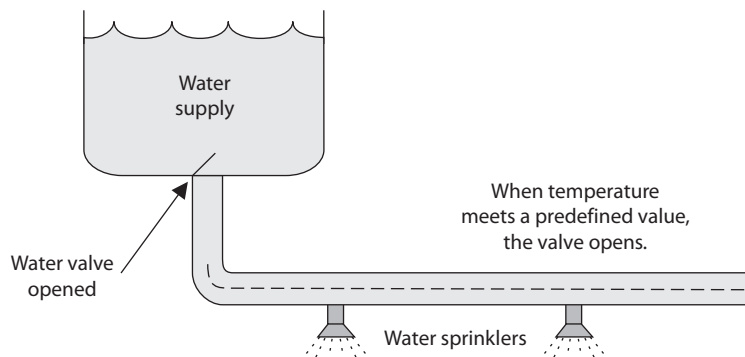


An organization should take great care in deciding which suppression agent and system is best for it. Four main types of water sprinkler systems are available:

- **Wet pipe** Wet pipe systems always contain water in the pipes and are usually discharged by temperature control–level sensors. One disadvantage of wet pipe systems is that the water in the pipes may freeze in colder climates. Also, if there is a nozzle or pipe break, it can cause extensive water damage. These types of systems are also called closed-head systems.
- **Dry pipe** In dry pipe systems, the water is not actually held in the pipes. The water is contained in a “holding tank” until it is released. The pipes hold pressurized air, which is reduced when a fire or smoke alarm is activated, allowing the water valve to be opened by the water pressure. Water is not allowed into the pipes that feed the sprinklers until an actual fire is detected. First, a heat or smoke sensor is activated; then, the water fills the pipes leading to the sprinkler heads, the fire alarm sounds, the electric power supply is disconnected, and finally water is allowed to flow from the sprinklers. These pipes are best used in colder climates because the pipes will not freeze. Figure 10-13 depicts a dry pipe system.
- **Preaction** Preaction systems are similar to dry pipe systems in that the water is not held in the pipes, but is released when the pressurized air within the pipes is reduced. Once this happens, the pipes are filled with water, but it is not released right away. A thermal-fusible link on the sprinkler head has to melt before the water is released. The purpose of combining these two techniques is to give people more time to respond to false alarms or to small fires that can be handled by other means. Putting out a small fire with a handheld extinguisher is better than losing a lot of electrical equipment to water damage. These systems are usually used only in data processing environments rather than the whole building because of the higher cost of these types of systems.
- **Deluge** A deluge system has its sprinkler heads wide open to allow a larger volume of water to be released in a shorter period. Because the water being released is in such large volumes, these systems are usually not used in data processing environments.

Figure 10-13
Dry pipe
systems do not
hold water in
the pipes.



Environmental Issues

In drier climates, or during the winter, the air contains less moisture, which can cause static electricity when two dissimilar objects touch each other. This electricity usually travels through the body and produces a spark from a person's finger that can release several thousand volts. This can be more damaging than you would think. Usually, the charge is released on a system casing and is of no concern, but sometimes it is released directly to an internal computer component and causes damage. People who work on the internal parts of a computer usually wear antistatic armbands to reduce the chance of this happening.

In more humid climates, or during the summer, more humidity is in the air, which can also affect components. Particles of silver can begin to move away from connectors onto copper circuits, which cement the connectors into their sockets. This can adversely affect the electrical efficiency of the connection. A *hygrometer* is usually used to monitor humidity. It can be manually read, or an automatic alarm can be set up to go off if the humidity passes a set threshold.

Chapter Review

Physical security of our sites and facilities requires a deliberate planning, execution, and review process. In this chapter, we have discussed the most important topics you'll need to know about to ensure that your organization's physical spaces are secure, but it's up to you to apply them in your particular situations. One of the most important aspects of securing a facility is controlling access in and out of it. In our experience, it is rare for an auditor (such as a physical penetration tester) to not be able to breach this perimeter through social engineering, lockpicking, or simply waiting for someone to leave a door propped open when they shouldn't. This underscores the importance of applying defense in depth and the other principles we discussed in the first half of the chapter.

The practical application of these secure design principles happens through security controls. Though our focus is on physical security, these controls can be administrative (e.g., policies and procedures), technical (e.g., keycard entry systems and security cameras), or physical (e.g., fences and guards). By carefully balancing threats, resources, and controls in a deliberate manner, we can provide effective site and facility security.

Quick Review

- A site is a geographic area with fixed boundaries that typically contains at least one building and its supporting structures (e.g., a parking lot or electric substation).
- A facility is a building or a part of a building dedicated to a specific purpose, such as corporate headquarters or a data center.
- The secure design principles covered in Chapter 9 for information systems are just as applicable to the design of physical security.
- The value of property within the facility and the value of the facility itself need to be ascertained to determine the proper budget for physical security so that security controls are cost-effective.

- Some physical security controls may conflict with the safety of people. These issues need to be addressed; human life is always more important than protecting a facility or the assets it contains.
- When looking at locations for a facility, consider local crime; natural disaster possibilities; and distance to hospitals, police and fire stations, airports, and railroads.
- Crime Prevention Through Environmental Design (CPTED) combines the physical environment and sociology issues that surround it to reduce crime rates and the fear of crime.
- CPTED provides four main strategies, which are natural access control, natural surveillance, territorial reinforcement, and maintenance.
- Natural access control is the guidance of people entering and leaving a space by the placement of doors, fences, lighting, and even landscaping.
- The goal of natural surveillance is not only to make criminals feel uncomfortable by providing many ways observers could potentially see them but also to make authorized personnel feel safe and comfortable by providing an open and well-designed environment.
- Territorial reinforcement creates physical designs (e.g., using walls, fences, landscaping) that emphasize or extend the organization's physical sphere of influence so legitimate users feel a sense of ownership of that space.
- CPTED's maintenance principle focuses on deterring criminal activity by making sites look well cared for, thus implying that site personnel are more attentive, well resourced, and alert.
- Target hardening focuses on denying access through physical and artificial barriers (alarms, locks, fences, and so on).
- If interior partitions do not go all the way up to the true ceiling, an intruder can remove a ceiling tile and climb over the partition into a critical portion of the facility.
- The primary power source is what is used in day-to-day operations, and the alternative power source is a backup in case the primary source fails.
- Smoke detectors should be located on and above suspended ceilings, below raised floors, and in air ducts to provide maximum fire detection.
- A fire needs high temperatures, oxygen, and fuel. To suppress it, one or more of those items needs to be reduced or eliminated.
- Portable fire extinguishers should be located within 50 feet of electrical equipment and should be inspected quarterly.
- CO₂ is a colorless, odorless, and potentially lethal substance because it removes the oxygen from the air in order to suppress fires.
- Window types that should be understood are standard, tempered, acrylic, wired, and laminated.

Questions

Please remember that these questions are formatted and asked in a certain way for a reason. Keep in mind that the CISSP exam is asking questions at a conceptual level. Questions may not always have the perfect answer, and the candidate is advised against always looking for the perfect answer. Instead, the candidate should look for the best answer in the list.

1. When should a CO₂ fire extinguisher be used?
 - A. When electrical equipment is on fire
 - B. When gases are on fire
 - C. When a combustible metal is on fire
 - D. In workspaces with paper products
2. When should a water sprinkler system be used?
 - A. When electrical equipment is on fire
 - B. When gases are on fire
 - C. When a combustible metal is on fire
 - D. In workspaces with paper products
3. Which of the following is not a main component of CPTED?
 - A. Natural access control
 - B. Natural surveillance
 - C. Territorial reinforcement
 - D. Target hardening
4. Which problems may be caused by humidity in an area with electrical devices?
 - A. High humidity causes excess electricity, and low humidity causes corrosion.
 - B. High humidity causes corrosion, and low humidity causes static electricity.
 - C. High humidity causes power fluctuations, and low humidity causes static electricity.
 - D. High humidity causes corrosion, and low humidity causes power fluctuations.
5. The fourth principle of Crime Prevention Through Environmental Design (CPTED) is maintenance. What does this principle entail?
 - A. Ensuring that target hardening controls remain in working order
 - B. Periodically assessing designs to ensure they remain effective
 - C. Maintaining visibility over all elements of the environmental design
 - D. Deterring criminal activity by making sites look well cared for

6. Which of the following answers contains a category of controls that does not belong in a physical security program?
- A. Deterrence and delaying
 - B. Response and detection
 - C. Assessment and detection
 - D. Delaying and lighting

Use the following scenario to answer Questions 7–9. You are the CISO for a data analytics company and, after reading this chapter, have decided to review the physical security of your facility. You currently lease the top three floors of a high-rise building in a major metropolitan area. The top floor contains the executive suites and conference facilities. The next floor down houses your data center and research and development (R&D) and software development teams. The lower floor is where your administrative and sales staff workspaces are located.

7. As part of your lease agreement, the building owners provide a reception area in the lobby with receptionists, security guards, keycard-based access controls, and security cameras. What secure design principle is this arrangement an example of?
- A. Zero trust
 - B. Trust but verify
 - C. Shared responsibility
 - D. Separation of duties
8. As you ride the elevator up, you notice that your company employees can use their keycards to access any of the top three floors. What secure design principle could this violate?
- A. Shared responsibility
 - B. Simplicity
 - C. Defense in depth
 - D. Secure defaults
9. Based on your preliminary findings, you are concerned about the security of your facility and decide to redo the entire plan. What would you do first?
- A. Determine the resources available to you
 - B. Conduct an audit of the current posture
 - C. Apply the secure design principles
 - D. Conduct a risk assessment

Answers

1. **A.** CO₂ fire extinguishers work by displacing oxygen and thus choking the fire. This approach can put human lives in danger of also choking, so it should not be used in workspaces. It is also not ideal for gas or metal fires. It is, however, very well suited for electrical fires since it minimizes the risk of electric shock and damage to the equipment.
2. **D.** Water sprinklers are very effective against wood and paper fires and are safe for humans, so they are a good choice for use in regular workspaces. They are, however, particularly bad choices to use against the other types of fires listed (electric, gas, metal).
3. **D.** Target hardening has to do with implementing locks, security guards, and proximity devices. Natural access control is the use of the environment to control access to entry points, such as using landscaping and bollards. An example of natural surveillance is the construction of pedestrian walkways so there is a clear line of sight of all the activities in the surroundings. Territorial reinforcement gives people a sense of ownership of a property, giving them a greater tendency to protect it. These concepts are all parts of CPTED.
4. **B.** High humidity can cause corrosion, and low humidity can cause excessive static electricity. Static electricity can short out devices or cause loss of information.
5. **D.** CPTED's maintenance principle focuses on deterring criminal activity by making sites look well cared for, thus implying that site personnel are more attentive, well resourced, and possibly alert.
6. **D.** The categories of controls that should make up any physical security program are deterrence, delaying, detection, assessment, and response. Lighting is a control itself, not a category of controls.
7. **C.** Shared responsibility usually involves different organizations that have a services agreement, such as leasing office spaces and having the landlord provide security in the entry points. Separation of duties is similar but applies to staff responsibilities divided between individuals to deter and prevent any one party from damaging the company. While elements of zero trust and trust but verify are present in the scenario, they are not the salient ones.
8. **C.** Defense in depth means you should create concentric rings around your most critical resources. In this case, only certain individuals should have access to the executive spaces and, possibly, the data center floor, unless you have additional security controls in place as soon as someone enters those two floors.
9. **D.** Site and facility security, like any other type of security, must start with an understanding of the risks faced by the organization. These risks determine the required resources for the effort and the manner in which the secure design principles are applied to your particular situation.

This page intentionally left blank

PART IV

Communication and Network Security

- **Chapter 11** Networking Fundamentals
- **Chapter 12** Wireless Networking
- **Chapter 13** Securing the Network
- **Chapter 14** Network Components
- **Chapter 15** Secure Communications Channels

This page intentionally left blank

Networking Fundamentals

This chapter presents the following:

- Data communications foundations
- Networking protocols
- Local, metropolitan, and wide area networks

The Internet...it's a series of tubes.

—Ted Stevens

Before we dive into communications and network security, it makes sense to review (and maybe pick up a bit of new information on) the fundamentals of data communications networks. Data communications and networking are complex topics, mainly because so many technologies are involved. Our current technologies are constantly evolving, and every month there seems to be new “emerging” ones that we have to learn, understand, implement, and secure. As security professionals, we need a solid grasp of networking software, protocols, services, and devices. We have to be able to identify and deal with interoperability issues (ideally before developing or acquiring a new system). Armed with all this knowledge and skill, we need to anticipate or discover vulnerabilities, both in individual components and in their interactions with each other, and devise effective controls for them. This can be a challenging task. However, if you are knowledgeable, have a solid practical skill set, and are willing to continue to learn, you can have more career opportunities than you know what to do with.

In this chapter we will start with the basics of data communications and networking, and then build upon them and identify many of the security issues that are involved. We'll follow up in the next chapter with a discussion of wireless technologies that take the place of the cables described in this chapter. Then, in Chapter 13 we'll dive into the protocols that drive the Internet. This will set the stage for understanding how we secure this hodgepodge of technologies, the focus of Chapters 14 and 15.

Data Communications Foundations

Data communications have made amazing advances in a relatively short period of time. In the beginning of the Computer Age, mainframes were the name of the game. They were isolated powerhouses, and many had “dumb” terminals hanging off them, but this

was not true networking. In the late 1960s and early 1970s, some technical researchers came up with ways of connecting all the mainframes and Unix systems to enable them to communicate. This marked the Internet's first baby steps.

While access to shared resources was a major drive in the evolution of networking, today the infrastructure that supports these shared resources and the services these components provide is really the secret to the secret sauce. As we will see, networks are made up of routers, switches, servers, proxies, firewalls, intrusion detection/prevention systems (IDS/IPSS), storage systems, virtual private network (VPN) concentrators, public key infrastructure, private branch exchanges (PBXs), and more. While functionality is critical, there are other important requirements that need to be understood when architecting a network, such as scalability, redundancy, performance, security, manageability, and maintainability.

Infrastructure provides foundational capabilities that support almost every aspect of our lives. When most people think of technology, they focus on the end systems that they interact with—laptops, mobile phones, tablet PCs, workstations, and so on—or the applications they use, such as e-mail, Facebook, websites, instant messaging, Twitter, and online banking. Most people do not even give a thought to how this stuff works under the covers, and many people do not fully realize all the other stuff that is dependent upon technology: medical devices, critical infrastructure, weapon systems, transportation, satellites, telephony, and so forth. People say it is love that makes the world go around, but let them experience one day without the Internet. We are all more dependent upon the Matrix than we fully realize, and as security professionals we need to not only understand the Matrix but also secure it.

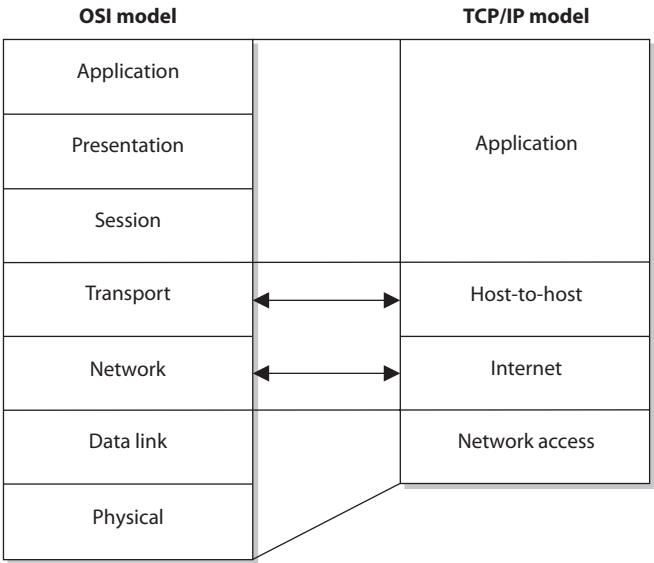
Before we get into the weeds of actual devices, systems, and services, it will be helpful to have a reference model so that we can put like pieces in the same bin. This will allow us to compare things that perform similar functions so that we can better see how they differ and what those differences may mean for security. Let's start off by introducing the two most common models for understanding networked systems: the OSI model and the TCP/IP model.

Network Reference Models

In the early 1980s, the International Organization for Standardization (as it is referred to internationally, but commonly abbreviated as ISO) worked to develop a protocol set that would be used by all vendors throughout the world to allow the interconnection of network devices. This movement was fueled with the hopes of ensuring that all vendor products and technologies could communicate and interact across international and technical boundaries. The actual protocol set did not catch on as a standard, but the *Open Systems Interconnection (OSI) reference model* was adopted and is used as an abstract framework to which most operating systems and protocols adhere.

Many people think that the OSI reference model arrived at the beginning of the computing age as we know it and helped shape and provide direction for many, if not all, networking technologies. However, this is not true. In fact, it was introduced in 1984, at which time the basics of the Internet had already been developed and implemented, and the basic Internet protocols had been in use for many years.

Figure 11-1
The OSI and TCP/
IP networking
models



The Transmission Control Protocol/Internet Protocol (TCP/IP) suite actually has its own model that predates the OSI model by several years. As a bit of background, the Internet as we know it grew from the Advanced Research Project Agency Network (ARPANET) program that started in the late 1960s. By 1978, ARPANET researchers realized that a monolithic approach to networking was not going to scale well. That’s when they split what had until then been known as the Transmission Control Program (which encompassed all aspects of getting data from point A to point B) into two distinct layers: TCP and IP. Everything that happened below IP was the domain of network access engineers, and everything above TCP was the domain of application developers. The idea caught on and the TCP/IP reference model was born. It is often used today when examining and understanding networking issues. Figure 11-1 shows the differences between the OSI and TCP/IP networking models. In this chapter, we will focus on the OSI model.



NOTE The host-to-host layer is sometimes called the transport layer in the TCP/IP model. The application layer in the TCP/IP architecture model is equivalent to a combination of the application, presentation, and session layers in the OSI model.

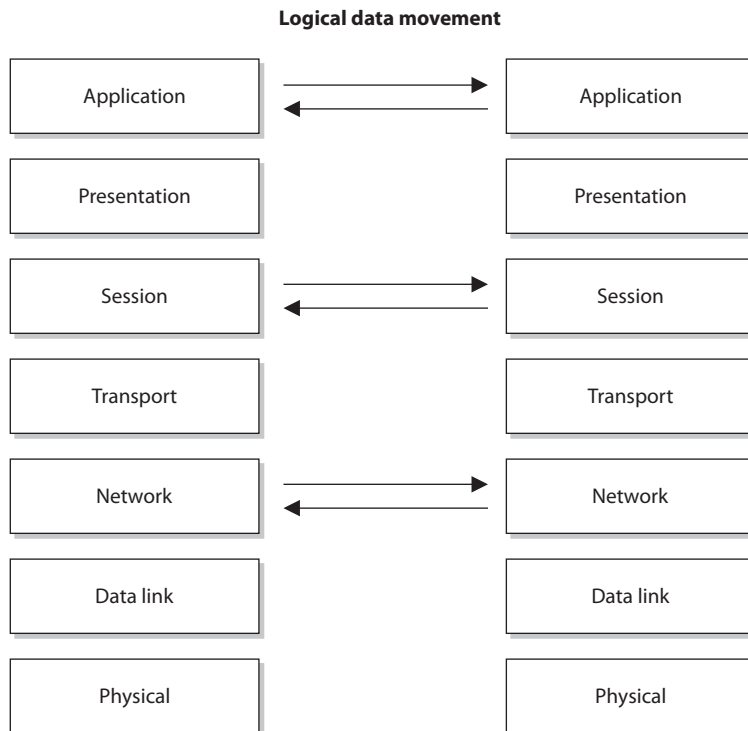
Protocols

Before we delve into the details of each layer of the OSI model, we need to examine the concept of network protocols. A *network protocol* is a standard set of rules that determines how systems will communicate across networks. Two different systems that use the same protocol can communicate and understand each other despite their differences, similar to how two people can communicate and understand each other by using the same language.

The OSI reference model, as described by ISO Standard 7498-1, provides important guidelines used by vendors, engineers, developers, and others. The model segments the networking tasks, protocols, and services into different layers. Each layer has its own responsibilities regarding how two computers communicate over a network. Each layer has certain functionalities, and the services and protocols that work within that layer fulfill them.

The OSI model's goal is to help vendors develop products that will work within an open network architecture. An *open network* architecture is one that no vendor owns, that is not proprietary, and that can easily integrate various technologies and vendor implementations of those technologies. Vendors have used the OSI model as a jumping-off point for developing their own networking frameworks. These vendors use the OSI model as a blueprint and develop their own protocols and services to produce functionality that is different from, or overlaps, that of other vendors. However, because these vendors use the OSI model as their starting place, integration of other vendor products is an easier task, and the interoperability issues are less burdensome than if the vendors had developed their own networking framework from scratch.

Although computers communicate in a physical sense (electronic signals are passed from one computer over a wire to the other computer), they also communicate through logical channels. Each protocol at a specific OSI layer on one computer communicates with a corresponding protocol operating at the same OSI layer on another computer. This happens through *encapsulation*.



Here's how encapsulation works: A message is constructed within a program on one computer and is then passed down through the network protocol's stack. A protocol at each layer adds its own information to the message, creating a *protocol data unit (PDU)*. Thus, the message grows in size as it goes down the protocol stack. The message is then sent to the destination computer, and the encapsulation is reversed by taking the packet apart through the same steps used by the source computer that encapsulated it. At the data link layer, the PDU pertaining to the data link layer is deconstructed, and the packet is sent up to the next layer. Then at the network layer, the network layer PDU is stripped and processed, and the message is again passed up to the next layer, and so on. This is how computers communicate logically. The information stripped off at the destination computer informs it how to interpret and process the packet properly. Data encapsulation is shown in Figure 11-2.

A protocol at each layer has specific responsibilities and control functions it performs, as well as data format syntaxes it expects. Each layer has a special interface (connection point) that allows it to interact with three other layers: (1) communications from the interface of the layer above it, (2) communications to the interface of the layer below it, and (3) communications with the same layer in the interface of the target packet address. The control functions, added by the protocols at each layer, are in the form of headers and trailers of the packet.

The benefit of modularizing these layers, and the functionality within each layer, is that various technologies, protocols, and services can interact with each other and provide the proper interfaces to enable communications. This means a computer can use an application protocol developed by Microsoft, a transport protocol developed by Apple,

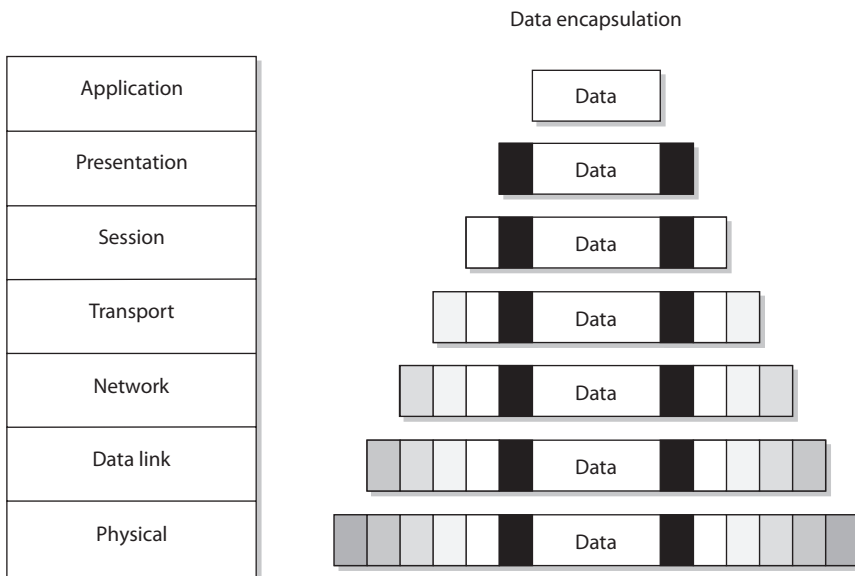


Figure 11-2 Each OSI layer protocol adds its own information to the data packet.

Attacks at Different Layers

As we examine the different layers of a common network stack, we will also look at the specific attack types that can take place at each layer. One concept to understand at this point is that a network can be used as a *channel for an attack*, or the network can be the *target of an attack*. If the network is a channel for an attack, this means the attacker is using the network as a resource. For example, when an attacker sends a virus from one system to another system, the virus travels through the network channel. If an attacker carries out a denial-of-service (DoS) attack, which sends a large amount of bogus traffic over a network link to bog it down, then the network itself is the target. As you will see throughout this book, it is important to understand how attacks take place and where they take place so that the correct countermeasures can be put into place.

and a data link protocol developed by Cisco to construct and send a message over a network. The protocols, technologies, and computers that operate within the OSI model are considered *open systems*. Open systems are capable of communicating with other open systems because they implement international standard protocols and interfaces. The specification for each layer's interface is very structured, while the actual code that makes up the internal part of the software layer is not defined. This makes it easy for vendors to write plug-ins in a modularized manner. Systems are able to integrate the plug-ins into the network stack seamlessly, gaining the vendor-specific extensions and functions.

Understanding the functionalities that take place at each OSI layer and the corresponding protocols that work at those layers helps you understand the overall communication process between computers. Once you understand this process, a more detailed look at each protocol will show you the full range of options each protocol provides and the security weaknesses embedded into each of those options.

Application Layer

The *application layer*, layer 7, works closest to the user and provides file transmissions, message exchanges, terminal sessions, and much more. This layer does not include the actual applications, but rather the protocols that support the applications. When an application needs to send data over the network, it passes instructions and the data to the protocols that support it at the application layer. This layer processes and properly formats the data and passes it down to the next layer within the OSI model. This happens until the data the application layer constructed contains the essential information from each layer necessary to transmit the data over the network. The data is then put on the network cable and transmitted until it arrives at the destination computer.

As an analogy, let's say that you write a letter that you would like to send to your congressman. Your job is to write the letter, your clerk's job is to figure out how to get it to him, and the congressman's job is to read your letter and respond to it. You (the application) create the content (message) and hand it to your assistant (application layer protocol).

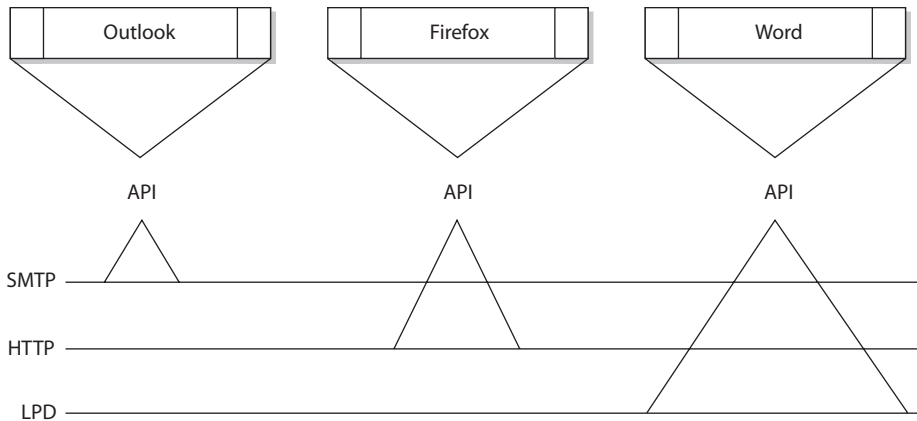


Figure 11-3 Applications send requests to an API, which is the interface to the supporting protocol.

Your assistant puts the content into an envelope, writes the congressman's address on the envelope (inserts headers and trailers), and puts it into the mailbox (passes it on to the next protocol in the network stack). When your assistant checks the mailbox a week later, there is a letter from the congressman (the remote application) addressed to you. Your assistant opens the envelope (strips off headers and trailers) and gives you the message (passes the message up to the application).

Some examples of the protocols working at the application layer are the Simple Mail Transfer Protocol (SMTP), the Hypertext Transfer Protocol (HTTP), and the Line Printer Daemon (LPD) protocol. Figure 11-3 shows how applications communicate with the underlying protocols through application programming interfaces (APIs). If a user makes a request to send an e-mail message through her e-mail client Outlook, the e-mail client sends this information to SMTP. SMTP adds its information to the user's message and passes it down to the presentation layer.

Presentation Layer

The *presentation layer*, layer 6, receives information from the application layer protocol and puts it in a format that any process operating at the same layer on a destination computer following the OSI model can understand. This layer provides a common means of representing data in a structure that can be properly processed by the end system. This means that when a user creates a Word document and sends it out to several people, it does not matter whether the receiving computers have different word processing programs; each of these computers will be able to receive this file and understand and present it to its user as a document. It is the data representation processing that is done at the presentation layer that enables this to take place. For example, when a Windows 10 computer receives a file from another computer system, information within the file's header indicates what type of file it is. The Windows 10 operating system has a list of file types it understands and a table describing what program should be used to open and

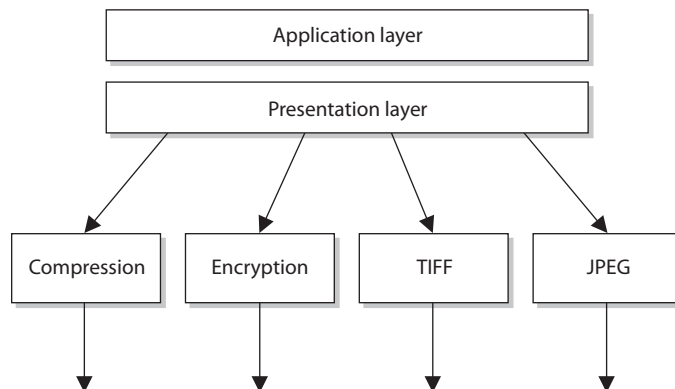
manipulate each of these file types. For example, suppose the sender e-mails a Portable Document Format (PDF) file created in Word and the receiver uses a Linux system. The receiver can open this file because the presentation layer on the sender's system encoded the file and added a descriptive header in accordance with the Multipurpose Internet Mail Extensions (MIME) standards, and the receiver's computer interprets the header's MIME type (Content-Type: application/pdf), decodes the file, and knows to open it with its PDF viewer application.

The presentation layer is concerned not with the meaning of data but with the syntax and format of that data. It works as a translator, translating the format an application is using to a standard format used for passing messages over a network. If a user uses a graphics application to save a file, for example, the graphic could be a Tagged Image File Format (TIFF), Graphic Interchange Format (GIF), or Joint Photographic Experts Group (JPEG) format. The presentation layer adds information to tell the destination computer the file type and how to process and present it. This way, if the user sends this graphic to another user who does not have the same graphics application, the receiving user's operating system can still present the graphic because it has been saved in a standard format. Figure 11-4 illustrates the conversion of a file into different standard file types.

This layer also handles data compression and encryption issues. If a program requests a certain file to be compressed and encrypted before being transferred over the network, the presentation layer provides the necessary information for the destination computer. It provides information on how the file was encrypted and/or compressed so that the receiving system knows what software and processes are necessary to decrypt and decompress the file. Let's say Sara compresses a file using WinZip and sends it to you. When your system receives this file, it looks at data within the header (Content-Type: application/zip) and knows what application can decompress the file. If your system has WinZip installed, then the file can be decompressed and presented to you in its original form. If your system does not have an application that understands the compression/decompression instructions, the file will be presented to you with an unassociated icon.

Figure 11-4

The presentation layer receives data from the application layer and puts it into a standard format.



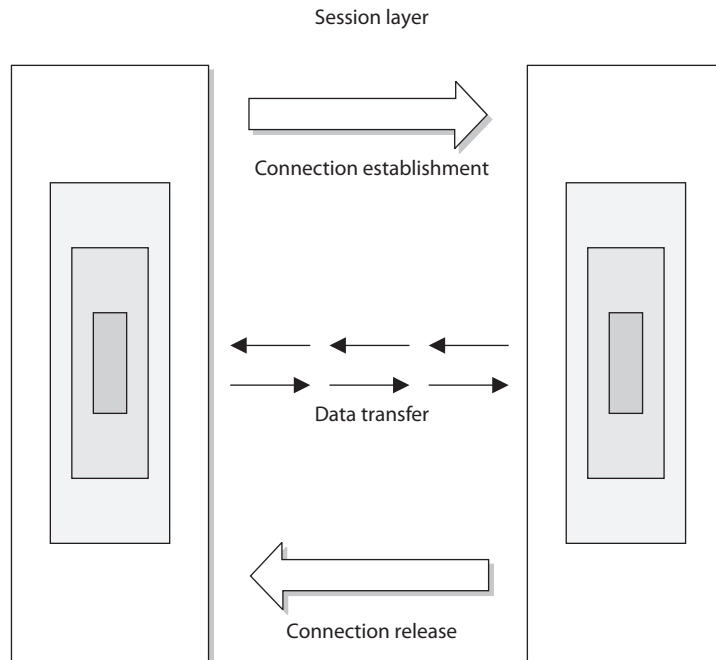
Session Layer

When two applications need to communicate or transfer data between themselves, a connection may need to be set up between them. The *session layer*, layer 5, is responsible for establishing a connection between the two applications, maintaining it during the transfer of data, and controlling the release of this connection. A good analogy for the functionality within this layer is a telephone conversation. When Kandy wants to call a friend, she uses the telephone. The telephone network circuitry and protocols set up the connection over the telephone lines and maintain that communication path, and when Kandy hangs up, they release all the resources they were using to keep that connection open.

Similar to how telephone circuitry works, the session layer works in three phases: connection establishment, data transfer, and connection release. It provides session restart and recovery if necessary and provides the overall maintenance of the session. When the conversation is over, this path is broken down and all parameters are set back to their original settings. This process is known as *dialog management*. Figure 11-5 depicts the three phases of a session. Some protocols that work at this layer are the Layer 2 Tunneling Protocol (L2TP), Point-to-Point Tunneling Protocol (PPTP), and Remote Procedure Call (RPC).

Figure 11-5

The session layer sets up the connection, maintains it, and tears it down once communication is completed.



The session layer protocol can enable communication between two applications to happen in three different modes:

- **Simplex** Communication takes place in one direction, though in practice this is very seldom the case.
- **Half-duplex** Communication takes place in both directions, but only one application can send information at a time.
- **Full-duplex** Communication takes place in both directions, and both applications can send information at the same time.

Many people have a hard time understanding the difference between what takes place at the session layer versus the transport layer because their definitions sound similar. Session layer protocols control application-to-application communication, whereas the transport layer protocols handle computer-to-computer communication. For example, if you are using a product that is working in a client/server model, in reality you have a small piece of the product on your computer (client portion) and the larger piece of the software product is running on a different computer (server portion). The communication between these two pieces of the same software product needs to be controlled, which is why session layer protocols even exist. Session layer protocols take on the functionality of middleware, which allows software on two different computers to communicate.

Session layer protocols provide interprocess communication channels, which allow a piece of software on one system to call upon a piece of software on another system without the programmer having to know the specifics of the software on the receiving system. The programmer of a piece of software can write a function call that calls upon a subroutine. The subroutine could be local to the system or be on a remote system. If the subroutine is on a remote system, the request is carried over a session layer protocol. The result that the remote system provides is then returned to the requesting system over the same session layer protocol. This is how RPC works. A piece of software can execute components that reside on another system. This is the core of distributed computing.



NOTE One security issue common to RPC (and similar interprocess communication systems) is improperly configured authentication or the use of unencrypted communications.

Session layer protocols are the least used protocols in a network environment; thus, many of them should be disabled on systems to decrease the chance of them being exploited. RPC, NetBIOS, and similar distributed computing calls usually only need to take place within a network; thus, firewalls should be configured so this type of traffic is not allowed into or out of a network. Firewall filtering rules should be in place to stop this type of unnecessary and dangerous traffic.

Transport Layer

When two computers are going to communicate through a connection-oriented protocol, they first agree on how much information each computer will send at a time, how to verify the integrity of the data once received, and how to determine whether a packet was lost along the way. The two computers agree on these parameters through a handshaking process at the *transport layer*, layer 4. The agreement on these issues before transferring data helps provide more reliable data transfer, error detection, correction, recovery, and flow control, and it optimizes the network services needed to perform these tasks. The transport layer provides end-to-end data transport services and establishes the logical connection between two communicating computers.



NOTE Connection-oriented protocols, such as Transmission Control Protocol (TCP), provide reliable data transmission when compared to connectionless protocols, such as User Datagram Protocol (UDP). This distinction is covered in more detail in the “Internet Protocol Networking” section, later in the chapter.

The functionality of the session and transport layers is similar insofar as they both set up some type of session or virtual connection for communication to take place. The difference is that protocols that work at the session layer set up connections between *applications*, whereas protocols that work at the transport layer set up connections between *computer systems*. For example, we can have three different applications on computer A communicating with three applications on computer B. The session layer protocols keep track of these different sessions. You can think of the transport layer protocol as the bus. It does not know or care what applications are communicating with each other. It just provides the mechanism to get the data from one system to another.

The transport layer receives data from many different applications and assembles the data into a stream to be properly transmitted over the network. The main protocols that work at this layer are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Information is passed down from different entities at higher layers to the transport layer, which must assemble the information into a stream, as shown in Figure 11-6. The stream is made up of the various data segments passed to it. Just like a bus can carry a variety of people, the transport layer protocol can carry a variety of application data types.



NOTE Different references can place specific protocols at different layers. For example, many references place the Transport Layer Security (TLS) protocol in the session layer, while other references place it in the transport layer. Neither placement is right or wrong. The OSI model tries to draw boxes around reality, but some protocols straddle the different layers.

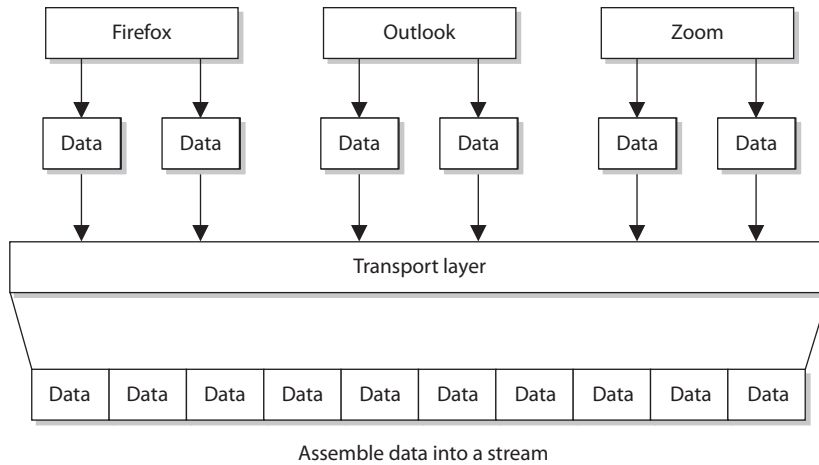


Figure 11-6 TCP formats data from applications into a stream to be prepared for transmission.

Network Layer

The main responsibilities of the *network layer*, layer 3, are to insert information into the packet's header so it can be properly addressed and routed, and then to actually route the packet to its proper destination. In a network, many routes can lead to one destination. The protocols at the network layer must determine the best path for the packet to take. Routing protocols build and maintain their routing tables. These tables are maps of the network, and when a packet must be sent from computer A to computer M, the protocols check the routing table, add the necessary information to the packet's header, and send it on its way.

The protocols that work at this layer do not ensure the delivery of the packets. They depend on the protocols at the transport layer to catch any problems and resend packets if necessary. The Internet Protocol (IP) is the predominant protocol working at the network layer, although other routing and routed protocols work there as well. Some of the other protocols are the Internet Control Message Protocol (ICMP), Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and Internet Group Management Protocol (IGMP). Figure 11-7 shows that a packet can take many routes and that the network layer enters routing information into the header to help the packet arrive at its destination.

Data Link Layer

As we continue down the protocol stack, we are getting closer to the actual transmission channel (e.g., network wire) over which all the data will travel. The network layer has already figured out how to route the packet through the various network devices to its final destination, but we still need to get the data over to the next, directly connected device. This happens at the *data link layer*, layer 2.

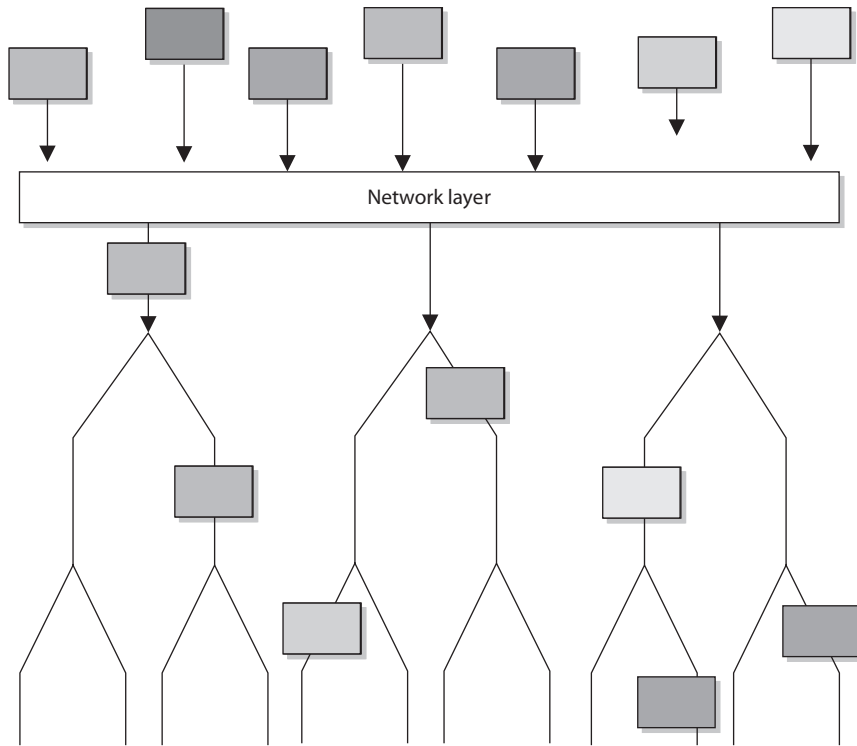


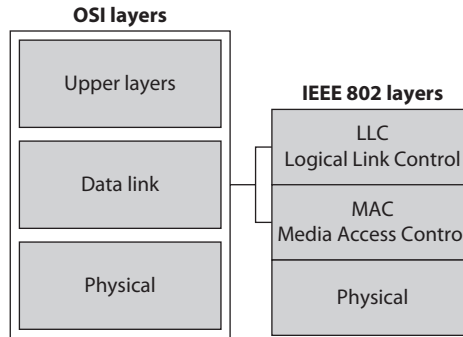
Figure 11-7 The network layer determines the most efficient path for each packet to take.

Different networking technologies, as we will shortly discuss in detail, can use different protocols, network interface cards (NICs), cables, and transmission methods. Each of these components has a different header data format structure, and they interpret electromagnetic signals in different ways. The data link layer is where the network stack knows in what format the data frame must be in order to transmit it properly over Ethernet, wireless, or frame relay links. If the network is an Ethernet network, for example, all the computers will expect packet headers to be a certain length, the flags to be positioned in certain field locations within the header, and the trailer information to be in a certain place with specific fields. Compared to Ethernet, frame relay network technology has different frame header lengths, flag values, and header formats.

The data link layer can be further divided into two functional sublayers: *Logical Link Control (LLC)*, whose job is to interface with the network layer above, and *Media Access Control (MAC)*, which is designed to interface with the physical layer below. Let's drill into this a bit. The LLC sublayer, which is defined in the ISO/IEC 8802-2 standard for Ethernet networks, receives a message (say, an IP packet) from layer 3 and negotiates the manner in which it should be sent out over the network link. This could include keeping track of which layer 3 protocol was used, determining what kind of service (e.g.,

Figure 11-8

The data link layer is made up of two sublayers.



connectionless or connection-oriented) will be used, and performing flow control so the link doesn't get saturated. The LLC sublayer then hands over the data to the MAC sublayer, which encapsulates it into a frame of the right type depending on the networking technology in use by the physical layer. Generally, LLC is implemented in software (as a device driver), while MAC is built in firmware on a physical device. Figure 11-8 shows these two sublayers that make up the data link layer.

As data is passed down the network stack, it has to go from the network layer to the data link layer. The protocol at the network layer does not know if the underlying network is Ethernet, wireless, or frame relay—it does not need to have this type of insight. The protocol at the network layer just adds its header and trailer information to the packet and passes it on to the next layer, which is the LLC sublayer. The LLC sublayer takes care of flow control and error checking. Data coming from the network layer passes down through the LLC sublayer and goes to the MAC sublayer. The technology at the MAC sublayer knows if the network is Ethernet, wireless, or frame relay, so it knows how to put the last header and trailer on the packet before it “hits the wire” for transmission.

Some of the protocols that work at the data link layer are the Point-to-Point Protocol (PPP), Asynchronous Transfer Mode (ATM), Layer 2 Tunneling Protocol (L2TP), Fiber Distributed Data Interface (FDDI), Ethernet, and Token Ring.

Each network technology (Ethernet, wireless, frame relay, and so on) defines the compatible physical transmission type (coaxial, twisted pair, fiber, wireless) that is required to enable network communication. Each network technology also has defined electronic signaling and encoding patterns. For example, if we were transmitting a bit with the value of 1 over an Ethernet network, the MAC sublayer would tell the physical layer to create a +0.5-volt electric signal. In the “language of Ethernet” this means that 0.5 volts is the encoding value for a bit with the value of 1. If the next bit the MAC sublayer receives is 0, the MAC layer would tell the physical layer to transmit 0 volts. The different network types will have different encoding schemes.

NICs bridge the data link and physical layers. Data is passed down through the first six layers and reaches the NIC at the data link layer. Depending on the network technology being used, the NIC encodes the bits at the data link layer, which are then turned into electricity states at the physical layer and placed onto the wire for transmission.



EXAM TIP When the data link layer applies the last header and trailer to the data message, this is referred to as *framing*. The unit of data is now called a *frame*.

Physical Layer

The *physical layer*, layer 1, converts bits into electromagnetic signals for transmission. Signals and voltage schemes have different meanings for different LAN and WAN technologies, as covered earlier. If a user sends data through the radio transceiver on a smartphone, the data format, electrical signals, and control functionality are much different than if that user sends data through an Ethernet NIC and onto an unshielded twisted pair (UTP) wire for LAN communication. The mechanisms that control this data going onto the radio waves, or the UTP wire, work at the physical layer. This layer controls synchronization, data rates, line noise, and transmission techniques. Specifications for the physical layer include the timing of voltage changes, voltage levels, and the physical connectors for electrical, optical, and mechanical transmission.



EXAM TIP To remember all the layers within the OSI model in the correct order, memorize “All People Seem To Need Data Processing.” Remember that you are starting at layer 7, the application layer, at the top.

Functions and Protocols in the OSI Model

For the CISSP exam, you will need to know the functionality that takes place at the different layers of the OSI model, along with specific protocols that work at each layer. The following is a quick overview of each layer and its components.

Application

The protocols at the application layer handle file transfer, virtual terminals, network management, and fulfilling networking requests of applications. A few of the protocols that work at this layer include

- File Transfer Protocol (FTP)
- Network Time Protocol (NTP)
- Simple Mail Transfer Protocol (SMTP)
- Internet Message Access Protocol (IMAP)
- Hypertext Transfer Protocol (HTTP)

Presentation

The services of the presentation layer handle translation into standard formats, data compression and decompression, and data encryption and decryption. No protocols work at this layer, just services. The following lists some of the presentation layer standards:

- American Standard Code for Information Interchange (ASCII)
- Tagged Image File Format (TIFF)

- Joint Photographic Experts Group (JPEG)
- Motion Picture Experts Group (MPEG)
- Musical Instrument Digital Interface (MIDI)

Session

The session layer protocols set up connections between applications; maintain dialog control; and negotiate, establish, maintain, and tear down the communication channel. Some of the protocols that work at this layer include

- Layer 2 Tunneling Protocol (L2TP)
- Network Basic Input Output System (NetBIOS)
- Password Authentication Protocol (PAP)
- Point-to-Point Tunneling Protocol (PPTP)
- Remote Procedure Call (RPC)

Transport

The protocols at the transport layer handle end-to-end transmission and segmentation of a data stream. The following protocols work at this layer:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Stream Control Transmission Protocol (SCTP)
- Resource Reservation Protocol (RSVP)
- QUIC (not an acronym)

Network

The responsibilities of the network layer protocols include internetworking service, addressing, and routing. The following lists some of the protocols that work at this layer:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)

Data Link

The protocols at the data link layer convert data into LAN or WAN frames for transmission and define how a computer accesses a network. This layer is divided into the Logical Link Control (LLC) and the Media Access Control (MAC) sublayers. Some protocols that work at this layer include the following:

- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)
- Serial Line Internet Protocol (SLIP)
- Ethernet (IEEE 802.3)
- Wireless Ethernet (IEEE 802.11)

Physical

Network interface cards and drivers convert bits into electrical signals and control the physical aspects of data transmission, including optical, electrical, and mechanical requirements. The following are some of the standard interfaces at this layer:

- RS/EIA/TIA-422, RS/EIA/TIA-423, RS/EIA/TIA-449, RS/EIA/TIA-485
- 10Base-T, 10Base2, 10Base5, 100Base-TX, 100Base-FX, 100Base-T, 1000Base-T, 1000Base-SX
- Integrated Services Digital Network (ISDN)
- Digital subscriber line (DSL)
- Synchronous Optical Networking (SONET)

Tying the Layers Together

The OSI model is used as a framework for many network-based products and is used by many types of vendors. Various types of devices and protocols work at different parts of this seven-layer model. The main reason that a Cisco switch, a Microsoft web server, a Barracuda firewall, and a Belkin wireless access point can all communicate properly on one network is because they all work within the OSI model. They do not have their own individual ways of sending data; they follow a standardized manner of communication, which allows for interoperability and allows a network to be a network. If a product does not follow the OSI model, it will not be able to communicate with other devices on the network because the other devices will not understand its proprietary way of communicating.

The different device types work at specific OSI layers. For example, computers can interpret and process data at each of the seven layers, but routers can understand information only up to the network layer because a router's main function is to route packets, which does not require knowledge about any further information within the packet. A router peels back the header information until it reaches the network layer data, where the routing and IP address information is located. The router looks at this information to make its decisions on where the packet should be routed. Bridges and switches understand only up to the data link layer, and repeaters understand traffic only at the physical layer. So if you hear someone mention a "layer 3 device," the person is referring to a device that works at the network layer. A "layer 2 device" works at the data link layer. Figure 11-9 shows what layer of the OSI model each type of device works within.

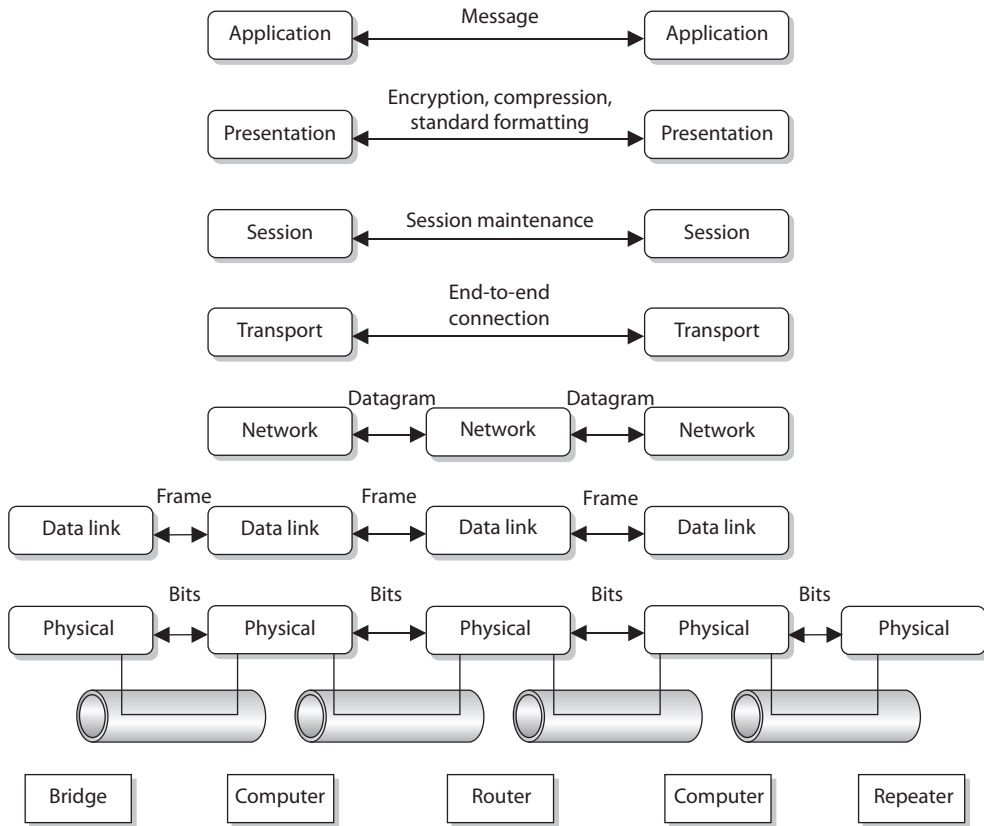


Figure 11-9 Each device works at a particular layer within the OSI model.



NOTE Some techies like to joke that all computer problems reside at layer 8. The OSI model does not have an eighth layer, and what these people are referring to is the user of a computer. So if someone states that there is a problem at layer 8, this is code for “the user is the problem.”

Let’s walk through an example. You just logged into a website and the landing page includes an image that your web browser automatically requests from the server. The web server now has to move this file over the network to your computer, so it generates an HTTP response, which includes a handle to the file, and hands it over to the presentation layer for encoding. HTTP doesn’t directly handle binary files like JPEG images, so the file must first be “serialized” or converted to a sequence of printable characters, which is what the presentation layer does.

Once the response is “presentable” (pardon the pun), the presentation layer hands it to the session layer to figure out which of the clients currently communicating with

the server should receive this image; the session layer figures out that it is for you and that you have been authenticated. The session layer then forwards the response to the transport layer, telling it the connection (or network socket) on which it should go out. Based on this connection identifier, the transport layer encapsulates the payload it just got from the session layer in a TCP datagram (which is what HTTP runs on), writes the protocol and port numbers in its header, and passes it on to layer 3.

The image you're requesting is quite large, so the network layer breaks it up into a numbered sequence of chunks, encapsulates each in its own IP packet, figures out how to route the packets, and then hands each to the data link layer. Layer 2 takes each packet and, based on its destination IP address, determines the next hop on its journey back to you, which is probably the DMZ firewall. It encapsulates the packet into an Ethernet frame, writes the MAC address of the firewall in it, and sends it down to the physical layer, which turns the 1's and 0's into electric currents on the network cable. And, just like that, your image starts its journey toward your screen.

Local Area Networks

Now that we've taken a quick look down all seven layers of the OSI model, let's circle back and take a more detailed look from the ground up. If you connect two general-purpose computers that are right next to each other, you create a local area network. A *local area network (LAN)* is a group of interconnected computers in close physical proximity to each other. A LAN is the basic building block for most organizational networks. As we'll shortly see, there are multiple ways to build LANs both physically and logically. In the following sections, we'll discuss the various technologies that allow us to physically interconnect devices.

Network Topology

The arrangement of computers and devices is called a *network topology*. Topology refers to the manner in which a network is physically connected and shows the layout of resources and systems. A difference exists between the physical network topology and the logical topology. A network can be configured as a physical star but work logically as a ring, as in the Token Ring technology.

The best topology for a particular network depends on such things as how nodes are supposed to interact; which protocols are used; the types of applications that are available; the reliability, expandability, and physical layout of a facility; existing wiring; and the technologies implemented. The wrong topology or combination of topologies can negatively affect the network's performance, productivity, and growth possibilities.

This section describes the basic types of network topologies. Most networks are much more complex and are usually implemented using a combination of topologies.

Bus Topology

In a simple *bus topology*, a single cable runs the entire length of the network. Nodes are attached to the network through drop points on this cable. Data communications transmit the length of the medium, and each packet transmitted has the capability of being

“looked at” by all nodes. Each node decides to accept or ignore the packet, depending upon the packet’s destination address.

Bus topologies are of two main types: linear and tree. The *linear bus topology* has a single cable with nodes attached. A *tree topology* has branches from the single cable, and each branch can contain many nodes. In simple implementations of a bus topology, if one workstation fails, other systems can be negatively affected because of the degree of interdependence. In addition, because all nodes are connected to one main cable, the cable itself becomes a potential single point of failure.

Bus topologies were common years ago on the first Ethernet networks. Today, you are very unlikely to encounter them in local area networks. However, this topology is prevalent in vehicular networks, of which the Controller Area Network (CAN) bus is by far the most popular standard.

Star Topology

In a *star topology*, all nodes connect to a central device such as a switch. Each node has a dedicated link to the central device. The central device needs to provide enough throughput that it does not become a detrimental bottleneck for the network as a whole. Because a central device is required, it is a potential single point of failure, so redundancy may need to be implemented. Switches can be configured in flat or hierarchical implementations so larger organizations can use them.

When one workstation fails on a star topology, it does not affect other systems, as in the bus topologies. In a star topology, each system is not as dependent on others as it is dependent on the central connection device. This topology generally requires less cabling than other types of topologies. As a result, cut cables are less likely, and detecting cable problems is an easier task.

Mesh Topology

In a *mesh topology*, all systems and resources are connected to each other in a way that does not follow the uniformity of the previous topologies, as shown in Figure 11-10. This arrangement is usually a network of interconnected routers and switches that provides

Figure 11-10

In a mesh topology, each node is connected to all other nodes, which provides for redundant paths.

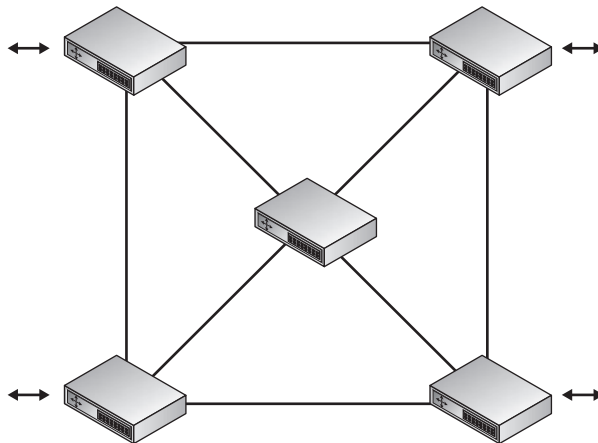
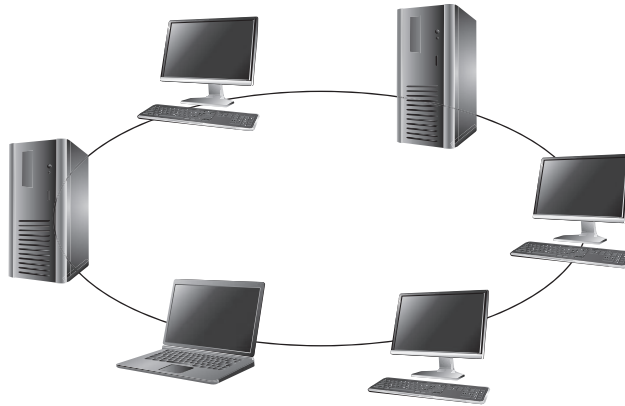


Figure 11-11
A ring topology
forms a closed-
loop connection.



multiple paths to all the nodes on the network. In a full mesh topology, every node is directly connected to every other node, which provides a great degree of redundancy. A typical Internet of Things (IoT) home automation network using ZigBee is an example of a full mesh topology. In a partial mesh topology, every node is not directly connected. The Internet is an example of a partial mesh topology.

Ring Topology

A *ring topology* has a series of devices connected by unidirectional transmission links, as shown in Figure 11-11. These links form a closed loop and do not connect to a central system, as in a star topology. In a physical ring formation, each node is dependent upon the preceding nodes. In simple networks, if one system fails, all other systems could be negatively affected because of this interdependence. Compensating for such failures, coupled with the difficulty of installing a ring network in the first place, makes ring topologies more expensive than other topologies. Although you have to know about ring topologies for the CISSP exam, you are extremely unlikely to find them in the real world. Still, who knows? Maybe one day they'll make a comeback.

Network Topologies Summary

A summary of the different network topologies and their important characteristics is provided in Table 11-1.

Medium Access Control Mechanisms

The physical topology of a network is the lower layer, or foundation, of a network. It determines what type of physical media will be used to connect network devices and what these connections look like. *Medium access control (MAC)* mechanisms deal with how computer systems communicate over these media and are built into the network interfaces. MAC mechanisms set up the rules of how computers will communicate on a LAN, how errors are handled, the maximum transmission unit (MTU) size of frames, and much more. These rules enable all computers and devices to communicate and

Topology	Characteristics	Problems
Bus	Uses a linear, single cable for all computers attached. All traffic travels the full cable and can be viewed by all other computers.	If one station experiences a problem, it can negatively affect surrounding computers on the same cable.
Ring	All computers are connected by a unidirectional transmission link, and the cable is in a closed loop.	If one station experiences a problem, it can negatively affect surrounding computers on the same ring.
Star	All computers are connected to a central device, which provides more resilience for the network.	The central device is a single point of failure.
Tree	A bus topology with branches off of the main cable.	Multiple single points of failure.
Mesh	Computers are connected to each other, which provides redundancy.	Requires more expense in cabling and extra effort to track down cable faults.

Table 11-1 Summary of Network Topologies

recover from problems, and enable users to be productive in accomplishing their networking tasks. Each participating entity needs to know how to communicate properly so all other systems will understand the transmissions, instructions, and requests.

LAN MAC mechanisms reside at the data link layer of the OSI model. Remember that as a message is passed down through a network stack, it is encapsulated by the protocols and services at each layer. When the data message reaches the data link layer, the protocol at this layer adds the necessary headers and trailers that will allow the message to traverse a specific type of network (Ethernet, Token Ring, FDDI, etc.)

No matter what type of medium access technology is being used, the main resource that has to be shared by all systems and devices on the network is the network transmission channel. This transmission channel could be coaxial cabling, UTP cabling, optical fibers, or free space (e.g., using radio waves). There must be methods in place to make sure that each system gets its fair share of access to the channel, that the system's data is not corrupted during transmission, and that there is a way to control traffic in peak times. Let's take a closer look at the three most common approaches to MAC: carrier sense multiple access, Token Ring, and polling.

Carrier Sense Multiple Access

By far, the most common approach to MAC is called *carrier sense multiple access (CSMA)*, which provides a standard way to access the shared medium, communicate, and recover from any errors that may occur. A transmission is called a *carrier*, so if a computer is transmitting frames, it is performing a carrier activity. Despite their best efforts, networked computers sometimes talk over each other, which creates what is called a *collision*. Think about it: it takes time for electromagnetic signals to move from point A (where one transmitter sits) to point B (where another computer also wants to transmit). Both computers listen to the medium and, detecting no other traffic on it, simultaneously transmit their messages.

Sometime later, these messages meet each other on the medium and *collide*, or corrupt each other. There are two variants of CSMA with which you should be familiar: collision detection (CSMA/CD) and collision avoidance (CSMA/CA).

When computers use the *carrier sense multiple access with collision detection* (CSMA/CD) protocol, they monitor the transmission activity, or carrier activity, on the wire so they can determine when would be the best time to transmit data. Each node monitors the wire continuously and waits until the wire is free before it transmits its data. As an analogy, consider several people gathered in a group talking here and there about this and that. If a person wants to talk, she usually listens to the current conversation and waits for a break before she proceeds to talk. If she does not wait for the first person to stop talking, she will be speaking at the same time as the other person, and the people around them may not be able to understand fully what each is trying to say. If a computer puts frames on the wire and its frames collide with another computer's frames, it will abort its transmission and alert all other stations that a collision just took place. All stations will start a random collision timer to force a delay before they attempt to transmit data. This random collision timer is called the *back-off algorithm*. CSMA/CD was important in the early days of Ethernet LANs when hubs and bridges were common, but is now largely deprecated.

The other variant of CSMA is *carrier sense multiple access with collision avoidance* (CSMA/CA), in which all stations with data to transmit first check the medium to see if it's quiet. If it is, they send their data. If it isn't, they start a random timer before they check again. In some implementations, the station wishing to send data first sends a request to send (RTS) frame to a controller or the destination and then waits for a clear to send (CTS) frame before transmitting its data. CSMA/CA is most commonly used in wireless networks.

Token Passing

Some of North America's indigenous people devised a clever way to ensure only one of them spoke (and was not interrupted) at meetings. It was called the talking stick, and only the one holding it was allowed to speak. Once that person was done, they would put it down, allowing the next individual to pick it up and speak. This token ensured that all participants had a chance to speak uninterrupted. Some MAC technologies also use tokens, which are 24-bit control frames used to control which computers communicate at what intervals. The token is passed from computer to computer, and only the computer that has the token can actually put frames onto the wire. The token grants a computer the right to communicate. The token contains the data to be transmitted and source and destination address information. When a system has data it needs to transmit, it has to wait to receive the token. The computer then connects its message to the token and puts it on the wire. Each computer checks this message to determine whether it is addressed to it, which continues until the destination computer receives the message. The destination computer makes a copy of the message and flips a bit to tell the source computer it did indeed get its message. Once this gets back to the source computer, it removes the frames from the network. The destination computer makes a copy of the message, but only the originator of the message can remove the message from the token and the network.

Carrier-Sensing and Token-Passing Access Methods

Overall, carrier-sensing access methods are faster than token-passing access methods, but the former do have the problem of collisions. A network segment with many devices can cause too many collisions and slow down the network's performance. Token-passing technologies do not have problems with collisions, but they do not perform at the speed of carrier-sensing technologies. Network routers can help significantly in isolating the network resources for both the CSMA/CD and the token-passing methods.

If a computer that receives the token does not have a message to transmit, it sends the token to the next computer on the network. An empty token has a header, data field, and trailer, but a token that has an actual message has a new header, destination address, source address, and a new trailer. This type of media-sharing method is used by Token Ring and FDDI technologies.



NOTE Some applications and network protocols work better if they can communicate at determined intervals, instead of “whenever the data arrives.” In token-passing technologies, traffic arrives in this type of deterministic nature because not all systems can communicate at one time; only the system that has control of the token can communicate.



NOTE When there is just one transmission medium (i.e., UTP cable) that has to be shared by all nodes and devices in a network, this is referred to as a *contention-based* environment. Each system has to “compete” to use the transmission line, which can cause contention.

Collision and Broadcast Domains

As previously indicated, a collision occurs on Ethernet networks when two computers transmit data at the same time. Other computers on the network detect this collision because the overlapping signals of the collision increase the voltage of the signal above a specific threshold. The more devices on a contention-based network, the more likely collisions will occur, which increases network latency (data transmission delays). A *collision domain* is a group of devices that are contending, or competing, for the same shared communication medium. For example, all devices that are connected to a particular wireless access point (WAP) belong to the same collision domain.



NOTE Collision domains used to be a problem in wired networks back when hubs were more prevalent than switches. You may still come across these examples, but they are exceptionally rare in real networks nowadays.

The compromised system will not be able to listen to traffic on other broadcast and collision domains, and this can greatly reduce the amount of traffic and information available to an attacker.

Polling

The third type of media-sharing method, besides CSMA and token passing, is polling. *Polling* is a medium access control mechanism that relies on a primary station that periodically polls all others in its collision domain. Each polled device responds by stating whether or not it has anything to send. In some implementations, the secondary stations (devices) can also let the primary station know how much data they want to send, where it's going to, and how urgent it is. After polling all secondary stations, the primary station allocates the channel according to whatever policy it follows. It could, for example, prioritize traffic from one station, or type, or evenly divide the channel among all stations that asked to access it.

The main thing to remember about polling MAC is that each device needs to wait until the primary station completes a poll and then tells that device how much of the channel it can use. Only then can a secondary station transmit data. This approach is very uncommon in LANs, though it is used in some wireless networks. Polling is much more common in wide area networks.

Layer 2 Protocols

Now that we've discussed LAN topologies and MAC mechanisms, let's look at how these are implemented in reality. For the CISSP exam, you should know three layer 2 protocols: Ethernet, Token Ring, and FDDI. In reality, you are almost certainly to only use Ethernet and its wireless cousin Wi-Fi. Token Ring and FDDI are exceptionally rare in LANs, though FDDI is still used as the backbone of metropolitan area networks (MANs).

Ethernet

Ethernet is a set of technologies that enables several devices to communicate on the same network. Ethernet usually uses a bus or star topology. If a linear bus topology is used, all devices connect to one cable. If a star topology is used, each device is connected to a cable that is connected to a centralized device, such as a switch. Ethernet was developed in the 1970s, became commercially available in 1980, and was officially defined through the IEEE 802.3 standard.

Ethernet has seen quite an evolution in its short history, from purely coaxial cable installations that worked at 10 Mbps to mostly twisted-pair cable that works at speeds of 100 Mbps, 1,000 Mbps (1 Gbps), and up to 40 Gbps.

Ethernet is defined by the following characteristics:

- Contention-based technology (all resources use the same shared communication medium)
- Uses broadcast and collision domains

Table 11-2 Ethernet Implementation Types	Ethernet Type	IEEE Standard	Cable Type (minimum)	Speed
	10Base-T	802.3i-1990	Cat3 UTP	10 Mbps
	100Base-TX, Fast Ethernet	802.3u-1995	Cat5 UTP	100 Mbps
	1000Base-T, Gigabit Ethernet	802.3ab-1999	Cat5 UTP	1,000 Mbps
	10GBase-T	802.3an-2006	Cat6a UTP	10,000 Mbps

- Uses the CSMA access method
- Supports full-duplex communication
- Can use coaxial, twisted-pair, or fiber-optic cabling types, but most commonly uses UTP cables
- Is defined by the IEEE 802.3 family of standards

Ethernet addresses how computers share a common network and how they deal with collisions, data integrity, communication mechanisms, and transmission controls. These are the common characteristics of Ethernet, but Ethernet does vary in the type of cabling schemes and transfer rates it can supply. Several types of Ethernet implementations are available, as outlined in Table 11-2.

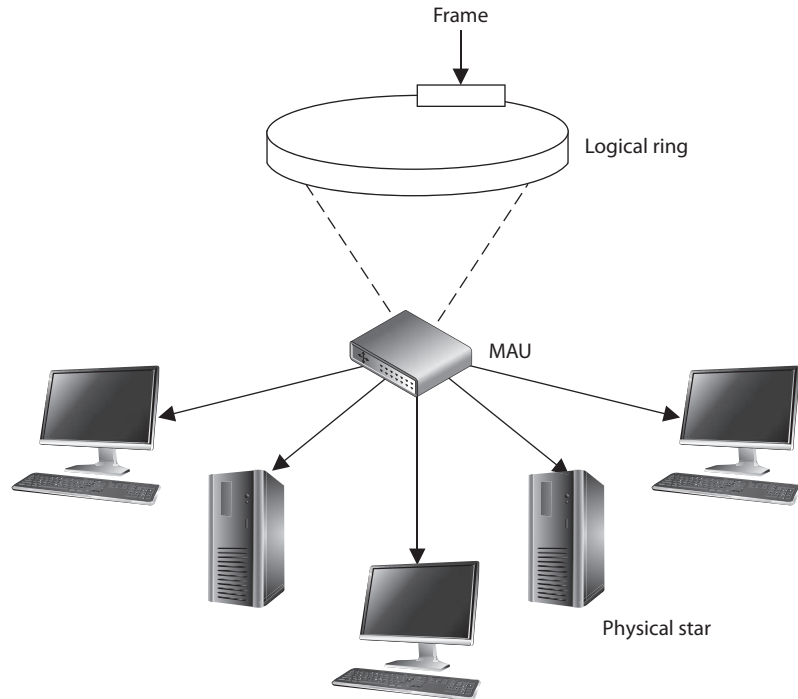
10Base-T was considered heaven-sent when it first arrived on the networking scene, but soon many users were demanding more speed and power. It is now considered a legacy standard and is very rarely seen in organizational networks. There are ongoing efforts, however, to develop variants of this obsolete standard for use in automotive and IoT applications. Today, the most widely deployed type of Ethernet in organizational networks is 1000Base-T.

Token Ring

The *Token Ring* technology was originally developed by IBM and then defined by the IEEE 802.5 standard. At first, Token Ring technology had the ability to transmit data at 4 Mbps. Later, it was improved to transmit at 16 Mbps. It uses a token-passing technology with a star-configured topology. The *ring* part of the name pertains to how the signals travel, which is in a logical ring. Each computer is connected to a central hub, called a *Multistation Access Unit (MAU)*. Physically, the topology can be a star, but the signals and transmissions are passed in a logical ring.

As previously described, *token-passing technology* is one in which a device cannot put data on the network wire without having possession of a *token*, a control frame that travels in a logical circle and is “picked up” when a system needs to communicate. This is different from Ethernet, in which all the devices attempt to communicate at the same time. This is why Ethernet is referred to as a “chatty protocol” and has collisions. Token Ring does not endure collisions, since only one system can communicate at a time, but this also means communication takes place more slowly compared to Ethernet.

Figure 11-13
A Token Ring
network



Token Ring employs a couple of mechanisms to deal with problems that can occur on this type of network. The *active monitor* mechanism removes frames that are continuously circulating on the network. This can occur if a computer locks up or is taken offline for one reason or another and cannot properly receive a token destined for it. With the *beaconing* mechanism, if a computer detects a problem with the network, it sends a beacon frame. This frame generates a failure domain, which is between the computer that issued the beacon and its neighbor downstream. The computers and devices within this failure domain will attempt to reconfigure certain settings to try to work around the detected fault. Figure 11-13 depicts a Token Ring network in a physical star configuration. Token Ring networks were popular in the 1980s and 1990s, and although some are still around, Ethernet is much more popular.

FDDI

Fiber Distributed Data Interface (FDDI) technology, developed by the American National Standards Institute (ANSI), is a high-speed, token-passing, medium access technology. FDDI has a data transmission speed of up to 100 Mbps and is usually used as a backbone network using fiber-optic cabling. FDDI also provides fault tolerance by offering a second counter-rotating fiber ring. The primary ring has data traveling clockwise and is used for regular data transmission. The second ring transmits data in a counterclockwise fashion and is invoked only if the primary ring goes down. Sensors watch the primary ring and, if it goes down, invoke a ring *wrap* so the data will be diverted to the second

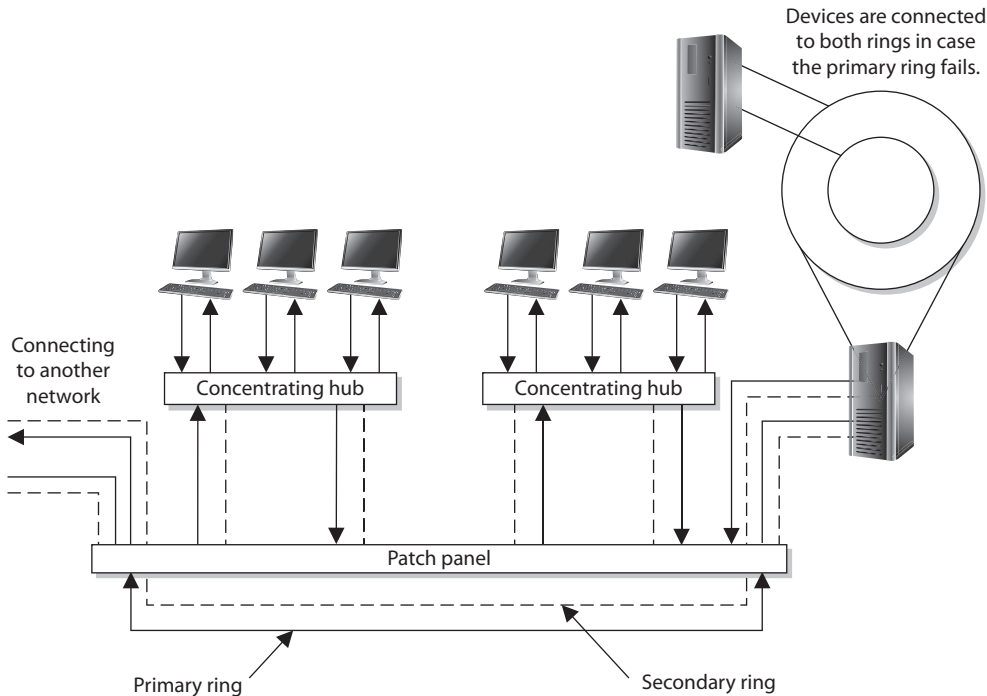


Figure 11-14 FDDI rings can be used as backbones to connect different LANs.

ring. Each node on the FDDI network has relays that are connected to both rings, so if a break in the ring occurs, the two rings can be joined.

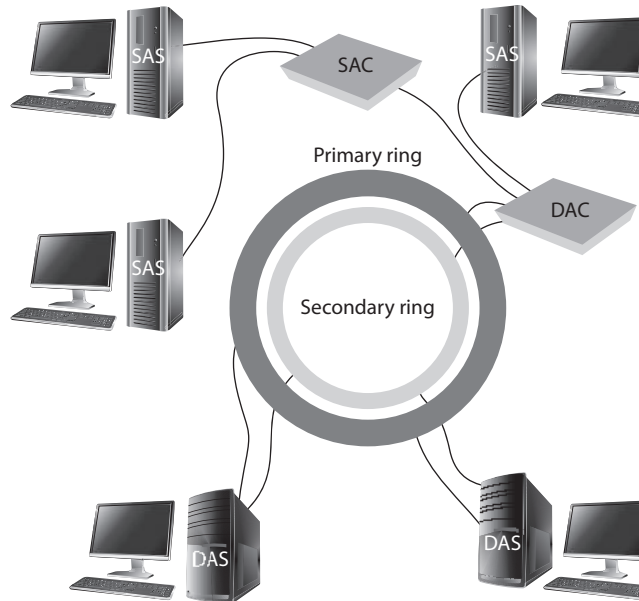
When FDDI is used as a backbone network, it usually connects several different networks, as shown in Figure 11-14.

Before Fast Ethernet and Gigabit Ethernet hit the market, FDDI was used mainly as campus and service provider backbones. Because FDDI can be employed for distances up to 100 kilometers, it was often used in MANs. The benefit of FDDI is that it can work over long distances and at high speeds with minimal interference. It enables several tokens to be present on the ring at the same time, causing more communication to take place simultaneously, and it provides predictable delays that help connected networks and devices know what to expect and when.



NOTE A version of FDDI, Copper Distributed Data Interface (CDDI), can work over UTP cabling.

Figure 11-15
FDDI device
types



Devices that connect to FDDI rings fall into one of the following categories:

- **Single-attachment station (SAS)** Attaches to only one ring (the primary) through a concentrator
- **Dual-attachment station (DAS)** Has two ports and each port provides a connection for both the primary and the secondary rings
- **Single-attached concentrator (SAC)** Concentrator that connects an SAS device to the primary ring
- **Dual-attached concentrator (DAC)** Concentrator that connects DAS, SAS, and SAC devices to both rings

The different FDDI device types are illustrated in Figure 11-15.



NOTE Ring topologies are considered deterministic, meaning that the rate of the traffic flow can be predicted. Since traffic can only flow if a token is in place, the maximum time that a node will have to wait to receive traffic can be determined. This can be beneficial for time-sensitive applications.

Layer 2 Network Protocol Summary

Table 11-3 sums up the important characteristics of the technologies described in the preceding sections.

LAN Implementation	Standard	Characteristics
Ethernet	IEEE 802.3	Uses broadcast and collision domains. Uses CSMA medium access control method. Can use coaxial, twisted-pair, or fiber-optic media. Transmission speeds of 10 Mbps to 10 Gbps.
Token Ring	IEEE 802.5	Token-passing media access method. Transmission speeds of 4 to 16 Mbps. Uses an active monitor and beaconing. Effectively defunct.
FDDI	ANSI standard Based on IEEE 802.4	Dual counter-rotating rings for fault tolerance. Transmission speeds of 100 Mbps. Operates over long distances at high speeds and is therefore used as a backbone. CDDI works over UTP. Very rarely seen in the enterprise.

Table 11-3 LAN Media Access Methods

Transmission Methods

A packet may need to be sent to only one workstation, to a set of workstations, or to all workstations on a particular subnet. If a packet needs to go from the source computer to one particular system, a *unicast* transmission method is used. If the packet needs to go to a specific group of systems, the sending system uses the *multicast* method. If a system wants all computers on its subnet to receive a message, it uses the *broadcast* method.

Unicast is pretty simple because it has a source address and a destination address. The data goes from point A to point B, it is a one-to-one transmission, and everyone is happy. Multicast is a bit different in that it is a one-to-many transmission. Multicasting enables one computer to send data to a selective group of computers. A good example of multicasting is tuning into a radio station on a computer. Some computers have software that enables the user to determine whether she wants to listen to rock, Latin, or a talk radio station, for example. Once the user selects one of these genres, the software must tell the NIC driver to pick up not only packets addressed to its specific MAC address but also packets that contain a specific multicast address.

The difference between broadcast and multicast is that in a broadcast one-to-all transmission, everyone gets the data, whereas in a multicast, only certain nodes receive the data. So how does a server three states away multicast to one particular computer on a specific network and no other networks in between? Suppose a user tunes in to her favorite Internet radio station. An application running on her computer (say, a web browser) has to tell her local router she wants to get frames with this particular multicast address passed her way. The local router must tell the router upstream, and this process continues so each router between the source and destination knows where to pass this multicast data. This ensures that the user can get her rock music without other networks being bothered with this extra data.

IPv4 multicast protocols use a Class D address (224.0.0.0 to 239.255.255.255), which is a special address space reserved for multicasting. IPv6 multicast addresses start with eight 1's (that is, 1111 1111). Multicasting can be used to send out information; multimedia data; and even real-time video, music, and voice clips.

Internet Group Management Protocol (IGMP) is used to report multicast group memberships to routers. When a user chooses to accept multicast traffic, she becomes a member of a particular multicast group. IGMP is the mechanism that allows her computer to inform the local routers that she is part of this group and to send traffic with a specific multicast address to her system. IGMP can be used for online streaming video and gaming activities. The protocol allows for efficient use of the necessary resources when supporting these types of applications.

Like most protocols, IGMP has gone through a few different versions, each improving upon the earlier one. In version 1, multicast agents periodically send queries to systems on the network they are responsible for and update their databases, indicating which system belongs to which group membership. Version 2 provides more granular query types and allows a system to signal to the agent when it wants to leave a group. Version 3 allows the systems to specify the sources it wants to receive multicast traffic from. Each version is backward-compatible because versions 1 and 2 are still in use in legacy equipment.



NOTE The previous statements are true pertaining to IPv4. IPv6 is more than just an upgrade to the original IP protocol; it functions differently in many respects, including how it handles multicasting, which has caused many interoperability issues and delay in its full deployment.

Layer 2 Security Standards

As frames pass from one network device to another device, attackers could sniff the data; modify the headers; redirect the traffic; spoof traffic; carry out man-in-the-middle attacks, DoS attacks, and replay attacks; and indulge in other malicious activities. It has become necessary to secure network traffic at the frame level, which is layer 2 of the OSI model.

802.1AE is the IEEE MAC Security (MACSec) standard, which defines a security infrastructure to provide data confidentiality, data integrity, and data origin authentication. Where a VPN connection provides protection at the higher networking layers, MACSec provides hop-by-hop protection at layer 2, as shown in Figure 11-16.

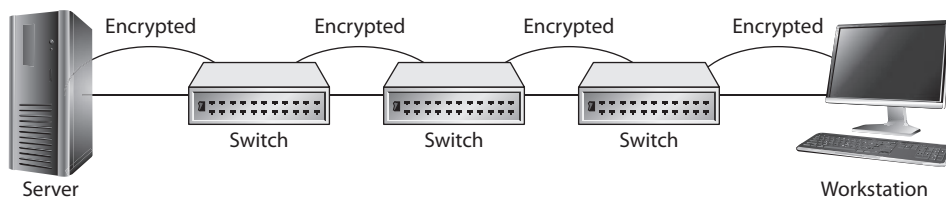


Figure 11-16 MACSec provides layer 2 frame protection.

MACSec integrates security protection into wired Ethernet networks to secure LAN-based traffic. Only authenticated and trusted devices on the network can communicate with each other. Unauthorized devices are prevented from communicating via the network, which helps prevent attackers from installing rogue devices and redirecting traffic between nodes in an unauthorized manner. When a frame arrives at a device that is configured with MACSec, the MACSec Security Entity (SecY) decrypts the frame if necessary and computes an integrity check value (ICV) on the frame and compares it with the ICV that was sent with the frame. If the ICVs match, the device processes the frame. If they do not match, the device handles the frame according to a preconfigured policy, such as discarding it.

The *IEEE 802.1AR* standard specifies unique per-device identifiers (DevID) and the management and cryptographic binding of a device (router, switch, access point) to its identifiers. A verifiable unique device identity allows establishment of the trustworthiness of devices, and thus facilitates secure device provisioning.

As a security administrator you really only want devices that are allowed on your network to be plugged into your network. But how do you properly and uniquely identify devices? The manufacturer's serial number is not available for a protocol to review. MAC addresses, hostnames, and IP addresses are easily spoofed. 802.1AR defines a globally unique per-device secure identifier cryptographically bound to the device through the use of public cryptography and digital certificates. These unique hardware-based credentials can be used with the Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) authentication framework. Each device that is compliant with IEEE 802.1AR comes with a single built-in initial secure device identity (iDevID). The iDevID is an instance of the general concept of a DevID, which is intended to be used with authentication protocols such as EAP, which is supported by IEEE 802.1X.

So 802.1AR provides a unique ID for a device. 802.1AE provides data encryption, integrity, and origin authentication functionality. 802.1AF carries out key agreement functions for the session keys used for data encryption. Each of these standards provides specific parameters to work within an 802.1X EAP-TLS framework, as shown in Figure 11-17.

As Figure 11-17 shows, when a new device is installed on the network, it cannot just start communicating with other devices, receive an IP address from a Dynamic Host Configuration Protocol (DHCP) server, resolve names with the Domain Name System (DNS) server, and so on. The device cannot carry out any network activity until it is authorized to do so. So 802.1X port authentication kicks in, which means that only authentication data is allowed to travel from the new device to the authenticating server. The authentication data is the digital certificate and hardware identity associated with that device (802.1AR), which is processed by EAP-TLS. Once the device is authenticated, usually by a Remote Authentication Dial-In User Server (RADIUS) server, encryption keying material is negotiated and agreed upon between surrounding network devices. Once the keying material is installed, then data encryption and frame integrity checking can take place (802.1AE) as traffic goes from one network device to the next.

These IEEE standards are new and evolving and at different levels of implementation by various vendors. One way the unique hardware identity and cryptographic material

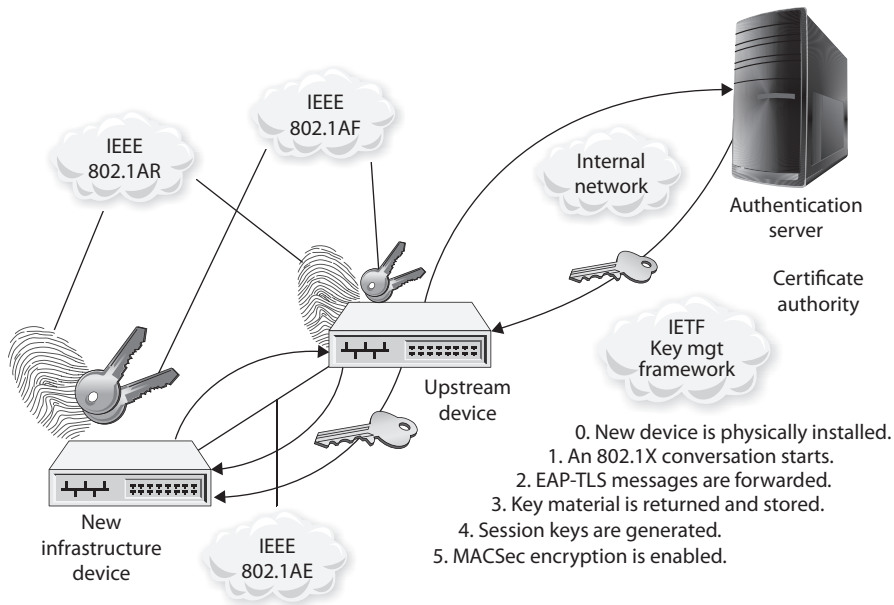


Figure 11-17 Layer 2 security protocols

are embedded in new network devices is through the use of a Trusted Platform Module (TPM; described in Chapter 9).

Internet Protocol Networking

Unless your network consists of only a few devices, isolated from the Internet (and what good is that?), you will need to move from layer 2 into layer 3 and above to do anything meaningful. Recall that the data link layer is concerned with exchanging data between devices that are directly connected to each other (in other words, in the same collision domain). Beyond that, we need layer 3 (network) and 4 (transport) protocols, such as TCP/IP.

The *Transmission Control Protocol/Internet Protocol (TCP/IP)* is a suite of protocols that governs the way data travels from one device to another. IP is a network layer protocol and provides datagram routing services. IP's main task is to support internetwork addressing and packet routing. It is a connectionless protocol that envelops data passed to it from the transport layer. The IP protocol addresses the datagram with the source and destination IP addresses. The protocols within the TCP/IP suite work together to break down the data passed from the application layer into pieces that can be moved along a network. They work with other protocols to transmit the data to the destination

IP

IP is a connectionless protocol that provides the addressing and routing capabilities for each package of data. The data, IP, and network relationship can be compared to the relationship between a letter and the postal system:

- Data = Letter
- IP = Addressed envelope
- Network = Postal system

The message is the letter, which is enveloped and addressed by IP, and the network and its services enable the message to be sent from its origin to its destination, like the postal system.

computer and then reassemble the data back into a form that the application layer can understand and process.

Two main protocols work at the transport layer: TCP and UDP. *TCP* is a reliable and *connection-oriented protocol*, which means it ensures packets are delivered to the destination computer. If a packet is lost during transmission, TCP has the ability to identify this issue and resend the lost or corrupted packet. TCP also supports packet sequencing (to ensure each and every packet was received), flow and congestion control, and error detection and correction. *UDP*, on the other hand, is a *best-effort* and *connectionless protocol*. It has neither packet sequencing nor flow and congestion control, and the destination does not acknowledge every packet it receives.

TCP

TCP is referred to as a connection-oriented protocol because before any user data is actually sent, handshaking takes place between the two systems that want to communicate. Once the handshaking completes successfully, a virtual connection is set up between the two systems. UDP is considered a connectionless protocol because it does not go through these steps. Instead, UDP sends out messages without first contacting the destination computer and does not know if the packets were received properly or dropped. Figure 11-18 shows the difference between a connection-oriented protocol and a connectionless protocol.

UDP and TCP sit together on the transport layer, and developers can choose which to use when developing applications. Many times, TCP is the transport protocol of choice because it provides reliability and ensures the packets are delivered. TCP provides a full-duplex, reliable communication mechanism, and if any packets are lost or damaged, they are re-sent; however, TCP requires a lot of system overhead compared to UDP.

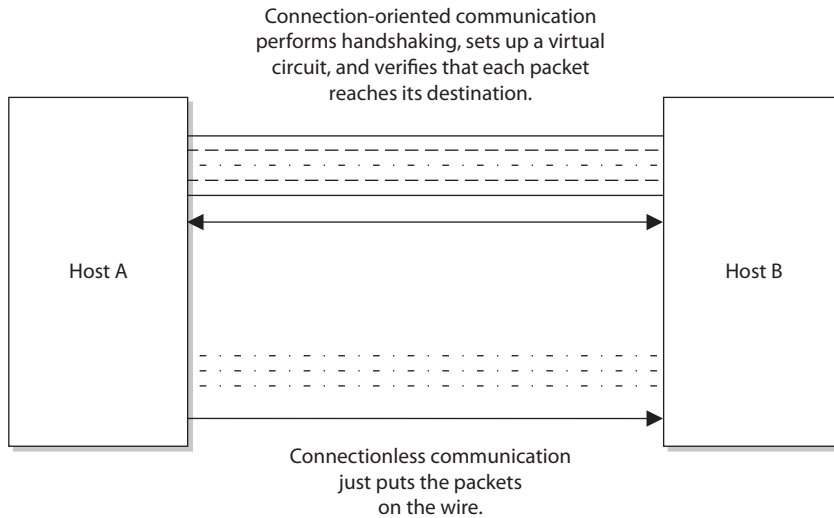


Figure 11-18 Connection-oriented protocol vs. connectionless protocol functionality

If developers know that data being dropped during transmission is not detrimental to the application, they may choose to use UDP because it is faster and requires fewer resources. For example, UDP is a better choice than TCP when a server sends status information to all listening nodes on the network. A node will not be negatively affected if, by some chance, it did not receive this status information, because the information will be re-sent every 60 seconds.

UDP and TCP are transport protocols that applications use to get their data across a network. They both use *ports* to communicate with upper OSI layers and to keep track of various conversations that take place simultaneously. The ports are also the mechanism used to identify how other computers access services. When a TCP or UDP message is formed, source and destination ports are contained within the header information along with the source and destination IP addresses. The combination of protocol (TCP or UDP), port, and IP address makes up a *socket*, and is how packets know where to go (by the address) and how to communicate with the right service or protocol on the other computer (by the port number). The IP address acts as the doorway to a computer, and the port acts as the doorway to the actual protocol or service. To communicate properly, the packet needs to know these doors. Figure 11-19 shows how packets communicate with applications and services through ports.

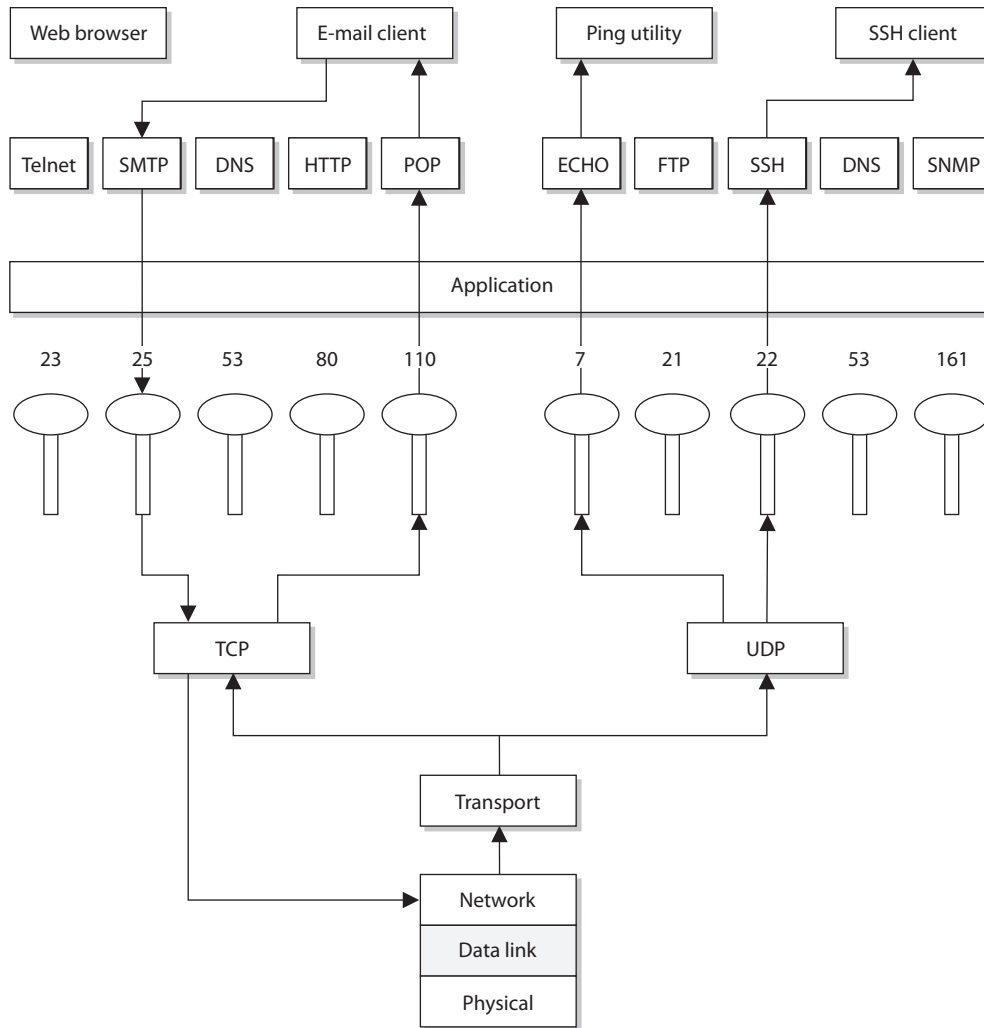


Figure 11-19 The packet can communicate with upper-layer protocols and services through a port.

The difference between TCP and UDP can also be seen in the message formats. Because TCP offers more services than UDP, it must contain much more information within its packet header format, as shown in Figure 11-20. Table 11-4 lists the major differences between TCP and UDP.

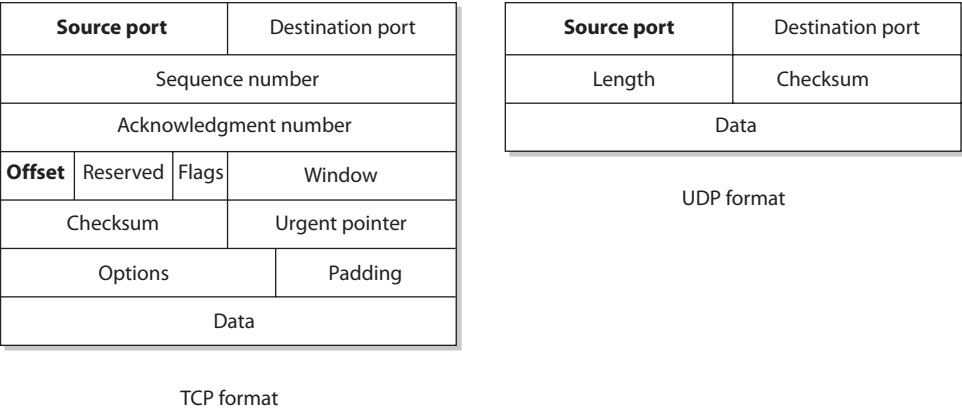


Figure 11-20 TCP carries a lot more information within its segment because it offers more services than UDP.

Property	TCP	UDP
Reliability	Ensures that packets reach their destinations, returns ACKs when packets are received, and is a reliable protocol.	Does not return ACKs and does not guarantee that a packet will reach its destination. Is an unreliable protocol.
Connection	Connection-oriented. It performs handshaking and develops a virtual connection with the destination computer.	Connectionless. It does no handshaking and does not set up a virtual connection.
Packet sequencing	Uses sequence numbers within headers to make sure each packet within a transmission is received.	Does not use sequence numbers.
Congestion controls	The destination computer can tell the source if it is overwhelmed and thus slow the transmission rate.	The destination computer does not communicate back to the source computer about flow control.
Usage	Used when reliable delivery is required. Intended for relatively small amounts of data transmission.	Used when reliable delivery is not required and high volumes of data need to be transmitted, such as in streaming video and status broadcasts.
Speed and overhead	Uses a considerable amount of resources and is slower than UDP.	Uses fewer resources and is faster than TCP.

Table 11-4 Major Differences Between TCP and UDP

Port Types

Port numbers up to 1023 (0 to 1023) are called *well-known ports*, and almost every computer in the world has the exact same protocol mapped to the exact same port number. That is why they are called well known—everyone follows this same standardized approach. This means that on almost every computer, port 25 is mapped to SMTP, port 80 is mapped to HTTP, and so on. This mapping between lower-numbered ports and specific protocols is a de facto standard, which just means that we all do this and that we do not have a standards body dictating that it absolutely has to be done this way. The fact that almost everyone follows this approach translates to more interoperability among systems all over the world.

Because this is a de facto standard and not a standard that absolutely must be followed, administrators can map different protocols to different port numbers if that fits their purpose. However, one thing to note is that ports 0 to 1023 can be used only by privileged system or root processes.

The following shows some of the most commonly used protocols and the ports to which they are usually mapped:

- Secure Shell (SSH) port 22
- SMTP port 25
- DNS port 53
- HTTP port 80
- NTP port 123
- IMAP port 143
- HTTP Secure (HTTPS) port 443

Registered ports are 1024 to 49151, which can be registered with the Internet Assigned Numbers Authority (IANA) for a particular use. Vendors register specific ports to map to their proprietary software. *Dynamic ports* (also known as *ephemeral ports*) are 49152 to 65535 and are available to be used by any application on an “as needed” basis. Typically, these ports are used on the client side of a connection. For instance, if you look at a specific connection between your web browser and a website you visit, you may notice that the destination port is 80 and the source port (on your client) is 53042. In some cases, however, the server may be listening on a well-known port (e.g., 135 for RPC) and hand off the server port to an ephemeral one. This means that in some cases, you will see both the client and the server port for a connection in the range of 49152 to 65535.

TCP Handshake

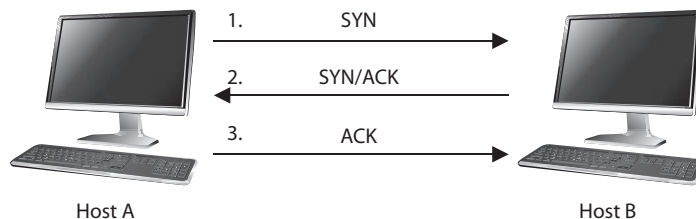
TCP must set up a virtual connection between two hosts before any data is sent. This means the two hosts must agree on certain parameters, data flow, windowing, error detection, and options. These issues are negotiated during the handshaking phase, as shown in Figure 11-21.

The host that initiates communication sends a synchronization (SYN) packet to the receiver. The receiver acknowledges this request by sending a SYN/ACK packet. This packet translates into, “I have received your request and am ready to communicate with you.” The sending host acknowledges this with an acknowledgment (ACK) packet, which translates into, “I received your acknowledgment. Let’s start transmitting our data.” This completes the handshaking phase, after which a virtual connection is set up, and actual data can now be passed. The connection that has been set up at this point is considered *full duplex*, which means transmission in both directions is possible using the same transmission line.

If an attacker sends a target system SYN packets with a spoofed address, then the victim system replies to the spoofed address with SYN/ACK packets. Each time the victim system receives one of these SYN packets, it sets aside resources to manage the new connection. If the attacker floods the victim system with SYN packets, eventually the victim system allocates all of its available TCP connection resources and can no longer process new requests. This is a type of DoS attack that is referred to as a *SYN flood*. To thwart this type of attack you can use a number of mitigations, the most common of which are described in the Internet Engineering Task Force’s (IETF) Request for Comments (RFC) 4987. One of the most effective techniques described in RFC 4987 is the use of SYN caches, which delays the allocation of a socket until the handshake is completed.

Another attack vector we need to understand is TCP sequence numbers. One of the values that is agreed upon during a TCP handshake between two systems is the sequence numbers that will be inserted into the packet headers. Once the sequence number is agreed upon, if a receiving system receives a packet from the sending system that does not have this predetermined value, it disregards the packet. This means that an attacker cannot just spoof the address of a sending system to fool a receiving system; the attacker has to spoof the sender’s address and use the correct sequence number values. If an attacker can correctly predict the TCP sequence numbers that two systems will use, then she can create packets containing those numbers and fool the receiving system into thinking that the packets are coming from the authorized sending system. She can then take over the TCP connection between the two systems, which is referred to as *TCP session hijacking*.

Figure 11-21
The TCP three-way handshake



Data Structures

As stated earlier, the message is formed and passed to the application layer from a program and sent down through the protocol stack. Each protocol at each layer adds its own information to the message to create a PDU and passes it down to the next layer. This activity is referred to as *encapsulation*. As the message is passed down the stack, it goes through a sort of evolution, and each stage has a specific name that indicates what is taking place. When an application formats data to be transmitted over the network, the PDU is called a *message* or *data*. The message is sent to the transport layer, where TCP does its magic on it. The PDU is now a *segment*. The segment is sent to the network layer. The network layer adds routing and addressing, and now the PDU is called a *packet*. The network layer passes off the packet to the data link layer, which frames the packet with a header and a trailer, and now it is called a *frame*. Figure 11-22 illustrates these stages.



EXAM TIP If the message is being transmitted over TCP, it is referred to as a “segment.” If it is being transmitted over UDP, it is referred to as a “datagram.”

Sometimes when an author refers to a segment, she is specifying the stage in which the data is located within the protocol stack. If the literature is describing routers, which work at the network layer, the author might use the word “packet” because the data at this layer has routing and addressing information attached. If an author is describing network traffic and flow control, she might use the word “frame” because all data actually ends up in the frame format before it is put on the network wire. The important thing here is that you understand the various steps a data package goes through when it moves up and down the protocol stack.

Figure 11-22
Data goes through its own evolutionary stages as it passes through the layers within the network stack.

