CHAPTER 1

Shooting Trouble

This chapter serves as the basis for the troubleshooting exercises throughout this book. In addition to a solid understanding of specific technologies, effective troubleshooting requires that you follow consistent procedures that are based on industry standards and reliable methods. The Open System Interconnection (OSI) model and the TCP/IP suite can help you methodically divide and conquer a problem or learn a new internetworking topic (by taking a layer-by-layer approach, for instance). This chapter presents an introduction to troubleshooting; a review of standards, protocols, and industry models; and practical troubleshooting, including baselining and documentation techniques. These standards, models, and techniques are covered in this chapter so that you can refer to them as you work through the specific troubleshooting tasks in this book. This chapter includes a Trouble Ticket designed to give you practical experience in solving real-world issues using Cisco's troubleshooting approach.

This chapter covers the following topics:

- Do You Shoot Trouble or Does Trouble Shoot You?
- Standards and Protocols
- Models and Methods
- Practical Troubleshooting

Do You Shoot Trouble or Does Trouble Shoot You?

Troubleshooting is all about reducing guesswork and eliminating the obvious. Following a systematic method is essential during the troubleshooting process. Methodical problem solving is the core of the CIT course, the CCNP Troubleshooting test, and this book, regardless of technical intricacies. Many times, whether or not you use a systematic method determines if you shoot trouble or if trouble shoots you.

Shooting trouble is often about questions. Do you ask the equipment or the user? Who is waiting for the results? What has happened? When did it occur? Why? Where did it happen? Are you using 10/100-Mbps Ethernet to the desktop; 155-Mbps ATM; or carrier services such as cable modems, digital subscriber line (DSL), wireless, ISDN, Frame Relay, Switched Multimegabit Data Service (SMDS), ATM, or long-haul Ethernet? The protocols, technologies, media, and topologies entail lots of complexity and the only thing constant is change. So where do you begin?

NOTE Appendix A material from the Cisco instructor-led Cisco Internetwork Troubleshooting (CIT) course for the CCNP Support exam is covered throughout this chapter and in more detail in the relevant chapters of this book. Consider this chapter fertile with test material; even more importantly, it makes an excellent practical review.

The first topic is standards and protocols. Think back for a moment to the last time you chatted with a friend. Certainly you and your friend had something to share, regardless of the method used to communicate. If you made a phone call, you were listening to each other talk. If you sent an e-mail or used a chat client, you were sending data back and forth. Whether it was your home phone, wireless phone, or PC, communications media was in place nonetheless. I assume that you waited for the friend to say hello first and that you took turns talking. You spoke the same language or understood multiple languages. Hopefully, you were polite enough to not talk while the other person was talking. You may have had to troubleshoot some issues while talking with the friend. Perhaps a lightning storm hit your phone line or you dialed the wrong number. Maybe you didn't pay your phone bill and the service was turned off. The friend may not have answered or the phone may have been busy. Maybe your friend had caller ID and picked up right away because it was you. Regardless of your exact scenario, throughout the contact you had to decide your next step.

NOTE Continue to think about your communications with your friend as you read through this chapter. You may begin to see how a different perspective or an analogy can help you to simplify complex topics. Throughout this book, I include occasional analogies I have found to be very helpful to my students learning in the classroom.

Standards and Protocols

Communication rules are referred to as *standards* and *protocols*. Playing with the right rules to the game normally means you are more apt to communicate well in the networking game. Standards are rules, conditions, and requirements that can be de jure, de facto, proprietary, or open. *De jure standards* are official; by legislation they are endorsed by a standards body, such as those listed in Table 1-1.

Table 1-1 *Standards Bodies*

Standards Body	Acronym	Examples
American National Standards Institute www.ansi.org	ANSI	C Cobol Fortran X3T9.5
International Telecommunication Union www.itu.int	ITU	V.22 V.32 V.34 V.42
Institute of Electrical and Electronic Engineers standards.ieee.org www.ieee.org	IEEE	802.2 LLC* 802.3 Ethernet 802.5 Token Ring
International Organization for Standardization www.iso.org	ISO (not an acronym)	OSI IS-IS
Electronic Industries Alliance/ Telecommunications Industry Association www.eia.org www.tiaonline.org	EIA/TIA	EIA/TIA 568 Commercial Building Telecommunications Wiring Standard RS-232 EIA/TIA 232
Internet Engineering Task Force www.ietf.org	IETF	RFCs
Internet Assigned Numbers Authority www.iana.org	IANA	Port and protocol numbers

*LLC = Logical Link Control

TCP/IP and OSI are examples of nonproprietary open standards that are widely used today. Standards are wonderful things; that's why we have so many. Webopedia (www.webopedia.com) defines standard as a definition or format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry. Standards exist for programming languages, operating systems, data formats, communications protocols, and electrical interfaces.

As an example of an evolution of technology through standards, consider the creation of the Internet. According to "20 Questions: How the Net Works," by Scot Finnie at www.scotfinnie.com/20quests/hownet.htm#Q1, no one person or group can claim this fame; however, in 1962 a series of memos discussed the "Galactic Network Concept" from MIT's J.C.R. Licklider. Licklider later became the head of the Department of Defense (DoD)

Advanced Research Projects Agency (ARPA). TCP/IP research began in 1961, and in 1967 ARPA's Lawrence Roberts published his plan for the worldwide network. Tests were conducted for several years, and e-mail and the Internet made their first public appearances in 1972. TCP/IP protocols and services made their way into the network in the 1970s. The World Wide Web (WWW) was born in the late 1980s. The National Science Foundation (NSF) took over the management of ARPANET in 1990. In the mid-1990s, NSFnet was turned over to a consortium of public providers we know today as Internet service providers (ISPs). Many standards bodies are responsible for the Internet's existence and maintenance, including the following:

- Internet Society (ISOC), which includes the Internet Architecture Board (IAB) for broad direction and overall architecture and the Internet Engineering Steering Group (IESG).
- Internet Assigned Numbers Authority (IANA) and Internet Network Information Center (InterNIC) for IP addresses, domain names, and other numbers.
- World Wide Web Consortium (WC3) for HTML and web standards.
- Internet Engineering Task Force (IETF) for RFCs and smooth operations.
- Internet Research Task Force (IRTF) for ongoing research.

TCP/IP open standards (nonproprietary) are based on Request For Comments (RFCs); whereas, proprietary standards are vendor-specific. Refer to www.rfc-editor.org to read RFCs and for more detail on the RFC process, including a tribute to Jon Postel who was *the* RFC Editor. Figure 1-1 shows the RFC Editor. Also refer to www.ietf.org/rfc/rfc2026.txt for particulars.

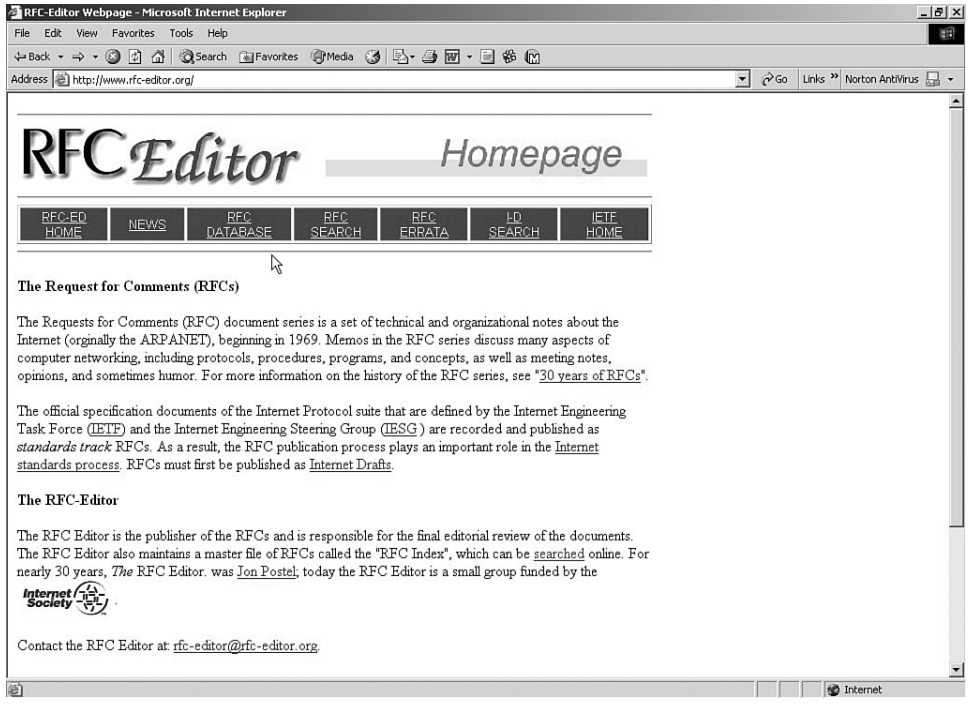
Anyone can propose a new standard, which then goes through various levels toward maturity. All RFCs start out as drafts, but not all drafts mature to RFCs. When published, RFCs do not change. Updates get a new RFC number. You can review private addressing in RFC 1918, for example, which obsoletes the original RFC 1597.

De facto standards include examples such as the Hayes command set for controlling modems, the Kermit and Xmodem communications protocols, and the printer control language (PCL) and postscript for laser printers. Although numerous de facto standards may have started as proprietary implementations, by the time they are regarded as de facto standards there are many different vendor implementations. One example of this is quite relevant to Chapter 7, "Shooting Trouble with VLANs on Routers and Switches"; the example is the two different ways that Ethernet trunking can occur:

- InterSwitch Link (ISL), which is the Cisco proprietary method
- 802.1Q, which is the IEEE standard

Standards are important. They enable different people (and different vendors) to approach a task in a similar way to achieve a similar solution that works. Standards can be categorized by how they are recognized: proprietary or open. Proprietary beginnings tend to produce de facto standards (Hayes, Kermit, and PCL). Open beginnings tend to produce de jure standards (TCP/IP and RFCs). If it is truly a proprietary solution and other vendors cannot use it, it probably is not a standard. A standard really refers to a solution available to multiple vendors.

Figure 1-1 *RFC Editor*



Now that I have defined standards and the standards process, what about the need for protocols? *Communications protocols* are rules governing the transmitting (Tx) and receiving (Rx) of data so that different end systems or applications can communicate with one another. A *protocol* is an agreed-upon format for transmitting data between two devices. The protocol determines how the sending and receiving devices communicate, such as the indicator for sending and receiving a message. The protocol also defines the type of error-checking and data-compression methods if any are used.

Examples of protocol suites include TCP/IP, OSI, IEEE, AppleTalk, DECnet, Novell Internetwork Packet Exchange (IPX), and IBM Systems Network Architecture (SNA). A protocol suite or stack is like many subcontractors building a house. Brick layers take care of the foundation, the electricians put in the wires, the plumbers install the pipes, the framers frame it up, roofers carry out their part, and finally the homeowners do their own finishing touches. In networking, different protocols operate at each layer to carry out fundamental functions such as encapsulation, segmentation and re-assembly, connection control (connection-oriented or connectionless), flow control, error control, multiplexing, and delivery.

These protocols use rules to dictate how communication is established. Unless everyone plays by the same rules, communication is not possible. As a fun demonstration of the importance of

standards and protocols, I gave several groups of technology students a card game to play. The rules were on a piece of paper given to each group. Unbeknownst to them, each group was given a slightly different set of rules. (One sheet said ace is high, another said the joker is a wildcard, and yet another said joker loses.) Each group was instructed to play by the rules and not to talk. When they were comfortable in their own little groups of four or five, I moved one person from each team to another group. It was chaos, to say the least, as they tried to play the game with different understandings of the rules. They finally figured it out and agreed that a standard set of rules (protocol) is definitely beneficial.

NOTE Understanding standards and protocols and their layered approach will assist you in applying internetworking skills and shooting trouble in a practical environment. In addition, with such understanding you will be on your way to passing many certification tests.

Models and Methods

Models are guidelines for communications and methods for troubleshooting. This section covers the ISO's OSI model, the DoD's TCP/IP suite, and Cisco's seven-step approach to troubleshooting.

The OSI Model

You have probably dealt with the OSI model more times than you care to remember. Hopefully, however, this review will make the OSI model meaningful to you. Use it to troubleshoot the practical lab scenarios that follow as well as to understand and review internetworking topics.

ISO began work on the OSI model in the late 1970s and published the OSI reference model in 1984 to facilitate interoperability among vendors. It is one of the best troubleshooting models around, and every certification vendor will test to make sure you are an expert in this area. Be aware, however, that every vendor has its own approach to OSI. (I write from experience here; I have been heavily involved in not only Cisco, but also Microsoft, Novell, and CompTIA (A+/Network+) certification course delivery over the years.)

Although the focus here is on understanding the OSI model and using it to troubleshoot, the OSI model provides other benefits as well (such as interoperability and standardization, and it enables you to subdivide developer tasks without having to alter other layers). For example, network interface card (NIC) vendors really don't want to be concerned with what upper-layer applications and protocols run over the hardware. However, NIC vendors must be concerned with LAN technologies such as Ethernet, Token Ring, and what physical specifications (cable and connectors) to follow.

Please Do Not Threaten Support People Again

I love mnemonics. They may seem simple, but they can be surprisingly effective in helping commit principles to memory. In this case, **P**lease **D**o **N**ot **T**hreaten **S**upport **P**eople **A**gain is a tool to help you remember the seven layers of the OSI model, as displayed in Table 1-2. Note the layers and protocol data units (PDUs) in Table 1-2. Although often referred to as just plain old packets, PDUs actually came from the ISO.

Table 1-2 *OSI Layers and PDUs*

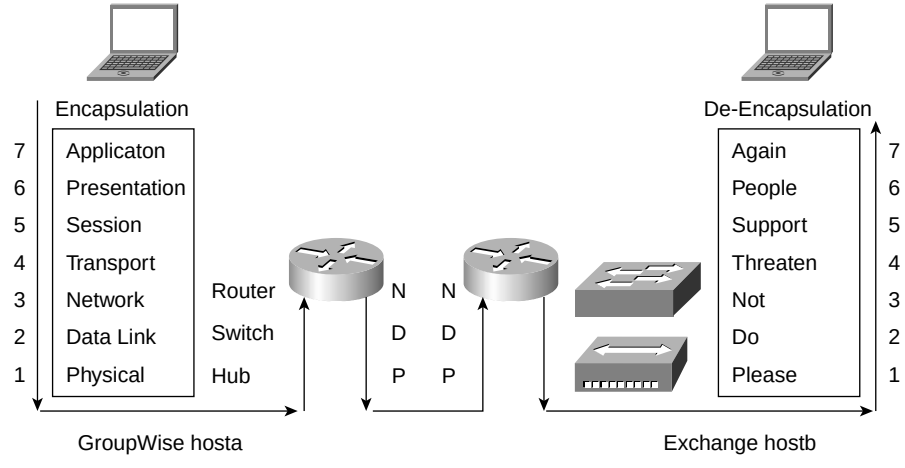
OSI Layer Number	OSI Layer Name	PDU	Mnemonic
7	**A**pplication	Messages (data, voice, video)	**A**gain
6	**P**resentation	Messages (data, voice, video)	**P**eople
5	**S**ession	Messages (data, voice, video)	**S**upport
4	**T**ransport	Segments (TCP*)/datagrams (UDP*)	**T**hreaten
3	**N**etwork	Packets/datagrams	**N**ot
2	**D**ata Link	Frames	**D**o
1	**P**hysical	Bits	**P**lease

*TCP = Transport Control Protocol

*UDP = User Datagram Protocol

Take each layer and examine the services provided to or from the next layer. It is helpful to draw a picture of two end systems communicating to understand the layers. (See Figure 1-2.) Often you miss a lot of important host-to-host activity if you look only at the source or the destination host of one protocol stack. Figure 1-2 shows GroupWise hosta, which sends e-mail to Exchange hostb. The general layered approach is presented here, not all the application details.

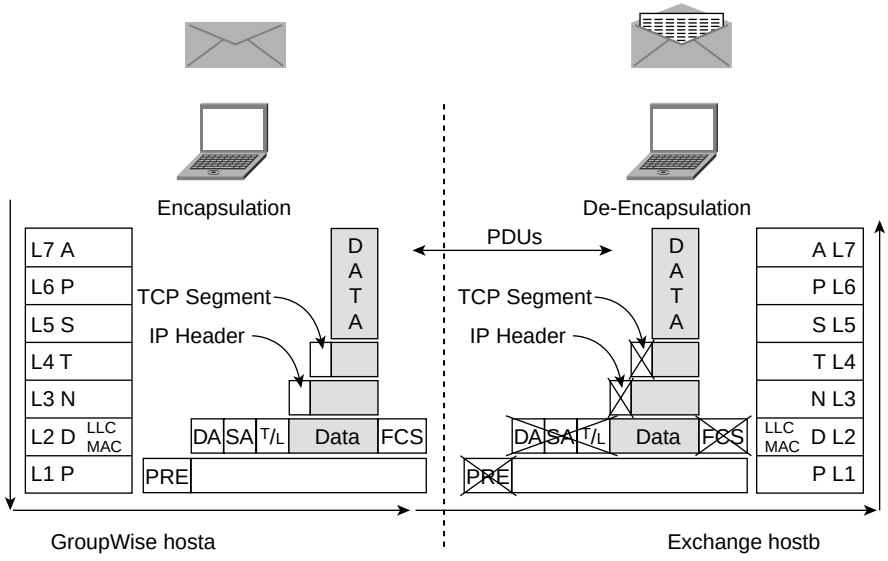
Figure 1-2 *End-System Data Flow*



Notice how the source GroupWise hosta encapsulates the message as it works its way from Layer 7 to Layer 1 across the wire. Assuming that the destination Exchange hostb is on another network, lots of encapsulation/de-encapsulation occurs between Layers 1 through 3 until the packet gets to the destination host (router-to-router operations). The destination host pulls the frames off the wire and processes (de-encapsulates) them up the stack from Layer 1 to Layer 7 so that the e-mail application can read the e-mail. The processing includes any necessary re-ordering and re-assembly of packets that result from packet routing and fragmentation.

Understanding a layered approach and packet flows is critical to being a good troubleshooter. That's why vendors put all that theory stuff in their courses. If you don't know how things work correctly, how in the world do you know what is wrong? End system–to–end system Exchange and GroupWise messaging is the main example I share with students in many of my classes. Take a look again at Figure 1-2 and then at Figure 1-3 (on encapsulation) to review the packet flow and layer operations. For more detail, refer to materials on CCNA and CCNP from Cisco Press and other publishers. I particularly like Jeff Doyle's *Routing TCP/IP*, Volumes I and II, and think they belong on everyone's shelf.

Figure 1-3 *Encapsulation*



Encapsulation (framing) is like wrapping presents for someone else so that he or she can tear the wrapping paper off. Another analogy many people use when referring to encapsulation is that it is like writing a letter and stuffing the letter in an IP envelope to be delivered to a destination. Think of encapsulation as placing a letter in your mailbox and putting the flag up to let the postal worker know to pick up and deliver the letter; this analogy will help you analyze the fields of the IP header in Chapter 3, "Shooting Trouble with IP." Each hop (Layer 3 device)

along the way strips off the packaging (Layer 2 framing/encapsulation) and repackages (Layer 2 framing/encapsulation) for the next hop closer to the destination (Layer 3). Figure 1-3 illustrates the encapsulation /de-encapsulation process for Ethernet, including the destination address (DA), source address (SA), type or length field (T/L), and the frame check sequence (FCS), all of which are examined in more detail in Chapter 5, "Shooting Trouble with Ethernet."

Each layer adds a *header*, which is nothing more than a set of instructions for its peer layer. With TCP/IP, for example, the upper-layer messages (data, voice, or video packets) get encapsulated (stuffed) inside of a TCP segment or UDP datagram at the Transport Layer for delivery. The Transport Layer segment (connection-oriented) or datagram (connectionless) gets encapsulated (stuffed) inside of the Network Layer IP packet or datagram (connectionless). The number of segments sent before acknowledgement is required may vary (windowing). The IP packet gets encapsulated (stuffed) inside of the Data Link Layer frame. In Ethernet, for instance, the preamble (PRE) starts the frame and the trailer (cyclic redundancy check [CRC] or FCS) ends the frame. If necessary, an Address Resolution Protocol (ARP) packet is broadcast (local broadcast) to resolve the destination IP address (Layer 3) to its equivalent Media Access Control (MAC) address (Layer 2). If the destination host is on the same subnet, the MAC is the destination host's address. If the destination host is on a different subnet, the resulting resolution is generally the default gateway (local router interface) MAC address. ARP is not necessary across a serial point-to-point link because it is not a broadcast segment like Ethernet. The IP packet destination IP address doesn't change during normal destination-based routing; however, the Layer 2 MAC addresses change each hop along the way.

NOTE The preceding paragraph discusses IP, but this layered approach certainly applies to various protocol stacks (such as Novell IPX, IBM SNA, AppleTalk, and so on).

Networking is limited by the standards that prevail. Even though 10-MB, 100-MB, and Gigabit Ethernet standards are available today, for example, the frame size is still limited to 1500 bytes. What if everything doesn't fit into the frame? Think of it like sending a box of Christmas gifts rather than just a Christmas card. You could get a bigger box to put all the presents in or send lots of smaller ones. Just like the Christmas box, if everything doesn't fit in the frame, IP fragments the data into smaller packets (chunks) each hop along the way according to the frame type or the maximum transmission unit (MTU) set on the interface. The initial packet ID number may be randomly generated, but the subsequent packet IDs are sequential in nature for re-ordering and re-assembly purposes. Some Layer 3 protocols, such as IPX, don't fragment the data at all. The Physical Layer requires bits (0s and 1s) to traverse the wire. A lot of activity occurs among the lower layers until the packets reach the destination host.

De-encapsulation is like opening envelopes or presents. Each layer reads and carries out the instructions from its peer layer, discards the header (instructions), and sends the packets up the stack for further processing. Each layer receives services from the layer below and provides services to the layer above it.

The following sections cover the OSI model layer by layer. It is assumed that you are somewhat familiar with the layers and abbreviations and acronyms discussed with regards to each layer. If not, you can find more information at websites such as www.acronymfinder.com, www.shoretraining.com, www.learntcpip.com, www.computerlanguage.com, www.whatis.com, www.amazon.com, www.certificationzone.com, and www.cisco.com.

NOTE Remember that protocols and applications are written to perform functions, and the focus here is using the OSI model as a *model* to understand and troubleshoot them. If you really want to know the technical details (for an engineering standpoint), you should read the ISO documents.

Layer 7: The Application Layer

Layer 7, the Application Layer, is all about servers providing services and users requesting to use those services. Servers provide shared services, such as file, print, message, database, network management, communications, and application services. Clients request the same services. This reminds me of going out to eat. The restaurant hostess seats you with the menus, and a server comes to the table to take your order (providing you with services). You, as a customer (client), order your food (request services) and indulge as usual.

Application Layer examples include the user interface, X.400 Mail services, X.500 Directory services, Simple Mail Transport Protocol (SMTP), Internet Message Access Protocol (IMAP), Post Office Protocol (POP), Simple Network Management Protocol (SNMP), FTP, TFTP, HTTP, telnet, Domain Name System (DNS), Bootstrap Protocol/Dynamic Host Configuration Protocol (BOOTP/DHCP*), Network File System (NFS), gateways, Border Gateway Protocol (BGP*), Routing Information Protocol (RIP*), and so on. Rout**ing** protocols are generally thought of at the Network Layer (Layer 3). Because BGP operates over TCP port number 179 and RIP operates over UDP port 520, however, many people choose to list them here. DHCP operates over UDP ports 67 and 68.

NOTE Many different opinions exist as to how to best classify routing protocols. It is important to keep in mind that many management and control type protocols obviously support Layer 3 functions rather than transfer data. Examples include such services as DHCP, BGP, and RIP, which I have marked with an asterisk (*) in the preceding paragraph. It is impossible to make everything fit nicely into the layers.

Layer 6: The Presentation Layer

Layer 6, the Presentation Layer, is the *translator.* Presentation is everything. How about that big hunk of cheesecake for desert with strawberry glaze on the plate? The waiter wrote down your order, but can the person serving the desert interpret it?

Think of translation from one application to another application (translation of, for example, such things as character codes and syntax, encryption, and compression). In the Cisco environment, compression is often thought to relate to Layer 2; I will cover that detail in the WAN chapters (Chapter 8, "Shooting Trouble with Frame Relay," and Chapter 9, "Shooting Trouble with HDLC, PPP, ISDN BRI, and Dial Backup"). Presentation Layer examples include ASCII, Extended Binary Coded Decimal Interchange Code (EBCDIC), Tagged Image File Format (TIFF), Joint Photographic Experts Group (JPEG), Musical Instrument Digital Interface (MIDI), MPEG-I Audio Layer III (MP3), Moving Picture Experts Group (MPEG), Rivest Shamir Adleman (RSA), Data Encryption Standard (DES), Secure Sockets Layer (SSL), and Transport Layer Security (TLS).

Layer 5: The Session Layer

Layer 5, the Session Layer, is the operator or dialog layer. It establishes, maintains, and tears down communication sessions within the operating system using protocols such as remote-procedure calls (RPCs), Lightweight Directory Access Protocol (LDAP), Network Basic Input/Output System (NetBIOS), sockets, Server Message Block (SMB), or Network Control Program (NCP). Communications examples include the following:

- Simplex (one way, like a television or radio broadcast)
- Half-duplex (one way at a time, like my Nextel walkie-talkie phone)
- Full-duplex (simultaneous, like telephones and networks)

NOTE The upper three layers of the OSI model are referred to as the Application Layer in the TCP/IP suite of protocols. From a troubleshooting standpoint, these layers typically relate to software problems in end systems and name resolution issues.

Layer 4: The Transport Layer

Layer 4, the Transport Layer, is all about host-to-host delivery. This layer hides lower-layer problems from upper layers in that it provides error detection and correction on the receiving end (host). In addition, it segments and re-assembles data for upper-layer applications based on various TCP and UDP port numbers. Application multiplexing is common (just like when you press Alt+Tab to cycle through your open programs in Windows). For example, you may be running a web browser (HTTP port 80), telnetted into a router (TCP port 23), and copying configurations to a TFTP server (UDP port 69) or FTP server (TCP ports 20 and 21) all at the same time. Normally, systems run out of resources before they run out of *ports* (pointers to applications).

TCP and UDP are the most common examples at Layer 4 for TCP/IP; the equivalent IPX/Sequenced Packet Exchange (SPX) transport is SPX. TCP is *connection-oriented*, which means the host must establish a logical connection, such as a 3-way handshake, before

communications can occur. Flow control occurs through windowing, and TCP is *reliable* in that it uses acknowledgements (acks) and negative acknowledgements (naks). UDP is *connectionless*, which means it does not require an established connection before communications can occur. It is unreliable at the Transport Layer, which means that the reliability is left up to the Application Layer. TCP is like the certified mail protocol; whereas, UDP is like the regular mail protocol.

Rout**ing** protocols are generally thought of as relating to the Network Layer (Layer 3). Because Interior Gateway Routing Protocol (IGRP, protocol number 9), Enhanced IGRP (EIGRP, protocol number 88), and Open Shortest Path First (OSPF, protocol number 89) operate side-by-side with TCP and UDP, however, they are often discussed as Layer 4 protocols. This leaves reliability up to the upper-layer protocols.

NOTE Protocol numbers and port numbers are different. Port numbers link the Transport Layer to the upper layers. FTP is an application that operates based on TCP ports 20 and 21; TFTP is an application that operates based on UDP port 69. Protocol numbers link the Network Layer to the Transport Layer, whereas service access points (SAPs) or type codes link the Layer 2 frame to point to Layer 3. You can access an excellent site for details on protocol and port numbers by the layers at www.networksorcery.com/enp/topic/ipsuite.htm.

Layer 3: The Network Layer

Layer 3, the Network Layer, is where routers or Layer 3 switches operate. By the way, Layer 3 switches are routers, and Layer 2 switches are bridges. Path determination and routing is all about moving things from one place to another. You do it every day with the telephone, mail, planes, trains, cars, boats, busses, subways, and so on. Do you take the fastest route, the best roads, the scenic route, or do you figure it out as you go? The routing table directs the packets as to where to go and drops them in the bit bucket if it doesn't know what to do with them. Do you use a map (link-state rout**ing** protocols) or do you just stop at the gas stations along the way (distance vector rout**ing** protocols)? Either way, your car or other form of transportation (rout**ed** protocol such as IP or IPX) carries you (the data) and any upper-layer instructions (headers) hop-by-hop to your destination. The router strips off the old framing (Layer 2 packaging) and re-encapsulates the packet for the outbound interface according to the destination IP address in the data packet header. Layer 2 addresses change from hop-to-hop, but the Layer 3 addresses stay the same assuming normal destination-based routing.

NOTE According to the ISO documents, rout**ing** protocols stand outside the basic protocol stack in a management plane and provide management services for the Network Layer. Although this discussion focuses on the OSI model as a model, it is more than just any old model. It is a set of ISO documents. Spend the money and read the ISO documents. Alternately, for a small fee you can subscribe to www.certificationzone.com for some very comprehensive OSI study guides by Howard Berkowitz and Katherine Tallis.

As displayed in Figure 1-4, routers route using a hop-to-hop relay system to get packets one step closer to their destination. Routers accept a frame on one interface, strip off the Layer 2 header, and select an outbound interface closer to the destination. The router adds a new Layer 2 header (re-encapsulates the packet) and switches (forwards) from the inbound interface to the outbound interface within the router to transmit the packet.

Figure 1-4 *Routing and Switching Process (Within the Router)*

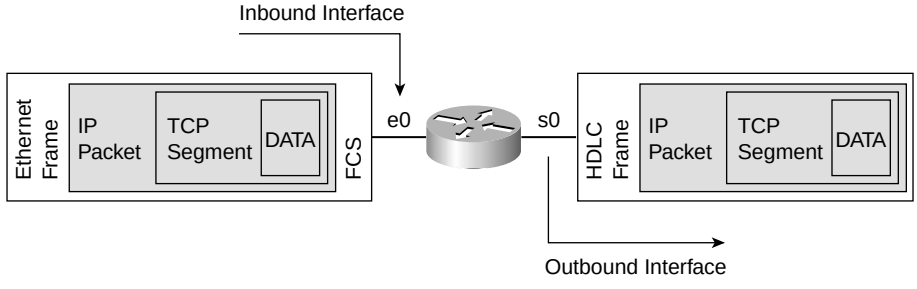


Figure 1-4 illustrates how the router accepts the Ethernet frame on inbound interface e0 and strips off the Layer 2 Ethernet frame leaving the upper-layer data intact. According to the destination IP address in the IP header, the router does a route table lookup to see which outbound interface will get the packet closer to its destination network. The router adds a new Layer 2 header to encapsulate the data and forwards it to its next hop. Next-hop reachability is not only a key point in getting packets to their destination network, it is also a key point in troubleshooting.

Routers *route* to the destination network address. They buffer and *switch* packets from the inbound interface to the outbound interface within the router. Performance is definitely affected by the switching type. *Fast switching* refers to when a router does a route table lookup for the first packet toward a destination and caches it so that it doesn't have to perform a route table lookup on each and every packet. (Imagine the overhead if a router actually performs a route table lookup on each and every packet, which is called *process switching* and is used when you perform such tasks as debug commands.) Newer devices offer Cisco Express Forwarding (CEF) as a switching type, whereby even the first packet gets cached. Remember these important points: Routers *route* hop-to-hop, and routers *switch* from the inbound interface to the outbound interface of the router at Layer 3. Chapters 6, "Shooting Trouble with CatOS and IOS," and 7, "Shooting Trouble with VLANs on Routers and Switches," discuss switching types (architectures) in more detail.

Much activity occurs at Layer 3. IP, the connectionless Internet Protocol, is the heart of TCP/IP-based applications. Connectionless is unplanned and without prior coordination (as is UDP at Layer 4). Each packet stands alone; no negotiation occurs. Think about this when you travel to various locations and mail postcards to people you haven't seen for a while.

IP and IPX are rout**ed** protocols responsible for delivery of packets, including rout**ing** protocol packets that are based on IP and IPX respectively. Rout**ing** protocols exchange routes with other routers. Rout**ed** protocols deliver packets; they send user data. This section briefly reviews rout**ing**/rout**ed** protocols for troubleshooting purposes.

NOTE Refer to the book *CCNP Practical Studies: Routing* (Cisco Press) or ACRC/BSCN/BSCI-related courses and books for more details on routing protocols. Although they all have good information, Building Switched Cisco Internetworks (BSCI) replaces Building Scalable Cisco Networks (BSCN), which replaced Advanced Cisco Router Configuration (ACRC).

Routed protocols transport packets through routers. Routing is a relay system, a hop-by-hop paradigm from one network to another. Routers filter based on Layer 3 logical network addresses. The router strips and rebuilds the Layer 2 framing according to the outbound interface. Route filters, such as access control lists (ACLs), distribution lists, route maps, and prefix lists, allow further filtering.

TCP handles end-to-end connectivity, whereas the transport of the data is handled by the connectionless IP. Each router from the source to the destination makes a decision.

Rout**ing** protocols *route* rout**ed** protocols. Rout**ing** protocols give directions; rout**ed** protocols carry the data. Rout**ing** protocols are used by routers to exchange data. Table 1-3 gives a brief comparison of routing protocols.

Besides learning from routing protocols, routers know about directly connected routes. Directly connected routes are like your arms and legs; they are attached networks. Basically, the router needs driving directions just as you and I need them. For example, you know where your immediate family and friends live. They might be in the same state, town, or even on the same street. You can also learn of other locations; perhaps the location you're looking for is right next door (directly connected routes), perhaps you look up an address on a website such as Yahoo! Maps (link-state routing protocols), or perhaps someone else gives you directions (distance vector routing protocols).

Table 1-3 *Routing Protocol Comparison*

Feature	IP RIP	IGRP	EIGRP	OSPF	IS-IS	BGP
Open or Proprietary	Open	Proprietary	Proprietary	Open, but IP support only	Open	Open
Network size	Small	Medium	Large	Large	Very Large	Very Large
Distance vector or link state	Distance vector (Routing by rumor)	Distance vector (Routing by rumor)	Advance distance vector (hybrid) (Routing by rumor)	Link-state (Routing by map)	Link-state (Routing by rumor)	Path vector (Routing between autonomous systems)
Interior or exterior	IGP	IGP	IGP	IGP	IGP	EGP
Updates	30-second broadcast updates RIPv2 is 224.0.0.9	90-second broadcast updates	Triggered updates 224.0.0.10	224.0.0.5 224.0.0.6 Link-state packets	Triggered updates Link-state Packets	Triggered unicast updates

Table 1-3 *Routing Protocol Comparison (Continued)*

Feature	IP RIP	IGRP	EIGRP	OSPF	IS-IS	BGP
Port or protocol number	UDP port 520	Protocol number 9	Protocol number 88	Protocol number 89	Protocol I CLNP (81) ES-ES (82) IS-IS (83) IP (CC)	TCP port 179
Administrative distance	120	100	90/170	110	115	200/20
Metrics	Hop count	Bandwidth Delay Reliability Load MTU (**B**ig **D**ogs **R**eally **L**ike **M**eat)	Bandwidth Delay Reliability Load MTU (**B**ig **D**ogs **R**eally **L**ike **M**eat)	Cost	Default (optional) Delay Expense Error	Attributes: Weight Local pref MED Origin AS-path Next-hop Community Others
Algorithm	Bellman-Ford algorithm	Bellman-Ford algorithm	DUAL algorithm	Dijkstra/SPF* algorithm	Dijkstra/ S algorithm	Shortest AS* path
Support for VLSM* and summarization	VLSM and summarization (in RIPv2)	N/A	VLSM and summarization Automatic classful summarization by default Manual summarization per interface	VLSM and summarization Manual summarization at ABR*/ ASBR* only	VLSM and summarization	VLSM and summarization

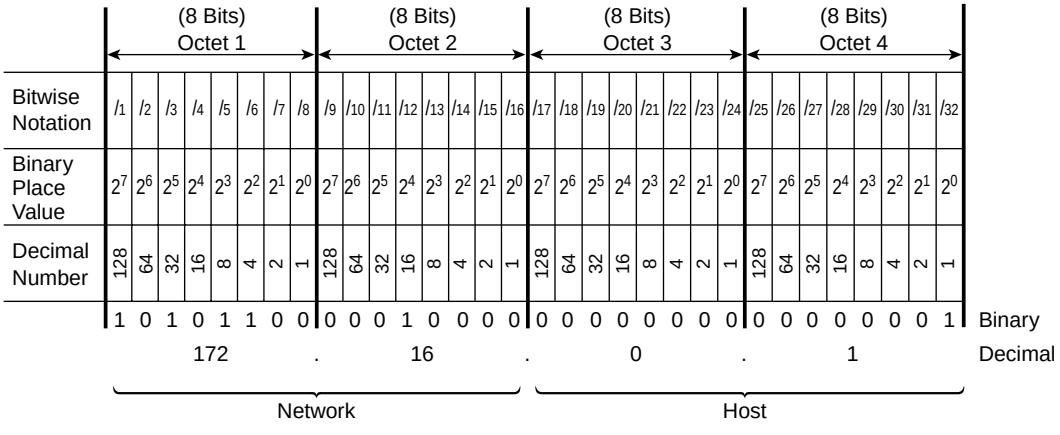
*AS = Autonomous system ABR = Area Border Router
SPF = Shortest Path First VLSM = Variable-Length Subnet Masking
ASBR = Autonomous System Boundary Router (or Border)

The routing table is also populated by static and default routes, which are not always automatically propagated to other routers by default. Default routes are very useful in stub network scenarios where there is only one way in and one way out. Static and default routes can eliminate routing update traffic in many cases—but don't be tricked into packets getting sent but not returned because they don't have a return route.

At this point in your CCNP preparation, you should be very comfortable with routing, IP addressing, subnetting, and summarizing. The labs in later chapters will certainly determine whether you have mastered these concepts. In the meantime, some IP examples appear here for your review.

IP version 4 mathematically allows for 4.2 billion addresses (2^{32}). Base 2, 32-bit, dotted-decimal addresses such as 172.16.1.1 are used. Figure 1-5 shows all 4 octets, which are comprised of a total of 32 bits (8 bits each). It is common to see the subnet mask listed as / *number* in the routing table to illustrate the number of network bits in the mask (as shown in the bitwise notation row). The next row is a power of two for the binary place value. The last row is the decimal equivalent or base 10 representation of the binary place value above it. Use a graphic such as this to assist you with subnetting and summarizing.

Figure 1-5 *Binary Place Values*



NOTE IP version 4 addresses are in a 4-octet, dotted-decimal, 32-bit format, whereas version 6 is 128 bits written as 8 groups of 4 hex digits separated by colons (such as 0000:AAAA:1111:BBBB:2222:CCCC:3333:DDDD).

By the way, IP version 4 still has a long life ahead of it because of address-exhaustion solutions such as private addresses, proxy servers, Network Address Translation (NAT), and Classless Interdomain Routing (CIDR). For a valid public address, contact your ISP or www.arin.net in the Americas for details. ARIN is one of the three regional internet registries the authority in the U.S. RIPE NCC is for Europe, the Middle East, North Africa, and parts of Asia. APNIC is the Asia Pacific Network Information Centre.

Private addresses are not routed on the Internet and fall within the following ranges:

- 10.0.0.0/8
- 172.16.0.0 through 172.31.0.0/12
- 192.168.0.0 through 192.168.255.0/16

NOTE Private addresses are used throughout the rest of the chapters so that I don't step on anyone's toes.

Table 1-4 displays Class A, B, and C addresses, which are available for hosts, and also shows Class D and E addresses, which are reserved for other purposes. Notice the pattern of **0**, 1**0**, 11**0** in the first octet binary range. You may be familiar with the Jackson 5 "A-B-C" song; the first octet follows that tune precisely: "A-B-C, it's easy as 1-2-3…" where the 1-2-3 is the bit position of the 0. (Thank you Glenn Tapley.)

Table 1-4 *Classes, Masks, Networks, and Hosts*

Class	First Octet Binary Range	First Octet Decimal Range	Default or Natural Mask	Number of Networks	Number of Hosts
A	**0**0000001 **0**1111111	1–127*	/8 255.0.0.0	126 $2^7 - 2$	16,777,214 $2^{24} - 2$ $(256 \times 256 \times 256) - 2$
B	**10**000000 **10**111111	128–191	/16 255.255.0.0	16,384 2^{14}	65,534 $2^{16} - 2$ $(256 \times 256) - 2$
C	**110**00000 **110**11111	192–223	/24 255.255.255.0	2,097,152 2^{21}	254 $2^8 - 2$ $(256) - 2$
D	**1110**0000 **1110**1111	224–239	Multicast		
E	**1111**0000 **1111**1111	240–255	Experimental		

* 127.0.0.0/8 denotes loopback addressing

Table 1-4 illustrates the classes of networks. It is essential to recognize the class by the first octet range so that you are familiar with the default or natural mask. When subnetting, you must borrow bits from the host portion; if you know the default mask, the host portion is where the 0 bits are. For example, 10.0.0.1/8 is a Class A address with a default subnet mask of 255.0.0.0 or /8. If you need more networks, you subnet by borrowing the required number of bits from the host octets 2, 3, and 4 contiguously. To determine the number of bits to borrow, use the following formula:

$2^x >=$ the number of subnets you need

Solve for x to know how many bits to borrow. It is not wrong to use the formula $2^x - 2$, but the minus 2 is for the 0 subnet and the all 1s subnet (broadcast), which are certainly valid today. Suppose you need 250 subnets. The formula to solve is $2^x >= 250$. In this example, you borrow 8 bits to give you a new subnet mask of 255.255.0.0 or /16. The class is still a Class A, however, not a Class B.

Default subnet masks, class ranges, networks, hosts, and their associated logical addresses are important practical topics that will help you with troubleshooting. Shortcuts are nice, but the binary really tells all. Check out www.learntosubnet.com for a refresher on subnetting and Chapter 3 of this book for practical application thereof. In the meantime, look at a couple of examples of subnetting and summarization here. Practice makes perfect, and Cisco tests aren't as lenient as others in allowing you to use a calculator. Use Figure 1-5 and Table 1-4 to assist in your calculations.

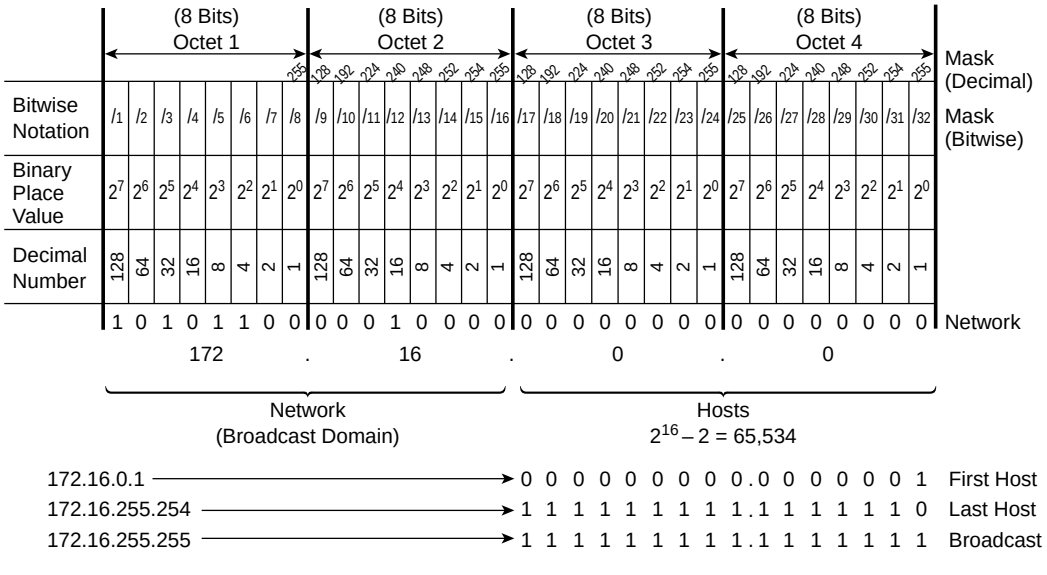
1 How many hosts are on 172.16.0.0/16? What are they?

Answer: 172.16.0.0 is one big network (broadcast domain) with 65,534 hosts ($2^{16} - 2$). The first host is 172.16.0.1. The last host is 172.16.255.254, and the broadcast is 172.16.255.255. The binary representation of this is critical to understanding bits and boundaries. Often it is helpful to write out the hosts in ranges, such as the following:

- 172.16.0.1 through 172.16.0.255
- 172.16.1.0 through 172.16.1.255
- 172.16.2.0 through 172.16.2.255
- ...
- 172.16.255.0 through 172.16.255.254
- 172.16.255.255 (broadcast)

Figure 1-6 illustrates the calculations using binary place values. Notice how the first host is the network address plus one; the broadcast address is the same as turning all bits on (1) for the hosts portion; and the last host is the broadcast address minus one.

Figure 1-6 *Binary Place Values (Question 1 Answer)*



2 How many networks and hosts are on 172.16.0.0/24? What are they?

Answer: 172.16.0.0 is 1 subnet (broadcast domain) with 254 hosts ($2^8 – 2$). The first host for this subnet is 172.16.0.1. The last host for this subnet is 172.16.0.254, and the broadcast address is 172.16.0.255. The binary representation of this is critical to understanding bits and boundaries. Often it is helpful to write out the hosts in ranges, such as the following:

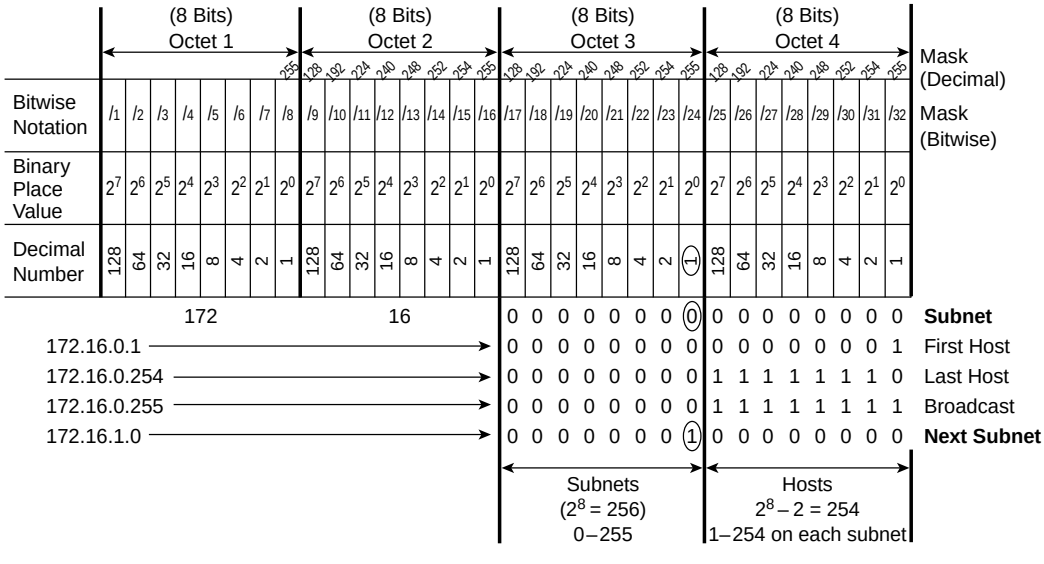
- 172.16.0.1 through 172.16.0.254 (hosts on subnet 172.16.0.0/24)
- 172.16.0.255 (broadcast on subnet 172.16.0.0/24)

This is a Class B address with a default subnet mask of /16. The given mask is /24, which means that 8 host bits were borrowed to provide more networks (subnets). This is subnetting. There are $2^8 = 256$ available subnets in this scenario with 254 hosts ($2^8 – 2$) on each one. These subnets increment by 1 because the lowest 1 (network) bit is in the 1 or 2^0 binary position. The next two subnets are 172.16.1.0/24 and 172.16.2.0/24.

Use Figure 1-7 to verify your calculations and to relate the following general rules:

- The rightmost available host bit is turned on (1) for the first host. All other host bits are off (0).
- The rightmost available host bit is turned off (0) for the last host. All other host bits are on (1).
- All host bits are on (1) for the broadcast address. The broadcast for a subnet is one less than the next subnet.
- Subnets increment by the lowest 1 bit (rightmost bit) in the mask. The subnet increment and the first two subnets are circled in Figure 1-7.

Figure 1-7 *Binary Place Values (Question 2 Answer)*



3 How many networks and hosts are on 172.16.1.4/30? What are they? What is the next available subnet?

Answer: 172.16.1.4 is one subnet (broadcast domain) with two hosts ($2^2 – 2$). The first host for this subnet is 172.16.1.5. The last host for this subnet is 172.16.1.6, and the broadcast address is 172.16.1.7. The binary representation of this is critical to understanding bits and boundaries. Often it is helpful to write out the hosts in ranges, such as the following:

- 172.16.1.5 through 172.16.1.6 (hosts on subnet 172.16.1.4/30)
- 172.16.1.7 (broadcast on subnet 172.16.1.4/30)
- 172.16.1.8/30 (next subnet)

This is a Class B address with a default subnet mask of /16. The given mask is /30, which means that 14 host bits were borrowed to give more networks (subnets). This is subnetting. There are $2^{14} = 16{,}384$ possible subnets in this scenario with 2 hosts ($2^2 – 2$) on each one. The subnet increment is circled in Table 1-5. These subnets increment by 4 because the lowest 1 (network) bit is in the 4 or 2^2 binary position. The next two subnets are 172.16.1.8/30 and 172.16.1.12/30. The shading in Table 1-5 indicates the subnet portion. Only the last octet is shown.

Table 1-5 *Binary Place Values for the Last Octet (Question 3 Answer)*

Subnet	**Subnet**	**Subnet**	**Subnet**	**Subnet**	**Subnet**	**Hosts**	**Hosts**	
/25	**/26**	**/27**	**/28**	**/29**	**/30**	**/31**	**/32**	**Mask (Bitwise)**
128	192	224	240	248	252	254	255	Mask (Decimal)
2^7	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**	**Binary**
128	64	32	16	8	(4)	2	1	Subnet increment
0	0	0	**0**	0	1	0	0	(Subnet) **172.16.1.4**
0	0	0	**0**	0	1	0	1	(First host) 172.16.1.5
0	0	0	**0**	0	1	1	0	(Last host) 172.16.1.6
0	0	0	**0**	0	1	1	1	(Broadcast) 172.16.1.7
0	0	0	**0**	**1**	0	0	0	(Subnet) **172.16.1.8**
0	0	0	**0**	**1**	0	0	1	(First host) 172.16.1.9

Table 1-5 *Binary Place Values for the Last Octet (Question 3 Answer) (Continued)*

Subnet	Subnet	Subnet	Subnet	Subnet	Subnet	Hosts	Hosts	
0	0	0	**0**	**1**	0	1	0	(Last host) 172.16.1.10
0	0	0	**0**	**1**	0	1	1	(Broadcast) 172.16.1.11
0	0	**0**	**0**	**1**	1	0	0	(Subnet) **172.16.1.12**

4 How many networks and hosts are on 10.1.1.0/28? What are they? List the hosts and broadcast address for the next available subnet.

Answer: 10.1.1.0 is 1 subnet (broadcast domain) with 14 hosts ($2^4 – 2$). The first host for this subnet is 10.1.1.1. The last host for this subnet is 10.1.1.14, and the broadcast address is 10.1.1.15. The binary representation of this is critical to understanding bits and boundaries. Often it is helpful to write out the hosts in ranges, such as the following:

- 10.1.1.1 through 10.1.1.14 (hosts on subnet 10.1.1.0/28)
- 10.1.1.15 (broadcast on subnet 10.1.1.0/28)
- 10.1.1.16 (next subnet)
- 10.1.1.17 through 10.1.1.30 (hosts on subnet 10.1.1.16/28)
- 10.1.1.31 (broadcast on subnet 10.1.1.16/28)
- 10.1.1.32 (next subnet)

This is a Class A address with a default subnet mask of /8. The given mask is /28, which means that 20 host bits were borrowed to give more networks. This is subnetting. There are 2^{20} = 1,048,576 possible subnets in this scenario with 14 hosts ($2^4 – 2$) on each one. These subnets increment by 16 because the lowest 1 (network) bit is in the 16 (2^4) binary position. The next two subnets are 10.1.1.16/28 and 10.1.1.32/28. The shading in Table 1-6 indicates the subnet portion. Only the last octet is shown.

NOTE If you have ever taken any of my classes, you know that all 4 octets with all 32 bits get drawn on the board first. Then I write out only the octets where the mask is less than 255 or 8 bits. One student suggested I actually write the last 2 octets on the back of my business card and hand them out to future students.

Table 1-6 *Binary Place Values for the Last Octet (Question 4)*

Subnet	Subnet	Subnet	Subnet	Hosts	Hosts	Hosts	Hosts	
/25	**/26**	**/27**	**/28**	**/29**	**/30**	**/31**	**/32**	**Mask (Bitwise)**
128	192	224	240	248	252	254	255	Mask (Decimal)
2^7	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**	**Binary**
128	64	32	(16)	8	4	2	1	Subnet increment
0	0	0	**0**	0	0	0	0	(Subnet) **10.1.1.0**
0	0	0	**0**	0	0	0	1	(First host) 10.1.1.1
0	0	0	**0**	1	1	1	0	(Last host) 10.1.1.14
0	0	0	**0**	1	1	1	1	(Broadcast) 10.1.1.15
0	0	0	**1**	0	0	0	0	(Subnet) **10.1.1.16**
0	0	0	**1**	0	0	0	1	(First host) 10.1.1.17
0	0	0	**1**	1	1	1	0	(Last host) 10.1.1.30
0	0	0	**1**	1	1	1	1	(Broadcast) 10.1.1.31
0	0	**1**	**0**	0	0	0	0	(Subnet) **10.1.1.32**

5 Assuming you have all point-to-point serial links to assign addresses to and that you are given the network 10.1.1.0/28, can you squeeze any more subnets out of it? If so, how many and what are they? What is this called?

Answer: VLSM is just subnetting again. You move the bit boundary to the right to get more subnets. In this example, the subnet boundary is at /28. Because point-to-point serial links never need more than 2 host addresses, you can borrow out to a /30 or 255.255.255.252 subnet mask. This gives $2^2 = 4$ VLSM subnets (0, 4, 8, and 12) with 2 hosts each. *Caution*: No overlap is allowed with VLSM! If subnet 0 has already been assigned, for instance, you *cannot* subnet that subnet. VLSM is common practice on WAN links, but the routing protocol must support it.

- 10.1.1.0/30 (VLSM subnet)
- 10.1.1.4/30 (VLSM subnet)
- 10.1.1.8/30 (VLSM subnet)
- 10.1.1.12/30 (VLSM subnet)

Table 1-7 illustrates subnet 10.1.1.0/28, its VLSM subnets (0, 4, 8, 12), and its hosts. The first host on VLSM subnet 0 is 10.1.1.1/30, for example, the last host is 10.1.1.2/30, and the broadcast is 10.1.1.3.30. The next VLSM subnet is 10.1.1.4/30. Its first host is 10.1.1.5, the last host is 10.1.1.6/30, and the broadcast is 10.1.1.7/30. The next VLSM subnet is 10.1.1.8/30 and so on. The lighter shading indicates subnets, and the darker shading indicates VLSM subnets.

Table 1-7 *Binary Place Values for the Last Octet (Question 5)*

Subnet	**Subnet**	**Subnet**	**Subnet**	**VLSM Subnet**	**VLSM Subnet**	**Hosts**	**Hosts**	
/25	**/26**	**/27**	**/28**	**/29**	**/30**	**/31**	**/32**	**Mask (Bitwise)**
128	192	224	240	248	252	254	255	Mask (Decimal)
2^7	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**	**Binary**
128	64	32	16	8	(4)	2	1	Subnet increment
0	0	0	0	**0**	**0**	0	0	(Subnet) **10.1.1.0**
0	0	0	0	**0**	**0**	0	1	(First host) 10.1.1.1
0	0	0	0	**0**	**0**	1	0	(Last host) 10.1.1.2
0	0	0	0	**0**	**0**	1	1	(Broadcast) 10.1.1.3
0	0	0	0	**0**	**1**	0	0	(Subnet) **10.1.1.4**
0	0	0	0	**0**	**1**	0	1	(First host) 10.1.1.5
0	0	0	0	**0**	**1**	1	0	(Last host) 10.1.1.6
0	0	0	0	**0**	**1**	1	1	(Broadcast) 10.1.1.7
0	0	0	0	**1**	**0**	0	0	(Subnet) **10.1.1.8**

Table 1-7 *Binary Place Values for the Last Octet (Question 5) (Continued)*

Subnet	Subnet	Subnet	Subnet	VLSM Subnet	VLSM Subnet	Hosts	Hosts	
0	0	0	**0**	**1**	0	0	1	(First host) 10.1.1.9
0	0	0	**0**	**1**	0	1	0	(Last host) 10.1.1.10
0	0	0	**0**	**1**	0	1	1	(Broadcast) 10.1.1.11
0	0	0	**0**	**1**	1	0	0	(Subnet) **10.1.1.12**
0	0	0	**0**	**1**	1	0	1	(First host) 10.1.1.13
0	0	0	**0**	**1**	1	1	0	(Last host) 10.1.1.14
0	0	0	**0**	**0**	0	1	1	(Broadcast) 10.1.1.15

6 Summarize the following into the fewest number of statements possible.

- 192.168.168.0/24
- 192.168.169.0/24
- 192.168.170.0/24
- 192.168.171.0/24
- 192.168.172.0/24
- 192.168.173.0/24
- 192.168.174.0/24
- 192.168.175.0/24

Answer: 192.168.168.0/21

Table 1-8 illustrates the third octet in binary so that you can easily identify the best pattern as to summarize in the fewest number of statements. As the darker shading shows, all bits match from bit /1 through /21; therefore you can capture eight lines into one. Although not as efficient,

you can summarize using two statements (192.168.168.0/22 and 192.168.172.0/22) or four statements (192.168.168.0/23, 192.168.170.0/23, 192.168.172.0/23, and 192.168.174.0/23).

Table 1-8 *Summarization (Question 6 Answer)*

Mask (Bitwise)	**/17**	**/18**	**/19**	**/20**	**/21**	**/22**	**/23**	**/24**
Mask (Decimal)	128	192	224	240	248	252	254	255
Binary Place Value	**2^7**	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
Decimal Number	128	64	32	16	8	4	2	1
168	1	0	1	0	1	0	0	0
169	1	0	1	0	1	0	0	1
170	1	0	1	0	1	0	1	0
171	1	0	1	0	1	0	1	1
172	1	0	1	0	1	1	0	0
173	1	0	1	0	1	1	0	1
174	1	0	1	0	1	1	1	0
175	1	0	1	0	1	1	1	1

NOTE If you think you need more review and practical application of subnetting, see Chapter 3. Additionally, review the *CCNA Practical Studies* and *CCNP Practical Studies: Routing* titles as well as the www.learntosubnet.com website.

So what else happens at Layer 3? IP is responsible for delivery and fragmentation at Layer 3 and it has various helper protocols to accomplish these tasks. Internet Control Message Protocol (ICMP) is for status and error reporting. Address Resolution Protocol (ARP) resolves an IP address to MAC on a broadcast-based network such as a LAN. Network cards and router interfaces have burned-in addresses (BIAs) for the MAC. By the way, ARP is not needed for IPX addressing because the MAC is the host address on the wire in Novell. ARP is also not required on point-to-point media either. ARP is initiated with a local broadcast, but the reply is a unicast. Think of it this way: I have the IP, but I need the MAC. You experiment and learn a little more about ARP in Chapter 3 in the section titled “Protocols and Packets.” Until then, think about what would happen in the following circumstances:

- **Local ARP request**—If you were to ping a host on a local network and look at the ARP cache (**arp –a**), what would you expect to see?
- **Remote ARP request**—If you were to ping a host on a different network and look at the ARP cache, what would you expect to see?

I would expect to see the host MAC address for the destination host in the ARP table for a local ARP request. If I were to ping a host on a different network and look at the ARP cache, however, I would expect to see the MAC address associated with the local interface of the router (default gateway). Learning to follow the ARP is quite beneficial in troubleshooting.

Now turn your attention to RARP, which sounds like something out of the TV show *Mork & Mindy*. RARP is Reverse Address Resolution Protocol. First there was RARP, then BOOTP, and now DHCP too. You explore DHCP in one of the later labs.

NOTE The Transport Layer is often referred to as the host-to-host layer and the Network Layer as the Internet layer. If you can't ping the destination host but can ping another local host and your default gateway, the problem may be in the path from the source to the destination. Use **traceroute** (**tracert**) to help you determine exactly where the problem is.

Layer 2: The Data Link Layer

Layer 2, the Data Link Layer, is where bridges and switches operate. Bridges and switches are covered in much more detail in Chapters 5, 6, and 7.

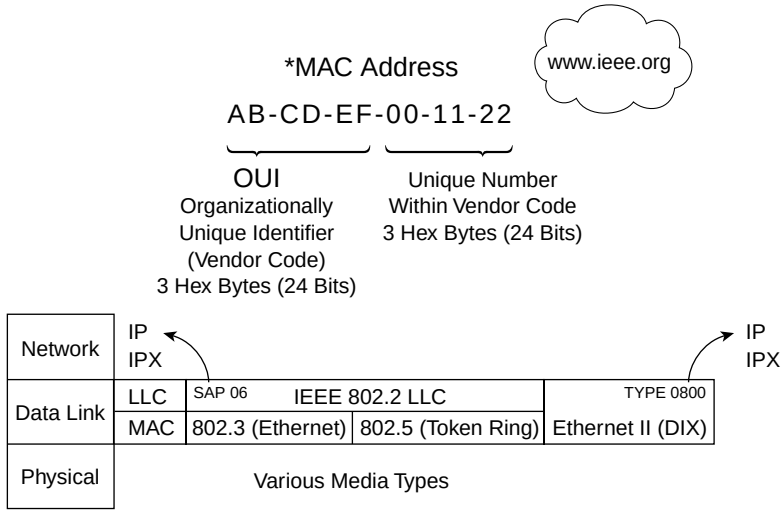
The IEEE Layer 2-defined sublayers include Logical Link Control (LLC) and Media Access Control (MAC) as represented in Figure 1-8. LLC is responsible for synchronization and connection services via Service Access Points (SAPs) to the upper layers. MAC is responsible for physical (hardware) addressing, logical topology, and shared media access.

Digital Intel Xerox (DIX) Ethernet II uses a Type field to point to the Layer 3 protocol (0800 is IP), but IEEE 802.3 Ethernet uses a valid length field and 802.2 LLC SAPs to link to the Layer 3 protocol. SAPs are pointers or software controls to manage multiple Layer 3 protocols. For example, the hex SAP value of 06 is a link to IP; the hex SAP value of e0 is a link to IPX. Table 1-9 provides more detail on LLC types.

Table 1-9 *LLC Types*

LLC Type	Connection	Reliability	Description
LLC Type 1	Connectionless	Unacknowledged	Does not confirm data transfers Used in LANs
LLC Type 2	Connection-oriented	Acknowledged	Establishes logical connection and confirms data upon receipt Used in IBM SNA
LLC Type 3	Connectionless	Acknowledged	Confirms data upon receipt but does not establish logical connection Used in factory automation

Figure 1-8 *LLC and MAC*



*48 Bits/12 Hex Characters/6 Hex Bytes

IEEE-assigned MAC addresses are often referred to as hardware addresses, Layer 2 addresses, BIAs, or physical addresses that are coded into the network card or interface on a router. A 3-byte IEEE-assigned Organizationally Unique Identifier (OUI) is used to generate universal MAC addresses for vendors. Table 1-10 offers some examples from Cisco, 3Com, Intel, DEC, and Madge. This is not by any means a comprehensive list; see www.ieee.org for more details. Download them all in a text file from standards.ieee.org/regauth/oui/oui.txt.

Table 1-10 *IEEE-Assigned MAC Addresses*

Vendor	Identification (OUI)
Cisco	00-00-0C
	00-01-42
	00-01-43
	00-01-63
	00-01-64
	00-E0-F7
	00-E0-F9
	00-E0-FE
	08-00-58

Table 1-10 *IEEE-Assigned MAC Addresses (Continued)*

Vendor	Identification (OUI)
3Com	00-01-02 00-01-03 02-C0-8C 08-00-4E
Intel	00-01-2A 00-02-B3 00-AA-01 00-AA-02
DEC	AA-00-00 AA-00-01 AA-00-02 AA-00-03 AA-00-04
Madge	00-00-6F 00-00-C1 00-80-E9

The MAC sublayer is for taking turns on the wire as well as error checking and addressing. It is like the traffic cop on the medium. Table 1-11 provides a brief review of access methods.

Table 1-11 *Access Methods*

Access Method	Description	Examples
CSMA/CD*	Polite conversation at a cocktail party. You listen (carrier sense) and if you and another person talk simultaneously (multiple access), you both wait a random amount of time and talk again.	Ethernet II IEEE Ethernet 802.3
CSMA/CA**	Collision avoidance Signal the intent to transmit	AppleTalk
Token Passing	Must hold the token to talk	IBM Token Ring IEEE 802.5 Token Ring ANSI X3T9.5 FDDI

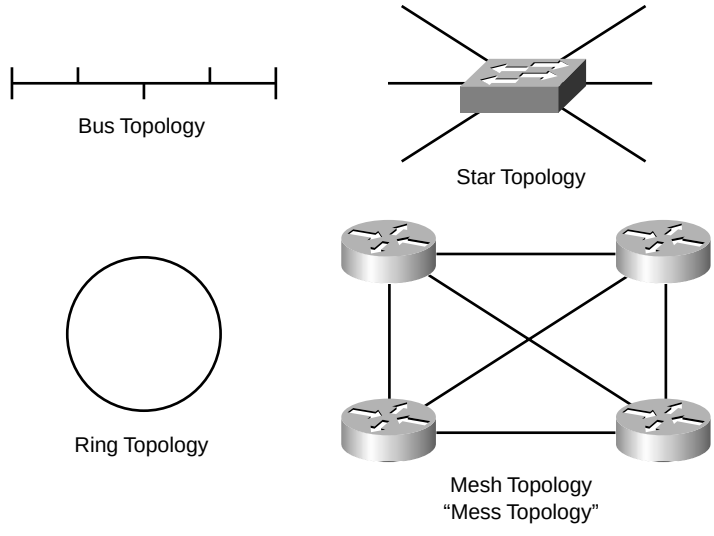
*CSMA/CD = Carrier sense multiple access with collision detection

**CSMA/CA = Carrier sense multiple access with collision avoidance

Ethernet, whether a physical star or bus, uses the carrier sense multiple access collision detect (CSMA/CD) logical access method because logically it acts like a bus. Token Ring and FDDI use a token-passing access method in a logical ring topology over a physical star or ring. Collisions do not occur in Token Ring because a device must have the token to talk. Access methods are nothing you and I set; they are a function of the network architecture, such as Ethernet or Token Ring, that allows devices to share the media.

Topologies encompass the Data Link (logical) and Physical Layers. Ethernet is typically a physical star, logical bus; whereas Token Ring is a physical star, logical ring topology. (See Figure 1-9.)

Figure 1-9 *Topology*



The PDU for Layer 2 is frames. Control bits mark the beginning and end of frames just as picture frames mark the edges of a picture. Layer 2 is the LAN/WAN layer in many respects. You have seen how it allows different devices to take turns on the media and how the network works (logical topologies). But how is the data actually packaged at Layer 2? Figures 1-10 (Ethernet), 1-11 (Token Ring), and 1-12 (FDDI) show some basic frame format (encapsulation) examples. Use the following legend for the abbreviations:

PRE = Preamble

DA = Destination address

SA = Source address

T/L = Type or length

FCS = Frame check sequence

DEL = Delimiter

FS = Frame status

Figure 1-10 *Ethernet Frame Format*

Ethernet Frame Format					
PRE	DA	SA	T/L	DATA	FCS

Figure 1-11 *Token Ring Frame Format*

Token Ring Frame Format							
START DEL	ACCESS CTRL	FRAME CTRL	DA	SA	DATA	FCS	END DEL

Figure 1-12 *FDDI Frame Format*

FDDI Format								
PRE	START DEL	FRAME CTRL	DA	SA	DATA	FCS	END DEL	FS

NOTE Ethernet dominates typical LAN topologies today and is further discussed in Chapter 5.

While I discuss other Layer 2 activities, think back to the earlier analogy of the waiter who took your order. Did you get the big-endian cheesecake or the little-endian cheesecake…ekaceseehc for desert? *Big-endian* systems, such as IBM, RISC, and Motorola processors, read left to right, or high-order to low-order bits and bytes. *Little-endian* systems, such as Intel processors and DEC Alphas, read right to left, or low-order to high-order bits and bytes. Likewise, Ethernet is canonical and Token Ring is noncanonical. Use Table 1-12 to review the hex calculations (base 16) used in Figure 1-13. Also remember that A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15 in hexadecimal.

Table 1-12 *Hex Place Values*

2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1

NOTE According to www.whatis.com, "Big-endian and little-endian derive from Jonathan Swift's *Gulliver's Travels* in which the Big Endians were a political faction that broke their eggs at the large end ("the primitive way") and rebelled against the Lilliputian King who required his subjects (the Little Endians) to break their eggs at the small end."

Figure 1-13 *Canonical Names*

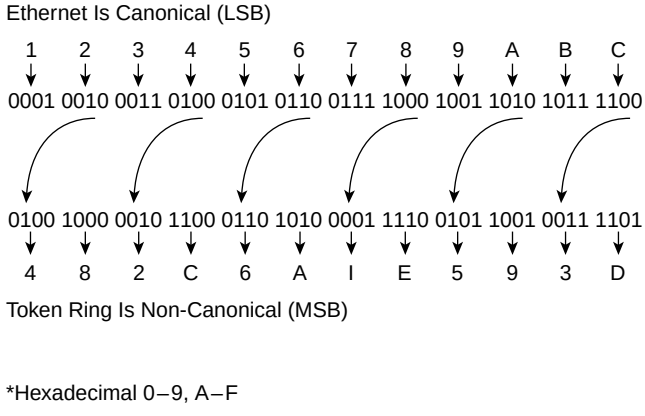


Figure 1-13 illustrates that Ethernet is canonical, and the least significant bit (LSB) is read first. In contrast, Token Ring is noncanonical, and the most significant bit (MSB) is read first. The picture also is a great review of binary-to-hex conversion, but most people use calculators for that anyhow.

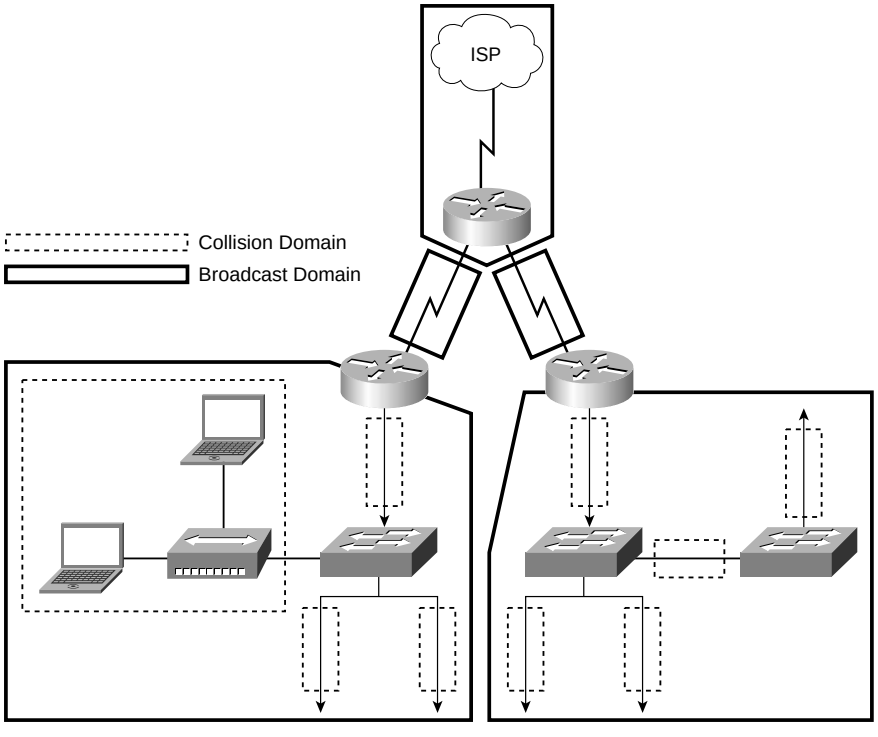
NOTE Cisco offers a tool on their website that enables you to automatically convert canonical to noncanonical and vice versa. Search for the "bitswap tool" on www.cisco.com to see for yourself.

Layer 1: The Physical Layer

Layer 1, the Physical Layer, is all about the shape of the network. How things work is more a matter of Layer 2 logical topologies, but Layer 1 is concerned with physical topologies such as star, bus, ring, or mesh. Ethernet is typically a physical star, logical bus (10BASE-T/100BASE-T/1000BASE-T). Token Ring and FDDI are typically wired as a physical star, logical ring. Without a concentrator, FDDI is truly a physical ring topology.

Hubs are Layer 1 devices that repeat or regenerate the signal to allow connectivity and assist with attenuation issues. Layer 1 devices just extend the network; they do no filtering. Hubs spit bits, including collisions and broadcasts. Switches (Layer 2 devices) assist with collisions and make filtering decisions based on physical addresses. Routers (Layer 3 devices or VLANs) assist with collisions and broadcasts; they make filtering decisions based on logical addresses. A collision domain is a separate CSMA/CD network in Ethernet or a separate ring in Token Ring where devices are taking turns for use of the wire. Collision domains exist between two Layer 2 devices and for each user-dedicated port on a switch. Broadcast domains are subnets in TCP/IP. They exist between routers and for each Layer 3 interface. Figure 1-14 shows collision domains and broadcast domains. Technically there are no collisions on the serial links, so you shouldn't count them as collision domains as you examine Figure 1-14.

Figure 1-14 *Collision and Broadcast Domains*



*The broadcast domains assume one VLAN per switch here.

Hubs with any intelligence at all (such as multiple speeds or network management capabilities) are not just Layer 1 devices; they move up the OSI stack according to the built-in intelligence. Most people know hub and repeater as the same thing; however, repeater is an IEEE term. When selecting a hub or any connectivity device for that matter, consider network architecture (such as Ethernet or Token Ring, or FDDI; port density and speed; management; cable types; and modularity).

Physical star topologies are easier to troubleshoot but take more cable. Failure of one device doesn't usually interfere with another. If a user calls and says the network is down, you can start your troubleshooting between the user and the hub or switch port connection. Switches are typically used rather than hubs today because the price per port is declining all the time. The big advantage is that each port is a separate collision domain; whereas all hub ports are in the same collision domain. After all, you don't need too many packet fights. Cat5E unshielded twisted-pair cable is still by far the most common.

Frozen yellow garden hose and vampire taps come to mind when I think of the old 10BASE5 Ethernet backbones using RG-8 or RG-11 standard Ethernet coax cable rated at 50-ohms impedance. 10BASE2 could be a bus or star depending on whether hubs are in the picture. Think of a two-pole clothesline setup on which you hang your clothes out to dry. The poles at each end are the terminators connecting one of the RG-58 family of cables. Although inexpensive to implement, the disadvantages include heavy traffic patterns on the bus and tedious troubleshooting, to say the least, unless you have a time domain reflectometer (TDR) to help you find the fault within the coax cable. Without the proper test equipment, carrying the terminator from one station to another is about as exciting as relocating your clothesline poles. Today the backbone is normally twisted-pair or fiber cable, so these issues are not as relevant; be aware, however, that many certification tests think you should know the specifications.

FDDI and Token Ring are typically a physical star, logical ring topology. The *active monitor* monitors the token circulating around the ring. Problem isolation and network reconfiguration are issues. It used to be that Token Ring wiring was all shielded twisted-pair Type 1, 2, 3, 6, and so on with hermaphroditic connectors, but now it is primarily UTP with RJ-45 connectors.

In meshes, fault tolerance may be maximized, but from the troubleshooting perspective, it is often a mess. I refer to the mesh topology as the *mess topology,* and it is more often used for backup links on the WAN (for example, ISDN backup for Frame Relay links).

Installation, reconfiguration, and cost normally lead us to some hybrid of the preceding topologies in a practical environment.

The PDU for Layer 1 is bits. Remember that the Physical Layer is responsible for transmitting bits (0s and 1s) and coordinating rules for transmitting (Tx) and receiving (Rx) them. Mechanical, electrical, optical, and signaling are among the many specifications at Layer 1. Layer 1 is a good place for trouble to shoot you if you are not careful.

NOTE I had to laugh at myself the other day when I connected my PC to the hub in the front of the classroom. I had lights on the hub, but not on the PC dongle. "This is a little strange," I thought to myself. I thought maybe I had the wrong cable type, because I didn't know what was on the other side of the wall. I thought I would eliminate the hub and plug directly into the wall. Neither a straight-through nor a crossover cable worked. I even tried different cables. Finally I picked up the laptop PC, took it to another room, and connected it fine. So I returned to the classroom, connected up again, but still no dongle light. I decided to just test things from the laptop PC anyhow and found I could ping and use the Internet, which is all I originally wanted to do. The funny part was that I had put a red mark on this dongle to trick a student who needed a little more of a challenge. Refer back to this scenario as I discuss using models and methods to troubleshoot by the layers.

Do you work with the physical aspects or is that done for you? (I have thrown brooms through the ceiling and used slingshots to get the cable a little farther down the hall when I didn't have any other tools at hand.) Because you are in a physical mindset at the moment, take a minute to look at a list of what uses RJ-45 (see Table 1-13). Only the active pins are displayed.

Table 1-13 *What Uses RJ-45?*

What Uses RJ-45?	Active Pins
10BASE-T Ethernet	1–2 3–6
100BASE-TX UTP (2 pair Category 5)*	1–2 3–6
Token Ring	4–5 3–6
Console cable	All pins rolled 1–8 2–7 3–6 4–5
T-1	1–2 4–5
ISDN U (North America; *U* for *unpowered*)	4–5 single pair
ISDN S/T	1–2(pwr) 4–5(data) 7–8(pwr)

*Although not commonly used, 100BASE-T4 uses 4 pair of Category 3, 4, or 5 cabling.

Figure 1-15 shows the RJ-45 connector. When you point the clip toward the floor, pin 1 is on the left and pin 8 is on the right. Compare it to the smaller RJ-11 connector.

Figure 1-15 *RJ-11 and RJ-45 Connectors*

Back in the mid-1980s, companies were concerned with cabling standards in particular. EIA/TIA has definitely permeated the cabling industry, particularly with EIA/TIA 568, and has very high recognition among users and vendors alike. Various committees have developed cabling standards and continue to provide updates with Technical Service Bulletins (TSBs) as the industry evolves. 568A and 568B are technically identical, as you can verify in Table 1-14. 568B is very widespread because it is basically the same as AT&T 258A; however, 568A allows two pairs for voice to make it a little more compatible in the telco environment.

Table 1-14 *568A and 568B Standards*

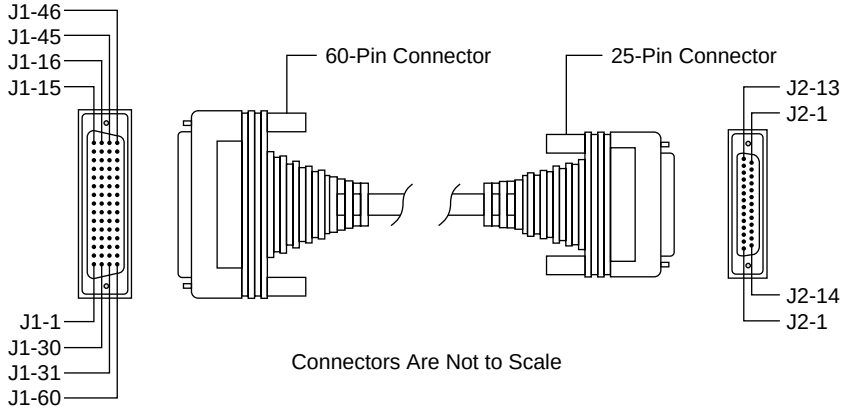
568A (EIA/TIA Where Orange and Green Are Reversed to Be More Compatible with Telco)	568B (The Old AT&T Standard That Is Very Widespread Today)
Pin	Pin
1 white/green (Rx+)	1 white/orange (Tx+)
2 green (Rx–)	2 orange (Tx–)
3 white/orange (Tx+)	3 white/green (Rx+)
6 orange (Tx–)	6 green (Rx–)

NOTE Although only one pair is used for Tx and one pair for Rx, the RJ-45 connector, which holds four pair (eight wires) is standard. Compare it to the RJ-11 connector back in Figure 1-15, which only physically holds two pair (four wires).

Besides the connectors and the pinouts, the wire thickness varies too according to the American Wire Gauge (AWG). For example, one-pair UTP 16 AWG speaker wire for my outside BOSE speakers is much larger than the four-pair UTP 24-gauge running my network.

Figure 1-16 shows a DB-60 to DB-25 serial cable used for WAN connectivity.

Figure 1-16 *EIA/TIA-232 Cable Assembly*



Part II, "Supporting IP and IPX," and Part III, "Supporting Ethernet, Switches, and VLANs," of this book discuss the Physical Layer and Data Link Layer as they relate to LANs/WANs in more detail. In addition, you can check out the following cable sites on your own:

- www.cisco.com
- www.belden.com
- www.belkin.com
- www.stonewallcable.com
- www.amp.com

Now look at some practical application of the Physical Layer. Do you know when to use a straight-through cable compared to a crossover cable? Perhaps a better question is what is a crossover cable? A straight-through cable is wired pin 1 to 1, 2 to 2, 3 to 3, and 6 to 6. A crossover is 1 to 3 and 2 to 6; it crosses between active pairs. Generally speaking, *unlike devices* require a *straight-through cable*, whereas *like devices* require a *crossover cable*. Repeat this rule to yourself as you review Table 1-15. As with any rule, exceptions apply. Therefore, check the cable documentation that comes with your switch or router. For example, a hub may have an uplink port and when in the normal position it requires a crossover cable to connect two devices together. When in the uplink position, the cross is already performed in the device hardware and a straight-through cable is appropriate. Many of the Cisco switch ports are designated with an X above the port or a media dependent interface (MDI/MDI-X) toggle and some are not. Connecting two devices with Xs normally means they are like; therefore you need which kind of cable? Check your answer in Table 1-15.

Table 1-15 *Do You Need a Straight-Through or Crossover Cable?*

Straight-Through (Unlike Devices)	Crossover (Like Devices)	Rollover	T1-Crossover
1-1	1-3	1-8	1-4
2-2	2-6	2-7	
3-3	3-1	3-6	
4-4	4-4	4-5	2-5
5-5	5-5	5-4	
6-6	6-2	6-3	
7-7	7-7	7-2	
8-8	8-8	8-1	
PC to hub	PC to PC (PC to server)	Console cable	T1 - RJ-45 jack
PC to switch	Hub to hub		
Switch to router	Switch to switch		
	Hub to switch*		
	Router to router		
	PC to router*		

*Doesn't follow the general rule of like devices use crossover cable and unlike use straight-through cable. The devices marked with an asterisk require a crossover.

The examples marked with an asterisk are exceptions to the general rule of like devices needing a crossover cable and unlike requiring a straight-through cable. If you draw a line between Layer 2 and Layer 3, however, any device on the same side of the line uses crossover cables.

NOTE I think of hubs and switches as Access Layer devices; because you use them to connect users, in the cabling respect they are the same. I think of PCs and routers as being the same for cabling purposes because both can route using routing protocols.

Wireless media is hot these days and is going to get hotter. It is great for places where wires are not possible (when you can't dig up the street because you don't have the right-of-way, for instance, or over a body of water where you choose not to lay cable or cable is just not feasible). It is becoming conveniently popular in schools, universities, and homes. Some examples follow. Infrared technologies enable you to transfer files or print as easily as you flip TV channels. Spread-spectrum radio is a cost-effective way to divide frequencies into channels instead of leasing lines from the service providers. Encrypted full-duplex data is carried at a fraction of

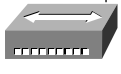
the cost. Cellular digital packet data (CDPD) uses the network for data when not used for voice. Microwave is still very widespread. Take a trip to Maryland's NASA Goddard Space Flight Center sometime or check out the towers at Chincoteague Island, Virginia. New cars are coming out with what rental cars have had for some time; global positioning systems (GPSs) are more popular than ever. If you are out on a boat, your latitude and longitude location is pretty significant to your whereabouts on the bay.

Table 1-16 provides a concise yet comprehensive review of the OSI model.

Table 1-16 *The OSI Reference Model*

Again	Application (Layer 7)	Messages, data, packets (User interface) (Services)	Telnet, NFS, FTP/ TFTP, HTTP, DNS, X.400, X.500, *RIP, *BGP, *DHCP	Service advertisement Service use Name resolution (DNS)	File, print, message, application, database, user interface, file transfer, e-mail
People	Presentation (Layer 6)	Messages, data packets (Translator)	ASCII, EBCIDIC, JPEG, MIDI, MPEG	Translation, encryption, compression	Bit order/byte order, character codes, file syntax, public key/ private key
Support	Session (Layer 5)	Messages, data packets (Operator/dialog)	NetBIOS, Sockets, RPC, LDAP, drive mappings	Dialog Session administration	Simplex, half-duplex, full-duplex connection establishment/data transfer
Threaten	Transport (Layer 4)	Datagrams, segments, packets (Certified mail)	*OSPF, *IGRP, *EIGRP, SPX, TCP, UDP	Addressing, sequencing Connection services Disassembly/re-assembly Delivery/ acknowledgment	Segment sequencing Error/flow control (end-to-end) Guaranteed delivery Hides lower layer intricacies from upper layers

Table 1-16 *The OSI Reference Model (Continued)*

N o t	Network (Layer 3) L3 Switch  Router 	Datagrams, packets (Path determination) (Routing)	IP, *ARP, RARP, ICMP, IGMP	Logical addressing Address resolution (ARP) Switching, sequencing Route discovery/ selection Connection services Gateway services	Unique IP/IPX (internal network number) Packet/message/circuit Distance vector/link state Static/dynamic Flow/error/sequence control Network Layer translation
D o	Data Link (Layer 2)  L2 Switch  N I C	Frames (Carpenter/framer) (Data packaging) (Encapsulation)	Ethernet, Token Ring, FDDI, Frame Relay, HDLC, SDLC, PPP, ISDN, LAPD	LLC sublayer Synchronization Connection services Logical topology Media access Physical addressing MAC sublayer	Logical link control Asynchronous/ synchronous/ isochronous Flow/error control Organizes 0s and 1s into frames Media Access Control Bus/ring Contention/token passing/polling MAC address (physical device address)
P l e a s e	Physical (Layer 1) Hub 	Bits (0s and 1s) (Coordinate rules for bit transmission)	UTP/Cat5E, HSSI, RJ-45, coax, fiber, wireless	Connection types Physical topology Digital/analog signaling Bit synchronization Bandwidth use Multiplexing	Point-to-point/ multipoint Cable layout (bus,ring, star, mesh, cellular) Current state/transition Asynchronous/ synchronous Baseband (TDM)/ broadband (FDM)

*Protocols and applications are written to perform functions. Analyze the layers by looking at protocol analyzer traces. Routing in general is discussed in more detail with regard to Layer 3 (although many ride on TCP/UDP or contain their own reliability mechanisms).

As you work through this book, you will encounter more detailed information and investigate specific troubleshooting targets. At all times, remember that although it is certainly helpful to understand how things work when you are shooting trouble, a methodical approach to troubleshooting is actually more important.

Troubleshooting by Layers

You must train yourself to systematically analyze, resolve, and escalate problems. Troubleshooting by OSI layers is certainly one way to accomplish this. The OSI layers are built and stacked for a reason. For troubleshooting, start at the Physical Layer and work your way up to the Application Layer. A layer problem will lead you to a box and a solution. It is pretty frustrating to just compare what works to what doesn't (the swap-til-you-drop approach), especially when you don't have anything left to swap. Use Table 1-17 to help you troubleshoot by layers.

Table 1-17 *Troubleshooting by Layers*

OSI Layer Number	OSI Layer Name	Basic Troubleshooting
7	Application	Software problem in end system
6	Presentation	Software problem in end system
5	Session	Software problem in end system Host name (Sockets) or NetBIOS name issue
4	Transport	Software problem in end system Cisco/UNIX Traceroute tests up to L4
3	Network	Ping tests up to L3 Microsoft Tracert tests up to L3
2	Data Link	Ping tests up to and through L3
1	Physical	Ping tests up to and through L3

The reality of it all is that the OSI layers are a good approach to discussing networking and internetworking technologies and provide a very good foundation from which to troubleshoot. Be aware, however, that they do not necessarily answer all interoperability issues. As you can see, many industry standards and protocols exist, and obviously there is a lot more to know.

NOTE Perhaps the ISO should have included a Layer 0 for Power and Layers 8, 9, and 10 for Finance, Politics, and Religion. Should I dare say lowest bid wins again, many decisions are quite political in nature, and the methodology (religion) is because we have always done it that way? Although these layers are not part of the ISO specifications, they do appear to be part of most practical environments (whether anyone actually admits it or not).

Many internetworking topics can be examined by reviewing the technical details of the OSI model. I have tried to give you a taste of them in this chapter and to introduce the importance of troubleshooting by layers. My OSI model examples have purposely been IP-related due to the practical application of the book, but they certainly didn't have to be. I could have just as easily used another protocol stack.

The DoD TCP/IP Suite

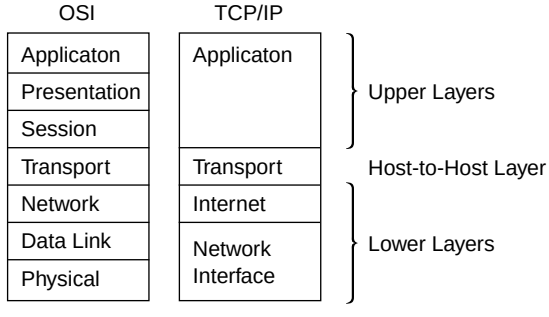
Other industry standard models, such as the DoD TCP/IP suite, provide a way to take a systematic approach to troubleshooting. Table 1-18 compares the DoD TCP/IP suite with the OSI model.

Table 1-18 *Comparing the ISO's OSI Model to the DoD's TCP/IP Suite*

OSI Layer Number	OSI Layer Name	PDU	DoD TCP/IP Suite
7	Application	Messages	Application
6	Presentation	Messages	
5	Session	Messages	
4	Transport	Segments (TCP) Datagrams (UDP)	Transport Host-to-host
3	Network	Packets/datagrams	Internet
2	Data Link	Frames	Data Link
1	Physical	Bits	Physical

TCP/IP came from ARPANET. It is old, but definitely not outdated. Prior to the acceptance of the TCP/IP suite, single-vendor solutions, such as IBM SNA and Novell IPX, prevailed. TCP/IP allows for heterogeneous operating systems, platforms, and hardware, hence open systems. Many vendors and resources discuss the TCP/IP suite using four layers (see Figure 1-17); however, the DoD standards call for five layers, dividing the Network Interface Layer into separate Physical and Link Layers (see Table 1-18).

Figure 1-17 *Upper, Host-to-Host, and Lower Layers*

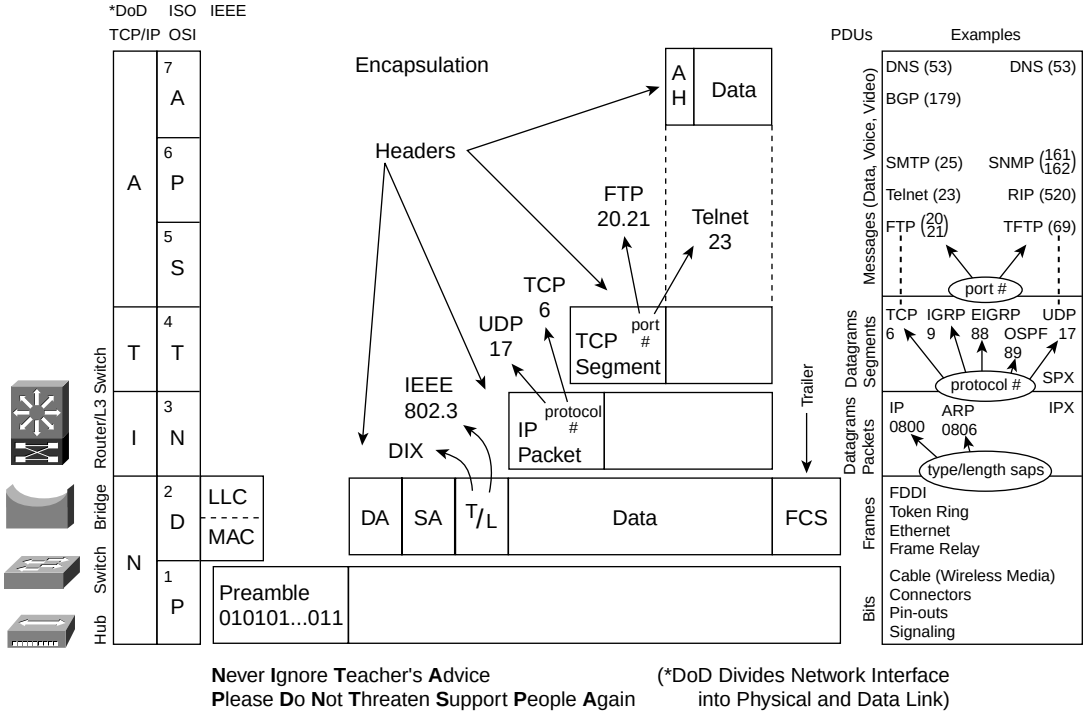


This model gives you more mnemonics from the bottom up: **N**ever **I**gnore **T**eacher's **A**dvice:

- **A**pplication
- **T**ransport
- **I**nternet
- **N**etwork Interface

Industry models enable you to take a layered approach to understanding technology and troubleshooting it. Cisco even recommends a layered approach to design. The *Access Layer* (user layer) is typically comprised of low-end switches operating at 10/100 Mbps. The *Distribution Layer* (decision-making layer) is typically comprised of 100-Mbps routers, whereas the *Core Layer* is typically a 100/1000-Mbps backbone of multilayer switches to switch packets as fast as possible from the source to the destination network. Knowing the layers and your network is a big part of troubleshooting. Compare the models once again in Figure 1-18 before you move on to the Cisco approach to troubleshooting.

Figure 1-18 *Compare the Models*



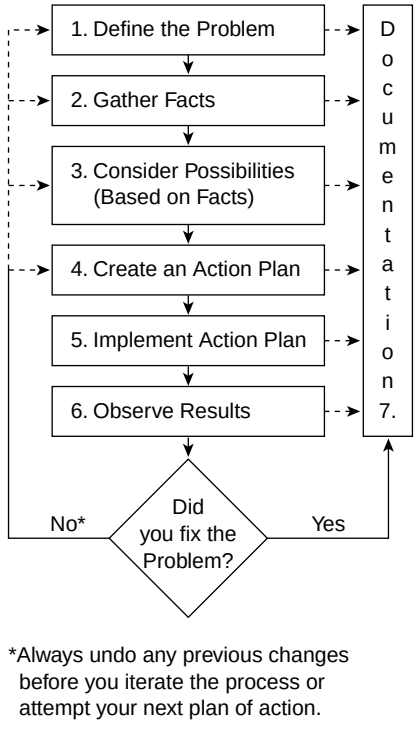
As you work through the scenarios and Trouble Tickets throughout the rest of the book, and particularly when you tackle problems in real life, it will become more and more apparent that you need an understanding of standards and protocols as well as systematic models and methods to

effectively support your LANs and WANs. The OSI model and TCP/IP suite certainly offer a layered approach to understanding and troubleshooting complex internetworks. However, there are many other approaches. As a matter of fact, Cisco offers a systematic approach of their own. Take the time to review the Cisco troubleshooting model. You can find it on the Documentation CD-ROM or search at Cisco.com for "Internetwork Troubleshooting Guide, Troubleshooting Overview" to find the Cisco approach to troubleshooting.

The Cisco Troubleshooting Approach

The Cisco approach to shooting trouble can be an effective way to troubleshoot, particularly if you don't already have a working method. This method is *critical* to the CCNP Support exam objectives, so you should study Figure 1-19 very carefully. From a practical viewpoint, you do not need to change to the Cisco strategy if the troubleshooting method/model you have works.

Figure 1-19 *The Cisco Approach to Troubleshooting*



The Cisco troubleshooting approach includes seven steps for resolving problems. (See Figure 1-19.) First you define the problem. Next you gather facts and then consider possibilities based on those facts. This is another way of saying evaluate your alternatives. Create an action plan, which may be kind of an if-then-else action plan. Implement the plan, observe the results, and

verify that you and everyone involved thinks you fixed the problem at hand. If you did not fix the problem, be sure to undo any previous changes before you continue with the next plan of action. If you have exhausted all your if-then-else courses of action in your action plan, you may have to start at the top of the Cisco ladder to ensure you defined the right problem. Cisco suggests step seven as the place to document the solution. However, documentation is very important at each step in this process (as indicated on Figure 1-19). I define some examples at each step in the following list. Use them to apply this method to your own environment and to work through the chapter Trouble Ticket.

1. Define the problem.

   For example, an end user calls in and reports the network is down. You should identify the symptoms, isolate the problem, and document the findings.

2. Gather facts.

   Perform pertinent tests. For example, can you ping or trace? From the PC? From the router? Can you use another application such as HTTP or FTP? Can you telnet to the port?

   Find out when it last worked, if it ever worked, and whether it is a recurring problem. What has changed since it last worked?

   Determine how many people/devices are affected. Is it a local or remote issue? If you did a network baseline up front, you have some comparison information. See the "Practical Troubleshooting" section for more on baselining and documentation.

   Work as a team; collaborate with other engineers and colleagues. Contact users, network administrators, managers, and other key people.

   Use your tools. For example, network management systems (NMS) such as CiscoWorks or Cisco Info Center (CIC) enable you to map your network and track changes. Take advantage of protocol analyzer captures from programs such as Sniffer or NetMon. Monitor syslog or other logs. Interpret Cisco **show** and **debug** output and research Cisco.com and other sites and tools. Time and date stamps are valuable for troubleshooting; Network Time Protocol (NTP) is free, so you should take advantage of it and show the clock while gathering facts. Answer the questions that help you identify which tool to use; remember that different tools operate at different layers.

NOTE Chapter 2, "What's in Your Tool Bag," covers tools relevant to the CCNP Support exam objectives and the practical world of internetworking.

   The most important thing about this step is to determine what the "real" and "full" problem is. If you open Trouble Tickets based on user complaints, remember to consider the user's description of the problem in light of the user's technical expertise and understanding.

Document the findings.

3. Consider possibilities (based on facts).

   Brainstorm and narrow down the possibilities so that you can focus on what is relevant. Find out whether anyone else has tried to fix the problem. Just as you did in the fact-gathering step, work with the people on your team, not against them.

   Document the findings. More times than one I have been the victim of my own circumstance because I did not document the relevant possibilities or make a checklist of what had and had not been completed. Documentation should be so good that someone should be able to immediately pick up where you left off in a Trouble Ticket.

4. Create an action plan.

   Determine what has to be done to fix the problem. Take a divide-and-conquer approach. List the most likely cause first and plan to change *only* one variable at a time so that you know what change has what impact. Identify any special resource requirements. Prioritize possibilities so that you start with the most likely solution first. Who or what will be affected as a result of your action plan?

   Document the findings.

5. Implement action plan.

   Follow a step-by-step approach to carrying out the action plan. Change only *one thing at a time* and measure the results; *always* maintain a fallback plan. Make sure you don't make things worse or add additional problems. Documenting each step of the way and following your plan systematically and meticulously will assist with this.

   Limit the impact on others as required. For example, shops that work around the clock (7×24) are more likely to have a more stringent change process.

   Call or e-mail TAC if you can't resolve a problem after putting it through the rigorous online tests that Cisco gives you at Cisco.com.

   There's nothing worse than trying to troubleshoot more than one problem at a time—particularly if the embedded problem is something you have helped create! This is why you undo a plan when it does not solve the problem.

   Document the findings.

6. Observe results.

   Determine whether you permanently solved the problem or whether you just implemented a temporary solution.

   Make sure the affected party/parties think you fixed the problem. Then document the results and action plan. If you did not fix the problem, go back and try the next item on your action plan. *Always* undo any previous changes before you iterate the process or attempt your next plan of action.

If you have not fixed the problem, consider taking time away from the problem; you might be able to come back with a fresh perspective. *Always* have a backup plan.

Document the findings.

7 Document the solution.

Document each step along the way and the final solution to improve overall expertise as you support your internetwork. Many people forget this step.

Whether manual or automated, maintain a database and change log for each piece of equipment. For example you should do things such as maintain version control, comment your configurations, add descriptions to your interfaces, and capture your logs for later review. Include change notes for yourself and others in the configurations with **remark**, **!**, or **#** for comments. If you are capturing your logs, show the clock a couple of times to show when things happened.

Record what you have done, have a fallback plan, and provide a history for yourself and others.

Plan for people and equipment upgrades (future expansion). Emergency changes to fix problems are one thing, but planned changes should be coordinated properly to assess the risk, plan for the change, communicate the change, implement the change, test the change, and document it.

One of the major goals of the CIT course and the Support exam is to make sure you establish a methodical mindset for troubleshooting so that your network operates with a minimum amount of downtime. The Cisco generic systematic approach is meticulous, disciplined, and optimistic. Any method that you are already used to is probably fine for practical purposes, as long as you are sure it takes advantage of the benefits a systematic approach can bring. For exam purposes, however, be very familiar with the Cisco problem-solving method.

NOTE Cisco offers another method called VISTA (View, Isolate, Solve, Test, Apply), which may be a little easier to recall in the real world of troubleshooting. Cisco's latest methodology says, "Define the problem, then Isolate, and Correct."

The following Trouble Ticket gives you a chance to apply the Cisco model to a sample network problem. The objectives are twofold. I want you to troubleshoot a particular technology by applying the seven-step Cisco troubleshooting approach presented earlier in this chapter. Figure 1-20 shows a graphic view of the scenario. Walk through the Trouble Ticket with me as I use the Cisco method to solve the problem and summarize some important technical concepts.

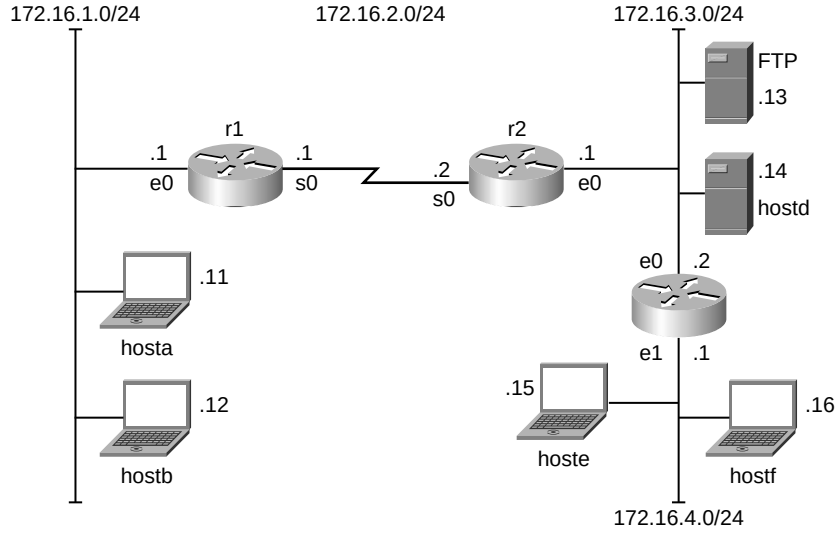
NOTE Even though the issue is not something previously discussed in the book, it is something you should be familiar with in a Cisco environment.

Trouble Ticket: Users Are Not Losers

An end user (hosta) calls in and reports, "I can't get to the FTP server."

This and Figure 1-20 is all the information you have been given, so you must brainstorm accordingly. Check your thoughts against the Trouble Ticket solution that follows.

Figure 1-20 *Users Are Not Losers*



Trouble Ticket Solution: Users Are Not Losers

1. Define the problem.

   hosta can't get to the FTP server, or at least that is what the user is telling you.

2. Gather facts.

   Fact gathering requires you to ask lots of questions of users and devices and to collaborate with others. As long as you are systematic and methodical, you can divide and conquer a bit while you are gathering facts to eliminate checking everything. For example, hosta can't get to the FTP server. Instead of asking too many questions of the users, you can try a simple ping and tracert from hosta to both of the servers on network 172.16.3.0. If you can't ping, you know you must test for Physical, Data Link, and/or Network Layer issues between the source and destination networks. Perhaps you don't have a route to the destination network; but if you can ping, you can move your testing above the Network Layer. Maybe there is a problem with a router in the path. Tracert (or traceroute) is helpful there. You should have gathered and documented facts such as the following:

   - hosta can ping hostb, its gateway 172.16.1.1, both devices on remote network 172.16.3.0, and everything on network 172.16.4.0.

- hosta can tracert to all hosts on its local network and all remote networks shown in Figure 1-20.
- hosta and hostb on network 172.16.1.0 can't FTP to the FTP server on network 172.16.3.0. Other remote hosts and routers can FTP to the FTP server on network 172.16.3.0.
- You are not sure whether this ever worked because you don't have any other documentation.

3 Consider possibilities (based on facts).

Narrow down the facts and possibilities so that you can focus on what is relevant. Your facts should help you further define the original problem. The real issue is that hosta can't FTP to the FTP server 172.16.3.13. Because r1, r2, r3, and the hosts on network 172.16.4.0 can FTP, you know the issue is not with the FTP services on the server; instead the problem more likely involves the 172.16.1.0 network off of r1. You also know that your problem is not a Physical or Data Link Layer issue because all pings are successful.

4 Create an action plan.

Start with the most likely cause and change only one variable at a time. r1 is a very likely target, and everyone on the local 172.16.1.0 network is affected. Your action plan should include further investigation of r1. Save your configurations and write down every step you intend to perform. The Cisco IOS **show access-lists** command is by far the most relevant here.

5 Implement action plan.

The **show access-lists** command reveals the following on r1:

```
r1#sh access-lists
Extended IP access list ftp
     deny tcp any any (16 matches)
     permit icmp any any (4 matches)
     permit tcp any 0.0.0.0 255.255.0.0 eq ftp-data
     permit tcp any 0.0.0.0 255.255.0.0 eq ftp
```

The access list is the reason hosta can't FTP to the server. After you found the access list, you determined that it was applied outbound on interface s0 by examining the running configuration. You verified this by typing **show ip interface s0**, and sure enough the FTP access list was applied outbound.

Your analysis requires that you further define your action plan. You may temporarily decide to remove the statement off the interface so that the access list is not applied (**no ip access-group ftp out**). If this allows hosta to FTP, you should fix the access list for a more permanent solution. You should shut the interface down until all changes have been made. Note that the access list denies all TCP communications from anywhere to anywhere outbound. Your pings should succeed because of the ICMP statement. However, the two **permit tcp** statements don't accomplish a thing because they use the subnet mask

rather than the necessary wildcard mask. In this example, it is probably easiest to completely remove the old access list and create another one. You may need to go back to the top of the Cisco troubleshooting ladder on this one to gather more facts to determine exactly what the access list should do. Assume you did that and hosta, hostb, and any other hosts except host 172.16.1.13 added to network 172.16.1.0 should be able to FTP to 172.16.3.13. Although host 172.16.1.13 should not be able to FTP, all other IP-related commands should be allowed. You also want to determine whether host 172.16.1.13 ever attempts to FTP to the FTP server. Your new action plan should attempt to create and apply the following access list on r1:

```
r1(config)#ip access-list extended ftp
r1(config-ext-nacl)#deny tcp host 172.16.1.13 host 172.16.3.13 eq 20
r1(config-ext-nacl)#deny tcp host 172.16.1.13 host 172.16.3.13 eq 21
r1(config-ext-nacl)#permit ip any any

r1(config-ext-nacl)#interface serial 0
r1(config-if)#ip access-group ftp out
r1(config-if)#end
r1#copy running-config startup-config
```

6 Observe results.

Test your new access list by making sure that hosta and hostb can ping and FTP to the FTP server. If possible, add host 172.16.1.13 to ensure that it can't FTP to the FTP server; it should, however, be able to ping and tracert. It is critical to make sure you fixed the problem at hand and did not introduce any others. In addition, everyone must be content with your solution; otherwise it is still a problem. Make sure you save your configurations and document the findings. Now that things are working, you may consider revisiting your action plan. Alternatively, you could place this ACL inbound on interface e0 to filter the traffic closer to the source.

7 Document the solution.

Document all changes and your new configurations.

```
r1#show access-lists
Extended IP access list ftp
    deny tcp host 172.16.1.13 host 172.16.3.13 eq ftp-data
    deny tcp host 172.16.1.13 host 172.16.3.13 eq ftp
    permit ip any any
r1#show ip interface serial 0
Serial0 is up, line protocol is up
  Internet address is 172.16.2.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is ftp
  Inbound  access list is not set
  Proxy ARP is enabled
```

This Trouble Ticket has provided you with the opportunity to practice solving a problem using the Cisco seven-step approach and to review the following about access lists:

- When coding your ACLs, use top-down processing. Place the more specific items at the top and the general items at the bottom. Don't rely on a particular version of the IOS to order things for you.
- At least one **permit** statement is required; otherwise the implicit default of **deny any any** applies.
- Wildcard masks are used in ACLs. They predate subnet masks. Many people write a 0 with a line through it anyhow, so just draw a checkmark over it so that you remember that 0 means check. Think of the 1 with a dot above it so it looks more like the letter *i*, for *ignore*. For example, 172.16.1.13 0.0.0.0 means check all bits if this statement is in the ACL. Instead of spelling out the 0.0.0.0, I used the keyword **host** in my revised access list. By the same token, 172.16.0.0 0.0.255.255 implies to check the first 16 bits and ignore the last 16. 172.16.0.0 255.255.255.255 is the same as ignore all bits or the keyword **any**; an example being **permit ip any any**, which says permit all IP traffic from anywhere to anywhere.
- An ACL will not block traffic originating from the router. You observed this when you could FTP from r1 to 172.16.3.13 but not from hosta, where you had to go through the affected router.
- Named ACLs enable you to remove individual lines of code; although in this example, it was just as easy to delete it and start again.
- To delete an ACL, it is best practice to type **no** in front of the lines to create and apply the ACL to take care of any version inconsistencies. An empty ACL that is applied to an interface used to deny everything; now it permits everything. Just don't apply an empty ACL. Always create before you apply and use third-party tools for editing because you can't add or delete lines within the ACL in the Cisco command-line interface (CLI). However, you can delete lines within a named ACL.
- When troubleshooting ACLs, use **debug** and **log**; matches are also helpful. For example, the **log** keyword is how you can determine whether the host you wanted to deny ever attempts to FTP anyhow.
- Although not apparent in this example, you should not have problems modifying an outbound ACL remotely. However, you could potentially lock yourself out of the router with an inbound.
- Remember, one ACL per protocol, per interface, per direction. This would have been an issue if you would not have temporarily issued the **no ip access-group** statement so that the incorrect ACL wasn't applied.

- Now that you have gone through the entire exercise and technically solved the implied problem, you probably should have asked yourself whether the user was *supposed to* have access to the FTP server! Be aware of any policies in place outside the actual configuration of networking equipment. This should be part of your "gathering facts" step.

Practical Troubleshooting

One day you will get a call that says the network is down. Be very prepared to divide and conquer to get to the real problem. Work through the affected layers. Remember that shooting trouble is often about questions. Do you ask the equipment or the user? Who is waiting for the results? What has happened? When did it occur? Why? Where did it happen? Plug it in; turn it on. Make sure you have lights and power. Did it ever work? What has changed since it last worked? Check the obvious. Who is complaining? Is it an end-system issue? Check the application and configuration if it is an individual person or machine. Is it a group of people or machines? Check connectivity and performance. Run through the OSI layers; remember ping and trace; check the routing tables. Is it a local segment issue or does it extend through routers? Is a bad NIC, cable, or device causing performance degradation? Ping yourself, ping someone local, ping the default gateway, or start by pinging a remote network to test all of these. Trace the problem. What is slow: cabling, link, devices? Do you have a baseline comparison? Use ping, trace, a protocol analyzer, and other tools on an ongoing basis. Did someone else try to fix the problem? Never be too proud to ask for help.

Actually it is quite helpful to have people with different backgrounds on your team, whether it be in a test lab or practical environment. You must be able to prioritize problem areas and people for that matter. Normally if the CEO has a problem, you take care of it immediately; if everyone else in the company is *down*, however, obviously they take precedence (one of those 8, 9, 10 layer things—finance, politics, and religion). Modern day prioritization says let the CEO wait so that when you ask for more people or resources the CEO recognizes the need.

Models and Methods

I would like to credit my REDI model source, but it is something I learned about in college while at Johns Hopkins. I think it came from a systems design or database textbook. In any case, the REDI model gives me a systematic mindset for whatever I am doing. It is quite effective yet easy to remember. The basic tenants of the REDI model are as follows:

- Define Your **R**equirements
- **E**valuate the Alternatives
- **D**esign and Develop
- **I**mplement (and then do it all over again)

If the design and development work is done, you are probably troubleshooting or starting the life cycle all over again. Whether it is taking a certification test, a new consulting gig, or applying for a job, taking a structured approach and documenting appropriately are of utmost importance.

Baselining and Documentation

Baselining and documentation are crucial to your long-term success with internetwork troubleshooting. This is not just theory; for if you don't know what is normal, how do you know where to begin with troubleshooting. What if you get the call saying the network is slow? Slow compared to what? Did you collect any data when the network was installed and running properly, do you audit it from time to time, or have you just taken the put-out-the-fire approach to network management? You should know what information to collect, how to store it, and who is affected by what. Utilization (CPU and bandwidth); memory; error statistics; protocol distribution; traffic statistics; changes in hardware, software, and configuration; and past troubleshooting documentation are all important aspects for troubleshooting. Track patterns and trends. When you find out who or what is affected, time of day, day of week, and month of year, you can compare this to your baseline.

In the form of pictures, charts, maps, tables, and databases, your baseline should include items such as the following:

Model number	Serial number
RAM/Flash memory	IOS version
Config-register settings	Interface statistics
Bandwidth/speed	Clocking
Encapsulation	Duplex
Descriptions	Addresses
Passwords	Spanning-tree portfast
VLANs	Routed protocols
Routing protocols	Bridged protocols

In practical application, other things that are valuable to document include the detailed location of equipment (down to the country, state, city, building, wiring closet, rack, and position). Store this information in a log book, on your network, or your personal digital assistant (PDA) for that matter.

From a practical viewpoint, pictures are wonderful resources. Physical layouts, logical maps, lists of protocols (routed, bridged, and routing including redistribution and filtering) can aid you in the process. Include your Internet connections, addressing plans, DHCP, NAT, security plans, and application implementations in your diagrams. What is normal for you may not be what is normal for the next person, so documentation and diagrams are invaluable. Change is truly the only thing constant in this industry—software, hardware, and configuration. Doctors keep records on your children from the time they are born throughout their life, documenting such things as shots, diseases, symptoms, cures, operations, and so on. Do the same for your

network. The answer to your problem will be easier to find if it happened before and you documented it in a database of some kind.

Practical troubleshooting is all about taking the previous methods and models and applying them to the real world. Regardless of the model/method you follow, if you take a systematic approach you will be able to narrow the problem down. Amateurs and pros alike should be able to analyze new and complex problems with an effective strategy. It is not necessary to be a know-it-all to be an effective troubleshooter. A successful troubleshooter is a logical thinker with common sense and people skills. Divide and conquer as you did with the access list Trouble Ticket; narrow possibilities down by the layers. Analyze and resolve. If you can't, escalate the issue to the team that can.

An unsystematic approach is time-consuming and costly. This concept is stressed on the CCNP Troubleshooting exam, CCIE exams, and CCSI exams. Troubleshooting models and methods help reduce a large set of causes to a smaller set of causes or, better yet, a single cause. Then you can solve the problem and document it for future reference to help mitigate the pressures of supporting critical complex internetworks. Remember, however, that vendor interoperability is far less smooth than theory models pretend it to be.

Review Questions

Use this chapter and your practical troubleshooting knowledge and skills to answer the following questions. The answers are located in Appendix A, "Answers to Review Questions."

1. The Transport Layer is the host-to-host layer in the OSI model and the TCP/IP suite. It is in-between the upper and lower layers and depending on the protocol is responsible for delivery, error detection, and correction. Describe the upper layers of the OSI model and include examples.
2. Describe the lower layers of the OSI model and include examples.
3. Draw a picture showing the differences between OSI layers and TCP/IP layers.
4. Explain encapsulation using the appropriate protocol data unit terminology.
5. Explain de-encapsulation, including how Layer 2 hands off to Layer 3, how Layer 3 hands off to Layer 4, and so on.
6. What is the difference between a hub, switch, and router?
7. What is the difference between routed and routing protocols? Give examples of each.
8. Describe packet flows through routers.
9. How can the OSI model assist in troubleshooting?
10. List the seven steps of the Cisco troubleshooting model?

Summary

This chapter presented an introduction to troubleshooting, a review of standards and protocols, industry models, troubleshooting methods, baselining, and documentation techniques. The chapter covered the OSI model and included information on how an understanding of that model can aid you in the troubleshooting process. The DoD TCP/IP suite was also covered and compared to the OSI model. The Trouble Ticket offered you an opportunity to apply the Cisco troubleshooting method and other techniques you learned in this chapter. Now that you have reviewed and studied a systematic approach to troubleshooting, you should determine whether you really have the right tools for the job.