Chapter One

Introduction to Computer Network

A **computer network**, or data network, is a digital telecommunications network which allows nodes to share resources. In computer networks, computing devices exchange data with each other using connections (data links) between nodes These data links are established over cable media such as wires or optic cables, or wireless media such as WiFi.

Early ***data networks*** were limited to exchanging character-based information between connected computer systems. Current ***networks*** have evolved to carry voice, video streams, text, and graphics between many different types of devices.
The use of the ***Internet*** spread quickly as connectivity became available in the 1990s. The early users of the World Wide Web were mostly university researchers exchanging information, but other people and businesses quickly figured out how to take advantage of web based communications.

The Internet has quickly become an integral part of our daily routines. The complex interconnection of electronic devices and media that comprises the network is transparent to the millions of users who make it a valued and personal part of their lives. Data networks that were once the transport of information from business to business are now also used to improve the quality of life for people everywhere. In the course of a day, resources available through the Internet can help you do the following:

- Decide what to wear using online current weather conditions.
- Find the least-congested route to your destination, displaying weather and traffic video from webcams.
- Check your bank balance and pay bills electronically.
- Receive and send e-mail at an Internet cafe over lunch.
- Obtain health information and nutritional advice from experts all over the world, and post to a forum to share related health or treatment information.
- ***Download*** new recipes and cooking techniques to create a spectacular dinner.
- Post and share your photographs, home videos, and experiences with friends or with the world.
- Use Internet phone services.
- Shop and sell at online auctions.
- Use instant messaging and chat for both business and personal use.

Networks Supporting the Way We Learn

The advances in the Internet and collaboration tools have been the force behind major changes in education. As web reliability and access have increased, more institutions have come to depend on technology to perform core educational functions. For example, **distance education** was once limited to correspondence, videos, or video and audio conferences. With newer collaboration tools and stronger web technologies, *online learning can engage remote students in interactive learning and real-time assessment.* The classes can use document sharing, wikis, online video, and online testing software to enhance learning opportunities. Student learning is becoming less dependent on location and schedule, which opens courses to potential students who previously could not attend classes. The methods of both face-to-face and online instruction are changing with the introduction of web tools such as wikis. Traditionally, a teacher provided course content and the class might have benefited from some discussions. With online tools equally available to all students, many classes focus on sharing the opinions and expertise of students. Students can communicate with the instructor and fellow students using online tools like e-mail, chat rooms, and instant messaging. Links provide access to learning resources outside the courseware. Blended e-learning provides the benefits of computer-based training while retaining the advantages of an instructor-led curriculum. Students have the opportunity to work online at their own pace and skill level while still having access to an instructor and other live resources. In addition to the benefits for the student, networks have improved the management and administration of courses as well. Some of these online functions include **enrollment**, **assessment**

delivery, and **grade books**. In the **business world**, the use of networks to provide efficient and cost-effective employee training is increasing in acceptance. Online learning opportunities can decrease time consuming and costly travel yet still ensure that all employees are adequately trained to perform their jobs in a safe and productive manner.

Online courseware and delivery offer many benefits to businesses, including the following:

- **Current and accurate training materials:** Collaboration among vendors, equipment manufacturers, and training providers ensures that the courseware is up to date with the latest processes and procedures. When errors in materials are found and corrected, the new courseware is immediately available to all employees.

- **Availability of training to a wide audience:** Online training is not dependent on travel schedules, instructor availability, or physical class size. Employees can be given deadlines by which training is to be completed, and they can access the courseware when it is convenient.

- **Consistent quality of instruction:** The quality of the instruction does not vary as it would if different instructors were delivering an in-person course. The online curriculum provides a consistent core of instruction to which instructors can add additional expertise.

- **Cost reduction:** In addition to reducing the cost of travel and the lost time associated with travel, there are other cost-reducing factors for business related to online training. It is usually less expensive to revise and update online courseware than it is to update paper-based material. Facilities to support in-person training can also be reduced or eliminated.

Networks Supporting the Way We Work

Many companies use collaboration software packages that allow distributed work groups—people working together but not in the same physical location—to interactively create documents and contribute to projects in real time. These collaboration tools demonstrate the global nature of online business and are now essential to large and small businesses alike.

Different companies use different types of networks. Employees can meet on the Internet, or they can join a restricted group on a company ***intranet***, which allows only internal employee access. Another type of network is an ***extranet***, a type of network that allows outside vendors special access to limited information in a company. To reap the benefits of these technology tools, businesses must provide the continuing training and education of workers. The ability to learn and adopt new ways to implement technology into the workplace is a valuable skill sought after by most employers. Most of the preceding examples highlight the benefits that larger corporations experience from computer networks. Networks also have enabled small businesses to achieve success.

What Is Communication?

People have many ways of communicating with each other. Whether the communication is verbal or nonverbal, face-to-face or over the telephone, or in a handwritten letter or in a chat room, successful communication requires common rules. The rules of communication are also known as **protocols**. Some of the protocols required for communication to occur include the presence of:

- An identified sender and receiver
- An agreed-upon method of communicating (face-to-face, telephone, letter, photograph, and so on)
- Common language and grammar
- An agreed-upon speed and timing of delivery (for example, “Please slow down so that I can understand you.”)
- Confirmation or acknowledgment requirements (for example, “Is that clear?” “Yes, thank you.”)

Not all communications have the same agreed-upon protocols. For example, an important legal letter can require a signature and response from the recipient, but personal letters need no such acknowledgment. People are unaware of many of the rules they follow while communicating because they are ingrained in language and culture. Tone of voice, pausing between thoughts, and polite ways to interrupt are just a few examples of implicit rules that humans follow.

Quality of Communication

Successful communication between computer networks devices, just as is true with communication between people, occurs when the meaning of the message understood by the recipient matches the meaning intended by the sender. There are many potential barriers to successful communication between computers on a network. The process of sending a message on a computer network can be complex and have many steps and conditions, and any step poorly performed or condition not properly met can potentially destroy the message. The steps and conditions, or factors, can be separated into internal and external groups. The external factors stem from the complexity of the network and the number of devices handling the message en route to the destination.

Examples of **external factors include the following:**

- The quality of the pathway between the sender and the recipient
- The number of times the message has to change form
- The number of times the message has to be redirected or readdressed
- The number of other messages being transmitted simultaneously on the communication network
- The amount of time allotted for successful communication

Internal factors include the following:

- The size of the message
- The complexity of the message
- The importance of the message

More **complex** messages can be more difficult for the recipient to understand, and **larger** messages have a greater potential to be distorted or incomplete at the destination.

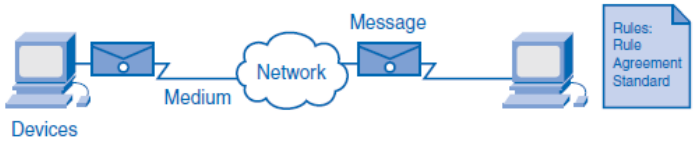
Elements of a Network

All networks have four basic elements in common:

- **Rules or agreements:** Rules or agreements (**protocols**) govern how the messages are sent, directed, received, and interpreted.
- **Messages:** The messages or units of information travel from one device to another.
- **Medium:** A medium is a means of interconnecting these devices, that is, a medium can transport the messages from one device to another.
- **Devices:** Devices on the network exchange messages with each other.

Figure 1-1 depicts a small network featuring rules, messages, a medium, and two devices.

Figure 1-1 Elements of a Network



Early networks had varying ***standards*** and, as a result, could not communicate easily with each other. Now global **standardization** of these elements enables easy communication between networks regardless of the equipment manufacturer. People use many technologies and devices that they do not completely understand. Driving a car, for example, is a common function for many people. When a driver starts a car, puts it into gear, and steps on the gas, many systems begin to work together. The car moves because an ignition system started the car, a fuel system regulates power, electrical systems run lights and gauges, and a complex transmission chooses appropriate gears to make the car move as directed by the driver. All of this happens under the hood and out of sight and mind to the driver, who focuses on the task of driving safely to a destination. Most drivers know little or nothing about how a car works but are still able to use it effectively for their own purpose.

Computer networks are similar to cars in the example. Two people communicating on end devices in different networks can do so only if many complex processes are successfully completed. These processes include a **message**, some form of **media**, various **devices**, and **protocols** working together.

❖ Message

Messages is a generic term that contains **web pages**, **e-mail**, **instant messages**, **telephone calls,** and other forms of communication enabled by the Internet. The message must be one that the network can carry. First, the messages must be supported in software at the end devices. Instant messaging and chat, for example, require some software setup before a session can begin. Different software is required for audio and video conferencing. These software programs that support communication functions are called services, and to initiate a message, a service must be installed. Examples of services include e-mail, IP telephony, and use of the World Wide Web. It does not matter whether the message is text, voice, or video, because all forms are converted into ***bits***, ***binary***-coded digital signals, to be carried over a wireless, copper, or fiberoptic connection. The digital signal can change with the media, but the original message content will remain intact.

❖ Medium

The medium that physically carries the message can change several times between the sender and the receiver. Network connections can be ***wired*** or ***wireless***. In **wired connections**, the **medium** is either **copper**, which carries **electrical signals**, or **optical fiber**, which carries **light signals**. The copper medium includes cables, such as **twisted pair** telephone wire, **coaxial cable**, or most commonly, what is known as **Category 5 unshielded twisted-pair (UTP)** cable. **Optical fibers**, thin strands of **glass or plastic** that carry **light signals**, are another form of networking media.

In **wireless connections**, the medium is **the Earth's atmosphere**, or **space**, and the **signals are microwaves**. Wireless media can include the home wireless connection between a wireless router and a computer with a wireless network card, the terrestrial wireless connection between two ground stations, or the communication between devices on Earth and satellites. In a typical journey across the Internet, a message can travel across a variety of media.

❖ Devices

Several devices, such as **switches** and **routers**, work to see that the message is properly directed from the ***source***, or originating device, to the **destination** device. At the destination network there can be more switches, cable, or perhaps a wireless router that will deliver the instant message to the receiver.

Graphics and icons are common when reading about networks. Icons, or small pictures arranged to represent a network's layout, can greatly clarify information about the design of the network. Figure 1-2 shows various network device symbols. **The desktop, laptop, and IP phone** represent **end-user devices**, whereas the rest of the icons depict network equipment or media used to connect the end devices. These icons do not refer to specific models or features on devices, which can vary greatly. Table 1-1 briefly describes the network symbols.

Figure 1-2 Network Device Symbols

Table 1-1 Internal and External Factors Affecting Successful Communication

Symbol	Description
Desktop computer	A common computer used in a home or office
Laptop	A portable computer
Server	A computer dedicated to providing application services to end users on a network
IP phone	A digital telephone that carries voice as data over data networks instead of analog phone lines
LAN media	Local-area network media, usually copper cable
Wireless media	Depicts local-area network wireless access
LAN switch	The most common device for interconnecting local-area networks
Firewall	A device that provides security to networks
Router	A device that helps direct messages between networks
Wireless router	A specific type of router often found in home networks
Cloud	A symbol used to summarize a group of networking devices out of local management control, often the Internet itself
WAN media	One form of wide-area network (WAN) interconnection, represented by the lightning bolt–shaped line

❖ Rules

All communication processes happen, as far as humans can tell, in an instant, and tens of thousands of processes can happen in a single second. To work properly, the network processes must be tightly controlled. Rules govern every step of the process, from the way cables are designed to the way the digital signals are sent. These **rules** are called **protocols**, and the communications industry has **standardized** most of them to allow people in different places with different equipment to communicate. The most common protocols are ***IP (Internet Protocol)*** and ***TCP (Transmission Control Protocol)***. These protocols work together and are usually known as the **TCP/IP protocol stack**.

Table 1-2 lists some common services and the protocols that support them.

Table 1-2 Services and Their Protocols

Service	Protocol ("Rule")
World Wide Web (WWW)	HTTP (Hypertext Transport Protocol)
E-mail	SMTP (Simple Mail Transport Protocol) and POP (Post Office Protocol)
Instant message (Jabber, AIM)	XMPP (Extensible Messaging and Presence Protocol) and OSCAR (Open System for Communication in Realtime)
IP telephony	SIP (Session Initiation Protocol)

People often only picture networks in the abstract sense: We create and send a text message, and it almost immediately shows up on the destination device. Although we know that between our sending device and the receiving device there is a network over which our message travels, we rarely think about all the parts and pieces that make up that infrastructure. The following list ties together how the elements of networks—devices, media, and services—are connected by rules to deliver a message:

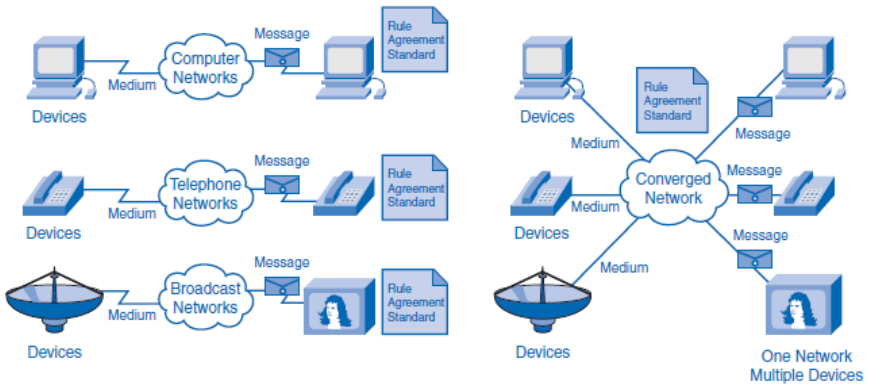
1. An end user types an instant message to a friend using an application on a PC.

2. The instant message gets converted into a format that can be transmitted on the network. All types of message formats—text, video, voice, or data—must be converted to **bits** before being sent to their destinations. After the instant message is converted to bits, it is ready to be sent onto the network for delivery.
3. The network interface card (NIC) inside the PC generates electrical signals to represent the bits and places the bits on the medium so that they can travel to the first network device.
4. The bits are passed from device to device in the local network.
5. If the bits need to leave the local network, they leave through a router connecting to a different network. There can be dozens, even hundreds, of devices handling the bits as they are routed to their destination.
6. As the bits get close to their destination, they once again get passed through local devices.
7. Finally, the NIC on the destination device accepts the bits and converts them back into a readable text message.

Converged Networks

Communication technologies evolved at different times and in different places in the twentieth century. Many developments in radio broadcast technology were driven by military necessity, yet developments in broadcast television grew to answer a market demand. The telephone evolved as a wired technology and then as a wireless technology. Computer communication developments came much later in the century. For example, the first text e-mail message was sent in the 1960s, but e-mail did not become popular until the 1980s. Now it is quite common to use a computer for instant messaging, telephone calls, and video sharing. The technology and protocols of each of these communication methods developed largely independent of each other, and most users of TV, telephone, and computer services pay different providers for each service. But recent developments in each area have driven broadcast and telephony to the digital technology already used by computers. This coming together of technologies onto a digital platform is called ***convergence***. **Convergence** occurs when telephones, broadcasts, and computer communications **all use the same rules, devices, and media to transport their messages**. On a converged network, or platform, different devices, such as televisions or cell phones, will use a common network infrastructure to communicate.
Figure 1-3 demonstrates the concept of **nonconverged** systems on the **left** and a **converged** network on the **right**.

Figure 1-3 Convergence



The Network Architecture

The Internet's design meets four fundamental expectations:

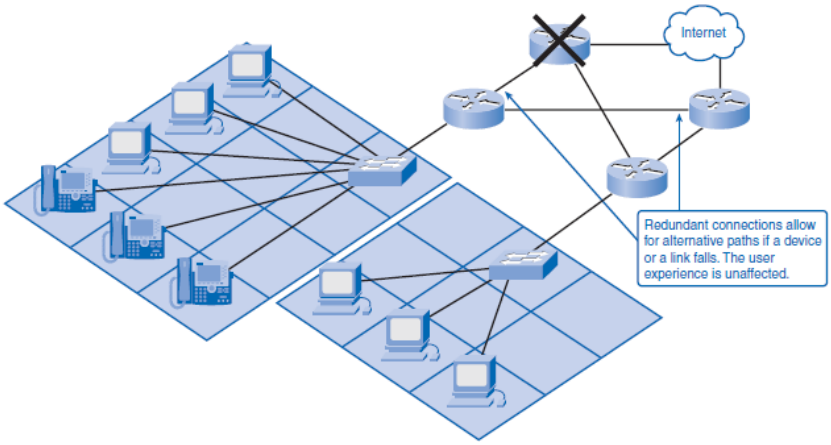
- **Fault tolerance,**
- **Scalability,**
- **Quality of service (QoS),**
- **Security**.

These topics are introduced here, and their implementation is discussed in the following section.

Fault tolerance, simply stated, means that the Internet will continue to function normally even when some of the components of the network fail. ***Redundancy***, or the duplication of equipment and media, is a key factor in fault tolerance. If a server fails, a redundant server performing the same functions should be able to pick up the work until

repairs are made. If a data link fails on a fault-tolerant network, messages will be routed to the destination on a duplicate route. Figure 1-4 depicts a fault-tolerant network with a failed network router.

Figure 1-4 Fault Tolerance

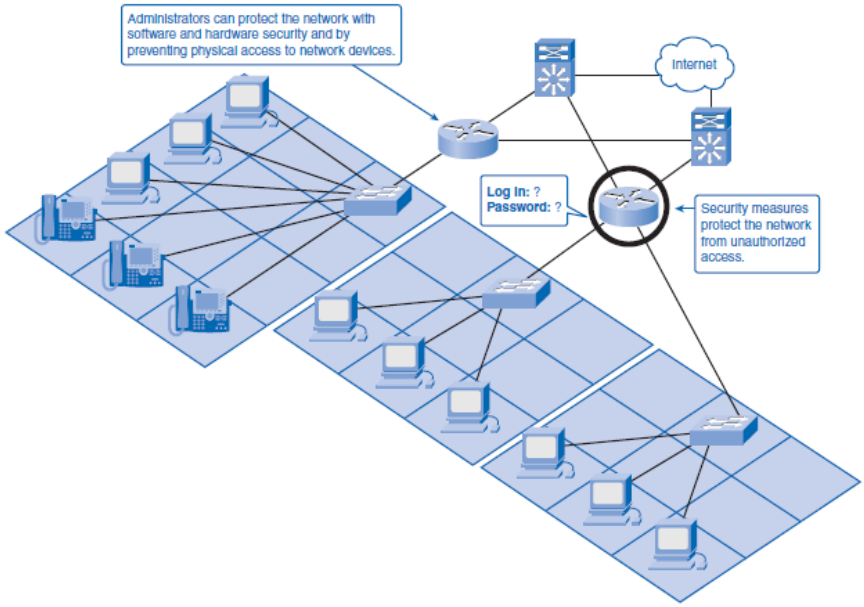


Scalability describes the network's ability to grow and react to future changes. A scalable network can accept new users and equipment without having to start over on the design. As mentioned earlier, it is certain that changes in the ways networks are used will occur, and having an adaptable, or scalable, network will allow the insertion of new users without having to rebuild the entire network. A scalable network will be able to grow internally and externally, joining other networks to form an ***internetwork*** that can grow to keep pace with user demand.

QoS indicates the performance level of services offered through the network. Services such as live video or voice can require more resources than services such as e-mail. Because many technologies are converged onto one platform, the separation of types of services on that platform can allow **higher priority** for one service over another. For example, a network administrator can determine that the data from attendees of a web meeting has priority over e-mail service. Configuring devices to prioritize types of data is an example of QoS.

Network security is essential if the public is to have **confidence** when using the Internet. People using the Internet to do business demand security for their financial transactions, and government and businesses that require personal information (for example, a hospital or doctor's office) must provide the protection of their clients' privacy. Encrypted messages and the use of security devices at the gate of a local network are methods of implementing security. Encryption and firewalls are not necessarily enough to protect a network, however. The security and privacy expectations that result from the use of internetworks to exchange confidential and business-critical information exceed what the current architecture can deliver. As a result, much effort is being devoted to this area of research and development. In the meantime, many tools and procedures are being implemented to combat inherent security flaws in the network architecture.
Figure 1-5 indicates how **firewall** settings on a **router** add security to network architecture by controlling network access.

Figure 1-5 Network Security



Fault-Tolerant Network Architecture

The architects of the Internet began their designs with fault tolerance as a high priority. The Internet came about when the United States Department of Defense (DoD) planners wanted to design a communication medium that could withstand widespread destruction of telephone and other communication infrastructure.

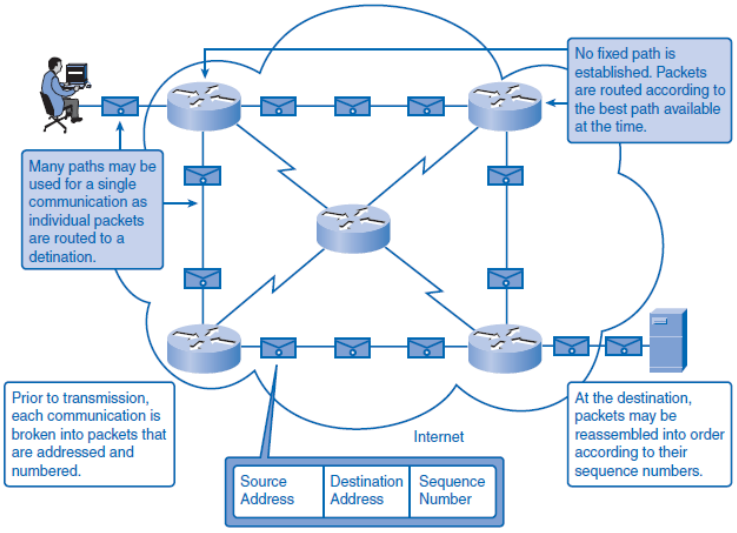
Circuit-Switched, Connection-Oriented Networks

The existing infrastructure at the time was a circuit-switched, connection-oriented network. Phone operators and primitive dial systems connected telephone calls by setting up a temporary circuit that was a physical connection on which the phone signal would travel from sender to receiver. The technology was connection oriented because any physical disconnect or service problem between the two users would drop the call. This would require initiating a new call and provisioning a new circuit. The circuit-switched design provided new service to customers, but it had flaws as well. For example, only one phone call occupied each circuit, and no other calls could use the circuit until the previous call ended. This inefficiency limited the capacity of the phone system and made it expensive, especially for long-distance calls. From the DoD perspective, the system was vulnerable to easy disruption from enemy attacks.

Packet-Switched, Connectionless Networks

The answer to the fault tolerance issue was converting to a packet-switched, connectionless network. On a packet-switched network, a single message is broken into small blocks of data, known as ***packets***, which address information for the sender and the receiver. The packets then travel through one or more networks along various paths and reassemble at the destination. The packets travel independently of each other and often take different routes to a destination. Messages are usually broken into thousands of packets, and it is common for some of them to be lost along the way. Protocols allow for this and contain methods for requesting retransmission of packets lost en route. Packet-switched technology is connectionless because it does not require an active connection for the call to go through. This allows more efficiency than circuit-switched networks because multiple users can use network circuits simultaneously. Packet-switched technology is fault tolerant because it avoids the perils of relying on a single circuit for service reliability. If one network path fails, another network path can deliver the entire message.
Figure 1-6 depicts a packet-switched network with several alternative routes between the source and destination.

Figure 1-6 Packet-Switched Network

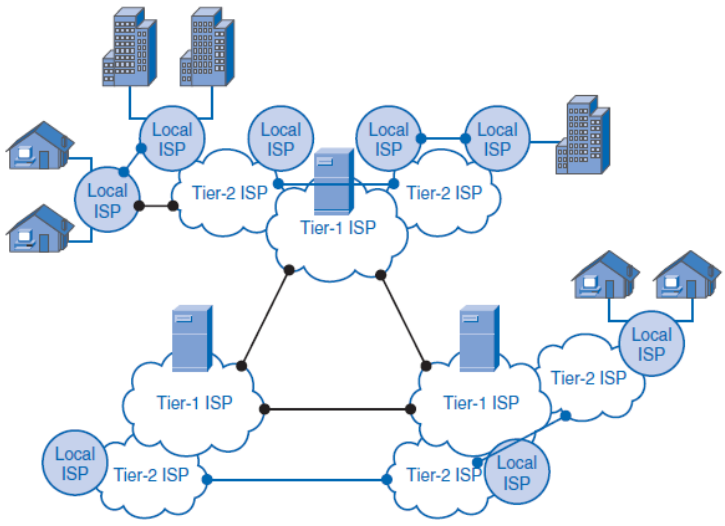


Packet-switched networks are the standard for the Internet, but a niche market remains for circuit-switched networks. Circuit-switched networks today allow circuit failure and session reestablishment, and some customers like the reliability and security that come from modern dedicated circuits. Circuit-switched connections are more expensive than packet switched networks, but many institutions require the constant circuit availability and security and are willing to pay the extra price.

Scalable Network Architecture

A scalable network is able to grow without undergoing fundamental change at its core. The Internet is an example of scalable design. The Internet has grown exponentially in the past decade or so, and the core design is unchanged. The Internet is a collection of many private and public networks interconnected by routers. Large tier-1 Internet service providers (ISP) house the largest domain servers that track Internet addresses. That information is replicated and shared in the lower tiers in the system. This hierarchical, multitiered design allows most traffic to be processed away from the upper-tier servers. This distribution of processing work means that changes made at the lower tiers, such as adding a new ISP, do not affect the upper levels. Figure 1-7 demonstrates the hierarchical design of the web. Traffic between lower tiers can bypass the upper-tier servers in the Internet. This allows upper tiers to work more efficiently as well as to provide alternate paths for peak web traffic.

Figure 1-7 Hierarchical Internet



Providing Quality of Service

When the Internet first came into public use, people were amazed at the new tasks they could do and were tolerant of delays and dropped messages. Now, however, users have adapted to higher speeds and a greater ***quality of service (QoS)***. QoS refers to the mechanisms that manage congested network traffic. Congestion is caused when the demand on the network resources exceeds the available capacity. There are some constraints on network resources that cannot be avoided. Constraints include technology limitations, costs, and the local availability of high-bandwidth service. Network ***bandwidth*** is the measure of the data-carrying capacity of the network. When simultaneous communications are attempted across the network, the demand for network bandwidth can exceed its availability. The obvious fix for this situation is to increase the amount of available bandwidth. But, because of the previously stated constraints, this is not always possible. Using QoS, a manager can choose which traffic gets priority for processing in the network.

For example, most people expect telephone service to be reliable and clear. Many companies want to save money by moving their long-distance phone calls onto the Internet using Voice over Internet Protocol (VoIP) services. If the users cannot distinguish any difference between regular phones and VoIP phones, they will not mind the change. But if network congestion causes the VoIP phones to experience delays and dropped calls, users will return to the old expensive service. The network administrator must ensure that the quality of voice service is as high as possible, and she can do this by giving voice traffic priority over other web traffic. Different companies and institutions have different needs and priorities. Some companies might prioritize voice traffic, others might want to give priority to video traffic, and still others might want to give priority to traffic carrying financial data. These various needs can be met by classifying network traffic and assigning priorities to each classification. Classification of traffic means putting web traffic into categories. Because so many types of web traffic exist, assigning each its own priority is not practical. Thus, using one category for time-sensitive traffic such as voice and video and another category for less sensitive traffic like e-mail and document transfers is a way to sort traffic into manageable groups. Not every network will have the same priorities, and different institutions will assign data types into different categories according to their needs. After traffic types are categorized, they
can be put into queues.

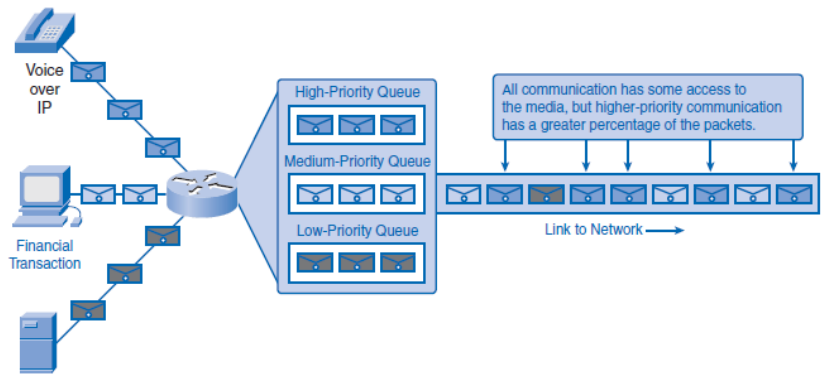
Examples of priority decisions for an organization might include

- **Time-sensitive communication:** Increase priority for services like telephony or video distribution.
- **Non-time-sensitive communication:** Decrease priority for web page retrieval or e-mail.
- **High importance to organization:** Increase priority for production control or business transaction data.
- **Undesirable communication:** Decrease priority or block unwanted activity, like peer-to-peer file sharing or live entertainment.

Getting into a queue means getting into a line. An example of ***priority queuing*** can be found at an airport check-in counter. There are two classes of passengers in queues: a queue for coach passengers and a separate queue at the end of the counter for first-class passengers. When airline agents become available, they will choose to help passengers from the first-class queue ahead of someone who has been waiting longer in the coach class queue. The coach class passengers are still important, and the agents will eventually help them, but the airlines give priority to first-class passengers because they value the extra revenue those customers bring to the company. In networking priority, queuing is much the same as the airline ticket counter process. Network managers assign priorities to traffic categories and allow the more important categories to have better access to the network's bandwidth.

Figure 1-8 demonstrates different classes of traffic having different priority access to bandwidth.

Figure 1-8 Priority Queuing



Providing Network Security

The Internet has proven to be fertile ground for business, and business-to-business transactions and e-commerce are sustaining significant growth every year. The same environment that attracts legitimate business, however, also attracts scam artists and vandals. Compromising the integrity of company assets could have serious business and financial repercussions. As a result, network security is a major concern of web providers and users, and web safety is a key part of any network management plan. To provide security, a network manager must address two areas:

- Network infrastructure security
- Content security

Securing the network infrastructure means protecting the devices from outside contact. Locking computer room doors and using quality password protection on network equipment and software are simple steps that can go a long way to securing an infrastructure. Securing network content means protection of data stored on network devices and the protection of packets carrying data into or out of the network. Content security on a network means ensuring confidentiality, maintaining communication integrity, and ensuring network availability.

Ensuring Confidentiality

Data privacy is maintained by allowing only the intended and authorized recipients— individuals, processes, or devices—to read the data. Different methods ensure data confidentiality. Having a strong system for user ***authentication***, enforcing **passwords** that are difficult to guess, and requiring users to change passwords frequently help restrict access to communications and to data stored on network-attached devices. Where appropriate, encrypting content ensures confidentiality and minimizes unauthorized disclosure or theft of information.

Maintaining Communication Integrity

Data integrity means having the assurance that the information has **not been altered** in transmission, from origin to destination. Data integrity can be compromised when information has been corrupted—willfully or accidentally—before the intended recipient receives it. Source integrity is the assurance that the identity of the sender has been validated. Source integrity is compromised when a user or device fakes its identity and supplies incorrect information to a recipient. **Digital signatures**, **hashing algorithms**, and **checksum** mechanisms provide source and data integrity across a network to prevent unauthorized modification of information.

Ensuring Availability

Making sure that resources are available to authorized users is an important part of a security plan. If a network is unavailable to a company using web-based business practices, the business can be slowed to a halt. Computer **virus** attacks and **denial of service** (DoS) attacks can bring a network down. A DoS attack occurs when outside computers **flood** a network with so many requests for service that valid users cannot access the network resources. **Tools to combat virus and DoS** attacks include **antivirus** software on servers and desktops, and ***firewalls***, which are routers and servers that are network gatekeepers that analyze traffic entering and exiting a network. Building fully redundant network infrastructures, with few ***single points of failure*** that can bring the network down, can reduce the impact of these threats.

Chapter Two

NETWORK STANDARDS

Protocol Suites and Industry Standards

In the early days of networking, each manufacturer had proprietary network equipment and protocols to support it. This worked well as long as the company that purchased the equipment did not need to share data outside its own network. As companies started to do business with other companies who were using different network systems, the need for a cross platform standard for network communication became apparent. People from the telecommunications industry gathered to standardize the way network communication works by writing common protocols. These standards are practices that are endorsed by representatives from industry groups and are followed to ensure interoperability between vendors. For example, Microsoft, Apple, and Linux operating systems each have a way to implement the TCP/IP protocol stack. This allows the users of different operating systems to have common access to network communication. The organizations that standardize networking protocols are the ***Institute of Electrical and Electronics Engineers (IEEE)*** and the ***Internet Engineering Task Force (IETF)***.

Interaction of Protocols

An example of the use of a protocol suite in network communications is the interaction between a web server and a web browser. This interaction uses a number of protocols and standards in the process of exchanging information between them. The different protocols work together to ensure that the messages are received and understood by both parties. Examples of these protocols are as follows:

- **Hypertext Transfer Protocol (HTTP):** HTTP is a common protocol that governs the way that a web server and a web client interact. HTTP defines the content and formatting of the requests and responses exchanged between the client and server. Both the client and the web server software implement HTTP as part of the application. The HTTP protocol relies on other protocols to govern how the messages are transported between client and server.
- **Transport protocol:** Transmission Control Protocol (TCP) is the transport protocol that manages the individual conversations between web servers and web clients. TCP divides the HTTP messages into smaller pieces, called segments, to be sent to the destination client. It is also responsible for controlling the size and rate at which messages are exchanged between the server and the client.
- **Internetwork protocol:** The most common internetwork protocol is Internet Protocol (IP). IP is responsible for taking the formatted segments from TCP, encapsulating them into packets, assigning the appropriate addresses, and selecting the best path to the destination host.
- **Network access protocols:** Network access protocols describe two primary functions: data-link management and the physical transmission of data on the media. Data-link management protocols take the packets from IP and format them to be transmitted over the media. The standards and protocols for the physical media govern how the signals are sent over the media and how they are interpreted by the receiving clients. Transceivers on the network interface cards implement the appropriate standards for the media that is being used.

Technology-Independent Protocols

Protocols that guide the network communication process are not dependent on any specific technology to carry out the task. Protocols describe what must be done to communicate, not how the task is to be completed. For example, in a classroom, the protocol for asking a question might be to raise a hand for attention. The protocol instructs students to raise their hands, but it does not specify how high to raise them or specify whether the right hand or left hand is better or whether waving the hand is helpful. Each student can raise his or her hand in a slightly different way, but if the hand is raised, the teacher will likely give attention to the student. So network communication protocols state what tasks must be completed, not how to complete them. This is what enables different types of devices, such as telephones and computers, to use the same network infrastructure to communicate. Each device has its own technology, but it is able to interact with different devices at the network level. In the previous example of Apple, Microsoft, and Linux, the operating systems must find a way to present data to others using TCP/IP, but each operating system will have its own way to do it.

Using Layered Models

The IT industry uses ***layered models*** to describe the complex process of network communication. Protocols for specific functions in the process are grouped by purpose into well defined layers.

The Benefits of a Layered Model

By breaking the network communication process into manageable layers, the industry can benefit in the following ways:

- Defines common terms that describe the network functions to those working in the industry and allows greater understanding and cooperation.
- Segments the process to allow technologies performing one function to evolve independently of technologies performing other functions. For example, advancing technologies of wireless media is not dependent on advances in routers.
- Fosters competition because products from different vendors can work together.
- Provides a common language to describe networking functions and capabilities.
- Assists in protocol design, because protocols that operate at a specific layer have defined information that they act upon and a defined interface to the layers above and below.

As an IT student, you will benefit from the layered approach as you build your understanding of the network communication process.

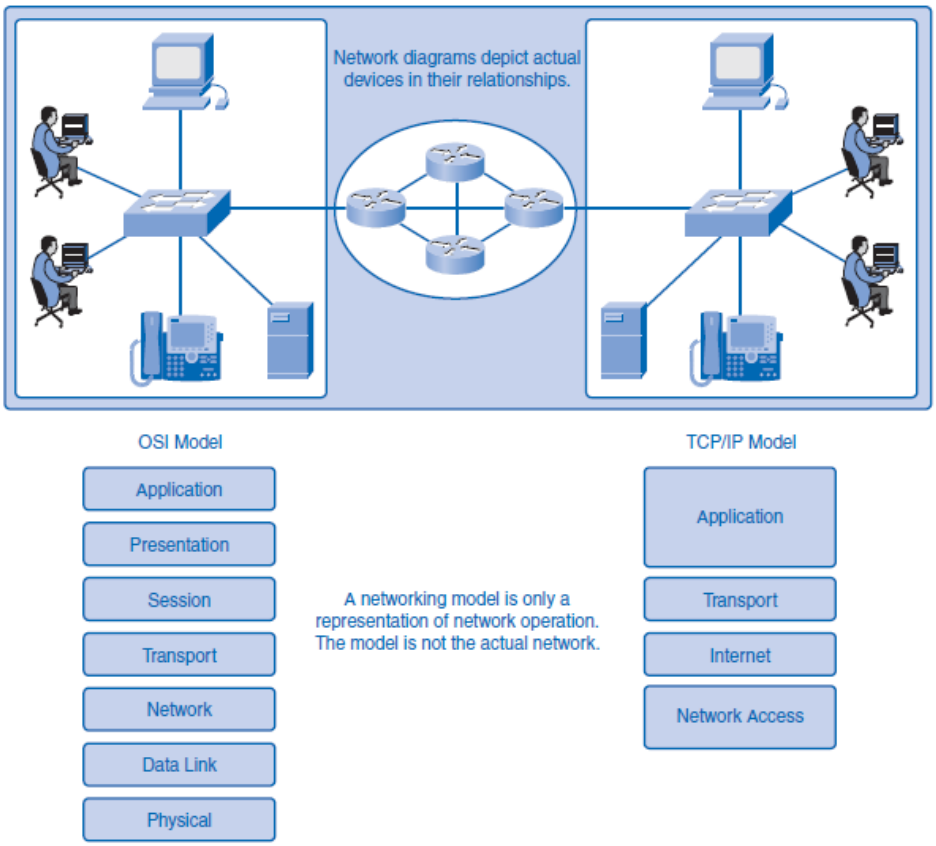
Protocol and Reference Models

Networking professionals use two networking models to communicate within the industry: protocol models and reference models. Both were created in the 1970s when network communication was in its infancy. A protocol model provides a model that closely matches the structure of a particular protocol suite. The hierarchical set of related protocols in a suite typically represents all the functionality required to interface the human network with the data network. The ***TCP/IP*** model is a protocol model because it describes the functions that occur at each layer of protocols within the TCP/IP suite. A reference model provides a common reference for maintaining consistency within all types of network protocols and services. A reference model is not intended to be an implementation specification or to provide a sufficient level of detail to define precisely the services of the network architecture. The primary purpose of a reference model is to aid inclearer understanding of the functions and process involved. The Open Systems Interconnection (OSI) model is the most widely known internetwork reference model. The OSI model describes the entire communication process in detail, and the TCP/IP model describes the communication process in terms of the TCP/IP protocol suite and the way it functions. It is important to know details of the OSI model to understand the entire network communication process and to know the TCP/IP model to understand how the process is implemented in current networks.

The OSI model is used to reference the process of communication, not to regulate it. Many protocols in use today apply to more than one layer of the OSI model. This is why some of the layers of the OSI model are combined in the TCP/IP model. Some manufacturers use variations on these models to demonstrate the functions of their products within the industry.

Figure 2-8 shows both OSI and TCP/IP models.

Figure 2-8 OSI and TCP/IP Models



TCP/IP Model

The TCP/IP model defines the four communication functions that protocols perform. TCP/IP is an open standard, which means that one company does not control it. The rules and implementations of the TCP/IP model were cooperatively developed by members of the industry using Request for Comments (RFC) documents. RFC documents are publicly accessible documents that define specifications and policies of the protocols and of the Internet in general. Solicitation and maintenance of RFCs are the responsibility of the IETF.
Table 2-3 briefly describes the functions of each layer of the TCP/IP model.

Table 2-3 Layers of the TCP/IP Model

Layer	Description
Application	Represents application data to the user. For example, the HTTP presents data to the user in a web browser application like Internet Explorer.
Transport	Supports communication between devices and performs error correction.
Internet	Finds the best path through the network.
Network access	Controls hardware devices and media.

Communication Process

The TCP/IP model describes the functionality of the protocols that make up the TCP/IP protocol suite. These protocols, which are implemented on both the sending and receiving hosts, interact to provide end-to-end delivery of applications over a network. A complete communication process includes these steps:

1. Creation of data at the application layer of the originating source end device.
2. Segmentation and encapsulation of data as it passes down the protocol stack in the source end device.
3. Generation of the data onto the media at the network access layer of the stack.
4. Transportation of the data through the internetwork, which consists of media and any intermediary devices.
5. Reception of the data at the network access layer of the destination end device.
6. Decapsulation and reassembly of the data as it passes up the stack in the destination device.
You learn more about the encapsulation and decapsulation processes in the next section.
7. Passing this data to the destination application at the application layer of the destination end device.

Protocol Data Units and Encapsulation

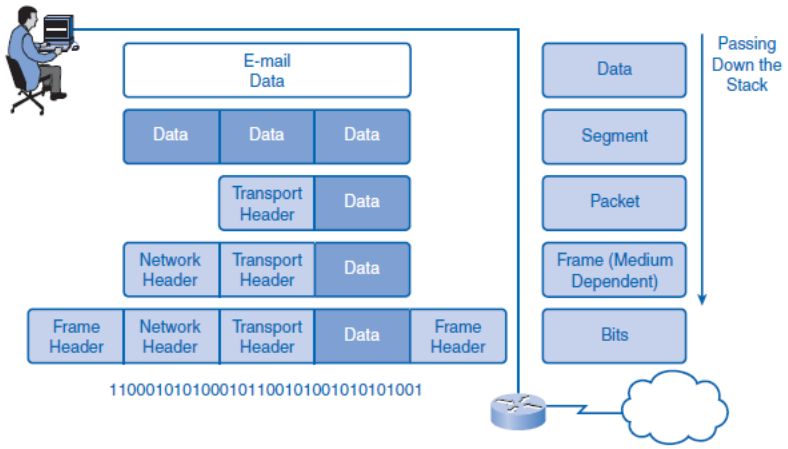
For application data to travel uncorrupted from one host to another, header (or control data), which contains control and addressing information, is added to the data as it moves down the layers. The process of adding control information as it passes through the layered model is called ***encapsulation***. ***Decapsulation*** is the process of removing the extra information and sending only the original application data up to the destination application layer. Each layer adds control information at each step. The generic term for data at each level is ***protocol data unit (PDU)***, but a PDU is different at each layer. For example, a PDU at the internetwork layer is different from the PDU at the transport layer, because internetwork layer data has been added to the transport layer data. The different names for PDUs at each layer are listed in Table 2-4.

Table 2-4 Protocol Data Unit Naming Conventions

PDU Name	Layer
Data	Application layer PDU
Segment	Transport layer PDU
Packet	Internetwork layer PDU
Frame	Network access layer PDU
Bits	PDU used for the physical transmission of binary data over media

Figure 2-9 depicts the encapsulation process and shows how PDUs are modified.

Figure 2-9 Encapsulation

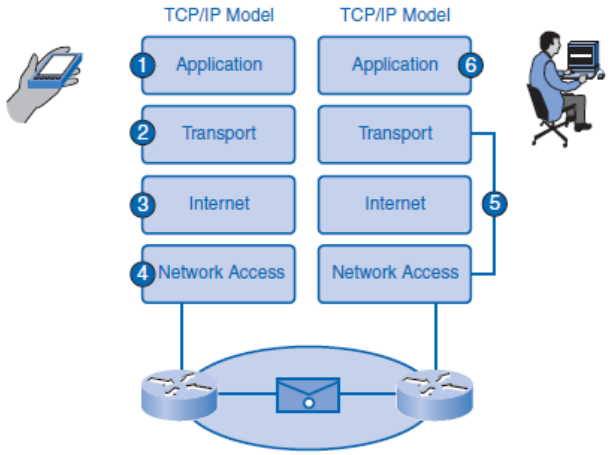


Sending and Receiving Process

The common task of sending an e-mail has many steps in the process. Using the proper terms for PDUs and the TCP/IP model, the process of sending the e-mail is as follows:

1. An end user, using an e-mail application, creates data. The application layer codes the data as e-mail and sends the data to the transport layer.
2. The message is segmented, or broken into pieces, for transport. The transport layer adds control information in a header so that it can be assigned to the correct process and all segments put into proper order at the destination. The segment is sent down to the internetwork layer.
3. The internetwork layer adds IP addressing information in an IP header. The segment is now an addressed packet that can be handled by routers en route to the destination. The internetwork layer sends the packet down to the network access layer.
4. The network access layer creates an Ethernet frame with local network physical address information in the header. This enables the packet to get to the local router and out to the web. The frame also contains a trailer with error-checking information. After the frame is created, it is encoded into bits and sent onto the media to the destination.
5. At the destination host, the process is reversed. The frame is decapsulated to a packet, then to a segment, and then the transport layer puts all segments into the proper order.
6. When all data has arrived and is ready, it is sent to the application layer, and then the original application data goes to the receiver's e-mail application. The message is successful.
Figure 2-10 depicts these steps as an encapsulated message travels down the TCP/IP model on the source and is en route to the destination for decapsulation.

Figure 2-10 Steps in the Communication Process



OSI Model

The ***Open Systems Interconnection (OSI)*** model, known as the OSI model, provides an abstract description of the network communication process. Developed by the ***International Organization for Standardization (ISO)*** to provide a road map for nonproprietary protocol development, the OSI model did not evolve as readily as the TCP/IP model. Many of the OSI protocols are no longer in use, but knowledge of the model as a reference is a basic expectation for networking professionals. Many professionals refer to the layers by number rather than name, so it is important to know both. The OSI model is just a reference model, so manufacturers have been free to create protocols and products that combine functions of one or more layers. New protocols might not exactly match the functions described at each layer but might fit into parts of two different layers. As designed, the communication process begins at the application layer of the source, and data is passed down to each lower layer to be encapsulated with supporting data until it reaches the physical layer and is put out on the media. When the data arrives at the destination, it is passed back up through layers and decapsulated by each layer. Each layer provides data services to the layer directly above by preparing information coming down the model or going up.
Table 2-5 briefly describes each layer of the OSI model

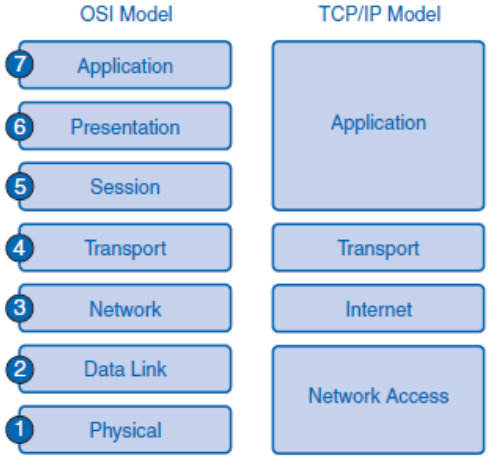
Table 2-5 OSI Model

No.	Layer Name	Description
7	Application	Performs services for the applications used by the end users.
6	Presentation	Provides data format information to the application. For example, the presentation layer tells the application layer whether there is encryption or whether it is a .jpg picture.
5	Session	Manages sessions between users. For example, the session layer will synchronize multiple web sessions and voice and video data in web conferences.
4	Transport	Defines data segments and numbers them at the source, transfers the data, and reassembles the data at the destination.
3	Network	Creates and addresses packets for end-to-end delivery through intermediary devices in other networks.
2	Data Link	Creates and addresses frames for host-to-host delivery on the local LANs and between WAN devices.
1	Physical	Transmits binary data over media between devices. Physical layer protocols define media specifications.

Comparing the OSI Model to the TCP/IP Model

The TCP/IP model evolved faster than the OSI model and is now more practical in describing network communication functions. The OSI model describes in detail functions that occur at the upper layers on the hosts, while networking is largely a function of the lower layers. Figure 2-11 shows the two models side by side for comparison. When juxtaposed, you can see that the functions of the application, presentation, and session layers of the OSI model are combined into one application layer in the TCP/IP model. The bulk of networking functions reside at the transport and the network layers, so they remain individual layers. TCP operates at the transport layer, and IP operates at the Internet layer. The data link and physical layers of the OSI model combine to make the network access layer of the TCP/IP model.

Figure 2-11 Comparing the OSI and TCP/IP Models



Network Addressing

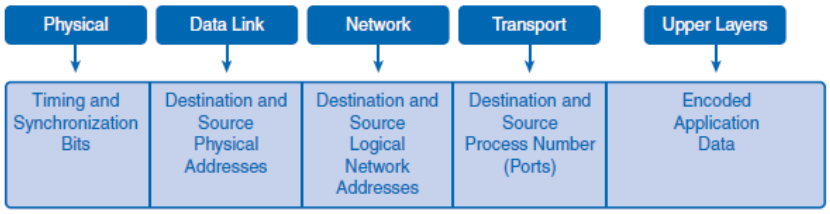
Successful communication requires that a sender and a receiver know how to get messages to each other. Postal systems use geography to deliver mail to physical addresses, but getting messages between computers is a more complicated matter. With the Internet, computers can communicate regardless of physical location.

Instead of using a geographical addressing scheme for computers, engineers devised a logical addressing scheme using numeric network addresses. The following sections introduce the addressing process. Chapter 6, "Addressing the Network: IPv4," explores network addressing in greater detail.

Addressing in the Network

There are millions of computers in use on the web and billions of messages traversing networks at any given time, so proper addressing is essential to make sure that the sent message arrives intact at the proper destination. Addressing of data happens in three different layers of the OSI model. The PDU at each layer adds address information for use by the peer layer at the destination. Figure 2-12 depicts the different addressing information added by each layer.

Figure 2-12 Addressing Added at Each Layer



Getting Data to the End Device

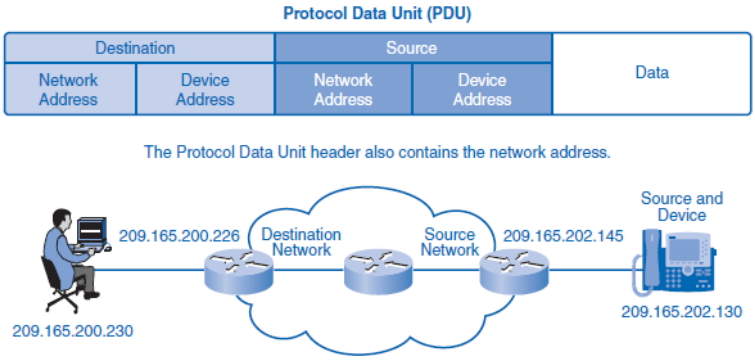
During the process of encapsulation, address identifiers are added to the data as it travels down the protocol stack on the source host. There are two layers of addressing added to ensure that data is delivered to the destination.

The first identifier, the host physical address, is contained in the header of the Layer 2 PDU, called a frame. Layer 2 is concerned with the delivery of messages on a single local network. The Layer 2 address is unique on the local network and represents the address of the end device on the physical media. The physical address comes from codes placed on the NIC by the manufacturer. In a LAN using Ethernet, this address is called the MAC address. The terms *physical address* and *MAC address* are often used interchangeably. When two end devices communicate on the local Ethernet network, the frames that are exchanged between them contain the destination and source MAC addresses. After a frame is successfully received by the destination host, the Layer 2 address information is removed as the data is decapsulated and moved up the protocol stack to Layer 3.

Getting Data Through the Internetwork

Layer 3 protocols are primarily designed to move data from one local network to another local network within an internetwork. Whereas Layer 2 addresses are only used to communicate between devices on a single local network, Layer 3 addresses must include identifiers that enable intermediary network devices to locate hosts on different networks. In the TCP/IP protocol suite, every IP host address contains information about the network where the host is located. At the boundary of each local network, an intermediary network device, usually a router, decapsulates the frame to read the destination host address contained in the header of the packet, the Layer 3 PDU. Routers use the network identifier portion of this address to determine which path to use to reach the destination host. When the path is determined, the router encapsulates the packet in a new frame and sends it on its way toward the destination end device. When the frame reaches its final destination, the frame and packet headers are removed and the data moved up to Layer 4. The journey from source to destination is depicted in Figure 2-13.

Figure 2-13 IP Addressing



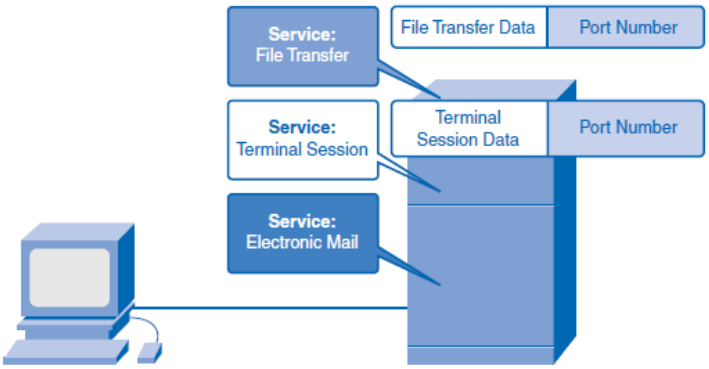
Getting Data to the Right Application

At Layer 4, information contained in the PDU header does not identify a destination host or a destination network. What it does identify is the specific process or service running on the destination host device that will act on the data being delivered. Hosts, whether they are clients or servers on the Internet, can run multiple network applications simultaneously. People using PCs often have an e-mail client running at the same time as a web browser, an instant messaging program, some streaming media, and perhaps even a game. All these separately running programs are examples of individual processes. Viewing a web page invokes at least one network process. Clicking a hyperlink causes a web browser to communicate with a web server. At the same time, in the background, an e-mail client can be sending and receiving e-mail, and a colleague or friend can be sending an instant message.

Think about a computer that has only one network interface on it. All the data streams created by the applications that are running on the PC enter and leave through that one interface; yet instant messages do not pop up in the middle of a word processor document or e-mail showing up in a game. This is because the transport layer adds ***port*** numbers to its segment header information to ensure that the destination host knows which application process is to receive the packet.

The end host assigns a port number to each type of traffic going in and out. A user can send and receive many types of traffic over a single network interface, and using port numbers for each segment keeps traffic for web pages separate from e-mail traffic and so on. The segment contains both source and destination ports in case the receiver needs to contact the sender. Figure 2-14 shows different data types for two different services on an end device.

Figure 2-14 Port Addressing



Presentation Layer

The presentation layer has three primary functions:

- Coding and conversion of application layer data to ensure that data from the ***source device*** can be interpreted by the appropriate application on the destination device.
- Compression of the data in a manner that can be decompressed by the destination device.
- Encryption of the data for transmission and decryption of data upon receipt by the destination layer

implementations are not typically associated with a particular protocol stack. The standards for video and graphics are examples. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding. Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF and JPEG are compression and coding standards for graphic images, and TIFF is a standard coding format for graphic images.

Session Layer

Functions at the session layer create and maintain dialogs between source and destination applications. The session layer handles the exchange of information to initiate dialogs and keep them active, and to restart sessions that are disrupted or idle for a long period of time.

TCP/IP Application Layer Protocols

The most widely known TCP/IP application layer protocols are those that provide the exchange of user information. These protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are the following:

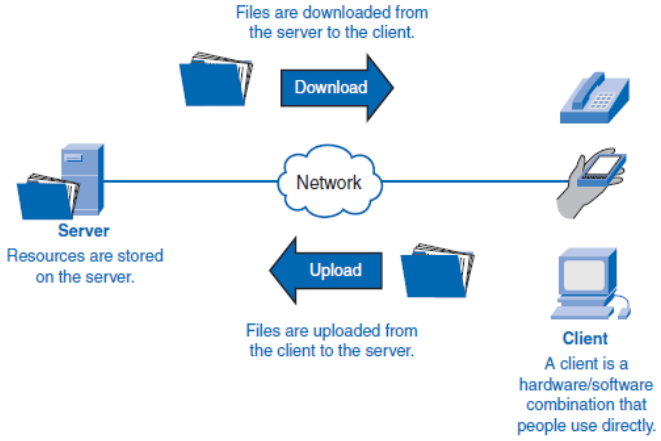
- ***Domain Name System (DNS)*** is used to resolve Internet names to IP addresses.
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the web pages of the World Wide Web.
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
- File Transfer Protocol (FTP) is used for interactive file transfer between systems.

The protocols in the TCP/IP suite are generally defined by ***Requests for Comments (RFC)***.
The Internet Engineering Task Force (IETF) maintains the RFCs as the standards for the TCP/IP suite.

Client/Server Model

In the client/server model, the device requesting the information is called a ***client*** and the device responding to the request is called a server. Client and ***server*** processes are considered to be in the application layer. The client begins the exchange by requesting data from the server, which responds by sending one or more streams of data to the client. Application layer protocols describe the design of the requests and responses between clients and servers. In addition to the actual data transfer, this exchange can require control information, such as user authentication and the identification of a data file to be transferred.
One example of a client/server network is a corporate environment where employees use a company e-mail server to send, receive, and store e-mail. The e-mail client on an employee computer issues a request to the e-mail server for any unread mail. The server responds by sending the requested e-mail to the client. Although data is typically described as flowing from the server to the client, some data always flows from the client to the server. Data flow can be equal in both directions or can even be greater in the direction going from the client to the server. For example, a client might transfer a file to the server for storage purposes. Data transfer from a client to a server is referred to as an *upload*, and data from a server to a client is a *download*. Figure 3-6 shows the client/server model concept.

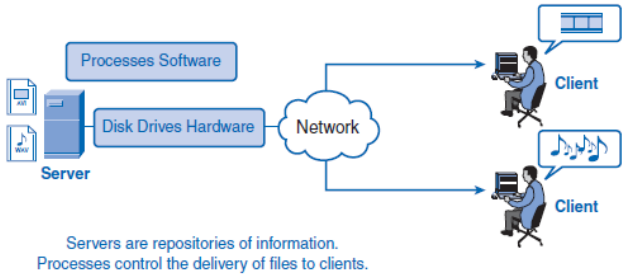
Figure 3-6 Client/Server Model



Servers

In a general networking context, any device that responds to requests from client applications is functioning as a server. A server is usually a computer that contains information to be shared with many client systems. For example, web pages, documents, databases, pictures, video, and audio files can all be stored on a server and delivered to requesting clients. In other cases, such as a network printer, the print server delivers the client print requests to the specified printer. Different types of server applications can have different requirements for client access. Some servers can require authentication of user account information to verify whether the user has permission to access the requested data or to use a particular operation. Such servers rely on a central list of user accounts and the authorizations, or permissions (both for data access and operations), granted to each user. When using an FTP client, for example, if you request to upload data to the FTP server, you might have permission to write to your individual folder but not to read other files on the site.

Figure 3-7 Servers



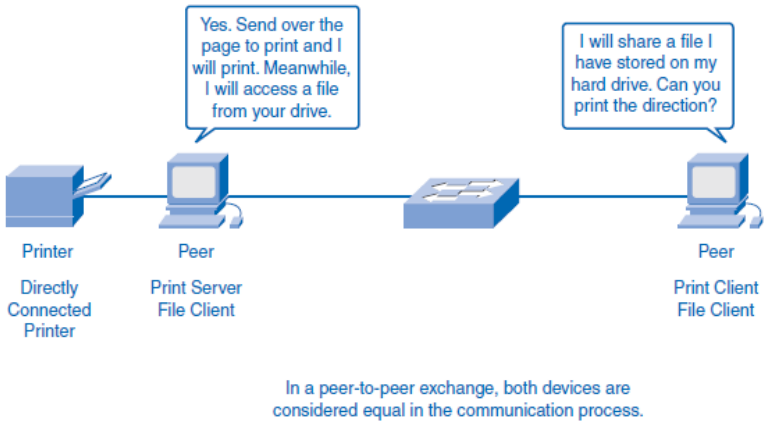
Peer-to-Peer (P2P) Networking and Applications

In addition to the client/server model for networking, there is a peer-to-peer (P2P) model. P2P networking involves two distinct forms: peer-to-peer network design and peer-to-peer applications. Both forms have similar features but in practice work very differently.

P2P Networks

In a peer-to-peer network, two or more computers are connected through a network and can share resources such as printers and files without having a dedicated server. Every connected end device, known as a peer, can function as either a server or a client. One computer might assume the role of server for one transaction while simultaneously serve as a client for another. The roles of client and server are set on a per-request basis, as shown in Figure 3-9. The figure shows one peer asking the other peer to provide print services, while at the same time acting as a file server that shares one of its files.

Figure 3-9 Peer-to-Peer Networking



A simple home network with two connected computers sharing a printer is an example of a peer-to-peer network. Each person can set his or her computer to share files, enable networked games, or share an Internet connection. Another example of peer-to-peer network functionality is two computers connected to a large network that use software applications to share resources between one another through the network.

Unlike the client/server model, which uses dedicated servers, peer-to-peer networks decentralize the resources on a network. Instead of locating information to be shared on dedicated servers, information can be located anywhere on any connected device. Most of the current operating systems support file and print sharing without requiring additional server software. Because peer-to-peer networks usually do not use centralized user accounts, permissions, or monitors, it is difficult to enforce security and access policies in networks containing more than just a few computers. User accounts and access rights must be set individually on each ***peer*** device.

Application Layer Protocols and Services Examples

Now that you have a better understanding of how applications provide an interface for the user and provide access to the network, you will take a look at some specific commonly used protocols. As you will see later in this book, the

transport layer uses an addressing ***scheme*** called a port number. Port numbers identify applications and application layer services that are the source and destination of data. Server programs generally use predefined port numbers that are commonly known by clients. As you examine the different TCP/IP application layer protocols and services, you will be referring to the TCP and UDP port numbers normally associated with these services.
Some of these services are:

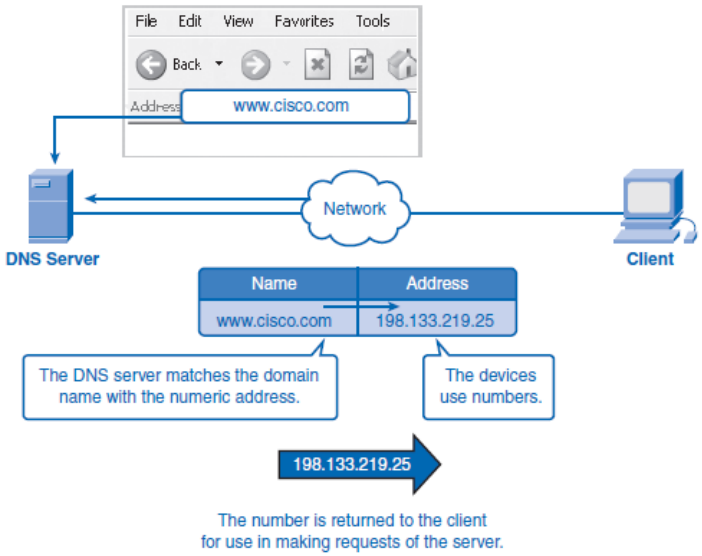
- **Domain Name System (DNS):** TCP/UDP port 53
- **HTTP:** TCP port 80
- **Simple Mail Transfer Protocol (SMTP):** TCP port 25
- **Post Office Protocol (POP):** UDP port 110
- **Telnet:** TCP port 23
- **DHCP:** UDP port 67
- **FTP:** TCP ports 20 and 21

The next sections take a closer look at DNS, world wide web services, and HTTP.

DNS Services and Protocol

In data networks, devices are assigned ***IP addresses*** so that they can participate in sending and receiving messages over the network. However, most people have a hard time remembering this numeric address. Hence, domain names were created to convert the numeric address into a simple, recognizable name.
On the Internet, these domain names, such as http://www.cisco.com, are much easier for people to remember than 198.132.219.25, which, at the time of this writing, is the numeric address for this server. Also, if Cisco decides to change the numeric address, it is transparent to the user, because the ***domain name*** will remain http://www.cisco.com. The new address will simply be linked to the existing domain name and connectivity is maintained, as shown in Figure 3-11. When networks were small, it was a simple task to maintain the mapping between domain names and the addresses they represented. However, as networks began to grow and the number of devices increased, this manual system became unworkable.

Figure 3-11 Resolving DNS Addresses



When a client makes a query, the "named" process first looks at its own records to see whether it can resolve the name. If it is unable to resolve the name using its stored records, it contacts other servers to resolve the name. The request can be passed along to a number of servers, which can take extra time and consume bandwidth. When a match is found and returned to the original requesting server, the server temporarily stores the numbered address that matches the name in the ***cache***. If that same name is requested again, the first server can return the address by using the value stored in its name cache. Caching reduces both the DNS query data network traffic and the workloads of servers higher up the hierarchy. The DNS client service on Windows PCs optimizes the performance of DNS name resolution by storing previously resolved names in memory, as well. The **ipconfig/displaydns** command displays all the cached DNS entries on a Windows XP or 2000 computer system.

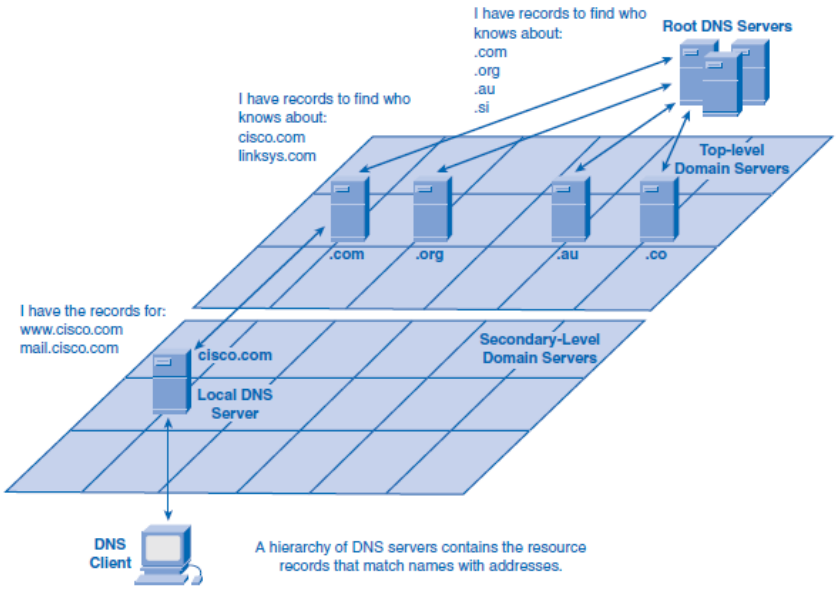
DNS Hierarchy

DNS uses a hierarchical system to create a name database to provide name resolution. The hierarchy looks like an inverted tree with the root at the top and branches below. At the top of the hierarchy, the root servers maintain records about how to reach the top level domain servers, which in turn have records that point to the secondary-level domain servers and so on. The different top-level domains represent either the type of organization or the country of origin. The following are examples of top-level domains are:

- **.au:** Australia
- **.co:** Colombia
- **.com:** A business or industry
- **.jp:** Japan
- **.org:** A nonprofit organization

After top-level domains are second-level domain names, and below them are other lower-level domains. A great example of that is the domain name http://www.cisco.netacad.net. The .net is the top-level domain, .netacad is the second-level domain, and .cisco is at the lower level. Each domain name is a path down this inverted tree starting from the root. For example, as shown in Figure 3-12, the root DNS servers might not know exactly where the e-mail server mail.cisco.com is located, but they maintain a record for the .com domain within the top-level domain. Likewise, the servers within the .com domain might not have a record for mail.cisco.com, but they do have a record for the cisco.com secondary-level domain. The servers within the cisco.com domain have a record (an MX record to be precise) for mail.cisco.com.

Figure 3-12 DNS Server Hierarchy



WWW Service and HTTP

When a web address (or URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server using HTTP. URLs and URIs (uniform resource identifiers) are the names most people associate with web addresses.

The URL http://www.cisco.com/index.html refers to a specific resource—a web page named index.html on a server identified as cisco.com. Web browsers are the client applications computers use to connect to the World Wide Web and access resources stored on a web server. As with most server processes, the web server runs as a background service and makes different types of files available. To access the content, web clients make connections to the server and request the desired resources. The server replies with the resources and, upon receipt, the browser interpret the data and present it to the user. To better understand how the web browser and web client interact, you can examine how a web page is opened in a browser. For this example, consider the URL **http://www.cisco.com/web-server.htm.**

First, the browser interprets the three parts of the URL:

- http: The protocol or scheme
- www.cisco.com: The server name
- web-server.htm: The specific filename requested

For secure communication across the Internet, the Secure HTTP (HTTPS) protocol is used for accessing and posting web server information. HTTPS can use authentication and ***encryption*** to secure data as it travels between the client and server. HTTPS specifies additional rules for passing data between the application layer and the transport layer.

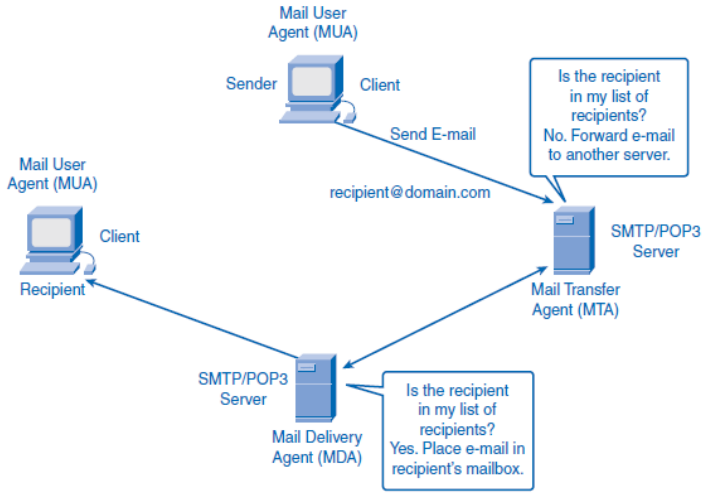
E-Mail Services and SMTP/POP Protocols

E-mail, the most popular network service, has revolutionized how people communicate through its simplicity and speed. Yet to run on a computer or other end device, e-mail requires several applications and services. Two examples of application layer protocols are ***Post Office Protocol (POP)*** and ***Simple Mail Transfer Protocol (SMTP)***. As with HTTP, these protocols define client/server processes.
POP and POP3 (Post Office Protocol, version 3) are inbound mail delivery protocols and are typical client/server protocols. They deliver e-mail from the e-mail server to the client (MUA).
SMTP, on the other hand, governs the transfer of outbound e-mail from the sending client to the e-mail server (MDA), as well as the transport of e-mail between e-mail servers (MTA). (These acronyms are defined in the next section.) SMTP enables e-mail to be transported across data networks between different types of server and client software and makes e-mail exchange over the Internet possible.
When people compose e-mail messages, they typically use an application called a ***Mail User Agent (MUA)***, or e-mail client. The MUA allows messages to be sent and places received messages into the client mailbox, both of which are distinct processes, as shown in Figure 3-14.

Figure 3-14 E-Mail Client (MUA)

To receive e-mail messages from an e-mail server, the e-mail client can use POP. Sending e-mail from either a client or a server uses message formats and command strings defined by the SMTP protocol. Usually an e-mail client provides the functionality of both protocols within one application.

Figure 3-16 E-Mail Server: MDA

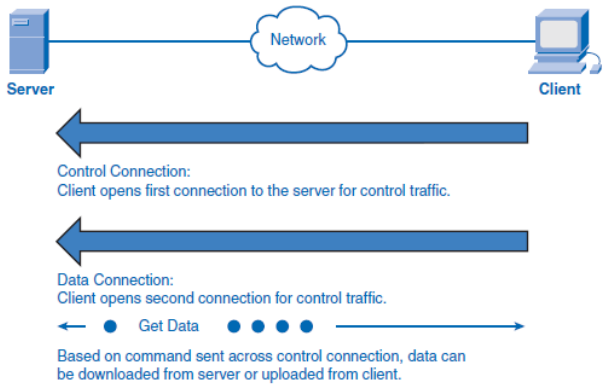


FTP

FTP is another commonly used application layer protocol. FTP was developed to allow file transfers between a client and a server. An FTP client is an application that runs on a computer that is used to push and pull files from a server running the FTP daemon (FTPd). To successfully transfer files, FTP requires two connections between the client and the server: one for commands and replies, and the other for the actual file transfer.

The client establishes the first connection to the server on TCP port 21. This connection is used for control traffic, consisting of client commands and server replies. The client establishes the second connection to the server over TCP port 20. This connection is for the actual file transfer and is created every time a file is transferred. The file transfer can happen in either direction, as shown in Figure 3-17. The client can download (pull) a file from the server or upload (push) a file to the server.

Figure 3-17 FTP Process



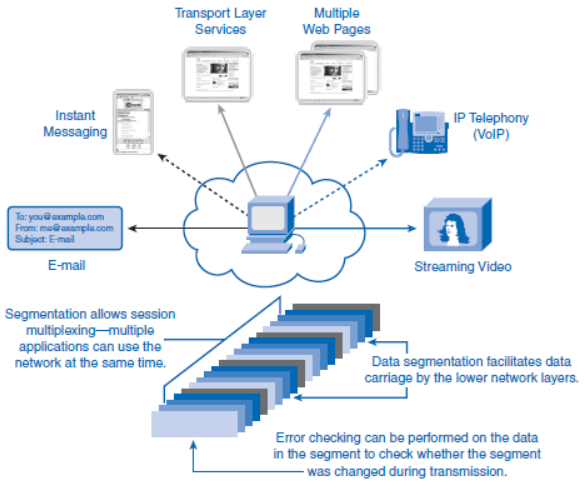
Purpose of the Transport Layer

The following are the primary responsibilities of the transport layer:

- Tracking the individual communications between applications on the source and destination hosts
- Segmenting data and managing each piece
- Reassembling the segments into streams of application data
- Identifying the different applications
- Performing flow control between end users
- Enabling error recovery
- Initiating a session

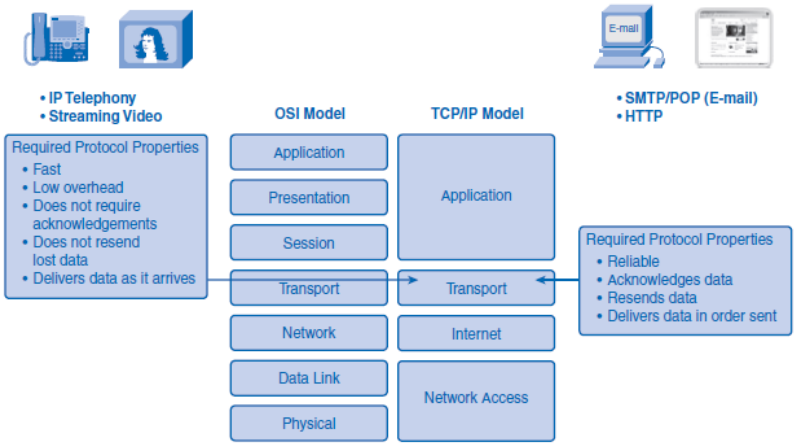
The transport layer enables applications on devices to communicate, as shown in Figure 4-2..

Figure 4-2 Enabling Applications on Devices to Communicate



Application developers must choose which transport protocol type is appropriate based on the requirements of their applications, as shown in Figure 4-5. At the transport layer, protocols specify methods for either reliable, guaranteed delivery or best-effort delivery. In the context of networking, best-effort delivery is referred to as unreliable, because the destination does not acknowledge whether it received the data.

Figure 4-5 Transport Layer Protocols



Applications, such as databases, web pages, and e-mail, require that all the sent data arrive at the destination in its original condition for the data to be useful. Any missing data could cause a corrupt communication that is either incomplete or unreadable. Therefore, these applications are designed to use a transport layer protocol that implements reliability. The additional network overhead is considered to be required for these applications.

Other applications are more tolerant of the loss of small amounts of data. For example, if one or two segments of a video stream fail to arrive, it would only create a momentary disruption in the stream. This can appear as distortion in the image but might not even be noticeable to the user. Imposing overhead to ensure reliability for this application could reduce the usefulness of the application. The image in a streaming video would be greatly degraded if the destination device had to account for lost data and delay the stream while waiting for its arrival. It is better to render the best image possible at the time with the segments that arrive and forego reliability. If reliability is required for some reason, these applications can provide error checking and retransmission requests.

TCP and UDP

The two most common transport layer protocols of the TCP/IP protocol suite are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Both protocols manage the communication of multiple applications. The differences between the two are the specific functions that each protocol implements.

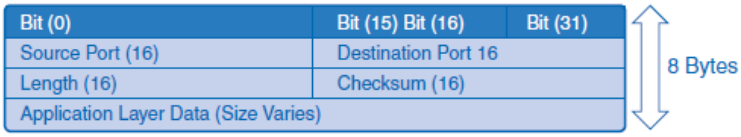
User Datagram Protocol (UDP)

UDP is a simple, connectionless protocol, described in RFC 768. It has the advantage of providing low-overhead data delivery. The segments of communication in UDP are called *datagrams*. UDP sends datagrams as "best effort." Applications that use UDP include:

- Domain Name System (DNS)
- Video streaming
- Voice over IP (VoIP)

Figure 4-6 illustrates a UDP datagram.

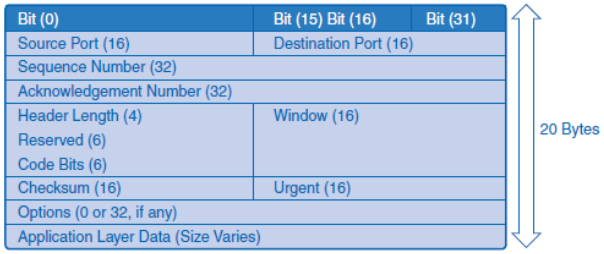
Figure 4-6 UDP Datagram



Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol, described in RFC 793. TCP incurs additional overhead to gain functions. Additional functions specified by TCP are same-order delivery, reliable delivery, and flow control. Each TCP segment has 20 bytes of overhead in the header encapsulating the application layer data, whereas each UDP segment has only 8 bytes of overhead. Figure 4-7 shows the TCP datagram

Figure 4-7 TCP Datagram



. The following applications use TCP:

- Web browsers
- E-mail
- File transfers

TCP Congestion Control: Minimizing Segment Loss

TCP provides congestion control through the use of flow control and dynamic window sizes. The following sections discuss how these techniques minimize segment loss that minimizes network overhead caused by retransmission of lost segments.

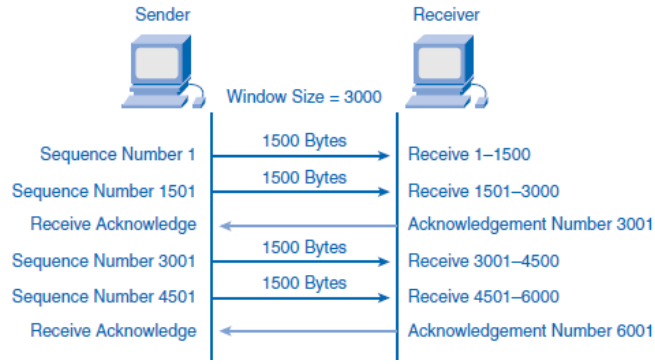
Flow Control

Flow control assists the reliability of TCP transmission by adjusting the effective rate of data flow between the two services in the session. When the source is informed that the specified amount of data in the segments is received, it can continue sending more data for this session. The window size field in the TCP header specifies the amount of data that can be transmitted before an acknowledgment must be received. The initial window size is determined during the session startup through the three-way handshake.

The TCP feedback mechanism adjusts the effective rate of data transmission to the maximum flow that the network and destination device can support without loss. TCP attempts to manage the rate of transmission so that all data will be received and retransmissions will be minimized.

Figure 4-14 shows a simplified representation of window size and acknowledgments. In this example, the initial window size for a TCP session represented is set to 3000 bytes. When the sender has transmitted 3000 bytes, it waits for an acknowledgment of these bytes before transmitting more segments in this session. After the sender has received this acknowledgment from the receiver, the sender can transmit an additional 3000 bytes.

Figure 4-14 TCP Segment Acknowledgment and Window Size

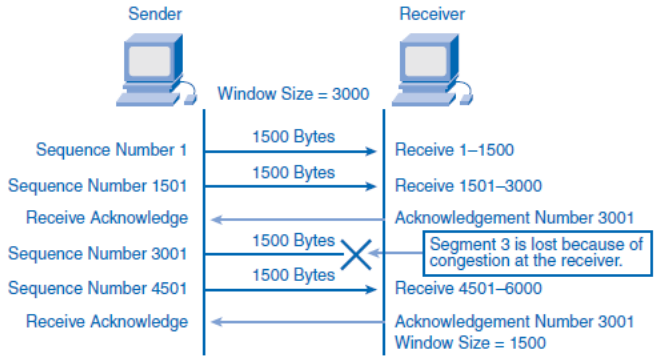


During the delay in receiving the acknowledgment, the sender will not send additional segments for this session. In periods when the network is congested or the resources of the receiving host are strained, the delay can increase. As this delay grows longer, the effective transmission rate of the data for this session decreases. The slowdown in data rate helps reduce the resource contention.

Dynamic Window Sizes

Another way to control the data flow is to use dynamic window sizes. When network resources are constrained, TCP can reduce the window size to require that received segments be acknowledged more frequently. This effectively slows the rate of transmission because the source must wait for data to be acknowledged. The TCP receiving host sends the window size value to the sending TCP to indicate the number of bytes that it is prepared to receive as a part of this session. If the destination needs to slow the rate of communication because of limited buffer memory, it can send a smaller window size value to the source as part of an acknowledgment. As shown in Figure 4-15, if a receiving host has congestion, it can respond to the sending host with a segment with a reduced window size. Figure 4-15 shows a loss of one of the segments. The receiver changed the window size field in the TCP header of the returning segments in this conversation from 3000 to 1500. This caused the sender to reduce the window size to 1500.

Figure 4-15 TCP Congestion and Flow Control



After periods of transmission with no data losses or constrained resources, the receiver will begin to increase the window size field. This reduces the overhead on the network because fewer acknowledgments need to be sent. Window size will continue to increase until data loss occurs, which will cause the window size to be decreased.

This dynamic increasing and decreasing of window size is a continuous process in TCP, which determines the optimum window size for each TCP session. In highly efficient networks, window sizes can become very large because data is not being lost. In networks where the underlying infrastructure is being stressed, the window size will likely remain small.

CHAPTER THREE
Computer Network Classification and connectivity devices

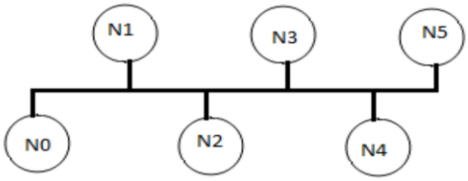
Computer Network Topology

PHYSICAL TOPOLOGY

Physical Network Topology emphasizes the hardware associated with the system including workstations, remote terminals, servers, and the associated wiring between assets. Physical topology defines how the systems are physically connected. It means the arrangement of devices on a computer network through the actual cables that transmit data. The shape of the cabling layout used to link devices is called the physical topology of the network. This refers to the layout of cabling, the locations of nodes, and the interconnections between the nodes and the cabling. The physical topology of a network is determined by the capabilities of the network access devices and media, the level of control or fault tolerance desired, and the cost associated with cabling or telecommunications circuits. There are five basic topologies. In below each of these topologies are described_

BUS TOPOLOGY

The bus topology carries the transmitted message along the cable. As the message arrives at each device (node), the nodes checks the destination address contained in the message to see if it matches its own. In this topology, a single network cable runs in the building or campus and all nodes are linked along with this communication line with two endpoints called the bus or backbone. By this type of topology, if one node goes faulty all nodes may be affected as all nodes share the same cable for the sending and receiving of information. The cabling cost of bus systems is the least of all the different topologies. Each end of the cable is terminated using a special terminator.



A. Advantages

1) Reliable in very small networks as well as easy to use and understand.
2) Requires least amount of cable to connect the computers (nodes) together and therefore is less expensive than other cabling arrangements.
3) It's easy to extend, Two cables can be easily joined with a connector, making a longer cable for more computers to join the network.
4) A repeater can also be used to extend a bus configuration.

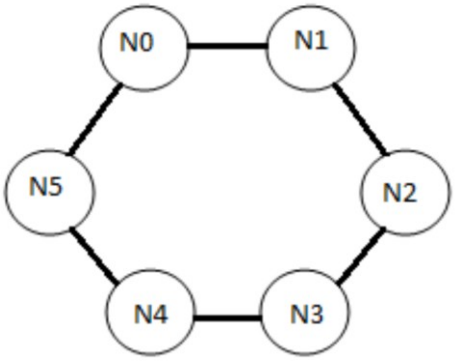
B. Disadvantages

1) Heavy network traffic can slow a bus considerably because any computer can transmit at any time. But networks do not
2) Coordinate when information is sent. Computer interrupting each other can use a lot of bandwidth.
3) Each connection between two cables weakens the electrical signal.
4) The bus configuration can be difficult to find and can cause the whole networks to stop functioning.

IV. RING TOPOLOGY

In a ring topology, every device has exactly two neighbors for communication purposes. All messages travel through a ring in the same direction (either “clockwise” or “counter clock wise”). There is a direct point-to-point link between two neighboring nodes (the Next and the Previous). These links are unidirectional which ensures that transmission by a node traverses the whole ring and comes back to the node, which made the transmission as shown in figure_

Ring



Faulty nodes can be isolated from the ring. When the workstation is powered on, it connects itself to the ring. When power is off, it disconnects itself from the ring and allows the information to bypass the node.
The most common implementation of this topology is token ring. A break in the ring causes the entire network to fail. Individual nodes can be isolated from the ring.

A. Advantages

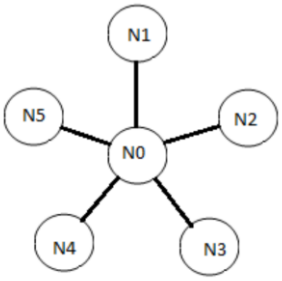
1) Ring networks offer high performance for a small number of workstations or for larger networks where each station has a similar workload.
2) Ring networks can span longer distances than other types of networks.
3) Ring networks are easily extendable.
4) Unlike Bus topology, there is no signal loss in Ring topology because the tokens are data packets that are re-generated at each node.

B. Disadvantages

1) Relatively expensive and difficult to install
2) Failure of one computer on the network can affect the whole network.
3) It is difficult to find fault in a ring network.
4) Adding or removing computers can disrupt the network.

V. STAR TOPOLOGY

Star topology uses a central hub through which, all components are connected. In a Star topology, the central hub is the host computer, and at the end of each connection is a node. Nodes communicate across the network by passing data through the hub. A star network uses a significant amount of cable as each node is wired back to the central hub, even if two nodes are side by side but several hundred meters away from the host. The central hub makes all routing decisions, and all other workstations can be simple. An advantage of the star topology is that failure, in one of the nodes does not affect any other node; however, failure of the central hub affects all terminals. This type of topology is frequently used to connect terminals to a large time-sharing host computer. Many home networks use the star topology.



A. Advantages

1) It is more reliable (if one connection fails, it does not affect others)
2) The center of a star network is a good place to diagnose network faults and if one computer fails whole network is not disturbed. Hub detects the fault and isolates the faulty computer.
3) It is easy to replace, install or remove hosts or other of the network by simply running a new line from the computer to the central location and plugging it to the hub.
4) Use of multiple cable types in a same network with a hub.
5) It has good performance

B. Disadvantages

1) It is expensive to install as it requires more cable, it costs more to cable a star network because all network cables must be pulled to one central point, requiring more cable length than other networking topologies.

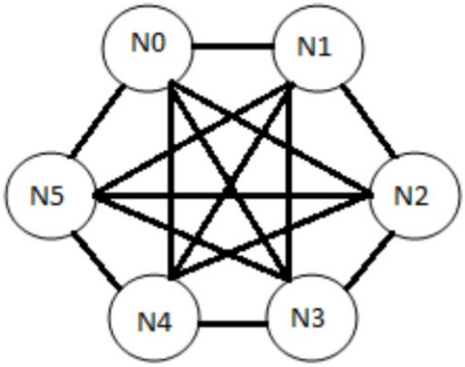
2) Central node dependency, if central hub fails, the whole network fails to operate.

3) Many star networks require a device at the central point to rebroadcast or switch the network traffic.

VI. MESH TOPOLOGY

Mesh topologies involve the concept of routes. Devices are connected with many redundant interconnections between network nodes. In a well-connected topology, every node has a connection to every other node in the network. The cable requirements are high, but there are redundant paths built in.

Failure in one of the computers does not cause the network to break down, as they have alternative paths to other computers.



Mesh topologies are used in critical connection of host computers (typically telephone exchanges). Alternate paths allow each computer to balance the load to other computer systems in the network by using more than one of the connection paths available.

A fully connected mesh network therefore has no (n-1) /2 physical channels to link n devices. To accommodate these, every device on the network must have (n-1) input/output ports.

A. Advantages

1) Yield the greatest amount of redundancy in the event that one of the nodes fails where network traffic can be redirected to another node.

2) Point-to-point link makes fault isolation easy.

3) Privacy between computers is maintained as messages travel along dedicated path.

4) Network problems are easier to diagnose.

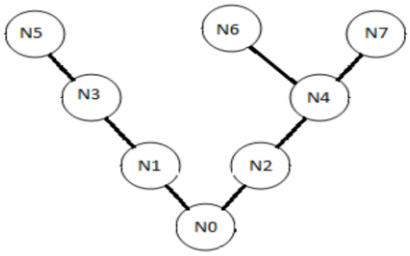
B. Disadvantages

1) The amount of cabling required is high.

2) A large number of I/O (input/output) ports are required.

VII. TREE TOPOLOGY

The most common topology known as Tree topology, Tree topology is a LAN topology in which only one route exists between any two nodes on the network. The pattern of connection resembles a tree in which all branches spring from one root.



Tree topology is a hybrid topology, it is similar to the star topology but the nodes are connected to the secondary hub, which in turn is connected to the central hub. Tree topology is a combination of two or more bus and the star topology. In this topology group of star-configured networks are connected to a linear bus backbone.

A. Advantages

1) Installation and configuration of network are easy.

2) The addition of the secondary hub allows more devices to be attached to the central hub.
3) Less expensive when compared to mesh topology.
4) Faults in the network can be detected traces.

B. Disadvantages

1) Failure in the central hub brings the entire network to a halt.
2) More cabling is required when compared to the bus topology because each node is connected to the central hub.

LOGICAL TOPOLOGY

Logical Network Topology emphasizes the representation of data flow between nodes. It means logical topology is associated with the arrangement of devices on a computer network and how they communicate with one another. The main role of logical topology is to communicate across the physical topologies among different systems. Logical topologies are often closely associated with Media Access Control methods and protocols. Logical topologies are able to be dynamically reconfigured by special types of equipment such as routers and switches. There are two categories of logical topologies: Shared media topology and token-based topology.

A. Shared Media Topology

In shared media topology the systems have unrestricted access to the physical media that is all the systems in a network have the ability to access the physical layout whenever they need it. Collision is the main disadvantage of this topology as more than one system send information out on the wire at the same time, the packets collide and as a result this collision kills the packets. Ethernet is an example of a shared media topology. As a remedy some huge networks are broken down into smaller networks. Some Ethernet uses Carrier Sense Multiple Access protocol to reduce the number of collisions.

B. Token Based Topology

In token based topology a token is used which travels around the network to access the physical media. If any node wants to send a packet to another one it should wait for the token which is traverse within the network either clockwise or anti-clockwise direction.

Components of the Network

The path that a message takes from source to destination can be as simple as a single cable connecting one computer to another or as complex as a network that literally spans the globe. This network infrastructure is the platform that supports our human network. It provides the stable and reliable channel over which our communications can occur.
Devices and media are the physical elements or hardware of the network. Hardware is often the visible components of the network platform such as a laptop, a PC, a ***switch***, or the cabling used to connect the devices. Occasionally, some components might not be so visible.
In the case of wireless media, messages are transmitted through the air using invisible radio frequency or infrared waves. Services and processes are the communication programs, called software, that run on the networked devices. A network service provides information in response to a request. Services include many of the common network applications people use every day, like e-mail hosting services and web hosting services. Processes provide the functionality that directs and moves the messages through the network. Processes are less obvious to us but are critical to the operation of networks.

End Devices and Their Role on the Network

An ***end device*** refers to a piece of equipment that is either the source or the destination of a message on a network. Network users usually only see and touch an end device, which is most often a computer. Another generic term for an end device that sends or receives messages is a ***host***. A host can be one of several pieces of equipment performing a wide variety of functions. Examples of hosts and end devices are as follows:

- Computers, including workstations, laptops, and servers connected to a network
- Network printers
- Voice over Internet Protocol (VoIP) phones
- Cameras on a network, including webcams and security cameras
- Handheld devices such as PDAs and handheld scanners
- Remote monitoring stations for weather observation

An end user is a person or group using an end device. Not all end devices are operated by people all of the time, though. For example, file servers are end devices that are set up by people but perform their tasks on their own. Servers are hosts that are set up to store and share information with other hosts called ***clients***. Clients request information and services, like e-mail and web pages, from servers, and servers reply with the requested information if they recognize the client.
When hosts communicate with each other, they use addresses to find each other. The ***host address*** is a unique ***physical address*** used by hosts inside a local-area network (LAN), and when a host sends a message to another host, it uses the physical address of the destination device.

Intermediary Devices and Their Role on the Network

End devices are the hosts that initiate communications and are the ones that people are most familiar with. But getting a message from the source to the destination can be a complex task involving several ***intermediary devices*** along the way. Intermediary devices connect the individual hosts to the network and can connect multiple individual networks to form an internetwork. Intermediary devices are not all the same. Some work inside the LAN performing switching functions, and others help route messages between networks. Table 2-1 lists some intermediary devices and their functions.

Table 2-1 Intermediary Devices

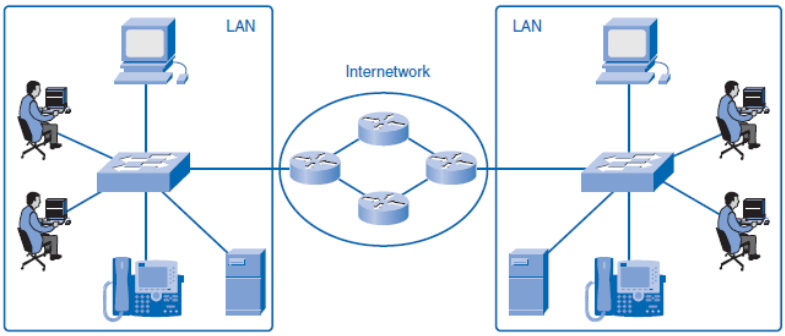
Device Type	Description
Network access devices	Connect end users to their network. Examples are hubs, switches, and wireless access points.
Internetwork devices	Connect one network to one or more other networks. Routers are the main example.
Communication servers	Route services such as IPTV and wireless broadband.
Modems	Connect users to servers and networks through telephone or cable.
Security devices	Secure the network with devices such as firewalls that analyze traffic exiting and entering networks.

The management of data as it flows through the network is also a role of the intermediary devices. These devices use the destination host address, in conjunction with information about the network interconnections, to determine the path that messages should take through the network. Processes running on the intermediary network devices perform these functions:

- Regenerate and retransmit data signals
- Maintain information about what pathways exist through the network and internetwork
- Notify other devices of errors and communication failures
- Direct data along alternate pathways when there is a link failure
- Classify and direct messages according to quality of service (QoS) priorities
- Permit or deny the flow of data, based on security settings

Figure 2-3 depicts two LANs with end devices connected by intermediary switches in the LANs and routers between the LANs.

Figure 2-3 LANs Connected by Routers



LANs, WANs, and Internetworks

Networks come in many sizes and serve a wide variety of functions. Following are some of the basic differences:

- The size of the area covered
- The number of users connected
- The number and types of services available

Three distinct groups of networks accommodate different groups and extend geographic boundaries: local-area networks (LANs), wide-area networks (WANs), and internetworks.

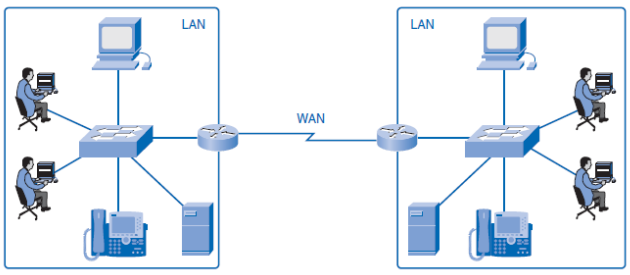
Local-Area Networks

A ***local-area network (LAN)*** is a group of end devices and users under the control of a common administration. The term *local* first meant that the computers were grouped geographically close together and had the same purpose in an organization. This is still true in many situations, but as technologies evolve, the definition of local has evolved as well. A LAN can consist of one group of users on one floor, but the term can also be used to describe all users on a multibuilding campus.

Wide-Area Networks

A wide-area network (WAN) is a network that is used to connect LANs that are located geographically far apart. If a company has offices in different cities, it will contract with a telecommunications service provider (TSP) to provide data lines between LANs in each city. The leased lines will vary in service and bandwidth, depending on the terms of the contract. The TSP is responsible for the intermediary devices on the WAN that transports messages, while LANs at both ends are controlled by the company. The sole purpose of WANs is to connect LANs, and there are usually no end users on WANs. Figure 2-5 depicts two LANs connected by a WAN.

Figure 2-5 Network with a WAN Connection



The Internet: A Network of Networks

In years past, LANs changed the way people worked, but they were limited to the resources within each network. Now workers who are not restricted to their own LAN can access other LANs on an internetwork. An *internetwork* is a collection of two or more LANs connected by WANs. Internetworks are referred to interchangeably as *data networks* or simply *networks*. The most popular internetwork is the Internet, which is open to public use. With LANs able to communicate with other LANs using WANs, many organizations developed intranets. A term often confused with the Internet, an *intranet* is a private web of networks closed to the public but open for employees to browse. For

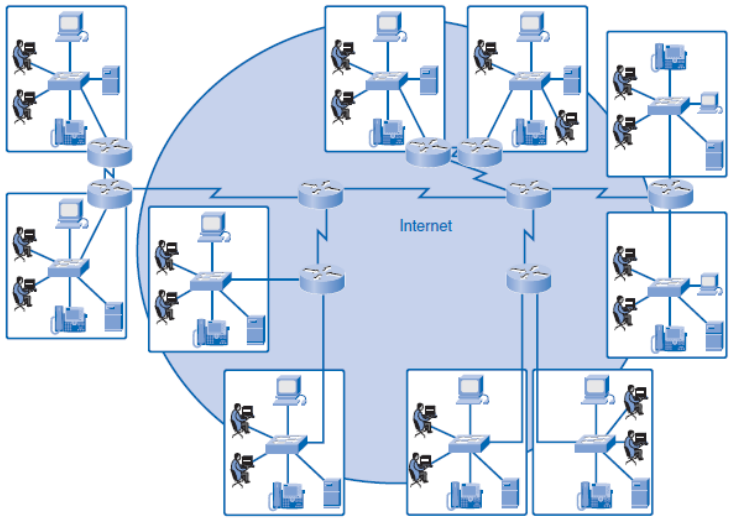
example, many companies use intranets to share company information and training across the globe to far-away employees. Documents are shared and projects are managed securely over great distances on an intranet.
Internet service providers (ISP), which are often also TSPs, connect their customers to the Internet. The customer can be a home user, a company, or a government institution. All Internet users access the web through ISPs. The ISPs cooperate with TSPs and other ISPs to make sure that all users have access to the web. This involves implementing rules and standards that enable any user to communicate with any other user regardless of location and equipment type. Figure 2-6 demonstrates how many WANs connect to form the Internet. Note the difference in symbols representing LAN connections to routers and the WAN connections between routers.

Network Representations

Chapter 1, "Living in a Network-Centric World," introduced many common data network symbols pictured in Figure 2-7. When discussing how devices and media connect to each other, remember these important terms:

- **Network interface card (NIC):** A *NIC*, or *LAN adapter*, provides the physical connection to the network at the PC or other host device. The media connecting the PC to the networking device plugs directly into the NIC. Each NIC has a unique physical address that identifies it on the LAN.
- **Physical port:** A *physical port* is a connector or outlet on a networking device where the media is connected to a host or other networking device. You can assume that all network host devices used in this book have a physical port that allows a connection to the network.
- **Interface:** The term *interface* refers to how the device can allow two different networks to communicate. Routers connect to different networks, and the specialized NICs on routers are simply called interfaces. The interface on a router device has a unique physical address and appears as a host on the local network.

Figure 2-6 Internetworks Made Up of LANs and WANs



Hub:

A hub, also called a network hub, is a common connection point for devices in a network. Hubs are devices commonly used to connect segments of a LAN. The hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets. Hubs and switches serve as a central connection for all of your network equipment and handles a data type known as frames. Frames carry your data. When a frame is received, it is amplified and then transmitted on to the port of the destination PC.

Historical Progression: Hubs, Bridges, and Switches

First, think back to the first UTP-based Ethernet standard, 10BASE-T, introduced in 1990. 10BASE-T used a centralized cabling model similar to today's Ethernet LANs, with each device connecting to the LAN using a UTP cable. However, instead of a LAN switch, the early 10BASE-T networks used hubs, because LAN switches had not yet been created. Figure 6-1 depicts the typical topology for 10BASE-T with a hub.

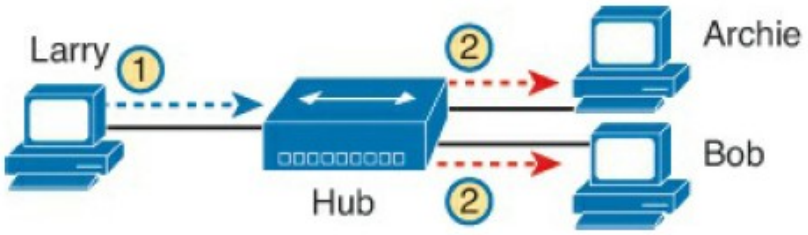


Figure 6-1. 10BASE-T (with a Hub)

Although using 10BASE-T with a hub improved Ethernet as compared to the older standards, several drawbacks continued to exist, even with 10BASE-T using hubs:
When hubs receive an electrical signal in one port (step 1 in Figure 6-1), the hub repeats the signal out all other ports (step 2 in the figure).
When two or more devices send at the same time, an electrical collision occurs, making both signals corrupt.
As a result, devices must take turns by using carrier sense multiple access with collision detection (CSMA/CD) logic, so the devices share the (10-Mbps) bandwidth.
Broadcasts sent by one device are heard by, and processed by, all other devices on the LAN.
Unicast frames are heard by all other devices on the LAN.
Over time, the performance of many Ethernet networks based on hubs started to degrade. People developed applications to take advantage of the LAN bandwidth. More devices were added to each Ethernet. However, the devices on the same Ethernet could not send (collectively) more than 10 Mbps of traffic because they all shared the 10 Mbps of bandwidth. In addition, the increase in traffic volumes resulted in an increased number of collisions, requiring more retransmissions and wasting more LAN capacity.
Ethernet transparent bridges, or simply bridges, helped solve this performance problem with 10BASE-T. After they were added to a 10BASE-T LAN, the following improvements were made:
Bridges separated devices into groups called *collision domains*.
Bridges reduced the number of collisions that occurred in the network, because frames inside one collision domain did not collide with frames in another collision domain.
Bridges increased bandwidth by giving each collision domain its own separate bandwidth, with one sender at a time per collision domain.
Figure 6-2 shows the effect of migrating from using a 10BASE-T hub without a bridge (as in Figure 6-1) to a network that uses a bridge. The bridge in this case separates the network into two separate collision domains (CD).

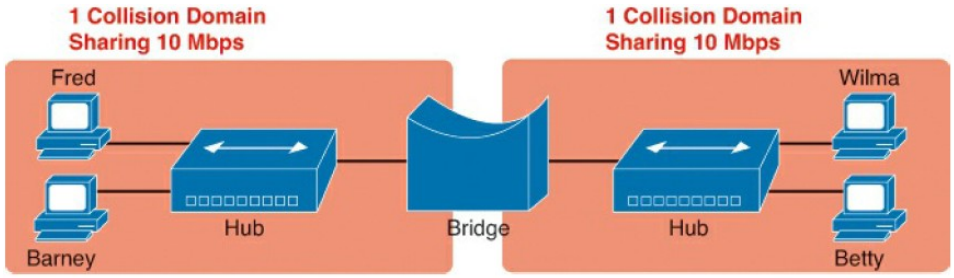


Figure 6-2. Bridge Creates Two Collision Domains and Two Shared Ethernets

The bridge, a predecessor to today's Ethernet LAN switch, uses logic so that the frames in one CD do not collide with frames in the other CD. The bridge forwards frames between its two interfaces, and unlike a hub, a bridge will buffer or queue the frame until the outgoing interface can send the frame. For example, Fred and Betty can both send a frame to Barney at the same time, and the bridge will queue the frame sent by Betty, waiting to forward it to the CD on the left, until the CD on the left is not busy.
Adding the bridge in Figure 6-2 really creates two separate 10BASE-T networks—one on the left and one on the right. The 10BASE-T network on the left has its own 10 Mbps to share, as does the network on the right. So, in this example, the total network bandwidth is doubled to 20 Mbps as compared with the 10BASE-T network in Figure 6-1, because devices on each side of the network can send at 10 Mbps at the same time.
LAN switches perform the same basic core functions as bridges, but at much faster speeds and with many enhanced features. Like bridges, switches segment a LAN into separate collision domains, each with its own capacity. And if the network does not have a hub, each single link is considered its own CD, even if no collisions can actually occur in that case.
For example, Figure 6-3 shows a simple LAN with a switch and four PCs. The switch creates four CDs, with the ability to send at 100 Mbps in this case on each of the four links. And with no hubs, each link can run at full-duplex, doubling the capacity of each link.

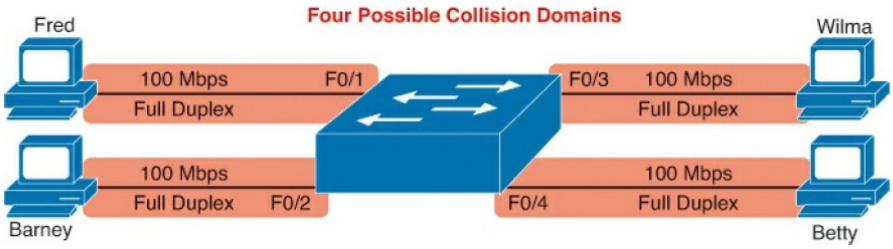


Figure 6-3. Switch Creates Four Collision Domains and Four Ethernet Segments

Switching Logic

Ultimately, the role of a LAN switch is to forward Ethernet frames. To achieve that goal, switches use logic—logic based on the source and destination MAC address in each frame's Ethernet header.

This book discusses how switches forward unicast frames and broadcast frames, ignoring multicast Ethernet frames. Unicast frames have a unicast address as a destination; these addresses represent a single device. A broadcast frame has a destination MAC address of FFFF.FFFF.FFFF; this frame should be delivered to all devices on the LAN.

The Forward-Versus-Filter Decision

To decide whether to forward a frame, a switch uses a dynamically built table that lists MAC addresses and outgoing interfaces. Switches compare the frame's destination MAC address to this table to decide whether the switch should forward a frame. For example, consider the simple network shown in Figure 6-4, with Fred sending a frame to Barney.

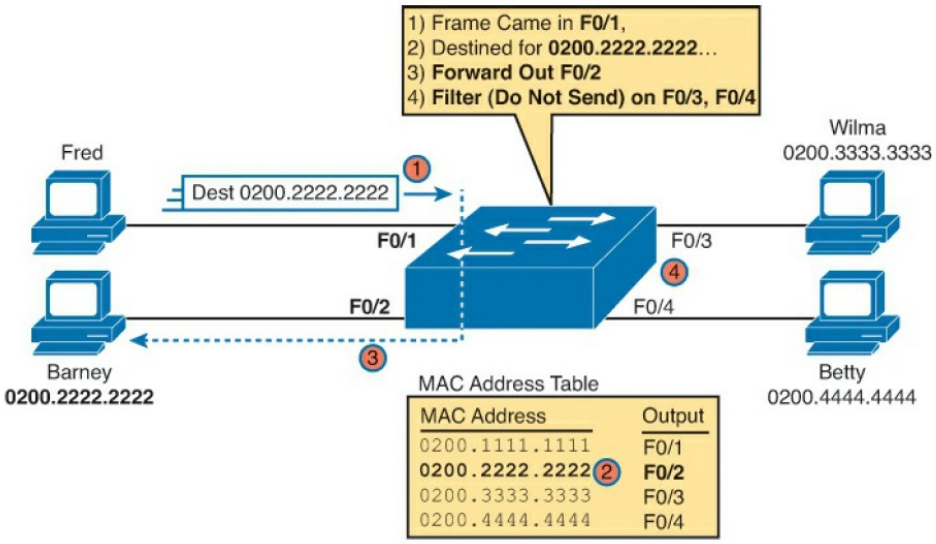
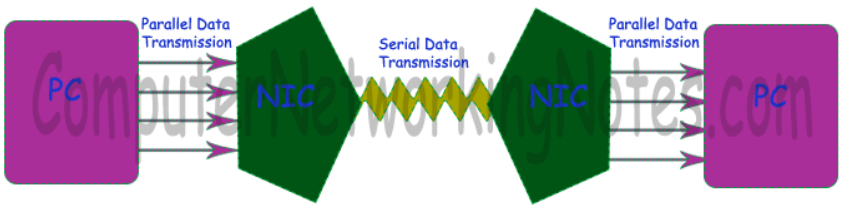


Figure 6-4. Sample Switch Forwarding and Filtering Decision

In this figure, Fred sends a frame with destination address 0200.2222.2222 (Barney's MAC address). The switch compares the destination MAC address (0200.2222.2222) to the MAC address table, matching the bold table entry. That matched table entry tells the switch to forward the frame out port F0/2, and only port F0/2.

Network Interface Card (NIC)

In the list of networking devices, NIC stands on first place. Without this device, networking cannot be done. This is also known as network adapter card, Ethernet Card and LAN card. NIC allows our PC to communicate with other PCs. Basically it converts data transmission technology. A PC uses parallel data transmission technology to transmit data between its internal parts while the media that connects this PC with other PCs uses serial data transmission technology. A NIC converts parallel data stream into serial data stream and vice versa serial data stream is get converted in parallel data stream.



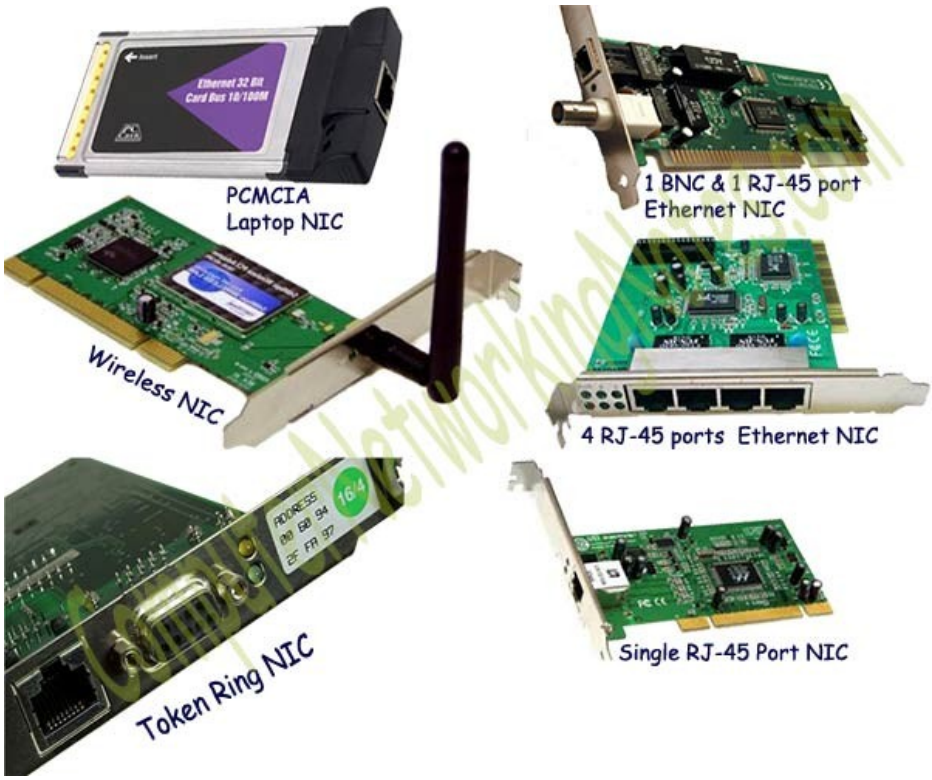
Usually all modern PCs have integrated NICs in motherboard. NICs are also available separately. For desktop or server system they are available in adapter format which can be plugged into the available slots of motherboard. For laptop or other small size devices they available in PCMCIA (Personal Computer Memory Card International Association) card format which can be inserted in PCMCIA slots.

Types of NICs

There are two types of NICs

Media Specific :- Different types of NICs are required to connect with different types of media. For example we cannot connect wired media with wireless NIC card. Just like this, we cannot connect coaxial cable with Ethernet LAN card. We have to use the LAN card that is particularly built for the media type which we have.

Network Design Specific :- A specific network design needs a specific LAN card. For example FDDI, Token Ring and Ethernet have their own distinctive type of NICs card. They cannot use other's NIC card.



HUB

HUB is used to connect multiple computers in a single workgroup LAN network. Typically HUBs are available with 4,8,12,24,48 ports. Based on port type, there are two types of HUB:-

Ethernet HUB :- In this type of HUB all ports have RJ-45 connectors.

Combo HUB :- In this type of HUB ports have several different types of connectors such RJ-45, BNC, and AUI.

HUBs generally have LED (light-emitting diode) indicator lights on each port to indicate the status of link, collisions, and other information.

To understand the functionality of hub let's take an example from real life.

There are four friends who share everything. One of them finds a photo of Amitabh Bachchan. To share this with friends, he will make three photo copies from Xerox machine and give one copy to each friend. He doesn't need a copy of photo for himself as he has the original one.

Now change the characters in this example. Replace friends with HUB's port, photo with data signal and Xerox machine with HUB.

There is a HUB which has four ports. Ports share everything. One port received data signal from its connected device. It will make three copies of data signal from HUB and give one copy to each port. Receiver port doesn't need a copy of data signal for itself as it has it the original version.

This is what exactly a HUB do. When a hub receives signal on its port, it repeats the signal and forwards that signal from all ports except the port on which the signal arrived.

There are two types of HUB

Passive HUB:- It forwards the data signal from all ports except the port on which signal arrived. It doesn't interfere in data signal.

Active HUB:- It also forwards the data signal from all ports except the port on which signal arrived. But before forwarding, it **improves the quality of data signal** by **amplifying** it. Due to this added features active HUB is also known as repeaters.

Usually HUB has one or more uplink ports that is used to connect it with another HUB. Right cable type is required to connect two HUBs.

Logically HUB creates a star topology where it sits in the center of the topology and all connected systems stay at the points of the star.

Physically HUB creates a bus topology where all connected systems share the same bus connection.

There are two similar devices to HUB, MAU and Patch Panel.

Patch Panel :- It is used to organize the UTP cables systematically. It doesn't interfere in data signal.

HUB and Repeater works at Layer 1 (Physical layer). These devices only understand the signals. Signals received on incoming port are forwarded from all available ports.

While signals are being forwarded no other ports should receive another signals (data) form their connected devices. If other port receives signals (data) while current transmission is going on, it will create collision.

Collision

Collision is the effect of two devices sending transmissions simultaneously in Ethernet. When they meet on the physical media, the signals from each device collide and damage.

Collision domain

Collision domain is the group of devices that share same collision effects over the Ethernet network.

CSMA/CD

It is a mechanism of removing collision from network. When two or more nodes simultaneously sense the wire and find no signals, they assume that wire is available for transmission. So they all put their own signal in wire simultaneously. These signals collide in wire and create a collision. CSMA/CD solves this problem. Let's understand this mechanism in little bit more detail.

Before placing any signal in wire, NIC (Network Interface Card) examines the wire for any existing signal. This method is known as **CS** (*Carrier Sense*).

If two NICs sense wire on exactly same time and see no signal then both will place their signals in wire. This is known as **MA** (Multiple Access).

If the NICs see a collision for their transmitted signals, they have to resend the signals. In this situation, each NIC that was transmitting a frame when a collision occurred creates a special signal, called a jam signal, on the wire, waits a small random time period, and examine the wire again. If there is no signal in the wire, NIC will retransmit its original signals again.

If the NICs detect a collision for their transmitted signals, they take following actions: -

- Stop sending any further signals in wire

- Create a jam signal for random time period
- Once this time period is expired, sense wire again
- If collision still exists, create another jam signal and repeat the process
- If collision is removed, retransmit the original signals again.

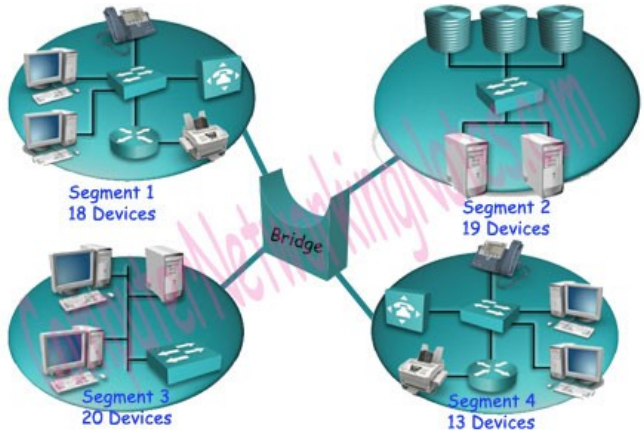
The more devices you place on a segment, the more likely you are going to experience the collisions. More devices means more random time interval, creating even more collisions, gradually slowing down a device's access when trying to transmit the data.

If a 100Mbps HUB has 10 ports, it means each port will effectively get only 10 Mbps (one tenth of total bandwidth.) Situation can be worse if we connect this HUB with another HUB on uplink port. Each new node will decrease the available bandwidth for other nodes in network.

Bridge solves these (bandwidth and collision) major issues of HUB.

Bridge

To improve the performance, usually networks are divided in smaller segments. Bridge is used to divide a large network in smaller segments. For example a network has 70 nodes. Without segmentation all these nodes will share same collision domain that will bring down overall network performance. To run a network smoothly we should not place more than 20 nodes in a collision domain. To deal with this situation we can use Bridge. Bridge has per port collision domain. It means if a port faces collision, other ports will not effect from this collision. If we use a four ports bridge in our example network, we will get four collision domains.



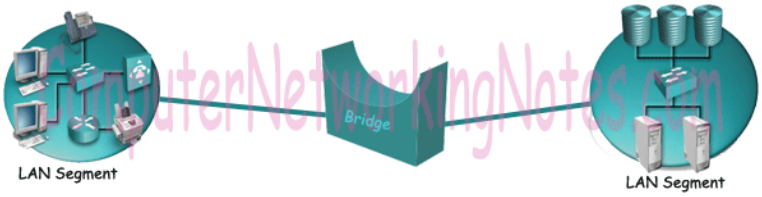
Basic functions of Bridge are following:-

- Break a large network in smaller segments.
- Join different media types such as UTP with fiber optic.
- Join different network architectures such as Ethernet with Token Ring.

A bridge can connect two different types of media or network architecture but it cannot connect two different types of network layer protocol such as TCP/IP or IPX. Bridge requires same network layer protocol in all segments.

There are three types of bridge:-

Local Bridge: - This Bridge connects two LAN segments directly. In Ethernet Implementation it is known as Transparent Bridge. In Token Ring network it is called Source-Routed Bridge.



Remote Bridge: - This Bridge connects with another bridge over the WAN link.

Wireless Bridge :- This bridge connects with another bridge without wiring between them.

In OSI Layer model Bridge works at physical layer and data link layer.

Bridges have following issues:-

- Bridges have limited ports.
- In bridge forward decision are made through the **software** which **slow** down overall performance of network.
- Bridges use age old technology which is not capable to fulfill the requirement of modern networks effectively.

Switch and Router solves these issues.

Switch

Just like Hub and Bridge, switch is also used to connect multiple computers together in a LAN segment. Switches available with 4,8,12,24,48,64 ports. Each switch port has a separate collision domain. Switch works at layer two in OSI Layer model. At layer two, data signals are formatted in frames.

When a switch receives frame, it checks FCS (Frame checksum sequence) field in it. Switch process the frame only if it is valid. All invalided frames are automatically dropped. All valid frames are processed and forwarded to their destination MAC address.

Switch makes their switching decisions in hardware by using application specific integrated circuits (ASICs). Unlike generic processor such as we have in our PC, ASICs are specialized processors built only to perform very few particular tasks. In cisco switch ASICs has single task, switch frames blazingly fast. For example an entry level catalyst 2960 switch has frame rate of 2.7 million frames per second. Higher end switches have higher FPS rate such as Catalyst 6500 has a rate of 400 million FPS rate.

Switches support three methods of switching.

- Store and Forward
- Cut and Through
- Fragment Free

Store and Forward

This is the basic mode of switching. In this mode Switch buffers entire frame into the memory and run FCS (Frame Check Sequence) to ensure that frame is valid and not corrupted. A frame less than 64bytes and higher than 1518bytes is invalid. Only valid frames are processed and all invalid frames are automatically dropped. Among these three methods, this method has highest latency. Latency is the time taken by device in passing frame from it.

Cut and Through

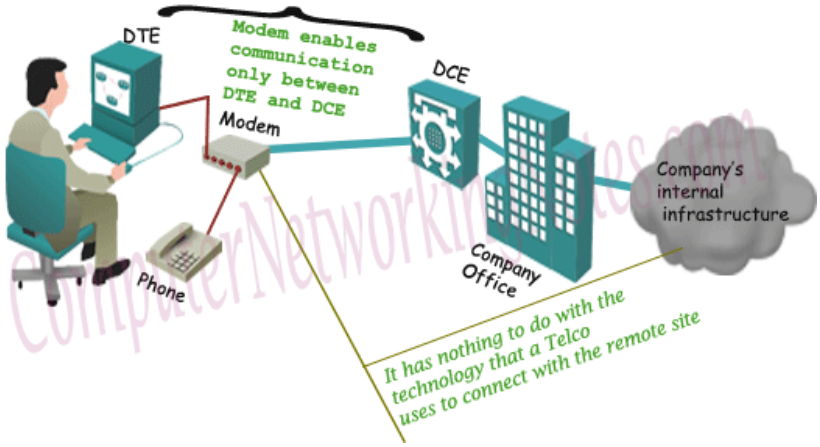
This has lowest latency. In this method, Switch only reads first six bytes from frame after the preamble. These six bytes are the destination address of frame. This is the fastest method of switching. This method also process invalid frames. Only advantage of this method is speed.

Fragment Free

This is a hybrid version of **Store and Forward** method and **Cut and Through** method. It takes goodies from both methods and makes a perfect method for switching. It checks first 64 bytes of frame for error. It processes only those frames that have first 64bytes valid. Any frame less than 64 bytes is known as runt. Runt is an invalid frame type. This method filters runt while maintaining the speed.

Modem

In simple language modem is a device that is used to connect with internet. Technically it is a device which enables digital data transmission to be transmitted over the telecommunication lines. A Telco company uses entirely different data transmission technology from the technology that a PC uses for data transmission. A modem understands both technologies. It converts the technology that a PC uses in the technology which a Telco company understand. It enables communication between PC (Known as DTE) and Telco company's office (Known as DCE).



There are two types of connection line between DCE and DTE

Analog connection line

An existing telephone or cable TV network line that uses analog signals (sound waves) for transportation. Instead of supporting Internet, these lines were primarily installed for their respective requirements.

Digital connection Line

A separate connection line between DTE and DCE. Since it is installed primarily for internet, it uses digital signals for data transportation.

For analog connection line we have to use analog modem and for digital line we need to use digital modem.

Analog Modem

Analog modem converts analog signal in digital signal and vice versa.

There are two types of analog modem; internal and external.

Internal Modem

Internal modem is available as interface card for desktop and as PCMCIA card for laptop . We need to install it on available slot of motherboard. In comparison with external modem these are inexpensive. As these modems usage computer's CPU for data encoding and decoding. We have to purchase these modem separately.

External Modem

External modem is a separate device that has its own CPU and memory. Usually Telco company provide this with connection. Depending on subscription it may be free or chargeable.

Digital Modem

Instead of signal conversion, digital modem performs modulation known as line coding. Line coding is used to modulate the digital signal in such a way that they can be transmitted over the digital line. DSL, ADSL and ISDN modem are the examples of digital modems.

DTE

DTE (Data Terminal Equipment) is a device (usually a router or PC) that converts data frame into signals and reconvert received signals in data frame. DTE device communicates with DCE device.

DCE

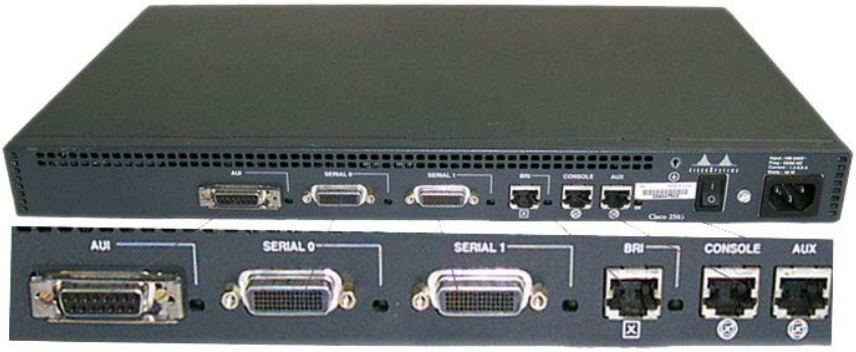
DCE (Data circuit terminating equipment) is a device (usually modem, CSU/DSU or Frame Relay switch) that provides clock rate and synchronization.

CSU/DSU

A CSU/DSU (Channel Service Unit/Data Service Unit) is a device that converts data signal between LAN network and WAN network. LAN network and WAN network uses separate communication technology. A CSU/DSU understands both technologies. DSL and cable modems are the example of CSU/DSU.

Router

Router is a layer three device which forwards data packet from one logical network segment to another. Router forwards packets on the bases of their destination address. For this, router keeps record of the path that packets can use as they move across the network. These records are maintained in a database table known as routing table. Routing table can be built statically or dynamically.



Basically routers are used :-

- To connect different network segments.
- To connect different network protocols such as IP and IPX.
- To connect several smaller networks into a large network (known as internetwork)
- To break a large network in smaller networks (Known as subnet usually created to improve the performance or manageability)
- To connect two different media types such as UTP and fiber optical.
- To connect two different network architectures such as token ring and Ethernet.
- To connect LAN network with Telco company's office (Known as DTE device).
- To access DSL services (known as DSL Router).

Brouters

Brouters are the combination of router and bridge. It can be used as a bridge or as a router. Brouters are the earlier implementation of the routers.

At layer two it's a fairly expensive device. which cost more than other high end switches that work much faster than it. At layer three it has a lot of complexity. Due to these drawbacks it is rarely used. Gradually it has been replaced by high end switch at layer 2 and by router at layer three.

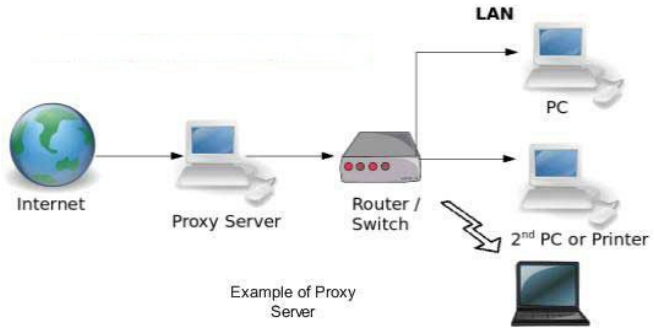
Gateway

Gateway is used to forward the packets which are intended for remote network from local network. Till host is configured with default gateway address, every packet should have default gateway address. A default gateway address is the address of gateway device. If packet does not find its destination address in local network then it would take the help of gateway device to find the

destination address in remote network. A gateway device knows the path of remote destination address. If require, it also change the encapsulation of packet so it can travel in other network to get its destination address.

Proxy

Proxy can be a dedicate device or it can be an application software. Proxy is used to hide the internal network from external world. If we use proxy then there would be no direct communication between internal network and external network. All communication will go through the proxy. External computer will be able to access only proxy. Thus Proxy makes tampering with an internal system from the external network more difficult.



Example of Proxy Server

Firewall

A firewall is a security layer which once configured keeps internal network safe from unauthorized external users. There are two types of firewall; software firewall and hardware firewall.

Software firewall

Software firewall runs as application software. It does not need any dedicate resources. It can be installed in any device which is already running other applications. It is less effective than hardware firewall but provides sufficient functionality for home and small office requirement. The biggest advantage of software firewall is that it is cost effective. Almost all modern platforms which can connect with Internet are equipped with basic firewall.

Following figure shows pre-installed software firewall in Windows 7

If require, user can install and use advance firewall. Advance firewalls are available at very nominal charges. Some open source firewalls are even available free of cost. You can download and installed them in PC, in router or in modem to protect your internal network from hackers.

Hardware firewall

Hardware firewall runs from a dedicate device. It is highly effective but costs a lot of money. Usually it is used in company environment where security is the top priority. Besides filtering data packet, a hardware firewall provides several other services such as spoofing, encryption and decryption, authentication and proxy services. Each additional service cost an additional amount of money. A lot of companies make hardware firewall nowadays. To select a hardware firewall which matches with your requirement, you can check the product manuals.

Following figure shows a Cisco Hardware Firewall

Chapter Four
TCP/IP Protocol Suite and IP Addressing:

This chapter describes the process the network layer uses to convert transport layer segments into packets and get them started on their journey down the right path across different networks to the destination network. You learn how the network layer divides networks into groups of hosts to manage the flow of data packets. You also consider how communication between networks is facilitated. This facilitation of communication between networks is called *routing*.

IPv4

The network layer, or Open Systems Interconnection (OSI) Layer 3, provides services to exchange the individual pieces of data over the network between identified end devices. To accomplish this end-to-end transport, Layer 3 uses the processes outlined in the following sections to address the packet to the proper destination, encapsulate the packet with necessary data for delivery, route the packet through the web of connected networks that will deliver the packet to the destination network for delivery, and finally, have the destination host decapsulate the data for processing. The details of these processes are explored further in the next sections.

Network Layer: Communication from Host to Host

The network layer, or OSI Layer 3, receives segments of data, or PDUs, from the transport layer. These bits of data have been processed into a transportable size and numbered for reliability. It is now up to the network layer to use protocols to add addressing and other information to the PDU and send it to the next router along the best path, or ***route***, to the destination network.

Network layer protocols, such as the widely used IP, are rules and instructions that devices use to enable sharing of upper-layer information between hosts. When the hosts are in different networks, additional routing protocols are used to choose routes between networks. Network layer protocols specify the addressing and packaging of a transport layer PDU and describe how the PDU is to be carried with minimum overhead.

The network layer describes four tasks to be performed:

1. Addressing packets with an IP address.
2. Encapsulation**.**
3. Routing.
4. Decapsulation.

The next sections describe each task in more detail and describe popular network layer protocols.

Addressing

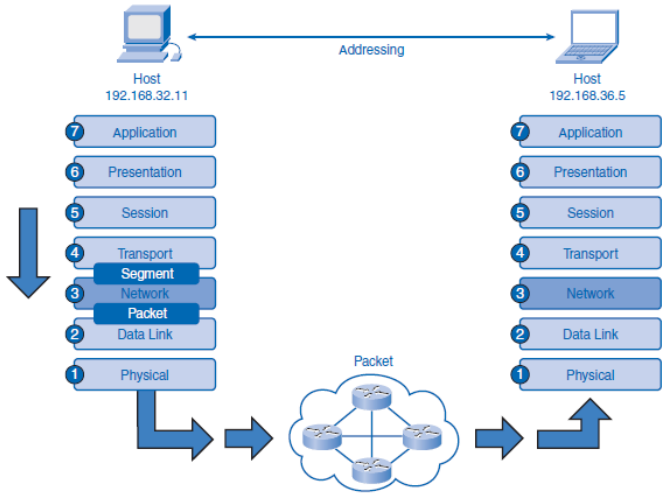
IP requires each sending and receiving device to have a unique IP address. Devices in IP networks that have IP addresses are called *hosts*. The IP address of the sending host is known as the ***source IP address***, and the IP address of the receiving host is referred to as the ***destination IP address***.

Encapsulation

Each PDU sent between networks needs to be identified with source and destination IP addresses in an ***IP header***. The IP header contains the address information and some other bits that identify the PDU as a network layer PDU. This process of adding information is called *encapsulation*. When an OSI Layer 4 PDU has been encapsulated at the network layer, it is referred to as a *packet*.

Figure 5-1 displays how segments are encapsulated at the network layer and become IP packets. The process is reversed at the destination.

Figure 5-1 Network Layer Encapsulation



Routing

When a packet is encapsulated at the network layer, it contains all the information necessary to travel to networks near and far. The journey between networks can be very short and relatively simple, or it can be complex and involve many steps between routers connected to different networks.

Routers are devices that connect networks. They specialize in understanding OSI Layer 3 packets and protocols as well as calculating the best path for the packets. ***Routing*** is the process routers perform when receiving packets, analyzing the destination address information, using the address information to select a path for the packet, and then forwarding the packet on to the next router on the selected network. Each route that a packet takes to reach the next device is called a ***hop***. A packet can hop between several different routers en route to the destination. Each router examines the address information in the packet, but neither the IP address information nor the encapsulated transport layer data in the packet is changed or removed until the packet reaches the destination network.

Figure 5-2 shows how there can be several different paths in the internetwork cloud between a source host and a destination host.

Figure 5-2 Multiple Network Paths Between Hosts

At the network layer, the router opens the packet and looks in the packet header for IP address information. The router, depending on how it is configured and what it knows about the destination network, will choose the best network to deliver the packet. The router then forwards the packet out of the interface connected to the chosen network. The last router along the path will realize that the packet belongs to a ***directly connected network*** and will forward it out the correct network interface for final delivery on the local network. For a network layer packet to travel between hosts, it must be handed down to the data link layer (OSI Layer 2) for another layer of encapsulation called *framing*, and then encoded and put onto the physical layer (OSI layer 1) to be sent to the next router.

Decapsulation

An IP packet arrives at a router's network interface encapsulated in a Layer 2 frame on the physical OSI layer. The router's network interface card (NIC) accepts the packet, removes the Layer 2 encapsulation data, and sends the packet up to the network layer. The process of removing encapsulation data at different layers is referred to as *decapsulation*. Encapsulation and decapsulation occur at all layers of the OSI model. As a packet travels from network to network to its destination, there can be several instances in which Layers 1 and 2 are encapsulated and decapsulated by routers. The network layer only decapsulates the IP packet at the final destination after examining the destination addresses and determining that the journey is over. The IP packet is no longer useful, so it is

discarded by the destination host. When the IP packet is decapsulated, the information in the packet is handed up to the upper layers for delivery and processing.

IPv4: Example Network Layer Protocol

Version 4 of IP (IPv4) is currently the most widely used version of IP. It is the only Layer 3 protocol that is used to carry user data over the Internet and is the focus of the CCNA. Therefore, it will be the example you use for network layer protocols in this course. IP version 6 (IPv6) is developed and being implemented in some areas. IPv6 will operate alongside IPv4 and might replace it in the future. The services provided by IP, as well as the packet header structure and contents, are specified by either IPv4 or IPv6.

The characteristics of IPv4 and IPv6 are different. Understanding these characteristics will allow you to understand the operation of the services described by this protocol. IP was designed as a protocol with low overhead. It provides only the functions that are necessary to deliver a packet from a source to a destination over an interconnected system of networks. The protocol was not designed to track and manage the flow of packets. These functions are performed by other protocols in other layers.

IPv4 basic characteristics include the following:

- **Connectionless:** IPv4 does not establish a connection before sending data packets.
- **Best effort (unreliable):** IPv4 does not use processes that guarantee packet delivery, which reduces processing time on routers and saves the bandwidth that acknowledgment messages would otherwise require.
- **Media independent:** IPv4 operates independently of the medium carrying the data.

The next sections describe these three traits in greater detail.

Connectionless

As you learned in Chapter 4, "OSI Transport Layer," TCP's reliability comes from being ***connection oriented***. TCP uses a connection between the sender and the receiver to exchange control data and ensure reliability of packet delivery. IP is ***connectionless***, meaning that there is no established connection between the sender and the receiver. IP simply sends packets without informing the receiver. Lacking a connection is not a problem for IP and is part of the "best effort" design. This is why IP and TCP work together so well in a TCP/IP stack: If a packet is lost or late, TCP will correct the problem at Layer 4, and IP can work more efficiently at Layer 3.

Because IP does not have to be accountable for reliability or keep a connection, it does not need as much information in the header as a TCP segment does. Because IP requires less data to perform the required tasks, it uses much less processing power and bandwidth, called ***overhead***, than TCP.

Best Effort

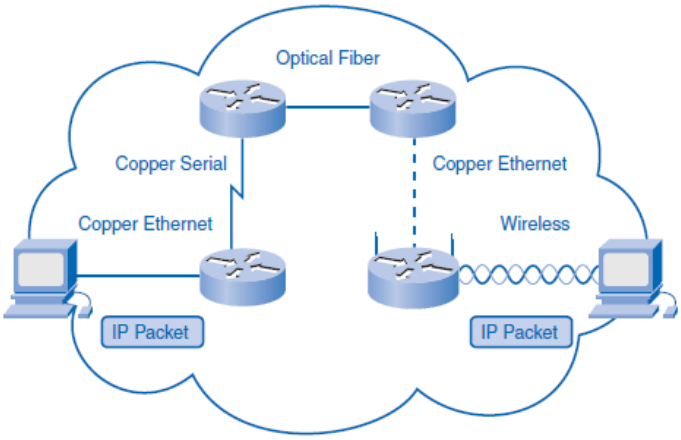
In Chapter 4, you also learned that TCP is reliable. It is reliable because communication is established with the receiver and receipt of the data is confirmed by the receiver. If packets are lost, the receiver communicates with the sender to request a retransmission. The TCP segment contains information that allows reliability to be ensured. IP is an unreliable, ***best-effort*** protocol in that it is unaware of the quality of job it is performing. IP packets are sent without certainty that they will be received. The IP protocol makes a "best effort" to deliver packets, but it has no way of determining whether the packets are delivered successfully or whether they are lost en route. IP has no way to inform the sender of reliability problems. TCP can be relied on to inform the sender of delivery problems.

Media Independent

IP is ***media independent***, which means it is not concerned with the physical medium that carries the packet. Internetwork communication is likely to be a multimedia journey using a combination of wireless, Ethernet cable, fiber-optic cable, and other OSI Layer 1 media. The arrangement of bits in the IP packet and header will not be changed as the packet transfers from wireless to fiber or any other media.

Figure 5-3 shows how there can be several different physical layer media between the source host and destination host.

Figure 5-3 IP Packets Are Media Independent



One important consideration, however, is the size of the PDU. Some networks have media restrictions and must enforce a ***maximum transmission unit (MTU)***. The MTU is determined by the OSI data link layer, and that requirement is passed to the network layer. The network layer then builds the packets according to specification. Should the packet come across a network that requires smaller packets, the router connected to the network will fragment the packets before forwarding them on the network's medium. This process is called ***fragmentation***.

IPv4 Packet: Packaging the Transport Layer PDU

IPv4 encapsulates, or packages, the transport layer segment or datagram so that the network can deliver it to the destination host. The IPv4 encapsulation remains in place from the time the packet leaves the network layer of the originating host until it arrives at the network layer of the destination host. The process of encapsulating data by layer enables the services at the different layers to develop and scale without affecting other layers. This means that transport layer segments can be readily packaged by existing network layer protocols, such as IPv4 and IPv6, or by any new protocol that might be developed in the future. Routers can implement these different network layer protocols to operate concurrently over a network to and from the same or different hosts. The routing performed by these intermediary devices only considers the contents of the packet header that encapsulates the segment.

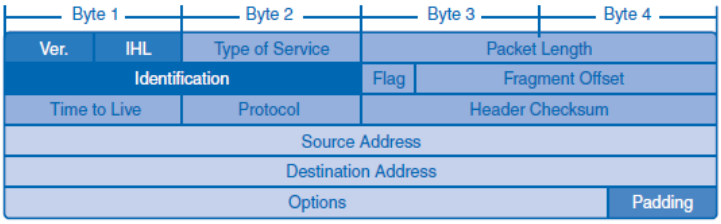
In all cases, the data portion of the packet—that is, the encapsulated transport layer PDU—remains unchanged during the network layer processes.

IPv4 Packet Header

The IP header holds the delivery and handling instructions for an IP packet. For example, when a packet arrives on a router's interface, the router needs to know whether the packet is IPv4 or IPv6. The router looks to a specific field in the header to see which type is arriving. The header also contains addressing information and other data about how to handle the packet along the way.

Figure 5-4 shows an outline of an IP packet header. There are several fields in the packet, and not every network uses every field. There are highlighted fields that are important to understanding how the IP header helps routers route IP packets successfully.

Figure 5-4 Components of an IP Header



The key fields are as follows:

- **IP Source Address:** Contains a 32-bit binary value that represents the host that will receive the packet. Routers will use this data to forward the packet to the correct network.

- **IP Destination Address:** Contains a 32-bit binary value that represents the host that will receive the packet. Routers will use this data to forward the packet to the correct network.
- ***Time to Live (TTL)*:** The 8-bit TTL field describes the maximum hops the packet can take before it is considered "lost" or undeliverable. Each router that handles the packet decrements the TTL field by at least 1. The packet will be dropped if the TTL value reaches 0. This keeps the Internet from being cluttered with lost packets.
- **Type of Service (ToS):** Each of the 8 bits in this field describes a level of throughput priority a router should use in processing the packet. For example, a packet containing IP voice data gets precedence over a packet containing streaming music. The way a router handles a packet from this data is known as *QoS*, or *quality of service*.
- **Protocol:** This 8-bit field indicates the upper-layer protocol—for example, TCP, UDP, or ICMP—that will receive the packet when it is decapsulated and given to the transport layer.
- **Flag and Fragment Offset:** A router might have to fragment a packet when forwarding it from one medium to another medium that has a smaller MTU. When fragmentation occurs, the IPv4 packet uses the Fragment Offset field and the MF flag in the IP header to reconstruct the packet when it arrives at the destination host. The Fragment Offset field identifies the order in which to place the packet fragment in the reconstruction.

Other fields are as follows:

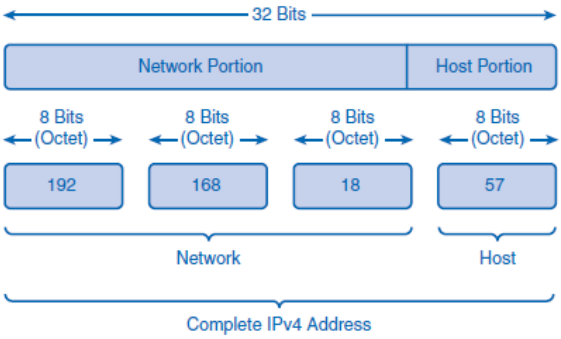
- **Version:** Indicates IP version 4 or 6.
- **Internet Header Length (IHL):** Tells the router how long the header is. The length is not always the same because of variable data in the Options field.
- **Packet Length:** This is the total length of the datagram, including the header. The minimum length of a packet is 20 bytes (header with no data), and the maximum length with data is 65,535 bytes.
- **Identification:** Sent by the source to help reassemble any fragments.
- **Header Checksum:** This data is used to indicate the length of the header and is checked by each router along the way. An algorithm is run by each router, and if the checksum is invalid, the packet is assumed to be corrupted and is dropped. Because the TTL value is changed by each router that handles the packet, the header checksum is recalculated at each hop.
- **Options:** A rarely used field that can provide special routing services.
- **Padding:** Padding is used to fill in bits when header data does not end on a 32-bit boundary.

Dividing Networks from Networks

The IPv4 address is composed of 32 bits divided into two parts: the network address and the host address. The network portion of the address acts like a postal code and tells routers where to find the general neighborhood of a network. Routers forward packets between networks by referring only to the network portion. When the packet arrives at the last router, like a letter arriving at the last postal station, the local portion of the address identifies the destination host. The IPv4 addressing system is flexible. If a large network needs to be divided into smaller subnets, additional network codes can be created using some of the bits designated for the host in a process called *subnetting*. Network managers use this flexibility to customize their private networks. IPv4's ability to scale to the ever-growing demands of the Internet has contributed to its wide use.

Figure 5-11 shows the basic structure of an IPv4 address. In this address, the three ***octets*** to the left are the general network address, and the last octet is used by the destination router to identify the local host.

Figure 5-11 Hierarchical IPv4 Address



The portion of the address that is network and the portion that is host can vary.

IPv4 Addresses

For communication to take place between hosts, the appropriate addresses must be applied to these devices. Managing the addressing of the devices and understanding the IPv4 address structure and its representation are essential.

Anatomy of an IPv4 Address

Each device on a network must be uniquely defined by a network layer address. At this layer, the packets of the communication are also identified with the source and destination addresses of the two end systems. With IPv4, each packet uses a 32-bit source address and a 32-bit destination address in the Layer 3 header.

These addresses are represented in the data network as binary patterns. Inside the devices, ***digital logic*** is applied for the interpretation of these addresses. For the human network, a string of 32 bits is difficult to interpret and even more difficult to remember. For this reason, IPv4 addresses are represented using ***dotted decimal*** format.

Dotted Decimal

IPv4 addresses are easier to remember, write, and verbally communicate than strings of 32 bits. Representing IPv4 addresses as dotted decimals begins by separating the 32 bits of the address into bytes. Each byte of the binary pattern, called an *octet*, is separated with a dot. The bytes are called an octet, because each of the decimal numbers represents 1 byte, or 8 bits.

For example, the following address:

10101100000100000000010000010100
is expressed in dotted decimal as
172.16.4.20

Keep in mind that devices use binary logic. The dotted decimal format makes it easier for people to use and remember addresses.

Network and Host Portions

IPv4 addresses have two parts: the network portion and the host portion. For each IPv4 address, some portion of the most significant bits, or ***high-order bits***, represents the network address. At Layer 3, a *network* is defined as a group of hosts that have identical bit patterns in the network address portion of their addresses. That is, all the bits in the network portion of their addresses are identical.

In the following example, the two addresses have identical network portions. Therefore,
hosts assigned these two addresses would be on the same logical network:

172.16.4.20 **172.16.4**.32
network host **network** host
portion portion **portion** portion

Although all 32 bits define the IPv4 host address, a variable number of bits represent the host portion of the address. The number of bits used in this host portion determines the number of hosts within the network. In the previous example, the last octet, the lowest 8 bits, are the host portion. This means that the bits for the upper three octets represent the network portion.

You determine how many bits are required for the host portion based on the number of hosts that a network requires. If a particular network requires at least 200 hosts, you would need to use enough bits in the host portion to be able to represent at least 200 different bit patterns. To assign a unique address to 200 hosts, you would use the entire last octet. With 8 bits, a total of 256 different bit patterns can be achieved. As with the previous example, this means that the bits for the upper three octets represent the network portion. Calculating the number of hosts and determining which portion of the 32 bits of an IPv4 address refers to the network portion will be covered in the section "Calculating Network, Hosts, and Broadcast Addresses," later in this chapter.

Addressing Types of Communication:

Unicast, Broadcast, Multicast

In an IPv4 network, the hosts can communicate in one of three different ways:

- **Unicast:** The process of sending a packet from one host to an individual host.

- **Broadcast:** The process of sending a packet from one host to all hosts in the network.
- **Multicast:** The process of sending a packet from one host to a selected group of hosts.

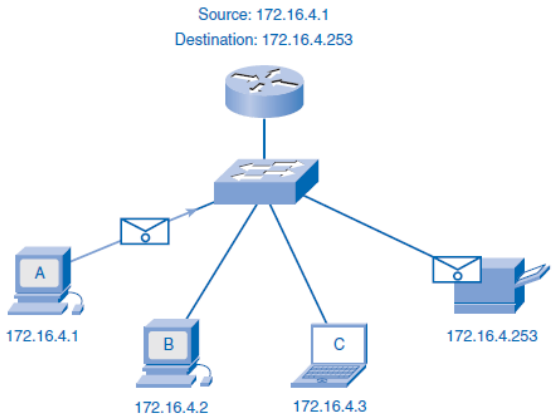
Each of these three types supports different types of communication in the data networks and uses different IPv4 destination addresses. In all three cases, the IPv4 address of the originating host is placed in the packet header as the source address.

Unicast Communication and Addresses

The most common type of communication is unicast. This is the normal host-to-host communication in both a client/server and a peer-to-peer network. For unicast communication, the host addresses assigned to the two end devices are used as the source and destination IPv4 addresses. During the encapsulation process, the source host places its IPv4 address in the unicast packet header as the source host address and the IPv4 address of the destination host in the packet header as the destination address. The communication using a unicast packet can be forwarded through an internetwork using the same addresses.

Figure 6-6 shows an example of IPv4 unicast communication from computer A with the address 172.16.4.1 to the printer with the address 172.16.4.253. In the communication represented, computer A creates a single packet addressed to the Layer 3 address of the printer. This packet is then forwarded by the services at the lower layers to the printer. If a copy of this packet should arrive at an end device whose address does not match this address, that host will discard the packet.

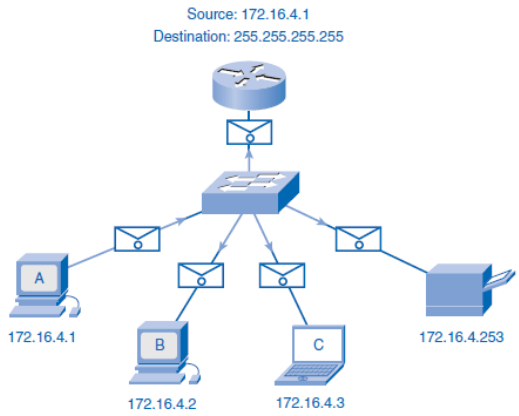
Figure 6-6 Unicast Communication



Broadcast Communication and Addresses

Layer 4 broadcast communication is the process of sending a packet from one host to all hosts in the network. Unlike unicast communication, which uses the destination host address, broadcast and multicast communication use special addresses as the destination address. This special address, called the ***broadcast address***, allows all the receiving hosts to accept the packet. When a host receives a packet with the broadcast address as the destination, it processes the packet as it would a packet to its unicast address. Using these special addresses, broadcasts are generally restricted to the local network.

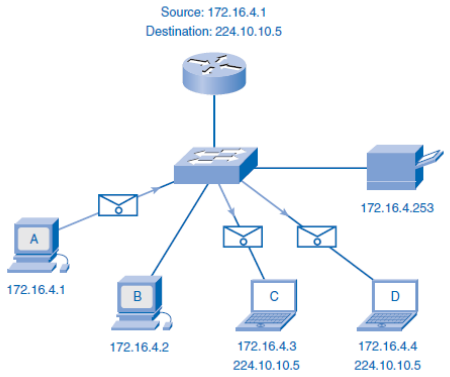
Figure 6-7 Broadcast Communication



Multicast Communication and Addresses

Multicast transmission is designed to conserve the bandwidth of the IPv4 network. It reduces traffic by allowing a host to send a single packet to a selected set of hosts. To reach multiple destination hosts using unicast communication, a source host would need to send an individual packet addressed to each host. With multicast, the source host can send a single packet that can reach thousands of destination hosts.

Figure 6-8 Multicast Communication



IPv4 Addresses for Different Purposes

Besides the addresses in the IPv4 address range that are reserved for multicast, many of the IPv4 unicast addresses have been reserved for special purposes. Some of these addresses limit the scope or functionality of the hosts to which they are assigned. Other reserved addresses cannot be assigned to hosts. In the next sections, some of these reserved addresses will be presented.

Types of Addresses in an IPv4 Network Range

Within each IPv4 network, there are three types of addresses:

- **Network address:** A special address that refers to the network
- **Broadcast address:** A special address used to send data to all hosts in the network
- **Host addresses:** The unicast addresses assigned to the end devices in the network

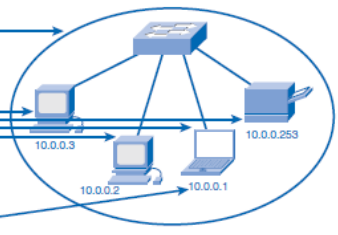
Within each network, there are two addresses that cannot be assigned to devices: network address and broadcast address. The other addresses allocated to a network are the host addresses to be assigned to the individual devices.

Network Address

The network address is a standard way to refer to a network. For example, you could refer to the network inside the circle in Figure 6-9 as "the 10.0.0.0 network." This is a much more convenient and descriptive way to refer to the network than using a term like "the first network." All hosts in the 10.0.0.0 network will have the same network bits. This address cannot be assigned to a device and is, therefore, not used as an address for communication in the network. It is only used as a reference to the network. Within the IPv4 address range of a network, the lowest address is reserved for the network address. This address has a 0 for each host bit in the host portion of the address.

Figure 6-9 Network, Broadcast, and Host Addresses

Network			Host
			Network Address
10	**0**	**0**	**0**
00001010	00000000	00000000	00000000
			Broadcast Address
10	**0**	**0**	**255**
00001010	00000000	00000000	11111111
			Host Address
10	**0**	**0**	**1**
00001010	00000000	00000000	00000001



.

Broadcast Address

The IPv4 broadcast address within a network is the directed broadcast address. Unlike the network address, this address is used in communication to all the hosts in a network. This special address for each network allows a single packet to communicate to all the hosts in that network. To send data to all hosts in a network, a host can send a single packet that is addressed to the broadcast address of the network. For example in Figure 6-9 in the preceding section, to communicate with all the hosts in this network, use a destination address 10.0.0.255, which is the broadcast address for the network. The broadcast address uses the highest address in the network range. This is the address in which the bits in the host portion are all 1s. For the network 10.0.0.0 with 24 network bits in Figure 6-9, the broadcast address would be 10.0.0.255.

Host Addresses

As described previously, every end device requires a unique unicast address to deliver a packet to that host. In IPv4 addresses, you can assign the values between the network address and the broadcast address to the devices in that network. These are called the host addresses. In Figure 6-9, the addresses between the network address of 10.0.0.0 and the broadcast address of 10.0.0.255 are the host addresses. This means that the addresses 10.0.0.1 to 10.0.0.254 can be assigned to the hosts in this logical network.

Network Prefixes

When you examine a network address, you might ask, "How do you know how many bits of this address represent the network portion and how many bits represent the host portion?" The answer is the prefix mask. When an IPv4 network address is expressed, you add a ***prefix length*** to the network address. This prefix length is the number of bits in the address that gives the network portion. This prefix length is written in ***slash format***. That is a forward slash (/) followed by the number of network bits. For example, in 172.16.4.0 /24, the /24 is the prefix length. This tells you that the first 24 bits are the network address. The remaining 8 bits, the last octet, are the host portion.

Networks are not always assigned a /24 prefix. Depending on the number of hosts on the network, the prefix assigned can be different. Having a different prefix number changes the host range and broadcast address for each network. Notice that the network addresses in Table 6-9 remain the same, but the host range and the broadcast address are different for the different prefix lengths. You can also see that the number of hosts that can be addressed on the network changes as well.

Table 6-9 Using Different Prefixes for the 172.16.4.0 Network

Network	Network Address	Host Range	Broadcast Address
172.16.4.0 /24	172.16.4.0	172.16.4.1–172.16.4.254	172.16.4.255
172.16.4.0 /25	172.16.4.0	172.16.4.1–172.16.4.126	172.16.4.127
172.16.4.0 /26	172.16.4.0	172.16.4.1–172.16.4.62	172.16.4.63
172.16.4.0 /27	172.16.4.0	172.16.4.1–172.16.4.30	172.16.4.31

Subnet Mask: Defining the Network and Host Portions of the Address

Another question you might ask is, "How do the network devices know how many bits are the network portion and how many bits are the host portion?" The answer to this question is the subnet mask.

The prefix and the subnet mask are different ways of representing the same information: the network portion of an address. The prefix length tells you the number of bits in the address that are the network portion in a way that is easier to communicate to humans. The subnet mask is used in data networks to define this network portion for the devices.

The subnet mask is a 32-bit value used with the IPv4 address that specifies the network portion of the address to the network devices. The subnet mask uses 1s and 0s to indicate which bits of the IPv4 address are network bits and which bits are hosts bits. The subnet mask is expressed in the same dotted decimal format as the IPv4 address.

There is a one-to-one correlation between the bits in the IPv4 address and the subnet mask. The subnet mask is created by placing a binary 1 in each appropriate bit position that represents a network bit of the address and placing a binary 0 in the remaining bit positions that represent the host portion of the address.

A /24 prefix represents a subnet mask of 255.255.255.0

(11111111.11111111.11111111.00000000). The first three octets, the higher-order 24 bits, are all 1s. The remaining low-order bits of the subnet mask are 0s, indicating the host address within the network.
For example, examine the host 172.16.4.35/27 shown in Table 6-10.

Table 6-10 Determining the Network Address for the Host 172.16.4.35 /27

	Dotted Decimal				Binary Octets			
Host	172	16	4	35	10101100	00010000	00000100	00100011
Mask	255	255	255	224	11111111	11111111	11111111	11100000
Network	172	16	4	32	10101100	00010000	00000100	00100000

Because the high-order bits of the subnet masks are contiguous 1s, there are only a limited number of subnet values within an octet. You only need to expand an octet if the network and host division falls within that octet. Therefore, there is a limited number of 8-bit patterns used in address masks. These bit patterns for the subnet masks, the number of network bits and the number of data bits within the octet, are shown in Table 6-11.
If the subnet mask for an octet is represented by 255, all the equivalent bits in that octet of the address are network bits. Similarly, if the subnet mask for an octet is represented by 0, all the equivalent bits in that octet of the address are host bits. In each of these cases, it is not necessary to expand this octet to binary to determine the network and host portions.

Table 6-11 Subnet Mask Values Within an Octet

Mask (Decimal)	Mask (Binary)	Network Bits	Host Bits
0	00000000	0	8
128	10000000	1	7
192	11000000	2	6
224	11100000	3	5
240	11110000	4	4
248	11111000	5	3
252	11111100	6	2
254	11111110	7	1
255	11111111	8	0

Special Unicast IPv4 Addresses

In addition to the addresses that cannot be assigned to hosts, special addresses can also be assigned to hosts but with restrictions on how those hosts can interact within the network.
These special addresses include the following:

- Default route
- Loopback address
- Link-local address
- Test-net addresses

Table 6-12 Major Reserved and Special-Purpose IPv4 Addresses

Type	Block	Range	Reference
Multicast	224.0.0.0 /4	224.0.0.0–239.255.255.255	RFC 1700
Network address	—	—	One per network
Broadcast address	—	—	One per network plus 255.255.255.255
Experimental addresses	240.0.0.0 /4	240.0.0.0–255.255.255.254	RFC 3330
Private space addresses	10.0.0.0 /8 172.16.0.0 /12 192.168.0.0 /16	10.0.0.0–10.255.255.255 172.16.0.0–172.31.255.255 192.168.0.0–192.168.255.255	RFC 1918
Default route	0.0.0.0 /8	0.0.0.0–0.255.255.255	RFC 1700
Loopback	127.0.0.0 /8	127.0.0.0–127.255.255.255	RFC 1700
Link-local addresses	169.254.0.0 /16	169.254.0.0–169.254.255.255	RFC 3927
Test-net addresses	192.0.2.0 /24	192.0.2.0–192.0.2.255	—

Legacy IPv4 Addressing

In the early 1980s, the IPv4 addressing range was divided into three different classes: class A, class B, and class C. Each class of addresses represented networks of a specific fixed size. At that time in the development of IP, there were no subnet masks to specify the network and host portion of the addresses. To distinguish between the network sizes, each of these classes of addresses was assigned address ranges. Devices could examine the high order address to determine how many network bits were used to define the network. For example, for the address 192.168.2.2, because this address is in the class C addressing range, a network device recognized this as a class C network and identified the standard class C prefix of /24.

In the late 1980s and early 1990s, the subnet mask was added to the IPv4 addressing scheme to allow these fixed-size networks to be subdivided or subnetted. However, many of the restrictions of these classes remained.

By the mid-1990s, most of the restrictions of this class-based addressing system had been removed from the standards and the equipment operation. However, the associated practices developed over the decade perpetuated this classful system. Even today, some remnants of this addressing system still affect network practices and operation. For this reason, you should be familiar with these network classes. Table 6-13 summarizes the address classes.

Table 6-13 IPv4 Network Classes

Address Class	First Octet Range	Prefix and Mask	Number of Possible Networks	Number of Hosts per Network
A	1 to 127	/8 255.0.0.0	126 (2^7)	16,777,214 (2^{24}–2)
B	128 to 191	/16 255.255.0.0	16,384 (2^{14})	65,534 (2^{16}–2)
C	192 to 223	/24 255.255.255.0	2,097,159 (2^{21})	254 (2^8–2)

Limits to the Classful Addressing System

Not all organizations' addressing requirements fit well into one of these three classes. Classful allocation of address space often wasted many addresses, which exhausted the availability of IPv4 addresses. For example, a company that had a network with 260 hosts would need to be given a class B address with more than 65,000 addresses.

Even though this classful system was all but abandoned in the late 1990s, the remnants of it are in effect in networks today. For example, when you assign an IPv4 address to a computer, the operating system examines the address being assigned to determine whether it is a class A, class B, or class C address. The operating system then assumes the prefix used by that class and makes the appropriate subnet mask assignment.
Another example is the assumption of the mask by some routing protocols. When some routing protocols receive an advertised route, it can assume the prefix length based on the class of the address.

Classless Addressing

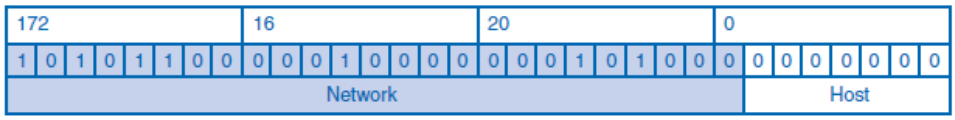
The system that is currently in use is referred to as ***classless addressing***. With the classless system, address blocks appropriate to the number of hosts are assigned to companies or organizations without regard to the unicast class. Associated with this classless addressing system are other practices, such as using networks of fixed sized, that have made IPv4 addressing more viable.

Calculating Network, Hosts, and Broadcast Addresses

Calculating the Network Address

First, determine the network address. The network address is the lowest address in the address block. To represent a network address, all the host bits are 0. With a 25-bit prefix, the last 7 bits are host bits and are 0s. Figure 6-14 shows the network address for the 172.16.20.0 /25 network.

Figure 6-14 Network Address for the 172.16.20.0 /25 Network

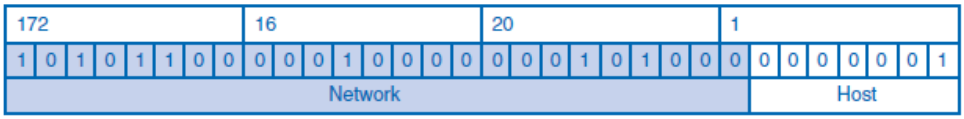


Host bits are all 0s: 0+0+0+0+0+0+0+0=0. So, the network address is 172.16.20.0. This makes the last octet of the address 0. Therefore, the network address is 172.16.20.0.

Calculating the Lowest Host Address

Next, you should calculate the lowest host address. This is always 1 greater than the network address. Therefore, using binary counting, you increment the 1s bit, making the last host bit a 1. Figure 6-15 shows the lowest host address for the network 172.16.20.0 /25.

Figure 6-15 Lowest Host Address for the 172.16.20.0 /25 Network

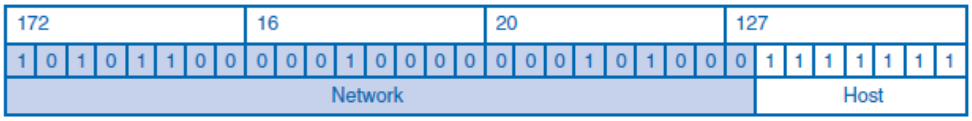


All host bits except the least significant address are all 0s: 0+0+0+0+0+0+0+1=1. With the lowest bit of the host address set to a 1, the address is 172.16.20.1. So, the lowest host address is 172.16.20.1.

Calculating the Broadcast Address

Although it can seem a little out of sequence, it is often easier to calculate the broadcast address before calculating the highest host address. The broadcast address of a network is the highest address in the address block. It requires all the host bits to be set. Therefore, all seven host bits used in this example network are 1s, as shown in Figure 6-16.

Figure 6-16 Broadcast Address for the 172.16.20.0 /25 Network

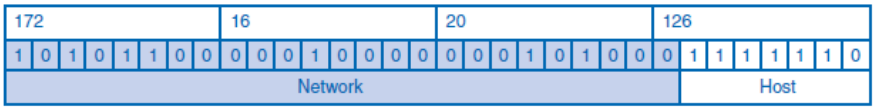


All host bits are 1s: 64+32+16+8+4+2+1=127. From the calculation, the value of the last octet is 127. This gives you a broadcast address of 172.16.20.127 for the network 172.16.20.0 /25.

Calculating the Highest Host Address

After determining the broadcast address, you can easily determine the highest host address. It is 1 less than the broadcast address. In Figure 6-17, with a broadcast address of 172.16.20.127, the highest host address would be 172.16.20.126. To determine the highest host address, make the lowest host bit a 0 and all other host bits a 1.

Figure 6-17 Highest Host Address for the 172.16.20.0 /25 Network



All host bits, except the lowest address, are all 1s: 64+32+16+8+4+2+0=126. This makes the highest host address in this example 172.16.20.126.

Determining the Host Address Range

Finally, you need to determine the host range for the network. The host range of the network includes all the addresses from the lowest host address to the highest host address inclusive. Therefore in this network, the address range is 172.16.20.1 to 172.16.20.126

These IPv4 unicast addresses can be assigned to the hosts in the logical network 172.16.20.0 /25. A host that is assigned any other address will be in a different logical network.

Basic Subnetting

Another IPv4 addressing skill helpful for a network associate is the ability to plan the subnetting of a network. The address range used in an internetwork needs to be divided into networks. Each of these networks must be assigned a portion of these addresses called a subnet. Many factors and techniques are used to create a subnetting plan. These sections will present some of these factors and techniques.

Subnetting allows creating multiple logical networks from a single address block. Because a router connects these networks, each interface on a router must have a unique network ID. Every node on that link is on the same network. You create the subnets by reassigning one or more of the host bits as network bits. This is done by extending the prefix to "borrow" some of the bits from the host portion of the address to create additional network bits. The more host bits borrowed, the more subnets that can be defined. For each bit borrowed, you double the number of subnetworks available.

For example, if you borrow 1 bit, you can define two subnets. If you borrow 2 bits, you can have four subnets. However, with each bit you borrow, you have fewer host bits to define the host addresses in each subnet. Therefore, there are fewer host addresses available per subnet. Additionally, because you have two addresses for each network—network address and broadcast address—that cannot be assigned to hosts, the total number of hosts in the entire network decreases.

Creating Two Subnets

Router A in Figure 6-18 has two interfaces to interconnect two networks. Given an address block of 192.168.1.0 /24, you will create two subnets. You borrow 1 bit from the host portion by using a subnet mask of 255.255.255.128, instead of the original 255.255.255.0 mask. This makes the most significant bit in the last octet a network bit instead of a host bit. This bit is used to distinguish between the two subnets. For one of the subnets, this bit is a 0, and for the other subnet, this bit is a 1. The information for these two subnets is shown in Table 6-15.

Figure 6-18 Borrowing a Bit to Create Two Subnets

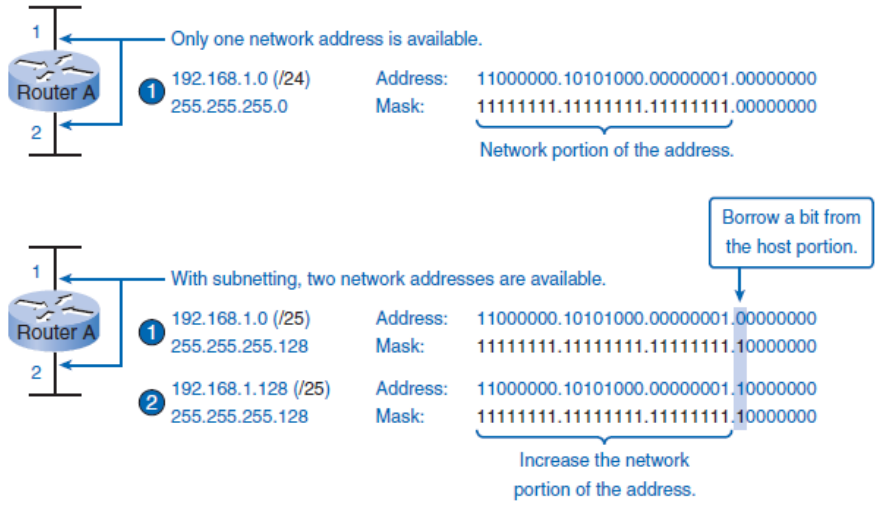


Table 6-15 Subnets for the 192.168.1.0 /24 Network with 1 Borrowed Bit

Subnet	Network Address	Host Range	Broadcast Address
1	192.168.1.0/25	192.168.1.1–192.168.1.126	192.168.1.127
2	192.168.1.128/25	192.168.1.129–192.168.1.254	192.168.1.255

Use this formula to calculate the number of subnets:
2^n, where n = the number of bits borrowed
In the example of Figure 6-18 and Table 6-15, the calculation is $2^1 = 2$ subnets
For each subnet, examine the last octet of the subnet address in binary. The values in these octets for the two networks are
Subnet 1: **0**0000000 = 0
Subnet 2: **1**0000000 = 128
To calculate the number of hosts per network, you use the formula of $2_n - 2$, where n = the number of bits left for hosts.
Applying this formula to the two-subnet example in Figure 6-18 and Table 6-15,
$2^7 - 2 = 126$ shows that each of these subnets can have 126 hosts.

Creating Three Subnets

Beginning with the previous example, consider an internetwork that requires three subnets. The network in Figure 6-19 starts with the same 192.168.1.0 /24 address block. Borrowing a single bit would only provide two subnets. To provide more networks, you change the subnet mask to 255.255.255.192 and borrow 2 bits. These 2 bits will provide four subnets. These networks are shown in Table 6-16. The calculations follow.

Figure 6-19 Borrowing 2 Bits to Create Subnets

Table 6-16 Subnets for the 192.168.1.0 /24 Network with 2 Borrowed Bits

Subnet	Network Address	Host Range	Broadcast Address
0	192.168.1.0/26	192.168.1.1–192.168.1.62	192.168.1.63
1	192.168.1.64/26	192.168.1.65–192.168.1.126	192.168.1.127
2	192.168.1.128/26	192.168.1.129–192.168.1.190	192.168.1.191
3	192.168.1.192/26	192.168.1.193–192.168.1.254	192.168.1.255

Calculate the number of subnets with this formula:
$2^2 = 4$ subnets
To calculate the number of hosts, begin by examining the last octet. Notice these subnets:
Subnet 1: 0 = **00**000000
Subnet 2: 64 = **01**000000
Subnet 3: 128 = **10**000000
Subnet 4: 192 = **11**000000
Apply the host calculation formula:
$2^6 – 2 = 62$ hosts per subnet

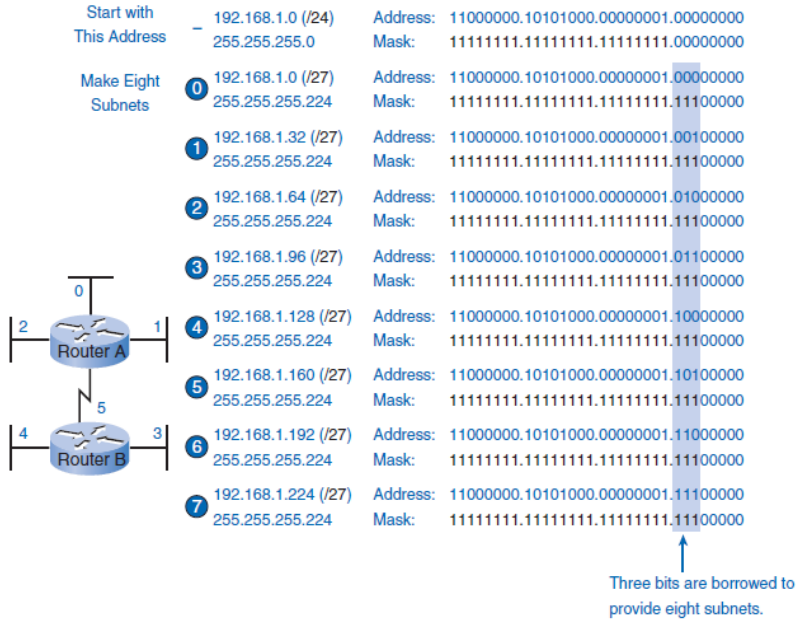
Creating Six Subnets

Consider the example in Figure 6-20 with five LANs and a WAN, for a total of six networks. The network information for this example is shown in Table 6-17, and the calculations follow.

Table 6-17 Subnets for the 192.168.1.0 /24 Network with 3 Borrowed Bits

Subnet	Network Address	Host Range	Broadcast Address
0	192.168.1.0/27	192.168.1.1–192.168.1.30	192.168.1.31
1	192.168.1.32/27	192.168.1.33–192.168.1.62	192.168.1.63
2	192.168.1.64/27	192.168.1.65–192.168.1.94	192.168.1.95
3	192.168.1.96/27	192.168.1.97–192.168.1.126	192.168.1.127
4	192.168.1.128/27	192.168.1.129–192.168.1.158	192.168.1.159
5	192.168.1.160/27	192.168.1.161–192.168.1.190	192.168.1.191
6	192.168.1.192/27	192.168.1.193–192.168.1.222	192.168.1.223
7	192.168.1.224/27	192.168.1.225–192.168.1.254	192.168.1.255

Figure 6-20 Borrowing 3 Bits to Create Subnets



To accommodate six networks, subnet 192.168.1.0 /24 into address blocks using this formula:
$2^3 = 8$
To get at least six subnets, borrow 3 host bits. A subnet mask of 255.255.255.224 provides the 3 additional network bits.
To calculate the number of hosts, begin by examining the last octet. Notice these subnets:

0 = **000**00000
32 = **001**00000
64 = **010**00000
96 = **011**00000
128 = **100**00000
160 = **101**00000
192 = **110**00000
224 = **111**00000

Apply the host calculation formula:
$2^5 - 2 = 30$ hosts per subnet

Subnetting: Dividing Networks into Right Sizes

Every network within the internetwork of a corporation or organization is designed to accommodate a finite number of hosts. Some networks, such as point-to-point WAN links, only require a maximum of two hosts. Other networks, such as a user LAN in a large building or department, might need to accommodate hundreds of hosts. Network administrators need to devise the internetwork addressing scheme to accommodate the maximum number of hosts for each network. The number of hosts in each division should allow growth in the number of hosts.

To examine this process, see the example network in Figure 6-21. Each step of this process in the following sections will use this as an example. Subnetting an address block for an internetwork uses the following steps:

Step 1. Determine the total number of addresses.

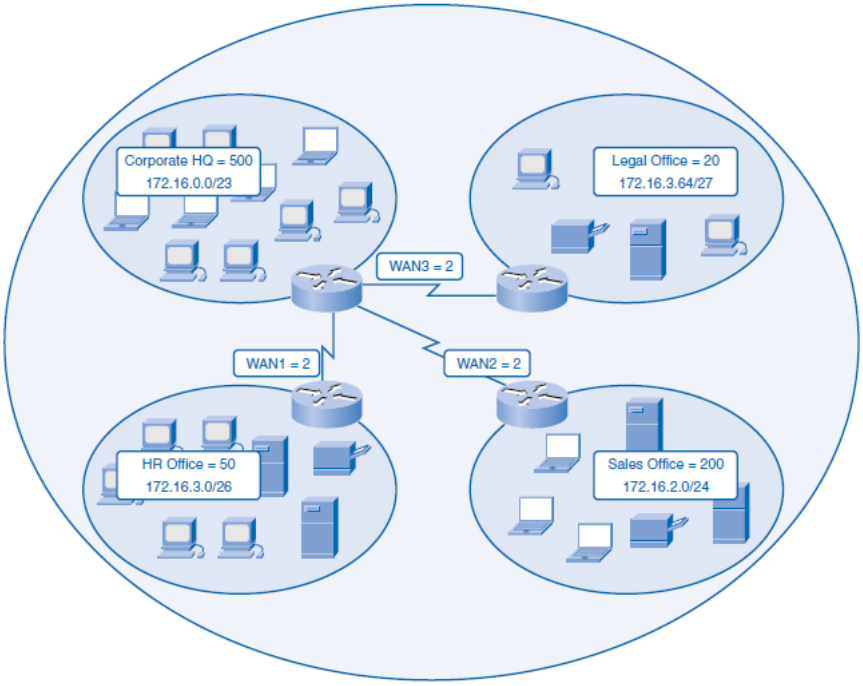
Step 2. Determine the number of networks and the number of hosts in each network.

Step 3. Partition the address block to create a network of appropriate size for the largest subnet network.

Step 4. Create another partition of appropriate size for the next largest network.

Step 5. Continue to create partitions for each subsequently smaller network until all subnets have address blocks assigned.

Figure 6-21 Example Network to Subnet



Allocating Addresses

With a count of the networks and the number of hosts for each network completed, you need to start allocating addresses from your overall block of addresses. This process begins by allocating network addresses for the locations that require the most hosts and work downward to the point-to-point links. This process ensures that large enough blocks of addresses are made available to accommodate the hosts and networks for these locations.

When making the divisions and assignment of available subnets, make sure that there are adequately sized address blocks available for the larger demands. Also, plan carefully to ensure that the address blocks assigned to the subnet do not overlap. A helpful tool in this planning process is a spreadsheet. You can place the addresses in columns to visualize the allocation of the addresses. This helps prevent the duplication of address assignments.

Figure 6-22 shows the use of a spreadsheet to plan address allocation. With the major blocks of the example network allocated, you subnet any of the locations that require further dividing. In this example, you divide the corporate HQ into two networks.

The subnets for this location are shown in Figure 6-23. This further division of the addresses is often called *subnetting the subnets*. As with any subnetting, you need to carefully plan the address allocation so that you have available blocks of addresses.

Figure 6-22 Subnets Planned on a Spreadsheet

Corporate Net	HQ	Sales	HR	Legal	WAN1	WAN2	WAN3	Unused
172.16.0.0/22	172.16.0.0/23	172.16.2.0/24	172.16.3.0/26	172.16.3.64/27	172.16.3.128/30	172.16.3.132/30	172.16.3.136/30	
172.16.0.1	172.16.0.1							
	172.16.1.255							
		172.16.2.0						
		172.16.2.255						
			172.16.3.0					
			172.16.3.63					
				172.16.3.64				
				172.16.3.127				
					172.16.3.129			
					172.16.3.131			
						172.16.3.132		
						172.16.3.135		
							172.16.3.136	
							172.16.3.139	
								172.16.3.140
172.16.3.255								172.16.3.255

Figure 6-23 Additional Subnetting of the HQ Location

HQ	HQ1	HQ2
172.16.0.0/23	172.16.0.0/24	172.16.1.0/24
172.16.0.1	172.16.0.0	
	172.16.0.255	
		172.16.1.0
172.16.1.255		172.16.1.255

As previously presented, the creation of new, smaller networks from a given address block is done by extending the length of the prefix, that is, adding 1s to the subnet mask. Doing this allocates more bits to the network portion of the address to provide more patterns for the new subnet. For each bit you borrow, you double the number of networks you have. For example, if you use 1 bit, you have the potential to divide that block into two smaller networks. With a single bit pattern, you can produce two unique bit patterns, 1 and 0. If you borrow 2 bits, you can provide four unique patterns to represent networks 00, 01, 10, and 11. Three bits would allow eight blocks, and so on.

Determine the Total Number of Hosts

Recall from the previous section that as you divide the address range into subnets, you lose two host addresses for each new network. These are the network address and broadcast address. The formula for calculating the number of hosts in a network is

Usable hosts = 2^n-2

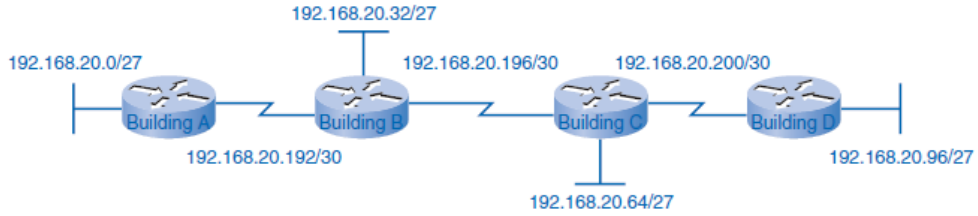
where *n* is the number of bits remaining to be used for hosts.

Subnetting a Subnet

Subnetting a subnet, or using variable-length subnet mask (VLSM), was designed to maximize addressing efficiency. VLSM is a practice associated with classless addressing. When identifying the total number of hosts using traditional subnetting, you allocate the same number of addresses for each subnet. If all the subnets have the same requirements for the number of hosts, these fixed-size address blocks would be efficient. However, that is most often not the case.

For example, the topology in Figure 6-24 shows a subnet requirement of seven subnets, one for each of the four LANs and one for each of the three WANs. With the given address of 192.168.20.0, you need to borrow 3 bits from the host bits in the last octet, which provides eight subnets, to meet your subnet requirement of seven subnets.

Figure 6-24 VLSM Subnetting



Subnet Number	Subnet Address
Subnet 0	192.168.20.0/27
Subnet 1	192.168.20.32/27
Subnet 2	192.168.20.64/27
Subnet 3	192.168.20.96/27
Subnet 4	192.168.20.128/27
Subnet 5	192.168.20.160/27
Subnet 6	192.168.20.192/27
Subnet 7	192.168.20.224/27

Subnet Number	Subnet Address
Subnet 0	192.168.20.192/30
Subnet 1	192.168.20.196/30
Subnet 2	192.168.20.200/30
Subnet 3	192.168.20.204/30
Subnet 4	192.168.20.208/30
Subnet 5	192.168.20.212/30
Subnet 6	192.168.20.216/30
Subnet 7	192.168.20.20/30

These bits are borrowed bits by changing the corresponding subnet mask bits to 1s to indicate that these bits are now being used as network bits. The last octet of the mask is then represented in binary by 11100000, which is 224. The new mask of 255.255.255.224 is represented with the /27 notation to represent a total of 27 bits for the mask.
In binary, this subnet mask is represented as 11111111.11111111.11111111.11100000.
After borrowing 3 of the host bits to use as network bits, this leaves 5 host bits. These 5 bits will allow up to 30 hosts per subnet. Although you have accomplished the task of dividing the network into an adequate number of subnets, it was done with a significant waste of unused addresses. For example, only two addresses are needed in each subnet for the WAN links. There are 28 unused addresses in each of the three WAN subnets that have been locked into these address blocks. Furthermore, this limits future growth by reducing the total number of subnets available. This inefficient use of addresses is characteristic of fixed-block sizes that are a carryover from practices with classful addressing. Applying a standard subnetting scheme to this scenario is inefficient. In fact, this example is a good model for showing how subnetting a subnet can be used to maximize address utilization.

Testing the Network Layer

Ping is a utility for testing IP connectivity between hosts. Ping sends out requests for responses from a specified host address. Ping uses a Layer 3 protocol that is a part of the TCP/IP suite called Internet Control Message Protocol (ICMP). Ping uses an ICMP echo request datagram.
If the host at the specified address receives the echo request, it responds with an ICMP echo reply datagram. For each packet sent, ping measures the time required for the reply. As each response is received, ping provides a display of the time between the ping being sent and the response being received. This is a measure of the network performance. Ping has a timeout value for the response. If a response is not received within that timeout, ping gives up and provides a message indicating that a response was not received. After all the sending of the requests, the ping utility provides an output with the summary of the responses. This output includes the success rate and average round-trip time to the destination.

Example 6-1 Successful Loopback Ping

```
C:\> ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Traceroute (tracert): Testing the Path

Ping is used to indicate the connectivity between two hosts. Traceroute (tracert) is a utility that allows you to observe the path between these hosts. The trace generates a list of hops that were successfully reached along the path. This list can provide you with important verification and troubleshooting information.

If the data reaches the destination, the trace lists the interface on every router in the path. If the data fails at some hop along the way, you have the address of the last router that responded to the trace. This is an indication of where the problem or security restrictions are.

Round-Trip Time (RTT)

Using traceroute provides the ***round-trip time (RTT)*** for each hop along the path and indicates whether a hop fails to respond. The RTT is the time a packet takes to reach the remote host and for the response from the host to return. An asterisk (*) is used to indicate a lost packet.

You can use this information to locate a problematic router in the path. High response times or data losses from a particular hop indicate that the resources of the router or its connections might be stressed.

Time to Live (TTL)

Traceroute uses a function of the Time to Live (TTL) field in the Layer 3 header and ICMP Time Exceeded message. The TTL field is used to limit the number of hops that a packet can cross. When a packet enters a router, the TTL field is decremented by 1. When the TTL reaches 0, a router will not forward the packet, and the packet is dropped.

In addition to dropping the packet, the router normally sends an ICMP Time Exceeded message addressed to the originating host. This ICMP message will contain the IP address of the router that responded.

The first sequence of messages sent from traceroute will have a TTL field of 1. This causes the TTL to time out the packet at the first router. This router then responds with an ICMP message. Traceroute now has the address of the first hop. Traceroute then progressively increments the TTL field (2, 3, 4, and so on) for each sequence of messages. This provides the trace with the address of each hop as the packets time out farther down the path. The TTL field continues to be increased until the destination is reached or it is incremented to a predefined maximum.

In Example 6-2, the tracert to www.cisco.com shows responses from the routers along the path. The local host sends a packet to the designation address of 198.133.219.2. The first response is a response from the host's default gateway, 10.20.0.94. The packet sent from the local host had a TTL = 1. When it reached this first router, the TTL was decremented to 0.

The router sends an ICMP message to indicate that the packet was dropped. The RTT indicates the amount of time required for this response. The local host sends two additional packets with a TTL = 1. For each one, the local gateway responds with a message, and an RTT is recorded.

Example 6-2 Trace to www.cisco.com

```
C:\> tracert www.cisco.com

Tracing route to www.cisco.com [198.133.219.25]
over a maximum of 30 hops:

  1    87 ms    87 ms    89 ms  sjck-access-gw2-vla30.cisco.com [10.20.0.94]
  2    89 ms    88 ms    87 ms  sjce-sbb1-gw1-gig3-7.cisco.com [171.69.14.245]
  3    88 ms    87 ms    88 ms  sjck-rbb-gw2-ten7-1.cisco.com [171.69.14.45]
  4    90 ms    87 ms    95 ms  sjck-corp-gw1-gig1-0-0.cisco.com [171.69.7.174]
  5    90 ms    88 ms    92 ms  sjce-dmzbb-gw1.cisco.com [128.107.236.38]
  6     *        *        *     Request timed out.
  7     *        *     ^C
C:\>
```

Overview of IPv6

In the early 1990s, the Internet Engineering Task Force (IETF) grew concerned about the exhaustion of the IPv4 network addresses and began to look for a replacement for this protocol.

This activity led to the development of what is now known as IPv6. This section presents a brief introduction to IPv6. Creating expanded addressing capabilities was the initial motivation for developing this new protocol. Other issues were also considered during the development of IPv6, such as these:

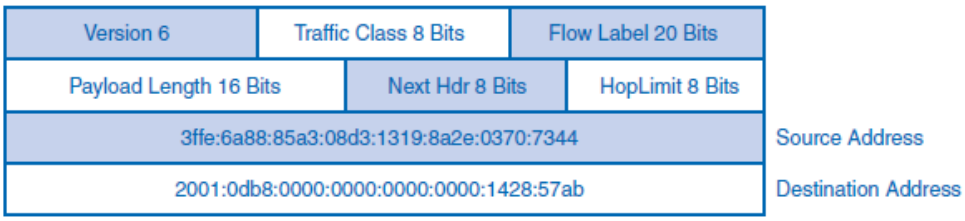
- Improved packet handling
- Increased scalability and longevity
- Quality of service (QoS) mechanisms
- Integrated security

To provide these features, IPv6 offers the following:

- 128-bit hierarchical addressing to expand addressing capabilities
- Header format simplification to improve packet handling
- Improved support for extensions and options for increased scalability/longevity and improved packet handling
- Flow-labeling capabilities as QoS mechanisms
- Authentication and privacy capabilities to integrate security

IPv6 is not merely a new Layer 3 protocol; it is a new protocol suite. New protocols at various layers of the stack have been developed to support this new protocol. There is a new messaging protocol called ICMPv6 and new routing protocols. Because of the increased size of the IPv6 header shown in Figure 6-29, it also impacts the underlying network infrastructure.

Figure 6-29 IPv6 Header



As you can see from this brief introduction, IPv6 has been designed with scalability to allow for years of internetwork growth. However, IPv6 is being implemented slowly and in select networks. Because of better tools, technologies, and address management in the past few years, IPv4 is still very widely used and is likely to remain so for some time into the future. However, IPv6 might eventually replace IPv4 as the dominant Internet protocol.

Chapter Six
Routing

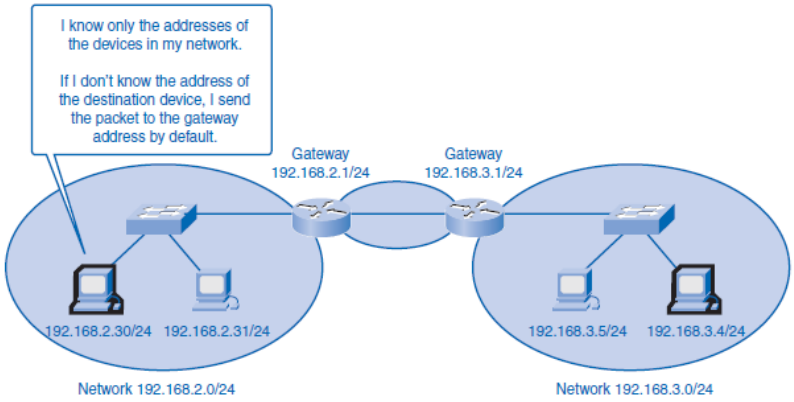
Routing: How Data Packets Are Handled

Communication within a network, or subnet, happens without a network layer device. When a host communicates outside the local network, a router acts as a gateway and performs the network layer function of choosing a path for the packet.

Device Parameters: Supporting Communication Outside the Network

As a part of its configuration, a host has a ***default gateway*** address defined. As shown in Figure 5-12, this gateway address is the address of a router interface that is connected to the same network as the host. The router interface is actually a host on the local network, so the host IP address and the default gateway address must be on the same network. Figure 5-12 shows that default gateways are members of their own local networks.

Figure 5-12 Gateways Enable Communications Between Networks



The default gateway is configured on a host. On a Windows computer, the Internet Protocol (TCP/IP) Properties tools are used to enter the default gateway IPv4 address. Both the host IPv4 address and the gateway address must have the same network (and subnet, if used) portion of their respective addresses.

IP Packets: Carrying Data End to End

The role of the network layer is to transfer data from the host that originates the data to the host that uses it. During encapsulation at the source host, an IP packet is constructed at Layer 3 to transport the Layer 4 PDU. If the destination host is in the same network as the source host, the packet is delivered between the two hosts on the local media without the need for a router.

However, if the destination host and source host are not in the same network, the packet can be carrying a transport layer PDU across many networks and through many routers. As it does, the information contained within is not altered by any routers when forwarding decisions are made.

At each hop, the forwarding decisions are based on the information in the IP packet header. The packet with its network layer encapsulation also is basically intact throughout the complete process, from the source host to the destination host. If communication is between hosts in different networks, the local network delivers the packet from the source to its gateway router. The router examines the network portion of the packet destination address and forwards the packet to the appropriate interface. If the destination network is directly connected to this router, the packet is forwarded directly to that host. If the destination network is not directly connected, the packet is forwarded to a second router that is the next-hop router.

The packet forwarding then becomes the responsibility of this second router. Many routers or hops along the way can process the packet before reaching the destination.

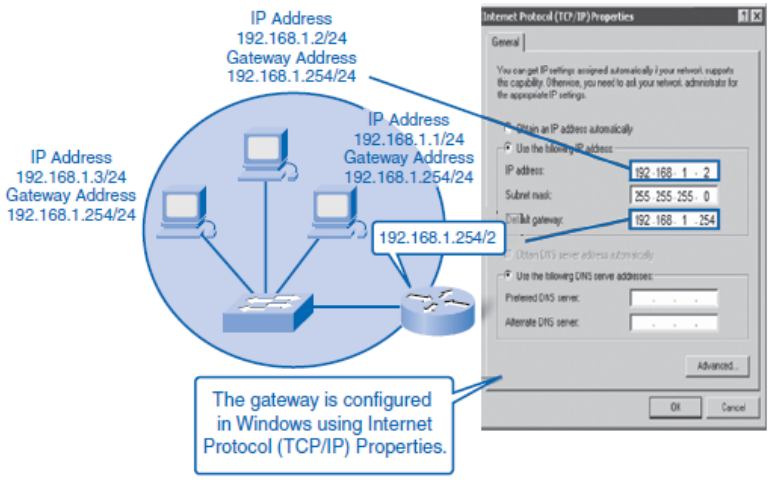
Gateway: The Way Out of the Network

The gateway, also known as the default gateway, is needed to send a packet out of the local network. If the network portion of the destination address of the packet is different from the network of the originating host, the packet has to be routed outside the original network. To do this, the packet is sent to the gateway. This gateway is a router interface connected to the local network. The gateway interface has a network layer address that matches the network address of the hosts. The hosts are configured to recognize that address as the gateway.

Default Gateway

The default gateway is configured on a host. On a Windows computer, the Internet Protocol (TCP/IP) Properties tools are used to enter the default gateway IPv4 address. Both the host IPv4 address and the gateway address must have the same network (and subnet, if used) portion of their respective addresses. Figure 5-13 depicts the Windows TCP/IP Properties configuration.

Figure 5-13 IP Address and Gateway Configuration in Windows



No packet can be forwarded without a route. Whether the packet is originating in a host or being forwarded by an intermediary device, the device must have a route to identify where to forward the packet.

A host must either forward a packet to the host on the local network or to the gateway, as appropriate. To forward the packets, the host must have routes that represent these destinations. A router makes a forwarding decision for each packet that arrives at the gateway interface. This forwarding process is referred to as *routing*. To forward a packet to a destination network, the router requires a route to that network. If a route to a destination network does not exist, the packet cannot be forwarded.

The destination network can be a number of routers or hops away from the gateway. The route to that network would only indicate the next-hop router to which the packet is to be forwarded, not the final router. The routing process uses a route to map the destination network address to the next hop and then forwards the packet to this next-hop address.

Confirming the Gateway and Route

An easy way to check the host IP address and default gateway is by issuing the **ipconfig** command at the command-line prompt of a Windows XP computer:

Step 1. Open the command-prompt window by clicking the Windows Start button in the lower-left corner of the desktop.

Step 2. Choose the Run icon.

Step 3. In the text box, type **cmd** and press **Enter**.

Step 4. The c:\Windows\system32\cmd.exe program is running. At the prompt, type **ipconfig** and press Enter. The Windows IP configuration will display with the IP address, subnet mask, and default gateway addresses.

Example 5-1 shows a sample of the **ipconfig** output with the host's IP address information.

Example 5-1 Confirming the IP Address and Gateway Route

```
C:\> ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . :

IP Address.....: 192.168.1.2
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.1.254
```

Route: A Path to a Network

A route for packets for remote destinations is added using the default gateway address as the next hop. Although it is not usually done, a host can also have routes manually added through configurations.

Like end devices, routers also add routes for the connected networks to their ***routing table***. When a router interface is configured with an IP address and subnet mask, the interface becomes part of that network. The routing table now includes that network as a directly connected network. All other routes, however, must be configured or acquired through a routing protocol. To forward a packet, the router must know where to send it. This information is available as routes in a routing table.

The routing table stores information about connected and remote networks. Connected networks are directly attached to one of the router interfaces. These interfaces are the gateways for the hosts on different local networks. Remote networks are networks that are not directly connected to the router. Routes to these networks can be manually configured on the router by the network administrator or learned automatically using dynamic routing protocols.

Routes in a routing table have three main features:

- Destination network
- Next-hop
- Metric

The router matches the destination address in the packet header with the destination network of a route in the routing table and forwards the packet to the next-hop router specified by that route. If there are two or more possible routes to the same destination, the metric is used to decide which route appears on the routing table.

Figure 5-14 shows a sample network with a local router and a remote router. Example 5-2 displays the routing table in the local router, which you can examine with the **show ip route** command from a router's console. From left to right, the output contains the destination network, the metric of [120/1], and the next hop through 192.168.2.2.

Figure 5-14 Confirming the Gateway and Route

Example 5-2 Router's Routing Table

```
Local_Router# show ip route

10.0.0.0/24 is subnetted, 2 subnets
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

As you know, packets cannot be forwarded by the router without a route. If a route representing the destination network is not on the routing table, the packet will be dropped (that is, not forwarded). The matching route could be either a connected route or a route to a remote network. The router can also use a default route to forward the packet. The ***default route*** is used when the destination network is not represented by any other route in the routing table.

Host Routing Table

Hosts require a local routing table to ensure that network layer packets are directed to thecorrect destination network. Unlike the routing table in a router, which contains both local and remote routes, the local table of the host typically contains its direct connection or connections (hosts can belong to more than one local network) and its own default route to the gateway. Configuring the default gateway address on the host creates the local default route.
Without a default gateway or route, packets destined outside the network will be dropped.
Figure 5-15 shows a simple network for the host routing table example that follows. The routing table of a computer host can be examined at the Windows command line by issuing the **netstat –r** or the **route print** command. Note that the host (192.168.1.2) serves as its own gateway to its own network (192.168.1.0) and has a default gateway for destinations outside the network pointing to the router interface (192.168.1.254).

Figure 5-15 Simple Network for Example 5-3

Follow these steps to display a local routing table on a host:
Step 1. Open the command-prompt window by clicking the Windows Start button in the lower-left corner of the desktop.
Step 2. Choose the Run icon.
Step 3. In the text box, type **cmd** and click the OK button or press **Enter**.
Step 4. The c:\Windows\system32\cmd.exe program is running. At the prompt, type **route print** or **netstat -r** and press Enter. The route table listing all known routes on the host will display.
Example 5-3 shows the host routing table.

Example 5-3 Host IP Routing Table Commands

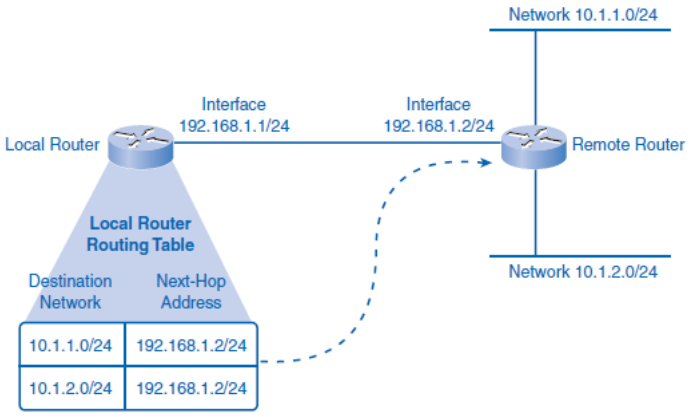
```
C:\> netstat -r

Route Table
===========================================================
Interface List
0x2….00 0f fe 26 f7 7b …Gigabit Ethernet - Packet Scheduler Miniport
===========================================================
Active Routes:
Network Destination    Netmask          Gateway          Interface        Metric
0.0.0.0                0.0.0.0          192.168.1.254    192.168.1.2      20
192.168.1.0            255.255.255.0    192.168.1.2      192.168.1.2      20
Default Gateway:       192.168.1.254
// output omitted //
===========================================================
```

When a host creates packets, it uses the routes it knows to forward them to the locally connected destination. These local network packets are delivered on the local route within the network without using a router. No packet is forwarded without a route. Whether the packet is originating in a host or being forwarded by an intermediary router, the device must have a route to identify which interface will be used to forward the packet. A host must either forward a packet to the host on the local network or to the gateway, as appropriate.

Routing

Routing is the process a router performs when making forwarding decisions for each packet arriving at the gateway interface. To forward a packet to a destination network, the router requires a route to that network. If a route to a destination network does not exist on the router, the packet will be forwarded to a default gateway. If no default gateway is configured, the packet cannot be forwarded. The destination network can be a number of routers or hops away from the gateway. If the router has an entry for the network in its routing table, it would only indicate the next-hop router to which the packet is to be forwarded, not the exact route to the final router. The routing process uses a routing table to map the destination network address to the next hop and then forwards the packet to this next-hop address. Figure 5-16 depicts a portion of a local router's routing table.

Figure 5-16 Local Router's Routing Table



Destination Network

For a router to route a packet to a destination network efficiently, it needs information about the route in its routing table. With millions of routes on the Internet, however, it is not reasonable to expect every route to be known to the router. The following sections describe how routers use information in routing tables and how packets can be forwarded when no information about routes can be found.

Routing Table Entries

The route, or destination network, in a routing table entry represents a range of host addresses and sometimes a range of network and host addresses. The hierarchical nature of Layer 3 addressing means that one route entry can refer to a large general network and another entry can refer to a subnet of that same network. When forwarding a packet, the router will select the most specific route that it knows. If a specific subnet is not in the routing table but the larger network that holds the subnet is known, the router will send it to the larger network, trusting that another router will find the subnet.

Consider Example 5-4. If a packet arrives at the router with the destination address of 10.1.1.55, the router forwards the packet to a next-hop router associated with a route to network 10.1.1.0. If a route to 10.1.1.0 is not listed in the routing table but a route to 10.1.0.0 is available, the packet is forwarded to the next-hop router for that network.

Example 5-4 Routes in a Routing Table

```
10.0.0.0/24 is subnetted, 2 subnets
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

The precedence the router uses for route selection for the packet going to 10.1.1.55 is as follows:

1. 10.1.1.0
2. 10.1.0.0
3. 10.0.0.0
4. 0.0.0.0 (default route if configured)
5. Dropped

In this case, the 10.1.1.0 network is known through 192.168.2.2, which is out the FastEthernet 0/0 interface.

Default Route

Remember that a default route is the route used if no specific route is available to be selected for delivery. In IPv4 networks, the address 0.0.0.0 is used for this purpose. Packets with a destination network address that does not match a more specific route in the routing table are forwarded to the next-hop router associated with the default route. The default route is also known as the *gateway of last resort*. When a default route is configured in a router, you can see it in the output, as noted in the first line of Example 5-5.

Example 5-5 Gateway of Last Resort

```
Gateway of Last Resort is 192.168.2.2 to Network 0.0.0.0
10.0.0.0/24 is subnetted, 2 subnets
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 192.168.2.2
```

Next Hop: Where the Packet Goes Next

The *next hop* is the address of the device that will process the packet next. For a host on a network, the address of the default gateway (router interface) is the next hop for all packets destined for another network.

As each packet arrives at a router, the destination network address is examined and compared to the routes in the routing table. The routing table lists an IP address for the next-hop router for the routes it knows. If a matching route is determined, the router then forwards the packet out the interface to which the next-hop router is connected. Example 5-6 outlines the association of routes with next hops and router interfaces.

Example 5-6 Routing Table Output with Next Hops

```
10.0.0.0/24 is subnetted, 2 subnets
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
R    10.1.1.0  [120/1] via 192.168.2.2, 00:00:08, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

As you can see in Example 5-6, some routes can have multiple next hops. This indicates that there are multiple paths to the same destination network. These are parallel routes that the router can use to select paths and forward packets.

Packet Forwarding: Moving the Packet Toward Its Destination

Routing is performed packet by packet and hop by hop. Each packet is treated independently by each router along the path. At each hop, the router examines the destination IP address for each packet and then checks the routing table for forwarding information. The router will then do one of the following with the packet:

- Forward it to the next-hop router
- Forward it to the destination host
- Drop it

A router takes the following steps to determine the appropriate action:

1. As an intermediary device, a router processes the packet at the network layer. However, packets that arrive at a router's interfaces are encapsulated as a data link layer (Layer 2) PDU. The router first discards the Layer 2 encapsulation so that the IP packet can be examined.

2. The router examines the IP address.

3. The router checks the routing table for a match.

4. The router selects the next hop. In the router, the destination address in a packet header is examined. If a matching route in the routing table shows that the destination network is directly connected to the router, the packet is forwarded to the interface to which that network is connected.

5. The router then does one of the following:

- **Scenario A: The router forwards the packet.** If the route matching the destination network of the packet is a remote network, the packet is forwarded to the indicated interface, encapsulated by the Layer 2 protocol, and sent to the next hop address. If the destination network is on a directly connected network, the packet has to be first reencapsulated by the Layer 2 protocol and then forwarded out the proper interface to the local network.
- **Scenario B: The router uses the default route.** If the routing table does not contain a more specific route entry for an arriving packet, the packet is forwarded to the interface indicated by a default route, if one exists. At this interface, the packet is encapsulated by the Layer 2 protocol and sent to the next-hop router.

The default route is also known as the *gateway of last resort*.

- **Scenario C: The router drops the packet.** If a packet is dropped, IP, by design, has no provision to return a packet to the sender or previous router. Such a function would detract from the protocol's efficiency and low overhead. Other protocols are used to report such errors.

Routing Processes: How Routes Are Learned

Routers need information about other networks to build a reliable routing table. Networks and routes are constantly changing, with new networks coming on and routes going down. If a router has bad information about routes, it is likely it will forward packets incorrectly, causing packets to be delayed or dropped. It is vital that routers have current information about neighboring routers to reliably forward packets. The two ways in which a router can learn information about routes is through static routing and dynamic routing. The following sections also introduce common routing protocols used by routers to dynamically share information.

Static Routing

The route information can be manually configured on the router, creating what is known as a ***static route***. An example of a static route is a default route. Static routing requires a network administrator for initial setup and for any changes to routes. Static routes are considered very reliable, and the router does not use much overhead to process packets. On the other hand, static routes do not update automatically and have higher continuing administrative costs.

If the router is connected to a number of other routers, knowledge of the internetworking structure is required. To ensure that the packets are routed to use the best possible next hops, each known destination network needs to either have a route or a default route configured. Because packets are forwarded at every hop, every router must be configured with static routes to next hops that reflect its location in the internetwork. Furthermore, if the internetwork structure changes or if new networks become available, these changes have to be manually updated on every router. If updating is not done in a timely fashion, the routing information can be incomplete or inaccurate, resulting in packet delays and possible packet loss.

Dynamic Routing

Routers can also learn about routes automatically from other routers in the same internetwork, which is known as ***dynamic routing***. Dynamic routing updates arrive from other routers and are used by the receiving router without administrative configuration. Dynamic routing has higher router processing overhead but little administrative cost after initial setup. If dynamic routing is not enabled and configured on a router, static routes to the next hops must be in place for the router to know where to forward packets.

Routing Protocols

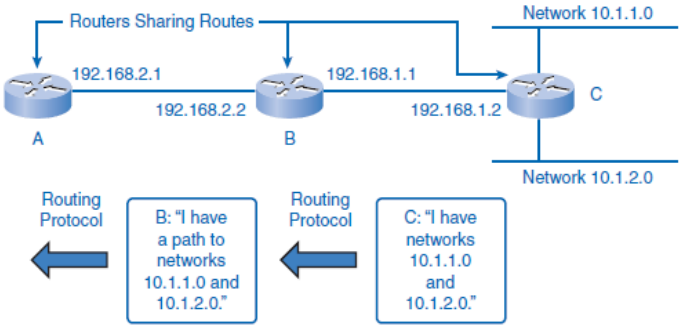
It is imperative that all routers in an internetwork have up-to-date and extensive route knowledge. Maintaining the routing table by manual static configuration is not always feasible. Configuring one of several available dynamic routing protocols on network routers is a much more efficient way to keep the routers updated.

Routing protocols are the set of rules by which routers dynamically share their routing information. As routers become aware of changes to the networks for which they act as the gateway, or changes to links between other routers, the information is passed on to other routers. When a router receives information about new or changed routes, it updates its own routing table and, in turn, passes the information to other routers. In this way, all routers have accurate routing tables that are updated dynamically and can learn about routes to remote networks that are many hops away. An example of routers sharing routes is shown in Figure 5-17.

The most common routing protocols used in this book are

- Routing Information Protocol (RIP)
- Enhanced Interior Gateway Protocol (EIGRP)
- Open Shortest Path First (OSPF)

Figure 5-17 Dynamic Route Sharing



Router B learns about Router C's networks dynamically.
Router B's next hop to 10.1.1.0 and 10.1.2.0 is 192.168.1.2 (Router C).
Router A learns about Router C's networks dynamically from Router B.
Router A's next hop to 10.1.1.0 and 10.1.2.0 is 192.168.2.2 (Router B).

The advantage of routing protocols providing routers with up-to-date routing tables is tempered by added overhead costs. The exchange of route information adds overhead by consuming network bandwidth. This overhead can be an issue with low-bandwidth links between routers. Another cost is the router's processing overhead. Not only does each packet need to be processed and routed, but updates from routing protocols also require complicated algorithmic calculations before the route information can be used in a routing table.

This means that routers employing these protocols must have sufficient processing capacity to both implement the protocol's algorithms and to perform timely packet routing and forwarding, which can add to initial network setup costs. Static routing does not produce network overhead and places entries directly into the routing table with no route processing required by the router. The cost for static routing, as mentioned earlier, is administrative time taken to manually configure and maintain routing tables in a manner that ensures efficient routing. In most internetworks, a combination of static (including default) and dynamic routes is used to provide efficient routing.

Chapter seven
Transmission Medium

Physical Layer Fundamental Principles

Communication at the physical layer is a process involving physical components that carry encoded data sent out as a signal appropriate to the medium. The following three components of Layer 1 communication are key to understanding how the physical layer functions:

- Physical components
- Encoding
- Signaling

There is some parallel between human communication and the processes of the physical layer. In a simplified communication model, when a person wants to communicate an idea to another, she processes an abstract thought into words, which are then encoded into speech sounds and sent out through the medium of air. At the other end, the receiver interprets the signal of sound, recognizes patterns in the sound that denote words, and then processes the meaning of the words into the original idea.

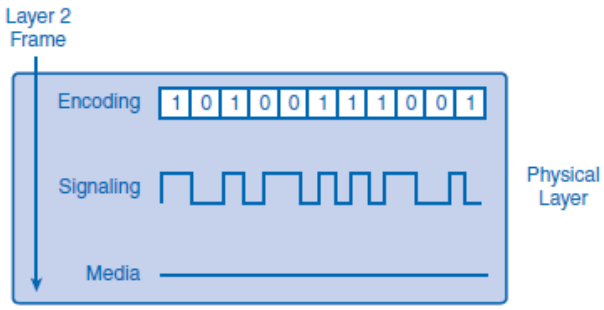
Exploring the analogy further, humans can be media independent when communicating by using a medium other than sound signals through the air. Gestures are conveyed in light, and written letters are conveyed by ink and paper. Each of these media has a unique way of ordering bits of communication into recognizable patterns indicating when messages begin and end.

This is also true of communication at the physical layer. Physical components carry the message in a reliable and consistent manner so that the receiver gets the message as it was sent. Encoding is another major function of the physical layer. The bits in the encapsulated data link layer frame need to be grouped, or encoded, into patterns recognized by Layer 1 devices. After transmission, the receiving Layer 1 device decodes patterns and hands the frame up to the data link layer. Another function of encoding is control information. Just as human speech uses pauses to indicate the start and end of sentences, the physical layer inserts a control code to indicate the beginning and end of frames. The control code is a specific pattern of 1s and 0s added to each end of the encoded frame. You learn more about encoding later in this chapter in the section "Encoding: Grouping Bits."

After the frame and control information are encoded into a string of binary digits, the bits are converted into a signal that will carry the pattern to the destination. Signaling is another key function of the physical layer. The process of signaling involves determining how to represent the binary bit on a specific medium. For example, if the medium is copper, the signal will be in the form of positive and negative patterns of voltage.

Figure 8-3 displays the process of a Layer 2 frame being encoded, converted to a signal, and then put onto the physical medium.

Figure 7-3 Physical Layer Processes



The processes of encoding and signaling complete the preparation of data for transport. The physical layer sends these bits out one at a time onto the medium as a signal, and those signals get picked up and decoded at the receiving end. There are several possible methods used to represent the binary digits as a signal, which are explored in the next section.

Signaling Bits for the Media

There are several different methods of representing these binary digits on physical media as a signal. Each method finds a way to convert a pulse of energy into a defined amount of time known as a ***bit time***. Bit time is the time it

takes for a NIC at OSI Layer 2 to generate 1 bit of data and send it out to the media as a signal. The signal will exist somewhere within the bit time and indicate the value of the bit to the receiver. The type of signal within the bit time depends on the method of signaling used.

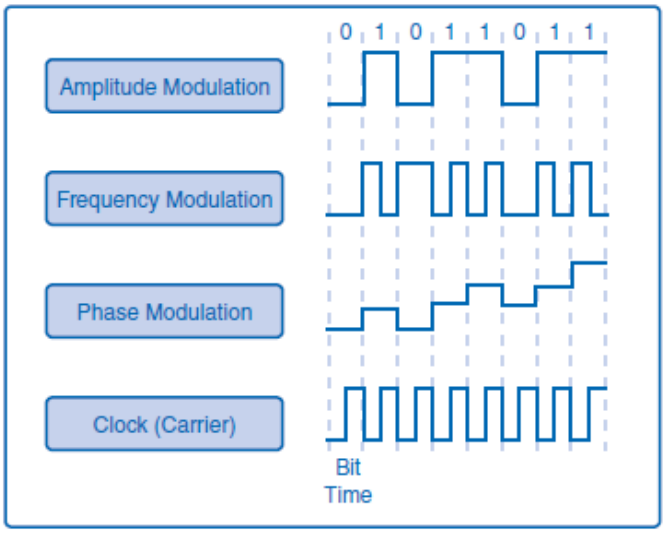
The amount of real time a bit time consumes depends on the speed of the NIC. Faster NICs produce shorter bit times. It is essential that the bits are read in order, and because the bit time can vary between different devices, there must be synchronization between the sending and the receiving units. Synchronization means that both sending and receiving units agree on the timing of the signals. Synchronization of the signals assures that the bits will be in order and can be properly interpreted by the receiving NIC. In local-area networks, each device keeps its own clock, and some signal methods include predictable transitions in the signal to provide synchronization.

Different signaling methods vary in the way they represent bits in the bit time. Three possible variations of a signal that can represent encoded bits are:

- Amplitude
- Frequency
- Phase

For example, amplitude is a measure of the variation of the signal cycle. The peak level of amplitude can represent a binary 1, and a lesser level of amplitude can denote a binary 0. Figure 8-4 demonstrates how each of these three characteristics can change within one bit time.

Figure 7-4 Signal Methods



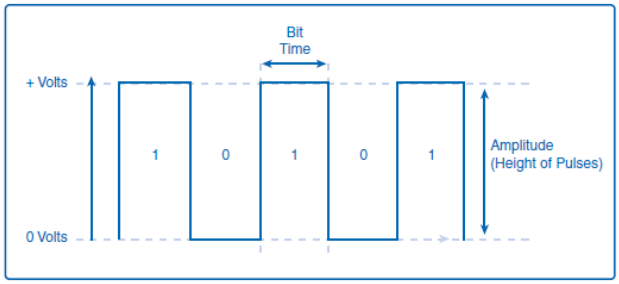
Different signal methods have different advantages and performance standards, but it is essential that all devices on the network use the same method so that the messages from sending devices can be read by the receiving devices. Signaling methods can be very complex, and an in-depth study of them is beyond the scope of this book, but a closer look at two methods—nonreturn to zero (NRZ) and Manchester encoding—provides a fundamental understanding of their function in the physical layer.

Nonreturn to Zero

The signaling method known as ***nonreturn to zero (NRZ)*** samples the voltage level on the medium during a bit time. The method defines which voltage levels represent 1 and 0, with a low voltage being 0 and a higher voltage representing a 1. The actual amount of voltage in the bit time can vary by standard. NRZ, as its name implies, has no constant zero voltage, so additional signaling is sometimes necessary for synchronization with other devices. This additional signal requirement limits the efficiency of NRZ and increases the risk of distortion if any common electromagnetic interference is present. This inefficiency relegates NRZ to use on lower-speed links.

Figure 8-5 depicts an NRZ signal representing both 1 and 0.

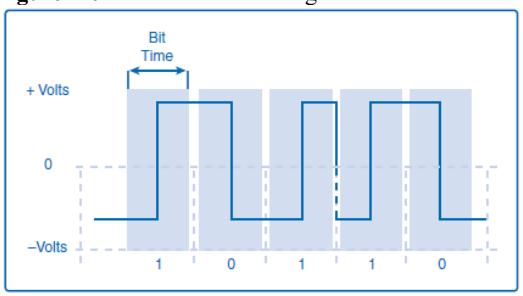
Figure 7-5 NRZ Encoding



Manchester Encoding

Manchester encoding is a signaling method that looks for a change in voltage in the middle of a bit time. A voltage change from low to high within the bit time represents a 1. Conversely, a voltage drop within the bit time from a high to a low voltage represents a 0. When there are repeating bit values, meaning consecutive 1s or 0s, a transition will occur at the edge of the bit time, so a repeated rise or fall will occur in the middle of the bit time. Figure 8-6 demonstrates the Manchester encoding method of voltage changes in the middle of a bit times as well as a repeating bit transition at the edge of the bit time.

Figure 7-6 Manchester Encoding



Data-Carrying Capacity

Each physical layer medium carries data at a different speed. There are three different ways to analyze the transfer speed of data on a medium:

- Theoretically as bandwidth
- Practically as throughput
- Qualitatively as goodput

Although each of these items measures a different aspect of data transfer, all three are measured by the same standard of bits per second.

Bandwidth is the capacity of a medium to carry data in a given amount of time. The standard measure for bandwidth is in bits per second (bps). As the technologies have improved over the years, it has become more practical to refer to bandwidth in ***kilobits***, or thousands of bits per second (kbps), and ***megabits***, or millions of bits per second (Mbps). The bandwidth measurement takes into account the physical properties of the medium and the signaling method applied to it. Table 8-2 lists the four most common units of measure for bandwidth along with the mathematical equivalence for each.

Table 7-2 Bandwidth Units of Measure

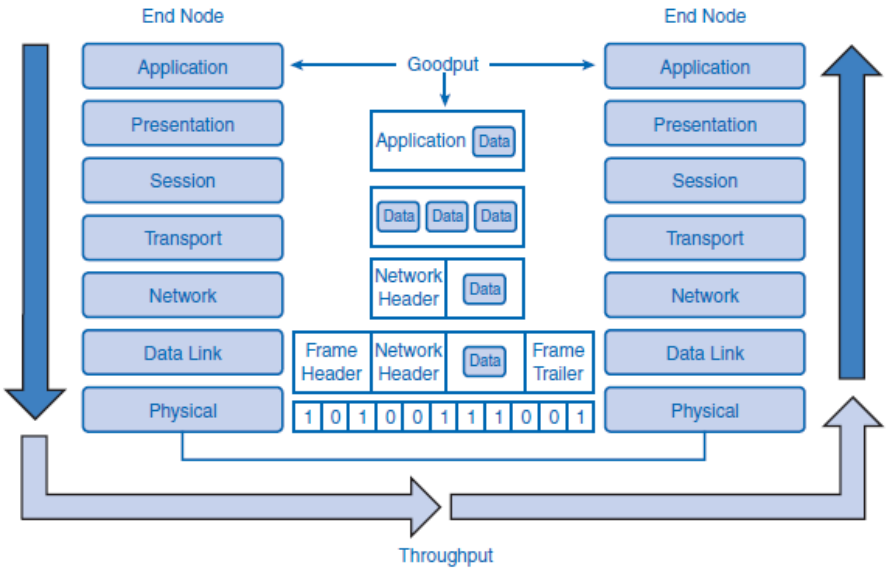
Units of Bandwidth	Abbreviation	Equivalence
Bits per second	bps	1 bps = Base unit
Kilobits per second	kbps	1 kbps = 1000 bps = 10^3 bps
Megabits per second	Mbps	1 Mbps = 1,000,000 bps = 10^6 bps
Gigabits per second	Gbps	1 Gbps = 1,000,000,000 bps = 10^9 bps
Terabits per second	Tbps	1 Tbps = 1,000,000,000,000 bps = 10^{12} bps

Throughput is the actual transfer rate of data over the medium in a period of time. Bandwidth is the capacity for moving data, but attaining that capacity is rare because of factors such as interference and errors. It is useful to plan networks around expected throughput and the actual rate of speed rather than theoretical bandwidth. Throughput, like bandwidth, is measured in bits per second.
Many factors influence throughput, including the following:

- The amount of traffic
- The type of traffic
- The number of network devices encountered on the network being measured

In a multiaccess topology such as Ethernet, nodes are competing for media access and its use. Therefore, the throughput of each node is degraded as usage of the media increases. In an internetwork or network with multiple segments, throughput cannot be faster than the slowest link of the path from source to destination. Even if all or most of the segments have high bandwidth, it will take only one segment in the path with low throughput to create a bottleneck to the throughput of the entire network.
Goodput is the transfer rate of actual usable data bits. Goodput is the data throughput less the protocol overhead bits, error corrections, and retransmission requests. The difference between goodput and throughput can vary greatly depending on the quality of network connections and devices.
Unlike throughput, which measures the transfer of bits and not the transfer of usable data, goodput accounts for bits devoted to protocol overhead? Goodput is throughput minus traffic overhead for establishing sessions, acknowledgments, and encapsulation.
As an example, consider two hosts on a LAN transferring a file. The bandwidth of the LAN is 100 Mbps. Because of the sharing and media overhead, the throughput between the computers is only 60 Mbps. With the overhead of the encapsulation process of the TCP/IP stack, the actual rate of the data received by the destination computer, goodput, is only 40 Mbps.
Figure 8-10 depicts the difference between throughput and goodput. In this example, throughput measures network performance and goodput measures the transfer rate of application layer data.

Figure 7-10 Throughput and Goodput



Physical Media: Connecting Communication

The physical layer is concerned with network media and signaling. This layer produces the representation and groupings of bits as voltages, radio frequencies, or light pulses. Various standards organizations have contributed to the definition of the physical, electrical, and mechanical properties of the media available for different data communications. These specifications guarantee that cables and connectors will function as anticipated with different data link layer implementations.

Types of Physical Media

The physical layer defines the performance standards for the physical components of a network such as copper and fiber cables and the connectors used on them. The physical layer also defines how bits are presented in the form of voltage, light pulses, and radio signals. The design of the physical layer differs from the design of upper layers in that it deals with the physics and electrical properties of the media rather than the logical processes. This section explores copper, fiber, and wireless media. Table 8-3 lists several different Ethernet standards for copper and fiber-optic media.

	Media	Maximum Segment Length	Topology	Connector
10BASE-T	EIA/TIA Category 3, 4, or 5 UTP, four-pair	100 m (328 feet)	Star	ISO 8877
100BASE-TX	EIA/TIA Category 5 UTP, two-pair	100 m (328 feet)	Star	—
100BASE-FX	5.0/62.5-micron multimode fiber	2 km (6562 feet)	Star	ISO 8877 (RJ-45)
100BASE-CX	STP	25 m (82 feet)	Star	ISO 8877 (RJ-45)
1000BASE-T	EIA/TIA Category 5 (or greater) UTP, four-pair	100 m (328 feet)	Star	—
1000BASE-SX	5.0/62.5-micron multimode fiber	Up to 550 m (1804 feet), depending on fiber used	Star	—
1000BASE-LX	5.0/62.5-micron multimode fiber or 9-micron	550-m multimode fiber, 10-km single-mode fiber	Star	—
1000BASE-ZX	Single-mode fiber	Approx. 70 km	Star	—
10GBASE-ZR	Single-mode fiber	Up to 80 km	Star	—

Copper Media

The most pervasive media in use for data transfer in local networks is copper. Copper cable standards and technologies have evolved over the past few decades, but copper remains the most common medium for connecting network devices. Copper connects hosts to devices such as routers, switches, and hubs within a LAN. Copper media has standards defined for each of the following:

- Type of copper cabling used
- Bandwidth of the communication
- Type of connectors used
- Pinout and color codes of connections to the media
- Maximum distance of the media

Copper is an effective medium because it conducts electrical signals very well, but it has its limitations. Data travels on copper cables as small pulses of electrical voltage. The voltage is quite low and easily distorted by outside interference and signal attenuation. ***Attenuation*** is the loss of energy in a signal as it travels longer distances. The timing and voltage values of these signals are susceptible to interference or ***noise*** from outside the communications system. These unwanted signals can distort and corrupt the data signals being carried by copper media. Radio waves and electromagnetic devices, such as fluorescent lights, electric motors, and other devices, are potential sources of noise.

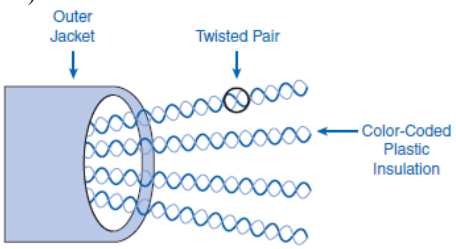
The advances in copper cable design have improved data transfer rates by reducing the effects of noise and signal attenuation on the wire. But improving the cable design is only part of a solution to interference. Architects designing new buildings can locate network devices away from building systems that generate electromagnetic interference. Cable installers can use quality cabling practices to enhance reliability at the physical layer, and choosing the proper cable type for the intended environment ensures the best possible performance.

There are different types of copper cable designed to meet the specific needs of different networks. The most common is unshielded twisted-pair (UTP) cabling, as it is used in Ethernet LANs. Others are coaxial cable and shielded twisted-pair cables. The features of each type are described in the following sections.

Unshielded Twisted-Pair (UTP) Cable

The most common copper network media is ***unshielded twisted-pair (UTP) cable***. UTP in Ethernet consists of eight wires twisted into four color-coded pairs and then wound inside a cable jacket. The colored pairs identify the wires for proper connection at the terminals.
Figure 8-11 depicts the twisted-wire pairs inside the cable jacket.

Figure 7-11 Unshielded Twisted-Pair (UTP) Cable

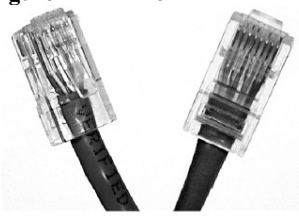


direction, keeping them close together with twisting will cause the magnetic fields on the wire pair to cancel each other. This magnetic interference from wires within the cable is called *crosstalk*. The rate of twisting in each pair of wires is different so that each pair self cancels and reduces crosstalk to a minimum.
Computer equipment manufactures build equipment to industry standards so that different systems can interoperate. It is important that these standards apply to the installation of cables and connectors in LANs. The previously mentioned TIA/EIA engineering groups in the telecommunications industry define the following standards for UTP cable installations:

- Cable types
- Cable lengths
- Connectors
- Cable termination
- Methods of testing cable

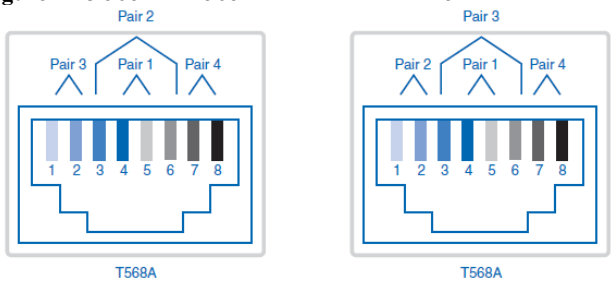
It is essential that the physical layer cable installations and connections closely follow industry standards. Poor cabling is a very common culprit in poor network performance. There are several categories of UTP cable. Each category indicates a level of bandwidth performance as defined by the IEEE. Important cable category changes were from Category 3 (Cat 3) to Category 5 (Cat 5), where UTP cable improvements allowed 100-megabit transmissions. In 1999, the Cat 5 standard improved to Cat 5e, which enabled full-duplex Fast Ethernet gigabit transmission over UTP cable.
In 2002, Category 6 (Cat 6) was defined. Cat 6 cable offers stricter manufacturing and termination standards that allow higher performance and less crosstalk. Cat5e is still acceptable for most LANs, but Cat 6 is the current recommended standard for gigabit connections and in new installations, as it will more readily allow future growth in LAN performance. Like all cable category upgrades, Cat 6 remains backward compatible with previous generation cable categories. The most common UTP cable connector in LAN devices is an ***RJ-45*** connector. Most computers accessing a network through cable use an RJ-45 connector plugged into the computer network interface card at one end and a hub or switch device at the other. An RJ-45 connector is commonly mistaken for a telephone jack, but an RJ-45 jack is larger and has a different cable termination. Figure 8-12 depicts RJ-45 connectors terminating a UTP cable.

Figure 7-12 RJ-45 Connector



The cable wires inserted into the RJ-45 connector are not always ordered in the same way. The required order of the wires in the connector, called the ***pinout***, varies according to where the cable fits in the network. The order of the wires in the pinouts is defined by TIA/EIA standards 568A and 568B. Each device connection requires a specific cable pinout to ensure that signals transmitted on a wire at one end arrive on the correct "receive" circuit at the other end of the cable.
Figure 8-13 shows the color patterns for TIA/EIA 568A and 568B pinouts. As you can see in the figure, the difference between 568A and 568B is simply the switched position of wire pair 2 and wire pair 3.

Figure 7-13 568A and 568B Pinouts on an RJ-45 Connector



At one time, different geographic areas adapted 568A and others 568B. One is not necessarily better than the other, but the important thing is to use the same standard throughout the network. Three UTP cable types are described in this book. Table 8-4 lists the specifications and purpose of each cable.

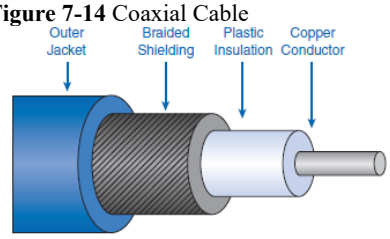
Table 7-4 UTP Cable Types

Cable Type	TIA/EIA Standard	Cable Use
Straight-through cable	Both ends the same, either 568A or 568B.	Connects a network host to a hub or switch.
Crossover cable	One end 568A, and the other 568B. It does not matter which end goes to which device.	Directly connects like devices, such as two hosts, two switches, or two routers. Also used to directly connect a host to a router.
Rollover cable (also known as a "Cisco" cable)	Cisco-proprietary.	Connects a workstation serial port to a Cisco device console port.

Other Copper Cable Types

One of the first types of copper cable used in LANs was coaxial cable. ***Coaxial cable***, also known as ***coax***, has a single, coated copper wire center and an outer metal mesh that acts as both a grounding circuit and an electromagnetic shield to reduce interference. The outer layer is the plastic cable jacket. The use of coax has migrated from LAN media, where it was once common but now is a legacy technology, to uses in wireless implementations connecting antennas to wireless devices. Figure 7-14 depicts the structure of coaxial cable.

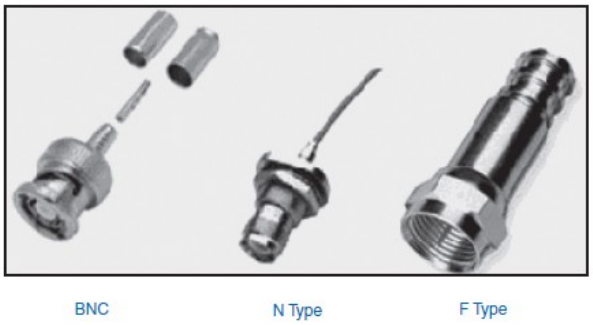
Figure 7-14 Coaxial Cable



For decades, coax has carried high-frequency radio and television signals over wire. The "cable" in cable TV is coax, and its use in TV is evolving from a one-way broadcast system to a two-way communication medium using new coax technologies known as ***hybrid fibercoax (HFC)***. HFC combines the electrical properties of coax and the bandwidth and distance benefits of fiber-optic cable.

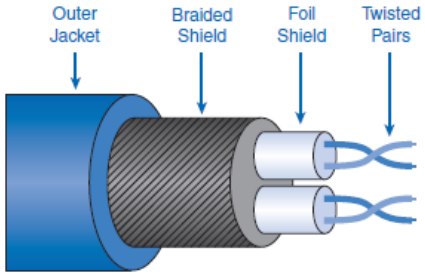
Coax cable connects to a host's NIC and other devices with a barrel connector. Some of the connectors have special terminators to help control the electrical interference on the line.
Figure 7-15 depicts the examples of coaxial cable connectors.

Figure 7-15 Coaxial Cable Connectors



Shielded twisted-pair (STP) cable is a LAN technology that has become less commonly used in recent years. STP cable was a standard in the IBM Token Ring network technology, but its use has faded as Token Ring networks have been replaced with other Ethernet technologies.
STP cable combines two methods of noise reduction by twisting the pairs of wire inside the cable to reduce interference and then shielding the cable in a wire mesh. STP can still be useful in installations where electromagnetic interference (EMI) is an issue, but STP cable is much more expensive than other available cable, so its use is quite limited at this time. Figure 7-16 depicts the structure of STP cable.

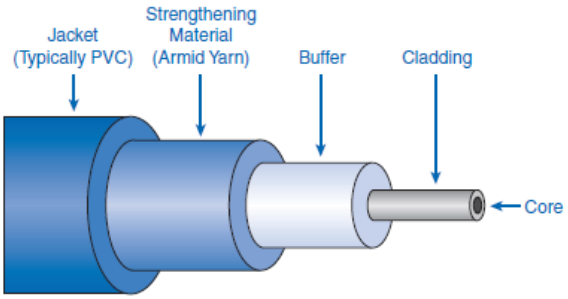
Figure 7-16 Shielded Twisted-Pair (STP) Cable



Fiber Media

At the physical layer, diverse technologies can perform the same function of data transfer. Fiber-optic cable is very different from copper, yet both effectively carry data over networks. Whereas copper uses electrical voltage to represent data on the wire, ***fiber-optic cable*** uses light pulses conducted through special glass conductors to carry data. The cable is engineered to be as pure as possible and to allow reliable light signals to traverse the medium.
Like copper media, the standards and performance levels of fiber are constantly improving. Fiber has some advantages over copper, but there are also some challenges when installing fiber in a network. For example, fiber has greater bandwidth and can run much farther than cable without needing a signal enhanced, but the higher cost of fiber-optic cable and connectors, along with special training required for installing fiber, limits its feasibility to special uses. Fiber cable also requires more special handling than copper cable.
Fiber is an answer to the safety issues of long copper runs mentioned in the previous section. Because fiber does not carry voltage and current, it is immune to the earth ground and lightning concerns. Because it is safer and can carry data much farther than copper, fiberoptic cable is usually considered the best choice for backbone connections between floors and wiring closets in large buildings and for connections between buildings on a campus.
Fiber-optic cable starts with a core strand of glass or special plastic on which the light signal travels. Around the glass is *cladding*, a special material that reflects escaping light into the core. Outer layers protect and strengthen the vulnerable center core from moisture and damage. Figure 7-17 depicts a cutout of a fiber-optic cable.

Figure 7-17 Fiber-Optic Cable



Fiber-optic cable can carry light in only one direction, so fiber cables usually include a pair of fiber cores. This allows full-duplex transmission, which is the transmitting and receiving of data simultaneously on one cable.

The light carried on fiber cables is generated by either a laser or a light emitting diode (LED) that converts the data to light pulses. The lasers used in fiber-optic cables can be intense and can damage the human eye, so great care is required when troubleshooting or installing the cable.

At the receiving end, devices called *photodiodes* interpret the light signal, decode the bit pattern, and send it up to the data link layer. There are two basic types of fiber-optic cable: single-mode and multimode.

Figure 7-18 displays single-mode and multimode cables.

Figure 7-18 Single-Mode and Multimode Fiber-Optic Cable

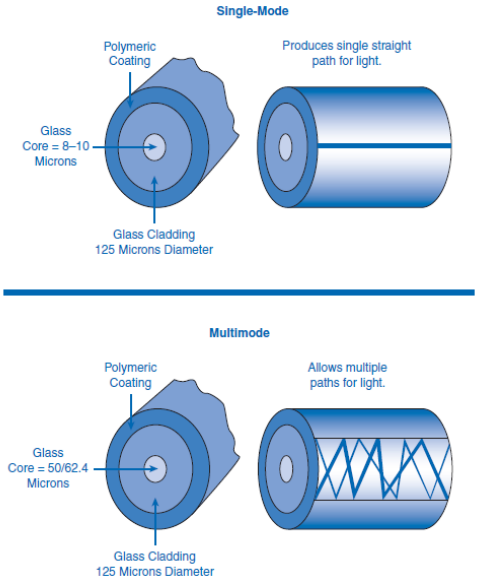


Table 7-5 describes the differences between single-mode and multimode fiber-optic cable.

Table 7-5 Single-Mode and Multimode Fiber-Optic Cable

Single-Mode	Multimode
Small glass core: 8–10 microns	Larger core: 50+ microns, can be glass or plastic
Less dispersion of light	Greater dispersion (loss of light)
Longer distance: Up to about 100 km	Shorter distance: Up to 2 km
Uses lasers as light source	Uses LEDs as light source on shorter runs

Dispersion of the light signal means that it separates as it travels. Because the cladding helps contain the intense laser light in the smaller glass core, the single-mode fiber can carry data greater distances. The dispersion rate is greater in multimode fiber, so the signal does not travel as far. Fiber-optics are economical on longer, high-speed,

point-to-point backbone runs, but they are not currently well suited for local connections between hosts and other network devices.

Wireless Media

A third distinct physical layer technology is wireless. Wireless media carry electromagnetic radio signals that represent the binary data of the data-link frame. Wireless technologies transmit and receive signals through the medium of the open atmosphere, which frees users from having to connect to a copper or fiber cable connection.

Open areas are best for wireless connections. Within buildings, interference occurs from physical objects such as walls, metal air ducts, and floors and machinery. The wireless signal is also subject to degradation from small appliances, microwave ovens, fluorescent lighting, and household wireless devices like phones and Bluetooth devices.

Although wireless has advantages, there are some disadvantages to its use. A wireless connection is usually slower than a cable connection, and because the medium is open to anyone with a wireless receiver, it is more susceptible to security breaches than other media.

The IEEE and telecommunications industry standards for wireless data communications cover both the data link and physical layers. Following are four common data communications standards that apply to wireless media:

- **Standard IEEE 802.11:** Commonly referred to as *Wi-Fi*, 802.11 is a wireless LAN (WLAN) technology that uses a contention or nondeterministic system with a carrier sense multiple access/collision avoid (CSMA/CA) media access process.
- **Standard IEEE 802.15:Wireless Personal-Area Network (WPAN):** Commonly known as Bluetooth, 802.15 uses a device-pairing process to communicate over distances from 1 to 100 meters.
- **Standard IEEE 802.16:** Commonly known as *WiMAX (Worldwide Interoperability for Microwave Access)*, 802.16 uses a point-to-multipoint topology to provide wireless broadband access.
- **Global System for Mobile Communication (GSM):** Includes physical layer specifications that enable the implementation of the Layer 2 General Packet Radio Service (GPRS) protocol to provide data transfer over mobile cellular telephony networks.

Other wireless technologies, such as satellite communications, provide data network connectivity for locations without another means of connection. Protocols including GPRS enable data to be transferred between earth stations and satellite links.

In each of these examples, physical layer specifications are applied to areas that include data–to–radio signal encoding, frequency and power of transmission, signal reception and decoding requirements, and antenna design and construction.

Wireless LAN

A common wireless data implementation is enabling devices to wirelessly connect through a LAN. In general, a wireless LAN requires the following network devices:

- **Wireless access point (AP):** Concentrates the wireless signals from users and connects, usually through a copper cable, to the existing copper-based network infrastructure such as Ethernet
- **Wireless NIC adapter:** Provides wireless communication capability to each network host As the technology has developed, a number of WLAN Ethernet-based standards have emerged. Care needs to be taken in purchasing wireless devices to ensure compatibility and interoperability.

Table 8-6 describes four of the basic 802.11 standards in use. Within each standard, the physical layer specifications pertain to the radio signal, the encoding of data, and the frequency and power of the transmission signals.

Table 7-6 802.11 Wireless LAN Standards

IEEE Standard	Description
IEEE 802.11a	■ Operates in the 5-GHz frequency band ■ Speeds of up to 54 Mbps ■ Small coverage area ■ Not interoperable with the 802.11b and 802.11g standards
IEEE 802.11b	■ Operates in the 2.4-GHz frequency band ■ Speeds of up to 11 Mbps ■ Longer range and better able to penetrate building structures than devices based on 802.11a
IEEE 802.11g	■ Operates in the 2.4-GHz frequency band ■ Speeds of up to 54 Mbps ■ Same radio frequency and range as 802.11b but with the bandwidth of 802.11a
IEEE 802.11n	■ Standard is currently in draft form ■ Proposed 2.4-GHz or 5-GHz ■ Expected data rates are 100 Mbps to 210 Mbps, with a distance range of up to 70 meters

The cost savings and ease of access are the major benefits of wireless LANS, with network security being the major caveat.