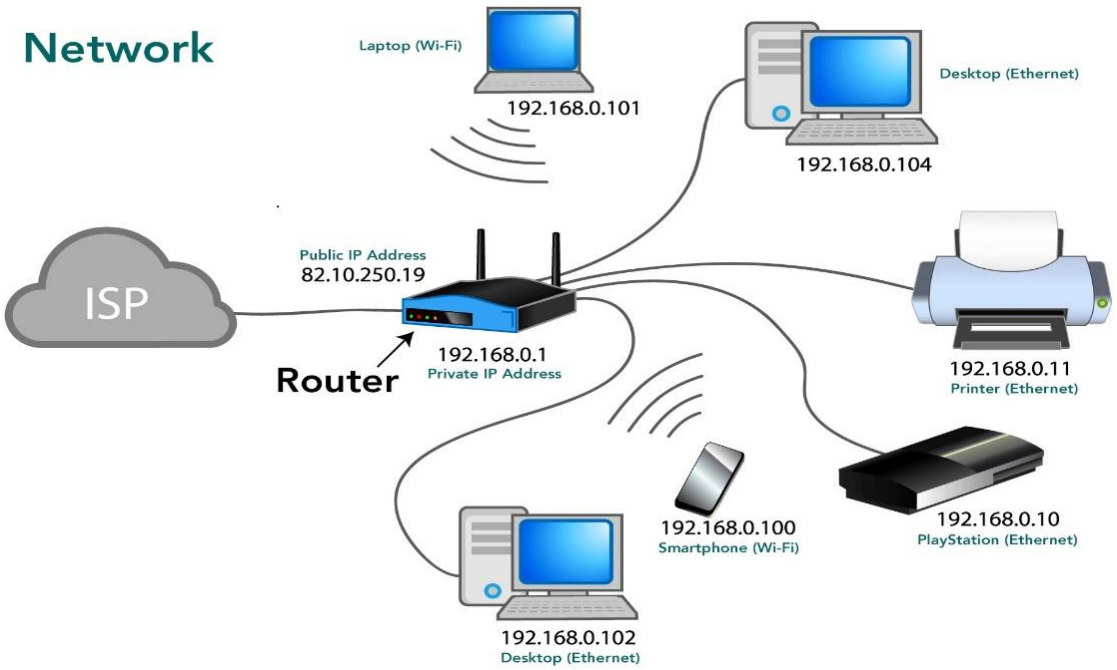
1.1.1-> what is a Network?

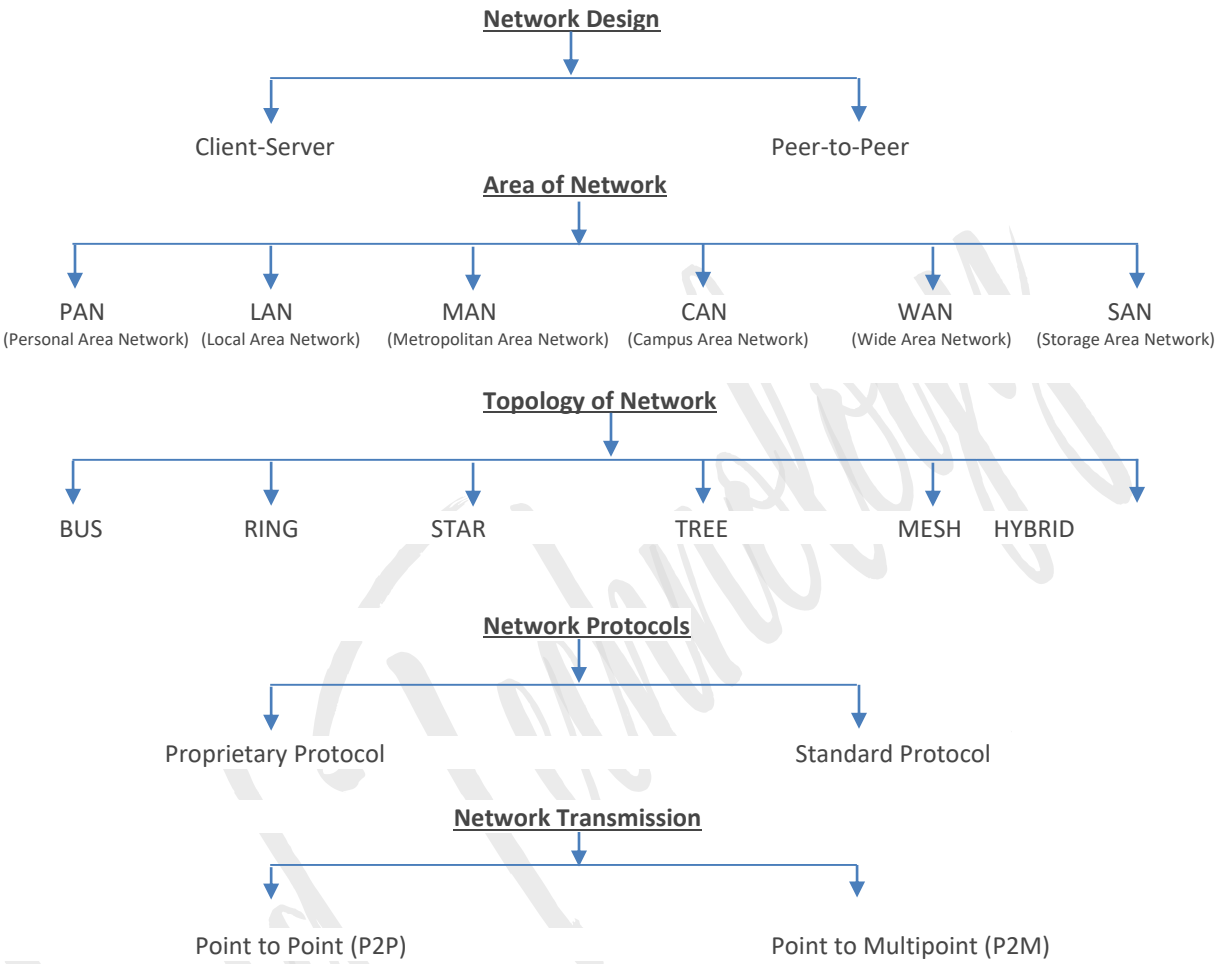
A **network** is a collection of computers, servers, mainframes, network devices, peripherals, or other devices connected to one another to allow the sharing of data. or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.

An example of a network is the Internet, which connects millions of people all over the world.



Types of Network-

- Network Design
- Area of Network
- Network Topology
- Network Protocols
- Network Transmission

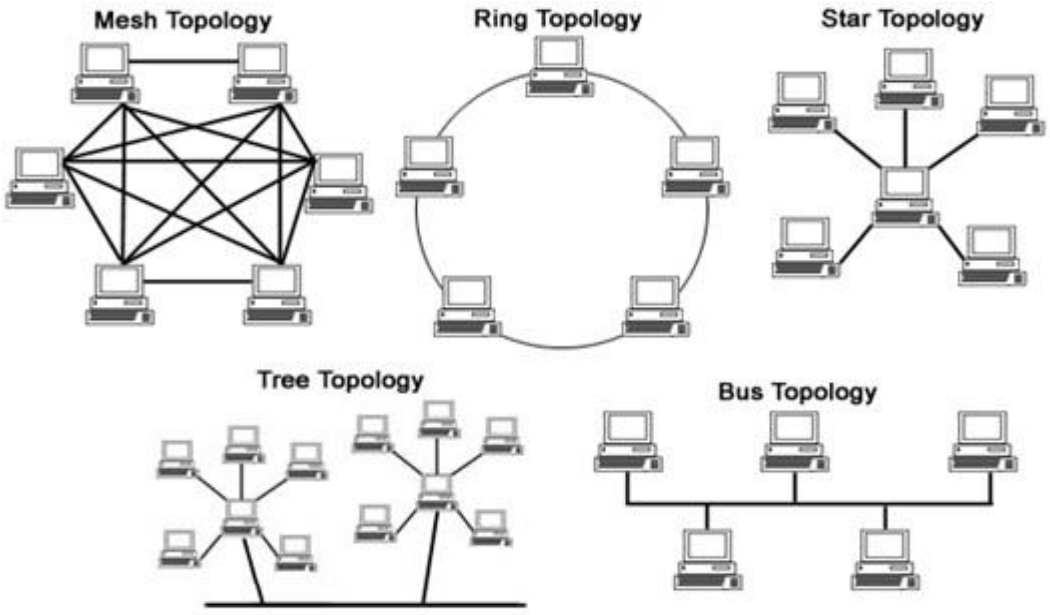


What is Topology?

Topology defines the structure of the network of how all the components are interconnected to each other.

There are two types of topology:

1> physical topology
2> logical topology



Bus Topology

- The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
- Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable.
- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- The bus topology is mainly used in 802.3 (Ethernet) and 802.4 standard networks.
- The configuration of a bus topology is quite simpler as compared to other topologies.
- The backbone cable is considered as a **"single lane"** through which the message is broadcast to all the stations.
- The most common access method of the bus topologies is **CSMA** (Carrier Sense Multiple Access).

CSMA: It is a media access control used to control the data flow so that data integrity is maintained, i.e., the packets do not get lost. There are two alternative ways of handling the problems that occur when two nodes send the messages simultaneously.

- **CSMA CD:** CSMA CD (**Collision detection**) is an access method used to detect the collision. Once the collision is detected, the sender will stop transmitting the data. Therefore, it works on "**recovery after the collision**".

- **CSMA CA: CSMA CA (Collision Avoidance)** is an access method used to avoid the collision by checking whether the transmission media is busy or not. If busy, then the sender waits until the media becomes idle. This technique effectively reduces the possibility of the collision. It does not work on "recovery after the collision".

Advantages of Bus topology:

- **Low-cost cable:** In bus topology, nodes are directly connected to the cable without passing through a hub. Therefore, the initial cost of installation is low.
- **Moderate data speeds:** Coaxial or twisted pair cables are mainly used in bus-based networks that support upto 10 Mbps.
- **Familiar technology:** Bus topology is a familiar technology as the installation and troubleshooting techniques are well known, and hardware components are easily available.
- **Limited failure:** A failure in one node will not have any effect on other nodes.

Disadvantages of Bus topology:

- **Extensive cabling:** A bus topology is quite simpler, but still it requires a lot of cabling.
- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Signal interference:** If two nodes send the messages simultaneously, then the signals of both the nodes collide with each other.
- **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
- **Attenuation:** Attenuation is a loss of signal leads to communication issues. Repeaters are used to regenerate the signal.

Ring Topology

- Ring topology is like a bus topology, but with connected ends.
- The node that receives the message from the previous computer will retransmit to the next node.
- The data flows in one direction, i.e., it is unidirectional.
- The data flows in a single loop continuously known as an endless loop.
- It has no terminated ends, i.e., each node is connected to other node and having no termination point.
- The data in a ring topology flow in a clockwise direction.
- The most common access method of the ring topology is **token passing**.
    - **Token passing:** It is a network access method in which token is passed from one node to another node.
    - **Token:** It is a frame that circulates around the network.

Working of Token passing

- A token moves around the network, and it is passed from computer to computer until it reaches the destination.
- The sender modifies the token by putting the address along with the data.
- The data is passed from one device to another device until the destination address matches. Once the token received by the destination device, then it sends the acknowledgment to the sender.

- In a ring topology, a token is used as a carrier.

Advantages of Ring topology:

- **Network Management:** Faulty devices can be removed from the network without bringing the network down.
- **Product availability:** Many hardware and software tools for network operation and monitoring are available.
- **Cost:** Twisted pair cabling is inexpensive and easily available. Therefore, the installation cost is very low.
- **Reliable:** It is a more reliable network because the communication system is not dependent on the single host computer.

Disadvantages of Ring topology:

- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Failure:** The breakdown in one station leads to the failure of the overall network.
- **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
- **Delay:** Communication delay is directly proportional to the number of nodes. Adding new devices increases the communication delay.

Star Topology

- Star topology is an arrangement of the network in which every node is connected to the central hub, switch or a central computer.
- The central computer is known as a **server**, and the peripheral devices attached to the server are known as **clients**.
- Coaxial cable or RJ-45 cables are used to connect the computers.
- Hubs or Switches are mainly used as connection devices in a **physical star topology**.
- Star topology is the most popular topology in network implementation.

Advantages of Star topology

- **Efficient troubleshooting:** Troubleshooting is quite efficient in a star topology as compared to bus topology. In a bus topology, the manager has to inspect the kilometers of cable. In a star topology, all the stations are connected to the centralized network. Therefore, the network administrator has to go to the single station to troubleshoot the problem.
- **Network control:** Complex network control features can be easily implemented in the star topology. Any changes made in the star topology are automatically accommodated.
- **Limited failure:** As each station is connected to the central hub with its own cable, therefore failure in one cable will not affect the entire network.
- **Familiar technology:** Star topology is a familiar technology as its tools are cost-effective.
- **Easily expandable:** It is easily expandable as new stations can be added to the open ports on the hub.
- **Cost effective:** Star topology networks are cost-effective as it uses inexpensive coaxial cable.
- **High data speeds:** It supports a bandwidth of approx 100Mbps. Ethernet 100BaseT is one of the most popular Star topology networks.

Disadvantages of Star topology

- **A Central point of failure:** If the central hub or switch goes down, then all the connected nodes will not be able to communicate with each other.
- **Cable:** Sometimes cable routing becomes difficult when a significant amount of routing is required.

Tree Topology

- Tree topology combines the characteristics of bus topology and star topology.
- A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
- The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
- There is only one path exists between two nodes for the data transmission. Thus, it forms a parent-child hierarchy.

Advantages of Tree topology

- **Support for broadband transmission:** Tree topology is mainly used to provide broadband transmission, i.e., signals are sent over long distances without being attenuated.
- **Easily expandable:** We can add the new device to the existing network. Therefore, we can say that tree topology is easily expandable.
- **Easily manageable:** In tree topology, the whole network is divided into segments known as star networks which can be easily managed and maintained.
- **Error detection:** Error detection and error correction are very easy in a tree topology.
- **Limited failure:** The breakdown in one station does not affect the entire network.
- **Point-to-point wiring:** It has point-to-point wiring for individual segments.

Disadvantages of Tree topology

- **Difficult troubleshooting:** If any fault occurs in the node, then it becomes difficult to troubleshoot the problem.
- **High cost:** Devices required for broadband transmission are very costly.
- **Failure:** A tree topology mainly relies on main bus cable and failure in main bus cable will damage the overall network.
- **Reconfiguration difficult:** If new devices are added, then it becomes difficult to reconfigure.

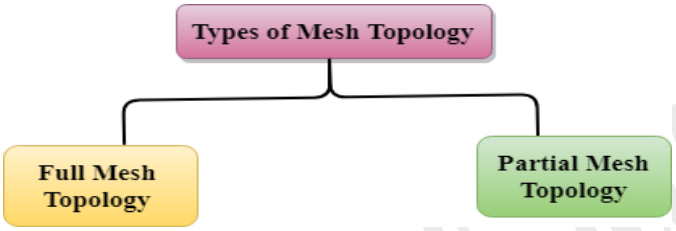
Mesh Topology

- Mesh technology is an arrangement of the network in which computers are interconnected with each other through various redundant connections.
- There are multiple paths from one computer to another computer.
- It does not contain the switch, hub or any central computer which acts as a central point of communication.
- The Internet is an example of the mesh topology.
- Mesh topology is mainly used for WAN implementations where communication failures are a critical concern.
- Mesh topology is mainly used for wireless networks.
- Mesh topology can be formed by using the formula:
  Number of cables = (n*(n-1))/2;

Where n is the number of nodes that represents the network.

Mesh topology is divided into two categories:

- Fully connected mesh topology
- Partially connected mesh topology



- **Full Mesh Topology:** In a full mesh topology, each computer is connected to all the computers available in the network.
- **Partial Mesh Topology:** In a partial mesh topology, not all but certain computers are connected to those computers with which they communicate frequently.

Advantages of Mesh topology:

Reliable: The mesh topology networks are very reliable as if any link breakdown will not affect the communication between connected computers.

Fast Communication: Communication is very fast between the nodes.

Easier Reconfiguration: Adding new devices would not disrupt the communication between other devices.

Disadvantages of Mesh topology

- **Cost:** A mesh topology contains a large number of connected devices such as a router and more transmission media than other topologies.
- **Management:** Mesh topology networks are very large and very difficult to maintain and manage. If the network is not monitored carefully, then the communication link failure goes undetected.
- **Efficiency:** In this topology, redundant connections are high that reduces the efficiency of the network.

Hybrid Topology

- The combination of various different topologies is known as **Hybrid topology**.
- A Hybrid topology is a connection between different links and nodes to transfer the data.
- When two or more different topologies are combined together is termed as Hybrid topology and if similar topologies are connected with each other will not result in Hybrid topology. For example, if there exist a ring topology in one branch of SCB bank and bus topology in another branch of SCB bank, connecting these two topologies will result in Hybrid topology.

Advantages of Hybrid Topology

- **Reliable:** If a fault occurs in any part of the network will not affect the functioning of the rest of the network.
- **Scalable:** Size of the network can be easily expanded by adding new devices without affecting the functionality of the existing network.
- **Flexible:** This topology is very flexible as it can be designed according to the requirements of the organization.

- **Effective:** Hybrid topology is very effective as it can be designed in such a way that the strength of the network is maximized and weakness of the network is minimized.

Disadvantages of Hybrid topology

- **Complex design:** The major drawback of the Hybrid topology is the design of the Hybrid network. It is very difficult to design the architecture of the Hybrid network.
- **Costly Hub:** The Hubs used in the Hybrid topology are very expensive as these hubs are different from usual Hubs used in other topologies.
- **Costly infrastructure:** The infrastructure cost is very high as a hybrid network requires a lot of cabling, network devices, etc.

2.3.2→ Network Protocols

At the human level, some communication rules are formal and others are simply understood, or implicit, based on custom and practice. For devices to successfully communicate, a network protocol suite must describe precise requirements and interactions.

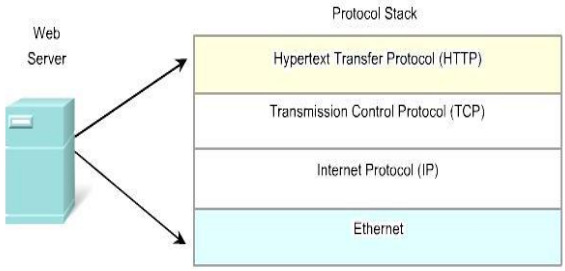
Networking protocol suites describe processes such as:

- The format or structure of the message
- The method by which networking devices share information about pathways with other networks
- How and when error and system messages are passed between devices
- The setup and termination of data transfer sessions

Individual protocols in a protocol suite may be vendor-specific and proprietary. Proprietary, in this context, means that one company or vendor controls the definition of the protocol and how it functions. Some proprietary protocols can be used by different organizations with permission from the owner. Others can only be implemented on equipment manufactured by the proprietary vendor.

2.3.4 → The 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:



Application Protocol:

Hypertext Transfer Protocol (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.

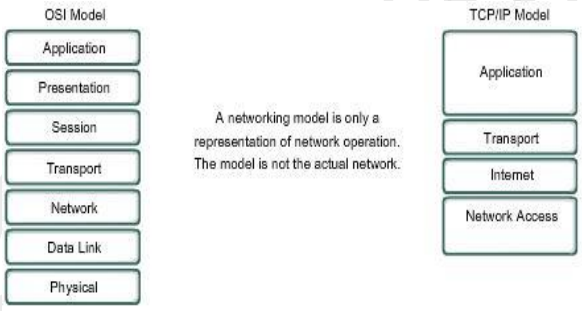
Network Access Protocols:

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.

2.4.2→ Protocol and Reference Models

There are two basic types of networking models: protocol models and reference models.

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.



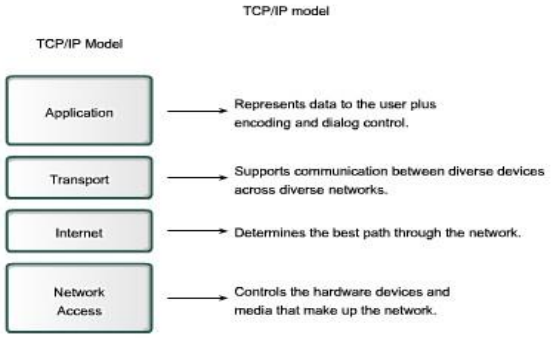
A reference model provides a common reference for maintaining consistency within all types of network protocols and services. A reference model is not intended to be an implementation specification or to provide a sufficient level of detail to define precisely the services of the network architecture. The primary purpose of a reference model is to aid in clearer understanding of the functions and process involved.

The Open Systems Interconnection (OSI) model is the most widely known internetwork reference model. It is used for data network design, operation specifications, and troubleshooting.

Although the TCP/IP and OSI models are the primary models used when discussing network functionality, designers of network protocols, services, or devices can create their own models to represent their products. Ultimately, designers are required to communicate to the industry by relating their product or service to either the OSI model or the TCP/IP model, or to both.

2.4.3→ The TCP/IP Model

The first layered protocol model for internetwork communications was created in the early 1970s and is referred to as the Internet model. It defines four categories of functions that must occur for communications to be successful. The architecture of the TCP/IP protocol suite follows the structure of this model. Because of this, the Internet model is commonly referred to as the TCP/IP model.

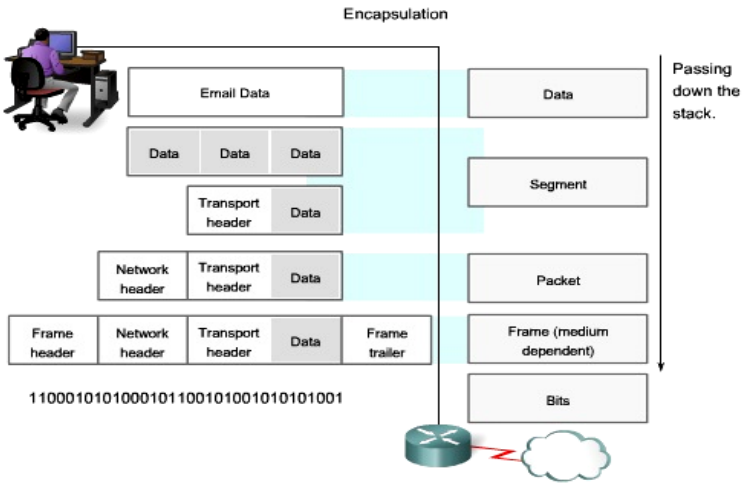


2.4.5→ Protocol Data Units and Encapsulation:

As application data is passed down the protocol stack on its way to be transmitted across the network media, various protocols add information to it at each level. This is commonly known as the encapsulation process.

The form that a piece of data takes at any layer is called a Protocol Data Unit (PDU). During encapsulation, each succeeding layer encapsulates the PDU that it receives from the layer above in accordance with the protocol being used. At each stage of the process, a PDU has a different name to reflect its new appearance. Although there is no universal naming convention for PDUs, in this course, the PDUs are named according to the protocols of the TCP/IP suite.

- Data - The general term for the PDU used at the Application layer
- Segment - Transport Layer PDU
- Packet - Internetwork Layer PDU
- Frame - Network Access Layer PDU
- Bits - A PDU used when physically transmitting data over the medium



2.4.7→ The OSI Model

Initially the OSI model was designed by the International Organization for Standardization (ISO) to provide a framework on which to build a suite of open systems protocols. The vision was that this set of protocols would be used to develop an international network that would not be dependent on proprietary systems.

Unfortunately, the speed at which the TCP/IP based Internet was adopted, and the rate at which it expanded, caused the OSI Protocol Suite development and acceptance to lag behind. Although few of the protocols developed using the OSI specifications are in widespread use today, the seven-layer OSI model has made major contributions to the development of other protocols and products for all types of new networks.

As a reference model, the OSI model provides an extensive list of functions and services that can occur at each layer. It also describes the interaction of each layer with the layers directly above and below it. Although the content of this course will be structured around the OSI Model the focus of discussion will be the protocols identified in the TCP/IP protocol stack.

Note that whereas the TCP/IP model layers are referred to only by name, the seven OSI model layers are more often referred to by number than by name.

3.1.1→ OSI and TCP/ IP Model

The Open Systems Interconnection reference model is a layered, abstract representation created as a guideline for network protocol design. The OSI model divides the networking process into seven logical layers, each of which has unique functionality and to which are assigned specific services and protocols.

In this model, information is passed from one layer to the next, starting at the Application layer on the transmitting host, proceeding down the hierarchy to the Physical layer, then passing over the communications channel to the destination host, where the information proceeds back up the hierarchy, ending at the Application layer. The figure depicts the steps in this process.

Figure: See **< 2.4.7 >**

The Application layer, Layer seven, is the top layer of both the OSI and TCP/IP models. It is the layer that provides the interface between the applications we use to communicate and the underlying network over which our messages are transmitted. Application layer protocols are used to exchange data between programs running on the source and destination hosts. There are many Application layer protocols and new protocols are always being developed.

Although the TCP/IP protocol suite was developed prior to the definition of the OSI model, the functionality of the TCP/IP application layer protocols fit roughly into the framework of the top three layers of the OSI model: Application, Presentation and Session layers.

Most TCP/IP application layer protocols were developed before the emergence of personal computers, graphical user interfaces and multimedia objects. As a result, these protocols implement very little of the functionality that is specified in the OSI model Presentation and Session layers.

Figure: See **<2.4.8>**

The 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 the decryption of data upon receipt by the destination.

Presentation 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.

The Session Layer

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

Most applications, like web browsers or e-mail clients, incorporate functionality of the OSI layers 5, 6 and 7.

The most widely-known TCP/IP Application layer protocols are those that provide for 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:

- Domain Name Service Protocol (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 (RFCs). The Internet Engineering Task Force maintains the RFCs as the standards for the TCP/IP suite.

3.1.2→ Application Layer Software

The functions associated with the Application layer protocols enable our human network to interface with the underlying data network. When we open a web browser or an instant message window, an application is started, and the program is put into the device's memory where it is executed. Each executing program loaded on a device is referred to as a process.

Within the Application layer, there are two forms of software programs or processes that provide access to the network: applications and services.

Network-Aware Applications

Applications are the software programs used by people to communicate over the network. Some end-user applications are network-aware, meaning that they implement the application layer protocols and are able to communicate directly with the lower layers of the protocol stack. E-mail clients and web browsers are examples of these types of applications.

Application layer Services

Other programs may need the assistance of Application layer services to use network resources, like file transfer or network print spooling. Though transparent to the user, these services are the programs that interface with the network and prepare the data for transfer. Different types of data - whether it is text, graphics, or video - require different network services to ensure that it is properly prepared for processing by the functions occurring at the lower layers of OSI model.

Each application or network service uses protocols which define the standards and data formats to be used. Without protocols, the data network would not have a common way to format and direct data. In order to understand the function of various network services, it is necessary to become familiar with the underlying protocols that govern their operation.

3.1.4→ Application Layer Protocol Functions

Application layer protocols are used by both the source and destination devices during a communication session. In order for the communications to be successful, the application layer protocols implemented on the source and destination host must match.

Protocols establish consistent rules for exchanging data between applications and services loaded on the participating devices. Protocols specify how data inside the messages is structured and the types of messages that are sent between source and destination. These messages can be requests for services, acknowledgments, data messages, status messages, or error messages. Protocols also define message dialogues, ensuring that a message being sent is met by the expected response and the correct services are invoked when data transfer occurs.

Many different types of applications communicate across data networks. Therefore, Application layer services must implement multiple protocols to provide the desired range of communication experiences. Each protocol has a specific purpose and contains the characteristics required to meet that purpose. The right protocol details in each layer must be followed so that the functions at one layer interface properly with the services in the lower layer.

3.2.1→ The Client-Server Model

The 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 format of the requests and responses between clients and servers. In addition to the actual data transfer, this exchange may also 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 may be equal in both directions, or may even be greater in the direction going from the client to the server. For example, a client may 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 as a download.

3.2.2→ 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 may have different requirements for client access. Some servers may require authentication of user account information to verify if 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 may have permission to write to your individual folder but not to read other files on the site.

In a client/server network, the server runs a service, or process, sometimes called a server daemon. Like most services, daemons typically run in the background and are not under an end user's direct control. Daemons are described as "listening" for a request from a client, because they are programmed to respond whenever the server receives a request for the service provided by the daemon. When a daemon "hears" a request from a client, it exchanges appropriate messages with the client, as required by its protocol, and proceeds to send the requested data to the client in the proper format.

3.2.4→ Peer-to-Peer Networking and Applications (P2P)

The Peer-to-Peer Model

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

Peer-to-Peer Networks

In a peer-to-peer network, two or more computers are connected via 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 serving as a client for another. The roles of client and server are set on a per request basis.

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.

Peer-to-Peer Applications

A peer-to-peer application (P2P), unlike a peer-to-peer network, allows a device to act as both a client and a server within the same communication. In this model, every client is a server and every server a client. Both can initiate a communication and are considered equal in the communication process. However, peer-to-peer applications require that each end device provide a user interface and run a background service. When you launch a specific peer-to-peer application it invokes the required user interface and background services. After that the devices can communicate directly.

Some P2P applications use a hybrid system where resource sharing is decentralized but the indexes that point to resource locations are stored in a centralized directory. In a hybrid system, each peer accesses an index server to get the location of a resource stored on another peer. The index server can also help connect two peers, but once connected, the communication takes place between the two peers without additional communication to the index server.

Peer-to-peer applications can be used on peer-to-peer networks, client/server networks, and across the Internet.

3.3.1→ DNS Services and Protocol

Now that we have a better understanding of how applications provide an interface for the user and provide access to the network, we will take a look at some specific commonly used protocols.

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 we examine the different TCP/IP Application layer protocols and services, we 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
- Hypertext Transfer Protocol (HTTP) - TCP Port 80
- Simple Mail Transfer Protocol (SMTP) - TCP Port 25
- Post Office Protocol (POP) - UDP Port 110
- Telnet - TCP Port 23
- Dynamic Host Configuration Protocol - UDP Port 67
- File Transfer Protocol (FTP) - TCP Ports 20 and 21

The Domain Name System (DNS) was created for domain name to address resolution for these networks. DNS uses a distributed set of servers to resolve the names associated with these numbered addresses.

The DNS protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data formats

DNS is a client/server service; however, it differs from the other client/server services that we are examining. While other services use a client that is an application (such as web browser, e-mail client), the DNS client runs as a service itself. The DNS client, sometimes called the DNS resolver, supports name resolution for our other network applications and other services that need it.

DNS is a client/server service; however, it differs from the other client/server services that we are examining. While other services use a client that is an application (such as web browser, e-mail client), the DNS client runs as a service itself. The DNS client, sometimes called the DNS resolver, supports name resolution for our other network applications and other services that need it.

When configuring a network device, we generally provide one or more DNS Server addresses that the DNS client can use for name resolution. Usually the Internet service provider provides the addresses to use for the DNS servers. When a user's application requests to connect to a remote device by name, the requesting DNS client queries one of these name servers to resolve the name to a numeric address.

3.3.2→ 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 the HTTP protocol. URLs (or Uniform Resource Locator) and URIs (Uniform Resource Identifier) are the names most people associate with web addresses.

Web browsers are the client applications our 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

The Hypertext Transfer Protocol (HTTP), one of the protocols in the TCP/IP suite, was originally developed to publish and retrieve HTML pages and is now used for distributed, collaborative information systems. HTTP is used across the World Wide Web for data transfer and is one of the most used application protocols.

HTTP specifies a request/response protocol. When a client, typically a web browser, sends a request message to a server, the HTTP protocol defines the message types the client uses to request the web page and also the message types the server uses to respond. The three common message types are GET, POST, and PUT.

GET is a client request for data. A web browser sends the GET message to request pages from a web server.

POST and **PUT** are used to send messages that upload data to the web server. For example, when the user enters data into a form embedded in a web page, POST includes the data in the message sent to the server.

PUT uploads resources or content to the web server.

3.3.3→ 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 example Application layer protocols are Post Office Protocol (POP) and Simple **Mail Transfer Protocol (SMTP),** shown in the figure. As with HTTP, these protocols define client/server processes.

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's mailbox, both of which are distinct processes.

In order 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.

E-mail Server Processes - MTA and MDA

The e-mail server operates two separate processes:

- Mail Transfer Agent (MTA)
- Mail Delivery Agent (MDA)

As mentioned earlier, e-mail can use the protocols, POP and SMTP (see the figure for an explanation of how they each work). 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). The MDA listens for when a client connects to a server. Once a connection is established, the server can deliver the e-mail to the client.

The **Simple Mail Transfer Protocol (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). 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.

The SMTP protocol message format uses a rigid set of commands and replies. These commands support the procedures used in SMTP, such as session initiation, mail transaction, forwarding mail, verifying mailbox names, expanding mailing lists, and the opening and closing exchanges.

3.3.4→ FTP

The File Transfer Protocol (FTP) is another commonly used Application layer protocol. FTP was developed to allow for 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, 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 there is a file transferred.

The file transfer can happen in either direction. The client can download (pull) a file from the server or, the client can upload (push) a file to the server.

3.3.5→ DHCP

The **Dynamic Host Configuration Protocol (DHCP)** service enables devices on a network to obtain IP addresses and other information from a DHCP server. This service automates the assignment of IP addresses, subnet masks, gateway and other IP networking parameters.

DHCP allows a host to obtain an IP address dynamically when it connects to the network. The DHCP server is contacted and an address requested. The DHCP server chooses an address from a configured range of addresses called a pool and assigns ("leases") it to the host for a set period.

DHCP distributed addresses are not permanently assigned to hosts but are only leased for a period of time. If the host is powered down or taken off the network, the address is returned to the pool for reuse. This is especially helpful with mobile users that come and go on a network. Users can freely move from location to location and re-establish network connections. The host can obtain an IP address once the hardware connection is made, either via a wired or wireless LAN.

DHCP makes it possible for you to access the Internet using wireless hotspots at airports or coffee shops. As you enter the area, your laptop DHCP client contacts the local DHCP server via a wireless connection. The DHCP server assigns an IP address to your laptop.

3.3.6 → File Sharing Services and SMB Protocol

The Server Message Block (SMB) is a client/server file sharing protocol. IBM developed Server Message Block (SMB) in the late 1980s to describe the structure of shared network resources, such as directories, files, printers, and serial ports. It is a request-response protocol. Unlike the file sharing supported by FTP, clients establish a long term connection to servers. Once the connection is established, the user of the client can access the resources on the server as if the resource is local to the client host.

The SMB protocol describes file system access and how clients can make requests for files. It also describes the SMB protocol inter-process communication. All SMB messages share a common format. This format uses a fixed-sized header followed by a variable-sized parameter and data component.

SMB messages can:

- Start, authenticate, and terminate sessions
- Control file and printer access
- Allow an application to send or receive messages to or from another device

3.3.8→ Telnet Services and Protocol

Telnet provides a standard method of emulating text-based terminal devices over the data network. Both the protocol itself and the client software that implements the protocol are commonly referred to as Telnet.

Appropriately enough, a connection using Telnet is called a Virtual Terminal (VTY) session, or connection. Rather than using a physical device to connect to the server, Telnet uses software to create a virtual device that provides the same features of a terminal session with access to the server command line interface (CLI).

To support Telnet client connections, the server runs a service called the Telnet daemon. A virtual terminal connection is established from an end device using a Telnet client application. Most operating systems include an Application layer Telnet client. On a Microsoft Windows PC, Telnet can be run from the command prompt.

Telnet is a client/server protocol and it specifies how a VTY session is established and terminated. It also provides the syntax and order of the commands used to initiate the Telnet session, as well as control commands that can be issued during a session. Each Telnet command consists of at least two bytes. The first byte is a special character called the Interpret as Command (IAC) character. As its name implies, the IAC defines the next byte as a command rather than text.

Some sample Telnet protocol commands include:

Are You There (AYT) - Lets the user request that something appear on the terminal screen to indicate that the VTY session is active.

Erase Line (EL) - Deletes all text from the current line.

Interrupt Process (IP) - Suspends, interrupts, aborts, or terminates the process to which the Virtual Terminal is connected. For example, if a user started a program on the Telnet server via the VTY, he or she could send an IP command to stop the program.

While the Telnet protocol supports user authentication, it does not support the transport of encrypted data. All data exchanged during a Telnet sessions is transported as plain text across the network. This means that the data can be intercepted and easily understood.

If security is a concern, the Secure Shell (SSH) protocol offers an alternate and secure method for server access. SSH provides the structure for secure remote login and other secure network services. It also provides stronger authentication than Telnet and supports the transport of session data using encryption. **As a best practice, network professionals should always use SSH in place of Telnet, whenever possible.**

4.1.1→ Purpose of the Transport Layer

The Transport layer provides for the segmentation of data and the control necessary to reassemble these pieces into the various communication streams. Its primary responsibilities to accomplish this are:

- Tracking the individual communication 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

Tracking Individual Conversations

Any host may have multiple applications that are communicating across the network. Each of these applications will be communicating with one or more applications on remote hosts. It is the responsibility of the Transport layer to maintain the multiple communication streams between these applications.

Segmenting Data

As each application creates a stream data to be sent to a remote application, this data must be prepared to be sent across the media in manageable pieces. The Transport layer protocols describe services that segment this data from the Application layer. This includes the encapsulation required on each piece of data. Each piece of application data requires headers to be added at the Transport layer to indicate to which communication it is associated.

Reassembling Segments

At the receiving host, each piece of data may be directed to the appropriate application. Additionally, these individual pieces of data must also be reconstructed into a complete data stream that is useful to the Application layer. The protocols at the Transport layer describe the how the Transport layer header information is used to reassemble the data pieces into streams to be passed to the Application layer.

4.1.2→ Controlling the Conversations

The primary functions specified by all Transport layer protocols include:

Segmentation and Reassembly - Most networks have a limitation on the amount of data that can be included in a single PDU. The Transport layer divides application data into blocks of data that are an appropriate size. At the destination, the Transport layer reassembles the data before sending it to the destination application or service.

Conversation Multiplexing - There may be many applications or services running on each host in the network. Each of these applications or services is assigned an address known as a port so that the Transport layer can determine with which application or service the data is identified.

In addition to using the information contained in the headers, for the basic functions of data segmentation and reassembly, some protocols at the Transport layer provide:

- Connection-oriented conversations
- Reliable delivery
- Ordered data reconstruction
- Flow control

Establishing a Session

The Transport layer can provide this connection orientation by creating a sessions between the applications. These connections prepare the applications to communicate with each other before any data is transmitted. Within these sessions, the data for a communication between the two applications can be closely managed.

Reliable Delivery

For many reasons, it is possible for a piece of data to become corrupted, or lost completely, as it is transmitted over the network. The Transport layer can ensure that all pieces reach their destination by having the source device to retransmit any data that is lost.

Same Order Delivery

Because networks may provide multiple routes that can have different transmission times, data can arrive in the wrong order. By numbering and sequencing the segments, the Transport layer can ensure that these segments are reassembled into the proper order.

Flow Control

Network hosts have limited resources, such as memory or bandwidth. When Transport layer is aware that these resources are overtaxed, some protocols can request that the sending application reduce the rate of data flow. This is done at the Transport layer by regulating the amount of data the source transmits as a group. Flow control can prevent the loss of segments on the network and avoid the need for retransmission.

4.1.3→ Supporting Reliable Communication

Recall that the primary function of the Transport layer is to manage the application data for the conversations between hosts. However, different applications have different requirements for their data, and therefore different Transport protocols have been developed to meet these requirements.

A Transport layer protocol can implement a method to ensure reliable delivery of the data. In networking terms, reliability means ensuring that each piece of data that the source sends arrives at the destination. At the Transport layer the three basic operations of reliability are:

- tracking transmitted data
- acknowledging received data
- retransmitting any unacknowledged data

This requires the processes of Transport layer of the source to keep track of all the data pieces of each conversation and the retransmit any of data that did were not acknowledged by the destination. The Transport layer of the receiving host must also track the data as it is received and acknowledge the receipt of the data.

These reliability processes place additional overhead on the network resources due to the acknowledgement, tracking, and retransmission. To support these reliability operations, more control data is exchanged between the sending and receiving hosts. This control information is contained in the Layer 4 header.

This creates a trade-off between the value of reliability and the burden it places on the network. Application developers must choose which transport protocol type is appropriate based on the requirements of their applications. At the Transport layer, there are protocols that 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 there is no acknowledgement that the data is received at the destination.

4.1.4→ TCP and UDP

The two most common Transport layer protocols of 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 for low overhead data delivery. The pieces of communication in UDP are called datagrams. These datagrams are sent as "best effort" by this Transport layer protocol.

Applications that use UDP include:

Domain Name System (DNS)

Video Streaming

Voice over IP (VoIP)

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 the 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 only has 8 bytes of overhead. See the figure for a comparison.

Applications that use TCP are:

Web Browsers

E-mail

File Transfers

TCP and UDP Headers

TCP Segment

Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Sequence Number (32)		
Acknowledgement Number (32)		
Header Length (4) Reserved (6) Code Bits (6)	Window (16)	
Checksum (16)	Urgent (16)	
Options (0 or 32 if any)		
APPLICATION LAYER DATA (Size varies)		

20 Bytes

UDP Datagram

Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Length (16)	Checksum (16)	
APPLICATION LAYER DATA (Size varies)		

8 Bytes

4.1.5→ Port Addressing

Identifying the Conversations

Consider the earlier example of a computer simultaneously receiving and sending e-mail, instant messages, web pages, and a VoIP phone call.

The TCP and UDP based services keep track of the various applications that are communicating. To differentiate the segments and datagrams for each application, both TCP and UDP have header fields that can uniquely identify these applications. These unique identifiers are the port numbers.

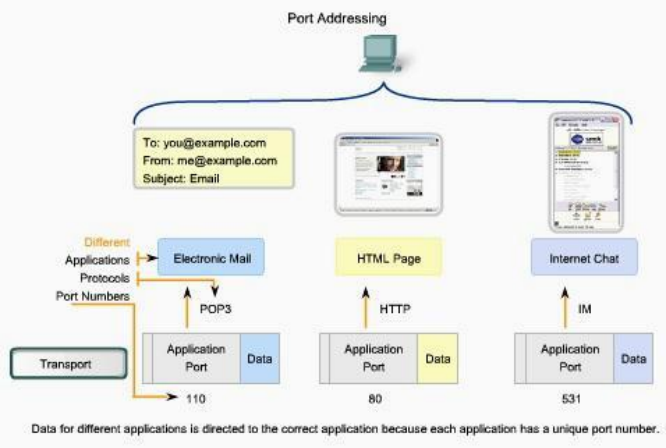
In the header of each segment or datagram, there is a source and destination port. The source port number is the number for this communication associated with the originating application on the local host. The destination port number is the number for this communication associated with the destination application on the remote host.

Port numbers are assigned in various ways, depending on whether the message is a request or a response. While server processes have static port numbers assigned to them, clients dynamically choose a port number for each conversation.

When a client application sends a request to a server application, the destination port contained in the header is the port number that is assigned to the service daemon running on the remote host. The client software must know what port number is associated with the server process on the remote host. This destination port number is configured, either by default or manually. For example, when a web browser application makes a request to a web server, the browser uses TCP and port number 80 unless otherwise specified. This is because TCP port 80 is the default port assigned to web-serving applications. Many common applications have default port assignments.

The source port in a segment or datagram header of a client request is randomly generated. As long as it does not conflict with other ports in use on the system, the client can choose any port number. This port number acts like a return address for the requesting application. The Transport layer keeps track of this port and the application that initiated the request so that when a response is returned, it can be forwarded to the correct application. The requesting application port number is used as the destination port number in the response coming back from the server.

The combination of the Transport layer port number and the Network layer IP address assigned to the host uniquely identifies a particular process running on a specific host device. This combination is called a socket. Occasionally, you may find the terms port number and socket used interchangeably. In the context of this course, the term socket refers only to the unique combination of IP address and port number. A socket pair, consisting of the source and destination IP addresses and port numbers, is also unique and identifies the conversation between the two hosts.



For example, an HTTP web page request being sent to a web server (port 80) running on a host with a Layer 3 IPv4 address of 192.168.1.20 would be destined to socket 192.168.1.20:80.

If the web browser requesting the web page is running on host 192.168.100.48 and the Dynamic port number assigned to the web browser is 49152, the socket for the web page would be 192.168.100.48:49152.

The **Internet Assigned Numbers Authority (IANA)** assigns port numbers. IANA is a standards body that is responsible for assigning various addressing standards.

There are different types of port numbers:

Well Known Ports (Numbers 0 to 1023) - These numbers are reserved for services and applications. They are commonly used for applications such as HTTP (web server) POP3/SMTP (e-mail server) and Telnet. By defining these well-known ports for server applications, client applications can be programmed to request a connection to that specific port and its associated service.

Registered Ports (Numbers 1024 to 49151) - These port numbers are assigned to user processes or applications. These processes are primarily individual applications that a user has chosen to install rather than common applications that would receive a Well Known Port. When not used for a server resource, these ports may also be used dynamically selected by a client as its source port.

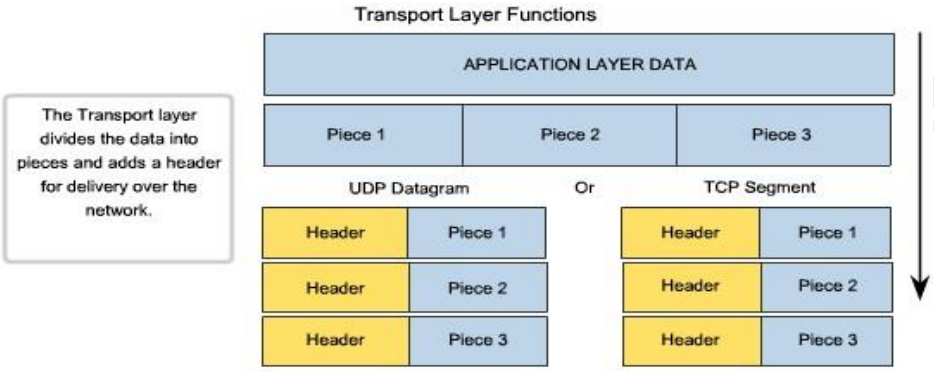
Dynamic or Private Ports (Numbers 49152 to 65535) - Also known as Ephemeral Ports, these are usually assigned dynamically to client applications when initiating a connection. It is not very common for a client to connect to a service using a Dynamic or Private Port (although some peer-to-peer file sharing programs do).

Using both TCP and UDP

Some applications may use both TCP and UDP. For example, the low overhead of UDP enables DNS to serve many client requests very quickly. Sometimes, however, sending the requested information may require the reliability of TCP. In this case, the well-known port number of 53 is used by both protocols with this service.

4.1.6→ Segmentation and Reassembly – Divide and Conquer

Dividing application data into pieces both ensures that data is transmitted within the limits of the media and that data from different applications can be multiplexed on to the media.



TCP and UDP Handle Segmentation Differently.

In TCP, each segment header contains a sequence number. This sequence number allows the Transport layer functions on the destination host to reassemble segments in the order in which they were transmitted. This ensures that the destination application has the data in the exact form the sender intended.

Although services using UDP also track the conversations between applications, they are not concerned with the order in which the information was transmitted, or in maintaining a connection. There is no sequence number in the UDP header. UDP is a simpler design and generates less overhead than TCP, resulting in a faster transfer of data.

4.2.1→ TCP – Making Conversations Reliable

The key distinction between TCP and UDP is reliability. The reliability of TCP communication is performed using connection-oriented sessions. Before a host using TCP sends data to another host, the Transport layer initiates a process to create a connection with the destination. This connection enables the tracking of a session, or communication stream between the hosts. This process ensures that each host is aware of and prepared for the communication. A complete TCP conversation requires the establishment of a session between the hosts in both directions.

After a session has been established, the destination sends acknowledgements to the source for the segments that it receives. These acknowledgements form the basis of reliability within the TCP session. As the source receives an acknowledgement, it knows that the data has been successfully delivered and can quit tracking that data. If the source does not receive an acknowledgement within a predetermined amount of time, it retransmits that data to the destination.

Part of the additional overhead of using TCP is the network traffic generated by acknowledgements and retransmissions. The establishment of the sessions creates overhead in the form of additional segments being

exchanged. There is also additional overhead on the individual hosts created by the necessity to keep track of which segments are awaiting acknowledgement and by the retransmission process.

TCP Segment Header Fields

The fields of the TCP header enable TCP to provide connection-oriented, reliable data communications.

4.2.3→ TCP Connection Establishment and Termination

When two hosts communicate using TCP, a connection is established before data can be exchanged. After the communication is completed, the sessions are closed and the connection is terminated. The connection and session mechanisms enable TCP's reliability function.

Each connection represents two one-way communication streams, or sessions. To establish the connection, the hosts perform a three-way handshake. Control bits in the TCP header indicate the progress and status of the connection. The three-way handshake:

- Establishes that the destination device is present on the network
- Verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use for the session
- Informs the destination device that the source client intends to establish a communication session on that port number

In TCP connections, the host serving as a client initiates the session to the server. The three steps in TCP connection establishment are:

1. The initiating client sends a segment containing an initial sequence value, which serves as a request to the server to begin a communications session.

2. The server responds with a segment containing an acknowledgement value equal to the received sequence value plus 1, plus its own synchronizing sequence value. The value is one greater than the sequence number because the ACK is always the next expected Byte or Octet. This acknowledgement value enables the client to tie the response back to the original segment that it sent to the server.

3. Initiating client responds with an acknowledgement value equal to the sequence value it received plus one. This completes the process of establishing the connection.

To understand the three-way handshake process, it is important to look at the various values that the two hosts exchange. Within the TCP segment header, there are six 1-bit fields that contain control information used to manage the TCP processes. Those fields are:

URG - Urgent pointer field significant

ACK - Acknowledgement field significant

PSH - Push function

RST - Reset the connection

SYN - Synchronize sequence numbers

FIN - No more data from sender

These fields are referred to as flags, because the value of one of these fields is only 1 bit and, therefore, has only two values: 1 or 0. When a bit value is set to 1, it indicates what control information is contained in the segment.

Using a four-step process, flags are exchanged to terminate a TCP connection.

4.2.4→ TCP Three-Way Handshake

Step 1

A TCP client begins the three-way handshake by sending a segment with the SYN (Synchronize Sequence Number) control flag set, indicating an initial value in the sequence number field in the header. This initial value for the sequence number, known as the Initial Sequence Number (ISN), is randomly chosen and is used to begin tracking the flow of data from the client to the server for this session. The ISN in the header of each segment is increased by one for each byte of data sent from the client to the server as the data conversation continues.

The SYN control flag is set and the relative sequence number is at 0. Although the protocol analyzer in the graphic indicates the relative values for the sequence and acknowledgement numbers, the true values are 32 bit binary numbers. We can determine the actual numbers sent in the segment headers by examining the Packet Bytes pane. Here you can see the four bytes represented in hexadecimal.

Step 2

The TCP server needs to acknowledge the receipt of the SYN segment from the client to establish the session from the client to the server. To do so, the server sends a segment back to the client with the ACK flag set indicating that the Acknowledgment number is significant. With this flag set in the segment, the client recognizes this as an acknowledgement that the server received the SYN from the TCP client.

The value of the acknowledgment number field is equal to the client initial sequence number plus 1. This establishes a session from the client to the server. The ACK flag will remain set for the balance of the session. Recall that the conversation between the client and the server is actually two one-way sessions: one from the client to the server, and the other from the server to the client. In this second step of the three-way handshake, the server must initiate the response from the server to the client. To start this session, the server uses the SYN flag in the same way that the client did. It sets the SYN control flag in the header to establish a session from the server to the client. The SYN flag indicates that the initial value of the sequence number field is in the header. This value will be used to track the flow of data in this session from the server back to the client.

Step 3

Finally, the TCP client responds with a segment containing an ACK that is the response to the TCP SYN sent by the server. There is no user data in this segment. The value in the acknowledgment number field contains one more than the initial sequence number received from the server. Once both sessions are established between client and server, all additional segments exchanged in this communication will have the ACK flag set.

Security can be added to the data network by:

- Denying the establishment of TCP sessions
- Only allowing sessions to be established for specific services
- Only allowing traffic as a part of already established sessions

This security can be implemented for all TCP sessions or only for selected sessions.

4.2.5→ TCP Session Termination

To close a connection, the FIN (Finish) control flag in the segment header must be set. To end each one-way TCP session, a two-way handshake is used, consisting of a FIN segment and an ACK segment. Therefore, to terminate a single conversation supported by TCP, four exchanges are needed to end both sessions. Note: In this explanation, the terms client and server are used in this description as a reference for simplicity, but the termination process can be initiated by any two hosts that complete the session:

1. When the client has no more data to send in the stream, it sends a segment with the FIN flag set.

2. The server sends an ACK to acknowledge the receipt of the FIN to terminate the session from client to server.

3. The server sends a FIN to the client, to terminate the server to client session.

4. The client responds with an ACK to acknowledge the FIN from the server.

When the client end of the session has no more data to transfer, it sets the FIN flag in the header of a segment. Next, the server end of the connection will send a normal segment containing data with the ACK flag set using the acknowledgment number, confirming that all the bytes of data have been received. When all segments have been acknowledged, the session is closed.

The session in the other direction is closed using the same process. The receiver indicates that there is no more data to send by setting the FIN flag in the header of a segment sent to the source. A return acknowledgement confirms that all bytes of data have been received and that session is, in turn, closed.

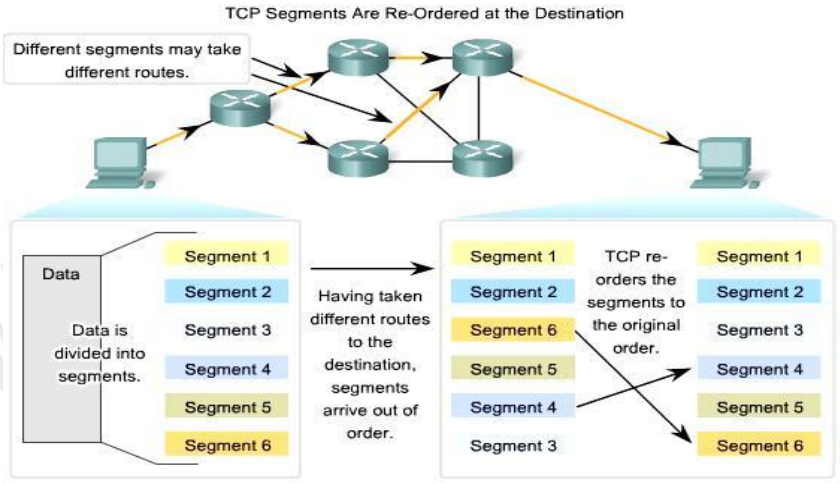
It is also possible to terminate the connection by a three-way handshake. When the client has no more data to send, it sends a FIN to the server. If the server also has no more data to send, it can reply with both the FIN and ACK flags set, combining two steps into one. The client replies with an ACK.

4.3.1→ TCP Segment Reassembly

Resequencing Segments to Order Transmitted

When services send data using TCP, segments may arrive at their destination out of order. For the original message to be understood by the recipient, the data in these segments is reassembled into the original order. Sequence numbers are assigned in the header of each packet to achieve this goal.

During session setup, an initial sequence number (ISN) is set. This initial sequence number represents the starting value for the bytes for this session that will be transmitted to the receiving application. As data is transmitted during the session, the sequence number is incremented by the number of bytes that have been transmitted. This tracking of data byte enables each segment to be uniquely identified and acknowledged. Missing segments can be identified.



Segment sequence numbers enable reliability by indicating how to reassemble and reorder received segments, as shown in the figure.

The receiving TCP process places the data from a segment into a receiving buffer. Segments are placed in the proper sequence number order and passed to the Application layer when reassembled. Any segments that arrive with noncontiguous sequence numbers are held for later processing. Then, when the segments with the missing bytes arrive, these segments are processed.

4.3.2→ TCP Acknowledgement with Windowing

Confirming Receipt of Segments

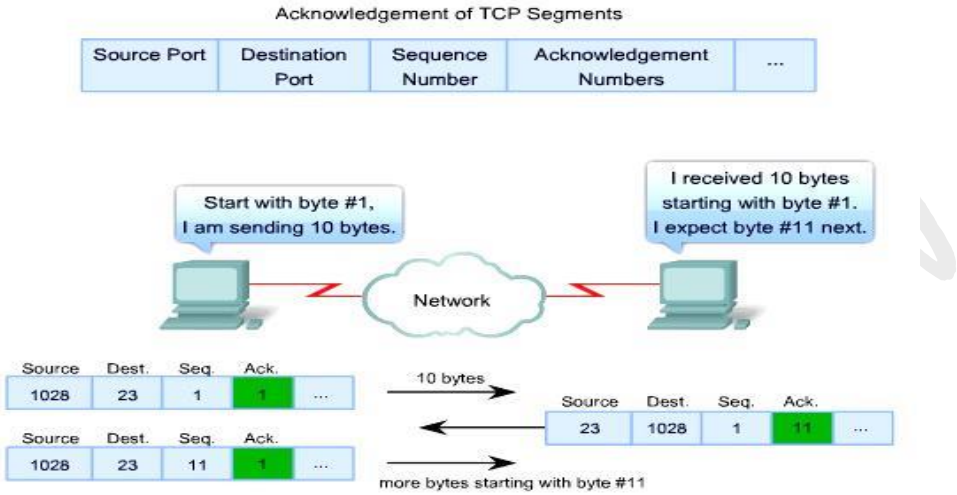
One of TCP's functions is making sure that each segment reaches its destination. The TCP services on the destination host acknowledge the data that it has received to the source application.

The segment header sequence number and acknowledgement number are used together to confirm receipt of the bytes of data contained in the segments. The sequence number is the relative number of bytes that have been

transmitted in this session plus 1 (which is the number of the first data byte in the current segment). TCP uses the acknowledgement number in segments sent back to the source to indicate the next byte in this session that the receiver expects to receive. This is called expectational acknowledgement.

The source is informed that the destination has received all bytes in this data stream up to, but not including, the byte indicated by the acknowledgement number. The sending host is expected to send a segment that uses a sequence number that is equal to the acknowledgement number.

Remember, each connection is actually two one-way sessions. Sequence numbers and acknowledgement numbers are being exchanged in both directions.



In the example in the figure, the host on the left is sending data to the host on the right. It sends a segment containing 10 bytes of data for this session and a sequence number equal to 1 in the header.

The receiving host on the right receives the segment at Layer 4 and determines that the sequence number is 1 and that it has 10 bytes of data. The host then sends a segment back to the host on the left to acknowledge the receipt of this data. In this segment, the host sets the acknowledgement number to 11 to indicate that the next byte of data it expects to receive in this session is byte number 11.

When the sending host on the left receives this acknowledgement, it can now send the next segment containing data for this session starting with byte number 11.

Looking at this example, if the sending host had to wait for acknowledgement of the receipt of each 10 bytes, the network would have a lot of overhead. To reduce the overhead of these acknowledgements, multiple segments of data can be sent before and acknowledged with a single TCP message in the opposite direction. This acknowledgement contains an acknowledgement number based on the total number of bytes received in the session.

For example, starting with a sequence number of 2000, if 10 segments of 1000 bytes each were received, an acknowledgement number of 12001 would be returned to the source.

The amount of data that a source can transmit before an acknowledgement must be received is called the window size. Window Size is a field in the TCP header that enables the management of lost data and flow control.

4.3.3→TCP Retransmission

Handling Segment Loss

No matter how well designed a network is, data loss will occasionally occur. Therefore, TCP provides methods of managing these segment losses. Among these is a mechanism to retransmit segments with unacknowledged data.

A destination host service using TCP usually only acknowledges data for contiguous sequence bytes. If one or more segments are missing, only the data in the segments that complete the stream are acknowledged.

For example, if segments with sequence numbers 1500 to 3000 and 3400 to 3500 were received, the acknowledgement number would be 3001. This is because there are segments with the sequence numbers 3001 to 3399 that have not been received.

When TCP at the source host has not received an acknowledgement after a predetermined amount of time, it will go back to the last acknowledgement number that it received and retransmit data from that point forward.

The retransmission process is not specified by the RFC, but is left up to the particular implementation of TCP.

For a typical TCP implementation, a host may transmit a segment, put a copy of the segment in a retransmission queue, and start a timer. When the data acknowledgment is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer expires, the segment is retransmitted.

4.3.4→ TCP Congestion Control – Minimizing Segment Loss

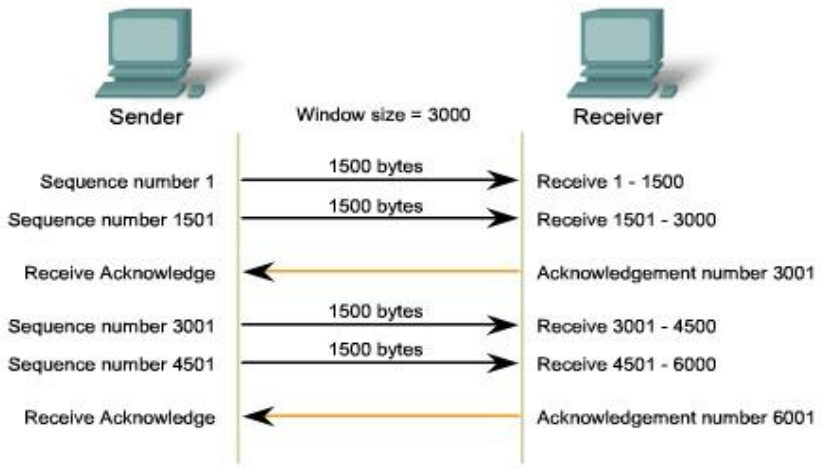
Flow Control

TCP also provides mechanisms for 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.

This Window Size field in the TCP header specifies the amount of data that can be transmitted before an acknowledgement must be received. The initial window size is determined during the session startup via the three-way handshake.

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.

TCP Segment Acknowledgement and Window Size



The window size determines the number of bytes sent before an acknowledgment is expected.
The acknowledgement number is the number of the next expected byte.

See the figure for a simplified representation of window size and acknowledgements. 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 acknowledgement of these bytes before transmitting more segments in this session.

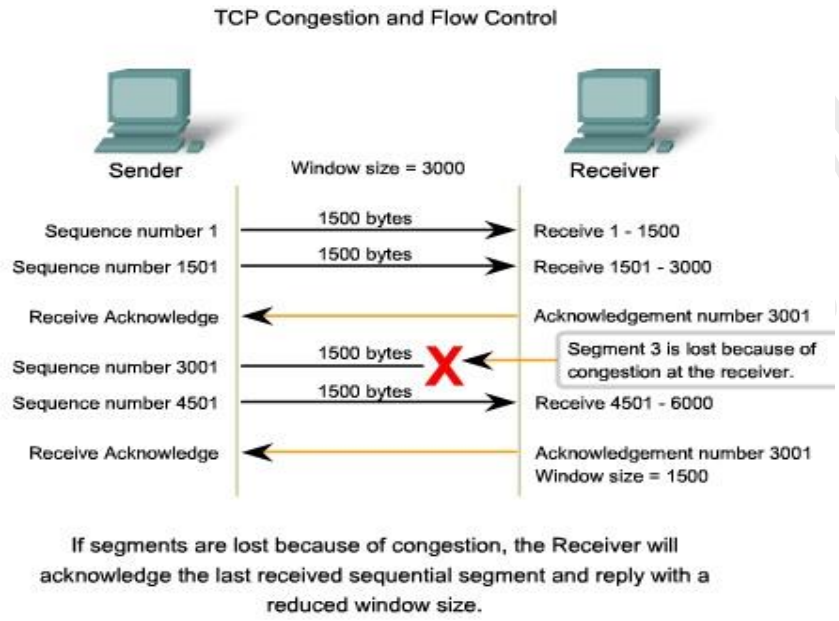
Once the sender has received this acknowledgement from the receiver, the sender can transmit an additional 3000 bytes.

During the delay in receiving the acknowledgement, the sender will not be sending any additional segments for this session. In periods when the network is congested or the resources of the receiving host are strained, the delay may 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.

Reducing Window Size

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 down the rate of transmission because the source waits for data to be acknowledged more frequently.

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 down the rate of communication because of limited buffer memory, it can send a smaller window size value to the source as part of an acknowledgement.



As shown in the figure, if a receiving host has congestion, it may respond to the sending host with a segment with a reduced window size. In this graphic, there was a loss of one of the segments. The receiver changed the window field in the TCP header of the returning segments in this conversation from 3000 down to 1500. This caused the sender to reduce the window size to 1500.

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

4.4.1→ UDP – Low Overhead vs. Reliability

UDP is a simple protocol that provides the basic Transport layer functions. It much lower overhead than TCP, since it is not connection-oriented and does not provide the sophisticated retransmission, sequencing, and flow control mechanisms.

This does not mean that applications that use UDP are always unreliable. It simply means that these functions are not provided by the Transport layer protocol and must be implemented elsewhere if required.

Although the total amount of UDP traffic found on a typical network is often relatively low, key Application layer protocols that use UDP include:

- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)

- Dynamic Host Configuration Protocol (DHCP)
- Routing Information Protocol (RIP)
- Trivial File Transfer Protocol (TFTP)
- Online games

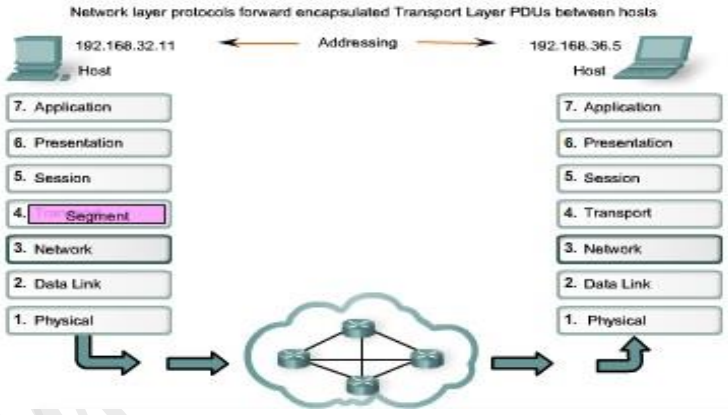
Some applications, such as online games or VoIP, can tolerate some loss of some data. If these applications used TCP, they may experience large delays while TCP detects data loss and retransmits data. These delays would be more detrimental to the application than small data losses. Some applications, such as DNS, will simply retry the request if they do not receive a response, and therefore they do not need TCP to guarantee the message delivery.

The low overhead of UDP makes it very desirable for such applications.

5.1.1→ Network Layer – Communication from Host to Host

The Network layer, or 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 four basic processes:

- Addressing
- Encapsulation
- Routing
- Decapsulation



Addressing

First, the Network layer must provide a mechanism for addressing these end devices. If individual pieces of data are to be directed to an end device, that device must have a unique address. In an IPv4 network, when this address is added to a device, the device is then referred to as a host.

Encapsulation

Second, the Network layer must provide encapsulation. Not only must the devices be identified with an address, the individual pieces - the Network layer PDUs - must also contain these addresses. During the encapsulation process, Layer 3 receives the Layer 4 PDU and adds a Layer 3 header, or label, to create the Layer 3 PDU. When referring to the Network layer, we call this PDU a packet. When a packet is created, the header must contain, among other information, the address of the host to which it is being sent. This address is referred to as the destination address. The Layer 3 header also contains the address of the originating host. This address is called the source address.

After the Network layer completes its encapsulation process, the packet is sent down to the Data Link layer to be prepared for transportation over the media.

Routing

Next, the Network layer must provide services to direct these packets to their destination host. The source and destination hosts are not always connected to the same network. In fact, the packet might have to travel through many different networks. Along the way, each packet must be guided through the network to reach its final

destination. Intermediary devices that connect the networks are called routers. The role of the router is to select paths for and direct packets toward their destination. This process is known as routing.

During the routing through an internetwork, the packet may traverse many intermediary devices. Each route that a packet takes to reach the next device is called a hop. As the packet is forwarded, its contents (the Transport layer PDU), remain intact until the destination host is reached.

Decapsulation

Finally, the packet arrives at the destination host and is processed at Layer 3. The host examines the destination address to verify that the packet was addressed to this device. If the address is correct, the packet is decapsulated by the Network layer and the Layer 4 PDU contained in the packet is passed up to the appropriate service at Transport layer.

Unlike the Transport layer (OSI Layer 4), which manages the data transport between the processes running on each end host, Network layer protocols specify the packet structure and processing used to carry the data from one host to another host. Operating without regard to the application data carried in each packet allows the Network layer to carry packets for multiple types of communications between multiple hosts.

Network Layer Protocols

Protocols implemented at the Network layer that carry user data include:

- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Novell Internetwork Packet Exchange (IPX)
- AppleTalk
- Connectionless Network Service (CLNS/DECNet)

The Internet Protocol (IPv4 and IPv6) is the most widely-used Layer 3 data carrying protocol and will be the focus of this course. Discussion of the other protocols will be minimal.

5.1.2→ The IP v4 Protocol – Example Network Layer Protocol

Role of IPv4

The Network layer services implemented by the TCP/IP protocol suite are the Internet Protocol (IP). 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 we 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 may replace it in the future. The services provided by IP, as well as the packet header structure and contents, are specified by either IPv4 protocol or IPv6 protocol. These services and packet structure are used to encapsulate UDP datagrams or TCP segments for their trip across an internetwork.

The characteristics of each protocol are different. Understanding these characteristics will allow you to understand the operation of the services described by this protocol.

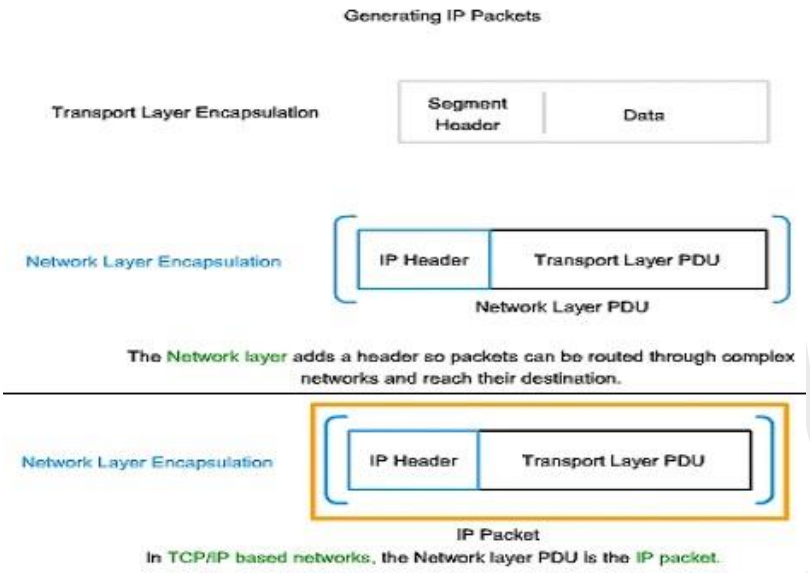
The Internet Protocol was designed as a protocol with low overhead. It provides only the functions that are necessary to deliver a packet from a source to a destination over an interconnected system of networks. The protocol was not designed to track and manage the flow of packets. These functions are performed by other protocols in other layers.

IPv4 basic characteristics:

- Connectionless - No connection is established before sending data packets.
- Best Effort (unreliable) - No overhead is used to guarantee packet delivery.
- Media Independent - Operates independently of the medium carrying the data.

5.1.6→ IP v4 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. Click the steps in the figure to see this process. 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.

5.1.7→ IP v4 Packet Header

As shown in the figure, an IPv4 protocol defines many different fields in the packet header. These fields contain binary values that the IPv4 services reference as they forward packets across the network.

This course will consider these 6 key fields:

- IP Source Address
- IP Destination Address
- Time-to-Live (TTL)
- Type-of-Service (ToS)
- Protocol
- Fragment Offset

Key IPv4 Header Fields

Roll over each field on the graphic to see its purpose.

IP Destination Address

The IP Destination Address field contains a 32-bit binary value that represents the packet destination Network layer host address.

IP Source Address

The IP Source Address field contains a 32-bit binary value that represents the packet source Network layer host address.

Time-to-Live

The Time-to-Live (TTL) is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop). When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow. This mechanism prevents packets that cannot reach their destination from being forwarded indefinitely between

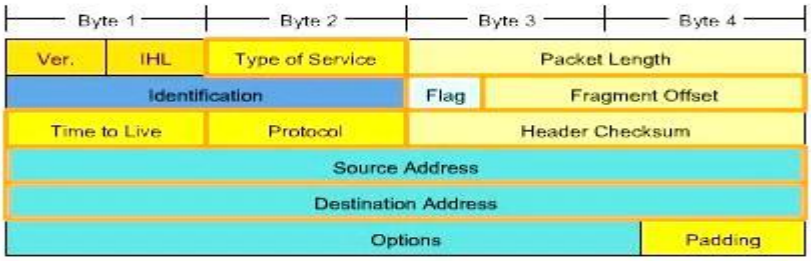
routers in a routing loop. If routing loops were permitted to continue, the network would become congested with data packets that will never reach their destination. Decrementing the TTL value at each hop ensures that it eventually becomes zero and that the packet with the expired TTL field will be dropped.

Protocol

This 8-bit binary value indicates the data payload type that the packet is carrying. The Protocol field enables the Network layer to pass the data to the appropriate upper-layer protocol.

Example values are:

- 01 ICMP
- 06 TCP
- 17 UDP



Type-of-Service

The Type-of-Service field contains an 8-bit binary value that is used to determine the priority of each packet. This value enables a Quality-of-Service (QoS) mechanism to be applied to high priority packets, such as those carrying telephony voice data. The router processing the packets can be configured to decide which packet it is to forward first based on the Type-of-Service value.

Fragment Offset

As mentioned earlier, a router may 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.

More Fragments flag

The More Fragments (MF) flag is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packets. The More Fragments flag bit is set, it means that it is not the last fragment of a packet. When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet. An unfragmented packet has all zero fragmentation information (MF = 0, fragment offset =0).

Don't Fragment flag

The Don't Fragment (DF) flag is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.

Other IPv4 Header Fields

Roll over each field on the graphic to see its purpose.

Version - Contains the IP version number (4).

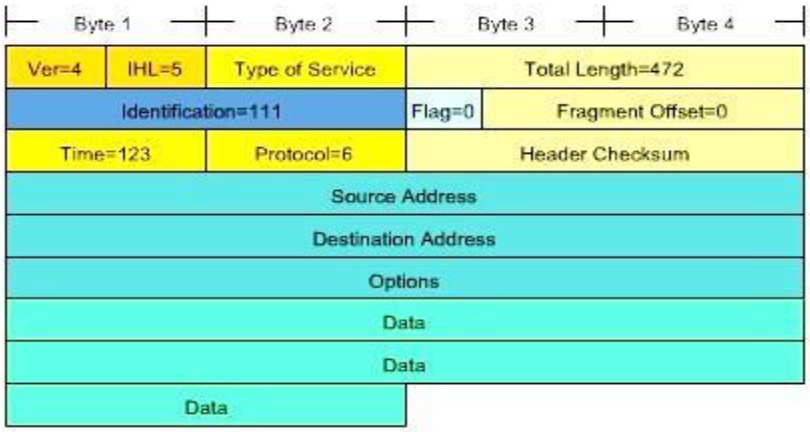
Header Length (IHL) - Specifies the size of the packet header.

Packet Length - This field gives the entire packet size, including header and data, in bytes.

Identification - This field is primarily used for uniquely identifying fragments of an original IP packet.

Header Checksum - The checksum field is used for error checking the packet header.

Options - There is provision for additional fields in the IPv4 header to provide other services but these are rarely used.



Typical IP Packet

The figure represents a complete IP packet with typical header field values.

Ver = 4; IP version.

IHL = 5; size of header in 32 bit words (4 bytes). This header is 5*4 = 20 bytes, the minimum valid size.

Total Length = 472; size of packet (header and data) is 472 bytes.

Identification = 111; original packet identifier (required if it is later fragmented).

Flag = 0; denotes packet can be fragmented if required.

Fragment Offset = 0; denotes that this packet is not currently fragmented (there is no offset).

Time to Live = 123; denotes the Layer 3 processing time in seconds before the packet is dropped (decremented by at least 1 every time a device processes the packet header).

Protocol = 6; denotes that the data carried by this packet is a TCP segment.

5.2.2 → Why Separate Hosts Into Networks? - Performance

As networks grow larger they present problems that can be at least partially alleviated by dividing the network into smaller interconnected networks.

Common issues with large networks are:

- Performance degradation
- Security issues
- Address Management

Improving Performance

Large numbers of hosts connected to a single network can produce volumes of data traffic that may stretch, if not overwhelm, network resources such as bandwidth and routing capability.

Dividing large networks so that hosts who need to communicate are grouped together reduces the traffic across the internetworks.

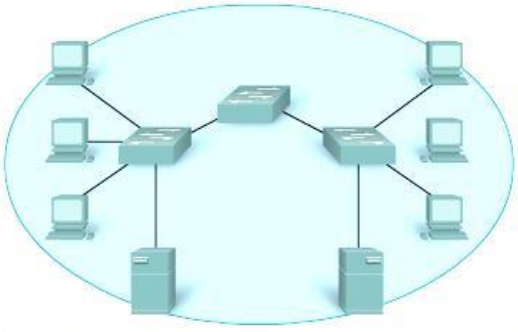
In addition to the actual data communications between hosts, network management and control traffic (overhead) also increases with the number of hosts. A significant contributor to this overhead can be network broadcasts.

A broadcast is a message sent from one host to all other hosts on the network. Typically, a host initiates a broadcast when information about another unknown host is required. Broadcasts are a necessary and useful tool used by protocols to enable data communication on networks. However, large numbers of hosts generate large numbers of broadcasts that consume network bandwidth. And because every other host has to process the broadcast packet it receives, the other productive functions that a host is performing are also interrupted or degraded.

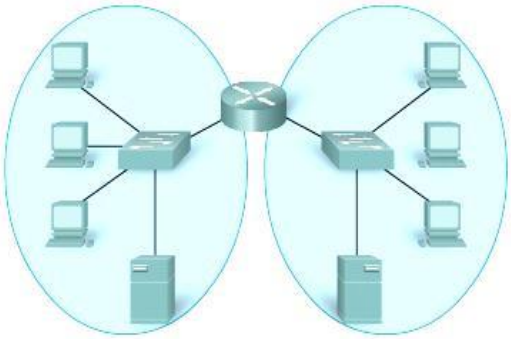
Broadcasts are contained within a network. In this context, a network is also known as a broadcast domain. Managing the size of broadcast domains by dividing a network into subnets ensures that network and host performances are not degraded to unacceptable levels. Improving Performance

Large numbers of hosts connected to a single network can produce volumes of data traffic that may stretch, if not overwhelm, network resources such as bandwidth and routing capability.

Dividing large networks so that hosts who need to communicate are grouped together reduces the traffic across the internetworks.



All devices in this network are connected in one broadcast domain when the switch is set to the factory default settings. Since switches forward broadcasts by default, broadcasts are processed by all devices in this network.



Replacing the middle switch with a router creates 2 IP subnets, hence, 2 distinct broadcast domains. All devices are connected but local broadcasts are contained.

In addition to the actual data communications between hosts, network management and control traffic (overhead) also increases with the number of hosts. A significant contributor to this overhead can be network broadcasts.

A broadcast is a message sent from one host to all other hosts on the network. Typically, a host initiates a broadcast when information about another unknown host is required. Broadcasts are a necessary and useful tool used by protocols to enable data communication on networks. However, large numbers of hosts generate large numbers of broadcasts that consume network bandwidth. And because every other host has to process the broadcast packet it receives, the other productive functions that a host is performing are also interrupted or degraded.

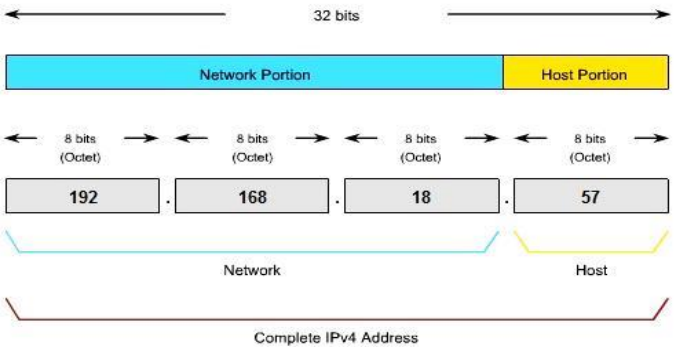
Broadcasts are contained within a network. In this context, a network is also known as a broadcast domain. Managing the size of broadcast domains by dividing a network into subnets ensures that network and host performances are not degraded to unacceptable levels.

5.2.6→ Dividing the Networks – Networks from Netwroks

If a large network has to be divided, additional layers of addressing can be created. Using hierarchical addressing means that the higher levels of the address are retained; with a subnetwork level and then the host level.

The logical 32-bit IPv4 address is hierarchical and is made up of two parts. The first part identifies the network and the second part identifies a host on that network. Both parts are required for a complete IP address.

For convenience IPv4 addresses are divided in four groups of eight bits (octets). Each octet is converted to its decimal value and the complete address written as the four decimal values separated by a dot (period).



For example - 192.168.18.57

In this example, as the figure shows, the first three octets, (192.168.18), can identify the network portion of the address, and the last octet, (57) identifies the host.

This is hierarchical addressing because the network portion indicates the network on which each unique host address is located. Routers only need to know how to reach each network, rather than needing to know the location of each individual host.

With IPv4 hierarchical addressing, the network portion of the address for all hosts in a network is the same. To divide a network, the network portion of the address is extended to use bits from the host portion of the address. These borrowed host bits are then used as network bits to represent the different subnetworks within the range of the original network.

Given that an IPv4 address is 32 bits, when host bits are used to divide a network the more subnetworks created results in fewer hosts for each subnetwork. Regardless of the number of subnetworks created however, all 32 bits are required to identify an individual host.

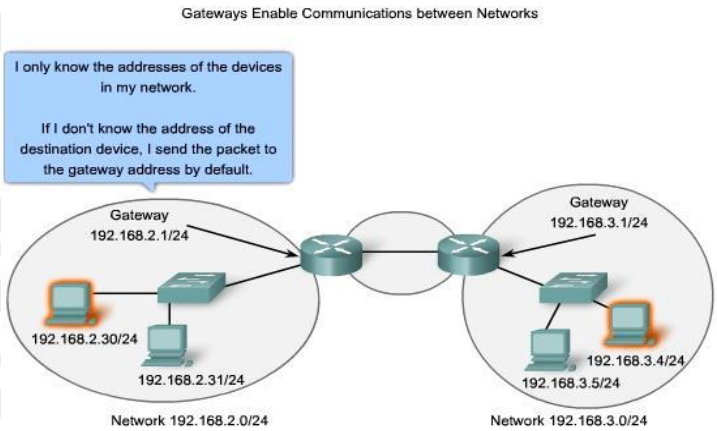
The number of bits of an address used as the network portion is called the prefix length. For example if a network uses 24 bits to express the network portion of an address the prefix is said to be /24. In the devices in an IPv4 network, a separate 32-bit number called a subnet mask indicates the prefix.

Extending the prefix length or subnet mask enables the creation of these subnetworks. In this way network administrators have the flexibility to divide networks to meet different needs, such as location, managing network performance, and security, while ensuring each host has a unique address.

For the purposes of explanation, however in this chapter the first 24 bits of an IPv4 address will be used as the network portion.

5.3.1→ Device Parameters - Supporting Communication outside Our Network

Within a network or a subnet work, hosts communicate with each other without the need for any Network layer intermediary device. When a host needs to communicate with another network, an intermediary device, or router, acts as a gateway to the other network.



As a part of its configuration, a host has a default gateway address defined. As shown in the figure, this gateway address is the address of a router interface that is connected to the same network as the host.

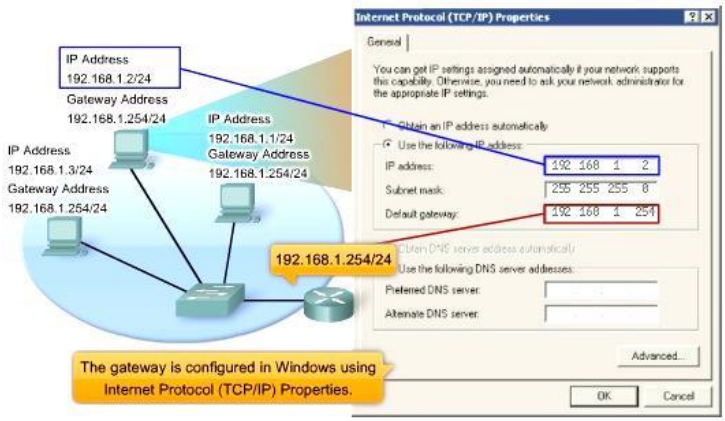
Keep in mind that it is not feasible for a particular host to know the address of every device on the Internet with which it may have to communicate. To communicate with a device on another network, a host uses the address of this gateway, or default gateway, to forward a packet outside the local network.

The router also needs a route that defines where to forward the packet next. This is called the next-hop address. If a route is available to the router, the router will forward the packet to the next-hop router that offers a path to the destination network.

5.3.3→ A Gateway – The Way out of Our 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.

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 may 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.

5.3.4 → A Route – The 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 via 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.

The routing table in a Cisco router can be examined with the **show ip route** command.

5.3.5 → The Destination Network

Default Route

A router can be configured to have a default route. A default route is a route that will match all destination networks. In IPv4 networks, the address 0.0.0.0 is used for this purpose. The default route is used to forward packets for which there is no entry in the routing table for the destination network. 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.

5.3.7 → Packet Forwarding – Moving the Packet Toward its Destination

Routing is done packet-by-packet and hop-by-hop. Each packet is treated independently in 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.

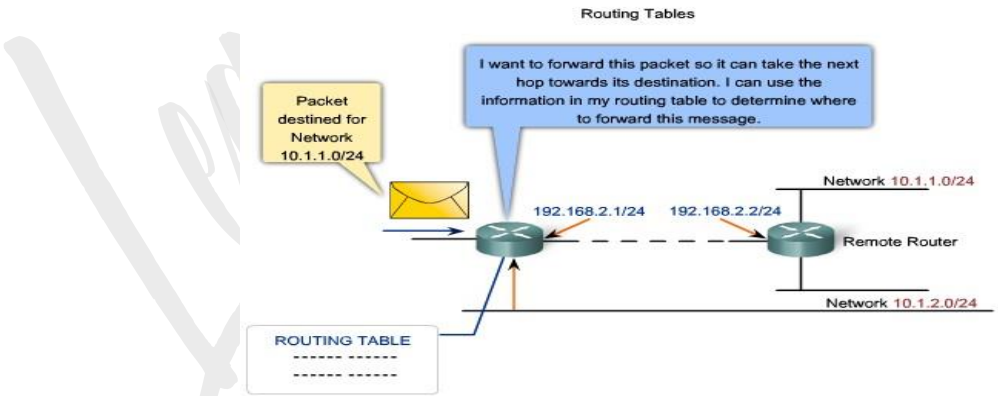
The router will do one of three things with the packet:

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

5.4.1 → Routing Protocols – Sharing the Routes

Routing requires that every hop, or router, along the path to a packet's destination have a route to forward the packet. Otherwise, the packet is dropped at that hop. Each router in a path does not need a route to all networks. It only needs to know the next hop on the path to the packet's destination network.

The routing table contains the information that a router uses in its packet forwarding decisions. For the routing decisions, the routing table needs to represent the most accurate state of network pathways that the router can access. Out-of-date routing information means that packets may not be forwarded to the most appropriate next-hop, causing delays or packet loss.



This route information can be manually configured on the router or learned dynamically from other routers in the same internetwork. After the interfaces of a router are configured and operational, the network associated with each interface is installed in the routing table as a directly connected route.

5.4.2 → Static Routing

Routes to remote networks with the associated next hops can be manually configured on the router. This is known as **static routing**. A default route can also be statically configured.

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.

Further, 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 may be incomplete or inaccurate, resulting in packet delays and possible packet loss.

5.4.3 → Dynamic Routing

Although it is essential for all routers in an internetwork to have up-to-date extensive route knowledge, maintaining the routing table by manual static configuration is not always feasible. Therefore, dynamic routing protocols are used. 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 routers, this 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 way. An example of router sharing routes is shown in the figure.

Common routing protocols are:

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

Although routing protocols provide routers with up-to-date routing tables, there are costs. First, the exchange of route information adds overhead that consumes network bandwidth. This overhead can be an issue, particularly for low bandwidth links between routers. Second, the route information that a router receives is processed extensively by protocols such as EIGRP and OSPF to make routing table entries. 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.

Static routing does not produce any network overhead and places entries directly into the routing table; no processing is required by the router. The cost for static routing is administrative - the manual configuration and maintenance of the routing table to ensure efficient and effective routing.

In many internetworks, a combination of static, dynamic, and default routes are used to provide the necessary routes.

6.1.1 → The Anatomy of an IPv4 Address

Each device on a network must be uniquely defined. At the Network layer, the packets of the communication need to be identified with the source and destination addresses of the two end systems. With IPv4, this means that each packet has a 32-bit source address and a 32-bit destination address in the Layer 3 header.

These addresses are used in the data network as binary patterns. Inside the devices, digital logic is applied for their interpretation. For us in the human network, a string of 32 bits is difficult to interpret and even more difficult to remember. Therefore, we represent IPv4 addresses using dotted decimal format.

Dotted Decimal

Binary patterns representing IPv4 addresses are expressed as dotted decimals by separating each byte of the binary pattern, called an octet, with a dot. It is called an octet because each decimal number represents one byte or 8 bits.

For example, the address:

10101100000100000000010000010100

is expressed in dotted decimal as:

172.16.4.20

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

Network and Host Portions

For each IPv4 address, some portion of the high-order bits represents the network address. At Layer 3, we define a network as a group of hosts that have identical bit patterns in the network address portion of their addresses.

Although all 32 bits define the IPv4 host address, we have a variable number of bits that are called the host portion of the address. The number of bits used in this host portion determines the number of hosts that we can have within the network.

6.1.2 → Knowing the Numbers - Binary to Decimal Conversion

To understand the operation of a device in a network, we need to look at addresses and other data the way the device does - in binary notation. This means that we need to have some skill in binary to decimal conversion.

Data represented in binary may represent many different forms of data to the human network. In this discussion, we refer to binary as it relates to IPv4 addressing. This means that we look at each byte (octet) as a decimal number in the range of 0 to 255.

Positional Notation

Learning to convert binary to decimal requires an understanding of the mathematical basis of a numbering system called positional notation. Positional notation means that a digit represents different values depending on the position it occupies. More specifically, the value that a digit represents is that value multiplied by the power of the base, or radix, represented by the position the digit occupies. Some examples will help to clarify how this system works.

For the decimal number 245, the value that the 2 represents is 2*10^2 (2 times 10 to the power of 2). The 2 is in what we commonly refer to as the "100s" position. Positional notation refers to this position as the base^2 position because the base, or radix, is 10 and the power is 2.

Using positional notation in the base 10 number system, 245 represents:

245 = (2 * 10^2) + (4 * 10^1) + (5 * 10^0)

or

245 = (2 * 100) + (4 * 10) + (5 * 1)

Binary Numbering System

In the binary numbering system, the radix is 2. Therefore, each position represents increasing powers of 2. In 8-bit binary numbers, the positions represent these quantities:

2^7 2^62^5 2^4 2^32^2 2^1 2^0

128 64 32 16 8 4 2 1

The base 2 numbering system only has two digits: 0 and 1.

When we interpret a byte as a decimal number, we have the quantity that position represents if the digit is a 1 and we do not have that quantity if the digit is a 0, as shown in the figure.

1 1 1 1 1 1 1 1

128 64 32 16 8 4 2 1

A 1 in each position means that we add the value for that position to the total. This is the addition when there is a 1 in each position of an octet. The total is 255.

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

A 0 in each position indicates that the value for that position is not added to the total. A 0 in every position yields a total of 0.

0 0 0 0 0 0 0 0

128 64 32 16 8 4 2 1

0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0.

6.1.4 → Knowing the Numbers - Decimal to Binary Conversions

:

:

6.2.1→ Type of Address in an IPv4 Network

Within the address range of each IPv4 network, we have three types of addresses:

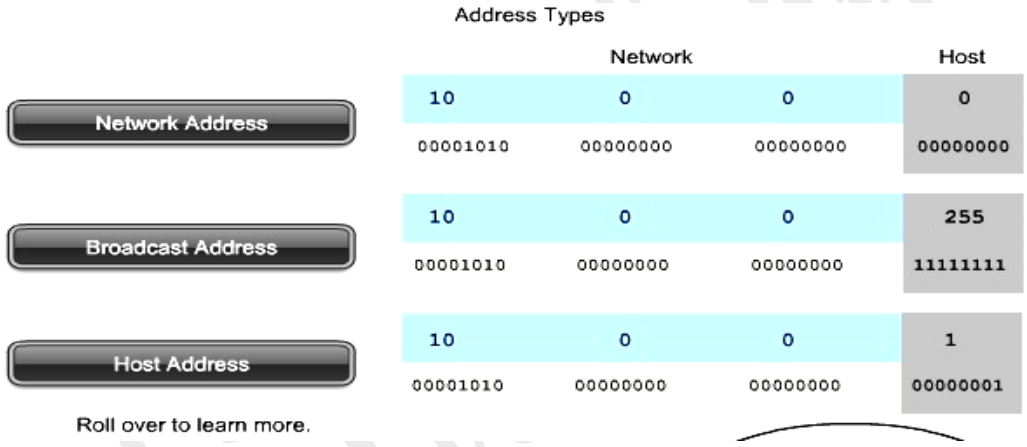
Network address - The address by which we refer to the network

Broadcast address - A special address used to send data to all hosts in the network

Host addresses - The addresses assigned to the end devices in the network

Network Address

The network address is a standard way to refer to a network. For example, we could refer to the network shown in the figure 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. 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



Broadcast Address

The IPv4 broadcast address is a special address for each network that allows communication 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.

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, the broadcast address would be 10.0.0.255. This address is also referred to as the directed broadcast.

Host Addresses

As described previously, every end device requires a unique address to deliver a packet to that host. In IPv4 addresses, we assign the values between the network address and the broadcast address to the devices in that network.

Network Prefixes

An important question is: How do we know how many bits represent the network portion and how many bits represent the host portion? When we express an IPv4 network address, we add a prefix length to the network address. **The prefix length is the number of bits in the address that gives us the network portion.** For example, in 172.16.4.0 /24, the /24 is the prefix length - it tells us that the first 24 bits are the network address. This leaves the remaining 8 bits, the last octet, as the host portion.

Networks are not always assigned a /24 prefix. Depending on the number of hosts on the network, the prefix assigned may be different. Having a different prefix number changes the host range and broadcast address for each network.

6.2.3→ Unicast, Broadcast, Multicast—Types of communication

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

These three types of communication are used for different purposes in the data networks. In all three cases, the IPv4 address of the originating host is placed in the packet header as the source address.

Unicast Traffic

Unicast communication is used for the normal host-to-host communication in both a client/server and a peer-to-peer network. Unicast packets use the host address of the destination device as the destination address and can be routed through an internetwork. Broadcast and multicast, however, use special addresses as the destination address. Using these special addresses, broadcasts are generally restricted to the local network. The scope of multicast traffic also may be limited to the local network or routed through an internetwork.

Broadcast Transmission

Because broadcast traffic is used to send packets to all hosts in the network, a packet uses a special broadcast address. 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.

Broadcast transmission is used for the location of special services/devices for which the address is not known or when a host needs to provide information to all the hosts on the network.

Some examples for using broadcast transmission are:

Mapping upper layer addresses to lower layer addresses

Requesting an address

Exchanging routing information by routing protocols

When a host needs information, the host sends a request, called a query, to the broadcast address. All hosts in the network receive and process this query. One or more of the hosts with the requested information will respond, typically using unicast.

Similarly, when a host needs to send information to the hosts on a network, it creates and sends a broadcast packet with the information.

Unlike unicast, where the packets can be routed throughout the internetwork, broadcast packets are usually restricted to the local network. This restriction is dependent on the configuration of the router that borders the network and the type of broadcast. There are two types of broadcasts: directed broadcast and limited broadcast.

Multicast Transmission

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.

Some examples of multicast transmission are:

- Video and audio distribution
- Routing information exchange by routing protocols
- Distribution of software
- News feeds

Multicast Clients

Hosts that wish to receive particular multicast data are called multicast clients. The multicast clients use services initiated by a client program to subscribe to the multicast group.

Each multicast group is represented by a single IPv4 multicast destination address. When an IPv4 host subscribes to a multicast group, the host processes packets addressed to this multicast address as well as packets addressed to its uniquely allocated unicast address. As we will see, IPv4 has set aside a special block of addresses from 224.0.0.0 to 239.255.255.255 for multicast groups addressing.

6.2.4 → Reserved IPv4 address Ranges

Expressed in dotted decimal format, the IPv4 address range is 0.0.0.0 to 255.255.255.255. As you have already seen, not all of these addresses can be used as host addresses for unicast communication.

Experimental Addresses

One major block of addresses reserved for special purposes is the IPv4 experimental address range 240.0.0.0 to 255.255.255.254. Currently, these addresses are listed as reserved for future use (RFC 3330). This suggests that they could be converted to usable addresses. Currently, they cannot be used in IPv4 networks. However, these addresses could be used for research or experimentation.

Multicast Addresses

As previously shown, another major block of addresses reserved for special purposes is the IPv4 multicast address range 224.0.0.0 to 239.255.255.255. Additionally, the multicast address range is subdivided into different types of addresses: **reserved link local addresses** and **globally scoped addresses**. One additional type of multicast address is the **administratively scoped addresses**, also called limited scope addresses.

The IPv4 multicast addresses 224.0.0.0 to 224.0.0.255 are reserved link local addresses. These addresses are to be used for multicast groups on a local network. Packets to these destinations are always transmitted with a time-to-live (TTL) value of 1. Therefore, a router connected to the local network should never forward them. A typical use of reserved link-local addresses is in routing protocols using multicast transmission to exchange routing information.

The globally scoped addresses are 224.0.1.0 to 238.255.255.255. They may be used to multicast data across the Internet. For example, 224.0.1.1 has been reserved for **Network Time Protocol (NTP)** to synchronize the time-of-day clocks of network devices.

Host Addresses

After accounting for the ranges reserved for experimental addresses and multicast addresses, this leaves an address range of 0.0.0.0 to 223.255.255.255 that could be used for IPv4 hosts. However, within this range are many addresses that are already reserved for special purposes. Although we have previously covered some of these addresses, the major reserved addresses are discussed in the next section.

6.2.5 → Public and private address

Although most IPv4 host addresses are public addresses designated for use in networks that are accessible on the Internet, there are blocks of addresses that are used in networks that require limited or no Internet access. These addresses are called private addresses.

Private Addresses

The private address blocks are:

- 10.0.0.0 to 10.255.255.255 (10.0.0.0 /8)
- 172.16.0.0 to 172.31.255.255 (172.16.0.0 /12)
- 192.168.0.0 to 192.168.255.255 (192.168.0.0 /16)

Private space address blocks, as shown in the figure, are set aside for use in private networks. The use of these addresses need not be unique among outside networks. Hosts that do not require access to the Internet at large may make unrestricted use of private addresses. However, the internal networks still must design network address schemes to ensure that the hosts in the private networks use IP addresses that are unique within their networking environment.

Many hosts in different networks may use the same private space addresses. Packets using these addresses as the source or destination should not appear on the public Internet. The router or firewall device at the perimeter of these private networks must block or translate these addresses. Even if these packets were to make their way to the Internet, the routers would not have routes to forward them to the appropriate private network.

Network Address Translation (NAT)

With services to translate private addresses to public addresses, hosts on a privately addressed network can have access to resources across the Internet. These services, called Network Address Translation (NAT), can be implemented on a device at the edge of the private network.

NAT allows the hosts in the network to "borrow" a public address for communicating to outside networks. While there are some limitations and performance issues with NAT, clients for most applications can access services over the Internet without noticeable problems.

Note: NAT will be covered in detail in a subsequent course.

Public Addresses

The vast majority of the addresses in the IPv4 unicast host range are public addresses. These addresses are designed to be used in the hosts that are publicly accessible from the Internet. Even within these address blocks, there are many addresses that are designated for other special purposes.

6.2.6 → Special IPv4 Address

There are certain addresses that cannot be assigned to hosts for various reasons. There are also special addresses that can be assigned to hosts but with restrictions on how those hosts can interact within the network.

Network and Broadcast Addresses

As explained earlier, within each network the first and last addresses cannot be assigned to hosts. These are the network address and the broadcast address, respectively.

Default Route

Also presented earlier, we represent the IPv4 default route as 0.0.0.0. The default route is used as a "catch all" route when a more specific route is not available. The use of this address also reserves all addresses in the 0.0.0.0 - 0.255.255.255 (0.0.0.0 /8) address block.

Loopback

One such reserved address is the IPv4 loopback address 127.0.0.1. The loopback is a special address that hosts use to direct traffic to themselves. The loopback address creates a shortcut method for TCP/IP applications and services that run on the same device to communicate with one another. By using the loopback address instead of the assigned IPv4 host address, two services on the same host can bypass the lower layers of the TCP/IP stack. You can also ping the loopback address to test the configuration of TCP/IP on the local host.

Although only the single 127.0.0.1 address is used, addresses 127.0.0.0 to 127.255.255.255 are reserved. Any address within this block will loop back within the local host. No address within this block should ever appear on any network.

Link-Local Addresses

IPv4 addresses in the address block 169.254.0.0 to 169.254.255.255 (169.254.0.0 /16) are designated as link-local addresses. These addresses can be automatically assigned to the local host by the operating system in environments where no IP configuration is available. These might be used in a small peer-to-peer network or for a host that could not automatically obtain an address from a Dynamic Host Configuration Protocol (DHCP) server.

Communication using IPv4 link-local addresses is only suitable for communication with other devices connected to the same network, as shown in the figure. A host must not send a packet with an IPv4 link-local destination address to any router for forwarding and should set the IPv4 TTL for these packets to 1.

Link-local addresses do not provide services outside of the local network. However, many client/server and peer-to-peer applications will work properly with IPv4 link-local addresses.

TEST-NET Addresses

The address block 192.0.2.0 to 192.0.2.255 (192.0.2.0 /24) is set aside for teaching and learning purposes. These addresses can be used in documentation and network examples. Unlike the experimental addresses, network devices will accept these addresses in their configurations. You may often find these addresses used with the domain names example.com or example.net in RFCs, vendor, and protocol documentation. Addresses within this block should not appear on the Internet.

6.2.7 → Legacy IP v4 Addressing

Historic Network Classes

Historically, RFC1700 grouped the unicast ranges into specific sizes called class A, class B, and class C addresses. It also defined class D (multicast) and class E (experimental) addresses, as previously presented.

The unicast address classes A, B, and C defined specifically-sized networks as well as specific address blocks for these networks, as shown in the figure. A company or organization was assigned an entire class A, class B, or class C address block. This use of address space is referred to as classful addressing.

Class A Blocks

A class A address block was designed to support extremely large networks with more than 16 million host addresses. Class A IPv4 addresses used a fixed /8 prefix with the first octet to indicate the network address. The remaining three octets were used for host addresses.

To reserve address space for the remaining address classes, all class A addresses required that the most significant bit of the high-order octet be a zero. This meant that there were only 128 possible class A networks, 0.0.0.0 /8 to 127.0.0.0 /8, before taking out the reserved address blocks. Even though the class A addresses reserved one-half of the address space, because of their limit of 128 networks, they could only be allocated to approximately 120 companies or organizations.

Class B Blocks

Class B address space was designed to support the needs of moderate to large size networks with more than 65,000 hosts. A class B IP address used the two high-order octets to indicate the network address. The other two octets specified host addresses. As with class A, address space for the remaining address classes needed to be reserved.

For class B addresses, the most significant two bits of the high-order octet were 10. This restricted the address block for class B to 128.0.0.0 /16 to 191.255.0.0 /16. Class B had slightly more efficient allocation of addresses than class A because it equally divided 25% of the total IPv4 address space among approximately 16,000 networks.

Class C Blocks

The class C address space was the most commonly available of the historic address classes. This address space was intended to provide addresses for small networks with a maximum of 254 hosts.

Class C address blocks used a /24 prefix. This meant that a class C network used only the last octet as host addresses with the three high-order octets used to indicate the network address.

Class C address blocks set aside address space for class D (multicast) and class E (experimental) by using a fixed value of 110 for the three most significant bits of the high-order octet. This restricted the address block for class C to 192.0.0.0 /16 to 223.255.255.0 /16. Although it occupied only 12.5% of the total IPv4 address space, it could provide addresses to 2 million networks.

Limits to the Class-based System

Not all organizations' 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, you will see remnants of it in networks today. For example, when you assign an IPv4 address to a computer, the operating system examines the address being assigned to determine if this address is a class A, class B, or class C. 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 may assume the prefix length based on the class of the address.

Classless Addressing

The system that we currently 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

IP Address Classes

Address Class	1st octet range (decimal)	1st octet bits (green bits do not change)	Network(N) and Host(H) parts of address	Default subnet mask (decimal and binary)	Number of possible networks and hosts per
A	1-127**	00000000-01111111	N.H.H.H	255.0.0.0	126 nets (2^7) 16,777,214 hosts per net (2^24-2)
B	128-191	10000000-10111111	N.N.H.H	255.255.0.0	16,384 nets (2^14) 65,534 hosts per net (2^16-2)
C	192-223	11000000-11011111	N.N.N.H	255.255.255.0	2,097,150 nets (2^21) 254 hosts per net (2^8-2)
D	224-239	11100000-11101111	NA (multicast)		
E	240-255	11110000-11111111	NA (experimental)		

** All zeros (0) and all ones (1) are invalid hosts addresses.

6.3.1 → Planning to address the Network

The allocation of Network layer address space within the corporate network needs to be well designed. Network administrators should not randomly select the addresses used in their networks. Nor should address assignment within the network be random.

The allocation of these addresses inside the networks should be planned and documented for the purpose of:

- Preventing duplication of addresses
- Providing and controlling access
- Monitoring security and performance
- Preventing Duplication of Addresses

As you already know, each host in an internetwork must have a unique address. Without the proper planning and documentation of these network allocations, we could easily assign an address to more than one host.

Providing and Controlling Access

Some hosts provide resources to the internal network as well as to the external network. One example of these devices is servers. Access to these resources can be controlled by the Layer 3 address. If the addresses for these resources are not planned and documented, the security and accessibility of the devices are not easily controlled. For example, if a server has a random address assigned, blocking access to its address is difficult and clients may not be able to locate this resource.

Monitoring Security and Performance

Similarly, we need to monitor the security and performance of the network hosts and the network as a whole. As part of the monitoring process, we examine network traffic looking for addresses that are generating or receiving excessive packets. If we have proper planning and documentation of the network addressing, we can identify the device on the network that has a problematic address.

Assigning Addresses within a Network

As you have already learned, hosts are associated with an IPv4 network by a common network portion of the address. Within a network, there are different types of hosts.

Some examples of different types of hosts are:

- End devices for users
- Servers and peripherals

- Hosts that are accessible from the Internet
- Intermediary devices

Each of these different device types should be allocated to a logical block of addresses within the address range of the network.

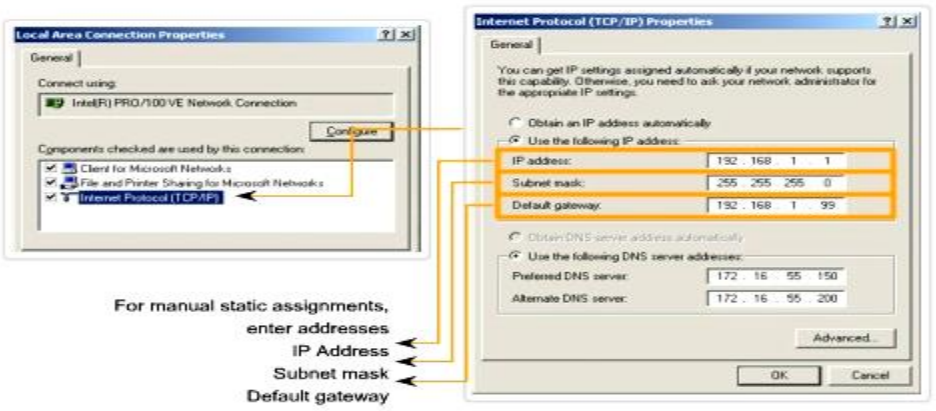
6.3.2 → Static or Dynamic Addressing For end user Devices

Static Assignment of Addresses

With a static assignment, the network administrator must manually configure the network information for a host, as shown in the figure. At a minimum, this includes entering the host IP address, subnet mask, and default gateway.

Static addresses have some advantages over dynamic addresses. For instance, they are useful for printers, servers, and other networking devices that need to be accessible to clients on the network. If hosts normally access a server at a particular IP address, it would cause problems if that address changed. Additionally, static assignment of addressing information can provide increased control of network resources. However, it can be time-consuming to enter the information on each host.

Addressing End Devices



Dynamic Assignment of Addresses

Because of the challenges associated with static address management, end user devices often have addresses dynamically assigned, using Dynamic Host Configuration Protocol (DHCP), as shown in the figure.

DHCP enables the automatic assignment of addressing information such as IP address, subnet mask, default gateway, and other configuration information. The configuration of the DHCP server requires that a block of addresses, called an address pool, be defined to be assigned to the DHCP clients on a network. Addresses assigned to this pool should be planned so that they exclude any addresses used for the other types of devices.

DHCP is generally the preferred method of assigning IP addresses to hosts on large networks because it reduces the burden on network support staff and virtually eliminates entry errors.

Another benefit of DHCP is that an address is not permanently assigned to a host but is only "leased" for a period of time. If the host is powered down or taken off the network, the address is returned to the pool for reuse. This feature is especially helpful for mobile users that come and go on a network.

6.3.4→ Who assigns the Different Address?

Internet Assigned Numbers Authority (IANA) (http://www.iana.net) is the master holder of the IP addresses. The IP multicast addresses and the IPv6 addresses are obtained directly from IANA. Until the mid-1990s, all IPv4 address space was managed directly by the IANA. At that time, the remaining IPv4 address space was allocated to various other registries to manage for particular purposes or for regional areas. These registration companies are called Regional Internet Registries (RIRs), as shown in the figure.

The major registries are:

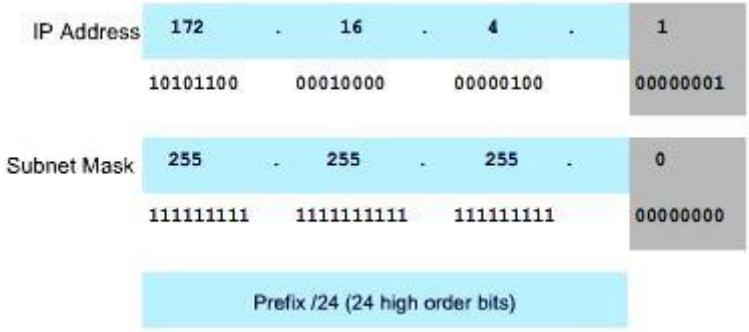
- AfriNIC (African Network Information Centre) - Africa Region http://www.afrinic.net
- APNIC (Asia Pacific Network Information Centre) - Asia/Pacific Region http://www.apnic.net
- ARIN (American Registry for Internet Numbers) - North America Region http://www.arin.net
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) - Latin America and some Caribbean Islands http://www.lacnic.net
- RIPE NCC (Reseaux IP Europeans) - Europe, the Middle East, and Central Asia http://www.ripe.net

6.4.1 → The Subnet Mask-Defining the network and Host Portions

As we learned earlier, an IPv4 address has a network portion and a host portion. We referred to the prefix length as the number of bits in the address giving us the network portion. The prefix is a way to define the network portion that is human readable. The data network must also have this network portion of the addresses defined.

To define the network and host portions of an address, the devices use a separate 32-bit pattern called a subnet mask, as shown in the figure. We express the subnet mask in the same dotted decimal format as the IPv4 address. The subnet mask is created by placing a binary 1 in each bit position that represents the network portion and placing a binary 0 in each bit position that represents the host portion.

The prefix and the subnet mask are different ways of representing the same thing - the network portion of an address.



As shown in the figure, a /24 prefix is expressed as a subnet mask as 255.255.255.0 (11111111.11111111.11111111.00000000). The remaining bits (low order) of the subnet mask are zeroes, indicating the host address within the network.

The subnet mask is configured on a host in conjunction with the IPv4 address to define the network portion of that address.

For example, let's look at the host 172.16.4.35/27:

address

172.16.20.35

10101100.00010000.00010100.00100011

subnet mask

255.255.255.224

11111111.11111111.11111111.11100000

network address

172.16.20.32

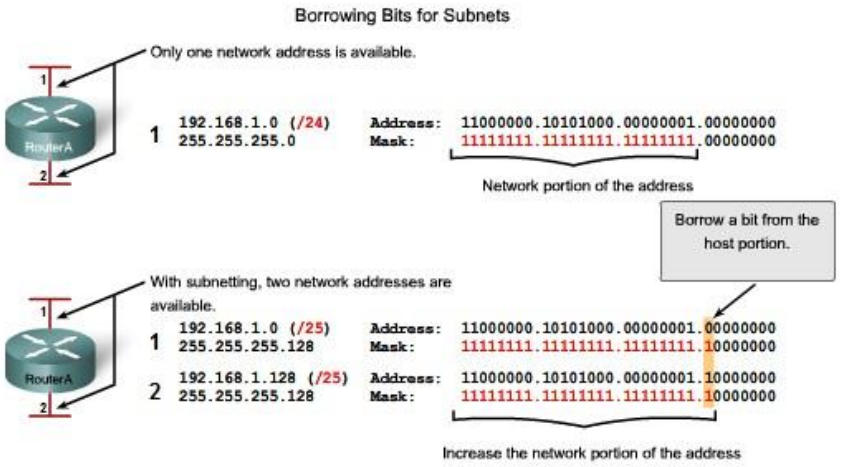
10101100.00010000.00010100.00100000

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 will recall that we only need to expand an octet if the network and host division falls within that octet. Therefore, there are a limited number 8 bit patterns used in address masks.

6.5.1 → Basic Subnetting

Subnetting allows for creating multiple logical networks from a single address block. Since we use a router to connect these networks together, each interface on a router must have a unique network ID. Every node on that link is on the same network.

We create the subnets by using one or more of the host bits as network bits. This is done by extending the mask to borrow some of the bits from the host portion of the address to create additional network bits. The more host bits used, the more subnets that can be defined. For each bit borrowed, we double the number of subnetworks available. For example, if we borrow 1 bit, we can define 2 subnets. If we borrow 2 bits, we can have 4 subnets. However, with each bit we borrow, fewer host addresses are available per subnet.



RouterA in the figure has two interfaces to interconnect two networks. Given an address block of 192.168.1.0 /24, we will create two subnets. We borrow one bit from the host portion by using a subnet mask of 255.255.255.128, instead of the original 255.255.255.0 mask. The most significant bit in the last octet 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".

Formula for calculating subnets

Use this formula to calculate the number of subnets:

2^n where n = the number of bits borrowed

In this example, the calculation looks like this:

2^1 = 2 subnets

The number of hosts

To calculate the number of hosts per network, we use the formula of 2^n - 2 where n = the number of bits left for hosts.

Applying this formula, (2^7 - 2 = 126) shows that each of these subnets can have 126 hosts.

For each subnet, examine the last octet in binary. The values in these octets for the two networks are:

Subnet 1: 00000000 = 0

Subnet 2: 10000000 = 128

See the figure for the addressing scheme for these networks.

6.6.1 →Ping 127.0.0.1 – Testing the Local Stack

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 on 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 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 requests are sent, 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.

6.6.2 → Ping Gateway – Testing connectivity to the Local LAN

6.6.3 → Ping Remote Host – Testing Connectivity to Remote LAN

6.6.4 → Traceroute (tracert) – Testing the Path

Ping is used to indicate the connectivity between two hosts. Traceroute (tracert) is a utility that allows us 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 us with important verification and troubleshooting information. If the data reaches the destination, then the trace lists the interface on every router in the path.

If the data fails at some hop along the way, we 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 round trip time (RTT) for each hop along the path and indicates if a hop fails to respond. The round trip time (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.

This information can be used to locate a problematic router in the path. If we get high response times or data losses from a particular hop, this is an indication that the resources of the router or its connections may be stressed.

Time to Live (TTL)

Traceroute makes use of 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 zero, 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.

Play the animation in the figure to see how Traceroute takes advantage of TTL.

The first sequence of messages sent from traceroute will have a TTL field of one. 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...) for each sequence of messages. This provides the trace with the address of each hop as the packets timeout further down the path. The TTL field continues to be increased until the destination is reached or it is incremented to a predefined maximum.

Once the final destination is reached, the host responds with either an ICMP Port Unreachable message or an ICMP Echo Reply message instead of the ICMP Time Exceeded message.

6.6.5 → ICMPv4- The Protocol Supporting Testing and Messaging

Although IPv4 is not a reliable protocol, it does provide for messages to be sent in the event of certain errors. These messages are sent using services of the Internet Control Messaging Protocol (ICMPv4). The purpose of these messages is to provide feedback about issues related to the processing of IP packets under certain conditions, not to make IP reliable. ICMP messages are not required and are often not allowed for security reasons.

ICMP is the messaging protocol for the TCP/IP suite. ICMP provides control and error messages and is used by the ping and traceroute utilities. Although ICMP uses the basic support of IP as if it were a higher-level protocol ICMP, it is actually a separate Layer 3 of the TCP/IP suite.

The types of ICMP messages - and the reasons why they are sent - are extensive. We will discuss some of the more common messages.

ICMP messages that may be sent include:

- Host confirmation
- Unreachable Destination or Service
- Time exceeded
- Route redirection
- Source quench

Host Confirmation

An ICMP Echo Message can be used to determine if a host is operational. The local host sends an ICMP Echo Request to a host. The host receiving the echo message replies with the ICMP Echo Reply, as shown in the figure. This use of the ICMP Echo messages is the basis of the ping utility.

Unreachable Destination or Service

The ICMP Destination Unreachable can used to notify a host that the destination or service is unreachable. When a host or gateway receives a packet that it cannot deliver, it may send an ICMP Destination Unreachable packet to the host originating the packet. The Destination Unreachable packet will contain codes that indicate why the packet could not be delivered.

Among the Destination Unreachable codes are:

0 = net unreachable

1 = host unreachable

2 = protocol unreachable

3 = port unreachable

Codes for net unreachable and host unreachable are responses from a router when it cannot forward a packet. If a router receives a packet for which it does not have a route, it may respond with an ICMP Destination Unreachable with a code = 0, indicating net unreachable. If a router receives a packet for which it has an attached route but is unable to deliver the packet to the host on the attached network, the router may respond with an ICMP Destination Unreachable with a code = 1, indicating that the network is known but the host is unreachable.

The codes 2 and 3 (protocol unreachable and port unreachable) are used by an end host to indicate that the TCP segment or UDP datagram contained in a packet could not be delivered to the upper layer service.

When the end host receives a packet with a Layer 4 PDU that is to be delivered to an unavailable service, the host may respond to the source host with an ICMP Destination Unreachable with a code = 2 or code = 3, indicating that the service is not available. The service may not be available because no daemon is running providing the service or because security on the host is not allowing access to the service.

Time Exceeded

An ICMP Time Exceeded message is used by a router to indicate that a packet cannot be forwarded because the TTL field of the packet has expired. If a router receives a packet and decrements the TTL field in the packet to zero, it discards the packet. The router may also send an ICMP Time Exceeded message to the source host to inform the host of the reason the packet was dropped.

Route Redirection

A router may use the ICMP Redirect Message to notify the hosts on a network that a better route is available for a particular destination. This message may only be used when the source host is on the same physical network as both gateways. If a router receives a packet for which it has a route and for which the next hop is attached to the same interface as the packet arrived, the router may send an ICMP Redirect Message to the source host. This message will inform the source host of the next hop contained in a route in the routing table.

Source Quench

The ICMP Source Quench message can be used to tell the source to temporarily stop sending packets. If a router does not have enough buffer space to receive incoming packets, a router will discard the packets. If the router has to do so, it may also send an ICMP Source Quench message to source hosts for every message that it discards.

A destination host may also send a source quench message if datagrams arrive too fast to be processed.

When a host receives an ICMP Source Quench message, it reports it to the Transport layer. The source host can then use the TCP flow control mechanisms to adjust the transmission.

7.1.1 → Data Link Layer – Supporting & Connecting to Upper Services

The Data Link layer provides a means for exchanging data over a common local media.

The Data Link layer performs two basic services:

- Allows the upper layers to access the media using techniques such as framing
- Controls how data is placed onto the media and is received from the media using techniques such as media access control and error detection

As with each of the OSI layers, there are terms specific to this layer:

Frame - The Data Link layer PDU

Node - The Layer 2 notation for network devices connected to a common medium

Media/medium (physical)* - The physical means for the transfer of information between two nodes

Network (physical)** - Two or more nodes connected to a common medium

The Data Link layer is responsible for the exchange of frames between nodes over the media of a physical network.

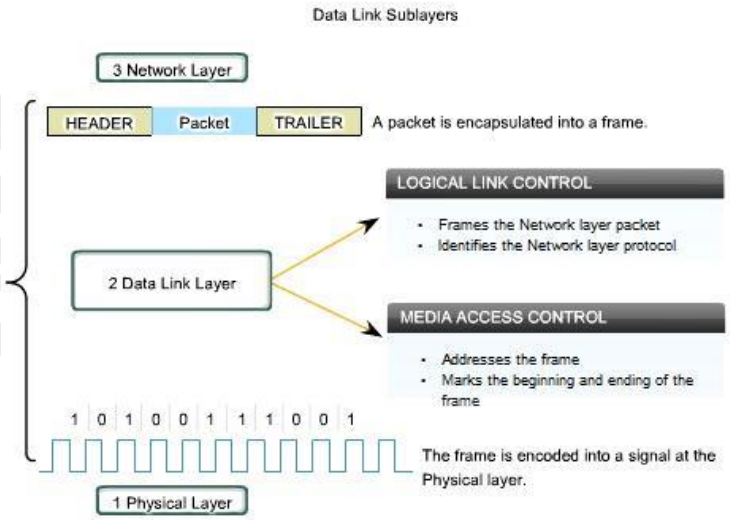
7.1.4 →Data Link Layer – Connecting Upper Layer Services to the Media

Data Link Sublayers

To support a wide variety of network functions, the Data Link layer is often divided into two sublayers: an upper sublayer and an lower sublayer.

- The upper sublayer defines the software processes that provide services to the Network layer protocols.
- The lower sublayer defines the media access processes performed by the hardware.

Separating the Data Link layer into sublayers allows for one type of frame defined by the upper layer to access different types of media defined by the lower layer. Such is the case in many LAN technologies, including Ethernet.



The two common LAN sublayers are:

Logical Link Control

Logical Link Control (LLC) places information in the frame that identifies which Network layer protocol is being used for the frame. This information allows multiple Layer 3 protocols, such as IP and IPX, to utilize the same network interface and media.

Media Access Control

Media Access Control (MAC) provides Data Link layer addressing and delimiting of data according to the physical signaling requirements of the medium and the type of Data Link layer protocol in use.

7.3.5 →Data Link Layer Protocols – The Frame

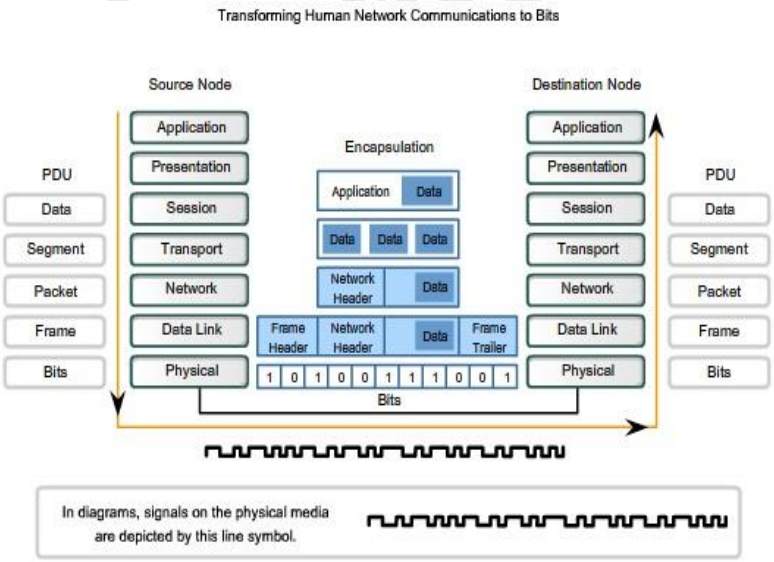
In a TCP/IP network, all OSI Layer 2 protocols work with the Internet Protocol at OSI Layer 3. However, the actual Layer 2 protocol used depends on the logical topology of the network and the implementation of the Physical layer. Given the wide range of physical media used across the range of topologies in networking, there are a correspondingly high number of Layer 2 protocols in use.

Protocols that will be covered in CCNA courses include:

- Ethernet
- Point-to-Point Protocol (PPP)
- High-Level Data Link Control (HDLC)
- Frame Relay
- Asynchronous Transfer Mode (ATM)

8.1.1 →Physical Layer – Purpose

The OSI Physical layer provides the means to transport across the network media the bits that make up a Data Link layer frame. This layer accepts a complete frame from the Data Link layer and encodes it as a series of signals that are transmitted onto the local media. The encoded bits that comprise a frame are received by either an end device or an intermediate device.



The delivery of frames across the local media requires the following Physical layer elements:

- The physical media and associated connectors
- A representation of bits on the media
- Encoding of data and control information
- Transmitter and receiver circuitry on the network devices

At this stage of the communication process, the user data has been segmented by the Transport layer, placed into packets by the Network layer, and further encapsulated as frames by the Data Link layer. The purpose of the Physical layer is to create the electrical, optical, or microwave signal that represents the bits in each frame. These signals are then sent on the media one at a time.

It is also the job of the Physical layer to retrieve these individual signals from the media, restore them to their bit representations, and pass the bits up to the Data Link layer as a complete frame.

8.1.2 → Physical Layer – Operation

The media does not carry the frame as a single entity. The media carries signals, one at a time, to represent the bits that make up the frame.

There are three basic forms of network media on which data is represented:

- Copper cable
- Fiber
- Wireless

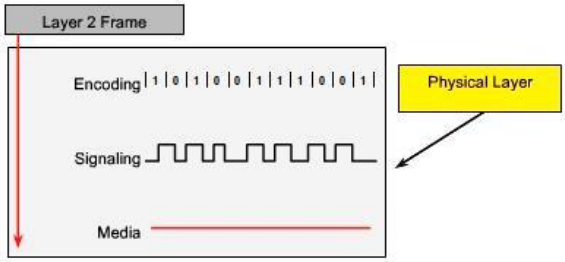
The representation of the bits - that is, the type of signal - depends on the type of media. For copper cable media, the signals are patterns of electrical pulses. For fiber, the signals are patterns of light. For wireless media, the signals are patterns of radio transmissions

8.1.4 → Physical Layer Fundamental Principles

The three fundamental functions of the Physical layer are:

- The physical components
- Data encoding
- Signaling

The physical elements are the electronic hardware devices, media and connectors that transmit and carry the signals to represent the bits.



Encoding

Encoding is a method of converting a stream of data bits into a predefined code. Codes are groupings of bits used to provide a predictable pattern that can be recognized by both the sender and the received. Using predictable patterns helps to distinguish data bits from control bits and provide better media error detection.

In addition to creating codes for data, encoding methods at the Physical layer may also provide codes for control purposes such as identifying the beginning and end of a frame. The transmitting host will transmit the specific pattern of bits or a code to identify the beginning and end of the frame.

Signaling

The Physical layer must generate the electrical, optical, or wireless signals that represent the "1" and "0" on the media. The method of representing the bits is called the signaling method. The Physical layer standards must define what type of signal represents a "1" and a "0". This can be as simple as a change in the level of an electrical signal or optical pulse or a more complex signaling method.

8.2.3 → Data Carrying Capacity:

Different physical media support the transfer of bits at different speeds. Data transfer can be measured in three ways:

- Bandwidth
- Throughput
- Goodput

Bandwidth

The capacity of a medium to carry data is described as the raw data bandwidth of the media. **Digital bandwidth measures the amount of information that can flow from one place to another in a given amount of time**. Bandwidth is typically measured in kilobits per second (kbps) or megabits per second (Mbps).

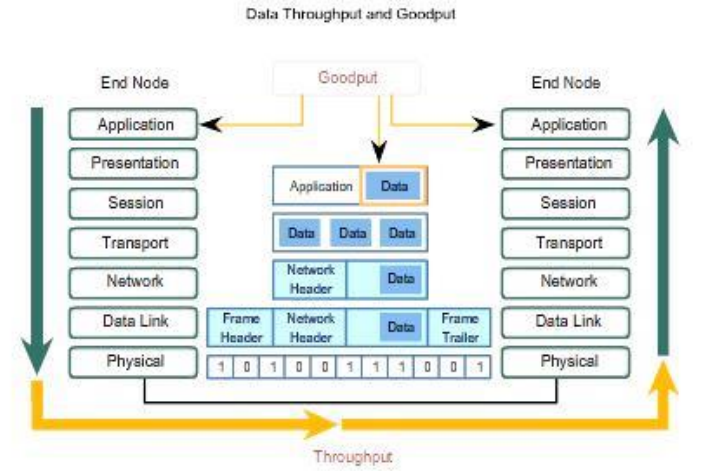
The practical bandwidth of a network is determined by a combination of factors: the properties of the physical media and the technologies chosen for signaling and detecting network signals.

Throughput

Throughput is the measure of the transfer of bits across the media over a given period of time. Due to a number of factors, throughput usually does not match the specified bandwidth in Physical layer implementations such as Ethernet.

Many factors influence throughput. Among these factors are the amount of traffic, the type of traffic, and the number of network devices encountered on the network being measured. In a multi-access 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 only take one segment in the path with low throughput to create a bottleneck to the throughput of the entire network.



Data throughput is actual network performance. Goodput is a measure of the transfer of usable data after protocol overhead traffic has been removed.

Goodput

A third measurement has been created to measure the transfer of usable data. That measure is known as goodput. Goodput is the measure of usable data transferred over a given period of time, and is therefore the measure that is of most interest to network users.

As shown in the figure, goodput measures the effective transfer of user data between Application layer entities, such as between a source web server process and a destination web browser device.

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, acknowledgements, and encapsulation.

As an example, consider two hosts on a LAN transferring a file. The bandwidth of the LAN is 100 Mbps. Due to 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 40Mbps.

8.3.2 → Types of Physical Media:

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.

Copper Media

The most commonly used media for data communications is cabling that uses copper wires to signal data and control bits between network devices. Cabling used for data communications usually consists of a series of individual copper wires that form circuits dedicated to specific signaling purposes.

Other types of copper cabling, known as coaxial cable, have a single conductor that runs through the center of the cable that is encased by, but insulated from, the other shield. The copper media type chosen is specified by the Physical layer standard required to link the Data Link layers of two or more network devices.

These cables can be used to connect nodes on a LAN to intermediate devices, such as routers and switches. Cables are also used to connect WAN devices to a data services provider such as a telephone company. Each type of connection and the accompanying devices have cabling requirements stipulated by Physical layer standards.

8.3.3 →Unshielded Twisted Pair (UTP) Cable

Unshielded twisted-pair (UTP) cabling, as it is used in Ethernet LANs, consists of four pairs of color-coded wires that have been twisted together and then encased in a flexible plastic sheath. As seen in the figure, the color codes identify the individual pairs and wires in the pairs and aid in cable termination.

UTP Cable Types

UTP cabling, terminated with RJ-45 connectors, is a common copper-based medium for interconnecting network devices, such as computers, with intermediate devices, such as routers and network switches.

Different situations may require UTP cables to be wired according to different wiring conventions. This means that the individual wires in the cable have to be connected in different orders to different sets of pins in the RJ-45 connectors. The following are main cable types that are obtained by using specific wiring conventions:

- Ethernet Straight-through
- Ethernet Crossover
- Rollover

Two other types of copper cable are used:

1. Coaxial

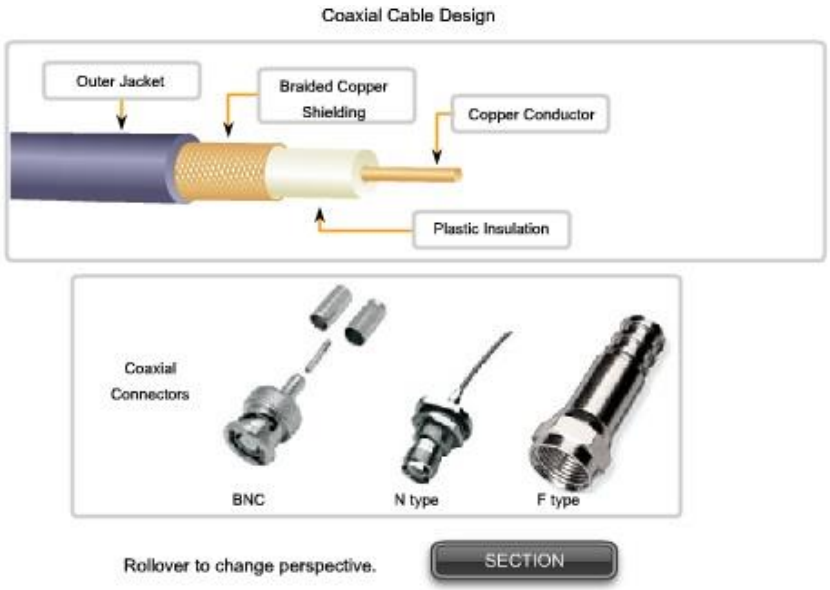
2. Shielded Twisted-Pair (STP)

Coaxial Cable

Coaxial cable consists of a copper conductor surrounded by a layer of flexible insulation, as shown in the figure.

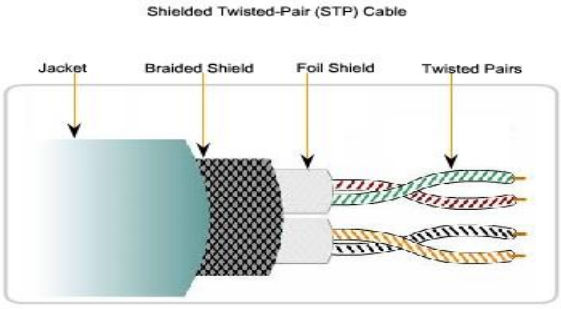
Over this insulating material is a woven copper braid, or metallic foil, that acts as the second wire in the circuit and as a shield for the inner conductor. This second layer, or shield, also reduces the amount of outside electromagnetic interference. Covering the shield is the cable jacket.

All the elements of the coaxial cable encircle the center conductor. Because they all share the same axis, this construction is called coaxial, or coax for short.



Shielded Twisted-Pair (STP) Cable

Another type of cabling used in networking is shielded twisted-pair (STP). As shown in the figure, STP uses two pairs of wires that are wrapped in an overall metallic braid or foil.

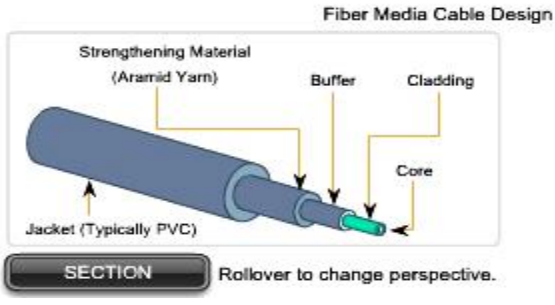


STP cable shields the entire bundle of wires within the cable as well as the individual wire pairs. STP provides better noise protection than UTP cabling, however at a significantly higher price.

For many years, STP was the cabling structure specified for use in Token Ring network installations. With the use of Token Ring declining, the demand for shielded twisted-pair cabling has also waned. The new 10 GB standard for Ethernet has a provision for the use of STP cabling. This may provide a renewed interest in shielded twisted-pair cabling.

8.3.6 –> Fiber Media

Fiber-optic cabling uses either glass or plastic fibers to guide light impulses from source to destination. The bits are encoded on the fiber as light impulses. Optical fiber cabling is capable of very large raw data bandwidth rates. Most current transmission standards have yet to approach the potential bandwidth of this media.



Generating and Detecting the Optical Signal

Either lasers or light emitting diodes (LEDs) generate the light pulses that are used to represent the transmitted data as bits on the media. Electronic semi-conductor devices called photodiodes detect the light pulses and convert them to voltages that can then be reconstructed into data frames.

Single-mode and Multimode Fiber

Fiber optic cables can be broadly classified into two types: single-mode and multimode.

Single-mode optical fiber carries a single ray of light, usually emitted from a laser. Because the laser light is uni-directional and travels down the center of the fiber, this type of fiber can transmit optical pulses for very long distances.

Multimode fiber typically uses LED emitters that do not create a single coherent light wave. Instead, light from an LED enters the multimode fiber at different angles. Because light entering the fiber at different angles takes different amounts of time to travel down the fiber, long fiber runs may result in the pulses becoming blurred on reception at the receiving end. This effect, known as modal dispersion, limits the length of multimode fiber segments.

Multimode fiber, and the LED light source used with it, are cheaper than single-mode fiber and its laser-based emitter technology.

8.3.7 → Wireless Media

The Wireless LAN

A common wireless data implementation is enabling devices to wirelessly connect via 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 adapters - 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.

8.3.8 → Media Connectors

Common Optical Fiber Connectors

Fiber-optic connectors come in a variety of types. The figure shows some of the most common:

Straight-Tip (ST) (trademarked by AT&T) - a very common bayonet style connector widely used with multimode fiber.

Subscriber Connector (SC) - a connector that uses a push-pull mechanism to ensure positive insertion. This connector type is widely used with single-mode fiber.

Lucent Connector (LC) - A small connector becoming popular for use with single-mode fiber and also supports multi-mode fiber.

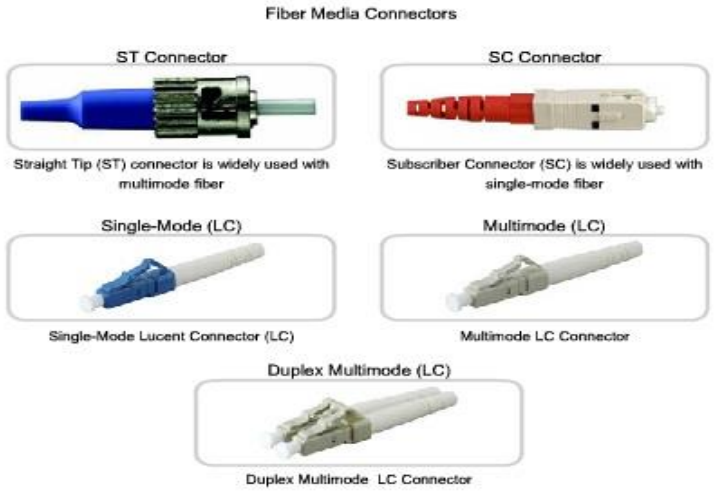
Terminating and splicing fiber-optic cabling requires special training and equipment. Incorrect termination of fiber optic media will result in diminished signaling distances or complete transmission failure.

Three common types of fiber-optic termination and splicing errors are:

- Misalignment - the fiber-optic media are not precisely aligned to one another when joined.
- End gap - the media do not completely touch at the splice or connection.
- End finish - the media ends are not well polished or dirt is present at the termination.

It is recommended that an Optical Time Domain Reflectometer (OTDR) be used to test each fiber-optic cable segment. This device injects a test pulse of light into the cable and measures back scatter and reflection of light detected as a function of time. The OTDR will calculate the approximate distance at which these faults are detected along the length of the cable.

A field test can be performed by shining a bright flashlight into one end of the fiber while observing the other end of the fiber. If light is visible, then the fiber is capable of passing light. Although this does not ensure the performance of the fiber, it is a quick and inexpensive way to find a broken fiber.



9.1.2 → Ethernet – Layer 1 and Layer 2

Ethernet operates across two layers of the OSI model. The model provides a reference to which Ethernet can be related but it is actually implemented in the lower half of the Data Link layer, which is known as the Media Access Control (MAC) sublayer, and the Physical layer only.

Ethernet at Layer 1 involves signals, bit streams that travel on the media, physical components that put signals on media, and various topologies. Ethernet Layer 1 performs a key role in the communication that takes place between devices, but each of its functions has limitations.

Layer 2 Addresses Layer 1 Limitations

Layer 1 Limitations	Layer 2 Functions
Cannot communicate with upper layers	Connects to upper layers via Logical Link Control (LLC)
Cannot identify devices	Uses addressing schemes to identify devices
Only recognizes streams of bits	Uses frames to organize bits into groups
Cannot determine the source of a transmission when multiple devices are transmitting	Uses Media Access Control (MAC) to identify transmission sources

As the figure shows, Ethernet at Layer 2 addresses these limitations. The Data Link sublayers contribute significantly to technological compatibility and computer communications. The MAC sublayer is concerned with the physical components that will be used to communicate the information and prepares the data for transmission over the media..

The Logical Link Control (LLC) sublayer remains relatively independent of the physical equipment that will be used for the communication process.

9.1.3 →Logical Link Control – Connecting to the Upper Layers

Ethernet separates the functions of the Data Link layer into two distinct sublayers: the Logical Link Control (LLC) sublayer and the Media Access Control (MAC) sublayer. The functions described in the OSI model for the Data Link layer are assigned to the LLC and MAC sublayers. The use of these sublayers contributes significantly to compatibility between diverse end devices.

For Ethernet, the IEEE 802.2 standard describes the LLC sublayer functions, and the 802.3 standard describes the MAC sublayer and the Physical layer functions. Logical Link Control handles the communication between the upper layers and the networking software, and the lower layers, typically the hardware. The LLC sublayer takes the network protocol data, which is typically an IPv4 packet, and adds control information to help deliver the packet to the destination node. Layer 2 communicates with the upper layers through LLC.

LLC is implemented in software, and its implementation is independent of the physical equipment. In a computer, the LLC can be considered the driver software for the Network Interface Card (NIC). The NIC driver is a program that interacts directly with the hardware on the NIC to pass the data between the media and the Media Access Control sublayer.

9.1.4 → MAC – Getting Data to the Media

Media Access Control (MAC) is the lower Ethernet sublayer of the Data Link layer. Media Access Control is implemented by hardware, typically in the computer Network Interface Card (NIC).

The Ethernet MAC sublayer has two primary responsibilities:

- Data Encapsulation
- Media Access Control

Data Encapsulation

Data encapsulation provides three primary functions:

- Frame delimiting
- Addressing
- Error detection

The data encapsulation process includes frame assembly before transmission and frame parsing upon reception of a frame. In forming the frame, the MAC layer adds a header and trailer to the Layer 3 PDU. The use of frames aids in the transmission of bits as they are placed on the media and in the grouping of bits at the receiving node.

The framing process provides important delimiters that are used to identify a group of bits that make up a frame. This process provides synchronization between the transmitting and receiving nodes.

The encapsulation process also provides for Data Link layer addressing. Each Ethernet header added in the frame contains the physical address (MAC address) that enables a frame to be delivered to a destination node.

An additional function of data encapsulation is error detection. Each Ethernet frame contains a trailer with a cyclic redundancy check (CRC) of the frame contents. After reception of a frame, the receiving node creates a CRC to compare to the one in the frame. If these two CRC calculations match, the frame can be trusted to have been received without error.

Media Access Control

The MAC sublayer controls the placement of frames on the media and the removal of frames from the media. As its name implies, it manages the media access control. This includes the initiation of frame transmission and recovery from transmission failure due to collisions.

Logical Topology

The underlying logical topology of Ethernet is a multi-access bus. This means that all the nodes (devices) in that network segment share the medium. This further means that all the nodes in that segment receive all the frames transmitted by any node on that segment.

Because all the nodes receive all the frames, each node needs to determine if a frame is to be accepted and processed by that node. This requires examining the addressing in the frame provided by the MAC address.

Ethernet provides a method for determining how the nodes share access to the media. The media access control method for classic Ethernet is Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

9.3.1 → The Frame – Encapsulating the Packet

9.3.5 → Ethernet Unicast, Multicast & Broadcast

9.4.2 → CSMA/CD – The Process:

Carrier Sense

In the CSMA/CD access method, all network devices that have messages to send must listen before transmitting.

If a device detects a signal from another device, it will wait for a specified amount of time before attempting to transmit.

When there is no traffic detected, a device will transmit its message. While this transmission is occurring, the device continues to listen for traffic or collisions on the LAN. After the message is sent, the device returns to its default listening mode.

Multi-access

If the distance between devices is such that the latency of one device's signals means that signals are not detected by a second device, the second device may start to transmit, too. The media now has two devices transmitting their signals at the same time. Their messages will propagate across the media until they encounter each other. At that point, the signals mix and the message is destroyed. Although the messages are corrupted, the jumble of remaining signals continues to propagate across the media.

Collision Detection

When a device is in listening mode, it can detect when a collision occurs on the shared media. The detection of a collision is made possible because all devices can detect an increase in the amplitude of the signal above the normal level.

Once a collision occurs, the other devices in listening mode - as well as all the transmitting devices - will detect the increase in the signal amplitude. Once detected, every device transmitting will continue to transmit to ensure that all devices on the network detect the collision.

Jam Signal and Random Backoff

Once the collision is detected by the transmitting devices, they send out a jamming signal. This jamming signal is used to notify the other devices of a collision, so that they will invoke a backoff algorithm. This backoff algorithm causes all devices to stop transmitting for a random amount of time, which allows the collision signals to subside.

After the delay has expired on a device, the device goes back into the "listening before transmit" mode. A random backoff period ensures that the devices that were involved in the collision do not try to send their traffic again at the same time, which would cause the whole process to repeat. But, this also means that a third device may transmit before either of the two involved in the original collision have a chance to re-transmit.

Hubs and Collision Domains

Given that collisions will occur occasionally in any shared media topology - even when employing CSMA/CD - we need to look at the conditions that can result in an increase in collisions. Because of the rapid growth of the Internet:

- More devices are being connected to the network.
- Devices access the network media more frequently.
- Distances between devices are increasing.

Recall that hubs were created as intermediary network devices that enable more nodes to connect to the shared media. Also known as multi-port repeaters, hubs retransmit received data signals to all connected devices, except the one from which it received the signals. Hubs do not perform network functions such as directing data based on addresses.

Hubs and repeaters are intermediary devices that extend the distance that Ethernet cables can reach. Because hubs operate at the Physical layer, dealing only with the signals on the media, collisions can occur between the devices they connect and within the hubs themselves.

Further, using hubs to provide network access to more users reduces the performance for each user because the fixed capacity of the media has to be shared between more and more devices.

The connected devices that access a common media via a hub or series of directly connected hubs make up what is known as a collision domain. A collision domain is also referred to as a network segment. Hubs and repeaters therefore have the effect of increasing the size of the collision domain.

The interconnection of hubs form a physical topology called an extended star. The extended star can create a greatly expanded collision domain.

An increased number of collisions reduces the network's efficiency and effectiveness until the collisions become a nuisance to the user.

Although CSMA/CD is a frame collision management system, it was designed to manage collisions for only limited numbers of devices and on networks with light network usage. Therefore, other mechanisms are required when large numbers of users require access and when more active network access is needed.

We will see that using switches in place of hubs can begin to alleviate this problem.

9.5.1 → Overview of Ethernet Physical Layer:

The differences between standard Ethernet, Fast Ethernet, Gigabit Ethernet, and 10 Gigabit Ethernet occur at the Physical layer, often referred to as the Ethernet PHY.

Ethernet is covered by the IEEE 802.3 standards. Four data rates are currently defined for operation over optical fiber and twisted-pair cables:

- 10 Mbps - 10Base-T Ethernet
- 100 Mbps - Fast Ethernet
- 1000 Mbps - Gigabit Ethernet
- 10 Gbps - 10 Gigabit Ethernet

9.6.1 →

Latency

Network latency is the amount of time it takes a signal to reach all destinations on the media. Each node in a hub-based network has to wait for an opportunity to transmit in order to avoid collisions. Latency can increase significantly as the distance between nodes is extended. Latency is also affected by a delay of the signal across the media as well as the delay added by the processing of the signals through hubs and repeaters. Increasing the length of media or the number of hubs and repeaters connected to a segment results in increased latency. With greater latency, it is more likely that nodes will not receive initial signals, thereby increasing the collisions present in the network.

Collisions

According to CSMA/CD, a node should not send a packet unless the network is clear of traffic. If two nodes send packets at the same time, a collision occurs and the packets are lost. Then both nodes send a jam signal, wait for a random amount of time, and retransmit their packets. Any part of the network where packets from two or more nodes can interfere with each other is considered a collision domain. A network with a larger number of nodes on the same segment has a larger collision domain and typically has more traffic. As the amount of traffic in the network increases, the likelihood of collisions increases.

Switches provide an alternative to the contention-based environment of classic Ethernet.

9.6.3 → Swithes – Selective Forwarding

Switch Operation

To accomplish their purpose, Ethernet LAN switches use five basic operations:

- Learning
- Aging
- Flooding
- Selective Forwarding
- Filtering

Learning

The MAC table must be populated with MAC addresses and their corresponding ports. The Learning process allows these mappings to be dynamically acquired during normal operation.

As each frame enters the switch, the switch examines the source MAC address. Using a lookup procedure, the switch determines if the table already contains an entry for that MAC address. If no entry exists, the switch creates a new entry in the MAC table using the source MAC address and pairs the address with the port on which the entry arrived. The switch now can use this mapping to forward frames to this node.

Aging

The entries in the MAC table acquired by the Learning process are time stamped. This timestamp is used as a means for removing old entries in the MAC table. After an entry in the MAC table is made, a procedure begins a countdown, using the timestamp as the beginning value. After the value reaches 0, the entry in the table will be refreshed when the switch next receives a frame from that node on the same port.

Flooding

If the switch does not know to which port to send a frame because the destination MAC address is not in the MAC table, the switch sends the frame to all ports except the port on which the frame arrived. The process of sending a frame to all segments is known as flooding. The switch does not forward the frame to the port on which it arrived because any destination on that segment will have already received the frame. Flooding is also used for frames sent to the broadcast MAC address.

Selective Forwarding

Selective forwarding is the process of examining a frame's destination MAC address and forwarding it out the appropriate port. This is the central function of the switch. When a frame from a node arrives at the switch for which the switch has already learned the MAC address, this address is matched to an entry in the MAC table and the frame is forwarded to the corresponding port. Instead of flooding the frame to all ports, the switch sends the frame to the destination node via its nominated port. This action is called forwarding.

Filtering

In some cases, a frame is not forwarded. This process is called frame filtering. One use of filtering has already been described: a switch does not forward a frame to the same port on which it arrived. A switch will also drop a corrupt frame. If a frame fails a CRC check, the frame is dropped. An additional reason for filtering a frame is security. A switch has security settings for blocking frames to and/or from selective MAC addresses or specific ports.

10.1.1 → Hub and Switch

Intranet work Devices

To create a LAN, we need to select the appropriate devices to connect the end device to the network. The two most common devices used are hubs and switches.

Hub

A hub receives a signal, regenerates it, and sends the signal over all ports. The use of hubs creates a logical bus. This means that the LAN uses multiaccess media. The ports use a shared bandwidth approach and often have reduced performance in the LAN due to collisions and recovery. Although multiple hubs can be interconnected, they remain a single collision domain.

Hubs are less expensive than switches. A hub is typically chosen as an intermediary device within a very small LAN, in a LAN that requires low throughput requirements, or when finances are limited.

Switch

A switch receives a frame and regenerates each bit of the frame on to the appropriate destination port. This device is used to segment a network into multiple collision domains. Unlike the hub, a switch reduces the collisions on a LAN. Each port on the switch creates a separate collision domain. This creates a point-to-point logical topology to the device on each port. Additionally, a switch provides dedicated bandwidth on each port, which can increase LAN performance. A LAN switch can also be used to interconnect network segments of different speeds.

In general, switches are chosen for connecting devices to a LAN. Although a switch is more expensive than a hub, its enhanced performance and reliability make it cost effective.

There is a range of switches available with a variety of features that enable the interconnection of multiple computers in a typical enterprise LAN setting.

10.1.2 → Device Selection Factors:

To meet user requirements, a LAN needs to be planned and designed. Planning ensures that all requirements, cost factors and deployment options are given due consideration.

When selecting a device for a particular LAN, there are a number of factors that need to be considered. These factors include, but are not limited to:

- Cost
- Speed and Types of Ports/Interfaces
- Expandability
- Manageability
- Additional Features and Services

10.2.2 → Making LAN Connections

Types of Interfaces

Typically, when connecting different types of devices, use a straight-through cable. And when connecting the same type of device, use a crossover cable.

Use straight-through cables for the following connectins:

- Switch to router
- Computer to switch
- Computer to hub

Use crossover cables directly connect the following device on a LAN :

- Switch to switch
- Switch to hub
- Hub to hub
- Router to router
- Computer to computer
- Computer to router

10.2.3 → Making WAN Connections

Data Communications Equipment and Data Terminal Equipment

The following terms describe the types of devices that maintain the link between a sending and a receiving device:

Data Communications Equipment (DCE) - A device that supplies the clocking services to another device. Typically, this device is at the WAN access provider end of the link.

Data Terminal Equipment (DTE) - A device that receives clocking services from another device and adjusts accordingly. Typically, this device is at the WAN customer or user end of the link.

If a serial connection is made directly to a service provider or to a device that provides signal clocking such as a channel service unit/data service unit (CSU/DSU), the router is considered to be data terminal equipment (DTE) and will use a DTE serial cable.

Be aware that there will be occasions, especially in our labs, when the local router is required to provide the clock rate and will therefore use a data communications equipment (DCE) cable.

DCEs and DTEs are used in WAN connections. The communication via a WAN connection is maintained by providing a clock rate that is acceptable to both the sending and the receiving device. In most cases, the telco or ISP provides the clocking service that synchronizes the transmitted signal.

10.5.1 → Device Interfaces

It is important to understand that Cisco devices, routers, and switches have several types of interfaces associated with them. You have worked with these interfaces in the labs. These interfaces, also commonly called ports, are where cables are connected to the device. See the figure for some example interfaces.

LAN Interfaces - Ethernet

The Ethernet interface is used for connecting cables that terminate with LAN devices such as computers and switches. This interface can also be used to connect routers to each other. This use will be covered in more detail in future courses.

Several conventions for naming Ethernet interfaces are popular, including AUI (older Cisco devices using a transceiver), Ethernet, FastEthernet and Fa 0/0. The name used depends on the type and model of the device.

WAN Interfaces - Serial

Serial WAN interfaces are used for connecting WAN devices to the CSU/DSU. A CSU/DSU is a device used to make the physical connection between data networks and WAN provider's circuits.

Serial interfaces between routers will also be used in our labs as part of various courses. For lab purposes, we will make a back-to-back connection between two routers using serial cables, and set a clock rate on one of the interfaces.

You may also need to configure other Data Link and Physical layer parameters on a router. To establish communication with a router via a console on a remote WAN, a WAN interface is assigned a Layer 3 address (IPv4 address).

Console Interface

The console interface is the primary interface for initial configuration of a Cisco router or switch. It is also an important means of troubleshooting. It is important to note that with physical access to the router's console interface, an unauthorized person can interrupt or compromise network traffic. Physical security of network devices is extremely important.

Auxiliary (AUX) Interface

This interface is used for remote management of the router. Typically, a modem is connected to the AUX interface for dial-in access. From a security standpoint, enabling the option to connect remotely to a network device carries with it the responsibility of maintaining vigilant device management.

11.1.1 → Cisco IOS

Access Methods

There are several ways to access the CLI environment. The most usual methods are:

- Console
- Telnet or SSH
- AUX port

Console

The CLI can be accessed through a console session, also known as the CTY line. A console uses a low speed serial connection to directly connect a computer or terminal to the console port on the router or switch.

The console port is a management port that provides out-of-band access to a router. The console port is accessible even if no networking services have been configured on the device. The console port is often used to access a device when the networking services have not been started or have failed.

Examples of console use are:

- The initial configuration of the network device
- Disaster recovery procedures and troubleshooting where remote access is not possible
- Password recovery procedures

When a router is first placed into service, networking parameters have not been configured. Therefore, the router cannot communicate via a network. To prepare for the initial startup and configuration, a computer running terminal emulation software is connected to the console port of the device. Configuration commands for setting up the router can be entered on the connected computer.

During operation, if a router cannot be accessed remotely, a connection to the console can enable a computer to determine the status of the device. By default, the console conveys the device startup, debugging, and error messages.

For many IOS devices, console access does not require any form of security, by default. However, the console should be configured with passwords to prevent unauthorized device access. In the event that a password is lost, there is a special set of procedures for bypassing the password and accessing the device. The device should be located in a locked room or equipment rack to prevent physical access.

Telnet and SSH

A method for remotely accessing a CLI session is to telnet to the router. Unlike the console connection, Telnet sessions require active networking services on the device. The network device must have at least one active interface configured with a Layer 3 address, such as an IPv4 address. Cisco IOS devices include a Telnet server process that launches when the device is started. The IOS also contains a Telnet client.

A host with a Telnet client can access the vty sessions running on the Cisco device. For security reasons, the IOS requires that the Telnet session use a password, as a minimum authentication method. The methods for establishing logins and passwords will be discussed in a later section.

The Secure Shell (SSH) protocol is a more secure method for remote device access. This protocol provides the structure for a remote login similar to Telnet, except that it utilizes more secure network services.

SSH provides stronger password authentication than Telnet and uses encryption when transporting session data. The SSH session encrypts all communications between the client and the IOS device. This keeps the user ID, password, and the details of the management session private. As a best practice, always use SSH in place of Telnet whenever possible.

Most newer versions of the IOS contain an SSH server. In some devices, this service is enabled by default. Other devices require the SSH server to be enabled.

IOS devices also include an SSH client that can be used to establish SSH sessions with other devices. Similarly, you can use a remote computer with an SSH client to start a secure CLI session. SSH client software is not provided by default on all computer operating systems. You may need to acquire, install, and configure SSH client software for your computer.

AUX

Another way to establish a CLI session remotely is via a telephone dialup connection using a modem connected to the router's AUX port. Similar to the console connection, this method does not require any networking services to be configured or available on the device.

The AUX port can also be used locally, like the console port, with a direct connection to a computer running a terminal emulation program. The console port is required for the configuration of the router, but not all routers have an auxiliary port. The console port is also preferred over the auxiliary port for troubleshooting because it displays router startup, debugging, and error messages by default.

Generally, the only time the AUX port is used locally instead of the console port is when there are problems using the console port, such as when certain console parameters are unknown.

11.1.2 → Configuration Files

Network devices depend on two types of software for their operation: operating system and configuration. Like the operating system in any computer, the operating system facilitates the basic operation of the device's hardware components.

Configuration files contain the Cisco IOS software commands used to customize the functionality of a Cisco device. Commands are parsed (translated and executed) by the Cisco IOS software when the system is booted (from the startup-config file) or when commands are entered in the CLI while in configuration mode.

A network administrator creates a configuration that defines the desired functionality of a Cisco device. The configuration file is typically a few hundred to a few thousand bytes in size.

Types of Configuration Files

A Cisco network device contains two configuration files:

- The running configuration file - used during the current operation of the device
- The startup configuration file - used as the backup configuration and is loaded when the device is started

A configuration file may also be stored remotely on a server as a backup.

Startup Configuration File

The startup configuration file (startup-config) is used during system startup to configure the device. The startup configuration file or startup-config file is stored in non-volatile RAM (NVRAM). Since NVRAM is non-volatile, when the Cisco device is turned off, the file remains intact. The startup-config files are loaded into RAM each time the router is started or reloaded. Once the configuration file is loaded into RAM, it is considered the running configuration or running-config.

Running Configuration

Once in RAM, this configuration is used to operate the network device.

The running configuration is modified when the network administrator performs device configuration. Changes to the running configuration will immediately affect the operation of the Cisco device. After making any changes, the administrator has the option of saving those changes back to the startup-config file so that they will be used the next time the device restarts.

Because the running configuration file is in RAM, it is lost if the power to the device is turned off or if the device is restarted. Changes made to the running-config file will also be lost if they are not saved to the startup-config file before the device is powered down.

11.1.3 → Cisco IOS Modes

The Cisco IOS is designed as a modal operating system. The term modal describes a system where there are different modes of operation, each having its own domain of operation. The CLI uses a hierarchical structure for the modes.

In order from top to bottom, the major modes are:

- User executive mode
- Privileged executive mode
- Global configuration mode
- Other specific configuration modes

Each mode is used to accomplish particular tasks and has a specific set of commands that are available when in that mode. For example, to configure a router interface, the user must enter interface configuration mode. All configurations that are entered in interface configuration mode apply only to that interface.

Some commands are available to all users; others can be executed only after entering the mode in which that command is available. Each mode is distinguished with a distinctive prompt, and only commands that are appropriate for that mode are allowed.

The hierarchical modal structure can be configured to provide security. Different authentication can be required for each hierarchal mode. This controls the level of access that network personnel can be granted.

Command Prompts

When using the CLI, the mode is identified by the command-line prompt that is unique to that mode. The prompt is composed of the words and symbols on the line to the left of the entry area. The word prompt is used because the system is prompting you to make an entry.

By default, every prompt begins with the device name. Following the name, the remainder of the prompt indicates the mode. For example, the default prompt for the global configuration mode on a router would be:

Router(config)#

Primary Modes

The two primary modes of operation are:

- User EXEC
- Privileged EXEC

As a security feature, the Cisco IOS software separates the EXEC sessions into two access modes. These two primary access modes are used within the Cisco CLI hierarchical structure.

Each mode has similar commands. However, the privileged EXEC mode has a higher level of authority in what it allows to be executed.

User Executive Mode

The user executive mode, or user EXEC for short, has limited capabilities but is useful for some basic operations. The user EXEC mode is at the top of the modal hierarchical structure. This mode is the first entrance into the CLI of an IOS router.

The user EXEC mode allows only a limited number of basic monitoring commands. This is often referred to as view-only mode. The user EXEC level does not allow the execution of any commands that might change the configuration of the device.

By default, there is no authentication required to access the user EXEC mode from the console. It is a good practice to ensure that authentication is configured during the initial configuration.

Moving between the User EXEC and Privileged EXEC Modes

The enable and disable commands are used to change the CLI between the user EXEC mode and the privileged EXEC mode, respectively.

In order to access the privileged EXEC mode, use the enable command. The privileged EXEC mode is sometimes called the enable mode.

The syntax for entering the enable command is:

Router>enable

This command is executed without the need for an argument or keyword. Once <Enter> is pressed, the router prompt changes to:

Router#

The # at the end of the prompt indicates that the router is now in privileged EXEC mode.

If password authentication has been configured for the privileged EXEC mode, the IOS prompts for the password.

The user EXEC mode is identified by the CLI prompt that ends with the > symbol. This is an example that shows the > symbol in the prompt:

Switch>

Privileged EXEC Mode

The execution of configuration and management commands requires that the network administrator use the privileged EXEC mode, or a specific mode further down the hierarchy.

The privileged EXEC mode can be identified by the prompt ending with the # symbol.

Switch#

By default, privileged EXEC does not require authentication. It is a good practice to ensure that authentication is configured.

Global configuration mode and all other more specific configuration modes can only be reached from the privileged EXEC mode. In a later section of this chapter, we will examine device configuration and some of the configuration modes.

Moving between the User EXEC and Privileged EXEC Modes

The enable and disable commands are used to change the CLI between the user EXEC mode and the privileged EXEC mode, respectively.

In order to access the privileged EXEC mode, use the enable command. The privileged EXEC mode is sometimes called the enable mode.

The syntax for entering the enable command is:

Router>enable

This command is executed without the need for an argument or keyword. Once <Enter> is pressed, the router prompt changes to:

Router#

The # at the end of the prompt indicates that the router is now in privileged EXEC mode.

If password authentication has been configured for the privileged EXEC mode, the IOS prompts for the password.

For example:

Router>enable

Password:

Router#

The disable command is used to return from the privileged EXEC to the user EXEC mode.

For example:

Router#disable

Router>

11.1.7 → IOS Configuration Modes

Global Configuration Mode

The primary configuration mode is called **global configuration** or **global config**. From global config, CLI configuration changes are made that affect the operation of the device as a whole.

We also use the global config mode as a precursor to accessing specific configuration modes.

The following CLI command is used to take the device from privileged EXEC mode to the global configuration mode and to allow entry of configuration commands from a terminal:

Router#configure terminal

Once the command is executed, the prompt changes to show that the router is in global configuration mode.

Router(config)#

Specific Configuration Modes

From the global config mode, there are many different configuration modes that may be entered. Each of these modes allows the configuration of a particular part or function of the IOS device. The list below shows a few of them:

- Interface mode - to configure one of the network interfaces (Fa0/0, S0/0/0,..)
- Line mode - to configure one of the lines (physical or virtual) (console, AUX, VTY,..)
- Router mode - to configure the parameters for one of the routing protocols

The figure shows the prompts for some modes. **Remember, as configuration changes are made within an interface or process, the changes only affect that interface or process.**

To exit a specific configuration mode and return to global configuration mode, enter exit at a prompt. To leave configuration mode completely and return to privileged EXEC mode, enter end or use the key sequence Ctrl-Z.

Once a change has been made from the global mode, it is good practice to save it to the startup configuration file stored in NVRAM. This prevents changes from being lost due to power failure or a deliberate restart. The command to save the running configuration to startup configuration file is:

Router#copy running-config startup-config

11.2.1 → Devices Need Names

Configure IOS Hostname

From the privileged EXEC mode, access the global configuration mode by entering the configure terminal command:

Router#configure terminal

After the command is executed, the prompt will change to:

Router(config)#

In the global mode, enter the hostname:

Router(config)#hostname AtlantaHQ

After the command is executed, the prompt will change to:

AtlantaHQ(config)#

Notice that the hostname appears in the prompt. To exit global mode, use the exit command.

Always make sure that your documentation is updated each time a device is added or modified. Identify devices in the documentation by their location, purpose, and address.

Note: To negate the effects of a command, preface the command with the no keyword.

For example, to remove the name of a device, use:

AtlantaHQ(config)# no hostname

Router(config)#

Notice that the no hostname command caused the router to revert to the default hostname of "Router."

11.2.2 → Limiting Device Access – Configuring Passwords and Using Banners

Physically limiting access to network devices with closets and locked racks is a good practice; however, passwords are the primary defense against unauthorized access to network devices. Every device should have locally configured passwords to limit access. In a later course, we will introduce how to strengthen security by requiring a userID along with a password. For now, we will present basic security precautions using only passwords.

As discussed previously, the IOS uses hierarchical modes to help with device security. As part of this security enforcement, the IOS can accept several passwords to allow different access privileges to the device.

The passwords introduced here are:

- **Console password -** limits device access using the console connection
- **Enable password -** limits access to the privileged EXEC mode
- **Enable secret password -** encrypted, limits access to the privileged EXEC mode
- **VTY password -** limits device access using Telnet

As good practice, use different authentication passwords for each of these levels of access. Although logging in with multiple and different passwords is inconvenient, it is a necessary precaution to properly protect the network infrastructure from unauthorized access.

Console Password

The console port of a Cisco IOS device has special privileges. The console port of network devices must be secured, at a bare minimum, by requiring the user to supply a strong password. This reduces the chance of unauthorized personnel physically plugging a cable into the device and gaining device access.

The following commands are used in global configuration mode to set a password for the console line:

Switch(config)#line console 0

Switch(config-line)#password password

Switch(config-line)#login

From global configuration mode, the command line console 0 is used to enter line configuration mode for the console. The zero is used to represent the first (and in most cases only) console interface for a router.

The second command, password password specifies a password on a line.

The login command configures the router to require authentication upon login. When login is enabled and a password set, there will be a prompt to enter a password.

Once these three commands are executed, a password prompt will appear each time a user attempts to gain access to the console port.

Enable and Enable Secret Passwords

To provide additional security, use the enable password command or the enable secret command. Either of these commands can be used to establish authentication before accessing privileged EXEC (enable) mode.

Always use the enable secret command, not the older enable password command, if possible. The enable secret command provides greater security because the password is encrypted. The enable password command can be used only if enable secret has not yet been set.

The enable password command would be used if the device uses an older copy of the Cisco IOS software that does not recognize the enable secret command.

The following commands are used to set the passwords:

Router(config)#enable passwordpassword

Router(config)#enable secret password

Note: If no enable password or enable secret password is set, the IOS prevents privileged EXEC access from a Telnet session.

Without an enable password having been set, a Telnet session would appear this way:

Switch>enable

% No password set

Switch>

VTY Password

The vty lines allow access to a router via Telnet. By default, many Cisco devices support five VTY lines that are numbered 0 to 4. A password needs to be set for all available vty lines. The same password can be set for all connections. However, it is often desirable that a unique password be set for one line to provide a fall-back for administrative entry to the device if the other connections are in use.

The following commands are used to set a password on vty lines:

Router(config)#line vty 0 4

Router(config-line)#passwordpassword

Router(config-line)#login

By default, the IOS includes the login command on the VTY lines. This prevents Telnet access to the device without first requiring authentication. If, by mistake, the no login command is set, which removes the requirement for authentication, unauthorized persons could connect to the line using Telnet. This would be a major security risk.

Encrypting Password Display

Another useful command prevents passwords from showing up as plain text when viewing the configuration files. This is the service password-encryption command.

This command causes the encryption of passwords to occur when a password is configured. The service password-encryption command applies weak encryption to all unencrypted passwords. This encryption does not apply to passwords as they are sent over media only in the configuration. The purpose of this command is to keep unauthorized individuals from viewing passwords in the configuration file.

If you execute the show running-config or show startup-config command prior to the service password-encryption command being executed, the unencrypted passwords are visible in the configuration output. The service password-encryption can then be executed and the encryption will be applied to the passwords. Once the encryption has been applied, removing the encryption service does not reverse the encryption.

Banner Messages

The creation of banners is a simple process; however, banners should be used appropriately. When a banner is utilized it should never welcome someone to the router. It should detail that only authorized personnel are allowed to access the device. Further, the banner can include scheduled system shutdowns and other information that affects all network users.

The IOS provides multiple types of banners. One common banner is the message of the day (MOTD). It is often used for legal notification because it is displayed to all connected terminals.

Configure MOTD using the banner motd command from global mode.

To configure a MOTD, from global configuration mode enter the banner motd command:

Switch(config)#banner motd # message #

Once the command is executed, the banner will be displayed on all subsequent attempts to access the device until the banner is removed.

11.2.3 →Managing Configuration Files

As we have discussed, modifying a running configuration affects the operation of the device immediately.

After making changes to a configuration, consider these options for the next step:

Make the changed configuration the new startup configuration.

- Return the device to its original configuration.
- Remove all configuration from the device.
- Make the Changed Configuration the New Startup Configuration

Remember, because the running configuration is stored in RAM, it is temporarily active while the Cisco device is running (powered on). If power to the router is lost or if the router is restarted, all configuration changes will be lost unless they have been saved.

Saving the running configuration to the startup configuration file in NVRAM preserves the changes as the new startup configuration.

Before committing to the changes, use the appropriate show commands to verify the device's operation. As shown in the figure, the show running-config command can be used to see a running configuration file.

When the changes are verified to be correct, use the copy running-config startup-config command at the privileged EXEC mode prompt. The following example shows the command:

Switch#copy running-config startup-config

Once executed, the running configuration file replaces the startup configuration file.

Return the Device to Its Original Configuration

If the changes made to the running configuration do not have the desired effect, it may become necessary to restore the device to its previous configuration. Assuming that we have not overwritten the startup configuration with the changes, we can replace the running configuration with the startup configuration. This is best done by restarting the device using the reload command at the privileged EXEC mode prompt.

When initiating a reload, the IOS will detect that the running config has changes that were not saved to startup configuration. A prompt will appear to ask whether to save the changes made. To discard the changes, enter n or no.

An additional prompt will appear to confirm the reload. To confirm, press the Enter key. Pressing any other key will abort the process.

For example:

Router#reload

System configuration has been modified. Save? [yes/no]: n

Proceed with reload? [confirm]

*Apr 13 01:34:15.758: %SYS-5-RELOAD: Reload requested by console. Reload Reason:

Backing Up Configurations Offline

Configuration files should be stored as backup files in the event of a problem. Configuration files can be stored on a Trivial File Transfer Protocol (TFTP) server, a CD, a USB memory stick, or a floppy disk stored in a safe place. A configuration file should also be included in the network documentation.

Backup Configuration on TFTP Server

```
Router#copy running-config tftp
Remote host []? 131.108.2.155
Name of configuration file to write[tokyo-config]?tokyo.2
Write file tokyo.2 to 131.108.2.155? [confirm] y
Writing tokyo.2 !!!!!! [OK]
```

As shown in the figure, one option is to save the running configuration or the startup configuration to a TFTP server. Use either the copy running-config tftp or copy startup-config tftp command and follow these steps:

1. Enter the copy running-config tftp command.
2. Enter the IP address of the host where the configuration file will be stored.
3. Enter the name to assign to the configuration file.
4. Answer yes to confirm each choice.

See the figure to view this process.

Removing All Configurations

If undesired changes are saved to the startup configuration, it may be necessary to clear all the configurations. This requires erasing the startup configuration and restarting the device.

The startup configuration is removed by using the erase startup-config command.

To erase the startup configuration file use erase NVRAM:startup-config or erase startup-config at the privileged EXEC mode prompt:

Router#erase startup-config

Once the command is issued, the router will prompt you for confirmation:

Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]

Confirm is the default response. To confirm and erase the startup configuration file, press the Enter key. Pressing any other key will abort the process.

After removing the startup configuration from NVRAM, reload the device to remove the current running configuration file from RAM. The device will then load the default startup configuration that was originally shipped with the device into the running configuration.

11.2.4 →Configuring Interfaces

Configuring Router Ethernet Interfaces

Router Ethernet interfaces are used as the gateways for the end devices on the LANs directly connected to the router.

Each Ethernet interface must have an IP address and subnet mask to route IP packets.

To configure an Ethernet interface follow these steps:

1. Enter global configuration mode.
2. Enter interface configuration mode.
3. Specify the interface address and subnet mask.
4. Enable the interface.

Configure the Ethernet IP address using the following commands:

Router(config)#interface FastEthernet 0/0

Router(config-if)#ip address ip_address netmask

Router(config-if)#no shutdown

Enabling the Interface

By default, interfaces are disabled. To enable an interface, enter the no shutdown command from the interface configuration mode. If an interface needs to be disabled for maintenance or troubleshooting, use the shutdown command.

Configuring Router Serial Interfaces

Serial interfaces are used to connect WANs to routers at a remote site or ISP.

To configure a serial interface follow these steps:

1. Enter global configuration mode.
2. Enter interface mode.

3. Specify the interface address and subnet mask.

4. Set the clock rate if a DCE cable is connected. Skip this step if a DTE cable is connected.

5. Turn on the interface.

Each connected serial interface must have an IP address and subnet mask to route IP packets.

Configure the IP address with the following commands:

Router(config)#interface Serial 0/0/0

Router(config-if)#ip address ip_address netmask

Serial interfaces require a clock signal to control the timing of the communications. In most environments, a DCE device such as a CSU/DSU will provide the clock. By default, Cisco routers are DTE devices, but they can be configured as DCE devices.

On serial links that are directly interconnected, as in our lab environment, one side must operate as DCE to provide a clocking signal. The clock is enabled and the speed is specified with the clock rate command. Some bit rates might not be available on certain serial interfaces. This depends on the capacity of each interface.

In the lab, if a clock rate needs to be set on an interface identified as DCE, use the 56000 clock rate.

The commands that are used to set a clock rate and enable a serial interface are:

Router(config)#interface Serial 0/0/0

Router(config-if)#clock rate 56000

Router(config-if)#no shutdown

Once configuration changes are made to the router, remember to use the show commands to verify the accuracy of the changes, and then save the changed configuration as the startup configuration.

As the hostname helps to identify the device on a network, an interface description indicates the purpose of the interface. A description of what an interface does or where it is connected should be part of the configuration of each interface. This description can be useful for troubleshooting.

The interface description will appear in the output of these commands: show startup-config, show running-config, and show interfaces.

For example, this description provides valuable information about the purpose of the interface:

Interface F0/0 is connected to the main switch in the administration building.

When support personnel can easily identify the purpose of an interface or connected device, they can more easily understand the scope of a problem, and this can lead to reaching a resolution sooner.

To create a description, use the command description. This example shows the commands used to create a description for a FastEthernet interface:

HQ-switch1#configure terminal

HQ-switch1(config)#interface fa0/0

HQ-switch1(config-if)#description Connects to main switch in Building A

Once the description is applied to the interface, use the show interfaces command to verify the description is correct.

Configuring a Switch Interface

A LAN switch is an intermediary device that interconnects segments within a network. Therefore, the physical interfaces on the switch do not have IP addresses. Unlike a router where the physical interfaces are connected to different networks, a physical interface on a switch connects devices within a network.

Switch interfaces are also enabled by default.

In order to be able to manage a switch, we assign addresses to the device to it. With an IP address assigned to the switch, it acts like a host device. Once the address is assigned, we access the switch with telnet, ssh or web services.

The address for a switch is assigned to a virtual interface represented as a Virtual LAN interface (VLAN). In most cases, this is the interface VLAN 1. we assign an IP address to the VLAN 1 interface. Like the physical interfaces of a router, we also must enable this interface with the no shutdown command.

Like any other host, the switch needs a gateway address defined to communicate outside of the local network.

we assign this gateway with the **ip default-gateway** command.

11.3.2 → Testing the Interface Assignment

In the same way that you use commands and utilities to verify a host configuration, you need to learn commands to verify the interfaces of intermediary devices. The IOS provides commands to verify the operation of router and switch interfaces.

Verifying the Router Interfaces

One of the most used commands is the show ip interface brief command. This provides a more abbreviated output than the show ip interface command. This provides a summary of the key information for all the interfaces.

Switch Connections

One additional tool that can be helpful is a mapping of how hosts are connected to a switch. This mapping can be obtained by issuing the show mac-address-table command.

Using a command line from a switch, enter the show command with the mac-address-table argument:

Sw1-2950#show mac-address-table