



CCIE Routing and Switching Written Exam Study Guide

for the CCIE Routing and Switching Written Exam version 4.0



Email

sales@ccbootcamp.com

Phone

1.877.NLI.CCIE (654.2243)

Int'l: +1 702.968.5100

Website

www.ccbootcamp.com

Forums

www.routerie.com

www.securityie.com

www.voiceie.com



CCBOOTCAMP'S CCIE Routing and Switching Written Exam Study Guide

for the CCIE Routing and Switching Written Exam version 4.0

For questions about this workbook please visit: www.routerie.com

CCBOOTCAMP

375 N. Stephanie Street
Building 21, Suite 2111
Henderson, NV 89014
1.877.654.2243 Toll Free

www.ccbootcamp.com

"Cisco," the "Cisco Logo," "CCNA," "CCNP," "CCDP," "CCDA," "CCIE," "Cisco Certified Network Associate," "Cisco Certified Design Professional," "Cisco Certified Design Associate," "and "Cisco Certified Network Professional," are registered trademarks of Cisco Systems, Inc. The contents contained wherein, is not associated or endorsed by Cisco Systems, Inc.

PLEASE READ THIS SUBSCRIPTION LICENSE AGREEMENT CAREFULLY BEFORE USING THIS PRODUCT. THIS SUBSCRIPTION LICENSE AGREEMENT APPLIES TO **CCBOOTCAMP's CCIE Routing and Switching Written Exam Study Guide**.

BY ORDERING THIS PRODUCT YOU ARE CONSENTING TO BE BOUND BY THIS LICENSING AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS LICENSE, THEN DO NOT PURCHASE THIS PRODUCT.

License Agreement

CCBOOTCAMP's CCIE Routing and Switching Written Exam Study Guide is copyrighted. In addition, this product is at all times the property of CCBOOTCAMP, and the customer shall agree to use this product only for themselves, the licensed user. The license for the specific customer remains valid from the purchase date until they pass their CCIE Routing and Switching written exam.

CCBOOTCAMP's CCIE Routing and Switching Written Exam Study Guide is licensed by individual customer. This material cannot be resold, transferred, traded, sold, or have the price shared in any way. Each specific individual customer must have a license to use this product. The customer agrees that this product is always the property of CCBOOTCAMP, and they are just purchasing a license to use it. A Customer's license will be revoked if they violate this licensing agreement in any way.

Copies of this material in any form or fashion are strictly prohibited. If for any reason a licensed copy of this material is lost or damaged a new copy will be provided free of charge, except for the cost of printing, shipping and handling.

Individuals or entities that knowingly violate the terms of this licensing agreement may be subject to punitive damages that CCBOOTCAMP could seek in civil court. Damages will be limited to a maximum of \$500,000.00 per individual and \$2,000,000.00 per entity. In addition, individuals or entities that knowingly violate the terms of this license agreement may be subject to criminal penalties as are allowed by law.

The venue of any dispute, controversy, litigation or proceeding (formal or informal) arising out of or pertaining to this licensing agreement or the subject hereof shall lie exclusively in the County of Clark, State of Nevada. Provided, however, that if any such dispute, controversy, litigation or proceeding requires or permits jurisdiction in a federal court or agency of the United States, then venue shall lie in no federal court or agency other than those located in (or nearest to) the County of Clark, State of Nevada.

Term and Termination of License Agreement

This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of written and electronic material of said product. Customer's rights under this License will terminate immediately without notice from CCBOOTCAMP, if Customer fails to comply with any provision of this License. Upon termination, Customer must destroy all copies of material in its possession or control. The license for the specific user remains valid from the purchase date until the user passes their lab exam pertaining to the purchased subscription. Once the customer passes the relevant lab exam the license is terminated and all material written or electronic in their possession or control must be destroyed or returned to CCBOOTCAMP.

Warranty

No warranty of any kind is provided with this product. There are no guarantees that the use of this product will help a customer pass any exams, tests, or certifications, or enhance their knowledge in any way. The product is provided on an "AS IS" basis. In no event will CCBOOTCAMP, its suppliers, or licensed resellers be liable for any incurred costs, lost revenue, lost profit, lost data, or any other damages regardless of the theory of liability arising out of use or inability to use this product.

Chapter 1..... 25

Implement Layer 2 Technologies..... 25

Implement Spanning Tree Protocol (STP) 25

Spanning-Tree Protocol (STP) 802.1d 25

Types of STP on Cisco Switches 25

Redundancy Without Loops 28

Root Bridges and Switches 29

Bridge Protocol Data Units (BPDUs) 29

Operation under STP 34

STP Step-by-Step..... 35

Port State Progression in STP..... 36

STP Broadcast Domain Characteristics 37

802.1w Rapid Spanning Tree Port States..... 38

802.1w Summary Table 38

802.1w Rapid Spanning Tree Port Roles 38

802.1w vs. 802.1d 39

Topology Changes – TCN, TCA, and TC 40

Spanning-Tree Configuration 40

STP Timers..... 42

Spanning-Tree Features 43

Storm Control..... 50

<i>Unicast Flooding</i>	52
Implement VLAN and VLAN Trunking Protocol (VTP)	53
<i>Virtual LAN (VLAN)</i>	53
<i>VLAN Trunk Protocol (VTP)</i>	54
<i>VTP Modes</i>	55
<i>VTP Advertisements</i>	56
<i>VTP message types:</i>	57
<i>VTP Pruning</i>	59
Implement Trunk and Trunk Protocols, EtherChannel, and Load-Balance	59
<i>Trunking Overview</i>	59
<i>Trunking Modes</i>	60
<i>Trunking Encapsulation Types</i>	61
<i>Trunking Configuration</i>	62
<i>EtherChannel</i>	62
<i>Port-Channel Interfaces</i>	64
<i>Port Aggregation Protocol</i>	65
<i>Link Aggregation Control Protocol</i>	66
<i>Load Balancing and Forwarding Methods</i>	68
Implement Ethernet Technologies	69
<i>Speed and Duplex</i>	69
<i>Ethernet, Fast Ethernet, and Gigabit Ethernet</i>	70

PPP over Ethernet (PPPoE) 71

Implement Switched Port Analyzer (SPAN), Remote Switched Port Analyzer (RSPAN), and Flow Control 74

SPAN 75

Remote SPAN 76

Flow Control (IEEE 802.3x) 77

Implement Frame Relay 78

Frame Relay Devices 79

Types of Circuits 80

Data Link Connection Identifier (DLCI)..... 81

Local Management Interface (LMI) 81

DLCI Capacity Calculation..... 82

Encapsulation 83

Speed Elements 84

Congestion 84

Frame Relay Error Checking 85

Frame Relay Compression 86

Frame-Relay Mapping 86

Other Frame Relay Issues 88

Frame Relay Adaptive Traffic Shaping 88

Split Horizon and Frame Relay Interfaces 89

Implement High-Level Data Link Control (HDLC) and PPP 91

<i>High-Level Data Link Control (HDLC)</i>	91
<i>HDLC Configuration</i>	92
<i>Point-to-Point Protocol (PPP)</i>	93
<i>Multilink PPP</i>	94
<i>MLP Interleaving</i>	95
<i>PPP Configuration with CHAP Authentication</i>	95
<i>Microsoft Point-to-Point Compression (MPPC) Configuration</i>	96
<i>Multilink PPP Configuration on Synchronous Serial Interfaces</i>	96
<i>MLP Interleaving and Queueing for Real-Time Traffic Configuration</i>	98
Chapter 2	100
Implement IPv4	100
Implement IP version 4 (IPv4) Addressing, Subnetting, and Variable Length Subnet Masking (VLSM)	100
<i>IP Addressing</i>	100
<i>Subnetting</i>	102
<i>CIDR and VLSM</i>	103
Implement IPv4 Tunneling and Generic Routing Encapsulation (GRE)	104
<i>Example GRE Configuration:</i>	104
Implement IPv4 RIP version 2 (RIPv2)	105
<i>Timers</i>	107
<i>RIP Unicast Updates</i>	107
<i>Route Summarization</i>	108

Split Horizon..... 109

Split Horizon in a Hub and Spoke Network..... 109

Split Horizon Example..... 110

Implement IPv4 Open Shortest Path First (OSPF) 112

Other OSPF Features: 113

OSPF Traffic Types: 113

OSPF Metrics 113

Passive OSPF Interface 115

OSPF Multicast Addresses..... 115

Default Routes 115

OSPF Timers..... 115

OSPF LSAs 116

Types of OSPF Areas..... 117

OSPF Route Preference 119

OSPF Neighbor States..... 119

Adjacencies on Point-to-Point Interfaces 122

Adjacencies on Non-Broadcast Multi-Access (NBMA) Networks 122

Point-to-Multipoint Interfaces 123

OSPF Graceful Restart..... 124

Route Summarization between OSPF Areas 126

Route Summarization When Redistributing Routes into OSPF..... 127

Virtual Links 127

OSPF Troubleshooting 127

Implement Enhanced Interior Gateway Routing Protocol (EIGRP) 128

EIGRP and Split Horizon 130

Types of EIGRP Successors..... 130

Feasibility Condition 130

Attributes of EIGRP 131

EIGRP Tables..... 132

Choosing routes 132

Init Flag 134

EIGRP Stub Routing..... 135

Simple Hub and Spoke Network 137

Route Summary..... 138

Auto-Summarization..... 138

Process ID for an Autonomous System..... 139

Show IP Route EIGRP 139

Show IP EIGRP Topology..... 140

Show Ip Eigrp Neighbor 143

Implement Border Gateway Protocol (BGP) 144

Situations that may require BGP: 145

Interior Border Gateway Protocol (IBGP) 146

Exterior Border Gateway Protocol (EBGP)..... 146

BGP Attributes 146

Weight Attribute..... 147

Local Preference Attribute 148

Multi-Exit Discriminator Attribute 150

Origin Attribute..... 152

AS_Path Attribute..... 152

Next-Hop Attribute 154

Community Attribute 155

Cluster-List..... 155

Originator ID 156

BGP Neighbor Connectivity 156

Synchronization/Full Mesh 158

Next-Hop-Self Command..... 158

Private AS numbers..... 159

BGP Path Selection 159

Scalability Problems with Internal BGP (IBGP) 160

Peer Groups 161

Confederations..... 162

Route Reflectors..... 162

Route Summary..... 162

BGP Clusters..... 164

Route Maps 164

No Export..... 164

Route Dampening..... 164

<i>Backdoor</i>	165
<i>Enabling BGP Routing</i>	165
<i>Controlling BGP Routes</i>	166
<i>RIB-Failure</i>	167
Implement Policy-Based Routing	167
<i>Policy-Based Routing Benefits</i>	168
<i>Data Forwarding Using Policy-Based Routing</i>	169
<i>Tagging Network Traffic</i>	170
<i>Applying Policy-Based Routing</i>	170
<i>Policy Route Maps</i>	170
<i>Match Clauses Define the Criteria</i>	171
<i>Set Clauses Define the Route</i>	172
<i>Source-Sensitive and Equal-Access Routing</i>	173
<i>Configuration Examples</i>	174
Implement Performance Routing (PfR) and Cisco Optimized Edge Routing (OER)	175
<i>OER Overview</i>	176
<i>OER Network Performance Loop</i>	177
<i>The five OER phases:</i>	177
<i>OER Components</i>	178
<i>Configuration Example</i>	179
Chapter 3	181

Implement IPv6 181

**Implement IP version 6 (IPv6) Addressing and different addressing types
 181**

Internet Protocol Version 6 (IPv6) 181

Unchanged characteristics of Addressing in IPv6 181

Addressing 182

Zero Compression in IPv6 Addresses 183

IPv6 Mixed Notation 183

IPv6 Address Prefix Length Representation 184

IPv6 Address Types 184

Important IPv6 address blocks 186

Implement IPv6 Neighbor Discovery Protocol 186

Host-Router Discovery Functions 187

Host-Host Communication Functions 188

Redirect Function 188

IPv6 ND Functions Compared to Equivalent IPv4 Functions 189

Host-Router Discovery Functions Performed By Routers 190

Host-Router Discovery Functions Performed By Hosts 190

Next-Hop Determination 191

Address Resolution 191

Updating Neighbors Using Neighbor Advertisement Messages 192

Neighbor Unreachability Detection and the Neighbor Cache 192

Duplicate Address Detection 193

Implement Basic IPv6 Functionality Protocols **193**

ICMP 193

Cisco Discovery Protocol 194

DNS 194

Unicast Reverse Path Forwarding 194

Path MTU Discovery..... 195

Implement Tunneling Techniques **196**

IPv6 Manually Configured Tunnels 197

IPv6 over IPv4 GRE Tunnels 197

Automatic 6to4 Tunnels 198

Automatic IPv4-Compatible IPv6 Tunnels 198

*The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP Tunnels)*198

Configuration Examples: 199

Manual IPv6 Tunnels Example..... 199

GRE Tunnel Running IS-IS and IPv6 Traffic Configuration Example:.. 200

6to4 Tunnels Configuration Example: 201

IPv4-Compatible IPv6 Tunnels Configuration Example: 202

ISATAP Tunnels Configuration Example:..... 203

Implement OSPF version 3 (OSPFv3) **203**

LSA Types for IPv6 205

NBMA in OSPF for IPv6 206

Importing Addresses into OSPF for IPv6 207

OSPF for IPv6 Authentication Support with IPSec 207

OSPF for IPv6 Virtual Links..... 209

OSPFv3 Graceful Restart 210

OSPF v3 Configuration 211

Implement EIGRP version 6 (EIGRPv6) 211

EIGRP Components 213

EIGRP v6 Configuration..... 215

Implement Filtering and Route Redistribution 216

Access Control Lists for IPv6 Traffic Filtering..... 216

Access Class Filtering in IPv6 216

Configuration Examples 217

Chapter 4..... 218

Implement MPLS Layer 3 VPNs..... 218

Multi Protocol Label Switching (MPLS)..... 218

MPLS Overview 221

Forwarding Equivalence Class (FEC) 222

Architectural Blocks of MPLS..... 223

Control plane: 223

Data plane: 223

Label Switch Router (LSR)..... 224

Label Switched Path (LSP)..... 225

<i>Label definition</i>	226
<i>Label Format</i>	226
<i>Label imposition/disposition.....</i>	227
<i>Penultimate Hop Popping:</i>	227
<i>Label allocation in Frame-Mode MPLS Networks</i>	229
<i>Label allocation in Cell-Mode MPLS networks</i>	229
<i>Label Distribution</i>	230
<i>Label Distribution Protocol (LDP):.....</i>	231
<i>MPLS Virtual Private Networks</i>	231
<i>VPN Operation</i>	232
<i>VPN Route Target Communities.....</i>	233
<i>MPLS Forwarding.....</i>	235
<i>Route distinguisher (RD)</i>	236
<i>MPLS VPN Virtual Routing/Forwarding Tables.....</i>	239
<i>Distribution of VPN Routing Information in an MPLS VPN</i>	240
<i>BGP Distribution of VPN Routing Information</i>	241
<i>MPLS Forwarding.....</i>	242
<i>Major Components of MPLS VPNs</i>	242
Chapter 5.....	245
Implement IP Multicast	245
<i>Benefits of IP Multicast</i>	246
<i>Communication Frame Types.....</i>	246

<i>IGMP and CGMP Multicast Protocols</i>	248
<i>Designated Querier</i>	250
<i>Querier router election</i>	251
<i>IGMP Versions 1, 2, and 3</i>	251
<i>IP Multicast Group Addressing</i>	254
<i>Calculating a Multicast Address</i>	254
<i>IP Multicast Address Scoping</i>	255
<i>Globally Scoped Addresses</i>	256
<i>Source Specific Multicast Addresses</i>	256
<i>Layer 2 Multicast Addresses</i>	258
<i>IP Multicast Delivery Modes</i>	258
<i>Protocol Independent Multicast (PIM)</i>	259
<i>PIM Commands</i>	261
<i>Bidirectional PIM</i>	261
<i>Rendezvous Points</i>	263
<i>Auto-RP</i>	264
<i>Sparse-Dense Mode for Auto-RP</i>	265
<i>Bootstrap Router</i>	266
<i>Multicast Source Discovery Protocol</i>	267
<i>Anycast RP</i>	268
<i>Multicast Forwarding</i>	269
<i>Multicast Distribution Source Tree (Shortest Path Tree)</i>	270
<i>Multicast Distribution Shared Tree</i>	271

Reverse Path Forwarding (RPF)..... 273

RPF Check..... 274

PIM and Distance Vector Multicast Routing Protocol (DVMRP)..... 275

Sparse Mode with Auto-RP Configuration Example..... 276

Sparse Mode with Bootstrap Router: Example..... 276

Sparse Mode with a Single Static RP: Example..... 277

Implement IPv6 Multicast..... **277**

IPv6 Multicast Groups..... 278

Multicast Listener Discovery Protocol for IPv6 278

Explicit Tracking of Receivers..... 279

IPv6 Multicast Addresses..... 279

IPv6 Multicast Address Format..... 280

IPV6 Multicast..... 281

IP Multicast Terms..... 282

Chapter 6..... **288**

Implement Network Security..... **288**

Implement Access Lists..... **288**

Access Lists..... 288

Sample Standard Access List 290

Sample Extended Access-list 291

Sample Named Access-list..... 293

Implement Zone Based Firewall..... **293**

<i>Zone-Based Policy Configuration Model</i>	<i>294</i>
<i>Rules For Applying Zone-Based Policy Firewall</i>	<i>295</i>
<i>Zone-Based Policy Network Security</i>	<i>296</i>
<i>Cisco Policy Language (CPL) Configuration</i>	<i>297</i>
<i>Configuring Zone-Based Policy Firewall Class-Maps.....</i>	<i>297</i>
<i>Configuring Zone-Based Policy Firewall Policy-Maps.....</i>	<i>298</i>
<i>Zone-Based Policy Firewall Actions</i>	<i>298</i>
<i>Configuring Zone-Policy Firewall Parameter-Maps.....</i>	<i>300</i>
<i>Applying Logging For Zone-Based Policy Firewall Policies.....</i>	<i>300</i>
<i>Private Internet Policy Configuration Example.....</i>	<i>301</i>
Implement Unicast Reverse Path Forwarding (uRPF)	302
<i>Unicast RPF with ACLs and Logging Configuration Example.....</i>	<i>304</i>
Implement IP Source Guard.....	305
Implement Authentication, Authorization, and Accounting (AAA).....	306
<i>AAA Philosophy</i>	<i>309</i>
<i>Benefits of Using AAA</i>	<i>309</i>
<i>Method Lists</i>	<i>309</i>
<i>The First Step, or Where to Begin.....</i>	<i>311</i>
<i>Overview of the AAA Configuration Process</i>	<i>311</i>
Implement Control Plane Policing (CoPP)	312
<i>CoPP Policy Construction and Deployment Concepts</i>	<i>314</i>
<i>CoPP Configuration</i>	<i>315</i>

Implement Cisco IOS Firewall	317
<i>Traffic Filtering.....</i>	<i>317</i>
<i>Traffic Inspection</i>	<i>318</i>
<i>How CBAC Works – Overview</i>	<i>318</i>
<i>Ethernet Interface Configuration Example</i>	<i>319</i>
Implement Cisco IOS Intrusion Prevention System (IPS)	320
<i>The Signature Definition File.....</i>	<i>321</i>
<i>Default Signatures Configuration Example.....</i>	<i>322</i>
<i>Attack-drop.sdf Configuration Example</i>	<i>322</i>
Implement Secure Shell (SSH).....	323
<i>Configuring SSH Server</i>	<i>323</i>
<i>Verifying SSH</i>	<i>325</i>
Implement 802.1x.....	325
<i>Device Roles.....</i>	<i>326</i>
Implement NAT	328
<i>How NAT Works</i>	<i>328</i>
<i>NAT Addresses.....</i>	<i>329</i>
<i>Types of NAT.....</i>	<i>329</i>
<i>Configuring NAT to Allow Internal Users to Access the Internet Using Overloading</i>	<i>330</i>
Implement Routing Protocol Authentication	331
<i>Protocols That Use Neighbor Authentication.....</i>	<i>331</i>

How Neighbor Authentication Works 332

Key Management (Key Chains) 332

RIP Authentication..... 333

OSPF Authentication 334

EIGRP Authentication..... 335

BGP Authentication..... 336

Implement Device Access Control 336

Cisco IOS CLI Modes..... 336

User EXEC Mode 337

Privileged EXEC Mode 338

Global Configuration Mode..... 338

Cisco IOS CLI Sessions 338

Local CLI Sessions..... 338

Remote CLI Sessions 339

Terminal Lines are Used for Local and Remote CLI Sessions 339

Cisco IOS Privilege Levels..... 339

Implement Security Features 340

Private VLANs..... 340

DHCP Snooping..... 344

Dynamic ARP Inspection (DAI)..... 345

Port Security 346

Secure MAC Addresses..... 347

Security Violations..... 349

TCP Intercept 351

Chapter 7..... **353**

Implement Network Services **353**

Implement Hot Standby Router Protocol (HSRP) **353**

Implement Gateway Load Balancing Protocol (GLBP) **357**

Implement Virtual Router Redundancy Protocol (VRRP) **358**

Implement Network Time Protocol (NTP) **360**

Association Modes 364

NTP Architecture 368

Clock Technology and Public Time Servers..... 369

The NTP Server Global Configuration Command 371

Implement Dynamic Host Configuration Protocol (DHCP) **372**

Implement Web Cache Communication Protocol (WCCP)..... **374**

Chapter 8..... **377**

Implement Quality of Service (QoS) **377**

QoS Overview..... 379

Five Benefits for Implementing QoS in the Enterprise Networks 379

How a Converged Network Behaves Without QoS..... 379

QoS framework..... 380

Call Admission Control Functionality 381

Integrated Services vs. Differentiated Services 381

Configure QoS Policy using Modular QoS CLI 383

Classification and Marking 391

Purposes of Classification and Marking 391

Difference between Classification and Marking 392

Class of Service, IP Precedence and DiffServ Code Points 393

Network Based Application Recognition (NBAR)..... 395

Classify and Mark Traffic 395

Congestion Management 399

Identify and Differentiate Between IOS Queuing Techniques 399

Apply Each Queuing Technique to the Appropriate Application..... 403

IP RTP Priority and Low Latency Queuing (LLQ) Differences 405

Configure WFQ, CBWFQ, and LLQ..... 406

Congestion Avoidance 409

Explain How TCP Responds to Congestion 409

Explain Tail Drop and Global Synchronization..... 410

Identify and Differentiate Between: RED, WRED, FRED 411

Configure IOS Congestion Avoidance Features 412

Link Efficiency Tools 413

The Need for Link Efficiency Tools 413

Real Time Protocol Header Compression (CRTP)..... 416

Configure and Monitor Various LFI methods and CRTP 418

Policing and Shaping..... 420

The Difference between Policing and Shaping and How Each Relates to QoS 420

When to Apply and How to Configure Policing Mechanisms 421

Different Types of Traffic Shaping and How to Apply Them 422

Configure the Different Types of Traffic Shaping..... 424

Congestion-Control Mechanisms..... 426

Traffic Shaping Parameters..... 427

Traffic Shaping Calculation 427

First-In, First-Out (FIFO)..... 428

Weighted Fair Queuing (WFQ)..... 429

Priority Queuing 429

Custom Queuing 432

Class-Based Weighted Fair Queuing..... 437

Packet over SONET/SDH (PoS) and IP Precedence..... 437

IP Precedence..... 438

Random Early Detection (RED) 439

Weighted Random Early Detection (WRED)..... 440

Weighted Round-Robin (WRR)/Queue Scheduling..... 441

Class of Service (CoS) 443

Shaping vs. Policing..... 443

Traffic Shaping..... 444

Committed Access Rate (CAR) 447

Network-Based Application Recognition (NBAR) 448

Configuring NBAR..... 450

Differentiated Services Code Point (DSCP)..... 451

<i>Resource Reservation Protocol (RSVP)</i>	453
<i>Load Balancing</i>	454
<i>802.1x and QoS</i>	455
<i>Custom Queuing (CQ)</i>	457
<i>Why Use CQ?</i>	458
<i>Restrictions</i>	458
<i>Configuring a Traffic Policy</i>	459
<i>Attaching a Traffic Policy to an Interface</i>	460
<i>Configuring a Traffic Class with NBAR Example</i>	461
<i>ToS Byte</i>	462
<i>DiffServ Field</i>	462
<i>Differences between Traffic-Shaping Mechanisms</i>	464
<i>CQ and Extended Burst Capability</i>	468
<i>Committed Access Rate (CAR) definition</i>	469
<i>Analysis</i>	472
<i>Connecting from Spoke to Spoke</i>	475
<i>Cisco AutoQoS - VoIP</i>	476
<i>Cisco AutoQoS for the Enterprise</i>	478
<i>Enabling the Auto-Discovery Phase: Example</i>	479
<i>Enabling the AutoQoS Template Generation Phase: Example</i>	479
Chapter 9	480
Troubleshoot a Network	480

<i>Duplex mismatch error</i>	480
<i>Etherchannel</i>	481
<i>Trunking Mode Mismatch</i>	482
<i>Tools for Troubleshooting IP Problems</i>	483
<i>General IP Troubleshooting Suggestions</i>	484
<i>Narrowing Down the Problem Domain</i>	485
<i>Troubleshooting IP Connectivity and Routing Problems</i>	485
<i>Determining Where to Start</i>	486
<i>Check for Resources</i>	487
<i>Check for Connectivity</i>	488
<i>Check for ACLs</i>	489
<i>Check for Network Address Translation</i>	490
<i>Troubleshooting Upper-Layer Problems</i>	490
<i>Generic</i>	491
<i>Hypertext Transport Protocol</i>	492
<i>FTP</i>	493
<i>MAIL (IMAP, POP, and SMTP)</i>	495
<i>Telnet</i>	496
<i>Routing Loops and Split Horizon</i>	497
<i>Troubleshooting routing loop</i>	497
Chapter 10	499
Optimize a Network	499

Implement Syslog and Local Logging	499
<i>Enabling System Message Logging</i>	<i>500</i>
<i>Setting the Syslog Destination</i>	<i>500</i>
<i>Configuring Synchronization of Logging Messages</i>	<i>501</i>
<i>Enabling Time-Stamps on Log Messages</i>	<i>502</i>
<i>Limiting the Error Message Severity Level and Facilities</i>	<i>502</i>
Implement IP Service Level Agreement (SLA)	504
Implement NetFlow	506
<i>NetFlow Flows.....</i>	<i>507</i>
<i>NetFlow Export Formats.....</i>	<i>508</i>
Implement Router IP Traffic Export (RITE).....	509
<i>Router IP Traffic Export (RITE).....</i>	<i>509</i>
<i>Configuring IP Traffic Export.....</i>	<i>510</i>
Implement Simple Network Management Protocol (SNMP).....	511
<i>SNMP Operations</i>	<i>512</i>
<i>SNMP Versions.....</i>	<i>513</i>
Implement Cisco IOS Embedded Event Manager	515
<i>EEM Policy Creation</i>	<i>516</i>
Implement Remote Monitoring (RMON)	517
Implement FTP.....	519
<i>Configuring a Router to Use FTP Connections.....</i>	<i>519</i>

Implement TFTP.....	520
Implement TFTP Server on Router	521
Implement Secure Copy Protocol (SCP)	522
<i>SCP Server-Side Configuration Using Local Authentication: Example</i>	<i>523</i>
<i>SCP Server-Side Configuration Using Network-Based Authentication: Example.....</i>	<i>523</i>
Implement HTTP and HTTPS	524
<i>Configuring the HTTP 1.1 Web Server: Example.....</i>	<i>524</i>
<i>Configuring HTTP Connection Characteristics for File Transfers: Example.....</i>	<i>526</i>
<i>Downloading a File from a Remote Server Using HTTP or HTTPs: Example</i>	<i>526</i>
<i>Uploading a File from Flash to the Remote HTTP Server: Example</i>	<i>526</i>
<i>Downloading a File from the Remote HTTP Server to Flash Memory: Example.....</i>	<i>527</i>
<i>Uploading a File to a Remote Server Using HTTP or HTTPs....</i>	<i>528</i>
Implement Telnet.....	528
<i>Hiding Telnet Addresses.....</i>	<i>529</i>

Chapter 1

Implement Layer 2 Technologies

Implement Spanning Tree Protocol (STP)

Spanning-Tree Protocol (STP) 802.1d

Spanning-Tree Protocol (STP) is a Layer 2 link management protocol using the Spanning Tree Algorithm (STA) to calculate the best loop-free path through a switched network. STP is designed to run on bridges and switches to provide path redundancy and prevent undesirable loops from forming in the network.

Switches send and receive spanning-tree frames at regular intervals. These spanning-tree frames are used to construct a loop-free path, forcing redundant data paths into a standby (blocked) state. This is operationally transparent to the network hosts.

Types of STP on Cisco Switches

- Common Spanning Tree (CST)

When connecting a Cisco switch to a non-Cisco device through an 802.1Q trunk, the Cisco switch combines the spanning-tree instance of the 802.1Q native VLAN of the trunk with the spanning-tree instance of the non-Cisco 802.1Q switch. The primary advantages of CST are that only one set of BPDU's are used; it is only necessary to track changes for a single instance of STP, and non-Cisco switches can be added to the mesh. However, with only one STP algorithm running, sub-optimal paths are more likely to be selected than under other methods. With CST, less bandwidth will be used to negotiate a root bridge, although with only one Root Bridge for the entire network, it may take longer for STP to recalculate when a change occurs.

- Per-VLAN Spanning Tree (PVST)

A Cisco proprietary method of connecting through ISL VLAN trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunk. This is the default STP used on ISL trunks. Since each VLAN has its own instance of STP, there is more granular control of the path selection process, and fewer sub-optimal paths may be invoked. Since the size of the STP topology is reduced, convergence time is reduced increasing scalability and stability.

- Per VLAN Spanning-Tree Plus (PVST+)

A Cisco proprietary method of connecting through 802.1Q VLAN trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunk, versus non-Cisco 802.1Q switches which maintain one instance for ALL VLANs. This is the default STP used on ISL trunks. Since each VLAN has its own instance of STP, there is more granular control of the path selection process, and fewer sub-optimal paths may be

invoked. Since the size of the STP topology is reduced, convergence time is reduced increasing scalability and stability.

PVST+ has replaced PVST and is now the default spanning tree protocol used on Cisco switches. Besides the features offered by PVST, PVST+ also offers Layer 2 load balancing.

- Multiple Spanning Trees (MST) 802.1s

Multiple Spanning Tree (MST) is an IEEE standard inspired from the Cisco proprietary Multiple Instances Spanning Tree Protocol (MISTP) implementation. MSTs (IEEE 802.1s), by supporting multiple instances of spanning tree combine the best aspects from both the PVST+ and the 802.1q. Several VLANs can be mapped to a reduced number of spanning tree instances because most networks do not need more than a few logical topologies. MST uses the port states and timers from Rapid Spanning Tree.

- Rapid Spanning Tree Protocol (RSTP) 802.1w

The IEEE 802.1w standard, Rapid Spanning Tree Protocol (RSTP) is an enhancement to 802.1d. RSTP offers more rapid spanning tree convergence, and industry standard alternatives to Cisco's PortFast, UplinkFast and BackboneFast. With RSTP, you can reduce the convergence time of a port from the 50-second default of 802.1d to, as low as, 1 second. This fast convergence is critical for latency-sensitive applications like voice and video.

Redundancy Without Loops

STP runs on bridges and switches in order to prevent loops in the network. There are different flavors of STP, with IEEE 802.1d being the most common. It is found in situations where you want to allow redundant links, but not loops.

Redundant links are important backups in case of a link failure in a network. If a primary fails, the backup links are activated so that users can continue using the network with minimal service interruptions. Without STP running on the bridges and switches, such a situation could result in loops.

To provide this desired path redundancy, as well as to avoid a loop condition, STP defines a series of paths that span all switches in an extended network. It forces less desirable links into a standby (blocked) state, while leaving others in a forwarding state. If a link in forwarding state becomes unavailable, STP recognizes the topology change in the network, and recalculates the best paths.

STP allows the redundant links, but with all but the best blocked, so that the switched fabric looks logically (but not physically) like this diagram:

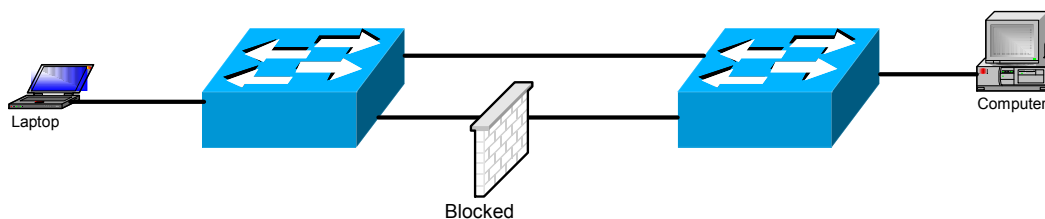


Figure 2-2. STP redundant links

Root Bridges and Switches

The key to STP is the selection of a root bridge, which becomes the focal point in the network. All other decisions in the network, such as which ports are blocked and which ports are put in forwarding mode, are made from the perspective of this root bridge.

When implemented in a switched network, the root bridge is usually referred to as the "root switch." Depending on the type of spanning-tree enabled, each VLAN may have its own root bridge/switch. In this case, the root for the different VLANs may all reside in a single switch, or it can reside in varying switches, depending on the estimates of the Network Architect.

You should remember that selection of the root switch for a particular VLAN is extremely important. You can allow the network to decide the root using arbitrary criteria, or you can define it yourself. With either option there is risk; the switches are unlikely to select the optimal root by themselves, but a bad network architect can make an even worse choice. You should control the selection of the root whenever possible, but make sure you understand what the consequences of a bad decision can be.

Bridge Protocol Data Units (BPDUs)

All switches exchange information to use in the selection of the root switch, as well as for subsequent configuration of the network. This information is carried in Bridge Protocol Data Units (BPDU). The BPDU contains parameters that the switches use in the selection process. Each switch compares the parameters in the BPDU that they are sending

to their neighbor with the one that they are receiving from their neighbor.

BPDUs are multicast frames sent out periodically by switches to announce their existence, resources, and recent changes to a switch's configuration. They:

- Propagate bridge IDs in order for the selection of the root switch to take place.
- Are used to determine loop locations within a network.
- Provide notification of network topology changes.
- Remove loops by placing redundant switch ports in a backup state.

Each configuration BPDU contains the following information:

- The unique bridge ID of the switch that the sending switch identifies as the root switch
- The spanning-tree path cost to the root
- The bridge ID of the sending switch
- Message age
- The identifier of the sending interface
- Values for the hello, forward delay, and max-age protocol timers

When a switch receives a configuration BPDU that contains *superior* information (lower bridge ID, lower path cost, and so forth), it stores the information for that port. If this BPDU is received on the root port of the switch, the switch

also forwards it with an updated message to all attached LANs for which it is the designated switch.

If a switch receives a configuration BPDU that contains *inferior* information to that currently stored for that port, it discards the BPDU. If the switch is a designated switch for the LAN from which the inferior BPDU was received, it sends that LAN a BPDU containing the up-to-date information stored for that port. In this way, inferior information is discarded, and superior information is propagated on the network.

A BPDU exchange results in these actions:

- One switch in the network is elected as the root switch (the logical center of the spanning-tree topology in a switched network).

For each VLAN, the switch with the highest switch priority (the lowest numerical priority value) is elected as the root switch. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root switch. The switch priority value occupies the most significant bits of the bridge ID.

- A root port is selected for each switch (except the root switch). This port provides the best path (lowest cost) when the switch forwards packets to the root switch.
- The shortest distance to the root switch is calculated for each switch based on the path cost.
- A designated switch for each LAN segment is selected. The designated switch incurs the lowest path cost when forwarding packets from that LAN to the root switch.

The port through which the designated switch is attached to the LAN is called the designated port.

All paths that are not needed to reach the root switch from anywhere in the switched network are placed in the spanning-tree blocking mode.

As a BPDU leaves a port, it applies the root port cost. Path Cost is the total sum of all of the port costs, and is what STP uses to determine which ports should forward and which ports should block. If the path cost is the same for several ports, STP will use the lowest port ID.

The thing to remember in the STP root selection process is that "smaller is better." If the Root ID advertised on Switch A is lower than the Root ID that its neighbor (Switch B) is advertising, then Switch A's information is better. Switch B stops advertising its Root ID, and accepts the Root ID of Switch A.

In a bridged network environment running under IEEE 802.1, a bridge takes maximum age, forwarding delay, and hello time parameters from the root bridge BPDU.

The BPDUs are in the following format:

2	1	1	1	8	4	8	2	2	2	2	2	2 Octets
Protocol ID	Version	Message Type	Flags	Root ID	Root Cost	Bridge ID	Port ID	Message Age	Max Age	Hello Time	Forward Delay	

- Protocol ID – the packet is a BPDU.
- Version – BPDU version used.
- Message Type – the stage of the negotiation.

- Flags – two bits used to indicate a change in topology and indicate acknowledgement of the TCN BPDU.
- Root ID – Root bridge priority (2 bytes) followed by the MAC address (6 bytes).
- Root Path Cost – Total cost to or from this bridge to the designated root bridge.
- Bridge ID – Bridge priority (2 bytes) followed by the MAC address (6 bytes), where smaller is better, and the lowest value wins! The default bridge priority is 0x8000 (3276810).
- Port ID – ID of the port from which BPDUs are sent, a root port, made up of the configured port priority and the bridge MAC address.
- Message Age – Timers for aging messages (only have effect on the network if the root bridge is configured with this parameter).
- Maximum Age – Maximum message age before information from a BPDU is dropped for being too old when no other BPDUs are being received. (Only has effect on the network if the root bridge is configured with this parameter). The default value is 20 seconds.
- Hello Time – Time between BPDU configuration messages sent by the root bridge (only has effect on the network if the root bridge is configured with this parameter). The default value is 2 seconds.
- Forward Delay – Stops a bridge from forwarding data temporarily to allow information about a topology change to disseminate to all parts of the network.

This allows ports which need to be turned off in the new topology to be switched off before the new ports are turned on (only has effect on the network if the root bridge is configured with this parameter).

Operation under STP

Selection of the root switch is the most important STP decision. Before deciding to configure STP, you should determine which switch will be the root of the spanning tree. It does not necessarily have to be the most powerful switch, but it should be the most central switch on the network. The network will be logically laid out from the perspective of this root device.

You should not change the root switch configuration if you can possibly avoid it, since reconfiguration triggers spanning tree recalculation, which will affect network performance.

Backbone switches, since they rarely have their configuration changed, are often defined as root switches. Backbone switches are usually more powerful than other switches, and are centrally placed within the network. They are also less likely to be disturbed during moves and changes within the network.

Selecting a backup root bridge is recommended as a matter of good practice. The backup will have a bridge priority in between the default and the primary root bridge. If the primary root switch were to fail, human intelligence would still be the determining factor in recalculating the spanning tree.

“Bridge priority” is the main variable that defines the root bridge. The switch with the lowest bridge priority will be selected as the root switch.

STP Step-by-Step

The root switch selection process starts with each switch transmitting BPDUs to its directly connected switch neighbors on a per-VLAN basis. As the BPDUs go through the network, each switch compares the BPDU it sent out to the ones it has received from its neighbors.

From this comparison of transmitted and received BPDU's, the switches determine the root switch. The switch with the lowest priority in the network wins this election process. (Remember, there may only be one root switch identified for each VLAN, depending on the type of STP selected.)

Any time STP is calculated or recalculated, the switches use these rules:

Rule One for STP: All ports of the root switch must be in forwarding mode. Next, each switch determines their best path to get to the root. The switches compare the information in all the BPDUs received on all their ports. The port with the smallest information contained in its BPDU is used to get to the root switch; that port is called the root port. After a switch figures out its root port, it proceeds to Rule Two.

Rule Two for STP: The selected root ports need to be set to forwarding mode. For each LAN segment, the switches communicate with each other to determine which switch on that LAN segment is best to use for moving data from that segment to the root bridge. This switch is called the designated switch.

Rule Three for STP: In any given LAN segment, the port of the designated switch that connects to the LAN segment must be placed in forwarding mode.

Rule Four for STP: All other ports in every switch within the VLAN must be placed in blocking mode. This is only for ports that are connected to other bridges or switches. Ports connected to workstations or PCs should not be affected by STP calculation; they remain forwarded. Users on your network may, however, see a delay during the recalculation process, as all data transmissions are halted.

Port State Progression in STP

STP domain ports will progress through the following states:

- **Blocking** – Listens for BPDUs from other bridges, but does not forward them or any traffic.
- **Listening** – An interim state while moving from blocking to learning. Listens for frames and detects available paths to the root bridge, but will not collect host MAC addresses for its address table.
- **Learning** – Examines the data frames for source MAC addresses to populate its address table, but no user data is passed.
- **Forwarding** – Once the learning state is complete, the port will begin its normal function of gathering MAC addresses and passing user data.

- **Disabled** – Either there has been an equipment failure, a security issue or the port has been disabled by the system administrator.

Notes about STP Port States:

- A port in blocking state does not participate in frame forwarding. The switch always goes into blocking state immediately following switch initialization.
- When a port changes from the listening state to the learning state, it is preparing to participate in frame forwarding.
- A port in the forwarding state actually forwards frames (User data, BPDUs, etc.).

STP Broadcast Domain Characteristics

- Where redundant links exist, any but the one with the least distance from the root switch are blocked.
- STP convergence can take upwards of 50 seconds.
- Broadcast traffic within the layer-2 domain (VLAN) interrupts every host.
- Broadcast storms within the layer-2 domain affect the whole domain.
- Isolating problems can be time consuming.
- Network security options within the layer-2 domain (VLAN) are limited.

802.1w Rapid Spanning Tree Port States

There are only three port states left in RSTP that correspond to the three possible operational states. The 802.1D disabled, blocking, and listening states are merged into a unique 802.1w discarding state.

- **Discarding** - (802.1d Blocking, Listening, Disabled)
- **Learning**
- **Forwarding**

802.1w Summary Table

STP (802.1D) Port State	RSTP (802.1w) Port State	Is Port Included in Active Topology?	Is Port Learning MAC Addresses?
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	Yes	Yes

802.1w Rapid Spanning Tree Port Roles

The role is now a variable assigned to a given port. The root port and designated port roles remain, while the blocking port role is split into the backup and alternate

port roles. The Spanning Tree Algorithm (STA) determines the role of a port based on Bridge Protocol Data Units (BPDUs). In order to simplify matters, the thing to remember about a BPDU is there is always a method to compare any two of them and decide whether one is more useful than the other. This is based on the value stored in the BPDU and occasionally on the port on which they are received.

- **Root Port** - The port that receives the best BPDU on a bridge is the root port. This is the port that is the closest to the root bridge in terms of path cost.
- **Designated Port** - A port is designated if it can send the best BPDU on the segment to which it is connected.
- **Alternate Port** - An alternate port receives more useful BPDUs from another bridge and is a port blocked.
- **Backup Port** - A backup port receives more useful BPDUs from the same bridge it is on and is a port blocked.
- **Disabled Port**

802.1w vs. 802.1d

802.1w provides faster spanning tree convergence after a topology change. 802.1w has additional features similar to Cisco PortFast, UplinkFast, and BackboneFast. With RSPT the port states transition within 1 sec instead of 50 seconds in 802.1.d.

Topology Changes – TCN, TCA, and TC

Normally, BPDU traffic is received from the root bridge on the root port, but is never sent to the root bridge. If a bridge has a topology change, it sends a TCN (Topology Change Notification) BPDU out its root port towards the root bridge. The next upstream bridge acknowledges receipt of the TCN by replying with a BPDU which has the TCA (Topology Change Acknowledgement) bit set. The TCN/TCA process is repeated hop-by-hop, until the Root Bridge receives the TCN BPDU. When the root is aware of a change, it sends out BPDUs with the TC (Topology Change) bit set.

Spanning-Tree Configuration

- Spanning-Tree Mode - The switch supports three spanning-tree modes: PVST+, rapid PVST+, or MSTP. By default, the switch runs the PVST+ protocol.

```
spanning-tree mode {pvst | mst | rapid-pvst}
```

Recommended for rapid-PVST+ mode only

```
interface interface-id  
spanning-tree link-type point-to-point
```

- Disabling Spanning-Tree - Spanning tree is enabled by default on VLAN 1 and on all newly created supported VLANs. Disable spanning tree only if you are sure there are no loops in the network topology.

```
no spanning-tree vlan vlan-id
```

- Root Switch - The default switch priority value is 32768. When entering the below command, the switch sets its own priority for the specified VLAN to 24576. If there is a switch with a root value less than 24576, the switch will set its own priority for the specified VLAN to 4096 less than the lowest switch priority.

```
spanning-tree vlan vlan-id root primary
```

- Secondary Root Switch - When you configure a switch as the secondary root, the switch priority is modified from the default value (32768) to 28672.

```
spanning-tree vlan vlan-id root secondary
```

- Port Priority - If a loop occurs, spanning tree uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want selected first and lower priority values (higher numerical values) that you want selected last. If all interfaces have the same priority value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

```
interface interface-id  
spanning-tree port-priority priority
```

- Path Cost - The spanning-tree path cost default value is derived from the media speed of an interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost values that you want selected last. If all interfaces have the same cost

value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

```
interface interface-id  
spanning-tree cost cost
```

- **Switch Priority** - You can configure the switch priority and make it more likely that the switch will be chosen as the root switch.

```
spanning-tree vlan vlan-id priority priority
```

STP Timers

- **Hello timer** - How often the switch broadcasts Hello messages to other switches.

The default hello timer is 2 seconds. The hello timer range is 1 to 10 seconds. The command to configure the hello timer is:

```
spanning-tree vlan vlan-id hello-time seconds
```

- **Forward delay timer** - Amount of time a port will remain in the listening and learning states before going into the forwarding state.

The default forward delay timer is 15 seconds. The forward delay timer range is 4 to 30 seconds. The command to configure the forward delay timer is:

```
spanning-tree vlan vlan-id forward-time seconds
```

- **Maximum age timer** - How long protocol information received on a port is stored by the switch.

The default maximum age timer is 20 seconds. The maximum after timer range is 6 to 40. The command to configure the maximum age timer is:

```
spanning-tree vlan vlan-id max-age seconds
```

- **Transmit Hold Count** – Controls the number of BPDUs that can be sent before pausing for 1 second.

The default transmit hold count is 6. The transmit hold count range is 1 to 20. The command to configure the transmit hold count is:

```
spanning-tree transmit hold-count value
```

Spanning-Tree Features

- **PortFast** – PortFast immediately brings an interface configured as an access or trunk port to the forwarding state from a blocking state, bypassing the listening and learning states. You can use PortFast on interfaces connected to a single workstation or server, to allow these devices to immediately connect to the network, rather than waiting for the spanning tree to converge.

Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). An interface with PortFast enabled goes through the normal cycle of spanning-tree status changes when the switch is restarted.

You can enable this feature by using the **spanning-tree portfast** interface configuration or the **spanning-tree portfast default** global configuration command.

- **BPDU GUARD** - The BPDU guard feature can be globally enabled on the switch or can be enabled per port, but the feature operates with some differences.

At the global level, you enable BPDU guard on PortFast enabled ports by using the **spanning-tree portfast bpduguard default** global configuration command. Spanning tree shuts down ports that are in a Port Fast-operational state if any BPDU is received on them. In a valid configuration, Port Fast-enabled ports do not receive BPDUs. Receiving a BPDU on a Port Fast-enabled port means an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the error-disabled state. When this happens, the switch shuts down the entire port on which the violation occurred.

To prevent the port from shutting down, you can use the **errdisable detect cause bpduguard shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

At the interface level, you enable BPDU guard on any port by using the **spanning-tree bpduguard enable** interface configuration command without also enabling the Port Fast feature. When the port receives a BPDU, it is put in the error-disabled state.

The BPDU guard feature provides a secure response to invalid configurations because you must manually put the interface back in service. Use the BPDU guard feature in a service-provider network to prevent an access port from participating in the spanning tree.

- **BPDU Filtering** - The BPDU filtering feature can be globally enabled on the switch or can be enabled per interface, but the feature operates with some differences.

At the global level, you can enable BPDU filtering on Port Fast-enabled interfaces by using the **spanning-tree portfast bpdupfilter default** global configuration command. This command prevents interfaces that are in a Port Fast-operational state from sending or receiving BPDUs. The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs. You should globally enable BPDU filtering on a switch so that hosts connected to these interfaces do not receive BPDUs. If a BPDU is received on a Port Fast-enabled interface, the interface loses its Port Fast-operational status, and BPDU filtering is disabled.

At the interface level, you can enable BPDU filtering on any interface by using the **spanning-tree bpdupfilter enable** interface configuration command without also enabling the Port Fast feature. This command prevents the interface from sending or receiving BPDUs.

- **UplinkFast** - Switches in hierarchical networks can be grouped into backbone switches, distribution switches, and access switches. If a switch loses connectivity, it begins using the alternate paths as soon as the spanning tree selects a new root port. By enabling UplinkFast with the **spanning-tree uplinkfast** global configuration command, you can accelerate the choice of a new root port when a link or switch fails or when the spanning tree reconfigures itself. The root port transitions to the forwarding state immediately without going through the listening and learning

states, as it would with the normal spanning-tree procedures.

When the spanning tree reconfigures the new root port, other interfaces flood the network with multicast packets, one for each address that was learned on the interface. You can limit these bursts of multicast traffic by reducing the max-update-rate parameter (the default for this parameter is 150 packets per second). However, if you enter zero, station-learning frames are not generated, so the spanning-tree topology converges more slowly after a loss of connectivity.

UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

- **BackboneFast** - BackboneFast detects indirect failures in the core of the backbone. BackboneFast is a complementary technology to the UplinkFast feature, which responds to failures on links directly connected to access switches. BackboneFast optimizes the maximum-age timer, which controls the amount of time the switch stores protocol information received on an interface. When a switch receives an inferior BPDU from the designated port of another switch, the BPDU is a signal that the other switch might have lost its path to the root, and BackboneFast tries to find an alternate path to the root.

BackboneFast, which is enabled by using the **spanning-tree backbonefast** global configuration command, starts when a root port or blocked interface on a switch receives inferior BPDUs from its designated switch. An inferior BPDU identifies a switch that declares itself as both the root bridge and the designated switch. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an *indirect* link) has failed (that is, the designated switch has lost its connection to the root switch). Under spanning-tree rules, the switch ignores inferior BPDUs for the configured maximum aging time specified by the **spanning-tree vlan vlan-id max-age** global configuration command.

The switch tries to find if it has an alternate path to the root switch. If the inferior BPDU arrives on a blocked interface, the root port and other blocked interfaces on the switch become alternate paths to the root switch. (Self-looped ports are not considered alternate paths to the root switch.) If the inferior BPDU arrives on the root port, all blocked interfaces become alternate paths to the root switch. If the inferior BPDU arrives on the root port and there are no blocked interfaces, the switch assumes that it has lost connectivity to the root switch, causes the maximum aging time on the root port to expire, and becomes the root switch according to normal spanning-tree rules.

If the switch has alternate paths to the root switch, it uses these alternate paths to send a root link query (RLQ) request. The switch sends the RLQ request on all alternate paths and waits for an RLQ reply from other switches in the network.

If the switch discovers that it still has an alternate path to the root, it expires the maximum aging time on the interface that received the inferior BPDU. If all the alternate paths to the root switch indicate that the switch has lost connectivity to the root switch, the switch expires the maximum aging time on the interface that received the RLQ reply. If one or more alternate paths can still connect to the root switch, the switch makes all interfaces on which it received an inferior BPDU its designated ports and moves them from the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

- **EtherChannel Guard** - You can use EtherChannel guard to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the switch interfaces are configured in an EtherChannel, but the interfaces on the other device are not. A misconfiguration can also occur if the channel parameters are not the same at both ends of the EtherChannel.

If the switch detects a misconfiguration on the other device, EtherChannel guard places the switch interfaces in the error-disabled state, and displays an error message.

You can enable this feature by using the **spanning-tree etherchannel guard misconfig** global configuration command.

- **Root Guard** - The Layer 2 network of a service provider (SP) can include many connections to switches that are not owned by the SP. In such a topology, the spanning tree can reconfigure itself and select a *customer switch* as the root switch. You can avoid this

situation by enabling root guard on SP switch interfaces that connect to switches in your customer's network. If spanning-tree calculations cause an interface in the customer network to be selected as the root port, root guard then places the interface in the root-inconsistent (blocked) state to prevent the customer's switch from becoming the root switch or being in the path to the root.

If a switch outside the SP network becomes the root switch, the interface is blocked (root-inconsistent state), and spanning tree selects a new root switch. The customer's switch does not become the root switch and is not in the path to the root.

If the switch is operating in multiple spanning-tree (MST) mode, root guard forces the interface to be a designated port. If a boundary port is blocked in an internal spanning-tree (IST) instance because of root guard, the interface also is blocked in all MST instances. A boundary port is an interface that connects to a LAN, the designated switch of which is either an IEEE 802.1D switch or a switch with a different MST region configuration.

Root guard enabled on an interface applies to all the VLANs to which the interface belongs. VLANs can be grouped and mapped to an MST instance.

You can enable this feature by using the **spanning-tree guard root** interface configuration command.

- **Loop Guard** - You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is enabled on the entire switched network. Loop guard

prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

You can enable this feature by using the **spanning-tree loopguard default** global configuration command.

When the switch is operating in PVST+ or rapid-PVST+ mode, loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in MST mode, BPDUs are not sent on nonboundary ports only if the interface is blocked by loop guard in all MST instances. On a boundary port, loop guard blocks the interface in all MST instances.

Storm Control

Storm control prevents traffic on a LAN from being disrupted by a broadcast, multicast, or unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation, mistakes in network configurations, or users issuing a denial-of-service attack can cause a storm.

Storm control (or traffic suppression) monitors packets passing from an interface to the switching bus and determines if the packet is unicast, multicast, or broadcast. The switch counts the number of packets of a specified type received within the 1-second time interval and compares the measurement with a predefined suppression-level threshold.

Storm control uses one of these methods to measure traffic activity:

- Bandwidth as a percentage of the total available bandwidth of the port that can be used by the broadcast, multicast, or unicast traffic.
- Traffic rate in packets per second at which broadcast, multicast, or unicast packets are received.
- Traffic rate in bits per second at which broadcast, multicast, or unicast packets are received.
- Traffic rate in packets per second and for small frames. This feature is enabled globally. The threshold for small frames is configured for each interface.

With each method, the port blocks traffic when the rising threshold is reached. The port remains blocked until the traffic rate drops below the falling threshold (if one is specified) and then resumes normal forwarding. If the falling suppression level is not specified, the switch blocks all traffic until the traffic rate drops below the rising suppression level. In general, the higher the level, the less effective the protection against broadcast storms.

By default, unicast, broadcast, and multicast storm control are disabled on the switch interfaces; that is, the suppression level is 100 percent.

Storm control is supported on physical interfaces. You can also configure storm control on an EtherChannel. When storm control is configured on an EtherChannel, the storm control settings propagate to the EtherChannel physical interfaces.

Storm control is configured on a per interface basis.

The command is:

```
storm-control {broadcast|multicast|unicast} level {level [level-low] | bps bps [bps-low] | pps pps [pps-low]}
```

Example:

```
switch(config-if)#storm-control broadcast level 40 20  
switch(config-if)#storm-control unicast bps 1000 500  
switch(config-if)#storm-control multicast pps 500 250
```

- Level 100 permits everything
- Level 0.00 disables the frame type

Unicast Flooding

By default, the switch floods packets with unknown destination MAC addresses out of all ports. If unknown unicast and multicast traffic is forwarded to a protected port, there could be security issues. To prevent unknown unicast or multicast traffic from being forwarded from one port to another, you can block a port (protected or nonprotected) from flooding unknown unicast or multicast packets to other ports.

The interface can be a physical interface or an EtherChannel group. When you block multicast or unicast traffic for a port channel, it is blocked on all ports in the port-channel group.

The command is:

```
switchport block {multicast | unicast}
```

Example:

```
switch(config-if)# switchport block ?  
  multicast  Block unknown multicast addresses  
  unicast    Block unknown unicast addresses
```

With multicast traffic, the port blocking feature blocks only pure Layer 2 packets. Multicast packets that contain IPv4 or IPv6 information in the header are not blocked.

Implement VLAN and VLAN Trunking Protocol (VTP)

Virtual LAN (VLAN)

A VLAN is an extended logical network configured independently from the physical network layout. Each port on a switch can be defined to join a specific VLAN.

VLAN ports on a switch can be assigned statically using a VLAN management application or by working directly within the switch. Dynamic VLANs are a more convenient approach in which ports on a switch that can automatically determine their VLAN assignments. Each hub segment connected to a switch port can be assigned to only one VLAN.

Unlike a physical subnet, VLAN devices do not need to be connected to a single physical cable segment. Devices can be part of a subnet and still be connected to different switches in different locations. However, since each VLAN is a separate broadcast domain, routing between them must be enabled if data is to be passed.

Most VLANs use frame filtering (frame tagging) with user-defined offsets to examine particular information about

each frame, and uniquely assign a user-defined ID to each frame header.

There are two steps in properly configuring a VLAN.

- Create the VLAN

```
vlan vlan-id
```

- Assign switchports to that VLAN

```
switchport access vlan vlan-id
```

VLAN Trunk Protocol (VTP)

The name VLAN Trunk Protocol (VTP) is misleading, since it doesn't really have much to do with actual trunking, except that switches use the VTP information to know what VLANs to carry over a trunk line. VTP is really a way to update the VLANs that a switch recognizes as valid on multiple switches by updating a single "server" switch. The rest of the switches in the VTP domain get a copy of the list of valid VLANs from the server switch.

In a switched environment a subnet corresponds to a VLAN, and a VLAN may map to a single Layer 2 switch, or it may span several switches, especially at the access layer. Also, it is likely that one or more VLANs may be present on any particular switch. VLAN Trunk Protocol (VTP) is a layer-2 messaging protocol that centralizes the management of VLAN additions, deletions, and changes on a network-wide basis. This simplifies the management of large switched networks with many VLANs.

When you add a new VLAN to the network you only have to define it on a single switch. The rest of the switches are

defined automatically through their VTP membership. If you decide not to use VTP, then each VLAN will need to be configured manually on each individual switch, which is a lot of work and can easily lead to problems if you mistype a single character.

If you have more than one group of switches, and each group has a different set of VLANs that it has to recognize, you should assign a separate domain for each group of switches. VTP domains consist of one or more interconnected switches that share the same VLAN configuration. A switch can only be configured as a member of a single VTP domain. You can specify the global VLAN configuration for the domain using either the CLI or an SNMP session.

Early switching design specifications promoted the ability of VTP to create global VLAN groups that could span vast networks. In reality all this does is generate unnecessary and expensive wide area traffic for little gain. Most network designers want to limit VTP advertisements sent over slow and expensive wide area links, so the recommendation is not to set up VTP domains which span WAN links.

Configuration Command:

```
vtp domain domain-name
```

VTP Modes

Switches which are part of VTP domains can be configured to operate in one of four VTP modes:

- **Server** – Advertises VLAN configuration to other switches in the same VTP domain and synchronize with other switches in the domain. Can create, modify, and

delete VLANs as well as modify VLAN configuration parameters such as VTP version and VTP pruning for the whole domain. This is the default mode for a switch.

- **Client** – Advertises VLAN configuration to other switches in the same VTP domain and synchronize their VLAN configuration with other switches using advertisements received over trunk links. Unable to create, change, or delete VLAN configurations.
- **Transparent** – Does not advertise its VLAN configuration and does not synchronize its VLAN configuration with other switches. A switch running VTP version 2 will forward VTP advertisements, but will not act on them.
- **Off** – A switch in VTP off mode functions in the same manner as a VTP transparent switch, except that it does not forward VTP advertisements on trunks.

Configuration Command:

```
vtp mode {client | server | transparent | off}
```

VTP Advertisements

Each switch in the VTP domain sends periodic global configuration advertisements from each trunk port to a reserved multicast address. Neighboring switches receive these advertisements and update their VTP and VLAN configurations as necessary.

Advertisement types include: requests from clients, summary advertisements and subset advertisements. Advertisements carry two types of information:

- **Global Information** – Includes VTP Domain Name, VTP Configuration Revision Number, Update Identity, Update Timestamp, MD5 Digest, Frame Format
- **VLAN Information** – Includes VLAN ID, VLAN Name, VLAN Type, VLAN State, Additional VLAN configuration information specific to the VLAN type

In VTP version 3, VTP advertisements also include the primary server ID, an instance number, and a start index.

VTP message types:

- Summary advertisements
- Subset advertisement
- Advertisement requests
- VTP join messages

VTP advertisements carry configuration revision numbers that are incremented every time a change is made. This allows identification of the most recent updates to the network topology. When a switch finds an advertisement with a higher configuration revision number, it will save the new VTP database, writing over the old one. A VLAN that does not exist in the new database is automatically deleted from the switch. Any ports that were in the VLAN will be orphaned.

A common mistake is to add a switch that has been used on a separate or test network to a production network, without being aware of the revision number. Since test networks change much more frequently than production networks, the new switch is likely to have a higher configuration revision number than the production VTP domain. The result

is that the entire production domain's VTP database gets overwritten and any ports assigned to the lost VLANs lose their VLAN membership and become unavailable to users.

If you ever receive a call that all the switched ports on a network have suddenly locked up and no traffic is being passed, one of the first places to look is the new switch added to the network. Good documentation and control over physical access to network devices are probably your best defense against this. Also, to prevent this from happening, the command **clear config all** (on a set-based switch) or **write erase** (on an IOS-based switch) should always be used before any new switch is added to a production network.

Note that all switches in the VTP domain must run the same VTP version. By default, VTP operates in version 1.

VTP Version 2 supports these features that are not supported in version 1:

- Token Ring support
- Unrecognized Type-Length-Value (TLV) support
- Version-Depended Transparent Mode
- Consistency Checks

VTP Version 3 supports these features that are not supported in version 1 or version 2:

- Enhanced authentication
- Support for extended range VLAN (1006 - 4094) database propagation
- Private VLAN support
- Support for any database in a domain
- VTP Primary and VTP Secondary servers
- The option to turn VTP on or off on a per-trunk (per-port) basis

Configuration Command:

```
ntp version {1 | 2 | 3}
```

VTP Pruning

VTP pruning increases bandwidth. It does this by controlling traffic flow to the vital trunk links, and blocking flooded traffic to VLANs in the pruning eligible list. VTP pruning can only be enabled on a VTP server, and that will enable it for the entire management domain. VLAN 1 is always pruning-ineligible, and VLANs 2 through 1000 are pruning-eligible. VTP pruning is disabled by default.

Configuration Command:

```
ntp pruning
```

Implement Trunk and Trunk Protocols, EtherChannel, and Load-Balance

Trunking Overview

A trunk is a point-to-point link between one or more Ethernet switch interfaces and another networking device such as a router or a switch. Ethernet trunks carry the traffic of multiple VLANs over a single link, and you can extend the VLANs across an entire network.

Ethernet trunk interfaces support different trunking modes. You can set an interface as trunking or nontrunking or to negotiate trunking with the neighboring interface. To autonegotiate trunking, the interfaces must be in the same VTP domain.

You can configure a trunk on a single Ethernet interface or on an EtherChannel bundle.

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP), which is a Point-to-Point Protocol. However, some internetworking devices might forward DTP frames improperly, which could cause misconfigurations.

To avoid this, you should configure interfaces connected to devices that do not support DTP to not forward DTP frames, that is, to turn off DTP.

- If you do not intend to trunk across those links, use the **switchport mode access** interface configuration command to disable trunking.
- To enable trunking to a device that does not support DTP, use the **switchport mode trunk** and **switchport nonegotiate** interface configuration commands to cause the interface to become a trunk but to not generate DTP frames. Use the **switchport trunk encapsulation isl** or **switchport trunk encapsulation dot1q** interface to select the encapsulation type on the trunk port.

You can also specify on DTP interfaces whether the trunk uses ISL or IEEE 802.1Q encapsulation or if the encapsulation type is autonegotiated. The DTP supports autonegotiation of both ISL and IEEE 802.1Q trunks.

Trunking Modes

- **Auto** - Makes the interface able to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to *trunk* or

desirable mode. The default switchport mode for all Ethernet interfaces is **dynamic auto**.

- **Desirable** - Makes the interface actively attempt to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to *trunk*, *desirable*, or *auto* mode.
- **On** - Puts the interface into permanent trunking mode and negotiates to convert the neighboring link into a trunk link. The interface becomes a trunk interface even if the neighboring interface is not a trunk interface.
- **Nonegotiate** - Prevents the interface from generating DTP frames. You can use this command only when the interface switchport mode is **access** or **trunk**. You must manually configure the neighboring interface as a trunk interface to establish a trunk link.
- **Off** - Forces the port to become non-trunking, even if the neighboring port does not agree with the change.

Trunking Encapsulation Types

- **Inter-Switch Link (ISL)** - A Cisco proprietary trunking encapsulation that adds a 26-byte header and 4-byte trailer to the frame. ISL supports the processing of untagged frames. ISL supports up to 1024 VLANs.
- **IEEE 802.1Q (dot1q)** - An industry standard trunking encapsulation that does not change the size of the frame. Since multiple vendors support dot1q, it is becoming more common in newer switched networks. 802.1q uses a tag protocol ID of 0x8100 (Ethertype

8100). 802.1q allows the encapsulation of multiple trunks within a single trunk. 802.1q tag is only 4 bytes in length. 802.1q supports up to 4096 VLANs.

- **Negotiate** - Specifies that the interface negotiate with the neighboring interface to become an ISL (preferred) or IEEE 802.1Q trunk, depending on the configuration and capabilities of the neighboring interface. This is the default for the switch.

The trunking mode, the trunk encapsulation type, and the hardware capabilities of the two connected interfaces decide whether a link becomes an ISL or IEEE 802.1Q trunk.

Trunking Configuration

- Encapsulation

```
switchport trunk encapsulation {isl | dot1q | negotiate}
```

- Mode

```
switchport mode {dynamic {auto | desirable} | trunk}
```

EtherChannel

EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. You can use it to increase the bandwidth between the wiring closets and the data center, and you can deploy it anywhere in the network where bottlenecks are likely to occur. EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, EtherChannel redirects traffic from the failed

link to the remaining links in the channel without intervention.

An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link.

The EtherChannel provides full-duplex bandwidth up to 800 Mb/s (Fast EtherChannel) or 8 Gb/s (Gigabit EtherChannel) between your switch and another switch or host. Each EtherChannel can consist of up to eight compatibly configured Ethernet ports.

All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports. The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command.

You can configure an EtherChannel in one of these modes: Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), or On. Configure both ends of the EtherChannel in the same mode:

- When you configure one end of an EtherChannel in either PAgP or LACP mode, the system negotiates with the other end of the channel to determine which ports should become active. Incompatible ports are put into an independent state and continue to carry data traffic as would any other single link. The port configuration does not change, but the port does not participate in the EtherChannel.
- When you configure an EtherChannel in the **on** mode, no negotiations take place. The switch forces all compatible ports to become active in the EtherChannel. The other end of the channel (on the other switch)

must also be configured in the **on** mode; otherwise, packet loss can occur.

If a link within an EtherChannel fails, traffic previously carried over that failed link moves to the remaining links within the EtherChannel. If traps are enabled on the switch, a trap is sent for a failure that identifies the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.

Port-Channel Interfaces

When you create an EtherChannel, a port-channel logical interface is involved:

- With Layer 2 ports, use the **channel-group** interface configuration command to dynamically create the port-channel logical interface.

You also can use the **interface port-channel** *port-channel-number* global configuration command to manually create the port-channel logical interface, but then you must use the **channel-group** *channel-group-number* command to bind the logical interface to a physical port. The *channel-group-number* can be the same as the *port-channel-number*, or you can use a new number. If you use a new number, the **channel-group** command dynamically creates a new port channel.

- With Layer 3 ports, you should manually create the logical interface by using the **interface port-channel** global configuration command followed by the **no switchport** interface configuration command. Then you

manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command.

For both Layer 2 and Layer 3 ports, the **channel-group** command binds the physical port and the logical interface together.

Each EtherChannel has a port-channel logical interface number assigned to it. This port-channel interface number corresponds to the one specified with the **channel-group** interface configuration command.

After you configure an EtherChannel, configuration changes applied to the port-channel interface apply to all the physical ports assigned to the port-channel interface. Configuration changes applied to the physical port affect only the port where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply configuration commands to the port-channel interface, for example, spanning-tree commands or commands to configure a Layer 2 EtherChannel as a trunk.

Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco switches and on those switches licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports.

By using PAgP, the switch learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware,

administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single switch port.

PAgP Modes :

- **Auto** - Places a port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation. This setting minimizes the transmission of PAgP packets.
- **Desirable** - Places a port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets.

Link Aggregation Control Protocol

The LACP is defined in IEEE 802.3ad and enables Cisco switches to manage Ethernet channels between switches that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports.

By using LACP, the switch learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP

adds the group to the spanning tree as a single switch port.

LACP Modes:

- **Active** - Places a port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.
- **Passive** - Places a port into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation. This setting minimizes the transmission of LACP packets.

Layer 2 EtherChannel Configuration Examples:

This example shows how to configure an EtherChannel on a switch. It assigns two ports as static-access ports in VLAN 10 to channel 5 with the PAgP mode **desirable**:

```
Switch# configure terminal
Switch(config)# interface range gigabitethernet0/1 -2
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
Switch(config-if-range)# channel-group 5 mode desirable non-
silent
Switch(config-if-range)# end
```

This example shows how to configure an EtherChannel on a switch. It assigns two ports as static-access ports in VLAN 10 to channel 5 with the LACP mode **active**:

```
Switch# configure terminal
Switch(config)# interface range gigabitethernet0/1 -2
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
Switch(config-if-range)# channel-group 5 mode active
```

```
Switch(config-if-range)# end
```

Layer 3 EtherChannel Configuration Example:

To configure Layer 3 EtherChannels, you create the port-channel logical interface and then put the Ethernet ports into the port-channel as described in the next two sections.

This example shows how to create the logical port channel 5 and assign 172.10.20.10 as its IP address:

```
Switch# configure terminal
Switch(config)# interface port-channel 5
Switch(config-if)# no switchport
Switch(config-if)# ip address 172.10.20.10 255.255.255.0
Switch(config-if)# end
```

This example shows how to configure an EtherChannel. It assigns two ports to channel 5 with the LACP mode **active**:

```
Switch# configure terminal
Switch(config)# interface range gigabitethernet0/1 -2
Switch(config-if-range)# no ip address
Switch(config-if-range)# no switchport
Switch(config-if-range)# channel-group 5 mode active
Switch(config-if-range)# end
```

Load Balancing and Forwarding Methods

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination

addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

```
port-channel load-balance {dst-ip | dst-mac | src-dst-ip | src-  
dst-mac | src-ip | src-mac}
```

The default is **src-mac**.

Select one of these load-distribution methods:

- **dst-ip** – Load distribution is based on the destination-host IP address.
- **dst-mac** – Load distribution is based on the destination-host MAC address of the incoming packet.
- **src-dst-ip** – Load distribution is based on the source-and-destination host-IP address.
- **src-dst-mac** – Load distribution is based on the source-and-destination host-MAC address.
- **src-ip** – Load distribution is based on the source-host IP address.
- **src-mac** – Load distribution is based on the source-MAC address of the incoming packet.

Implement Ethernet Technologies

Speed and Duplex

Ethernet interfaces operate at 10, 100, or 1000 Mb/s, or 10,000 Mb/s and in either full- or half-duplex mode. In full-duplex mode, two stations can send and receive traffic at the same time. Normally, 10-Mb/s ports operate in half-duplex mode, which means that stations can either receive or send traffic.

Speed/Duplex Negotiation detects the speed (for example, 10Mbps, 100Mbps) and duplex (half-duplex or full-duplex) settings of the device on the other end of the wire and subsequently adjust to match those settings. During speed/duplex negotiation the device transmits its own abilities to the peer device so that the peer can use the appropriate settings.

Fast Ethernet (10/100-Mb/s) ports support all speed and duplex options. Gigabit Ethernet (10/100/1000-Mb/s) ports support all speed options and all duplex options (auto, half, and full). However, Gigabit Ethernet ports operating at 1000 Mb/s do not support half-duplex mode.

The interface commands **speed** and **duplex** are used to configure the speed and duplex settings.

Ethernet, Fast Ethernet, and Gigabit Ethernet

Ethernet

The term Ethernet refers to the family of local-area network (LAN) products covered by the IEEE 802.3 standard that defines what is commonly known as the CSMA/CD protocol. Four data rates are currently defined for operation over optical fiber and twisted-pair cables:

- 10 Mbps - 10Base-T Ethernet
- 100 Mbps - Fast Ethernet
- 1000 Mbps and 10,000Mbps - Gigabit Ethernet

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is the LAN access method used in Ethernet. When a device wants to gain access to the network, it checks to

see if the network is free. If the network is not free, the device waits a random amount of time before retrying. If the network is free and two devices access the line at exactly the same time, their signals collide. When a collision is detected, they both back off and wait a random amount of time before retrying.

Fast Ethernet

Fast Ethernet offers a speed increase ten times that of the 10BaseT Ethernet specification, while preserving such qualities as frame format, MAC mechanisms, and MTU. Such similarities allow the use of existing 10BaseT applications and network management tools on Fast Ethernet networks.

Gigabit Ethernet

Gigabit Ethernet is a standard described under IEEE 802.3z, defined in 1998. The IEEE 802.3ab standard described the 1000Base-T standard in 1999. Both describe Gigabit speed implementations, the difference being that the 802.3z uses fiber and 802.3ab uses copper (Category 5e/6).

PPP over Ethernet (PPPoE)

PPPoE combines two widely accepted standards, Ethernet and PPP, in order to provide an authenticated method that assigns IP Addresses to client systems. PPPoE clients are typically personal computers connected to an ISP over a remote broadband connection, such as DSL or cable service. ISPs deploy PPPoE because it is easier for customers to use and it uses their existing remote access infrastructure in order to support high-speed broadband access.