Computer Networks

Dr. Gaurav Varshney

Reference: Computer Networking, Kurose Ross, Sixth Edition, Pearson

What is Internet? Let us discuss an abstraction

- It is a internetwork that connects billions of computing devices around the world.
- The computing devices that it connects are knows as hosts or end systems [your computer, a server]
- End systems [Clients / Servers] are connected via communication links and packet switches.
- The data travels in the form of packets [common jargon].
  - The packets contain data as well as metadata [headers]
  - The metadata help intermediate systems to perform various functionalities to make the data reach successfully to the other end systems.
- Packet Switches
  - Routers
  - Link Layer Switches

PACKET SWITCHING RESEARCH IN PARALLEL

JCR at IPTO USA "Man Computer Symbiosis"

Paul Baran 1964 at RAND "Survivable Networks"

Donald Davies 1964 at NPL England "Packet Switching"

ARPANET

JCR Licklider and Roberts leads the Computer Science program at ARPA in 1967

First packet travels between UCLA and Stanford Research Institute in 1969

Interconnecting Networks

Standalone Packet Switching Networks [ALOHANET, DARPA's Packet Satellite Network 1970's

Vinton Cerf, Robert Kahn at DARPA, Internetworking 1974

Official Deployment of TCP/IP in 1983 for ARPANET

World Wide Web in 1990's by Tim Berners Lee

1995 Netscape Browser

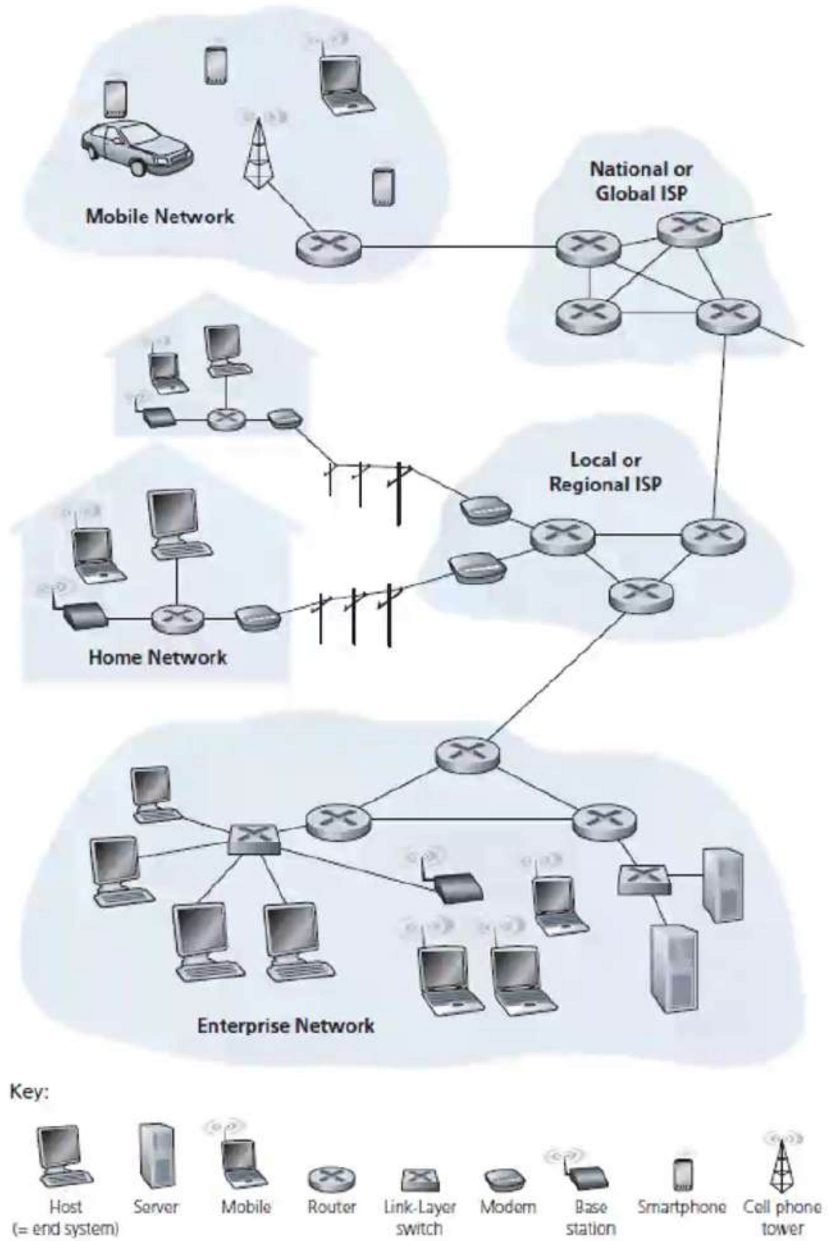
Dr. Gaurav Varshney IIT Jammu

Historical Timeline

- Sputnik Scare 1957
- Galactic Network MIT and ARPA in 1962
- NPL Packet Switching 1965
- ARPAnet UCLA to Stanford node to node communication Oct 29 1969
- ALOHAnet 1971.
- TCP/IP Vinton Cerf and Robert Kahn "all mini networks can communicate with each other" 1973
- Tim Berners Lee World Wide Web 1991
- MOSAIC(Netscape) browser in 1992

What is Internet?

- Packet Switching
  - In broad you can consider it as a cargo shipment
    - The trucks start from source and reach destination
    - Trucks use sign boards to decide the route
    - Trucks for same destination may take different routes
- End systems connect to network via ISPs
  - Internet Service Providers
  - Broadband
- Protocols
  - All devices follow a set of protocols.
    - Protocols[TCP/IP] provide a well defined procedure for communication between end systems/switches
  - It is Internet Engineering Task Force [IETF] which Develo
    and maintain standards for communication Generally specification
    of protocols via documents Known as RFCs or Request For Comments



What is a Protocol ?

- All communication on Internet between two end systems is governed by a Protocol.
- A protocol defines structure/format of message that are communicated over network.
- It also defines how the message should be interpreted and what actions should be performed by end systems on receipt of it.

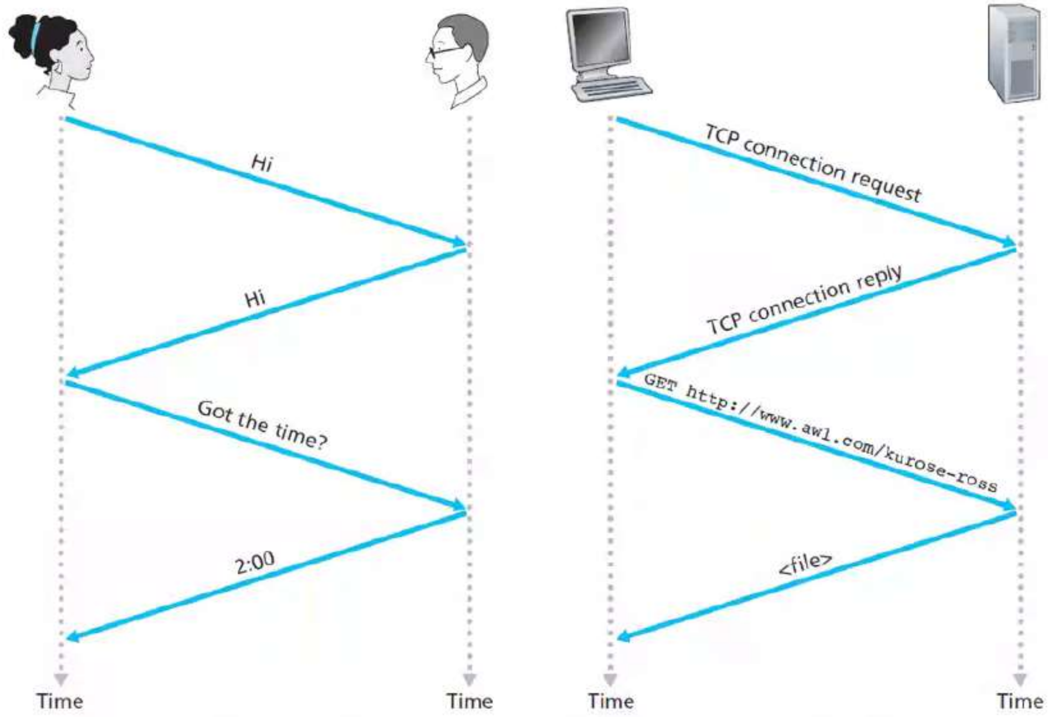


Figure 1.2 ♦ A human protocol and a computer network protocol

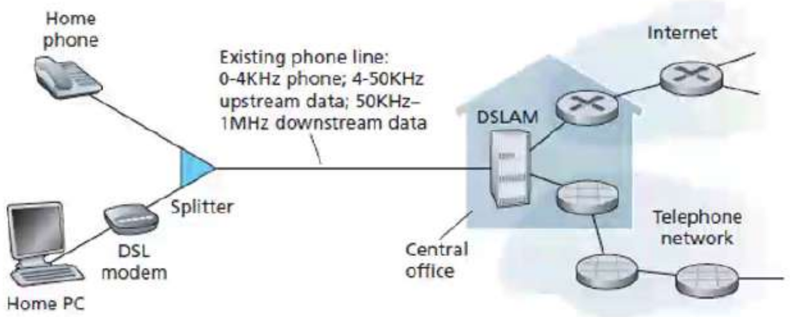


Figure 1.5 ♦ DSL Internet access

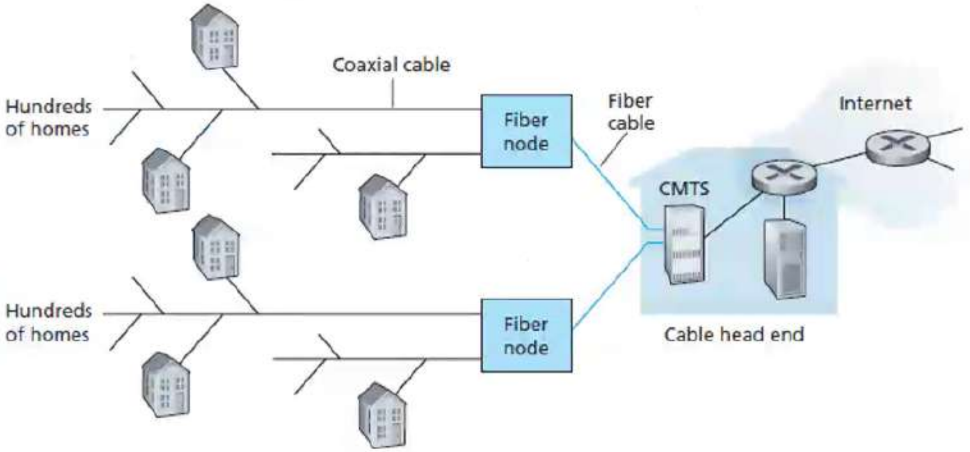


Figure 1.6 ♦ A hybrid fiber-coaxial access network

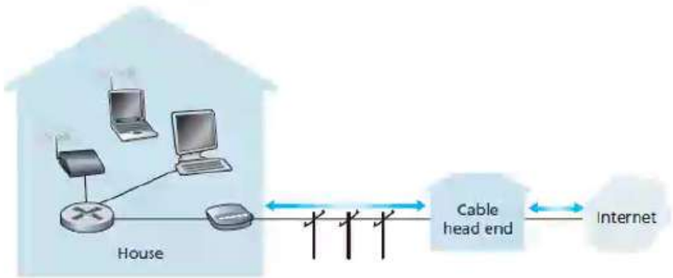


Figure 1.9 ♦ A typical home network

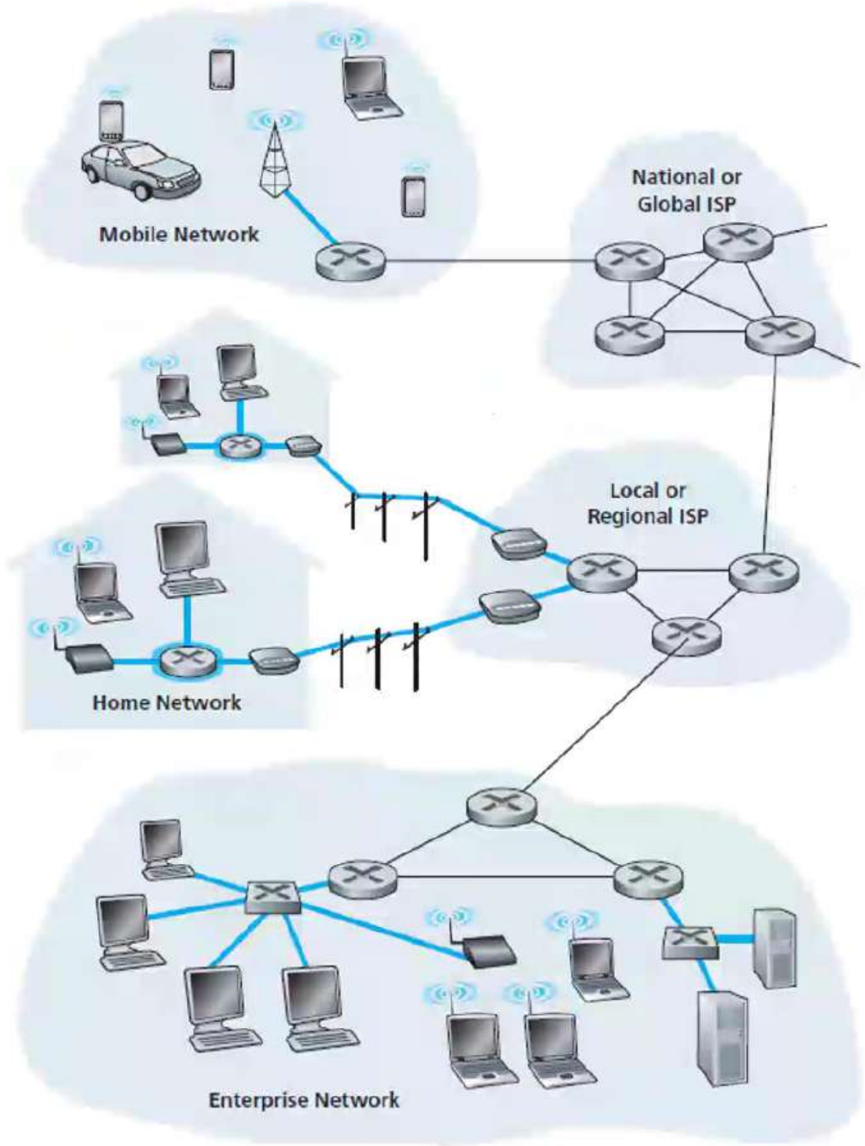


Figure 1.4 ♦ Access networks

Internet

Internetworking: Packet Switching is the core

- When end systems generate the data to be sent to another end system they are divided into small chunks which are sent over the communication links. We call them as packets in general.
- Packets are switched between communication links so that they can travel from source to destination
- Packet switches such as Routers [Layer 3] and Link Layer [Layer 2] switches does the necessary task of switching between end systems.
- Routers perform Store and Forward transmission.
  - Packet is stored some processing is performed to decide next link where packet needs to be transmitted

Store and Forward Switching

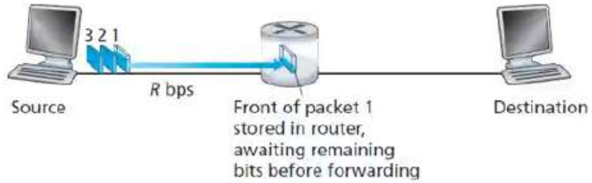


Figure 1.11 ♦ Store-and-forward packet switching

- If the data is L bits and the transmission rate of the channel is R bits/sec then time takes for putting L bits on the channel will be L/R sec. This is transmission time.
- When it arrives at a Router it is first received completely and then forwarded on the outgoing link [Store and Forward]
  - It is received and stored because the data [packet] requires certain processing. Deciding on which outgoing link packet has to be transmitted next.
  - Now router again needs to put L bits on to the channel taking L/R sec.
  - So the destination receives the packet at time 2L/R
  - If there are N routers in between then the destination will receive the packet at NL/R time.
  - Packets wait in the outgoing buffer on the router outgoing link and if it gets full other packets destined to the same outgoing link are dropped.

Store and Forward Switching

- Forwarding Tables in Routers
  - How we build them ?
    - Routing Protocols

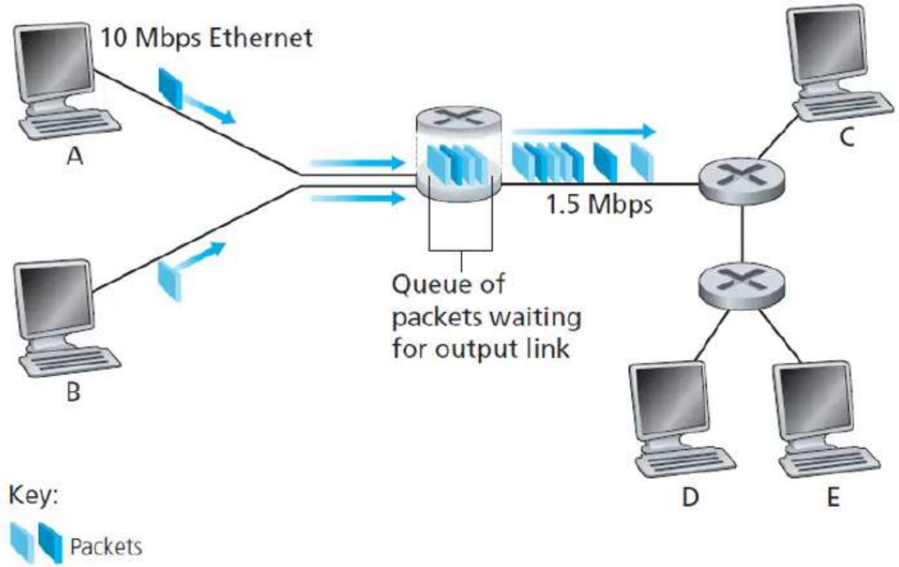
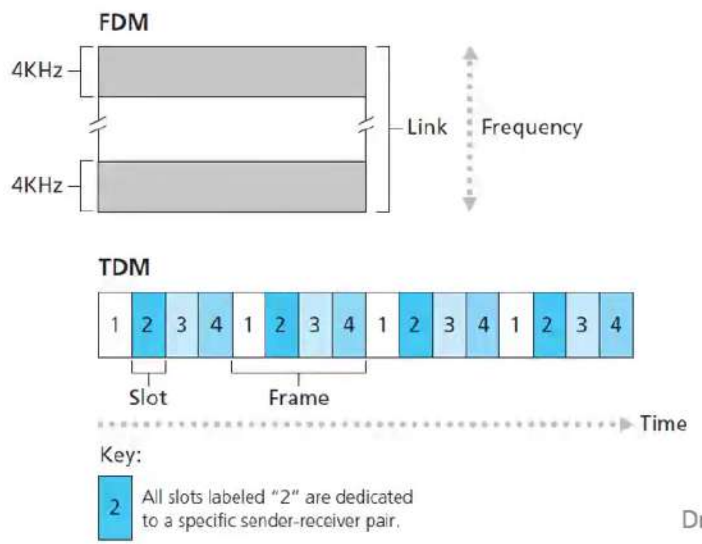


Figure 1.12 ♦ Packet switching

Circuit Switching Vs Packet Switching

- In Circuit Switching a dedicated path is reserved between two end systems.
  - The routers and switches along the path remembers the circuit and passes the data along a predefined path that spans across multiple intermediate devices.
  - The resources are reserved exclusively for a specific connection till the time connection is not terminated.
  - The end systems share resources: Time Division Multiplexing, Frequency Division Multiplexing



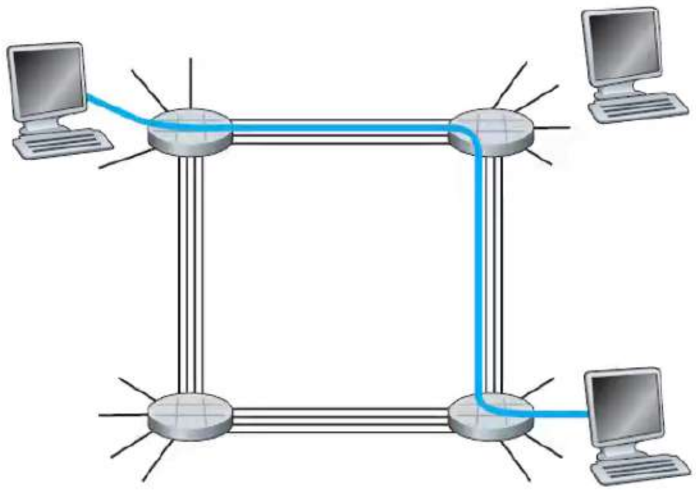
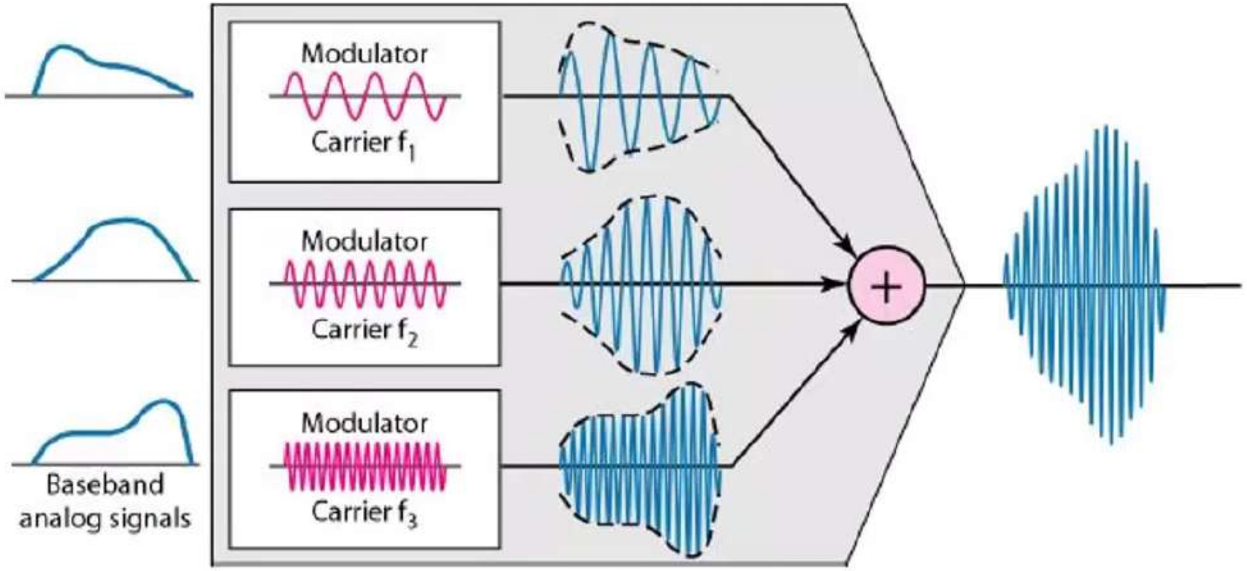


Figure 1.13 ♦ A simple circuit-switched network consisting of four switches and four links



https://datacommandnet.blogspot.com/p/multiplexing.html

Circuit Switching Vs Packet Switching

- Let us consider how long it takes to send a file of 640,000 bits from Host A to Host B over a circuit-switched network.
  - Suppose that all links in the network use TDM with 24 slots and have a bit rate of 1.536 Mbps. Also suppose that it takes 500 msec to establish an end-to-end circuit before Host A can begin to transmit the file. How long does it take to send the file?
  - Each circuit has a transmission rate of (1.536 Mbps)/24 = 64 kbps
  - so it takes (640,000 bits)/(64 kbps) = 10 seconds to transmit the file.
  - To this 10 seconds we add the circuit establishment time, giving 10.5 seconds to send the file.

Nodal Delays in Packet Switching

- Let us consider all the delays
  - Processing Delays at Intermediate Routers
  - Queuing Delays
  - Transmission Delay L/R
  - Propagation Delay D/S

  - **Total nodal delay:** Processing Delay + Queuing Delay + Transmission Delay + Propagation Delay
  - **Throughput** if a file of size F bits is received by a receiver in T seconds then the instantaneous throughput is F/T bits/sec

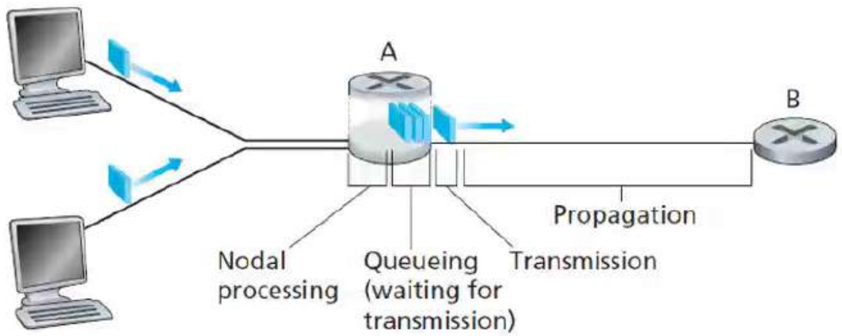
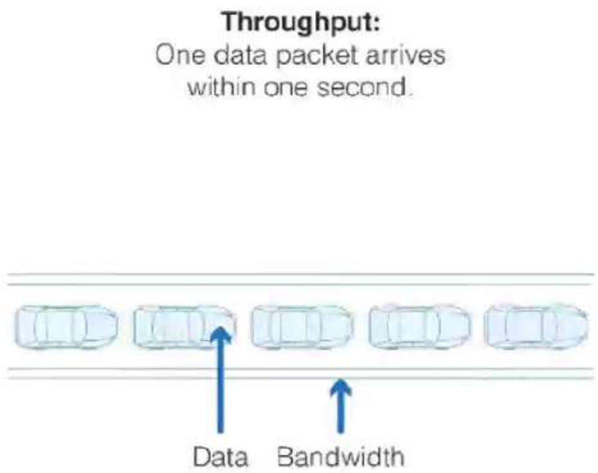


Figure 1.16 ♦ The nodal delay at router A



Q19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2= 2 Mbps, and R3
= 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?

b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B

c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Q2. Equation 1.1 gives a formula for the end-to-end delay of sending one packet of length L over N links of transmission rate R. Generalize this formula for sending P such packets back-to-back over the N links.

Q19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2= 2 Mbps, and R3 = 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?

b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B

c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Solution. A —-R1(500kbps)—-X—-—-R2(2Mbps)—-X—-—-R3(1Mbps)—— B

a. This is the concept of bottleneck link. Overall throughput will be the least bandwidth in the path between sender and receiver. So, throughput = 500 kbps

b. File size = 4*10^6 bytes, throughput = 500 kbps=> [4*(10^6)*8]/[500*10^3] = 64 seconds

c. **A —-R1(500kbps)—-X—-—-R2(100kbps)—-X—-—-R3(1Mbps)—— B**

Now the bottleneck link is R2 (100kbps). Repeat above calculation with R2 = 100kbps instead of R1 = 500kbps.

Q2. Equation 1.1 gives a formula for the end-to-end delay of sending one packet of length L over N links of transmission rate R. Generalize this formula for sending P such packets back-to-back over the N links.

Q19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2= 2 Mbps, and R3
= 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?
b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B
c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Solution. **A --R1(500kbps)--X---R2(2Mbps)--X---R3(1Mbps)-- B**

a. This is the concept of bottleneck link. Overall throughput will be the least bandwidth in the path between sender and receiver. So, throughput = 500 kbps

b. File size = 4*10^6 bytes, throughput = 500 kbps=> [4*(10^6)*8]/[500*10^3] = 64 seconds

c. **A --R1(500kbps)--X---R2(100kbps)--X---R3(1Mbps)-- B**

Now the bottleneck link is R2 (100kbps). Repeat above calculation with R2 = 100kbps instead of R1 = 500kbps.

Q2. Equation 1.1 gives a formula for the end-to-end delay of sending one packet of length L over N links of transmission rate R. Generalize this formula for sending P such packets back-to-back over the N links.

Solution. Transmission time is the time taken by a node to load the data on the channel or link and that is L/R for one node.

For given N links:-

1st packet will reach to destination at time N*(L/R)

The 2nd packet has to wait at sender, until the 1st packet will be transmitted completely (L/R).

So, after reaching the 1st packet at receiver (N*(L/R)), after L/R time 2nd packet will reach to the receiver.

At time N*(L/R) + L/R, the second packet has reached the receiver.

Continuing with this logic, after reaching 1st packet at receiver, after each L/R time next packet will be reaching at receiver.

So, the time required by all the packets to reach the destination is: =

= 1st packet time + 2nd packet time + 3rd packet time + 4th packet time + ... + (P-1)th packet time

N*(L/R) + (L/R) + (L/R) + (L/R) ....+ (P-1)th time (L/R)

=N*(L/R) + (P-1)*(L/R) = (N+P-1)*(L/R)

Q19. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 500 kbps, R2= 2 Mbps, and R3 = 1 Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?

b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B

c. Repeat (a) and (b), but now with R2 reduced to 100 kbps.

Solution. **A—-R1(500kbps)—-X—-—R2(2Mbps)—-X—-—R3(1Mbps)—— B**

a. This is the concept of bottleneck link. Overall throughput will be the least bandwidth in the path between sender and receiver. So, throughput = 500 kbps

b. File size = 4*10^6 bytes, throughput = 500 kbps=> [4*(10^6)*8]/[500*10^3] = 64 seconds

c. **A—-R1(500kbps)—-X—-—R2(100kbps)—-X—-—R3(1Mbps)—— B**

Now the bottleneck link is R2 (100kbps). Repeat above calculation with R2 = 100kbps instead of R1 = 500kbps.

Q2. Equation 1.1 gives a formula for the end-to-end delay of sending one packet of length L over N links of transmission rate R. Generalize this formula for sending P such packets back-to-back over the N links.

Solution. Transmission time is the time taken by a node to load the data on the channel or link and that is L/R for one node.

For given N links:-

1st packet will reach to destination at time N*(L/R)

The 2nd packet has to wait at sender, until the 1st packet will be transmitted completely (L/R).

So, after reaching the 1st packet at receiver (N*(L/R)), after L/R time 2nd packet will reach to the receiver.

At time N*(L/R) + L/R, the second packet has reached the receiver.

Continuing with this logic, after reaching 1st packet at receiver, after each L/R time next packet will be reaching at receiver.

So, the time required by all the packets to reach the destination is: =

= 1st packet time + 2nd packet time + 3rd packet time + 4th packet time + ... + (P-1)th packet time

N*(L/R) + (L/R) + (L/R) + (L/R) ....+ (P-1)th time (L/R)

=N*(L/R) + (P-1)*(L/R) = (N+P-1)*(L/R)

0:39 / 0:39

TCP/IP and OSI Reference Model

- Individual layer of the protocol stack perform a set of functions
- The layer is implemented as software or hardware using a set of protocols.

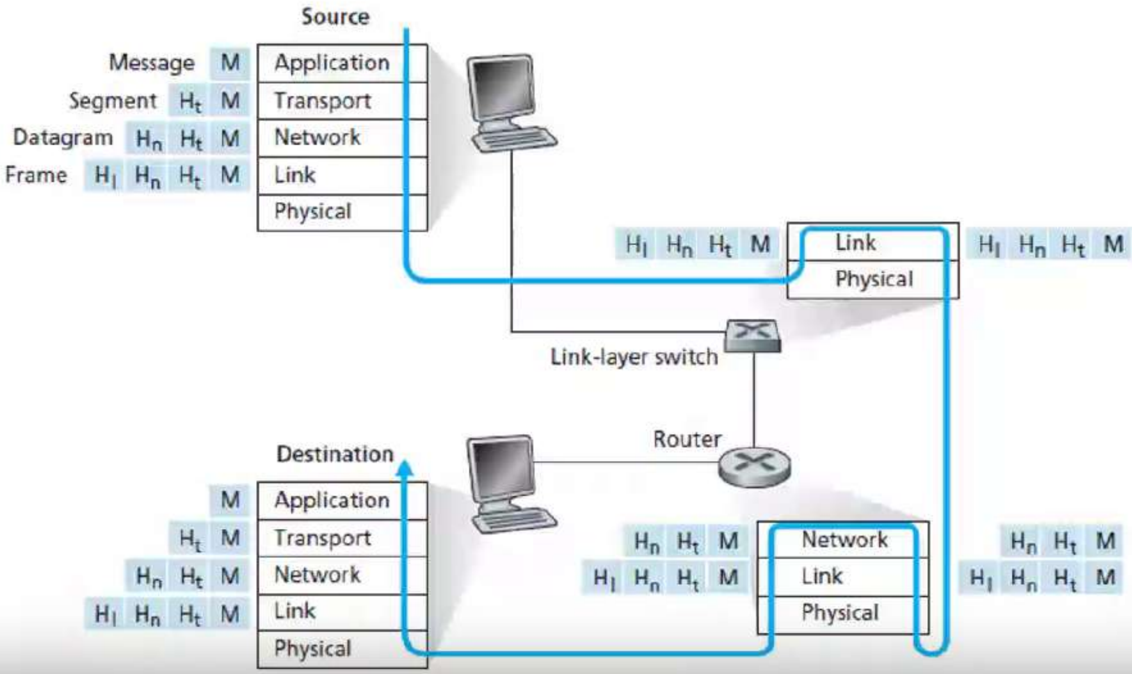


Figure 1.24 ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

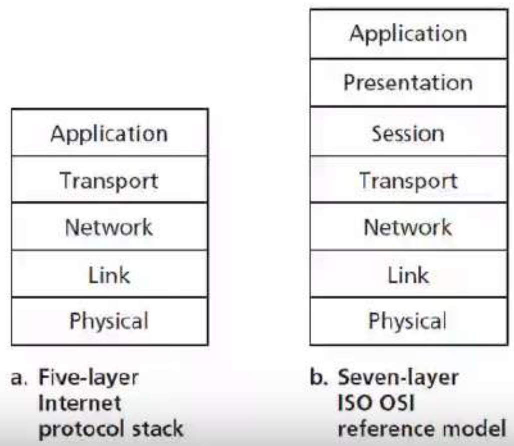
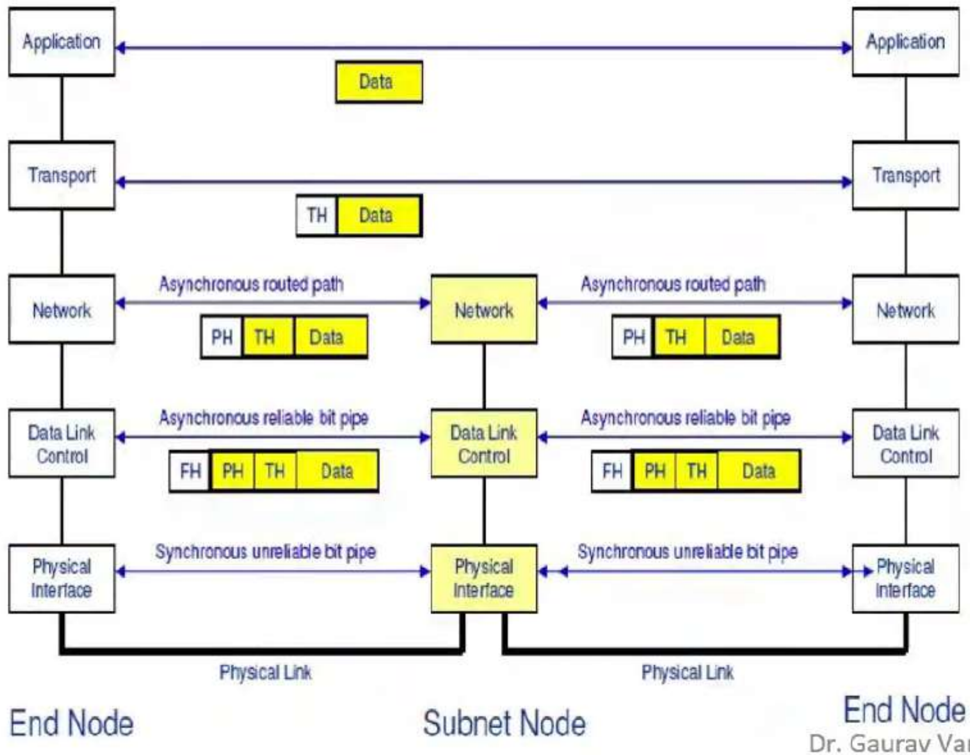


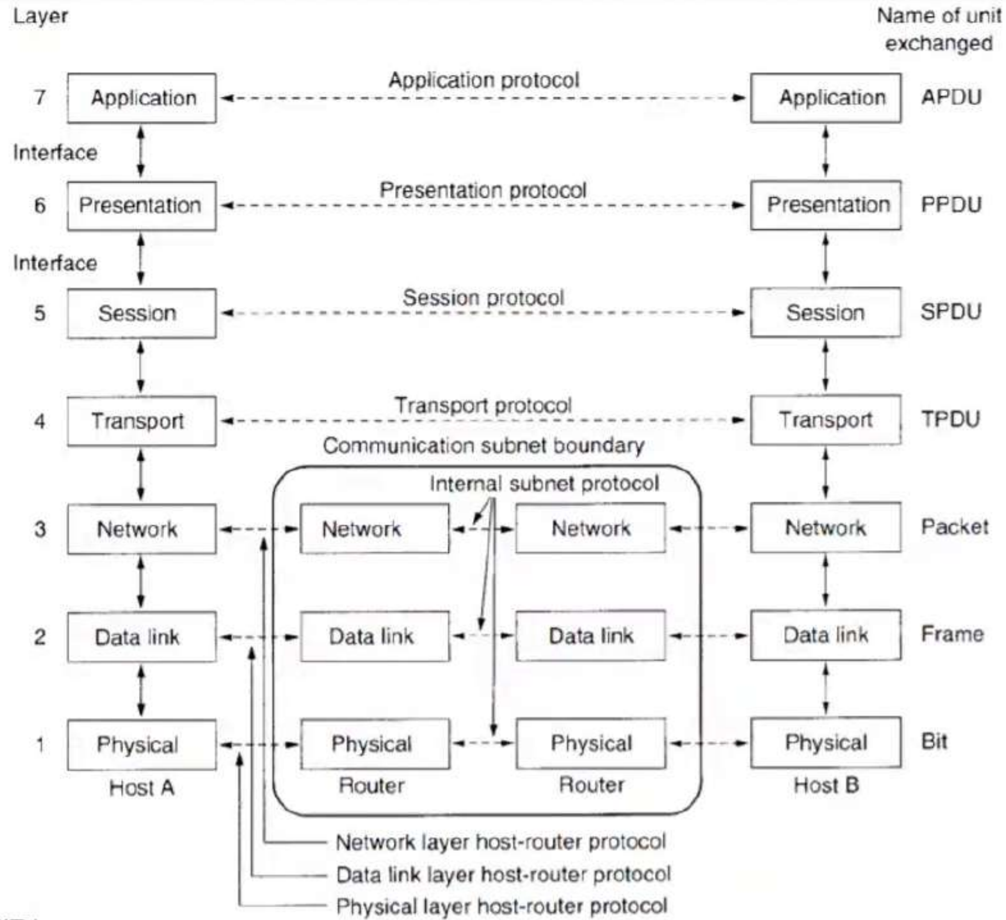
Figure 1.23 ♦ The Internet protocol stack (a) and OSI reference model (b)

- Individual layer of the protocol stack perform a set of functions
- The layer is implemented as software or hardware using a set of protocols.

Encapsulation



TCP/IP and OSI Reference Model

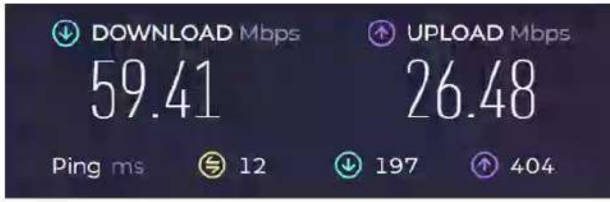


Dr. Gaurav Varshney IIT Jammu

Network Application Architecture

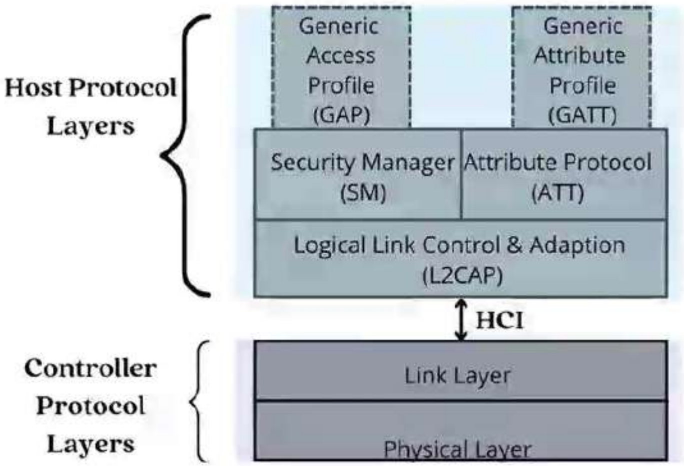
In the context of a communication session between a pair of processes, the process that initiates the communication (that is, initially contacts the other process at the beginning of the session) is labeled as the ***client****. The process that waits to be contacted to begin the session is the* ***server****.*

- **Client Server Architecture**
  - Server has a well known IP address and well known port number
  - Clients connect to servers for access to services
  - HTTP, FTP support client server application architecture
- **Peer to Peer Applications**
  - Peers communicate directly with each other without a central web server
  - In this way all peers are client and server at the same time.
  - Bit Torrent, Skype used to have Hybrid P2P protocol.
  - Advantage: Self Scalability
  - Disadvantage: ISP [upstream vs downstream], Security, Incentives [convince users to volunteer storage, bandwidth].



https://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf

https://novelbits.io/bluetooth-low-energy-protocol-stack-layers/



This section will explain the standard BitTorrent architecture as described in [4]. Later a new pure peer-to-peer approach will be explained.

The BitTorrent architecture normally consists of the following entities [27]:

- a static metainfo file (a "torrent file")
- a 'tracker'
- an original downloader ("seed")
- the end user downloader ("leecher")

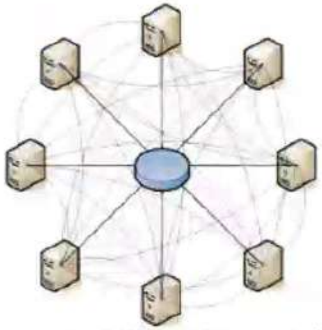


Figure 5 – BitTorrent in its original form matches the "hybrid" peer-to-peer concept. It's all about the torrent file, the centralized tracker and the associated swarm of peers. The centralized tracker provides the different entities with an address list over the available peers. These peers will then contact each other to download pieces of the file from each other.

The first step in publishing a file using BitTorrent is to create a metainfo file from the file that you want to publish. The metainfo file is called a "torrent". The torrent file contains the filename, size, hashing information and the URL of the "tracker". The "torrent" is needed by anyone who wants to download the file the torrent is created from. The torrent file can be distributed by e-mail, IRC, http etc.

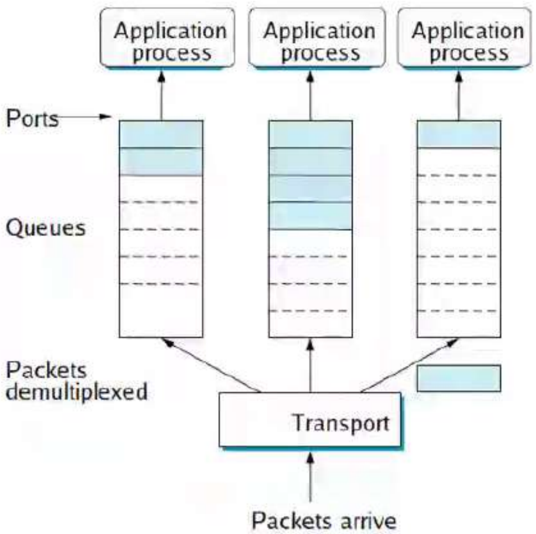
The torrent is created by using a free program. This functionality is also commonly included in the BitTorrent clients. To download or "seed" a file, you need a BitTorrent client. The BitTorrent client is a free application that administrates the download procedure. There are several different BitTorrent clients available [28]. They all support the standard BitTorrent protocol, but may differ and be incompatible with each other regarding certain features [29]. A BitTorrent download is started by opening the torrent file in the BitTorrent client.

The tracker keeps a log of peers that are currently downloading a file, and helps them find each other. The tracker is not directly involved in the transfer of data and does not have a copy of the file. The tracker and the downloading users exchange information using a simple protocol on top of HTTP. First, the user gives information to the tracker about which file it's downloading, ports it's listening on etc. The responce from the tracker is a list of other users which are downloading the same file and information on how to contact them. This group of peers that all share the same torrent represents a 'swarm'.

However, making a torrent file is not enough to make the file you want to distribute available. An original downloader known as a "seed" has to be started. A "seed" is a user that has the entire file. A downloading user that has nothing or only parts of a file is called a "leecher". The "seed" must upload at least one complete copy of the file. Once an entire copy is distributed amongst the other downloaders, the 'seed' can stop uploading and the download will still continue for all downloaders as long as there are enough people downloading the file, and all pieces of the file are available. For a popular file one complete copy from the seed

Sockets

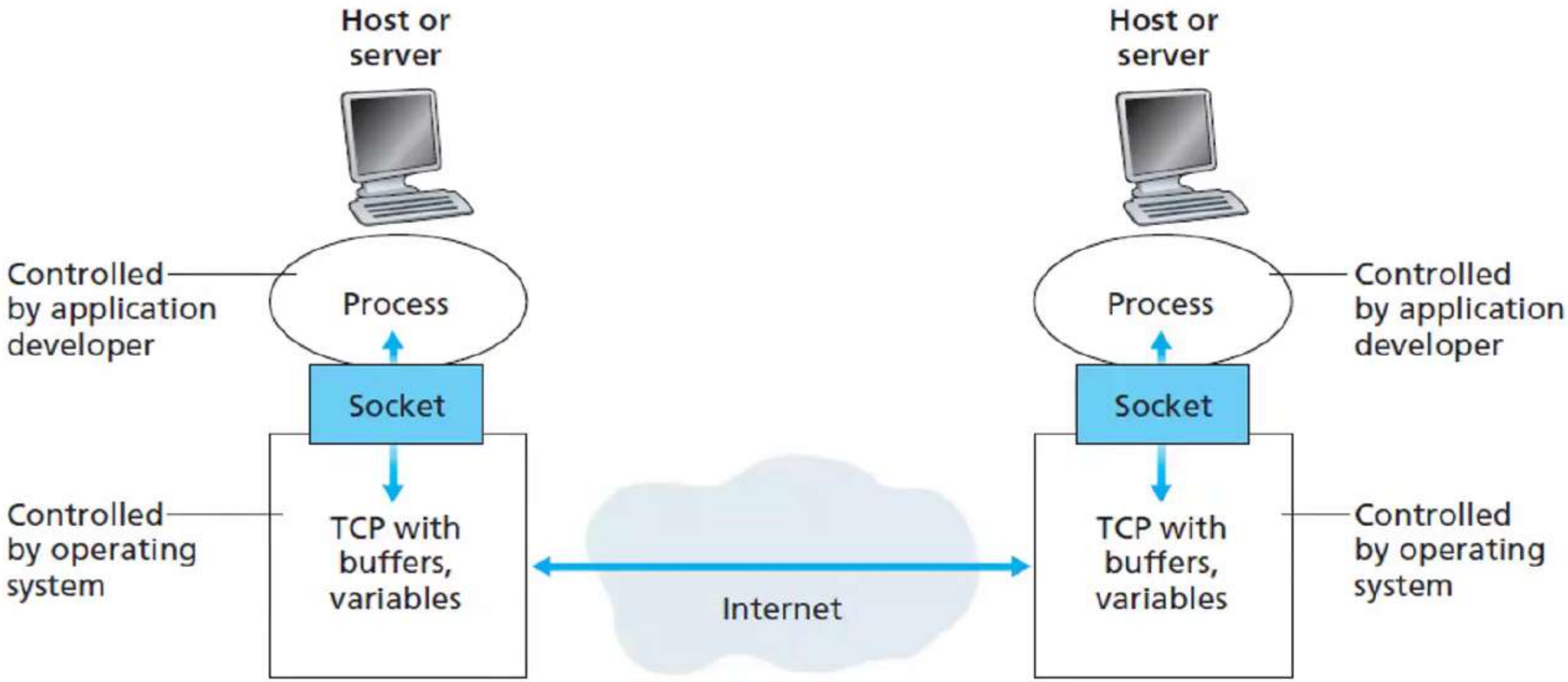
- Application generate data but how to push that data to Transport layer controlled by operating system.
- Application create socket to push the data to the below layer. A socket connect two endpoints in a network.
- An end point is defined by an IP address and a port.
  - Port is unique for every process and it provides multiplexing and demultiplexing at the Transport layer.



- Socket: An interface between an application process and transport layer
  - The application process can send/receive messages to/from another application process (local or remote)via a socket
- In Unix jargon, a socket is a file descriptor – an integer associated with an open file

Kameswari Chebrolu
Dept. of Electrical Engineering, IIT Kanpur

Application Layer to Transport Layer



Transport Services Application

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Figure 2.4 ♦ Requirements of selected network applications