Unit 1

[Unit 1] **6 Hrs**

Introduction: Applications of computer networks, Network hardware, Network software: Protocol Hierarchy, Design Issue, connection oriented vs. connectionless, Service Primitives, Reference models: OSI and TCP/IP, Example networks: Internet, Network standardization, Performance: Bandwidth and Latency, Delay and bandwidth product, High-Speed Network, Application Performance Needs.

INTRODUCTION

Applications of computer networks

USES OF COMPUTER NETWORKS

1. **Business Applications**

- to distribute information throughout the company (resource sharing). sharing physical resources such as printers, and tape backup systems, is sharing information
- **client-server model**. It is widely used and forms the basis of much network usage.
- **communication medium** among employees.email (electronic mail), which employees generally use for a great deal of daily communication.
- Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called IP telephony or **Voice over IP (VoIP**) when Internet technology is used.
- Desktop sharing lets remote workers see and interact with a graphical computer screen
- doing business electronically, especially with customers and suppliers. This new model is called e-commerce (electronic commerce) and it has grown rapidly in recent years.

2. .**Home Applications**

- peer-to-peer communication
  - person-to-person communication
- electronic commerce
  - entertainment.(game playing,)

3. **Mobile Users**

  - Text messaging or texting
  - Smart phones,
  - GPS (Global Positioning System)
  - m-commerce
  - NFC (Near Field Communication)

4 Social Issues

With the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues.

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies

like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a live-and-let-live view, but others feel that posting certain material (e.g., verbal attacks on particular countries or religions, pornography, etc.) is simply unacceptable and that such content must be censored. Different countries have different and conflicting laws in this area. Thus, the debate rages.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as employee rights versus employer rights.

Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights.

A new twist with mobile devices is location privacy. As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent and which medical center you visit.

Phishing ATTACK: Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message.

1.2 NETWORK HARDWARE

It is now time to turn our attention from the applications and social aspects of networking (the dessert) to the technical issues involved in network design (the spinach). There is no generally accepted taxonomy into which all computer networks fit, but two dimensions stand out as important: transmission technology and

scale. We will now examine each of these in turn.

Broadly speaking, there are two types of transmission technology that are in widespread use: **broadcast** links and **point-to-point** links.

Point-to-point links connect individual pairs of machines. To go from the source to the destination on a network made up of point-to-point links, short messages, called **packets** in certain contexts, may have to first visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so finding good ones is important in point-to-point networks. Point-to-point transmission with exactly one sender and exactly one receiver is sometimes called **unicasting**.

In contrast, on a broadcast network, the communication channel is shared by all the machines on the network; packets sent by any machine are received by all the others. An address field within each packet specifies the intended recipient.

Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

A wireless network is a common example of a broadcast link, with communication shared over a coverage region that depends on the wireless channel and the transmitting machine. As an analogy, consider someone standing in a meeting room and shouting ‘‘Watson, come here. I want you.’’ Although the packet may actually be received (heard) by many people, only Watson will respond; the others just ignore it.

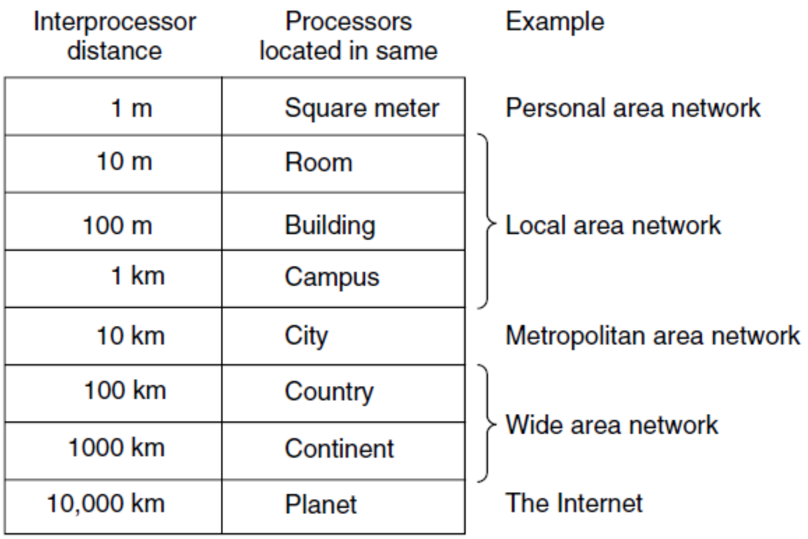
Broadcast systems usually also allow the possibility of addressing a packet to *all* destinations by using a special code in the address field. When a packet with this code is transmitted, it is received and processed by every machine on the network.

This mode of operation is called **broadcasting**. Some broadcast systems also support transmission to a subset of the machines, which known as **multicasting**.

An alternative criterion for classifying networks is by scale. Distance is important as a classification metric because different technologies are used at different scales.

In Fig. 1-6 we classify multiple processor systems by their rough physical size. At the top are the personal area networks, networks that are meant for one person. Beyond these come longer-range networks. These can be divided into local, metropolitan, and wide area networks, each with increasing scale. Finally, the connection of two or more networks is called an internetwork. The worldwide Internet is certainly the best-known (but not the only) example of an internetwork.

Soon we will have even larger internetworks with the Interplanetary Internet that connects networks across space (Burleigh et al., 2003).



Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Figure 1-6. Classification of interconnected processors by scale.

Types of Network based on size

The types of network are classified based upon the size, the area it covers and its physical architecture. The three primary network categories are LAN, WAN and MAN. Each network differs in their characteristics such as distance, transmission speed, cables and cost.

Basic types

LAN (Local Area Network)

Group of interconnected computers within a small area. (room, building, campus) Two or more pc's can from a LAN to share files, folders, printers, applications and other devices.

Coaxial or CAT 5 cables are normally used for connections.

Due to short distances, errors and noise are minimum.

Data transfer rate is 10 to 100 mbps.

Example: A computer lab in a school.

MAN (Metropolitan Area Network)

Design to extend over a large area.

Connecting number of LAN's to form larger network, so that resources can be shared.

Networks can be up to 5 to 50 km.

Owned by organization or individual.

Data transfer rate is low compare to LAN.

Example: Organization with different branches located in the city.

WAN (Wide Area Network)

Are country and worldwide network.

Contains multiple LAN's and MAN's.

Distinguished in terms of geographical range.

Uses satellites and microwave relays.

Data transfer rate depends upon the ISP provider and varies over the location.

Best example is the internet.

Other types

WLAN (Wireless LAN)

A LAN that uses high frequency radio waves for communication.

Provides short range connectivity with high speed data transmission.

PAN (Personal Area Network)

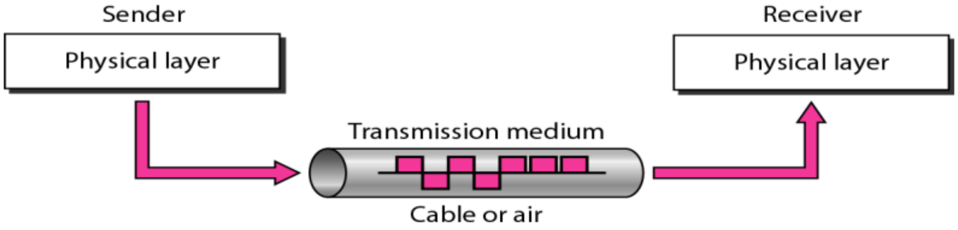
Network organized by the individual user for its personal use.

SAN (Storage Area Network)

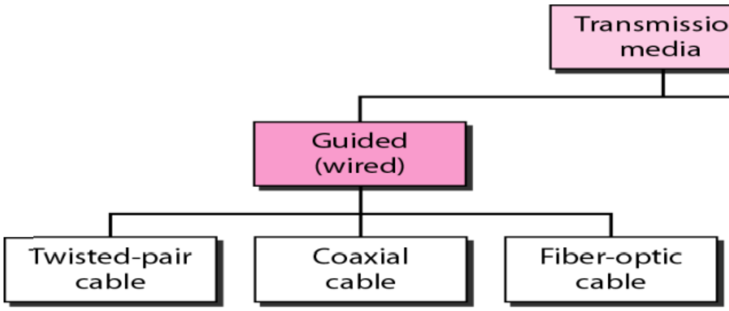
Connects servers to data storage devices via fiber-optic cables.

E.g.: Used for daily backup of organization or a mirror copy

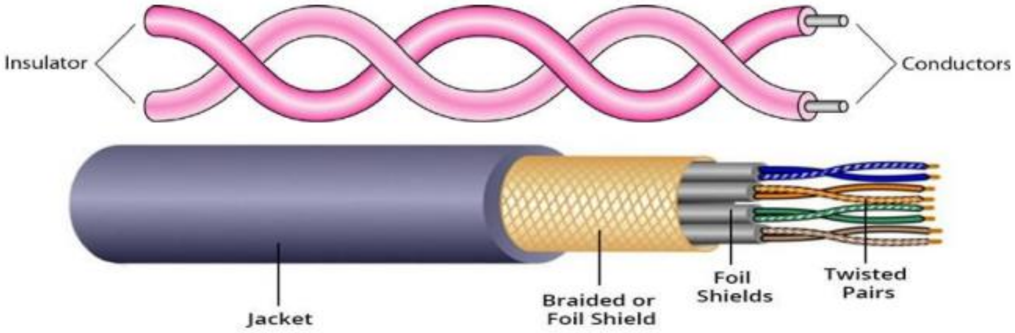
A transmission medium can be broadly defined as anything that can carry information from a source to a destination.



Classes of transmission media

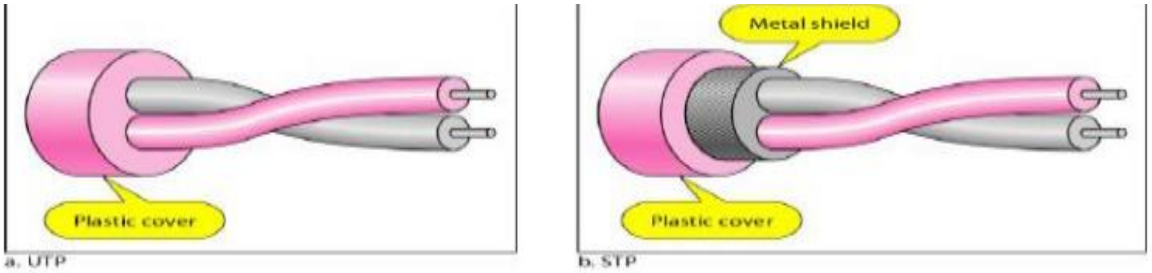


Guided Media: Guided media, which are those that provide a medium from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable. Twisted-Pair Cable: A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.



Unshielded Versus Shielded Twisted-Pair Cable The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive

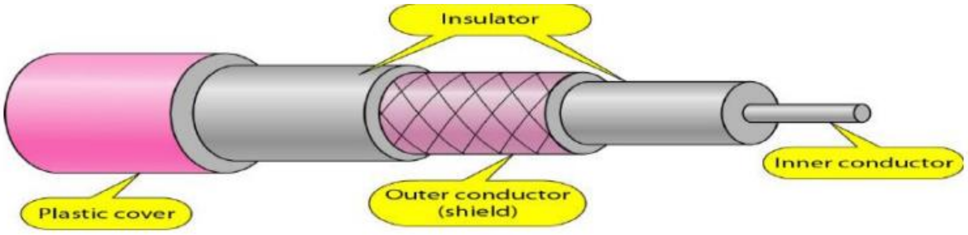
.

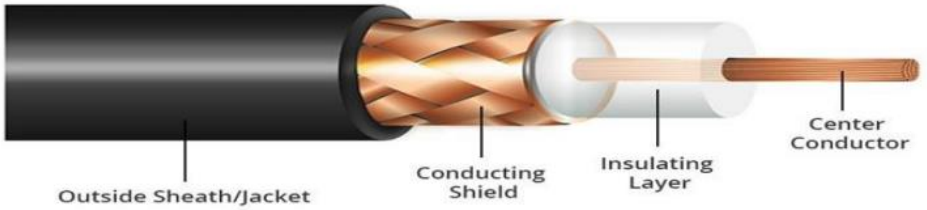


The most common UTP connector is RJ45 (RJ stands for registered jack)

Applications Twisted-pair cables are used in telephone lines to provide voice and data channels. Local-area networks, such as l0Base-T and l00Base-T, also use twisted-pair cables.

Coaxial Cable Coaxial cable (or coax) carries signals of higher frequency ranges than those in twisted pair cable. coax has a central core conductor of solid or stranded wire (usuallycopper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit.This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover.





The most common type of connector used today is the Bayone-Neill-Concelman (BNC), connector.

Applications

Coaxial cable was widely used in analog telephone networks,digital telephone networks

Cable TV networks also use coaxial cables.

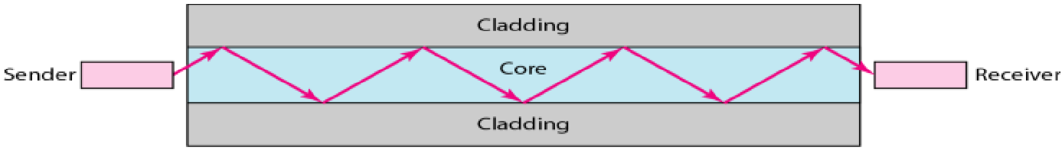
Another common application of coaxial cable is in traditional Ethernet LANs

Fiber-Optic Cable

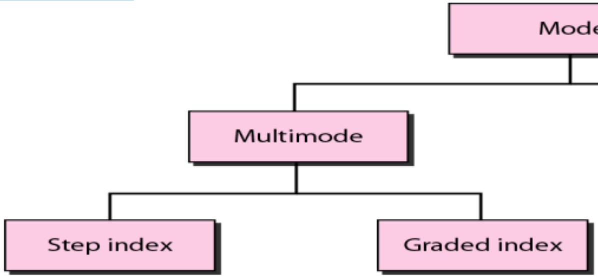
A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. Light travels in a straight line as long as it is moving through a single uniform substance.

If a ray of light traveling through one substance suddenly enters another substance(of a different density), the ray changes direction.

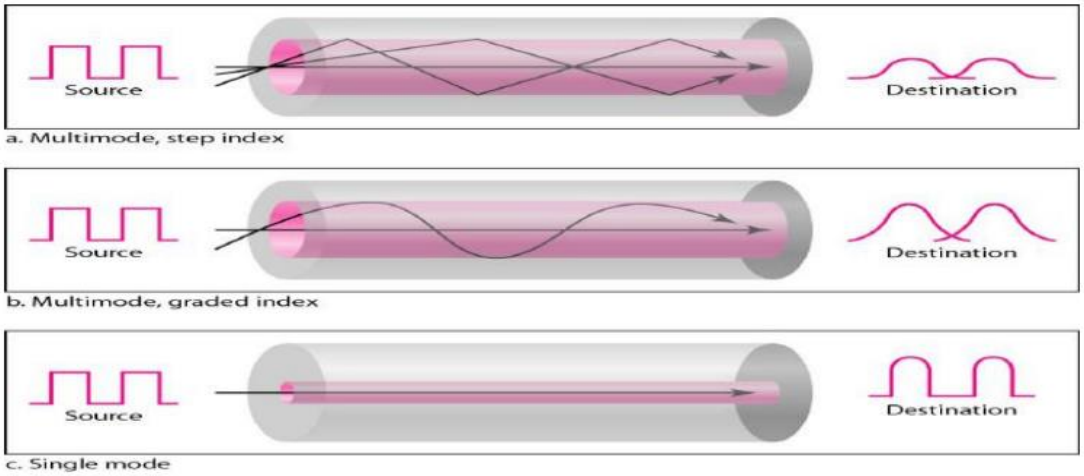
Optical fibers use reflection to guide light through a channel. A glass or plastic core is surrounded by a cladding of less dense glass or plastic.



Propagation Modes



Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core, as shown in Figure.



Advantages and Disadvantages of Optical Fiber Advantages Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial)

. 1 Higher bandwidth.

2 Less signal attenuation. Fiber-optic transmission distance is significantly greaterthan that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twistedpair cable.

3 Immunity to electromagnetic interference. Electromagnetic noise cannot affect fiber-optic cables.

4 Resistance to corrosive materials. Glass is more resistant to corrosive materials than copper.

5 Light weight. Fiber-optic cables are much lighter than copper cables.

6 Greater immunity to tapping. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

Disadvantages There are some disadvantages in the use of optical fiber.

1Installation and maintenance

2 Unidirectional light propagation. Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.

3 Cost. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

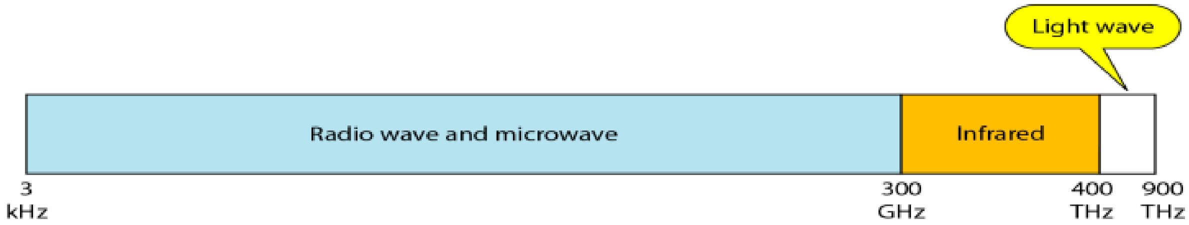
UNGUIDED MEDIA: WIRELESS

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

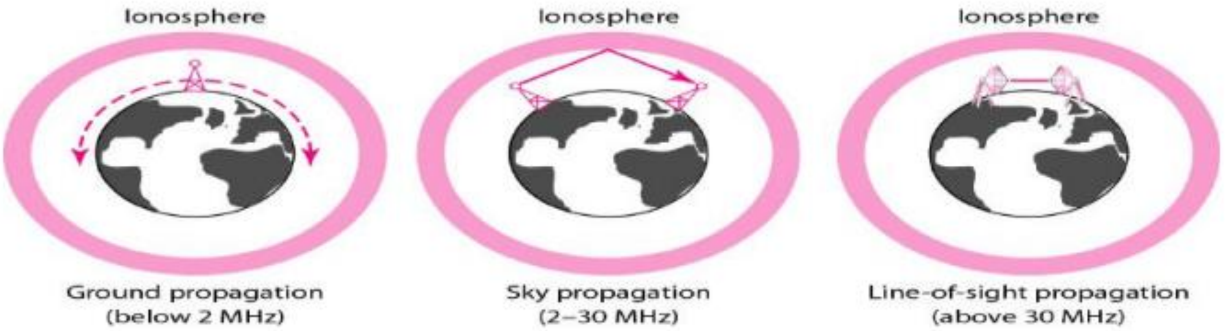
Radio Waves

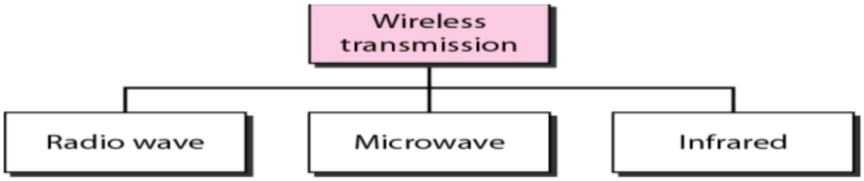
Microwaves

Infrared



Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure


Ionosphere
Ionosphere
Ionosphere
Ground propagation (below 2 MHz)
Sky propagation (2–30 MHz)
Line-of-sight propagation (above 30 MHz)



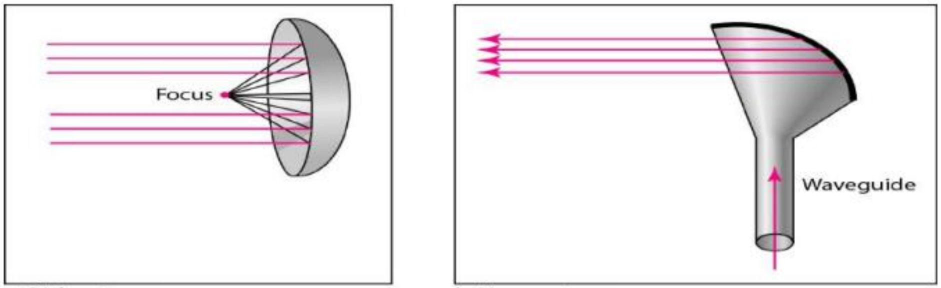
Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves. Radio waves are omni directional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omni directional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

The Omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

Microwaves Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. The sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn



a. Dish antenna

b. Horn antenna

Applications: Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs

Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous characteristic prevents interference between one system and another; a shortrange communication system in one room cannot be affected by another system in the next room. When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors.

Infrared signals useless for **long-range communication**. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication. Applications: Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

NETWORK SOFTWARE

The first computer networks were designed with the hardware as the main concern and the software as an afterthought. This strategy no longer works. Network software is now highly structured. In the following sections we examine the software structuring technique in some detail. The approach described here forms the keystone of the entire book and will occur repeatedly later on.

1.3.1 Protocol Hierarchies

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is actually a familiar one and is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

When layer n on one machine carries on a conversation with layer n on another machine, the rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a protocol is an agreement between the communicating parties on how communication is to proceed. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn, may decide to either shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

A five-layer network is illustrated in Fig. 1-13. The entities comprising the corresponding layers on different machines are called peers. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

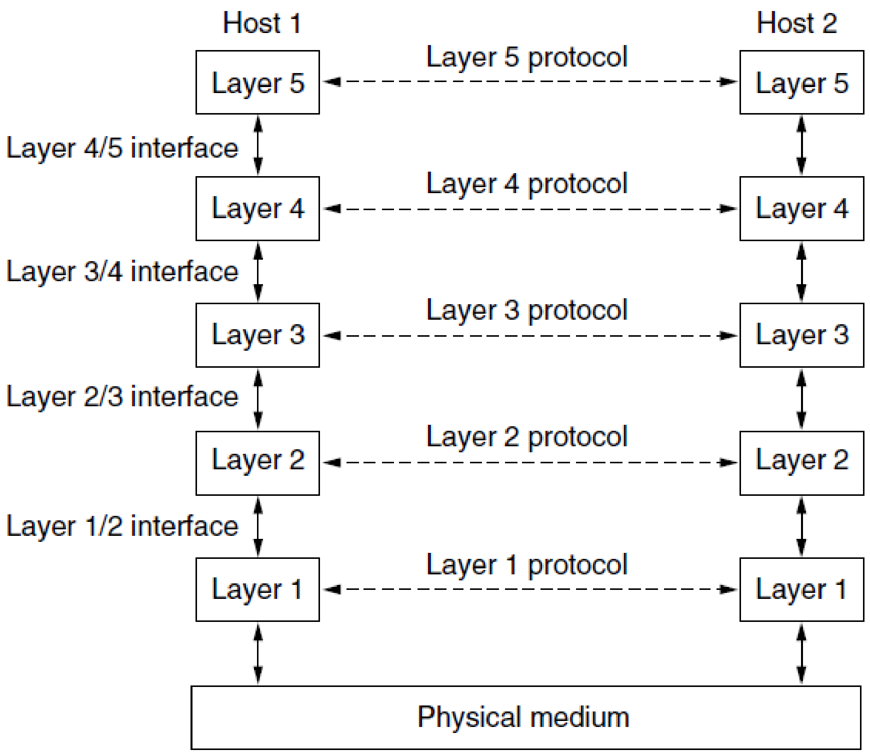


Figure 1-13. Layers, protocols, and interfaces.

Design Issues for the Layers

1. **Error detection and Correction**
2. **Routing**
3. **Addressing and Naming**
4. **Scalability**
5. **Flow control**
6. **Security**

Connection-Oriented Versus Connectionless Service

Connection-oriented service is modeled after the telephone system.

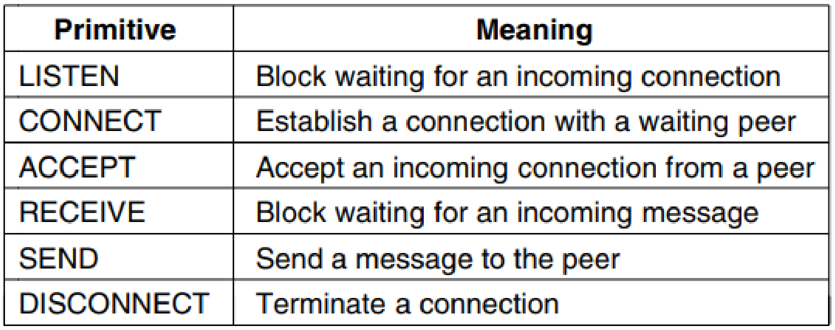
To use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection.

Criteria	Connection Oriented	Connectionless
Connection	Prior connection needs to be established	No prior connection needs to be established
Resource Allocation	Resources need to be allocated	No prior resources need to be allocated
Reliability	It ensures reliable transfer of data	Reliability is not guaranteed as it is a best effort service.
Congestion	Congestion is not at all possible	Congestion can occur likely
Transfer mode	It can be implemented either using Circuit switching or VCs	It is implemented using Packet Switching
Retransmission	It is possible to retransmit the lost data bits	It is not possible
Suitability	It is suitable for long and steady communication	It is suitable for busty transmission
Signalling	Connection is established through process of signaling	There is no concept of signaling
Packet Travel	In this packets travel to their destination node in a sequential manner.	In this packets travel to their destination node in random manner.
Delay	There is more delay in transfer of information, but once connection established faster delivery.	There is no delay due absence of connection establishment

Service Primitives

A service is formally specified by a set of primitives (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets. The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service. As a minimal example of the service primitives that might provide a reliable byte stream, consider the primitives listed in Fig. 1-17.

They will be familiar to fans of the Berkeley socket interface, as the primitives are a simplified version of that interface.



Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Figure 1-17. Six service primitives that provide a simple connection-oriented service.

These primitives might be used for a request-reply interaction in a client-server environment. To illustrate how, We sketch a simple protocol that implements the service using acknowledged datagrams.

First, the server executes LISTEN to indicate that it is prepared to accept incoming connections. A common way to implement LISTEN is to make it a blocking system call. After executing the primitive, the server process is blocked until a request for connection appears.

Next, the client process executes CONNECT to establish a connection with theserver. The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address. The operating system then typically sends a packet to the peer asking it to connect, as shown by (1) in Fig. 1-18. The client process is suspended until there is a response.

When the packet arrives at the server, the operating system sees that the packet is requesting a connection. It checks to see if there is a listener, and if so it unblocks the listener. The server process can then establish the connection with the ACCEPT call. This sends a response (2) back to the client process to accept connection. The arrival of this response then releases the client.

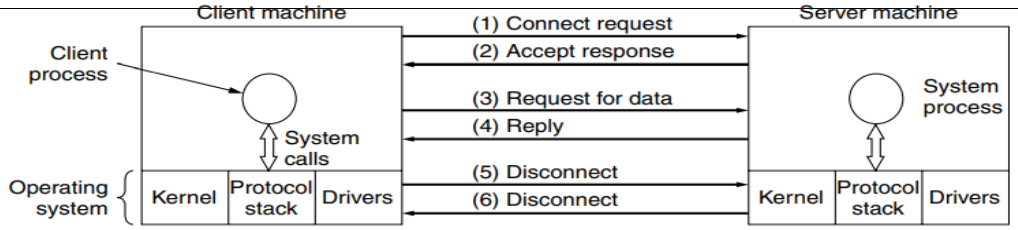


Figure 1-18. A simple client-server interaction using acknowledged datagrams.

At this point the client and server are both running and they have a connection established.

The obvious analogy between this protocol and real life is a customer (client) calling a company's customer service manager. At the start of the day, the service manager sits next to his telephone in case it rings. Later, a client places a call.

When the manager picks up the phone, the connection is established.

The next step is for the server to execute RECEIVE to prepare to accept the first request. Normally, the server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the server.

Then the client executes SEND to transmit its request (3) followed by the execution of RECEIVE to get the reply. The arrival of the request packet at the server machine unblocks the server so it can handle the request. After it has done the

work, the server uses SEND to return the answer to the client (4). The arrival of this packet unblocks the client, which can now inspect the answer. If the client has additional requests, it can make them now.

When the client is done, it executes DISCONNECT to terminate the connection (5). Usually, an initial DISCONNECT is a blocking call, suspending the client and sending a packet to the server saying that the connection is no longer needed. When the server gets the packet, it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6). When the server's packet gets back to the client machine, the client process is released and the connection is broken. In a nutshell, this is how connection-oriented communication works.

Of course, life is not so simple. Many things can go wrong here. The timing can be wrong (e.g., the CONNECT is done before the LISTEN), packets can get lost, and much more. We will look at these issues in great detail later, but for the

moment, Fig. 1-18 briefly summarizes how client-server communication might work with acknowledged datagrams so that we can ignore lost packets.

Given that six packets are required to complete this protocol, one might wonder why a connectionless protocol is not used instead. The answer is that in a perfect world it could be, in which case only two packets would be needed: one for the request and one for the reply. However, in the face of large messages in either direction (e.g., a megabyte file), transmission errors, and lost packets, the situation changes. If the reply consisted of hundreds of packets, some of which could be lost during transmission, how would the client know if some pieces were missing? How would the client know whether the last packet actually received was really the last packet sent? Suppose the client wanted a second file. How could it tell packet 1 from the second file from a lost packet 1 from the first file that suddenly found its way to the client? In short, in the real world, a simple request-reply protocol over an unreliable network is often inadequate.

Reference models: OSI and TCP/IP

OSI model was first introduced in 1984 by the International Organization for Standardization (ISO).

– Outlines **WHAT** needs to be done to send data from one computer to another.

– Not **HOW** it should be done.

– Protocols stacks handle how data is prepared for transmittal (to be transmitted)

● In the OSI model, The specification needed – are contained in 7 different layers that interact with each other.

What is "THE MODEL?"

● Commonly referred to as the OSI reference model.

● The OSI model

– is a theoretical blueprint that helps us understand how data gets from one user's computer to another.

– It is also a model that helps develop standards so that all of our hardware and software talks nicely to each other.

– It aids standardization of networking technologies by providing an organized structure for hardware and software developers to follow, to insure there products are compatible with current and future technologies.

7 Layer OSI Model

● Why use a reference model?

– Serves as an outline of rules for how protocols can be used to allow communication between computers.

– Each layer has its own function and provides support to other layers.

● Other reference models are in use.

– Most well known is the TCP/IP reference model.

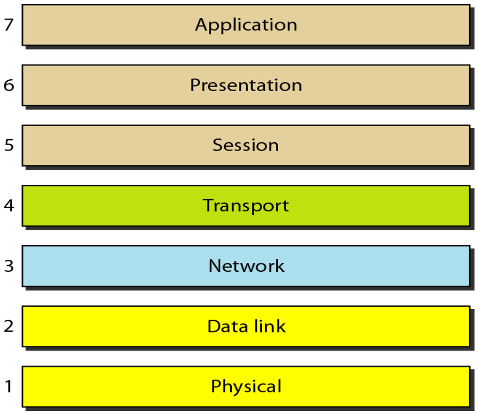
– We will compare OSI and TCP/IP models

● As computing requirements increased, the network modeling had to evolve to meet ever increasing demands of larger networks and multiple venders.

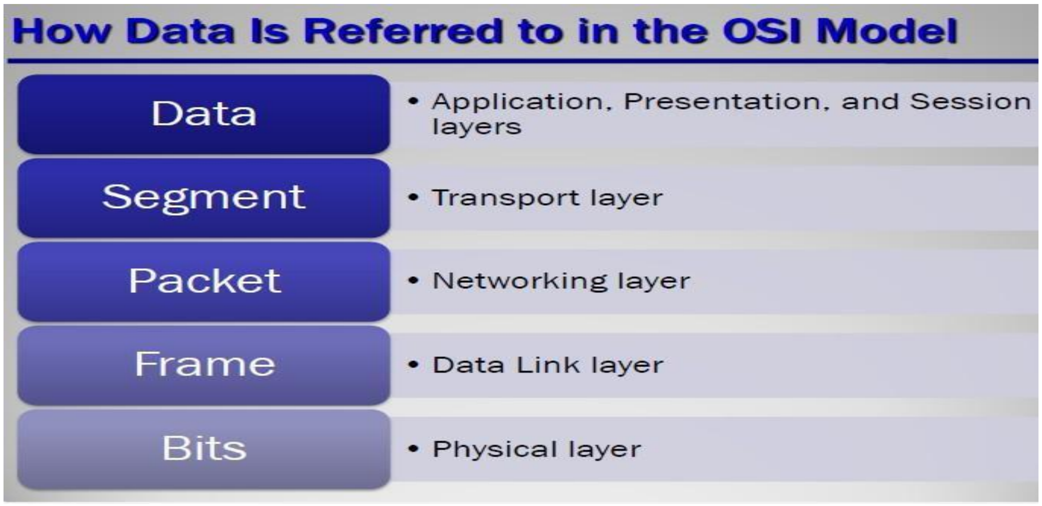
● Problems and technology advances also added to the demands for changes in network modeling.

OSI

- OSI stands for Open Systems Interconnection
- Created by International Standards Organization(ISO)
- Was created as a framework and reference model to explain how different networking technologies work together and interact
- It is not a standard that networking protocols must follow
- Each layer has specific functions it is responsible for
- All layers work together in the correct order to move data around a network



Top to bottom
–All People Seem To Need Data Processing Bottom to top
–Please Do Not Throw Sausage Pizza Away



How Data Is Referred to in the OSI Model

Data	• Application, Presentation, and Session layers
Segment	• Transport layer
Packet	• Networking layer
Frame	• Data Link layer
Bits	• Physical layer

Physical Layer

- Deals with all aspects of physically moving data from one computer to the next
- Converts data from the upper layers into 1s and 0s for transmission over media
- Defines how data is encoded onto the media to transmit the data
- Defined on this layer: Cable standards, wireless standards, and fiber optic standards.

Copper wiring, fiber optic cable, radio frequencies, anything that can be used to transmit data is defined on the Physical layer of the OSI Model

- Device example: Hub
- Used to transmit data

Data Link Layer

- Is responsible for moving frames from node to node or computer to computer
- Can move frames from one adjacent computer to another, cannot move frames across routers
- Encapsulation = frame
- Requires MAC address or *physical address*
- Protocols defined include Ethernet Protocol and Point-to-Point Protocol (PPP)
- Device example: Switch
- Two sublayers: Logical Link Control (LLC) and the Media Access Control (MAC)

○ Logical Link Control (LLC)

□ –Data Link layer addressing, flow control, address notification, error control

○ Media Access Control (MAC)

□ –Determines which computer has access to the network media at any given time

□ –Determines where one frame ends and the next one starts, called frame synchronization

Network Layer

- Responsible for moving packets (data) from one end of the network to the other, called *end-to-end communications*
- Requires *logical addresses* such as IP addresses
- Device example: Router
- –Routing is the ability of various network devices and their related software to move data packets from source to destination

Transport Layer

- Takes data from higher levels of OSI Model and breaks it into segments that can be sent to lower-level layers for data transmission
- Conversely, reassembles data segments into data that higher-level protocols and applications can use
- Also puts segments in correct order (called sequencing) so they can be reassembled in correct order at destination
- Concerned with the reliability of the transport of sent data
- May use a *connection-oriented protocol* such as TCP to ensure destination received segments
- May use a *connectionless protocol* such as UDP to send segments without assurance of delivery
- Uses port addressing

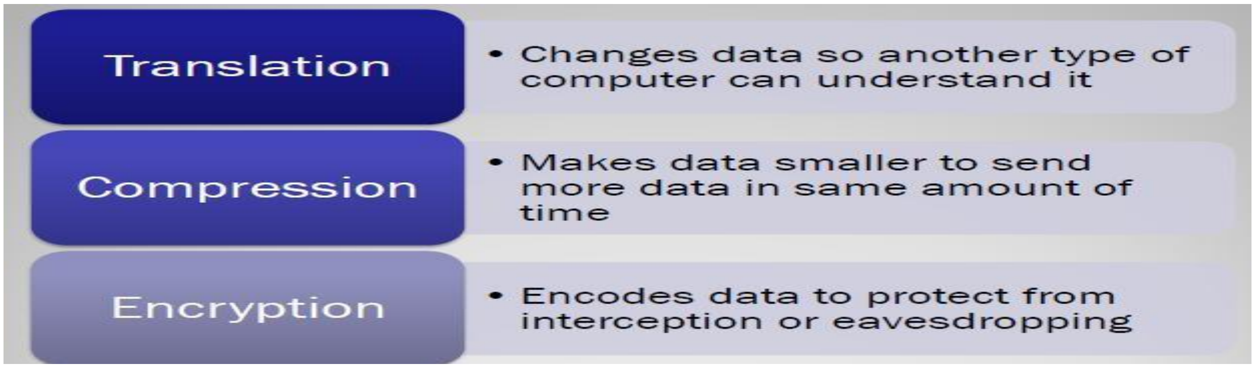
Session Layer

- Responsible for managing the dialog between networked devices
- Establishes, manages, and terminates connections
- Provides duplex, half-duplex, or simplex communications between devices

- Provides procedures for establishing checkpoints, adjournment, termination, and restart or recovery procedures

Presentation Layer

- Concerned with how data is presented to the network
- Handles three primary tasks: –Translation , –Compression , –Encryption



Translation

- Changes data so another type of computer can understand it

Compression

- Makes data smaller to send more data in same amount of time

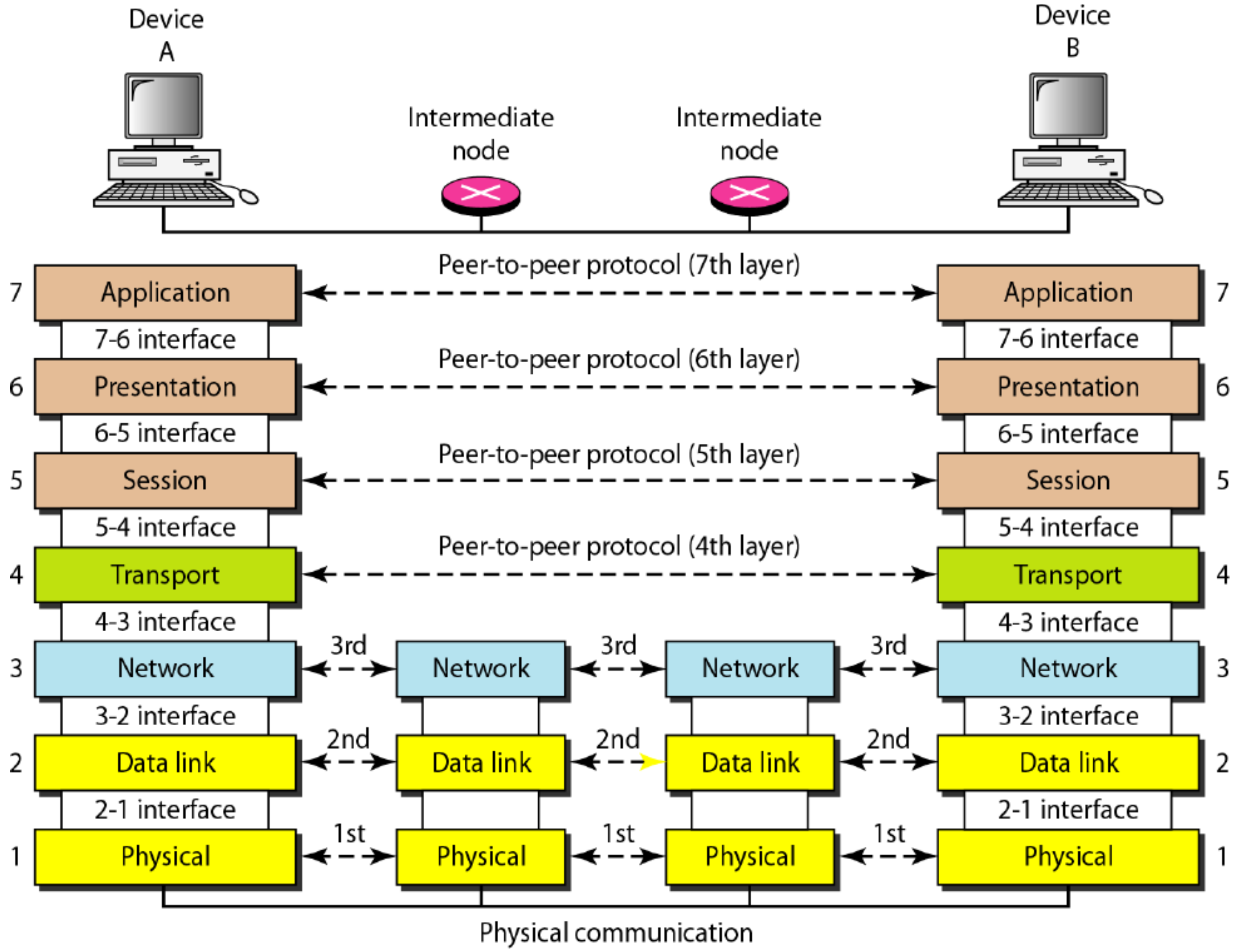
Encryption

- Encodes data to protect from interception or eavesdropping

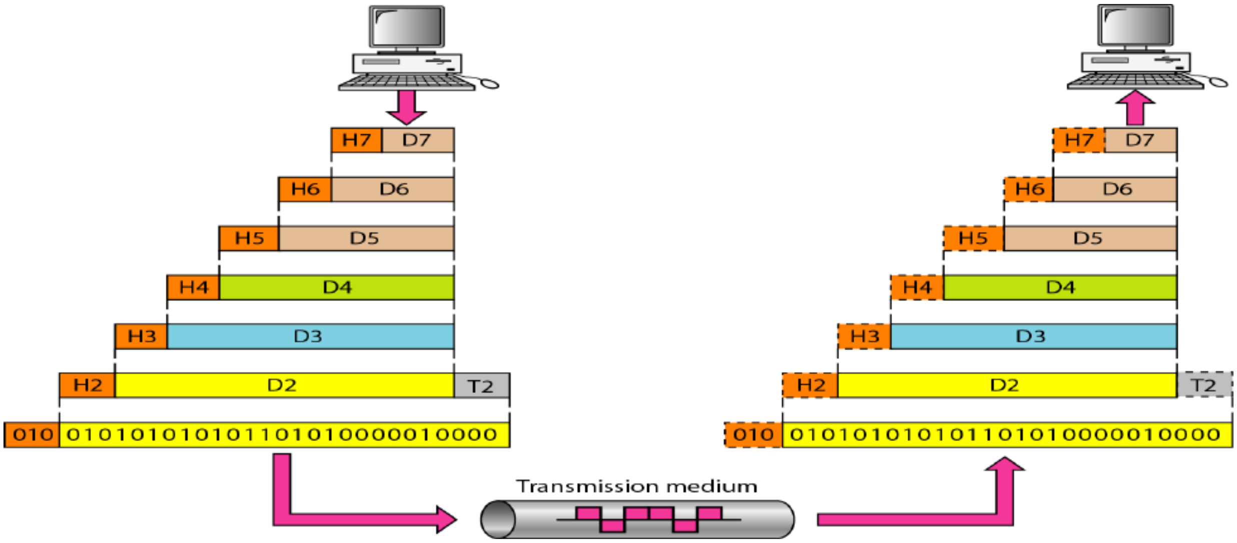
Application Layer

- Contains all services or protocols needed by application software or operating system to communicate on the network
- Examples
  - –Firefox web browser uses HTTP (Hyper-Text Transport Protocol)
  - –E-mail program may use POP3 (Post Office Protocol version 3) to read e-mails and SMTP (Simple Mail Transport Protocol) to send e-mails

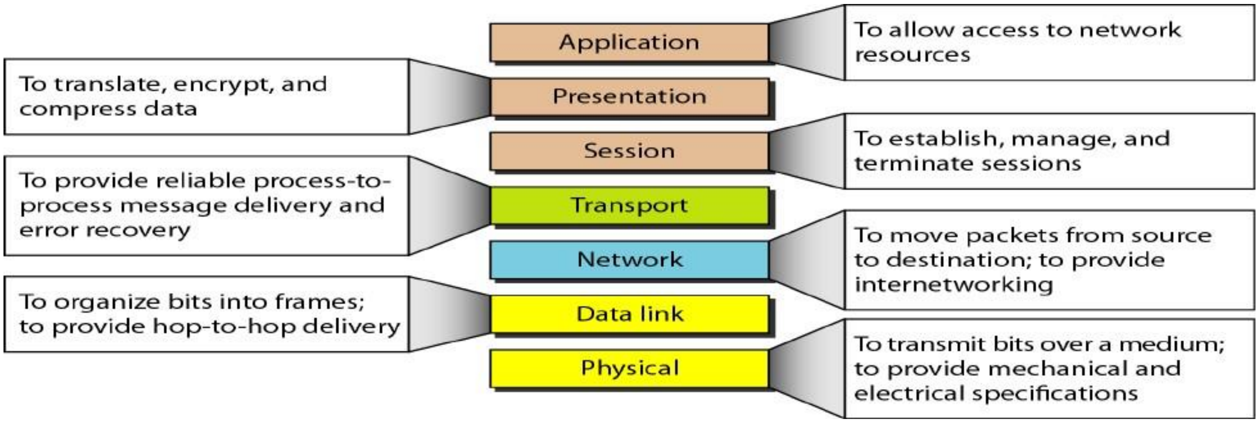
The interaction between layers in the OSI model



An exchange using the OSI model

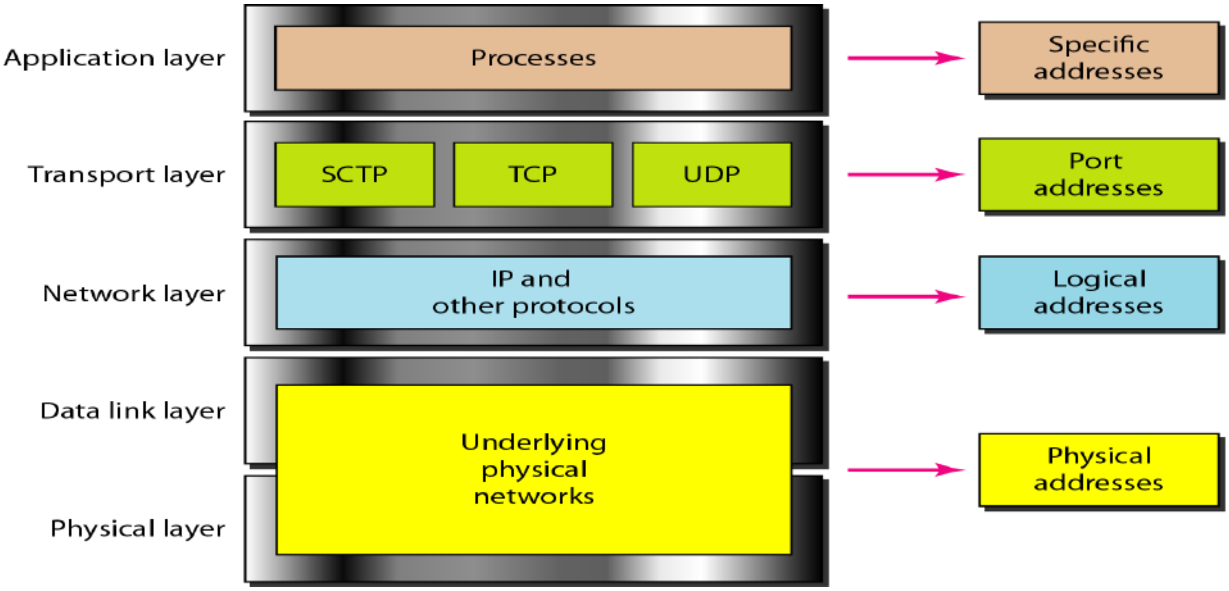


summery



Application
To allow access to network resources
Presentation
To translate, encrypt, and compress data
Session
To establish, manage, and terminate sessions
Transport
To provide reliable process-to-process message delivery and error recovery
Network
To move packets from source to destination; to provide internetworking
Data link
To organize bits into frames; to provide hop-to-hop delivery
Physical
To transmit bits over a medium; to provide mechanical and electrical specifications

<u>TCP/IP Model (Transmission Control Protocol/Internet Protocol)</u>



–A *protocol suite* is a large number of related protocols that work together to allow networked computers to communicate

Relationship of layers and addresses in TCP/IP

<u>Application Layer</u>

- Application layer protocols define the rules when implementing specific network applications
- Rely on the underlying layers to provide accurate and efficient data delivery
- Typical protocols:

○ FTP – File Transfer Protocol

□ For file transfer

○ Telnet – Remote terminal protocol

□ For remote login on any other computer on the network

○ SMTP – Simple Mail Transfer Protocol

□ For mail transfer

○ HTTP – Hypertext Transfer Protocol

□ For Web browsing

- Encompasses same functions as these OSI Model layers Application Presentation Session

Transport Layer

- TCP is a connection-oriented protocol
  - Does not mean it has a physical connection between sender and receiver
  - TCP provides the function to allow a connection virtually exists – also called virtual circuit
- UDP provides the functions:
  - Dividing a chunk of data into segments
  - Reassembly segments into the original chunk
  - Provide further the functions such as reordering and data resend
- Offering a reliable byte-stream delivery service
- Functions the same as the Transport layer in OSI
- Synchronize source and destination computers to set up the session between the respective

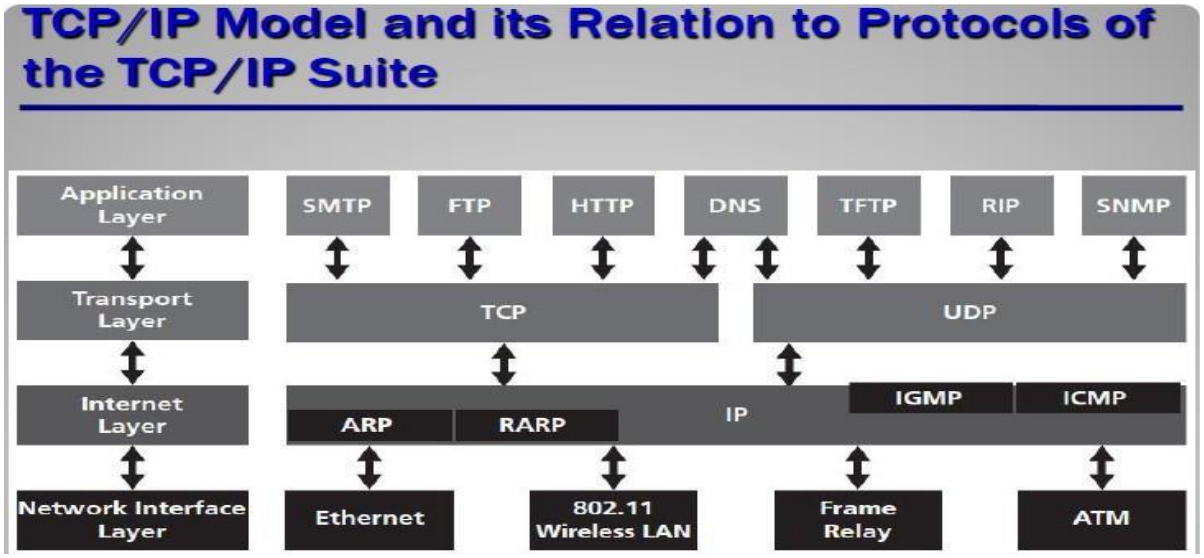
  computers

Internet Layer

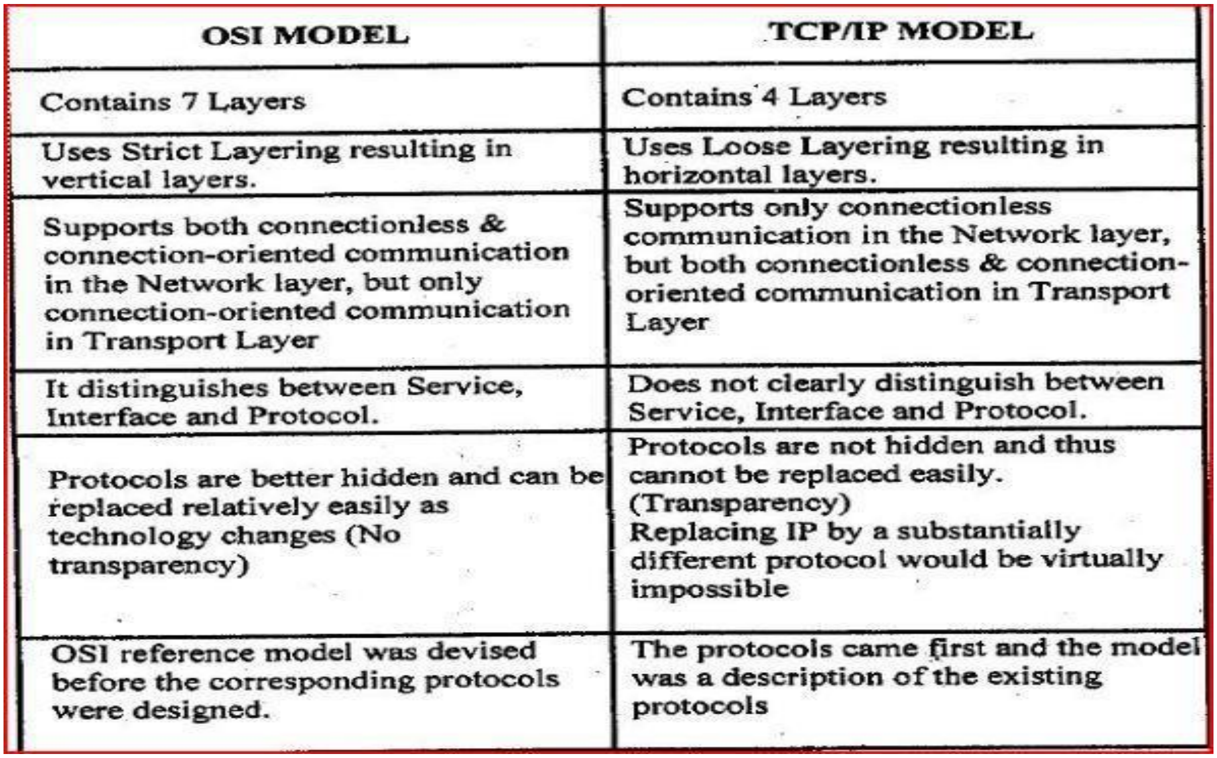
- The network layer, also called the internet layer, deals with packets and connects independent networks to transport the packets across network boundaries. The network layer protocols are the IP and the Internet Control Message Protocol (ICMP), which is used for error reporting.

Host-to-network layer

The **Host-to-network layer** is the lowest **layer** of the **TCP**/**IP** reference model. It combines the link **layer** and the physical **layer** of the ISO/OSI model. At this **layer**, data is transferred between adjacent **network** nodes in a WAN or between nodes on the same LAN.



TCP/IP Model and its Relation to Protocols of the TCP/IP Suite



OSI MODEL	TCP/IP MODEL
Contains 7 Layers	Contains 4 Layers
Uses Strict Layering resulting in vertical layers.	Uses Loose Layering resulting in horizontal layers.
Supports both connectionless & connection-oriented communication in the Network layer, but only connection-oriented communication in Transport Layer	Supports only connectionless communication in the Network layer, but both connectionless & connection-oriented communication in Transport Layer
It distinguishes between Service, Interface and Protocol.	Does not clearly distinguish between Service, Interface and Protocol.
Protocols are better hidden and can be replaced relatively easily as technology changes (No transparency)	Protocols are not hidden and thus cannot be replaced easily. (Transparency) Replacing IP by a substantially different protocol would be virtually impossible
OSI reference model was devised before the corresponding protocols were designed.	The protocols came first and the model was a description of the existing protocols

THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. Or maybe you researched a medical topic, booked a hotel reservation, chatted with a fellow Trekkie, or comparison-shopped for a car. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

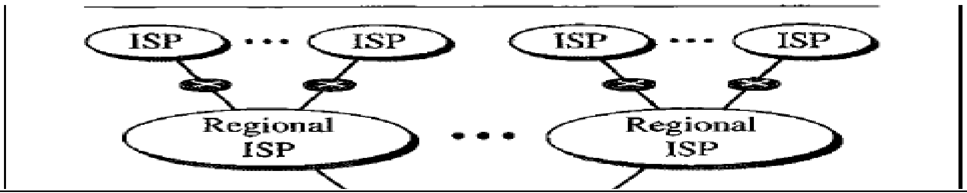
In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency (ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

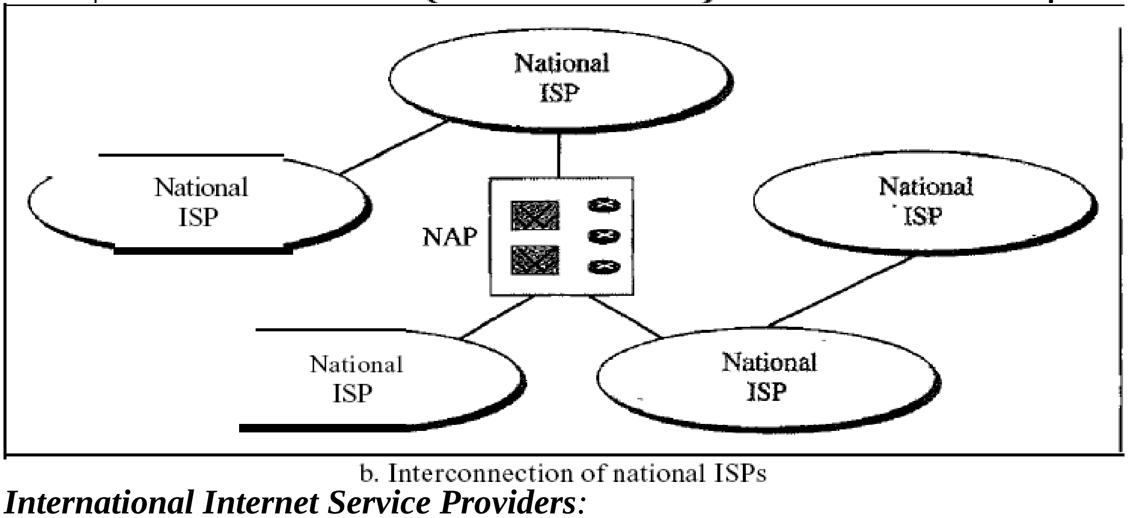
In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *inteiface message processor* (IMP). The IMPs, in tum, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Projec1.* Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end- to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (lP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCPIIP.

The Internet Today

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchical structure. It is made up of many wide- and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing-new networks are being added, existing networks are adding addresses, and networks of defunct companies are being removed. Today most end users who want Internet connection use the services of Internet service providers (lSPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure 1.13 shows a conceptual (not geographic) view of the Internet.





b. Interconnection of national ISPs

International Internet Service Providers:

At the top of the hierarchy are the international service providers that connect nations together.

National Internet Service Providers:

The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are SprintLink, PSINet, UUNet Technology, AGIS, and internet Mel. To provide connectivity between

the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called *peering points.* These normally operate at a high data rate (up to 600 Mbps).

Regional Internet Service Providers:

Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate. ***Local Internet Service Providers****:*

Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

Performance

Performance of a network pertains to the measure of service quality of a network as perceived by the user. There are different ways to measure the performance of a network, depending upon the nature and design of the network. The characteristics that measure the performance of a network are :

Bandwidth
Throughput
Latency (Delay)
Bandwidth – Delay Product
Jitter

BANDWIDTH

One of the most essential conditions of a website's performance is the amount of bandwidth allocated to the network. Bandwidth determines how rapidly the webserver is able to upload the requested information. While there are different factors to consider with respect to a site's performance, bandwidth is every now and again the restricting element.

Bandwidth is characterized as the measure of data or information that can be transmitted in a fixed measure of time. The term can be used in two different contexts with two distinctive estimating values. In the case of digital devices, the bandwidth is measured in bits per second(bps) or bytes per second. In the case of analogue devices, the bandwidth is measured in cycles per second, or Hertz (Hz).

Bandwidth is only one component of what an individual sees as the speed of a network. People frequently mistake bandwidth with internet speed in light of the fact that internet service providers (ISPs) tend to claim that they have a fast "40Mbps connection" in their advertising campaigns. True internet speed is actually the amount of data you receive every second and that has a lot to do with latency too.

"Bandwidth" means "Capacity" and "Speed" means "Transfer rate".

More bandwidth does not mean more speed. Let us take a case where we have double the width of the tap pipe, but the water rate is still the same as it was when the tap pipe was half the width. Hence, there will be no improvement in speed. When we consider WAN links, we mostly mean bandwidth but when we consider LAN, we mostly mean speed. This is on the grounds that we are generally constrained by expensive cable bandwidth over WAN rather than hardware and interface data transfer rates (or speed) over LAN.

Bandwidth in Hertz: It is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass. For example, let us consider the bandwidth of a subscriber telephone line as 4 kHz.

Bandwidth in Bits per Seconds: It refers to the number of bits per second that a channel, a link, or rather a network can transmit. For example, we can say the bandwidth of a Fast Ethernet network is a maximum of 100 Mbps, which means that the network can send 100 Mbps of data.

Note: There exists an explicit relationship between the bandwidth in hertz and the bandwidth in bits per second. An increase in bandwidth in hertz means an increase in bandwidth in bits per second. The relationship depends upon whether we have baseband transmission or transmission with modulation.

THROUGHPUT

Throughput is the number of messages successfully transmitted per unit time. It is controlled by available bandwidth, the available signal-to-noise ratio and hardware limitations. The maximum throughput of a network may be consequently higher than the actual throughput achieved in everyday consumption. The terms 'throughput' and 'bandwidth' are often thought of as the same, yet they are different. Bandwidth is the potential measurement of a link, whereas throughput is an actual measurement of how fast we can send data.

Throughput is measured by tabulating the amount of data transferred between multiple locations during a specific period of time, usually resulting in the unit of bits per second(bps), which has evolved to bytes per second(Bps), kilobytes per second(KBps), megabytes per second(MBps) and gigabytes per second(GBps).

Throughput may be affected by numerous factors, such as the hindrance of the underlying analogue physical medium, the available processing power of the system components, and end-user behaviour. When numerous protocol expenses are taken into account, the use rate of the transferred data can be significantly lower than the maximum achievable throughput.

Let us consider: A highway which has a capacity of moving, say, 200 vehicles at a time. But at a random time, someone notices only, say, 150 vehicles moving through it due to some congestion on the road. As a result, the capacity is likely to be 200 vehicles per unit time and the throughput is 150 vehicles at a time.

Example:

Input:A network with bandwidth of 10 Mbps can pass only an average of 12, 000 frames per minute where each frame carries an average of 10, 000 bits. What will be the throughput for this network?

Output: We can calculate the throughput as-

Throughput = (12, 000 x 10, 000) / 60 = 2 Mbps

The throughput is nearly equal to one-fifth of the bandwidth in this case.

Difference between Bandwidth and Throughput:

S.No.	Comparison	Bandwidth	Throughput
1.	Basic	Data capacity is travelled via a channel.	Practical measure of the amount of data actually transmitted through a channel.
2.	Measured in	Bits	Average rate is measured depending on bandwidth. It is measured in terms of bits transferred per second (bps).
3.	Concerned with	Transfer of data by some means.	Communication between two entities
4.	Relevance to layer	Physical layer property.	Work at any of the layers in the OSI model.
5.	Dependence	Not depend on the latency.	It depends on the latency.

6.	Definition	It refers to the maximum amount of the data that can be passed from one point to another.	It is considered as the actual measurement of the data that is being moved through the media at any particular time.
7.	Effect	It is not affected by physical obstruction because it is a theoretical unit to some extent.	It can be easily affected by change in interference, traffic in network, network devices, transmission errors and the host of other type.
8.	Real world Example(Water Tap Example).	It is the speed of tap at which water is coming out.	It is the total amount of water that comes out.

LATENCY

In a network, during the process of data communication, latency(also known as delay) is defined as the total time taken for a complete message to arrive at the destination, starting with the time when the first bit of the message is sent out from the source and ending with the time when the last bit of the message is delivered at the destination. The network connections where small delays occur are called "Low-Latency-Networks" and the network connections which suffer from long delays are known as "High-Latency-Networks".

High latency leads to the creation of bottlenecks in any network communication. It stops the data from taking full advantage of the network pipe and conclusively decreases the bandwidth of the communicating network. The effect of the latency on a network's bandwidth can be temporary or never-ending depending on the source of the delays. Latency is also known as a ping rate and is measured in milliseconds(ms).

In simpler terms: latency may be defined as the time required to successfully send a packet across a network.

It is measured in many ways like round trip, one way, etc.

It might be affected by any component in the chain which is utilized to vehiculate data, like workstations, WAN links, routers, LAN, servers and eventually may be limited for large networks, by the speed of light.

Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay

Propagation Time: It is the time required for a bit to travel from the source to the destination. Propagation time can be calculated as the ratio between the link length (distance) and the propagation speed over the communicating medium. For example, for an electric signal, propagation time is the time taken for the signal to travel through a wire.

Propagation time = Distance / Propagation speed

Example:

Input: What will be the propagation time when the distance between two points is
12, 000 km? Assuming the propagation speed to be 2.4 * 10^8 m/s in cable.

Output: We can calculate the propagation time as-
Propagation time = (12000 * 10000) / (2.4 * 10^8) = 50 ms

Transmission Time: Transmission time is a time based on how long it takes to send the signal down the transmission line. It consists of time costs for an EM signal to propagate from one side to the other, or costs like the training signals that are usually put on the front of a packet by the sender, which helps the receiver synchronize clocks. The transmission time of a message relies upon the size of the message and the bandwidth of the channel.

Transmission time = Message size / Bandwidth

Example:

Input: What will be the propagation time and the transmission time for a 2.5-kbyte message when the bandwidth of the network is 1 Gbps? Assuming the distance between sender and receiver is 12, 000 km and speed of light is 2.4 * 10^8 m/s.

Output: We can calculate the propagation and transmission time as-
Propagation time = (12000 * 10000) / (2.4 * 10^8) = 50 ms
Transmission time = (2560 * 8) / 10^9 = 0.020 ms

Note: Since the message is short and the bandwidth is high, the dominant factor is the propagation time and not the transmission time(which can be ignored).

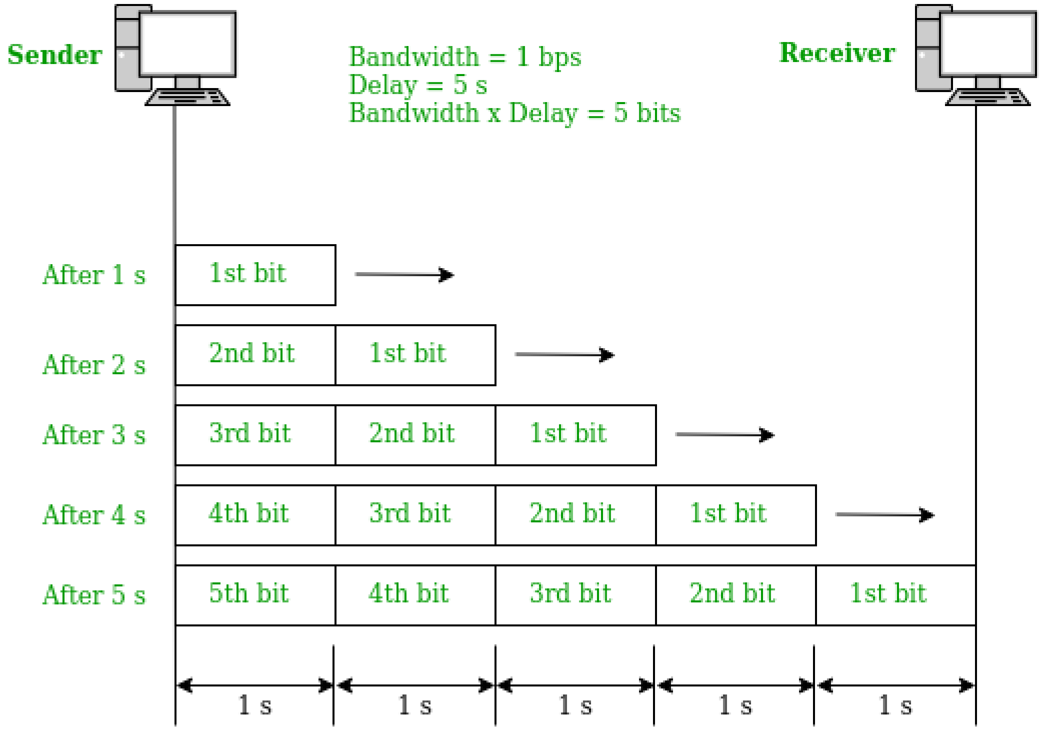
Queuing Time: Queuing time is a time based on how long the packet has to sit around in the router. Quite frequently the wire is busy, so we are not able to transmit a packet immediately. The queuing time is usually not a fixed factor, hence it changes with the load thrust in the network. In cases like these, the packet sits waiting, ready to go, in a queue. These delays are predominantly characterized by the measure of traffic on the system. The more the traffic, the more likely a packet is stuck in the queue, just sitting in the memory, waiting.

Processing Delay: Processing delay is the delay based on how long it takes the router to figure out where to send the packet. As soon as the router finds it out, it will queue the packet for transmission. These costs are predominantly based on the complexity of the protocol. The router must decipher enough of the packet to make sense of which queue to put the packet in. Typically the lower-level layers of the stack have simpler protocols. If a router does not know which physical port to send the packet to, it will send it to all the ports, queuing the packet in many queues immediately. Differently, at a higher level, like in IP protocols, the processing may include making an ARP request to find out the physical address of the destination before queuing the packet for transmission. This situation may also be considered as a processing delay.

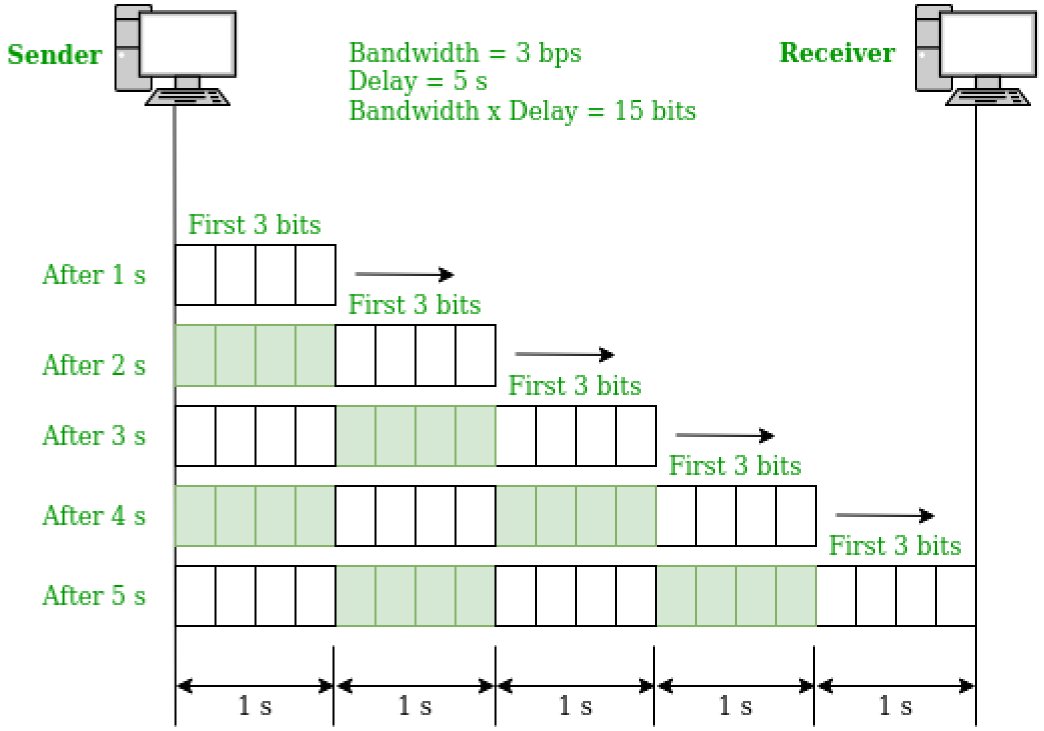
BANDWIDTH – DELAY PRODUCT

Bandwidth and delay are two performance measurements of a link. However, what is significant in data communications is the product of the two, the bandwidth-delay product.
Let us take two hypothetical cases as examples.

Case 1: Assume a link is of bandwidth 1bps and the delay of the link is 5s. Let us find the bandwidth-delay product in this case. From the image, we can say that this product 1 x 5 is the maximum number of bits that can fill the link. There can be close to 5 bits at any time on the link.



Case 2: Assume a link is of bandwidth 3bps. From the image, we can say that there can be a maximum of 3 x 5 = 15 bits on the line. The reason is that, at each second, there are 3 bits on the line and the duration of each bit is 0.33s.



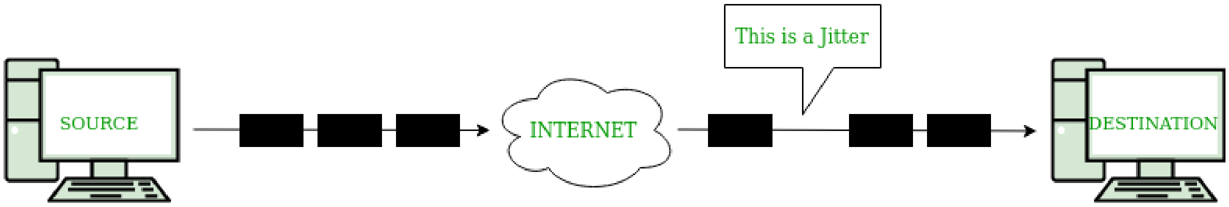
For both examples, the product of bandwidth and delay is the number of bits that can fill the link. This estimation is significant in the event that we have to send data in bursts and wait for the acknowledgement of each burst before sending the following one. To utilize the maximum ability of the link, we have to make the size of our burst twice the product of bandwidth and delay. Also, we need to fill up the full-duplex channel. The sender ought to send a burst of data of (2*bandwidth*delay) bits. The sender at that point waits for the receiver's acknowledgement for part of the burst before sending another burst. The amount: 2*bandwidth*delay is the number of bits that can be in transition at any time.

JITTER

Jitter is another performance issue related to delay. In technical terms, jitter is a "packet delay variance". It can simply mean that jitter is considered as a problem when different packets of data face different delays in a network and the data at the receiver application is time-sensitive, i.e. audio or video data. Jitter is measured in milliseconds(ms). It is defined as an interference in the normal order of sending data packets. For example: if the delay for the first packet is 10 ms, for the second is 35 ms, and for the third is 50 ms, then the real-time destination application that uses the packets experiences jitter.
Simply, jitter is any deviation in, or displacement of, the signal pulses in a high-frequency digital signal. The deviation can be in connection with the amplitude, the width of the signal pulse or the phase timing. The major causes of jitter are electromagnetic interference(EMI) and crosstalk between signals. Jitter can lead to flickering of a display screen, affects the capability of a processor in a desktop or server to proceed as expected, introducing clicks or other undesired impacts in audio signals, and loss of transmitted data between network devices.

Jitter is negative and causes network congestion and packet loss.

- Congestion is like a traffic jam on the highway. In a traffic jam, cars cannot move forward at a reasonable speed. Like the traffic jam, in congestion, all the packets come to a junction at the same time. Nothing can get loaded.
- The second negative effect is packet loss. When packets arrive at unexpected intervals, the receiving system is not able to process the information, which leads to missing information also called "packet loss". This has negative effects on video viewing. If a video becomes pixelated and is skipping, the network is experiencing jitter. The result of the jitter is packet loss. When you are playing a game online, the effect of packet loss can be that a player begins moving around on the screen randomly. Even worse, the game goes from one scene to the next, skipping over part of the gameplay.



In the above image, it can be noticed that the time it takes for packets to be sent is not the same as the time in which he will arrive at the receiver side. One of the packets faces an unexpected delay on its way and is received after the expected time. This is jitter.

A jitter buffer can reduce the effects of jitter, either in a network, on a router or switch, or on a computer. The system at the destination receiving the network packets usually receives them from the buffer and not from the source system directly. Each packet is fed out of the buffer at a regular rate. Another approach to diminish jitter in case of multiple paths for traffic is to selectively route traffic along the most stable paths or to always pick the path that can come closest to the targeted packet delivery rate.

High-Speed Networks

The seeming continual increase in bandwidth causes network designers to start thinking about what happens in the limit or, stated another way, what is the impact on network design of having infinite bandwidth available.

Although high-speed networks bring a dramatic change in the bandwidth available to applications, in many respects their impact on how we think about networking comes in what does *not* change as bandwidth increases: the speed of light. To quote Scotty from *Star Trek*, “Ye cannae change the laws of physics.” In other words, “high speed” does not mean that latency improves at the same rate as bandwidth; the transcontinental RTT of a 1-Gbps link is the same 100 ms as it is for a 1-Mbps link.

To appreciate the significance of ever-increasing bandwidth in the face of fixed latency, consider what is required to transmit a 1-MB file over a 1-Mbps network versus over a 1-Gbps network, both of which have an RTT of 100 ms. In the case of the 1-Mbps network, it takes 80 round-trip times to transmit the file; during each RTT, 1.25% of the file is sent. In contrast, the same 1-MB file doesn’t even come close to filling 1 RTT’s worth of the 1-Gbps link, which has a delay × bandwidth product of 12.5 MB.

Figure 19 illustrates the difference between the two networks. In effect, the 1-MB file looks like a stream of data that needs to be transmitted across a 1-Mbps network, while it looks like a single packet on a 1-Gbps network. To help drive this point home, consider that a 1-MB file is to a 1-Gbps network what a 1-KB *packet* is to a 1-Mbps network.

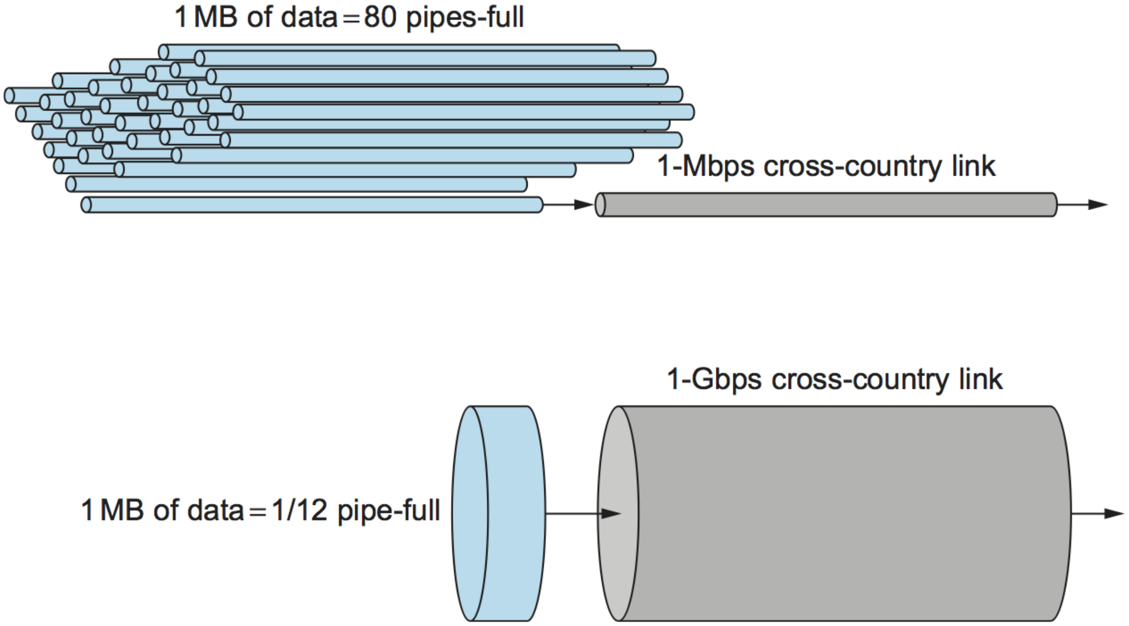

1 MB of data = 80 pipes-full
1-Mbps cross-country link
1-Gbps cross-country link
1 MB of data = 1/12 pipe-full

Figure 19. *Relationship between bandwidth and latency. A 1-MB file would fill the 1-Mbps link 80 times but only fill 1/12th of a 1-Gbps link.*

Another way to think about the situation is that more data can be transmitted during each RTT on a high-speed network, so much so that a single RTT becomes a significant amount of time. Thus, while you wouldn’t think twice about the difference between a file transfer taking 101 RTTs rather than 100 RTTs (a relative difference of only 1%), suddenly the difference between 1 RTT and 2 RTTs is significant—a 100% increase. In other words, latency, rather than throughput, starts to dominate our thinking about network design.

Perhaps the best way to understand the relationship between throughput and latency is to return to basics. The effective end-to-end throughput that can be achieved over a network is given by the simple relationship

Throughput = TransferSize / TransferTime

where TransferTime includes not only the elements of one-way identified earlier in this section, but also any additional time spent requesting or setting up the transfer. Generally, we represent this relationship as

TransferTime = RTT + 1/Bandwidth x TransferSize

We use in this calculation to account for a request message being sent across the network and the data being sent back. For example, consider a situation where a user wants to fetch a 1-MB file across a 1-Gbps with a round-trip time of 100 ms. This includes both the transmit time for 1 MB (1 / 1 Gbps × 1 MB = 8 ms) and the 100-ms RTT, for a total transfer time of 108 ms. This means that the effective throughput will be

1 MB / 108 ms = 74.1 Mbps

not 1 Gbps. Clearly, transferring a larger amount of data will help improve the effective throughput, where in the limit an infinitely large transfer size will cause the effective throughput to approach the network bandwidth. On the other hand, having to endure more than 1 RTT—for example, to retransmit missing packets—will hurt the effective throughput for any transfer of finite size and will be most noticeable for small transfers.

Application Requirements

The discussion in this section has taken a network-centric view of performance; that is, we have talked in terms of what a given link or channel will support. The unstated assumption has been that application programs have simple needs—they want as much bandwidth as the network can provide. This is certainly true of the aforementioned digital library program that is retrieving a 250-MB image; the more bandwidth that is available, the faster the program will be able to return the image to the user.

However, some applications are able to state an upper limit on how much bandwidth they need. Video applications are a prime example. Suppose one wants to stream a video that is one quarter the size of a standard TV screen; that is, it has a resolution of 352 by 240 pixels. If each pixel is represented by 24 bits of information, as would be the case for 24-bit color, then the size of each frame would be (352 × 240 × 24) / 8 = 247.5 KB If the application needs to support a frame rate of 30 frames per second, then it might request a throughput rate of 75 Mbps. The ability of the network to provide more bandwidth is of no interest to such an application because it has only so much data to transmit in a given period of time.

Unfortunately, the situation is not as simple as this example suggests. Because the difference between any two adjacent frames in a video stream is often small, it is possible to compress the video by transmitting only the differences between adjacent frames. Each frame can also be compressed because not all the detail in a picture is readily perceived by a human eye. The compressed video does not flow at a constant rate, but varies with time according to factors such as the amount of action and detail in the picture and the compression algorithm being used. Therefore, it is possible to say what the average bandwidth requirement will be, but the instantaneous rate may be more or less.

The key issue is the time interval over which the average is computed. Suppose that this example video application can be compressed down to the point that it needs only 2 Mbps, on average. If it transmits 1 megabit in a 1-second interval and 3 megabits in the following 1-second interval, then over the 2-second interval it is transmitting at an average rate of 2 Mbps; however, this will be of little consolation to a channel that was engineered to support no more than 2 megabits in any one second. Clearly, just knowing the average bandwidth needs of an application will not always suffice.

Generally, however, it is possible to put an upper bound on how large a burst an application like this is likely to transmit. A burst might be described by some peak rate that is maintained for some period of time. Alternatively, it could be described as the number of bytes that can be sent at the peak rate before reverting to the average rate or some lower rate. If this peak rate is higher than the available channel capacity, then the excess data will have to be buffered somewhere, to be transmitted later. Knowing how big of a burst might be sent allows the network designer to allocate sufficient buffer capacity to hold the burst.

Analogous to the way an application’s bandwidth needs can be something other than “all it can get,” an application’s delay requirements may be more complex than simply “as little delay as possible.” In the case of delay, it sometimes doesn’t matter so much whether the one-way latency of the network is 100 ms or 500 ms as how much the latency varies from packet to packet. The variation in latency is called *jitter*.

Consider the situation in which the source sends a packet once every 33 ms, as would be the case for a video application transmitting frames 30 times a second. If the packets arrive at the destination spaced out exactly 33 ms apart, then we can deduce that the delay experienced by each packet in the network was exactly the same. If the spacing between when packets arrive at the destination—sometimes called the *inter-packet gap*—is variable, however, then the delay experienced by the sequence of packets must have also been variable, and the network is said to have introduced jitter into the packet stream, as shown in Figure 20. Such variation is generally not introduced in a single physical link, but it can happen when packets experience different queuing delays in a multihop packet-switched network. This queuing delay corresponds to the component of latency defined earlier in this section, which varies with time.

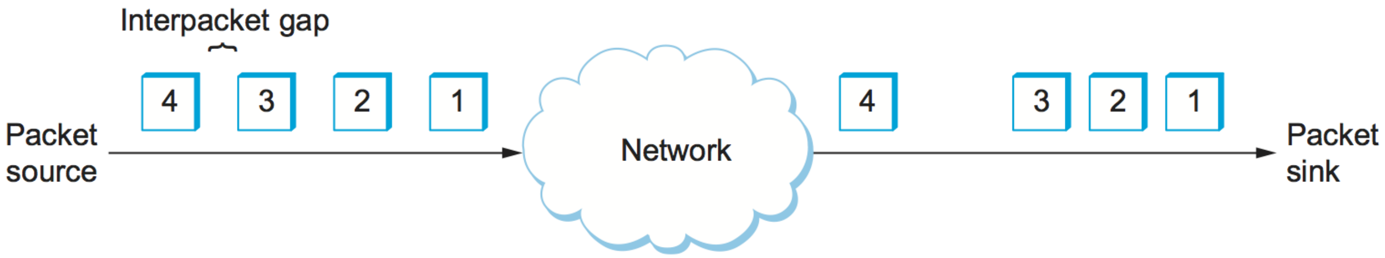


Figure 20. *Network-induced jitter.*ℑ

To understand the relevance of jitter, suppose that the packets being transmitted over the network contain video frames, and in order to display these frames on the screen the receiver needs to receive a new one every 33 ms. If a frame arrives early, then it can simply be saved by the receiver until it is time to display it. Unfortunately, if a frame arrives late, then the receiver will not have the frame it needs in time to update the screen, and the video quality will suffer; it will not be smooth. Note that it is not necessary to eliminate jitter, only to know how bad it is. The reason for this is that if the receiver knows the upper and lower bounds on the latency that a packet can experience, it can delay the time at which it starts playing back the video (i.e., displays the first frame) long enough to ensure that in the future it will always have a frame to display when it needs it. The receiver delays the frame, effectively smoothing out the jitter, by storing it in a buffer.