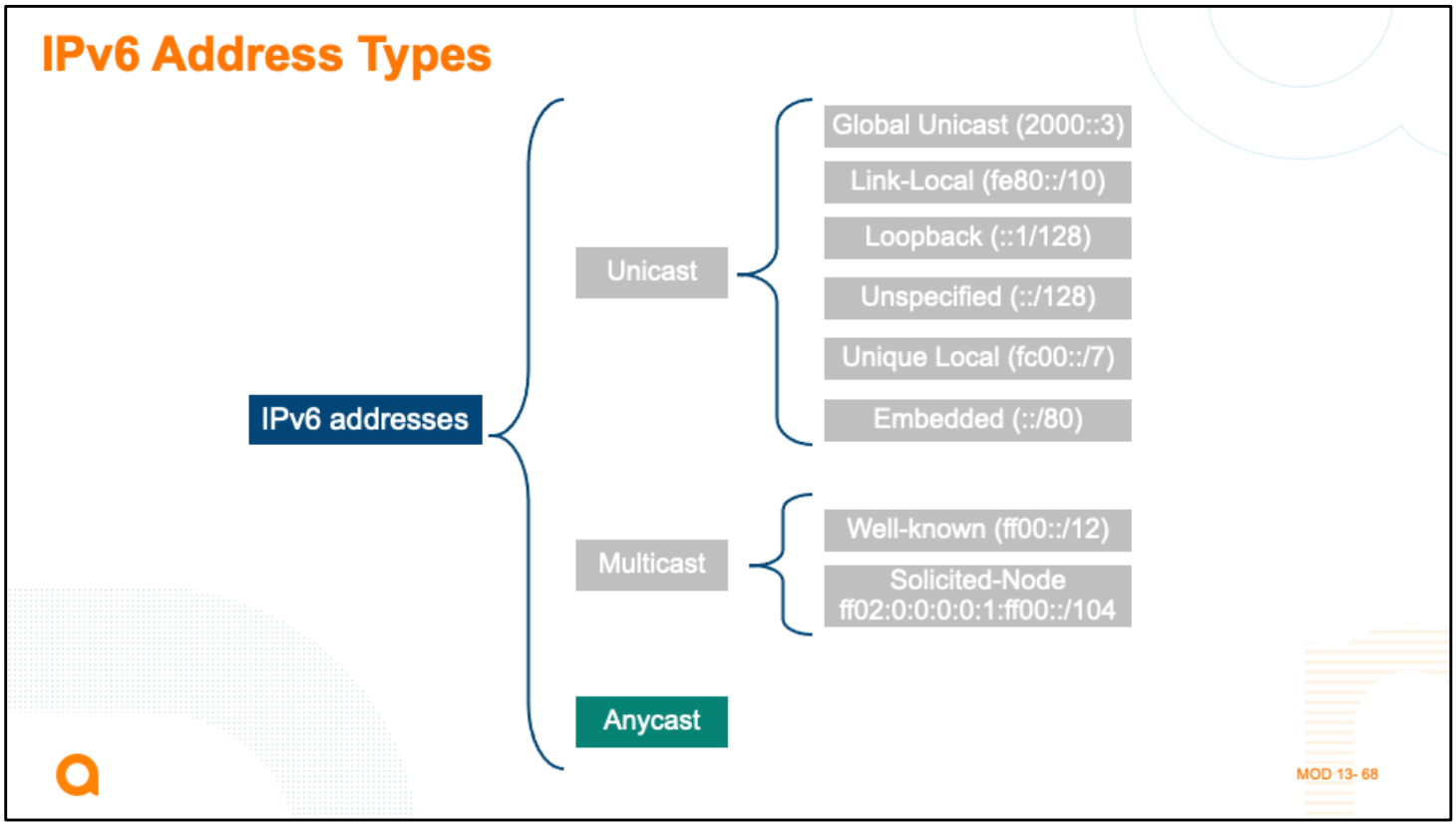
1The filtering process must be set by an administrator it does not happen automatically.



IPv6 multicast addresses, including solicited-node and well-known multicast addresses, are mapped to Ethernet MAC addresses. This is where you see the real advantage of using a multicast address over a broadcast address.

33-33-xx-xx-xx-xx is the reserved Ethernet MAC address when carrying an IPv6 multicast packet, as described in RFC 7042, IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters. The lower 32 bits of the MAC address (xx-xx-xx-xx) are copied from the lower 32 bits of the IPv6 multicast address.

The figure shows how a Solicited-node and the Ethernet addresses are generated based on the Global Unicast address.

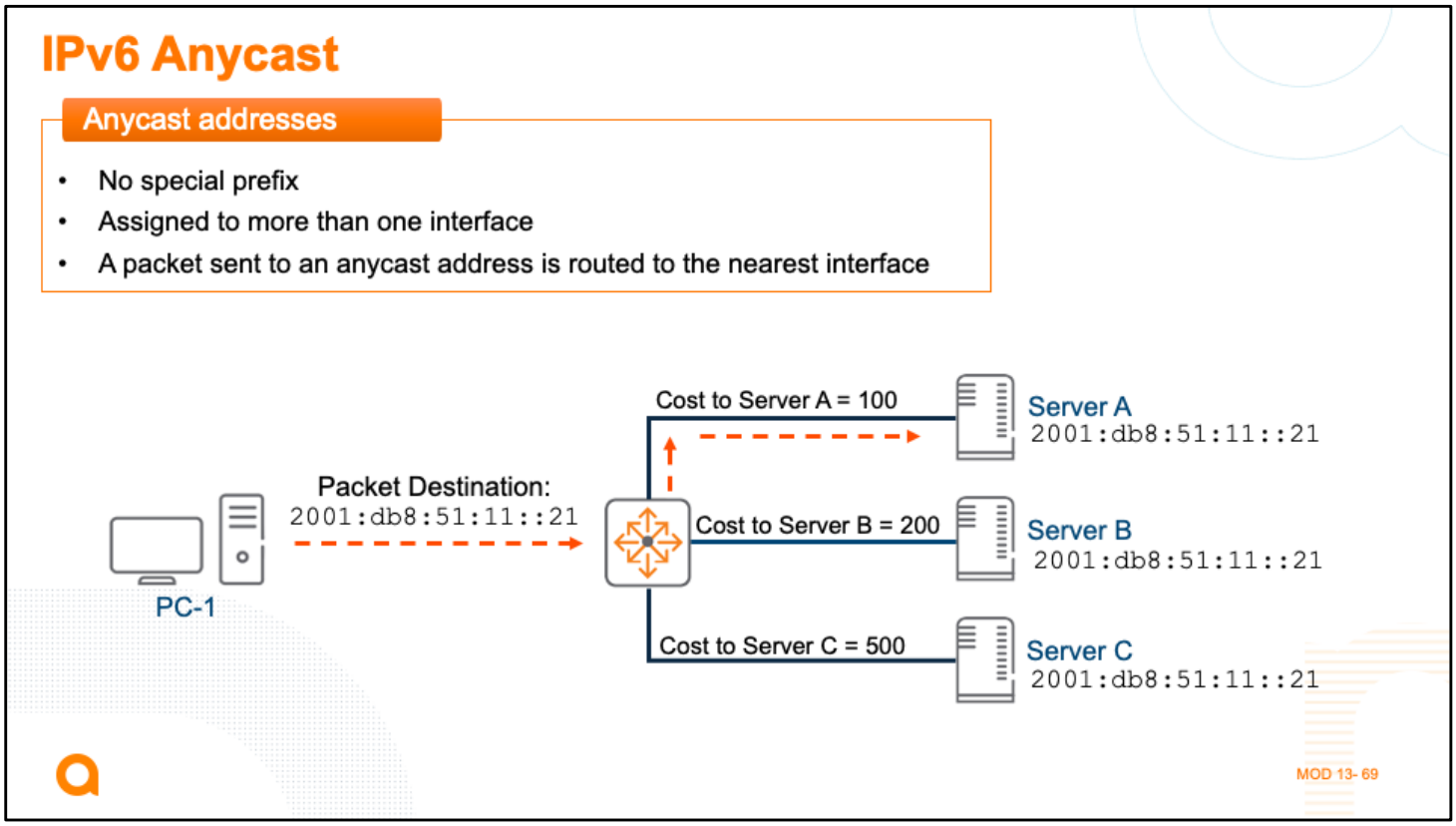


RFC 4291 defines the IPv6 address types, as listed below. You will now learn about Anycast addressing.

Unicast
- Global Unicast address (GUA)
- Link Local address (LLA)
- Loopback address
- Unspecified address
- Unique Local address
- Embedded address

Multicast
- Well-known address
- Solicited-node address

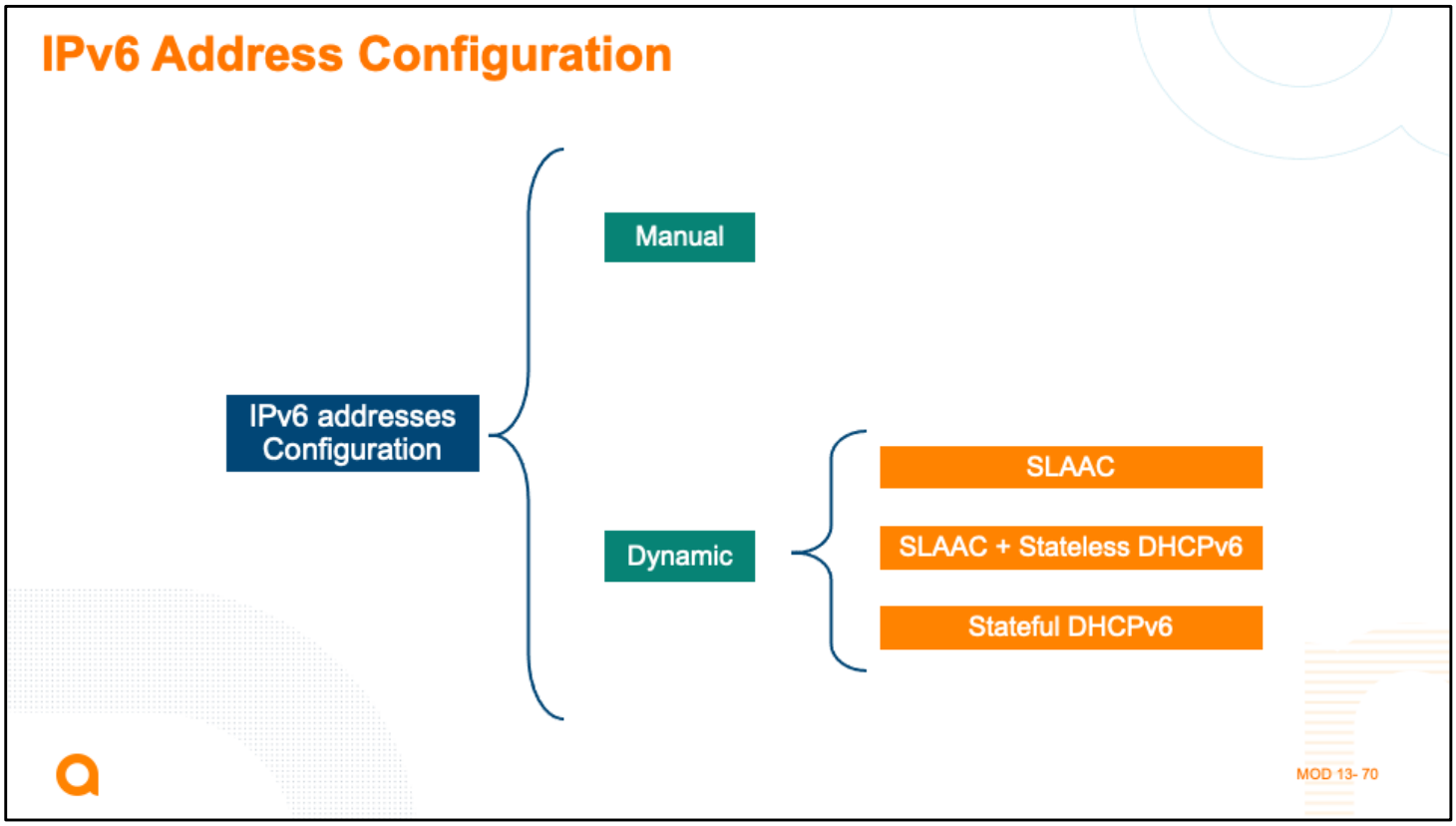
Anycast



An IPv6 anycast address can be assigned to more than one interface, typically on different devices. In other words, multiple devices can have the same anycast address. A packet sent to an anycast address is routed to the “nearest” interface having that address, according to the router’s routing table.

There is no special prefix for an IPv6 anycast address. An IPv6 anycast address uses the same address range as global unicast addresses.

In the example, you can see that all servers have the same address 2001:db8:51:11::21, the decision of where to send the packet depends on the routing table on the multilayer switch, in this case the packet is sent out from the interface that connect to Server A since from the switch perspective this path has the lower cost.



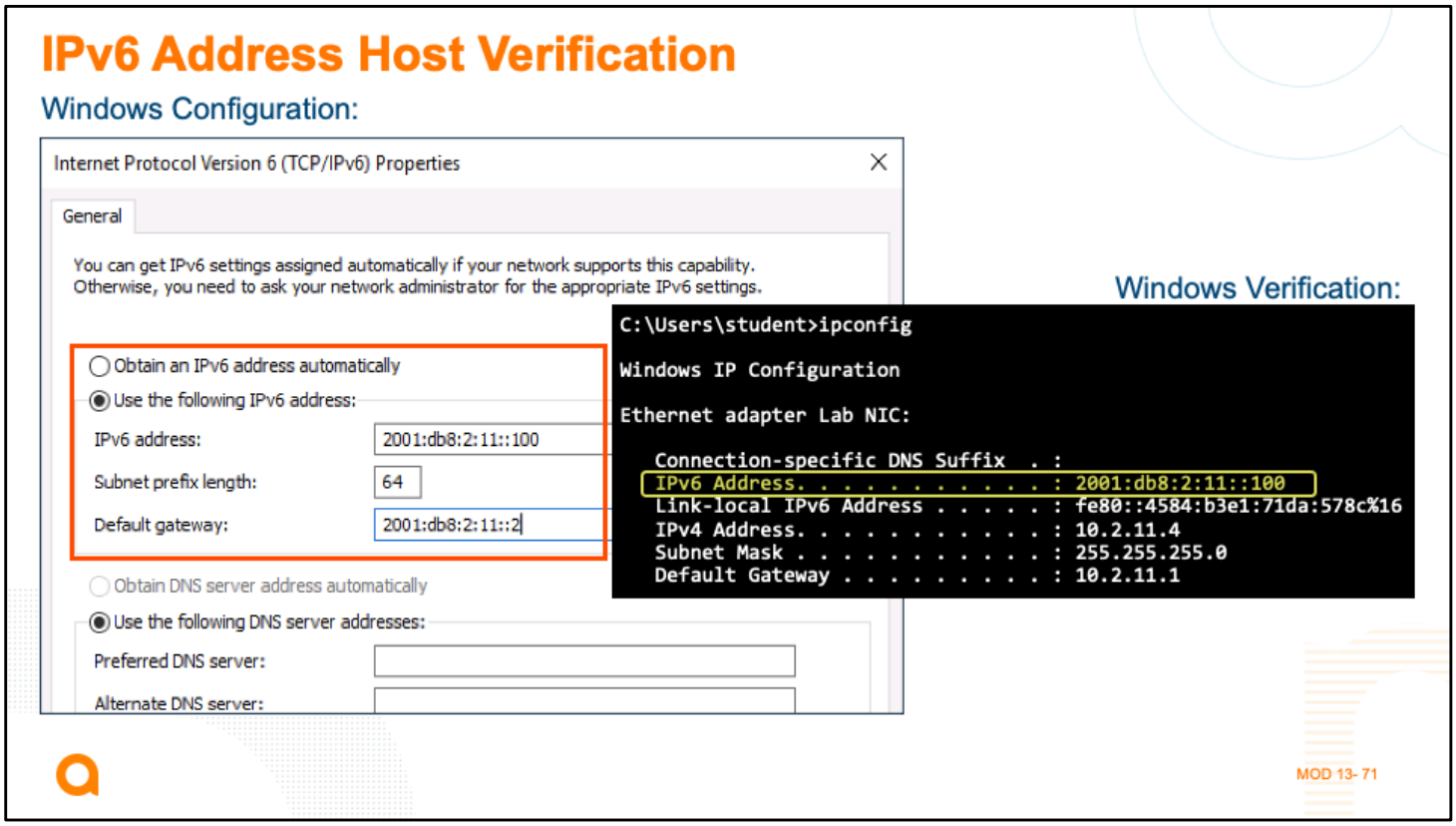
Endpoints can configure their interfaces with an IPv6 address using two options:

Manual configuration. This requires the intervention of the administrator or end user.

Dynamic configuration. This option helps devices to automatically obtain their IPv6 address, there are three methods that can be used:

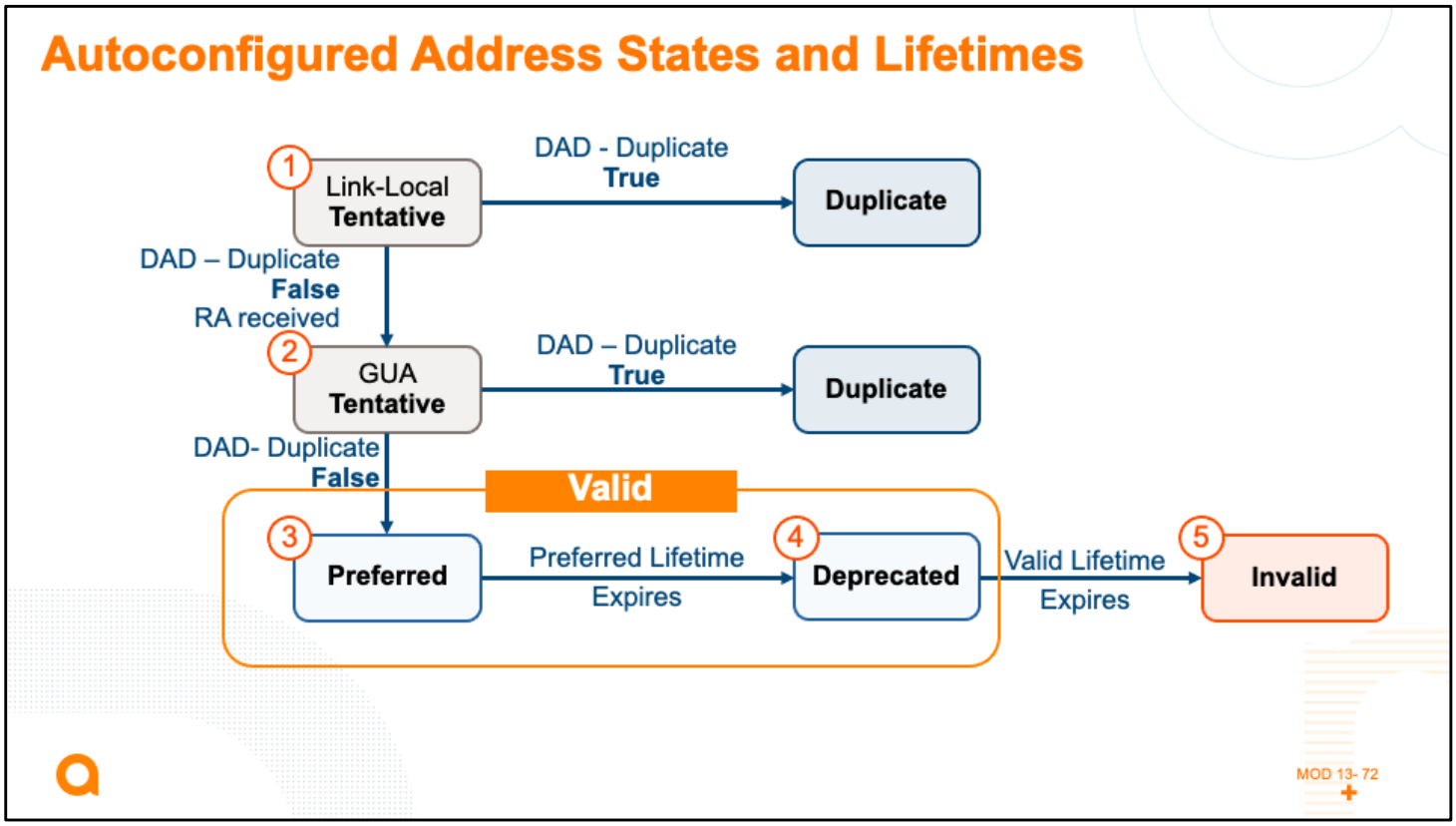
- SLAAC (Stateless Address Autoconfiguration)
- SLAAC and Stateless DHCPv6
- Stateful DHCPv6

Note. Stateful DHCP is a service that keeps the state of the leases. It can also provide Global Unicast Addresses. However, Stateless DHCP servers do not maintain lease states. It is only used to offer extra parameters (Not Global Unicast Addresses) to endpoints such as Domain, NTP or TFTP. You will learn more about these concepts in the following pages.



Manual configuration in a windows station is the same as an IPv4 address. Go to the interface properties and configure the IPv6 address.

To verify the address simply use ipconfig command.



A Temporal IPv6 address (which includes GUA addresses) moves from multiple states, as shown in the figure and described below.

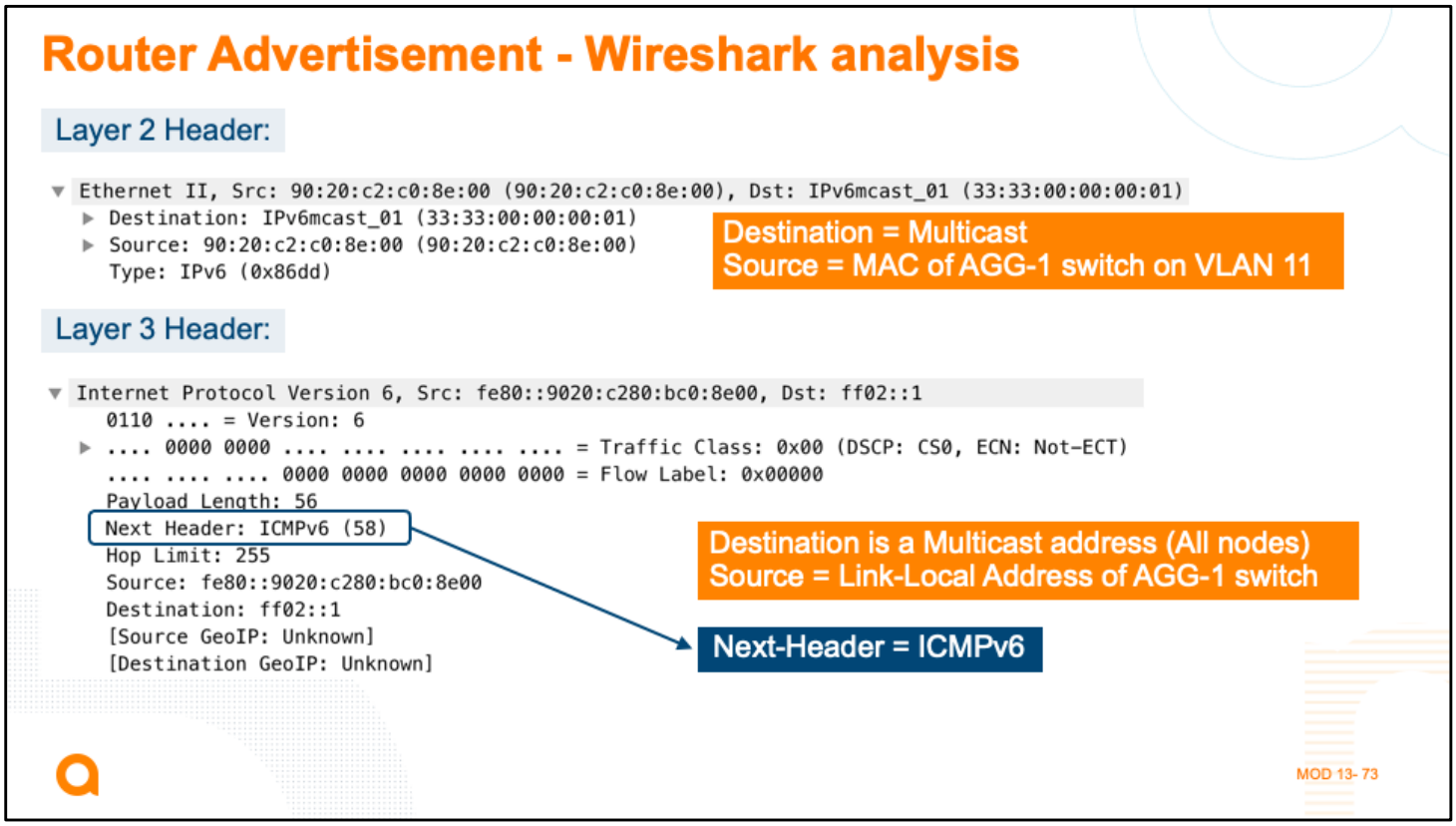
IPv6 devices automatically create an IPv6 link-local address. You know addresses must be unique, so DAD must run. If the result is a duplicated address, then the state is DUPLICATE. If there is no other device with the same Link-Local address, then DAD verifies that the address is unique.

The device receives a Router Advertisement and generates a Global address using SLAAC. DAD validates address uniqueness as before. If DAD verifies that this address is unique then it moves to VALID/PREFERRED state.

The Preferred Lifetime and Valid Lifetime counters engage. Remember that in this state new sessions can be originated using the IPv6 address as source.

When the Preferred Lifetime expires the state moves to VALID/DEPRECATED state. On this state, the device stops using the address to generate new sessions.

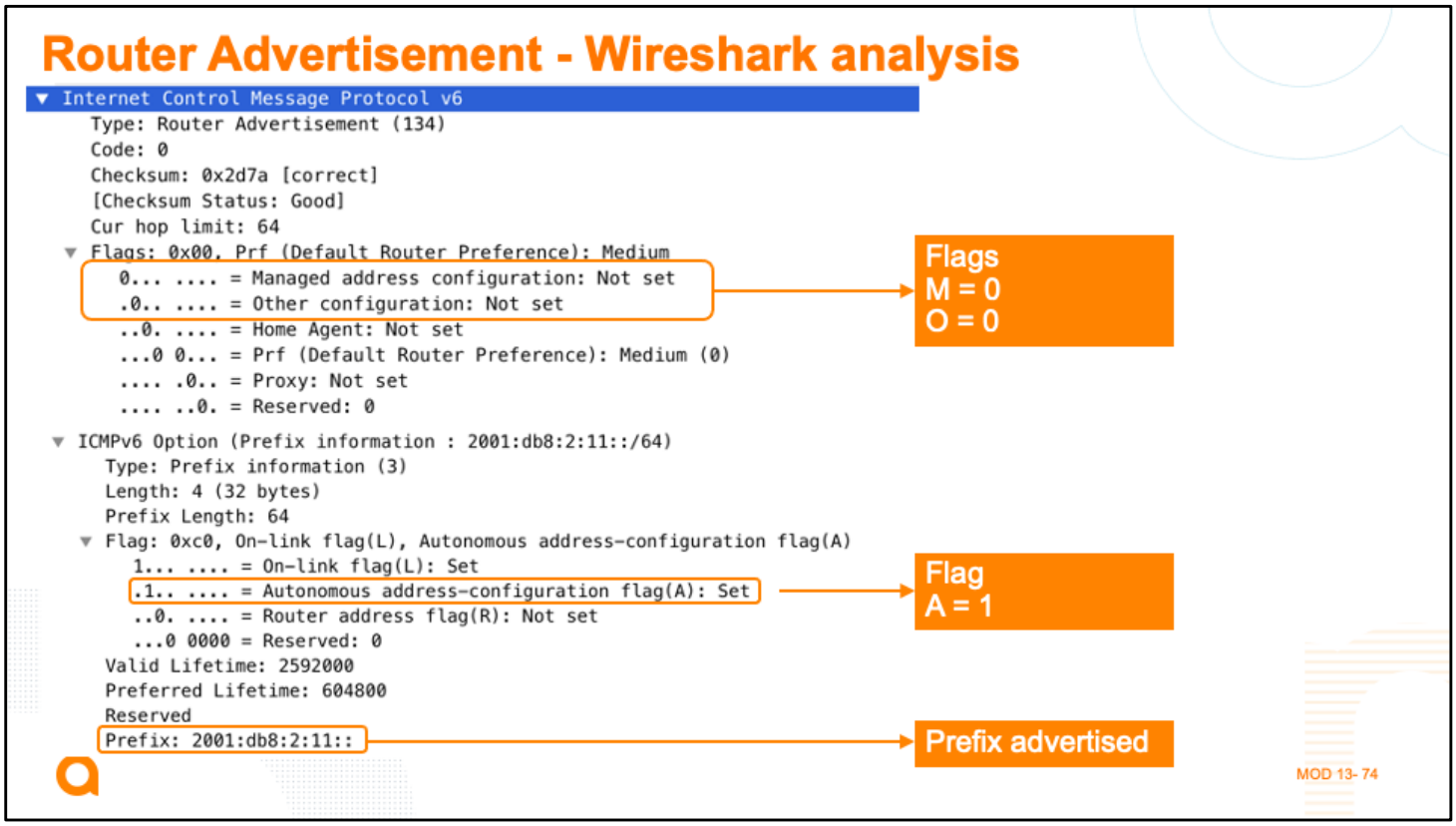
When the Valid lifetime expires the state moves to INVALID state. Here the use of this address is not permitted.



A Wireshark analysis helps to confirm everything that we have learned.

In Layer 2, you see the source and destination MAC address. Noticed that a RA destination is multicast.

In Layer 3, the main IPv6 header points to the next header ICMPv6



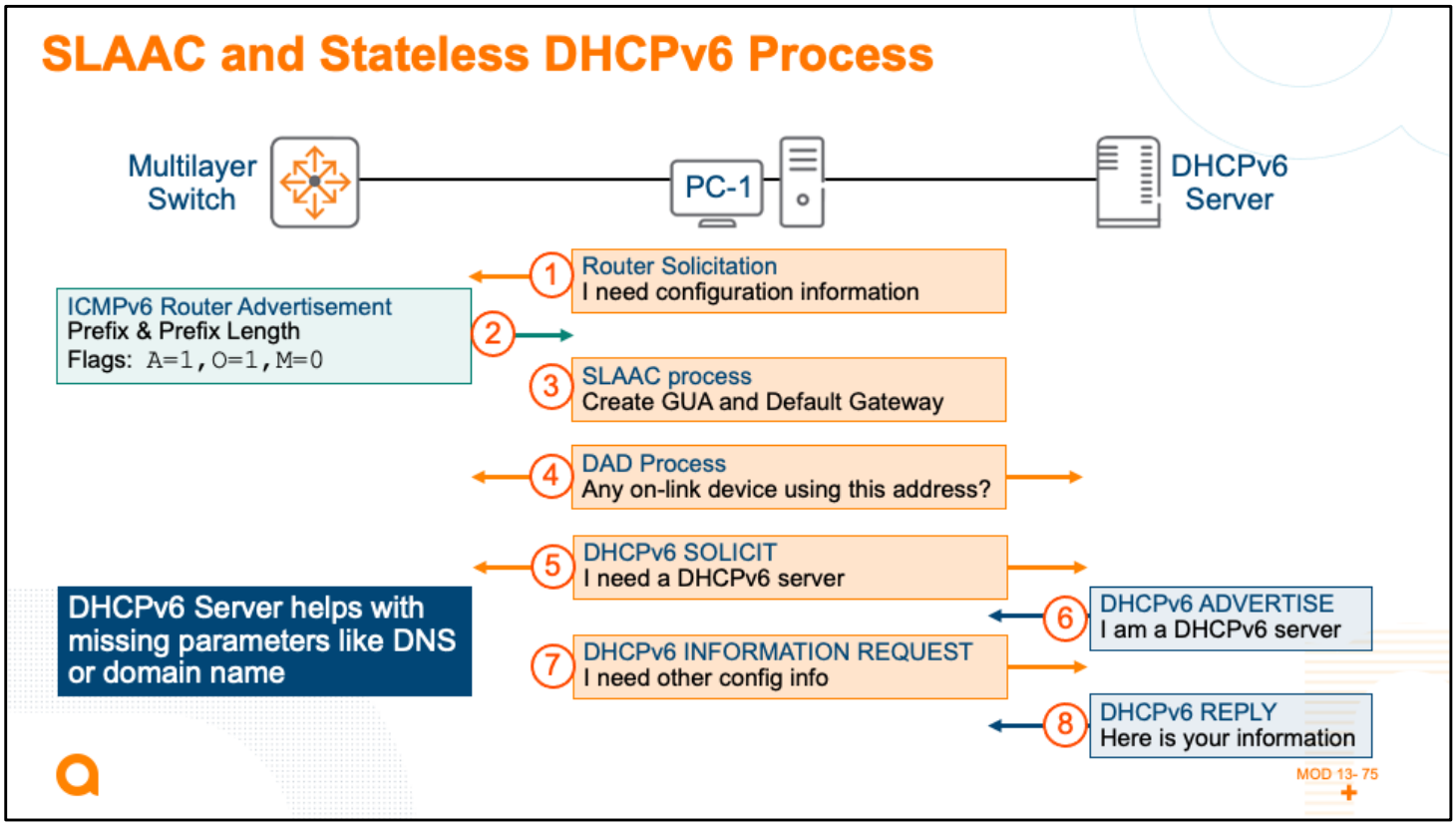
Here you can see the prefix advertised (2001:db8:2:1::) and the flags:

A = 1

M = 0

O = 0

Noticed that in this case MTU is not advertised (AOS-CX does not include this parameter by default).



SLAAC and Stateless DHCPv6 Process:

The station sends a Router Solicitation message.

The Router Advertisement message is sent with the default A flag set to 1 - suggesting that the receiving host use SLAAC. The O flag is set to 1, suggesting that other configuration information is available from a stateless DHCPv6 server. The M flag is not set, indicating that the services of a stateful DHCPv6 server are not needed.

Upon RA receipt, the station uses the RA source IPv6 address as its default gateway. Because the A flag is set to 1, the station performs the SLAAC process:

- Using the prefix in the RA message, it creates the Global Unicast address. The interface ID is generated using either EUI-64 or a random 64-bit value.
- If the station uses the random (privacy extension) method a temporary address is created.

The station performs DAD to ensure that any GUA address created is unique. The station will send out a Neighbor Solicitation message looking for its own IPv6 address, if it does not receive a reply, the it knows the address is unique.

The O-Flag message was set to 1, suggesting that additional information is available from a stateless DHCPv6 server. The station then send outs a DHCPv6 SOLICIT message to ff02::1:2 (all DHCPv6 servers)

One or more DHCPv6 servers respond with an ADVERTISE message indicating that they are available for DHCPv6 service.

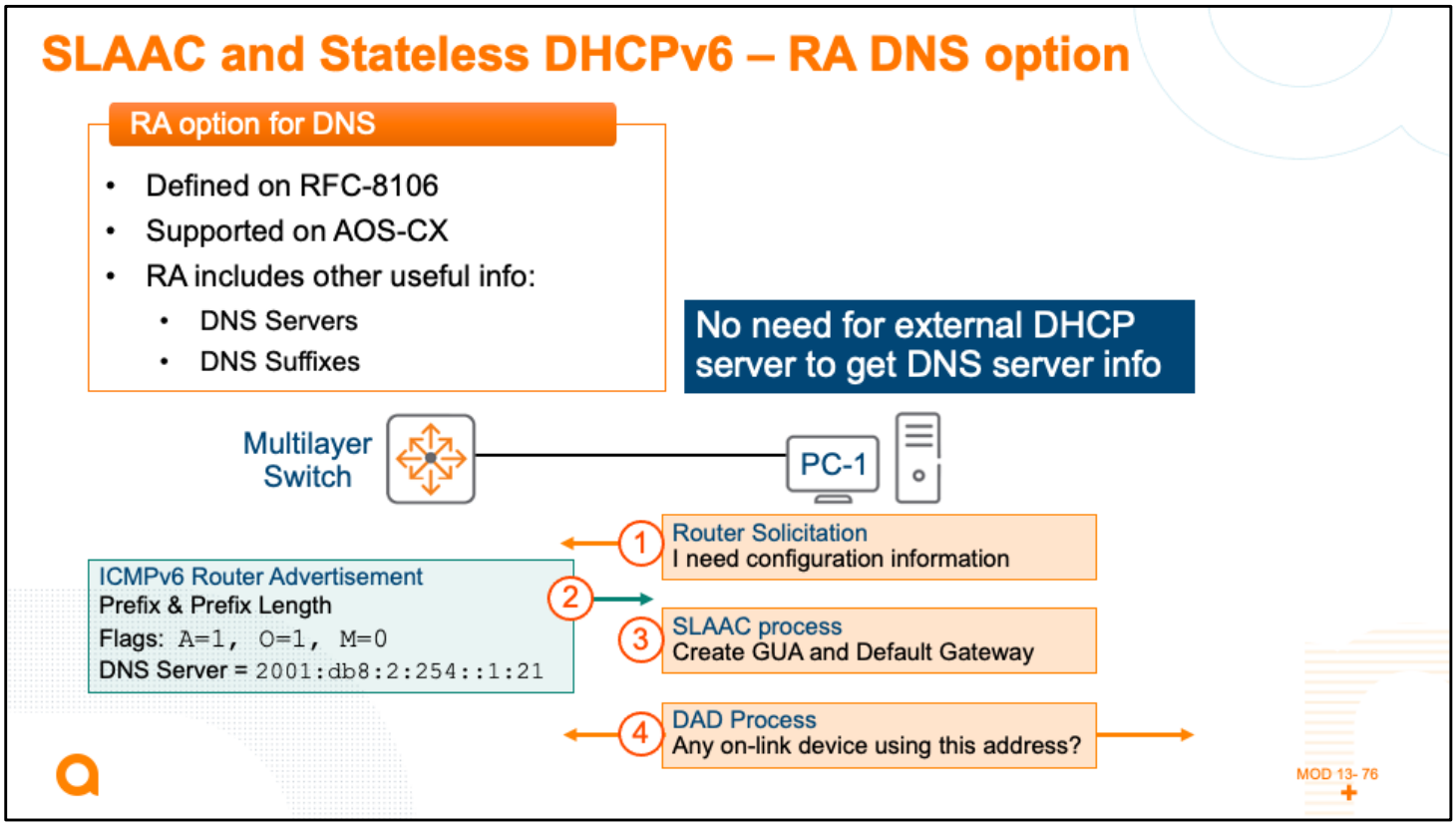
The station responds to the selected server by sending an INFORMATION REQUEST

message, asking for other configuration information.

The DHCPv6 server responds with a REPLY message that contains the information.

As a conclusion, the DHCPv6 server is used to complement the information that cannot be obtained from the SLAAC process.

The DHCPv6 relay process is discussed later, in the stateful DHCPv6 section of this module.



What if the DNS server information is missing information after running SLAAC? Should you deploy a DHCPv6 server – for the sole purpose of providing this one parameter?

Thankfully, AOS-CX supports RFC-8106, which enables RAs to include DNS server and DNS Suffixes. As shown in the figure, the process is shorter. Router Advertisements offer all basic information to a device.

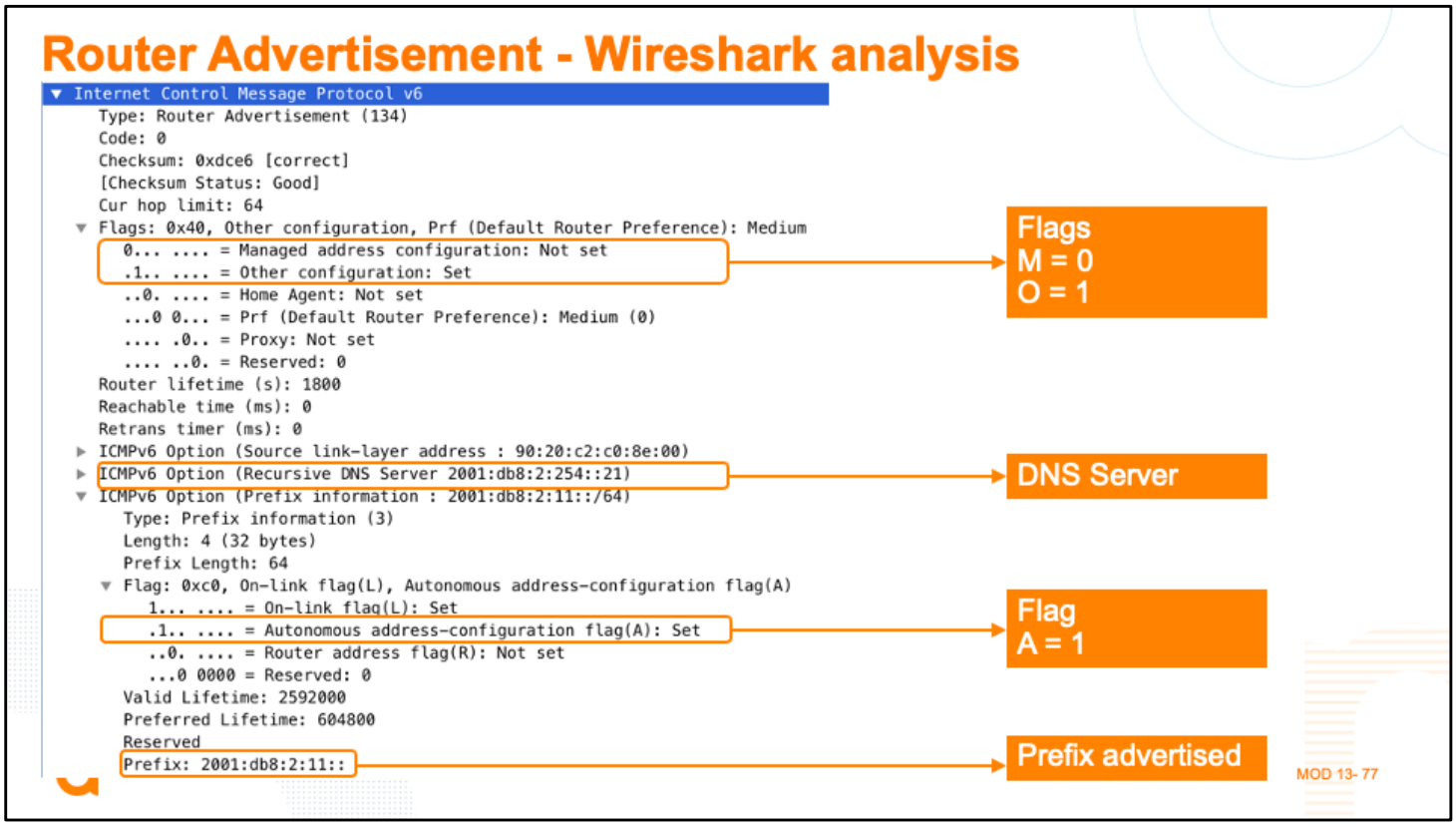
Global Unicast address: obtained from RA

Prefix length: obtained from RA

Default-Gateway: used the source IP address of the RA (Link Local Address of the router)

DNS Server: obtained from RA

Note. A DNS suffix allows for a more friendly interaction with some domain devices. The DNS suffix is automatically appended to all domain names when making a DNS lookup. For example, if a company office has a server called "app.example.com", they can announce DNS suffix "example.com" via DHCP and the same server becomes accessible simply as "app". Whenever you visit http://app/ the OS will automatically try http://app.<DNS_suffix>/ instead, saving employees a bit of unnecessary typing.



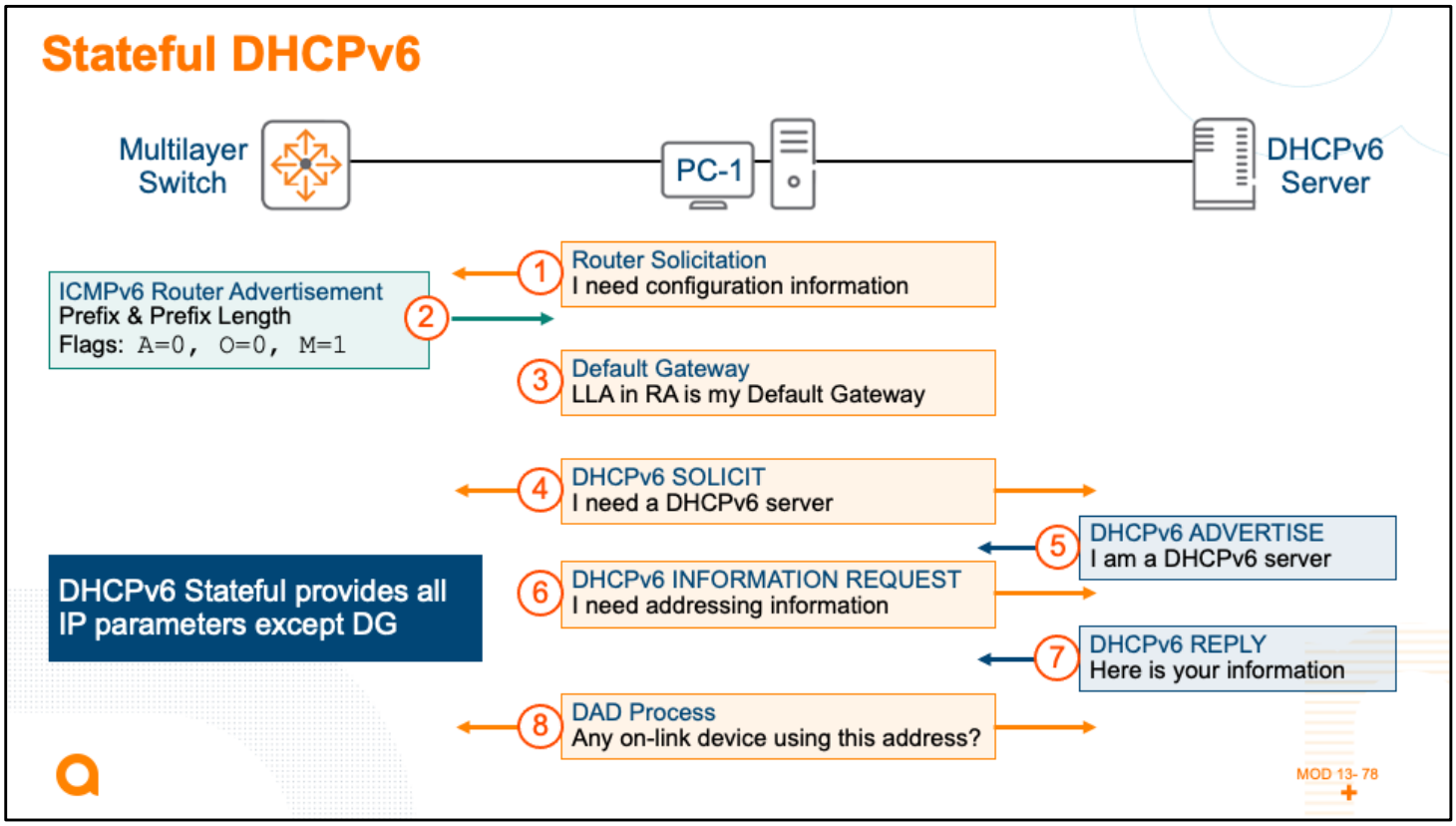
Here you can see the prefix advertised (2001:db8:2:1::) and the flags:

A = 1

M = 0

O = 1

DNS Server is now advertised in the RA.



Stateful DHCPv6 Process:

The station sends a Router Solicitation message.

The RA message is sent with the A and O flags set to 0, the M flag is set to 1, indicating that the services of a stateful DHCPv6 server are needed.

Upon receiving the RA the station uses the source IPv6 address of the RA as its default gateway.

The station then send outs a DHCPv6 SOLICIT message to ff02::1:2 (all DHCPv6 servers).

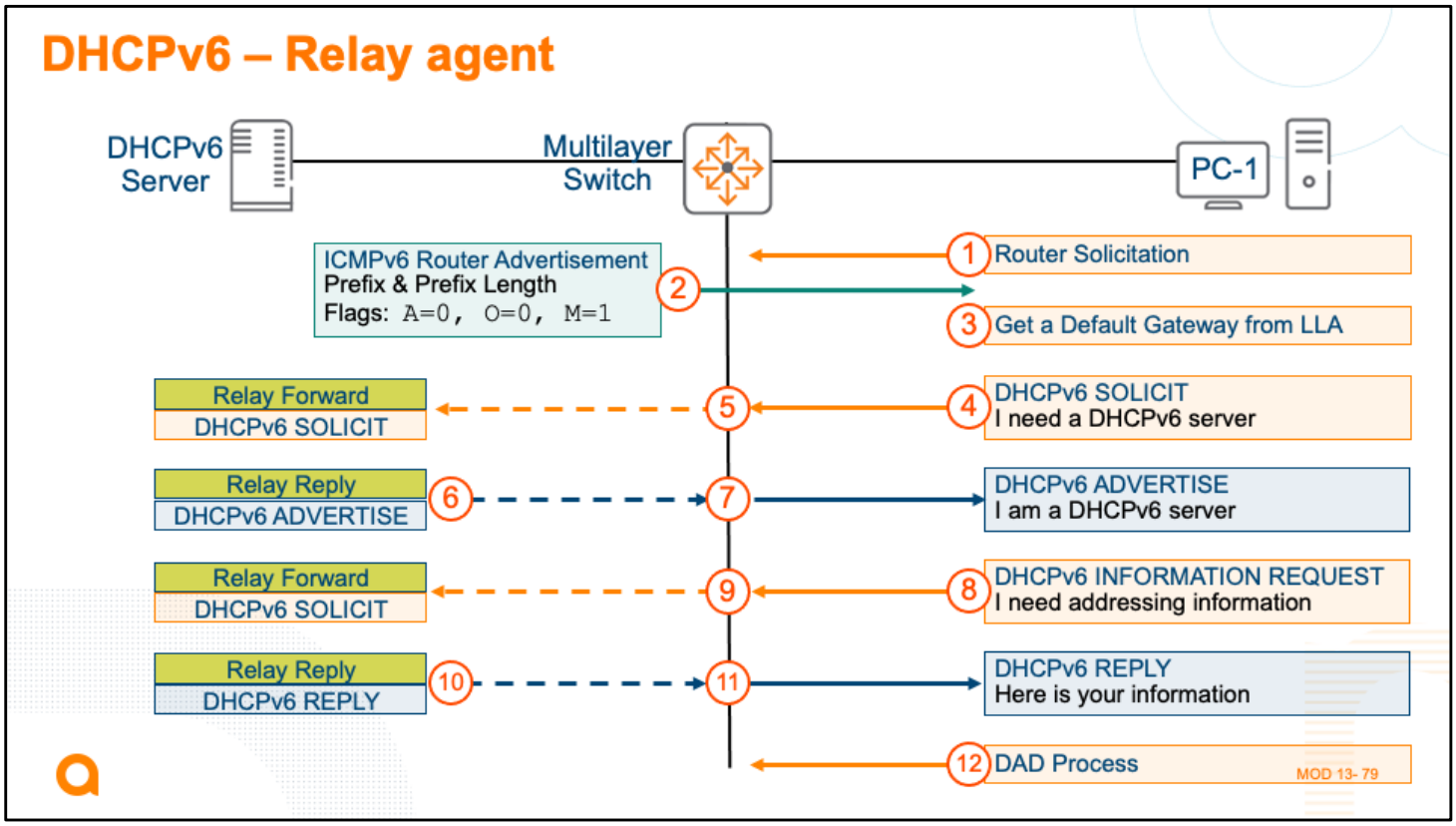
One or more DHCPv6 servers respond with an ADVERTISE message indicating that they are available for DHCPv6 service.

The station responds to the selected server by sending INFORMATION REQUEST message, asking for other configuration information.

The DHCPv6 server responds with a REPLY message that contains the information.

The station perform Duplicated Address Detection (DAD) to ensure that any GUA address received is unique. The station will send out a Neighbor Solicitation message looking for its own IPv6 address, if it does not receive a reply, the it knows the address is unique.

As a conclusion, the DHCPv6 server will offer all parameters to the endpoint.



If the DHCPv6 server is not on the same subnet, you must configure a relay agent.

Stateful DHCPv6 Process with relay agent:

The station sends a Router Solicitation message.

The Router Advertisement message is sent with the A and O flags set to 0, the M flag is set to 1, indicating that stateful DHCPv6 services are needed.

Upon receiving the RA the station uses the RA source IPv6 address as its default gateway.

The station sends a DHCPv6 SOLICIT message to ff02::1:2 (all DHCPv6 servers). In this case the Multilayer switch listens to this port and accepts the message.

The Multilayer switch is configured as a DHCP relay agent. Thus, it forwards a unicast packet to the DHCPv6 server. The switch encapsulates the original SOLICIT message inside a “Relay forward” header.

The server receives the SOLICIT message and responds with an ADVERTISE message, encapsulated inside a “Relay Replay” header.

The multilayer switch receives this message, extracts the original ADVERTISE message and sends it to the endpoint.

The station responds to the selected server by sending INFORMATION REQUEST message, asking for other configuration information.

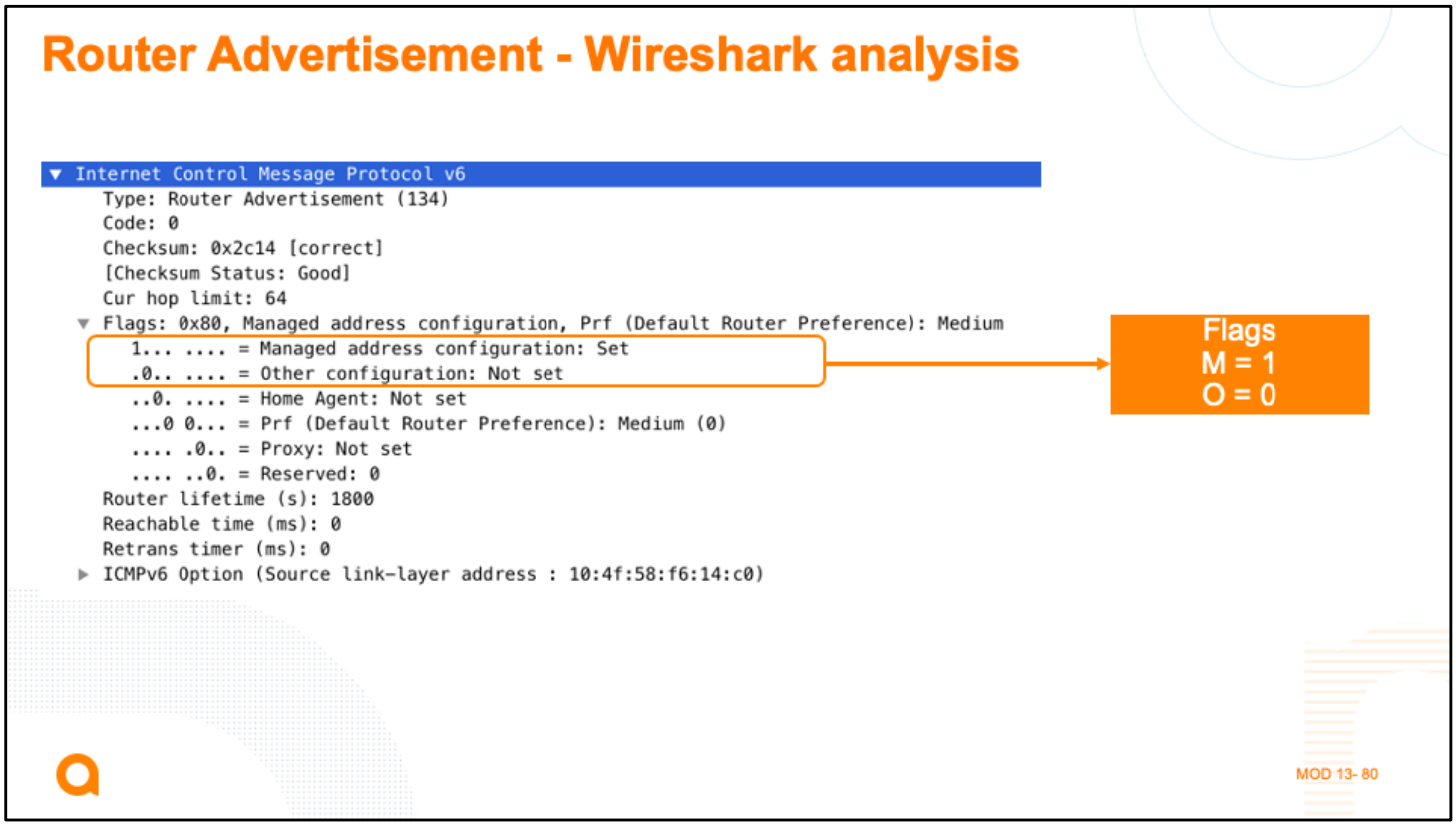
The Multilayer switch creates a RELAY-FORWARD message containing the original

INFORMATION REQUEST message and forwards it to the DHCPv6 server.

The DHCPv6 server responds with a REPLY message which is again encapsulated into a RELAY-REPLY message.

This message is received by the Multilayer switch, which forwards only the REPLY message to the endpoint.

The station performs DAD to ensure that any GUA address received is unique. The station will send out a Neighbor Solicitation message looking for its own IPv6 address, if it does not receive a reply, the it knows the address is unique.

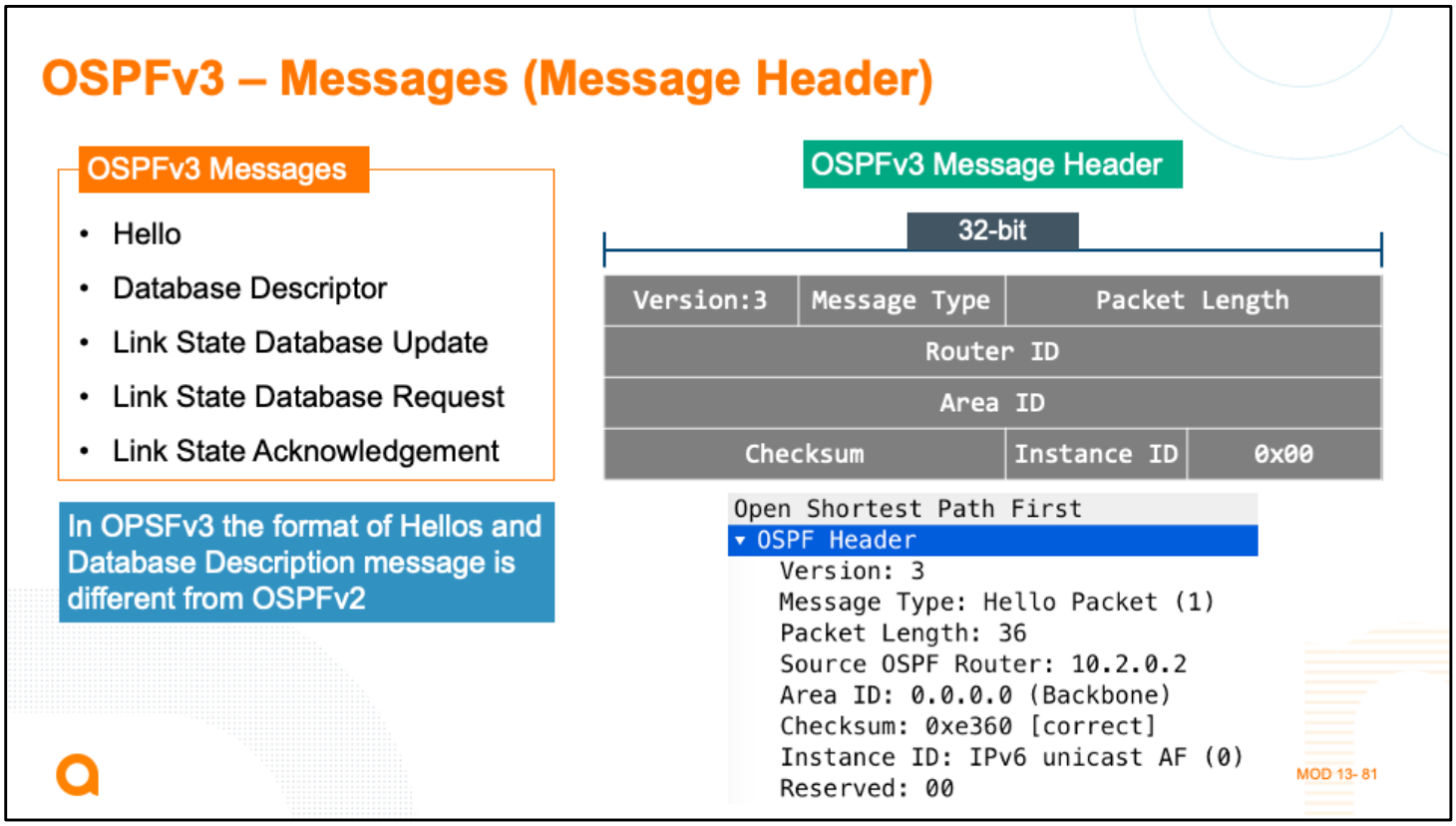


The figure shows a Wireshark capture, where you can see the flags:

A = 0

M = 1

O = 0

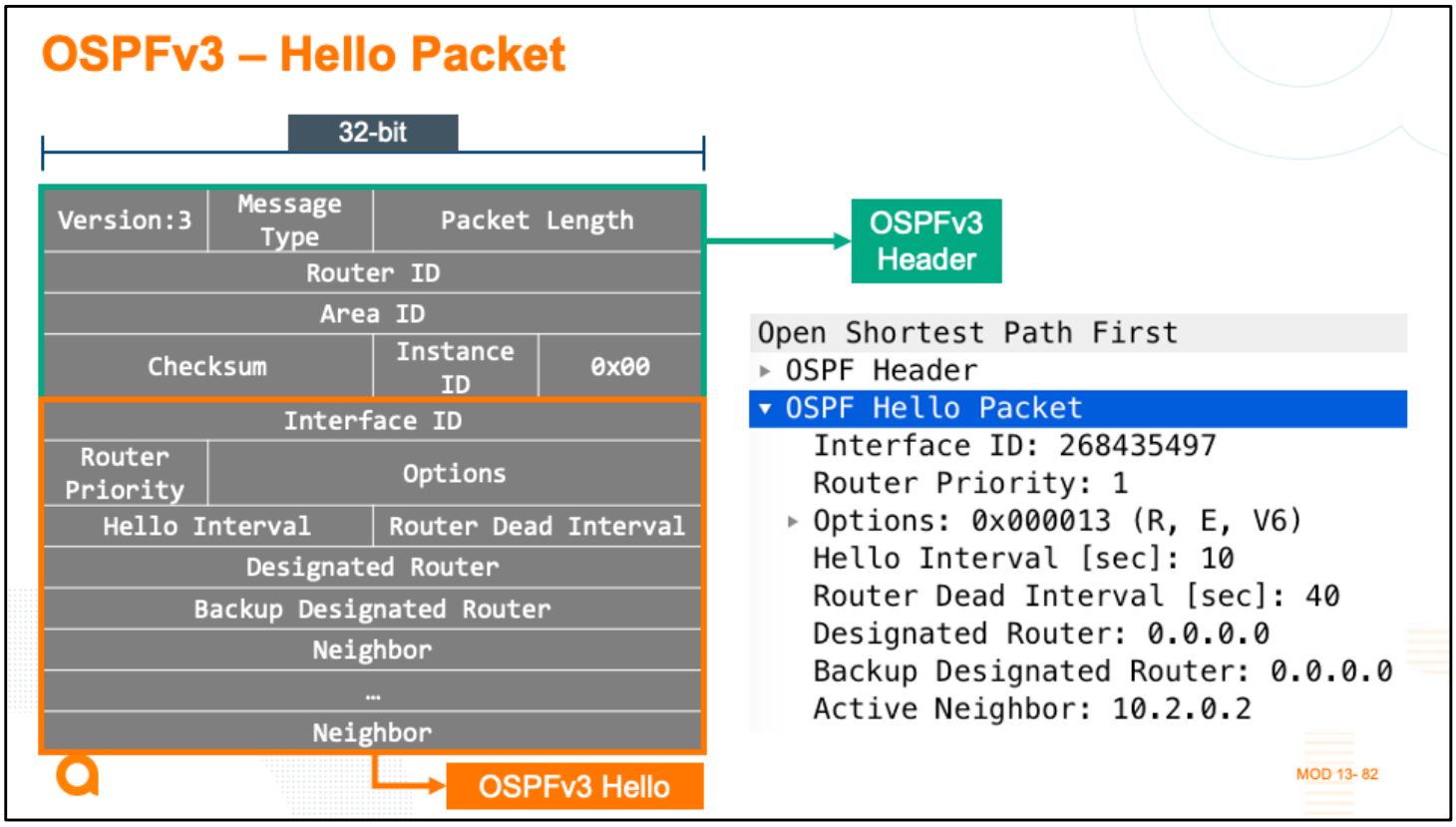


OSPFv3 uses the same five message types as OSPFv2:

Hello
Database descriptor
Link State Database update
Link State Database request
Link State Acknowledgement

The OSPFv3 message header is somewhat different from the OSPFv2 message header. The version number, of course, is 3 rather than 2. But more important, there are no fields for authentication. There is an Instance ID, which allows multiple OSPFv3 instances to run on the same link. The Instance ID has local link significance only, because the OSPFv3 message is not forwarded beyond the link on which is originated.

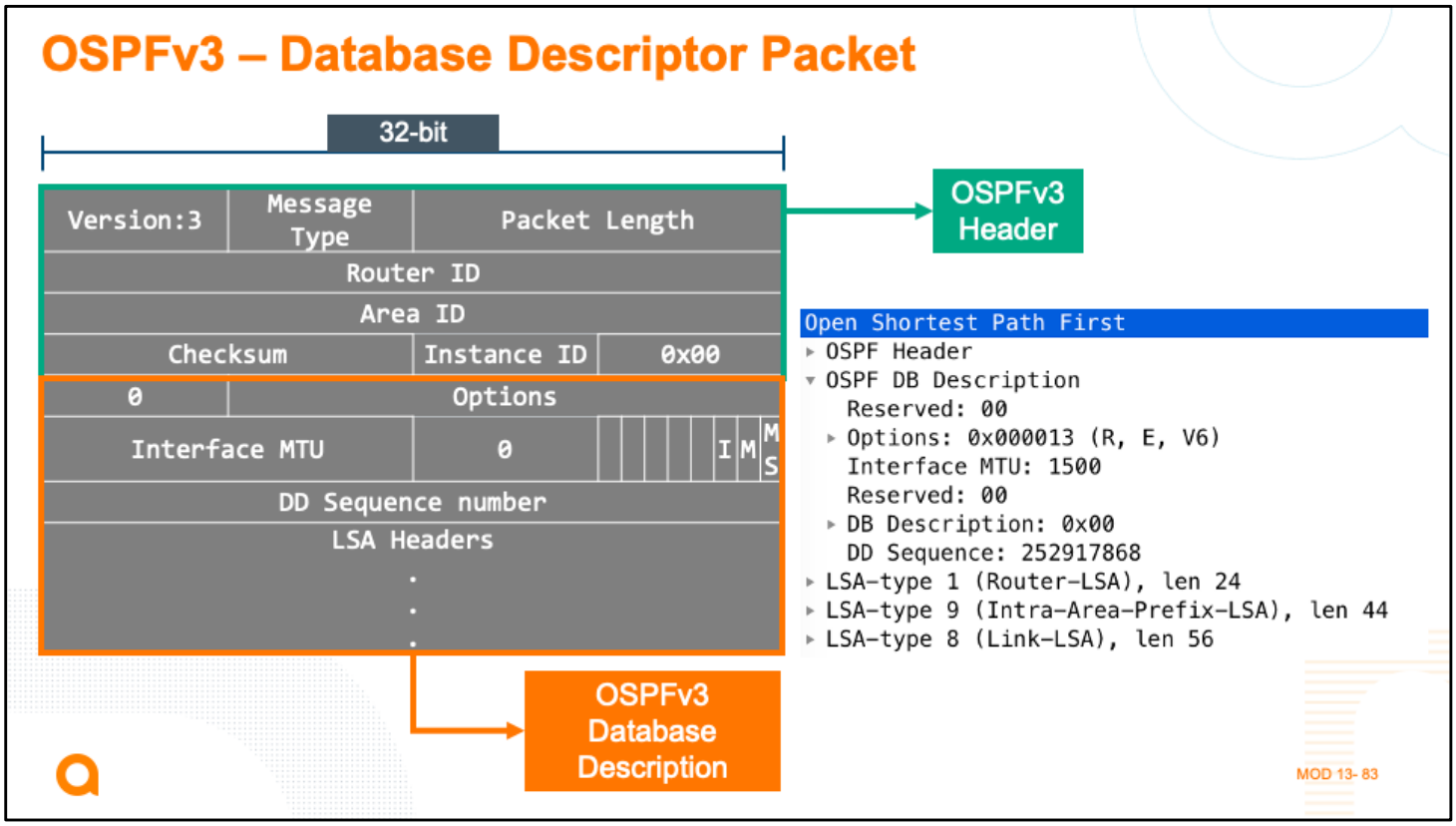
Aside from the message header, the format of only two of the OSPFv3 messages is different from their OSPFv2 counterparts: Hellos and Database Description messages.



The image shows the format of the OSPFv3 Hello message. Unlike OSPFv2, there is no Network Mask field because IPv6 has no need for it. Beyond that, the same fields appear in both packets. The Options field increases in size to 24 bits, and the Router Dead Interval is decreased from 32 bits to 16 bits. The implication of this second filed is that the theoretical maximum Router Dead interval is decreased to 65535 seconds. This change has no bearing on operational networks.

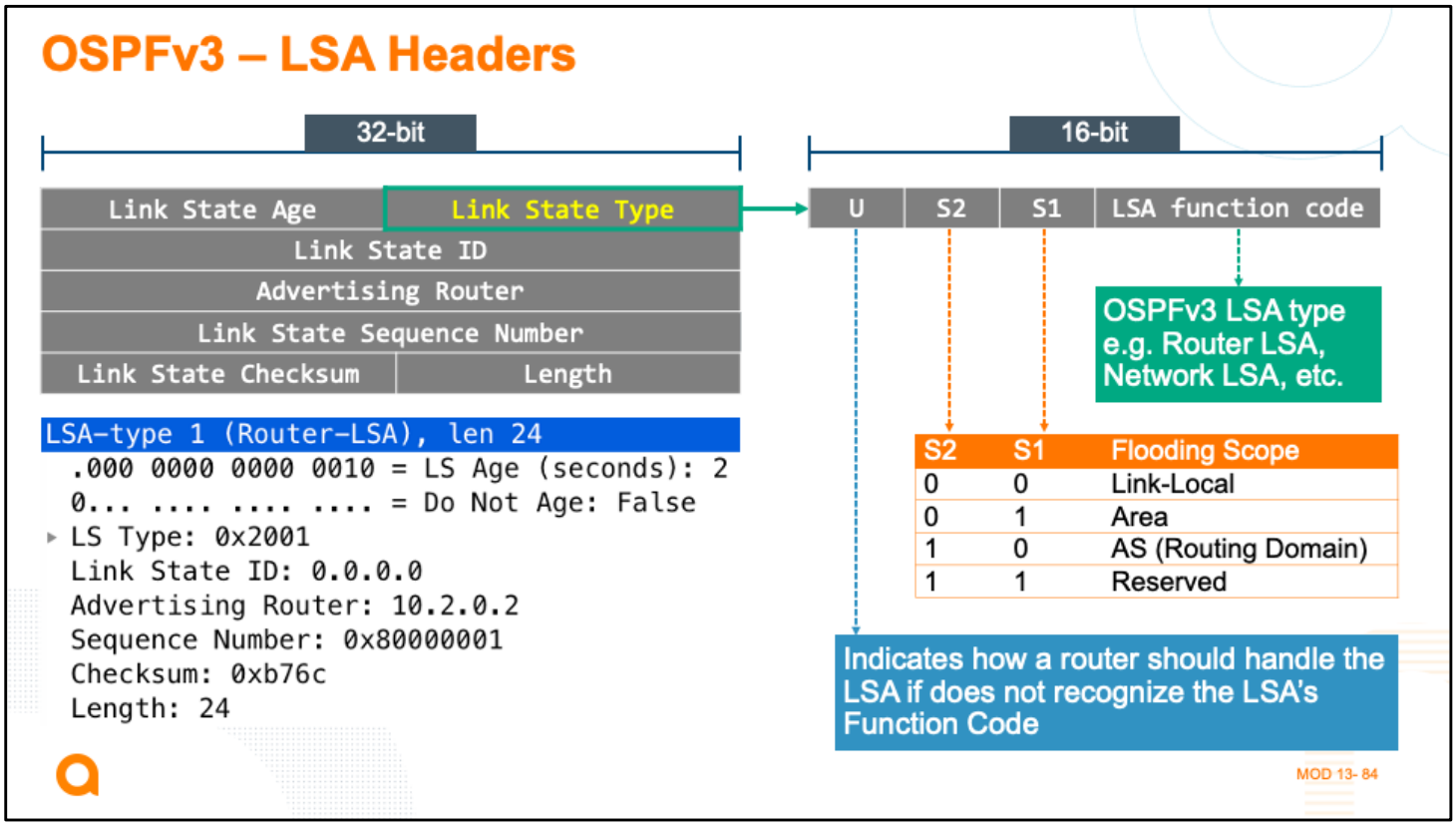
The Interface ID field contains a locally assigned value that identifies the interface that sent the packet over the link. RFC 2740, OSPF for IPv6, suggests that in some implementations, the MIB-II IfIndex (RFC 2863, The Interfaces Group MIB) may be used instead of the link-local address.

Instead of including the IPv4 address of the designated, backup designated, and neighbor routers, the router IDs of each type of router are used. Router ID is a 32-bit number that is assigned locally to each router that is managed by OSPF.



The OSPFv3 Database descriptor packet differs from its OSPFv2 counterpart only in larger Options field.

The other three message types – Link State Request, Link State Update and Link State Acknowledgement – have formats that do not differ from those of OSPFv2 and are not shown in this module.



The OSPFv3 LSA header is shown in the figure. In OSPFv3 The Link State Type field The reason that the Link State Type field is 16 bits rather than 8-bit Type field of OSPFv2. The reason is longer is because it includes three preceding bits, as shown in the figure.

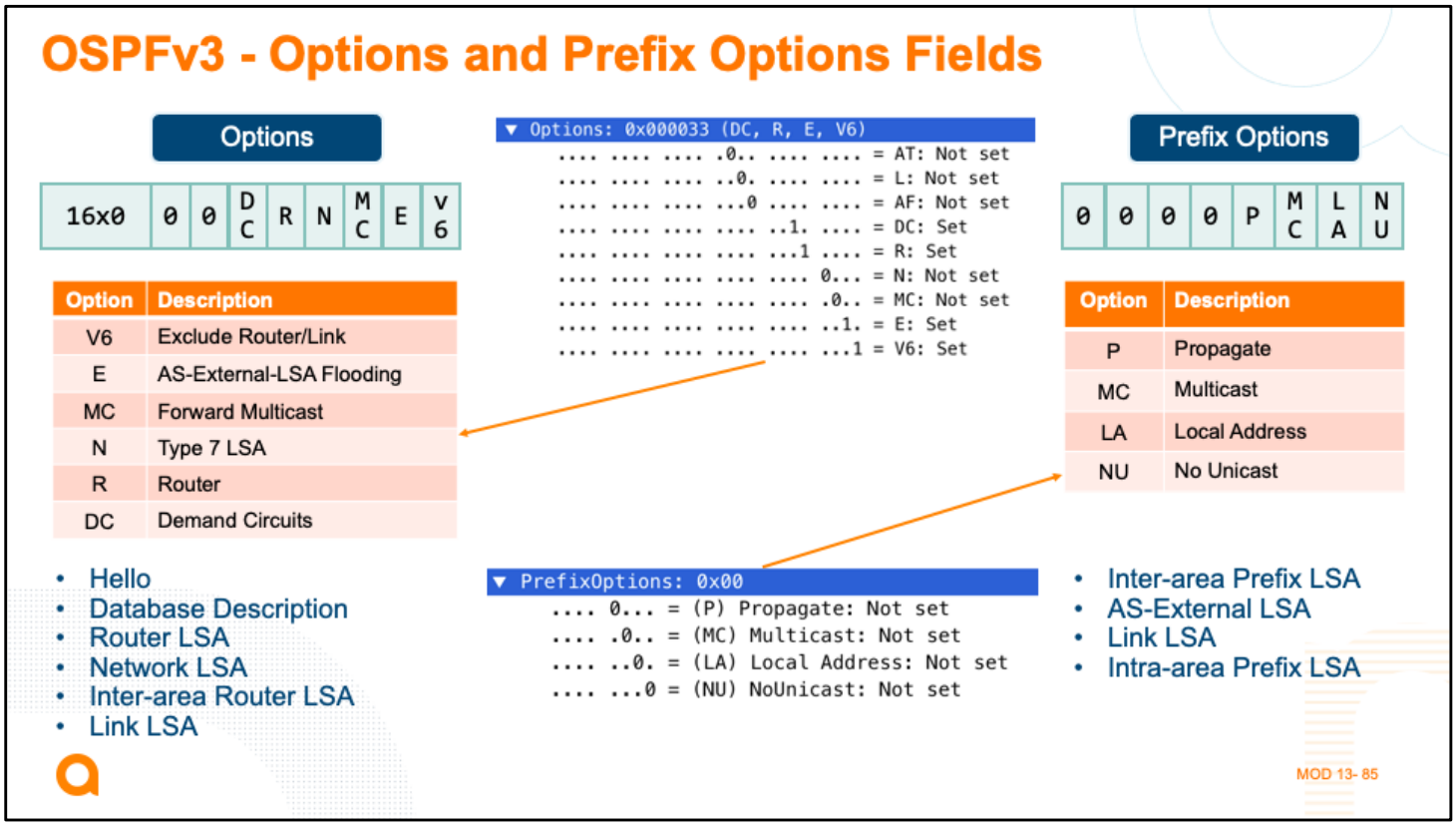
U indicates how a router should handle the LSA if it does not recognize the LSA's Function Code. If the bit is cleared, the unknown LSA is to be treated as if it had link-local flooding scope. If the U bit is set, the unknown LSA is to be stored and flooded as if it is understood.

S2 and S1 indicate the LSA's flooding scope. The next table shows the possible values of these two bits and the associated flooding scopes.

LSA Function Code, the last 13 bits of the LS Type field, correspond to the OSPFv2 Type field.

LS Type	Name
0x2001	Router LSA
0x2002	Network LSA
0x2003	Inter-Area Prefix LSA
0x2004	Inter-Area Router LSA
0x4005	AS-External LSA
0x2006	Group Membership LSA

0x2007	Type-7 LSA
0x0008	Link LSA
0x2009	Intra-Area Prefix LSA



The 24-bit OSPF Options field is present in OSPF Hello packets, Database Description packets, and certain LSAs - router-LSAs, network- LSAs, inter-area-router-LSAs, and link-LSAs. The Options field enables controls OSPF router support for optional capabilities, and to communicate their capability level to other OSPF routers. Through this mechanism, routers of differing capabilities can be mixed within an OSPF routing domain.

An option mismatch between routers can cause a variety of behaviors, depending on the option. Some option mismatches prevent neighbor relationships from forming (e.g., the E-bit below). These mismatches are discovered through sending and receiving Hello packets. Some option mismatches prevent particular LSA types from being flooded across adjacencies. These are discovered by sending and receiving Database Description packets. Some option mismatches prevent routers from being included in one or more of the various routing calculations because of their reduced functionality. These mismatches are discovered by examining LSAs.

Seven bits of the OSPF Options field have been assigned. Each bit is described briefly below.

V6-bit. If this bit is clear, the router/link should be excluded from IPv6 routing calculations [1 Should not] [0 Should]

E-bit. This bit describes the way AS-external-LSAs are flooded, for the formation of stub areas

N-bit. This bit indicates whether the router is attached to an NSSA. [1 Support] [0 No Support]

R-bit. This bit (the ‘Router’ bit) indicates whether the originator is an active router. If the router bit is clear, then routes that transit the advertising node cannot be computed. Clearing the router bit is appropriate for a multi-homed host that wants to participate in routing but does not want to forward non-locally addressed packets.

DC-bit. This bit describes the router's handling of demand circuits. [1 Support] [0 No Support]

Each prefix is advertised along with an 8-bit field of capabilities. These serve as input to the various routing calculations. For example, they can indicate that prefixes are to be ignored in some cases or are to be marked as not re-advertisable in others.

NU-bit. The "no unicast" capability bit. If set, the prefix should be excluded from IPv6 unicast calculations. If not set, it should be included.

LA-bit. The "local address" capability bit. If set, the prefix is actually an IPv6 interface address of the Advertising Router.

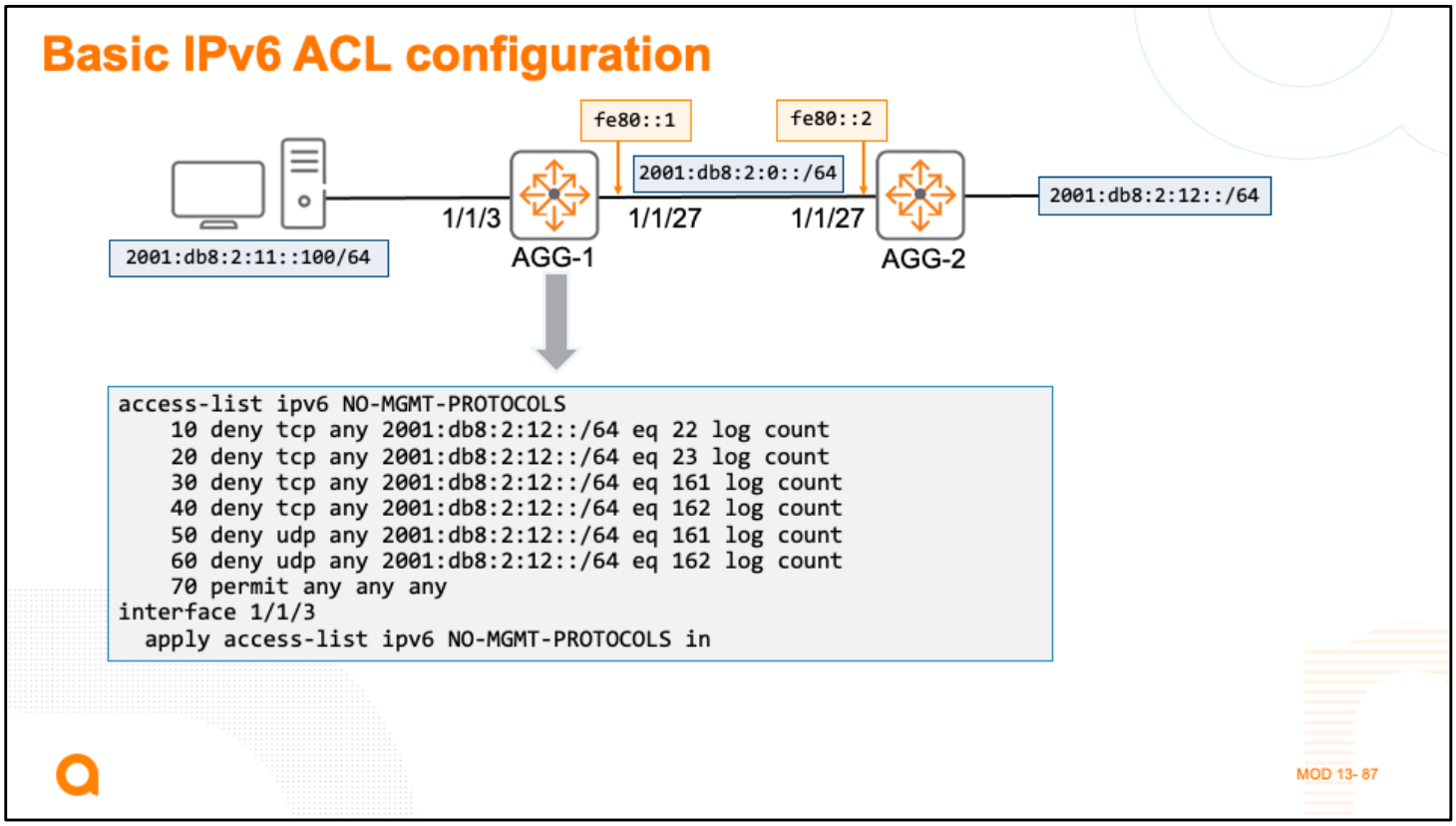
P-bit. The "propagate" bit. Set on NSSA area prefixes that should be re-advertised by the translating NSSA area border.

MC-bit. Multicast bit. If set Multicast prefix should be included in IPv6 multicast routing calculations. If not set, it should not.

aruba
a Hewlett Packard Enterprise company

IPv6 Access Control Lists

Configuration and Verification



ACLs in IPv6 use the same rules on IPv4. The ACL is made of one or more Access Control Entries (ACEs) ordered and prioritized by sequence number. The lowest sequence number is the highest prioritized ACE.

The example shows a simple ACL - created to block SSH, Telnet, and SNMP traffic to any device under the prefix 2001:db8:2:12::/64. As usual, this ACL must be applied on an interface - in this case, inbound on 1/1/3.

Basic IPv6 ACL verification

```
AGG-1# show access-list hitcounts ipv6 NO-MGMT-PROTOCOLS
Statistics for ACL NO-MGMT-PROTOCOLS (ipv6):
Interface 1/1/3* (in):
           Hit Count   Configuration
                   6   10 deny tcp any 2001:db8:2:12::/64 eq 22 log count
                   0   20 deny tcp any 2001:db8:2:12::/64 eq 23 log count
                   0   30 deny tcp any 2001:db8:2:12::/64 eq 161 log count
                   0   40 deny tcp any 2001:db8:2:12::/64 eq 162 log count
                   0   50 deny udp any 2001:db8:2:12::/64 eq 161 log count
                   0   60 deny udp any 2001:db8:2:12::/64 eq 162 log count
                   -   70 permit any any any log
```

```
AGG-1# show logging -r
2020-07-11T12:43:40.848785-04:00 AGG-1 ops-switchd[679]:
Event|10002|LOG_INFO|MSTR|1|NO-MGMT-PROTOCOLS on 1/1/3 (in):          5 10 deny tcp
any 2001:db8:2:12::/64 eq 22 log count
```



MOD 13- 88

To verify if the ACL is working, use show access-list hitcounts. If the ACE has specified a log, then you display records with the show logging –r command.


Lab Activity
Lab 13

3333 Scott Blvd, Santa Clara, CA 95054
TEL: 408.227.4500 | FAX: 408.227.4550
www.ARUBANETWORKS.com

EDU-ASTS-ILT-v20.41