Part IX

Appendixes

This page intentionally left blank

APPENDIX A

Numeric Reference Tables

This appendix provides several useful reference tables that list numbers used throughout this book. Specifically:

Table A-1: A decimal-binary cross reference, useful when converting from decimal to binary and vice versa.

Table A-1 Decimal-Binary Cross Reference, Decimal Values 0–255

Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value
0	00000000	32	00100000	64	01000000	96	01100000
1	00000001	33	00100001	65	01000001	97	01100001
2	00000010	34	00100010	66	01000010	98	01100010
3	00000011	35	00100011	67	01000011	99	01100011
4	00000100	36	00100100	68	01000100	100	01100100
5	00000101	37	00100101	69	01000101	101	01100101
6	00000110	38	00100110	70	01000110	102	01100110
7	00000111	39	00100111	71	01000111	103	01100111
8	00001000	40	00101000	72	01001000	104	01101000
9	00001001	41	00101001	73	01001001	105	01101001
10	00001010	42	00101010	74	01001010	106	01101010
11	00001011	43	00101011	75	01001011	107	01101011
12	00001100	44	00101100	76	01001100	108	01101100
13	00001101	45	00101101	77	01001101	109	01101101
14	00001110	46	00101110	78	01001110	110	01101110
15	00001111	47	00101111	79	01001111	111	01101111
16	00010000	48	00110000	80	01010000	112	01110000
17	00010001	49	00110001	81	01010001	113	01110001
18	00010010	50	00110010	82	01010010	114	01110010
19	00010011	51	00110011	83	01010011	115	01110011
20	00010100	52	00110100	84	01010100	116	01110100
21	00010101	53	00110101	85	01010101	117	01110101
22	00010110	54	00110110	86	01010110	118	01110110
23	00010111	55	00110111	87	01010111	119	01110111
24	00011000	56	00111000	88	01011000	120	01111000
25	00011001	57	00111001	89	01011001	121	01111001
26	00011010	58	00111010	90	01011010	122	01111010
27	00011011	59	00111011	91	01011011	123	01111011
28	00011100	60	00111100	92	01011100	124	01111100
29	00011101	61	00111101	93	01011101	125	01111101
30	00011110	62	00111110	94	01011110	126	01111110
31	00011111	63	00111111	95	01011111	127	01111111

Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value
128	10000000	160	10100000	192	11000000	224	11100000
129	10000001	161	10100001	193	11000001	225	11100001
130	10000010	162	10100010	194	11000010	226	11100010
131	10000011	163	10100011	195	11000011	227	11100011
132	10000100	164	10100100	196	11000100	228	11100100
133	10000101	165	10100101	197	11000101	229	11100101
134	10000110	166	10100110	198	11000110	230	11100110
135	10000111	167	10100111	199	11000111	231	11100111
136	10001000	168	10101000	200	11001000	232	11101000
137	10001001	169	10101001	201	11001001	233	11101001
138	10001010	170	10101010	202	11001010	234	11101010
139	10001011	171	10101011	203	11001011	235	11101011
140	10001100	172	10101100	204	11001100	236	11101100
141	10001101	173	10101101	205	11001101	237	11101101
142	10001110	174	10101110	206	11001110	238	11101110
143	10001111	175	10101111	207	11001111	239	11101111
144	10010000	176	10110000	208	11010000	240	11110000
145	10010001	177	10110001	209	11010001	241	11110001
146	10010010	178	10110010	210	11010010	242	11110010
147	10010011	179	10110011	211	11010011	243	11110011
148	10010100	180	10110100	212	11010100	244	11110100
149	10010101	181	10110101	213	11010101	245	11110101
150	10010110	182	10110110	214	11010110	246	11110110
151	10010111	183	10110111	215	11010111	247	11110111
152	10011000	184	10111000	216	11011000	248	11111000
153	10011001	185	10111001	217	11011001	249	11111001
154	10011010	186	10111010	218	11011010	250	11111010
155	10011011	187	10111011	219	11011011	251	11111011
156	10011100	188	10111100	220	11011100	252	11111100
157	10011101	189	10111101	221	11011101	253	11111101
158	10011110	190	10111110	222	11011110	254	11111110
159	10011111	191	10111111	223	11011111	255	11111111

Table A-2: A hexadecimal-binary cross reference, useful when converting from hex to binary and vice versa.

Table A-2 Hex-Binary Cross Reference

Hex	4-Bit Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Table A-3: Powers of 2, from 2^1 through 2^{32}.

Table A-3 Powers of 2

X	2^X	X	2^X
1	2	17	131,072
2	4	18	262,144
3	8	19	524,288
4	16	20	1,048,576
5	32	21	2,097,152
6	64	22	4,194,304
7	128	23	8,388,608
8	256	24	16,777,216
9	512	25	33,554,432
10	1024	26	67,108,864
11	2048	27	134,217,728
12	4096	28	268,435,456
13	8192	29	536,870,912
14	16,384	30	1,073,741,824
15	32,768	31	2,147,483,648
16	65,536	32	4,294,967,296

Table A-4: Table of all 33 possible subnet masks, in all three formats.

Table A-4 All Subnet Masks

Decimal	Prefix	Binary
0.0.0.0	/0	00000000 00000000 00000000 00000000
128.0.0.0	/1	10000000 00000000 00000000 00000000
192.0.0.0	/2	11000000 00000000 00000000 00000000
224.0.0.0	/3	11100000 00000000 00000000 00000000
240.0.0.0	/4	11110000 00000000 00000000 00000000
248.0.0.0	/5	11111000 00000000 00000000 00000000
252.0.0.0	/6	11111100 00000000 00000000 00000000
254.0.0.0	/7	11111110 00000000 00000000 00000000
255.0.0.0	/8	11111111 00000000 00000000 00000000
255.128.0.0	/9	11111111 10000000 00000000 00000000
255.192.0.0	/10	11111111 11000000 00000000 00000000
255.224.0.0	/11	11111111 11100000 00000000 00000000
255.240.0.0	/12	11111111 11110000 00000000 00000000
255.248.0.0	/13	11111111 11111000 00000000 00000000
255.252.0.0	/14	11111111 11111100 00000000 00000000
255.254.0.0	/15	11111111 11111110 00000000 00000000
255.255.0.0	/16	11111111 11111111 00000000 00000000
255.255.128.0	/17	11111111 11111111 10000000 00000000
255.255.192.0	/18	11111111 11111111 11000000 00000000
255.255.224.0	/19	11111111 11111111 11100000 00000000
255.255.240.0	/20	11111111 11111111 11110000 00000000
255.255.248.0	/21	11111111 11111111 11111000 00000000
255.255.252.0	/22	11111111 11111111 11111100 00000000
255.255.254.0	/23	11111111 11111111 11111110 00000000
255.255.255.0	/24	11111111 11111111 11111111 00000000
255.255.255.128	/25	11111111 11111111 11111111 10000000
255.255.255.192	/26	11111111 11111111 11111111 11000000
255.255.255.224	/27	11111111 11111111 11111111 11100000
255.255.255.240	/28	11111111 11111111 11111111 11110000
255.255.255.248	/29	11111111 11111111 11111111 11111000
255.255.255.252	/30	11111111 11111111 11111111 11111100
255.255.255.254	/31	11111111 11111111 11111111 11111110
255.255.255.255	/32	11111111 11111111 11111111 11111111

APPENDIX B

Exam Topics Cross-Reference

This appendix lists the exam topics associated with the CCNA 200-301 exam blueprint Version 1.1. Cisco lists the exam topics on its website. Even though changes to the exam topics are rare, you should always review those exam topics for any updates; check www.cisco.com/go/certifications and navigate to the correct exam.

Cisco organizes each list of exam topics by domains, which are major topic areas. Cisco states the percentage of the exam that should come from each domain, so you get some idea of the areas of importance. Traditionally, the score report you receive after taking the exam shows your percentage score in each domain.

This appendix includes two separate types of indices for exam topics:

- **CCNA 200-301 Exam Topic Order:** This section lists the CCNA 200-301 V1.1 exam topics in the same order Cisco lists them on its website, with a list of associated book chapters. This first list shows a cross-reference from each exam topic to the chapters that include at least some material about each topic.
- **Book Chapter Order Versus CCNA 200-301 Exam Topics:** This lists the same CCNA 200-301 V1.1 exam topics but indexed by chapter instead of exam topic. This section lists the chapters in this book, along with the exam topics that the chapter includes. This section basically relists the kind of information found on the first page of each chapter, just in condensed form in one place.

CCNA 200-301 Exam Topic Order

The CCNA 200-301 exam includes six major topic areas (domains), each with a percentage listed. Table B-1 lists the domains and their percentages.

Table B-1 CCNA 200-301 V1.1 Exam Topic Domains

Domain	Percentage
Domain 1: Network Fundamentals	20%
Domain 2: Network Access	20%
Domain 3: IP Connectivity	25%
Domain 4: IP Services	10%
Domain 5: Security Fundamentals	15%
Domain 6: Automation and Programmability	10%

Tables B-2 through B-7 list the exam topics within each of the six domains. Note that the *CCNA 200-301 Official Cert Guide, Volume 2*, Second Edition, covers some of the exam topics, while this book covers the rest.

Table B-2 CCNA 200-301 V1.1 Domain 1 Exam Topics (Network Fundamentals)

Exam Topic	Vol 1 Chapter(s)	Vol 2 Chapter(s)
1.1 Explain the role and function of network components	2, 3, 5, 7	1, 10, 18, 21, 22
1.1.a Routers	3, 16, 18	
1.1.b Layer 2 and Layer 3 Switches	2, 5, 7, 18	
1.1.c Next-generation firewalls and IPS		10
1.1.d Access points		1
1.1.e Controllers		4, 22
1.1.f Endpoints		21
1.1.g Servers		21
1.1.h PoE		18
1.2 Describe characteristics of network topology architectures	2, 3	18–21
1.2.a Two-tier		18
1.2.b Three-tier		18
1.2.c Spine-leaf		21
1.2.d WAN	3	19
1.2.e Small office/home office (SOHO)	2, 16	18
1.2.f On-premises and cloud		20
1.3 Compare physical interface and cabling types	1, 2, 7	18
1.3.a Single-mode fiber, multimode fiber, copper	1, 2	18
1.3.b Connections (Ethernet shared media and point-to-point)	1, 2, 7	18
1.4 Identify interface and cable issues (collisions, errors, mismatch duplex, and/or speed)	7	
1.5 Compare TCP to UDP		5
1.6 Configure and verify IPv4 addressing and subnetting	6, 11–16, 18	
1.7 Describe private IPv4 addressing	11, 12, 17	14
1.8 Configure and verify IPv6 addressing and prefix	25–28	
1.9 Describe IPv6 address types	25–28	
1.9.a Unicast (global, unique local, and link local)	26–28	
1.9.b Anycast	26, 27	
1.9.c Multicast	27	
1.9.d Modified EUI 64	27, 28	
1.10 Verify IP parameters for Client OS (Windows, Mac OS, Linux)	19	
1.11 Describe wireless principles		1, 3
1.11.a Nonoverlapping Wi-Fi channels		1
1.11.b SSID		1

Exam Topic	Vol 1 Chapter(s)	Vol 2 Chapter(s)
1.11.c RF		1
1.11.d Encryption		3
1.12 Explain virtualization fundamentals (server virtualization, containers, and VRFs)		20
1.13 Describe switching concepts	5, 8	
1.13.a MAC learning and aging	5, 8	
1.13.b Frame switching	5, 8	
1.13.c Frame flooding	5, 8	
1.13.d MAC address table	5, 8	

Table B-3 CCNA 200-301 V1.1 Domain 2 Exam Topics (Network Access)

Exam Topic	Vol 1 Chapter(s)	Vol 2 Chapter(s)
2.1 Configure and verify VLANs (normal range) spanning multiple switches	8, 18	
2.1.a Access ports (data and voice)	8	
2.1.b Default VLAN	8	
2.1.c InterVLAN Connectivity	8, 18	
2.2 Configure and verify interswitch connectivity	8	
2.2.a Trunk ports	8	
2.2.b 802.1Q	8	
2.2.c Native VLAN	8	
2.3 Configure and verify Layer 2 discovery protocols (Cisco Discovery Protocol and LLDP)		13
2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP)	8–10, 18	
2.5 Interpret basic operations of Rapid PVST+ Spanning Tree Protocol	5, 9, 10	
2.5.a Root port, root bridge (primary/secondary), and other port names	9, 10	
2.5.b Port states and roles	9, 10	
2.5.c PortFast	9, 10	
2.5.d Root Guard, loop guard, BPDU filter, BPDU guard	9, 10	
2.6 Compare Cisco Wireless Architectures and AP modes		2
2.7 Describe physical infrastructure connections of WLAN components (AP, WLC, access/trunk ports, and LAG)		4

Exam Topic	Vol 1 Chapter(s)	Vol 2 Chapter(s)
2.8 Describe network device management access (Telnet, SSH, HTTP, HTTPS, console, and TACACS+/RADIUS, and cloud managed)	4, 6, 20	4
2.9 Interpret the wireless LAN GUI configuration for client connectivity, such as WLAN creation, security settings, QoS profiles, and advanced settings		4

Table B-4 CCNA 200-301 Domain 3 Exam Topics (IP Connectivity)

Exam Topic	Vol 1 Chapter(s)	Vol 2 Chapter(s)
3.1 Interpret the components of routing table	17, 29	
3.1.a Routing protocol code	17, 29	
3.1.b Prefix	17, 29	
3.1.c Network mask	17, 29	
3.1.d Next hop	17, 29	
3.1.e Administrative distance	17, 24, 29	
3.1.f Metric	17	
3.1.g Gateway of last resort	17	
3.2 Determine how a router makes a forwarding decision by default	17, 21–24	
3.2.a Longest prefix match	17, 24	
3.2.b Administrative distance	17, 21–24	
3.2.c Routing protocol metric	21, 21–24	
3.3 Configure and verify IPv4 and IPv6 static routing	17, 20, 29	
3.3.a Default route	17, 20, 29	
3.3.b Network route	17, 20, 29	
3.3.c Host route	17, 20, 29	
3.3.d Floating static	17, 20, 29	
3.4 Configure and verify single area OSPFv2	21–24	
3.4.a Neighbor adjacencies	21–24	
3.4.b Point-to-point	21–24	
3.4.c Broadcast (DR/BDR selection)	21–24	
3.4.d Router ID	21–24	
3.5 Describe the purpose, functions, and concepts of first hop redundancy protocols		16

Table B-5 CCNA 200-301 V1.1 Domain 4 Exam Topics (IP Services)

Exam Topics	Vol 1 Chapter(s)	Vol 2 Chapter(s)
4.1 Configure and verify inside source NAT using static and pools		14
4.2 Configure and verify NTP operating in a client and server mode		13
4.3 Explain the role of DHCP and DNS within the network	19	5
4.4 Explain the function of SNMP in network operations		17
4.5 Describe the use of syslog features including facilities and severity levels		13
4.6 Configure and verify DHCP client and relay	6, 19	
4.7 Explain the forwarding per-hop behavior (PHB) for QoS such as classification, marking, queuing, congestion, policing, and shaping		15
4.8 Configure network devices for remote access using SSH	6	10
4.9 Describe the capabilities and functions of TFTP/FTP in the network		17

Table B-6 CCNA 200-301 Domain 5 Exam Topics (Security Fundamentals)

Exam Topics	Vol 1 Chapter(s)	Vol 2 Chapter(s)
5.1 Define key security concepts (threats, vulnerabilities, exploits, and mitigation techniques)		9
5.2 Describe security program elements (user awareness, training, and physical access control)		9
5.3 Configure device access control using local passwords	6	10
5.4 Describe security password policies elements, such as management, complexity, and password alternatives (multifactor authentication, certificates, and biometrics)		9
5.5 Describe IPsec remote access and site-to-site VPNs		19
5.6 Configure and verify access control lists		6, 7, 8
5.7 Configure Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)		11, 12
5.8 Compare authentication, authorization, and accounting concepts		9

Exam Topics	Vol 1 Chapter(s)	Vol 2 Chapter(s)
5.9 Describe wireless security protocols (WPA, WPA2, and WPA3)		3
5.10 Configure and verify WLAN within the GUI using WPA2 PSK		4

Table B-7 CCNA 200-301 V1.1 Domain 6 Exam Topics (Programmability and Automation)

Exam Topics	Vol 1 Chapter(s)	Vol 2 Chapter(s)
6.1 Explain how automation impacts network management		21
6.2 Compare traditional networks with controller-based networking		
6.3 Describe controller-based, software-defined architecture (overlay, underlay, and fabric)		21
6.3.a Separation of control plane and data plane		21, 22
6.3.b Northbound and Southbound APIs		21, 22
6.4 Explain AI (generative and predictive) and machine learning in network operations		22
6.5 Describe characteristics of REST-based APIs (authentication types, CRUD, HTTP verbs, and data encoding)		23
6.6 Recognize the capabilities of configuration management mechanisms such as Ansible and Terraform		24
6.7 Recognize components of JSON-encoded data		23

Book Chapter Order Versus CCNA 200-301 Exam Topics

Cisco organizes its exam topics based on the outcome of your learning experience, which is typically not a reasonable order for building the content of a book or course. This section lists the book chapters in sequence, with the exam topics covered in each chapter.

Table B-8 CCNA 200-301 V1.1: Chapter-to-Exam Topic Mapping

Book Chapter	Exam Topics Covered
Part I: Introduction to Networking	
Chapter 1: Introduction to TCP/IP Networking	**1.0 Network Fundamentals** 1.3 Compare physical interface and cabling types *1.3.a Single-mode fiber, multimode fiber, copper* *1.3.b Connections (Ethernet shared media and point-to-point)*

Book Chapter	Exam Topics Covered
Chapter 2: Fundamentals of Ethernet LANs	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.b Layer 2 and Layer 3 switches* 1.2 Describe characteristics of network topology architectures *1.2.e Small office/home office (SOHO)* 1.3 Compare physical interface and cabling types *1.3.a Single-mode fiber, multimode fiber, copper* *1.3.b Connections (Ethernet shared media and point-to-point)*
Chapter 3: Fundamentals of WANs and IP Routing	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.a Routers* 1.2 Describe characteristics of network topology architectures *1.2.d WAN*
Part II: Implementing Ethernet LANs	
Chapter 4: Using the Command-Line Interface	**2.0 Network Access** 2.8 Describe network device management access (Telnet, SSH, HTTP, HTTPs, console, TACACS+/RADIUS, and cloud managed)
Chapter 5: Analyzing Ethernet LAN Switching	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.b Layer 2 and Layer 3 switches* 1.13 Describe switching concepts *1.13.a MAC learning and aging* *1.13.b Frame switching* *1.13.c Frame flooding* *1.13.d MAC address table* **2.0 Network Access** 2.5 Interpret basic operations of Rapid PVST+ Spanning Tree Protocol

Book Chapter	Exam Topics Covered
Chapter 6: Configuring Basic Switch Management	**1.0 Network Fundamentals** 1.6 Configure and calculate IPv4 addressing and subnetting **2.0 Network Access** 2.8 Describe network device management access (Telnet, SSH, HTTP, HTTPs, console, TACACS+/RADIUS, and cloud managed) **4.0 IP Services** 4.6 Configure and verify DHCP client and relay 4.8 Configure network devices for remote access using SSH **5.0 Security Fundamentals** 5.3 Configure device access control using local passwords
Chapter 7: Configuring and Verifying Switch Interfaces	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.b Layer 2 and Layer 3 Switches* 1.3 Compare physical interface and cabling types *1.3.b Connections (Ethernet shared media and point-to-point)* 1.4 Identify interface and cable issues (collisions, errors, mismatch duplex, and/or speed)
Part III: Implementing VLANs and STP	
Chapter 8: Implementing Ethernet Virtual LANs	**1.0 Network Fundamentals** 1.13 Describe switching concepts *1.13.a MAC learning and aging* *1.13.b Frame switching* *1.13.c Frame flooding* *1.13.d MAC address table* **2.0 Network Access** 2.1 Configure and verify VLANs (normal range) spanning multiple switches *2.1.a Access ports (data and voice)* *2.1.b Default VLAN* *2.1.c InterVLAN Connectivity*

Book Chapter	Exam Topics Covered
	2.2 Configure and verify interswitch connectivity *2.2.a Trunk ports* *2.2.b 802.1Q* *2.2.c Native VLAN*
Chapter 9: Spanning Tree Protocol Concepts	**2.0 Network Access** 2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP) 2.5 Interpret basic operations of Rapid PVST+ Spanning Tree Protocol *2.5.a Root port, root bridge (primary/secondary), and other port names* *2.5.b Port states and roles* *2.5.c PortFast* *2.5.d Root Guard, loop guard, BPDU filter, BPDU guard*
Chapter 10: RSTP and EtherChannel Configuration	**2.0 Network Access** 2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP) 2.5 Interpret basic operations of Rapid PVST+ Spanning Tree Protocol *2.5.a Root port, root bridge (primary/secondary), and other port names* *2.5.b Port states and roles* *2.5.c PortFast* *2.5.d Root Guard, loop guard, BPDU filter, BPDU guard*
Part IV: IPv4 Addressing	
Chapter 11: Perspectives on IPv4 Subnetting	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting 1.7 Describe the need for private IPv4 addressing
Chapter 12: Analyzing Classful IPv4 Networks	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting 1.7 Describe the need for private IPv4 addressing

Book Chapter	Exam Topics Covered
Chapter 13: Analyzing Subnet Masks	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting
Chapter 14: Analyzing Existing Subnets	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting
Chapter 15: Subnet Design	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting
Part V: IPv4 Routing	
Chapter 16: Operating Cisco Routers	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.a Routers* 1.2 Describe characteristics of network topology architectures *1.2.e Small office/home office (SOHO)* 1.6 Configure and verify IPv4 addressing and subnetting
Chapter 17: Configuring IPv4 Addresses and Static Routes	**1.0 Network Fundamentals** 1.6 Configure and verify IPv4 addressing and subnetting **3.0 IP Connectivity** 3.1 Interpret the components of routing table *3.1.a Routing protocol code* *3.1.b Prefix* *3.1.c Network mask* *3.1.d Next hop* *3.1.e Administrative distance* *3.1.f Metric* *3.1.g Gateway of last resort* 3.2 Determine how a router makes a forwarding decision by default *3.2.a Longest prefix match* *3.2.b Administrative distance* 3.3 Configure and verify IPv4 and IPv6 static routing *3.3.a Default route* *3.3.b Network route* *3.3.c Host route* *3.3.d Floating static*

Book Chapter	Exam Topics Covered
Chapter 18: IP Routing in the LAN	**1.0 Network Fundamentals** 1.1 Explain the role and function of network components *1.1.a Routers* *1.1.b Layer 2 and Layer 3 Switches* 1.6 Configure and verify IPv4 addressing and subnetting **2.0 Network Access** 2.1 Configure and verify VLANs (normal range) spanning multiple switches *2.1.c InterVLAN connectivity* 2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP)
Chapter 19: IP Addressing on Hosts	**1.0 Network Fundamentals** 1.10 Verify IP parameters for Client OS (Windows, Mac OS, Linux) **4.0 IP Services** 4.3 Explain the role of DHCP and DNS within the network 4.6 Configure and verify DHCP client and relay
Chapter 20: Troubleshooting IPv4 Routing	**2.0 Network Access** 2.8 Describe network device management access (Telnet, SSH, HTTP, HTTPS, console, TACACS+/RADIUS, and cloud managed) **3.0 IP Connectivity** 3.3 Configure and verify IPv4 and IPv6 static routing *3.3.a Default route* *3.3.b Network route* *3.3.c Host route* *3.3.d Floating static*

Book Chapter	Exam Topics Covered
Part VI: OSPF	
Chapter 21: Understanding OSPF Concepts	**3.0 IP Connectivity** 3.2 Determine how a router makes a forwarding decision by default *3.2.b Administrative distance* *3.2.c Routing protocol metric* 3.4 Configure and verify single area OSPFv2 *3.4.a Neighbor adjacencies* *3.4.b Point-to-point* *3.4.c Broadcast (DR/BR selection)* *3.4.d Router ID*
Chapter 22: Implementing Basic OSPF Features	**3.0 IP Connectivity** 3.2 Determine how a router makes a forwarding decision by default *3.2.b Administrative distance* *3.2.c Routing protocol metric* 3.4 Configure and verify single area OSPFv2 *3.4.a Neighbor adjacencies* *3.4.c Broadcast (DR/BR selection)* *3.4.d (Router ID)*
Chapter 23: Implementing Optional OSPF Features	**3.0 IP Connectivity** 3.2 Determine how a router makes a forwarding decision by default *3.2.b Administrative distance* *3.2.c Routing protocol metric* 3.4 Configure and verify single area OSPFv2 *3.4.a Neighbor adjacencies* *3.4.b Point-to-point* *3.4.c Broadcast (DR/BR selection)* *3.4.d Router ID*

Book Chapter	Exam Topics Covered
Chapter 24: OSPF Neighbors and Route Selection	**3.0 IP Connectivity** 3.1 Interpret components of the routing table *3.1.e Administrative distance* 3.2 Determine how a router makes a forwarding decision by default *3.2.a Longest prefix match* *3.2.b Administrative distance* *3.2.c Routing protocol metric* 3.4 Configure and verify single area OSPFv2 *3.4.a Neighbor adjacencies* *3.4.b Point-to-point* *3.4.c Broadcast (DR/BR selection)* *3.4.d Router ID*
Part VII: IP Version 6	
Chapter 25: Fundamentals of IP Version 6	**1.0 Network Fundamentals** 1.8 Configure and verify IPv6 addressing and prefix
Chapter 26: IPv6 Addressing and Subnetting	**1.0 Network Fundamentals** 1.8 Configure and verify IPv6 addressing and prefix 1.9 Describe IPv6 address types
Chapter 27: Implementing IPv6 Addressing on Routers	**1.0 Network Fundamentals** 1.8 Configure and verify IPv6 addressing and prefix 1.9 Compare and contrast IPv6 address types *1.9.a Unicast (global, unique local, and link local)* *1.9.b Anycast* *1.9.c Multicast* *1.9.d Modified EUI 64*
Chapter 28: Implementing IPv6 Addressing on Hosts	**1.0 Network Fundamentals** 1.8 Configure and verify IPv6 addressing and prefix 1.9 Describe IPv6 address types *1.9.a Unicast (global, unique local, and link local)* *1.9.d Modified EUI 64*

Book Chapter	Exam Topics Covered
Chapter 29: Implementing IPv6 Routing	**3.0 IP Connectivity** 3.1 Interpret components of the routing table *3.1.a Routing protocol code* *3.1.b Prefix* *3.1.c Network mask* *3.1.d Next hop* *3.1.e Administrative distance* 3.3 Configure and verify IPv4 and IPv6 static routing *3.3.a Default route* *3.3.b Network route* *3.3.c Host route* *3.3.d Floating static*

APPENDIX C

Answers to the “Do I Know This Already?” Quizzes

Chapter 1

1. D and F. Of the remaining answers, Ethernet defines both physical and data-link protocols, PPP is a data-link protocol, IP is a network layer protocol, and SMTP and HTTP are application layer protocols.
2. A and G. Of the remaining answers, IP is a network layer protocol, TCP and UDP are transport layer protocols, and SMTP and HTTP are application layer protocols.
3. B. Adjacent-layer interaction occurs on one computer, with two adjacent layers in the model. The higher layer requests services from the next lower layer, and the lower layer provides the services to the next higher layer.
4. B. Same-layer interaction occurs on multiple computers. The functions defined by that layer typically need to be accomplished by multiple computers—for example, the sender setting a sequence number for a segment and the receiver acknowledging receipt of that segment. A single layer defines that process, but the implementation of that layer on multiple devices is required to accomplish the function.
5. A. Encapsulation is defined as the process of adding a header in front of data supplied by a higher layer (and possibly adding a trailer as well).
6. D. By convention, the term *frame* refers to the part of a network message that includes the data-link header and trailer, with encapsulated data. The term *packet* omits the data-link header and trailer, leaving the network layer header with its encapsulated data. The term *segment* omits the network layer header, leaving the transport layer header and its encapsulated data.

Chapter 2

1. A. The IEEE defines Ethernet LAN standards, with standard names that begin with 802.3, all of which happen to use cabling. The IEEE also defines wireless LAN standards, with standard names that begin with 802.11, which are separate standards from Ethernet.
2. C. The number before the word *BASE* defines the speed, in megabits per second (Mbps): 1000 Mbps equals 1 gigabit per second (1 Gbps). The *T* in the suffix implies twisted-pair or UTP cabling, so 1000BASE-T is the UTP-based Gigabit Ethernet standard name.
3. B. Crossover cables cross the wire at one node’s transmit pin pair to the different pins used as the receive pins on the other device. For 10- and 100-Mbps Ethernet, the specific crossover cable wiring connects the pair at pins 1 and 2 on each end of the cable to pins 3 and 6 on the other end of the cable, respectively.

4. B, D, and E. Routers, wireless access point Ethernet ports, and PC NICs all send using pins 1 and 2, whereas hubs and LAN switches transmit on pins 3 and 6. Straight-through cables connect devices that use opposite pin pairs for sending, because the cable does not need to cross the pairs.

5. B and D. Multimode fiber works with LED-based transmitters rather than laser-based transmitters. Two answers mention the type of transmitters, making one of those answers correct and one incorrect.

   Two answers mention distance. The answer that mentions the longest distance possible is incorrect because single-mode cables, not multimode cables, provide the longest distances. The other (correct) answer mentions the tradeoff of multimode being used for distances just longer than UTP's 100-meter limit, while happening to use less expensive hardware than single mode.

6. B. NICs (and switch ports) use the carrier sense multiple access with collision detection (CSMA/CD) algorithm to implement half-duplex logic. CSMA/CD attempts to avoid collisions, but it also notices when collisions do occur, with rules about how the Ethernet nodes should stop sending, wait, and try again later.

7. C. The 4-byte Ethernet FCS field, found in the Ethernet trailer, allows the receiving node to see what the sending node computed with a math formula that is a key part of the error-detection process. Note that Ethernet defines the process of detecting errors (error detection), but not error recovery.

8. B, C, and E. The pre-assigned universal MAC address, given to each Ethernet port when manufactured, breaks the address into two 3-byte halves. The first half is called the organizationally unique identifier (OUI), which the IEEE assigns to the company that builds the product as a unique hex number to be used only by that company.

9. C and D. Ethernet supports unicast addresses, which identify a single Ethernet node, and group addresses, which can be used to send one frame to multiple Ethernet nodes. The two types of group addresses are the *broadcast address* and *multicast address*.

Chapter 3

1. D. The correct answer lists one term for an Ethernet WAN link between two sites: E-line, short for Ethernet line. The other answers list common synonyms for a serial link.

2. B and D. The physical installation uses a model in which each router uses a physical Ethernet link to connect to some SP device in an SP facility called a point of presence (PoP). The Ethernet link does not span from each customer device to the other. From a data-link perspective, both routers use the same Ethernet standard header and trailer used on LANs; HDLC does not matter on these Ethernet WAN links.

3. A. PC1 will send an Ethernet frame to Router1, with PC1's MAC address as the source address and Router1's MAC address as the destination address. Router1 will remove the encapsulated IP packet from that Ethernet frame, discarding the frame header and trailer, not using it again. Router1 will forward the IP packet by first encapsulating it inside a PPP frame. Router1 will not encapsulate the original

Ethernet frame in the PPP frame but rather the IP packet. The PPP header uses different addresses than Ethernet, so the original frame’s Ethernet addresses are not used.

4. C. Routers compare the packet’s destination IP address to the router’s IP routing table, making a match and using the forwarding instructions in the matched route to forward the IP packet.
5. C. IPv4 hosts generally use basic two-branch logic. To send an IP packet to another host on the same IP network or subnet that is on the same LAN, the sender sends the IP packet directly to that host. Otherwise, the sender sends the packet to its default router (also called the default gateway).
6. A and C. Routers perform all the actions listed in the answers. However, the routing protocol does the functions in the two correct answers. Independent of the routing protocol, a router learns routes for IP subnets and IP networks directly connected to its interfaces. Routers also forward (route) IP packets, but that process is called IP routing, or IP forwarding, and is an independent process compared to the work of a routing protocol.
7. C. Address Resolution Protocol (ARP) does allow PC1 to learn information, but the information is not stored on a server. The **ping** command does let the user at PC1 learn whether packets can flow in the network, but it again does not use a server. With the Domain Name System (DNS), PC1 acts as a DNS client, relying on a DNS server to respond with information about the IP addresses that match a given hostname.

Chapter 4

1. A and B. The command in the question is an EXEC command that happens to require only user mode access. As such, you can use this command in both user mode and enable mode. Because it is an EXEC command, you cannot use the command (as shown in the question) in configuration mode. Note that you can put the word **do** in front of the EXEC command while in configuration mode (for example, **do show mac address-table**) to issue the command from inside any configuration mode.
2. B. The command referenced in the question, the **reload** command, is an EXEC command that happens to require privileged mode, also known as enable mode. This command is not available in user mode. Note that you can put the word **do** in front of the EXEC command while in configuration mode (for example, **do reload**) to issue the command from inside any configuration mode.
3. B. SSH provides a secure remote login option, encrypting all data flows, including password exchanges. Telnet sends all data (including passwords) as clear text.
4. A. Switches (and routers) keep the currently used configuration in RAM, using NVRAM to store the configuration file that is loaded when the switch (or router) next loads the IOS.
5. F. The startup-config file is in NVRAM, and the running-config file is in RAM.
6. B and C. The **exit** command moves the user one config mode backward, toward global configuration mode, or if already in global configuration mode, it moves the user back to enable mode. From console mode, it moves the user back to global configuration mode. The **end** command and the Ctrl+Z key sequence both move the user back to enable mode regardless of the current configuration submode.

Chapter 5

1. A. A switch compares the destination MAC address to the MAC address table. If a matching entry is found, the switch forwards the frame out the appropriate interface. If no matching entry is found, the switch floods the frame.
2. B. A switch floods broadcast frames, multicast frames (if no multicast optimizations are enabled), and unknown unicast destination frames (frames whose destination MAC address is not in the MAC address table). FFFF.FFFF.FFFF is the Ethernet broadcast address, so the switch floods the frame, which means that the switch forwards copies of the frame out all other ports except the arrival port.
3. A. A switch floods broadcast frames, multicast frames (if no multicast optimizations are enabled), and unknown unicast destination frames (frames whose destination MAC address is not in the MAC address table). Of the available answers, the correct answer defines how a switch floods a frame.

   Of the incorrect answers, one incorrect answer describes how a switch forwards known unicast frames by finding the matching entry in the MAC address table. Another describes MAC learning, in which the switch learns the source MAC address of incoming frames. Yet another incorrect answer mentions comparing the destination IP address to the destination MAC address, which the switch does not do.
4. B. Switches need to learn the location of each MAC address used in the LAN relative to that local switch. When a switch receives a frame, the source MAC identifies the sender. The interface in which the frame arrives identifies the local switch interface closest to that node in the LAN topology.
5. C. The **show interfaces status** command lists one line of output per interface. Cisco Catalyst switches name the type of interface based on the fastest speed of the interface, so 10/100 interfaces would be Fast Ethernet. With a working connection, ports from FastEthernet 0/1 through 0/10 would be listed in a connected state, while the rest would be listed in a notconnected state.
6. D. For the correct answer, each entry lists the learned MAC address. By definition, dynamically learned MAC addresses are learned by looking at the source MAC address of received frames. (That fact rules out one of the incorrect answers as well.)

   The **show mac address-table dynamic** command lists the current list of MAC table entries, with three known entries at the point at which the command output was gathered. The counter in the last line of output lists the number of current entries, not the total number of learned MAC addresses since the last reboot. For instance, the switch could have learned other MAC addresses whose entries timed out from the MAC address table.

   Finally, the answer that claims that port Gi0/2 connects directly to a device with a particular MAC address may or may not be true. That port could connect to another switch, and another, and so on, with one of those switches connecting to the device that uses the listed MAC address.

Chapter 6

1. B. If both commands are configured, IOS accepts only the password as configured in the **enable secret** command.

2. A. To answer this question, it might be best to first think of the complete configuration and then find any answers that match the configuration. The commands, in vty line configuration mode, would be **password** *password* and **login**. Only one answer lists a vty subcommand that is one of these two commands.

   Of note in the incorrect answers:

   One answer mentions console subcommands. The console does not define what happens when remote users log in; those details sit in the vty line configuration.

   One answer mentions the **login local** command; this command means that the switch should use the local list of configured usernames/passwords. The question stated that the engineer wanted to use passwords only, with no usernames.

   One answer mentions the **transport input ssh** command, which, by omitting the **telnet** keyword, disables Telnet. While that command can be useful, SSH does not work when using passwords only; SSH requires both a username and a password. So, by disabling Telnet (and allowing SSH only), the configuration would allow no one to remotely log in to the switch.

3. B and C. SSH requires the use of usernames in addition to a password. Using the **username** global command would be one way to define usernames (and matching passwords) to support SSH. The vty lines would also need to be configured to require the use of usernames, with the **login local** vty subcommand being one such option.

   The **transport input ssh** command could be part of a meaningful configuration, but it is not a global configuration command (as claimed in one wrong answer). Likewise, one answer refers to the **username** command as a command in vty config mode, which is also the wrong mode.

4. A, D, and F. To allow access through Telnet, the switch must have password security enabled, at a minimum using the **password** vty line configuration subcommand. In addition, the switch needs an IP address (configured under one VLAN interface) and a default gateway when the switch needs to communicate with hosts in a different subnet.

5. B and C. To allow SSH or Telnet access, a switch must have a correct IP configuration. That includes the configuration of a correct IP address and mask on a VLAN interface. That VLAN interface then must have a path out of the switch via ports assigned to that VLAN. In this case, with all ports assigned to VLAN 2, the switch must use interface VLAN 2 (using the **interface vlan 2** configuration command).

   To meet the requirement to support login from hosts outside the local subnet, the switch must configure a correct default gateway setting with the **ip default-gateway 172.16.2.254** global command in this case.

6. A. The **logging synchronous** line subcommand synchronizes the log message display with other command output so the log message does not interrupt a **show**

command's output. The **no ip domain-lookup** command is not a line subcommand. The other two incorrect answers are line subcommands but do not configure the function listed in the question.

Chapter 7

1. A and C. Because both devices use IEEE autonegotiation, they declare their speed and duplex capabilities to the other via messages sent out-of-band using Fast Link Pulses (FLPs). Then both devices choose the fastest speed that both support. They also use the best duplex setting that both support, with full duplex being better than half.
2. E. Cisco switches support per-interface settings for speed (with the **speed** command) and duplex (with the **duplex** command) in interface configuration mode.
3. B and E. Because the PC disables autonegotiation, it does not send autonegotiation FLP messages. However, it does start sending Ethernet frames based on the physical layer standard configured on the PC. After not receiving any autonegotiation FLP messages, the switch port analyzes the incoming signal to determine the standard used by the PC to send Ethernet frames. That analysis identifies the speed used by the PC, so the switch chooses to also use that speed. The switch then chooses the duplex based on a table of defaults: full duplex for speeds of 1 Gbps and faster or half duplex for speeds slower than 1 Gbps.
4. C. The **shutdown** interface subcommand administratively disables the interface, while the engineer can log in remotely on the weekend and configure the **no shutdown** interface subcommand to re-enable the interface. Note that unplugging the cable would prevent the interface from being used but would not allow the engineer to enable the interface remotely during the weekend change window. The **disable** and **enable** commands shown in a few answers do not exist.
5. B and D. The **interface range** global configuration command identifies a set of interfaces and moves the user into interface configuration mode. At that point, the switch applies any interface subcommands to all the interfaces in that range. However, the switch does not keep the **interface range** command in the configuration. For instance, in this case, the switch would list the **description** command under all the interfaces in the range: interfaces GigabitEthernet1/0/10 through GigabitEthernet 1/0/20 (11 interfaces in total).
6. A, B, and D. The disabled state in the **show interfaces status** command is the same as an "administratively down and down" state shown in the **show interfaces** command. The interface must be in a connected state (per the **show interfaces status** command) before the switch can send frames out of the interface.
7. A and C. First, note that some Cisco switch ports disable autonegotiation when configured with both **speed** and **duplex** on a port, as described in this question. A problem occurs when the combination of configured speed and duplex leads the autonegotiating device on the other end of the link to choose a different duplex setting—an effect called a duplex mismatch.

   To summarize the scenario in this case, SW1 uses autonegotiation but receives no autonegotiation messages from SW2. As a result, SW1 then begins using alternate logic (called parallel detection) that does not rely on autonegotiation messages. SW2 does not use autonegotiation but has a speed set, so SW2 begins sending Ethernet frames per that speed on the link. SW1 senses the speed of those signals and uses the same speed, so a speed mismatch will not exist. However, SW1 then has to choose the duplex based on a default table, choosing half duplex if the speed is 10 or 100 Mbps, and full duplex if the speed is 1 Gbps or faster.

In the case of SW2 using **speed 100** and **duplex full** settings, SW1 senses the 100-Mbps speed and defaults to use half duplex, resulting in a duplex mismatch. Similar logic applies to the **speed 10** and **duplex full** case.

8. D. For the two answers about a duplex mismatch, that condition does cause collisions, particularly late collisions. However, only the side using CSMA/CD logic (the half-duplex side) has any concept of collisions. So, if switch SW1 were using half duplex and SW2 using full duplex, SW1 would likely see late collisions and show the counter incrementing over time.

   If switch SW2 had shut down its interface, switch SW1's interface would be in a down/down state, and none of the counters would increment. Also, if both switch ports use different speed settings, the ports would be in a down/down state, and none of the interface counters would increment.

Chapter 8

1. B. A VLAN is a set of devices in the same Layer 2 broadcast domain. A subnet often includes the exact same set of devices, but it is a Layer 3 concept. A collision domain refers to a set of Ethernet devices, but with different rules than VLAN rules for determining which devices are in the same collision domain.
2. D. Although a subnet and a VLAN are not equivalent concepts, the devices in one VLAN are typically in the same IP subnet and vice versa.
3. B. The 802.1Q trunking defines a 4-byte header, inserted after the original frame's destination and source MAC address fields. The insertion of this header does not change the original frame's source or destination address. The header itself holds a 12-bit VLAN ID field, which identifies the VLAN associated with the frame.
4. A and C. The **dynamic auto** setting means that the switch can negotiate trunking, but it can only respond to negotiation messages, and it cannot initiate the negotiation process. So, the other switch must be configured to trunk (**switchport mode trunk**) or to initiate the dynamic negotiation process (**switchport mode dynamic desirable**).
5. A and B. The configured VTP setting of VTP transparent mode means that the switch can configure VLANs, so the VLAN is configured. In addition, the VLAN configuration details, including the VLAN name, show up as part of the running-config file.
6. B and C. The **show interfaces switchport** command lists both the administrative and operational status of each port. When a switch considers a port to be trunking, this command lists an operational trunking state of "trunk." The **show interfaces trunk** command lists a set of interfaces—the interfaces that are currently operating as trunks. So, both of these commands identify interfaces that are operational trunks.

7. A and B. On switches that do not use VTP (by using VTP modes off or transparent), the switch lists all VLAN configuration in the configuration file (making one answer correct). Also, the **show vlan brief** command lists all defined VLANs, regardless of VTP mode and regardless of shutdown state. As a result, the two answers that mention commands are correct.

   The other two answers are incorrect because VLAN 30 has been shut down, which means the switch will not forward frames in that VLAN, regardless of whether they arrive on access or trunk ports.

8. B. The first list of VLAN IDs includes all VLANs (1–4094) except those overtly removed per the details in any **switchport trunk allowed vlan** interface subcommands on the trunk interface. If no such commands are configured, the first list in the output will include 1–4094. The two incorrect answers that mention VLAN 30 list conditions that change the second of two lists of VLANs in the command output, while STP's choice to block an interface would impact the third list.

Chapter 9

1. A and B. Listening and learning are transitory port states, used only when moving from the blocking to the forwarding state. Discarding is not an STP port state.

2. C. The smallest numeric bridge ID wins the election. The bridge IDs in the answers break into a decimal priority on the left, a colon, and then the 12-digit hexadecimal MAC address of the switch. Of all the answers, two tie with the lowest priority (4097). Of those, the correct answer lists a lower MAC address.

3. C and D. Listening and learning are transitory port states used only when moving from the blocking to the forwarding state. Discarding is not an STP port state. Forwarding and blocking are stable states.

4. B. Nonroot switches forward Hellos received from the root; the root sends these Hellos based on the root's configured Hello timer.

5. B and D. RSTP uses port states forwarding, learning, and discarding. Forwarding and learning perform the same functions as the port states used by traditional STP.

6. A and D. With RSTP, an alternate port is an alternate to the root port when a switch's root port fails. A backup port takes over for a designated port if the designated port fails.

7. D. The PortFast feature allows STP/RSTP to move a port from blocking to forwarding without going through the interim listening and learning states. STP allows this exception when the link is known to have no switch on the other end of the link, removing the risk of a switching loop. Cisco created PortFast before the IEEE released RSTP, but RSTP included the equivalent feature as well, so it is a feature of both STP and RSTP.

   BPDU Guard is a common feature to use at the same time as PortFast because it watches for incoming bridge protocol data units (BPDUs), which should not happen on an access port, and prevents the loops from a rogue switch by disabling the port.

8. C. Root Guard on a switch interface monitors incoming STP BPDUs, processing them as normal with STP, with one exception: superior BPDUs. Superior BPDUs identify a

new root switch with a better (lower) Bridge ID. Root Guard reacts to receiving such a BPDU and disables the port.

For the other answers, BPDU Guard monitors for incoming BPDUs but disables the port for any received BPDU on the port. Neither PortFast nor Loop Guard monitor incoming BPDUs or react to them.

Chapter 10

1. A. Of the four answers, only **pvst** and **rapid-pvst** are valid options on the command. Of those, the **rapid-pvst** option enables Rapid Per VLAN Spanning Tree (RPVST+), which uses RSTP. The **pvst** option enables Per VLAN Spanning Tree (PVST), which uses STP, not RSTP. The other two options, if attempted, would cause the command to be rejected because these option do not exist.
2. A and C. The system ID extension (or extended system ID) part of a bridge ID contains 12 bits and sits after the 4-bit priority field and before the 48-bit system ID. Switches use this field to store the VLAN ID when using STP or RSTP to build spanning trees per VLAN. So, of the two answers that mention the system ID extension, the one that lists the VLAN ID, in this case 5, is correct.

   The output also lists a priority of 32773. However, that output lists the decimal equivalent of the 16-bit priority value. In reality, this decimal value is the sum of the configured decimal priority plus the VLAN ID: 32768 + 5 = 32773. So in this case, the root’s configured priority is 32,768.
3. A, B, and D. Cisco’s Rapid Per VLAN Spanning Tree (RPVST+) creates one spanning tree instance per VLAN. To do so, it sends BPDUs per VLAN. Each switch identifies itself with a unique Bridge ID (BID) per VLAN, made unique per VLAN by adding the VLAN ID to the system ID extension 12-bit field of the BID. RVPST also adds a new Type-Length Value (TLV) to the BPDU itself, which includes a place to list the VLAN ID. Finally, when transmitting the BPDUs over VLAN trunks, the switch uses a trunking header that lists the VLAN ID (a practice sometimes called tunneling in 802.1Q). The receiving switch can check all three locations that list the VLAN ID to ensure that they all agree about what VLAN the BPDU is describing. Of the four answers, the three correct answers describe the three actual locations in which RPVST+ lists the VLAN ID.
4. B and C. BPDU Guard disables a port by placing it into an error disabled (err-disabled) interface state. BPDU Guard does so when it is enabled on the interface, regardless of whether PortFast is also enabled. The two correct answers both state that BPDU Guard is enabled, while the two incorrect answers list it as disabled.
5. A and E. Root Guard reacts to the receipt of a superior BPDU by disabling the port. To do so, it leaves the interface state as is in a connected state. Instead, it manipulates the STP port state, changing it to a special state called broken. The commands display the broken state with letters BRK. So, the **show interfaces status** command lists a connected interface state, while the **show spanning-tree** command lists a port state of BRK, or broken, which stops all traffic on the interface.

6. D. IOS uses the **channel-group** configuration command to create an EtherChannel. Then the term *etherchannel* is used in the **show etherchannel** command, which displays the status of the channel. The output of this **show** command then names the channel a *PortChannel*. The only answer that is not used somewhere in IOS to describe this multilink channel is *Ethernet-Channel*.

7. B and D. The **channel-group** command will direct the switch to use LACP to dynamically negotiate to add a link to an EtherChannel when the command uses the **active** and **passive** keywords, respectively. The **desirable** and **passive** keywords direct the switch to use PAgP instead of LACP. Of the four answers, the two correct answers use two LACP values, while the two incorrect answers use at least one value that would cause the switch to use PAgP, making the answers incorrect.

   Of the two correct answers, both combinations result in the switches attempting to add the link to an EtherChannel using LACP as the negotiation protocol. If both switches used the **passive** keyword, they would both sit and wait for the other switch to begin sending LACP messages and therefore never attempt to add the link to the channel.

8. C. EtherChannel load distribution, or load balancing, on Cisco Catalyst switches uses an algorithm. The algorithm examines some fields in the various headers, so messages that have the same values in those fields always flow over the same link in a particular EtherChannel. Note that it does not break the frames into smaller fragments or use a round-robin approach that ignores the header values, and it does not examine link utilization when making the choice.

Chapter 11

1. B and D. The general rule to determine whether two devices' interfaces should be in the same subnet is whether the two interfaces are separated from each other by a router. To provide a way for hosts in one VLAN to send data to hosts outside that VLAN, a local router must connect its LAN interface to the same VLAN as the hosts and have an address in the same subnet as the hosts. All the hosts in that same VLAN on the same switch would not be separated from each other by a router, so these hosts would also be in the same subnet. However, another PC, connected to the same switch but in a different VLAN, will require its packets to flow through a router to reach Host A, so Host A's IP address would need to be in a different subnet compared to this new host.

2. D. By definition, two address values in every IPv4 subnet cannot be used as host IPv4 addresses: the first (lowest) numeric value in the subnet for the subnet ID and the last (highest) numeric value in the subnet for the subnet broadcast address.

3. B and C. At least 7 subnet bits are needed because $2^6 = 64$, so 6 subnet bits could not number 100 different subnets. Seven subnet bits could because $2^7 = 128 => 100$. Similarly, 6 host bits is not enough because $2^6 - 2 = 62$, but 7 host bits is enough because $2^7 - 2 = 126 => 100$.

   The number of network, subnet, and host bits must total 32 bits, making one of the answers incorrect. The answer with 8 network bits cannot be correct because the question states that a Class B network is used, so the number of network bits must always be 16. The two correct answers have 16 network bits (required because the question states the use of a Class B network) and at least 7 subnet and host bits each.

4. A and C. The private IPv4 networks, defined by RFC 1918, are Class A network 10.0.0.0, the 16 Class B networks from 172.16.0.0 to 172.31.0.0, and the 256 Class C networks that begin with 192.168.
5. A, D, and E. The private IPv4 networks, defined by RFC 1918, are Class A network 10.0.0.0, the 16 Class B networks from 172.16.0.0 to 172.31.0.0, and the 256 Class C networks that begin with 192.168. The three correct answers are from the public IP network range, and none are reserved values.
6. A and C. An unsubnetted Class A, B, or C network has two parts: the network and host parts.
7. B. An unsubnetted Class A, B, or C network has two parts: the network and host parts. The subnet part does not exist in that case. To perform subnetting, the engineer creates a new subnet part by choosing to use a subnet mask, defining a smaller number of host bits, which makes space for some bits to be used to number different subnets. So, the host part of the address structure gets smaller in the after-subnetting case. The subnet part of the address structure moves from size 0 (nonexistent) to some number of subnet bits after the engineer chooses a subnet (nondefault) mask. The network part remains a constant size throughout, whether subnetting or not.

Chapter 12

1. B and C. Class A networks have a first octet in the range of 1–126, inclusive, and their network IDs have a 0 in the last three octets. A network ID of 130.0.0.0 is actually a Class B network (first octet range 128–191, inclusive). All addresses that begin with 127 are reserved, so 127.0.0.0 is not a Class A network.
2. E. All Class B networks begin with values between 128 and 191, inclusive, in their first octets. The network ID has any value in the 128–191 range in the first octet, and any value from 0 to 255 inclusive in the second octet, with decimal 0s in the final two octets. Two of the answers show a 255 in the second octet, which is acceptable. Two of the answers show a 0 in the second octet, which is also acceptable.
3. B and D. The first octet (172) is in the range of values for Class B addresses (128–191). As a result, the network ID can be formed by copying the first two octets (172.16) and writing 0s for the last two octets (172.16.0.0). The default mask for all Class B networks is 255.255.0.0, and the number of host bits in all unsubnetted Class B networks is 16.
4. A and C. The first octet (192) is in the range of values for Class C addresses (192–223). As a result, the network ID can be formed by copying the first three octets (192.168.6) and writing 0 for the last octet (192.168.6.0). The default mask for all Class C networks is 255.255.255.0, and the number of host bits in all unsubnetted Class C networks is 8.
5. D. To find the network broadcast address, first determine the class, and then determine the number of host octets. At that point, convert the host octets to 255 to create the network broadcast address. In this case, 10.1.255.255 is in a Class A network, with the last three octets as host octets, for a network broadcast address of

10.255.255.255. For 192.168.255.1, it is a Class C address, with the last octet as the host part, for a network broadcast address of 192.168.255.255. Address 224.1.1.255 is a Class D address, so it is not in any unicast IP network and the question does not apply. For 172.30.255.255, it is a Class B address, with the last two octets as host octets, so the network broadcast address is 172.30.255.255.

Chapter 13

1. C. If you think about the conversion one octet at a time, the first two octets each convert to 8 binary 1s. The 254 converts to 8-bit binary 11111110, and the decimal 0 converts to 8-bit binary 00000000. So, the total number of binary 1s (which defines the prefix length) is 8 + 8 + 7 + 0 = /23.
2. B. If you think about the conversion one octet at a time, the first three octets each convert to 8 binary 1s. The 240 converts to 8-bit binary 11110000, so the total number of binary 1s (which defines the prefix length) is 8 + 8 + 8 + 4 = /28.
3. B. Remember that /30 is the equivalent of the mask that in binary has 30 binary 1s. To convert that to DDN format, write down all the binary 1s (30 in this case), followed by binary 0s for the remainder of the 32-bit mask. Then take 8 bits at a time and convert from binary to decimal (or memorize the nine possible DDN mask octet values and their binary equivalents). Using the /30 mask in this question, the binary mask is 11111111 11111111 11111111 11111100. Each of the first three octets is all binary 1s, so each converts to 255. The last octet, 11111100, converts to 252, for a DDN mask of 255.255.255.252. See Appendix A, "Numeric Reference Tables," for a decimal/binary conversion table.
4. C. The size of the network part is always either 8, 16, or 24 bits, based on whether it is Class A, B, or C, respectively. As a Class A address, N=8. The mask 255.255.255.0, converted to prefix format, is /24. The number of subnet bits is the difference between the prefix length (24) and N, so S=16 in this case. The size of the host part is a number that, when added to the prefix length (24), gives you 32, so H=8 in this case.
5. A. The size of the network part is always either 8, 16, or 24 bits, based on whether it is Class A, B, or C, respectively. As a Class C address, N=24. The number of subnet bits is the difference between the prefix length (27) and N, so S=3 in this case. The size of the host part is a number that, when added to the prefix length (27), gives you 32, so H=5 in this case.
6. D. Classless addressing rules define a two-part IP address structure: the prefix and the host part. This logic ignores Class A, B, and C rules and can be applied to the 32-bit IPv4 addresses from any address class. By ignoring Class A, B, and C rules, classless addressing ignores any distinction as to the network part of an IPv4 address.
7. A and B. The masks in binary define a number of binary 1s, and the number of binary 1s defines the length of the prefix (network + subnet) part. With a Class B network, the network part is 16 bits. To support 100 subnets, the subnet part must be at least 7 bits long. Six subnet bits would supply only 2^6 = 64 subnets, while 7 subnet bits supply 2^7 = 128 subnets. The /24 answer supplies 8 subnet bits, and the 255.255.255.252 answer supplies 14 subnet bits.

Chapter 14

1. D. When using classful IP addressing concepts as described in Chapter 13, “Analyzing Subnet Masks,” addresses have three parts: network, subnet, and host. For addresses in a single classful network, the network parts must be identical for the numbers to be in the same network. For addresses in the same subnet, both the network and subnet parts must have identical values. The host part differs when comparing different addresses in the same subnet.
2. B and D. In any subnet, the subnet ID is the smallest number in the range, the subnet broadcast address is the largest number, and the usable IP addresses sit between them. All numbers in a subnet have identical binary values in the prefix part (classless view) and network + subnet part (classful view). To be the lowest number, the subnet ID must have the lowest possible binary value (all 0s) in the host part. To be the largest number, the broadcast address must have the highest possible binary value (all binary 1s) in the host part. The usable addresses do not include the subnet ID and subnet broadcast address, so the addresses in the range of usable IP addresses never have a value of all 0s or 1s in their host parts.
3. C. The mask converts to 255.255.255.0. To find the subnet ID, for each octet of the mask that is 255, you can copy the IP address’s corresponding values. For mask octets of decimal 0, you can record a 0 in that octet of the subnet ID. As such, copy the 10.7.99 and write a 0 for the fourth octet, for a subnet ID of 10.7.99.0.
4. C. First, the resident subnet (the subnet ID of the subnet in which the address resides) must be numerically smaller than the IP address, which rules out one of the answers. The mask converts to 255.255.255.252. As such, you can copy the first three octets of the IP address because of their value of 255. For the fourth octet, the subnet ID value must be a multiple of 4, because 256 – 252 (mask) = 4. Those multiples include 96 and 100, and the right choice is the multiple closest to the IP address value in that octet (97) without going over. So, the correct subnet ID is 192.168.44.96.
5. C. The resident subnet ID in this case is 172.31.77.192. You can find the subnet broadcast address based on the subnet ID and mask using several methods. Following the decimal process in the book, the mask converts to 255.255.255.224, making the interesting octet be octet 4, with magic number 256 – 224 = 32. For the three octets where the mask = 255, copy the subnet ID (172.31.77). For the interesting octet, take the subnet ID value (192), add magic (32), and subtract 1, for 223. That makes the subnet broadcast address 172.31.77.223.
6. C. To answer this question, you need to find the range of addresses in the subnet, which typically then means you need to calculate the subnet ID and subnet broadcast address. With a subnet ID/mask of 10.1.4.0/23, the mask converts to 255.255.254.0. To find the subnet broadcast address, following the decimal process described in this chapter, you can copy the subnet ID’s first two octets because the mask’s value is 255 in each octet. You write a 255 in the fourth octet because the mask has a 0 on the fourth octet. In octet 3, the interesting octet, add the magic number (2) to the subnet ID’s value (4), minus 1, for a value of 2 + 4 – 1 = 5. (The magic number in this case is calculated as 256 – 254 = 2.) That makes the broadcast address 10.1.5.255. The last usable address is 1 less: 10.1.5.254. The range that includes the last 100 addresses is 10.1.5.155 – 10.1.5.254.

Chapter 15

1. A. With 50 percent growth, the mask needs to define enough subnet bits to create 150 subnets. As a result, the mask needs at least 8 subnet bits (7 subnet bits supply 2^7, or 128, subnets, and 8 subnet bits supply 2^8, or 256, subnets). Similarly, the need for 50 percent growth in the size for the largest subnet means that the host part needs enough bits to number 750 hosts/subnet. Nine host bits are not enough ($2^9 - 2 = 510$), but 10 host bits supply 1022 hosts/subnet ($2^{10} - 2 = 1022$). With 16 network bits existing because of the choice to use a Class B network, the design needs a total of 34 bits (at least) in the mask (16 network, 8 subnet, 10 host), but only 32 bits exist—so no single mask meets the requirements.

2. B. With a growth of 20 percent, the design needs to support 240 subnets. Seven subnet bits do not meet the need ($2^7 = 128$), but 8 subnet bits do meet the need ($2^8 = 256$). To support 120 hosts/subnet, with 20% growth, the mask should support 144 hosts/subnet. That number requires 8 host bits ($2^8 - 2 = 254$). As a result, you need a minimum 8 subnet bits and 8 host bits.

   The right answer, 10.0.0.0/22, has 8 network bits because the network class is Class A, 14 subnet bits (/22 – 8 = 14), and 10 host bits (32 – 22 = 10). This mask supplies at least 8 subnet bits and at least 8 host bits.

   The answer with the /25 mask supplies only 7 host bits, making it incorrect. The answer showing 172.16.0.0/23 supplies 9 host bits, which is enough; however, it uses 16 network bits with the Class B network, leaving too few subnet bits (7). The answer that shows Class C network 192.168.7.0 with mask /24 supplies 8 host bits but 0 subnet bits.

3. B. To support 1000 subnets, 10 subnet bits ($2^{10} = 1024$) are needed. The design uses a Class B network, which means that 16 network bits exist as well. So, the shortest mask that meets the requirements is 255.255.255.192, or /26, composed of 16 network plus 10 subnet bits. The /28 answer also supplies enough subnets to meet the need, but compared to /26, /28 supplies fewer host bits and so fewer hosts/subnet.

4. C and D. The mask converts to 255.255.252.0, so the difference from subnet ID to subnet ID (called the magic number in this chapter) is 256 – 252 = 4. So, the subnet IDs start with 172.30.0.0, then 172.30.4.0, then 172.30.8.0, and so on, adding 4 to the third octet. The mask, used with a Class B network, implies 6 subnet bits, for 64 total subnet IDs. The last of these, 172.30.252.0, can be recognized in part because the third octet, where the subnet bits sit, has the same value as the mask in that third octet.

5. A. The first (numerically lowest) subnet ID is the same number as the classful network number, or 192.168.9.0. The remaining subnet IDs are each 8 larger than the previous subnet ID, in sequence, or 192.168.9.8, 192.168.9.16, 192.168.9.24, 192.168.9.32, and so on, through 192.168.9.248.

6. D. Using mask /24 (255.255.255.0), the subnet IDs increment by 1 in the third octet. The reasoning is that with a Class B network, 16 network bits exist, and with mask /24, the next 8 bits are subnet bits, so the entire third octet contains subnet bits. All the subnet IDs will have a 0 as the last octet, because the entire fourth octet consists of host bits. Note that 172.19.0.0 (the zero subnet) and 172.19.255.0 (the broadcast subnet) might look odd but are valid subnet IDs.

Chapter 16

1. B and D. Cisco routers originally used IOS, with some models today still using IOS. Most of the enterprise router product line uses the newer IOS XE operating system. CatOS, short for Catalyst OS, refers to the original Cisco switch operating system.
2. B. The switch and router CLI follows the same basic flow with many commands in common. The three incorrect answers are incorrect because they describe actions that can occur on both routers and switches. However, the user must configure router interfaces with IP addresses. Switches, when used as Layer 2 switches only, do not need any IP addresses on their Layer 2 physical interfaces.
3. A and C. To route packets on an interface, the router interface configuration must include an IP address and mask. One correct command shows the correct single command used to configure both values, while one incorrect command shows those settings as two separate (nonexistent) commands. Also, to route packets, the interface much reach an "up/up" state; that is, the **show interfaces** and other commands list two status values, and both must be "up." The **no shutdown** command enables the interface so that it can reach an up/up state, assuming the interface has correct cabling and is connected to an appropriate device. One incorrect answer mentions the **description** command, which is useful but has nothing to do with making the interface work properly.
4. B. If the first of the two status codes is "down," it typically means that a Layer 1 problem exists. In this case, the question states that the router connects to a switch with a UTP straight-through cable, which is the correct cable pinout. Of the two answers that mention the **shutdown** command, if the router interface were shut down, the first router status code would be "administratively down," so that answer is incorrect. However, if the neighboring device interface sits in a shutdown state, the router will sense no electrical signals over the cable, seeing that as a physical problem, and place the interface into a "down/down" state, making that answer correct.

   Second, the two answers that mention interface IP addresses have no impact on the status codes of the **show interfaces brief** command. Both answers imply that the interface does not have an IP address configured; however, the IP address configuration has no effect on the interface status codes, making both answers incorrect.
5. C. The **show ip interface brief** command lists all the interface IPv4 addresses but none of the masks. The other three commands list both the address and mask.
6. B. A router has one IPv4 address for each interface in use, whereas a LAN switch has a single IPv4 address that is just used for accessing the switch. The rest of the answers list configuration settings that use the same conventions on both routers and switches.

Chapter 17

1. A and C. The route defines the group of addresses represented by the route using the subnet ID and mask. The router can use those numbers to find the range of addresses that should be matched by this route. The other two answers list facts useful when forwarding packets that happen to match the route.

2. D. Each time a router routes an IP packet, it de-encapsulates (removes) the IP packet from the incoming data-link frame. Once it decides where to forward the packet next, it re-encapsulates the packet in a new data-link frame. That occurs even if the incoming and outgoing data links happen to be the same type, as is the case in this scenario. So, all three routers de-encapsulate the IP packet. Since all links are Ethernet links, all three de-encapsulation actions removed the packet from an Ethernet frame.

3. A and D. First, for the subnetting math, address 10.1.1.100, with mask /26, implies a subnet ID of 10.1.1.64. Also, mask /26 converts to a DDN mask of 255.255.255.192. For any working router interface, after adding the **ip address** command to configure an address and mask, the router adds a connected route for the subnet. In this case, that means the router adds a connected route for subnet 10.1.1.64 255.255.255.192. The router also adds a route called a local route, which is a route for the interface IP address with a 255.255.255.255 mask. In this case, that means the router adds a local route for address 10.1.1.100 with mask 255.255.255.255.

4. B and C. The **ip route** command can refer to the IP address of the next-hop router on the link between the two routers, or to the local router's outgoing interface ID. The incorrect answers reverse those items, mentioning the local router's IP address and the next-hop router's interface ID.

5. A. The correct syntax lists a subnet number, then a subnet mask in dotted-decimal form, and then either an outgoing interface or a next-hop IP address. Of the incorrect answers, one omits the subnet mask, while two use a prefix-style mask instead of a DDN mask.

6. B. The network engineer issued the command, but the router did not add an IP route. So, either the command had a syntax error, or the router accepted the command but has some reason to believe that it should not add a route to the table.

   Two (incorrect) answers suggest the command has a syntax error: one answer with a general claim of a syntax error, and another explicitly stating that the next-hop IP address is missing. However, the **ip route 10.1.1.0 255.255.255.0 s0/0/0** command is syntactically correct. Note that with outgoing interface S0/0/0 listed, the command does not need a next-hop IP address.

   As for reasons why IOS would not add a route once it accepts the command into the configuration, IOS performs several checks of the contents of a valid **ip route** command before adding the route to the routing table. It checks whether the outgoing interface is up/up (as noted in this question's correct answer) and whether it has a route to reach the next-hop address. Also, if the router already has a route to the same subnet learned from another source, the router checks whether the other route has a better administrative distance.

Chapter 18

1. A and F. Of all the commands listed, only the two correct answers are syntactically correct router configuration commands. The command to enable 802.1Q trunking is **encapsulation dot1q** *vlan_id*.

2. B and C. Subinterface G0/1.1 must be in an administratively down state due to the **shutdown** command being issued on that subinterface. For subinterface G0/1.2, its

status cannot be administratively down because of the **no shutdown** command. G0/1.2’s state will then track to the state of the underlying physical interface. With a physical interface state of down/down, subinterface G0/1.2 will be in a down/down state in this case.

3. C. The configuration of the Layer 3 switch’s routing feature uses VLAN interfaces. The VLAN interface numbers must match the associated VLAN ID, so with VLANs 1, 2, and 3 in use, the switch will configure **interface vlan 1**, **interface vlan 2** (which is the correct answer), and **interface vlan 3**. The matching connected routes, like all connected IP routes, will list the VLAN interfaces.

   As for the incorrect answers, a list of connected routes will not list any next-hop IP addresses. Each route will list an outgoing interface; the outgoing interface will not be a physical interface, but rather a VLAN interface, because the question states that the configuration uses SVIs. Finally, all the listed subnets have a /25 mask, which is 255.255.255.128, so none of the routes will list a 255.255.255.0 mask.

4. C and D. First, for the correct answers, a Layer 3 switch will not route packets on a VLAN interface unless it is in an up/up state. When using autostate, a VLAN interface will only be up/up if the matching VLAN (with the same VLAN number) exists on the switch, is not shut down, and at least one port is up and active in that VLAN. For one correct answer, if the **no vlan 2** command were issued, deleting VLAN 2, the switch would move interface VLAN 2 to a up/down state so it could no longer route packets. For the other correct answer, disabling VLAN 2 with the **shutdown** command in VLAN configuration mode has the same result.

   As for the incorrect answers, when using autostate, a Layer 3 switch needs only one access port or trunk port forwarding for a VLAN to enable routing for that VLAN, so nine of the ten access ports in VLAN 2 could fail, leaving one working port, and the switch would keep routing for VLAN 2.

   A **shutdown** of VLAN 4 does not affect routing for VLAN interfaces 2 and 3. Had that answer listed VLAN 2 or 3, it would be a reason to make routing fail for that VLAN interface.

5. A and C. With a Layer 3 EtherChannel, the physical ports and the port-channel interface must disable the behavior of acting like a switch port and therefore act like a routed port, through the configuration of the **no switchport** interface subcommand. (The **routedport** command is not an IOS command.) Once created, the physical interfaces should not have an IP address configured. The port-channel interface (the interface representing the EtherChannel) should be configured with the IP address.

6. B and C. With a Layer 3 EtherChannel, two configuration settings must be the same on all the physical ports, specifically the speed and duplex as set with the **speed** and **duplex** commands. Additionally, the physical ports and port-channel port must all have the **no switchport** command configured to make each act as a routed port. So, having a different speed setting, or being configured with **switchport** rather than **no switchport**, would prevent IOS from adding interface G0/2 to the Layer 3 EtherChannel.

   As for the wrong answers, both would cause an issue adding the port to a Layer 2 EtherChannel but do not cause a problem with a Layer 3 EtherChannel. Once Layer 2 operations have been disabled because of the **no switchport** command, those settings do not then cause problems for the Layer 3 EtherChannel. So, Layer 2 settings about access VLANs, trunking allowed lists, and STP settings, which must match before an interface can be added to a Layer 2 EtherChannel, do not matter for a Layer 3 EtherChannel.

7. A. On a router that has some routed ports, plus some switched ports, IOS supports LAN switching subcommands on the switched ports only. So, when in interface configuration mode for one of a router's switched interfaces, IOS accepts the **switchport access** command but not the **ip address** command. The router supports the **description** subcommand on both switched and routed ports, making that answer incorrect. Finally, one answer lists a global command (**hostname**), making that answer incorrect because the question asks for interface subcommands.

Chapter 19

1. B and D. The client sends a Discover message, with the server returning an Offer message. The client then sends a Request, with the server sending back the IP address in the Acknowledgment message.
2. A and B. The two correct answers list the two primary facts that impact which IP addresses the server will lease to clients. For the incorrect answer about DNS servers, the DHCP server does supply the IP address of the DNS servers, but not the hostnames of the DNS servers. Also, the DHCP server supplies the IP address (but not the MAC address) of the default gateway in each subnet.
3. A and C. A router needs to act as a DHCP relay agent if DHCP clients exist on the connected subnet and there is no DHCP server in that subnet. If a DHCP server exists in the subnet, the router does not need to forward DHCP messages to a remote DHCP server (which is the function of a DHCP relay agent). The answer that mentions the **ip address dhcp** command makes the router interface act as a DHCP client and has nothing to do with DHCP relay agent.
4. D. The **ip address dhcp** command tells the router to obtain its address using DHCP. The router learns all the same information that a normal DHCP client would learn. The router uses the address listed as the default gateway to build a default route, using the default gateway IP address as the next-hop address. The router continues to work like a router always does, forwarding packets based on its IP routing table.
5. B and C. The output shows the MAC address, IP address, subnet mask (in hex format), and the subnet broadcast address. Of those, the DHCP server supplies the information in the two correct answers. The two incorrect answers mention the MAC address (not supplied by DHCP, but known to the device's NIC) and the subnet broadcast address (calculated by the host).
6. D. Windows supports both **ipconfig** and **ipconfig /all** commands, but the **ipconfig** command does not mention the DNS servers. Note that the **ifconfig** command works on Linux and macOS but not Windows, and the **ifconfig /all** command is an invalid command on all three.

Chapter 20

There are no questions for this chapter.

Chapter 21

1. D. Both versions of RIP use distance vector logic, and EIGRP uses a different kind of logic, characterized either as advanced distance vector or a balanced hybrid.
2. C and D. Both versions of RIP use the same hop-count metric, neither of which is affected by link bandwidth. EIGRP’s metric, by default, is calculated based on bandwidth and delay. OSPF’s metric is a sum of outgoing interfaces costs, with those costs (by default) based on interface bandwidth.
3. B, C, and D. Of the listed routing protocols, only the old RIP Version 1 (RIP-1) protocol does not support variable-length subnet masks (VLSM).
4. C. LSAs contain topology information that is useful in calculating routes, but the LSAs do not directly list the route that a router should add to its routing table. In this case, R1 would run a calculation called the Shortest Path First (SPF) algorithm, against the LSAs, to determine what IP routes to add to the IP routing table.
5. B. Neighboring OSPF routers that complete the database exchange are considered fully adjacent and rest in a full neighbor state. The up/up and final states are not OSPF states at all. The 2-way state is either an interim state or a stable state between some routers on the same VLAN.
6. C. The correct answer is the one advantage of using a single-area design. The three wrong answers are advantages of using a multiarea design, with all reasons being much more important with a larger internetwork.

Chapter 22

1. B. The **network 10.0.0.0 0.255.255.255 area 0** command matches all three interface IP addresses because it compares the first octet only (10) and matches in each case.

   The three incorrect answers do not match all three interface IP addresses because they each compare at least one octet that does not match the address in the **network** command:

   network 10.0.0.0 0.0.0.0 requires an exact match of all four octets (10.0.0.0), which matches no interfaces.

   network 10.0.0.0 0.0.0.255 requires an exact match of the first three octets (10.0.0), which matches none of the interface IP addresses.

   network 10.0.0.0 0.0.255.255 requires an exact match of the first two octets (10.0), which matches none of the interface IP addresses.
2. A. The **network 10.1.0.0 0.0.255.255 area 0** command matches all IP addresses that begin with 10.1, enabling OSPF in area 0 on all interfaces. The three incorrect answers do not match all three interface IP addresses because they each compare at least one octet that does not match the address in the **network** command:

   network 10.0.0.0 0.255.255.0 ignores the middle two octets but compares the first (10) and last (0) octets to the interface addresses. The first octet matches, but the fourth octet matches none of the addresses.

network 10.1.1.0 0.x.1x.0 does not meet syntax requirements because of the letters (x) in the wildcard mask. It would be rejected when attempted in configuration mode.

network 10.1.1.0 255.0.0.0 ignores the first octet but compares the last three octets (1.1.0) to the addresses. None of the addresses end in 1.1.0, so no addresses match this command.

3. A and E. Of the three wrong answers, two are real commands that simply do not list the OSPF neighbors. **show ip ospf interface brief** lists interfaces on which OSPF is enabled but does not list neighbors. **show ip interface** lists IPv4 details about interfaces, but none related to OSPF. One incorrect answer, **show ip neighbor**, is not a valid IOS command.
4. B. The rule for choosing the OSPF RID begins with the **router-id** command in the OSPF process configuration, but the router had no such command. The next rule considers all working (up/up) loopback interfaces, and among those, OSPF chooses the numerically highest IP address. In this case, two such loopback interfaces exist, with loopback 1, with address 10.8.8.8, having the numerically highest IP address.
5. B. With OSPFv2 interface configuration mode, the configuration looks just like the traditional configuration, with a couple of exceptions. The **network** router subcommand is no longer required. Instead, each interface on which OSPF should be enabled is configured with an **ip ospf** *process-id* **area** *area-id* interface subcommand. This command refers to the OSPF routing process that should be enabled on the interface and specifies the OSPFv2 area.
6. A and D. Many of the **show** commands for OSPF do not happen to note whether OSPF happens to be enabled due to an interface subcommand the (**ip ospf** interface subcommand) or a router subcommand (the **network** command). The **show ip protocols** command lists all interfaces on which OSPF has been enabled using the **ip ospf interface** subcommand under the heading "Routing on Interfaces Configured Explicitly." Additionally, the **show ip ospf interface** command, which lists many lines of output per interface, lists the phrase "Attached via Interface Enable." Also, although not in the answers, you can also look at the configuration with the **show running-config** or **show startup-config** command.

Chapter 23

1. B and D. By default, IOS assigns Ethernet interfaces an OSPF network type of broadcast, with an OSPF interface priority of 1. As a result, both routers attempt to discover the other routers on the link (which identifies one correct answer).

   The broadcast network type means that the routers also attempt to elect a DR and BDR. With a priority tied, the routers choose the DR based on the highest router ID (RID) values, meaning that R2 will become the DR and R1 will become the BDR. These facts show why the two incorrect answers are incorrect. The other correct answer is correct because the **show ip ospf neighbor** command lists the local router's neighbor relationship state (FULL) and the role filled by that neighbor (DR), which would be the output shown on R1 when R2 is acting as DR.

2. B and C. First, the OSPF point-to-point network type causes the two routers to dynamically discover neighbors, making one answer correct.

   Next, IOS assigns a default OSPF interface priority of 1, so R1’s configured priority of 11 would be better in a DR/BDR election. However, the point-to-point network type causes the router to not use a DR/BDR on the interface. As a result, the answer about R1 becoming the DR is incorrect (because no DR exists at all), and the answer listing a state of “FULL/DR” is incorrect for the same reason. However, the answer that claims that R2 will be neither DR nor BDR is true because no DR or BDR is elected.

3. D. The **show ip ospf interface brief** command lists a pair of counters under the heading “Nbrs F/C” on the far right of the output. The first of the two numbers represents the number of fully adjacent neighbors (2 in this case), and the second number represents the total number of neighbors.

4. B. The default OSPF priority setting is 1. Once configured with 100, R2 has a higher priority. However, the routers only use the priority values when electing a new DR, so as long as the neighbor relationship is stable, no new DR election will occur. So, any change to make R2 (with higher priority) the DR occurs only after a failure that breaks the current neighbor relationship.Two of the answers refer to other timing as to when R2 becomes the DR. Another distractor states that R2 will cease to serve as BDR, which is not the case.

5. B. SPF calculates the cost of a route as the sum of the OSPF interface costs for all outgoing interfaces in the route. The interface cost can be set directly (**ip ospf cost**), or IOS uses a default based on the reference bandwidth and the interface bandwidth. Of the listed answers, **delay** is the only setting that does not influence OSPFv2 metric calculations.

6. D. The configuration of the interface subcommand **ip ospf hello-interval 15** sets the Hello interval to 15. Also, without any explicit configuration of a Dead interval, IOS also sets the Dead interval to 4X the Hello interval or 60 in this case. The question stem describes the timing and purpose of the Dead interval: how long to wait after not receiving any more Hellos before believing the neighbor has failed.

Chapter 24

1. A and D. As worded, the correct answers list a scenario that would prevent the neighbor relationship. One correct answer mentions the use of two different OSPF areas on the potential OSPF neighbors; to become neighbors, the two routers must use the same area number. The other correct answer mentions the use of two different Hello timers, a mismatch that causes two routers to reject each other and to not become neighbors.

   The two incorrect answers list scenarios that do not cause issues, making them incorrect answers. One mentions mismatched OSPF process IDs; OSPF process IDs do not need to match for two routers to become neighbors. The other incorrect answer (that is, a scenario that does not cause a problem) mentions the use of two different priority values. The priority values give OSPF a means to prefer one router over the other when electing a DR/BDR, so the setting is intended to be set to different values on different routers and does not cause a problem.

2. C. As worded, the correct answers should be a scenario that would prevent the neighbor relationship. The answers all list values that are identical or similar on the two routers. Of those, the use of an identical OSPF Router ID (RID) on the two routers prevents them from becoming neighbors, making that one answer correct.

   Of the incorrect answers, both routers must have the same Dead interval, so both using a Dead interval of 40 causes no issues. The two routers can use any OSPF process ID (the same or different value, it does not matter), making that answer incorrect. Finally, the two routers' IP addresses must be in the same subnet, so again that scenario does not prevent R13 and R14 from becoming neighbors.
3. D. The OSPF **shutdown** command tells the OSPF process to stop operating. That process includes removing any OSPF-learned routes from the IP routing table, clearing the router's LSDB, and closing existing OSPF neighbor relationships. In effect, it causes OSPF to stop working on the router, but it does retain the configuration so that a **no shutdown** command in OSPF configuration mode will cause the router to start using OSPF again with no changes to the configuration.
4. B. OSPF uses an equal-cost multipath feature, in which when it calculates multiple routes for the same subnet that tie with the lowest metric, the router places multiple routes into the routing table. IOS limits the number of such routes for one destination subnet per the **maximum-paths** setting on the router, which typically defaults to 6. The router would not use the route with metric 15001, as it is worse than the other two routes' metric of 15000.
5. D. Within a routing protocol, the routing protocol will choose the best route based on the metric. As a result, OSPF picks the metric 1000 route while EIGRP chooses its metric 1,000,000 route. Then the router must choose between the two routing protocol sources using the administrative distance. With default settings, EIGRP has a better administrative distance of 90 versus OSPF's 110. As a result, the router places the best EIGRP route into its routing table, the route learned by EIGRP with metric 1,000,000.
6. D. Each route defines a range of IP addresses as follows:
   - 172.20.90.9/32: 172.20.90.9 only
   - 172.20.88.0/23: 172.20.88.0–172.20.89.255
   - 172.20.80.0/20: 172.20.80.0–172.20.95.255
   - 172.20.0.0/16: 172.20.0.0–172.20.255.255
   - 0.0.0.0/0: 0.0.0.0–255.255.255.255

   Given those ranges, a packet destined for address 172.20.89.100 matches all but the first route in the list.
7. C. Each route defines a range of IP addresses, as follows:
   - 172.20.90.9/32: 172.20.90.9 only
   - 172.20.88.0/23: 172.20.88.0–172.20.89.255
   - 172.20.80.0/20: 172.20.80.0–172.20.95.255

- 172.20.0.0/16: 172.20.0.0–172.20.255.255
- 0.0.0.0/0: 0.0.0.0–255.255.255.255

Given those ranges, a packet destined for address 172.20.90.1 matches the last three routes in the list. Among those, the router will use the most specific route, the route with the largest number of prefix bits. As a result, the router uses the route with prefix length /20, which has a next-hop address of 172.20.13.3.

Chapter 25

1. C. NAT, specifically the PAT feature that allows many hosts to use private IPv4 addresses while being supported by a single public IPv4 address, was one short-term solution to the IPv4 address exhaustion problem. IP version 5 existed briefly as an experimental protocol and had nothing to do with IPv4 address exhaustion. IPv6 directly addresses the IPv4 address exhaustion problem, but it is a long-term solution. ARP has no impact on the number of IPv4 addresses used.
2. A. Routers use the same process steps when routing IPv6 packets as they do when routing IPv4 packets. Routers route IPv6 packets based on the IPv6 addresses listed inside the IPv6 header by comparing the destination IPv6 address to the router's IPv6 routing table. As a result, the router discards the incoming frame's data-link header and trailer, leaving an IPv6 packet. The router compares the destination (not source) IPv6 address in the header to the router's IPv6 (not IPv4) routing table and then forwards the packet based on the matched route.
3. D. If you are following the steps in the book, the first step removes up to three leading 0s in each quartet, leaving FE80:0:0:0:100:0:0:123. This value leaves two strings of consecutive all-0 quartets; when you change the longest string of all 0s to ::, the address is FE80::100:0:0:123.
4. B. This question has many quartets that make it easy to make a common mistake: removing trailing 0s in a quartet of hex digits. Only leading 0s in a quartet and not trailing 0s should be removed. Many of the quartets have trailing 0s (0s on the right side of the quartet), so make sure not to remove those 0s.
5. A. The unabbreviated version of an IPv6 address must have 32 digits, and only one answer has 32 hex digits. In this case, the original number shows four quartets and a ::. So, the :: was replaced with four quartets of 0000, making the number eight. Then, for each quartet with fewer than four digits, leading 0s were added, so each quartet has four hex digits.
6. C. The /64 prefix length means that the last 64 bits, or last 16 digits, of the address should be changed to all 0s. That process leaves the unabbreviated subnet prefix as 2000:0000:0000:0005:0000:0000:0000:0000. The last four quartets are all 0s, making that string of all 0s be the longest and best string of 0s to replace with ::. After removing the leading 0s in other quartets, the answer is 2000:0:0:5::/64.

Chapter 26

1. C. Unique local addresses begin with FD in the first two digits.
2. A. Global unicast addresses begin with a hex 2 or 3.

3. D. The global routing prefix defines the address block, represented as a prefix value and prefix length, assigned to an organization by some numbering authority. The global routing prefix acts as the initial part of IPv6 addresses within the company for the number of bits defined by the prefix length. Similarly, when a company uses a public IPv4 address block, all the addresses have the same value in the network part, which also acts as the initial part of IPv4 addresses.
4. B. Subnetting a global unicast address block, using a single prefix length for all subnets, breaks the addresses into three parts. The parts are the global routing prefix, subnet ID, and interface ID.
5. D. Unique local addresses begin with a 2-hex-digit prefix of FD, followed by the 10-hex-digit global ID.

Chapter 27

1. A. The one correct answer lists the exact same IPv6 address listed in the question, with a /64 prefix length and no spaces in the syntax of the answer. Another (incorrect) answer is identical, except that it leaves a space between the address and prefix length, which is incorrect syntax. The two answers that list the **eui-64** parameter list an address and not a prefix; they should list a prefix to be correct. However, even if these two incorrect answers had listed the prefix of the address shown (2001:1:1:1::), the EUI-64 process would not have resulted in the IPv6 address listed in the question.
2. B. With the **eui-64** parameter, the router will calculate the interface ID portion of the IPv6 address based on its MAC address. Beginning with 5055.4444.3333, the router injects FF FE in the middle (5055.44FF.FE44.3333). Then the router inverts the seventh bit in the first byte. To see the change, hex 50 to binary 0101 0000. Then change bit 7, so the string becomes 0101 0010, which converts back to hex 52. The final interface ID value is 5255:44FF:FE44:3333. The wrong answers simply list a different value.
3. A and C. Of the four answers, the two correct answers show the minimal required configuration to support IPv6 on a Cisco router: enabling IPv6 routing (**ipv6 unicast-routing**) and enabling IPv6 on each interface, typically by adding a unicast address to each interface (**ipv6 address…**). The two incorrect answers list nonexistent commands.
4. B and D. The **show ipv6 route connected** command lists all known connected routes, with each route listing the prefix/length of the route. The **show ipv6 interface g0/0/0** command lists the interface address and the prefix/length calculated from the configured address/length.

   Of the incorrect answers, the **show ipv6 interface brief** command lists the interface address but not the prefix/length of the connected subnet. The **show ipv6 address** command does not exist, but is simply rejected as an invalid command if attempted.
5. A. With an **ipv6 address** command configured for a global unicast address but without a link-local address configured with an **ipv6 address** command, the router calculates its link-local address on the interface based on its MAC address and EUI-64 rules. The router does not use the global unicast IPv6 address to calculate the link-local address.

The first half of the link-local address begins FE80:0000:0000:0000. The router then calculates the second half of the link-local address value by taking the MAC address (0200.0001.000A), injecting FF FE in the middle (0200.00FF.FE01.000A), and flipping the seventh bit (0000.00FF.FE01.000A).

6. B. FF02::1 is used by all IPv6 hosts on the link, FF02::5 is used by all OSPFv3 routers, and FF02::A is used by all EIGRPv6 routers. FF02::2 is used to send packets to all IPv6 routers on a link.
7. A. The router sends the NDP NS message to the solicited-node multicast address based on the unicast address of 2001:db8:1:1::1234:5678. To create the correct solicited-node address, take the last six hex digits (34:5678 in this case), and prepend FF02::1:FF. The correct answer is FF02::1:FF34:5678. The other answers are similar values that do not follow the correct solicited-node rules.
8. B and C. First, for G0/0/1, with the **ipv6 enable** command, the router enables IPv6, creating an LLA using EUI-64 rules for the interface ID. Those facts identify one correct and one incorrect answer.

   Then, for the answer interface G0/0/2 and the **ipv6 autoconfig** command, the command enables IPv6 with SLAAC. As a result, it generates an LLA, using EUI-64 rules, and generates a routable unicast address using SLAAC, again using EUI-64 rules for the interface ID. As a result, G0/0/2’s LLA and global unicast address use the same interface ID values.

   Finally, for the answer about interface G0/0/3 and the **ipv6 address** subcommand, every interface that supports IPv6 must have an LLA. The router will again use EUI-64 to self-assign the interface ID portion of the interface’s LLA.

C

Chapter 28

1. B. PC1 needs to discover PC2’s MAC address. Unlike IPv4, IPv6 does not use ARP, instead using NDP. Specifically, PC1 uses the NDP Neighbor Solicitation (NS) message to request that PC2 send back an NDP Neighbor Advertisement (NA). SLAAC relates to address assignment and not to discovering a neighbor’s MAC address.
2. A and C. The NDP RA lists the router IPv6 address, the IPv6 prefixes known on the link, and the matching prefix lengths. The incorrect answers happen to list facts not included in the NDP RA message.
3. A. The **show ipv6 neighbors** command lists all IPv6 addresses of neighbors (both routers and hosts), plus their matching MAC addresses. It does not note which are routers, leaving that information for the **show ipv6 routers** command.
4. D. For the one correct answer, hosts can ask for (solicit) all routers to identify themselves by sending an NDP Router Solicitation (RS) message, with the routers sending back an NDP Router Advertisement (RA) message. For the incorrect answers, PC1 uses NDP Neighbor Solicitation (NS) but not for learning its default router IPv6 address. DAD is a function that uses NDP NS and NA messages, but its function does not include the discovery of the default router address. Finally, EUI-64 does not define a protocol or message, but is rather a convention to define 64-bit values to use as an IPv6 IID.

5. D. SLAAC gives the host a means to choose its unicast address. The host also uses NDP to learn its prefix length, plus the address(es) of any default routers. It then uses stateless DHCP to learn the addresses of the DNS server(s).

6. B and D. With SLAAC, the host learns the subnet prefix from a router using NDP RS/RA messages, and then the host builds the rest of the address (the interface ID). The host can randomly generate the interface ID or use modified EUI-64 rules. The host does not learn the interface ID from any other device, which helps make the process stateless because no other device needs to assign the host its full address.

7. A. The DHCPv6 protocol uses well-known multicast addresses, specifically FF02::1:2, for messages directed to DHCPv6 servers. However, because this multicast address has a link-local scope, those messages remain on the local LAN. A router connected to the LAN must implement a DHCPv6 relay agent function so that the router will replace the packet's FF02::1:2 multicast destination address with the unicast address of the DHCPv6 server. The routers then use normal unicast routing to forward the packet.

   For the two other incorrect answers, note that IPv6 does not use broadcast addresses at all. For instance, the all F's address in the answer is not an IPv6 broadcast address because there is no such thing in IPv6. Also, there is no mechanism to learn a DHCPv6 server's unicast address using NDP.

8. C. IPv6 routes on hosts and routers typically use the LLA of the next-hop device. For instance, PC1's default route would reference a router's LLA, not GUA. The **traceroute** command relies on those routes. However, the NDP messages that help the **traceroute** command identify how each router identifies the routable unicast address like the global unicast address. So, the **traceroute** command lists only GUAs in its output. Those facts determine the correct answer and rule out two answers as incorrect. One answer mentions the last line of **traceroute** output; because the command succeeded, that line lists the IPv6 address of the destination host rather than the address of the last router.

Chapter 29

1. A and C. With an IPv6 address on a working interface, the router adds a connected route for the prefix (subnet) implied by the **ipv6 address** command. It also adds a local route (with a /128 prefix length) based on the unicast address. The router does not add a route based on the link-local address.

2. A and C. The two correct answers show the correct subnet ID (subnet prefix) and prefix length for the two connected subnets: 3111:1:1:1::/64 and 3222:2:2:2::/64. The answer with the /128 prefix length exists in a local route, but the **show ipv6 route connected** command does not list local routes. The other incorrect answer lists the entire IPv6 address with a /64 prefix length rather than the prefix ID.

3. A and B. All the answers list the same destination subnet prefix (2000:1:2:3::/64), which is the subnet prefix on the LAN to the right of Router R1. The differences exist in the forwarding instructions in each route.

   For the two commands that list both the outgoing interface (G0/0/1) and the next-hop address, both refer to the correct outgoing interface on Router R5. One refers to the incorrect next-hop address—R1's own global unicast address (GUA), whereas the correct command lists neighboring Router R6's GUA (which ends in :6).

For the incorrect answer that lists only an outgoing interface, it lists the correct interface, and the router adds it to its routing table, but the route does not work. IPv6 static routes that refer to an outgoing Ethernet interface must also list a next-hop address for the router to know enough information to forward packets.

For the correct answer that lists only a next-hop GUA, it lists the correct GUA: R6's GUA on the link between R5 and R6 (which ends in :6).

4. B. All four answers show examples of commands that use a next-hop router IPv6 address.

   Two incorrect answers list a next-hop address for R5's WAN interface (one global unicast, one link-local). A correct next-hop address reference on Router R1 should refer to an address on Router R6 instead.

   For the two answers that list addresses on Router R6, the one that lists R6's global unicast address (2001:1:2:56::6) is correct. The command that lists R6's link-local address also requires R5's outgoing interface, so the router would reject the command in the answer that lists FE80::FF:FE00:6.

5. B. The **show ipv6 route** command, unlike the **show ip route** command, does not designate a gateway of last resort. Instead, it lists the default route like the other IPv6 routes, but with the special prefix/length of ::/0, which matches all possible IPv6 addresses.

   For the other incorrect answers, the prefix of ::/128 would match the host address of all 0s, rather than matching all addresses. A route that matches prefix 2000::/3 will match all global unicast addresses, but it does not match all IPv6 addresses, so it would not be a default route.

6. B. The **ipv6 route** command in the question uses correct syntax, so the router will at least accept the command into the configuration. Of note, the command uses the administrative distance (AD) setting at the end, with a value of 200. As a result, Router R1 treats this route as a floating static route because its AD value (200) is greater than the default OSPF AD (110). As such, R1 continues to use the better OSPF-learned route based on the better (lower) AD and does not add the static route to the routing table.

7. C. The question asks what could have caused the conditions in the question. The user typed the command and pressed Enter, but the question did not say whether the router accepted the command. The question also tells us that the IPv6 routing table lists no routes for prefix/length (2001:DB8:8:8::/64). The goal then is to consider the answers to determine if any of those answers could result in no routes appearing for this prefix.

   IOS will add a new static route to the IPv6 routing table if, when using a next-hop global unicast address, the router has a working route to reach that next-hop address and there is no better (lower administrative distance) route for the exact same subnet. So, the correct answer identifies one reason why the route would not appear.

The answer that mentions a better route with administrative distance of 110 is a valid reason for the static route not to appear. Still, the question states that no route for the subnet appears in the routing table, so clearly that competing route does not exist.

The other two incorrect answers mention the **ipv6 route** command. This command can use a link-local next-hop address but does not have to do so, showing the incorrect claim on one of those answers. For the other answer, when using a global unicast address as next-hop, the command does not also require an outgoing interface parameter, showing that answer as incorrect.

This page intentionally left blank

This page intentionally left blank

GLOSSARY

NUMERIC

10/100 A short reference to an Ethernet NIC or switch port that supports speed of 10 Mbps and 100 Mbps.

10/100/1000 A short reference to an Ethernet NIC or switch port that supports speeds of 10 Mbps, 100 Mbps, and 1000 Mbps (that is, 1 Gbps).

10BASE-T The 10-Mbps baseband Ethernet specification using two pairs of twisted-pair cabling (Categories 3, 4, or 5): one pair transmits data and the other receives data. 10BASE-T, which is part of the IEEE 802.3 specification, has a distance limit of approximately 100 m (328 feet) per segment.

100BASE-T A name for the IEEE Fast Ethernet standard that uses two-pair copper cabling, a speed of 100 Mbps, and a maximum cable length of 100 meters.

1000BASE-T A name for the IEEE Gigabit Ethernet standard that uses four-pair copper cabling, a speed of 1000 Mbps (1 Gbps), and a maximum cable length of 100 meters.

2-way state In OSPF, a neighbor state that implies that the router has exchanged Hellos with the neighbor and that all required parameters match.

802.1Q 802.1Q is the standard protocol for this tag. The most critical piece of information (for this discussion) in this tag is the VLAN ID.

A

AAA Authentication, authorization, and accounting. Authentication confirms the identity of the user or device. Authorization determines what the user or device is allowed to do. Accounting records information about access attempts, including inappropriate requests.

AAA server A server that holds security information and provides services related to user login, particularly authentication (is the user who he says he is?), authorization (once authenticated, what do we allow the user to do?), and accounting (tracking the user).

ABR *See* Area Border Router.

access interface A LAN network design term that refers to a switch interface connected to end-user devices, configured so that it does not use VLAN trunking.

access layer In a campus LAN design, the switches that connect directly to endpoint devices (servers, user devices), and also connect into the distribution layer switches.

access link In Frame Relay, the physical serial link that connects a Frame Relay DTE device, usually a router, to a Frame Relay switch. The access link uses the same physical layer standards as do point-to-point leased lines.

access point (AP) A device that provides wireless service for clients within its coverage area or cell, with the AP connecting to both the wireless LAN and the wired Ethernet LAN.

accounting In security, the recording of access attempts. *See also* AAA.

address block A set of consecutive IPv4 addresses. The term is most often used for a classless prefix as defined by CIDR but can also refer to any subnet or IPv4 network.

adjacent-layer interaction The general topic of how, on one computer, two adjacent layers in a networking architectural model work together, with the lower layer providing services to the higher layer.

administrative distance In Cisco routers, a means for one router to choose between multiple routes to reach the same subnet when those routes were learned by different routing protocols. The lower the administrative distance, the better the source of the routing information.

all-nodes multicast address A specific IPv6 multicast address, FF02::1, with link-local scope, used to send packets to all devices on the link that support IPv6.

all-routers multicast address A specific IPv6 multicast address, FF02::2, with link-local scope, used to send packets to all devices that act as IPv6 routers on the local link.

alternate port role With RSTP, a port role in which the port acts as an alternative to a switch's root port, so that when the switch's root port fails, the alternate port can immediately take over as the root port.

anycast address An address shared by two or more hosts that exist in different parts of the network, so that by design, the routers will forward packets to the nearest of the two servers, allowing clients to communicate with the nearest such server, not caring which particular server with which the client communicates.

APIPA Automatic Private IP Addressing. A convention per RFC 3927 for a process and reserved set of IPv4 addresses (169.254.0.0/16) that hosts use when they need to use an IPv4 address but they fail to lease an IPv4 address using dynamic processes like DHCP.

Area Border Router (ABR) A router using OSPF in which the router has interfaces in multiple OSPF areas.

ARP Address Resolution Protocol. An Internet protocol used to map an IP address to a MAC address. Defined in RFC 826.

ARP table A list of IP addresses of neighbors on the same VLAN, along with their MAC addresses, as kept in memory by hosts and routers.

ARPANET The first packet-switched network, first created around 1970, which served as the predecessor to the Internet.

ASBR Autonomous System Border Router. A router using OSPF in which the router learns routes via another source, usually another routing protocol, exchanging routes that are external to OSPF with the OSPF domain.

authentication In security, the verification of the identity of a person or a process. *See also* AAA.

authentication server (AS) An 802.1x entity that authenticates users or clients based on their credentials, as matched against a user database. In a wireless network, a RADIUS server is an AS.

authenticator An 802.1x entity that exists as a network device that provides access to the network. In a wireless network, a WLC acts as an authenticator.

authorization In security, the determination of the rights allowed for a particular user or device. *See also* AAA.

auto-MDIX An Ethernet standard feature, introduced at the same time as Gigabit Ethernet in 1998, that senses whether the link uses the correct UTP cable pinout (straight-through or crossover), and automatically swaps the signals internally to make the link work if the wrong UTP cable pinout is used.

autonegotiation An IEEE standard mechanism (802.3u) with which two nodes can exchange messages for the purpose of choosing to use the same Ethernet standards on both ends of the link, ensuring that the link functions and functions well.

autonomous system An internetwork in the administrative control of one organization, company, or governmental agency, inside which that organization typically runs an interior gateway protocol (IGP).

auxiliary port A physical connector on a router that is designed to be used to allow a remote terminal, or PC with a terminal emulator, to access a router using an analog modem.

B

backbone area In OSPFv2 and OSPFv3, the special area in a multiarea design, with all nonbackbone areas needing to connect to the backbone area, area 0.

back-to-back link A serial link between two routers, created without CSU/DSUs, by connecting a DTE cable to one router and a DCE cable to the other. Typically used in labs to build serial links without the expense of an actual leased line from the telco.

backup designated router An OSPF router connected to a multiaccess network that monitors the work of the designated router (DR) and takes over the work of the DR if the DR fails.

backup port role With RSTP, a port role in which the port acts as a backup to one of the switch's ports acting as a designated port. If the switch's designated port fails, the switch will use the backup port to immediately take over as the designated port.

bandwidth A reference to the speed of a networking link. Its origins come from earlier communications technology in which the range, or width, of the frequency band dictated how fast communications could occur.

binary mask An IPv4 subnet mask written as a 32-bit binary number.

bitwise Boolean AND A Boolean AND between two numbers of the same length in which the first bit in each number is ANDed, and then the second bit in each number, and then the third, and so on.

blocking state In STP, a port state in which no received frames are processed and the switch forwards no frames out the interface, with the exception of STP messages.

Boolean AND A math operation performed on a pair of one-digit binary numbers. The result is another one-digit binary number. 1 AND 1 yields 1; all other combinations yield a 0.

BPDU Bridge protocol data unit. The generic name for Spanning Tree Protocol messages.

BPDU Guard A Cisco switch feature that listens for incoming STP BPDU messages, disabling the interface if any are received. The goal is to prevent loops when a switch connects to a port expected to only have a host connected to it.

BPDU Filter A Cisco switch feature that uses the monitoring and filtering (discarding) of STP BPDUs to achieve goals, such as protecting against forwarding loops on PortFast ports and disabling STP by filtering all BPDU messages.

bridge ID (BID) An 8-byte identifier for bridges and switches used by STP and RSTP. It is composed of a 2-byte priority field followed by a 6-byte System ID field that is usually filled with a MAC address.

bridge protocol data unit *See* BPDU.

broadcast address Generally, any address that represents all devices, and can be used to send one message to all devices. In Ethernet, the MAC address of all binary 1s, or FFFF.FFFF.FFFF in hex. For IPv4, *see* subnet broadcast address.

broadcast domain A set of all devices that receive broadcast frames originating from any device within the set. Devices in the same VLAN are in the same broadcast domain.

broadcast frame An Ethernet frame sent to destination address FFFF.FFFF.FFFF, meaning that the frame should be delivered to all hosts on that LAN.

broadcast network type An OSPF interface setting, useful on links with more than two routers, resulting in an LSA flooding process managed by an elected designated router (DR).

broadcast subnet When subnetting a Class A, B, or C network, the one subnet in each classful network for which all subnet bits have a value of binary 1. The subnet broadcast address in this subnet has the same numeric value as the classful network's networkwide broadcast address.

broken (state) An STP port state on Cisco switches, used by Root Guard and Loop Guard as a method for STP to disable the use of the port. A port in the broken state does not forward or process received frames.

C

Channel-group One term Cisco switches use to reference a bundle of links that are, in some respects, treated like a single link. Other similar terms include *EtherChannel* and *PortChannel*.

CIDR Classless interdomain routing. An RFC-standard tool for global IP address range assignment. CIDR reduces the size of Internet routers' IP routing tables, helping deal with the rapid growth of the Internet. The term *classless* refers to the fact that the summarized groups of networks represent a group of addresses that do not conform to IPv4 classful (Class A, B, and C) grouping rules.

CIDR Block A set of consecutive public IPv4 addresses whose size can be any power of 2. Used as an alternative to the original process with public IP networks of three set sizes.

CIDR mask Another term for a prefix mask, one that uses prefix or CIDR notation, in which the mask is represented by a slash (/) followed by a decimal number.

CIDR notation *See* prefix notation.

Cisco Catalyst Edge Platform The brand name created by Cisco for a product family. The products primarily act as routers, but they also create a platform to run many other networking services, including SD-WAN, SASE, and cloud features.

cladding In fiber-optic cabling, the second layer of the cable, surrounding the core of the cable, with the property of reflecting light back into the core.

classful addressing A concept in IPv4 addressing that defines a subnetted IP address as having three parts: network, subnet, and host.

classful IP network An IPv4 Class A, B, or C network; called a classful network because these networks are defined by the class rules for IPv4 addressing.

classful routing protocol Does not transmit the mask information along with the subnet number and therefore must consider Class A, B, and C network boundaries and perform autosummarization at those boundaries. Does not support VLSM.

classless addressing A concept in IPv4 addressing that defines a subnetted IP address as having two parts: a prefix (or subnet) and a host.

classless interdomain routing The name of an RFC that defines several important features related to public IPv4 addressing: a global address assignment strategy to keep the size of IPv4 routing tables smaller, and the ability to assign public IPv4 addresses in sizes based on any prefix length.

classless prefix A range of public IPv4 addresses as defined by CIDR.

classless prefix length The mask (prefix length) used when defining a classless prefix.

classless routing protocol An inherent characteristic of a routing protocol, specifically that the routing protocol does send subnet masks in its routing updates, thereby removing any need to make assumptions about the addresses in a particular subnet or network, making it able to support VLSM and manual route summarization.

CLI Command-line interface. An interface that enables the user to interact with the operating system by entering commands and optional arguments.

collision domain A set of network interface cards (NIC) for which a frame sent by one NIC could result in a collision with a frame sent by any other NIC in the same collision domain.

command-line interface *See* CLI.

configuration mode A part of the Cisco IOS Software CLI in which the user can type configuration commands that are then added to the device's currently used configuration file (running-config).

connected The single-item status code listed by a **switch show interfaces status** command, with this status referring to a working interface.

connected route On a router, an IP route added to the routing table when the router interface is both up and has an IP address configured. The route is for the subnet that can be calculated based on the configured IP address and mask.

console port A physical socket on a router or switch to which a cable can be connected between a computer and the router/switch, for the purpose of allowing the computer to use a terminal emulator and use the CLI to configure, verify, and troubleshoot the router/switch.

contiguous network A network topology in which subnets of network X are not separated by subnets of any other classful network.

convergence The time required for routing protocols to react to changes in the network, removing bad routes and adding new, better routes so that the current best routes are in all the routers' routing tables.

core In fiber-optic cabling, the center cylinder of the cable, made of fiberglass, through which light passes.

crossover cable An Ethernet cable that swaps the pair used for transmission on one device to a pair used for receiving on the device on the opposite end of the cable. In 10BASE-T and 100BASE-TX networks, this cable swaps the pair at pins 1,2 to pins 3,6 on the other end of the cable, and the pair at pins 3,6 to pins 1,2 as well.

CSMA/CD Carrier sense multiple access with collision detection. A media-access mechanism in which devices ready to transmit data first check the channel for a carrier. If no carrier is sensed for a specific period of time, a device can transmit. If two devices transmit at once, a collision occurs and is detected by all colliding devices. This collision subsequently delays retransmissions from those devices for some random length of time.

D

data VLAN A VLAN used by typical data devices connected to an Ethernet, like PCs and servers. Used in comparison to a voice VLAN.

Database Description An OSPF packet type that lists brief descriptions of the LSAs in the OSPF LSDB.

DDN *See* dotted-decimal notation.

Dead Interval In OSPF, a timer used for each neighbor. A router considers the neighbor to have failed if no Hellos are received from that neighbor in the time defined by the timer.

decimal mask An IPv4 subnet mask written in dotted-decimal notation; for example, 255.255.255.0.

de-encapsulation On a computer that receives data over a network, the process in which the device interprets the lower-layer headers and, when finished with each header, removes the header, revealing the next-higher-layer PDU.

default gateway/default router On an IP host, the IP address of some router to which the host sends packets when the packet's destination address is on a subnet other than the local subnet.

default mask The mask used in a Class A, B, or C network that does not create any subnets; specifically, mask 255.0.0.0 for Class A networks, 255.255.0.0 for Class B networks, and 255.255.255.0 for Class C networks.

default route On a router, the route that is considered to match all packets that are not otherwise matched by some more specific route.

default VLAN A reference to the default setting of 1 (meaning VLAN ID 1) on the **switchport access vlan** *vlan-id* **interface** subcommand on Cisco switches, meaning that by default, a port will be assigned to VLAN 1 if acting as an access port.

designated port In both STP and RSTP, a port role used to determine which of multiple interfaces on multiple switches, each connected to the same segment or collision domain, should forward frames to the segment. The switch advertising the lowest-cost Hello BPDU onto the segment becomes the DP.

designated router In OSPF, on a multiaccess network, the router that wins an election and is therefore responsible for managing a streamlined process for exchanging OSPF topology information between all routers attached to that network.

DHCP Dynamic Host Configuration Protocol. A protocol used by hosts to dynamically discover and lease an IP address, and learn the correct subnet mask, default gateway, and DNS server IP addresses.

DHCP client Any device that uses DHCP protocols to ask to lease an IP address from a DHCP server, or to learn any IP settings from that server.

DHCP relay agent The name of the router IOS feature that forwards DHCP messages from client to servers by changing the destination IP address from 255.255.255.255 to the IP address of the DHCP server.

DHCP server Software that waits for DHCP clients to request to lease IP addresses, with the server assigning a lease of an IP address as well as listing other important IP settings for the client.

Dijkstra Shortest Path First (SPF) algorithm The name of the algorithm used by link-state routing protocols to analyze the LSDB and find the least-cost routes from that router to each subnet.

directed broadcast address *See* subnet broadcast address.

disabled port role In STP, a port role for nonworking interfaces—in other words, interfaces that are not in a connect or up/up interface state. The reason can be due to administrative setting (shutdown) or interface failure.

disabled state In STP but not RSTP, the state to be used for interfaces in the disabled port role.

discarding state An RSTP interface state, which primarily replaces the STP blocking state, as used for interfaces with port roles other than designated or root ports. In this state, the switch does not forward data frames out the interface, nor does it process received frames other than RSTP messages.

discontiguous network A network topology in which subnets of network X are separated by subnets of some other classful network.

distance vector The logic behind the behavior of some interior routing protocols, such as RIP. Distance vector routing algorithms call for each router to send its entire routing table in each update, but only to its neighbors. Distance vector routing algorithms can be prone to routing loops but are computationally simpler than link-state routing algorithms.

distribution layer In a campus LAN design, the switches that connect to access layer switches as the most efficient means to provide connectivity from the access layer into the other parts of the LAN.

DNS Domain Name System. An application layer protocol used throughout the Internet for translating hostnames into their associated IP addresses.

DNS Reply In the Domain Name System (DNS), a message sent by a DNS server to a DNS client in response to a DNS Request, identifying the IP address assigned to a particular hostname or fully qualified domain name (FQDN).

DNS Request In the Domain Name System (DNS), a message sent by a DNS client to a DNS server, listing a hostname or fully qualified domain name (FQDN), asking the server to discover and reply with the IP address associated with that hostname or FQDN.

DNS server An application acting as a server for the purpose of providing name resolution services per the Domain Name System (DNS) protocol and worldwide system.

DNS server list A list of IP addresses of DNS servers, known to an IP host, used by a host when sending DNS name resolution requests.

dotted-decimal notation (DDN) The format used for IP version 4 addresses, in which four decimal values are used, separated by periods (dots).

dual stack A mode of operation in which a host or router runs both IPv4 and IPv6.

duplex mismatch On opposite ends of any Ethernet link, the condition in which one of the two devices uses full-duplex logic and the other uses half-duplex logic, resulting in unnecessary frame discards and retransmissions on the link.

duplicate address detection (DAD) A term used in IPv6 to refer to how hosts first check whether another host is using a unicast address before the first host uses that address.

E

EIGRP Enhanced Interior Gateway Routing Protocol. An advanced version of IGRP developed by Cisco. Provides superior convergence properties and operating efficiency and combines the advantages of link-state protocols with those of distance vector protocols.

EIGRP version 6 The version of the EIGRP routing protocol that supports IPv6, and not IPv4.

electromagnetic interference (EMI) The name of the effect in which electricity passes through one cable as normal, inducing a magnetic field outside the conductor. That magnetic field, if it passes through another conductor, like a nearby cable, induces new electrical current in the second cable, interfering with the use of electricity to transmit data on the second cable.

enable mode A part of the Cisco IOS CLI in which the user can use the most powerful and potentially disruptive commands on a router or switch, including the ability to then reach configuration mode and reconfigure the router.

encapsulation The placement of data from a higher-layer protocol behind the header (and in some cases, between a header and trailer) of the next-lower-layer protocol. For example, an IP packet could be encapsulated in an Ethernet header and trailer before being sent over an Ethernet.

encryption Applying a specific algorithm to data to alter the appearance of the data, making it incomprehensible to those who are not authorized to see the information.

enterprise router A term to describe the general role of a router as a router at a permanent site owned or leased by the enterprise, like an office building, manufacturing facility, branch office, or retail location. These sites typically have enough users to justify separate routers, switches, and wireless access points, and are more likely to justify private WAN services, in comparison to SOHO routers.

equal-cost multipath (ECMP) A term for a router's forwarding logic when it has more than one route for the same subnet with the same metric (cost).

equal-cost route When a routing protocol computes all possible routes to one subnet, the case for which multiple of those routes have the same metric (cost).

error detection The process of discovering whether a data-link level frame was changed during transmission. This process typically uses a Frame Check Sequence (FCS) field in the data-link trailer.

error disabled An interface state on LAN switches that can be the result of one of many security violations.

error recovery The process of noticing when some transmitted data was not successfully received and resending the data until it is successfully received.

EtherChannel A feature in which up to eight parallel Ethernet segments exist between the same two devices, each using the same speed. May be a Layer 2 EtherChannel, which acts like a single link for forwarding and Spanning Tree Protocol logic, or a Layer 3 EtherChannel, which acts like a single link for the switch's Layer 3 routing logic.

EtherChannel Load Distribution The logic used by switches when forwarding messages over EtherChannels by which the switch chooses the specific physical link out which the switch will forward the frame.

Ethernet A series of LAN standards defined by the IEEE, originally invented by Xerox Corporation and developed jointly by Xerox, Intel, and Digital Equipment Corporation.

Ethernet address A 48-bit (6-byte) binary number, usually written as a 12-digit hexadecimal number, used to identify Ethernet nodes in an Ethernet network. Ethernet frame headers list a destination and source address field, used by the Ethernet devices to deliver Ethernet frames to the correct destination.

Ethernet frame A term referring to an Ethernet data-link header and trailer, plus the data encapsulated between the header and trailer.

Ethernet Line Service (E-Line) A specific carrier/metro Ethernet service defined by MEF (MEF.net) that provides a point-to-point topology between two customer devices, much as if the two devices were connected using an Ethernet crossover cable.

Ethernet link A generic term for any physical link between two Ethernet nodes, no matter what type of cabling is used.

Ethernet over MPLS (EoMPLS) A term referring specifically to how a service provider can create an Ethernet WAN service using an MPLS network. More generally, a term referring to Ethernet WAN services.

Ethernet port A generic term for the opening on the side of any Ethernet node, typically in an Ethernet NIC or LAN switch, into which an Ethernet cable can be connected.

EtherType Jargon that shortens the term *Ethernet Type*, which refers to the Type field in the Ethernet header. The Type field identifies the type of packet encapsulated inside an Ethernet frame.

EUI-64 Literally, a standard for an extended unique identifier that is 64 bits long. Specifically for IPv6, a set of rules for forming a 64-bit identifier, used as the interface ID in IPv6 addresses, by starting with a 48-bit MAC address, inserting FFFE (hex) in the middle, and inverting the seventh bit.

extended ping An IOS command in which the **ping** command accepts many other options besides just the destination IP address.

F

Fast Ethernet The common name for all the IEEE standards that send data at 100 megabits per second.

fiber-optic cable A type of cabling that uses glass fiber as a medium through which to transmit light.

filter Generally, a process or a device that screens network traffic for certain characteristics, such as source address, destination address, or protocol, and determines whether to forward or discard that traffic based on the established criteria.

firewall A device that forwards packets between the less secure and more secure parts of the network, applying rules that determine which packets are allowed to pass and which are not.

flash memory A type of read/write permanent memory that retains its contents even with no power applied to the memory, and uses no moving parts, making the memory less likely to fail over time.

floating static route A static IP route that uses a higher administrative distance than other routes, typically routes learned by a routing protocol. As a result, the router will not use the static route if the routing protocol route has been learned, but then use the static route if the routing protocol fails to learn the route.

flood/flooding The result of the LAN switch forwarding process for broadcasts and unknown unicast frames. Switches forward these frames out all interfaces, except the interface in which the frame arrived. Switches also flood multicasts by default, although this behavior can be changed.

forward To send a frame received in one interface out another interface, toward its ultimate destination.

forward delay An STP timer, defaulting to 15 seconds, used to dictate how long an interface stays in the listening state and the time spent in learning state. Also called the forward delay timer.

forward route From one host's perspective, the route over which a packet travels from that host to some other host.

forwarding state An STP and RSTP port state in which an interface operates unrestricted by STP.

frame A term referring to a data-link header and trailer, plus the data encapsulated between the header and trailer.

Frame Check Sequence A field in many data-link trailers used as part of the error-detection process.

full duplex Generically, any communication in which two communicating devices can concurrently send and receive data. In Ethernet LANs, the allowance for both devices to send and receive at the same time, allowed when both devices disable their CSMA/CD logic.

full state In OSPF, a neighbor state that implies that the two routers have exchanged the complete (full) contents of their respective LSDBs.

full update With IP routing protocols, the general concept that a routing protocol update lists all known routes.

fully adjacent In OSPF, a characterization of the state of a neighbor in which the two neighbors have reached the full state.

fully adjacent neighbor In OSPF, a neighbor with which the local router has also reached the OSPF full state, meaning that the two routers have exchanged their LSDBs directly with each other.

G

Gigabit Ethernet The common name for all the IEEE standards that send data at 1 gigabit per second.

global routing prefix An IPv6 prefix that defines an IPv6 address block made up of global unicast addresses, assigned to one organization, so that the organization has a block of globally unique IPv6 addresses to use in its network.

global unicast address A type of unicast IPv6 address that has been allocated from a range of public globally unique IP addresses, as registered through IANA/ICANN, its member agencies, and other registries or ISPs.

H

half duplex Generically, any communication in which only one device at a time can send data. In Ethernet LANs, the normal result of the CSMA/CD algorithm that enforces the rule that only one device should send at any point in time.

HDLC High-Level Data Link Control. A bit-oriented synchronous data-link layer protocol developed by the International Organization for Standardization (ISO).

header In computer networking, a set of bytes placed in front of some other data, encapsulating that data, as defined by a particular protocol.

Hello (Multiple definitions) 1) A protocol used by OSPF routers to discover, establish, and maintain neighbor relationships. 2) A protocol used by EIGRP routers to discover, establish, and maintain neighbor relationships. 3) In STP, refers to the name of the periodic message sourced by the root bridge in a spanning tree.

Hello BPDU The STP and RSTP message used for the majority of STP communications, listing the root's bridge ID, the sending device's bridge ID, and the sending device's cost with which to reach the root.

Hello Interval With OSPF and EIGRP, an interface timer that dictates how often the router should send Hello messages.

Hello timer In STP, the time interval at which the root switch should send Hello BPDUs.

history buffer In a Cisco router or switch, the function by which IOS keeps a list of commands that the user has used in this login session, both in EXEC mode and configuration mode. The user can then recall these commands for easier repeating or making small edits and issuing similar commands.

hop count The metric used by the RIP routing protocol. Each router in an IP route is considered a hop, so for example, if two other routers sit between a router and some subnet, that router would have a hop count of two for that route.

host Any device that uses an IP address.

host address The IP address assigned to a network card on a computer.

host part A term used to describe a part of an IPv4 address that is used to uniquely identify a host inside a subnet. The host part is identified by the bits of value 0 in the subnet mask.

host route A route with a /32 mask, which by virtue of this mask represents a route to a single host IP address.

hostname The alphanumeric name of an IP host.

hub A LAN device that provides a centralized connection point for LAN cabling, repeating any received electrical signal out all other ports, thereby creating a logical bus. Hubs do not interpret the electrical signals as a frame of bits, so hubs are considered to be Layer 1 devices.

I

IANA The Internet Assigned Numbers Authority (IANA). An organization that owns the rights to assign many operating numbers and facts about how the global Internet works, including public IPv4 and IPv6 addresses. *See also* ICANN.

ICANN The Internet Corporation for Assigned Names and Numbers. An organization appointed by IANA to oversee the distributed process of assigning public IPv4 and IPv6 addresses across the globe.

ICMP Internet Control Message Protocol. A TCP/IP network layer protocol that reports errors and provides other information relevant to IP packet processing.

ICMP echo reply One type of ICMP message, created specifically to be used as the message sent by the ping command to test connectivity in a network. The ping command expects to receive these messages from other hosts, after the ping command first sends an ICMP echo request message to the host.

ICMP echo request One type of ICMP message, created specifically to be used as the message sent by the ping command to test connectivity in a network. The ping command sends these messages to other hosts, expecting the other host to reply with an ICMP echo reply message.

IEEE Institute of Electrical and Electronics Engineers. A professional organization that develops communications and network standards, among other activities.

IEEE 802.11 The IEEE base standard for wireless LANs.

IEEE 802.1Q The IEEE standard VLAN trunking protocol. 802.1Q includes the concept of a native VLAN, for which no VLAN header is added, and a 4-byte VLAN header is inserted after the original frame's Type/Length field.

IEEE 802.2 An IEEE LAN protocol that specifies an implementation of the LLC sublayer of the data-link layer.

IEEE 802.3 A set of IEEE LAN protocols that specifies the many variations of what is known today as an Ethernet LAN.

IEEE 802.3 AD The IEEE standard for the functional equivalent of the Cisco-proprietary EtherChannel.

IETF The Internet Engineering Task Force. The IETF serves as the primary organization that works directly to create new TCP/IP standards.

IGP *See* interior gateway protocol.

inactivity timer For switch MAC address tables, a timer associated with each entry that counts time upward from 0 and is reset to 0 each time a switch receives a frame with the same MAC address. The entries with the largest timers can be removed to make space for additional MAC address table entries.

infrastructure mode The operating mode of an AP that is providing a BSS for wireless clients.

Integrated Services Router (ISR) Cisco's long-running term for several different model series of Enterprise-class routers, intended mostly for use as enterprise routers and some use as SOHO routers. ISR routers first serve as routers but, depending on the family or specific model, support all current types of WAN connections (private and Internet), LAN switching ports, Wireless APs, VPNs, and other integrated functions supported in a single device.

interface bandwidth In OSPF, the numerator in the calculation of an interface's default OSPF cost metric, calculated as the interface bandwidth divided by the reference bandwidth.

Interface ID The ending (rightmost) portion of the structure of an IPv6 address, usually 64 bits long.

interface-local scope A concept in IPv6 for which packets sent to an address using this scope should not physically exit the interface, keeping the packet inside the sending host.

interior gateway protocol (IGP) A routing protocol designed to be used to exchange routing information inside a single autonomous system.

interior routing protocol A synonym of interior gateway protocol. *See* interior gateway protocol.

Internal Border Gateway Protocol (iBGP) The use of BGP between two routers in the same ASN, with different rules compared to External BGP (eBGP).

internal router In OSPF, a router with all interfaces in the same nonbackbone area.

Internetwork Operating System The operating system (OS) of Cisco routers and switches, which provides the majority of a router's or switch's features, with the hardware providing the remaining features.

IOS *See* Internetwork Operating System.

IOS XE A Cisco operating system (OS) with a modern Linux-based multitasking software architecture used as the OS for many enterprise-class LAN switch and router products.

IP Internet Protocol. The network layer protocol in the TCP/IP stack, providing routing and logical addressing standards and services.

IP address (IP version 4) In IP version 4 (IPv4), a 32-bit address assigned to hosts using TCP/IP. Each address consists of a network number, an optional subnetwork number, and a host number. The network and subnetwork numbers together are used for routing, and the host number is used to address an individual host within the network or subnetwork.

IP address (IP version 6) In IP version 6 (IPv6), a 128-bit address assigned to hosts using TCP/IP. Addresses use different formats, commonly using a routing prefix, subnet, and interface ID, corresponding to the IPv4 network, subnet, and host parts of an address.

IP network *See* classful IP network.

IP packet An IP header, followed by the data encapsulated after the IP header, but specifically not including any headers and trailers for layers below the network layer.

IP routing table *See* routing table.

IP subnet Subdivisions of a Class A, B, or C network, as configured by a network administrator. Subnets allow a single Class A, B, or C network to be used instead of multiple networks, and still allow for a large number of groups of IP addresses, as is required for efficient IP routing.

IP version 4 Literally, the version of the Internet Protocol defined in an old RFC 791, standardized in 1980, and used as the basis of TCP/IP networks and the Internet for over 30 years.

IP version 6 A newer version of the Internet Protocol defined in RFC 2460, as well as many other RFCs, whose creation was motivated by the need to avoid the IPv4 address exhaustion problem.

IPv4 *See* IP version 4.

IPv4 address exhaustion The process by which the public IPv4 addresses, available to create the Internet, were consumed through the 1980s until today, with the expectation that eventually the world would run out of available IPv4 addresses.

IPv6 *See* IP version 6.

IPv6 address scope The concept of how far an IPv6 packet should be forwarded by hosts and routers in an IPv6 network. Includes interface-local, link-local, site-local, and organization-local scopes.

IPv6 administrative distance In Cisco routers, a means for one router to choose between multiple IPv6 routes to reach the same subnet when those routes were learned by different routing protocols. The lower the administrative distance, the better the source of the routing information.

IPv6 host route A route with a /128 mask, which by virtue of this mask represents a route to a single host IPv6 address.

IPv6 local route A route added to an IPv6 router's routing table for the router's interface IP address, with a /128 mask, which by virtue of this mask represents a route to only that router's IPv4 address.

IPv6 neighbor table The IPv6 equivalent of the ARP table. A table that lists IPv6 addresses of other hosts on the same link, along with their matching MAC addresses, as typically learned using Neighbor Discovery Protocol (NDP).

ISO International Organization for Standardization. An international organization that is responsible for a wide range of standards, including many standards relevant to networking. The ISO developed the OSI reference model, a popular networking reference model.

K–L

keepalive A proprietary feature of Cisco routers in which the router sends messages on a periodic basis as a means of letting the neighboring router know that the first router is still alive and well.

known unicast frame An Ethernet frame whose destination MAC address is listed in a switch's MAC address table, so the switch will forward the frame out the one port associated with that entry in the MAC address table.

LACP Link Aggregation Control Protocol is a messaging protocol defined by the IEEE 802.3ad standard that enables two neighboring devices to realize that they have multiple parallel links connecting to each other and then to decide which links can be combined into an EtherChannel.

Layer 2 EtherChannel (L2 EtherChannel) An EtherChannel that acts as a switched port (that is, not a routed port), and as such, is used by a switch's Layer 2 forwarding logic. As a result, the Layer 2 switch lists the Layer 2 EtherChannel in switch MAC address tables, and when forwarding a frame based on one of these MAC table entries, the switch balances traffic across the various ports in the Layer 2 EtherChannel.

Layer 3 EtherChannel (L3 EtherChannel) An EtherChannel that acts as a routed port (that is, not a switched port), and as such, is used by a switch's Layer 3 forwarding logic. As a result, the Layer 3 switch lists the Layer 3 EtherChannel in various routes in the switch's IP routing table, with the switch balancing traffic across the various ports in the Layer 3 EtherChannel.

Layer 3 protocol A protocol that has characteristics like OSI Layer 3, which defines logical addressing and routing. IPv4 and IPv6 are Layer 3 protocols.

Layer 3 switch *See* multilayer switch.

learning The process used by switches for discovering MAC addresses, and their relative location, by looking at the source MAC address of all frames received by a bridge or switch.

learning state In STP, a temporary port state in which the interface does not forward frames, but it can begin to learn MAC addresses from frames received on the interface.

leased line A serial communications circuit between two points, provided by some service provider, typically a telephone company (telco). Because the telco does not sell a physical cable between the two endpoints, instead charging a monthly fee for the ability to send bits between the two sites, the service is considered to be a leased service.

link state A classification of the underlying algorithm used in some routing protocols. Link-state protocols build a detailed database that lists links (subnets) and their state (up, down), from which the best routes can then be calculated.

link-local address (LLA) A unicast IPv6 address that begins FE80, used on each IPv6-enabled interface, used for sending packets within the attached link by applying a link-local scope.

link-local multicast address A multicast IPv6 address that begins with FF02, with the fourth digit of 2 identifying the scope as link-local, to which devices apply a link-local scope.

link-local scope With IPv6 multicasts, a term that refers to the parts (scope) of the network to which a multicast packet can flow, with link-local referring to the fact that the packet stays on the subnet in which it originated.

link-state advertisement (LSA) In OSPF, the name of the data structure that resides inside the LSDB and describes in detail the various components in a network, including routers and links (subnets).

link-state database (LSDB) In OSPF, the data structure in RAM of a router that holds the various LSAs, with the collective LSAs representing the entire topology of the network.

Link-State Request An OSPF packet used to ask a neighboring router to send a particular LSA.

Link-State Update An OSPF packet used to send an LSA to a neighboring router.

listening state A temporary STP port state that occurs immediately when a blocking interface must be moved to a forwarding state. The switch times out MAC table entries during this state. It also ignores frames received on the interface and doesn't forward any frames out the interface.

LLC Logical Link Control. The higher of the two sublayers of the data-link layer defined by the IEEE. Synonymous with IEEE 802.2.

local broadcast IP address IPv4 address 255.255.255.255. A packet sent to this address is sent as a data-link broadcast, but only flows to hosts in the subnet into which it was originally sent. Routers do not forward these packets.

local mode The default mode of a Cisco lightweight AP that offers one or more functioning BSSs on a specific channel.

local route A route added to an IPv4 router's routing table for the router's interface IP address, with a /32 mask, which by virtue of this mask represents a route to only that router's IPv4 address.

local username A username (with matching password), configured on a router or switch. It is considered local because it exists on the router or switch, and not on a remote server.

logical address A generic reference to addresses as defined by Layer 3 protocols that do not have to be concerned with the physical details of the underlying physical media. Used mainly to contrast these addresses with data-link addresses, which are generically considered to be physical addresses because they differ based on the type of physical medium.

longest prefix match When a router's IP routing table has more than one route that matches a packet's destination address, the choice to use the matching route with the longest mask (the mask with the largest number of binary 1s in the mask).

Loop Guard A complex Cisco switch mechanism that protects against STP loops in a specific common case. For switches whose switch-to-switch links settle into either a root port or an alternate port role, as expected per the STP design, Loop Guard disables use of those ports if normal STP operation attempts to assign them the designated port role.

LSA *See* link-state advertisement.

LSDB *See* link-state database.

M

MAC Media Access Control. The lower of the two sublayers of the data-link layer defined by the IEEE. Synonymous with IEEE 802.3 for Ethernet LANs.

MAC address A standardized data-link layer address that is required for every device that connects to a LAN. Ethernet MAC addresses are 6 bytes long and are controlled by the IEEE. Also known as a hardware address, a MAC layer address, and a physical address.

MAC address table A table of forwarding information held by a Layer 2 switch, built dynamically by listening to incoming frames and used by the switch to match frames to make decisions about where to forward the frame.

MaxAge In STP, a timer that states how long a switch should wait when it no longer receives Hellos from the root switch before acting to reconverge the STP topology. Also called the MaxAge timer.

media access control (MAC) layer A low-level function performed as part of Layer 2; in wireless networks, this function can be divided between a wireless LAN controller and a lightweight AP to form a split-MAC architecture.

message of the day One type of login banner that can be defined on a Cisco router or switch.

metric A unit of measure used by routing protocol algorithms to determine the best route for traffic to use to reach a particular destination.

Modified EUI-64 *See* EUI-64.

multiarea In OSPFv2 and OSPFv3, a design that uses multiple areas.

multiarea OSPF In OSPFv2 and OSPFv3, a design that uses more than one area within one OSPF domain (typically a single company).

multicast IP address A class D IPv4 address. When used as a destination address in a packet, the routers collectively work to deliver copies of the one original packet to all hosts who have previously registered to receive packets sent to that particular multicast address.

multilayer switch A LAN switch that can also perform Layer 3 routing functions. The name comes from the fact that this device makes forwarding decisions based on logic from multiple OSI layers (Layers 2 and 3).

multimode fiber A type of fiber cable that works well with transmitters like LEDs that emit multiple angles of light into the core of the cable; to accommodate the multiple angles of incident, the cable has a larger core in comparison to single-mode fiber cables.

N

name resolution The process by which an IP host discovers the IP address associated with a hostname, often involving sending a DNS request to a DNS server, with the server supplying the IP address used by a host with the listed hostname.

name server A server connected to a network that resolves network names into network addresses.

NAT Network Address Translation. A mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet, by translating those addresses into public addresses in the globally routable address space.

native VLAN The one VLAN ID on any 802.1Q VLAN trunk for which the trunk forwards frames without an 802.1Q header.

neighbor In routing protocols, another router with which a router decides to exchange routing information.

Neighbor Advertisement (NA) A message defined by the IPv6 Neighbor Discovery Protocol (NDP), used to declare to other neighbors a host's MAC address. Sometimes sent in response to a previously received NDP Neighbor Solicitation (NS) message.

Neighbor Discovery Protocol (NDP) A protocol that is part of the IPv6 protocol suite, used to discover and exchange information about devices on the same subnet (neighbors). In particular, it replaces the IPv4 ARP protocol.

Neighbor Solicitation (NS) A message defined by the IPv6 Neighbor Discovery Protocol (NDP), used to ask a neighbor to reply with a Neighbor Advertisement, which lists the neighbor's MAC address.

neighbor table For OSPF and EIGRP, a list of routers that have reached neighbor status.

network A collection of computers, printers, routers, switches, and other devices that can communicate with each other over some transmission medium.

network address *See* network number.

network broadcast address In IPv4, a special address in each classful network that can be used to broadcast a packet to all hosts in that same classful network. Numerically, the address has the same value as the network number in the network part of the address and all 255s in the host octets; for example, 10.255.255.255 is the network broadcast address for classful network 10.0.0.0.

network ID A number that identifies an IPv4 network, using a number in dotted-decimal notation (like IP addresses); a number that represents any single Class A, B, or C IP network.

network interface card (NIC) A computer card, sometimes an expansion card and sometimes integrated into the motherboard of the computer, that provides the electronics and other functions to connect to a computer network. Today, most NICs are specifically Ethernet NICs, and most have an RJ-45 port, the most common type of Ethernet port.

Network LSA In OSPF, a type of LSA that a designated router (DR) creates for the network (subnet) for which the DR is helping to distribute LSAs.

network number A number that uses dotted-decimal notation like IP addresses, but the number itself represents all hosts in a single Class A, B, or C IP network.

network part The portion of an IPv4 address that is either 1, 2, or 3 octets/bytes long, based on whether the address is in a Class A, B, or C network.

network route A route for a classful network.

networking model A generic term referring to any set of protocols and standards collected into a comprehensive grouping that, when followed by the devices in a network, allows all the devices to communicate. Examples include TCP/IP and OSI.

next-hop router In an IP route in a routing table, part of a routing table entry that refers to the next IP router (by IP address) that should receive packets that match the route.

NIC *See* network interface card.

NVRAM Nonvolatile RAM. A type of random-access memory (RAM) that retains its contents when a unit is powered off.

O

on-link prefix An IPv6 subnet prefix, advertised by a router in an NDP Router Advertisement (RA) message, that identifies to on-link hosts a subnet considered to exist on the local link.

Organization-local scope A concept in IPv6 for which packets sent to an address using this scope should be forwarded by routers inside the organization but not over any links connected to other organizations or over links connected to the Internet.

OSI Open System Interconnection reference model. A network architectural model developed by the ISO. The model consists of seven layers, each of which specifies particular network functions, such as addressing, flow control, error control, encapsulation, and reliable message transfer.

OSPF Open Shortest Path First. A popular link-state IGP that uses a link-state database and the Shortest Path First (SPF) algorithm to calculate the best routes to reach each known subnet.

OSPF neighbor A local router's attitude toward a second router that resides on a common subnet, when both use OSPF and use appropriate OSPF settings so that the routers will choose to exchange routing information with each other.

OSPF priority An OSPF interface setting that serves as the first comparison point in the designated router (DR) election process, with the router interface with the highest priority winning the election.

OSPF router-id In OSPF, the 32-bit number, normally shown in dotted-decimal notation but also allowed to be listed as an integer, intended as a unique identifier for each OSPF router in an OSPF domain.

OSPF version 2 The version of the OSPF routing protocol that supports IPv4, and not IPv6, and has been commonly used for over 20 years.

OSPF version 3 The version of the OSPF routing protocol that originally supported only IPv6, and not IPv4, but now supports IPv4 through the use of address family configuration.

outgoing interface In an IP route in a routing table, part of a routing table entry that refers to the local interface out which the local router should forward packets that match the route.

overlapping subnets An (incorrect) IP subnet design condition in which one subnet's range of addresses includes addresses in the range of another subnet.

P

packet A logical grouping of bytes that includes the network layer header and encapsulated data, but specifically does not include any headers and trailers below the network layer.

PAgP Port Aggregation Protocol (PAgP) is a messaging protocol defined by Cisco that enables two neighboring devices to realize that they have multiple parallel links connecting to each other and then to decide which links can be combined into an EtherChannel.

parallel detection The term for the branch of IEEE autonegotiation steps that applies to a device that uses autonegotiation but the device on the other end of the link does not.

partial mesh A network topology in which more than two devices could physically communicate but, by choice, only a subset of the pairs of devices connected to the network is allowed to communicate directly.

passive interface With a routing protocol, a router interface for which the routing protocol is enabled on the interface, but for which the routing protocol does not send routing protocol messages out that interface.

patch cable An Ethernet cable, usually short, that connects from a device's Ethernet port to a wall plate or switch. With wiring inside a building, electricians prewire from the wiring closet to each cubicle or other location, with a patch cable connecting the short distance from the wall plate to the user device.

periodic update With routing protocols, the concept that the routing protocol advertises routes in a routing update on a regular periodic basis. This is typical of distance vector routing protocols.

permanent IPv6 address When using IPv6 SLAAC, a host creates an address with an infinite preferred and valid lifetime, making the address permanent, using the address when listening for incoming connections to any services running on that host.

ping An Internet Control Message Protocol (ICMP) echo message and its reply; ping often is used in IP networks to test the reachability of a network device.

pinout The documentation and implementation of which wires inside a cable connect to each pin position in any connector.

point-to-point network type An OSPF interface setting, useful on links in a point-to-point topology with only two routers, resulting in flooding directly between the routers without the use of a designated router (DR).

port In TCP and UDP, a number that is used to uniquely identify the application process that either sent (source port) or should receive (destination port) data. In LAN switching, another term for *switch interface*.

PortChannel One term Cisco switches use to reference a bundle of links that are, in some respects, treated like a single link. Other similar terms include *EtherChannel* and *Channel-group*.

PortFast A switch STP feature in which a port is placed in an STP forwarding state as soon as the interface comes up, bypassing the listening and learning states. This feature is meant for ports connected to end-user devices.

preferred lifetime In the context of IPv6 addresses, a timer applied to a SLAAC-created address defining how long the address is preferred, meaning the host uses the address for new and existing application flows.

Prefix (prefix ID) In both IPv4 and IPv6, this term refers to the number that identifies a group of IPv4 or IPv6 addresses, respectively. Another term for *subnet identifier*.

prefix discovery IPv6 neighbor discovery protocol (NDP) function, specifically part of the Router Advertisement (RA) message, in which the router supplies a list of IPv6 subnet prefixes and prefix lengths that exist on the local link.

prefix length In IPv6, the number of bits in an IPv6 prefix.

prefix mask A term to describe an IPv4 subnet mask when represented as a slash (/) followed by a decimal number. The decimal number is the number of binary 1s in the mask.

prefix notation (IP version 4) A shorter way to write a subnet mask in which the number of binary 1s in the mask is simply written in decimal. For example, /24 denotes the subnet mask with 24 binary 1 bits in the subnet mask. The number of bits of value binary 1 in the mask is considered to be the prefix length.

primary root This term refers to the switch configured with the primary keyword on the **spanning-tree vlan x root {primary | secondary}** command. At time of configuration, this command causes the switch to choose a new priority setting that makes the switch become the root switch in the network.

private addresses IP addresses in several Class A, B, and C networks that are set aside for use inside private organizations. These addresses, as defined in RFC 1918, are not routable through the Internet.

private IP network Any of the IPv4 Class A, B, or C networks as defined by RFC 1918, intended for use inside a company but not used as public IP networks.

Protocol Type field A field in a LAN header that identifies the type of header that follows the LAN header. Includes the DIX Ethernet Type field, the IEEE 802.2 DSAP field, and the SNAP protocol Type field.

public IP address An IP address that is part of a registered network number, as assigned by an Internet Assigned Numbers Authority (IANA) member agency, so that only the organization to which the address is registered is allowed to use the address. Routers in the Internet should have routes allowing them to forward packets to all the publicly registered IP addresses.

public IP network Any IPv4 Class A, B, or C network assigned for use by one organization only, so that the addresses in the network are unique across the Internet, allowing packets to be sent through the public Internet using the addresses.

PVST+ An STP option in Cisco switches that creates an STP instance per VLAN. Cisco proprietary.

Q–R

quartet A term used in this book, but not in other references, to refer to a set of four hex digits in an IPv6 address.

RADIUS server An authentication server used with 802.1x to authenticate wireless clients.

RAM Random-access memory. A type of volatile memory that can be read and written by a microprocessor.

Rapid PVST+ An STP option in Cisco switches that creates an RSTP instance per VLAN. Cisco proprietary.

Rapid Spanning Tree Protocol (RSTP) Defined in IEEE 802.lw. Defines an improved version of STP that converges much more quickly and consistently than STP (802.Id).

reference bandwidth In OSPF, a configurable value for the OSPF routing process, used by OSPF when calculating an interface's default OSPF cost metric, calculated as the interface's bandwidth divided by the reference bandwidth.

Regional Internet Registry An organization (five globally) that receives allocations of public IPv4 addresses from IANA and then manages that address space in their major geographic region, performing public address allocations to ISPs and assignments directly to companies that use the addresses.

repeater A device that repeats or retransmits signals it receives, effectively expanding the wireless coverage area.

resident subnet Each IP subnet contains a number of unicast IP addresses; that subnet is the resident subnet for each of those addresses—that is, the subnet in which those addresses reside.

reverse route From one host's perspective, for packets sent back to the host from another host, the route over which the packet travels.

RFC Request For Comments. A document used as the primary means for communicating information about the TCP/IP protocols. Some RFCs are designated by the Internet Architecture Board (IAB) as Internet standards, and others are informational. RFCs are available online from numerous sources, including http://www.rfc-editor.org.

RIP Routing Information Protocol. An interior gateway protocol (IGP) that uses distance vector logic and router hop count as the metric. RIP version 2 (RIPv2) replaced the older RIP version 1 (RIPv1), with RIPv2 providing more features, including support for VLSM.

RIR *See* Regional Internet Registry.

RJ-45 A popular type of cabling connector used for Ethernet cabling. It is similar to the RJ-11 connector used for telephone wiring in homes in the United States. RJ-45 allows the connection of eight wires.

ROAS *See* Router-on-a-Stick.

rollover cable For connections to the RJ-45 console port rather than USB console port of a Cisco device, the type of cable used between the user PC and the console port. The UTP rollover cable uses a rollover pinout with eight pins, connecting pin 1 to pin 8, pin 2 to pin 7, and so on.

ROM Read-only memory. A type of nonvolatile memory that can be read but not written to by the microprocessor.

ROMMON A shorter name for ROM Monitor, which is a low-level operating system that can be loaded into Cisco routers for several seldom-needed maintenance tasks, including password recovery and loading a new IOS when flash memory has been corrupted.

root bridge *See* root switch.

root cost The STP cost from a nonroot switch to reach the root switch, as the sum of all STP costs for all ports out which a frame would exit to reach the root.

Root Guard A Cisco switch feature that protects against unexpected new root switches. When enabled on an interface, IOS uses normal STP rules except to disable the use of the port after receiving a superior BPDU.

root port role In STP and RSTP, the one port on a nonroot switch in which the least-cost Hello is received. Switches put root ports in a forwarding state.

root switch In STP and RSTP, the switch that wins the election by virtue of having the lowest bridge ID and, as a result, sends periodic Hello BPDUs (default, 2 seconds).

routed port A reference to the default logic on an interface on a router, such as de-encapsulating the Layer 3 packet from incoming Layer 2 frames and passing the packet to the Layer 3 forwarding logic of the device.

routed protocol A protocol that defines packets that can be routed by a router. Examples of routed protocols include IPv4 and IPv6.

Router Advertisement (RA) A message defined by the IPv6 Neighbor Discovery Protocol (NDP), used by routers to announce their willingness to act as an IPv6 router on a link. These can be sent in response to a previously received NDP Router Solicitation (RS) message.

router ID (RID) In EIGRP and OSPF, a 32-bit number, written in dotted-decimal notation, that uniquely identifies each router.

router LSA In OSPF, a type of LSA that a router creates to describe itself and the networks connected to it.

Router-on-a-Stick (ROAS) Jargon to refer to the Cisco router feature of using VLAN trunking on an Ethernet interface, which then allows the router to route packets that happen to enter the router on that trunk and then exit the router on that same trunk, just on a different VLAN.

Router Solicitation (RS) A message defined by the IPv6 Neighbor Discovery Protocol (NDP), used to ask any routers on the link to reply, identifying the router, plus other configuration settings (prefixes and prefix lengths).

routing protocol A set of messages and processes with which routers can exchange information about routes to reach subnets in a particular network. Examples of routing protocols include Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

routing table A list of routes in a router, with each route listing the destination subnet and mask, the router interface out which to forward packets destined to that subnet, and as needed, the next-hop router's IP address.

routing update A generic reference to any routing protocol's messages in which it sends routing information to a neighbor.

RSTP *See* Rapid Spanning Tree Protocol.

running-config file In Cisco IOS switches and routers, the name of the file that resides in RAM, holding the device's currently used configuration.

S

same-layer interaction The communication between two networking devices for the purposes of the functions defined at a particular layer of a networking model, with that communication happening by using a header defined by that layer of the model. The two devices set values in the header, send the header and encapsulated data, with the receiving devices interpreting the header to decide what action to take.

secondary root This term refers to the switch configured with the secondary keyword on the **spanning-tree vlan x root {primary | secondary}** command. At time of configuration, this command causes the switch to set its base priority to 28,762.

Secure Shell (SSH) A TCP/IP application layer protocol that supports terminal emulation between a client and server, using dynamic key exchange and encryption to keep the communications private.

segment In TCP, a term used to describe a TCP header and its encapsulated data (also called an L4PDU). Also in TCP, the process of accepting a large chunk of data from the application layer and breaking it into smaller pieces that fit into TCP segments. In Ethernet, a segment is either a single Ethernet cable or a single collision domain (no matter how many cables are used).

shared Ethernet An Ethernet that uses a hub, or even the original coaxial cabling, that results in the devices having to take turns sending data, sharing the available bandwidth.

shortest path first (SPF) algorithm The name of the algorithm used by link-state routing protocols to analyze the LSDB and find the least-cost routes from that router to each subnet.

single-area OSPF In OSPFv2 and OSPFv3, a design that uses a single area within one OSPF domain (typically a single company).

single-mode fiber A type of fiber cable that works well with transmitters like lasers that emit a single angle of light into the core of the cable, allowing for a smaller core in comparison to multimode fiber cables.

site-local scope A concept in IPv6 for which packets sent to an address using this scope should be forwarded by routers, but not forwarded over WAN links to other sites.

SOHO router A term to describe the general role of a router that exists as part of the enterprise network but resides at an employee's home or at a smaller business site, possibly with a short-term lease compared to larger enterprise sites. These sites typically have few devices, so it makes sense to use one device that integrates routing, switches, wireless, and other features into a single device (the SOHO router) and are more likely to justify Internet access as the primary WAN access method.

solicited-node multicast address A type of IPv6 multicast address, with link-local scope, used to send packets to all hosts in the subnet that share the same value in the last six hex digits of their unicast IPv6 addresses. Begins with FF02::1:FF00:0/104.

Spanning Tree Protocol (STP) A protocol defined by IEEE standard 802.ID. Allows switches and bridges to create a redundant LAN, with the protocol dynamically causing some ports to block traffic, so that the bridge/switch forwarding logic will not cause frames to loop indefinitely around the LAN.

SSH *See* Secure Shell.

standard access list A list of IOS global configuration commands that can match only a packet's source IP address, for the purpose of deciding which packets to discard and which to allow through the router.

star topology A network topology in which endpoints on a network are connected to a common central device by point-to-point links.

startup-config file In Cisco IOS switches and routers, the name of the file that resides in NVRAM memory, holding the device's configuration that will be loaded into RAM as the running-config file when the device is next reloaded or powered on.

stateful DHCPv6 A term used in IPv6 to contrast with stateless DHCP. Stateful DHCP keeps track of which clients have been assigned which IPv6 addresses (state information).

stateless address autoconfiguration (SLAAC) A feature of IPv6 in which a host or router can be assigned an IPv6 unicast address without the need for a stateful DHCP server.

stateless DHCPv6 A term used in IPv6 to contrast with stateful DHCP. Stateless DHCP servers don't lease IPv6 addresses to clients. Instead, they supply other useful information, such as DNS server IP addresses, but with no need to track information about the clients (state information).

static access interface A LAN network design term, synonymous with the term *access interface*, but emphasizing that the port is assigned to one VLAN as a result of static configuration rather than through some dynamic process.

static route An IP route on a router created by the user configuring the details of the route on the local router.

STP Shielded twisted-pair. This type of cabling has a layer of shielded insulation to reduce electromagnetic interference (EMI).

straight-through cable In Ethernet, a cable that connects the wire on pin 1 on one end of the cable to pin 1 on the other end of the cable, pin 2 on one end to pin 2 on the other end, and so on.

subinterface One of the virtual interfaces on a single physical interface.

subnet Subdivisions of a Class A, B, or C network, as configured by a network administrator. Subnets allow a single Class A, B, or C network to be used instead of multiple networks, and still allow for a large number of groups of IP addresses, as is required for efficient IP routing.

subnet address *See* subnet number.

subnet broadcast address A special address in each IPv4 subnet, specifically the largest numeric address in the subnet, designed so that packets sent to this address should be delivered to all hosts in that subnet.

subnet ID (IPv4) *See* subnet number.

subnet ID (IPv6) The number that represents the IPv6 subnet. Also known as the IPv6 prefix, or more formally as the subnet-router anycast address.

subnet ID (prefix ID) *See* subnet number.

subnet mask A 32-bit number that numerically describes the format of an IP address, by representing the combined network and subnet bits in the address with mask bit values of 1, and representing the host bits in the address with mask bit values of 0.

subnet number In IPv4, a dotted-decimal number that represents all addresses in a single subnet. Numerically, the smallest value in the range of numbers in a subnet, reserved so that it cannot be used as a unicast IP address by a host.

subnet part In a subnetted IPv4 address, interpreted with classful addressing rules, one of three parts of the structure of an IP address, with the subnet part uniquely identifying different subnets of a classful IP network.

subnet prefix The term for a number that represents an IPv6 subnet.

subnet router anycast address A special anycast address in each IPv6 subnet, reserved for use by routers as a way to send a packet to any router on the subnet. The address's value in each subnet is the same number as the subnet ID.

subnet zero An alternative term for *zero subnet*. *See* zero subnet.

subnetting The process of subdividing a Class A, B, or C network into smaller groups called subnets.

summary LSA In OSPFv2, a type of LSA, created by an Area Border Router (ABR), to describe a subnet in one area in the database of another area.

superior BPDU An STP bridge protocol data unit (BPDU) that lists a better (lower) root bridge ID (BID) as compared to the current bridge. A switch that receives a superior BPDU under normal conditions would begin using the new switch with the lower BID as the root switch.

switch A network device that filters, forwards, and floods Ethernet frames based on the destination address of each frame.

switched Ethernet An Ethernet that uses a switch, and particularly not a hub, so that the devices connected to one switch port do not have to contend to use the bandwidth available on another port. This term contrasts with *shared Ethernet*, in which the devices must share bandwidth, whereas switched Ethernet provides much more capacity, as the devices do not have to share the available bandwidth.

switched port A reference to the default logic on a Layer 2 switch port, such as learning the source MAC address of received frames and passing incoming frames to the Layer 2 forwarding logic of the device.

switched virtual interface (SVI) Another term for any VLAN interface in a Cisco switch. *See also* VLAN interface.

system ID extension The term for the formatting applied to the original 16-bit STP priority field to break it into a 4-bit priority field and a 12-bit VLAN ID field.

T

TCP Transmission Control Protocol. A connection-oriented transport layer TCP/IP protocol that provides reliable data transmission.

TCP/IP Transmission Control Protocol/Internet Protocol. A common name for the suite of protocols developed by the U.S. Department of Defense in the 1970s to support the construction of worldwide internetworks. TCP and IP are the two best-known protocols in the suite.

telco A common abbreviation for *telephone company*.

Telnet The standard terminal-emulation application layer protocol in the TCP/IP protocol stack. Telnet is used for remote terminal connection, enabling users to log in to remote systems and use resources as if they were connected to a local system. Telnet is defined in RFC 854.

temporary IPv6 address When using IPv6 SLAAC, a host creates an address with a relatively short preferred and valid lifetime, usually days long for each, making the address temporary. The host uses the temporary address for outgoing application connections.

topology database The structured data that describes the network topology to a routing protocol. Link-state and balanced hybrid routing protocols use topology tables, from which they build the entries in the routing table.

trace Short for traceroute. A program available on many systems that traces the path that a packet takes to a destination. It is used mostly to troubleshoot routing problems between hosts.

traceroute A program available on many systems that traces the path that a packet takes to a destination. It is used mostly to debug routing problems between hosts.

trailer In computer networking, a set of bytes placed behind some other data, encapsulating that data, as defined by a particular protocol. Typically, only data-link layer protocols define trailers.

transceiver A term formed from the words *transmitter* and *receiver*. The hardware used to both send (transmit) energy over some communications medium (e.g., wires in a cable), as well as to process received energy signals to interpret as a series of 1s and 0s.

trunk In campus LANs, an Ethernet segment over which the devices add a VLAN header that identifies the VLAN in which the frame exists.

trunk interface A switch interface configured so that it operates using VLAN trunking (either 802.1Q or ISL).

trunking Also called *VLAN trunking*. A method (using either the Cisco ISL protocol or the IEEE 802.1Q protocol) to support multiple VLANs, allowing traffic from those VLANs to cross a single link.

trunking administrative mode The configured trunking setting on a Cisco switch interface, as configured with the switchport mode command.

trunking operational mode The current behavior of a Cisco switch interface for VLAN trunking.

twisted-pair Transmission medium consisting of two insulated wires, with the wires twisted around each other in a spiral. An electrical circuit flows over the wire pair, with the current in opposite directions on each wire, which significantly reduces the interference between the two wires.

U

UDP User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery.

ULA global ID The unique local address (ULA) for one organization uses a /48 prefix, composed of a set hex FD in the first 8 bits, with a 10-hex digit (40 bit) global ID, which should be generated by a pseudo-random algorithm.

unicast address Generally, any address in networking that represents a single device or interface, instead of a group of addresses (as would be represented by a multicast or broadcast address).

unicast IP address An IP address that represents a single interface. In IPv4, these addresses come from the Class A, B, and C ranges.

unidirectional link A condition on fiber optic links in which one of the two required fibers fails in a way that also results in both attached switches leaving their interfaces in a working (connected) state—when, in reality, the link can pass frames only in a single direction.

unique local address A type of IPv6 unicast address meant as a replacement for IPv4 private addresses.

unknown unicast frame An Ethernet frame whose destination MAC address is not listed in a switch's MAC address table, so the switch must flood the frame.

up and up Jargon referring to the two interface states on a Cisco IOS router or switch (line status and protocol status), with the first "up" referring to the line status and the second "up" referring to the protocol status. An interface in this state should be able to pass data-link frames.

update timer The time interval that regulates how often a routing protocol sends its next periodic routing updates. Distance vector routing protocols send full routing updates every update interval.

user mode A mode of the user interface to a router or switch in which the user can type only nondisruptive EXEC commands, generally just to look at the current status, but not to change any operational settings.

UTP Unshielded twisted-pair. A type of cabling, standardized by the Telecommunications Industry Association (TIA), that holds twisted pairs of copper wires (typically four pair) and does not contain any shielding from outside interference.

V

valid lifetime In the context of IPv6 addresses, a timer applied to a SLAAC-created address defining how long the address is valid, meaning the host can continue to support existing application flows using the address.

variable-length subnet mask (VLSM) The capability to specify a different subnet mask for the same Class A, B, or C network number on different subnets. VLSM can help optimize available address space.

virtual LAN (VLAN) A group of devices, connected to one or more switches, with the devices grouped into a single broadcast domain through switch configuration. VLANs allow switch administrators to separate the devices connected to the switches into separate VLANs without requiring separate physical switches, gaining design advantages of separating the traffic without the expense of buying additional hardware.

virtual private network (VPN) The process of securing communication between two devices whose packets pass over some public and unsecured network, typically the Internet. VPNs encrypt packets so that the communication is private, and authenticate the identity of the endpoints.

VLAN *See* virtual LAN.

VLAN interface A configuration concept inside Cisco switches, used as an interface between IOS running on the switch and a VLAN supported inside the switch, so that the switch can assign an IP address and send IP packets into that VLAN.

VLAN Trunking Protocol (VTP) A Cisco-proprietary messaging protocol used between Cisco switches to communicate configuration information about the existence of VLANs, including the VLAN ID and VLAN name.

voice VLAN A VLAN defined for use by IP Phones, with the Cisco switch notifying the phone about the voice VLAN ID so that the phone can use 802.1Q frames to support traffic for the phone and the attached PC (which uses a data VLAN).

VoIP Voice over IP. The transport of voice traffic inside IP packets over an IP network.

VTP *See* VLAN Trunking Protocol.

VTP transparent mode One of three VTP operational modes. Switches in transparent mode can configure VLANs, but they do not tell other switches about the changes, and they do not learn about VLAN changes from other switches.

W

WAN *See* wide-area network.

web server Software, running on a computer, that stores web pages and sends those web pages to web clients (web browsers) that request the web pages.

wide-area network (WAN) A part of a larger network that implements mostly OSI Layer 1 and 2 technology, connects sites that typically sit far apart, and uses a business model in which a consumer (individual or business) must lease the WAN from a service provider (often a telco).

Wi-Fi Alliance An organization formed by many companies in the wireless industry (an industry association) for the purpose of getting multivendor certified-compatible wireless products to market in a more timely fashion than would be possible by simply relying on standardization processes.

wildcard mask The mask used in Cisco IOS ACL commands and OSPF and EIGRP network commands.

wired LAN A local-area network (LAN) that physically transmits bits using cables, often the wires inside cables. A term for local-area networks that use cables, emphasizing the fact that the LAN transmits data using wires (in cables) instead of wireless radio waves. *See also* wireless LAN.

wireless LAN A local-area network (LAN) that physically transmits bits using radio waves. The name “wireless” compares these LANs to more traditional “wired” LANs, which are LANs that use cables (which often have copper wires inside).

Z

zero subnet For every classful IPv4 network that is subnetted, the one subnet whose subnet number has all binary 0s in the subnet part of the number. In decimal, the zero subnet can be easily identified because it is the same number as the classful network number.

Index

This page intentionally left blank

APPENDIX D

Practice for Chapter 12: Analyzing Classful IPv4 Networks

Practice Problems

The practice problems in this appendix require that you determine a few basic facts about a network, given an IP address and an assumption that subnetting is not used in that network. To do so, refer to the processes described in Chapter 12 of *CCNA 200-301 Official Cert Guide, Volume 1*.

> **NOTE** You may also elect to do this same set of practice problems using the "Practice Exercise: Analyzing Classful IPv4 Networks" application on the companion website.

In particular, for the upcoming list of IP addresses, you should identify the following information:

- Class of the address
- Number of octets in the network part of the address
- Number of octets in the host part of the address
- Network number
- Network broadcast address

Find all these facts for the following IP addresses:

1. 10.55.44.3
2. 128.77.6.7
3. 192.168.76.54
4. 190.190.190.190
5. 9.1.1.1
6. 200.1.1.1
7. 201.1.77.5
8. 101.1.77.5
9. 119.67.99.240
10. 219.240.66.98

Answers

The process to answer these problems is relatively basic, so this section reviews the overall process and then lists the answers to problems 1–10.

The process starts by examining the first octet of the IP address:

- If the first octet of the IP address is a number between 1 and 126, inclusive, the address is a Class A address.
- If the first octet of the IP address is a number between 128 and 191, inclusive, the address is a Class B address.
- If the first octet of the IP address is a number between 192 and 223, inclusive, the address is a Class C address.

When no subnetting is used:

- Class A addresses have one octet in the network part of the address and three octets in the host part.
- Class B addresses have two octets each in the network and host part.
- Class C addresses have three octets in the network part and one octet in the host part.

After determining the class and the number of network octets, you can easily find the network number and network broadcast address. To find the network number, copy the network octets of the IP address and write down 0s for the host octets. To find the network broadcast address, copy the network octets of the IP address and write down 255s for the host octets.

Table D-1 lists all ten problems and their respective answers.

Table D-1 Answers to Problems

IP Address	Class	Number of Network Octets	Number of Host Octets	Network Number	Network Broadcast Address
10.55.44.3	A	1	3	10.0.0.0	10.255.255.255
128.77.6.7	B	2	2	128.77.0.0	128.77.255.255
192.168.76.54	C	3	1	192.168.76.0	192.168.76.255
190.190.190.190	B	2	2	190.190.0.0	190.190.255.255
9.1.1.1	A	1	3	9.0.0.0	9.255.255.255
200.1.1.1	C	3	1	200.1.1.0	200.1.1.255
201.1.77.55	C	3	1	201.1.77.0	201.1.77.255
101.1.77.55	A	1	3	101.0.0.0	101.255.255.255
119.67.99.240	A	1	3	119.0.0.0	119.255.255.255
219.240.66.98	C	3	1	219.240.66.0	219.240.66.255

Practice for Chapter 13: Analyzing Subnet Masks

This appendix begins with 23 mask conversion problems, followed by the matching answers and explanations. After that, the appendix lists 10 mask analysis problems, with the matching answers to follow.

NOTE You may also perform this same set of practice problems using the "Analyzing Subnet Masks" and "Mask Conversion" applications on the companion website.

Mask Conversion Problems

The problems in this appendix require you to convert dotted-decimal subnet masks to prefix format and vice versa. To do so, feel free to use the processes described in Chapter 13 of *CCNA 200-301 Official Cert Guide, Volume 1*.

Many people use the information in Table E-1 when converting masks. The table lists the nine dotted-decimal notation (DDN) mask values, the binary equivalent, and the number of binary 1s in the binary equivalent.

Table E-1 Nine Possible Values in One Octet of a Subnet Mask

Binary Mask Octet	DDN Mask Octet	Number of Binary 1s
00000000	0	0
10000000	128	1
11000000	192	2
11100000	224	3
11110000	240	4
11111000	248	5
11111100	252	6
11111110	254	7
11111111	255	8

Convert each DDN mask to prefix format and vice versa:

1. 255.240.0.0
2. 255.255.192.0
3. 255.255.255.224
4. 255.254.0.0.

5. 255.255.248.0
6. /30
7. /25
8. /11
9. /22
10. /24
11. 255.0.0.0
12. /29
13. /9
14. 255.192.0.0
15. 255.255.255.240
16. /26
17. /13
18. 255.255.254.0
19. 255.252.0.0
20. /20
21. /16
22. 255.255.224.0
23. 255.255.128.0

Answers to Mask Conversion Problems

Mask Conversion Problem 1: Answer

The answer is /12.

The binary process for converting the mask from dotted-decimal format to prefix format is relatively simple. The only hard part is converting the dotted-decimal number to binary. For reference, the process is as follows:

Step 1. Convert the dotted-decimal mask to binary.

Step 2. Count the number of binary 1s in the 32-bit binary mask; this is the value of the prefix notation mask.

For problem 1, mask 255.240.0.0 converts to the following:

11111111 11110000 00000000 00000000

You can see from the binary number that it contains 12 binary 1s, so the prefix format of the mask will be /12.

You can find the same answer without converting decimal to binary if you have memorized the nine DDN mask values, and the corresponding number of binary 1s in each, as listed earlier in Table E-1. Follow these steps:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 4 because the second mask octet of 240 includes four binary 1s.

Step 4. The resulting prefix is /12.

Mask Conversion Problem 2: Answer

The answer is /18.

For problem 2, mask 255.255.192.0 converts to the following:

```
11111111 11111111 11000000 00000000
```

You can see from the binary number that it contains 18 binary 1s, so the prefix format of the mask will be /18.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 2 because the third mask octet of 192 includes two binary 1s.

Step 5. The resulting prefix is /18.

Mask Conversion Problem 3: Answer

The answer is /27.

For problem 3, mask 255.255.255.224 converts to the following:

```
11111111 11111111 11111111 11100000
```

You can see from the binary number that it contains 27 binary 1s, so the prefix format of the mask will be /27.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 8 because the third mask octet of 255 includes eight binary 1s.

Step 5. (4th octet) Add 3 because the fourth mask octet of 224 includes three binary 1s.

Step 6. The resulting prefix is /27.

Mask Conversion Problem 4: Answer

The answer is /15.

For problem 4, mask 255.254.0.0 converts to the following:

11111111 11111110 00000000 00000000

You can see from the binary number that it contains 15 binary 1s, so the prefix format of the mask will be /15.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 7 because the second mask octet of 254 includes seven binary 1s.

Step 4. The resulting prefix is /15.

Mask Conversion Problem 5: Answer

The answer is /21.

For problem 5, mask 255.255.248.0 converts to the following:

11111111 11111111 11111000 00000000

You can see from the binary number that it contains 21 binary 1s, so the prefix format of the mask will be /21.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 5 because the third mask octet of 248 includes five binary 1s.

Step 5. The resulting prefix is /21.

Mask Conversion Problem 6: Answer

The answer is 255.255.255.252.

The binary process for converting the prefix version of the mask to dotted-decimal is straightforward, but again requires some binary math. For reference, the process runs like this:

Step 1. Write down *x* binary 1s, where *x* is the value listed in the prefix version of the mask.

Step 2. Write down binary 0s after the binary 1s until the combined 1s and 0s form a 32-bit number.

Step 3. Convert this binary number, 8 bits at a time, to decimal, to create a dotted-decimal number; this value is the dotted-decimal version of the subnet mask. (Refer to Table E-1, which lists the binary and decimal equivalents.)

For problem 6, with a prefix of /30, you start at Step 1 by writing down 30 binary 1s, as shown here:

11111111 11111111 11111111 111111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 11111111 11111100

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 7: Answer

The answer is 255.255.255.128.

For problem 7, with a prefix of /25, you start at Step 1 by writing down 25 binary 1s, as shown here:

11111111 11111111 11111111 1

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 11111111 10000000

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 8: Answer

The answer is 255.224.0.0.

For problem 8, with a prefix of /11, you start at Step 1 by writing down 11 binary 1s, as shown here:

11111111 111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11100000 00000000 00000000

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 9: Answer

The answer is 255.255.252.0.

For problem 9, with a prefix of /22, you start at Step 1 by writing down 22 binary 1s, as shown here:

11111111 11111111 111111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 111111**00 00000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 10: Answer

The answer is 255.255.255.0.

For problem 10, with a prefix of /24, you start at Step 1 by writing down 24 binary 1s, as shown here:

11111111 11111111 11111111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 11111111 **00000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 11: Answer

The answer is /8.

For problem 11, mask 255.0.0.0 converts to the following:

11111111 00000000 00000000 00000000

You can see from the binary number that it contains 8 binary 1s, so the prefix format of the mask will be /8.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 0 for the other octets because each mask octet of 0 includes zero binary 1s.

Step 4. The resulting prefix is /8.

Mask Conversion Problem 12: Answer

The answer is 255.255.255.248.

For problem 12, with a prefix of /29, you start at Step 1 by writing down 29 binary 1s, as shown here:

11111111 11111111 11111111 11111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 11111111 11111**000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 13: Answer

The answer is 255.128.0.0.

For problem 13, with a prefix of /9, you start at Step 1 by writing down 9 binary 1s, as shown here:

```
11111111 1
```

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

```
11111111 10000000 00000000 00000000
```

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 14: Answer

The answer is /10.

For problem 14, mask 255.192.0.0 converts to the following:

```
11111111 11000000 00000000 00000000
```

You can see from the binary number that it contains 10 binary 1s, so the prefix format of the mask will be /10.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 2 because the second mask octet of 192 includes two binary 1s.

Step 4. The resulting prefix is /10.

Mask Conversion Problem 15: Answer

The answer is /28.

For problem 15, mask 255.255.255.240 converts to the following:

```
11111111 11111111 11111111 11110000
```

You can see from the binary number that it contains 28 binary 1s, so the prefix format of the mask will be /28.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 8 because the third mask octet of 255 includes eight binary 1s.

Step 5. (4th octet) Add 4 because the fourth mask octet of 240 includes four binary 1s.

Step 6. The resulting prefix is /28.

Mask Conversion Problem 16: Answer

The answer is 255.255.255.192.

For problem 16, with a prefix of /26, you start at Step 1 by writing down 26 binary 1s, as shown here:

11111111 11111111 11111111 11

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 11111111 11**000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 17: Answer

The answer is 255.248.0.0.

For problem 17, with a prefix of /13, you start at Step 1 by writing down 13 binary 1s, as shown here:

11111111 11111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111**000 00000000 00000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 18: Answer

The answer is /23.

For problem 18, mask 255.255.254.0 converts to the following:

11111111 11111111 11111110 00000000

You can see from the binary number that it contains 23 binary 1s, so the prefix format of the mask will be /23.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 7 because the third mask octet of 254 includes seven binary 1s.

Step 5. The resulting prefix is /23.

Mask Conversion Problem 19: Answer

The answer is /14.

For problem 19, mask 255.252.0.0 converts to the following:

11111111 11111100 00000000 00000000

You can see from the binary number that it contains 14 binary 1s, so the prefix format of the mask will be /14.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 6 because the second mask octet of 252 includes six binary 1s.

Step 4. The resulting prefix is /14.

Mask Conversion Problem 20: Answer

The answer is 255.255.240.0.

For problem 20, with a prefix of /20, you start at Step 1 by writing down 20 binary 1s, as shown here:

11111111 11111111 1111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 1111**0000 00000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 21: Answer

The answer is 255.255.0.0.

For problem 21, with a prefix of /16, you start at Step 1 by writing down 16 binary 1s, as shown here:

11111111 11111111

At Step 2, you add binary 0s until you have 32 total bits, as shown next:

11111111 11111111 **00000000 00000000**

The only remaining work is to convert this 32-bit number to decimal, remembering that the conversion works with 8 bits at a time.

Mask Conversion Problem 22: Answer

The answer is /19.

For problem 22, mask 255.255.224.0 converts to the following:

11111111 11111111 11100000 00000000

You can see from the binary number that it contains 19 binary 1s, so the prefix format of the mask will be /19.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 3 because the third mask octet of 224 includes three binary 1s.

Step 5. The resulting prefix is /19.

Mask Conversion Problem 23: Answer

The answer is /17.

For problem 23, mask 255.255.128.0 converts to the following:

11111111 11111111 10000000 00000000

You can see from the binary number that it contains 17 binary 1s, so the prefix format of the mask will be /17.

If you memorized the number of binary 1s represented by each DDN mask value, you can possibly work faster with the following logic:

Step 1. Start with a prefix value of 0.

Step 2. (1st octet) Add 8 because the first mask octet of 255 includes eight binary 1s.

Step 3. (2nd octet) Add 8 because the second mask octet of 255 includes eight binary 1s.

Step 4. (3rd octet) Add 1 because the third mask octet of 128 includes one binary 1.

Step 5. The resulting prefix is /17.

Mask Analysis Problems

This appendix lists problems that require you to analyze an existing IP address and mask to determine the number of network, subnet, and host bits. From that, you should calculate the number of subnets possible when using the listed mask in the class of network shown in the problem, as well as the number of possible host addresses in each subnet.

To find this information, you can use the processes explained in Chapter 13 of *CCNA 200-301 Official Cert Guide, Volume 1*. When doing the problems, Table E-1, earlier in this appendix, which lists all possible DDN mask values, can be useful.

Each row of Table E-2 lists an IP address and mask. For each row, complete the table. Note that for the purposes of this exercise you can assume that the two special subnets in each network, the zero subnet and broadcast subnet, are allowed to be used.

Table E-2 Mask Analysis Problems

Problem Number	Problem	Network Bits	Subnet Bits	Host Bits	Number of Subnets in Network	Number of Hosts per Subnet
1	10.66.5.99, 255.255.254.0					
2	172.16.203.42, 255.255.252.0					
3	192.168.55.55, 255.255.255.224					
4	10.22.55.87/30					
5	172.30.40.166/26					
6	192.168.203.18/29					
7	200.11.88.211, 255.255.255.240					
8	128.1.211.33, 255.255.255.128					
9	9.211.45.65/21					
10	223.224.225.226/25					

Answers to Mask Analysis Problems

Table E-3 includes the answers to problems 1–10. The paragraphs following the table provide the explanations of each answer.

Table E-3 Answers to Problems in This Appendix

Problem Number	Problem	Network Bits	Subnet Bits	Host Bits	Number of Subnets in Network	Number of Hosts per Subnet
1	10.66.5.99, 255.255.254.0	8	15	9	$2^{15} = 32{,}768$	$2^9 – 2 = 510$
2	172.16.203.42, 255.255.252.0	16	6	10	$2^6 = 64$	$2^{10} – 2 = 1022$
3	192.168.55.55, 255.255.255.224	24	3	5	$2^3 = 8$	$2^5 – 2 = 30$
4	10.22.55.87/30	8	22	2	$2^{22} = 4{,}194{,}304$	$2^2 – 2 = 2$
5	172.30.40.166/26	16	10	6	$2^{10} = 1024$	$2^6 – 2 = 62$
6	192.168.203.18/29	24	5	3	$2^5 = 32$	$2^3 – 2 = 6$
7	200.11.88.211, 255.255.255.240	24	4	4	$2^4 = 16$	$2^4 – 2 = 14$
8	128.1.211.33, 255.255.255.128	16	9	7	$2^9 = 512$	$2^7 – 2 = 126$
9	9.211.45.65/21	8	13	11	$2^{13} = 8192$	$2^{11} – 2 = 2046$
10	223.224.225.226/25	24	1	7	$2^1 = 2$	$2^7 – 2 = 126$

Mask Analysis Problem 1: Answer

Address 10.66.5.99 is in Class A network 10.0.0.0, meaning that 8 network bits exist. Mask 255.255.254.0 converts to prefix /23, because the first 2 octets of value 255 represent 8 binary 1s, and the 254 in the third octet represents 7 binary 1s, for a total of 23 binary 1s. Therefore, the number of host bits is 32 – 23 = 9, leaving 15 subnet bits (32 – 8 network bits – 9 host bits = 15 subnet bits). The number of subnets in this Class A network, using mask 255.255.254.0, is $2^{15} = 32{,}768$. The number of hosts per subnet is $2^9 – 2 = 510$.

Mask Analysis Problem 2: Answer

Address 172.16.203.42, mask 255.255.252.0, is in Class B network 172.16.0.0, meaning that 16 network bits exist. Mask 255.255.252.0 converts to prefix /22, because the first 2 octets of value 255 represent 8 binary 1s, and the 252 in the third octet represents 6 binary 1s, for a total of 22 binary 1s. Therefore, the number of host bits is 32 – 22 = 10, leaving 6 subnet bits (32 – 16 network bits – 10 host bits = 6 subnet bits). The number of subnets in this Class B network, using mask 255.255.252.0, is $2^6 = 64$. The number of hosts per subnet is $2^{10} – 2 = 1022$.

Mask Analysis Problem 3: Answer

Address 192.168.55.55 is in Class C network 192.168.55.0, meaning that 24 network bits exist. Mask 255.255.255.224 converts to prefix /27, because the first 3 octets of value 255 represent 8 binary 1s, and the 224 in the fourth octet represents 3 binary 1s, for a total of 27 binary 1s. Therefore, the number of host bits is 32 – 27 = 5, leaving 3 subnet bits (32 – 24 network bits – 5 host bits = 3 subnet bits). The number of subnets in this Class C network, using mask 255.255.255.224, is 2^3 = 8. The number of hosts per subnet is 2^5 – 2 = 30.

Mask Analysis Problem 4: Answer

Address 10.22.55.87 is in Class A network 10.0.0.0, meaning that 8 network bits exist. The prefix format mask of /30 lets you calculate the number of host bits as 32 – prefix length (in this case, 32 – 30 = 2). This leaves 22 subnet bits (32 – 8 network bits – 2 host bits = 22 subnet bits). The number of subnets in this Class A network, using mask 255.255.255.252, is 2^{22} = 4,194,304. The number of hosts per subnet is 2^2 – 2 = 2. (Note that this mask is popularly used on serial links, which need only two IP addresses in a subnet.)

Mask Analysis Problem 5: Answer

Address 172.30.40.166 is in Class B network 172.30.0.0, meaning that 16 network bits exist. The prefix format mask of /26 lets you calculate the number of host bits as 32 – prefix length (in this case, 32 – 26 = 6). This leaves 10 subnet bits (32 – 16 network bits – 6 host bits = 10 subnet bits). The number of subnets in this Class B network, using mask /26, is 2^{10} = 1024. The number of hosts per subnet is 2^6 – 2 = 62.

Mask Analysis Problem 6: Answer

Address 192.168.203.18 is in Class C network 192.168.203.0, meaning that 24 network bits exist. The prefix format mask of /29 lets you calculate the number of host bits as 32 – prefix length (in this case, 32 – 29 = 3). This leaves 5 subnet bits, because 32 – 24 network bits – 3 host bits = 5 subnet bits. The number of subnets in this Class C network, using mask /29, is 2^5 = 32. The number of hosts per subnet is 2^3 – 2 = 6.

Mask Analysis Problem 7: Answer

Address 200.11.88.211 is in Class C network 200.11.88.0, meaning that 24 network bits exist. Mask 255.255.255.240 converts to prefix /28, because the first three octets of value 255 represent 8 binary 1s, and the 240 in the fourth octet represents 4 binary 1s, for a total of 28 binary 1s. This leaves 4 subnet bits (32 – 24 network bits – 4 host bits = 4 subnet bits). The number of subnets in this Class C network, using mask /28, is 2^4 = 16. The number of hosts per subnet is 2^4 – 2 = 14.

Mask Analysis Problem 8: Answer

Address 128.1.211.33, mask 255.255.255.128, is in Class B network 128.1.0.0, meaning that 16 network bits exist. Mask 255.255.255.128 converts to prefix /25, because the first 3 octets of value 255 represent 8 binary 1s, and the 128 in the fourth octet represents 1 binary 1, for a total of 25 binary 1s. Therefore, the number of host bits is 32 – 25 = 7, leaving 9 subnet bits (32 – 16 network bits – 7 host bits = 9 subnet bits). The number of subnets in this Class B network, using mask 255.255.255.128, is 2^9 = 512. The number of hosts per subnet is 2^7 – 2 = 126.

Mask Analysis Problem 9: Answer

Address 9.211.45.65 is in Class A network 10.0.0.0, meaning that 8 network bits exist. The prefix format mask of /21 lets you calculate the number of host bits as 32 – prefix length (in this case, 32 – 21 = 11). This leaves 13 subnet bits (32 – 8 network bits – 11 host bits = 13 subnet bits). The number of subnets in this Class A network, using mask /21, is 2^{13} = 8192. The number of hosts per subnet is 2^{11} – 2 = 2046.

Mask Analysis Problem 10: Answer

Address 223.224.225.226 is in Class C network 223.224.225.0, meaning that 24 network bits exist. The prefix format mask of /25 lets you calculate the number of host bits as 32 – prefix length (in this case, 32 – 25 = 7). This leaves 1 subnet bit (32 – 24 network bits – 7 host bits = 1 subnet bit). The number of subnets in this Class C network, using mask /25, is 2^1 = 2. The number of hosts per subnet is 2^7 – 2 = 126.

APPENDIX F

Practice for Chapter 14: Analyzing Existing Subnets

Practice Problems

This appendix lists practice problems related to Chapter 14, "Analyzing Existing Subnets." Each problem asks you to find a variety of information about the subnet in which an IP address resides. Each problem supplies an IP address and a subnet mask, from which you should find the following information:

- Subnet number
- Subnet broadcast address
- Range of valid IP addresses in this network

To find these facts, you can use any of the processes explained in Chapter 14.

In addition, these same problems can be used to review the concepts in Chapter 13, "Analyzing Subnet Masks." To use these same problems for practice related to Chapter 13, simply find the following information for each of the problems:

- Size of the network part of the address
- Size of the subnet part of the address
- Size of the host part of the address
- Number of hosts per subnet
- Number of subnets in this network

Feel free to either ignore or use the opportunity for more practice related to analyzing subnet masks.

Solve for the following problems:

1. 10.180.10.18, mask 255.192.0.0
2. 10.200.10.18, mask 255.224.0.0
3. 10.100.18.18, mask 255.240.0.0
4. 10.100.18.18, mask 255.248.0.0
5. 10.150.200.200, mask 255.252.0.0
6. 10.150.200.200, mask 255.254.0.0
7. 10.220.100.18, mask 255.255.0.0
8. 10.220.100.18, mask 255.255.128.0
9. 172.31.100.100, mask 255.255.192.0

10. 172.31.100.100, mask 255.255.224.0
11. 172.31.200.10, mask 255.255.240.0
12. 172.31.200.10, mask 255.255.248.0
13. 172.31.50.50, mask 255.255.252.0
14. 172.31.50.50, mask 255.255.254.0
15. 172.31.140.14, mask 255.255.255.0
16. 172.31.140.14, mask 255.255.255.128
17. 192.168.15.150, mask 255.255.255.192
18. 192.168.15.150, mask 255.255.255.224
19. 192.168.100.100, mask 255.255.255.240
20. 192.168.100.100, mask 255.255.255.248
21. 192.168.15.230, mask 255.255.255.252
22. 10.1.1.1, mask 255.248.0.0
23. 172.16.1.200, mask 255.255.240.0
24. 172.16.0.200, mask 255.255.255.192
25. 10.1.1.1, mask 255.0.0.0

Answers

This section includes the answers to the 25 problems listed in this appendix. The answer section for each problem explains how to use the process outlined in Chapter 14 to find the answers. Also, refer to Chapter 13 for details on how to find information about analyzing the subnet mask.

Answer to Problem 1

The answers begin with the analysis of the three parts of the address, the number of hosts per subnet, and the number of subnets of this network using the stated mask, as outlined in Table F-1. The binary math for subnet and broadcast address calculation follows. The answer finishes with the easier mental calculations for the range of IP addresses in the subnet.

Table F-1 Question 1: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.180.10.18	—
Mask	255.192.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	22	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{22} - 2 = 4{,}194{,}302$	$2^{\text{number-of-host-bits}} - 2$

Table F-2 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-2 Question 1: Binary Calculation of Subnet and Broadcast Addresses

Address	10.180.10.18	`00001010 10`**`110100 00001010 00010010`**
Mask	255.192.0.0	`11111111 11`**`000000 00000000 00000000`**
AND result (subnet number)	10.128.0.0	`00001010 10`**`000000 00000000 00000000`**
Change host to 1s (broadcast address)	10.191.255.255	`00001010 10`**`111111 11111111 11111111`**

To get the first valid IP address, just add 1 to the subnet number; to get the last valid IP address, just subtract 1 from the broadcast address. In this case:

10.128.0.1 through 10.191.255.254

10.128.0.0 + 1 = 10.128.0.1

10.191.255.255 – 1 = 10.191.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. The key parts of the process are as follows:

- The interesting octet is the octet for which the mask's value is not a decimal 0 or 255.
- The magic number is calculated as the value of the IP address's interesting octet, subtracted from 256.
- The subnet number can be found by copying the IP address octets to the left of the interesting octet, by writing down 0s for octets to the right of the interesting octet, and by finding the multiple of the magic number closest to, but not larger than, the IP address's value in that same octet.
- The broadcast address can be similarly found by copying the subnet number's octets to the left of the interesting octet, by writing 255s for octets to the right of the interesting octet, and by taking the subnet number's value in the interesting octet, adding the magic number, and subtracting 1.

Table F-3 shows the work for this problem, with some explanation of the work following the table. Refer to Chapter 14 for the detailed processes.

Table F-3 Question 1: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	192	0	0	
Address	10	180	10	18	
Subnet Number	10	128	0	0	Magic number = 256 – 192 = 64
First Address	10	128	0	1	Add 1 to last octet of subnet
Last Address	10	191	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	191	255	255	128 + 64 – 1 = 191

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that is closest to 180 but not higher than 180. So, the second octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 128 + 64 – 1 = 191.

Answer to Problem 2

Table F-4 Question 2: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.200.10.18	—
Mask	255.224.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	21	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	2^3 = 8	$2^{\text{number-of-subnet-bits}}$
Number of hosts	2^{21} – 2 = 2,097,150	$2^{\text{number-of-host-bits}}$ – 2

Table F-5 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-5 Question 2: Binary Calculation of Subnet and Broadcast Addresses

Address	10.200.10.18	00001010 110**01000 00001010 00010010**
Mask	255.224.0.0	11111111 111**00000 00000000 00000000**
AND result (subnet number)	10.192.0.0	00001010 110**00000 00000000 00000000**
Change host to 1s (broadcast address)	10.223.255.255	00001010 110**11111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.192.0.1 through 10.223.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-6 shows the work for this problem, with some explanation of the work following the table.

Table F-6 Question 2: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	224	0	0	
Address	10	200	10	18	
Subnet Number	10	192	0	0	Magic number = 256 – 224 = 32
First Address	10	192	0	1	Add 1 to last octet of subnet
Last Address	10	223	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	223	255	255	192 + 32 – 1 = 223

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 32 that is closest to 200 but not higher than 200. So, the second octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 192 + 32 – 1 = 223.

Answer to Problem 3

Table F-7 Question 3: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.100.18.18	—
Mask	255.240.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	20	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{20} - 2 = 1{,}048{,}574$	$2^{\text{number-of-host-bits}} - 2$

Table F-8 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-8 Question 3: Binary Calculation of Subnet and Broadcast Addresses

Address	10.100.18.18	00001010 0110**0100 00010010 00010010**
Mask	255.240.0.0	11111111 1111**0000 00000000 00000000**
AND result (subnet number)	10.96.0.0	00001010 0110**0000 00000000 00000000**
Change host to 1s (broadcast address)	10.111.255.255	00001010 0110**1111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.111.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-9 shows the work for this problem, with some explanation of the work following the table.

Table F-9 Question 3: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	240	0	0	—
Address	10	100	18	18	—
Subnet Number	10	96	0	0	Magic number = 256 – 240 = 16
First Address	10	96	0	1	Add 1 to last octet of subnet
Last Address	10	111	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	111	255	255	96 + 16 – 1 = 111

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that is closest to 100 but not higher than 100. So, the second octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 96 + 16 – 1 = 111.

Answer to Problem 4

Table F-10 Question 4: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.100.18.18	—
Mask	255.248.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{19} - 2 = 524{,}286$	$2^{\text{number-of-host-bits}} - 2$

Table F-11 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-11 Question 4: Binary Calculation of Subnet and Broadcast Addresses

Address	10.100.18.18	00001010 01100**100 00010010 00010010**
Mask	255.248.0.0	11111111 11111**000 00000000 00000000**
AND result (subnet number)	10.96.0.0	00001010 01100**000 00000000 00000000**
Change host to 1s (broadcast address)	10.103.255.255	00001010 01100**111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.103.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-12 shows the work for this problem, with some explanation of the work following the table.

Table F-12 Question 4: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	248	0	0	—
Address	10	100	18	18	—
Subnet Number	10	96	0	0	Magic number = 256 – 248 = 8
First Address	10	96	0	1	Add 1 to last octet of subnet
Last Address	10	103	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	103	255	255	96 + 8 – 1 = 103

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that is closest to 100 but not higher than 100. So, the second octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 96 + 8 – 1 = 103.

Answer to Problem 5

Table F-13 Question 5: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.150.200.200	—
Mask	255.252.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	18	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{18} - 2 = 262{,}142$	$2^{\text{number-of-host-bits}} - 2$

Table F-14 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-14 Question 5: Binary Calculation of Subnet and Broadcast Addresses

Address	10.150.200.200	`00001010 100101`**`10 11001000 11001000`**
Mask	255.252.0.0	`11111111 111111`**`00 00000000 00000000`**
AND result (subnet number)	10.148.0.0	`00001010 100101`**`00 00000000 00000000`**
Change host to 1s (broadcast address)	10.151.255.255	`00001010 100101`**`11 11111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.148.0.1 through 10.151.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-15 shows the work for this problem, with some explanation of the work following the table.

Table F-15 Question 5: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Mask	255	252	0	0	—
Address	10	150	200	200	—
Subnet Number	10	148	0	0	Magic number = 256 – 252 = 4
First Address	10	148	0	1	Add 1 to last octet of subnet
Last Address	10	151	255	254	Subtract 1 from last octet of broadcast
Broadcast	10	151	255	255	148 + 4 – 1 = 151

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 148 is the multiple of 4 that is closest to 150 but not higher than 150. So, the second octet of the subnet number is 148.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 148 + 4 – 1 = 151.

Answer to Problem 6

Table F-16 Question 6: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.150.200.200	—
Mask	255.254.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	17	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 = 128$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{17} - 2 = 131{,}070$	$2^{\text{number-of-host-bits}} - 2$

Table F-17 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-17 Question 6: Binary Calculation of Subnet and Broadcast Addresses

Address	10.150.200.200	00001010 10010110 **11001000 11001000**
Mask	255.254.0.0	11111111 11111110 **00000000 00000000**
AND result (subnet number)	10.150.0.0	00001010 10010110 **00000000 00000000**
Change host to 1s (broadcast address)	10.151.255.255	00001010 10010111 **11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.150.0.1 through 10.151.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-18 shows the work for this problem, with some explanation of the work following the table.

Table F-18 Question 6: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	254	0	0
Address	10	150	200	200
Subnet Number	10	150	0	0
First Valid Address	10	150	0	1
Last Valid Address	10	151	255	254
Broadcast	10	151	255	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 150 is the multiple of 2 that is closest to 150 but not higher than 150. So, the second octet of the subnet number is 150.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 150 + 2 – 1 = 151.

Answer to Problem 7

Table F-19 Question 7: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.220.100.18	—
Mask	255.255.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	16	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 = 256$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{16} - 2 = 65{,}534$	$2^{\text{number-of-host-bits}} - 2$

Table F-20 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-20 Question 7: Binary Calculation of Subnet and Broadcast Addresses

Address	10.220.100.18	00001010 11011100 **01100100 00010010**
Mask	255.255.0.0	11111111 11111111 **00000000 00000000**
AND result (subnet number)	10.220.0.0	00001010 11011100 **00000000 00000000**
Change host to 1s (broadcast address)	10.220.255.255	00001010 11011100 **11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-21 shows the work for this problem.

Table F-21 Question 7: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	0	0
Address	10	220	100	18
Subnet Number	10	220	0	0
First Valid Address	10	220	0	1
Last Valid Address	10	220	255	254
Broadcast	10	220	255	255

This subnetting scheme uses an easy mask because all the octets are a 0 or a 255. No math tricks are needed.

Answer to Problem 8

Table F-22 Question 8: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.220.100.18	—
Mask	255.255.128.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	15	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 = 512$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{15} - 2 = 32,766$	$2^{\text{number-of-host-bits}} - 2$

Table F-23 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-23 Question 8: Binary Calculation of Subnet and Broadcast Addresses

Address	10.220.100.18	00001010 11011100 0**1100100** **00010010**
Mask	255.255.128.0	11111111 11111111 1**0000000** **00000000**
AND result (subnet number)	10.220.0.0	00001010 11011100 0**0000000** **00000000**
Change host to 1s (broadcast address)	10.220.127.255	00001010 11011100 0**1111111** **11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.127.254

Table F-24 shows the work for this problem, with some explanation of the work following the table. Refer to Chapter 14 for the detailed processes.

Table F-24 Question 8: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	128	0
Address	10	220	100	18
Subnet Number	10	220	0	0
First Address	10	220	0	1
Last Address	10	220	127	254
Broadcast	10	220	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 0 + 128 – 1 = 127.

This example tends to confuse people, because a mask with 128 in it gives you subnet numbers that just do not seem to look right. Table F-25 gives you the answers for the first several subnets, just to make sure that you are clear about the subnets when using this mask with a Class A network.

Table F-25 Question 8: First Four Subnets

	Zero Subnet	2nd Subnet	3rd Subnet	4th Subnet
Subnet	10.0.0.0	10.0.128.0	10.1.0.0	10.1.128.0
First Address	10.0.0.1	10.0.128.1	10.1.0.1	10.1.128.1
Last Address	10.0.127.254	10.0.255.254	10.1.127.254	10.1.255.254
Broadcast	10.0.127.255	10.0.255.255	10.1.127.255	10.1.255.255

Answer to Problem 9

Table F-26 Question 9: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.100.100	—
Mask	255.255.192.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	14	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{14} - 2 = 16,382$	$2^{\text{number-of-host-bits}} - 2$

Table F-27 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-27 Question 9: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.100.100	`10101100 00011111 01`**`100100 01100100`**
Mask	255.255.192.0	`11111111 11111111 11`**`000000 00000000`**
AND result (subnet number)	172.31.64.0	`10101100 00011111 01`**`000000 00000000`**
Change host to 1s (broadcast address)	172.31.127.255	`10101100 00011111 01`**`111111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.64.1 through 172.31.127.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-28 shows the work for this problem, with some explanation of the work following the table.

Table F-28 Question 9: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	192	0
Address	172	31	100	100
Subnet Number	172	31	64	0
First Valid Address	172	31	64	1
Last Valid Address	172	31	127	254
Broadcast	172	31	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 64 is the multiple of 64 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 64.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, it is 64 + 64 – 1 = 127.

Answer to Problem 10

Table F-29 Question 10: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.100.100	—
Mask	255.255.224.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	13	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 = 8$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{13} - 2 = 8190$	$2^{\text{number-of-host-bits}} - 2$

Table F-30 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-30 Question 10: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.100.100	10101100 00011111 011**00100 01100100**
Mask	255.255.224.0	11111111 11111111 111**00000 00000000**
AND result (subnet number)	172.31.96.0	10101100 00011111 011**00000 00000000**
Change host to 1s (broadcast address)	172.31.127.255	10101100 00011111 011**11111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.96.1 through 172.31.127.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-31 shows the work for this problem, with some explanation of the work following the table.

Table F-31 Question 10: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	224	0
Address	172	31	100	100
Subnet Number	172	31	96	0
First Valid Address	172	31	96	1
Last Valid Address	172	31	127	254
Broadcast	172	31	127	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 32 that is closest to 100 but not higher than 100. So, the third octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky parts, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 96 + 32 – 1 = 127.

Answer to Problem 11

Table F-32 Question 11: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.200.10	—
Mask	255.255.240.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table F-33 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-33 Question 11: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.200.10	`10101100 00011111 1100`**`1000 00001010`**
Mask	255.255.240.0	`11111111 11111111 1111`**`0000 00000000`**
AND result (subnet number)	172.31.192.0	`10101100 00011111 1100`**`0000 00000000`**
Change host to 1s (broadcast address)	172.31.207.255	`10101100 00011111 1100`**`1111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.192.1 through 172.31.207.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-34 shows the work for this problem, with some explanation of the work following the table.

Table F-34 Question 11: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	240	0
Address	172	31	200	10
Subnet Number	172	31	192	0
First Valid Address	172	31	192	1
Last Valid Address	172	31	207	254
Broadcast	172	31	207	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 16 that is closest to 200 but not higher than 200. So, the third octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 192 + 16 – 1 = 207.

Answer to Problem 12

Table F-35 Question 12: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.200.10	—
Mask	255.255.248.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	11	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{11} - 2 = 2046$	$2^{\text{number-of-host-bits}} - 2$

Table F-36 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-36 Question 12: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.200.10	`10101100 00011111 11001`**`000 00001010`**
Mask	255.255.248.0	`11111111 11111111 11111`**`000 00000000`**
AND result (subnet number)	172.31.200.0	`10101100 00011111 11001`**`000 00000000`**
Change host to 1s (broadcast address)	172.31.207.255	`10101100 00011111 11001`**`111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.200.1 through 172.31.207.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-37 shows the work for this problem, with some explanation of the work following the table.

Table F-37 Question 12: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	248	0
Address	172	31	200	10
Subnet Number	172	31	200	0
First Valid Address	172	31	200	1
Last Valid Address	172	31	207	254
Broadcast	172	31	207	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 200 is the multiple of 8 that is closest to 200 but not higher than 200. So, the third octet of the subnet number is 200.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 200 + 8 – 1 = 207.

Answer to Problem 13

Table F-38 Question 13: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.50.50	—
Mask	255.255.252.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	10	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{10} - 2 = 1022$	$2^{\text{number-of-host-bits}} - 2$

Table F-39 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-39 Question 13: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.50.50	10101100 00011111 001100**10 00110010**
Mask	255.255.252.0	11111111 11111111 111111**00 00000000**
AND result (subnet number)	172.31.48.0	10101100 00011111 001100**00 00000000**
Change host to 1s (broadcast address)	172.31.51.255	10101100 00011111 001100**11 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.48.1 through 172.31.51.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-40 shows the work for this problem, with some explanation of the work following the table.

Table F-40 Question 13: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	252	0
Address	172	31	50	50
Subnet Number	172	31	48	0
First Valid Address	172	31	48	1
Last Valid Address	172	31	51	254
Broadcast	172	31	51	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 48 is the multiple of 4 that is closest to 50 but not higher than 50. So, the third octet of the subnet number is 48.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 48 + 4 – 1 = 51.

Answer to Problem 14

Table F-41 Question 14: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.50.50	—
Mask	255.255.254.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	9	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	2^7 = 128	$2^{\text{number-of-subnet-bits}}$
Number of hosts	2^9 – 2 = 510	$2^{\text{number-of-host-bits}}$ – 2

Table F-42 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-42 Question 14: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.50.50	10101100 00011111 0011001**0 00110010**
Mask	255.255.254.0	11111111 11111111 1111111**0** 00000000
AND result (subnet number)	172.31.50.0	10101100 00011111 0011001**0 00000000**
Change host to 1s (broadcast address)	172.31.51.255	10101100 00011111 0011001**1 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.50.1 through 172.31.51.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-43 shows the work for this problem, with some explanation of the work following the table.

Table F-43 Question 14: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	254	0
Address	172	31	50	50
Subnet Number	172	31	50	0
First Valid Address	172	31	50	1
Last Valid Address	172	31	51	254
Broadcast	172	31	51	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 50 is the multiple of 2 that is closest to 50 but not higher than 50. So, the third octet of the subnet number is 50.

The second part of this process calculates the subnet broadcast address with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 50 + 2 – 1 = 51.

Answer to Problem 15

Table F-44 Question 15: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.140.14	—
Mask	255.255.255.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	8	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 = 256$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^8 - 2 = 254$	$2^{\text{number-of-host-bits}} - 2$

Table F-45 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-45 Question 15: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.140.14	10101100 00011111 10001100 **00001110**
Mask	255.255.255.0	11111111 11111111 11111111 **00000000**
AND result (subnet number)	172.31.140.0	10101100 00011111 10001100 **00000000**
Change host to 1s (broadcast address)	172.31.140.255	10101100 00011111 10001100 **11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-46 shows the work for this problem.

Table F-46 Question 15: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	0
Address	172	31	140	14
Subnet Number	172	31	140	0
First Valid Address	172	31	140	1
Last Valid Address	172	31	140	254
Broadcast	172	31	140	255

This subnetting scheme uses an easy mask because all the octets are a 0 or a 255. No math tricks are needed.

Answer to Problem 16

Table F-47 Question 16: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.31.140.14	—
Mask	255.255.255.128	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	7	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 = 512$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^7 - 2 = 126$	$2^{\text{number-of-host-bits}} - 2$

Table F-48 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-48 Question 16: Binary Calculation of Subnet and Broadcast Addresses

Address	172.31.140.14	10101100 00011111 10001100 0**0001110**
Mask	255.255.255.128	11111111 11111111 11111111 1**0000000**
AND result (subnet number)	172.31.140.0	10101100 00011111 10001100 0**0000000**
Change host to 1s (broadcast address)	172.31.140.127	10101100 00011111 10001100 0**1111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.126

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-49 shows the work for this problem, with some explanation of the work following the table.

Table F-49 Question 16: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	128
Address	172	31	140	14
Subnet Number	172	31	140	0
First Valid Address	172	31	140	1
Last Valid Address	172	31	140	126
Broadcast	172	31	140	127

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 0 is the multiple of 128 that is closest to 14 but not higher than 14. So, the fourth octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, it is 0 + 128 – 1 = 127.

Answer to Problem 17

Table F-50 Question 17: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	192.168.15.150	—
Mask	255.255.255.192	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 = 4$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table F-51 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-51 Question 17: Binary Calculation of Subnet and Broadcast Addresses

Address	192.168.15.150	11000000 10101000 00001111 10**010110**
Mask	255.255.255.192	11111111 11111111 11111111 11**000000**
AND result (subnet number)	192.168.15.128	11000000 10101000 00001111 10**000000**
Change host to 1s (broadcast address)	192.168.15.191	11000000 10101000 00001111 10**111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.190

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-52 shows the work for this problem, with some explanation of the work following the table.

Table F-52 Question 17: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	192
Address	192	168	15	150
Subnet Number	192	168	15	128
First Valid Address	192	168	15	129
Last Valid Address	192	168	15	190
Broadcast	192	168	15	191

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that is closest to 150 but not higher than 150. So, the fourth octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 128 + 64 – 1 = 191.

Answer to Problem 18

Table F-53 Question 18: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	192.168.15.150	—
Mask	255.255.255.224	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	5	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 = 8$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^5 - 2 = 30$	$2^{\text{number-of-host-bits}} - 2$

Table F-54 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-54 Question 18: Binary Calculation of Subnet and Broadcast Addresses

Address	192.168.15.150	11000000 10101000 00001111 100**10110**
Mask	255.255.255.224	11111111 11111111 11111111 111**00000**
AND result (subnet number)	192.168.15.128	11000000 10101000 00001111 100**00000**
Change host to 1s (broadcast address)	192.168.15.159	11000000 10101000 00001111 100**11111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.158

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-55 shows the work for this problem, with some explanation of the work following the table.

Table F-55 Question 18: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	224
Address	192	168	15	150
Subnet Number	192	168	15	128
First Valid Address	192	168	15	129
Last Valid Address	192	168	15	158
Broadcast	192	168	15	159

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 128 is the multiple of 32 that is closest to 150 but not higher than 150. So, the fourth octet of the subnet number is 128.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, it is 128 + 32 – 1 = 159.

Answer to Problem 19

Table F-56 Question 19: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	192.168.100.100	—
Mask	255.255.255.240	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	4	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^4 - 2 = 14$	$2^{\text{number-of-host-bits}} - 2$

Table F-57 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-57 Question 19: Binary Calculation of Subnet and Broadcast Addresses

Address	192.168.100.100	11000000 10101000 01100100 0110**0100**
Mask	255.255.255.240	11111111 11111111 11111111 1111**0000**
AND result (subnet number)	192.168.100.96	11000000 10101000 01100100 0110**0000**
Change host to 1s (broadcast address)	192.168.100.111	11000000 10101000 01100100 0110**1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.110

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-58 shows the work for this problem, with some explanation of the work following the table.

Table F-58 Question 19: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	240
Address	192	168	100	100
Subnet Number	192	168	100	96
First Valid Address	192	168	100	97
Last Valid Address	192	168	100	110
Broadcast	192	168	100	111

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that is closest to 100 but not higher than 100. So, the fourth octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 96 + 16 – 1 = 111.

Answer to Problem 20

Table F-59 Question 20: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	192.168.100.100	—
Mask	255.255.255.248	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	3	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^3 - 2 = 6$	$2^{\text{number-of-host-bits}} - 2$

Table F-60 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-60 Question 20: Binary Calculation of Subnet and Broadcast Addresses

Address	192.168.100.100	11000000 10101000 01100100 01100**100**
Mask	255.255.255.248	11111111 11111111 11111111 11111**000**
AND result (subnet number)	192.168.100.96	11000000 10101000 01100100 01100**000**
Change host to 1s (broadcast address)	192.168.100.103	11000000 10101000 01100100 01100**111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.102

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-61 shows the work for this problem, with some explanation of the work following the table.

Table F-61 Question 20: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	248
Address	192	168	100	100
Subnet Number	192	168	100	96
First Valid Address	192	168	100	97
Last Valid Address	192	168	100	102
Broadcast	192	168	100	103

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that is closest to 100 but not higher than 100. So, the fourth octet of the subnet number is 96.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 96 + 8 – 1 = 103.

Answer to Problem 21

Table F-62 Question 21: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	192.168.15.230	—
Mask	255.255.255.252	—
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	2	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 = 64$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^2 - 2 = 2$	$2^{\text{number-of-host-bits}} - 2$

Table F-63 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-63 Question 21: Binary Calculation of Subnet and Broadcast Addresses

Address	192.168.15.230	`11000000 10101000 00001111 11100110`
Mask	255.255.255.252	`11111111 11111111 11111111 11111100`
AND result (subnet number)	192.168.15.228	`11000000 10101000 00001111 11100100`
Change host to 1s (broadcast address)	192.168.15.231	`11000000 10101000 00001111 11100111`

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.229 through 192.168.15.230

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-64 shows the work for this problem, with some explanation of the work following the table.

Table F-64 Question 21: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	252
Address	192	168	15	230
Subnet Number	192	168	15	228
First Valid Address	192	168	15	229
Last Valid Address	192	168	15	230
Broadcast	192	168	15	231

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 228 is the multiple of 4 that is closest to 230 but not higher than 230. So, the fourth octet of the subnet number is 228.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 228 + 4 – 1 = 231.

Answer to Problem 22

Table F-65 Question 22: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.1.1.1	—
Mask	255.248.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 = 32$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{19} - 2 = 524{,}286$	$2^{\text{number-of-host-bits}} - 2$

Table F-66 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-66 Question 22: Binary Calculation of Subnet and Broadcast Addresses

Address	10.1.1.1	00001010 00000**001 00000001 00000001**
Mask	255.248.0.0	11111111 11111**000 00000000 00000000**
AND result (subnet number)	10.0.0.0	00001010 00000**000 00000000 00000000**
Change host to 1s (broadcast address)	10.7.255.255	00001010 00000**111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.7.255.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 0000 1010 **0000 0**000 0000 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-67 shows the work for this problem, with some explanation of the work following the table.

Table F-67 Question 22: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	248	0	0
Address	10	1	1	1
Subnet Number	10	0	0	0
First Valid Address	10	0	0	1
Last Valid Address	10	7	255	254
Broadcast	10	7	255	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 8 that is closest to 1 but not higher than 1. So, the second octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address's value in the interesting octet. In this case, it is 0 + 8 – 1 = 7.

Answer to Problem 23

Table F-68 Question 23: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.16.1.200	—
Mask	255.255.240.0	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 = 16$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table F-69 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-69 Question 23: Binary Calculation of Subnet and Broadcast Addresses

Address	172.16.1.200	`10101100 00010000 0000`**`0001 11001000`**
Mask	255.255.240.0	`11111111 11111111 1111`**`0000 00000000`**
AND result (subnet number)	172.16.0.0	`10101100 00010000 0000`**`0000 00000000`**
Change host to 1s (broadcast address)	172.16.15.255	`10101100 00010000 0000`**`1111 11111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.1 through 172.16.15.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 1010 1100 0001 0000 **0000** 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-70 shows the work for this problem, with some explanation of the work following the table.

Table F-70 Question 23: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	240	0
Address	172	16	1	200
Subnet Number	172	16	0	0
First Valid Address	172	16	0	1
Last Valid Address	172	16	15	254
Broadcast	172	16	15	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The third octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 0 is the multiple of 16 that is closest to 1 but not higher than 1. So, the third octet of the subnet number is 0.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, it is 0 + 16 – 1 = 15.

Answer to Problem 24

Table F-71 Question 24: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	172.16.0.200	—
Mask	255.255.255.192	—
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	10	32 – (network size + host size)
Number of subnets	$2^{10} = 1024$	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table F-72 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-72 Question 24: Binary Calculation of Subnet and Broadcast Addresses

Address	172.16.0.200	`10101100 00010000 00000000 11`**`001000`**
Mask	255.255.255.192	`11111111 11111111 11111111 11`**`000000`**
AND result (subnet number)	172.16.0.192	`10101100 00010000 00000000 11`**`000000`**
Change host to 1s (broadcast address)	172.16.0.255	`10101100 00010000 00000000 11`**`111111`**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.193 through 172.16.0.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-73 shows the work for this problem, with some explanation of the work following the table.

Table F-73 Question 24: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	255	255	192
Address	172	16	0	200
Subnet Number	172	16	0	192
First Valid Address	172	16	0	193
Last Valid Address	172	16	0	254
Broadcast	172	16	0	255

This subnetting scheme uses a difficult mask because one of the octets is not a 0 or a 255. The fourth octet is “interesting” in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask’s value in the interesting octet). The subnet number’s value in the interesting octet (inside the box) is the multiple of the magic number that is not higher than the original IP address’s value in the interesting octet. In this case, 192 is the multiple of 64 that is closest to 200 but not higher than 200. So, the fourth octet of the subnet number is 192.

The second part of this process calculates the subnet broadcast address, with the tricky part, as usual, in the “interesting” octet. Take the subnet number’s value in the interesting octet, add the magic number, and subtract 1. That is the broadcast address’s value in the interesting octet. In this case, it is 192 + 64 – 1 = 255.

You can easily forget that the subnet part of this address, when using this mask, actually covers all the third octet as well as 2 bits of the fourth octet. For example, the valid subnet numbers in order are listed here:

172.16.0.0 (zero subnet)

172.16.0.64

172.16.0.128

172.16.0.192

172.16.1.0

172.16.1.64

172.16.1.128

172.16.1.192

172.16.2.0

172.16.2.64

172.16.2.128

172.16.2.192

172.16.3.0

172.16.3.64

172.16.3.128

172.16.3.192

And so on.

Answer to Problem 25

Congratulations! You made it through the extra practice in this appendix! Here is an easy one to complete your review—one with no subnetting at all.

Table F-74 Question 25: Size of Network, Subnet, Host, Number of Subnets, and Number of Hosts

Item	Example	Rules to Remember
Address	10.1.1.1	—
Mask	255.0.0.0	—
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	24	Always defined as number of binary 0s in mask
Number of subnet bits	0	32 – (network size + host size)
Number of subnets	0	$2^{\text{number-of-subnet-bits}}$
Number of hosts	$2^{24} - 2 = 16,777,214$	$2^{\text{number-of-host-bits}} - 2$

Table F-75 contains the important binary calculations for finding the subnet number and subnet broadcast address. To calculate the subnet number, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table F-75 Question 25: Binary Calculation of Subnet and Broadcast Addresses

Address	10.1.1.1	00001010 **00000001 00000001 00000001**
Mask	255.0.0.0	11111111 **00000000 00000000 00000000**
AND result (subnet number)	10.0.0.0	00001010 **00000000 00000000 00000000**
Change host to 1s (broadcast address)	10.255.255.255	00001010 **11111111 11111111 11111111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.255.255.254

Alternatively, you can use the processes that only use decimal math to find the subnet and broadcast address. Table F-76 shows the work for this problem.

Table F-76 Question 25: Subnet, Broadcast, and First and Last Addresses Calculated Using the Subnet Chart

	Octet 1	Octet 2	Octet 3	Octet 4
Mask	255	0	0	0
Address	10	1	1	1
Network Number	10	0	0	0
First Valid Address	10	0	0	1
Last Valid Address	10	255	255	254
Broadcast	10	255	255	255

Practice for Chapter 15: Subnet Design

This appendix exists as two halves to match the two major sections of the chapter. The first half lists mask design problems, and then the answers to those problems. The second half lists problems where you need to find the subnet ID, but with less than 8 subnet bits and with more than 8 subnet bits.

To solve these problems, use the processes explained in Chapter 15 of *CCNA 200-301 Official Cert Guide, Volume 1*, Second Edition, the current edition of this book you are reading.

Mask Design Practice Problems

This section lists problems with a short set of requirements regarding how a particular classful network should be subnetted. The requirements include the classful network, the number of subnets the design must support, and the number of hosts in each subnet. For each problem, supply the following information:

- The minimum number of subnet and host bits needed in the mask to support the design requirements
- The dotted-decimal format mask(s) that meet the requirements
- The mask you would choose if the problem said to maximize the number of subnets
- The mask you would choose if the problem said to maximize the number of hosts per subnet

Also note that you should assume that the two special subnets in each network—the zero subnet and broadcast subnet—are allowed to be used for these questions.

When doing the problems, the information in Table G-1 can be helpful. Note that Appendix A, "Numeric Reference Tables," in the printed book, also includes this table.

Table G-1 Powers of 2

Number of Bits	2^X	Number of Bits	2^X	Number of Bits	2^X	Number of Bits	2^X
1	2	5	32	9	512	13	8192
2	4	6	64	10	1024	14	16,384
3	8	7	128	11	2048	15	32,768
4	16	8	256	12	4096	16	65,536

Find the key facts for these sets of requirements:

1. Network 10.0.0.0, need 50 subnets, need 200 hosts/subnet
2. Network 172.32.0.0, need 125 subnets, need 125 hosts/subnet
3. Network 192.168.44.0, need 15 subnets, need 6 hosts/subnet
4. Network 10.0.0.0, need 300 subnets, need 500 hosts/subnet
5. Network 172.32.0.0, need 500 subnets, need 15 hosts/subnet
6. Network 172.16.0.0, need 2000 subnets, need 2 hosts/subnet

Mask Design Answers

This section includes the answers to the six problems listed in this appendix. The answer section for each problem explains how to use the process outlined in Chapter 15, "Subnet Design," to find the answers.

Answer to Mask Design Problem 1

Problem 1 shows a Class A network, with 8 network bits, with a minimum of 6 subnet bits and 8 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.252.0.0 (maximizes the number of hosts per subnet)
- 255.254.0.0
- 255.255.0.0
- 255.255.128.0
- 255.255.192.0
- 255.255.224.0
- 255.255.240.0
- 255.255.248.0
- 255.255.252.0
- 255.255.254.0
- 255.255.255.0 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

NOTE The following explanation uses step numbers that match the process listed in Chapter 15, but only the steps from that process that apply to this problem. As a result, the step numbers in the explanation are not sequential.

Step 1. The question lists Class A network 10.0.0.0, so there are 8 network bits.

Step 2. The question states that 50 subnets are needed. A mask with 5 subnet bits supplies only 2^5 (32) subnets, but a mask with 6 subnet bits supplies 2^6 (64) subnets. So, the mask needs at least 6 subnet bits.

Step 3. The question states that 200 hosts are needed per subnet. A mask with 7 host bits supplies only $2^7 – 2$ (126) hosts per subnet, but a mask with 8 host bits supplies $2^8 – 2$ (254) hosts per subnet. So, the mask needs at least 8 host bits.

Step 6A. With N=8, a minimum S=6, and a minimum H=8, multiple masks exist. The first mask, with the minimum number of subnet bits, is /14, found by adding N (8) to the minimum value of S (6). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 8. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 8 = /24.

Step 6C. All masks between /14 and /24 also meet the requirements.

Answer to Mask Design Problem 2

Problem 2 shows a Class B network, with 16 network bits, with a minimum of 7 subnet bits and 7 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.254.0 (maximizes the number of hosts/subnet)
- 255.255.255.0
- 255.255.255.128 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1. The question lists Class B network 172.32.0.0, so there are 16 network bits.

Step 2. The question states that 125 subnets are needed. A mask with 6 subnet bits supplies only 2^6 (64) subnets, but a mask with 7 subnet bits supplies 2^7 (128) subnets. So, the mask needs at least 7 subnet bits.

Step 3. The question states that 125 hosts are needed per subnet. A mask with 6 host bits supplies only $2^6 – 2$ (62) hosts per subnet, but a mask with 7 host bits supplies $2^7 – 2$ (126) hosts per subnet. So, the mask needs at least 7 host bits.

Step 6A. With N=16, a minimum S=7, and a minimum H=7, multiple masks exist. The first mask, with the minimum number of subnet bits, is /23, found by adding N (16) to the minimum value of S (7). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 7. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 7 = /25.

Step 6C. All masks between /23 and /25 also meet the requirements (/23, /24, and /25).

Answer to Mask Design Problem 3

Problem 3 shows a Class C network, with 24 network bits, with a minimum of 4 subnet bits and 3 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.240 (maximizes the number of hosts/subnet)
- 255.255.255.248 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1. The question lists Class C network 192.168.44.0, so there are 24 network bits.

Step 2. The question states that 15 subnets are needed. A mask with 3 subnet bits supplies only 2^3 (8) subnets, but a mask with 4 subnet bits supplies 2^4 (16) subnets. So, the mask needs at least 4 subnet bits.

Step 3. The question states that 6 hosts are needed per subnet. A mask with 2 host bits supplies only $2^2 - 2$ (2) hosts per subnet, but a mask with 3 host bits supplies $2^3 - 2$ (6) hosts per subnet. So, the mask needs at least 3 host bits.

Step 6A. With N=24, a minimum S=4, and a minimum H=3, multiple masks exist. The first mask, with the minimum number of subnet bits, is /28, found by adding N (24) to the minimum value of S (4). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 3. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 3 = /29.

Step 6C. Only masks /28 and /29 meet the requirements.

Answer to Mask Design Problem 4

Problem 4 shows a Class A network, with 8 network bits, with a minimum of 9 subnet bits and 9 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.128.0 (maximizes the number of hosts/subnet)
- 255.255.192.0
- 255.255.224.0
- 255.255.240.0
- 255.255.248.0
- 255.255.252.0
- 255.255.254.0 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1. The question lists Class A network 10.0.0.0, so there are 8 network bits.

Step 2. The question states that 300 subnets are needed. A mask with 8 subnet bits supplies only 2^8 (256) subnets, but a mask with 9 subnet bits supplies 2^9 (512) subnets. So, the mask needs at least 9 subnet bits.

Step 3. The question states that 500 hosts are needed per subnet. A mask with 8 host bits supplies only $2^8 - 2$ (254) hosts per subnet, but a mask with 9 host bits supplies $2^9 - 2$ (510) hosts per subnet. So, the mask needs at least 9 host bits.

Step 6A. With N=8, a minimum S=9, and a minimum H=9, multiple masks exist. The first mask, with the minimum number of subnet bits, is /17, found by adding N (8) to the minimum value of S (9). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 9. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 9 = /23.

Step 6C. All masks between /17 and /23 also meet the requirements (/17, /18, /19, /20, /21, /22, /23).

Answer to Mask Design Problem 5

Problem 5 shows a Class B network, with 16 network bits, with a minimum of 9 subnet bits and 5 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.128 (maximizes the number of hosts/subnet)
- 255.255.255.192
- 255.255.255.224 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1. The question lists Class B network 172.32.0.0, so there are 16 network bits.

Step 2. The question states that 500 subnets are needed. A mask with 8 subnet bits supplies only 2^8 (256) subnets, but a mask with 9 subnet bits supplies 2^9 (512) subnets. So, the mask needs at least 9 subnet bits.

Step 3. The question states that 15 hosts are needed per subnet. A mask with 4 host bits supplies only $2^4 - 2$ (14) hosts per subnet, but a mask with 5 host bits supplies $2^5 - 2$ (30) hosts per subnet. So, the mask needs at least 5 host bits.

Step 6A. With N=16, a minimum S=9, and a minimum H=5, multiple masks exist. The first mask, with the minimum number of subnet bits, is /25, found by adding N (16) to the minimum value of S (9). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 5. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 5 = /27.

Step 6C. All masks between /25 and /27 also meet the requirements (/25, /26, /27).

Answer to Mask Design Problem 6

Problem 6 shows a Class B network, with 16 network bits, with a minimum of 11 subnet bits and 2 host bits to meet the required number of subnets and hosts/subnet. The following masks all meet the requirements in this problem, with the masks that maximize the number of hosts/subnet and the number of subnets noted:

- 255.255.255.224 (maximizes the number of hosts/subnet)
- 255.255.255.240
- 255.255.255.248
- 255.255.255.252 (maximizes the number of subnets)

As for the process to find the answers, the following list explains the details:

Step 1. The question lists Class B network 172.16.0.0, so there are 16 network bits.

Step 2. The question states that 2000 subnets are needed. A mask with 10 subnet bits supplies only 2^{10} (1024) subnets, but a mask with 11 subnet bits supplies 2^{11} (2048) subnets. So, the mask needs at least 11 subnet bits.

Step 3. The question states that 2 hosts are needed per subnet. A mask with 2 host bits supplies $2^2 - 2$ (2) hosts per subnet. So, the mask needs at least 2 host bits.

Step 6A. With N=16, a minimum S=11, and a minimum H=2, multiple masks exist. The first mask, with the minimum number of subnet bits, is /27, found by adding N (16) to the minimum value of S (11). This mask maximizes the number of host bits and therefore maximizes the number of hosts/subnet.

Step 6B. The minimum value of H, the number of host bits, is 2. So, the mask with the fewest H bits, maximizing the number of subnets, is 32 – H = 32 – 2 = /30.

Step 6C. All masks between /27 and /30 also meet the requirements (/27, /28, /29, /30).

Practice Finding All Subnet IDs

The remainder of this Chapter 15ists two sets of problems. Both problem sets list an IP network and mask; your job is to list all the subnet IDs for each network/mask combination. The first problem set includes problems that happen to have 8 or fewer subnet bits, and the second problem set includes problems that happen to have more than 8 subnet bits. In particular, for each problem, find the following:

- All subnet numbers
- The subnet that is the zero subnet
- The subnet that is the broadcast subnet

To find this information, you can use the processes explained in Chapter 15.

Find Subnet IDs, Problem Set 1: 8 or Fewer Subnet Bits

The problems, which consist of a classful network and static-length mask, are as follows:

1. 172.32.0.0/22
2. 200.1.2.0/28

3. 10.0.0.0/15
4. 172.20.0.0/24

Find Subnet IDs, Problem Set 2: More Than 8 Subnet Bits

The problems, which consist of a classful network and static-length mask, are as follows:

1. 172.32.0.0/25
2. 10.0.0.0/21

Answers to Find Subnet IDs, Problem Set 1

This section includes the answers to the four problems listed in Problem Set 1.

Problem Set 1, Answer 1: 172.32.0.0/22

The answer is as follows:

- 172.32.0.0 (zero subnet)
- 172.32.4.0
- 172.32.8.0
- 172.32.12.0
- 172.32.16.0
- 172.32.20.0
- 172.32.24.0

  (Skipping many subnets; each new subnet is the same as the previous subnet, after adding 4 to the third octet.)
- 172.32.248.0
- 172.32.252.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information:

- The mask has fewer than 8 subnet bits (6 bits), because the network is a Class B network (16 network bits), and the mask has 22 binary 1s in it—implying 10 host bits and leaving 6 subnet bits.
- The mask in dotted-decimal format is 255.255.252.0. The interesting octet is the third octet because the subnet bits are all in the third octet.
- Each successive subnet number is 4 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 252 = 4.

As a result, in this case, all the subnets begin with 172.32, have a multiple of 4 in the third octet, and end in 0.

Table G-2 shows the results of the various steps of the process, as outlined in Chapter 15.

Table G-2 8 or Fewer Subnet Bits, Question 1: Answer Table

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	252	0
Magic Number (Step 3)			256 – 252 = 4	
Zero Subnet Number (Step 4)	172	32	0	0
Next Subnet (Step 5)	172	32	4	0
Next Subnet (Step 5)	172	32	8	0
Next Subnet (Step 5)	172	32	12	0
Next Subnet (Step 5)	172	32	16	0
(You might need many more such rows.)	172	32	X	0
Next Subnet	172	32	244	0
Next Subnet (Step 5)	172	32	248	0
Broadcast Subnet (Step 6)	172	32	252	0
Out of Range—Stop Process (Step 6)			256	

Problem Set 1, Answer 2: 200.1.2.0/28

The answer is as follows:

- 200.1.2.0 (zero subnet)
- 200.1.2.16
- 200.1.2.32
- 200.1.2.48
- 200.1.2.64
- 200.1.2.80

  (Skipping many subnets; each new subnet is the same as the previous subnet, after adding 16 to the fourth octet.)
- 200.1.2.224
- 200.1.2.240 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has fewer than 8 subnet bits (4 bits), because the network is a Class C network (24 network bits), and the mask has 28 binary 1s in it, which implies 4 host bits and leaves 4 subnet bits.
- The mask in dotted-decimal format is 255.255.255.240. The interesting octet is the fourth octet, because all the subnet bits are in the fourth octet.
- Each successive subnet number is 16 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 240 = 16.

As a result, in this case, all the subnets begin with 200.1.2 and have a multiple of 16 in the fourth octet.

Table G-3 shows the results of the various steps of the process, as outlined in Chapter 15.

Table G-3 Problem Set 1, Question 2: Answer Table

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	240
Magic Number (Step 3)				256 – 240 = 16
Zero Subnet Number (Step 4)	200	1	2	0
Next Subnet (Step 5)	200	1	2	16
Next Subnet (Step 5)	200	1	2	32
Next Subnet (Step 5)	200	1	2	48
(You might need many more such rows.) (Step 5)	200	1	2	X
Next Subnet (Step 5)	200	1	2	224
Broadcast Subnet (Step 6)	200	1	2	240
Out of Range—Stop Process (Step 6)				256

Problem Set 1, Answer 3: 10.0.0.0/15

The answer is as follows:

- 10.0.0.0 (zero subnet)
- 10.2.0.0
- 10.4.0.0
- 10.6.0.0

  (Skipping many subnets; each new subnet is the same as the previous subnet, after adding 2 to the second octet.)
- 10.252.0.0
- 10.254.0.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information:

- The mask has fewer than 8 subnet bits (7 subnet bits), because the network is a Class A network (8 network bits), and the mask has 15 binary 1s in it, which implies 17 host bits and leaves 7 subnet bits.
- The mask in dotted-decimal format is 255.254.0.0. The interesting octet is the second octet, because all the subnet bits exist in the second octet.
- Each successive subnet number is 2 higher than the previous subnet number, in the interesting octet, because the magic number is 256 – 254 = 2.

As a result, in this case, all the subnets begin with 10, have a multiple of 2 in the second octet, and end in 0.0.

Table G-4 shows the results of the various steps of the process, as outlined in Chapter 15.

Table G-4 Problem Set 1, Question 3: Answer Table

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	254	0	0
Magic Number (Step 3)		256 – 254 = 2		
Zero Subnet Number (Step 4)	10	0	0	0
Next Subnet (Step 5)	10	2	0	0
Next Subnet (Step 5)	10	4	0	0
Next Subnet (Step 5)	10	6	0	0
(You might need many more such rows.) (Step 5)	10	X	0	0
Next Subnet (Step 5)	10	252	0	0
Broadcast Subnet (Step 6)	10	254	0	0
Out of Range—Stop Process (Step 6)		256		

Problem Set 1, Answer 4: 172.20.0.0/24

This problem has an 8-bit subnet field, meaning that 2^8, or 256, possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 172.20.0.0 (zero subnet)
- 172.20.1.0
- 172.20.2.0
- 172.20.3.0
- 172.20.4.0

  (Skipping many subnets; each new subnet is the same as the previous subnet, after adding 1 to the third octet.)
- 172.20.252.0
- 172.20.253.0
- 172.20.254.0
- 172.20.255.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information:

- The mask has exactly 8 subnet bits, specifically all bits in the third octet, making the third octet the interesting octet.
- The magic number is 256 – 255 = 1, because the mask's value in the interesting (third) octet is 255.
- Beginning with the network number of 172.20.0.0, which is the same value as the zero subnet, just add the magic number (1) in the interesting octet.

Essentially, you just count by 1 in the third octet until you reach the highest legal number (255). The first subnet, 172.20.0.0, is the zero subnet, and the last subnet, 172.20.255.0, is the broadcast subnet.

Answers to Find Subnet IDs, Problem Set 2

Problem Set 2, Answer 1: 172.32.0.0/25

This problem has a 9-bit subnet field, meaning that 2^9, or 512, possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 172.32.0.0 (zero subnet)
- 172.32.0.128
- 172.32.1.0
- 172.32.1.128
- 172.32.2.0
- 172.32.2.128
- 172.32.3.0
- 172.32.3.128

  (Skipping many subnets; the subnets occur in blocks of two, with either 0 or 128 in the fourth octet, with each successive block being one greater in the third octet.)
- 172.32.254.0
- 172.32.254.128
- 172.32.255.0
- 172.32.255.128 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has more than 8 subnet bits (9 bits), because the network is a Class B network (16 network bits), and the mask has 25 binary 1s in it, which implies 7 host bits and leaves 9 subnet bits.
- Using the terminology in Chapter 15, octet 4 is the *interesting* octet, where the counting occurs based on the magic number. Octet 3 is the “just left” octet, in which the process counts by 1, from 0 to 255.
- The magic number, which will be used to calculate each successive subnet number, is 256 – 128 = 128.

To calculate the first subnet block, use the same six-step process as used in the simpler problems that have 8 or fewer subnet bits. In this case, with only 1 subnet bit in octet 4, only two subnets exist in each subnet block. Table G-5 shows the steps as compared to the six-step process to find the subnets in a subnet block.

Table G-5 Creating the First Subnet Block

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	255	128
Magic Number (Step 3)				256 – 128 = 128
Zero Subnet Number (Step 4)	172	32	0	0
Next Subnet (Step 5)	172	32	0	128
Step 6 Needs to Be Used Here (Sum of 256 in the 4th Octet)	172	32	0	256

The table represents the logic, but to make sure that the answer is clear, the first subnet block includes the following:

172.32.0.0

172.32.0.128

The next major task—to create subnet blocks for all possible values in the "just left" octet—completes the process. Essentially, create 256 blocks like the previous list. The first has a value of 0, in the "just left" octet; the next has a value of 1; the next, a value of 2; and so on, through a block that begins with 172.30.255. Figure G-1 shows the concept.

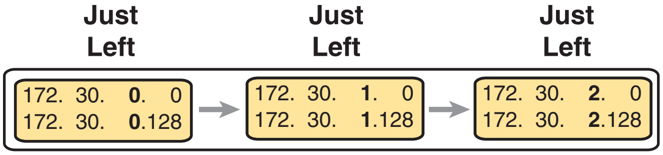


Figure G-1 *Creating Subnet Blocks by Adding 1 in the "Just Left" Octet*

Problem Set 2, Answer 2: 10.0.0.0/21

This problem has a 13-bit subnet field, meaning that 2^{13}, or 8192, possible subnets exist. The following list shows some of the subnets, which should be enough to see the trends in how to find all subnet numbers:

- 10.0.0.0 (zero subnet)
- 10.0.8.0
- 10.0.16.0
- 10.0.24.0

  (Skipping several subnets)
- 10.0.248.0
- 10.1.0.0
- 10.1.8.0
- 10.1.16.0

  (Skipping several subnets)
- 10.1.248.0

- 10.2.0.0
- 10.2.8.0
- 10.2.16.0

(Skipping several subnets)

- 10.255.232.0
- 10.255.240.0
- 10.255.248.0 (broadcast subnet)

The process to find all subnets depends on three key pieces of information, as follows:

- The mask has more than 8 subnet bits (13 bits), because the network is a Class A network (8 network bits), and the mask has 21 binary 1s in it, which implies 11 host bits and leaves 13 subnet bits.
- Using the terminology in Chapter 15, octet 3 is the interesting octet, where the counting occurs based on the magic number. Octet 2 is the "just left" octet, in which the process counts by 1, from 0 to 255.
- The magic number, which will be used to calculate each successive subnet number, is 256 – 248 = 8.

To calculate the first subnet block, use the same six-step process as used in the simpler problems that have 8 or fewer subnet bits. In this case, with 5 subnet bits in octet 3, 32 subnets exist in each subnet block. Table G-6 shows the steps as compared to the six-step process to find the subnets in a subnet block.

Table G-6 Creating the First Subnet Block

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet Mask (Step 1)	255	255	248	0
Magic Number (Step 3)			256 – 248 = 8	
Zero Subnet Number (Step 4)	10	0	0	0
Next Subnet (Step 5)	10	0	8	0
(Skipping several subnets)	10	0	X	0
Next Subnet (Step 5)	10	0	248	0
Step 6 Needs to Be Used Here (Sum of 256 in the 3rd Octet)	10	0	256	0

The table represents the logic, but to make sure that the answer is clear, the first subnet block includes the following:

10.0.0.0

10.0.8.0

10.0.16.0

10.0.24.0

10.0.32.0

10.0.40.0

10.0.48.0

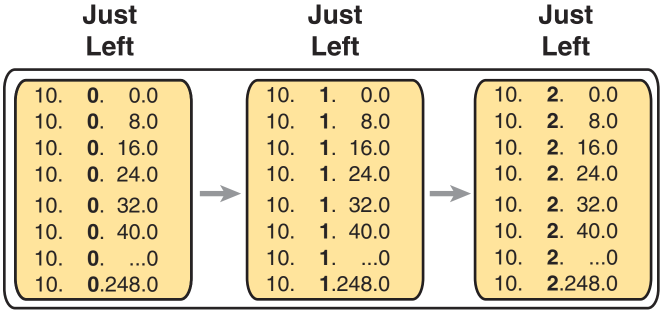
10.0.56.0

10.0.64.0

And so on…

10.0.248.0

The next major task—to create subnet blocks for all possible values in the "just left" octet—completes the process. Essentially, create 256 blocks like the previous list. The first has a value of 0, in the "just left" octet; the next has a value of 1; the next, a value of 2; and so on, through a block that begins with 10.255. Figure G-2 shows the concept.



Just Left	Just Left	Just Left
10. **0**. 0.0	10. **1**. 0.0	10. **2**. 0.0
10. **0**. 8.0	10. **1**. 8.0	10. **2**. 8.0
10. **0**. 16.0	10. **1**. 16.0	10. **2**. 16.0
10. **0**. 24.0	10. **1**. 24.0	10. **2**. 24.0
10. **0**. 32.0	10. **1**. 32.0	10. **2**. 32.0
10. **0**. 40.0	10. **1**. 40.0	10. **2**. 40.0
10. **0**. ...0	10. **1**. ...0	10. **2**. ...0
10. **0**.248.0	10. **1**.248.0	10. **2**.248.0

Figure G-2 *Creating Subnet Blocks by Adding 1 in the "Just Left" Octet*

APPENDIX H

Practice for Chapter 25: Fundamentals of IP Version 6

This appendix provides extra practice problems for two topics discussed in Chapter 25, "Fundamentals of IP Version 6," of the book. The first problems let you convert from a full 32-digit IPv6 address to its abbreviated form, or to do the reverse. The second set of problems begins with IPv6 addresses and prefix lengths, asking you to determine the IPv6 prefix (subnet).

Address Abbreviating and Expanding Problems

Chapter 25 discusses some reasons why you may need to be able to mentally convert from the full 32-digit IPv6 address to the abbreviated form, or vice versa. The practice problems in this section simply provide more opportunities to practice.

Table H-1 lists some practice problems, with the full 32-digit IPv6 address on the left and the best abbreviation on the right. The table gives you either the expanded or abbreviated address, and you need to supply the opposite value. The answers sit at the end of the appendix, in the section "Answers to Address Abbreviating and Expanding Problems."

Table H-1 IPv6 Address Abbreviation and Expansion Practice

	Full	Abbreviation
1	2987:BA11:B011:B00A:1000:0001:F001:F003	
2		3100::1010:D00D:D000:D00B:B00D
3	FD00:0001:0001:0001:0200:00FF:FE00:0001	
4		FDDF:8080:880:1001:0:FF:FE01:507
5	32CC:0000:0000:000D:210F:0000:0000:0000	
6		2100:E:E0::E00
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C	
8		3799:9F9F:F000:0:FFFF::1
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A	
10		3194::1:0:0:101
11	2001:0DB8:0000:0000:0001:0000:0002:0100	
12		2001:DB8::10:A000
13	3330:0000:0000:0100:0000:0002:0000:0003	
14		FD00::1000:2000:0:1:20
15	FD11:1000:0100:0010:0001:0000:1000:0100	
16		2000::2

Calculating the IPv6 Prefix Problems

Routers take the interface IPv6 address configuration and add a connected IPv6 route to the IPv6 routing table, for the IPv6 prefix (subnet) connected to that interface. This section provides some practice problems so that you can do the same math and predict the prefix value that the router will add to the routing table.

Table H-2 lists practice problems that all use the same prefix length (/64), which is the most common prefix length you see. Table H-3 that follows lists additional practice problems, with prefix lengths other than /64.

Table H-2 Finding the IPv6 Prefix When Using a /64 Prefix Length

	Address (Assume a /64 Prefix Length)	Prefix (Subnet)
1	2987:BA11:B011:B00A:1000:0001:F001:F003	
2	3100:0000:0000:1010:D00D:D000:D00B:B00D	
3	FD00:0001:0001:0001:0200:00FF:FE00:0001	
4	FDDF:8080:0880:1001:0000:00FF:FE01:0507	
5	32CC:0000:0000:000D:210F:0000:0000:0000	
6	2100:000E:00E0:0000:0000:0000:0000:0E00	
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C	
8	3799:9F9F:F000:0000:FFFF:0000:0000:0001	
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A	
10	3194:0000:0000:0000:0001:0000:0000:0101	
11	2001:0DB8:0000:0000:0001:0000:0002:0100	
12	2001:0DB8:0000:0000:0000:0000:0010:A000	
13	3330:0000:0000:0100:0000:0002:0000:0003	
14	FD00:0000:0000:1000:2000:0000:0001:0020	
15	FD11:1000:0100:0010:0001:0000:1000:0100	
16	2000:0000:0000:0000:0000:0000:0000:0002	

Table H-3 Finding the IPv6 Prefix Using a Prefix Length Other Than /64

	Address	Prefix (Subnet)
1	2987:BA11:B011:B00A:1000:0001:F001:F003 /60	
2	3100:0000:0000:1010:D00D:D000:D00B:B00D /56	
3	FD00:0001:0001:0001:0200:00FF:FE00:0001 /52	
4	FDDF:8080:0880:1001:0000:00FF:FE01:0507 /48	
5	32CC:0000:0000:000D:210F:0000:0000:0000 /44	
6	2100:000E:00E0:0000:0000:0000:0000:0E00 /60	
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C /56	
8	3799:9F9F:F000:0000:FFFF:0000:0000:0001 /52	
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A /48	
10	3194:0000:0000:0000:0001:0000:0000:0101 /44	

Answers to Address Abbreviating and Expanding Problems

Table H-4 lists the answers to the problems listed earlier in Table H-1.

Table H-4 Answers: IPv6 Address Abbreviation and Expansion Practice

	Full	Abbreviation
1	2987:BA11:B011:B00A:1000:0001:F001:F003	2987:BA11:B011:B00A:1000:1:F001:F003
2	3100:0000:0000:1010:D00D:D000:D00B:B00D	3100::1010:D00D:D000:D00B:B00D
3	FD00:0001:0001:0001:0200:00FF:FE00:0001	FD00:1:1:1:200:FF:FE00:1
4	FDDF:8080:0880:1001:0000:00FF:FE01:0507	FDDF:8080:880:1001:0:FF:FE01:507
5	32CC:0000:0000:000D:210F:0000:0000:0000	32CC:0:0:D:210F::
6	2100:000E:00E0:0000:0000:0000:0000:0E00	2100:E:E0::E00
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C	3A11:CA00::FF:FECC:C
8	3799:9F9F:F000:0000:FFFF:0000:0000:0001	3799:9F9F:F000:0:FFFF::1
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A	2A2A::2A2A
10	3194:0000:0000:0000:0001:0000:0000:0101	3194::1:0:0:101
11	2001:0DB8:0000:0000:0001:0000:0002:0100	2001:DB8::1:0:2:100
12	2001:0DB8:0000:0000:0000:0000:0010:A000	2001:DB8::10:A000
13	3330:0000:0000:0100:0000:0002:0000:0003	3330::100:0:2:0:3
14	FD00:0000:0000:1000:2000:0000:0001:0020	FD00::1000:2000:0:1:20
15	FD11:1000:0100:0010:0001:0000:1000:0100	FD11:1000:100:10:1:0:1000:100
16	2000:0000:0000:0000:0000:0000:0000:0002	2000::2

Answers to Calculating IPv6 Prefix Problems

Tables H-5 and H-6 list the answers to the problems listed earlier in Tables H-2 and H-3.

Table H-5 Answers: Finding the IPv6 Prefix, with a /64 Prefix Length

	Address (Assume a /64 Prefix Length)	Prefix (Subnet)
1	2987:BA11:B011:B00A:1000:0001:F001:F003	2987:BA11:B011:B00A::/64
2	3100:0000:0000:1010:D00D:D000:D00B:B00D	3100:0:0:1010::/64
3	FD00:0001:0001:0001:0200:00FF:FE00:0001	FD00:1:1:1::/64
4	FDDF:8080:0880:1001:0000:00FF:FE01:0507	FDDF:8080:880:1001::/64
5	32CC:0000:0000:000D:210F:0000:0000:0000	32CC:0:0:D::/64
6	2100:000E:00E0:0000:0000:0000:0000:0E00	2100:E:E0::/64
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C	3A11:CA00::/64
8	3799:9F9F:F000:0000:FFFF:0000:0000:0001	3799:9F9F:F000::/64
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A	2A2A::/64
10	3194:0000:0000:0000:0001:0000:0000:0101	3194::/64
11	2001:0DB8:0000:0000:0001:0000:0002:0100	2001:DB8::/64
12	2001:0DB8:0000:0000:0000:0000:0010:A000	2001:DB8::/64
13	3330:0000:0000:0100:0000:0002:0000:0003	3330:0:0:100::/64
14	FD00:0000:0000:1000:2000:0000:0001:0020	FD00:0:0:1000::/64
15	FD11:1000:0100:0010:0001:0000:1000:0100	FD11:1000:100:10::/64
16	2000:0000:0000:0000:0000:0000:0000:0002	2000::/64

Table H-6 Answers: Finding the IPv6 Prefix, with Other Prefix Lengths

	Address	Prefix (Subnet)
1	2987:BA11:B011:B00A:1000:0001:F001:F003 /60	2987:BA11:B011:B000::/60
2	3100:0000:0000:1010:D00D:D000:D00B:B00D /56	3100:0:0:1000::/56
3	FD00:0001:0001:0001:0200:00FF:FE00:0001 /52	FD00:1:1::/52
4	FDDF:8080:0880:1001:0000:00FF:FE01:0507 /48	FDDF:8080:880::/48
5	32CC:0000:0000:000D:210F:0000:0000:0000 /44	32CC::/44
6	2100:000E:00E0:0000:0000:0000:0000:0E00 /60	2100:E:E0::/60
7	3A11:CA00:0000:0000:0000:00FF:FECC:000C /56	3A11:CA00::/56
8	3799:9F9F:F000:0000:FFFF:0000:0000:0001 /52	3799:9F9F:F000::/52
9	2A2A:0000:0000:0000:0000:0000:0000:2A2A /48	2A2A::/48
10	3194:0000:0000:0000:0001:0000:0000:0101 /44	3194::/44

APPENDIX I

Practice for Chapter 27: Implementing IPv6 Addressing on Routers

This appendix provides practice problems for two types of addresses: unicast addresses formed with the EUI-64 feature and solicited node multicast addresses. With EUI-64, you take the 64-bit (16 hex digit) prefix and a MAC address, manipulate the MAC address into a 64-bit value, and use those 64 bits as the interface ID. Solicited node multicast addresses are formed from a standard 26 hex digit prefix, combined with the same last 6 hex digits as the unicast address.

EUI-64 and Solicited Node Multicast Problems

Table I-1 lists some practice problems. Each problem lists a prefix and a MAC address. Then, in Table I-2, record your answers for the unicast IPv6 address, assuming that EUI-64 rules are used. Also in Table I-2, list the solicited node multicast address associated with your calculated unicast address.

For each answer, use the best abbreviation, instead of a full 32-digit address.

The answers sit at the end of the appendix, in Table I-3.

Table I-1 IPv6 EUI-64 Unicast and Solicited Node Multicast Problems

	Prefix	MAC Address
1	2987:BA11:B011:B00A::/64	0000.1234.5678
2	3100:0000:0000:1010::/64	1234.5678.9ABC
3	FD00:0001:0001:0001::/64	0400.AAAA.0001
4	FDDF:8080:0880:1001::/64	0611.BABA.DADA
5	32CC:0000:0000:000D::/64	0000.0000.0001
6	2100:000E:00E0:0000::/64	0505.0505.0707
7	3A11:CA00:0000:0000::/64	0A0A.B0B0.0C0C
8	3799:9F9F:F000:0000::/64	F00F.0005.0041
9	2A2A:0000:0000:0000::/64	0200.0101.0101
10	3194:0000:0000:0000::/64	0C0C.000C.00CC

Table I-2 Blank Answer Table for Problems in Table I-1

	Unicast Address Using EUI-64	Solicited Node Multicast Address
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Answers to EUI-64 and Solicited Node Multicast Problems

Table I-3 lists the answers to the problems listed earlier in Table I-1.

Table I-3 Answers to Problems in Table I-1

	Unicast Address Using EUI-64	Solicited Node Multicast Address
1	2987:BA11:B011:B00A:200:12FF:FE34:5678	FF02::01:FF34.5678
2	3100::1010:1034:56FF:FE78:9ABC	FF02::01:FF78.9ABC
3	FD00:1:1:1:600:AAFF:FEAA:1	FF02::01:FFAA:1
4	FDDF:8080:880:1001:411:BAFF:FEBA:DADA	FF02::01:FFBA:DADA
5	32CC::D:200:FF:FE00:1	FF02::01:FF00:1
6	2100:E:E0:0:705:5FF:FE05:707	FF02::01:FF05:707
7	3A11:CA00::80A:B0FF:FEB0:C0C	FF02::01:FFB0:C0C
8	3799:9F9F:F000:0:F20F:FF:FE05:41	FF02::01:FF05:41
9	2A2A::1FF:FE01:101	FF02::01:FF01:101
10	3194::E0C:FF:FE0C:CC	FF02::01:FF0C:CC

Appendix J

Study Planner

Practice Test	Reading	Task

Element	Task	Goal Date	First Date Completed	Second Date Completed (Optional)	Notes
Introduction	Read Introduction				
Your Study Planner	Read Your Study Planner				
1. Introduction to TCP/IP Networking	Read Foundation Topics				
1. Introduction to TCP/IP Networking	Review Key Topics using the book or companion website				
1. Introduction to TCP/IP Networking	Define Key Terms using the book or companion website				
1. Introduction to TCP/IP Networking	Repeat DIKTA questions using the book or PTP exam engine				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
2. Fundamentals of Ethernet LANs	Read Foundation Topics				
2. Fundamentals of Ethernet LANs	Review Key Topics using the book or companion website				
2. Fundamentals of Ethernet LANs	Define Key Terms using the book or companion website				

2. Fundamentals of Ethernet LANs	Repeat DIKTA questions using the book or PTP exam engine				
2. Fundamentals of Ethernet LANs	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
3. Fundamentals of WANs and IP Routing	Read Foundation Topics				
3. Fundamentals of WANs and IP Routing	Review Key Topics using the book or companion website				
3. Fundamentals of WANs and IP Routing	Define Key Terms using the book or companion website				
3. Fundamentals of WANs and IP Routing	Repeat DIKTA questions using the book or PTP exam engine				
3. Fundamentals of WANs and IP Routing	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part I. Introduction to Networking	Complete all exercises in Part I Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
4. Using the Command-Line Interface	Read Foundation Topics				

4. Using the Command-Line Interface	Review Key Topics using the book or companion website				
4. Using the Command-Line Interface	Define Key Terms using the book or companion website				
4. Using the Command-Line Interface	Repeat DIKTA questions using the book or PTP exam engine				
4. Using the Command-Line Interface	Review the command tables				
4. Using the Command-Line Interface	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this chapter				
5. Analyzing Ethernet LAN Switching	Read Foundation Topics				
5. Analyzing Ethernet LAN Switching	Review Key Topics using the book or companion website				
5. Analyzing Ethernet LAN Switching	Define Key Terms using the book or companion website				
5. Analyzing Ethernet LAN Switching	Repeat DIKTA questions using the book or PTP exam engine				
5. Analyzing Ethernet LAN Switching	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
5. Analyzing Ethernet LAN Switching	Review the command tables				
5. Analyzing Ethernet LAN Switching	Watch video for this chapter using the companion website				

Practice Test	Take practice test in study mode using Part Review exam in practice test software for this chapter				
6. Configuring Basic Switch Management	Read Foundation Topics				
6. Configuring Basic Switch Management	Review Key Topics using the book or companion website				
6. Configuring Basic Switch Management	Define Key Terms using the book or companion website				
6. Configuring Basic Switch Management	Repeat DIKTA questions using the book or PTP exam engine				
6. Configuring Basic Switch Management	Complete config checklists in this chapter using the companion website				
6. Configuring Basic Switch Management	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
6. Configuring Basic Switch Management	Review command tables for this chapter				
6. Configuring Basic Switch Management	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
7. Configuring and Verifying Switch Interfaces	Read Foundation Topics				
7. Configuring and Verifying Switch Interfaces	Review Key Topics using the book or companion website				

7. Configuring and Verifying Switch Interfaces	Define Key Terms using the book or companion website				
7. Configuring and Verifying Switch Interfaces	Repeat DIKTA questions using the book or PTP exam engine				
7. Configuring and Verifying Switch Interfaces	Review command tables for this chapter				
7. Configuring and Verifying Switch Interfaces	Complete all memory tables in this chapter using the companion website				
7. Configuring and Verifying Switch Interfaces	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part II. Implementing Ethernet LANs	Complete all exercises in Part II Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
8. Implementing Ethernet Virtual LANs	Read Foundation Topics				
8. Implementing Ethernet Virtual LANs	Review Key Topics using the book or companion website				
8. Implementing Ethernet Virtual LANs	Define Key Terms using the book or companion website				
8. Implementing Ethernet Virtual LANs	Repeat DIKTA questions using the book or PTP exam engine				

8. Implementing Ethernet Virtual LANs	Complete config checklists in this chapter using the companion website				
8. Implementing Ethernet Virtual LANs	Review command tables for this chapter				
8. Implementing Ethernet Virtual LANs	Complete all memory tables in this chapter using the companion website				
8. Implementing Ethernet Virtual LANs	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
8. Implementing Ethernet Virtual LANs	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
9. Spanning Tree Protocol Concepts	Read Foundation Topics				
9. Spanning Tree Protocol Concepts	Review Key Topics using the book or companion website				
9. Spanning Tree Protocol Concepts	Define Key Terms using the book or companion website				
9. Spanning Tree Protocol Concepts	Repeat DIKTA questions using the book or PTP exam engine				
9. Spanning Tree Protocol Concepts	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				

10. RSTP and EtherChannel Configuration	Read Foundation Topics				
10. RSTP and EtherChannel Configuration	Review Key Topics using the book or companion website				
10. RSTP and EtherChannel Configuration	Define Key Terms using the book or companion website				
10. RSTP and EtherChannel Configuration	Repeat DIKTA questions using the book or PTP exam engine				
10. RSTP and EtherChannel Configuration	Complete config checklists in this chapter using the companion website				
10. RSTP and EtherChannel Configuration	Review command tables for this chapter				
10. RSTP and EtherChannel Configuration	Complete all memory tables in this chapter using the companion website				
10. RSTP and EtherChannel Configuration	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
10. RSTP and EtherChannel Configuration	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part III. Implementing VLANs and STP	Complete all exercises in Part III Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				

11. Perspectives on IPv4 Subnetting	Read Foundation Topics				
11. Perspectives on IPv4 Subnetting	Review Key Topics using the book or companion website				
11. Perspectives on IPv4 Subnetting	Define Key Terms using the book or companion website				
11. Perspectives on IPv4 Subnetting	Repeat DIKTA questions using the book or PTP exam engine				
11. Perspectives on IPv4 Subnetting	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
12. Analyzing Classful IPv4 Networks	Read Foundation Topics				
12. Analyzing Classful IPv4 Networks	Review Key Topics using the book or companion website				
12. Analyzing Classful IPv4 Networks	Define Key Terms using the book or companion website				
12. Analyzing Classful IPv4 Networks	Repeat DIKTA questions using the book or PTP exam engine				
12. Analyzing Classful IPv4 Networks	Complete all memory tables in this chapter using the companion website				
12. Analyzing Classful IPv4 Networks	Practice analyzing classful IPv4 networks using Appendix D on the companion website				

12. Analyzing Classful IPv4 Networks	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
13. Analyzing Subnet Masks	Read Foundation Topics				
13. Analyzing Subnet Masks	Review Key Topics using the book or companion website				
13. Analyzing Subnet Masks	Define Key Terms using the book or companion website				
13. Analyzing Subnet Masks	Repeat DIKTA questions using the book or PTP exam engine				
13. Analyzing Subnet Masks	Complete all memory tables in this chapter using the companion website				
13. Analyzing Subnet Masks	Practice analyzing subnet masks using Appendix E on the companion website				
13. Analyzing Subnet Masks	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
14. Analyzing Existing Subnets	Read Foundation Topics				
14. Analyzing Existing Subnets	Review Key Topics using the book or companion website				

14. Analyzing Existing Subnets	Define Key Terms using the book or companion website				
14. Analyzing Existing Subnets	Repeat DIKTA questions using the book or PTP exam engine				
14. Analyzing Existing Subnets	Complete all memory tables in this chapter using the companion website				
14. Analyzing Existing Subnets	Practice mask analysis using Appendix F on the companion website				
14. Analyzing Existing Subnets	Practice analyzing existing subnets using Appendix F on the companion website				
14. Analyzing Existing Subnets	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
15. Subnet Design	Read Foundation Topics				
15. Subnet Design	Review Key Topics using the book or companion website				
15. Subnet Design	Define Key Terms using the book or companion website				
15. Subnet Design	Repeat DIKTA questions using the book or PTP exam engine				
15. Subnet Design	Practice subnet design using Appendix G on the companion website				

15. Subnet Design	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part IV. IPv4 Addressing	Complete all exercises in Part IV Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
16. Operating Cisco Routers	Read Foundation Topics				
16. Operating Cisco Routers	Review Key Topics using the book or companion website				
16. Operating Cisco Routers	Define Key Terms using the book or companion website				
16. Operating Cisco Routers	Repeat DIKTA questions using the book or PTP exam engine				
16. Operating Cisco Routers	Review command tables for this chapter				
16. Operating Cisco Routers	Complete all memory tables in this chapter using the companion website				
16. Operating Cisco Routers	Do labs listed for this chapter at blog.certskills.com				
16. Operating Cisco Routers	Watch video for this chapter using the companion website				

Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
17. Configuring IPv4 Addresses and Static Routes	Read Foundation Topics				
17. Configuring IPv4 Addresses and Static Routes	Review Key Topics using the book or companion website				
17. Configuring IPv4 Addresses and Static Routes	Define Key Terms using the book or companion website				
17. Configuring IPv4 Addresses and Static Routes	Repeat DIKTA questions using the book or PTP exam engine				
17. Configuring IPv4 Addresses and Static Routes	Review command tables for this chapter				
17. Configuring IPv4 Addresses and Static Routes	Do labs listed for this chapter at blog.certskills.com				
17. Configuring IPv4 Addresses and Static Routes	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
18. IP Routing in the LAN	Read Foundation Topics				
18. IP Routing in the LAN	Review Key Topics using the book or companion website				
18. IP Routing in the LAN	Define Key Terms using the book or companion website				
18. IP Routing in the LAN	Repeat DIKTA questions using the book or PTP exam engine				

18. IP Routing in the LAN	Complete config checklists in this chapter using the companion website				
18. IP Routing in the LAN	Review command tables for this chapter				
18. IP Routing in the LAN	Do labs listed for this chapter using the Sim Lite app and/or at blog.certskills.com				
18. IP Routing in the LAN	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
19. IP Addressing on Hosts	Read Foundation Topics				
19. IP Addressing on Hosts	Review Key Topics using the book or companion website				
19. IP Addressing on Hosts	Define Key Terms using the book or companion website				
19. IP Addressing on Hosts	Repeat DIKTA questions using the book or PTP exam engine				
19. IP Addressing on Hosts	Review command tables for this chapter				
19. IP Addressing on Hosts	Watch video for this chapter using the companion website				
20. Troubleshooting IPv4 Routing	Read Foundation Topics				
20. Troubleshooting IPv4 Routing	Review Key Topics using the book or companion website				
20. Troubleshooting IPv4 Routing	Define Key Terms using the book or companion website				

Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part V. IPv4 Routing	Complete all exercises in Part V Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
21. Understanding OSPF Concepts	Read Foundation Topics				
21. Understanding OSPF Concepts	Review Key Topics using the book or companion website				
21. Understanding OSPF Concepts	Define Key Terms using the book or companion website				
21. Understanding OSPF Concepts	Repeat DIKTA questions using the book or PTP exam engine				
21. Understanding OSPF Concepts	Complete all memory tables in this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
22. Implementing Basic OSPF Features	Read Foundation Topics				
22. Implementing Basic OSPF Features	Review Key Topics using the book or companion website				
22. Implementing Basic OSPF Features	Define Key Terms using the book or companion website				

22. Implementing Basic OSPF Features	Repeat DIKTA questions using the book or PTP exam engine				
22. Implementing Basic OSPF Features	Complete config checklists in this chapter using the companion website				
22. Implementing Basic OSPF Features	Review command tables for this chapter				
22. Implementing Basic OSPF Features	Do labs listed for this chapter at blog.certskills.com				
22. Implementing Basic OSPF Features	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
23. Implementing Optional OSPF Features	Read Foundation Topics				
23. Implementing Optional OSPF Features	Review Key Topics using the book or companion website				
23. Implementing Optional OSPF Features	Define Key Terms using the book or companion website				
23. Implementing Optional OSPF Features	Repeat DIKTA questions using the book or PTP exam engine				
23. Implementing Optional OSPF Features	Complete all memory tables in this chapter using the companion website				
23. Implementing Optional OSPF Features	Complete config checklists in this chapter using the companion website				

23. Implementing Optional OSPF Features	Review command tables for this chapter				
23. Implementing Optional OSPF Features	Do labs listed for this chapter at blog.certskills.com				
23. Implementing Optional OSPF Features	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
24. OSPF Neighbors and Route Selection	Read Foundation Topics				
24. OSPF Neighbors and Route Selection	Review Key Topics using the book or companion website				
24. OSPF Neighbors and Route Selection	Define Key Terms using the book or companion website				
24. OSPF Neighbors and Route Selection	Repeat DIKTA questions using the book or PTP exam engine				
24. OSPF Neighbors and Route Selection	Review command tables for this chapter				
24. OSPF Neighbors and Route Selection	Complete all memory tables in this chapter using the companion website				
24. OSPF Neighbors and Route Selection	Watch video for this chapter using the companion website				
Part VI. OSPF	Complete all exercises in Part VI Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
25. Fundamentals of IP Version 6	Read Foundation Topics				

25. Fundamentals of IP Version 6	Review Key Topics using the book or companion website				
25. Fundamentals of IP Version 6	Define Key Terms using the book or companion website				
25. Fundamentals of IP Version 6	Repeat DIKTA questions using the book or PTP exam engine				
25. Fundamentals of IP Version 6	Review command tables for this chapter				
25. Fundamentals of IP Version 6	Complete all memory tables in this chapter using the companion website				
25. Fundamentals of IP Version 6	Practice abbreviating and expanding addresses using Appendix H on the companion website				
25. Fundamentals of IP Version 6	Practice calculating the IPv6 subnet prefix using Appendix H on the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
26. IPv6 Addressing and Subnetting	Read Foundation Topics				
26. IPv6 Addressing and Subnetting	Review Key Topics using the book or companion website				
26. IPv6 Addressing and Subnetting	Define Key Terms using the book or companion website				
26. IPv6 Addressing and Subnetting	Repeat DIKTA questions using the book or PTP exam engine				

26. IPv6 Addressing and Subnetting	Complete all memory tables in this chapter using the companion website				
26. IPv6 Addressing and Subnetting	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
27. Implementing IPv6 Addressing on Routers	Read Foundation Topics				
27. Implementing IPv6 Addressing on Routers	Review Key Topics using the book or companion website				
27. Implementing IPv6 Addressing on Routers	Define Key Terms using the book or companion website				
27. Implementing IPv6 Addressing on Routers	Repeat DIKTA questions using the book or PTP exam engine				
27. Implementing IPv6 Addressing on Routers	Review command tables for this chapter				
27. Implementing IPv6 Addressing on Routers	Complete all memory tables in this chapter using the companion website				
27. Implementing IPv6 Addressing on Routers	Practice EUI-64 and solicited node multicast problems using Appendix I on the companion website				
27. Implementing IPv6 Addressing on Routers	Do labs listed for this chapter at blog.certskills.com				
27. Implementing IPv6 Addressing on Routers	Watch video for this chapter using the companion website				

Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
28. Implementing IPv6 Addressing on Hosts	Read Foundation Topics				
28. Implementing IPv6 Addressing on Hosts	Review Key Topics using the book or companion website				
28. Implementing IPv6 Addressing on Hosts	Define Key Terms using the book or companion website				
28. Implementing IPv6 Addressing on Hosts	Repeat DIKTA questions using the book or PTP exam engine				
28. Implementing IPv6 Addressing on Hosts	Complete all memory tables in this chapter using the companion website				
28. Implementing IPv6 Addressing on Hosts	Review command tables for this chapter				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
29. Implementing IPv6 Routing	Read Foundation Topics				
29. Implementing IPv6 Routing	Review Key Topics using the book or companion website				
29. Implementing IPv6 Routing	Define Key Terms using the book or companion website				
29. Implementing IPv6 Routing	Repeat DIKTA questions using the book or PTP exam engine				
29. Implementing IPv6 Routing	Review command tables for this chapter				

29. Implementing IPv6 Routing	Complete all memory tables in this chapter using the companion website				
29. Implementing IPv6 Routing	Do labs listed for this chapter at blog.certskills.com				
29. Implementing IPv6 Routing	Watch video for this chapter using the companion website				
Practice Test	Take practice test in study mode using DIKTA exam in practice test software for this chapter				
Part VII. IP Version 6	Complete all exercises in Part VII Review				
Practice Test	Take practice test in study mode using Part Review exam in practice test software for this part				
Final Review	Take practice test in study mode for all book questions in practice test software				
Final Review	Review all Key Topics in all chapters or in the Key Topics app using the companion website				
Final Review	Review all Key Terms in all chapters or using the Key Terms flashcards on the companion website				
Final Review	Complete all memory tables for all chapters using the companion website				

Final Review	Take practice test in practice exam mode using Exam Bank #1 questions for all chapters				
Final Review	Take practice test in practice exam mode using Exam Bank #2 questions for all chapters				

APPENDIX K

Topics from Previous Editions

We base the books' contents on the Cisco exam topics; that is, the books attempt to cover the topics Cisco lists as exam topics. However, the book authoring process does create some challenges, particularly with the balance of what to include in the books and what to leave out.

For instance, when I compared a new exam to the old, Cisco had removed some topics—and I might want to keep the content in the book. There are a few reasons why. Sometimes I just expect that some readers will still want to read about that technology. Also, more than a few schools use these books as textbooks, and including some of the older-but-still-relevant topics can be a help. And keeping the old material available on each book's companion website takes only a little extra work, so we do just that.

Some of the older topics that I choose to keep on the companion website are small, so I collect them into this appendix. Other topics happen to have been an entire chapter in a previous edition of the books, so we include those topics each as a separate appendix. Regardless, the material exists here in this appendix, and in the appendices that follow, for your use if you have a need. But do not feel as though you must read this appendix for the current exam.

The topics in this appendix are as follows:

- IPv4 Address Types
- Bandwidth and Clock Rate on Serial Interfaces
- Implementing DHCP
- Using traceroute to Isolate Problems on Two Routers
- Default Routes with SLAAC on Router Interfaces

NOTE The content under the heading “IPv4 Address Types” was most recently published for the 100-105 Exam in 2016, in Chapter 20 of the *Cisco CCNA ICND1 100-105 Official Cert Guide*.

IPv4 Address Types

The IPv4 address space includes three major categories of addresses: unicast, broadcast, and multicast. For the current exam, Cisco lists one exam topic that asks you to compare and contrast these address types. To help you make those comparisons, this section explains multicast addressing, while pulling together the key ideas about unicast and broadcast IP addresses that have already been introduced, to pull the ideas together.

Review of Unicast (Class A, B, and C) IP Addresses

Unicast IP addresses are those Class A, B, and C IP addresses assigned to hosts, router interfaces, and other networking devices. Because most discussions about IP addressing refer to unicast IP addresses, most of us just refer to them as IP addresses and leave out the word *unicast*.

Just to be complete and define the concept, unicast addresses identify one interface on one device to IP. Just like your postal address gives the post office an address to use to send letters to your one specific house or apartment, a unicast IP address gives the IP network an address to use to send packets to one specific host. However, with IP, instead of addressing the device, unicast addresses identify individual interfaces. For example:

- A router with four LAN interfaces and two WAN interfaces has six unicast addresses, each in a different subnet, one for each interface.
- A PC with both an Ethernet network interface card (NIC) and a wireless NIC would have two unicast IPv4 addresses, one for each interface.

IP Broadcast Addresses

Broadcast IPv4 addresses give IP a way to send one packet that the network delivers to multiple hosts. IPv4 defines several types of broadcast addresses, with each type being used to reach a different set of hosts. These different broadcast IP addresses give different overhead protocols like DHCP the ability to efficiently reach all hosts in a specific part of the network. The following list reviews the three IP broadcast address types:

Local broadcast address: 255.255.255.255. Used to send a packet on a local subnet, knowing that routers will not forward the packet as is. Also called a *limited broadcast*.

Subnet broadcast address: One reserved address for each subnet, namely the numerically highest number in the subnet, as discussed in Chapter 13, “Analyzing Subnet Masks.” A packet sent to a subnet broadcast address can be routed to the router connected to that subnet and then sent as a data link broadcast to all hosts in that one subnet. Also called an *all-hosts broadcast* to emphasize that all hosts in a subnet are reached, and also called a *directed broadcast*.

Network broadcast address: One reserved address for each classful network, namely the numerically highest number in the network. Used to send one packet to all hosts in that one network. Also called an *all-subnets broadcast*, referring to the fact that the packet reaches all subnets in a network.

This chapter has already shown how a local broadcast works, sending the message over the same subnet in which it was first transmitted, but no further. However, the other two types are a little more interesting.

Subnet and network broadcasts provide a way to send packets to all hosts in a subnet or network (respectively) while reducing waste. For instance, with a subnet broadcast, routers forward the packet just like any other IP packet going to that subnet. When that packet arrives at the router connected to that subnet, the last router then encapsulates the packet in a LAN broadcast, so that all hosts receive a copy. Figure K-1 shows the idea.

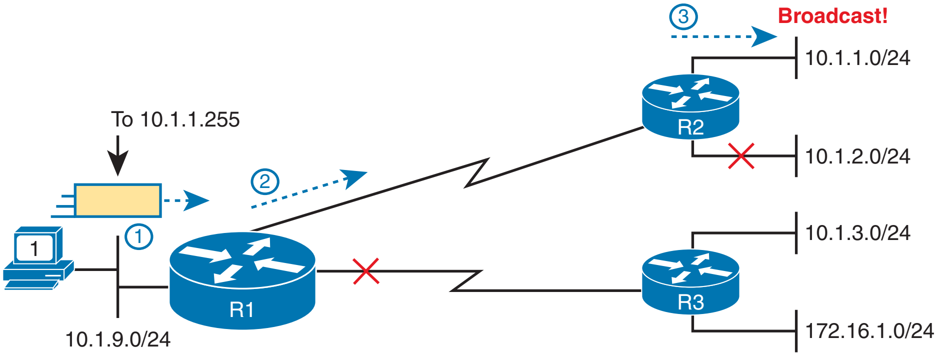


Figure K-1 *Example of a Subnet Broadcast to 10.1.1.255*

The figure shows two key points. R1 does not flood or broadcast the frame to all other routers, instead routing it to the next router (R2 in this case) so that the packet reaches subnet 10.1.1.0/24. R2, connected to subnet 10.1.1.0/24, forwards the packet onto the LAN, but encapsulates the packet in an Ethernet broadcast frame, so that it reaches all hosts in the subnet.

The figure shows the intended use of the subnet broadcast address; however, it presents a security issue today. Many attacks start with a ping to subnet broadcast addresses, hoping to get many hosts to reply. Cisco changed the IOS default many years ago to disable the forwarding of subnet broadcasts onto a connected subnet (that is, it disables Step 3 in Figure K-1). That default setting is based on the **no ip directed-broadcast** interface subcommand.

A network broadcast packet (a packet with a network broadcast address as the destination) works in a similar way. To reach all subnets, however, the routers create copies of the packet and flood it so it reaches all subnets inside the classful network. On any LAN interfaces, the packet is forwarded in a LAN broadcast, just as shown in Step 3 of Figure K-1.

IPv4 Multicast Addresses (Class D Addresses)

Multicast IP addresses and the related protocols help solve a similar problem as compared to broadcast addresses, but mainly for applications, and without the same security issues experienced by broadcast addresses. To see how it works, consider this example. A video application may be designed to show live video feeds. If 10 people at the same remote site in the same subnet want to watch the same video at the same time, the application could be designed so that the application sent the same video data 10 times, once to each client in the same subnet. An application designed to use Class D multicast addresses could send 1 packet, which the routers would route across the WAN, and then deliver a copy to all 10 hosts in the destination subnet.

When using multicast, all the hosts still use their individual unicast IP address for their normal traffic, while also using the same multicast IPv4 address for the multicast application. Any server or client that happens to use an application designed to take advantage of IP multicast then also uses the Class D multicast addresses that the application chooses to use. You can think of a Class D address more as a multicast group—in fact, it is often called that—because hosts join the group so that they can receive the packets sent by the multicast application.

Class D addresses begin with a first octet of between 224 and 239, with some ranges reserved for various purposes. Much of the Class D address space is set aside for a company to deploy one of these multicast applications, and then pick an address from the Class D range, and configure it to be used by a multicast application.

As an example, imagine the video application uses Class D address 226.1.1.1. Figure K-2 illustrates the process by which the application at the server on the left sends one multicast packet with destination address 226.1.1.1. Note that for this process to work, the hosts with * beside them registered with their local routers to notify the routers that the host wants to receive packets destined to multicast address 226.1.1.1. When the action in this figure begins, the routers collectively know which subnets have hosts that want a copy of multicasts sent to 226.1.1.1 and which subnets do not.

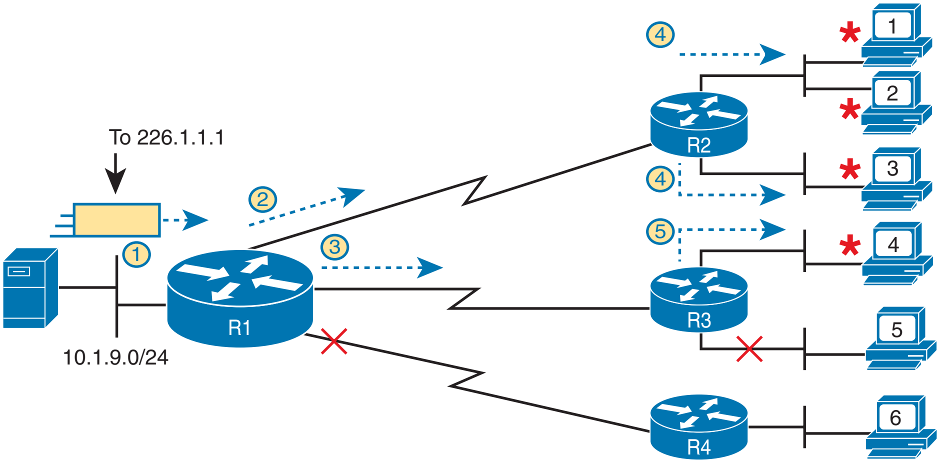


Figure K-2 *Example of a Multicast Packet Flow for Three Registered Hosts*

Following the steps in the figure:

1. The server on the left generates and sends a multicast packet.
2. Router R1 replicates the packet to send a copy to both R2…
3. …and to R3. R1 does not replicate and send a copy to R4, because there are no hosts near R4 listening for packets sent to 226.1.1.1.

4. R2 processes the multicast packet received from R1, and because of the earlier host registration process, R2 knows that at least one host off both its LAN interfaces is listening for packets sent to 226.1.1.1. R2 therefore forwards a copy of the packet out each of its LAN interfaces.
5. R3 receives the multicast packet from R1 and uses the same kind of logic as R2. However, R3 knows from the earlier host registration process that only one of its LAN interfaces connects to a subnet with hosts listening for packets sent to 226.1.1.1, so R3 forwards a copy of the packet out that one interface only.

As you can see from this example, the server sent one packet, and the routers replicated the packet so it reached all the correct locations in the network.

As another comparison between unicast and multicast addresses, note that multicast addresses may be used as destination IP addresses only, whereas unicast addresses may be used as both the destination and source address. For instance, consider the packets in the example shown in Figure K-2. All those packets flow from one host, so the packet uses a unicast IP address of that host's unicast IP address.

Finally, to complete one more comparison between unicast IP addressing and multicast IP addressing, think about that last hop router in the example shown in Figure K-1. If a router such as R2 or R3 had forwarded a unicast IP packet, the router would look in its ARP cache to find the unicast IP address for the destination in that connected subnet and the associated unicast MAC address. That will not work when forwarding a multicast packet with a multicast (Class D) destination IP address.

To encapsulate a multicast IP packet over an Ethernet LAN, IP multicast calculates the destination MAC address with a simple process. The process copies the last 23 bits of the IP address behind a reserved 25-bit prefix to form the 48-bit destination MAC address. The resulting MAC address, called a multicast MAC address, begins with hex 01005E. So, the multicast IP packet, encapsulated in the multicast Ethernet frame, is forwarded out the router interface onto the LAN. At that point, the switches take one of the following approaches to forwarding the frame so that all hosts that want a copy of the frame get a copy:

- Flood the multicast frame as if it were a broadcast
- Use other Ethernet multicast features that flood the frame only to those same devices that registered to receive a copy

If you feel as though these few pages probably left out some detail, indeed, several books have been written about IP multicast all to itself. The topic is indeed large. For this book's purposes, know the main comparison points with unicast addressing. Multicast addressing gives applications that need to communicate the same data at the same time to multiple hosts a much more efficient way to do that. If the application is written to make use of IP multicast, the application can consume much less traffic in the network, as compared to using unicast IP addresses and sending every host a copy of the packet.

Comparing and Contrasting IP Address Types

The preceding pages reviewed unicast and broadcast addresses and explained the core concepts behind IP multicast addresses. Table K-1 summarizes the key comparison points mentioned throughout this section for convenient study.

Table K-1 Comparisons of Unicast, Broadcast, and Multicast IP Addresses

	Unicast	Broadcast	Multicast
Primarily used for data sent by the most common user apps (web, email, chat, and so on)	Yes	No	No
Assigned to hosts with DHCP	Yes	No	No
Uses Class A, B, and C addresses	Yes	No	No
Primarily used by overhead protocols (DHCP, ARP) to send one message to more than one device	No	Yes	No
Used as destination IP address only	No	Yes	Yes
Primarily used by applications that send the same data at the same time to multiple clients	No	No	Yes
Uses Class D addresses	No	No	Yes

NOTE The content under the heading “Bandwidth and Clock Rate on Serial Interfaces” was most recently published for the 100-105 Exam in 2016, in Chapter 17 of the *CCENT/CCNA ICND1 100-105 Official Cert Guide*.

Bandwidth and Clock Rate on Serial Interfaces

WAN serial links can run at a wide variety of speeds. To deal with the wide range of speeds, routers physically slave themselves to the speed as dictated by the CSU/DSU through a process called *clocking*. As a result, routers can use serial links without the need for additional configuration or autonegotiation to sense the serial link’s speed. The CSU/DSU knows the speed, the CSU/DSU sends clock pulses over the cable to the router, and the router reacts to the clocking signal.

To build a serial link in a home lab, the routers can use serial interface cards that normally use an external CSU/DSU, and make a serial link, without requiring the expense of two CSU/DSUs. Figure K-3 shows the concept. To make it work, the link uses two serial cables—one a DTE cable and the other a DCE cable—which swap the transmit and receive pair on the cables.

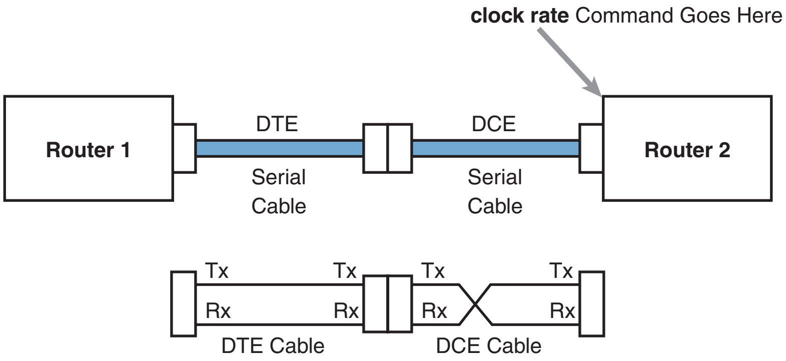


Figure K-3 *Serial Link in Lab*

Using the correct cabling works, as long as you add one command: the **clock rate** interface subcommand. This command tells that router the speed at which to transmit bits on a serial link like the one shown in Figure K-3. The **clock rate** command is not needed on real serial links, because the CSU/DSU provides the clocking. When you create a serial link in the lab using cables, without any real CSU/DSUs on the link, the router with the DCE cable must supply that clocking function, and the **clock rate** command tells the router to provide it.

NOTE Newer router IOS versions automatically add a default **clock rate 2000000** command on serial interfaces that have a DCE cable connected to them. While helpful, this speed might be too high for some types of back-to-back serial cables, so consider using a lower speed in lab.

Example K-1 shows the configuration of the **clock rate** command. The end of the example verifies that this router can use the **clock rate** command with the **show controllers** command. This command confirms that R1 has a V.35 DCE cable connected.

Example K-1 *Router R1 Configuration with the* **clock rate** *Command*

```
R1# show running-config
! lines omitted for brevity
interface Serial0/0/0
   ip address 172.16.4.1 255.255.255.0
   clock rate 2000000
!
interface Serial0/0/1
   ip address 172.16.5.1 255.255.255.0
   clock rate 128000

! lines omitted for brevity

R1# show controllers serial 0/0/1
Interface Serial0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 128000
idb at 0x8169BB20, driver data structure at 0x816A35E4
! Lines omitted for brevity
```

NOTE The **clock rate** command does not allow just any speed to be configured. However, the list of speeds does vary from router to router.

Some people confuse the router **bandwidth** command with the **clock rate** command. The **clock rate** command sets the actual Layer 1 speed used on the link, if no CSU/DSU is used, as just described. Conversely, every router interface has a bandwidth setting, either by default or configured. The bandwidth of the interface is the documented speed of the interface, which does not have to match the actual Layer 1 speed used on the interface.

That bandwidth setting does not impact how fast the interface transmits data. Instead, routers use the interface bandwidth setting as both documentation and as input to some other processes. For instance, the Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP) routing protocols base their routing protocol metrics on the bandwidth by default.

Example K-2 highlights the bandwidth setting on Router R1's S0/0/1 interface, as configured in the previous example. In that previous example, the **clock rate 128000** command sets the clock rate to 128 kbps, but it leaves the **bandwidth** command unset. As a result, IOS uses the default serial bandwidth setting of 1544, which means 1544 kbps—which is the speed of a T1 serial link.

Example K-2 *Router Bandwidth Settings*

```
R1# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is up
   Hardware is WIC MBRD Serial
   Description: link to R3
   Internet address is 10.1.13.1/24
   MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
      reliability 255/255, txload 1/255, rxload 1/255
   Encapsulation HDLC, loopback not set
```

The common mistake people make is to know about clock rate but mistakenly think that the bandwidth setting is just another term for "clock rate." It is not. Follow these rules to find these two interface settings:

- To see the clock rate, look for the **clock rate** interface subcommand in the configuration, or use the **show controllers serial** *number* command (as shown in Example K-1).
- To see the bandwidth setting on an interface, look for the **bandwidth** interface subcommand in the configuration, or use the **show interfaces** [*type number*] command (as shown in Example K-2).

Note that using default bandwidth settings on most router interfaces makes sense, with the exception of serial interfaces. IOS defaults to a bandwidth of 1544 (meaning 1544 kbps, or 1.544 Mbps) for serial interfaces, regardless of the speed dictated by the provider or by a **clock rate** command in the lab. Most engineers set the bandwidth to match the actual speed, for example, using the **bandwidth 128** interface subcommand on a link running at 128 kbps. On Ethernet 10/100 or 10/100/1000 interfaces, the router knows the speed used and dynamically sets the Ethernet interface's bandwidth to match.

NOTE The content under the heading "Implementing DHCP" was most recently published for the 100-105 Exam in 2016, in Chapter 20 of the *Cisco CCNA ICND1 100-105 Official Cert Guide.*

Implementing DHCP

This section includes DHCP implementation topics from an earlier edition of the book.

DHCP Server Configuration on Routers

A quick Google search on "DHCP server products" reveals that many companies offer DHCP server software. Cisco routers (and some Cisco switches) can also act as a DHCP server with just a little added configuration.

Configuring a Cisco router to act as a DHCP server uses a new configuration concept, one per subnet, called a *DHCP pool*. All the per-subnet settings go into a per-subnet DHCP pool. The only DHCP command that sits outside the pool is the command that defines the list of addresses excluded from being leased by DHCP. The Cisco IOS DHCP server configuration steps are as follows:

Step 1. Use the **ip dhcp excluded-address** *first last* command in global configuration mode to list addresses that should be excluded (that is, not leased by DHCP).

Step 2. Use the **ip dhcp pool** *name* command in global configuration mode to both create a DHCP pool for a subnet and to navigate into DHCP pool configuration mode. Then also

- **a.** Use the **network** *subnet-ID mask* or **network** *subnet-ID prefix-length* command in DHCP pool configuration mode to define the subnet for this pool.
- **b.** Use the **default-router** *address1 address2*… command in DHCP pool configuration mode to define default router IP address(es) in that subnet.
- **c.** Use the **dns-server** *address1 address2*… command in DHCP pool configuration mode to define the list of DNS server IP addresses used by hosts in this subnet.
- **d.** Use the **lease** *days hours minutes* command in DHCP pool configuration mode to define the length of the lease, in days, hours, and minutes.
- **e.** Use the **domain-name** *name* command in DHCP pool configuration mode to define the DNS domain name.
- **f.** Use the **next-server** *ip-address* command in DHCP pool configuration mode to define the TFTP server IP address used by any hosts (like phones) that need a TFTP server.

Of course, an example can help, particularly with so many configuration commands required. Figure K-4 shows the organization of the configuration, while sticking to pseudocode rather than the specific configuration commands. (Upcoming Example K-3 shows a matching configuration.) Note that for each of the two LAN subnets, there is a global command to exclude addresses, and then a group of settings for each of two different DHCP pools.

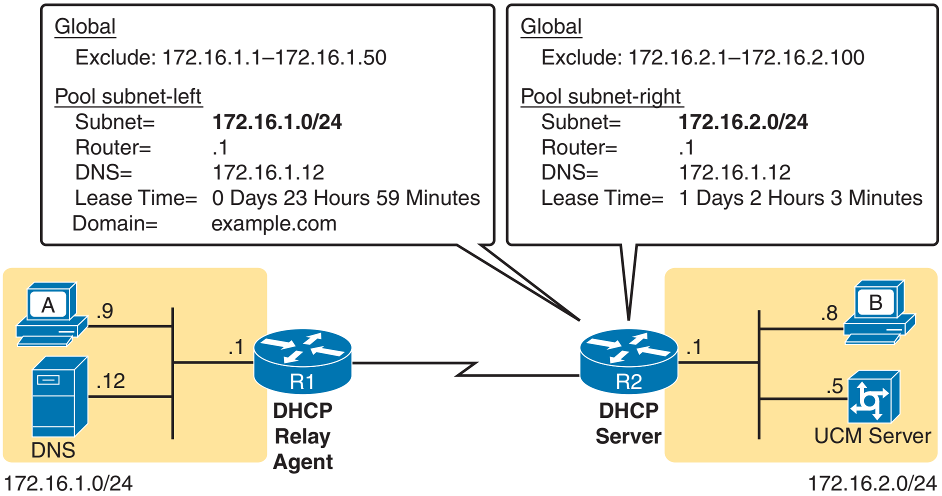


Figure K-4 *DHCP Server Configuration Pseudocode*

Example K-3 *R2 as a DHCP Server Per the Concepts in Figure K-4*

```
ip dhcp excluded-address 172.16.1.1 172.16.1.50
ip dhcp excluded-address 172.16.2.1 172.16.2.100
!
ip dhcp pool subnet-left
 network 172.16.1.0 255.255.255.0
 dns-server 172.16.1.12
 default-router 172.16.1.1
 lease 0 23 59
 domain-name example.com
 next-server 172.16.2.5
!
ip dhcp pool subnet-right
 network 172.16.2.0 /24
 dns-server 172.16.1.12
 default-router 172.16.2.1
 lease 1 2 3
 next-server 172.16.2.5
```

Focus on subnet 172.16.1.0/24 for a moment: the subnet is configured as pool subnet-left. The subnet ID and mask match the subnet ID chosen for that subnet. Then the global **ip dhcp excluded-address** command, just above, reserves 172.16.1.1 through 172.16.1.50, so that this DHCP server will not lease these addresses. The server will automatically exclude the subnet ID (172.16.1.0) as well, so this DHCP server will begin leasing IP addresses starting with the .51 address.

Now look at the details for subnet-right. It uses a DHCP pool **network** command with a prefix-style mask. It defines the same DNS server, as does the pool for the other subnet, but a different default router setting, because, of course, the default router in each subnet is different. This pool includes a lease time of 1:02:03 (1 day, 2 hours, and 3 minutes) just as an example.

Also note that both subnets list a TFTP server IP address of the Unified Communications Manager (UCM) server with the **next-server** command. In most cases, you would find this setting in the pools for subnets in which phones reside.

Finally, note that configuring a router as a DHCP server does not remove the need for the **ip helper-address** command. If DHCP clients still exist on LANs that do not have a DHCP server, then the routers connected to those LANs still need the **ip helper-address** command. For example, in Figure K-3, R1 would still need the **ip helper-address** command on its LAN interface. R2 would not need the command on its LAN interface, because R2 could service those requests, rather than needing to forward the DHCP messages to some other server.

IOS DHCP Server Verification

The IOS DHCP server function has several different **show** commands. These three commands list most of the details:

show ip dhcp binding: Lists state information about each IP address currently leased to a client

show ip dhcp pool [*poolname*]**:** Lists the configured range of IP addresses, plus statistics for the number of currently leased addresses and the high-water mark for leases from each pool

show ip dhcp server statistics: Lists DHCP server statistics

Example K-4 shows sample output from two of these commands, based on the configuration from Figure K-4 and Example K-3. In this case, the DHCP server leased one IP address from each of the pools—one for host A and one for host B—as shown in the highlighted portions of the output.

Example K-4 *Verifying Current Operation of a Router-Based DHCP Server*

```
R2# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
172.16.1.51         0063.6973.636f.2d30.    Oct 12 2012 02:56 AM    Automatic
                    3230.302e.3131.3131.
                    2e31.3131.312d.4661.
                    302f.30
172.16.2.101        0063.6973.636f.2d30.    Oct 12 2012 04:59 AM    Automatic
                    3230.302e.3232.3232.
                    2e32.3232.322d.4769.
                    302f.30
R2# show ip dhcp pool subnet-right
Pool subnet-right :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 1
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 172.16.2.102         172.16.2.1       - 172.16.2.254      1
```

Note that the output in Example K-4 does not happen to list the excluded addresses, but it does show the effects. The addresses assigned to the clients end with .51 (host A, subnet 172.16.1.0) and .101 (host B, subnet 172.16.2.0), proving that the server did exclude the addresses as shown in the configuration in Example K-3. The server avoided the .1 through .50 addresses in subnet 172.16.1.0, and the .1 through .100 addresses in subnet 172.16.2.0.

NOTE The DHCP server keeps status (state) information about each DHCP client that leases an address. Specifically, it remembers the DHCP client ID and the IP address leased to the client. As a result, an IPv4 DHCP server can be considered to be a stateful DHCP server.

Troubleshooting DHCP Services

To be prepared for the CCNA Simlet questions, you have to be ready to predict what symptoms would occur when the network was misconfigured in particular ways. This next section takes a similar approach, pointing out the most typical issues that could be introduced through incorrect or missing configuration, and then discussing what symptoms should happen and how to recognize those problems.

This section begins with a typical look at configuration mistakes and the symptoms that occur with those mistakes. In particular, this section looks at problems with the relay agent's helper address as well as the IOS DHCP server configuration. This section then looks at non-DHCP problems related to that data plane, breaking the problem into issues between the client and relay agent, and between the relay agent and DHCP server. The final section takes a short look at how a DHCP server prevents duplicate IP addresses between hosts that use static IP addresses and those that use DHCP.

DHCP Relay Agent Configuration Mistakes and Symptoms

One configuration mistake that prevents a DHCP client from leasing an IP address is the misconfiguration or the omission of the **ip helper-address** interface subcommand on the router acting as the DHCP relay agent. The relay agent takes the incoming DHCP message, changes the destination address of the packet to be the address on the **ip helper-address** *address* command, and forwards the packet to that address. If the command is missing, the router does not attempt to forward the DHCP messages at all; if it is incorrect, the relay agent forwards the DHCP packets, but they never arrive at the actual DHCP server.

The main problem symptom in this case is the failure of a DHCP client to lease an address. If you can identify a client that has a problem, and you know what VLAN or subnet in which that host resides, you can then work to identify any routers connected to that subnet, to find and correct the **ip helper-address** subcommands.

Beyond that step, this list summarizes a few other related points.

- The DHCP relay agent feature is needed on interfaces only if the DHCP server is on a different subnet; it is not needed if the DHCP server is on the same subnet as the client.
- On routers with VLAN trunks (with a router-on-a-stick [ROAS] subinterface configuration), the subinterfaces also need an **ip helper-address** command (assuming they meet the first criteria in this list).

- If an exam question does not allow you to look at the configuration, use the **show ip interface** [*type number*] command to view the **ip helper-address** setting on an interface.

About that last point, Example K-5 shows an example of the **show ip interface g0/0** command. In this case, the interface has been configured with the **ip helper-address 172.16.2.11** command; the **show** command output basically restates that fact. Note that if there were no **ip helper-address** configured on the interface, the text would instead read “Helper address is not set.”

Example K-5 *Listing the Current Helper Address Setting with* **show ip interface**

```
R1# show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 182.16.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is 172.16.2.11
! Lines omitted for brevity (about 20 lines)
```

IOS DHCP Server Configuration Mistakes and Symptoms

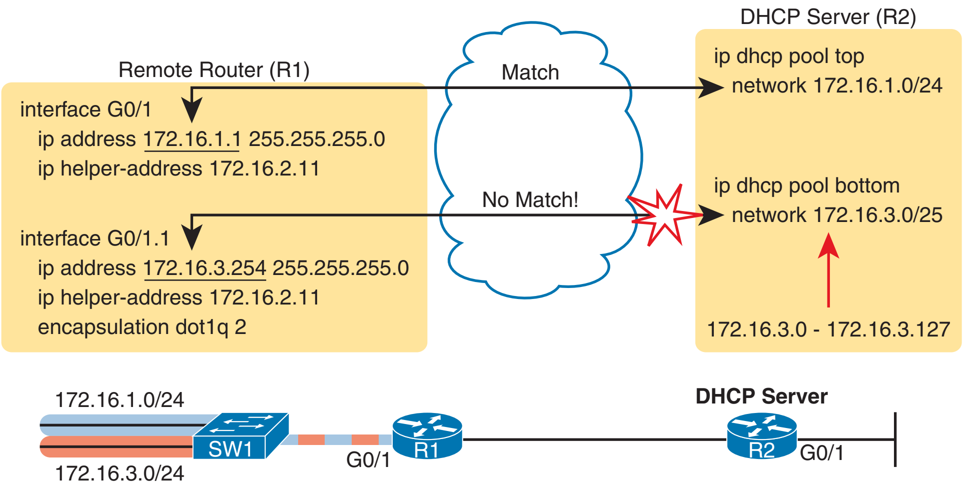
When using an IOS DHCP server, from a troubleshooting perspective, break issues into two broad categories: those that prevent DHCP clients from leasing an address and those that allow the lease but provide incorrect settings to the client.

First, the primary configuration mistake that causes a failure in the DHCP lease process is the misconfiguration of the **network** command. The problem revolves around these key facts:

- The packet from the relay agent to the DHCP server uses the relay agent’s interface IP address as the source IP address in the forwarded DHCP message.
- The DHCP server compares that source IP address in the received DHCP packet to the **network** commands in its DHCP pools to find the right pool.
- Each **network** *subnet mask* command implies a range of addresses, just like any other IP network or subnet shown with a subnet mask.
- If the source IP address of the packet is not in the range of addresses implied by any **network** command in all the pools, the DHCP server has no pool to use for that request. The DHCP server does not know how to respond, so it does not reply at all.

As an example of that failure, consider the configuration shown in Figure K-5. The left side shows the configuration on R1, a DHCP relay agent that has two interfaces configured with the **ip helper-address 172.16.2.11** command. The DHCP server configuration on the right lists two pools, intended as one pool for each subnet off Router R1. However, the **network 172.16.3.0 /25** command implies an address range of 172.16.3.0 to 172.16.3.127, and the relay agent’s interface address of 172.16.3.254 is not within that range of numbers. The solution would be to correct the DHCP server’s **network** command to use a /24 mask.

NOTE The **ip helper-address** configuration on the left is correct. The figure uses the ROAS configuration here just to reinforce the comment in the earlier section that ROAS subinterfaces also need an **ip helper-address** subcommand.



Remote Router (R1)

```
interface G0/1
  ip address 172.16.1.1 255.255.255.0
  ip helper-address 172.16.2.11

interface G0/1.1
  ip address 172.16.3.254 255.255.255.0
  ip helper-address 172.16.2.11
  encapsulation dot1q 2
```

DHCP Server (R2)

```
ip dhcp pool top
  network 172.16.1.0/24


ip dhcp pool bottom
  network 172.16.3.0/25
```

Figure K-5 *A Sample Misconfiguration of a DHCP Pool* **network** *Command*

While you ultimately need to find this kind of problem and fix the configuration, on the exam you need to be ready to discover the root cause based on symptoms and **show** commands as well. So, when troubleshooting DHCP issues, and the client fails to lease an address, look at the IOS DHCP server's **network** commands. Calculate the range of IP addresses as if that command were defining a subnet. Then compare that range of addresses by the **network** command in each pool to the interface addresses on the DHCP relay agent routers. Every relay agent interface (that is, every interface with an **ip helper-address** command configured) should be included in a pool defined at the IOS DHCP server.

The DHCP server can also be misconfigured in a way that allows the lease of an address, but then causes other problems. If the lease process works, but the rest of the parameters given to the client are incorrect or missing, the client could operate, but operate poorly. This list summarizes the kinds of mistakes and the resulting symptoms:

- With the DNS server IP addresses incorrectly configured on the server (or omitted), hosts would fail to resolve hostnames into their associated IP addresses.
- With the default gateway IP address incorrectly configured on the server (or omitted), hosts could not communicate outside the local subnet.
- With the TFTP server IP address incorrectly configured (or omitted), an IP phone would fail to correctly load its configuration.

IP Connectivity from DHCP Relay Agent to DHCP Server

For the DHCP process to work with a centralized server, IP broadcast packets must flow between the client and relay agent, and IP unicast packets must flow between the relay agent and the DHCP server. Any problem that prevents the flow of these packets also prevents DHCP from working.

For perspective, consider the topology in Figure K-6, which again shows the relay agent on the left and the DHCP server on the right. The server uses IP address 172.16.2.11, and the relay agent uses interface address 172.16.1.1. Any failure that prevents the flow of IP packets between those two IP addresses would prevent host A from leasing an IP address.

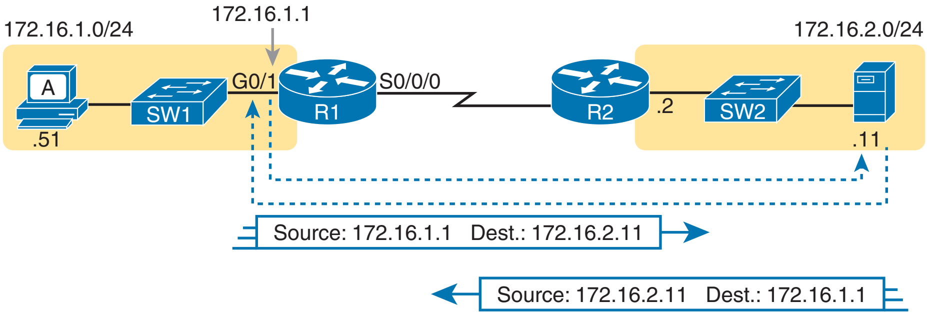

172.16.1.0/24
172.16.1.1
172.16.2.0/24
A
.51
SW1
G0/1
R1
S0/0/0
R2
.2
SW2
.11
Source: 172.16.1.1 Dest.: 172.16.2.11
Source: 172.16.2.11 Dest.: 172.16.1.1

Figure K-6 *Addresses Used Between Relay Agent and Server*

Remember that the IP addresses are used on the packets between the relay agent and server, and know that you may need to troubleshoot IP routing to ensure those packets can be delivered.

LAN Connectivity Between the DHCP Client and Relay Agent

You might encounter a network environment where DHCP messages on the same LAN as the DHCP client all show a destination IP address of 255.255.255.255. What does that really mean? When a packet uses this 255.255.255.255 address:

- The address is called the *local broadcast address*.
- Packets sent to this address are not forwarded as-is by routers.
- On a LAN, the sender of an IP local broadcast packet encapsulates these IP packets in an Ethernet frame with an Ethernet broadcast destination address (FFFF.FFFF.FFFF), so the LAN broadcasts the frame.

As a result of the logic in these steps, the broadcast DHCP messages can easily flow between the client and router, as long as the LAN works.

Summary of DHCP Troubleshooting

In summary, as a study tool, the following list summarizes the key troubleshooting ideas from this section on troubleshooting DHCP:

Step 1. If you're using a centralized DHCP server, at least one router on each remote subnet that has DHCP clients must act as DHCP relay agent and have a correctly configured **ip helper-address** *address* subcommand on the interface connected to that subnet.

Step 2. If you're using a centralized IOS DHCP server, make sure the DHCP pools' **network** commands match the entire network's list of router interfaces that have an **ip helper-address** command pointing to this DHCP server.

Step 3. Troubleshoot for any IP connectivity issues between the DHCP relay agent and the DHCP server, using the relay agent interface IP address and the server IP address as the source and destination of the packets.

Step 4. Troubleshoot for any LAN issues between the DHCP client and the DHCP relay agent.

Also, as one final note about DHCP in the real world, DHCP might seem dangerous at this point, with all the focus on potential problems in this section, combined with the importance of DHCP and its use by most end-user devices. However, DHCP has some great availability features. First, most DHCP servers set their lease times for at least a few days, often a week, or maybe longer. Combined with that, the DHCP protocol has several processes through which the client reconfirms the existing lease with the server, and re-leases the same IP address in advance of the expiration of the lease. Clients do not simply wait until the moment the lease would expire to then contact the DHCP server, hoping it is available. So the network can have outages, and DHCP clients that have already leased an address can continue to work without any problem.

Detecting Conflicts with Offered Versus Used Addresses

Beyond troubleshooting the types of problems that would prevent DHCP from working, the IOS DHCP server tries to prevent another type of problem: assigning IP addresses with DHCP when another host tries to statically configure that same IP address. Although the DHCP server configuration clearly lists the addresses in the pool, plus those to be excluded from the pool, hosts can still statically configure addresses from the range inside the DHCP pool. In other words, no protocols prevent a host from statically configuring and using an IP address from within the range of addresses used by the DHCP server.

Knowing that some host might have statically configured an address from within the range of addresses in the DHCP pool, both DHCP servers and clients try to detect such problems, called *conflicts*, before the client uses a newly leased address.

DHCP servers detect conflicts by using pings. Before offering a new IP address to a client, the DHCP server first pings the address. If the server receives a response to the ping, some other host must already be using the address, which lets the server know a conflict exists. The server notes that particular address as being in conflict, and the server does not offer the address, moving on to the next address in the pool.

The DHCP client can also detect conflicts, but instead of using ping, it uses ARP. In the client case, when the DHCP client receives from the DHCP server an offer to use a particular IP address, the client sends an Address Resolution Protocol (ARP) request for that address. If another host replies, the DHCP client has found a conflict.

Example K-6 lists output from the router-based DHCP server on R2, after host B detected a conflict using ARP. Behind the scenes, host B used DHCP to request a lease, with the process working normally until host B used ARP and found some other device already used 172.16.2.102. At that point, host B then sent a DHCP message back to the server, rejecting the use of address 172.16.2.102. The example shows the router's log message related to host B's discovery of the conflict and a **show** command that lists all conflicted addresses.

Example K-6 *Displaying Information About DHCP Conflicts in IOS*

```
*Oct 16 19:28:59.220: %DHCPD-4-DECLINE_CONFLICT: DHCP address conflict:   client
0063.6973.636f.2d30.3230.302e.3034.3034.2e30.3430.342d.4769.302f.30     declined
172.16.2.102.
R2# show ip dhcp conflict
IP address        Detection method   Detection time          VRF
172.16.2.102      Gratuitous ARP     Oct 16 2012 07:28 PM
```

The **show ip dhcp conflict** command lists the method through which the server added each address to the conflict list: either gratuitous ARP, as detected by the client, or ping, as detected by the server. The server avoids offering these conflicted addresses to any future clients, until the engineer uses the **clear ip dhcp conflict** command to clear the list.

NOTE The content under the heading "Using traceroute to Isolate the Problem to Two Routers" was most recently published for the 100-105 Exam in 2016, in Chapter 23 of the *Cisco CCNA ICND1 100-105 Official Cert Guide*.

Using traceroute to Isolate the Problem to Two Routers

One of the best features of the **traceroute** command, as compared to ping, is that when it does not complete, it gives an immediate clue as to where to look next. With ping, when the ping fails, the next step is usually to use more **ping** commands. With traceroute, it tells you what router to try to connect and look at the routes and in which direction.

NOTE As a reminder, this book uses the term *forward route* for routes that send the packets sent by the **ping** or **traceroute** command, and *reverse route* for the packets sent back.

When a problem exists, a **traceroute** command results in a partial list of routers. Then the command either finishes with an incomplete list, or it runs until the user must stop the command. In either case, the output does not list all routers in the end-to-end route because of the underlying problem.

NOTE In addition, the **traceroute** command may not finish even though the network has no problems. Routers and firewalls may filter the messages sent by the **traceroute** command, or the TTL Exceeded messages, which would prevent the display of portions or all or part of the path.

The last router listed in the output of a **traceroute** command's output tells us where to look next to isolate the problem, as follows:

- Connect to the CLI of the last router listed, to look at forward route issues.
- Connect to the CLI of the next router that should have been listed, to look for reverse route issues.

To see why, consider an example based on the internetwork in Figure K-7. In this case, R1 uses an extended traceroute to host 5.5.5.5, with source IP address 1.1.1.1. This command's output lists router 2.2.2.2, then 3.3.3.3, and then the command cannot complete.

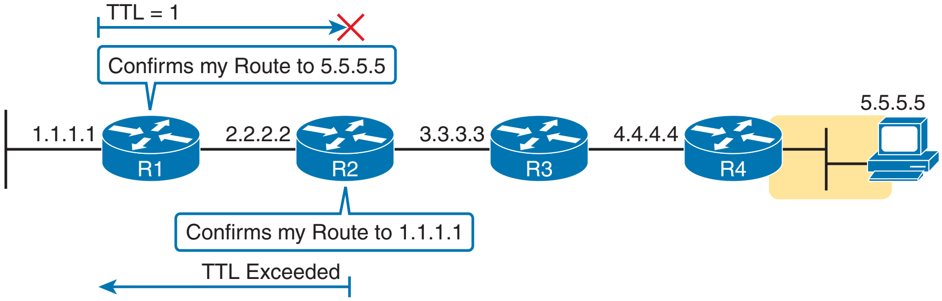


Figure K-7 *Messages That Cause the* **traceroute** *Command to List 2.2.2.2*

First, Figure K-7 focuses on the first line of output: the line that lists first-hop router 2.2.2.2.

The figure shows the TTL=1 message at the top and the TTL Exceeded message back on the bottom. This first pair of messages in the figure must have worked, because without them, the **traceroute** command on R1 cannot have learned about a router with address 2.2.2.2. The first (top) message required R1 to have a route for 5.5.5.5, which sent the packets to R2 next. The TTL Exceeded message required that R2 have a route that matched address 1.1.1.1, to send the packets back to R1's LAN IP address.

Next, Figure K-8 focuses on the messages that allow the second line of output on R1's sample **traceroute** command: the line that correctly lists 3.3.3.3 as the next router in the route.

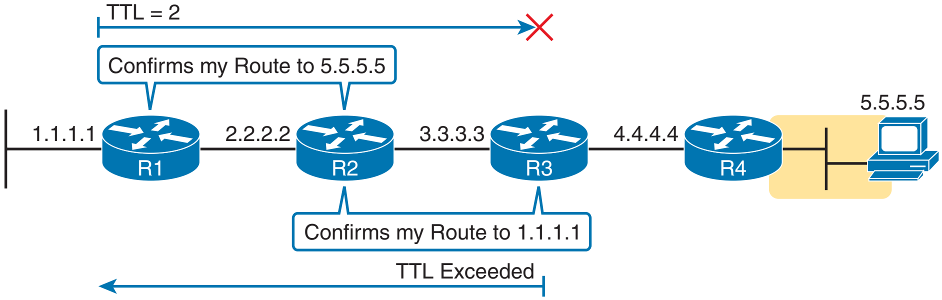


Figure K-8 *Messages That Cause the* **traceroute** *Command to List 3.3.3.3*

Following the same logic, the **traceroute** output lists 3.3.3.3 because the messages in Figure K-8 must have worked. For these messages to flow, the routes listed in Figure K-7 must exist, plus new routes listed in K-8. Specifically, the TTL=2 packet at the top requires R2 to have a route for 5.5.5.5, which sends the packets to R3 next. The TTL Exceeded message requires that R3 have a route that matches address 1.1.1.1, to send the packets back toward R1's LAN IP address.

In this example, the **traceroute 5.5.5.5** command does not list any routers beyond 2.2.2.2 and 3.3.3.3 However, based on the figures, it is clear that 4.4.4.4 should be the next IP address listed. To help isolate the problem further, why might the next messages—the message with TTL=3 and the response—fail?

Figure K-9 points out the routing issues that can cause this command to not be able to list 4.4.4.4 as the next router. First, R3 must have a forward route matching destination 5.5.5.5

and forwarding the packet to Router R4. The return message requires a reverse route matching destination 1.1.1.1 and forwarding the packet back to Router R3.

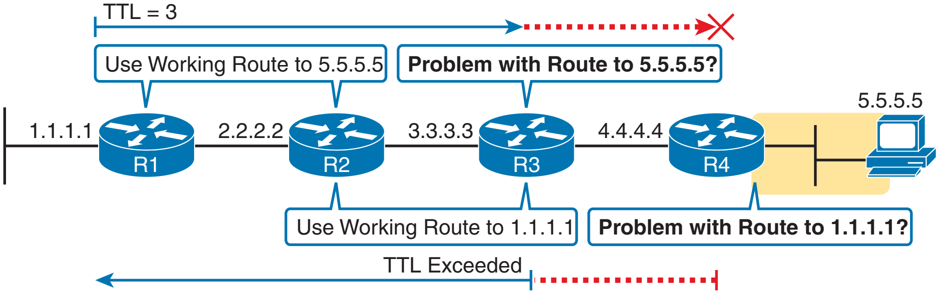

TTL = 3
Use Working Route to 5.5.5.5
Problem with Route to 5.5.5.5?
5.5.5.5
1.1.1.1
R1
2.2.2.2
R2
3.3.3.3
R3
4.4.4.4
R4
Use Working Route to 1.1.1.1
Problem with Route to 1.1.1.1?
TTL Exceeded

Figure K-9 *Issues That Could Prevent* **traceroute** *from Listing 4.4.4.4*

In conclusion, for this example, if a routing problem prevents the **traceroute** command from working, the problem exists in one of two places: the forward route to 5.5.5.5 on Router R3, or the reverse route to 1.1.1.1 on R4.

NOTE The content under the heading "Default Routes with SLAAC on Router Interfaces" was most recently published in Chapter 32 of the *Cisco CCNA ICND1 100-105 Official Cert Guide*.

Default Routes with SLAAC on Router Interfaces

Routers can use DHCP on their own interface and learn their IP address, mask, and even a default IPv4 route. In particular, that process can be useful on a router that connects to the Internet. The enterprise router uses DHCP as a client, learning its own IPv4 address with DHCP and adding a default route pointing to the ISP's router as the next-hop IPv4 address.

Routers can accomplish the same goals with IPv6, just with a few different protocols and methods. As with IPv4, the IPv6 enterprise router can dynamically learn its IPv6 address and dynamically create a default IPv6 route to the ISP's router. This section shows the details, with the enterprise router using SLAAC to learn its address and the information needed to create a default route.

First, the enterprise router that connects to the ISP, like Router R1 in Figure K-10, requires the configuration of the interface subcommand **ipv6 address autoconfig default**. This command tells the router that, on that interface, use SLAAC to build its own IPv6 address. R1 would act like any host that uses SLAAC, as shown in Step 2 of the figure, and send an NDP RS message over the link. As noted at Step 3, the ISP router would send back an RA message, announcing router ISP1's IPv6 address and the IPv6 prefix used on the link.

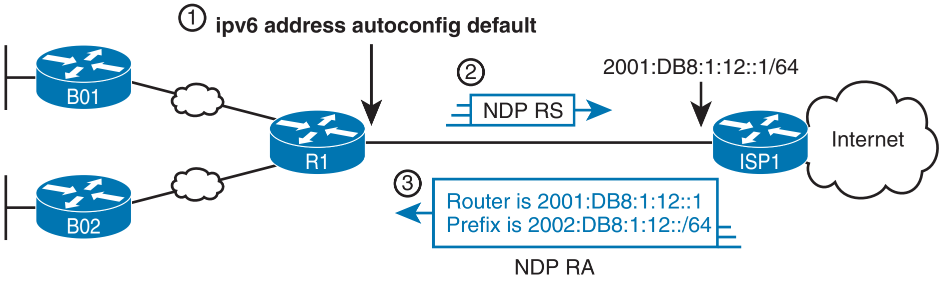

1
ipv6 address autoconfig default
2
NDP RS
2001:DB8:1:12::1/64
B01
R1
B02
ISP1
Internet
3
Router is 2001:DB8:1:12::1
Prefix is 2002:DB8:1:12::/64
NDP RA

Figure K-10 *Enterprise Router Using SLAAC to Build IPv6 Address and Default IPv6 Route*

When R1 receives the NDP RA message, it does the following:

Interface address: Builds its own interface IPv6 address using the SLAAC process, based on the prefix in the RA.

Local /128 route: Adds a local (/128) IPv6 route for the address, as it would for any interface IPv6 address.

Connected route for prefix: Adds a connected (/64) route for the prefix learned in the NDP RA message.

Default route: R1 adds a default route, to destination ::/0, with the next-hop address of ISP's link-local address, as learned in the RA sent by router ISP1.

Note that the router can be configured to add this default route or not. As shown in the figure, the router builds a default route. Using the **ipv6 address autoconfig** subcommand without the **default** keyword causes the router to build its address with SLAAC but not add a default route.

Example K-7 shows the three IPv6 routes on Router R1 just mentioned in the list. In particular, note the codes for the connected route and the default route; both codes begin with ND, meaning the route was learned with NDP. In particular, as highlighted in the legend part of the output, *ND* refers to an NDP-learned default route, and *NDp* refers to an NDP-learned prefix (as listed in the NDP RA message in Figure K-10 in this case). Note also that these same two routes have an administrative distance of 2, which is the default administrative distance of IPv6 routes learned with NDP.

Example K-7 *Learning an Address and Default Static Route with DHCP*

```
R1# show ipv6 route
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, la - LISP alt
       lr - LISP site-registrations, ld - LISP dyn-eid, a - Application
ND ::/0 [2/0]
     via FE80::22FF:FE22:2222, Serial0/0/0
NDp 2001:DB8:1:12::/64 [2/0]
     via Serial0/0/0, directly connected
L 2001:DB8:1:12:32F7:DFF:FE29:8560/128 [0/0]
    via Serial0/0/0, receive
! lines omitted for brevity
```

This page intentionally left blank

APPENDIX L

LAN Troubleshooting

NOTE This appendix contains content that was published as a chapter in one of the past editions of this book or a related book. The author includes this appendix with the current edition as extra reading for anyone interested in learning more. However, note that the content in this appendix has not been edited since it was published in the earlier edition, so references to exams and exam topics, and to other chapters, will be outdated. This appendix was previously published as Chapter 4 of the book *CCNA Routing and Switching ICND2 200-105 Official Cert Guide*, published in 2016.

This chapter discusses the LAN topics discussed in depth in the first three chapters, plus a few prerequisite topics, from a troubleshooting perspective.

Troubleshooting for any networking topic requires a slightly different mindset as compared to thinking about configuration and verification. When thinking about configuration and verification, it helps to think about basic designs, learn how to configure the feature correctly, and learn how to verify the correct configuration is indeed working correctly. However, to learn how to troubleshoot, you need to think about symptoms when the design is incorrect, or if the configuration does not match the good design. What symptoms occur when you make one type of mistake or another? This chapter looks at the common types of mistakes, and works through how to look at the status with **show** commands to find those mistakes.

This chapter breaks the material into four major sections. The first section tackles the largest topic, STP troubleshooting. STP is not likely to fail as a protocol; instead, STP may not be operating as designed, so the task is to find how STP is currently working and discover how to then make the configuration implement the correct design. The second major section then moves on to Layer 2 EtherChannels, which have a variety of small potential problems that can prevent the dynamic formation of an EtherChannel.

The third major section of the chapter focuses on the data plane forwarding of Ethernet frames on LAN switches, in light of VLANs, trunks, STP, and EtherChannels. That same section reviews the Layer 2 forwarding logic of a switch in light of these features. The fourth and final major section then examines VLAN and trunking issues, and how those issues impact switch forwarding.

Note that a few of the subtopics listed within the exam topics at the beginning of this chapter are not discussed in this chapter. This chapter does not discuss VTP beyond its basic features or Layer 3 EtherChannels.

Foundation Topics

Troubleshooting STP

STP questions tend to intimidate many test takers. STP uses many rules, with tiebreakers in case one rule ends with a tie. Without much experience with STP, people tend to distrust their own answers. Also, even those of us with networking jobs already probably do not troubleshoot STP very often, because STP works well. Often, troubleshooting STP is not about STP failing to do its job but rather about STP working differently than designed, with a different root switch, or different root ports (RP), and so on. Seldom does STP troubleshooting begin with a case in which STP has failed to prevent a loop.

This section reviews the rules for STP, while emphasizing some important troubleshooting points. In particular, this section takes a closer look at the tiebreakers that STP uses to make decisions. It also makes some practical suggestions about how to go about answering exam questions such as "which switch is the root switch?"

Determining the Root Switch

Determining the STP root switch is easy if you know all the switches' BIDs: Just pick the lowest value. If the question lists the priority and MAC address separately, as is common in some **show** command output, pick the switch with the lowest priority, or in the case of a tie, pick the lower MAC address value.

And just to be extra clear, STP does not have nor need a tiebreaker for electing the root switch. The BID uses a switch universal MAC address as the last 48 bits of the BID. These MAC addresses are unique in the universe, so there should never be identical BIDs or the need for a tiebreaker.

For the exam, a question that asks about the root switch might not be so simple as listing a bunch of BIDs and asking you which one is "best." A more likely question is a simulator (sim) question in which you have to do any **show** commands you like or a multiple choice question that lists the output from only one or two commands. Then you have to apply the STP algorithm to figure out the rest.

When faced with an exam question using a simulator, or just the output in an exhibit, use a simple strategy of ruling out switches, as follows:

Step 1. Begin with a list or diagram of switches, and consider all as possible root switches.

Step 2. Rule out any switches that have an RP (**show spanning-tree**, **show spanning-tree root**), because root switches do not have an RP.

Step 3. Always try **show spanning-tree**, because it identifies the local switch as root directly: "This switch is the root" on the fifth line of output.

Step 4. Always try **show spanning-tree root**, because it identifies the local switch as root indirectly: The RP column is empty if the local switch is the root.

Step 5. When using a sim, rather than try switches randomly, chase the RPs. For example, if starting with SW1, and SW1's G0/1 is an RP, next try the switch on the other end of SW1's G0/1 port.

Step 6. When using a sim, use **show spanning-tree vlan** *x* on a few switches and record the root switch, RP, and designated port (DP). This strategy can quickly show you most STP facts.

The one step in this list that most people ignore is the idea of ruling out switches that have an RP. Root switches do not have an RP, so any switch with an RP can be ruled out as not being the root switch for that VLAN. Example L-1 shows two commands on switch SW2 in some LAN that confirms that SW2 has an RP and is therefore not the root switch.

Example L-1 *Ruling Out Switches as Root Based on Having a Root Port*

```
SW2# show spanning-tree vlan 20 root

                                           Root  Hello Max Fwd
Vlan                   Root ID             Cost   Time Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0020         32788 1833.9d7b.0e80         4    2   20  15  Gi0/2

SW2# show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    32788
             Address     1833.9d7b.0e80
             Cost        4
             Port        26 (GigabitEthernet0/2)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32788  (priority 32768 sys-id-ext 20)
             Address     1833.9d7b.1380
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

Both commands identify SW2's G0/2 port as its RP, so if you follow the suggestions, the next switch to try in a sim question would be the switch on the other end of SW2's G0/2 interface.

Determining the Root Port on Nonroot Switches

Determining the RP of a switch when **show** command output is available is relatively easy. As shown recently in Example L-1, both **show spanning-tree** and **show spanning-tree root** list the root port of the local switch, assuming it is not the root switch. The challenge comes more when an exam question makes you think through how the switches choose the RP based on the root cost of each path to the root switch, with some tiebreakers as necessary.

As a review, each nonroot switch has one, and only one, RP for a VLAN. To choose its RP, a switch listens for incoming Hello bridge protocol data units (BPDU). For each received Hello, the switch adds the cost listed in the hello BPDU to the cost of the incoming interface (the interface on which the Hello was received). That total is the root cost over that path. The lowest root cost wins, and the local switch uses its local port that is part of the least root cost path as its root port.

Most humans can analyze what STP chooses by using a network diagram and a slightly different algorithm. Instead of thinking about Hello messages and so on, approach the question as this: the sum of all outgoing port costs between the nonroot switch and the root. Repeating a familiar example, with a twist, Figure L-1 shows the calculation of the root cost. Note that SW3's Gi0/1 port has yet again had its cost configured to a different value.

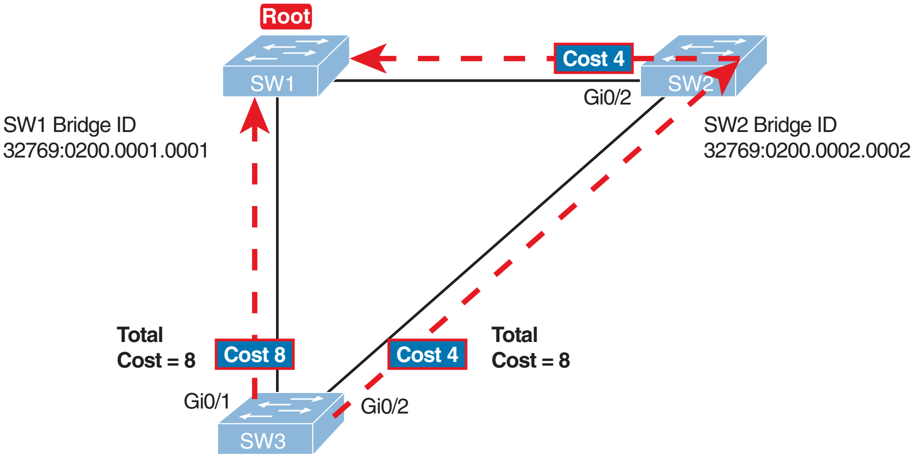


Figure L-1 *SW3's Root Cost Calculation Ends in a Tie*

STP Tiebreakers When Choosing the Root Port

Figure L-1 shows the easier process of adding the STP costs of the outgoing interfaces over each from SW3, a nonroot, to SW1, the root. It also shows a tie (on purpose), to talk about the tiebreakers.

When a switch chooses its root port, the first choice is to choose the local port that is part of the least root cost path. When those costs tie, the switch picks the port connected to the neighbor with the lowest BID. This tiebreaker usually breaks the tie, but not always. So, for completeness, the three tiebreakers are, in the order a switch uses them, as follows:

1. Choose based on the lowest neighbor bridge ID.
2. Choose based on the lowest neighbor port priority.
3. Choose based on the lowest neighbor internal port number.

(Note that the switch only considers the root paths that tie when thinking about these tiebreakers.)

For example, Figure L-1 shows that SW3 is not root and that its two paths to reach the root tie with their root costs of 8. The first tiebreaker is the lowest neighbor's BID. SW1's BID value is lower than SW2's, so SW3 chooses its G0/1 interface as its RP in this case.

The last two RP tiebreakers come into play only when two switches connect to each other with multiple links, as shown in Figure L-2. In that case, a switch receives Hellos on more than one port from the same neighboring switch, so the BIDs tie.

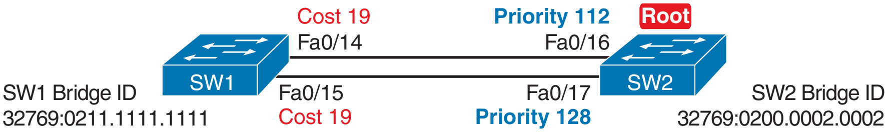


Figure L-2 *Topology Required for the Last Two Tiebreakers for Root Port*

In this particular example, SW2 becomes root, and SW1 needs to choose its RP. SW1's port costs tie, at 19 each, so SW1's root cost over each path will tie at 19. SW2 sends Hellos over each link to SW1, so SW1 cannot break the tie based on SW1's neighbor BID because both list SW2's BID. So, SW1 has to turn to the other two tiebreakers.

NOTE In real life, most engineers would put these two links into an EtherChannel.

The next tiebreaker is a configurable option: the neighboring switch's port priority on each neighboring switch interface. Cisco switch ports default to a setting of 128, with a range of values from 0 through 255, with lower being better (as usual). In this example, the network engineer has set SW2's F0/16 interface with the **spanning-tree vlan 10 port-priority 112** command. SW1 learns that the neighbor has a port priority of 112 on the top link and 128 on the bottom, so SW1 uses its top (F0/14) interface as the root port.

If the port priority ties, which it often does due to the default values, STP relies on an internal port numbering on the neighbor. Cisco switches assign an internal integer to identify each interface on the switch. The nonroot looks for the neighbor's lowest internal port number (as listed in the Hello messages) and chooses its RP based on the lower number.

Cisco switches use an obvious numbering, with Fa0/1 having the lowest number, then Fa0/2, then Fa0/3, and so on. So, in Figure L-2, SW2's Fa0/16 would have a lower internal port number than Fa0/17; SW1 would learn those numbers in the Hello; and SW1 would use its Fa0/14 port as its RP.

Suggestions for Attacking Root Port Problems on the Exam

Exam questions that make you think about the RP can be easy if you know where to look and the output of a few key commands is available. However, the more conceptual the question, the more you have to calculate the root cost over each path, correlate that to different **show** commands, and put the ideas together. The following list makes a few suggestions about how to approach STP problems on the exam:

1. If available, look at the **show spanning-tree** and **show spanning-tree root** commands. Both commands list the root port and the root cost (see Example L-1).
2. The **show spanning-tree** command lists cost in two places: the root cost at the top, in the section about the root switch; and the interface cost, at the bottom, in the per-interface section. Be careful, though; the cost at the bottom is the interface cost, not the root cost!

3. For problems where you have to calculate a switch's root cost:

   a. Memorize the default cost values: 100 for 10 Mbps, 19 for 100 Mbps, 4 for 1 Gbps, and 2 for 10 Gbps.

   b. Look for any evidence of the **spanning-tree cost** configuration command on an interface, because it overrides the default cost. Do not assume default costs are used.

   c. When you know a default cost is used, if you can, check the current actual speed as well. Cisco switches choose STP cost defaults based on the current speed, not the maximum speed.

Determining the Designated Port on Each LAN Segment

Each LAN segment has a single switch that acts as the designated port (DP) on that segment. On segments that connect a switch to a device that does not even use STP—for example, segments connecting a switch to a PC or a router—the switch always wins, because it is the only device sending a Hello onto the link. However, links with two switches require a little more work to discover which should be the DP. By definition:

Step 1. For switches connected to the same LAN segment, the switch with the lowest cost to reach the root, as advertised in the Hello they send onto the link, becomes the DP on that link.

Step 2. In case of a tie, among the switches that tied on cost, the switch with the lowest BID becomes the DP.

For example, consider Figure L-3. This figure notes the root, RPs, and DPs and each switch's least cost to reach the root over its respective RP.

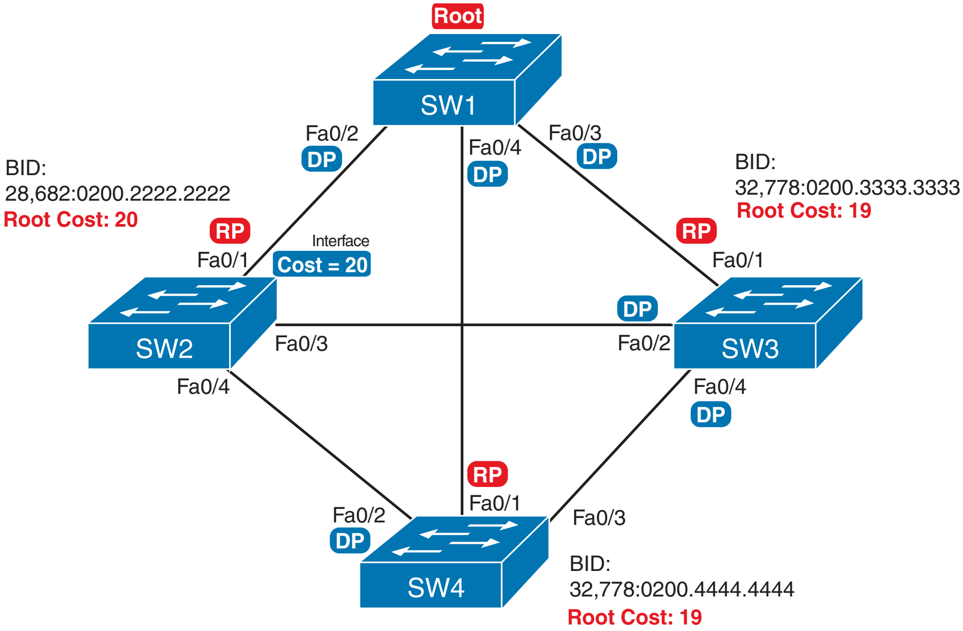


Figure L-3 *Picking the DPs*

Focus on the segments that connect the nonroot switches for a moment:

SW2–SW4 segment: SW4 wins because of its root cost of 19, compared to SW2's root cost of 20.

SW2–SW3 segment: SW3 wins because of its root cost of 19, compared to SW2's root cost of 20.

SW3–SW4 segment: SW3 and SW4 tie on root cost, both with root cost 19. SW3 wins due to its better (lower) BID value.

Interestingly, SW2 loses and does not become DP on the links to SW3 and SW4 even though SW2 has the better (lower) BID value. The DP tiebreaker does use the lowest BID, but the first DP criteria is the lowest root cost, and SW2's root cost happens to be higher than SW3's and SW4's.

NOTE A single switch can connect two or more interfaces to the same collision domain, and compete to become DP, if hubs are used. In such cases, two different switch ports on the same switch tie, the DP choice uses the same two final tiebreakers as used with the RP selection: the lowest interface STP priority, and if that ties, the lowest internal interface number.

Suggestions for Attacking Designated Port Problems on the Exam

As with exam questions asking about the RP, exam questions that make you think about the DP can be easy if you know where to look and the output of a few key commands is available. However, the more conceptual the question, the more you have to think about the criteria for choosing the DP: first the root cost of the competing switches, and then the better BID if they tie based on root cost.

The following list gives some tips to keep in mind when digging into a given DP issue. Some of this list repeats the suggestions for finding the RP, but to be complete, this list includes each idea as well.

1. If available, look at the **show spanning-tree** commands, at the list of interfaces at the end of the output. Then, look for the Role column, and look for Desg, to identify any DPs.
2. Identify the root cost of a switch directly by using the **show spanning-tree** command. But be careful! This command lists the cost in two places, and only the mention at the top, in the section about the root, lists the root cost.
3. For problems where you have to calculate a switch's root cost, do the following:
   a. Memorize the default cost values: 100 for 10 Mbps, 19 for 100 Mbps, 4 for 1 Gbps, and 2 for 10 Gbps.
   b. Look for any evidence of the **spanning-tree cost** configuration command on an interface, because it overrides the default cost. Do not assume default costs are used.
   c. When you know a default cost is used, if you can, check the current actual speed as well. Cisco switches choose STP cost defaults based on the current speed, not the maximum speed.

STP Convergence

STP puts each RP and DP into a forwarding state, and ports that are neither RP nor DP into a blocking state. Those states may remain as is for days, weeks, or months. But at some point, some switch or link will fail, a link may change speeds (changing the STP cost), or the STP configuration may change. Any of these events can cause switches to repeat their STP algorithm, which may in turn change their own RP and any ports that are DPs.

When STP converges based on some change, not all the ports have to change their state. For instance, a port that was forwarding, if it still needs to forward, just keeps on forwarding. Ports that were blocking that still need to block keep on blocking. But when a port needs to change state, something has to happen, based on the following rules:

- For interfaces that stay in the same STP state, nothing needs to change.
- For interfaces that need to move from a forwarding state to a blocking state, the switch immediately changes the state to blocking.
- For interfaces that need to move from a blocking state to a forwarding state, the switch first moves the interface to listening state, then learning state, each for the time specified by the forward delay timer (default 15 seconds). Only then is the interface placed into forwarding state.

Because the transition from blocking to forwarding does require some extra steps, you should be ready to respond to conceptual questions about the transition.

Troubleshooting Layer 2 EtherChannel

EtherChannels can prove particularly challenging to troubleshoot for a couple of reasons. First, you have to be careful to match the correct configuration, and there are many more incorrect configuration combinations than there are correct combinations. Second, many interface settings must match on the physical links, both on the local switch and on the neighboring switch, before a switch will add the physical link to the channel. This second major section in the chapter works through both sets of issues.

Incorrect Options on the channel-group Command

The rules for the small set of working configuration options on the **channel-group** command can be summarized as follows, for a single EtherChannel:

1. On the local switch, all the **channel-group** commands for all the physical interfaces must use the same channel-group number.
2. The channel-group number can be different on the neighboring switches.
3. If using the **on** keyword, you must use it on the corresponding interfaces of both switches.
4. If you use the **desirable** keyword on one switch, the switch uses PAgP; the other switch must use either **desirable** or **auto**.
5. If you use the **active** keyword on one switch, the switch uses LACP; the other switch must use either **active** or **passive**.

These rules summarize the correct configuration options, but the options actually leave many more incorrect choices. The following list shows some incorrect configurations that the switches allow, even though they would result in the EtherChannel not working. The list compares the configuration on one switch to another based on the physical interface configuration. Each lists the reasons why the configuration is incorrect.

- Configuring the **on** keyword on one switch, and **desirable**, **auto**, **active**, or **passive** on the other switch. The **on** keyword does not enable PAgP, and does not enable LACP, and the other options rely on PAgP or LACP.
- Configuring the **auto** keyword on both switches. Both use PAgP, but both wait on the other switch to begin negotiations.
- Configuring the **passive** keyword on both switches. Both use LACP, but both wait on the other switch to begin negotiations.
- Configuring the **active** keyword on one switch and either **desirable** or **auto** on the other switch. The **active** keyword uses LACP, whereas the other keywords use PAgP.
- Configuring the **desirable** keyword on one switch and either **active** or **passive** on the other switch. The **desirable** keyword uses PAgP, whereas the other keywords use LACP.

Example L-2 shows an example that matches the last item in the list. In this case, SW1's two ports (F0/14 and F0/15) have been configured with the **desirable** keyword, and SW2's matching F0/16 and F0/17 have been configured with the **active** keyword. The example lists some telling status information about the failure, with notes following the example.

Example L-2 *Incorrect Configuration Using Mismatched PortChannel Protocols*

```
SW1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SD)         PAgP      Fa0/14(I)   Fa0/15(I)

SW1# show interfaces status | include Po|14|15
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/14                       connected    301        a-full  a-100 10/100BaseTX
Fa0/15                       connected    301        a-full  a-100 10/100BaseTX
Po1                          notconnect   unassigned   auto   auto
```

Start at the top, in the legend of the **show etherchannel summary** command. The *D* code letter means that the channel itself is down, with *S* meaning that the channel is a Layer 2 EtherChannel. Code *I* means that the physical interface is working independently from the PortChannel (described as "stand-alone"). Then, the bottom of that command's output highlights PortChannel 1 (Po1) as Layer 2 EtherChannel in a down state (SD), with F0/14 and F0/15 as stand-alone interfaces (I).

Interestingly, because the problem is a configuration mistake, the two physical interfaces still operate independently, as if the PortChannel did not exist. The last command in the example shows that while the PortChannel 1 interface is down, the two physical interfaces are in a connected state.

NOTE As a suggestion for attacking EtherChannel problems on the exam, rather than memorizing all the incorrect configuration options, concentrate on the list of correct configuration options. Then look for any differences between a given question's configuration as compared to the known correct configurations and work from there.

Analyzing the Switch Data Plane Forwarding

STP and EtherChannel both have an impact on what a switch's forwarding logic can use. STP limits which interfaces the data plane even considers using by placing some ports in a blocking state (STP) or discarding state (RSTP), which in turn tells the data plane to simply not use that port. EtherChannel gives the data plane new ports to use in the switch's MAC address table—EtherChannels—while telling the data plane to not use the underlying physical interfaces in an EtherChannel in the MAC table.

This (short) third major section of the chapter explores the impact of STP and EtherChannel on data plane logic and a switch's MAC address table.

Predicting STP Impact on MAC Tables

Consider the small LAN shown in Figure L-4. The LAN has only three switches, with redundancy, just big enough to make the point for this next example. The LAN supports two VLANs, 1 and 2, and the engineer has configured STP such that SW3 blocks on a different port in each of the two VLANs. As a result, VLAN 1 traffic would flow from SW3 to SW1 next, and in VLAN 2, traffic would flow from SW3 to SW2 next instead.

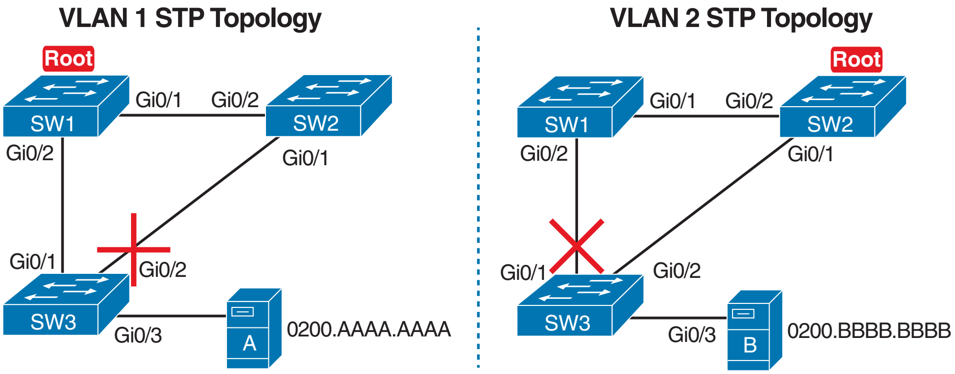


Figure L-4 *Two Different STP Topologies for Same Physical LAN, Two Different VLANs*

Looking at diagrams like those in Figure L-4 makes the forwarding path obvious. Although the figure shows the traffic path, that path is determined by switch MAC learning, which is then impacted by the ports on which STP has set a blocking or discarding state.

For example, consider VLAN 1's STP topology in Figure L-4. Remember, STP blocks on a port on one switch, not on both ends of the link. So, in the case of VLAN 1, SW3's G0/2 port blocks, but SW2's G0/1 does not. Even so, by blocking on a port on one end of the link, that act effectively stops any MAC learning from happening by either device on the link. That is, SW3 learns no MAC addresses on its G0/2 port, and SW2 learns no MAC addresses on its G0/1 port, for these reasons:

- **SW2 learns no MAC addresses on G0/1:** On the blocking (SW3) end of the SW3–SW2 trunk, SW3 will not send frames out that link to SW2, so SW2 will never receive frames from which to learn MAC addresses on SW2's G0/1.
- **SW3 learns no MAC addresses on G0/2:** On the not blocking (SW2) end of the SW3–SW2 trunk, SW2 will flood frames out that port. SW3 receives those frames, but because SW3 blocks, SW3 ignores those received frames and does not learn their MAC addresses.

Given that discussion, can you predict the MAC table entries on each of the three switches for the MAC addresses of servers A and B in Figure L-4? On switch SW2, the entry for server A, in VLAN 1, should refer to SW2's G0/2 port, pointing to SW1 next, matching the figure. But SW2's entry for server B, in VLAN 2, references SW2's G0/1 port, again matching the figure. Example L-3 shows the MAC tables on SW1 and SW2 as a confirmation.

Example L-3 *Examining SW1 and SW2 Dynamic MAC Address Table Entries*

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Gi0/2
  2     0200.BBBB.BBBB    DYNAMIC     Gi0/1
```

```
SW2# show mac address-table dynamic
          Mac Address Table
-------------------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Gi0/2
  2     0200.BBBB.BBBB    DYNAMIC     Gi0/1
```

Predicting EtherChannel Impact on MAC Tables

Most designs use multiple links between switches, with those links configured to be part of an EtherChannel. What does that do to the MAC forwarding logic? In short, the switch uses the PortChannel interfaces, and not the physical interfaces bundled into the EtherChannel, in the MAC address table. Specifically:

MAC learning: Frames received in a physical interface that is part of a PortChannel are considered to arrive on the PortChannel interface. So, MAC learning adds the PortChannel interface rather than the physical interface to the MAC address table.

MAC forwarding: The forwarding process will find a PortChannel port as an outgoing interface when matching the MAC address table. Then the switch must take the additional step to choose the outgoing physical interface, based on the load-balancing preferences configured for that PortChannel.

For example, consider Figure L-5, which updates previous Figure L-4 with two-link PortChannels between each pair of switches. With VLAN 1 blocking again on switch SW3, but this time on SW3's PortChannel3 interface, what MAC table entries would you expect to see in each switch? Similarly, what MAC table entries would you expect to see for VLAN 2, with SW3 blocking on its PortChannel2 interface?

The logic of which entries exist on which ports mirrors the logic with the earlier example surrounding Figure L-4. In this case, the interfaces just happen to be PortChannel interfaces. Example L-4 shows the same command from the same two switches as Example L-3: **show mac address-table dynamic** from both SW1 and SW2. (Note that to save length, the MAC table output shows only the entries for the two servers in Figure L-5.)

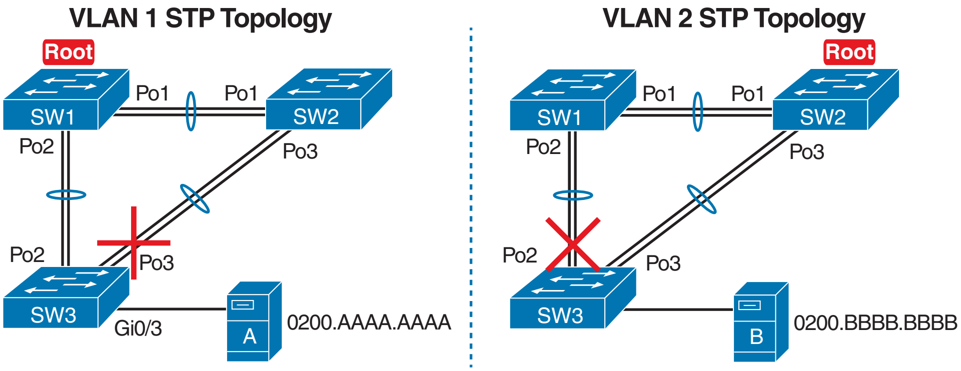


Figure L-5 *VLAN Topology with PortChannels Between Switches*

Example L-4 *SW1 and SW2 MAC Tables with PortChannel Ports Listed*

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Po2
  2     0200.BBBB.BBBB    DYNAMIC     Po1
```

```
SW2# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Po1
  2     0200.BBBB.BBBB    DYNAMIC     Po3
```

Switches use one of many load-balancing options to then choose the physical interface to use after matching MAC table entries like those shown in Example L-4. By default, Cisco Layer 2 switches often default to use a balancing method based on the source MAC address. In particular, the switch looks at the low-order bits of the source MAC address (which are on the far right of the MAC address in written form). This approach increases the chances that the balancing will be spread somewhat evenly based on the source MAC addresses in use.

Choosing the VLAN of Incoming Frames

To wrap up the analysis of switch data plane forwarding, this section mostly reviews topics already discussed, but it serves to emphasize some important points. The topic is simply this: How does a switch know which VLAN a frame is a part of as the frame enters a switch? You have seen all the information needed to answer this question already, but take the time to review.

First, some interfaces trunk, and in those cases, the frame arrives with a VLAN ID listed in the incoming trunking header. In other cases, the frame does not arrive with a trunking header, and the switch must look at local configuration. But because the switch will match both the destination MAC address and the frame VLAN ID when matching the MAC address table, knowing how the switch determines the VLAN ID is important.

The following list reviews and summarizes the key points of how a switch determines the VLAN ID to associate with an incoming frame:

Step 1. If the port is an access port, associate the frame with the configured access VLAN (**switchport access vlan** *vlan_id*).

Step 2. If the port is a voice port, or has both an IP Phone and PC (or other data device) connected to the phone:

- **a.** Associate the frames from the data device with the configured access VLAN (as configured with the **switchport access vlan** *vlan_id* command).
- **b.** Associate the frames from the phone with the VLAN ID in the 802.1Q header (as configured with the **switchport voice vlan** *vlan_id* command).

Step 3. If the port is a trunk, determine the frame's tagged VLAN, or if there is no tag, use that incoming interface's native VLAN ID (**switchport trunk native** *vlan_id*).

This page intentionally left blank

APPENDIX M

Variable-Length Subnet Masks

NOTE This appendix contains content that was published as a chapter in one of the past editions of this book or a related book. The author includes this appendix with the current edition as extra reading for anyone interested in learning more. However, note that the content in this appendix has not been edited since it was published in the earlier edition, so references to exams and exam topics, and to other chapters, will be outdated. This appendix was previously published as Chapter 22 of the book *CCENT/CCNA ICND1 100-105 Official Cert Guide*, published in 2016.

IPv4 addressing and subnetting use a lot of terms, a lot of small math steps, and a lot of concepts that fit together. While learning those concepts, it helps to keep things as simple as possible. One way this book has kept the discussion simpler so far was to show examples that use one mask only inside a single Class A, B, or C network.

This chapter removes that restriction by introducing variable-length subnet masks (VLSM). VLSM simply means that the subnet design uses more than one mask in the same classful network. VLSM has some advantages and disadvantages, but when learning, the main challenge is that a subnetting design that uses VLSM requires more math, and it requires that you think about some other issues as well. This chapter walks you through the concepts, the issues, and the math.

Foundation Topics

VLSM Concepts and Configuration

VLSM occurs when an internetwork uses more than one mask for different subnets of a single Class A, B, or C network. Figure M-1 shows an example of VLSM used in Class A network 10.0.0.0.

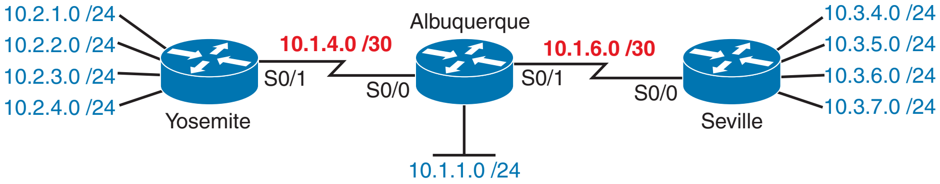


Figure M-1 *VLSM in Network 10.0.0.0: Masks /24 and /30*

Figure M-1 shows a typical choice of using a /30 prefix (mask 255.255.255.252) on point-to-point serial links, with mask /24 (255.255.255.0) on the LAN subnets. All subnets are of Class A network 10.0.0.0, with two masks being used, therefore meeting the definition of VLSM.

Oddly enough, a common mistake occurs when people think that VLSM means "using more than one mask in some internetwork" rather than "using more than one mask *in a single classful network*." For example, if in one internetwork diagram, all subnets of network 10.0.0.0 use a 255.255.240.0 mask, and all subnets of network 11.0.0.0 use a 255.255.255.0 mask, the design uses two different masks. However, Class A network 10.0.0.0 uses only one mask, and Class A network 11.0.0.0 uses only one mask. In that case, the design does not use VLSM.

VLSM provides many benefits for real networks, mainly related to how you allocate and use your IP address space. Because a mask defines the size of the subnet (the number of host addresses in the subnet), VLSM allows engineers to better match the need for addresses with the size of the subnet. For example, for subnets that need fewer addresses, the engineer uses a mask with fewer host bits, so the subnet has fewer host IP addresses. This flexibility reduces the number of wasted IP addresses in each subnet. By wasting fewer addresses, more space remains to allocate more subnets.

VLSM can be helpful for both public and private IP addresses, but the benefits are more dramatic with public networks. With public networks, the address savings help engineers avoid having to obtain another registered IP network number from regional IP address assignment authorities. With private networks, as defined in RFC 1918, running out of addresses is not as big a negative, because you can always grab another private network from RFC 1918 if you run out.

Classless and Classful Routing Protocols

Before you can deploy a VLSM design, you must first use a routing protocol that supports VLSM. To support VLSM, the routing protocol must advertise the mask along with each subnet. Without mask information, the router receiving the update would be confused.

For example, if a router learned a route for 10.1.8.0, but with no mask information, what does that mean? Is that subnet 10.1.8.0/24? 10.1.8.0/23? 10.1.8.0/30? The dotted-decimal number 10.1.8.0 happens to be a valid subnet number with a variety of masks, and because multiple masks can be used with VLSM, the router has no good way to make an educated guess. To effectively support VLSM, the routing protocol needs to advertise the correct mask along with each subnet so that the receiving router knows the exact subnet that is being advertised.

By definition, *classless routing protocols* advertise the mask with each advertised route, and *classful routing protocols* do not. The classless routing protocols, as noted in Table M-1, are the newer, more advanced routing protocols. Not only do these more advanced classless routing protocols support VLSM, but they also support manual route summarization, which allows a routing protocol to advertise one route for a larger subnet instead of multiple routes for smaller subnets.

Table M-1 Classless and Classful Interior IP Routing Protocols

Routing Protocol	Is It Classless?	Sends Mask in Updates?	Supports VLSM?	Supports Manual Route Summarization?
RIPv1	No	No	No	No
RIPv2	Yes	Yes	Yes	Yes
EIGRP	Yes	Yes	Yes	Yes
OSPF	Yes	Yes	Yes	Yes

Beyond VLSM itself, the routing protocols do not have to be configured to support VLSM or to be classless. There is no command to enable or disable the fact that classless routing protocols include the mask with each route. The only configuration choice you must make is to use a classless routing protocol.

VLSM Configuration and Verification

Cisco routers do not configure VLSM, enable or disable it, or need any configuration to use it. From a configuration perspective, VLSM is simply a side effect of using the **ip address** interface subcommand. Routers collectively configure VLSM by virtue of having IP addresses in the same classful network but with different masks.

For example, Example M-1 shows two of the interfaces from router Yosemite from Figure M-1. The example shows the IP address assignments on two interfaces, one with a /24 mask and one with a /30 mask, both with IP addresses in Class A network 10.0.0.0.

Example M-1 *Configuring Two Interfaces on Yosemite, Resulting in VLSM*

```
Yosemite# configure terminal
Yosemite(config)# interface Fa0/0
Yosemite(config-if)# ip address 10.2.1.1 255.255.255.0
Yosemite(config-if)# interface S0/1
Yosemite(config-if)# ip address 10.1.4.1 255.255.255.252
```

The use of VLSM can also be detected by a detailed look at the output of the **show ip route** command. This command lists routes in groups, by classful network, so that you see all the subnets of a single Class A, B, or C network all in a row. Just look down the list, and look to see, if any, how many different masks are listed. For example, Example M-2 lists the routing table on Albuquerque from Figure M-1; Albuquerque uses masks /24 and /30 inside network 10.0.0.0, as noted in the highlighted line in the example.

Example M-2 *Albuquerque Routing Table with VLSM*

```
Albuquerque# show ip route
! Legend omitted for brevity

     10.0.0.0/8 is variably subnetted, 14 subnets, 3 masks
D       10.2.1.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D       10.2.2.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D       10.2.3.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D       10.2.4.0/24 [90/2172416] via 10.1.4.1, 00:00:34, Serial0/0
D       10.3.4.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D       10.3.5.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D       10.3.6.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
D       10.3.7.0/24 [90/2172416] via 10.1.6.2, 00:00:56, Serial0/1
C       10.1.1.0/24 is directly connected, FastEthernet0/0
L       10.1.1.1/32 is directly connected, FastEthernet0/0
C       10.1.6.0/30 is directly connected, Serial0/1
L       10.1.6.1/32 is directly connected, Serial0/1
C       10.1.4.0/30 is directly connected, Serial0/0
L       10.1.4.1/32 is directly connected, Serial0/0
```

NOTE For the purposes of understanding whether a design uses VLSM, ignore the /32 "local" routes that a router automatically creates for its own interface IP addresses.

So ends the discussion of VLSM as an end to itself. This chapter is devoted to VLSM, but it took a mere three to four pages to fully describe it. Why the entire VLSM chapter? Well, to work with VLSM, to find problems with it, to add subnets to an existing design, and to design using VLSM from scratch—in other words, to apply VLSM to real networks—takes skill and practice. To do these same tasks on the exam requires skill and practice. The rest of this chapter examines the skills to apply VLSM and provides some practice for these two key areas:

- Finding VLSM overlaps
- Adding new VLSM subnets without overlaps

Finding VLSM Overlaps

Regardless of whether a design uses VLSM, the subnets used in any IP internetwork design should not overlap their address ranges. When subnets in different locations overlap their addresses, a router's routing table entries overlap. As a result, hosts in different locations can be assigned the same IP address. Routers clearly cannot route packets correctly in these cases. In short, a design that uses overlapping subnets is considered to be an incorrect design and should not be used.

This section begins with a short discussion about VLSM design, to drive home the ideas behind VLSM overlaps. It then gets into an operational and troubleshooting approach to the topic, by looking at existing designs and trying to find any existing overlaps.

Designing Subnetting Plans with VLSM

When creating a subnetting plan using VLSM, you have to be much more careful in choosing what subnets to use. First, whatever masks you use in a VLSM design, each subnet ID must be a valid subnet ID given the mask that you use for that subnet.

For example, consider a subnet plan for Class B network 172.16.0.0. To create a subnet with a /24 mask, the subnet ID must be a subnet ID that you could choose if you subnetted the whole Class B network with that same mask. Appendix L, "Subnet Design," discusses how to find those subnets in depth, but with a Class B network and a /24 mask, the possible subnet IDs should be easy to calculate by now: 172.16.0.0 (the zero subnet), then 172.16.1.0, 172.16.2.0, 172.16.3.0, 172.16.4.0, and so on, up through 172.16.255.0.

NOTE Subnet IDs must always follow this important binary rule as noted back in Chapter 14, "Analyzing Existing Subnets": In binary, each subnet ID has a host field of all binary 0s. If you use the math and processes to find all subnet IDs per Appendix L, all those subnet IDs happen to have binary 0s in the host fields.

Now expand your thinking about subnet IDs to a VLSM design. To begin, you would decide that you need some subnets with one mask, other subnets with another mask, and so on, to meet the requirements for different sizes of different subnets. For instance, imagine you start with a brand-new VLSM design, with Class B network 172.16.0.0. You plan to have some subnets with /22 masks, some with /23, and some with /24. You might develop then a planning diagram, or at least draw the ideas, with something like Figure M-2.

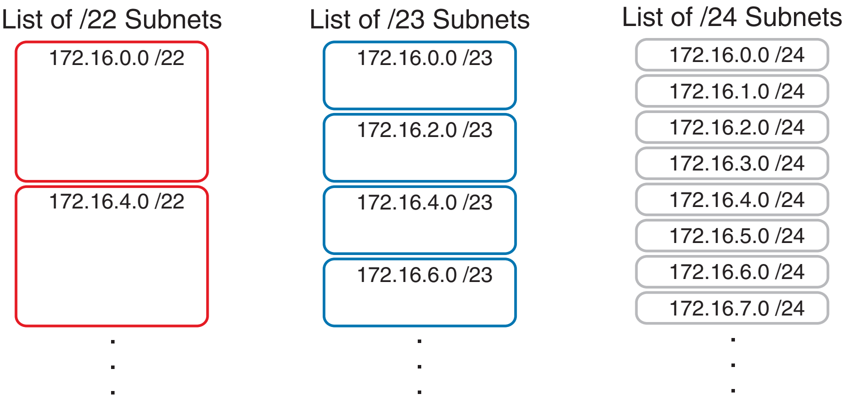


Figure M-2 *Possible Subnet IDs of Network 172.16.0.0, with /22, /23, and /24 Masks*

The drawing shows the first few subnet IDs available with each mask, but you cannot use all subnets from all three lists in a design. As soon as you choose to use one subnet from any column, you remove some subnets from the other lists because subnets cannot overlap. Overlapping subnets are subnets whose range of addresses include some of the same addresses.

As an example, Figure M-3 shows the same list of the first few possible /22, /23, and /24 subnets of Class B network 172.16.0.0. However, it shows a check mark beside two subnets that have been allocated for use; that is, on paper, the person making the subnetting plan has decided to use these two subnets somewhere in the network. The subnets with a dark gray shading and an *X* in them can no longer be used because they have some overlapping addresses with the subnets that have check marks (172.16.3.0/24 and 172.16.4.0/22).

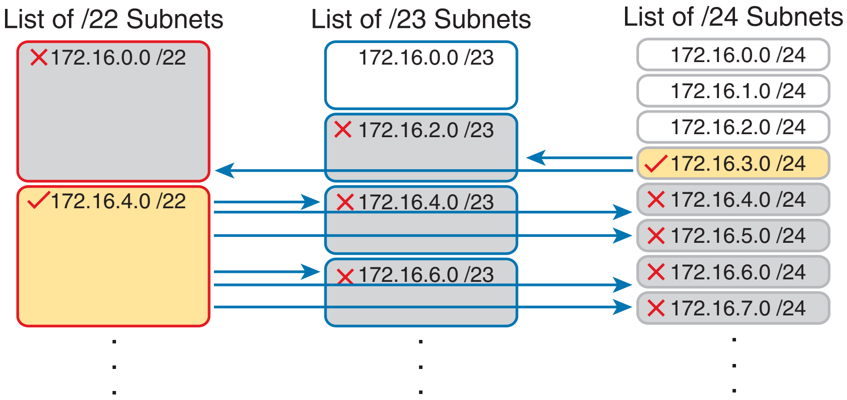


Figure M-3 *Selecting Two Subnets Disallows Other Subnets in Different Columns*

Just to complete the example, first look at subnet 172.16.4.0 on the lower left. That subnet includes addresses from the subnet ID of 172.16.4.0 through the subnet broadcast address of 172.16.7.255. As you can see just by looking at the subnet IDs to the right, all the subnets referenced with the arrowed lines are within that same range of addresses.

Now look to the upper right of the figure, to subnet 172.16.3.0/24. The subnet has a range of 172.16.3.0–172.16.3.255 including the subnet ID and subnet broadcast address. That subnet overlaps with the two subnets referenced to the left. For instance, subnet 172.16.0.0/22 includes the range from 172.16.0.0–172.16.3.255. But because there is some overlap, once the design has allocated the 172.16.3.0/24 subnet, the 172.16.2.0/23 and 172.16.0.0/22 subnets could not be used without causing problems, because:

> A subnetting design, whether using VLSM or not, should not allow subnets whose address ranges overlap. If overlapping subnets are implemented, routing problems occur and some hosts simply cannot communicate outside their subnets.

These address overlaps are easier to see when not using VLSM. When not using VLSM, overlapped subnets have identical subnet IDs, so to find overlaps, you just have to look at the subnet IDs. With VLSM, overlapped subnets may not have the same subnet ID, as was the case in this most recent example with the subnets across the top of Figure M-3. To find these overlaps, you have to look at the entire range of addresses in each subnet, from subnet ID to subnet broadcast address, and compare the range to the other subnets in the design.

An Example of Finding a VLSM Overlap

For example, imagine that a practice question for the CCENT exam shows Figure M-4. It uses a single Class B network (172.16.0.0), with VLSM, because it uses three different masks: /23, /24, and /30.

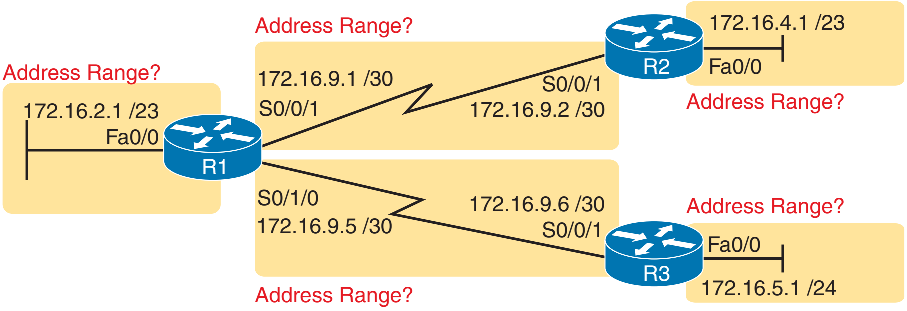


Figure M-4 *VLSM Design with Possible Overlap*

Now imagine that the exam question shows you the figure, and either directly or indirectly asks whether overlapping subnets exist. This type of question might simply tell you that some hosts cannot ping each other, or it might not even mention that the root cause could be that some of the subnets overlap. To answer such a question, you could follow this simple but possibly laborious process:

Step 1. Calculate the subnet ID and subnet broadcast address of each subnet, which gives you the range of addresses in that subnet.

Step 2. List the subnet IDs in numerical order (along with their subnet broadcast addresses).

Step 3. Scan the list from top to bottom, comparing each pair of adjacent entries, to see whether their range of addresses overlaps.

For example, Table M-2 completes the first two steps based on Figure M-4, listing the subnet IDs and subnet broadcast addresses, in numerical order based on the subnet IDs.

Table M-2 Subnet IDs and Broadcast Addresses, in Numerical Order, from Figure M-4

Subnet	Subnet Number	Broadcast Address
R1 LAN	172.16.2.0	172.16.3.255
R2 LAN	172.16.4.0	172.16.5.255
R3 LAN	172.16.5.0	172.16.5.255
R1-R2 serial	172.16.9.0	172.16.9.3
R1-R3 serial	172.16.9.4	172.16.9.7

The VLSM design is invalid in this case because of the overlap between R2's LAN subnet and R3's LAN subnet. As for the process, Step 3 states the somewhat obvious step of comparing the address ranges to see whether any overlaps occur. Note that, in this case, none of the subnet numbers are identical, but two entries (highlighted) do overlap. The design is invalid because of the overlap, and one of these two subnets would need to be changed.

As far as the three-step process works, note that if two adjacent entries in the list overlap, compare three entries at the next step. The two subnets already marked as overlapped can overlap with the next subnet in the list. For example, the three subnets in the following list overlap in that the first subnet overlaps with the second and third subnets in the list. If you followed the process shown here, you would have first noticed the overlap between the first two subnets in the list, so you would then also need to check the next subnet in the list to find out if it overlapped.

10.1.0.0/16 (subnet ID 10.1.0.0, broadcast 10.1.255.255)

10.1.200.0/24 (subnet ID 10.1.200.0, broadcast 10.1.200.255)

10.1.250.0/24 (subnet ID 10.1.250.0, broadcast 10.1.250.255)

Practice Finding VLSM Overlaps

As typical of anything to do with applying IP addressing and subnetting, practice helps. To that end, Table M-3 lists three practice problems. Just start with the five IP addresses listed in a single column, and then follow the three-step process outlined in the previous section to find any VLSM overlaps. The answers can be found near the end of this chapter, in the section "Answers to Earlier Practice Problems."

Table M-3 VLSM Overlap Practice Problems

Problem 1	Problem 2	Problem 3
10.1.34.9/22	172.16.126.151/22	192.168.1.253/30
10.1.29.101/23	172.16.122.57/27	192.168.1.113/28
10.1.23.254/22	172.16.122.33/30	192.168.1.245/29
10.1.17.1/21	172.16.122.1/30	192.168.1.125/30
10.1.1.1/20	172.16.128.151/20	192.168.1.122/30

Adding a New Subnet to an Existing VLSM Design

The task described in this section happens frequently in real networks: choosing new subnets to add to an existing design. In real life, you can use IP Address Management (IPAM) tools that help you choose a new subnet so that you do not cause an overlap. However, for the CCNA exam, you need to be ready to do the mental process and math of choosing a subnet that does not create an overlapped VLSM subnet condition. In other words, you need to pick a new subnet and not make a mistake!

For example, consider the internetwork shown earlier in Figure M-2, with classful network 172.16.0.0. An exam question might suggest that a new subnet, with a /23 prefix length, needs to be added to the design. The question might also say, "Pick the numerically lowest subnet number that can be used for the new subnet." In other words, if both 172.16.4.0 and 172.16.6.0 would work, use 172.16.4.0.

So, you really have a couple of tasks: To find all the subnet IDs that could be used, rule out the ones that would cause an overlap, and then check to see whether the question guides you to pick either the numerically lowest (or highest) subnet ID. This list outlines the specific steps:

Step 1. Pick the subnet mask (prefix length) for the new subnet, based on the design requirements (if not already listed as part of the question).

Step 2. Calculate all possible subnet numbers of the classful network using the mask from Step 1, along with the subnet broadcast addresses.

Step 3. Make a list of existing subnet IDs and matching subnet broadcast addresses.

Step 4. Compare the existing subnets to the candidate new subnets to rule out overlapping new subnets.

Step 5. Choose the new subnet ID from the remaining subnets identified at Step 4, paying attention to whether the question asks for the numerically lowest or numerically highest subnet ID.

An Example of Adding a New VLSM Subnet

For example, Figure M-5 shows an existing internetwork that uses VLSM. (The figure uses the same IP addresses as shown in Figure M-4, but with R3's LAN IP address changed to fix the VLSM overlap shown in Figure M-4.) In this case, you need to add a new subnet to support 300 hosts. Imagine that the question tells you to use the smallest subnet (least number of hosts) to meet that requirement. You use some math and logic you learned earlier in your study to choose mask /23, which gives you 9 host bits, for $2^9 - 2 = 510$ hosts in the subnet.

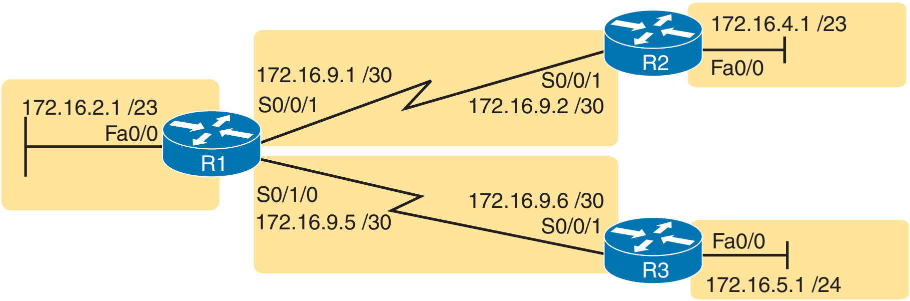


Figure M-5 *Internetwork to Which You Need to Add a /23 Subnet, Network 172.16.0.0*

At this point, just follow the steps listed before Figure M-5. For Step 1, you have already been given the mask (/23). For Step 2, you need to list all the subnet numbers and broadcast addresses of 172.16.0.0, assuming the /23 mask. You will not use all these subnets, but you need the list for comparison to the existing subnets. Table M-4 shows the results, at least for the first five possible /23 subnets.

Table M-4 First Five Possible /23 Subnets

Subnet	Subnet Number	Subnet Broadcast Address
First (zero)	172.16.0.0	172.16.1.255
Second	172.16.2.0	172.16.3.255
Third	172.16.4.0	172.16.5.255
Fourth	172.16.6.0	172.16.7.255
Fifth	172.16.8.0	172.16.9.255

Next, at Step 3, list the existing subnet numbers and broadcast addresses, as shown earlier in Figure M-5. To do so, do the usual math to take an IP address/mask to then find the subnet ID and subnet broadcast address. Table M-5 summarizes that information, including the locations, subnet numbers, and subnet broadcast addresses.

Table M-5 Existing Subnet IDs and Broadcast Addresses from Figure M-5

Subnet	Subnet Number	Subnet Broadcast Address
R1 LAN	172.16.2.0	172.16.3.255
R2 LAN	172.16.4.0	172.16.5.255
R3 LAN	172.16.6.0	172.16.6.255
R1-R2 serial	172.16.9.0	172.16.9.3
R1-R3 serial	172.16.9.4	172.16.9.7

At this point, you have all the information you need to look for the overlap at Step 4. Simply compare the range of numbers for the subnets in the previous two tables. Which of the possible new /23 subnets (Table M-4) overlap with the existing subnets (Table M-5)? In this case, the second through fifth subnets in Table M-4 overlap, so rule those out as candidates to be used. (Table M-4 denotes those subnets with gray highlights.)

Step 5 has more to do with the exam than with real network design, but it is still worth listing as a separate step. Multiple-choice questions sometimes need to force you into a single answer, and asking for the numerically lowest or highest subnet does that. This particular example asks for the numerically lowest subnet number, which in this case is 172.16.0.0/23.

NOTE The answer, 172.16.0.0/23, happens to be a zero subnet. For the exam, the zero subnet should be avoided if (a) the question implies the use of classful routing protocols or (b) the routers are configured with the **no ip subnet-zero** global configuration command. Otherwise, assume that the zero subnet can be used.

Answers to Earlier Practice Problems

Answers to Practice Finding VLSM Overlaps

This section lists the answers to the three practice problems in the section "Practice Finding VLSM Overlaps," as listed earlier in Table M-3. Note that the tables that list details of the answer reordered the subnets as part of the process.

In Problem 1, the second and third subnet IDs listed in Table M-6 happen to overlap. The second subnet's range completely includes the range of addresses in the third subnet.

Table M-6 VLSM Overlap Problem 1 Answers (Overlaps Highlighted)

Reference	Original Address and Mask	Subnet ID	Broadcast Address
1	10.1.1.1/20	10.1.0.0	10.1.15.255
2	10.1.17.1/21	10.1.16.0	10.1.23.255
3	10.1.23.254/22	10.1.20.0	10.1.23.255
4	10.1.29.101/23	10.1.28.0	10.1.29.255
5	10.1.34.9/22	10.1.32.0	10.1.35.255

In Problem 2, again the second and third subnet IDs (listed in Table M-7) happen to overlap, and again, the second subnet's range completely includes the range of addresses in the third subnet. Also, the second and third subnet IDs are the same value, so the overlap is more obvious.

Table M-7 VLSM Overlap Problem 2 Answers (Overlaps Highlighted)

Reference	Original Address and Mask	Subnet ID	Broadcast Address
1	172.16.122.1/30	172.16.122.0	172.16.122.3
2	172.16.122.57/27	172.16.122.32	172.16.122.63
3	172.16.122.33/30	172.16.122.32	172.16.122.35
4	172.16.126.151/22	172.16.124.0	172.16.127.255
5	172.16.128.151/20	172.16.128.0	172.16.143.255

In Problem 3, three subnets overlap. Subnet 1's range completely includes the range of addresses in the second and third subnets, as shown in Table M-8. Note that the second and third subnets do not overlap with each other, so for the process in this book to find all the overlaps, after you find that the first two subnets overlap, you should compare the next entry in the table (3) with both of the two known-to-overlap entries (1 and 2).

Table M-8 VLSM Overlap Problem 3 Answers (Overlaps Highlighted)

Reference	Original Address and Mask	Subnet ID	Broadcast Address
1	192.168.1.113/28	192.168.1.112	192.168.1.127
2	192.168.1.122/30	192.168.1.120	192.168.1.123
3	192.168.1.125/30	192.168.1.124	192.168.1.127
4	192.168.1.245/29	192.168.1.240	192.168.1.247
5	192.168.1.253/30	192.168.1.252	192.168.1.255

This page intentionally left blank

Pearson

Where are the companion content files?

Thank you for purchasing this Premium Edition version of CCNA 200-301 Official Cert Guide, Volume 2, 2nd Edition

This product comes with companion content. You have access to these files by following the steps below:

1. Go to **ciscopress.com/account** and log in.
2. Click on the "Access Bonus Content" link in the Registered Products section of your account page for this product, to be taken to the page where your downloadable content is available.

Please note that many of our companion content files can be very large, especially image and video files.

If you are unable to locate the files for this title by following the steps at left, please visit **ciscopress.com/support** and select the chat, phone, or web ticket options to get help from a tech support representative.

The Professional and Personal Technology Brands of Pearson

Addison Wesley Cisco Press informIT PEARSON IT Certification QUE SAMS

CCNA 200-301, Volume 1
Official Cert Guide

2nd Edition

In addition to the wealth of updated content, this new edition includes a series of free hands-on exercises to help you master several real-world configuration and troubleshooting activities. These exercises can be performed on the CCNA 200-301 Network Simulator Lite, Volume 1 software included for free on the companion website that accompanies this book. This software, which simulates the experience of working on actual Cisco routers and switches, contains the following 21 free lab exercises, covering topics in Part II and Part III, the first hands-on configuration sections of the book:

1. **Configuring Local Usernames**
2. **Configuring Hostnames**
3. **Interface Status I**
4. **Interface Status II**
5. **Interface Status III**
6. **Interface Status IV**
7. **Configuring Switch IP Settings**
8. **Switch IP Address**
9. **Switch IP Connectivity I**
10. **Switch CLI Configuration Process I**
11. **Switch CLI Configuration Process II**
12. **Switch CLI Exec Mode**
13. **Setting Switch Passwords**
14. **Interface Settings I**
15. **Interface Settings II**
16. **Interface Settings III**
17. **Switch Forwarding I**
18. **Switch Security I**
19. **Switch Interfaces and Forwarding Configuration Scenario**
20. **Configuring VLANs Configuration Scenario**
21. **VLAN Troubleshooting**

If you are interested in exploring more hands-on labs and practice configuration and troubleshooting with more router and switch commands, go to **www.pearsonitcertification.com/networksimulator** for demos and to review the latest products for sale.

Network Simulator System Requirements

Windows (Minimum)

- Windows 11, Windows 10, Windows 8.1
- Intel Pentium III (1GHz or faster processor) or 2 GB RAM
- 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
- DirectX 9 graphics device with WDDM 1.0 or higher driver
- Adobe Acrobat Reader version 8 and above
- Connection to the Internet during installation for access code validation

Mac (Minimum)

- macOS 14, 13, and 12
- Intel Pentium III (1GHz or faster processor)
- 512 MB RAM (1 GB recommended)
- 1.5 GB hard disk space
- 32-bit color depth at 1024x768 resolution
- Adobe Acrobat Reader version 8 and above
- Connection to the Internet during installation for access code validation