COMPUTER COMMUNICATION NETWORKS
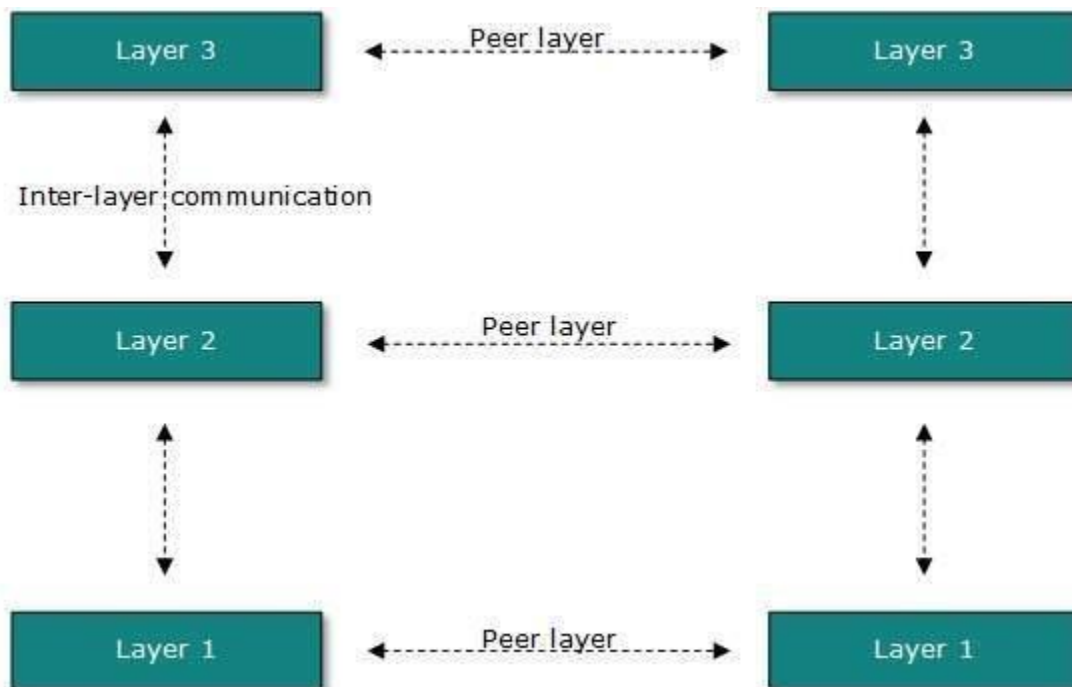
COMPUTER COMMUNICATION NETWORKS

UNIT 1 : INTRODUCTION TO NETWORKS

Networking engineering is a complicated task, which involves software, firmware, chip level engineering, hardware, and electric pulses. To ease network engineering, the whole networking concept is divided into multiple layers. Each layer is involved in some particular task and is independent of all other layers. But as a whole, almost all networking tasks depend on all of these layers. Layers share data between them and they depend on each other only to take input and send output.

Layered Tasks

In layered architecture of Network Model, one whole network process is divided into small tasks. Each small task is then assigned to a particular layer which works dedicatedly to process the task only. Every layer does only specific work.

In layered communication system, one layer of a host deals with the task done by or to be done by its peer layer at the same level on the remote host. The task is either initiated by layer at the lowest level or at the top most level. If the task is initiated by the-top most layer, it is passed on to the layer below it for further processing. The lower layer does the same thing, it processes the task and passes on to lower layer. If the task is initiated by lower most layer, then the reverse path is taken.

COMPUTER COMMUNICATION NETWORKS

Every layer clubs together all procedures, protocols, and methods which it requires to execute its piece of task. All layers identify their counterparts by means of encapsulation header and tail.

OSI reference model (Open Systems Interconnection)

OSI (Open Systems Interconnection) is reference model for how applications can communicate over a network. A reference model is a conceptual framework for understanding relationships. The purpose of the OSI reference model is to guide vendors and developers so the digital communication products and software programs they create will interoperate, and to facilitate clear comparisons among communications tools. Most vendors involved in telecommunications make an attempt to describe their products and services in relation to the OSI model. And although useful for guiding discussion and evaluation, OSI is rarely actually implemented, as few network products or standard tools keep all related functions together in well-defined layers as related to the model. The TCP/IP protocols, which define the Internet, do not map cleanly to the OSI model.

Developed by representatives of major computer and telecommunication companies beginning in 1983, OSI was originally intended to be a detailed specification of actual interfaces. Instead, the committee decided to establish a common reference model for which others could then develop detailed interfaces, which in turn could become standards. OSI was officially adopted as an international standard by the International Organization of Standards (ISO).

OSI layers

The main concept of OSI is that the process of communication between two endpoints in a telecommunication network can be divided into seven distinct groups of related functions, or layers. Each communicating user or program is at a computer that can provide those seven layers of function. So in a given message between users, there will be a flow of data down through the layers in the source computer, across the network and then up through the layers in the receiving computer. The seven layers of function are provided by a combination of applications, operating systems, network card device drivers and networking hardware that enable a system to put a signal on a network cable or out over Wi-Fi or other wireless protocol).

The seven Open Systems Interconnection layers are:

Layer 7: The application layer. This is the layer at which communication partners are identified (Is there someone to talk to?), network capacity is assessed (Will the network let me talk to them right now?), and that creates a thing to send or opens the thing received.   (This layer is not the application itself, it is the set of services an application should be able to make use of directly, although some applications may perform application layer functions.)

COMPUTER COMMUNICATION NETWORKS

Layer 6: The presentation layer. This layer is usually part of an operating system (OS) and converts incoming and outgoing data from one presentation format to another (for example, from clear text to encrypted text at one end and back to clear text at the other).

Layer 5: The session layer. This layer sets up, coordinates and terminates conversations. Services include authentication and reconnection after an interruption. On the Internet, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) provide these services for most applications.
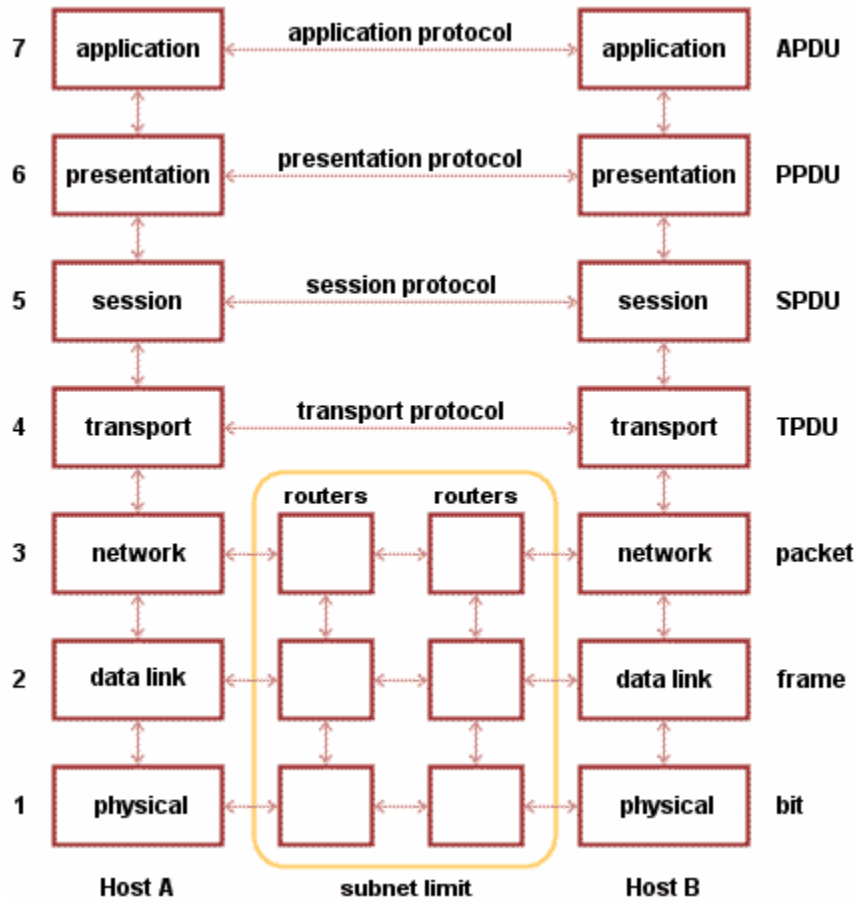
Layer 4: The transport layer. This layer manages packetization of data, then the delivery of the packets, including checking for errors in the data once it arrives. On the Internet, TCP and UDP provide these services for most applications as well.

Layer 3: The network layer. This layer handles the addressing and routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). IP is the network layer for the Internet.

Layer 2: The data-link layer. This layer sets up links across the physical network, putting packets into network frames. This layer has two sub-layers, the Logical Link Control Layer and the Media Access Control Layer. Ethernet is the main data link layer in use.

Layer 1: The physical layer. This layer conveys the bit stream through the network at the electrical, optical or radio level. It provides the hardware means of sending and receiving data on a carrier network.

COMPUTER COMMUNICATION NETWORKS

7	application	application protocol	application	APDU
6	presentation	presentation protocol	presentation	PPDU
5	session	session protocol	session	SPDU
4	transport	transport protocol	transport	TPDU
3	network	routers routers	network	packet
2	data link		data link	frame
1	physical		physical	bit
	Host A	subnet limit	Host B	

The TCP/IP Protocol Suite

The TCP/IP protocol suite maps to a four-layer conceptual model known as the DARPA model, which was named after the U.S. government agency that initially developed TCP/IP. The four layers of the DARPA model are: Application, Transport, Internet, and Network Interface. Each layer in the DARPA model corresponds to one or more layers of the seven-layer OSI model.

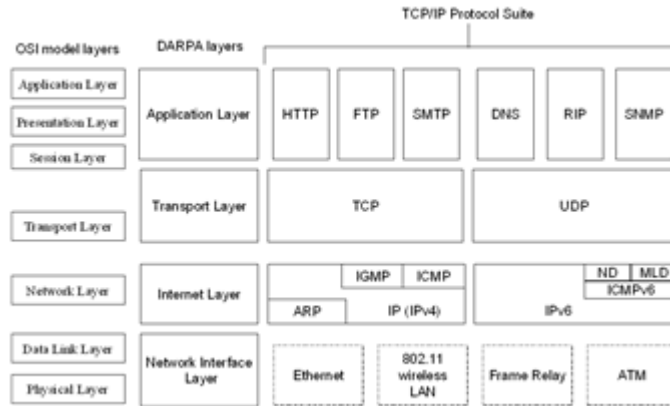Figure 2-1 shows the architecture of the TCP/IP protocol suite.

Figure 2-1 The architecture of the TCP/IP protocol suite

The TCP/IP protocol suite has two sets of protocols at the Internet layer:
- IPv4, also known as IP, is the Internet layer in common use today on private intranets and the Internet.
- IPv6 is the new Internet layer that will eventually replace the existing IPv4 Internet layer.

Network Interface Layer

The Network Interface layer (also called the Network Access layer) sends TCP/IP packets on the network medium and receives TCP/IP packets off the network medium. TCP/IP was designed to be independent of the network access method, frame format, and medium. Therefore, you can use TCP/IP to communicate across differing network types that use LAN technologies—such as Ethernet and 802.11 wireless LAN—and WAN technologies—such as Frame Relay and Asynchronous Transfer Mode (ATM). By being independent of any specific network technology, TCP/IP can be adapted to new technologies.

The Network Interface layer of the DARPA model encompasses the Data Link and Physical layers of the OSI model. The Internet layer of the DARPA model does not take advantage of sequencing and acknowledgment services that might be present in the Data Link layer of the OSI model. The Internet layer assumes an unreliable Network Interface layer and that reliable communications through session establishment and the sequencing and acknowledgment of packets is the responsibility of either the Transport layer or the Application layer.

Internet Layer

The Internet layer responsibilities include addressing, packaging, and routing functions. The Internet layer is analogous to the Network layer of the OSI model.
The core protocols for the IPv4 Internet layer consist of the following:
- The Address Resolution Protocol (ARP) resolves the Internet layer address to a Network Interface layer address such as a hardware address.

COMPUTER COMMUNICATION NETWORKS

- The Internet Protocol (IP) is a routable protocol that addresses, routes, fragments, and reassembles packets.
- The Internet Control Message Protocol (ICMP) reports errors and other information to help you diagnose unsuccessful packet delivery.
- The Internet Group Management Protocol (IGMP) manages IP multicast groups.

For more information about the core protocols for the IPv4 Internet layer, see "IPv4 Internet Layer" later in this chapter.

The core protocols for the IPv6 Internet layer consist of the following:

- IPv6 is a routable protocol that addresses and routes packets.
- The Internet Control Message Protocol for IPv6 (ICMPv6) reports errors and other information to help you diagnose unsuccessful packet delivery.
- The Neighbor Discovery (ND) protocol manages the interactions between neighboring IPv6 nodes.
- The Multicast Listener Discovery (MLD) protocol manages IPv6 multicast groups.

Transport Layer

The Transport layer (also known as the Host-to-Host Transport layer) provides the Application layer with session and datagram communication services. The Transport layer encompasses the responsibilities of the OSI Transport layer. The core protocols of the Transport layer are TCP and UDP.

TCP provides a one-to-one, connection-oriented, reliable communications service. TCP establishes connections, sequences and acknowledges packets sent, and recovers packets lost during transmission.

In contrast to TCP, UDP provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as the data that would fit into a single packet), when an application developer does not want the overhead associated with TCP connections, or when the applications or upper-layer protocols provide reliable delivery.

TCP and UDP operate over both IPv4 and IPv6 Internet layers.

Note The Internet Protocol (TCP/IP) component of Windows contains separate versions of the TCP and UDP protocols than the Microsoft TCP/IP Version 6 component does. The versions in the Microsoft TCP/IP Version 6 component are functionally equivalent to those provided with the Microsoft Windows NT® 4.0 operating systems and contain all the most recent security updates. The existence of separate protocol components with their own versions of TCP and UDP is known as a dual stack architecture. The ideal architecture is known as a dual IP layer, in

which the same versions of TCP and UDP operate over both IPv4 and IPv6 (as Figure 2-1 shows). Windows Vista has a dual IP layer architecture for the TCP/IP protocol components.

Application Layer

The Application layer allows applications to access the services of the other layers, and it defines the protocols that applications use to exchange data. The Application layer contains many protocols, and more are always being developed.

The most widely known Application layer protocols help users exchange information:

- The Hypertext Transfer Protocol (HTTP) transfers files that make up pages on the World Wide Web.
- The File Transfer Protocol (FTP) transfers individual files, typically for an interactive user session.
- The Simple Mail Transfer Protocol (SMTP) transfers mail messages and attachments.

Additionally, the following Application layer protocols help you use and manage TCP/IP networks:

- The Domain Name System (DNS) protocol resolves a host name, such as www.microsoft.com, to an IP address and copies name information between DNS servers.
- The Routing Information Protocol (RIP) is a protocol that routers use to exchange routing information on an IP network.
- The Simple Network Management Protocol (SNMP) collects and exchanges network management information between a network management console and network devices such as routers, bridges, and servers.

Windows Sockets and NetBIOS are examples of Application layer interfaces for TCP/IP applications.

IPv4 Internet Layer

The IPv4 Internet layer consists of the following protocols:

- ARP
- IP (IPv4)
- ICMP
- IGMP

The following sections describe each of these protocols in more detail.

ARP

When IP sends packets over a shared access, broadcast-based networking technology such as Ethernet or 802.11 wireless LAN, the protocol must resolve the media access control (MAC) addresses corresponding to the IPv4 addresses of the nodes to which the packets are being

forwarded, also known as the next-hop IPv4 addresses. As RFC 826 defines, ARP uses MAC-level broadcasts to resolve next-hop IPv4 addresses to their corresponding MAC addresses.

Based on the destination IPv4 address and the route determination process, IPv4 determines the next-hop IPv4 address and interface for forwarding the packet. IPv4 then hands the IPv4 packet, the next-hop IPv4 address, and the next-hop interface to ARP.

If the IPv4 address of the packet's next hop is the same as the IPv4 address of the packet's destination, ARP performs a direct delivery to the destination. In a direct delivery, ARP must resolve the IPv4 address of the packet's destination to its MAC address.

If the IPv4 address of the packet's next hop is not the same as the IPv4 address of the packet's destination, ARP performs an indirect delivery to a router. In an indirect delivery, ARP must resolve the IPv4 address of the router to its MAC address

To resolve the IPv4 address of a packet's next hop to its MAC address, ARP uses the broadcasting facility on shared access networking technologies (such as Ethernet or 802.11) to send out a broadcast ARP Request frame. In response, the sender receives an ARP Reply frame, which contains the MAC address that corresponds to the IPv4 address of the packet's next hop.

ARP Cache

To minimize the number of broadcast ARP Request frames, many TCP/IP protocol implementations incorporate an ARP cache, which is a table of recently resolved IPv4 addresses and their corresponding MAC addresses. ARP checks this cache before sending an ARP Request frame. Each interface has its own ARP cache.

Depending on the vendor implementation, the ARP cache can have the following qualities:
ARP cache entries can be dynamic (based on ARP replies) or static. Static ARP cache entries are permanent, and you add them manually using a TCP/IP tool, such as the Arp tool provided with Windows. Static ARP cache entries prevent nodes from sending ARP requests for commonly used local IPv4 addresses, such as those for routers and servers. The problem with static ARP cache entries is that you must manually update them when network adapter equipment changes. Dynamic ARP cache entries have time-out values associated with them so that they are removed from the cache after a specified period of time. For example, dynamic ARP cache entries for Windows are removed after no more than 10 minutes.

To view the ARP cache on a Windows–based computer, type **arp -a** at a command prompt. You can also use the Arp tool to add or delete static ARP cache entries.

ARP Process
When sending the initial packet as the sending host or forwarding the packet as a router, IPv4 sends the IPv4 packet, the next-hop IPv4 address, and the next-hop interface to ARP. Whether performing a direct or indirect delivery, ARP performs the following process:

Based on the next-hop IPv4 address and interface, ARP checks the appropriate ARP cache for an entry that matches the next-hop IPv4 address. If ARP finds an entry, ARP skips to step 6.

If ARP does not find an entry, ARP builds an ARP Request frame. This frame contains the MAC and IPv4 addresses of the interface from which the ARP request is being sent and the IPv4 packet's next-hop IPv4 address. ARP then broadcasts the ARP Request frame from the appropriate interface.
All nodes on the subnet receive the broadcasted frame and process the ARP request. If the next-hop address in the ARP request corresponds to the IPv4 address assigned to an interface on the subnet, the receiving node updates its ARP cache with the IPv4 and MAC addresses of the ARP requestor. All other nodes silently discard the ARP request.

The receiving node that is assigned the IPv4 packet's next-hop address formulates an ARP reply that contains the requested MAC address and sends the reply directly to the ARP requestor.

When the ARP requestor receives the ARP reply, the requestor updates its ARP cache with the address mapping. With the exchange of the ARP request and the ARP reply, both the ARP requestor and ARP responder have each other's address mappings in their ARP caches.

The ARP requestor sends the IPv4 packet to the next-hop node by addressing it to the resolved MAC address.

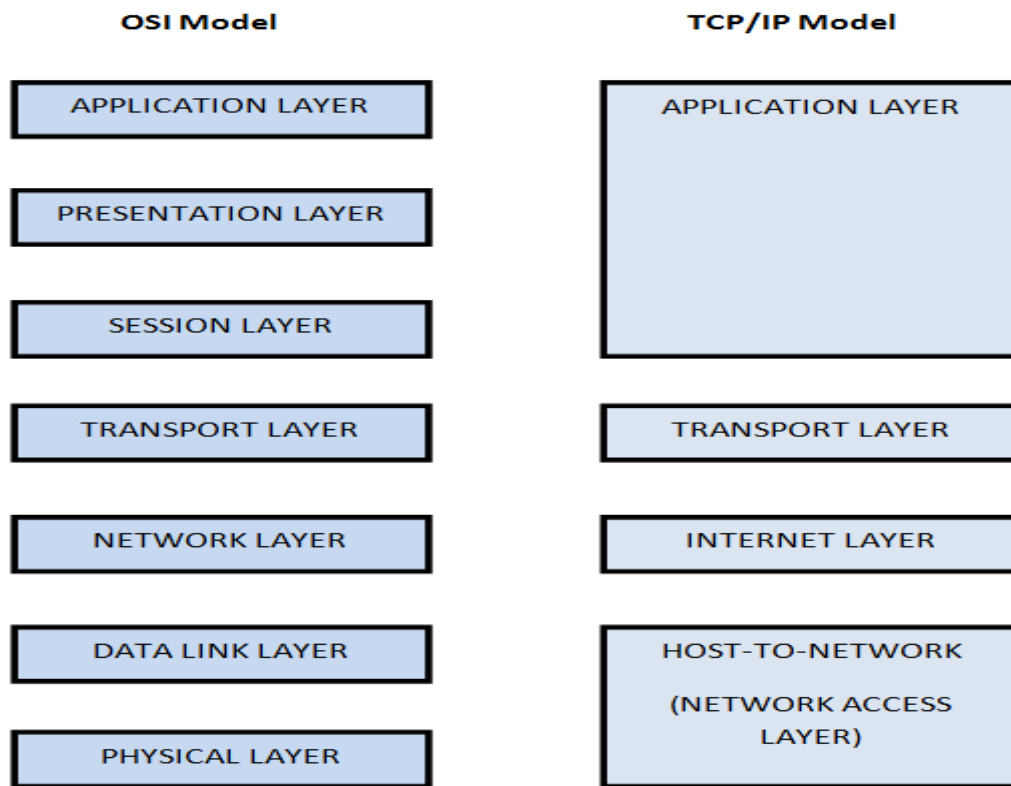## Comparison of OSI Reference Model and TCP/IP Reference Model

Following are some major differences between OSI Reference Model and TCP/IP Reference Model, with diagrammatic comparison below.

OSI(Open System Interconnection)	TCP/IP(Transmission Control Protocol / Internet Protocol)
1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
2. In OSI model the transport layer	2. In TCP/IP model the transport layer does

guarantees the delivery of packets.	not guarantees delivery of packets. Still the TCP/IP model is more reliable.
3. Follows vertical approach.	3. Follows horizontal approach.
4. OSI model has a separate Presentation layer and Session layer.	4. TCP/IP does not have a separate Presentation layer or Session layer.
5. OSI is a reference model around which the networks are built. Generally it is used as a guidance tool.	5. TCP/IP model is, in a way implementation of the OSI model.
6. Network layer of OSI model provides both connection oriented and connectionless service.	6. The Network layer in TCP/IP model provides connectionless service.
7. OSI model has a problem of fitting the protocols into the model.	7. TCP/IP model does not fit any protocol
8. Protocols are hidden in OSI model and are easily replaced as the technology changes.	8. In TCP/IP replacing protocol is not easy.
9. OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent.	9. In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent.
10. It has 7 layers	10. It has 4 layers

COMPUTER COMMUNICATION NETWORKS

OSI Model

| APPLICATION LAYER |
| PRESENTATION LAYER |
| SESSION LAYER |
| TRANSPORT LAYER |
| NETWORK LAYER |
| DATA LINK LAYER |
| PHYSICAL LAYER |

TCP/IP Model

| APPLICATION LAYER |
| TRANSPORT LAYER |
| INTERNET LAYER |
| HOST-TO-NETWORK (NETWORK ACCESS LAYER) |

Using Telephone and Cable Networks for Data Transmission

The telephone, which is referred to as the plain old telephone system (POTS), was originally an analog system. During the last decade, the telephone network has undergone many technical changes. The network is now digital as well as analog.

A home computer can access the Internet through the existing telephone system or through a cable TV system.

The telephone network is made of three major components: local loops, trunks, and switching offices. It has several levels of switching offices such as end offices, tandem offices, and regional offices.

Telephone companies provide two types of services: analog and digital.
The United States is divided into many local access transport areas (LATAs). The services offered inside a LATA are called intra-LATA services. The carrier that handles these services is called a local exchange carrier (LEC). The services between LATAs are handled by interexchange carriers (lXCs).

COMPUTER COMMUNICATION NETWORKS

A LATA is a small or large metropolitan area that according to the divestiture of 1984 was under the control of a single telephone-service provider.

In in-band signaling, the same circuit is used for both signaling and data. In out-of band signaling, a portion of the bandwidth is used for signaling and another portion for data. The protocol that is used for signaling in the telephone network is called Signaling System Seven (SS7).

- Telephone companies provide two types of services: analog and digital. We can categorize analog services as either analog switched services or analog leased services. The two most common digital services are switched/56 service and digital data service (DDS).

- Data transfer using the telephone local loop was traditionally done using a dial-up modem. The term modem is a composite word that refers to the two functional entities that make up the device: a signal modulator and a signal demodulator.

- Most popular modems available are based on the V-series standards. The V.32 modem has a data rate of 9600 bps. The V32bis modem supports 14,400-bps transmission. V90 modems, called 56K modems, with a downloading rate of 56 kbps and uploading rate of 33.6 kbps are very common. The standard above V90 is called V92. These modems can adjust their speed, and if the noise allows, they can upload data at the rate of 48 kbps.

- Telephone companies developed another technology, digital subscriber line (DSL), to provide higher-speed access to the Internet. DSL technology is a set of technologies, each differing in the first letter (ADSL, VDSL, HDSL, and SDSL. ADSL provides higher speed in the downstream direction than in the upstream direction. The high-bitrate digital subscriber line (HDSL) was designed as an alternative to the T-l line (1.544 Mbps). The symmetric digital subscriber line (SDSL) is a one twisted-pair version of HDSL. The very high-bit-rate digital subscriber line (VDSL) is an alternative approach that is similar to ADSL.

- DSL supports high-speed digital communications over the existing telephone local loops.

- ADSL technology allows customers a bit rate of up to 1 Mbps in the upstream direction and up to 8 Mbps in the downstream direction.

- ADSL uses a modulation technique called DMT which combines QAM and FDM.

COMPUTER COMMUNICATION NETWORKS

- ADSL is an asymmetric communication technology designed for residential users; it is not suitable for businesses.

- ADSL is an adaptive technology. The system uses a data rate based on the condition of the local loop line.

- SDSL, HDSL, and VDSL are other DSL technologies.

- Theoretically, the coaxial cable used for cable TV allows Internet access with a bit rate of up to 12 Mbps in the upstream direction and up to 30 Mbps in the downstream direction.

- An HFC network allows Internet access through a combination of fiber-optic and coaxial cables.

- The coaxial cable bandwidth is divided into a video band, a downstream data band, and an upstream data band. Both upstream and downstream bands are shared among subscribers.

- DOCSIS defines all protocols needed for data transmission on an HFC network.

- Synchronous Optical Network (SONET) is a synchronous high-data-rate TDM network for fiber-optic networks.

- SONET has defined a hierarchy of signals (similar to the DS hierarchy) called synchronous transport signals (STSs).

- Optical carrier (OC) levels are the implementation of STSs.

- A SONET frame can be viewed as a matrix of nine rows of 90 octets each.

- SONET is backward compatible with the current DS hierarchy through the virtual tributary (VT) concept. VT's are a partial payload consisting of an m-by-n block of octets. An STS payload can be a combination of several VT's.

- STSs can be multiplexed to get a new STS with a higher data range.
- Community antenna TV (CATV) was originally designed to provide video services for the community. The traditional cable TV system used coaxial cable end to end. The second generation of cable networks is called a hybrid fiber-coaxial (HFC) network. The network uses a combination of fiber-optic and coaxial cable.
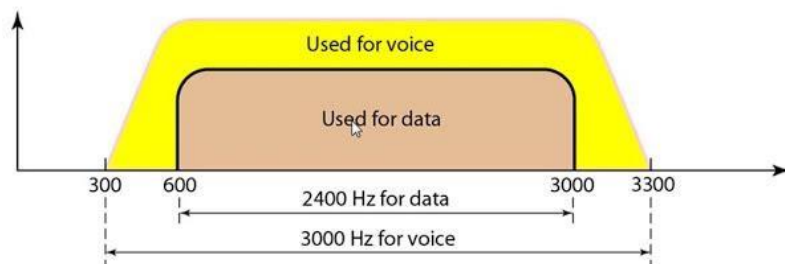
COMPUTER COMMUNICATION NETWORKS

- Communication in the traditional cable TV network is unidirectional.

- Communication in an HFC cable TV network can be bidirectional.

- To provide Internet access, the cable company has divided the available bandwidth of the coaxial cable into three bands: video, downstream data, and upstream data. The downstream-only video band occupies frequencies from 54 to 550 MHz. The downstream data occupies the upper band, from 550 to 750 MHz. The upstream data occupies the lower band, from 5 to 42 MHz.

In a telephone network, the telephone numbers of the caller and callee are serving as source and destination addresses. These are used only during the setup (dialing) and teardown (hanging-up)phases.

Three Major Components of Telephone System

- Local loops - a twisted-pair cable that connects the subscriber telephone to the nearest end office or local central office. The local loop, when used for voice, has a bandwidth of 4000 Hz (4 kHz). The existing local loops can handle bandwidths up to 1.1 MHz.

- Trunks - transmission media that handle the communication between offices. A trunk normally handles hundreds or thousands of connections through multiplexing. Transmission is usually through optical fibers or satellite links.

- Switching offices - A switch connects several local loops or trunks and allows a connection between different subscribers.

Telephone Line Bandwidth


Bandwidth division in ADSL

COMPUTER COMMUNICATION NETWORKS
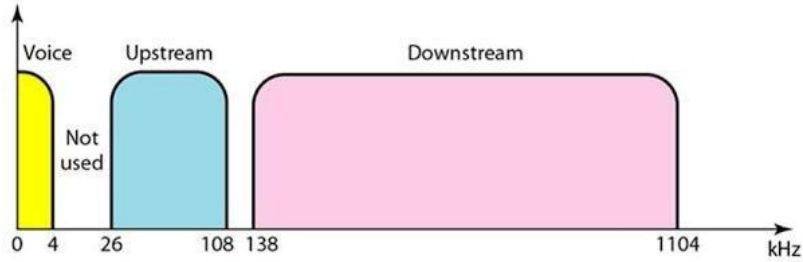
Summary of DSL technologies

Technology	Downstream Rate	Upstream Rate	Distance (ft)	Twisted Pairs	Line Code
ADSL	1.5–6.1 Mbps	16–640 kbps	12,000	1	DMT
ADSL Lite	1.5 Mbps	500 kbps	18,000	1	DMT
HDSL	1.5–2.0 Mbps	1.5–2.0 Mbps	12,000	2	2B1Q
SDSL	768 kbps	768 kbps	12,000	1	2B1Q
VDSL	25–55 Mbps	3.2 Mbps	3000–10,000	1	DMT

Division of coaxial cable band by CATV
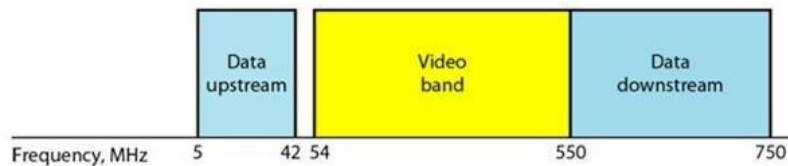
A SONET system can use the following equipment:

STS multiplexer - combines several optical signals to make an STS signal.
Regenerator - removes noise from an optical signal.
Add/drop multiplexer - adds STSs from different paths and removes STSs from a path.

Modem (also called a 'dial-up modem')

This is a piece of equipment used for sending and receiving data from one computer to another computer using the existing phone network.

A typical dial-up connection.
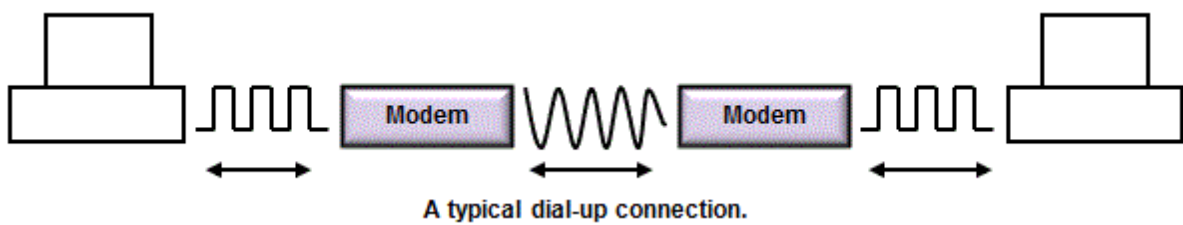
COMPUTER COMMUNICATION NETWORKS

A modem works by taking the packets of data (which are digital signals) from a computer and converting them into analogue signals, which the phone network uses. The analogue signals then pass along the phone network from computer to computer, from network to network, until the final destination is reached. At the destination, another modem converts back the analogue signals into the original digital ones and then passes these to the destination computer.

Advantages and disadvantages of modems

In this way, the existing, widespread phone network, which only uses analogue signals, can be used by computers, which are digital devices. On the other hand, modems cannot send large volumes of data at once. They are therefore of little use when streaming files, and the time taken to download files such as music or films may make them frustrating. In addition, the phone line, just like making a phone call, is engaged whilst the modem is in use. This is why it is also known as a 'dial-up' modem; you have to dial up the phone number provided by your **Internet Service Provider** (ISP) to make a connection to the Internet and all the time you are on the phone using the Internet, your phone cannot be used for other things, such as making and receiving phone calls or sending faxes.

DIGITAL SUBSCRIBER LINE

After traditional modems reached their peak data rate, telephone companies developed another technology, DSL, to provide higher-speed access to the Internet. Digital subscriber line (DSL) technology is one of the most promising for supporting high-speed digital communication over the existing local loops.

The original DSL system was designed to produce data rates of 1.5 Mbits/s to 8 Mbits/s downstream from the telephone company to the subscriber and a lower rate upstream. Most Internet access involves more downloading and less uploading of data. The resulting design is referred to as asymmetrical DSL or ADSL. Most DSL formats are asymmetrical, although there are DSL variations that deliver the same rates in both directions.

The great attenuation, noise, and crosstalk problems of bundling multiple twisted-pair lines are the primary limitations of the POTS. These lines are effectively long low-pass filters with upper frequency limits that reduce the bandwidth of the line and limit the data rate that can be achieved. Line bandwidth is a function of the length of the UTP. Shorter cable runs have wider bandwidths, so they are clearly more capable of high data rates than longer runs. But despite this limitation, developments in digital signal processing have made this once limited communications medium capable of high-speed data delivery.

COMPUTER COMMUNICATION NETWORKS

Cable TV Systems

Cable TV systems were developed to provide reliable TV service to local communities. Along with the hundreds of TV channels available, cable companies offer services such as high-speed Internet access. Some even offer voice over IP (VoIP) telephone service. Cable companies usually offer a "triple-play" package that bundles TV, phone, and Internet services.

Systems have been upgraded from pure analog transmission to digital. Early systems were based on coax cable, but today the most common configuration is fiber-optic cable and coax. Hybrid fiber coax is one of the most common configurations (Fig. 1).
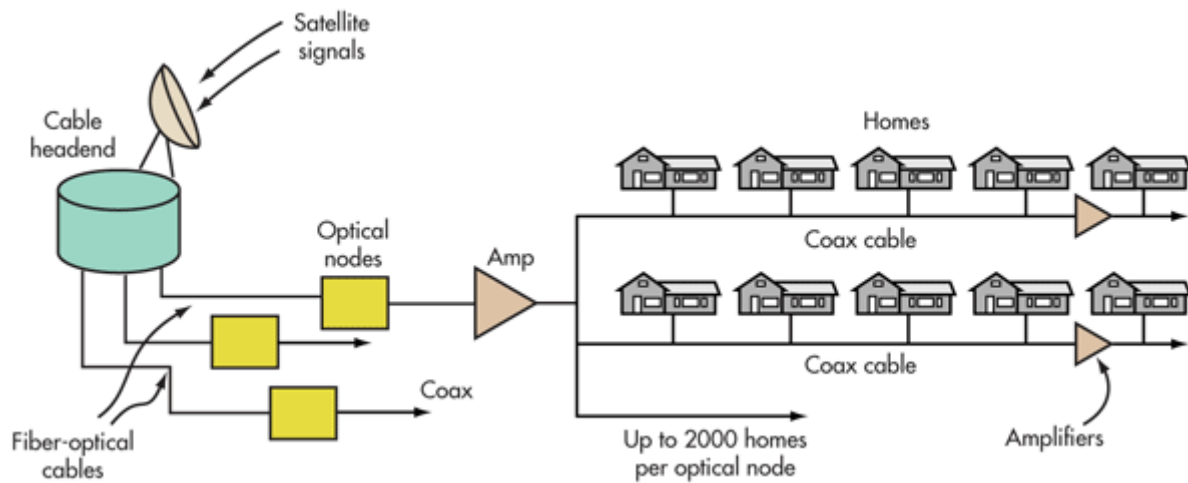
1. The typical hybrid fiber coax (HFC) cable TV distribution system used throughout the consists of fiber-optic cable to neighborhood nodes that then distribute the signals to homes with RG-6/U coax.

All of the services originate from the cable company's facilities, known as the headend, where the company collects the video from local TV stations and cable TV programming suppliers via satellite. The company then packages multiple channels into bundles for basic cable as well as two or three other options of premium movie and/or sports channels. The headend also has an interconnection to the Internet, where it can supply Internet services or connect to a separate Internet service provider.

The headend connects to the end user via a network of fiber-optic and coax cables. The TV channels and Internet channels are frequency multiplexed and modulated on to the main fiber-optic cable for transport out to distribution hubs that rejuvenate the signals over longer cable runs. From the one or more distribution hubs, the signal travels to multiple optical nodes located in various city or suburban neighborhoods. In a typical configuration, a   single
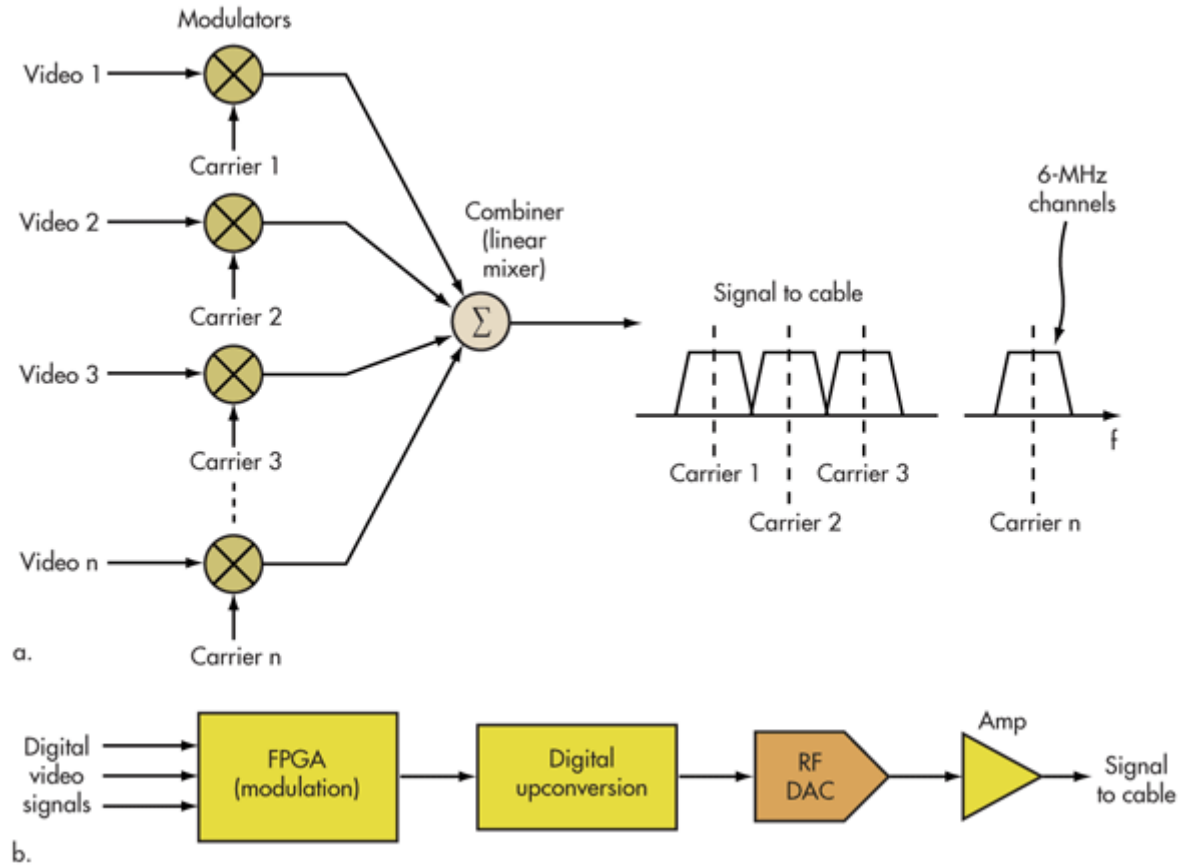
fiber is split to serve four fiber optical nodes. Most fiber nodes serve up to 500 homes. With this arrangement, each fiber serves up to 2000 homes, although not all homes passed have a cable modem or service.

The optical nodes convert the optical signals into electrical signals for the final distribution via coax cable. The most common cable is RG-6/U 75-ohm coax using F-type connectors. All of the homes receive the same signal, just like a bus network topology. In some areas with longer distances, amplifiers are added along the way to mitigate the large cable losses that are common.

All of the TV signals and Internet data are transmitted in a spectrum of 6-MHz wide channels. Since a coax cable has a bandwidth as wide as 850 MHz to 1 GHz, the system can accommodate from 140 to 170 downstream channels of 6 MHz each. The TV signals or Internet data are modulated on to carriers in each channel. There are also upstream channels that allow the consumer to transmit data back to the headend. This communication takes places in 6-MHz channels as well that occupy the cable spectrum from 5 MHz to 40 MHz or in some systems up to 65 MHz.

The composite video signal is developed in equipment called the cable modem termination system (CMTS). In older systems, the video information is modulated on to the 6-MHz channel carriers and then all channels are combined or linearly mixed to form the composited cable signal (Fig. 2a). However, today it's possible to synthesize a full block of modulated channels digitally. The digitized video is sent to an ASIC or FPGA programmed to produce the desired quadrature amplitude modulation (QAM) for each channel (Fig. 2b). The signals are then digitally upconverted to the final frequency and sent to a wideband digital-to-analog converter (DAC) that produces the composite multi-channel signal to be sent to the cable.

COMPUTER COMMUNICATION NETWORKS

The original DSL system was designed to produce data rates of 1.5 Mbits/s to 8 Mbits/s downstream from the telephone company to the subscriber and a lower rate upstream. Most Internet access involves more downloading and less uploading of data. The resulting design is referred to as asymmetrical DSL or ADSL. Most DSL formats are asymmetrical, although there are DSL variations that deliver the same rates in both directions.

The great attenuation, noise, and crosstalk problems of bundling multiple twisted-pair lines are the primary limitations of the POTS. These lines are effectively long low-pass filters with upper frequency limits that reduce the bandwidth of the line and limit the data rate that can be achieved. Line bandwidth is a function of the length of the UTP. Shorter cable runs have wider bandwidths, so they are clearly more capable of high data rates than longer runs. But despite this limitation, developments in digital signal processing have made this once limited communications medium capable of high-speed data delivery.

COMPUTER COMMUNICATION NETWORKS

UNIT 2 :

DATA LINK CONTROL - OSI Model

Data link layer is most reliable node to node delivery of data. It forms frames from the packets that are received from network layer and gives it to physical layer. It also synchronizes the information which is to be transmitted over the data. Error controlling is easily done. The encoded data are then passed to physical.

Error detection bits are used by the data link layer. It also corrects the errors. Outgoing messages are assembled into frames. Then the system waits for the acknowledgements to be received after the transmission. It is reliable to send message.

FUNCTIONS OF DATA LINK LAYER:

- Framing: Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.

- Physical Addressing: The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.

- Flow Control: A flow control mechanism to avoid a fast transmitter from running a slow receiver by buffering the extra bit is provided by flow control. This prevents traffic jam at the receiver side.

- Error Control: Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

- Access Control: Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

COMPUTER COMMUNICATION NETWORKS
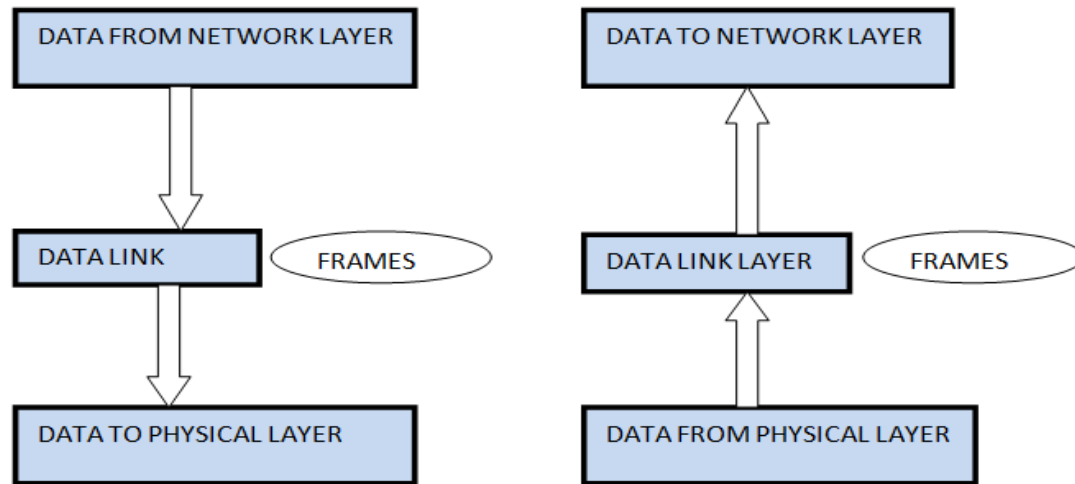
Data   Link   Layer   -   Creating   a   Frame

The description of a frame is a key element of each Data Link layer protocol. Data Link layer protocols require control information to enable the protocols to function. Control information may tell:

- Which nodes are in communication with each other
- When communication between individual nodes begins and when it ends
- Which errors occurred while the nodes communicated
- Which nodes will communicate next

The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame. Unlike the other PDUs that have been discussed in this course, the Data Link layer frame includes:

Data - The packet from the Network layer

Header - Contains control information, such as addressing, and is located at the beginning of the PDU

Trailer - Contains control information added to the end of the PDU

COMPUTER COMMUNICATION NETWORKS

Data Link Layer Services
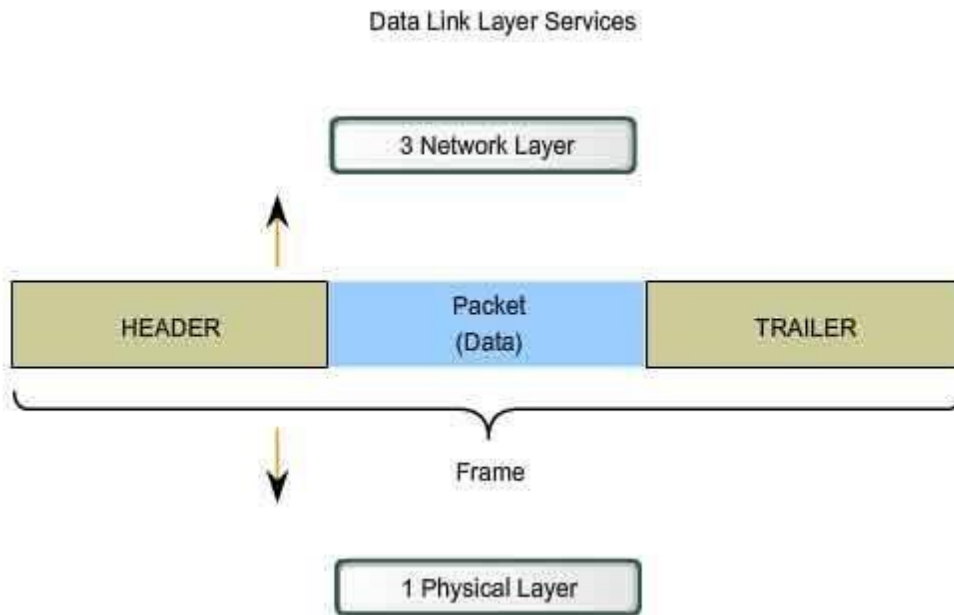
Formatting Data for Transmission

When data travels on the media, it is converted into a stream of bits, or 1s and 0s. If a node is receiving long streams of bits, how does it determine where a frame starts and stops or which bits represent the address?

Framing breaks the stream into decipherable groupings, with control information inserted in the header and trailer as values in different fields. This format gives the physical signals a structure that can be received by nodes and decoded into packets at the destination.

Typical field types include:

Start and stop indicator fields - The beginning and end limits of the frame
Naming or addressing fields

Type field - The type of PDU contained in the frame
Quality - control fields

A data field -The frame payload (Network layer packet)

Fields at the end of the frame form the trailer. These fields are used for error detection and mark the end of the frame. Not all protocols include all of these fields. The standards for a specific Data Link protocol define the actual frame format. Examples of frame formats will be discussed at the end of this chapter.

COMPUTER COMMUNICATION NETWORKS

Formatting Data for Transmission
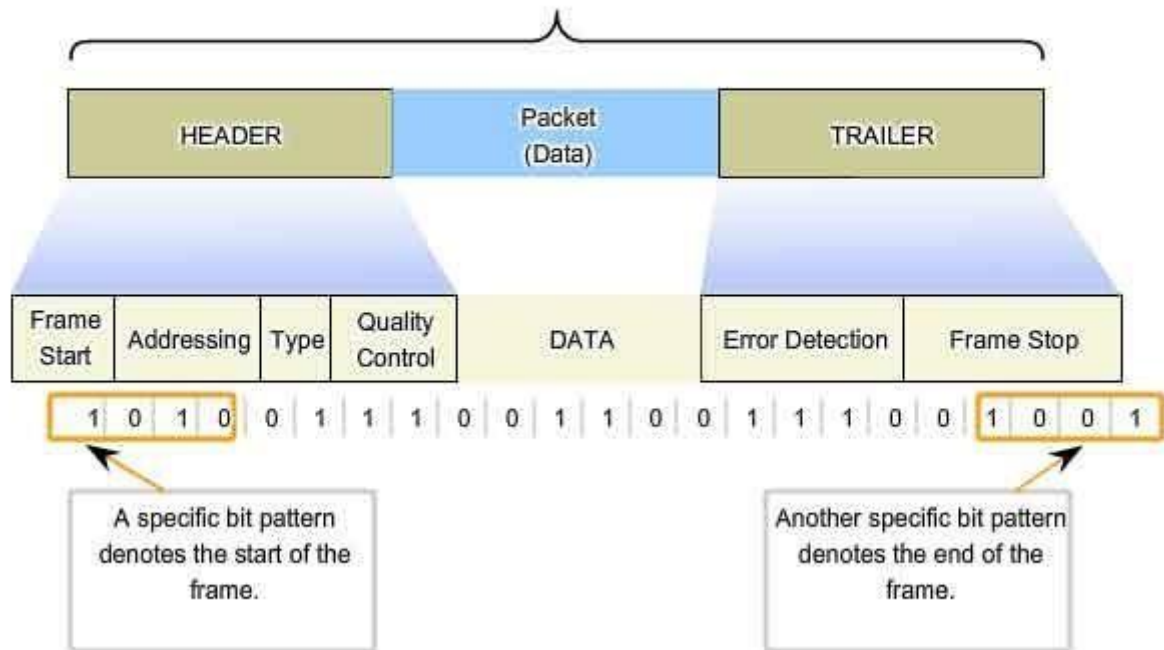
Framing technique does not work satisfactorily, because networks generally do not make any guarantees about the timing. So some other methods are derived.

Framing methods :

Following methods are used for carrying out framing.
1. Character count
2. Starting and ending characters, with character stuffing.
3. Starting and ending flags with bit stuffing.
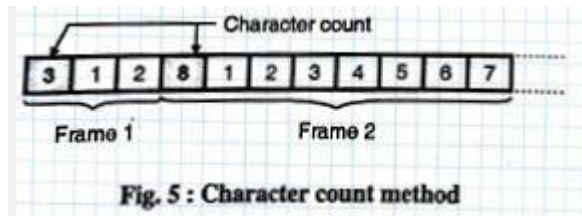4. Physical layer coding violations.

Character count :
In this method, a field in the header is used to specify the number of characters in the frame. This number helps the receiver to know the number of characters in the frame following this count.

The character count method is illustrated in Fig. 5.


Fig. 5 : Character count method

Character Count Method

The two frames shown in Fig. 5 are of 3 and 8 characters respectively. The disadvantage of this method is that, an error can change the character count. If the wrong character count number is received then the receiver will get out of synchronization and will be unable to locate the start of next frame. The character count method is rarely used in practice.

Starting and ending character with character stuffing :

The problem of character count method is solved here by using a starting character before the starting of each frame and an ending character at the end of each frame. Each frame is preceded by the transmission of ASCII character sequence DLE STX. (DLE stands for data link escape and STX is start of Text). After each frame the ASCII character sequence DLE ETX is transmitted. Here DLE stands for Data Link Escape and ETX stands for End of Text. So if the receiver loses the synchronization, it just to search for the DLE STX or DLE ETX characters to return back on track. This is shown in Fig. 6.


Fig. 6
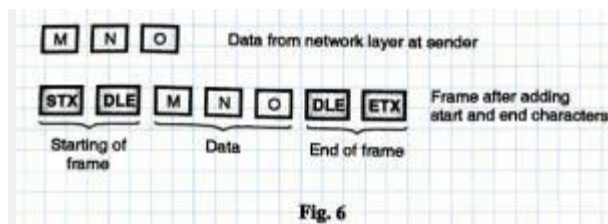
Starting and ending character with character stuffing

Character stuffing :

The problem with this system is that the characters DLE STX or DLE ETX can be a part of data as well. If so, they will be misinterpreted by the receiver as start or end of frame. This problem is solved by using a technique called character stuffing. Which is as follows :
The data link layer at the sending end inserts an ASCII DLE character just before each accidental

DLE character in the data. The data link layer at the receiving end will remove these DLE characters before handing over the data to the network layer.

Fig. 7 : Character stuffing
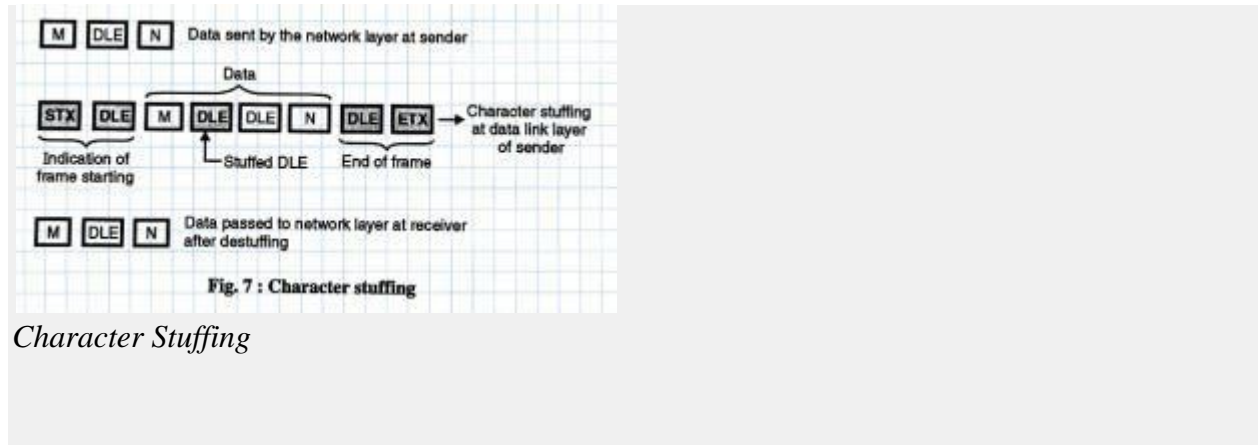
Character Stuffing

Thus the framing DLE STX or DLE ETX can be distinguished from the one in data because DLEs in the data are always doubled. This is called character stuffing and it is shown in Fig. 7. At the receiving end the destuffing is essential.

Starting and ending flags, with bit stuffing :

This technique allows the frames to contain an arbitrary number of bits and codes different from ASCII code. At the beginning and end of each frame, a specific bit pattern 0111 1110 called flag byte is Since there are six consecutive Is in this byte a technique called bit stuffing which is similar to character stuffing is used.

Bit stuffing :

Whenever the sender data link layer detects the presence of five consecutive ones in the data, it automatically stuffs a O bit into the outgoing bit stream. This is called bit stuffing and it is as shown in fig. 8.

Fig. 8 : Bit stuffing and destuffing
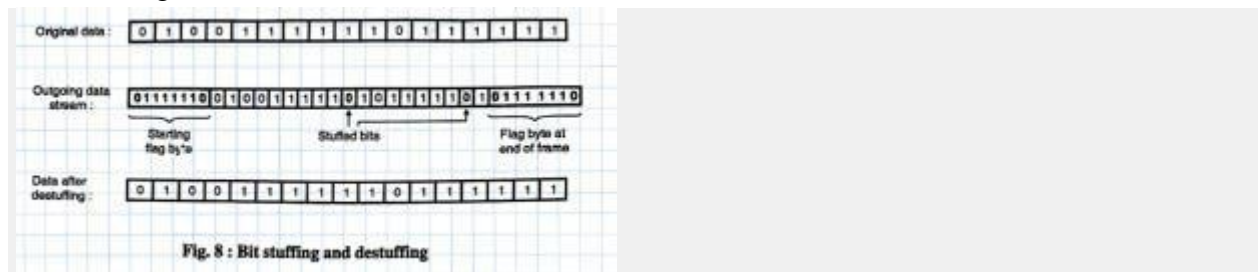
Bit Stuffing and Destuffing

When a receiver detects presence of five consecutive ones in the received bit stream, it automatically deletes the O bit following the five ones. This is called de-stuffing. It is shown in

COMPUTER COMMUNICATION NETWORKS

Fig. 8. Due to bit stuffing, the possible problem if the data contains the flag byte pattern (0111 1110) is eliminated.

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

Stop and Wait

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.
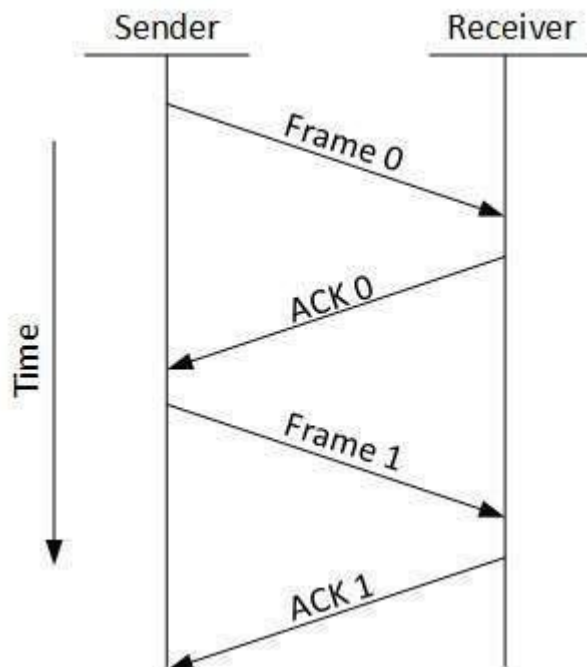
Sliding Window

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

COMPUTER COMMUNICATION NETWORKS

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

Error detection - The sender and receiver, either both or any, must ascertain that there is some error in the transit.

Positive ACK - When the receiver receives a correct frame, it should acknowledge it.

Negative ACK - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.

Retransmission: The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

- An Unrestricted Simplex Protocol
- A Simplex Stop-and-Wait Protocol
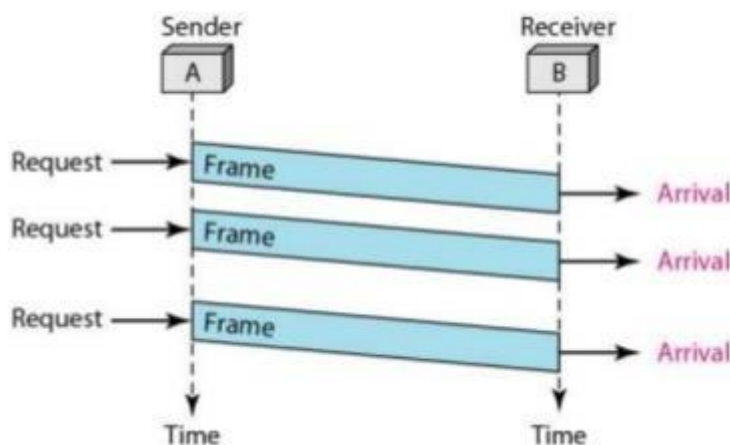- A Simplex Protocol for a Noisy Channel
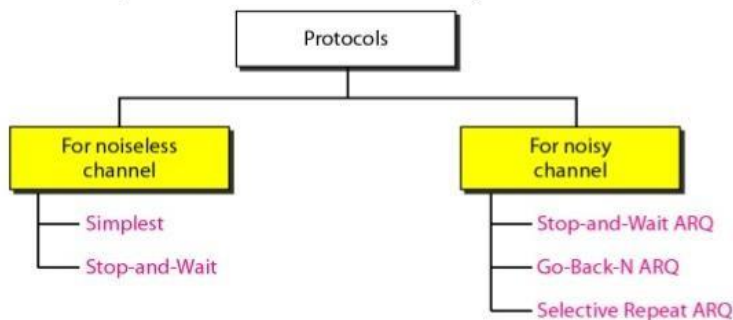
NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or

corrupted. We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does. Of course, neither has error control because we have assumed that the channel is a perfect noiseless channel.

AN UNRESTRICTED SIMPLEX PROTOCOL

- o In order to appreciate the step by step development of efficient and complex protocols we will begin with a simple but unrealistic protocol. In this protocol: Data are transmitted in one direction only
- o The transmitting (Tx) and receiving (Rx) hosts are always ready
- o Processing time can be ignored
- o Infinite buffer space is available
- o No errors occur; i.e. no damaged frames and no lost frames (perfect channel)

In order to appreciate the step by step development of efficient and complex protocols such as SDLC, HDLC etc., we will begin with a simple but unrealistic protocol. In this protocol:

- Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No errors occur; i.e. no damaged frames and no lost frames (perfect channel)

The protocol consists of two procedures, a sender and receiver as depicted below:
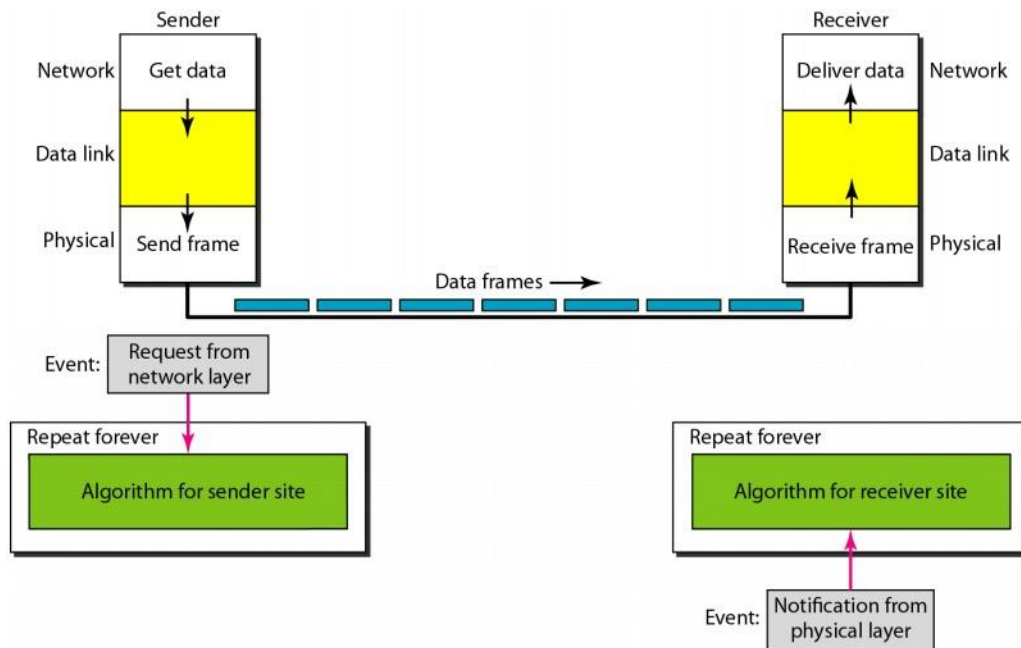
```
/* protocol 1 */
```

```
Sender()
{
        forever
        {
                from_host(buffer);
                S.info = buffer;
                sendf(S);
        }
}

Receiver()
{
        forever
        {
                wait(event);
                getf(R);
                to_host(R.info);
        }
}
```

The design of the simplest protocol with no flow or error control

COMPUTER COMMUNICATION NETWORKS

A simplex stop-and-wait protocol
In this protocol we assume that

- Data are transmitted in one direction only
- No errors occur (perfect channel)
- The receiver can only process the received information at a finite rate

These assumptions imply that the transmitter cannot send frames at a rate faster than the receiver can process them.

The problem here is how to prevent the sender from flooding the receiver.

A general solution to this problem is to have the receiver provide some sort of feedback to the sender. The process could be as follows: The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted. The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame. This protocol is known as *stop-and-wait*.

The protocol is as follows:
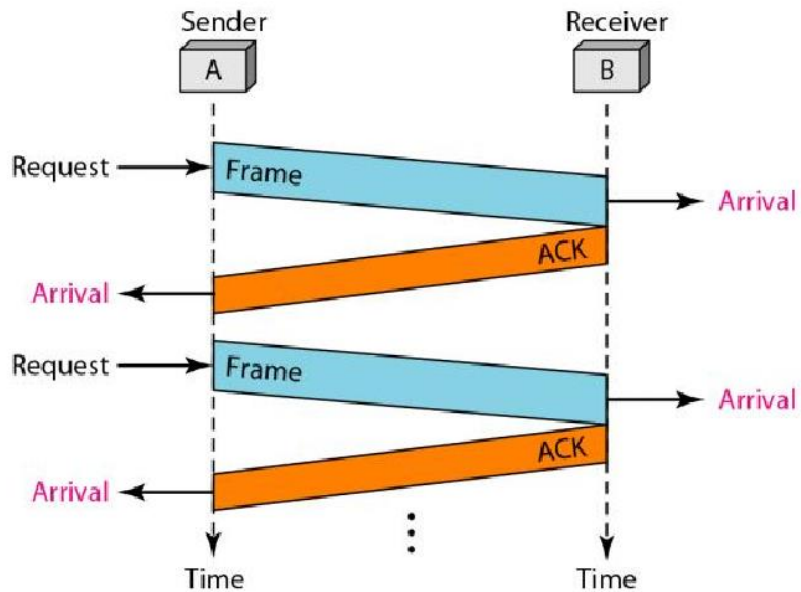
```
/* protocol 2 */

Sender()
{
        forever
        {
                from_host(buffer);
                S.info = buffer;
                sendf(S);
                wait(event);
        }
}

Receiver()
{
        forever
        {
                wait(event);
                getf(R);
                to_host(R.info);
                sendf(S);
        }}
```
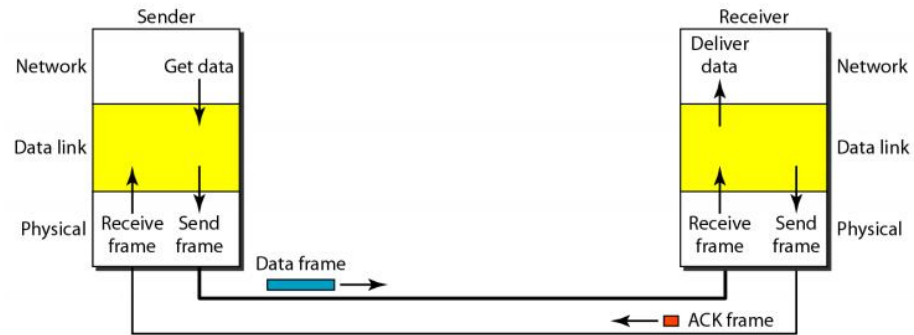
COMPUTER COMMUNICATION NETWORKS

Design of Stop-and-Wait Protocol

COMPUTER COMMUNICATION NETWORKS

A simplex protocol for a noisy channel

In this protocol the unreal "error free" assumption in protocol 2 is dropped. Frames may be either damaged or lost completely. We assume that transmission errors in the frame are detected by the hardware checksum.

One suggestion is that the sender would send a frame, the receiver would send an ACK frame only if the frame is received correctly. If the frame is in error the receiver simply ignores it; the transmitter would time out and would retransmit it.

One fatal flaw with the above scheme is that if the ACK frame is lost or damaged, duplicate frames are accepted at the receiver without the receiver knowing it.

Imagine a situation where the receiver has just sent an ACK frame back to the sender saying that it correctly received and already passed a frame to its host. However, the ACK frame gets lost completely, the sender times out and retransmits the frame. There is no way for the receiver to tell whether this frame is a retransmitted frame or a new frame, so the receiver accepts this duplicate happily and transfers it to the host. The protocol thus fails in this aspect.

To overcome this problem it is required that the receiver be able to distinguish a frame that it is seeing for the first time from a retransmission. One way to achieve this is to have the sender put a sequence number in the header of each frame it sends. The receiver then can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

The receiver needs to distinguish only 2 possibilities: a new frame or a duplicate; a 1-bit sequence number is sufficient. At any instant the receiver expects a particular sequence number. Any wrong sequence numbered frame arriving at the receiver is rejected as a duplicate. A correctly numbered frame arriving at the receiver is accepted, passed to the host, and the expected sequence number is incremented by 1 (modulo 2).

The protocol is depicted below:

```
/* protocol 3 */
```

```
Sender()
{
        NFTS = 0;                       /* NFTS = Next Frame To Send */
        from_host(buffer);
        forever
        {
                S.seq = NFTS;
                S.info = buffer;
                sendf(S);
                start_timer(S.seq);
                wait(event);
                if(event == frame_arrival)
                {
                        from_host(buffer);
                        ++NFTS;  /* modulo 2 operation */
                }
        }
}


Receiver()
{
        FE = 0;                         /* FE = Frame Expected */
        forever
        {
                wait(event);
                if(event == frame_arrival)
                {
                        getf(R);
                        if(R.seq == FE)
                        {
                                to_host(R.info);
                                ++FE;  /* modulo 2 operation */
                        }
                        sendf(S);         /* ACK */
                }
        }
}
```

This protocol can handle lost frames by timing out. The timeout interval has to be long enough to prevent premature timeouts which could cause a "deadlock" situation.
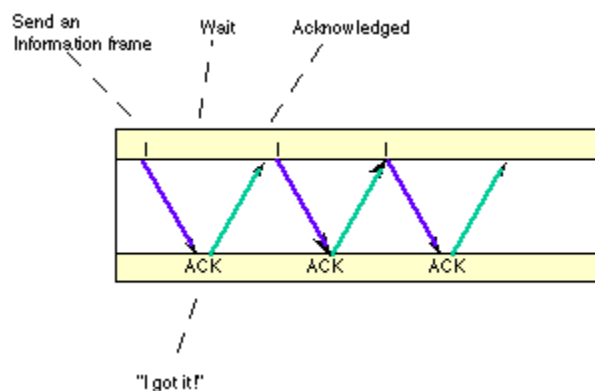
Stop and Wait ARQ

Stop and Wait transmission is the simplest reliability technique and is adequate for a very simple communications protocol. A stop and wait protocol transmits a Protocol Data Unit (PDU) of information and then waits for a response. The receiver receives

each PDU and sends an Acknowledgement (ACK) PDU if a data PDU is received correctly, and a Negative Acknowledgement (NACK) PDU if the data was not received. In practice, the receiver may not be able to reliably identify whether a PDU has been received, and the transmitter will usually also need to implement a timer to recover from the condition where the receiver does not respond.

Under normal transmission the sender will receive an ACK for the data and then commence transmission of the next data block. For a long delay link, the sender may have to wait an appreciable time for this response. While it is waiting the sender is said to be in the "idle" state and is unable to send further data.


Stop and Wait ARQ - Waiting for Acknowledgment (ACK) from the remote node.

The blue arrows show the sequence of data PDUs being sent across the link from the sender (top to the receiver (bottom). A Stop and Wait protocol relies on two way transmission (full duplex or half duplex) to allow the receiver at the remote node to return PDUs acknowledging the successful transmission. The acknowledgements are shown in green in the diagram, and flow back to the original sender. A small processing delay may be introduced between reception of the last byte of a Data PDU and generation of the corresponding ACK.

When PDUs are lost, the receiver will not normally be able to identify the loss (most receivers will not receive anything, not even an indication that something has been corrupted). The transmitter must then rely upon a timer to detect the lack of a response.
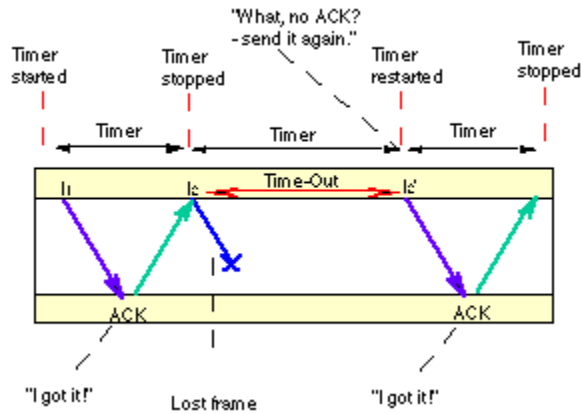
Stop and Wait ARQ - Retransmission due to timer expiry

In the diagram, the second PDU of Data is corrupted during transmission. The receiver discards the corrupted data (by noting that it is followed by an invalid data checksum). The sender is unaware of this loss, but starts a timer after sending each PDU. Normally an ACK PDU is received before this the timer expires. In this case no ACK is received, and the timer counts down to zero and triggers retransmission of the same PDU by the sender. The sender always starts a timer following transmission, but in the second transmission receives an ACK PDU before the timer expires, finally indicating that the data has now been received by the remote node.
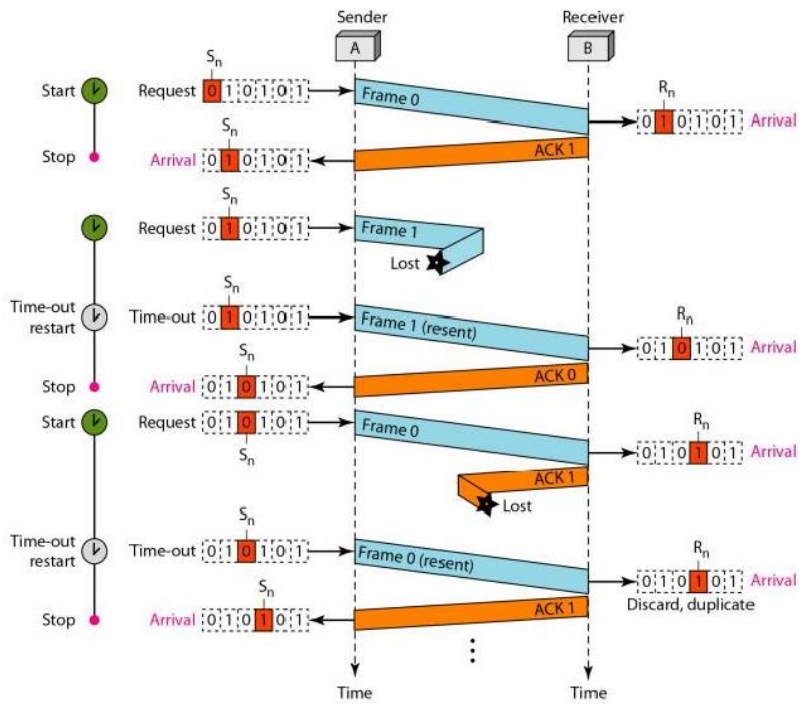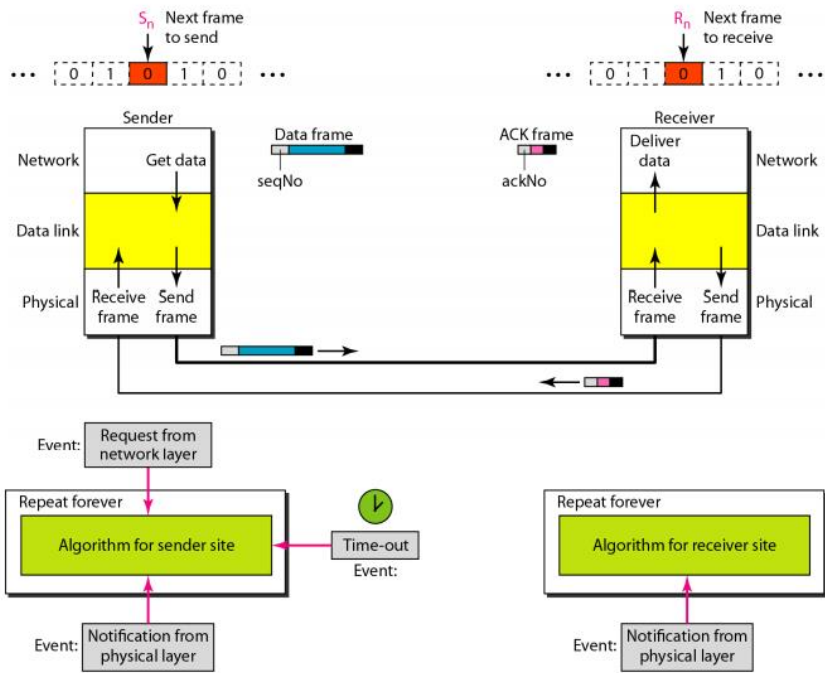
The state diagram (also showing the operation of NACK) is shown below:
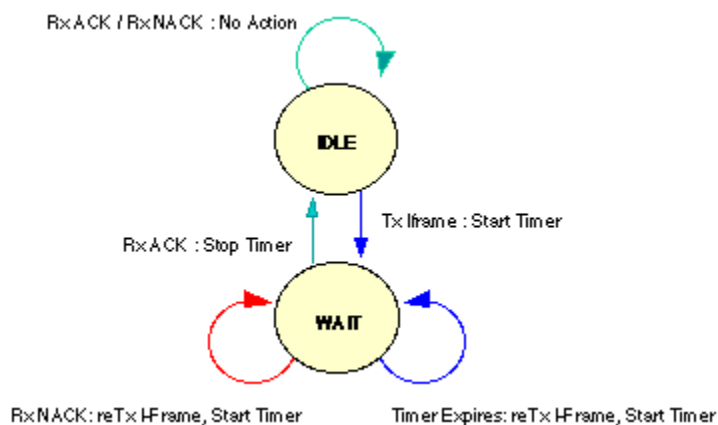
State Diagram for a simple stop and wait protocol

Design of the Stop-and-Wait ARQ Protocol

COMPUTER COMMUNICATION NETWORKS

Go Back n Protocol

Go Back n is a connection oriented protocol in which the transmitter has a window of sequence numbers that may be transmitted without acknowledgment. The receiver will only accept the next sequence number it is expecting - other sequence nubmers are silently ignored.
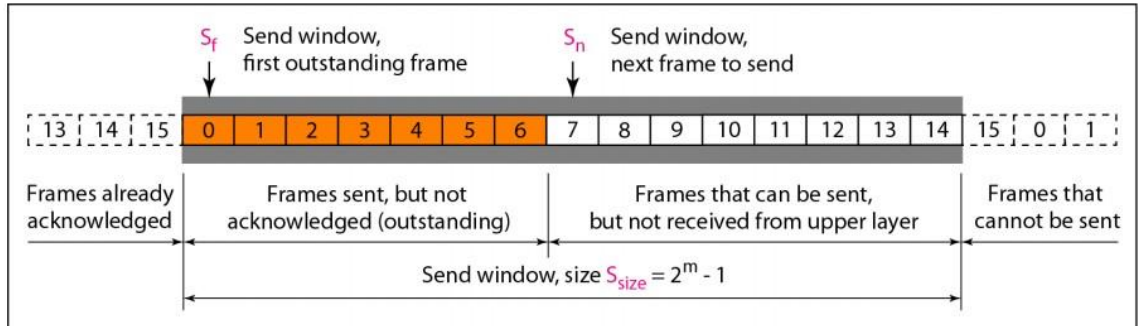
The protocol simulation shows a time-sequence diagram with users A and B, protocol entities A and B that support them, and a communications medium that carries messages. Users request data transmissions with *DatReq(DATAn),* and receive data transmissions as *DatInd(DATAn).* Data messages are simply numbered *DATA0*, *DATA1*, etc. without explicit content. The transmitting protocol sends the protocol message *DT(n)* that gives only the sequence number, not the data. Once sequence numbers reach a maximum number (like 7), they wrap back round to 0. An acknowledgement *AK(n)* means that the *DT* message numbered *n* is the next one expected (i.e. all messages up to but not including this number have been received). Since sequence numbers wrap round, an acknowledgement with sequence number 1 refers to messages 0, 1, 7, 6, etc. Note that if a*DT* message is received again due to re-transmission, it is acknowledged but discarded.

The protocol has a maximum number of messages that can be sent without acknowledgement. If this window becomes full, the protocol is blocked until an acknowledgement is received for the earliest outstanding message. At this point the transmitter is clear to send more messages.
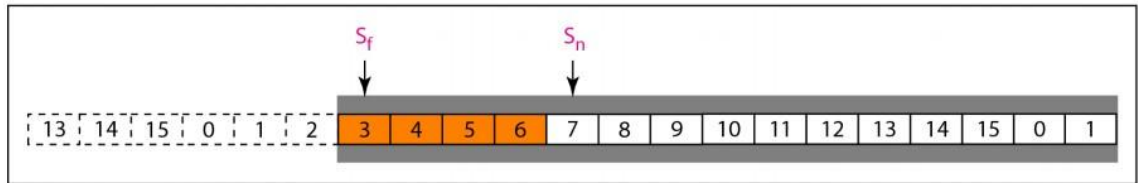
The receiver delivers the protocol messages *DT(n)* to the user in order. Any received out of order are ignored.

Send window for Go-Back-N ARQ

S_f Send window, first outstanding frame

S_n Send window, next frame to send

| 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 |

Frames already acknowledged

Frames sent, but not acknowledged (outstanding)

Frames that can be sent, but not received from upper layer

Frames that cannot be sent

Send window, size $S_{size} = 2^m - 1$

a. Send window before sliding

S_f

S_n

| 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 |

b. Send window after sliding

Receive window for Go-Back-N ARQ
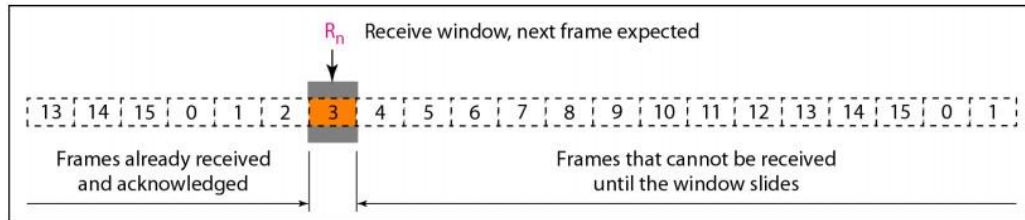
R_n Receive window, next frame expected

| 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 |

Frames already received and acknowledged

Frames that cannot be received until the window slides

a. Receive window

R_n

| 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 |
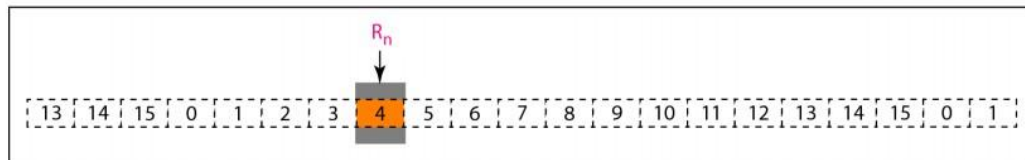
b. Window after sliding

Design of Go-Back-N ARQ
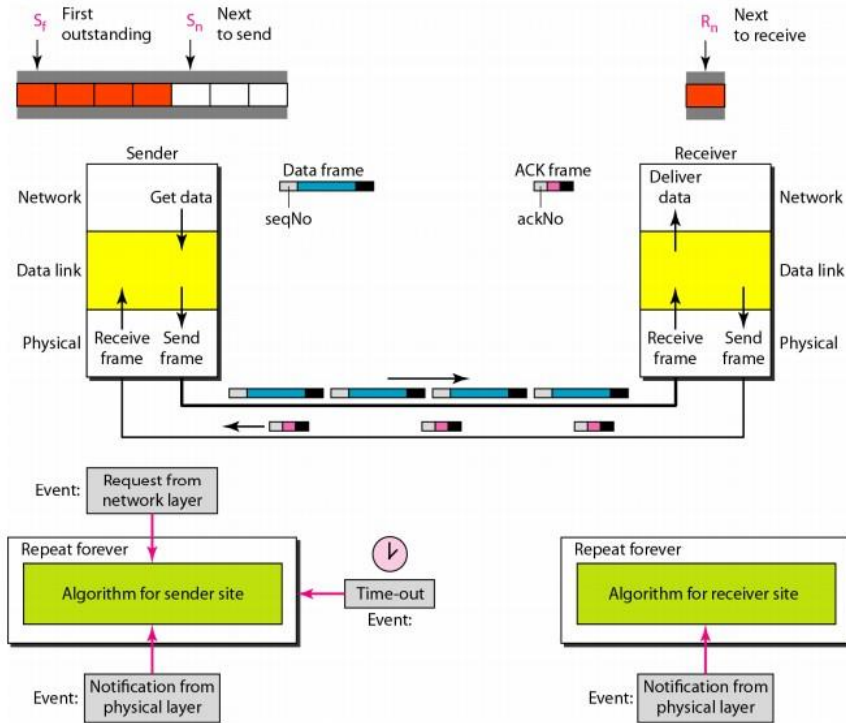
Window size for Go-Back-N ARQ
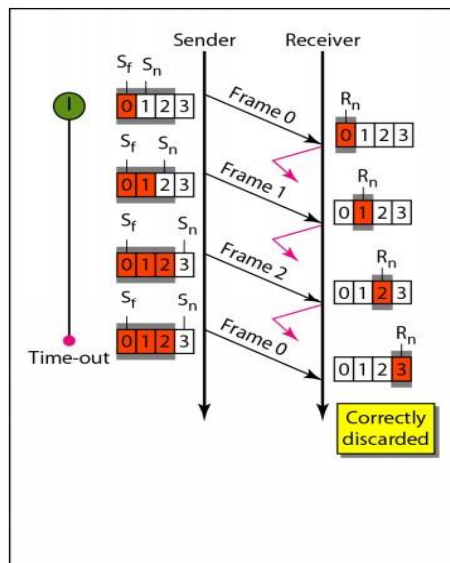
a. Window size < 2^m
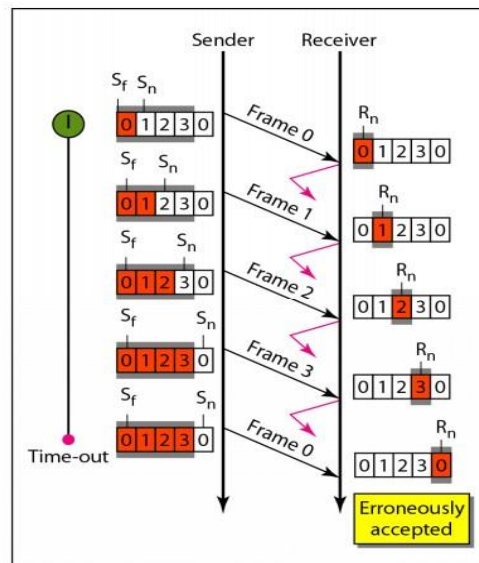
b. Window size = 2^m
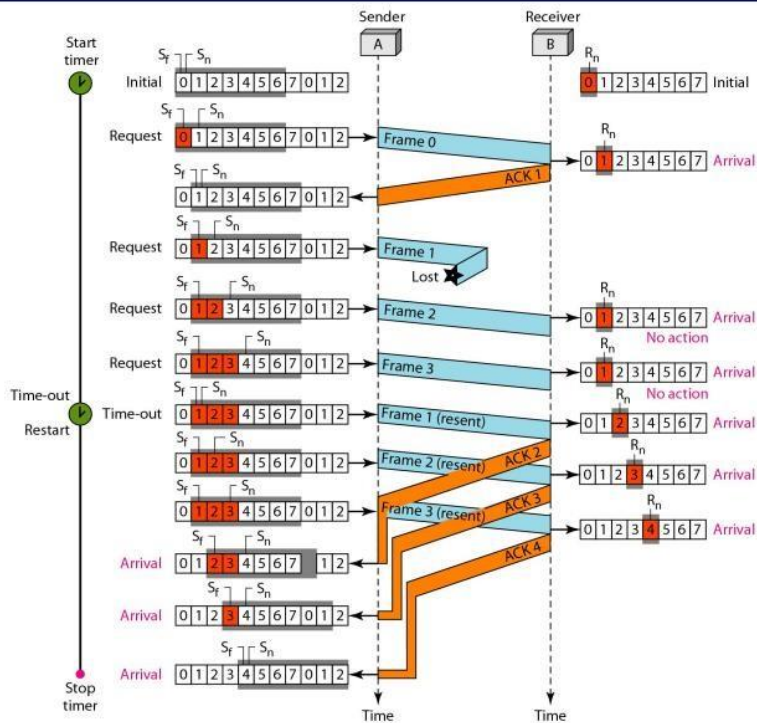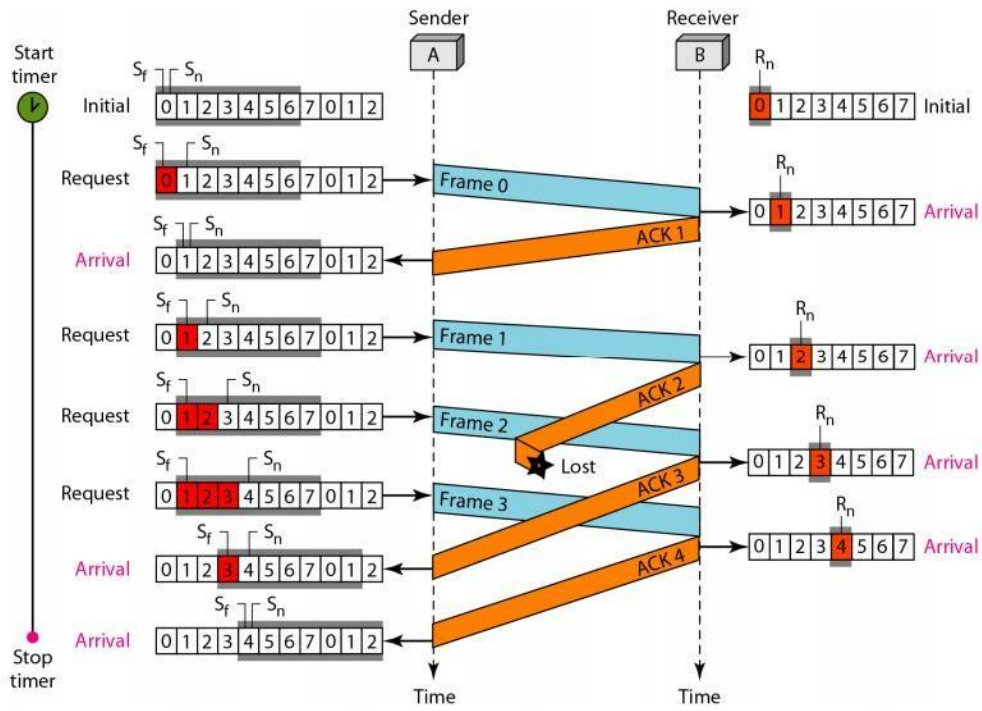
Selective Repeat Error Recovery

Selective Repeat error recovery is a procedure which is implemented in some communications protocols to provide reliability. It is the most complex of a set of procedures which may provide error recovery, it is however the most efficient scheme. Selective repeat is employed by the TCP transport protocol.

Features required for Selective Repeat ARQ

- To support Go-Back-N ARQ, a protocol must number each PDU which is sent. (PDUs are normally numbered using modulo arithmetic, which allows the same number to be re-used after a suitably long period of time. The time period is selected to ensure the same PDU number is never used again for a different PDU, until the first PDU has "left the network" (e.g. it may have been acknowledged)).
- The local node must also keep a buffer of all PDUs which have been sent, but have not yet been acknowledged.
- The receiver at the remote node keeps a record of the highest numbered PDU which has been correctly received. This number corresponds to the last acknowledgement PDU which it may have sent.

The above features are also required for Go-Back-N, however for selective repeat, the receiver must also maintain a buffer of frames which have been received, but not acknowledged.

Recovery of lost PDUs using Selective Repeat ARQ

The recovery of a corrupted PDU proceeds in four stages:

- First, the corrupted PDU is discarded at the remote node's receiver.
- Second, the remote node requests retransmission of the missing PDU using a control PDU (sometimes called a Selective Reject). The receiver then stores all out-of-sequence PDUs in the receive buffer until the requested PDU has been retransmitted.
- The sender receives the retransmission request and then transmits the lost PDU(s).

- The receiver forwards the retransmitted PDU, and all subsequent in-sequence PDUs which are held in the receive buffer.

A remote node may request retransmission of corrupted PDUs by initiating Selective Repeat error recovery by sending a control PDU indicating the **missing** PDU. This allows the remote node to instruct the sending node where to retransmit the PDU which has not been received. The remote**stores** any out-of-sequence PDUs (i.e. which do not have the expected sequence number) until the retransmission is complete.

Upon receipt of a Selective Repeat control PDU (by the local node), the transmitter **sends a single PDU** from its buffer of unacknowledged PDUs. The transmitter then **continues** normal transmission of new PDUs until the PDUs are acknowledged or another selective repeat request is received.

Send window for Selective Repeat ARQ

Send window, first S_f outstanding frame

S_n Send window, next frame to send

| 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 |

Frames already acknowledged

Frames sent, but not acknowledged

Frames that can be sent

Frames that cannot be sent
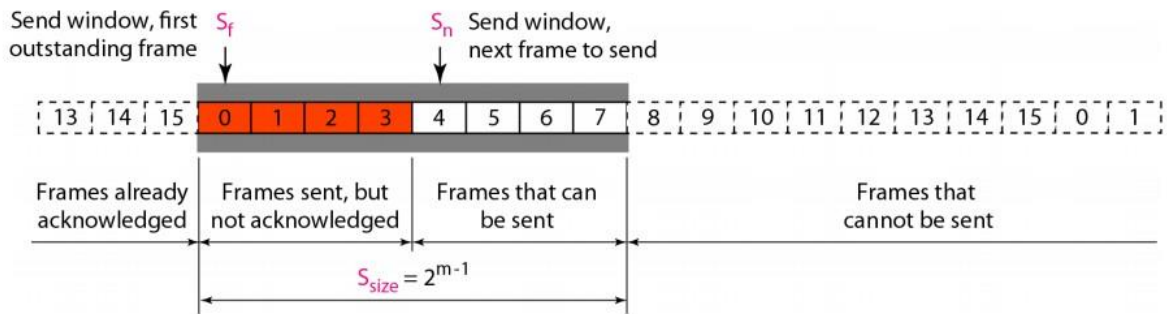
$$S_{size} = 2^{m-1}$$

Design of Selective Repeat ARQ

S_f First outstanding S_n Next to send

R_n Next to receive

Sender

Data frame

seqNo

ACK or NAK

ackNo or nakNo

Receiver

Deliver data

Network   Get data

Network

Data link

Data link

Physical   Receive   Send frame   frame

Receive   Send frame   frame   Physical

Event: Request from network layer

Repeat forever

Algorithm for sender site

Time-out
Event:

Repeat forever

Algorithm for receiver site

Event: Notification from physical layer

Event: Notification from physical layer

Receive window for Selective Repeat ARQ
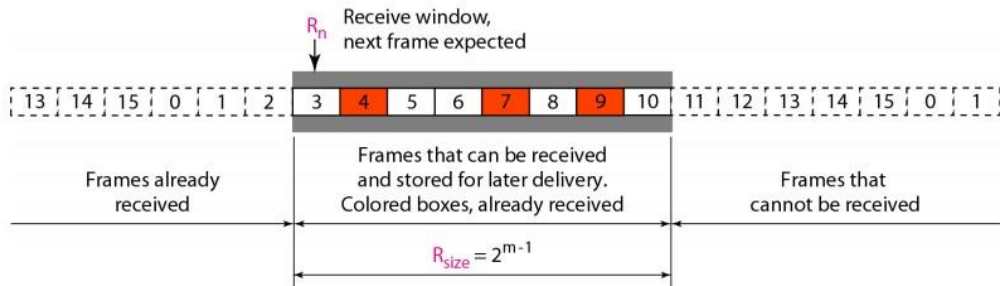
Selective Repeat ARQ, window size

a. Window size = 2^{m-1}

b. Window size > 2^{m-1}

Delivery of data in Selective Repeat ARQ
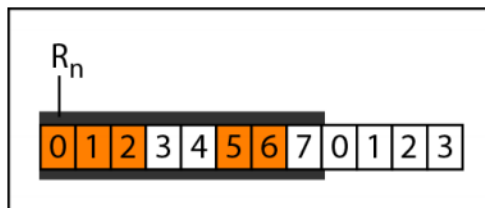
a. Before delivery
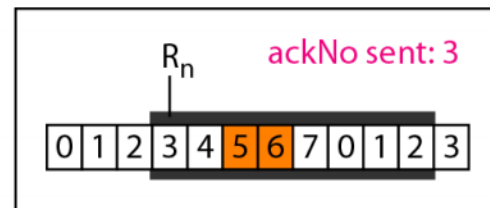
b. After delivery
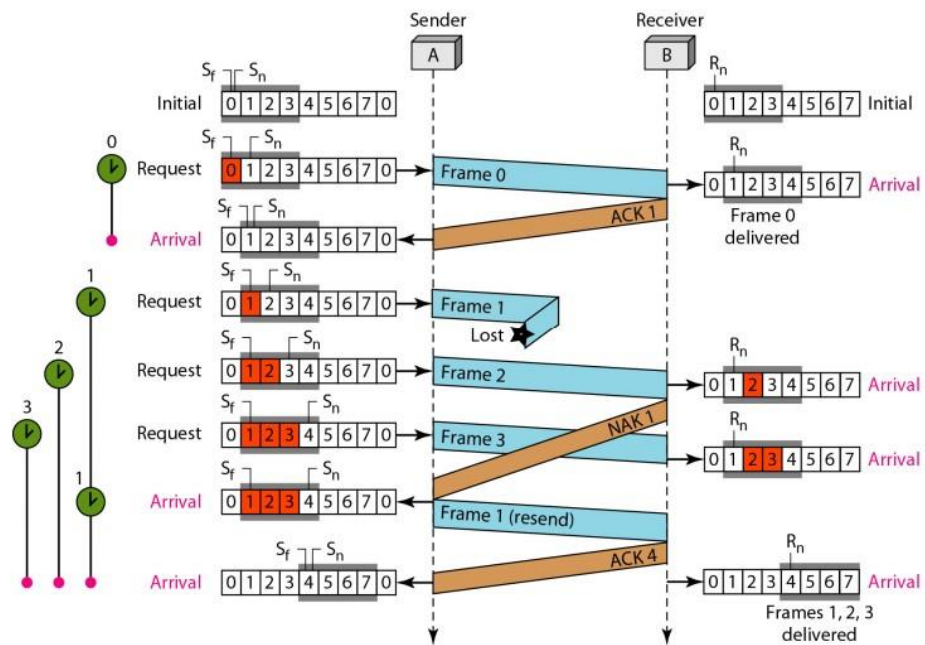
HDLC Encapsulation

ISO Standard - current ISO 13239

HDLC frames

Flag	Address	Control	Information	FCS	Flag

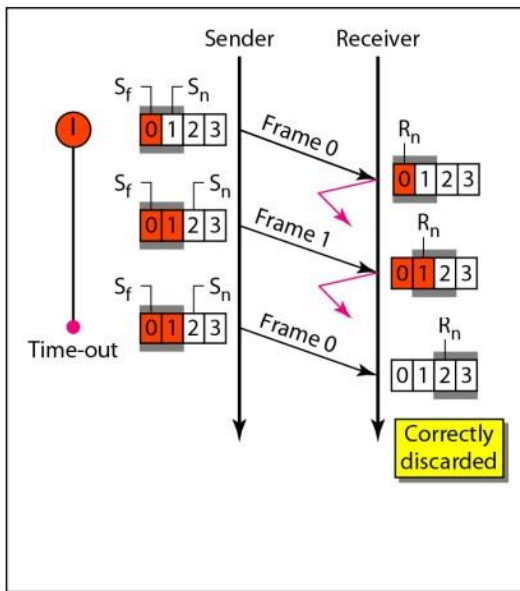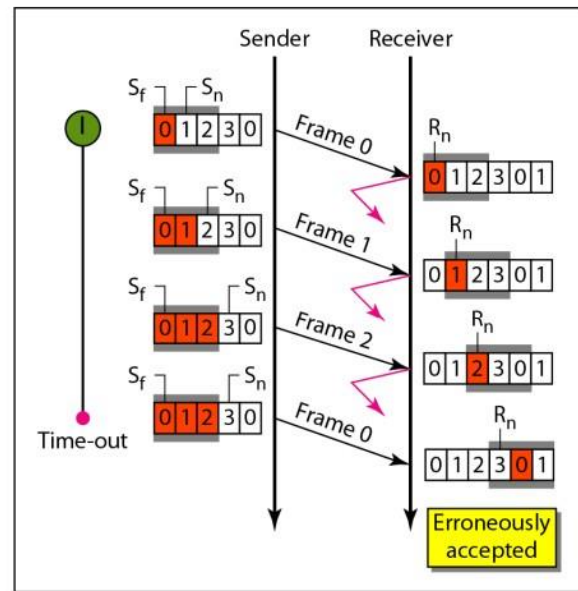←—8—→ ←—8—→ ←—8 or 16—→ ←———variable———→ ←——16 or 32——→ ←—8—→
bits   extendable

| Flag | Address | Control | User information | FCS | Flag | I-frame |

| Flag | Address | Control | FCS | Flag | S-frame |

| Flag | Address | Control | Management information | FCS | Flag | U-frame |
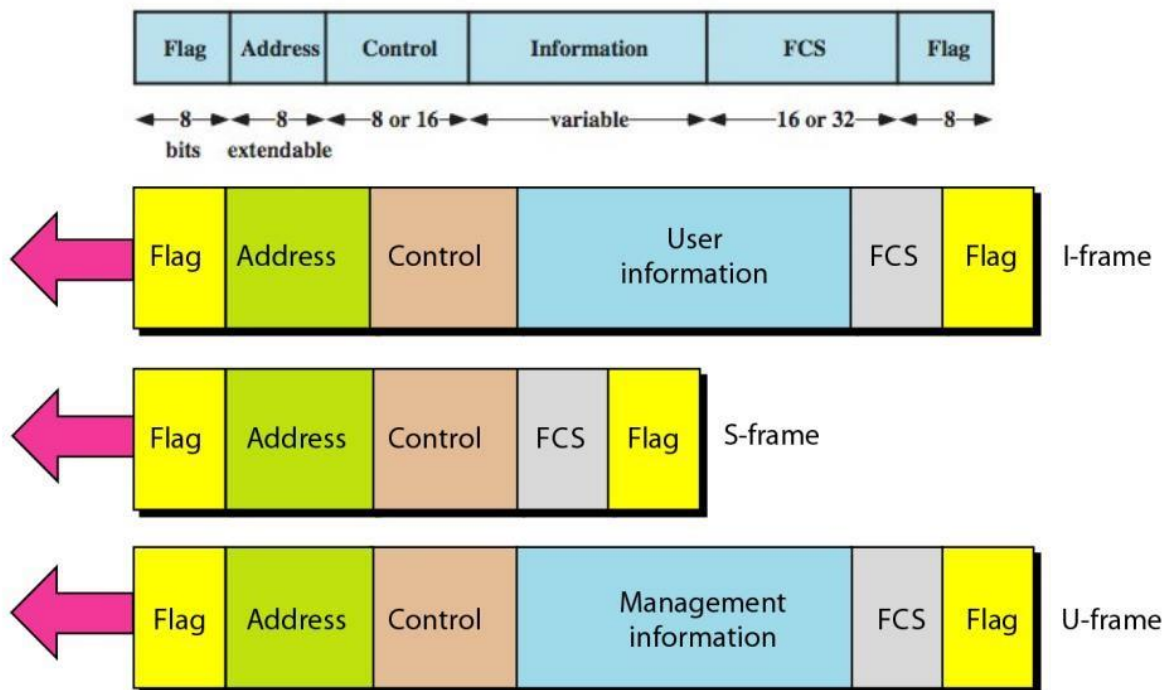
- bit-oriented protocol
- Developed from the Synchronous Data Link Control
- Provides connection-oriented and connectionless service
- Synchronous serial transmission to provide error-free communication between devices
- HDLC uses a frame delimiter to mark the beginning and end of each frame, something that a layer 1 protocol cannot do over synchronous nor asynchronous transfer.
- Cisco has a proprietary, but vendor-friendly, implementation of HDLC that solves multiprotocol support problems.

Notable Frame Fields

- Flag - The flag field initiates and terminates error checking. The frame always starts and ends with an 8-bit flag field. The bit pattern is 01111110. Because there is a likelihood that this pattern occurs in the actual data, the sending HDLC system always inserts a 0 bit after   every

five 1s in the data field, so in practice the flag sequence can only occur at the frame ends. The receiving system strips out the inserted bits. When frames are transmitted consecutively, the end flag of the first frame is used as the start flag of the next frame.

- Address - The address field contains the HDLC address of the secondary station. This address can contain a specific address, a group address, or a broadcast address. A primary address is either a communication source or a destination, which eliminates the need to include the address of the primary.

- Control
  - Information (I) frame: I-frames carry upper layer information and some control information. This frame sends and receives sequence numbers, and the poll final (P/F) bit performs flow and error control. The send sequence number refers to the number of the frame to be sent next. The receive sequence number provides the number of the frame to be received next. Both sender and receiver maintain send and receive sequence numbers. A primary station uses the P/F bit to tell the secondary whether it requires an immediate response. A secondary station uses the P/F bit to tell the primary whether the current frame is the last in its current response.
  - Supervisory (S) frame: S-frames provide control information. An S-frame can request and suspend transmission, report on status, and acknowledge receipt of I-frames. S-frames do not have an information field.
  - Unnumbered (U) frame: U-frames support control purposes and are not sequenced. A U-frame can be used to initialize secondaries. Depending on the function of the U-frame, its control field is 1 or 2 bytes. Some U-frames have an information field.

- Protocol - (only used in Cisco HDLC) This field specifies the protocol type encapsulated within the frame (e.g. 0x0800 for IP).

- Data - The data field contains a path information unit (PIU) or exchange identification (XID) information.

- Frame check sequence (FCS) - The FCS precedes the ending flag delimiter and is usually a cyclic redundancy check (CRC) calculation remainder. The CRC calculation is redone in the receiver. If the result differs from the value in the original frame, an error is assumed.

Point-to-Point Protocol (PPP)

• PPP is most commonly used data link protocol. It is used to connect the Home PC to the server of ISP via a modem.

• This protocol offers several facilities that were not present in SLIP. Some of these facilities are:

1. PPP defines the format of the frame to be exchanged between the devices.

2. It defines link control protocol (LCP) for:-

(a) Establishing the link between two devices.

(b) Maintaining this established link.

(c) Configuring this link.

(d) Terminating this link after the transfer.

3. It defines how network layer data are encapsulated in data link frame.

4. PPP provides error detection.

5. Unlike SLIP that supports only IP, PPP supports multiple protocols.

6. PPP allows the IP address to be assigned at the connection time i.e. dynamically. Thus a temporary IP address can be assigned to each host.

7. PPP provides multiple network layer services supporting a variety of network layer protocol. For this PPP uses a protocol called NCP (Network Control Protocol).

8. It also defines how two devices can authenticate each other.

PPP Frame Format

The frame format of PPP resembles HDLC frame. Its various fields are:

PPP Frame Format

The frame format of PPP resembles HDLC frame. Its various fields are:

PPP frame Format

1. **Flag field**: Flag field marks the beginning and end of the PPP frame. Flag byte is 01111110. (1 byte).

2. **Address field**: This field is of 1 byte and is always 11111111. This address is the broadcast address *i.e.* all the stations accept this frame.

3. **Control field**: This field is also of 1 byte. This field uses the format of the U-frame (unnumbered) in HDLC. The value is always 00000011 to show that the frame does not contain any sequence numbers and there is no flow control or error control.

4. **Protocol field**: This field specifies the kind of packet in the data field *i.e.* what is being carried in data field.

5. **Data field**: Its length is variable. If the length is not negotiated using LCP during line set up, a default length of 1500 bytes is used. It carries user data or other information.

6. **FCS field**: The frame checks sequence. It is either of 2 bytes or 4 bytes. It contains the checksum.

Transition Phases in PPP

The PPP connection goes through different states as shown in fig.

1. **Dead**: In dead phase the link is not used. There is no active carrier and the line is quiet.
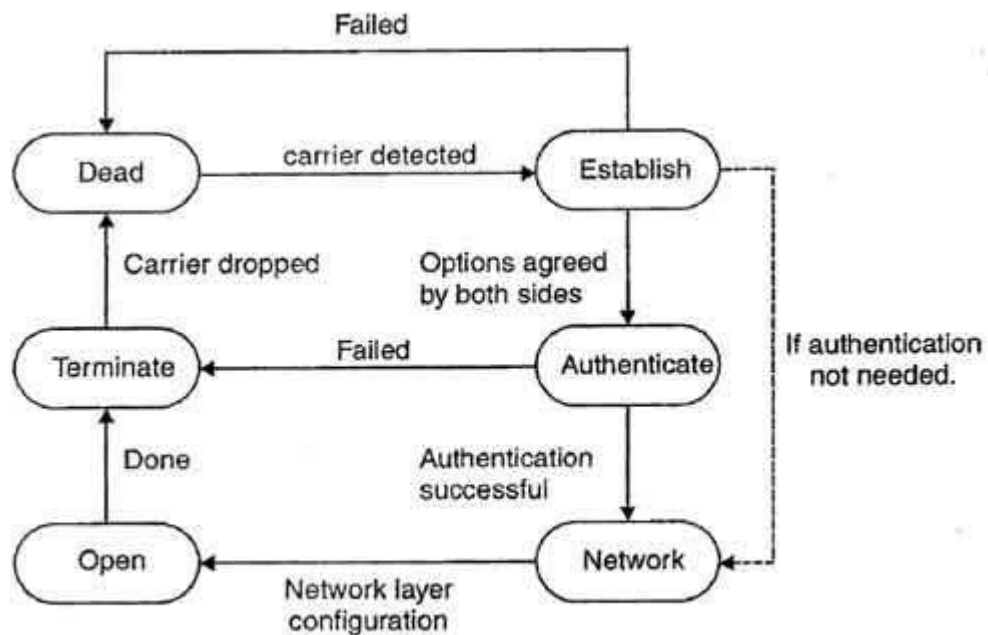
Transition phases

2. **Establish**: Connection goes into this phase when one of the nodes start communication. In this phase, two parties negotiate the options. If negotiation is successful, the system goes into authentication phase or directly to networking phase. LCP packets are used for this purpose.

3. **Authenticate**: This phase is optional. The two nodes may decide during the establishment phase, not to skip this phase. However if they decide to proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.

4. **Network**: In network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged. This is because PPP supports several protocols at network layer. If a node is running multiple protocols simultaneously at the network layer, the receiving node needs to know which protocol will receive the data.

5. **Open**: In this phase, data transfer takes place. The connection remains in this phase until one of the endpoints wants to end the connection.

6. **Terminate**: In this phase connection is terminated.

Point-to-point protocol Stack

PPP uses several other protocols to establish link, authenticate users and to carry the network layer data.

The various protocols used are:

1. Link Control Protocol

2. Authentication Protocol

3. Network Control Protocol

1. Link Control Protocol

• It is responsible for establishing, maintaining, configuring and terminating the link.

• It provides negotiation mechanism to set options between two endpoints.
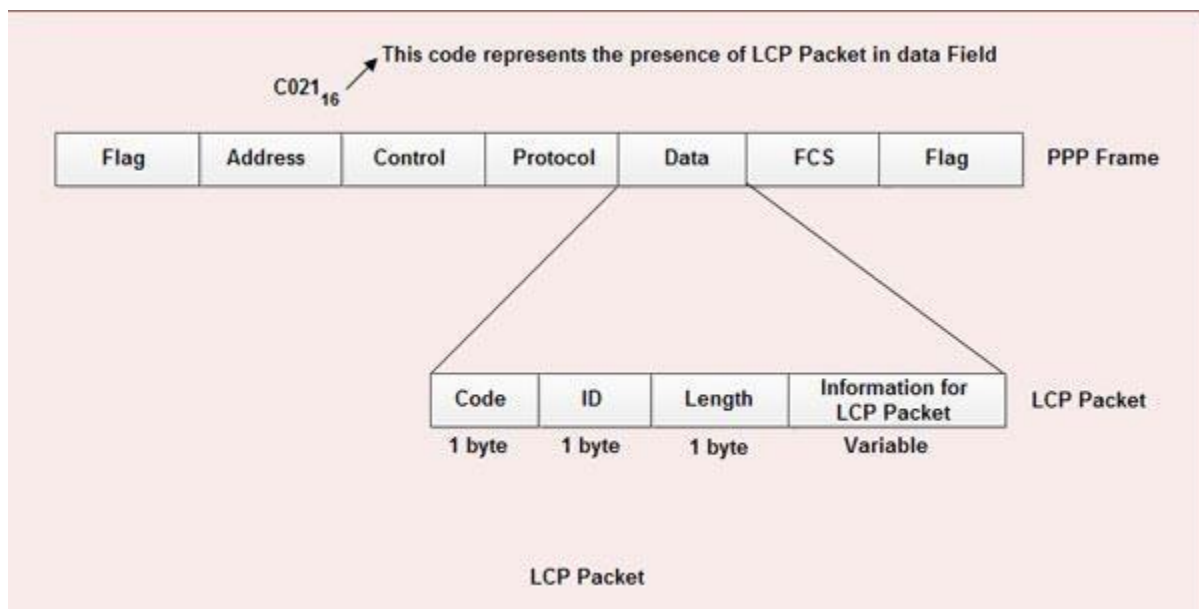
LCP Packet

• All LCP packets are carried in the data field of the PPP frame.

• The presence of a value $C021_{16}$ in the protocol field of PPP frame indicates that LCP packet is present in the data field.

• The various fields present in LCP packet are:

1. **Code**: 1 byte-specifies the type of LCP packet.

2. **ID**: 1 byte-holds a value used to match a request with the reply.

3. **Length**: 2 byte-specifies the length of entire LCP packet.

4. **Information**: Contains extra information required for some LCP packet.

• There are eleven different type of LCP packets. These are categorized in three groups:

1. **Configuration packet**: These are used to negotiate options between the two ends. For example: configure-request, configure-ack, configure-nak, configure-reject are some configuration packets.

2. **Link termination packets**: These are used to disconnect the link between two end points. For example: terminate-request, terminate-ack, are some link termination packets.

3. **Link monitoring and debugging packets**: These are used to monitor and debug the links. For example: code-reject, protocol-reject, echo-request, echo-reply and discard-request are some link monitoring and debugging packets.

2. Authentication Protocol

Authentication protocols help to validate the identity of a user who needs to access the resources.

There are two authentication protocols:

1. Password Authentication Protocols (PAP)

2. Challenge Handshake Authentication Protocol (CHAP)

1. PAP (Password Authentication Protocol)

This protocol provides two step authentication procedures:

Step 1: User name and password is provided by the user who wants to access a system.

Step 2: The system checks the validity of user name and password and either accepts or denies the connection.

• PAP packets are also carried in the data field of PPP frames.

• The presence of PAP packet is identified by the value $C023_{16}$ in the protocol field of PPP frame.

• There are three PAP packets.

1. **Authenticate-request**: used to send user name & password.

2. **Authenticate-ack**: used by system to allow the access.

3. **Authenticate-nak**: used by system to deny the access.

2. CHAP (Challenge Handshake Authentication Protocol)

• It provides more security than PAP.

• In this method, password is kept secret, it is never sent on-line.

• It is a three-way handshaking authentication protocol:

1. System sends. a challenge packet to the user. This packet contains a value, usually a few bytes.

2. Using a predefined function, a user combines this challenge value with the user password and sends the resultant packet back to the system.

3. System then applies the same function to the password of the user and challenge value and creates a result. If result is same as the result sent in the response packet, access is granted, otherwise, it is denied.

• **There are 4 types of CHAP packets:**

1. Challenge-used by system to send challenge value.

2. Response-used by the user to return the result of the calculation.

3. Success-used by system to allow access to the system.

4. Failure-used by the system to deny access to the system.

3. Network Control Protocol (NCP)

• After establishing the link and authenticating the user, PPP connects to the network layer. This connection is established by NCP.

• Therefore NCP is a set of control protocols that allow the encapsulation of the data coming from network layer.

• After the network layer configuration is done by one of the NCP protocols, the users can exchange data from the network layer.

• PPP can carry a network layer data packet from protocols defined by the Internet, DECNET, Apple Talk, Novell, OSI, Xerox and so on.

• None of the NCP packets carry networks layer data. They just configure the link at the network layer for the incoming data.

Piggy Backing

In all practical situations, the transmission of data needs to be bi-directional. This is called as full-duplex transmission.

We can achieve this full duplex transmission *i.e.* by having two separate channels-one for forward data transfer and the other for separate transfer *i.e.* for acknowledgements.

COMPUTER COMMUNICATION NETWORKS

• A better solution would be to use each channel (forward & reverse) to transmit frames both ways, with both channels having the same capacity. If A and B are two users. Then the data frames from A to Bare intermixed with the acknowledgements from A to B.

• One more improvement that can be made is piggybacking. The concept is explained as follows:

Example of piggybacking without error
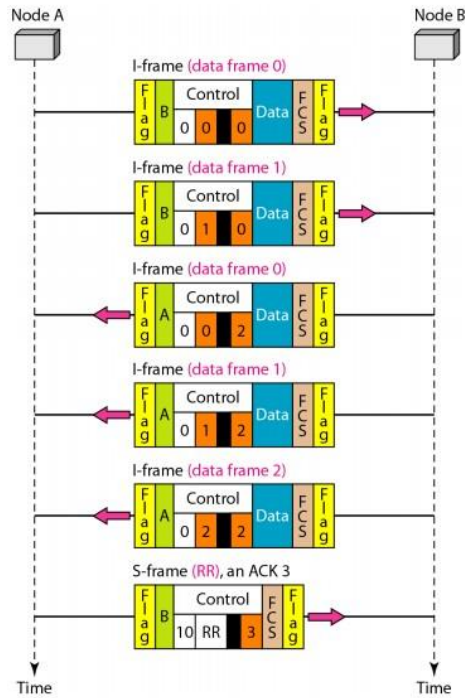
COMPUTER COMMUNICATION NETWORKS

Example of piggybacking with error
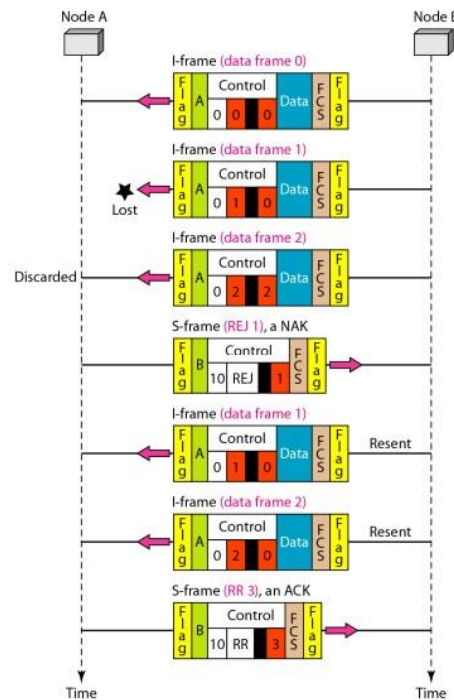
In two way communication, Whenever a data frame is received, the received waits and does not send the control frame (acknowledgement) back to the sender immediately.

The receiver waits until its network layer passes in the next data packet. The delayed acknowledgement is then attached to this outgoing data frame.

This technique of temporarily delaying the acknowledgement so that it can be hooked with next outgoing data frame is known as piggybacking.

• The major advantage of piggybacking is better use of available channel bandwidth.

• The disadvantages of piggybacking are:

1. Additional complexity.

2. If the data link layer waits too long before transmitting the acknowledgement, then retransmission of frame would take place.

COMPUTER COMMUNICATION NETWORKS

Objectives:

The student shall be able to:

☐ Describe how CSMA/CD and CSMA/CA work, and the differences between the two.

☐ Be able to determine the number of collisions and successful transmissions given an example situation for the following technologies: Aloha, Slotted Aloha, CSMA/CD, CSMA/CA.

☐ Define FDMA, TDMA, CDMA, Channelization, Scheduling, Reservation, Polling, and briefly define how they work.

☐ Define switch, bridge, hub.

COMPUTER COMMUNICATION NETWORKS

UNIT 3,4,5 :   Medium Access Control (MAC)
Local Area Networks (LAN)

Introduction

LAN: used to interconnect distributed terminals located within a single building or localized group of buildings.

```
                    ┌─────────┐
                    │ Network │
                    └────┬────┘
              ┌──────────┴──────────┐
         ┌────┴────┐          ┌─────┴─────┐
         │ Switche │          │ Broadcast │
         └─────────┘          └─────┬─────┘
                      ┌──────────────┴──────────┐
               ┌──────┴───────┐          ┌───────┴──┐
               │ Channelization│          │ Dynamic  │
               └──────────────┘          └────┬─────┘
                                   ┌───────────┴────────────┐
                            ┌──────┴─────┐           ┌───────┴──┐
                            │ Scheduling │           │  Random  │
                            └──────┬─────┘           └──────────┘
                          ┌────────┴────────┐
                    ┌─────┴─────┐     ┌──────┴──┐
                    │ Reservatio│     │ Polling │
                    └───────────┘     └─────────┘
```
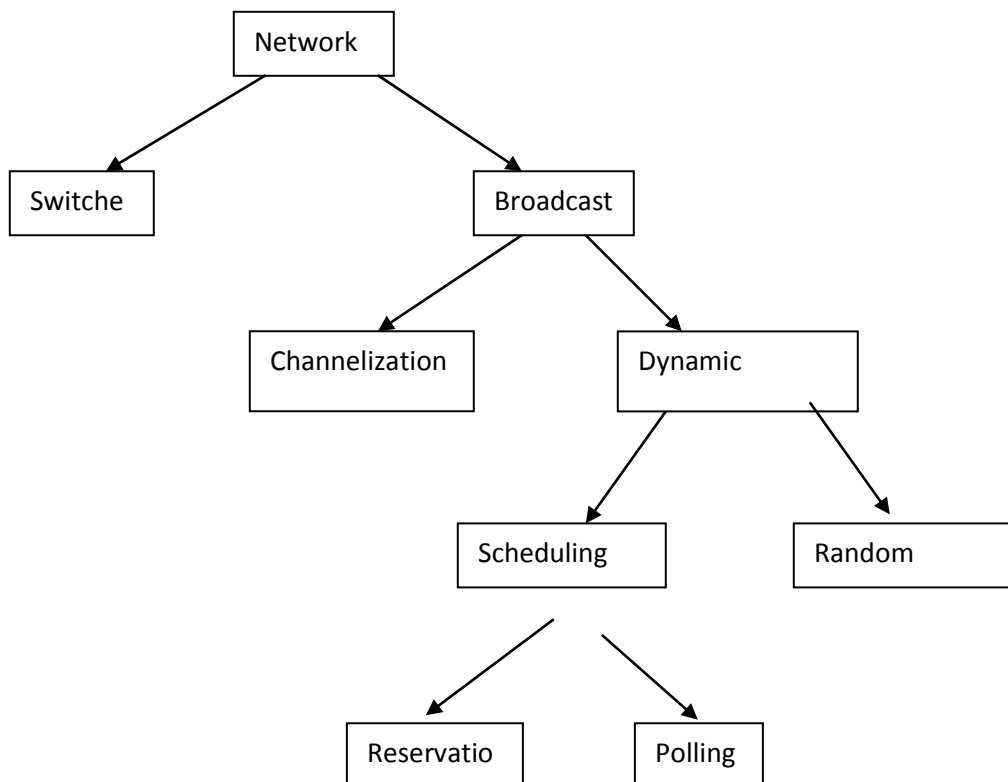
Medium Access Control: Coordinate access to a channel so that information can be transmitted from a source to a destination in a broadcast network.

Channelization Scheme: Partitions medium into separate channels that are dedicated to particular users.
- Base Station or Controller → Station: Forward or Downstream transmission
- Station → Base Station or Controller: Reverse or Upstream transmission
Examples: Cellular FDMA, TDMA, CDMA; Multidrop

Dynamic MAC Scheme: Direct communication between all M stations
☐ Dynamic sharing of the medium on a per packet basis.
☐ No central control
☐ Must minimize collision of packet transmission by multiple sources simultaneously.
Example: Ethernet, Ad Hoc Wireless LANs, Multitapped bus.

Dynamic Medium Access Control: Random Access

Random Access: Transmit immediately and then randomly if collision occurred
☐ Coordination improves throughput.

Aloha Protocol

Aloha protocol: *From University of Hawaii*
☐ Everyone transmits when they want to transmit.
☐ Result: a lot of collisions.
☐ Maximum throughput: 18.4% (when something to send 50% of time).

Throughput calculations:
    S = Throughput
    G = Average number of frames attempting transmission
    Assumes frames are of equal length
    $P(0 \text{ frames}) = e^{-2G}$
    $S = G(e^{-2G})$
    Highest throughput achieved when: G=0.5 S=18.4%

Slotted Aloha protocol:
☐ Terminal can only transmit during fixed time intervals called timeslots.
☐ Max throughput achieved at: 37% successes, 37% slots empty, 26% collisions *(assumes Poisson arrival rate).*
☐ Current Use: Cellular Random Access Channel

Throughput calculations:
    S = Throughput
    G = Average number of frames attempting transmission
    Assumes Poisson Distribution
    $P(0 \text{ frames}) = e^{-G}$
    $S = G(e^{-G})$

COMPUTER COMMUNICATION NETWORKS

E = Expected # of transmissions to achieve successful transmission:

$E = e^G$

Highest throughput achieved when G = 1

S = 36.79%

E = 2.718

COMPUTER COMMUNICATION NETWORKS

CSMA/CD

Carrier Sense Multiple Access with Collision Detection (CSMA/CD): Ethernet: 802.3
- **Carrier Sense**: Before transmitting a frame the source first listens to see if someone is using the medium, and waits until the transmission is complete.
- **Multiple Access**: Multiple stations can access media concurrently – broadcast mode
    - o **1**-Persistent CSMA: As soon as channel becomes idle, transmit packet
    - o **P**-Persistent CSMA: Transmit packet with probability P as channel becomes idle – else wait one propagation delay and try again.
- **Collision Detect**:
    1. When transmitting, listen to see if own signal is received back.
    2. If collision has occurred jam and then stop the transmission.
    3. Retransmit after a random period of time.

Collision Determination: How long will it take to determine if a collision has occurred?
- Assume a 1km long coax cable.
- t = propagation delay between 2 furthest end points = 5 usec
- Assume Node 1 is at one end of cable and transmits first.
- Assume Node 2 is at other end of cable and begins to transmit just as first bit is arriving at Node 2.
- Collision detect will take twice the propagation delay of sending bits from one end to other end of cable.

Example: Ethernet
- Uses optical fiber, twisted pair or coax from 10 Mbps to 1 Gbps.

Ethernet Frame format:
1. Preamble: 10101010: synchronizes sender and receiver.
2. Start of frame delimiter: 10101011: indicates start of frame.
3. Destination address *16 or 48 bits.*
4. Source address
5. Type of above layer: IP, ARP, or RARP
    - o IEEE 802.3: Replaces type with length.
6. Data field: 0 to 1500 bytes
7. Pad: fills out to minimum 64 bytes if necessary.
8. FCS: CRC *4 octets.*

Hub: Forwards transmissions in all directions
- Has ports to multiple terminals in star configuration

- ☐ Received frames broadcast to all other lines
- ☐ If collision detected, hub jams to cause backoff algorithm
- ☐ Used with 10BaseT: Twisted pair LAN at 10 Mbps.

Dynamic Medium Access Control: Scheduling

Scheduling: Dynamic form of Time Division Multiplexing
- ☐ Assumes centrally scheduled system

Reservation

Two Periods:

Reservation Interval: Stations bid for time
- ☐ M minislots: Each minislot allocated to 1 terminal OR
- ☐ Random Access bidding

Data Transmission Interval: Packets transmitted in order (optionally: and duration and number) as specified during the Reservation Interval
- ☐ Frame transmission = 1 + v   where v = time for minislot
- ☐ Maximum throughput = 1 / (1 + v)   for fixed-size 1-frame transmissions

Example: Cellular Packet Data (General Packet Radio Services)

Polling

Central Controller polls stations
- ☐ CC: Do you have anything to send?
- ☐ Station: Yes, here it is .. I am done OR No, I have nothing to send
- ☐ Walk time: time to poll a station: overhead
- ☐ Total Walk time: Sum of Walk times for 1 polling round.

Example:
- ☐ IBM SNRM,
- ☐ Token: Traverses the ring, giving owner of token permission to transmit
  - o Packet can be removed at destination or origination

Channelization

Frequency Division Multiple Access (FDMA): A station is allocated a frequency all the time

Time Division Multiple Access (TDMA): Stations have assigned slot to transmit and thus take turns transmitting.

Code Division Multiple Access (CDMA): Station transmits over all the frequency all the time.

- [ ] Each bit is transmitted multiple times according to the chip rate, using a unique binary pseudorandom sequence, called the chip sequence.
- [ ] The received signal is correlated with the terminals chip sequence and summed since:
    - $C_1{}^2 + C_2{}^2 + C_3{}^2 + \ldots + C_G{}^2 = G$
    - Where each C is either -1 or $+1$, and $-1^2 = +1$ and $1^2 = +1$

Frequency Division Duplex (FDD): Base station and terminal transmit to each other on different frequencies.

Time Division Duplex (TDD): Base station and terminals transmit on same frequency at different times.

COMPUTER COMMUNICATION NETWORKS

Switched Networks

Switch: A form of lower-layer router
- Has ports to multiple terminals in star configuration
- Incoming frames are transferred to single appropriate outgoing port
- May buffer data (packet switch)
- Uses routing tables to forward packets to correct destination
- Multiplexes packets toward destination.
- Forwards according to layer 2 address.

Two Modes:
1) Dedicated lines in both directions to single terminal: No collisions
2) Dedicated lines in both directions to hub and multiple terminals:
    - Collisions possible on uplink.

Fast Ethernet: 100 Mbps
- Uses frame structure of IEEE 802.3 standard
- Physical Layer:
    - 100BaseT4: 4 x UTP 3: single direction.
    - 100BaseTX: 2 x UTP 5: can operate in full-duplex
    - 100BaseFX: 2 x fiber: full duplex
- Uses hubs or switches

Gigabit Ethernet: 1 Gbps speed
- Uses frame structure of IEEE 802.3 standard
- Star configuration
- Physical layer:
    - 1000BaseSX: 2 x Optical fiber @ 550 m
    - 1000BaseLX: 2 x Optical fiber @ 5 km
    - 1000BaseCX: Shielded copper @ 25 m
    - 1000BaseT: Cat. 5 UTP @ 100 m
- Problem: TX of frame can occur without recognizing collision occurred.
- Solution: Minimum slot size of 512 bytes
    - Packet Bursting: TX a burst of small packets.

IEEE 802.11: Wireless LAN

IEEE 802.11 Wireless LAN protocol.
- Supports rates of 11, 5.5, 2, 1 Mbps.
- Cannot hear any interference since transmitted signal is always the loudest signal.
- Multiple stations may send simultaneously as long as intended receiver is closest to the desired transmitting station.
- **Hidden Terminal**: A terminal which can hear transmissions from few terminals

Time divided into two periods:
Point Coordination Function (PCF): Contention Free Period
- Access Point (AP) uses polling to ensure low-delay applications get serviced regularly

Distributed Coordination Function (DCF): Contention Period
- Uses CSMA/Collision Avoidance to coordinate Random Access

CSMA/CA: Carrier Sense Multiple Access / Collision Avoidance
- Only allowed to send if no other station is sending.
- **Collision Avoidance**: Random backoff procedure
- Certain types of packets may be transmitted if no transmission has occurred in a period of T time, called an Interframe Space (IFS)
  - o SIFS or Short IFS: Can transmit Acks, poll responses, remaining packet segments
  - o PIFS: PCF IFS: Access Point can request control, then Contention Free period starts
  - o DIFS: DCF IFS: A random access packet can be transmitted
- **Reservation** algorithm includes:
  1. Send a Request To Send (RTS) packet notifying all how long the message will be.
  2. Receiver replies with Clear To Send (CTS) packet acknowledging RTS and repeating length of message.
  3. RTS/CTS notifies all neighbors within earshot of both transmitter and receiver to not transmit during the message transmission.

COMPUTER COMMUNICATION NETWORKS

Exercise on Multiple Access:

Assume transmissions are to be transmitted from the indicated terminals at the following times. Assume each frame (for simplicity sake) is 0.99 unit long. Assume any propagation delay is at most 0.1. All numbers are in time units.
Terminals are configured as follows:

```
+--------+--------------------------------+--------------------------------+--------+
A        B                                E                                C        D
```
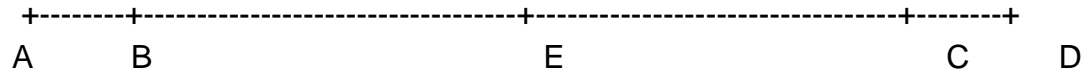
Terminal	Time Frame Generated
A->B	1.5
D->C	1.7
B->E	3.0
C->D	4.2
D->E	5.0
E->A	5.1

How many collisions will occur for:
- ☐ Aloha?
- ☐ Slotted Aloha?
- ☐ CSMA/CD?
- • IEEE 802.11?

COMPUTER COMMUNICATION NETWORKS

More LAN Vocabulary

LLC: Logical Link Control
- ☐ Addressing: Identifies the layer 3 protocol entity (IP, Novell IPX) in addition to:
- ☐ Error control:
  - o Type 1: Unacknowledged Connectionless (i.e. UI or Unnumbered Info)
  - o Type 2: Reliable Connection Oriented (i.e. SABM and I-frames)
  - o Type 3: Acknowledged Connectionless: Individual frames with acks

Layer 2 consists of:

```
        ┌──────────────┐
        │     LLC      │
        ├──────────────┤
   NIC: │     MAC      │
        └──────────────┘
```
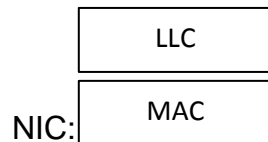
NIC: Network Interface Card or LAN Adapter Card
- ☐ Implements MAC protocol
- ☐ Transfers data in serial mode over network: performs parallel to serial conversion
- ☐ Own unique physical address is burned into ROM.
- ☐ Interface with computer with PCMCIA card or other type card into expansion slot.

Routing / Repeaters:
Repeater: Repeats at the Physical Layer:
- ☐ No intelligence: No checking of CRC

Bridge: Relay occurs at MAC layer.
- ☐ Interfaces two LANs.
- ☐ Builds routing tables from source addresses.

Router: Routing occurs at Network layer.

Application Layer: Application Gateway
Transport Layer: Transport Gateway
Network Layer: Router
Data Link Layer: Bridge, Switch
Physical Layer: Repeater, Hub

COMPUTER COMMUNICATION NETWORKS

UNIT 4

Wired Networks

Wired networks, also called Ethernet networks, are the most common type of local area network (LAN) technology. A wired network is simply a collection of two or more computers, printers, and other devices linked by Ethernet cables. Ethernet is the fastest wired network protocol, with connection speeds of 10 megabits per second (Mbps) to 100 Mbps or higher. Wired networks can also be used as part of other wired and wireless networks. To connect a computer to a network with an Ethernet cable, the computer must have an Ethernet adapter (sometimes called a network interface card, or NIC). Ethernet adapters can be internal (installed in a computer) or external (housed in a separate case). Some computers include a built-in Ethernet adapter port, which eliminates the need for a separate adapter (Microsoft). There are three basic network topologies that are most commonly used today. (Homenthelp.com)

The star network, a general more simplistic type of topology, has one central hub that connects to three or more computers and the ability to network printers. This type can be used for small businesses and even home networks. The star network is very useful for applications where some processing must be centralized and some must be performed locally. The major disadvantage is the star network is its vulnerability. All data must pass through one central host computer and if that host fails the entire network will fail.

On the other hand the bus network has no central computer and all computers are linked on a single circuit. This type broadcasts signals in all directions and it uses special software to identify which computer gets what signal. One disadvantage with this type of network is that only one signal can be sent at one time, if two signals are sent at the same time they will collide and the signal will fail to reach its destination. One advantage is that there is no central computer so if one computer goes down others will not be affected and will be able to send messages to one another. (Laudon)

The third type of network is the ring network. Similar to the bus network, the ring network does not rely on a central host computer either. Each computer in the network can communicate directly with any other computer, and each processes its own applications independently. A ring network forms a closed loop and
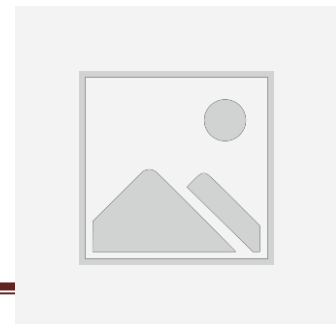
data is sent in one direction only and if a computer in the network fails the data is still able to be transmitted.

Typically the range of a wired network is within a 2,000-foot-radius. The disadvantage of this is that data transmission over this distance may be slow or nonexistent. The benefit of a wired network is that bandwidth is very high and that interference is very limited through direct connections. Wired networks are more secure and can be used in many situations; corporate LANs, school networks and hospitals. The biggest drawback to this type of network is that it must be rewired every time it is moved.

Wireless Networks

A wireless network, which uses high-frequency radio waves rather than wires to communicate between nodes, is another option for home or business networking. Individuals and organizations can use this option to expand their existing wired network or to go completely wireless. Wireless allows for devices to be shared without networking cable which increases mobility but decreases range. There are two main types of wireless networking; peer to peer or ad-hoc and infrastructure. (Wi-fi.com)

An ad-hoc or peer-to-peer wireless network consists of a number of computers each equipped with a wireless networking interface card. Each computer can communicate directly with all of the other wireless enabled computers. They can share files and printers this way, but may not be able to access wired LAN resources, unless one of the computers acts as a bridge to the wired LAN using special software.

An infrastructure wireless network consists of an access point or a base station. In this type of network the access point acts like a hub, providing connectivity for the wireless computers. It can connect or bridge the wireless LAN to a wired LAN, allowing wireless computer access to LAN resources, such as file servers or existing Internet Connectivity. (compnetworking.about.com)

There are four basic types of transmissions standards for wireless networking. These types are produced by the Institute of Electrical and Electronic Engineers (IEEE). These standards define all aspects of radio frequency wireless networking. They have established four transmission standards; 802.11, 802.11a, 802.11b, 802.11g.

The basic differences between these four types are connection speed and radio frequency. 802.11 and 802.11b are the slowest at 1 or 2 Mbps and 5.5 and 11Mbps respectively. They both

operate off of the 2.4 GHz radio frequency. 802.11a operates off of a 5 GHz frequency and can transmit up to 54 Mbps and the 802.11g operates off of the 2.4 GHz frequency and can transmit up to 54 Mbps. Actual transmission speeds vary depending on such factors as the number and size of the physical barriers within the network and any interference in the radio transmissions. (Wi-fi.com)

Wireless networks are reliable, but when interfered with it can reduce the range and the quality of the signal. Interference can be caused by other devices operating on the same radio frequency and it is very hard to control the addition of new devices on the same frequency. Usually if your wireless range is compromised considerably, more than likely, interference is to blame. (Laudon)

A major cause of interference with any radio signals are the materials in your surroundings, especially metallic substances, which have a tendency to reflect radio signals. Needless to say, the potential sources of metal around a home are numerous--things like metal studs, nails, building insulation with a foil backing and even lead paint can all possibly reduce the quality of the wireless radio signal. Materials with a high density, like concrete, tend to be harder for radio signals to penetrate, absorbing more of the energy. Other devices utilizing the same frequency can also result in interference with your wireless. For example, the 2.4GHz frequency used by 802.11b-based wireless products to communicate with each other. Wireless devices don't have this frequency all to themselves. In a business environment, other devices that use the 2.4GHz band include microwave ovens and certain cordless phones. (Laundon)

On the other hand, many wireless networks can increase the range of the signal by using many different types of hardware devices. A wireless extender can be used to relay the radio frequency from one point to another without losing signal strength. Even though this device extends the range of a wireless signal it has some drawbacks. One drawback is that it extends the signal, but the transmission speed will be slowed.

There are many benefits to a wireless network. The most important one is the option to expand your current wired network to other areas of your organization where it would otherwise not be cost effective or practical to do so. An organization can also install a wireless network without physically disrupting the current workplace or wired network. (Wi-Fi.org) Wireless networks are far easier to move than a wired network and adding users to an existing wireless network is easy. Organizations opt for a wireless network in conference rooms, lobbies and offices where adding to the existing wired network may be too expensive to do so.

Wired vs. Wireless Networking

The biggest difference between these two types of networks is one uses network cables and one uses radio frequencies. A wired network allows for a faster and more secure connection

and can only be used for distances shorter than 2,000 feet. A wireless network is a lot less secure and transmission speeds can suffer from outside interference. Although wireless networking is a lot more mobile than wired networking the range of the network is usually 150-300 indoors and up to 1000 feet outdoors depending on the terrain. (Homelanextream.com)

The cost for wired networking has become rather inexpensive. Ethernet cables, hubs and switches are very inexpensive. Some connection sharing software packages, like ICS, are free; some cost a nominal fee. Broadband routers cost more, but these are optional components of a wired network, and their higher cost is offset by the benefit of easier installation and built-in security features.

Wireless gear costs somewhat more than the equivalent wired Ethernet products. At full retail prices, wireless adapters and access points may cost three or four times as much as Ethernet cable adapters and hubs/switches, respectively. 802.11b products have dropped in price considerably with the release of 802.11g. (Homelanextream.com)

Wired LANs offer superior performance. A traditional Ethernet connection offers only 10 Mbps bandwidth, but 100 Mbps Fast Ethernet technology costs a little more and is readily available. Fast Ethernet should be sufficient for file sharing, gaming, and high-speed Internet access for many years into the future. (Wi-Fi.org) Wired LANs utilizing hubs can suffer performance slowdown if computers heavily utilize the network simultaneously. Use Ethernet switches instead of hubs to avoid this problem; a switch costs little more than a hub.

Wireless networks using 802.11b support a maximum bandwidth of 11 Mbps, roughly the same as that of old, traditional Ethernet. 802.11a and 802.11g LANs support 54 Mbps, that is approximately one-half the bandwidth of Fast Ethernet. Furthermore, wireless networking performance is distance sensitive, meaning that maximum performance will degrade on computers farther away from the access point or other communication endpoint. As more wireless devices utilize the 802.11 LAN more heavily, performance degrades even further. (Wi-Fi.org)

The greater mobility of wireless LANs helps offset the performance disadvantage. Mobile computers do not need to be tied to an Ethernet cable and can roam freely within the wireless network range. However, many computers are larger desktop models, and even mobile computers must sometimes be tied to an electrical cord and outlet for power. This undermines the mobility advantage of wireless networks in many organizations and homes.

For any wired network connected to the Internet, firewalls are the primary security consideration. Wired Ethernet hubs and switches do not support firewalls. However, firewall software products like Zone Alarm can be installed on the computers themselves. Broadband routers offer equivalent firewall capability built into the device, configurable through its own software.

COMPUTER COMMUNICATION NETWORKS

In theory, wireless LANs are less secure than wired LANs, because wireless communication signals travel through the air and can easily be intercepted. The weaknesses of wireless security are more theoretical than practical. (Wi-Fi.org) Wireless networks protect their data through the Wired Equivalent Privacy (WEP) encryption standard that makes wireless communications reasonably as safe as wired ones.

No computer network is completely secure. Important security considerations for organizations tend to not be related to whether the network is wired or wireless but rather ensuring that the firewall is properly configured, employees are aware of the dangers of spoof emails, they are away of spy ware and how to avoid and that anyone outside the organization does not have unauthorized access to the network.

Ethernet Network Topologies and Structures

LANs take on many topological configurations, but regardless of their size or complexity, all will be a combination of only three basic interconnection structures or network building blocks.

The simplest structure is the point-to-point interconnection, shown in Figure: Example Point-to-Point Interconnection. Only two network units are involved, and the connection may be DTE-to-DTE, DTE-to-DCE, or DCE-to-DCE. The cable in point-to-point interconnections is known as a network link. The maximum allowable length of the link depends on the type of cable and the transmission method that is used.

Figure: Example Point-to-Point Interconnection

The original Ethernet networks were implemented with a coaxial bus structure, as shown in Figure: Example Coaxial Bus Topology. Segment lengths were limited to 500 meters, and up to 100 stations could be connected to a single segment. Individual segments could be interconnected with repeaters, as long as multiple paths did not exist between any two stations on the network and the number of DTEs did not exceed 1024. The total path distance between the most-distant pair of stations was also not allowed to exceed a maximum prescribed value.

COMPUTER COMMUNICATION NETWORKS

Figure: Example Coaxial Bus Topology
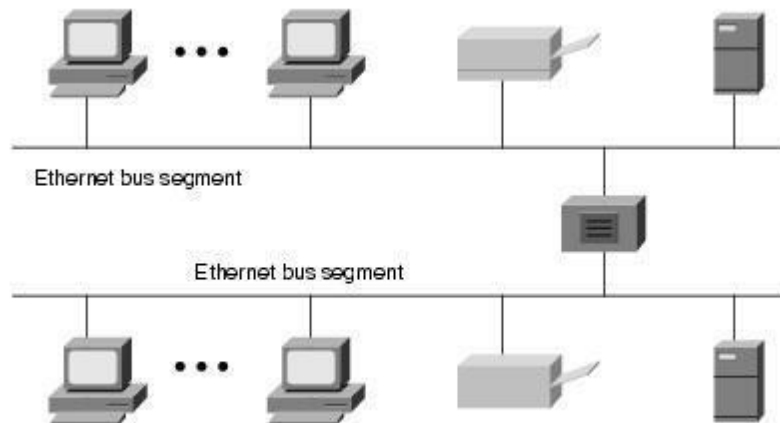
Although new networks are no longer connected in a bus configuration, some older bus-connected networks do still exist and are still useful.

Since the early 1990s, the network configuration of choice has been the star-connected topology, shown in Figure: Example Star-Connected Topology. The central network unit is either a multiport repeater (also known as a hub) or a network switch. All connections in a star network are point-to-point links implemented with either twisted-pair or optical fiber cable.

Figure: Example Star-Connected Topology
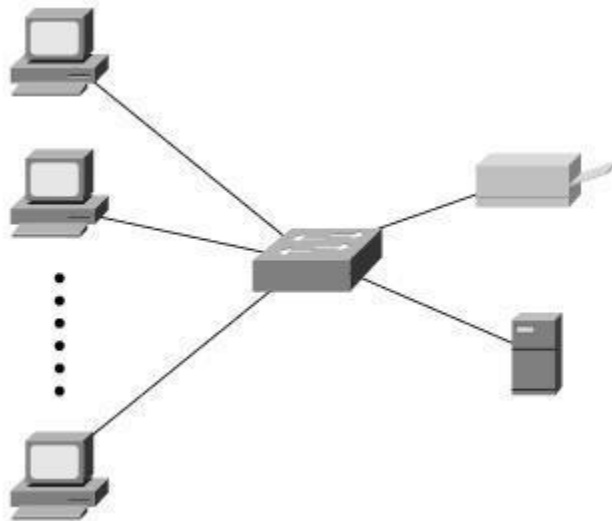

The IEEE 802.3 Logical Relationship to the OSI Reference Model

Figure: Ethernet's Logical Relationship to the OSI Reference Model shows the IEEE 802.3 logical layers and their relationship to the OSI reference model. As with all IEEE 802 protocols, the OSI data link layer is divided into two IEEE 802 sublayers, the Media Access Control

(MAC) sublayer and the MAC-client sublayer. The IEEE 802.3 physical layer corresponds to the OSI physical layer.

Figure: Ethernet's Logical Relationship to the OSI Reference Model
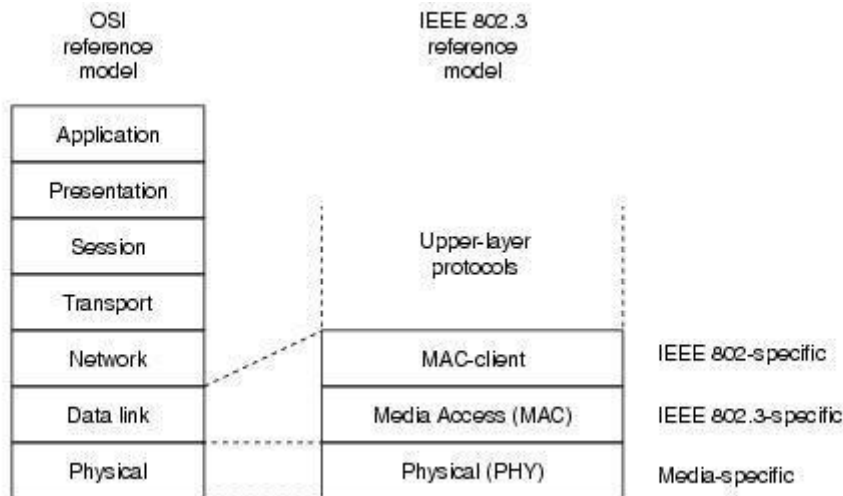
The MAC-client sublayer may be one of the following:

- Logical Link Control (LLC), if the unit is a DTE. This sublayer provides the interface between the Ethernet MAC and the upper layers in the protocol stack of the end station. The LLC sublayer is defined by IEEE 802.2 standards.
- Bridge entity, if the unit is a DCE. Bridge entities provide LAN-to-LAN interfaces between LANs that use the same protocol (for example, Ethernet to Ethernet) and also between different protocols (for example, Ethernet to Token Ring). Bridge entities are defined by IEEE 802.1 standards.

Because specifications for LLC and bridge entities are common for all IEEE 802 LAN protocols, network compatibility becomes the primary responsibility of the particular network protocol.

Figure: MAC and Physical Layer Compatibility Requirements for Basic Data Communication shows different compatibility requirements imposed by the MAC and physical levels for basic data communication over an Ethernet link.

COMPUTER COMMUNICATION NETWORKS

Figure: MAC and Physical Layer Compatibility Requirements for Basic Data Communication
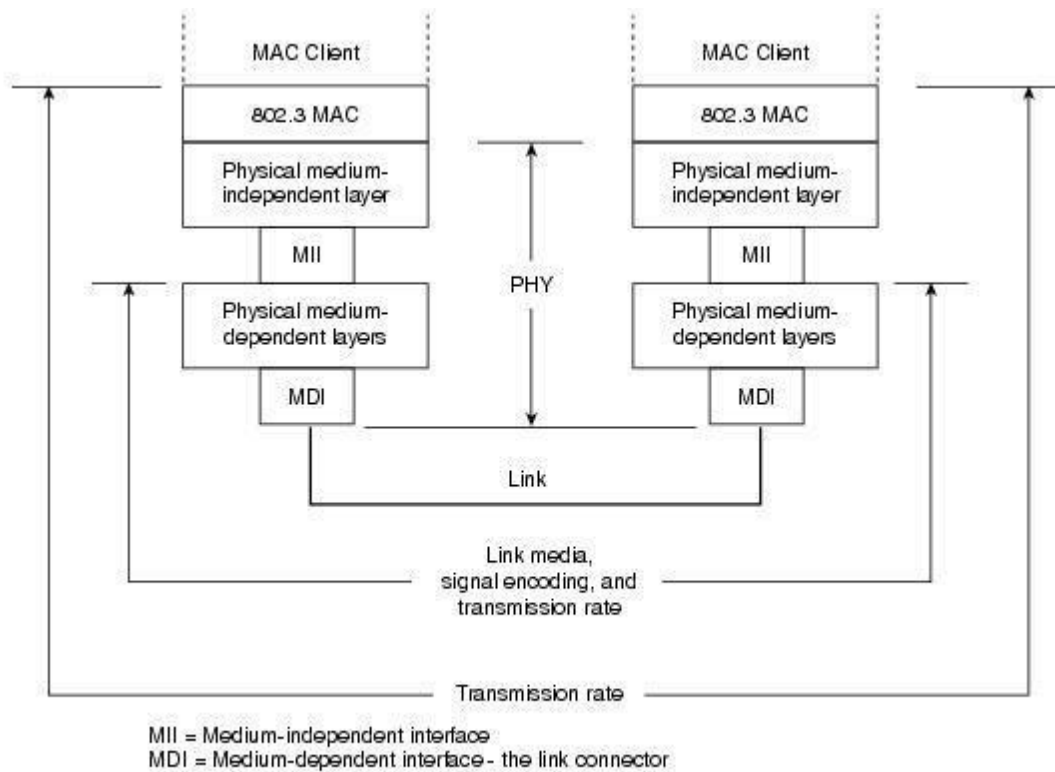
The MAC layer controls the node's access to the network media and is specific to the individual protocol. All IEEE 802.3 MACs must meet the same basic set of logical requirements, regardless of whether they include one or more of the defined optional protocol extensions. The only requirement for basic communication (communication that does not require optional protocol extensions) between two network nodes is that both MACs must support the same transmission rate.

The 802.3 physical layer is specific to the transmission data rate, the signal encoding, and the type of media interconnecting the two nodes. Gigabit Ethernet, for example, is defined to operate over either twisted-pair or optical fiber cable, but each specific type of cable or signal-encoding procedure requires a different physical layer implementation.

The Ethernet MAC Sublayer

The MAC sublayer has two primary responsibilities:

- Data encapsulation, including frame assembly before transmission, and frame parsing/error detection during and after reception
- Media access control, including initiation of frame transmission and recovery from transmission failure

COMPUTER COMMUNICATION NETWORKS

The Basic Ethernet Frame Format

The IEEE 802.3 standard defines a basic data frame format that is required for all MAC implementations, plus several additional optional formats that are used to extend the protocol's basic capability. The basic data frame format contains the seven fields shown in Figure: The Basic IEEE 802.3 MAC Data Frame Format.

- **Preamble (PRE)** - Consists of 7 bytes. The PRE is an alternating pattern of ones and zeros that tells receiving stations that a frame is coming, and that provides a means to synchronize the frame-reception portions of receiving physical layers with the incoming bit stream.

- **Start-of-frame delimiter (SOF)** - Consists of 1 byte. The SOF is an alternating pattern of ones and zeros, ending with two consecutive 1-bits indicating that the next bit is the left-most bit in the left-most byte of the destination address.

- **Destination address (DA)** - Consists of 6 bytes. The DA field identifies which station(s) should receive the frame. The left-most bit in the DA field indicates whether the address is an individual address (indicated by a 0) or a group address (indicated by a 1). The second bit from the left indicates whether the DA is globally administered (indicated by a 0) or locally administered (indicated by a 1). The remaining 46 bits are a uniquely assigned value that identifies a single station, a defined group of stations, or all stations on the network.

- **Source addresses (SA)** - Consists of 6 bytes. The SA field identifies the sending station. The SA is always an individual address and the left-most bit in the SA field is always 0.

- **Length/Type** - Consists of 2 bytes. This field indicates either the number of MAC-client data bytes that are contained in the data field of the frame, or the frame type ID if the frame is assembled using an optional format. If the Length/Type field value is less than or equal to 1500, the number of LLC bytes in the Data field is equal to the Length/Type field value. If the Length/Type field value is greater than 1536, the frame is an optional type frame, and the Length/Type field value identifies the particular type of frame being sent or received.

- **Data** - Is a sequence of n bytes of any value, where n is less than or equal to 1500. If the length of the Data field is less than 46, the Data field must be extended by adding a filler (a pad) sufficient to bring the Data field length to 46 bytes.

- **Frame check sequence (FCS)** - Consists of 4 bytes. This sequence contains a 32-bit cyclic redundancy check (CRC) value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames. The FCS is generated over the DA, SA, Length/Type, and Data fields.

COMPUTER COMMUNICATION NETWORKS

Figure: The Basic IEEE 802.3 MAC Data Frame Format
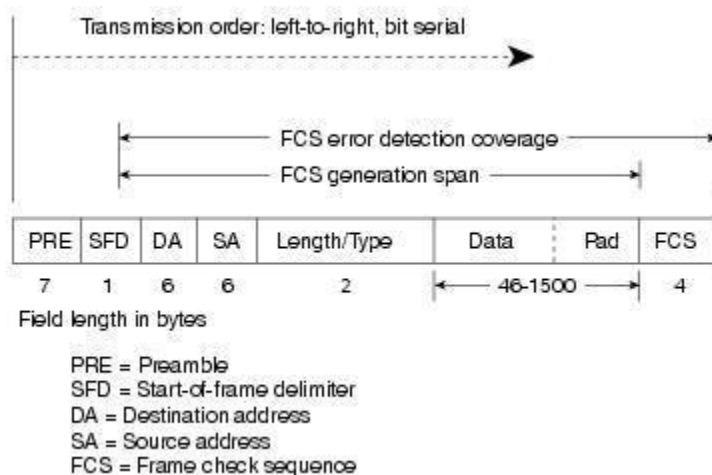
{{note:Individual addresses are also known as unicast addresses because they refer to a single MAC and are assigned by the NIC manufacturer from a block of addresses allocated by the IEEE. Group addresses (a.k.a. multicast addresses) identify the end stations in a workgroup and are assigned by the network manager. A special group address (all 1s-the broadcast address) indicates all stations on the network.}}

Frame Transmission

Whenever an end station MAC receives a transmit-frame request with the accompanying address and data information from the LLC sublayer, the MAC begins the transmission sequence by transferring the LLC information into the MAC frame buffer.

- The preamble and start-of-frame delimiter are inserted in the PRE and SOF fields.
- The destination and source addresses are inserted into the address fields.
- The LLC data bytes are counted, and the number of bytes is inserted into the Length/Type field.
- The LLC data bytes are inserted into the Data field. If the number of LLC data bytes is less than 46, a pad is added to bring the Data field length up to 46.
- An FCS value is generated over the DA, SA, Length/Type, and Data fields and is appended to the end of the Data field.

After the frame is assembled, actual frame transmission will depend on whether the MAC is operating in half-duplex or full-duplex mode.

The IEEE 802.3 standard currently requires that all Ethernet MACs support half-duplex operation, in which the MAC can be either transmitting or receiving a frame, but it cannot be doing both simultaneously. Full-duplex operation is an optional MAC capability that allows the MAC to transmit and receive frames simultaneously.

COMPUTER COMMUNICATION NETWORKS

Half-Duplex Transmission-The CSMA/CD Access Method

The CSMA/CD protocol was originally developed as a means by which two or more stations could share a common media in a switch-less environment when the protocol does not require central arbitration, access tokens, or assigned time slots to indicate when a station will be allowed to transmit. Each Ethernet MAC determines for itself when it will be allowed to send a frame.

The CSMA/CD access rules are summarized by the protocol's acronym:

- **Carrier sense** - Each station continuously listens for traffic on the medium to determine when gaps between frame transmissions occur.
- **Multiple access** - Stations may begin transmitting any time they detect that the network is quiet (there is no traffic).
- **Collision detect** - If two or more stations in the same CSMA/CD network (collision domain) begin transmitting at approximately the same time, the bit streams from the transmitting stations will interfere (collide) with each other, and both transmissions will be unreadable. If that happens, each transmitting station must be capable of detecting that a collision has occurred before it has finished sending its frame. Each must stop transmitting as soon as it has detected the collision and then must wait a quasirandom length of time (determined by a back-off algorithm) before attempting to retransmit the frame.

The worst-case situation occurs when the two most-distant stations on the network both need to send a frame and when the second station does not begin transmitting until just before the frame from the first station arrives. The collision will be detected almost immediately by the second station, but it will not be detected by the first station until the corrupted signal has propagated all the way back to that station. The maximum time that is required to detect a collision (the collision window, or "slot time") is approximately equal to twice the signal propagation time between the two most-distant stations on the network.

This means that both the minimum frame length and the maximum collision diameter are directly related to the slot time. Longer minimum frame lengths translate to longer slot times and larger collision diameters; shorter minimum frame lengths correspond to shorter slot times and smaller collision diameters.

The trade-off was between the need to reduce the impact of collision recovery and the need for network diameters to be large enough to accommodate reasonable network sizes. The compromise was to choose a maximum network diameter (about 2500 meters) and then to set the minimum frame length long enough to ensure detection of all worst-case collisions.

The compromise worked well for 10 Mbps, but it was a problem for higher data-rate Ethernet developers. Fast Ethernet was required to provide backward compatibility with earlier Ethernet networks, including the existing IEEE 802.3 frame format and error-detection procedures, plus all applications and networking software running on the 10-Mbps networks.

COMPUTER COMMUNICATION NETWORKS

Although signal propagation velocity is essentially constant for all transmission rates, the time required to transmit a frame is inversely related to the transmission rate. At 100 Mbps, a minimum-length frame can be transmitted in approximately one-tenth of the defined slot time, and any collision that occurred during the transmission would not likely be detected by the transmitting stations. This, in turn, meant that the maximum network diameters specified for 10-Mbps networks could not be used for 100-Mbps networks. The solution for Fast Ethernet was to reduce the maximum network diameter by approximately a factor of 10 (to a little more than 200 meters).

The same problem also arose during specification development for Gigabit Ethernet, but decreasing network diameters by another factor of 10 (to approximately 20 meters) for 1000-Mbps operation was simply not practical. This time, the developers elected to maintain approximately the same maximum collision domain diameters as 100-Mbps networks and to increase the apparent minimum frame size by adding a variable-length nondata extension field to frames that are shorter than the minimum length (the extension field is removed during frame reception).

Figure: MAC Frame with Gigabit Carrier Extension shows the MAC frame format with the gigabit extension field, and the following table shows the effect of the trade-off between the transmission data rate and the minimum frame size for 10-Mbps, 100-Mbps, and 1000-Mbps Ethernet.

Figure: MAC Frame with Gigabit Carrier Extension
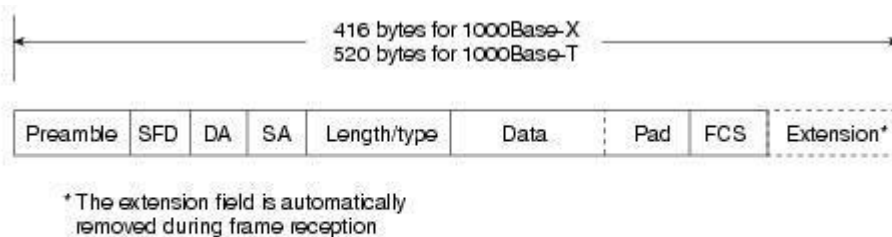
Table: Limits for Half-Duplex Operation

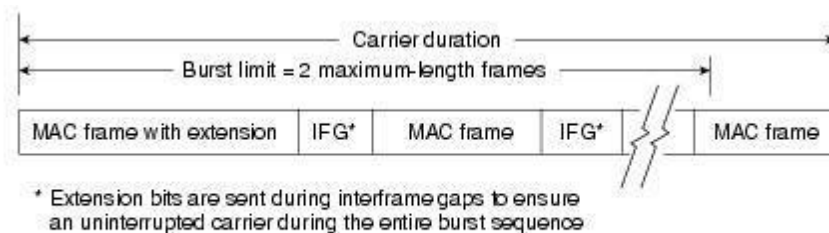| Parameter	10 Mbps	100 Mbps	1000 Mbps
Minimum frame size	64 bytes	64 bytes	520 bytes (with extension field added)
Maximum collision diameter, DTE to DTE	100 meters UTP	100 meters UTP	100 meters UTP

		412 meters fiber	316 meters fiber
Maximum collision diameter with repeaters	2500 meters	205 meters	200 meters
Maximum number of repeaters in network path	5	2	1

Note: 520 bytes applies to 1000Base-T implementations. The minimum frame size with extension field for 1000Base-X is reduced to 416 bytes because 1000Base-X encodes and transmits 10 bits for each byte.

Another change to the Ethernet CSMA/CD transmit specification was the addition of frame bursting for gigabit operation. Burst mode is a feature that allows a MAC to send a short sequence (a burst) of frames equal to approximately 5.4 maximum-length frames without having to relinquish control of the medium. The transmitting MAC fills each interframe interval with extension bits, as shown in Figure: A Gigabit Frame-Burst Sequence, so that other stations on the network will see that the network is busy and will not attempt transmission until after the burst is complete.

Figure: A Gigabit Frame-Burst Sequence

* Extension bits are sent during interframe gaps to ensure an uninterrupted carrier during the entire burst sequence

If the length of the first frame is less than the minimum frame length, an extension field is added to extend the frame length to the value indicated in Table: Limits for Half-Duplex Operation. Subsequent frames in a frame-burst sequence do not need extension fields, and a frame burst may continue as long as the burst limit has not been reached. If the burst limit is reached after a frame transmission has begun, transmission is allowed to continue until that entire frame has been sent.

Frame extension fields are not defined, and burst mode is not allowed for 10 Mbps and 100 Mbps transmission rates.

COMPUTER COMMUNICATION NETWORKS

Full-Duplex Transmission-An Optional Approach to Higher Network Efficiency

Full-duplex operation is an optional MAC capability that allows simultaneous two-way transmission over point-to-point links. Full duplex transmission is functionally much simpler than half-duplex transmission because it involves no media contention, no collisions, no need to schedule retransmissions, and no need for extension bits on the end of short frames. The result is not only more time available for transmission, but also an effective doubling of the link bandwidth because each link can now support full-rate, simultaneous, two-way transmission.

Transmission can usually begin as soon as frames are ready to send. The only restriction is that there must be a minimum-length interframe gap between successive frames, as shown in Figure: Full Duplex Operation Allows Simultaneous Two-Way Transmission on the Same Link, and each frame must conform to Ethernet frame format standards.

Figure: Full Duplex Operation Allows Simultaneous Two-Way Transmission on the Same Link
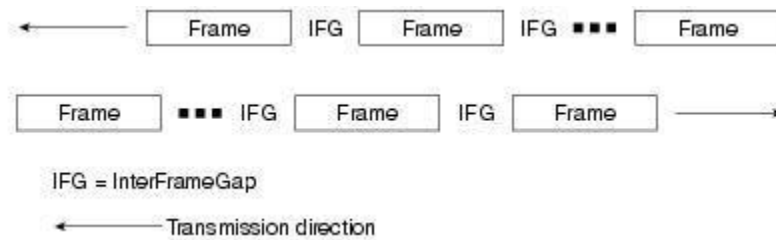
Flow Control

Full-duplex operation requires concurrent implementation of the optional flow-control capability that allows a receiving node (such as a network switch port) that is becoming congested to request the sending node (such as a file server) to stop sending frames for a selected short period of time. Control is MAC-to-MAC through the use of a pause frame that is automatically generated by the receiving MAC. If the congestion is relieved before the requested wait has expired, a second pause frame with a zero time-to-wait value can be sent to request resumption of transmission. An overview of the flow control operation is shown in Figure: An Overview of the IEEE 802.3 Flow Control Sequence.

COMPUTER COMMUNICATION NETWORKS

Figure: An Overview of the IEEE 802.3 Flow Control Sequence
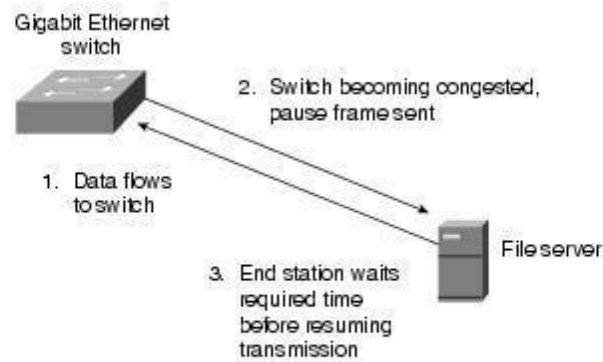
The full-duplex operation and its companion flow control capability are both options for all Ethernet MACs and all transmission rates. Both options are enabled on a link-by-link basis, assuming that the associated physical layers are also capable of supporting full-duplex operation.

Pause frames are identified as MAC control frames by an exclusive assigned (reserved) length/type value. They are also assigned a reserved destination address value to ensure that an incoming pause frame is never forwarded to upper protocol layers or to other ports in a switch.

Frame Reception

Frame reception is essentially the same for both half-duplex and full-duplex operations, except that full-duplex MACs must have separate frame buffers and data paths to allow for simultaneous frame transmission and reception.

Frame reception is the reverse of frame transmission. The destination address of the received frame is checked and matched against the station's address list (its MAC address, its group addresses, and the broadcast address) to determine whether the frame is destined for that station. If an address match is found, the frame length is checked and the received FCS is compared to the FCS that was generated during frame reception. If the frame length is okay and there is an FCS match, the frame type is determined by the contents of the Length/Type field. The frame is then parsed and forwarded to the appropriate upper layer.

The VLAN Tagging Option

VLAN tagging is a MAC option that provides three important capabilities not previously available to Ethernet network users and network managers:

▪ Provides a means to expedite time-critical network traffic by setting transmission priorities for outgoing frames.

- Allows stations to be assigned to logical groups, to communicate across multiple LANs as though they were on a single LAN. Bridges and switches filter destination addresses and forward VLAN frames only to ports that serve the VLAN to which the traffic belongs.
- Simplifies network management and makes adds, moves, and changes easier to administer.

A VLAN-tagged frame is simply a basic MAC data frame that has had a 4-byte VLAN header inserted between the SA and Length/Type fields, as shown in Figure: VLAN-Tagged Frames Are Identified When the MAC Finds the LAN Type Value in the Normal Length/Type Field Location.

Figure: VLAN-Tagged Frames Are Identified When the MAC Finds the LAN Type Value in the Normal Length/Type Field Location
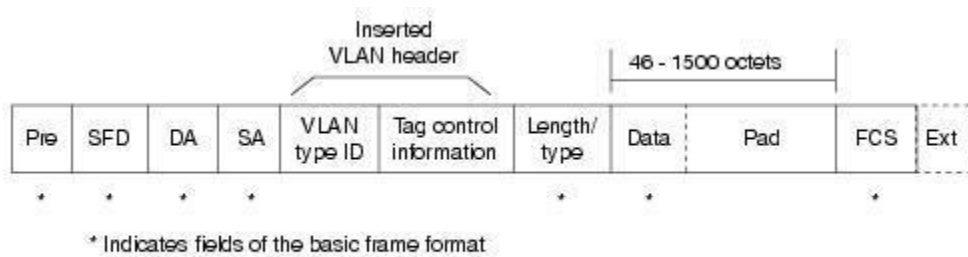
The VLAN header consists of two fields:

- A reserved 2-byte type value, indicating that the frame is a VLAN frame
- A two-byte Tag-Control field that contains both the transmission priority (0 to 7, where 7 is the highest) and a VLAN ID that identifies the particular VLAN over which the frame is to be sent

The receiving MAC reads the reserved type value, which is located in the normal Length/Type field position, and interprets the received frame as a VLAN frame. Then the following occurs:

- If the MAC is installed in a switch port, the frame is forwarded according to its priority level to all ports that are associated with the indicated VLAN identifier.
- If the MAC is installed in an end station, it removes the 4-byte VLAN header and processes the frame in the same manner as a basic data frame.

VLAN tagging requires that all network nodes involved with a VLAN group be equipped with the VLAN option.

The Ethernet Physical Layers

Because Ethernet devices implement only the bottom two layers of the OSI protocol stack, they are typically implemented as network interface cards (NICs) that plug into the host device's motherboard. The different NICs are identified by a three-part product name that is based on the physical layer attributes.

COMPUTER COMMUNICATION NETWORKS

The naming convention is a concatenation of three terms indicating the transmission rate, the transmission method, and the media type/signal encoding. For example, consider this:

- 10Base-T = 10 Mbps, baseband, over two twisted-pair cables
- 100Base-T2 = 100 Mbps, baseband, over two twisted-pair cables
- 100Base-T4 = 100 Mbps, baseband, over four-twisted pair cables
- 1000Base-LX = 1000 Mbps, baseband, long wavelength over optical fiber cable

A question sometimes arises as to why the middle term always seems to be "Base." Early versions of the protocol also allowed for broadband transmission (for example, 10Broad), but broadband implementations were not successful in the marketplace. All current Ethernet implementations use baseband transmission.

Encoding for Signal Transmission

In baseband transmission, the frame information is directly impressed upon the link as a sequence of pulses or data symbols that are typically attenuated (reduced in size) and distorted (changed in shape) before they reach the other end of the link. The receiver's task is to detect each pulse as it arrives and then to extract its correct value before transferring the reconstructed information to the receiving MAC.

Filters and pulse-shaping circuits can help restore the size and shape of the received waveforms, but additional measures must be taken to ensure that the received signals are sampled at the correct time in the pulse period and at same rate as the transmit clock:

- The receive clock must be recovered from the incoming data stream to allow the receiving physical layer to synchronize with the incoming pulses.
- Compensating measures must be taken for a transmission effect known as baseline wander.

Clock recovery requires level transitions in the incoming signal to identify and synchronize on pulse boundaries. The alternating 1s and 0s of the frame preamble were designed both to indicate that a frame was arriving and to aid in clock recovery. However, recovered clocks can drift and possibly lose synchronization if pulse levels remain constant and there are no transitions to detect (for example, during long strings of 0s).

Baseline wander results because Ethernet links are AC-coupled to the transceivers and because AC coupling is incapable of maintaining voltage levels for more than a short time. As a result, transmitted pulses are distorted by a droop effect similar to the exaggerated example shown in Figure: A Concept Example of Baseline Wander. In long strings of either 1s or 0s, the droop can become so severe that the voltage level passes through the decision threshold, resulting in erroneous sampled values for the affected pulses.

COMPUTER COMMUNICATION NETWORKS

Figure: A Concept Example of Baseline Wander
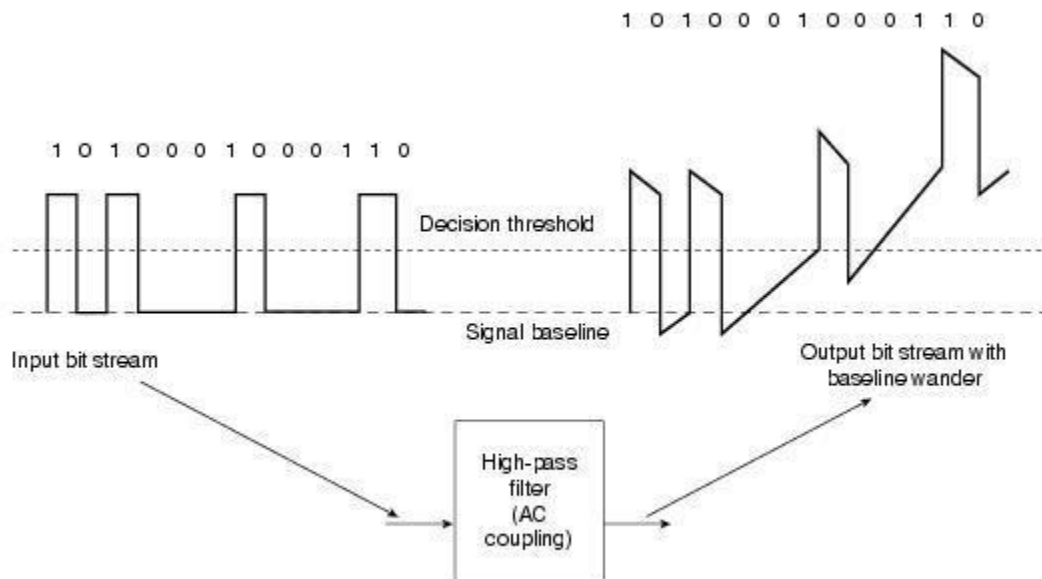
Fortunately, encoding the outgoing signal before transmission can significantly reduce the effect of both these problems, as well as reduce the possibility of transmission errors. Early Ethernet implementations, up to and including 10Base-T, all used the Manchester encoding method, shown in Figure: Transition-Based Manchester Binary Encoding. Each pulse is clearly identified by the direction of the midpulse transition rather than by its sampled level value.

Figure: Transition-Based Manchester Binary Encoding
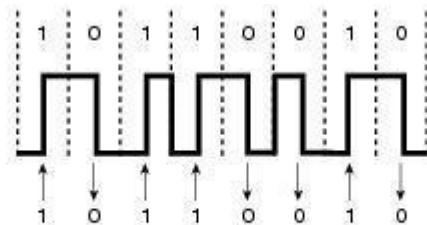
Unfortunately, Manchester encoding introduces some difficult frequency-related problems that make it unsuitable for use at higher data rates. Ethernet versions subsequent to 10Base-T all use different encoding procedures that include some or all of the following techniques:

- **Using data scrambling** - A procedure that scrambles the bits in each byte in an orderly (and recoverable) manner. Some 0s are changed to 1s, some 1s are changed to 0s, and some bits are left the same. The result is reduced run-length of same-value bits, increased transition density, and easier clock recovery.

- **Expanding the code space** - A technique that allows assignment of separate codes for data and control symbols (such as start-of-stream and end-of-stream delimiters, extension bits, and so on) and that assists in transmission error detection.
- **Using forward error-correcting codes** - An encoding in which redundant information is added to the transmitted data stream so that some types of transmission errors can be corrected during frame reception.

🖉 **Note**: Forward error-correcting codes are used in 1000Base-T to achieve an effective reduction in the bit error rate. Ethernet protocol limits error handling to detection of bit errors in the received frame. Recovery of frames received with uncorrectable errors or missing frames is the responsibility of higher layers in the protocol stack.

The 802.3 Physical Layer Relationship to the ISO Reference Model

Although the specific logical model of the physical layer may vary from version to version, all Ethernet NICs generally conform to the generic model shown in Figure: The Generic Ethernet Physical Layer Reference Model:

Figure: The Generic Ethernet Physical Layer Reference Model
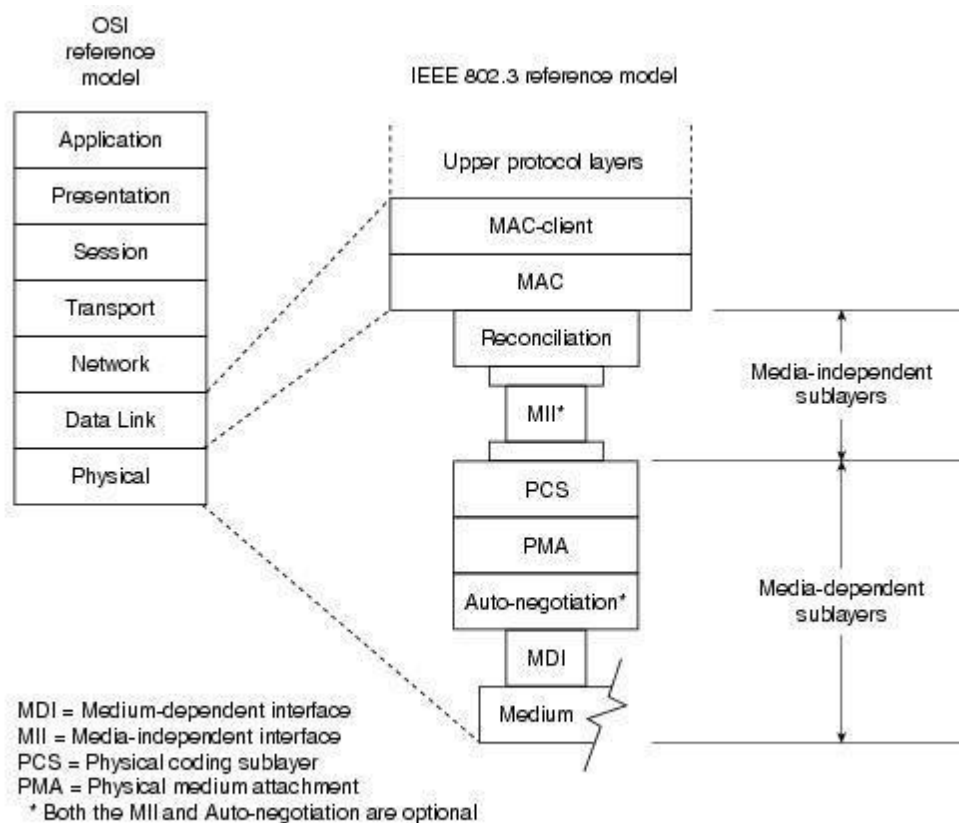
COMPUTER COMMUNICATION NETWORKS

The physical layer for each transmission rate is divided into sublayers that are independent of the particular media type and sublayers that are specific to the media type or signal encoding.

- The reconciliation sublayer and the optional media-independent interface (MII in 10-Mbps and 100-Mbps Ethernet, GMII in Gigabit Ethernet) provide the logical connection between the MAC and the different sets of media-dependent layers. The MII and GMII are defined with separate transmit and receive data paths that are bit-serial for 10-Mbps implementations, nibble-serial (4 bits wide) for 100-Mbps implementations, and byte-serial (8 bits wide) for 1000-Mbps implementations. The media-independent interfaces and the reconciliation sublayer are common for their respective transmission rates and are configured for full-duplex operation in 10Base-T and all subsequent Ethernet versions.
- The media-dependent physical coding sublayer (PCS) provides the logic for encoding, multiplexing, and synchronization of the outgoing symbol streams as well symbol code alignment, demultiplexing, and decoding of the incoming data.
- The physical medium attachment (PMA) sublayer contains the signal transmitters and receivers (transceivers), as well as the clock recovery logic for the received data streams.
- The medium-dependent interface (MDI) is the cable connector between the signal transceivers and the link.
- The Auto-negotiation sublayer allows the NICs at each end of the link to exchange information about their individual capabilities, and then to negotiate and select the most favorable operational mode that they both are capable of supporting. Auto-negotiation is optional in early Ethernet implementations and is mandatory in later versions.

Depending on which type of signal encoding is used and how the links are configured, the PCS and PMA may or may not be capable of supporting full-duplex operation.

10-Mbps Ethernet-10Base-T

10Base-T provides Manchester-encoded 10-Mbps bit-serial communication over two unshielded twisted-pair cables. Although the standard was designed to support transmission over common telephone cable, the more typical link configuration is to use two pair of a four-pair Category 3 or 5 cable, terminated at each NIC with an 8-pin RJ-45 connector (the MDI), as shown in Figure: The Typical 10Base-T Link Is a Four-Pair UTP Cable in Which Two Pairs Are Not Used pair is configured as a simplex link where transmission is in one direction only, the 10Base-T physical layers can support either half-duplex or full-duplex operation.

Figure: The Typical 10Base-T Link Is a Four-Pair UTP Cable in Which Two Pairs Are Not Used
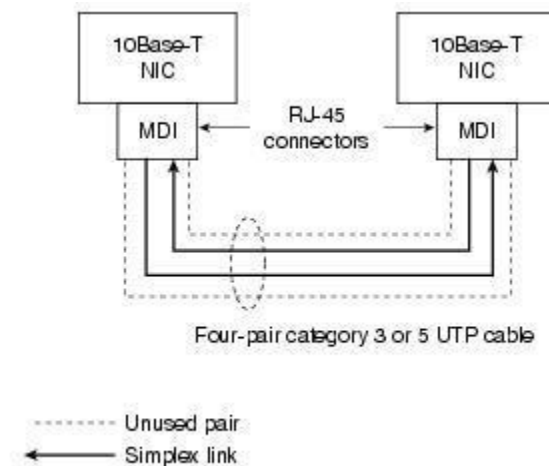
Although 10Base-T may be considered essentially obsolete in some circles, it is included here because there are still many 10Base-T Ethernet networks, and because full-duplex operation has given 10BaseT an extended life.

10Base-T was also the first Ethernet version to include a link integrity test to determine the health of the link. Immediately after powerup, the PMA transmits a normal link pulse (NLP) to tell the NIC at the other end of the link that this NIC wants to establish an active link connection:

- If the NIC at the other end of the link is also powered up, it responds with its own NLP.
- If the NIC at the other end of the link is not powered up, this NIC continues sending an NLP about once every 16 ms until it receives a response.

The link is activated only after both NICs are capable of exchanging valid NLPs.

100 Mbps-Fast Ethernet

Increasing the Ethernet transmission rate by a factor of ten over 10Base-T was not a simple task, and the effort resulted in the development of three separate physical layer standards for 100 Mbps over UTP cable: 100Base-TX and 100Base-T4 in 1995, and 100Base-T2 in 1997. Each was defined with different encoding requirements and a different set of media-dependent sublayers, even though there is some overlap in the link cabling.

Table: Summary of 100Base-T Physical Layer Characteristics compares the physical layer characteristics of 10Base-T to the various 100Base versions.

COMPUTER COMMUNICATION NETWORKS

Table: Summary of 100Base-T Physical Layer Characteristics

Ethernet Version	Transmit Symbol Rate	Encoding	Cabling	Full-Duplex Operation
10Base-T	10 MBd	Manchester	Two pairs of UTP Category - 3 or better	Supported
100Base-TX	125 MBd	4B/5B	Two pairs of UTP Category - 5 or Type 1 STP	Supported
100Base-T4	33 MBd	8B/6T	Four pairs of UTP Category - 3 or better	Not supported
100Base-T2	25 MBd	PAM5x5	Two pairs of UTP Category - 3 or better	Supported

🖉 **Note**: One baud is equal to one transmitted symbol per second, where the transmitted symbol may contain the equivalent value of 1 or more binary bits.

Although not all three 100-Mbps versions were successful in the marketplace, all three have been discussed in the literature, and all three did impact future designs. As such, all three are important to consider here.

100Base-X

100Base-X was designed to support transmission over either two pairs of Category 5 UTP copper wire or two strands of optical fiber. Although the encoding, decoding, and clock recovery procedures are the same for both media, the signal transmission is different-electrical pulses in copper and light pulses in optical fiber. The signal transceivers that were included as part of the PMA function in the generic logical model of the following figure were redefined as the separate physical media-dependent (PMD) sublayers shown in Figure: The 100Base-X Logical Model.

COMPUTER COMMUNICATION NETWORKS

Figure: The 100Base-X Logical Model
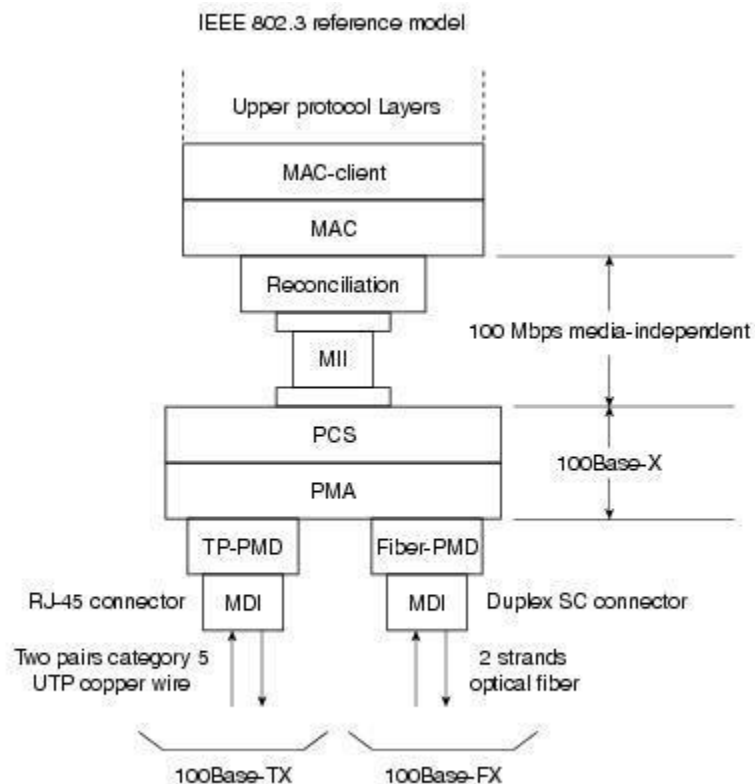
The 100Base-X encoding procedure is based on the earlier FDDI optical fiber physical media-dependent and FDDI/CDDI copper twisted-pair physical media-dependent signaling standards developed by ISO and ANSI. The 100Base-TX physical media-dependent sublayer (TP-PMD) was implemented with CDDI semiconductor transceivers and RJ-45 connectors; the fiber PMD was implemented with FDDI optical transceivers and the Low Cost Fibre Interface Connector (commonly called the duplex SC connector).

The 4B/5B encoding procedure is the same as the encoding procedure used by FDDI, with only minor adaptations to accommodate Ethernet frame control. Each 4-bit data nibble (representing half of a data byte) is mapped into a 5-bit binary code-group that is transmitted bit-serial over the link. The expanded code space provided by the 32 5-bit code-groups allow separate assignment for the following:

- The 16 possible values in a 4-bit data nibble (16 code-groups).
- Four control code-groups that are transmitted as code-group pairs to indicate the start-of-stream delimiter (SSD) and the end-of-stream delimiter (ESD). Each MAC frame is "encapsulated" to mark both the beginning and end of the frame. The first byte of preamble is

replaced with SSD code-group pair that precisely identifies the frame's code-group boundaries. The ESD code-group pair is appended after the frame's FCS field.

- A special IDLE code-group that is continuously sent during interframe gaps to maintain continuous synchronization between the NICs at each end of the link. The receipt of IDLE is interpreted to mean that the link is quiet.

- Eleven invalid code-groups that are not intentionally transmitted by a NIC (although one is used by a repeater to propagate receive errors). Receipt of any invalid code-group will cause the incoming frame to be treated as an invalid frame.

Figure: The 100Base-X Code-Group Stream with Frame Encapsulation shows how a MAC frame is encapsulated before being transmitted as a 100Base-X code-group stream.

Figure: The 100Base-X Code-Group Stream with Frame Encapsulation
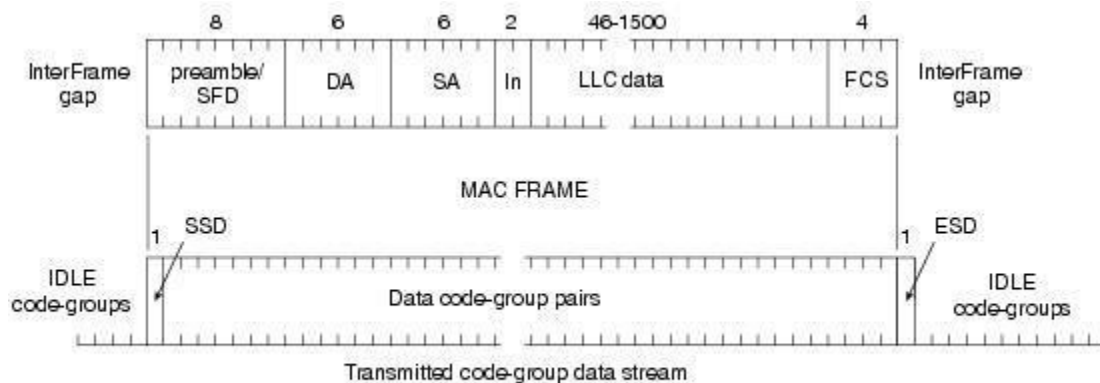
100Base-TX transmits and receives on the same link pairs and uses the same pin assignments on the MDI as 10Base-T. 100Base-TX and 100Base-FX both support half-duplex and full-duplex transmission.

100Base-T4

100Base-T4 was developed to allow 10BaseT networks to be upgraded to 100-Mbps operation without requiring existing four-pair Category 3 UTP cables to be replaced with the newer Category 5 cables. Two of the four pairs are configured for half-duplex operation and can support transmission in either direction, but only in one direction at a time. The other two pairs are configured as simplex pairs dedicated to transmission in one direction only. Frame transmission uses both half-duplex pairs, plus the simplex pair that is appropriate for the transmission direction, as shown in Figure: The 100Base-T4 Wire-Pair Usage During Frame Transmission. The simplex pair for the opposite direction provides carrier sense and collision detection. Full-duplex operation cannot be supported on 100Base-T4.

COMPUTER COMMUNICATION NETWORKS

Figure: The 100Base-T4 Wire-Pair Usage During Frame Transmission

Transmission paths for DTE1 shown in bold
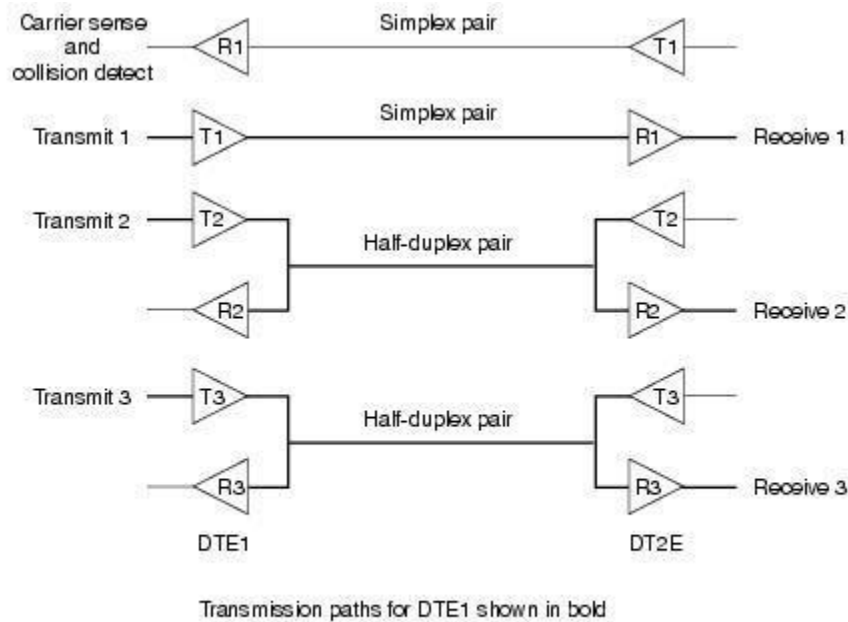
100Base-T4 uses an 8B6T encoding scheme in which each 8-bit binary byte is mapped into a pattern of six ternary (three-level: +1, 0, -1) symbols known as 6T code-groups. Separate 6T code-groups are used for IDLE and for the control code-groups that are necessary for frame transmission. IDLE received on the dedicated receive pair indicates that the link is quiet.

During frame transmission, 6T data code-groups are transmitted in a delayed round-robin sequence over the three transmit wire-pairs, as shown in Figure: The 100Base-T4 Frame Transmission Sequence. Each frame is encapsulated with start-of-stream and end-of-packet 6T code-groups that mark both the beginning and end of the frame, and the beginning and end of the 6T code-group stream on each wire pair. Receipt of a non-IDLE code-group over the dedicated receive-pair any time before the collision window expires indicates that a collision has occurred.

COMPUTER COMMUNICATION NETWORKS

Figure: The 100Base-T4 Frame Transmission Sequence
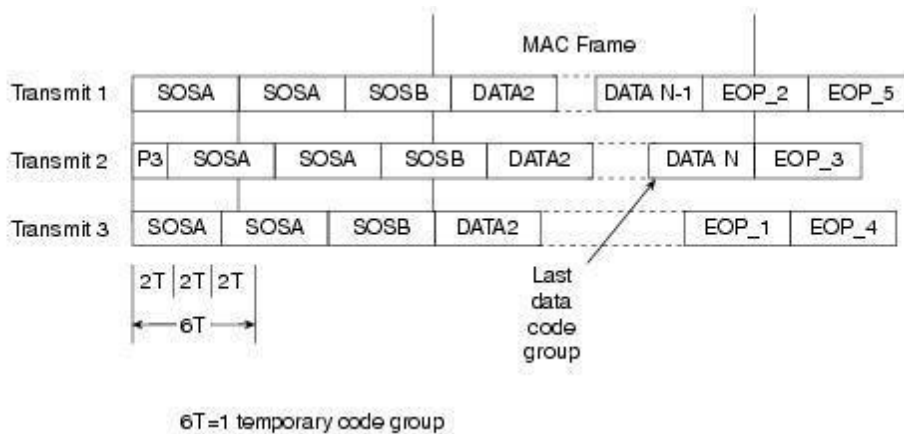
100Base-T2

The 100Base-T2 specification was developed as a better alternative for upgrading networks with installed Category 3 cabling than was being provided by 100Base-T4. Two important new goals were defined:

- To provide communication over two pairs of Category 3 or better cable
- To support both half-duplex and full-duplex operation

100Base-T2 uses a different signal transmission procedure than any previous twisted-pair Ethernet implementations. Instead of using two simplex links to form one full-duplex link, the 100Base-T2 dual-duplex baseband transmission method sends encoded symbols simultaneously in both directions on both wire pairs, as shown in Figure: The 100Base-T2 Link Topology. The term "TDX<3:2>" indicates the 2 most significant bits in the nibble before encoding and transmission. "RDX<3:2>" indicates the same 2 bits after receipt and decoding.

COMPUTER COMMUNICATION NETWORKS

Figure: The 100Base-T2 Link Topology
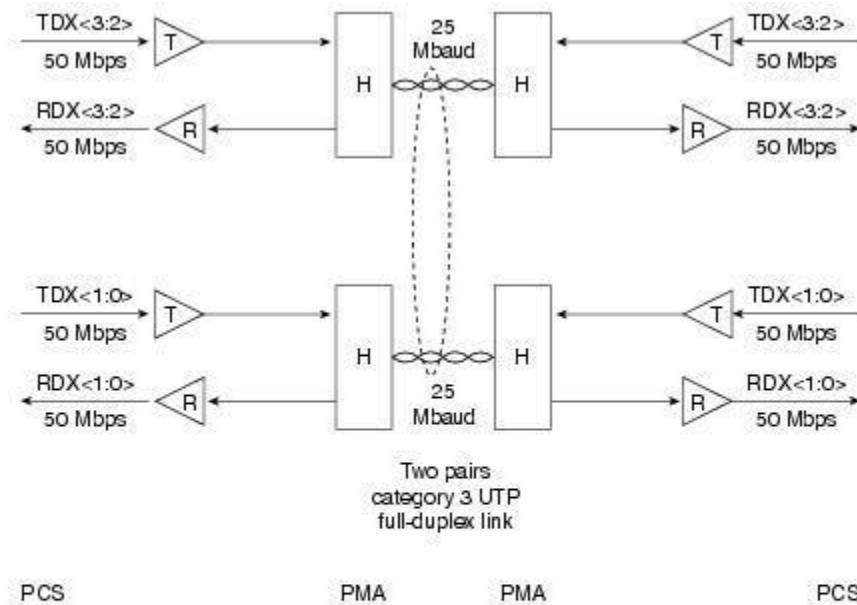
H = Hybrid canceller transceiver
T = Transmit encoder
R = Receive decoder
Two PAM5 code symbols = One nibble

Dual-duplex baseband transmission requires the NICs at each end of the link to be operated in a master/slave loop-timing mode. Which NIC will be master and which will be slave is determined by autonegotiation during link initiation. When the link is operational, synchronization is based on the master NIC's internal transmit clock. The slave NIC uses the recovered clock for both transmit and receive operations, as shown inFigure: The 100Base-T2 Loop Timing Configuration. Each transmitted frame is encapsulated, and link synchronization is maintained with a continuous stream of IDLE symbols during interframe gaps.

COMPUTER COMMUNICATION NETWORKS

Figure: The 100Base-T2 Loop Timing Configuration
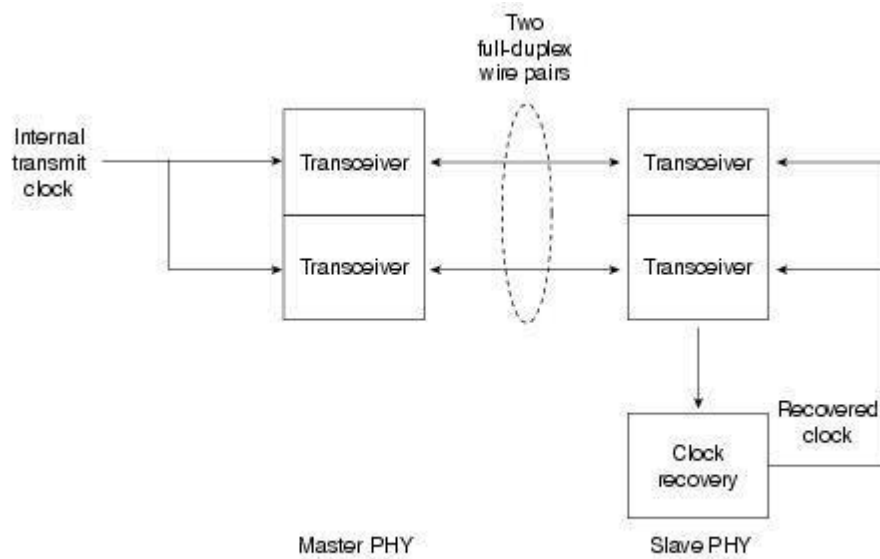
The 100Base-T2 encoding process first scrambles the data frame nibbles to randomize the bit sequence. It then maps the two upper bits and the two lower bits of each nibble into two five-level (+2, +1, 0, -1, -2) pulse amplitude-modulated (PAM5) symbols that are simultaneously transmitted over the two wire pairs (PAM5x5). Different scrambling procedures for master and slave transmissions ensure that the data streams traveling in opposite directions on the same wire pair are uncoordinated.

Signal reception is essentially the reverse of signal transmission. Because the signal on each wire pair at the MDI is the sum of the transmitted signal and the received signal, each receiver subtracts the transmitted symbols from the signal received at the MDI to recover the symbols in the incoming data stream. The incoming symbol pair is then decoded, unscrambled, and reconstituted as a data nibble for transfer to the MAC.
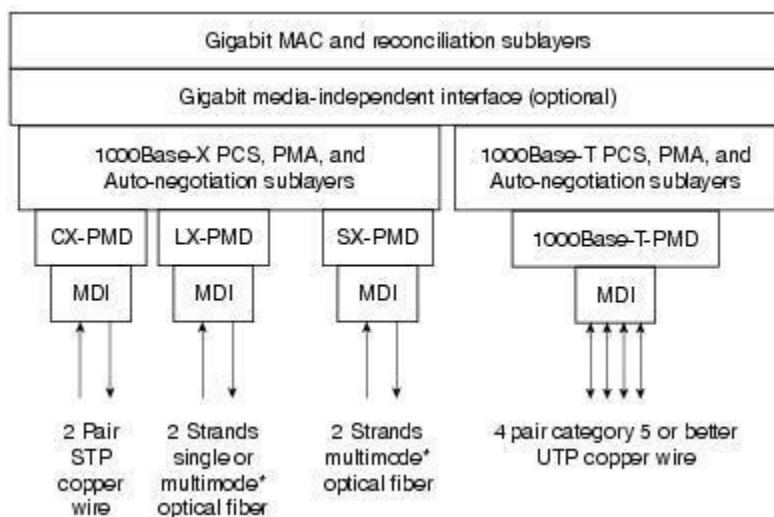
1000 Mbps-Gigabit Ethernet

The Gigabit Ethernet standards development resulted in two primary specifications: 1000Base-T for UTP copper cable and 1000Base-X STP copper cable, as well as single and multimode optical fiber (see Figure: Gigabit Ethernet Variations).

Figure 7-22 Gigabit Ethernet Variations

1000Base-T

1000Base-T Ethernet provides full-duplex transmission over four-pair Category 5 or better UTP cable. 1000Base-T is based largely on the findings and design approaches that led to the development of the Fast Ethernet physical layer implementations:

- 100Base-TX proved that binary symbol streams could be successfully transmitted over Category 5 UTP cable at 125 MBd.
- 100Base-T4 provided a basic understanding of the problems related to sending multilevel signals over four wire pairs.
- 100Base-T2 proved that PAM5 encoding, coupled with digital signal processing, could handle both simultaneous two-way data streams and potential crosstalk problems resulting from alien signals on adjacent wire pairs.

1000Base-T scrambles each byte in the MAC frame to randomize the bit sequence before it is encoded using a 4-D, 8-State Trellis Forward Error Correction (FEC) coding in which four PAM5 symbols are sent at the same time over four wire pairs. Four of the five levels in each PAM5 symbol represent 2 bits in the data byte. The fifth level is used for FEC coding, which enhances symbol recovery in the presence of noise and crosstalk. Separate scramblers for the master and slave PHYs create essentially uncorrelated data streams between the two opposite-travelling symbol streams on each wire pair.

The1000Base-T link topology is shown in Figure: The 1000Base-T Link Topology. The term "TDX<7:6>" indicates the 2 most significant bits in the data byte before encoding and transmission. "RDX<7:6>" indicates the same 2 bits after receipt and decoding.

COMPUTER COMMUNICATION NETWORKS

Figure: The 1000Base-T Link Topology
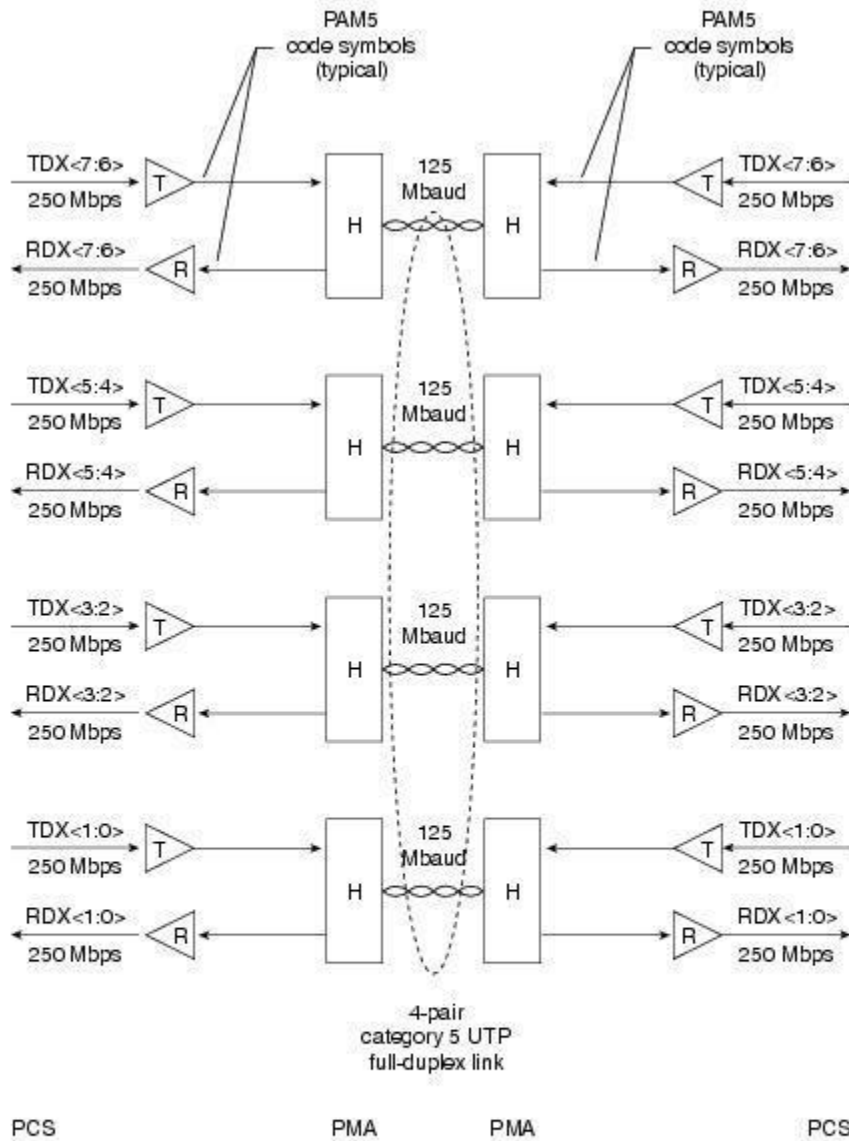
The clock recovery and master/slave loop timing procedures are essentially the same as those used in 100Base-T2 (see Figure: 1000Base-T Master/Slave Loop Timing Configuration). Which NIC will be master (typically the NIC in a multiport intermediate network node) and which will be slave is determined during auto negotiation.

COMPUTER COMMUNICATION NETWORKS

Figure: 1000Base-T Master/Slave Loop Timing Configuration
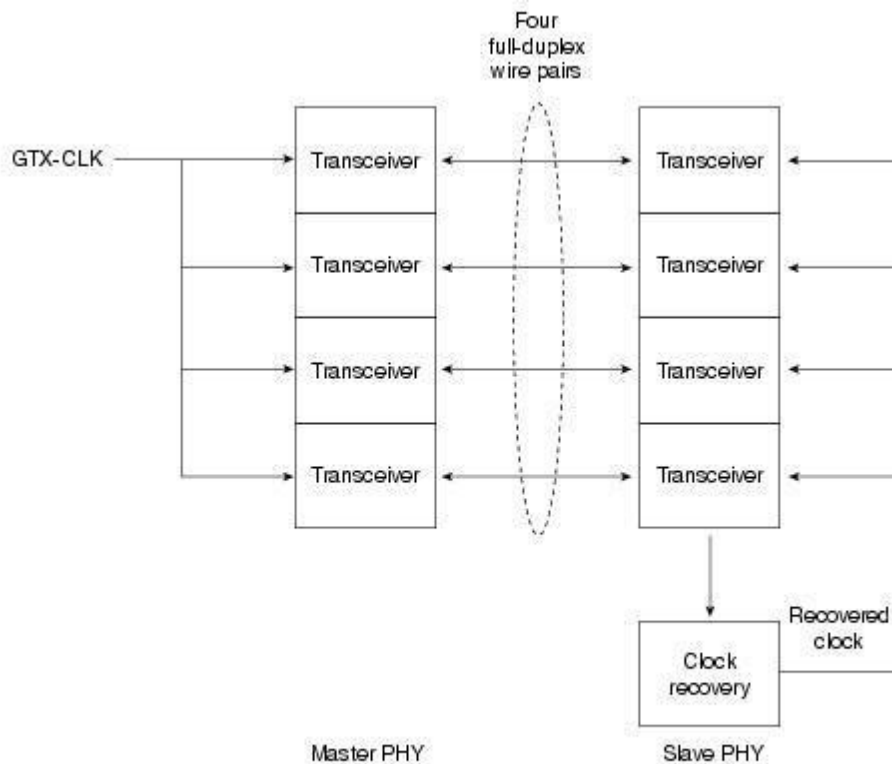
Each transmitted frame is encapsulated with start-of-stream and end-of-stream delimiters, and loop timing is maintained by continuous streams of IDLE symbols sent on each wire pair during interframe gaps. 1000Base-T supports both half-duplex and full-duplex operation.

1000Base-X

All three 1000Base-X versions support full-duplex binary transmission at 1250 Mbps over two strands of optical fiber or two STP copper wire-pairs, as shown in Figure: 1000Base-X Link Configuration. Transmission coding is based on the ANSI Fibre Channel 8B/10B encoding scheme. Each 8-bit data byte is mapped into a 10-bit code-group for bit-serial transmission. Like earlier Ethernet versions, each data frame is encapsulated at the physical layer before transmission, and link synchronization is maintained by sending a continuous stream of IDLE code-groups during interframe gaps. All 1000Base-X physical layers support both half-duplex and full-duplex operation.

COMPUTER COMMUNICATION NETWORKS

Figure: 1000Base-X Link Configuration
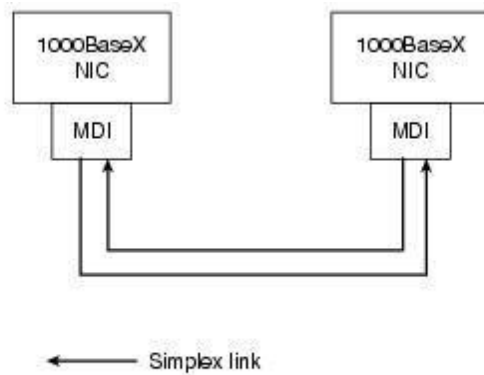
The principal differences among the 1000Base-X versions are the link media and connectors that the particular versions will support and, in the case of optical media, the wavelength of the optical signal (see Table: 1000Base-X Link Configuration Support).

Table: 1000Base-X Link Configuration Support

Link Configuration	1000Base-CX	1000Base-SX (850 nm Wavelength)	1000Base-LX (1300 nm Wavelength)
150 Ω STP copper	Supported	Not supported	Not supported
125/62.5 µm multimode optical fiber	Not supported	Supported	Supported
125/50 µm multimode optical fiber	Not supported	Supported	Supported
125/10 µm single mode optical fiber	Not supported	Not supported	Supported
Allowed connectors	IEC style 1 or Fibre Channel style 2	SFF MT-RJ or Duplex SC	SFF MT-RJ or Duplex SC

The 125/62.5 µm specification refers to the cladding and core diameters of the optical fiber.

Network Cabling-Link Crossover Requirements

Link compatibility requires that the transmitters at each end of the link be connected to the receivers at the other end of the link. However, because cable connectors at both ends of the link

are keyed the same, the conductors must cross over at some point to ensure that transmitter outputs are always connected to receiver inputs.

Unfortunately, when this requirement first came up in the development of 10Base-T, IEEE 802.3 chose not to make a hard rule as to whether the crossover should be implemented in the cable as shown in Figure: Alternative Ways for Implementing the Link Crossover Requirement (a) or whether it should be implemented internally as shown in Figure: Alternative Ways for Implementing the Link Crossover Requirement (b).

Figure: Alternative Ways for Implementing the Link Crossover Requirement

(a) Cable-based crossover

(b) Internal crossover
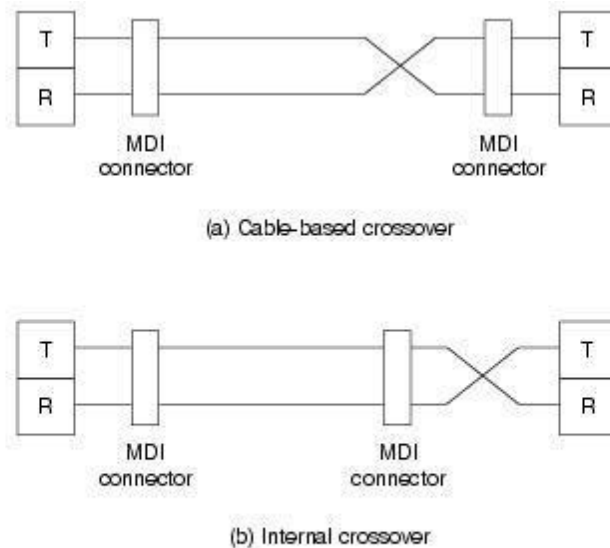
Instead, IEEE 802.3 defined two rules and made two recommendations:

- There must be an odd number of crossovers in all multi-conductor links.
- If a PMD is equipped with an internal crossover, its MDI must be clearly labeled with the graphical X symbol.
- Implementation of an internal crossover function is optional.
- When a DTE is connected to a repeater or switch (DCE) port, it is recommended that the crossover be implemented within the DCE port.

The eventual result was that ports in most DCEs were equipped with PMDs that contained internal crossover circuitry and that DTEs had PMDs without internal crossovers. This led to the following oft-quoted de facto installation rule:

- Use a straight-through cable when connecting DTE to DCE. Use a crossover cable when connecting DTE to DTE or DCE to DCE.

COMPUTER COMMUNICATION NETWORKS

Unfortunately, the de facto rule does not apply to all Ethernet versions that have been developed subsequent to 10Base-T. As things now stand, the following is true:

- All fiber-based systems use cables that have the crossover implemented within the cable.
- All 100Base systems using twisted-pair links use the same rules and recommendations as 10Base-T.
- 1000Base-T NICs may implement a selectable internal crossover option that can be negotiated and enabled during auto-negotiation. When the selectable crossover option is not implemented, 10Base-T rules and recommendations apply.

COMPUTER COMMUNICATION NETWORKS

UNIT 6: NETWORK LAYER

Protocol Design Concepts, IP and Routing

Introduction

A central idea in the design of protocols is that of layering; and a guiding principle of Internet protocols is the "end-to-end" principle. In this chapter, we review these ideas and describe the transport and network layers in the Internet stack.

Protocols and Layering

Protocols are complex, distributed pieces of software. Abstraction and modular design are standard techniques used by software engineers to deal with complexity. By abstraction, we mean that a subset of functions is carefully chosen and setup as a "black-box" or module (see Figure 1). The module has an interface describing its input/output behavior. The interface outlives the implementation the module in the sense that the technology used to implement the interface may change often, but the interface tends to remain constant. Modules may be built and maintained by different entities. The software modules are then used as building blocks in a larger design. Placement of functions to design the right building blocks and interfaces is a core activity in software engineering.

Input Interface → Software → Output Interface
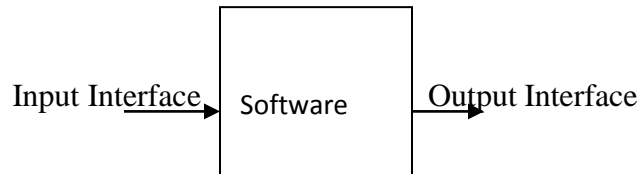
Figure 1: Abstraction of Functionality into Modules

Protocols have an additional constraint of being distributed. Therefore software modules have to communicate with one or more software modules at a distance. Such interfaces across a distance are termed as "***peer-to-peer***" interfaces; and the local interfaces are termed as "***service***" interfaces (Figure 2). Since protocol function naturally tend to be a sequence of functions, the modules on each end are organized as a (vertical) sequence called "***layers***". The set of modules

organized as layers is also commonly called a "***protocol stack***". The concept of layering is illustrated in Figure 3.

Service Interface   *Service Interface*

Input   *Peer-to-Peer Interface*   Output

Software   Software

Output   Input
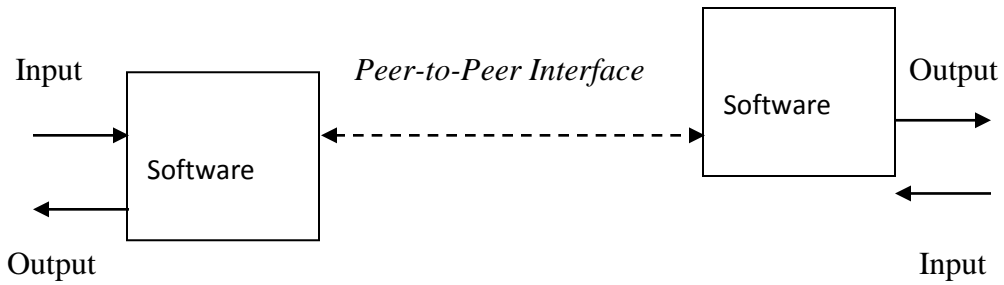
Figure 2: Communicating Software Modules

Service Interfaces

Layer *i+1*   *Peer-to-Peer Interface*   Layer *i+1*

Service Interfaces

Layer *i*   *Peer-to-Peer Interface*   Layer *i*
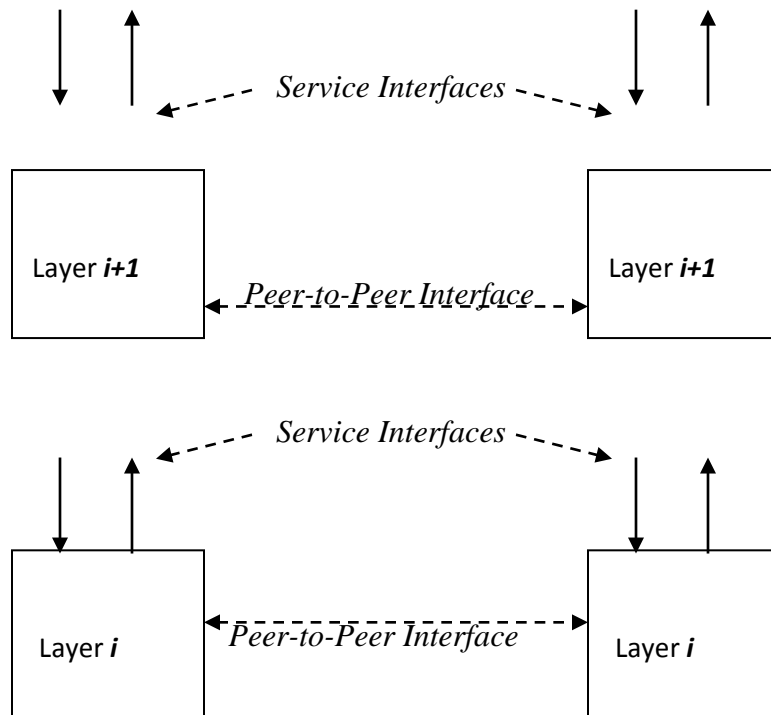
Figure 3: Layered Communicating Software Modules (Protocols)

COMPUTER COMMUNICATION NETWORKS

Over the years, some layered models have been standardized. The ISO Open Systems Interconnection (ISO/OSI) layered model has seven layers and was developed by a set of committees under the auspices of International Standards Organization (ISO). The TCP/IP has a 4-layered stack that has become a *de facto* standard of the Internet. The TCP/IP and ISO/OSI stacks and their rough correspondences are illustrated in Figure 4. The particular protocols of the TCP/IP stack are also shown.

Application	FTP	Telnet	HTTP		Application
					Presentation
Transport	TCP		UDP		Session
					Transport
Internetwork	IP				Network
Host to Network	Ether net	Packet Radio	Point-to-Point		Datalink
					Physical

Figure 4: TCP/IP vs ISO/OSI Protocol Stack

The physical layer deals with getting viable bit-transmission out of the underlying medium (fiber, copper, coax cable, air etc). The data-link layer then converts this bit-transmission into a framed-transmission on the link. "Frames" are used to allow multiplexing of several streams; and defines the unit of transmission used in error detection and flow control. In the event that the link is actually shared (i.e. multiple access), the data link layer defines the medium access control (MAC) protocol as well. The network layer deals with packet transmission across multiple links from the source node to the destination node. The functions here include routing, signaling and mechanisms to deal with the heterogeneous link layers at each link.

The transport layer protocol provides "end-to-end" communication services, i.e., it allows applications to multiplex the network service, and may add other capabilities like connection setup, reliability and flow/congestion control. Examples of communication abstractions provided by the transport layer include a reliable byte-stream service (TCP) and an unreliable datagram service (UDP). These abstractions are made available through application-level programming interfaces (APIs) such as the BSD socket interface. The application layers (session, presentation, application) then use the communication abstractions provided by the transport layer to create the basis for interesting applications like email, web, file transfer, multimedia conference, peer-to-peer applications etc. Examples of such protocols include SMTP, HTTP, DNS, H.323 and SIP.

COMPUTER COMMUNICATION NETWORKS

The End-to-End Principle in Internet Protocol Design

A key principle used in the design of the TCP/IP protocols is the so-called "end-to-end" principle that guides the placement of functionality in a complex distributed system. The principle suggests that "…*functions placed at the lower levels may be redundant or of little value when compared to the cost of providing them at the lower level…*" In other words, a system (or subsystem level) should consider only functions that can be *completely and correctly* implemented within it. All other functions are best moved to the system level where it can be completely and correctly implemented.

In the context of the Internet, it implies that several functions like reliability, congestion control, session/connection management are best moved to the end-systems (i.e. performed on an "end-to-end" basis), and the network layer focuses on functions which it can fully implement, i.e. routing and datagram delivery. As a result, the end-systems are intelligent and in control of the communication while the forwarding aspects of the network is kept simple. This leads to a philosophy diametrically opposite to the telephone world which sports dumb end-systems (the telephone) and intelligent networks. Indeed the misunderstanding of the end-to-end principle has been a primary cause for friction between the "telephony" and "internet" camps. Arguably the telephone world developed as such due to technological and economic reasons because intelligent and affordable end-systems were not possible until 1970s. Also, as an aside, note that there is a misconception that the end-to-end principle implies a "dumb" network. Routing is a good example of a very complex function that is consistent with the end-to-end principle, but is non-trivial in terms of complexity. Routing is kept at the network level because it can be completely implemented at that level, and the costs of involving the end-systems in routing are formidable.

The end-to-end principle further argues that even if the network layer did provide connection management and reliability, transport levels would have to add reliability to account for the interaction at the transport-network boundary; or if the transport needs more reliability than what the network provides. Removing these concerns from the lower layer packet-forwarding devices streamlines the forwarding process, contributing to system-wide efficiency and lower costs. In other words, the costs of providing the "incomplete" function at the network layers would arguably outweigh the benefits.

It should be noted that the end-to-end principle emphasizes function placement vis-a-vis correctness, completeness and overall system costs. The argument does say that, *"…sometimes an incomplete version of the function provided by the communication system may be useful as   a*

performance enhancement…" In other words, the principle does allow a cost-performance tradeoff, and incorporation of economic concerns. However, it cautions that the choice of such "incomplete versions of functions" to be placed inside the network should be made very prudently. Lets try to understand some implications of this aspect.

One issue regarding the "incomplete network-level function" is the degree of "state" maintained inside the network. Lack of state removes any requirement for the network nodes to notify each other as endpoint connections are formed or dropped. Furthermore, the endpoints are not, and need not be, aware of any network components other than the destination, first hop router(s), and an optional name resolution service. Packet integrity is preserved through the network, and transport checksums and any address-dependent security functions are valid end-to-end. If state is maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks (also termed "*fate-sharing*"), then as networks grow in size, likelihood of component failures affecting a connection becomes increasingly frequent. If failures lead to loss of communication, because key state is lost, then the network becomes increasingly brittle, and its utility degrades. However, if an endpoint itself fails, then there is no hope of subsequent communication anyway. Therefore one quick interpretation of the end-to-end model is that it suggests that only the endpoints should hold critical state. But this is flawed.

Let us consider the economic issues of Internet Service Provider (ISPs) into this mix. ISPs need to go beyond the commoditised mix of access and connectivity services to provide differentiated network services. Providing Quality of Service (QoS) and charging for it implies that some part of the network has to participate in decisions of resource sharing, and billing, which cannot be entrusted to end-systems. A correct application of the end-to-end principle in this scenario is as follows: due to the economic and trust model issues, these functions belong to the network. Applications may be allowed to participate in the decision process, but the control belongs to the network, not the end-system in this matter. The differentiated services architecture discussed later in this chapter has the notion of the "network edge" which is the repository of these functions.

In summary, the end-to-end principle has guided a vast majority of function placement decisions in the Internet and it remains relevant today even as the design decisions are intertwined with complex economic concerns of multiple ISPs and vendors.

COMPUTER COMMUNICATION NETWORKS

Network Layer

The network layer in the TCP/IP stack deals with internetworking and routing. The core problems of internetworking are *heterogeneity* and *scale*. Heterogeneity is the problem of dealing with disparate layer 2 networks to create a viable forwarding and addressing paradigm; and the problem of providing meaningful service to a range of disparate applications. Scale is the problem of allowing the Internet to grow without bounds to meet its intended user demands. The Internet design applies the end-to-end principle to deal with these problems.

Network Service Models

One way of dealing with heterogeneity is to provide translation services between the heterogeneous entities when forwarding across them is desired. Examples of such design include multi-protocol bridges and multi-protocol routers. But this gets too complicated and does not allow scaling because every new entity that wishes to join the Internet will require changes in all existing infrastructure. A more preferable requirement is to be able to "*incrementally upgrade*" the network. The alternative strategy is called an "overlay" model where a new protocol (IP) with its own packet format and address space is developed and the mapping is done between all protocols and this intermediate protocol.

IP has to be simple by necessity so that the mapping between IP and lower layer protocols is simplified. As a result, IP opts for a best-effort, unreliable datagram service model where it forwards datagrams between sources and destinations situated on, and separated by a set of disparate networks. IP expects a minimal link-level frame forwarding service from lower layers. The mapping between IP and lower layers involve address mapping issues (eg: address resolution) and packet format mapping issues (eg: fragmentation/reassembly). Experience has shown that this mapping is straightforward in many subnetworks, especially those that are not too large, and those which support broadcast at the LAN level. The address resolution can be a complex problem on non-broadcast multiple access (NBMA) sub-networks; and the control protocols associated with IP (esp BGP routing) can place other requirements on large sub-networks (eg: ATM networks) which make the mapping problems hard. Hybrid technologies like MPLS are used to address these mapping concerns, and to enable new traffic engineering capabilities in core networks.

For several applications, it turns out that the simple best-effort service provided by IP can be augmented with end-to-end transport protocols like TCP, UDP and RTP to be sufficient. Other applications having stringent performance expectations (eg: telephony) need to either adapt and/or use augmented QoS capabilities from the network. While several mechanisms and protocols for this have been developed in the last decade, a fully QoS-capable Internet is still a holy grail for the Internet community. The hard problems surround routing,   inter-domain/multi-

provider issues, and the implications of QoS on a range of functions (routing, forwarding, scheduling, signaling, application adaptation etc).

In summary, the best-effort, overlay model of IP has proved to be enormously successful, it has faced problems in being mapped to large NBMA sub-networks and continues to faces challenges in the inter-domain/multi-provider and QoS areas.

The Internet Protocol (IP): Forwarding Paradigm

The core service provided by IP is datagram forwarding over disparate networks. This itself is a non-trivial problem. The end-result of this forwarding service is to provide connectivity. The two broad approaches to getting connectivity are: direct connectivity and indirect connectivity. *Direct connectivity* refers to the case where the destination is only a single link away (this includes shared and unshared media). *Indirect connectivity* refers to connectivity achieved by going through intermediate components or intermediate networks. The intermediate components (bridges, switches, routers, NAT boxes etc) are dedicated to functions to deal with the problem of scale and/or heterogeneity. Indeed the function of providing indirect connectivity through intermediate networks can be thought of as a design of a large virtual intermediate component, the Internet. These different forms of connectivity are shown in Figures 5-7.

Bus

Full mesh
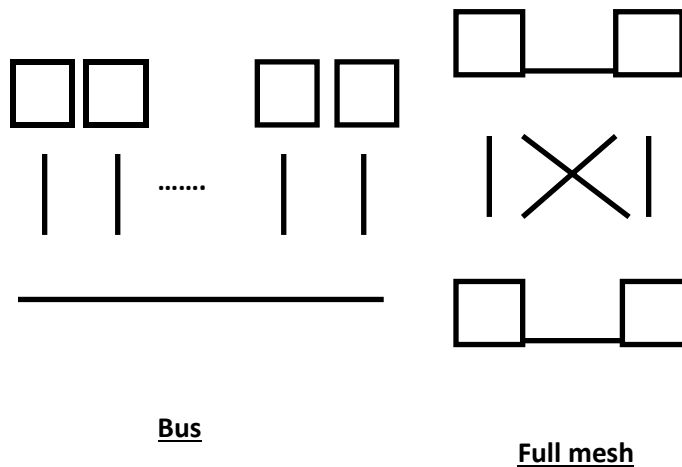
Figure 5: Direct Connectivity Architectures
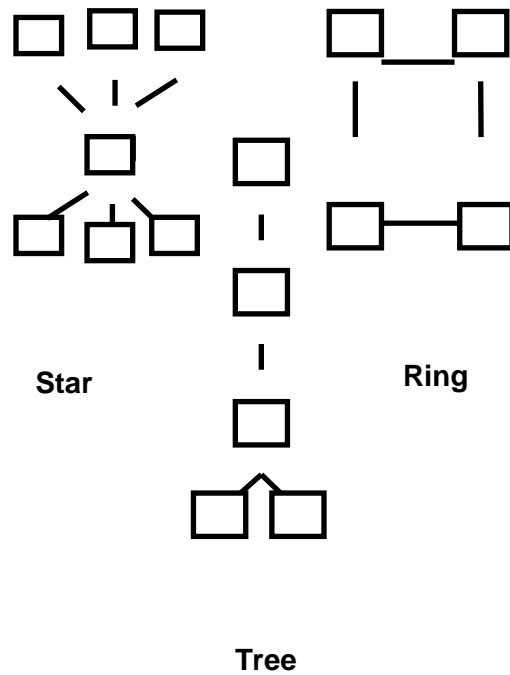
Figure 6: Indirect Connectivity though Intermediate Components
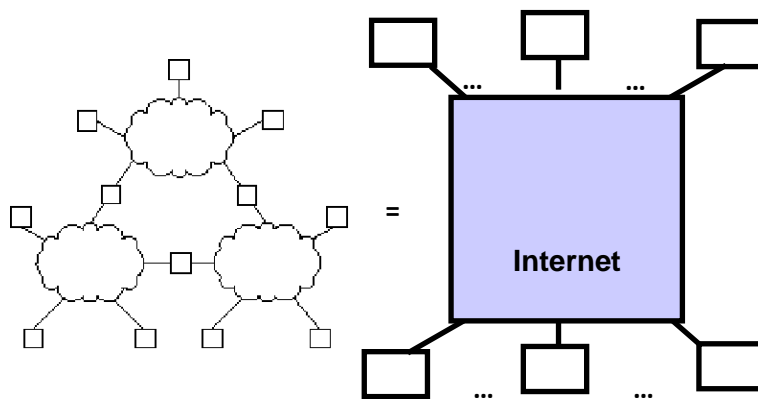
Figure 7: Indirect Connectivity though Intermediate Neworks & Components

The problem of scaling with respect to a parameter (eg: number of nodes) is inversely related to the efficiency characteristics of the architecture with respect to the same parameter. For example, direct connectivity architectures do not scale because of finite capacity of shared medium, or

finite interface slots; or high costs of provisioning a full mesh of links. A way to deal with this is to build a switched network, where the intermediate components ("*switches*") provide filtering and forwarding capabilities to isolate multiple networks to keep them within their scaling limits, and yet providing scalable interconnection. In general, the more efficient the filtering and forwarding of these components, the more scalable is the architecture. Layer 1 hubs do pure broadcast, and hence do no filtering, but can forward signals. Layer 2 bridges and switches can filter to an extent using forwarding tables learnt by snooping; but their default to flooding on a spanning tree when the forwarding table does not contain the address of the receiver. This default behavior of flooding or broadcast is inefficient, and hence limits scalability. This behavior if also partially a result of the flat addressing structure used by L2 networks.

In contrast, layer 3 (IP) switches (aka routers) *never* broadcast across sub-networks; and rely on a set of routing protocols and a concatenated set of local forwarding decisions to deliver packets across the Internet. IP addressing is designed hierarchically, and address assignment is coordinated with routing design. This enables intermediate node (or hosts) to do a simple determination: whether the destination is *directly* or *indirectly* connected. In the former case, simple layer 2 forwarding is invoked; and in the latter case, a layer 3 forwarding decision is made to determine the next-hop that is an intermediate node on the same sub-network, and then the layer 2 forwarding is invoked.

Heterogeneity is supported by IP because it invokes only a minimal forwarding service of the underlying L2 protocol. Before invoking this L2 forwarding service, the router has to a) determine the L2 address of the destination (or next-hop) -- an address resolution problem; and b) map the datagram to the underlying L2 frame format. If the datagram is too large, it has to do something -- fragmentation/reassembly. IP does not expect any other special feature in lower layers and hence can work over a range of L2 protocols.

In summary, the IP forwarding paradigm naturally comes out of the notions of direct and indirect connectivity. The "secret sauce" is in the way addressing is designed to enable the directly/indirectly reachable query; and the scalable design of routing protocols to aid the determination of the appropriate next-hop if the destination is indirectly connected. Heterogeneity leads to mapping issues, which are simplified because of the minimalist expectations of IP from its lower layers (only an forwarding capability expected). All other details of lower layers are abstracted out.

The Internet Protocol: Packet Format, Addressing, Fragmentation/Reassembly

IP Packet Format

The IP packet format is shown in Figure 8. The biggest fields in the header are the source and destination 32-bit IP *address* fields. The second 32-bit line (*ID, flags, frag offset*) are related to fragmentation/reassembly and will be explained later. The length field indicates the length of the entire datagram, and is required because IP accepts variable length payloads. The *checksum* field

covers only the header and not the payload and is used to catch any header errors to avoid mis-routing garbled packets. Error detection in the payload is the responsibility of the transport layer (both UDP and TCP provide error detection). The *protocol* field allows IP to demultiplex the datagram and deliver it to a higher-level protocol. Since it has only 8-bits, IP does not support application multiplexing. Providing port number fields to enable application multiplexing is another required function in transport protocols on IP.

The *time-to-live (TTL)* field is decremented at every hop and the packet is discarded if the field is 0; this prevents packets from looping forever in the Internet. The TTL field is also used a simple way to scope the reach of the packets, and can be used in conjunction with ICMP, multicast etc to support administrative functions. The *type-of-service (TOS)* field was designed to allow optional support for differential forwarding, but has not been extensively used. Recently, the differentiated services (diff-serv) WG in IETF renamed this field to the *DS byte* to be used to support diff-serv. The version field indicates the version of IP and allows extensibility. The current version of IP is version 4. IPv6 is the next generation of IP that may be deployed over the next decade to support a larger 128-bit IP address space. *Header length* is a field used because *options* can be variable length. But options are rarely used in modern IP deployments, so we don't discuss them any further.
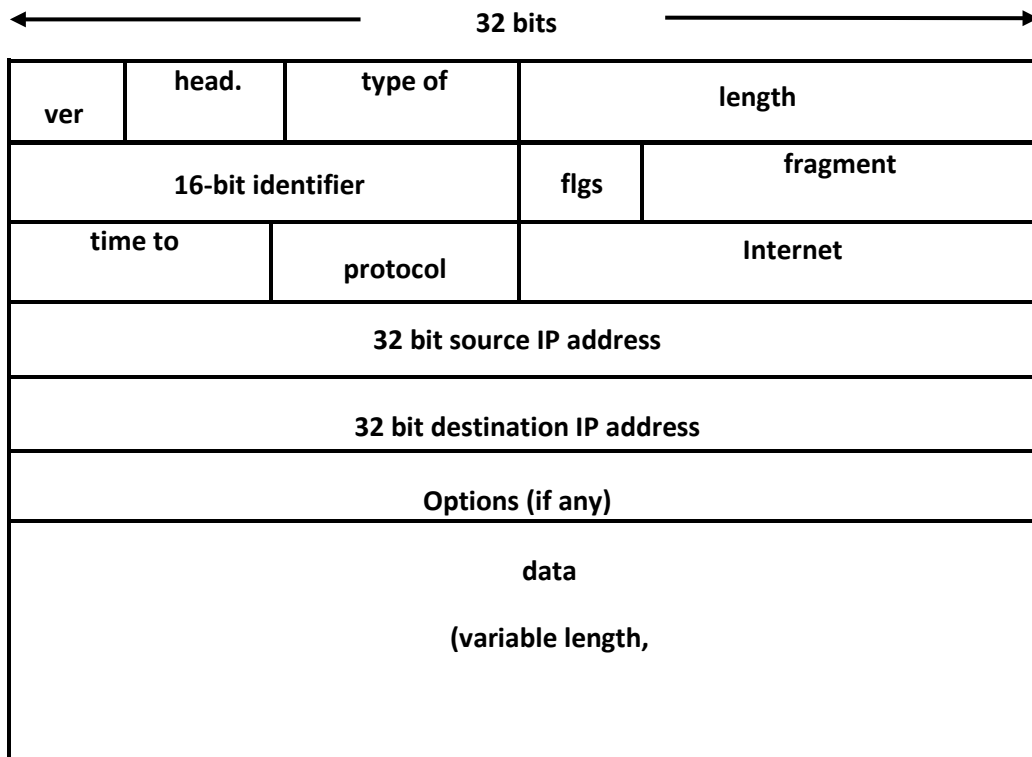

Figure 8: IP Packet Format

3.3.2 IP Addressing and Address Allocation

An address is a *unique "computer-understandable" identifier*. Uniqueness is defined in a domain. Outside that domain one needs to have either a larger address space, or do translation. An address should ideally be valid regardless of the location of the <u>source</u>, but may change if the <u>destination</u> moves. Fixed size addresses can be processed faster.

The concept of addresses is fundamental to networking. There is no (non-trivial) network without addresses. Address space size also limits the scalability of networks. A large address space allows a large network, i.e. it is fundamentally required for network scalability. Large address space also makes it easier to assign addresses and minimize configuration. In connectionless networks, the most interesting differences revolve around addresses. After all, a connectionless net basically involves putting an address in a packet and sending it hoping it will get to the destination.

IPv4 uses 32-bit addresses whereas IPv6 uses 128-bit addresses. For convenience of writing, a dotted decimal notation became popular. Each byte is summarized as a base-10 integer, and dots placed between these numbers (eg: 128.113.40.50).

IP addresses have two parts -- a network part (prefix), and a host part (suffix). This is illustrated in Figure 9. Recall that the intermediate nodes (or hosts) have to make a determination whether the destination is directly or indirectly connected. Examining the network part of the IP address allows us to make this determination. If the destination is directly connected, the network part matches the network part of an outgoing interface of the intermediate node. This hierarchical structure of addressing which is fundamental to IP scaling is not seen in layer 2 (IEEE 802) addresses. The structure has implications on address allocation because all interfaces on a single sub-network have to be assigned the same network part of the address (to enable the forwarding test mentioned above).

Network ID	Host ID

Demarcator ⟶
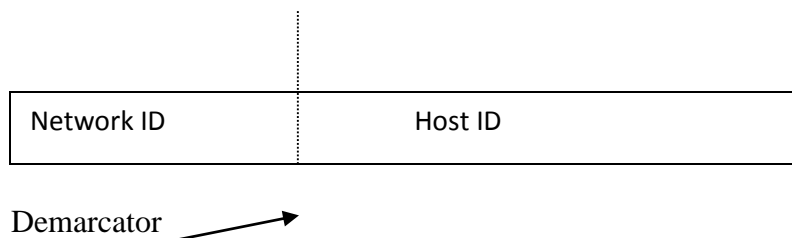
Figure 9: Hierarchical Structure of an IP Address

Unfortunately address allocation was not well thought out during the early days of IP, and hence it has followed a number of steps of evolution. Part of the evolution was forced because of the then unforeseen sustained exponential growth of the Internet. The evolution largely centered

around the placement of the conceptual demarcator between the network ID and Host ID as shown in Figure 9.

Initially, the addressing followed a "*classful*" scheme where the address space was divided into a few blocks and static demarcators assigned to each block. Class A has a 8-bit demarcator; Class B has a 16-bit demarcator; Class C has a 24-bit demarcator. Class D was reserved for multicast and Class E for future use. This scheme is shown in Figure 10. This scheme ran into trouble in early 1980s because of two reasons: a) class B's were popular (class Cs largely unallocated) and b) the host space in class As and class Bs were largely unused because no single sub-network (eg: Ethernets) was large enough to utilize the space fully. The solution to these problems is simple -- allow the host space to be further subdivided; and allow demarcators to be placed more flexibly rather than statically.

These realizations led to the development of "subnet" and "supernet" masking respectively. A mask is a 32-bit pattern, the ones of which indicate the bits belonging to the network ID and the zeros indicate the host ID bits. For simplicity, the ones in the masks are contiguous. For example, a subnet mask 255.255.255.0 applied to IP address 128.113.40.50 indicates that the network ID has been extended from 16-bits (since this is a class B address) to 24-bits. Supernet masks are used between autonomous systems to indicate address allocations or to advertise networks for routing. For example the notation 198.28.29.0/18 indicates an 18-bit address space. The supernet mask written as /18 is actually 255.255.192.0. Observe that the 198.28.29.0 belonged to the class C space according to the earlier classful scheme and class C admits only of /24 networks (i.e. with host space of 8 bits).

class

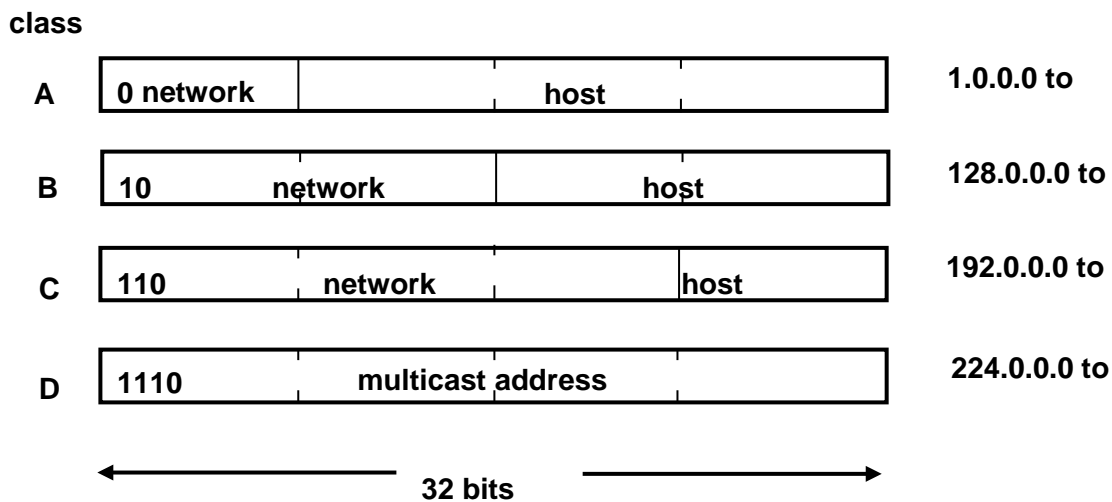|		
A	0 network host	**1.0.0.0 to**
B	10 network host	**128.0.0.0 to**
C	110 network host	**192.0.0.0 to**
D	1110 multicast address	**224.0.0.0 to**

← **32 bits** →

Figure 10: Initial Classful Addressing for IPv4

COMPUTER COMMUNICATION NETWORKS

Since these class boundaries are no longer valid with the supernet masks, this allocation scheme is also called "*classless*" allocation; and the routing scheme which accompanied this development is called *"Classless Inter-Domain Routing" (CIDR).* One effect of CIDR and supernet masking is that it is possible for a destination address to match multiple prefixes of different lengths. To resolve this, CIDR prescribes that the longest-prefix match be chosen for the L3 forwarding decision. As a result, all routers in the mid 1980s had to replace their forwarding algorithms. Similarly when subnet masking was introduced, hosts and routers had to be configured with subnet masks; and had to apply the mask in the forwarding process to determine the true network ID. Recall that the network ID is used to determine if the destination is directly or indirectly connected. These evolutionary changes are examples of how control-plane changes (CIDR and address allocation) could also affect the data-plane (IP forwarding) operation.

In modern networks, two other schemes are also used to further conserve public address space: DHCP and NAT. The Dynamic Host Configuration Protocol (DHCP) was originally a network "booting" protocol that configured essential parameters to hosts and routers. Now, it is primarily used to *lease* a pool of scarce public addresses among hosts who need it for connecting to the Internet. Observe that the leasing model means that host interfaces no longer "*own*" IP addresses.

The Network Address Translator (NAT) system enables the use of private address spaces within large enterprises. The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

  10.0.0.0   -   10.255.255.255 (10/8 prefix)

  172.16.0.0   -   172.31.255.255 (172.16/12 prefix)

  192.168.0.0   -   192.168.255.255 (192.168/16 prefix)

The NAT boxes at the edge of these private networks then translate public addresses to private addresses for all active sessions. Since early applications (eg: FTP) overloaded the semantics of IP addresses and included them in application-level fields, NAT has to transform these addresses as well. NAT breaks certain security protocols, notably IPSEC, which in part tries to ensure integrity of the IP addresses during transmission.

The combination of these techniques has delayed the deployment of IPv6 that proposes a more long-lived solution to address space shortage. IETF and the IPv6 Forum have been planning the deployment of IPv6 for over a decade now, and it remains to be seen what will be the major catalyst for IPv6 adoption. The potential growth of 3G wireless networks and/or the strain on inter-domain routing due to multi-homing have been recently cited as possible catalysts. ISPs

project that the IPv4 address space can be prolonged for another decade with the above techniques.

ARP, Fragmentation and Reassembly

Recall that the overlay model used by IP results in two mapping problems: address mapping and packet format mapping. The address mapping is resolved by a sub-IP protocol called ARP, while the packet mapping is done within IP by the fragmentation/reassembly procedures. The mapping problems in IP are far simpler than other internetworking protocols in the early 80s because IP has minimal expectations from the lower layers.

The address mapping problem occurs once the destination or next-hop is determined at the IP level (i.e. using the L3 forwarding table). The problem is as follows: the node knows the IP address of the next hop (which by definition is directly connected (i.e. accessible through layer 2 forwarding). But now to be able to use L2 forwarding, it needs to find out the next-hop's L2 address. Since the address spaces of L2 and L3 are independently assigned, the mapping is not a simple functional relationship, i.e., it has to be discovered dynamically. The protocol used to discover the L3 address to L2 address mapping is called the Address Resolution Protocol (ARP).

The ARP at the node sends out a link-level broadcast message requesting the mapping. Since the next hop is on the same layer-2 "*wire,*" it will respond with a unicast ARP reply to the node giving its L2 address. Then the node uses this L2 address, and encloses the IP datagram in the L2 frame payload and "*drops*" the frame on the L2 "*wire.*" ARP then uses caching (i.e. an ARP mapping table) to avoid the broadcast request-response for future packets. In fact, other nodes on the same L2 wire also snoop and update their ARP tables, thus reducing the need for redundant ARP broadcasts. Since the mapping between L3 and L2 addresses could change (because both L2 and L3 address can be dynamically assigned), the ARP table entries are aged and expunged after a timeout period.

The packet-mapping problem occurs when the IP datagram to be forwarded is larger than the maximum transmission unit (MTU) possible in the link layer. Every link typically has a MTU for reasons such as fairness in multiplexing, error detection efficiency etc. For example, Ethernet has an MTU of 1518 bytes. The solution is for the IP datagram to be fragmented such that each fragment fits the L2 payload. Each fragment now becomes an independent IP datagram; hence the IP header is copied over. However, it also needs to indicate the original datagram, the position (or offset) of the fragment in the original datagram and whether it is the last datagram. These pieces of information are filled into the fragmentation fields in the IP header (ID, flags, frag offset) respectively. The reassembly is then done at the IP layer in the ultimate destination. Fragments may come out-of-order or be delayed. A reassembly table data structure and a time-out per datagram is maintained at the receiver to implement this function. Reassembly is not attempted at intermediate routers because all fragments may not be routed through the same path.

COMPUTER COMMUNICATION NETWORKS

In general, though fragmentation is a necessary function for correctness, it has severe performance penalties. This is because any one of the fragments lost leads to the entire datagram being discarded at the receiver. Moreover, the remaining fragments that have reached the receiver (and are discarded) have consumed and effectively wasted scare resources at intermediate nodes. Therefore, modern transport protocols try to avoid fragmentation as much as possible by first discovering the minimum MTU of the path. This procedure is also known as "*path-MTU discovery*." Periodically (every 6 seconds or so), an active session will invoke the path-MTU procedure. The procedure starts by sending a maximum sized datagram with the "*do not fragment*" bit set in the flags field. When a router is forced to consider fragmentation due to a smaller MTU than the datagram, it drops the datagram and sends an ICMP message indicating the MTU of the link. The host then retries the procedure with the new MTU. This process is repeated till an appropriately sized packet reaches the receiver, the size of which is used as the maximum datagram size for future transmissions.

In summary, the mapping problems in IP are solved by ARP (a separate protocol) and fragmentation/reassembly procedures. Fragmentation avoidance is a performance imperative and is carried out through path MTU discovery. This completes the discussion of the key data-plane concepts in IP. The missing pieces now are the routing protocols used to populate forwarding tables such that a concatenation of local decisions (forwarding) leads to efficient global connectivity.

Routing in the Internet

Routing is the magic enabling connectivity. It is the control-plane function, which sets up the local forwarding tables at the intermediate nodes, such that a concatenation of local forwarding decisions leads to global connectivity. The global connectivity is also "efficient" in the sense that loops are avoided in the steady state.

Internet routing is scalable because it is hierarchical. There are two categories of routing in the Internet: inter-domain routing and intra-domain routing. *Inter*-domain routing is performed between autonomous systems (AS's). An autonomous system defines the locus of single administrative control and is internally connected, i.e., employs appropriate routing so that two internal nodes need not use an external route to reach each other. The internal connectivity in an AS is achieved through *intra*-domain routing protocols.

Once the nodes and links of a network are defined and the boundary of the routing architecture is defined, then the routing protocol is responsible for capturing and condensing the appropriate global state into local state (i.e. the forwarding table). Two issues in routing are *completeness* and *consistency*.

In the steady state, the routing information at nodes must be *consistent*, i.e., a series of independent local forwarding decisions must lead to connectivity between any (source, destination) pair in the network. If this condition is not true, then the routing algorithm is said to

not have "*converged*" to steady state, i.e., it is in a transient state. In certain routing protocols, convergence may take a long time. In general a part of the routing information may be consistent while the rest may be inconsistent. If packets are forwarded during the period of convergence, they may end up in loops or arbitrarily traverse the network without reaching the destination. This is why the TTL field in the IP header is used. In general, a faster convergence algorithm is preferred, and is considered more stable; but this may come at the expense of complexity. Longer convergence times also limit the scalability of the algorithm, because with more nodes, there are more routes, and each could have convergence issues independently.

Completeness means that every node has sufficient information to be able to compute all paths in the entire network locally. In general, with more complete information, routing algorithms tend to converge faster, because the chances of inconsistency reduce. But this means that more distributed state must be collected at each node and processed. The demand for more completeness also limits the scalability of the algorithm. Since both consistency and completeness pose scalability problems, large networks have to be structured hierarchically (eg: as areas in OSPF) where each area operates independently and views the other areas as a single border node.

Distance Vector and Link-State Algorithms and Protocols

In packet switched networks, the two main types of routing are link-state and distance vector. *Distance vector* protocols maintain information on a *per-node* basis (i.e. a vector of elements), where each element of the vector represents a distance or a path to that node. *Link state* protocols maintain information on a *per-link* basis where each element represents a weight or a set of attributes of a link. If a graph is considered as a set of nodes and links, it is easy to see that the link-state approach has complete information (information about links also implicitly indicates the nodes which are the end-points of the links) whereas the distance vector approach has incomplete information.

The basic algorithms of the distance vector (Bellman-Ford) and the link-state (Dijkstra) attempt to find the shortest paths in a graph, in a fully distributed manner, assuming that distance vector or link-state information can only be exchanged between immediate neighbors. Both algorithms rely on a simple recursive equation. Assume that the shortest distance path from node i to node j has distance D(i,j), and it passes through neighbor k to which the cost from i is c(i,k), then we have the equation:

$$D(i, j) = c(i,k) + D(k,j) \qquad (1)$$

In other words, the subset of a shortest path is also the shortest path between the two intermediate nodes.

COMPUTER COMMUNICATION NETWORKS

The *distance vector (Bellman-Ford) algorithm* evaluates this recursion iteratively by starting with initial distance values:

$D(i,i) = 0$;

$D(i,k) = c(i,k)$ if k is a neighbor (i.e. k is one-hop away); and

$D(i,k) = $ INFINITY for all other non-neighbors k.

Observe that the set of values $D(i,*)$ is a *distance vector at node i*. The algorithm also maintains a nexthop value for every destination j, initialized as:

next-hop(i) = i;

next-hop(k) = k if k is a neighbor, and

next-hop(k) = UNKNOWN if k is a non-neighbor.

Note that the next-hop values at the end of every iteration go into the forwarding table used at node i.

In every iteration each node i exchanges its distance vectors $D(i,*)$ with its immediate neighbors. Now each node i has the values used in equation (1), i.e. $D(i,j)$ for any destination and $D(k,j)$ and $c(i,k)$ for each of its neighbors k. Now if $c(i,k) + D(k,j)$ is smaller than the current value of $D(i,j)$, then $D(i,j)$ is replaced with $c(i,k) + D(k,j)$, as per equation (1). The next-hop value for destination j is set now to k. Thus after *m* iterations, each node knows the shortest path possible to any other node which takes *m* hops or less. Therefore the algorithm converges in $O(d)$ iterations where d is the maximum diameter of the network. Observe that each iteration requires information exchange between neighbors. At the end of each iteration, the next-hop values for every destination j are output into the forwarding table used by IP.

The *link state (Dijkstra) algorithm* pivots around the link cost $c(i,k)$ and the destinations j, rather than the distance $D(i,j)$ and the source i in the distance-vector approach. It follows a greedy iterative approach to evaluating (1), but it collects all the link states in the graph *before* running the Dijkstra algorithm *locally*. The Dijkstra algorithm at node i maintains two sets: set N that contains nodes to which the shortest paths have been found so far, and set M that contains all other nodes. Initially, the set N contains node i only, and the next hop (i) = i. For all other nodes k a value $D(i,k)$ is maintained which indicates the current value of the path cost (distance) from i to k. Also a value $p(k)$ indicates what is the predecessor node to k on the shortest known path from i (i.e. $p(k)$ is a neighbor of k). Initially,

$D(i,i) = 0$ and $p(i) = i$;

$D(i,k) = c(i,k)$ and $p(k) = i$ if k is a neighbor of i

COMPUTER COMMUNICATION NETWORKS

D(i,k) = INFINITY   and   p(k) = UNKNOWN if k is *not* a neighbor of i

Set N contains node i only, and the next hop (i) = i.

Set M contains all other nodes j.

In each iteration, a new node j is moved from set M into the set N. Such a node j has the minimum distance among all current nodes in M, i.e. $D(i,j) = \min_{\{l \,\varepsilon\, M\}} D(i,l)$. If multiple nodes have the same minimum distance, any one of them is chosen as j. Node j is moved from set M to set N, and the next-hop(j) is set to the neighbor of i on the shortest path to j. Now, in addition, the distance values of any neighbor k of j in set M is reset as:

If D(i,k) < c(j,k) + D(i,j), then D(i,k) = c(j,k) + D(i,j), and p(k) = j.

This operation called "*relaxing*" the edges of j is essentially the application of equation (1). This defines the end of the iteration. Observe that at the end of iteration *p* the algorithm has effectively explored paths, which are *p* hops or smaller from node i. At the end of the algorithm, the set N contains all the nodes, and knows all the next-hop(j) values which are entered into the IP forwarding table. The set M is empty upon termination. The algorithm requires *n* iterations where *n* is the number of nodes in the graph. But since the Dijkstra algorithm is a *local* computation, they are performed much quicker than in the distance vector approach. The complexity in the link-state approach is largely due to the need to wait to get all the link states c(j,k) from the entire network.

The protocols corresponding to the distance-vector and link-state approaches for *intra*-domain routing are RIP and OSPF respectively. In both these algorithms if a link or node goes down, the link costs or distance values have to be updated. Hence information needs to be distributed and the algorithms need to be rerun. RIP is used for fairly small networks mainly due to a convergence problem called "*count-to-infinity.*" The advantage of RIP is simplicity (25 lines of code!). OSPF is a more complex standard that allows hierarchy and is more stable than RIP. Therefore it is used in larger networks (esp enterprise and ISP internal networks). Another popular link-state protocol commonly used in ISP networks is IS-IS, which came from the ISO/OSI world, but was adapted to IP networks.

Routers and Routing Algorithms

Router: a network device working in the *network layer*; it receives packets, puts them in a queue and dispatches the packets to the links toward their destinations. To do this, it uses the *IP header* of packets together with its precalculated *forwarding table*.

Routing

- Intra-AS:

    - Routing within a single *autonomous system*: a network with a single administrator.

    - Example: OSPF and RIP routing protocols

- Inter-AS:

    - sometimes called *policy routing*;

    - competition and security issues may arise because of communicating among networks with different administrators.

    - Example: BGP routing protocol.

OSPF

OSPF[1] is a routing protocol based on the *Dijkstra's algorithm*. It constructs a shortest path tree for each source. It is robust against link failures and converges quite rapidly to new solutions in response to dynamic changes in the network.

RIP

RIP acts based on the *Bellman-Ford algorithm*. Each node computes and stores a local routing table containing information about *which nodes* can be reached by *how much cost (distance)* and through *which neighbor (next-hop)*. To propagate this information throughout the network, each node sends its (node, distance) information to all its neighbors. This happens on a regular basis after each *t* time units to ensure that the node is live, or when an update occurs.

COMPUTER COMMUNICATION NETWORKS

When a link cost goes down:

Consider the following example, where cost of the link zw changes from ∞ to 1.
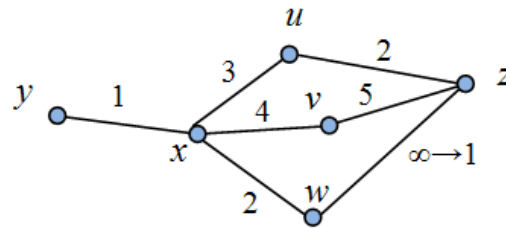
Initially, routing tables for w and z are:

z's Routing table		
node	dist	next-hop
u	2	u
v	5	v
w	7	u
x	5	u
y	6	u
z	0	z

w's Routing table		
node	dist	next-hop
u	5	x
v	6	x
w	0	w
x	2	x
y	3	x
z	7	x

After the link appears, z and w know that they are now connected with a weight 1 link. They send their routing tables to each other, and update the routing tables based on the new information:

z's Routing table		
node	dist	next-hop
u	2	u
v	5	v
w	1	w
x	3	w
y	4	w
z	0	z

w's Routing table		
node	dist	next-hop
u	3	z
v	6	x
w	0	w
x	2	x
y	3	x
z	1	z

Apparently, in this case the algorithm converges rapidly.

When a link cost goes up:

Now consider the graph below, where cost of the edge ab increases from 1 to 20.

COMPUTER COMMUNICATION NETWORKS
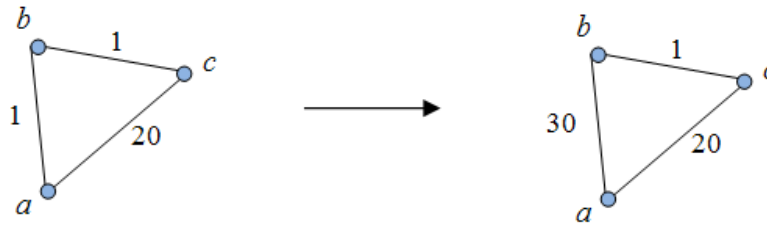
Initially, routing tables of *c* and *b* are as follows:

	b's Routing table				c's Routing table	
node	dist	next-hop		node	dist	next-hop
a	1	a		a	2	b
c	1	c		b	1	b

When *a* and *b* discover that the link cost has been increased, they update their tables and send them to their neighbors. As the link cost is high, *b* and *a* no more route through each other. But *c* doesn't realize the fact, and feeds *a* and *b* with obsolete information. This causes a long sequence of alternating changes in routing tables of *b* and *c*, until *c* realizes that to reach *a*, it shouldn't rout through *b*.

	b's Routing table				c's Routing table	
node	dist	next-hop		node	dist	next-hop
a	3	c		a	4	b
c	1	c		b	1	b

	b's Routing table				c's Routing table	
node	dist	next-hop		node	dist	next-hop
a	5	c		a	6	b
c	1	c		b	1	b

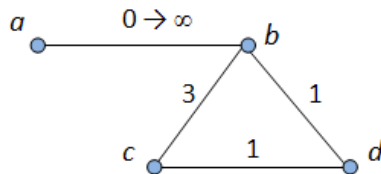|	b's Routing table				c's Routing table	
node	dist	next-hop		node	dist	next-hop
a	7	c		a	8	b
c	1	c		b	1	b

.
.
.

As you see, when link cost increases, convergence time is very high. This problem is known as "*The Counting to Infinity Problem*". It is caused by the fact that *b* and *c* are engaged in a   pattern

of mutual deception. Each claims to be able to get to *a* via the other. To solve the problem, we note that it is never useful to claim reachability for a destination node to the neighbor from which the route passes. "*Split horizon with poisoned reverse*" solves this problem based on this fact:

If node b is next hop on your path to a, advertise "infinity to a" to b.

But the problem still remains for longer loops:

Initially, the routing tables are:

b's Routing table				d's Routing table				c's Routing table		
node	dist	next-hop		node	dist	next-hop		node	dist	next-hop
a	0	a		a	1	b		a	2	d
c	2	d		b	1	b		b	2	d
d	1	d		c	1	c		d	1	d

Then, *b* finds out that it is no more linked to *a*. Also, *b* receives "∞ to a" from *d*. Thus, it can route toward *a* using *c*. What happens is in an infinite loop, b,c, and d update their route to a: b routes through c, c through d and d through a. Although the "Split horizon with poisoned reverse" scheme is applied, it turns out to fail for loops of length three or more.

b's Routing table				d's Routing table				c's Routing table		
node	dist	next-hop		node	dist	next-hop		node	dist	next-hop
a	5	c		a	6	b		a	7	d
c	2	d		b	1	b		b	2	d
d	1	d		c	3	c		d	1	d

b's Routing table				d's Routing table				c's Routing table		
node	dist	next-hop		node	dist	next-hop		node	dist	next-hop
a	10	c		a	11	b		a	12	d
c	2	d		b	1	b		b	2	d
d	1	d		c	3	c		d	1	d

b's Routing table				d's Routing table				c's Routing table		
node	dist	next-hop		node	dist	next-hop		node	dist	next-hop
a	15	c		a	16	b		a	17	d
c	2	d		b	1	b		b	2	d
d	1	d		c	3	c		d	1	d

.
.
.

COMPUTER COMMUNICATION NETWORKS

To come up with the problem: BGP has put an upperbound on the number of hops. Thus infinity is no more unachievable; a specific number e.g. 15 is considered as infinity.

Firewalls

- **Firewalls are sets of *Rules***: a list of criteria a packet should satisfy or it will be blocked and won't be allowed to enter the network.

- **TCP flags *SYN* and *FIN***: SYN and FIN flags in TCP header used to manage TCP connections. A SYN/FIN packet is the most well known illegal combination [2].

- Example rules:

  - If a packet comes from a trusted source, accept.

  - If SYN and FIN bits are both set, reject.

- If there is more than one match, the first match is important, i.e. we look through the rules and we stop once we find the first match.

In a sense, this works similar to the forwarding table. The forwarding table in a router determines to which link a packet should be dispatched, considering a prefix of its destination IP address. In case of a forwarding table, if multiple matches exist, the longest should be taken into account.

Forwarding tables

Tables used in network layer to route packets based on the prefix of their destination IP address.

Remark: IP classes

IP address is a 32 bit number. These addresses admit a hierarchical structure. All hosts within a single network have similar parts of IP used as the network address. To route a packet to its destination, we first route it toward the destination network which in turn directs the packet to its destination. Thus at the first step, the host address doesn't matter. This is analogous to hierarchical structure of mailing addresses: as long as a package hasn't reached your house, the apartment number is not important.

Previously, IP addresses belonged to one of the following classes[3]:

- Class A: A small number (126) of networks with huge number (2^{24}) of hosts.

---

[2]   Since a SYN packet is used to initiate a connection, it should never have the FIN flag set in conjunction. It is always a malicious attempt at getting past your firewall. Thus most firewalls are now aware of SYN/FIN packets.

[3]   There are also two other classes D and E.

COMPUTER COMMUNICATION NETWORKS

- Class B: For medium sized networks.

- Class C: For small networks of at most 256 hosts.
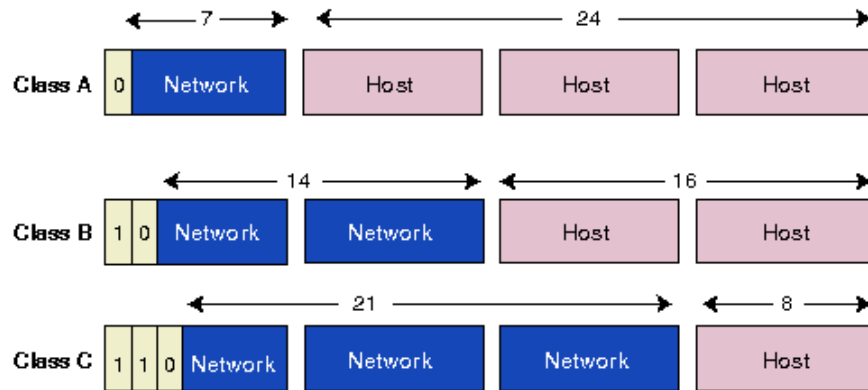
This way of classification caused waste of IP space for two reasons: Few people liked class C addresses because they were too restrictive; also no network with a class A IP could fill up a reasonable fraction of 2^{24} IPs. This happens because the gap between type B and C is large, making the options too discrete. To prevent waste of IP addresses, we have CIDR[4]. In CIDR, we dedicate some bits of the network address to the host address. Network addresses are numbers like 112.56.0/14 where /14 means we should only look at the first 14 bits for subnet mask (IP range).

Routing tables do "Longest Prefix Matching": they route packets based on the longest match in the routing table that is a prefix of destination address.

Example.

prefix	next-hop
1) 01101*	A
2) 011*	B
3) 001111*	A
4) 001110*	B
5) 110111*	C
6) 1*	A
7) *	C

---

4   Classless Inter-Domain Routing

COMPUTER COMMUNICATION NETWORKS

a) 01100001…….. : Matches 2 and 7, hence goes through B.

b) 01101111…….. : Matches 1, 2 and 7, thus goes through A based on (1).

c) 00110000…….. : Matches default column 7: goes through C.

To better represent routing tables, they are stored in data structures called *Trie*. A trie is a tree in which edges represent characters, and each of the nodes represents the string obtained by traversing all edges from the root to that node. To store the routing table using trie, we insert each of the rows as a string in the trie, and for the nodes at the end of each string, we determine the next-hop. To achieve a more compressed representation, we may combine the nodes which have only one child with their descendents. Here is a trie for the above example: