

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 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.

protected access credential (PAC) Special-purpose data that is used as an authentication credential in EAP-FAST.

Protected EAP (PEAP) An authentication method that uses a certificate on the AS for outer authentication and a TLS tunnel for inner authentication. Clients can provide their credentials through either MS-CHAPv2 or GTC.

Protected Management Frame (PMF) A service provided by WPA3 that protects a set of 802.11 robust management and action frames, to prevent spoofing of AP functions.

protocol data unit (PDU) A generic term referring to the header defined by some layer of a networking model, and the data encapsulated by the header (and possibly trailer) of that layer, but specifically not including any lower-layer headers and trailers.

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.

Public Key Infrastructure (PKI) An enterprisewide system that generates and revokes digital certificates for client authentication.

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 802.1w. Defines an improved version of STP that converges much more quickly and consistently than STP (802.1d).

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.

roaming The process a wireless client uses to move from one AP to another as it changes location.

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

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 port 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 port on a multilayer Cisco switch, configured with the `no switchport` command, that tells the switch to treat the port as if it were a Layer 3 port, like a router interface.

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).

serial cable A type of cable with many different styles of connectors used to connect a router to an external CSU/DSU on a leased-line installation.

serial interface A type of interface on a router, used to connect to some types of WAN links, particularly leased lines and Frame Relay access links.

service set identifier (SSID) A text string that is used to identify a wireless network.

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.

Simultaneous Authentication of Equals (SAE) A strong authentication method used in WPA3 to authenticate wireless clients and APs and to prevent dictionary attacks for discovering pre-shared keys.

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 802.1D. 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.

split-MAC architecture A wireless AP strategy based around the idea that normal AP functions are split or divided between a wireless LAN controller and lightweight APs.

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.

station (STA) An 802.11 client device that is associated with a BSS.

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 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.

supplicant An 802.1x entity that exists as software on a client device and serves to request network access.

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 port on a multilayer Cisco switch or a Layer 2 switch, configured with the normal default interface setting of switchport, that tells the switch to treat the port as if it were a Layer 2 port, resulting in the switch performing switch MAC learning, Layer 2 forwarding, and STP on that interface.

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

symmetric A feature of many Internet access technologies in which the downstream transmission rate is the same as the upstream transmission rate.

synchronous The imposition of time ordering on a bit stream. Practically, a device will try to use the same speed as another device on the other end of a serial link. However, by examining transitions between voltage states on the link, the device can notice slight variations in the speed on each end and can adjust its speed accordingly.

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

T1 A line from the telco that allows transmission of data at 1.544 Mbps, with the ability to treat the line as 24 different 64-kbps DS0 channels (plus 8 kbps of overhead).

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.

Temporal Key Integrity Protocol (TKIP) A wireless security scheme developed before 802.11i that provides a MIC for data integrity, a dynamic method for per-frame WEP encryption keys, and a 48-bit initialization vector. The MIC also includes a time stamp and the sender's MAC address

three-tier design *See* core design.

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.

transparent bridge The name of a networking device that was a precursor to modern LAN switches. Bridges forward frames between LAN segments based on the destination MAC address. Transparent bridging is so named because the presence of bridges is transparent to network end nodes.

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.

two-tier design *See* collapsed core design.

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.

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.

unified WLC deployment A wireless network design that places a WLC centrally within a network topology.

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

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 configuration database The name of the collective configuration of VLAN IDs and names on a Cisco switch.

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 client mode One of three VTP operational modes for a switch with which switches learn about VLAN numbers and names from other switches, but which does not allow the switch to be directly configured with VLAN information.

VTP server mode One of three VTP operational modes. Switches in server mode can configure VLANs, tell other switches about the changes, and learn about VLAN changes from other switches.

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.

Wi-Fi Protected Access (WPA) The first version of a Wi-Fi Alliance standard that requires pre-shared key or 802.1x authentication, TKIP, and dynamic key management; based on parts of the 802.11i amendment before it was ratified.

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

window Represents the number of bytes that can be sent without receiving an acknowledgment.

Wired Equivalent Privacy (WEP) An 802.11 authentication and encryption method that requires clients and APs to use a common WEP key.

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).

wireless LAN Controller (WLC) A device that cooperates with wireless lightweight access points (LWAP) to create a wireless LAN by performing some control functions for each LWAP and forwarding data between each LWAP and the wired LAN.

WLAN client A wireless device that wants to gain access to a wireless access point for the purpose of communicating with other wireless devices or other devices connected to the wired internetwork.

workgroup bridge (WGB) An AP that is configured to bridge between a wired device and a wireless network. The WGB acts as a wireless client.

WPA Version 2 (WPA2) The second version of a Wi-Fi Alliance standard that requires pre-shared key or 802.1x authentication, TKIP or CCMP, and dynamic encryption key management; based on the complete 802.11i amendment after its ratification.

WPA Version 3 (WPA3) The third version of a Wi-Fi Alliance standard introduced in 2018 that requires pre-shared key or 802.1x authentication, GCMP, SAE, and forward secrecy.

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

Symbols

? command, 94-95
:: (double colon), 531

Numbers

2-way state (OSPF), 453-454, 457
2.4-GHz band, 626
5-GHz band, 626
10BASE-T, 37, 42-45
10GBASE-T, 37
100BASE-T, 37, 42-45
802.11, 628-629
 BSS, 614-616
 DS, 616-618
 ESS, 618
 IBSS, 619
 WLAN, 614
802.1D STP, 228, 232
802.1Q, 182
802.1w RSTP, 228-232
802.1x, EAP integration, 658
1000BASE-LX, 37
1000BASE-T, UTP cabling pinouts, 45-46

A

AAA (Authentication, Authorization, and Accounting) servers, 136
abbreviating IPv6 addresses, 531-532

ABR (Area Border Routers), 460-461

access

 CLI, 87-94, 128-139, 355-356
 protected credentials, 659
 WPA, 662-663
 WPA2, 662-663
 WPA3, 662-663

access interfaces, 185

access points. *See* AP

access switches, 241

ad hoc wireless networks. *See* IBSS

addresses

 BIA, 52
 broadcast addresses, 50-52
 calculating hosts and subnets in networks, 313-315
 classless versus classful addressing, 312-313
 Ethernet addresses, 50-52
 exhaustion, 525
 experimental, 290
 first usable, 293-294
 group addresses, 51
 host addresses, 293
 IPv4 addresses. *See* individual entry
 IPv6 addresses. *See* individual entry
 LAN addresses, 52
 last usable, 293-294
 loopback address, 295
 MAC addresses, 50-52, 111-114, 117-124, 218
 multicast addresses, 50-52, 290
 NAT, 277
 network broadcast addresses, 293-295

- network numbers, 293-295
- NIC addresses, 52
- prefix part, 309-311
- private addresses, 542
- public addresses, 542
- range of subnet addresses, finding, 331
- sender MAC, 661
- subnet addresses, 272, 283, 324-327, 334-338
- unicast addresses, 50-52, 290, 322
- universal addresses, 51
- adjacencies (OSPF neighbors), trouble-shooting, 510-516**
- adjacent-layer interaction, 21-22**
- adjacent neighbors, 457**
- administrative distance, 382-383, 448-449, 594-595**
- administrative mode, trunking, 191**
- administratively shutdown interfaces, 217**
- AES (Advanced Encryption Standard), 661**
- aging MAC address tables, 121-122**
- algorithms**
 - AES, 661
 - CSMA/CD, 55
 - Dijkstra SPF, 451
 - IGP routing protocol algorithm, 445
 - key mixing, 661
 - RC4 cipher, 657
 - SPF, 457-459
 - STA, 216
- alternate ports, 229-232**
- anycast addresses (IPv6), 574-576**
- AP (Access Points), 35, 614, 629**
 - authentication, 654
 - autonomous, 634-635, 638
 - Bridge mode, 647
 - BSSID, 615
 - cloud-based AP architectures, 636-637
 - ESS, 618
 - fake, 654
 - Flex+Bridge mode, 647
 - FlexConnect mode, 647
 - IBSS, 619
 - LAP, 638-640
 - Local mode, 647
 - management interface, 674
 - Monitor mode, 647
 - multiple SSID, supporting, 617
 - noninfrastructure modes, 620-622
 - passing through, 615
 - roaming, 618
 - Rogue Detector mode, 647
 - SE Connect mode, 647
 - Sniffer mode, 647
 - SSID, 615
 - VLAN, 668
 - WLAN, 668-669
- application layer (TCP/IP), 19-20**
- architectures**
 - autonomous, 634-635, 638
 - centralized, 642-643
 - cloud-based
 - AP, 636-637
 - WLC deployments, 643
 - networking, 16
 - split-MAC, 638-642
- area design (OSPF), 459-462**
- ARIN (American Registry for Internet Numbers), 445**
- ARP (Address Resolution Protocol), 72, 77, 378-379**
- AS (Authentication Servers), 658**
- AS (Autonomous Systems), 444-445**
- ASN (AS Numbers), 445**
- assigning**
 - IPv6 addresses to hosts, 550
 - IPv6 subnets to internetwork topology, 549
 - subnets to different locations, 285

authentication. *See also security*

- AP, 654
- AS, 658
- clients, 653
- EAP, 657-658
- EAP-FAST, 659
- EAP-TLS, 660
- external authentication servers, 135-136
- LEAP, 659
- open authentication, 656
- PEAP, 659
- web (WebAuth), 657
- WEP, 657
- WLAN, 682
- WLC, 642
- WPA, 662-663
- WPA2, 662-663
- WPA3, 662-663

authenticators, 658

auto-cost reference-bandwidth
command, 493, 496

auto-mdix, 45

autonegotiation, 158-162

autonomous AP (Access Points),
634-635, 638

autonomous architectures, 634-635,
638

autonomous systems. *See AS*

auxiliary ports (routers), 362

B

backbone areas, 460-461

backbone routers, 461

backup ports, 230, 233

bandwidth

- frequencies, 626-627
- reference, 492
- router serial interfaces, 361

bandwidth command, 492, 496

Basic Service Areas. *See BSA*

Basic Service Sets. *See BSS*

BDR (Backup DR), 456-457, 504-506

Bellman-Ford protocols. *See distance*
vector protocols

Berners-Lee, Tim, 20

BGP (Border Gateway Protocol), 445

BIA (Burned-In Addresses), 52

BID (Bridge ID)

- STP, 218-219

- system ID extensions, 243-244

bidirectional communication, 613

binary/hexadecimal conversion chart
(IPv6), 531

binary masks, 304-308

binary subnet analysis, 326

- binary practice problems, 328-329

- Boolean math, 331

- finding

- range of addresses,* 331

- subnet ID,* 327

- shortcut for binary process, 330

blocking state, interfaces, 215-217

blueprint (networking), 16

Boolean AND, 331

Boolean math, 331

Boolean OR, 331

borrowing host bits to create subnet
bits, 280-281

BPDU (Bridge Protocol Data Units),
218, 225

BPDU Guard, 236

BPDU tunneling, 247

bridge ID. *See BID*

Bridge mode (AP), 647

bridges. *See switches*

bridging tables. *See MAC address*
tables

broadcast addresses, 50-52, 325-327

broadcast network type (OSPF),
500-506

broadcast storms, 213-215

BSA (Basic Service Areas), 614

BSS (Basic Service Sets), 614-618, 629

AP, 614

associations, 615

BSSID, 615

DS, 616-618

IBSS, 619

stations, 615

traffic flows, 615

burned-in MAC addresses, 218

C

CA (Certificate Authorities), 659

cables

CLI, cabling console connections,
88-90

enterprise networks, 351

Ethernet, 35

fiber-optic cabling, 38, 46-49

IP telephony, 197

leased-line cabling, 62-63

physical console connections, 88-90

pinouts

rollover pinouts, 89

straight-through cable pinout,
42-45

UTP, 37-46, 49

caches (ARP), 77

CAM (Content-Addressable Memory)

tables. *See* MAC address tables

candidate default routes, 384

CAPWAP (Control and Provisioning
of Wireless Access Points) tunneling
protocol, 639-640

carrier sense multiple access with col-
lision detection (CSMA/CD), 55

CCMP (Counter/CBC-MAC Protocol),
661

cells. *See* BSA

centralized architectures, 642-643

centralized controllers

dynamic interfaces, creating, 678

RADIUS servers, configuration, 676

WLAN security, 682

certificate authorities. *See* CA

CFN (Cisco Feature Navigator), 404

channel-group command, 248-249,
259

EtherChannels, 416

Layer 3 EtherChannels, trouble-
shooting, 413

channel-group number mode on
command, 411

channels, 627

dynamic assignment, 642

nonoverlapping, 628

CIDR (Classless Interdomain Routing),
subnet masks, 305

circuits. *See* leased-line WAN

Cisco Binary Game, 306

Cisco Catalyst switches, 86

Cisco integrated services routers, 352

cladding (fiber-optic cable), 47

Class A networks, 290-295, 312

Class B networks, 290-293, 312

Class C networks, 290-295, 312

Class D networks, 290

Class E networks, 290

classful IP addresses, 312-313

classful IP networks, 289, 296-297

address formats, 291-292

before subnetting, 279-280

calculating hosts per network, 293

classes in, 290-291

default masks, 292

network ID, 293-295

number of, 291

octet values, 290

size of, 291

- subnet masks, 302
 - unusual addresses, 295
- classful networks, 276-279**
- classful routing protocols, 447-448**
- classless addressing, 312-313**
- classless routing protocols, 447-448**
- clear ip arp [ip-address] command, 378, 391**
- clear ip ospf process command, 481, 497**
- clear mac address-table dynamic command, 122, 125**
- CLI (Command-Line Interface)**
 - accessing, 87-94
 - cabling console connections, 88-90
 - Cisco Catalyst switches, 86
 - command edit and recall, 95
 - common command prompts, 98
 - configuration files, 99-102
 - configuration mode, 96-97
 - configuration submodes and contexts, 97-99
 - help, 94-95
 - overview, 84-86
 - privileged EXEC mode, 91-93
 - router CLI, 355-356
 - security, 128-139
 - user EXEC mode, 91-93
- clients**
 - authentication, 653, 656-660
 - load balancing, 642
 - roaming, 642
 - Telnet clients, 91
 - WLAN, 684
- CLN (Cisco Learning Network), 306**
- clock rates, router serial interfaces, 361**
- cloud-based architectures, 636-637, 643**
- collisions, 167**

- commands**
 - ?, 94-95
 - auto-cost reference-bandwidth, 496
 - bandwidth, 496
 - channel-group, 248-249, 259, 413, 416
 - channel-group number mode on, 411
 - clear ip arp [ip-address], 378, 391
 - clear ip ospf process, 481, 497
 - clear mac address-table dynamic, 122, 125
 - com?, 94
 - command, 495
 - command ?, 94
 - command parm?, 94
 - command parm<Tab>, 94
 - command parm1 ?, 94
 - configure terminal, 97, 101, 104, 132, 189, 355
 - copy, 356
 - copy running-config startup-config, 102-104
 - copy startup-config running-config, 104
 - crypto key, 137
 - crypto key generate rsa, 137-139, 148
 - debug, 96
 - default-information originate, 489, 496
 - default-information originate always, 490
 - delete vlan.dat, 117
 - description, 153, 170, 363
 - disable, 104
 - duplex, 152-154, 165, 170, 355, 363
 - enable, 91, 104, 130
 - enable password, 131
 - enable secret, 131, 148
 - enable secret love, 94
 - encapsulation, 397-398
 - encapsulation dot1q, 415
 - encapsulation dot1q vlan_id, 397

- encapsulation dot1q vlan-id, 401
- end, 104, 355
- erase nvram, 104
- erase startup-config, 104, 117
- exec-timeout, 145, 148
- exit, 98, 101-103, 355
- history size, 145, 148
- hostname, 99-103, 117, 138, 148
- hostname Fred, 97
- how interfaces status, 156
- interface, 97, 103, 169, 185, 198, 356, 363, 391, 415
- interface ethernet, 357
- interface fastethernet, 357
- interface gigabitethernet, 357
- interface loopback, 470, 481, 496
- interface port-channel, 416
- interface port-channel number, 411
- interface range, 154, 169, 187
- interface type number.subint, 397
- interface vlan, 148, 415
- interface vlan 1, 142
- interface vlan vlan_id, 403
- ip -6 neighbor show, 600
- ip address, 142, 148, 360, 363, 381, 391-392, 397-398, 470
- ip address address mask, 397, 403, 411
- ip address dhcp, 148
- ip default-gateway, 142, 148
- ip domain-name, 139
- ip mtu, 515
- ip name-server, 142, 148
- ip ospf, 495
- ip ospf cost, 492, 496
- ip ospf dead-interval, 517
- ip ospf hello-interval, 517
- ip ospf process-id, 511
- ip ospf process-id area area-id, 483-485
- ip route, 367, 376, 380-385, 391
- ip routing, 391, 402-404, 415
- ip ssh version 2, 139
- ipv6 address, 557, 560, 564-568, 576-578, 583
- ipv6 address dhcp, 578
- ipv6 address eui-64, 563
- ipv6 address link-local, 568
- ipv6 enable, 568-569, 576-578
- ipv6 route, 586-597, 604
- ipv6 unicast-routing, 558, 578
- line aux 0, 362
- line con 0, 130-131
- line console 0, 97-98, 103, 147, 356
- line vty, 132, 147
- logging console, 145, 148
- logging synchronous, 145, 148
- login, 94, 103, 130-132, 147
- login local, 147
- mac-address, 564
- maximum-paths, 494-496
- name, 185, 207
- ndp -an, 600
- netsh interface ipv6 show neighbors, 600
- network, 473-475, 480-486, 511
- no debug all, 104
- no description, 157, 170
- no duplex, 157, 170
- no ip address, 412
- no ip domain-lookup, 146
- no logging console, 145, 148
- no passive-interface, 487, 496
- no password, 134
- no shutdown, 142, 155-157, 170, 207, 253, 356, 363, 399, 403-405
- [no] shutdown vlan number, 201
- no speed, 157, 170
- no switchport, 408, 411-415
- passive-interface, 487, 496, 517
- passive-interface default, 488
- password, 97, 103, 130-132, 147
- password faith, 94

- ping, 78, 419-429, 587
- port-channel load-balance method, 254
- quit, 104
- reload, 91-92, 102-104, 117, 402-404
- router-id, 470, 496
- router ospf, 470, 495
- router ospf 1, 472, 480
- router ospf process-id, 480, 510
- sdm prefer, 402-404
- sdm prefer lanbase-routing, 402, 415
- show, 95, 166, 361, 480, 508
- show crypto key mypubkey rsa, 149
- show dhcp lease, 143-144, 149
- show etherchannel, 248, 259, 416
- show etherchannel 1 summary, 250
- show etherchannel summary, 413
- show history, 145, 149
- show interfaces, 119-120, 156, 162-164, 167-170, 357-358, 361, 364, 376, 408, 416, 515-517, 583
- show interfaces description, 162, 170
- show interfaces interface-id trunk, 203-205
- show interfaces status, 118, 125, 153, 162-165, 408, 412
- show interfaces switchport, 192-199, 202-203, 208
- show interfaces trunk, 193-194, 199-205, 208, 401
- show interfaces type number switchport, 199
- show interfaces type number trunk, 200
- show interfaces vlan, 143-144, 149, 416
- show ip arp, 391
- show ip default-gateway, 144, 149
- show ip interface brief, 357-361, 364, 406
- show ip ospf, 481, 496, 510-511, 517
- show ip ospf database, 450, 462, 475, 497
- show ip ospf interface, 486-488, 496, 503-505, 510-513, 517
- show ip ospf interface [brief], 479-480, 511
- show ip ospf interface brief, 488, 491, 496, 503, 5.5, 508-510, 514, 517
- show ip ospf interface G0/0, 505
- show ip ospf neighbor, 452-453, 457, 475, 480, 497, 502, 505, 508-517
- show ip ospf neighbor interface brief, 513
- show ip protocols, 479, 485, 496, 517
- show ip route, 324, 356, 367, 376-391, 400-402, 408, 416, 449, 475-478, 497, 585
- show ip route address, 388
- show ip route [connected], 398
- show ip route EXEC, 404
- show ip route ospf, 387, 497
- show ip route static, 380, 490
- show ip ssh, 139, 149
- show ipv6 interface, 558-559, 567, 570-573, 579
- show ipv6 interface brief, 558-560, 567, 575, 579
- show ipv6 route, 566, 579, 585-590, 605
- show ipv6 route connected, 560, 586
- show ipv6 route local, 585-586
- show ipv6 route static, 587-590, 593, 595
- show mac address-table, 120, 125, 356
- show mac address-table aging-time, 122, 125
- show mac address-table count, 122, 125
- show mac address-table dynamic, 96, 117, 123-125, 170
- show mac address-table dynamic address, 125
- show mac address-table dynamic interface, 120-121, 125

- show mac address-table dynamic vlan, 125
- show mac address-table static, 170
- show mac address-table vlan, 121
- show protocols, 361, 364
- show running-config, 93, 101, 104, 132-133, 143, 149, 155, 158, 170, 398, 479, 488, 511, 584
- show running-config | interface, 170
- show spanning-tree, 249, 259
- show spanning-tree vlan, 259
- show spanning-tree vlan vlan-id, 204
- show ssh, 139, 149
- show startup-config, 101, 104, 158
- show vlan, 201, 208
- show vlan brief, 186-189, 202
- show vlan id, 187
- show vlans, 398-401, 416
- show vtp status, 190, 208
- shutdown, 143, 155, 170, 207, 253, 356, 359, 363, 399-401, 405
- shutdown command, 163
- spanning-tree, 259
- spanning-tree mode, 242-243, 259
- spanning-tree vlan, 244
- spanning-tree vlan x root primary, 244-245
- spanning-tree vlan x root secondary, 244-245
- speed, 98-99, 152-154, 165, 170, 355, 363
- switchport, 408, 415
- switchport access vlan, 185-189, 198-199, 207
- switchport mode, 191, 207
- switchport mode access, 185, 188, 198-199
- switchport mode dynamic auto, 202
- switchport mode dynamic desirable, 193
- switchport mode trunk, 191, 203, 396
- switchport nonegotiate, 195, 203, 207
- switchport trunk allowed vlan, 204, 207
- switchport trunk encapsulation, 191, 207
- switchport trunk native vlan, 207
- switchport trunk native vlan vlan-id, 205
- switchport voice vlan, 198-199, 207
- switchport voice vlan vlan-id, 200
- terminal history size, 145, 149
- test etherchannel load-balance EXEC, 255
- traceroute, 428-432, 587
- transport input, 138, 148, 356
- transport input all, 139
- transport input none, 139
- transport input ssh, 139
- transport input telnet ssh, 139
- undebg all, 104
- username, 134
- username secret, 134, 147
- vlan, 185, 198, 207
- vlan number, 201
- vtp mode, 207
- vtp mode off, 190
- vtp mode transparent, 190
- write erase, 104
- communication**
 - bidirectional, 613
 - passing through, 615
 - unidirectional, 613
- configuration BPDUs. See Hello BPDUs**
- configuration changes (STP topology, influencing), 223**
- configuration files, 99-102**
- configuration mode (CLI), 96-97**
- configure terminal command, 97, 101, 104, 132, 189, 355**
- connected routes, 366, 376-378, 583-585**

connectors

- pins, 40
- RJ-45, 41

console connections, cabling, 88-90

console passwords, 129

console ports, 672

context-setting commands, 97

control plane (cloud-based AP architectures), 637

controllers

- centralized, 676-678, 682
- dynamic interfaces, 674-675
- interfaces, 673, 681
- management interfaces, 674
- ports, 672-673
- redundancy management, 674
- service port interfaces, 674
- virtual interfaces, 674
- VLANs, mapping, 673
- WLAN controller configuration, 685
- WLC, 639-642

convergence, 216, 443

converting subnet mask formats, 305-309

copy command, 356

copy running-config startup-config command, 102-104

copy startup-config running-config command, 104

cores (fiber-optic cable), 47

costs (metrics)

- EIGRP, 446
- IGP, 446-447
- OSPF, 491-493
- ports, 247
- IEEE default*, 223
- STP*, 221
- RIPv2, 446-447

CRC (Cyclic Redundancy Checks), 167-168

crossover cable pinouts, 44-45

crosstalk, 40

crypto key command, 137

crypto key generate rsa command, 137-139, 148

CSMA/CD (Carrier Sense Multiple Access with Collision Detection), 55, 167

CUCM (Cisco Unified Communication Manager), 196

cycles, waves, 625

D

DAD (Duplicate Address Detection), 598, 602

data

- decryption, 655
- encapsulation
 - OSI terminology*, 30
 - TCP/IP terminology*, 27-28
- integrity, 656
- privacy, 655
- privacy/integrity methods, 660-661

data centers, 108

data link layer

- Ethernet, 38-39, 49-50
- TCP/IP, 25-26

data-link protocols, leased-line WAN, 63-64

data paths, autonomous wireless networks, 635

data plane (cloud-based AP architectures), 637

Data VLAN (Virtual Local Area Networks), 197-199

DDN (Dotted-Decimal Notation), 24, 305-309

de-encapsulating IP packets, 373-374

Dead Interval timers, 455

dead timers, troubleshooting, 512-513

debug command, 96

decimal masks. *See* DDN

decimal subnet analysis, 331

difficult masks, 334-338

easy masks, 332

finding

- subnet broadcast addresses,*
336-338
- subnet IDs, 334-336*

predictability in interesting octets,
333-334reference table: DDN mask values and
binary equivalent, 338-339**decrypting data, 655****default gateways, 70, 370-372**default-information originate always
command, 490default-information originate
command, 489, 496**default OSPF routes, 489-491****default routers, 70, 370-372****default routes, 379, 383-384****default VLAN (Virtual Local Area
Networks), 186****delete vlan.dat command, 117****description command, 153, 170, 363****designated ports. *See* DP****DHCP (Dynamic Host Configuration
Protocol), 143, 286****diagrams (networking), 15, 26****difficult subnet masks, 334-338****digital certificates, split-MAC archi-
tectures, 640****Dijkstra SPF algorithm, 451****directed broadcast addresses, 283****disable command, 104****disabling**

autonegotiation, 160

DTP, 203

ports, 230

switch interfaces, 155-156

VLAN, troubleshooting, 201-202

WLAN, 680

discarding state (RSTP), 229-230**discovering**

duplicate addresses, 602

neighbor link addresses, 598-600

routers, 600-601

distance vector protocols, 446**distributed architectures, 634-638****distribution switches, 241****distribution system ports, 672-673****distribution systems. *See* DS****DNS (Domain Name Systems), 76-77****documentation, subnet plans, 267****double colon (::), 531****DP (Designated Ports), 217, 222-223,
230****DR (Designated Routers)**

BDR, 456-457

elections, configuration with
broadcast network type (OSPF),
504-506**DRAM (Dynamic Random-Access
Memory), 99****DROthers routers, 457****DS (Distribution Systems), 616-618****DTP (Dynamic Trunking Protocol), 203****dual stacks, 529, 556****duplex command, 152-154, 165, 170,
355, 363****duplexes**configuration on switch interfaces,
152-154

mismatches, 161

troubleshooting, 161-166

Duplicate Address Detection. *See* DAD**dynamic auto trunking, 191****dynamic desirable trunking, 191****dynamic EtherChannels, configuration,
250-251****Dynamic Host Configuration Protocol
(DHCP), 143, 286****dynamic interfaces, 674-675, 678****dynamic IP address configuration,
DHCP, 143**

dynamic ranges per subnet, choosing, 286-287

dynamic unicast address configuration (IPv6), 564

E

E-Line, 66

EAP (Extensible Authentication Protocol), 657-660

EAP-FAST (EAP Flexible Authentication by Secure Tunneling), 659

EAP-TLS (EAP Transport Layer Security), 660

easy subnet masks, 332

echo requests/replies (ICMP), 78, 419

edge ports, 233

EGP (Exterior Gateway Protocol), 444

EIGRP (Enhanced Interior Gateway Routing Protocol), 446

EIGRPv6 (EIGRP for IPv6), 529

electric waves, traveling, 624

embedded WLC deployments, 644

enable command, 91, 104, 130

enable mode, 91-93

enable passwords, 130-131

enable secret command, 131, 148

enable secret love command, 94

encapsulation

IPv4, 70

OSI terminology, 30

TCP/IP terminology, 27-28

encapsulation command, 397-398

encapsulation dot1q command, 415

encapsulation dot1q vlan_id command, 397, 401

encoding schemes, 39

encryption (data), 655

end command, 104, 355

end-user perspectives on networking, 14-15

enterprise LAN (Local Area Networks), 36-37

enterprise mode (WPA), 663

enterprise networks, 15, 268, 350-352

enterprise routers, 350-353

EoMPLS (Ethernet over MPLS), 66

erase nvram command, 104

erase startup-config command, 104, 117

erasing switch configuration files, 102

errors

detection, FCS field, 53

TCP error recovery rates, 21

ESS (Extended Service Sets), 618

EtherChannel, 234, 407

configuration, 247-257

dynamic EtherChannels, 250-251

Layer 3 EtherChannels, 392, 410-414

load distribution, 253-257

manual Layer 2 EtherChannels, 248-250

troubleshooting, 251-253

Ethernet, 26

addresses, 52

cables, 35

E-Line, 66

emulation, 66-68

EoMPLS, 66

GBIC, 42

IPv6 static routes over Ethernet links, 591

LAN. *See also* subnets

enterprise LAN, 36-37

enterprise networks, 350

Ethernet addressing, 50-52

Ethernet data link protocols, 38-50

Ethernet frames, 38

Ethernet physical layer standards, 37

Ethernet ports, 40

- Ethernet Type field*, 52
- FCS field*, 53
- full-duplex logic*, 53-56
- half-duplex logic*, 54-56
- overview*, 32-34
- SOHO LAN*, 35
- switches*, 35, 106-124, 152-162
- troubleshooting*, 162-168
- UTP cables*, 37-46, 49
- VLAN*, 179-205
- links, 40
- OSPF
 - Ethernet links*, 456-457
 - Ethernet WAN*, 506-508
- point-to-point, 56
- shared media, 56
- switches, fiber-optic cables, 48
- WAN
 - enterprise networks*, 350
 - EoMPLS*, 66
 - Ethernet emulation*, 66-68
 - overview*, 65-66
 - point-to-point network type (OSPF)*, 506-508
- Ethernet Alliance web page, 38
- EtherType, 52
- EUI-64 (extended unique identifier), 560-564
- EXEC modes
 - privileged EXEC mode, 91-93
 - simple password configuration, 130-133
 - user EXEC mode, 91-93
- exec-timeout command, 145, 148
- exit command, 98, 101-103, 355
- expanding IPv6 addresses, 532
- experimental addresses, 290
- extended ping command, 423-426
- extended traceroute command, 431-432
- external authentication servers, 135-136
- F**

 - failed interfaces, 217
 - fake AP, 654
 - Fast Ethernet, 37
 - FCS (Frame Check Sequence) field, 53
 - fiber-optic cables, 37-38, 46-49
 - finding
 - IPv6 prefixes, 533-536
 - MAC address table entries, 120-121
 - mismatched Hello/dead timers, 512
 - range of subnet addresses, 331
 - routers best routes, 451
 - subnet broadcast addresses, 327, 336-338
 - subnet ID, 327, 334-336
 - first octet values, classes by, 290
 - first usable IP addresses, deriving, 293-294
 - flash memory, 100
 - Flex+Bridge mode (APs), 647
 - FlexConnect mode (APs), 647
 - floating static routes, 381-383, 593-595
 - flooding, 114, 450
 - Forward delay timers (STP), 225
 - forward secrecy, 663
 - forward-versus-filter decisions, 113
 - forwarding, 115
 - data. *See* routes/routing
 - IP packets, 68-75, 374-375
 - known unicast frames, 110-113
 - forwarding state, interfaces, 215-217
 - frames, 26-28, 38
 - broadcast storms, 213-215
 - CRC, 167
 - flooding, 114
 - giants, 167
 - IP routing, 373-376
 - looping frames, 213-215
 - multiple frame transmissions, 214-215

- packet output errors, 167
- runs, 167
- unknown unicast frames, 114
- frequencies, 613, 625-627
- full addresses (IPv6), 530
- full duplex logic, 53-56
- full VLAN configuration example, 186-188
- fully adjacent neighbors, 457, 502

G

- G0/0 status code, 359
- G0/1 status code, 359
- gateways (default), 370-372
- GBIC (Gigabit Ethernet Interface Converter), 42
- GCMP (Galois/Counter Mode Protocol), 661
- Get IEEE 802 program, 228
- GET requests (HTTP), 20
- GHz (Gigahertz), 625
- giants, 167
- Gigabit Ethernet, 37
- global routing prefix (IPv6), 543-544
- global unicast addresses, 542-550
- global unicast next-hop addresses, 589
- group addresses, 51
- groupings (IP address), 70
- GTC (Generic Token Cards), 660

H

- half-duplex logic, 54-56
- HDLC (High-Level Data Link Control), 63-64
- headers
 - Ethernet header fields, 50
 - HDLC, 63

- HTTP, 20
 - IP headers, 73
- Hello BDPDU, 218, 225
- Hello Interval timers, 455
- Hello messages, 219, 452
- Hello timers, 225, 512-513
- hexadecimal/binary conversion chart (IPv6), 531
- history buffer commands, 144-145
- history size command, 145, 148
- hopping (VLAN), 205
- host addresses, calculating number per network, 293
- host bits, 272
- host forwarding logic (IPv4), 69
- host part (of IP addresses), 292, 302, 311
- host routes, 378-379
 - IPv4 routing process, 370
 - static host routes, 381
- hostname command, 97-103, 117, 138, 148
- hostnames, 76, 427-428
- hosts, 68
 - analyzing subnet needs, 269-271
 - assigning addresses to, 550
 - calculating, 313-315
 - host bits, 272
 - IP settings, 24, 140-142
 - NDP, 598-603
 - subnets, 268-271
- HTTP (Hypertext Transfer Protocol), 19-20
- hubs
 - autonegotiation, 161-162
 - LAN hubs, 54-56
- Hypertext Transfer Protocol (HTTP), 19-20
- Hz (Hertz), 625

-
- IANA (Internet Assigned Numbers Authority), 445, 540
 - IBSS (Independent Basic Service Sets), 619. *See also* BSS
 - ICANN (Internet Corporation for Assigned Names and Numbers), 540
 - ICMP (Internet Control Message Protocol), 78, 419
 - ICMPv6 (Internet Control Message Protocol version 6), 526
 - ID (identification)
 - ID numbers, WLAN, 680
 - interface ID, 547
 - subnet ID, 272, 283, 324, 327, 330, 334-336, 548
 - system ID extensions, 245-246
 - VLAN ID, 180
 - IEEE (Institute of Electrical and Electronic Engineers), 18
 - 802.1D Spanning-Tree states, 227
 - 802.1D standard, 228
 - 802.1w amendment, 228
 - 802.1x, EAP integration, 658
 - default port costs, 223
 - Get IEEE 802 program, 228
 - IGP (Interior Gateway Protocol), 444-448
 - IGRP (Interior Gateway Routing Protocol), 446
 - inferior Hello messages, 219
 - infrastructure mode, 614
 - input errors, 166-167
 - integrated services routers (Cisco), 352
 - interarea routes, 461
 - interesting octets, predictability in, 333-334
 - interface command, 97, 103, 169, 185, 198, 356, 363, 391, 415
 - interface ethernet command, 357
 - interface fastethernet command, 357
 - interface gigabitethernet command, 357
 - interface ID, 547
 - interface loopback command, 470, 481, 496
 - interface port-channel command, 416
 - interface port-channel number command, 411
 - interface range command, 154, 169, 187
 - interface type number.subint command, 397
 - interface vlan command, 148, 415
 - interface vlan 1 command, 142
 - interface vlan vlan_id command, 403
 - interfaces, 87
 - administratively shutdown, 217
 - blocking state, 215
 - controllers, 673, 681
 - dynamic interfaces, 674-675, 678
 - EtherChannels, adding, 251-253
 - failed interfaces, 217
 - forwarding state, 215
 - Layer 1 problems, 166-168
 - learning state, 227
 - listening state, 227
 - management interfaces, 674
 - OSPF
 - metrics*, 493
 - passive interfaces*, 487-488
 - OSPFv2 configuration, 483-486
 - physical interface configuration, 251-253
 - ports, compared, 671
 - routed interfaces, Layer 3 (multilayer) switches, 407-409
 - routers, 356-357
 - bandwidth*, 361
 - clock rates*, 361
 - IP addresses*, 360-361
 - status codes*, 358-359
 - service port interfaces, 674

- speed and duplex issues, 163-166
- states, 216-217, 227
- status codes, 162-163, 358-359
- subcommands, 97
- subinterfaces, 396-397
- SVI, 392, 401-406
- switch interface configuration, 152-162
- troubleshooting, 162-168
- virtual interfaces, 674
- VLAN interfaces, 402
- WLC interfaces, 673-675
- working interfaces, 217
- interference, simultaneous transmissions, 613
- internal routers, 461
- Internet Protocol. *See* IP
- internetworks, 72, 268
- intra-area routes, 461
- intrusion protection, WLC, 642
- IOS configuration, 96-102
- IP (Internet Protocol), 22. *See also* IPv4; IPv6
 - addresses
 - management*, 635
 - ping command*, 427-428
 - subnets*, 283-284
 - forwarding
 - IP packets*, 374-375
 - longest prefix matches*, 386-389
 - IGP metrics, 446-447
 - routing, 366
 - ARP tables*, 378-379
 - de-encapsulating IP packets*, 373-374
 - encapsulating IP packets in new frames*, 375
 - example of*, 371-376
 - frames*, 373-376
 - host forwarding of IP packets to default routers (gateways)*, 372
 - IP forwarding*, 374-375, 386-389
 - IPv4 routing process*, 369-371
 - troubleshooting*, 419-434
 - routing tables, 70-72, 388-389
 - telephony, 196-200
- ip -6 neighbor show command, 600
- ip address address mask command, 397, 403, 411
- ip address command, 142, 148, 360, 363, 381, 391-392, 398
 - IP addresses on loopback interfaces, 470
 - subinterfaces, 397
- ip address dhcp command, 148
- ip address subcommand, 376
- ip_address parameter, network command, 473
- ip default-gateway command, 142, 148
- ip domain-name command, 139
- ip mtu command, 515
- ip name-server command, 142, 148
- ip ospf command, 495
- ip ospf cost command, 492, 496
- ip ospf dead-interval command, 517
- ip ospf hello-interval command, 517
- ip ospf process-id area area-id command, 483-485
- ip ospf process-id command, 511
- ip route command, 367, 376, 379-385, 391, 402-404, 415
- ip ssh version 2 command, 139
- IPv4 (Internet Protocol Version 4). *See also* IP
 - address exhaustion, 525
 - ARP, 72, 77
 - calculating hosts and subnets in network, 313-315
 - classes in, 290-291
 - classful IP networks, 289-297
 - classless versus classful addressing, 312-313
 - configuration on switch, 142-143

DNS, 76-77
 dynamic IP address configuration with DHCP, 143
 headers, 73
 hosts, 24, 140-142
 networks, 70-73, 293-295
 overview, 22-23, 68
 private addresses, 542
 public addresses, 542
 router support
 auxiliary ports, 362
 CLI access, 355-356
 interfaces, 356-361
 routing, 24-25, 369-371
 logic, 68-72
 protocols, 74-75
 subnets, 70, 73, 264-267, 322-339
 hosts, 268-271
 multiple subnet sizes, 274
 number of hosts, 271
 number of subnets, 270
 one-size subnets, 273
 single-size subnets, 273
 size of, 272-274
 subnet addresses, 272
 subnet ID, 272
 subnet masks, 272, 275, 279-283, 302-312, 315
 subnet numbers, 272
 switch settings, 140-142
 testing connectivity, 78
 troubleshooting tools
 ping command, 419-429
 SSH, 432-434
 Telnet, 432-434
 traceroute command, 428-432
 unusual addresses within classes, 295
 verifying on switch, 143-144
 VLSM, 275

IPv6 (Internet Protocol Version 6). *See also IP*
 abbreviating addresses, 531-532
 address configuration summary, 576
 assigning subnets to internetwork topology, 549
 dual-stack strategies, 556
 dynamic unicast address configuration, 564
 expanding addresses, 532
 global routing prefix, 543-544
 global unicast addresses, 542-550
 hexadecimal/binary conversion chart, 531
 history of, 524-525
 interface ID, 547
 link-local addresses, 566-569
 loopback addresses, 574
 multicast addresses, 569-576
 NDP, 573-574, 598-603
 overview, 524
 prefix length, 533-536
 protocols, 526-527
 representing full IPv6 addresses, 530
 routing, 527-530, 583-598
 static unicast address configuration, 557-564
 subnets, 543
 global unicast addresses, 545-549
 router anycast addresses, 549
 unique local addresses, 551-552
 unicast addresses, 556
 unique local addresses, 542, 551-553
 unknown addresses, 574
ipv6 address command, 557, 560, 564-568, 576-578, 583
ipv6 address dhcp command, 578
ipv6 address eui-64 command, 563
ipv6 address link-local command, 568
ipv6 enable command, 568-569, 576-578

ipv6 route command, 586-597, 604
 ipv6 unicast-routing command, 558, 578
 IS-IS (Integrated Intermediate System
 to Intermediate System), 446
 ISL (Inter-Switch Link), 182
 ISO (International Organization for
 Standardization), 17
 IV (Initialization Vectors), 661

J - K

keys

forward secrecy, 663
 mixing algorithm, 661
 PKIs, 660
 shared-key security, 657
 TKIP, 660-661
 WEP, 657
 kHz (kilohertz), 625
 kilohertz (kHz), 625
 known unicast frames, forwarding,
 110-113

L

LACP (Link Aggregation Control
 Protocol), 250
 LAG (link aggregation group), 673
 LAN (Local-Area Networks). *See also*
 subnets
 addresses, 52
 definition of, 179
 DP on each segment, choosing, 222-223
 enterprise LAN, 36-37
 Ethernet LAN, 32-46, 49-56
 enterprise networks, 350
 LAN switching, 106-124
 switch interface configuration,
 152-162
 troubleshooting, 162-168
 hubs, 54-56, 161-162

LAN switching, 106-124
 neighbors, testing, 425-426
 redundancy, 210, 214
 STP security exposures, 236
 switching, 35
 analyzing, 116
 flooding, 114
 interface configuration, 152-162
 MAC address table, 113-114,
 117-124
 overview, 106-109
 STP, 114-115
 summary, 115-116
 switch forwarding and filtering
 decisions, 110-113
 switch interfaces, 118-120,
 152-162
 switching logic, 109-110
 verifying, 116

VLAN

AP, 668
 configuration, 185-195, 198-199
 Data VLAN, 197-199
 default VLAN, 186
 disabled VLAN, 201-202
 IP telephony, 196-200
 native VLAN, 183, 205
 overview, 179-180
 routing, 183-184
 supported VLAN list on trunks,
 203-205
 tagging, 181-182
 troubleshooting, 201-205
 trunking, 180-182, 189-195
 undefined VLAN, 201-202
 VLAN ID, 180
 Voice VLAN, 197-199
 VTP, 189-190
 WLAN, 32
 802.11 WLAN, 614
 advanced settings, 684-685

- AP, 668-669
- BSS, 614-616
 - client session timeouts*, 684
 - configuration*, 675-678, 681-685
 - controller configuration*, 685
 - creating*, 679-681
 - creating too many*, 676
 - defined*, 675
 - displaying list of*, 679
 - DS, 616-618
 - ESS, 618
 - IBSS, 619
 - limiting*, 676
 - management access*, 685
 - mesh networks*, 622
 - outdoor bridges*, 621-622
 - QoS, 683-684
 - repeaters*, 620-621
 - security*, 681-684
 - topologies*, 614-622
 - WGBs, 621
 - WLCs, 669-675
- LAP (Lightweight Access Points), 639-642
- last usable IP addresses, deriving, 293-294
- late collisions, 167
- Layer 1 problems, troubleshooting, 166-168
- Layer 2 switches, 141, 183
- Layer 3 EtherChannel, 392
- Layer 3 (multilayer) switches, 141, 184
 - routed ports, 406-414
 - SVI, 401-406
- LEAP (Lightweight EAP), 659
- learning state, interfaces, 227
- leased-line WAN (Wide Area Networks), 61-65
- lightweight AP (Access Points), 638
- line aux 0 command, 362
- line con 0 command, 130-131
- line console 0 command, 97-98, 103, 147, 356
- line vty command, 132, 147
- link-local addresses (IPv6), 566-569
- link-local next-hop address, 589-590
- link-state protocols, 446
- list of subnets
 - building, 283-284
 - IPv6 subnets, 548-549
- listening state, interfaces, 227
- load balancing
 - clients, 642
 - OSPF, 494
- load distribution, EtherChannel, 253-257
- Local mode (AP), 647
- local routes, 378, 583-586
- local scope multicast addresses, 569-573
- logging console command, 145, 148
- logging synchronous command, 145, 148
- logical networks, user segregation, 676
- login command, 94, 103, 130-132, 147
- login local command, 147
- loopback address, 295, 574
- looping frames, 213-215
- loops, avoiding with STP, 114-115
- LSA (Link-State Advertisements), 449, 454
 - flooding, 450
 - LSDB relationship, 450
 - network LSA, 464
 - OSPF, 454-456, 459-464
 - router LSAs, 463
- LSDB (Link-State Database)
 - area design, 461-462
 - best routes, finding, 451
 - LSA relationship, 450
 - OSPF/LSDB neighbor exchanges, 454-456

LSU (Link-State Update) packets, 454
 LWAPP (Lightweight Access Point Protocol), 639

M

MAC address tables, 111

- aging, 121-122
- clearing, 122
- finding entries in, 120-121
- instability, 214-215
- multiple switches, 123-124
- overview, 113-114
- showing, 117-118

mac-address command, 564

MAC addresses, 50-52

- burned-in, 218
- sender MAC addresses, 661
- source MAC addresses, 113
- split-MAC architectures, 638-642

macrobending, 163

magic number, 334

magnetic waves, traveling, 624

man-in-the-middle attacks, 654

management access (WLAN), allowing, 685

management interfaces (controllers), 674

management IP addresses, autonomous AP, 635

manual Layer 2 EtherChannels, 248-250

mapping VLAN, 673

MaxAge timer (STP), 225

maximum-paths command, 494-496

memory, 99-100

Meraki, 636-637

mesh networks, 622

messages

- Hello, 219
- Hello BPDU, 218, 225

- inferior Hello, 219

- integrity, 656, 660-661

- OSPF Hello, 452

- privacy, 655, 660-661

- RSTP, 232

- sending, 623-624

- superior Hello, 219

metrics (costs)

- EIGRP, 446

- IGP, 446-447

- OSPF, 491-493

- ports, 247

- IEEE default*, 223

- STP*, 221

- RIPv2, 446-447

MHz (Megahertz), 625

MIC (Message Integrity Checks), 656, 660-661

Mobility Express WLC deployments, 645

models, networking

- OSI, 17, 28-30

- TCP/IP, 16-29

modified EUI-64 (Extended Unique Identifier-64), 560-564

Monitor mode (AP), 647

MP BGP-4 (Multiprotocol BGP version 4), 529

MSCHAPv2 (Microsoft Challenge Authentication Protocol version 2), 660

MSTP (Multiple Spanning Tree Protocol), 242-243

MTU (Maximum Transmission Units), 50, 515

multiarea OSPF (Open Shortest Path First), 482

multicast addresses, 50-52, 290, 569-576

multilayer switches, 141, 184, 401-414

multimode fiber-optic cables, 47-49

N

NA (Neighbor Advertisement), 599

name command, 185, 207

NAT (Network Address Translation), 277, 542

native VLAN (Virtual Local-Area Networks), 183, 205, 398

NDP (Neighbor Discovery Protocol), 526, 573-574, 598-603

ndp -an command, 600

neighbors

adjacent neighbors, 457

fully adjacent neighbors, 457, 502

link addresses, discovering, 598-600

NA, 599

NS, 599

OSPF, 451

broadcast network type, 502-506

LSA exchanges, 454-456

LSDB exchanges, 454-456

requirements, 508-510

RID, 452

states, 453, 457

troubleshooting adjacencies, 510-516

testing, 425-426

netsh interface ipv6 show neighbors command, 600

network command, 473-475, 480-486, 495, 511

network ID. *See* network numbers

network layer, 22-25

ARP, 77

DNS, 76-77

protocols, identifying with Ethernet Type field, 52

routing

LAN/WAN, 70-72

logic, 68-70

testing connectivity, 78

network numbers, 293-295

network types (OSPF)

broadcast, 500-506

point-to-point, 500-501, 506-508

troubleshooting mismatched network types, 515-516

networks

architectures, 16

blueprint, 16

broadcast addresses, 293-295

classful IP networks, 289-297

classful networks, 276-278

definition of, 268

diagrams, 15, 26

end-user perspectives, 14-15

enterprise networks, 15, 268, 350-352

internetworks, 268

IP networks, 70-73, 292, 302, 312

logical networks, user segregation, 676

LSA, 464

masks, 376

mesh, 622

NAT, 277

networking model overview, 16

OSI, 17, 28-30

overview, 12-14

private IP networks, 277-278

public IP networks, 276-278

routes, 379

SOHO networks, 15

subnets versus, 324

TCP/IP, 16-29

VLAN switches, 140

WAN, 60

Ethernet WAN, 65-68

leased-line WAN, 61-65

wireless networks, 628-629, 662-663

next-hop IPv6 addresses, 589-590

NIC addresses, 52

NIM (Network Interface Modules), 352

- no debug all command, 104
- no description command, 157, 170
- no duplex command, 157, 170
- no ip address command, Layer 3 Ether-Channels, 412
- no ip domain-lookup command, 146
- no logging console command, 145, 148
- no network network-id area area-id subcommands, 483
- no passive-interface command, 487, 496
- no password command, 134
- no shutdown command, 142, 155-157, 170, 207, 253, 356, 363, 399, 403-405
- [no] shutdown vlan number command, 201
- no speed command, 157, 170
- no switchport command, 408, 411-415
- nonoverlapping channels, 628
- nonworking states, troubleshooting, 162-163
- NS (Neighbor Solicitation), 599
- numbers
 - DDN, 24
 - magic number, 334
 - SEQ, 21
 - subnet numbers, 272, 283, 324, 327, 334-336
- NVRAM (nonvolatile RAM), 100

O

- one-size subnets, 273-274
- open authentication, 656
- operational view of subnetting, 267-268
- optical transmitters (fiber-optic cable), 47
- OSI (Open Systems Interconnection), 17, 28-30

- OSPF (Open Shortest Path First), 450
 - 2-way state, 453-454, 457
 - area design, 459-462
 - backbone areas, 460
 - broadcast network type, 500-506
 - calculating best routes with SPF, 457-459
 - configuration, 472, 479-481
 - default routes, 489-491
 - Dijkstra SPF algorithm, 451
 - DR, 456-457
 - Ethernet links, 456-457
 - Hello/dead timers, 512-513
 - Hello messages, 452
 - interfaces, 493
 - load balancing, 494
 - LSAs, 450, 459-464
 - metrics, 446-447, 491-493
 - mismatched network types, 515-516
 - MTU mismatched settings, 515
 - multiarea OSPF, 482
 - neighbors, 451
 - broadcast network type, 502-506*
 - LSA exchanges, 454-456*
 - LSDb exchanges, 454-456*
 - requirements, 508-510*
 - RIDs, 452*
 - states, 453, 457*
 - troubleshooting adjacencies, 510-516*
 - passive interfaces, 487-488
 - point-to-point network type, 500-501, 506-508
 - process-id, 472
 - processes, shutting down, 513-514
 - RID, 480-481, 511
 - verifying
 - configuration, 479-480*
 - operation, 475-478*

OSPFv2 (OSPF version 2), 440, 463
 interface configuration, 483-486
 load balancing, 494
 metrics, 493
 single-area configuration, 470-475
 OSPFv3 (OSPF version 3), 526, 529
 outdoor bridges, 621-622
 outgoing interfaces, IPv6 static routes
 with, 587-588

P

PAC (Protected Access Credentials), 659
 packets, 28
 data packets, routing VLAN, 184
 IP packets
 de-encapsulating, 373-374
 encapsulating in new frames, 375
 forwarding, 68-75, 374-375
 hot forwarding to default routers
 (*gateways*), 372
 output errors, 167
 PAgP (Port Aggregation Protocol), 250
 passing through (communications), 615
 passive-interface command, 487, 496, 517
 passive-interface default command, 488
 password command, 97, 103, 130-132, 147
 password faith command, 94
 passwords
 CLI, 93-94, 130-135
 console passwords, 129
 enable passwords, 130
 shared passwords, 130
 Telnet passwords, 129
 path selection, 69, 442
 PBX (Private Branch Exchange), 196
 PDU (Protocol Data Units), 30
 PEAP (Protected EAP), 659
 permanent keywords, 385
 personal mode (WPA), 663
 physical console connections, 88-90
 physical interfaces, configuration, 251-253
 physical layer (TCP/IP), 25-26
 ping command, 78, 419-429, 587
 pinouts (cables)
 10BASE-T, 42-45
 100BASE-T, 42-45
 1000BASE-T, 45-46
 rollover pinouts, 89
 pins (connectors), 40
 PKIs (Public Key Infrastructures), 660
 point-to-multipoint outdoor bridges, 622
 point-to-point (Ethernet), 56
 point-to-point edge ports, 233
 point-to-point lines. *See* leased-line WAN
 point-to-point network type (OSPF), 500-501, 506-508
 point-to-point outdoor bridges, 622
 point-to-point ports, 233
 policies, WLAN client exclusion, 684
 Port Aggregation Protocol. *See* PAgP
 port-channel load-balance method command, 254
 PortChannels. *See* EtherChannel
 PortFast, 235
 ports, 87
 802.1w RSTP roles, 230
 alternate, 229-232
 backup, 230
 blocking, choosing, 212
 console ports, 672
 controllers, 672-673
 costs, 247
 IEEE default, 223
 STP, 221

- disabled ports, 230
- distribution system ports, 672-673
- DP, 217, 222-223, 230
- Ethernet ports, 40
- interfaces, compared, 671
- redundancy ports, 672
- RJ-45, 40
- routed ports, VLAN routing, 406-414
- router auxiliary ports, 362
- RP, 217, 220, 230
- RSTP
 - backup*, 233
 - roles*, 230
- service ports, 672-674
- states, 232
- switch ports, 110
- switch roots, choosing, 220-221
- USB ports, 89
- WLC ports, 672-673
- postal service forwarding, 22
- predictability in interesting octet, 333-334
- prefixes
 - IP addresses, 292, 302
 - defined*, 309-310
 - dividing into network and subnet parts*, 312
 - host part and*, 311
 - length of*, 533-536
 - masks, 305-309
 - routing, 378
- primary root switches, 247
- priority, switches, 245-246
- privacy
 - CCMP, 661
 - data, 655
 - GCMP, 661
 - TKIP, 660-661
- private addresses (IPv4), 542
- private branch exchange. *See* PBX
- private IP networks, 277-278
- private lines. *See* leased-line WAN
- privileged EXEC mode, 91-93
- problem isolation, traceroute
 - command, 429-431
- process-ids (OSPF), 472
- proprietary routing protocols, 446
- protected access credentials. *See* PAC
- protocols
 - BGP, 445
 - BPDU, 218, 225
 - CAPWAP, 639
 - CCMP, 661
 - definition of, 16
 - distance vector, 446
 - DTP, 203
 - EAP, 657-658
 - EAP-FAST, 659
 - EAP-TLS, 660
 - GCMP, 661
 - IGRP, 446
 - LACP, 250
 - LEAP, 659
 - link-state, 446
 - LWAPP, 639
 - MSTP, 242-243
 - NDP, 573-574
 - OSPF, 450
 - 2-way state*, 453-454, 457
 - area design*, 459-462
 - backbone areas*, 460
 - broadcast network type*, 500-506
 - calculating best routes with SPF*, 457-459
 - configuration*, 472, 479-481
 - default routes*, 489-491
 - Dijkstra SPF algorithm*, 451
 - DR*, 456-457
 - Ethernet links*, 456-457
 - Hello/dead timers*, 512-513
 - Hello messages*, 452
 - interfaces*, 493

- load balancing*, 494
- LSAs*, 450, 459-464
- metrics*, 446-447, 491-493
- mismatched network types*, 515-516
- MTU mismatched settings*, 515
- multiarea OSPF*, 482
- neighbors*, 451-457, 502-516
- passive interfaces*, 487-488
- point-to-point network type*, 500-501, 506-508
- process-id*, 472
- processes, shutting down*, 513-514
- RID*, 480-481, 511
- verifying operation*, 475-478
- OSPFv2, 440, 463
 - interface configuration*, 483-486
 - load balancing*, 494
 - metrics*, 493
 - single-area configuration*, 470-475
- OSPFv3, 526, 529
- PAGP, 250
- PEAP, 659
- PVST+, 242-243
- RIP, 446
- routable protocols, 442
- routed protocols, 442
- routing protocols, 376-378, 442-449
- RPVST+, 242-243, 246
- RSTP, 228, 242-243
 - alternate ports*, 230-232
 - backup port role*, 233
 - BID*, 218
 - BPDU*, 218, 225
 - configurable priority values*, 244
 - configuration*, 240
 - discarding state*, 229
 - forwarding or blocking criteria*, 216-217
 - LAN segment DP*, 222-223
 - link types*, 233
 - looping frames, preventing*, 213
 - multiple spanning tree support*, 246
 - need for*, 213-215
 - ports*, 212, 230-233
 - processes*, 232
 - purpose of*, 215-217
 - root switches*, 218, 247
 - STA*, 216
 - standards*, 228
 - steady-state operation*, 225
 - STP, compared*, 229-230
 - switches*, 219-221, 247
 - topology influences*, 223-225
- STA, 216
- STP, 114-115
 - 802.1D standard*, 228
 - BID*, 218-219, 243-244
 - BPDU*, 218, 225
 - configurable priority values*, 244
 - configuration*, 240, 243-244
 - convergence*, 216
 - EtherChannels*, 234, 247-251
 - Forward delay timer*, 225
 - forwarding or blocking criteria*, 216-217
 - Hello timer*, 225
 - interface states, changing*, 227
 - LAN redundancy*, 210, 214
 - LAN segment DP*, 222-223
 - looping frames*, 213
 - MaxAge timer*, 225
 - modes*, 242
 - multiple STP*, 241
 - need for*, 213-215
 - PortFast*, 235
 - ports*, 212, 221, 232
 - purpose of*, 215-217
 - roles*, 227

- root switches*, 218-219
 - RSTP*, 229-230
 - security*, 236
 - STA*, 216
 - standards*, 242
 - states*, 227
 - steady-state operation*, 225
 - switch reactions to changes*, 226-227
 - switch RP*, 220-221
 - system ID extensions*, 243-244
 - timers*, 226-227
 - topology influences*, 223-225
 - TCP, 20-21
 - TCP/IP
 - application layer*, 19-20
 - compared to OSI*, 29
 - data encapsulation terminology*, 27-28
 - data-link layer*, 25-26
 - history of*, 16-17
 - HTTP*, 19-20
 - IPv4*, 22-25, 68-78, 140-144
 - network layer*, 22-25, 68-72, 76-78
 - overview*, 18
 - physical layer*, 25-26
 - RFC*, 18
 - transport layer*, 20-22
 - TKIP, 660-661
 - public addresses (IPv4), 542
 - public IP networks, 276-278
 - Public Key Infrastructures. *See* PKIs
 - PVST+ (Per VLAN Spanning Tree), 242-243
- ## Q - R
-
- QoS (Quality of Service), WLAN, 683-684
 - quit command, 104
 - RA (Router Advertisement), 600
 - radio frequencies. *See* RF
 - radios, selecting WLAN, 680
 - RADIUS servers
 - configuration, 676
 - WLAN authentication, 682
 - RAM (Random Access Memory), 99
 - ranges for global unicast addresses, 544-545
 - RC4 cipher algorithm, 657
 - receivers, communication, 613
 - redundancy
 - LAN, 210, 214
 - management, 674
 - ports, 672
 - reference bandwidth, defined, 492
 - registered private IP networks, 277-278
 - registered public IP networks, 276-278
 - reload command, 91-92, 102-104, 117, 402-404
 - remote subnets, 375
 - repeaters, 620-621
 - replies
 - ARP replies, 77
 - HTTP, 20
 - ICMP echo replies, 78
 - requests
 - ARP requests, 77
 - ICMP echo requests, 78
 - reserved multicast addresses, 569-571
 - resident subnets, 322
 - reverse routes, testing, 423-425
 - RF (Radio Frequencies), 613, 626, 642
 - RID (Router ID)
 - defined, 470
 - OSPF, 511
 - neighbors*, 452
 - RID configuration*, 480-481
 - troubleshooting, 511

- RIP (Routing Information Protocol), 446
- RIPng (RIP next generation), 529
- RIPv2 (Routing Information Protocol version 2), 446-447
- RIR (Regional Internet Registries), 524
- RJ-45 connectors, 41
- RJ-45 ports, 40
- roaming
 - AP, 618
 - clients, 642
- ROAS (Router-On-A-Stick), 392, 396-401
- Rogue Detector mode (AP), 647
- roles
 - alternate ports, 230-232
 - ports, 230, 233
 - RSTP port, 230
 - STP, 227
- rollover pinouts (cables), 89
- ROM (Read-Only Memory), 100
- root bridge ID, 218
- root costs, switches, 216
- root ports. *See* RP
- root switches, 217
 - electing, 218-219
 - RSTP root switches, 247
 - timer values, 218
- routeable protocols, 442
- route redistribution, 448
- routed ports, VLAN routing, 406
 - EtherChannels, 410-414
 - routed interfaces, 407-409
- routed protocols, 442
- router-id command, 470, 496
- router ospf command, 470, 495
- router ospf 1 command, 472, 480
- router ospf process-id command, 480, 510
- routers/routing, 35
 - ABR, 460-461
 - ARP tables, 378-379
 - auxiliary ports, 362
 - backbone, 461
 - best routes, finding, 451
 - candidate default routes, 384
 - Cisco integrated services routers, 352
 - classful versus classless, 313
 - CLI, 355-356
 - connected routes, 366, 376-378
 - default routers, 70, 370-372
 - default routes, 379, 383-384
 - discovering with NDP, 600-601
 - DR, 456-457
 - DROthers, 457
 - dynamic unicast address configuration, 564
 - enterprise routers, 350-353
 - floating static routes, 381-383
 - flooding, 450
 - host routes, 378-379
 - logic*, 370
 - static host routes*, 381
 - installation, 350-354
 - interfaces, 356-361
 - internal routers, 461
 - IP routing, 366, 369
 - ARP tables*, 378-379
 - de-encapsulating IP packets*, 373-374
 - encapsulating IP packets in new frames*, 375
 - example of*, 371-376
 - forwarding*, 374-375, 386-389
 - host forwarding of IP packets to default routers (gateways)*, 372
 - IPv4 routing*, 24-25, 68-75, 355-362, 369-371, 527
 - IPv6 routing*, 527-530, 558, 583-598
 - processing incoming frames*, 373
 - tables*, 388-389

- transmitting frames*, 376
- troubleshooting*, 419-434
- link-local address configuration, 566-569
- local routes, 378
- logic
 - host routing*, 370
 - IPv4 routing*, 371
- LSA, 463
- network masks, 378
- network routes, 379
- OSPF interface costs, 493
- overview, 348
- path selection, 69
- prefixes, 378
- protocol codes, 378
- protocols, 376
 - administrative distance*, 448-449
 - algorithms*, 445
 - AS*, 444
 - classful versus classless*, 313
 - classless/classful*, 447-448
 - convergence*, 443
 - defined*, 442
 - distance vector*, 446
 - EGP*, 444
 - EIGRP*, 446
 - functions*, 443
 - IGP*, 444-448
 - link-state*, 446
 - OSPF*, 446-447, 450-464, 475-482, 487-491
 - path selections*, 442
 - proprietary*, 446
 - IPv2*, 446-447
 - route redistribution*, 448
- remote subnets, 375
- reverse routes, testing, 423-425
- ROAS
 - configuration*, 396-398
 - subinterfaces*, 399-401
 - troubleshooting*, 400-401
 - verifying*, 398-400
- SOHO routers, 354
- static unicast address configuration, 557-564
- static routes, 367, 376
 - configuration*, 379-384
 - default routes*, 379
 - floating static routes*, 381-383
 - host routes*, 379-381
 - static default routes*, 383-384
 - static network routes*, 379
 - troubleshooting*, 385-386
- subnet router anycast addresses, 576
- VLAN routing, 183-184, 395
 - Layer 3 (multilayer) switch routed ports*, 406-414
 - Layer 3 (multilayer) switch SVI*, 401-406
 - ROAS*, 396-401
- WAN, 64-65
- RP (Root Ports), 217, 220-221, 230
- RPVST+ (Rapid Per VLAN Spanning Tree+), 242-243, 246
- RS (Router Solicitation), 600
- RSTP (Rapid Spanning Tree Protocol), 228, 242-243
 - alternate ports, 230-232
 - backup port role, 233
 - BID, 218
 - blocking criteria, 216-217
 - BPDU, 218, 225
 - configurable priority values, 244
 - configuration, 240
 - discarding state, 229
 - forwarding criteria, 216-217
 - LAN segment DP, 222-223
 - link types, 233
 - looping frames, preventing, 213
 - multiple spanning tree support, 246
 - need for, 213-215

- ports, 233
 - blocking*, 212
 - roles*, 230
 - states*, 232
- processes, 232
- purpose of, 215-217
- root switches, 218, 247
- STA, 216
- standards, 228
- steady-state operation, 225
- STP, compared, 229-230
- switches
 - electing*, 219
 - priority*, 247
 - RP, choosing*, 220-221
- topology influences, 223-225
- running-config file, 100
- runts, 167

S

- S0/0/0 status code, 359
- same-layer interaction, 21-22
- scopes of multicast addresses, 571-572
- sdm prefer command, 402-404
- sdm prefer lanbase-routing command, 402, 415
- SE Connect mode (APs), 647
- secondary root switches, 247
- Secure Shell. *See* SSH
- security. *See also* authentication
 - attacks, 654
 - CLI, 93-94, 128-139
 - data integrity, 656
 - data privacy, 655
 - decryption, 655
 - encryption, 655
 - fake AP, 654
 - forward secrecy, 663
 - intrusion protection, 642
 - MIC, 656
 - privacy/integrity methods, 660-661
 - shared-key, 657
 - STP, 236
 - transmissions reaching unintended recipients, 652
 - WLAN, 681-684
 - WLC authentication, 642
 - WPA, 662-663
 - WPA2, 662-663
 - WPA3, 662-663
- self-healing coverage, 642
- sender MAC addresses, 661
- SEQ (Sequence Numbers), 21
- sequence counters (TKIP), 661
- sequence numbers (SEQ), 21
- serial lines. *See* leased-line WAN
- Serial WAN (Wide Area Networks), 350
- servers
 - AAA servers, 136
 - AS, 658
 - external authentication servers, 135-136
 - RADIUS, 676, 682
 - Telnet servers, 91
- service ports, 672-674
- service set identifiers. *See* SSID
- session timeouts (WLAN), 684
- SFP (Small Form Pluggable), 42, 48
- SFP+ (Small Form Pluggable Plus), 42, 48
- shared-key security, 657
- shared media (Ethernet), 56
- shared passwords, 130
- shared ports, 234
- shorter VLAN configuration example, 189
- Shortest Path First algorithm. *See* SPF algorithm

- show arp command, 391
- show command, 95, 166, 361, 480, 508
- show crypto key mypubkey rsa command, 149
- show dhcp lease command, 143-144, 149
- show etherchannel 1 summary command, 250
- show etherchannel command, 248, 259, 416
- show etherchannel summary command, 413
- show history command, 145, 149
- show interfaces command, 119-120, 156, 162-164, 167-170, 357-358, 361, 364, 376, 408, 416, 515-517, 583
- show interfaces description command, 162, 170
- show interfaces interface-id trunk command, 203-205
- show interfaces status command, 118, 125, 153, 156, 162-165
 - Layer 3 EtherChannels, 412
 - routed ports, 408
- show interfaces switchport command, 192-195, 199, 202-203, 208
- show interfaces trunk command, 193-194, 199-200, 203-205, 208, 401
- show interfaces type number switchport command, 199
- show interfaces type number trunk command, 200
- show interfaces vlan command, 143-144, 149, 416
- show ip arp command, 391
- show ip default-gateway command, 144, 149
- show ip interface brief command, 357-361, 364, 406
- show ip ospf command, 481
 - defined, 496, 517
 - duplicate OSPF RID, 511
 - OSPF neighbors, troubleshooting, 510
- show ip ospf database command, 450, 462, 475, 497
- show ip ospf interface brief command, 479-480, 488, 491, 503-505, 508, 511, 514
 - defined, 496, 517
 - OSPF neighbors, troubleshooting, 510
- show ip ospf interface command, 488, 503-505, 513
 - defined, 496, 517
 - Hello/dead timer mismatches, 512
 - OSPF neighbors, troubleshooting, 510
 - OSPFv2 interface configuration, 486
- show ip ospf interface G0/0 command, 505
- show ip ospf neighbor command, 452-453, 457, 475, 480, 497, 502, 505, 508-511, 513-517
- show ip ospf neighbor interface brief command, 513
- show ip protocols command
 - defined, 496, 517
 - OSPFv2 interface configuration, 485
- show ip route address command, 388
- show ip route command, 324, 356, 367, 376, 378-391, 400-402, 408, 475-478, 585
 - administrative distance, 449
 - defined, 497
 - routing tables, displaying, 416
- show ip route [connected] command, 398
- show ip route EXEC command, 404
- show ip route ospf command, 387, 497
- show ip route static command, 380, 490
- show ip ssh command, 139, 149

- show ipv6 interface brief command, 558-560, 567, 575, 579
- show ipv6 interface command, 558-559, 567, 570-573, 579
- show ipv6 route command, 566, 579, 585-590, 605
- show ipv6 route connected command, 560, 586
- show ipv6 route local command, 585-586
- show ipv6 route static command, 587-590, 593-595
- show mac address-table aging-time command, 122, 125
- show mac address-table command, 120, 125, 356
- show mac address-table count command, 122, 125
- show mac address-table dynamic address command, 125
- show mac address-table dynamic command, 96, 117, 123-125, 170
- show mac address-table dynamic interface command, 120-121, 125
- show mac address-table dynamic vlan command, 125
- show mac address-table static command, 170
- show mac address-table vlan command, 121
- show protocols command, 361, 364
- show running-config | interface command, 170
- show running-config command, 93, 101, 104, 132-133, 143, 149, 155, 158, 170, 398, 479, 488, 511, 584
- show spanning-tree command, 249, 259
- show spanning-tree vlan command, 259
- show spanning-tree vlan vlan-id command, 204
- show ssh command, 139, 149
- show startup-config command, 101, 104, 158
- show vlan brief command, 186-189, 202
- show vlan command, 201, 208, 398-401, 416
- show vlan id command, 187
- show vtp status command, 190, 208
- shutdown command, 143, 155, 163, 170, 207, 253, 356, 359, 363, 399-401, 405
- signals
 - sending messages, 623
 - waves, 623-627
- single-area OSPF, 459
- single-area OSPFv2, 470-475
- single-mode fiber-optic cables, 47-49
- single-size subnets, 273-274
- SLAAC (Stateless Address Auto Configuration), 560, 598, 601
- slash masks, 305
- small office/home office (SOHO) LANs, 35
- small office/home office (SOHO) networks, 15
- SNA (Systems Network Architecture), 16
- Sniffer mode (APs), 647
- software configuration
 - common command prompts, 98
 - configuration files, 99-102
 - configuration mode, 96-97
 - configuration submodes and contexts, 97-99
- SOHO (Small Offices/Home Offices) LAN, 35
- networks, 15
- routers, 354
- solicited-node multicast addresses, 573-574
- source MAC addresses, 113

spanning-tree algorithm. *See* STA

spanning-tree commands, 259

spanning-tree mode command,
242-243, 259

Spanning Tree Protocol. *See* STP

spanning-tree vlan command, 244

**spanning-tree vlan x root primary
command,** 244-245

**spanning-tree vlan x root secondary
command,** 244-245

speed, switch interface configurations,
152-154

speed command, 98-99, 152-154, 165,
170, 355, 363

SPF (Shortest Path First) algorithm

Dijkstra SPF, 451

OSPF best routes, calculating, 457-459

split-MAC architectures, 638-643

SSH (Secure Shell), 91, 136-139,
432-434

SSID (Service Set Identifiers), 615

broadcasting, 681

multiple on one AP, supporting, 617

STA (spanning-tree algorithm), 216

startup-config file, 100

state change reactions (STP topology),
224-225

Stateless Address Auto Configuration.
See SLAAC

states

discarding, 230

interfaces, 215-217, 227

ports, 232

STP, 227

static default routes (IPv6), 592-593

static host routes (IPv6), 593

static ranges per subnet, choosing,
286-287

static routes, 367, 376

configuration, 379-384

default routes, 379

floating static routes, 381-383,
593-595

global unicast next-hop address, 589

host routes, 379-381

link-local next-hop address, 589-590

outgoing interface, 587-588

over Ethernet links, 591

overview, 586

static default routes, 383-384, 592-593

static host routes, 593

static network routes, 379

troubleshooting, 385-386, 595-598

**static unicast address configuration
(IPv6)**

configuration full 128-bit address,
557-558

enabling IPv6 routing, 558

generating unique interface ID with
modified EUI-64, 560-564

verifying, 558-560

status codes

routers, 358-359

troubleshooting, 162-163

STP (Spanning Tree Protocol),
114-115, 210, 243

802.1D standard, 228

BID, 218-219, 243-244

blocking criteria, 212, 216-217

BPDU, 218, 225

configurable priority values, 244

configuration, 240, 243-244

convergence, 216

EtherChannels, 234, 247-251

Forward delay timer, 225

forwarding criteria, 216-217

Hello timer, 225

interface states, changing, 227

LAN

redundancy, 210, 214

segment DPs, choosing, 222-223

- looping frames, preventing, 213
- MaxAge timer, 225
- modes, 242
- multiple STP, 241
- need for, 213-215
- PortFast, 235
- ports
 - blocking criteria*, 212, 216-217
 - cost*, 221
 - states*, 232
- purpose of, 215-217
- roles, 227
- root switches, electing, 218-219
- RSTP, compared, 229-230
- security, 236
- STA, 216
- standards, 242
- states, 227
- steady-state operation, 225
- switch reactions to changes, 226-227
- switch RP, choosing, 220-221
- system ID extensions, 243-244
- timers, 226-227
- topology influences, 223-225
- straight-through cable pinouts, 42-45**
- subcommands, 97**
 - auto-cost reference-bandwidth, 493
 - bandwidth, 492
 - ip address, 376
 - no network network-id area area-id, 483
 - switchport trunk allowed vlan, 204
- subdivided networks. *See* subnets**
- subinterfaces, 396-401**
- subnet masks, 272, 302. *See also* subnets**
 - classful IP networks before subnetting, 279-280
 - converting between formats, 305-309
 - difficult masks, 334-338
 - easy masks, 332
 - formats for, 304-305
 - hosts
 - borrowing bits to create subnet bits*, 280-281
 - calculating in network*, 313-315
 - choosing bits*, 281
 - mask formats, 282-283
 - prefix part, 309-312
 - sample design, 282
 - VLSM, 275
- subnet numbers, 272, 283, 334-336**
- subnets, 543. *See also* subnet masks**
 - addresses, 272, 283, 324, 327, 334-336
 - analyzing
 - subnet needs*, 269, 271
 - with decimal math*, 332, 339
 - assigning to different locations, 285
 - binary math, 326
 - Boolean math*, 331
 - finding range of addresses*, 331
 - finding subnet IDs*, 327
 - practice problems*, 328-329
 - shortcut for binary process*, 330
 - Boolean math, 331
 - broadcasts, 272, 283, 325-327, 336-338
 - building list of, 283-284
 - calculating, 313-315
 - decimal math, 331
 - difficult masks*, 334-338
 - easy masks*, 332
 - finding subnet broadcast addresses*, 336-338
 - predictability in interesting octet*, 333-334
 - reference table: DDN mask values and binary equivalent*, 339
 - definition of, 267, 322

- design choices, 276-284
- design views, 267-268
- dynamic ranges, choosing, 286-287
- examples of
 - networks with four subnets, 322-323*
 - simple example, 267*
- hosts, 268-271
- ID, 272, 283, 324, 330
 - finding with binary math, 327*
 - finding with decimal math, 334-336*
 - IPv4, 548*
 - IPv6, 548*
- IP addresses, 283-284, 302, 312
- IPv4, 70, 73, 545
- IPv6
 - assigning to internetwork topology, 549*
 - interface ID, 547*
 - listing, 548-549*
 - with global unicast addresses, 545-549*
 - with unique local addresses, 551-552*
- multiple subnet sizes, 274
- networks versus, 324
- number of hosts, 271
- number of subnets, 270
- one-size subnets, 273
- operational view, 267-268
- overview, 266
- plan documents, 267
- planning implementations, 284-287
- range of usable addresses, 325
- remote subnets, 375
- resident subnets, 322
- router anycast addresses, 549, 576
- simple example, 267
- single-size subnets, 273
- size of, 272-274
- static ranges, choosing, 286-287
- subnet numbers, 272, 283, 324, 327, 334-336
- VLSM, 275
- superior Hello messages, 219
- suplicants, 658
- SVI (Switched Virtual Interfaces), 392, 401-406
- switch ports, 110
- switches
 - access switches, 241
 - alternate ports, 229
 - auto-mdix, 45
 - backup ports, 230
 - BID, 218, 243-244
 - BPDU, 218, 225
 - Cisco Catalyst switches, 86
 - configuration files, 99-102
 - DHCP, 143
 - distribution switches, 241
 - EtherChannels, 234
 - Ethernet switches, 48
 - filtering decisions, 110-113
 - forwarding decisions, 110-113
 - history buffer commands, 144-145
 - interfaces, 87, 110, 118-120
 - autonegotiation, 158-162*
 - description, 152-154*
 - duplex, 152-154, 163-166*
 - enabling/disabling interfaces, 155-156*
 - Layer 1 problems, 166-168*
 - multiple interfaces, 154-155*
 - overview, 152*
 - removing configuration, 157-158*
 - speed, 152-154, 163-166*
 - status codes, 162-163*
 - troubleshooting, 162-168*
- IPv4, 140-144
- LAN segment DP, choosing, 222-223

- LAN switches, 35
 - analyzing*, 116
 - flooding*, 114
 - interface configuration*, 152-162
 - MAC address table*, 113-114, 117-124
 - overview*, 106-109
 - STP*, 114-115
 - summary*, 115-116
 - switch forwarding and filtering decisions*, 110-113
 - switch interfaces*, 118-120, 152-162
 - switching logic*, 109-110
 - verifying*, 116
- Layer 2 switches, 141, 183
- Layer 3 (multilayer) switches, 141, 184, 401-414
- links, 233
- MAC address tables, 111, 214-215
- management
 - DHCP*, 143
 - history buffer commands*, 144-145
 - IPv4*, 140-144
 - overview*, 126
 - security*, 128-139
- multilayer switches, 184
- PortFast, 235
- ports, 87, 230-233
- priority, 245-246
- root costs, 216
- root switches, 217-219, 247
- RP, choosing, 220-221
- RSTP switch priority, 247
- security, 128-139
- STP
 - reacting to changes*, 226-227
 - topology influences*, 223-225
- system ID extensions, 245-246
- unknown unicast frames, 114
- VLAN configuration, 140
- voice switches, 196
- switching tables. *See* MAC address tables
- switchport access vlan command, 185-189, 198-199, 207
- switchport command
 - Layer 3 switches, 415
 - routed ports, 408
- switchport mode access command, 185, 188, 198-199
- switchport mode command, 191, 207
- switchport mode dynamic auto command, 202
- switchport mode dynamic desirable command, 193
- switchport mode trunk command, 191, 203, 396
- switchport nonegotiate command, 195, 203, 207
- switchport trunk allowed vlan command, 204, 207
- switchport trunk encapsulation command, 191, 207
- switchport trunk native vlan command, 207
- switchport trunk native vlan vlan-id command, 205
- switchport voice vlan command, 198-199, 207
- switchport voice vlan vlan-id command, 200
- system ID extensions, 243-246

T

T1. *See* leased-line WAN tables

- ARP tables, 77, 378-379
- IP routing tables, 70-72, 388-389
- MAC address tables, 111-124, 214-215

tagging (VLAN), 181-182

TCP (Transmission Control Protocol), 20-21

TCP/IP (Transmission Control Protocol/Internet Protocol)

application layer, 19-20

data encapsulation terminology, 27-28

data-link layer, 25-26

history of, 16-17

HTTP, 19-20

IPv4, 22-25, 68-78, 140-144

network layer, 22-25

ARP, 77

DNS, 76-77

routing, 68-72

testing connectivity, 78

OSI, compared, 29

overview, 18

physical layer, 25-26

RFC, 18

transport layer, 20-22

Telnet, 90-91, 129, 432-434

terminal history size command, 145, 149

test etherchannel load-balance EXEC command, 255

testing

IPv4 connectivity, 78

LAN neighbors, 425-426

reverse routes, 423-425

WAN neighbors, 427

three-area OSPF (Open Shortest Path First), 460

time stamps, 661

timers

Hello/dead mismatches, troubleshooting, 512-513

Hello messages, 455

STP, 226-227

TKIP (Temporal Key Integrity Protocol), 660-661

topologies

AP noninfrastructure modes, 620-622

STP, 223-225

WLAN, 614-622

traceroute command, 428-432, 587

traffic flows, BSS, 615

trailer fields (Ethernet), 50

transmissions

bidirectional communication, 613

interference, 613

unidirectional communication, 613

unintended recipients, 652

transmitters, communication, 613

transmitting

frames, IP routing, 376

optimizing transmit power, 642

transport input all command, 139

transport input command, 138, 148, 356

transport input none command, 139

transport input ssh command, 139

transport input telnet ssh command, 139

transport layer (TCP/IP), 20-22

troubleshooting

EtherChannels, 251-253

Ethernet LAN, 166-168

Hello/dead timers, 512-513

interfaces, 162-168

IP routing

ping command, 419-429

SSH, 432-434

Telnet, 432-434

traceroute command, 428-432

Layer 3 EtherChannels, 413-414

Layer 3 (multilayer) switch SVI, 404-406

native VLAN, 205

neighbor adjacencies, 510-516

OSPF

- mismatched MTU settings*, 515
 - mismatched network types*, 515-516
 - neighbor adjacencies*, 510-516
 - shutting down processes*, 513-514
 - ping command, 419-429, 587
 - RID, 511
 - ROAS, 400-401
 - SSH, 432-434
 - static IPv6 routes, 595-598
 - static routes, 385-386
 - Telnet, 432-434
 - traceroute command, 428-432, 587
 - VLAN, 201-205
 - trunking**
 - 802.1Q, 182
 - administrative mode, 191
 - configuration, 191-195
 - dynamic auto mode, 191
 - dynamic desirable mode, 191
 - ISL, 182
 - overview, 180-181
 - type of, 191
 - VLAN
 - mismatched native VLAN*, 205
 - mismatched trunking operational states*, 202-203
 - supported VLAN list on trunks*, 203-205
 - tagging*, 181-182
 - VTP, 189-190
 - TTL (Time To Live), 429
 - TTL Exceeded (Time-to-Live Exceeded), 429-431
 - tunneling, CAPWAP, 639-640
 - two-switch topology, 123-124
- ## U
-
- UDP (User Datagram Protocol), 20
 - unabbreviated addresses (IPv6), 530
 - undebug all command, 104
 - undefined VLAN, troubleshooting, 201-202
 - unicast addresses, 50-52, 290, 322, 540, 556-564
 - unidirectional communication, 613
 - unified architectures. *See* centralized architectures
 - unique local addresses, 542, 551-553
 - universal addresses, 51
 - unknown addresses (IPv6), 574
 - unknown unicast frames, 114
 - URI (Universal Resource Identifiers), 20
 - URL (Uniform Resource Locators), 20
 - USB ports, 89
 - User Datagram Protocol (UDP), 20
 - user EXEC mode, 91-93
 - user mode**
 - external authentication servers, 135-136
 - passwords, 130-135
 - usernames, 133-135, 147
 - users, segregating into logical networks, 676
 - UTP (Unshielded Twisted-Pair) cables, 37
 - cabling pinouts, 42-49
 - overview, 39-40
 - UTP Ethernet links, 40-41
 - uWGB (Universal Workgroup Bridges), 621

V

verifying

- Data VLAN, 198-199
- EtherChannel configuration before adding interfaces, 251-253
- Ethernet switching, 116
- IPv4 on switch, 143-144
- Layer 3 (multilayer) switch SVI, 403-404
- OSPF
 - configuration*, 479-480
 - operation*, 475-478
- OSPFv2 interface configuration, 485-486
- ROAS, 398-400
- static unicast address configuration, 558-560
- Voice VLAN, 198-199

virtual interfaces (controllers), 674

VLAN (Virtual Local Area Networks)

- AP, 635, 668
- configuration, 185-195, 198-199
- Data VLAN, 197-199
- default VLAN, 186
- disabled VLAN, troubleshooting, 201-202
- dynamic interface ID, 678
- hopping, 205
- ID, 180
- interfaces, 402
- IP telephony, 196-200
- LAN trunking, 182
- mapping, 673
- native VLAN, 183, 205, 398
- overview, 179-180
- PVST+, 242-243
- routing, 183-184, 395-414
- split-MAC architecture, 640
- supported VLAN list on trunks, 203-205

- switches, 140
- tagging, 181-182
- troubleshooting
 - disabled VLAN*, 201-202
 - supported VLAN list on trunks*, 203-205
 - trunking*, 202-205
 - undefined VLAN*, 201-202
- trunking, 180-182, 189-195
- VLAN ID, 180
- Voice VLAN, 197-199
- vlan command, 185, 198, 207
- vlan number command, 201
- VLSM (Variable Length Subnet Masks), 275
- voice switches, 196
- VTP (VLAN Trunking Protocol), 189-190
- vtp mode command, 207
- vtp mode off command, 190
- vtp mode transparent command, 190

W - X - Y - Z

WAN (Wide Area Networks), 32, 60

- Ethernet WAN, 65-68
 - enterprise networks*, 350
 - point-to-point network type (OSPF)*, 506-508
- leased-line WAN, 61-65
- neighbors, testing, 427
- Serial WAN, enterprise networks, 350

waves

- continuous pattern, 623
- cycles, 625
- electric/magnetic, 624
- electromagnetic, 624
- frequency, 625-627
- propagation with idealistic antenna, 624

- WebAuth (Web Authentication), 657
- WEP (Wired Equivalent Privacy), 657
- WGB (Workgroup Bridges), 621
- wildcard masks, 473-475
- wired LAN. *See* Ethernet, LAN
- wired networks, 612-613
- wireless band frequencies, 627
- wireless LAN, 32
- wireless networks
 - 802.11 standard, 628-629
 - waves, 625
 - wired networks, compared, 612-613
 - WPA, 662-663
 - WPA2, 662-663
 - WPA3, 662-663
- WLAN (Wireless Local Area Networks)
 - 802.11 WLAN, 614
 - advanced settings, 684-685
 - AP, 668-669
 - BSS, 614-616
 - client session timeouts, 684
 - configuration, 675
 - advanced settings, 684-685*
 - controller configuration, 685*
 - dynamic interfaces, 678*
 - QoS, 683-684*
 - RADIUS servers, 676*
 - security, 681-682*
 - creating, 679-681
 - defined, 675
 - DS, 616-618
 - dynamic interfaces, creating, 678
 - ESS, 618
 - IBSS, 619
 - limiting, 676
 - listings of, displaying, 679
 - management access, allowing, 685
 - mesh networks, 622
 - outdoor bridges, 621-622
 - QoS, 683-684
 - RADIUS server, configuration, 676
 - repeaters, 620-621
 - security, 681-684
 - too many, creating, 676
 - topologies, 614-622
 - user segregation into logical networks, 676
 - WGB, 621
 - WLC, 669-675
- WLC (Wireless LAN Controllers)
 - activities, 642
 - centralized, 642-643
 - cloud-based architectures, 643
 - dynamic interfaces, 674-675
 - embedded deployments, 644
 - interfaces, 673-675
 - LAP, 639-640
 - management interfaces, 674
 - Mobility Express WLC deployments, 645
 - ports, 672-673
 - redundancy management, 674
 - service port interfaces, 674
 - virtual interfaces, 674
 - WLAN, 669-675
- working interfaces, defined, 217
- WPA (Wi-Fi Protected Access), 662-663
- WPA2 (Wi-Fi Protected Access version 2), 662-663
- WPA3 (Wi-Fi Protected Access version 3), 662-663
- write erase command, 104



REGISTER YOUR PRODUCT at CiscoPress.com/register Access Additional Benefits and SAVE 35% on Your Next Purchase

- Download available product updates.
- Access bonus material when applicable.
- Receive exclusive offers on new editions and related products.
(Just check the box to hear from us when setting up your account.)
- Get a coupon for 35% for your next purchase, valid for 30 days.
Your code will be available in your Cisco Press cart. (You will also find it in the Manage Codes section of your account page.)

Registration benefits vary by product. Benefits will be listed on your account page under Registered Products.

CiscoPress.com – Learning Solutions for Self-Paced Study, Enterprise, and the Classroom

Cisco Press is the Cisco Systems authorized book publisher of Cisco networking technology, Cisco certification self-study, and Cisco Networking Academy Program materials.

At CiscoPress.com you can

- Shop our books, eBooks, software, and video training.
- Take advantage of our special offers and promotions (ciscopress.com/promotions).
- Sign up for special offers and content newsletters (ciscopress.com/newsletters).
- Read free articles, exam profiles, and blogs by information technology experts.
- Access thousands of free chapters and video lessons.

Connect with Cisco Press – Visit CiscoPress.com/community

Learn about Cisco Press community events and programs.



Cisco Press

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
119.6799.240	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

APPENDIX E

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 11111100 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 11111000
```

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 11000000
```

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 11111000 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 11110000 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 ($32 - 24 \text{ network bits} - 5 \text{ host bits} = 3 \text{ subnet bits}$). 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 $32 - \text{prefix length}$ (in this case, $32 - 30 = 2$). This leaves 22 subnet bits ($32 - 8 \text{ network bits} - 2 \text{ host bits} = 22 \text{ subnet bits}$). 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 $32 - \text{prefix length}$ (in this case, $32 - 26 = 6$). This leaves 10 subnet bits ($32 - 16 \text{ network bits} - 6 \text{ host bits} = 10 \text{ subnet bits}$). 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 $32 - \text{prefix length}$ (in this case, $32 - 29 = 3$). This leaves 5 subnet bits, because $32 - 24 \text{ network bits} - 3 \text{ host bits} = 5 \text{ subnet bits}$. 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 ($32 - 24 \text{ network bits} - 4 \text{ host bits} = 4 \text{ subnet bits}$). 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 ($32 - 16 \text{ network bits} - 7 \text{ host bits} = 9 \text{ subnet bits}$). 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 $32 - \text{prefix length}$ (in this case, $32 - 21 = 11$). This leaves 13 subnet bits ($32 - 8 \text{ network bits} - 11 \text{ host bits} = 13 \text{ subnet bits}$). 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 $32 - \text{prefix length}$ (in this case, $32 - 25 = 7$). This leaves 1 subnet bit ($32 - 24 \text{ network bits} - 7 \text{ host bits} = 1 \text{ subnet bit}$). 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	$32 - (\text{network size} + \text{host size})$
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 11000000 00000000 00000000
AND result (subnet number)	10.128.0.0	00001010 10000000 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 11001000 00001010 00010010
Mask	255.224.0.0	11111111 11100000 00000000 00000000
AND result (subnet number)	10.192.0.0	00001010 11000000 00000000 00000000
Change host to 1s (broadcast address)	10.223.255.255	00001010 11011111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 01100 100 00010010 00010010
Mask	255.240.0.0	11111111 11110000 00000000 00000000
AND result (subnet number)	10.96.0.0	00001010 01100000 00000000 00000000
Change host to 1s (broadcast address)	10.111.255.255	00001010 01101 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 10.111.255.254

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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 01100100 00010010 00010010
Mask	255.248.0.0	11111111 11111000 00000000 00000000
AND result (subnet number)	10.96.0.0	00001010 01100000 00000000 00000000
Change host to 1s (broadcast address)	10.103.255.255	00001010 01100111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 10010110 11001000 11001000
Mask	255.252.0.0	11111111 11111100 00000000 00000000
AND result (subnet number)	10.148.0.0	00001010 10010100 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:

10.148.0.1 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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	$32 - (\text{network size} + \text{host size})$
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 01100100 00010010
Mask	255.255.128.0	11111111 11111111 10000000 00000000
AND result (subnet number)	10.220.0.0	00001010 11011100 00000000 00000000
Change host to 1s (broadcast address)	10.220.127.255	00001010 11011100 01111111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 01100100 01100100
Mask	255.255.192.0	11111111 11111111 11000000 00000000
AND result (subnet number)	172.31.64.0	10101100 00011111 01000000 00000000
Change host to 1s (broadcast address)	172.31.127.255	10101100 00011111 01111111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 01100100 01100100
Mask	255.255.224.0	11111111 11111111 11100000 00000000
AND result (subnet number)	172.31.96.0	10101100 00011111 01100000 00000000
Change host to 1s (broadcast address)	172.31.127.255	10101100 00011111 01111111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 11001000 00001010
Mask	255.255.240.0	11111111 11111111 11110000 00000000
AND result (subnet number)	172.31.192.0	10101100 00011111 11000000 00000000
Change host to 1s (broadcast address)	172.31.207.255	10101100 00011111 11001111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 11001000 00001010
Mask	255.255.248.0	11111111 11111111 11111000 00000000
AND result (subnet number)	172.31.200.0	10101100 00011111 11001000 00000000
Change host to 1s (broadcast address)	172.31.207.255	10101100 00011111 11001111 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 00110010 00110010
Mask	255.255.252.0	11111111 11111111 11111100 00000000
AND result (subnet number)	172.31.48.0	10101100 00011111 00110000 00000000
Change host to 1s (broadcast address)	172.31.51.255	10101100 00011111 00110011 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 00110010 00110010
Mask	255.255.254.0	11111111 11111111 11111110 00000000
AND result (subnet number)	172.31.50.0	10101100 00011111 00110010 00000000
Change host to 1s (broadcast address)	172.31.51.255	10101100 00011111 00110011 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 0000 1110
Mask	255.255.255.128	11111111 11111111 11111111 100 00000
AND result (subnet number)	172.31.140.0	10101100 00011111 10001100 000 00000
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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 10010110
Mask	255.255.255.224	11111111 11111111 11111111 11100000
AND result (subnet number)	192.168.15.128	11000000 10101000 00001111 10000000
Change host to 1s (broadcast address)	192.168.15.159	11000000 10101000 00001111 10011111

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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 0000 0001 00000001 00000001
Mask	255.248.0.0	11111111 1111 1000 00000000 00000000
AND result (subnet number)	10.0.0.0	00001010 0000 0000 00000000 00000000
Change host to 1s (broadcast address)	10.7.255.255	00001010 0000 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:

10.0.0.1 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** 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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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 ($256 - \text{mask's value in the interesting octet}$). 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	$32 - (\text{network size} + \text{host size})$
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