



Indian Institute of Information Technology, Sri City, Chittoor
(An Institute of National Importance under an Act of Parliament)

Computer Communication Networks

Introduction, Communication link, Multiplexing

Dr. Raja Vara Prasad

Assistant Professor

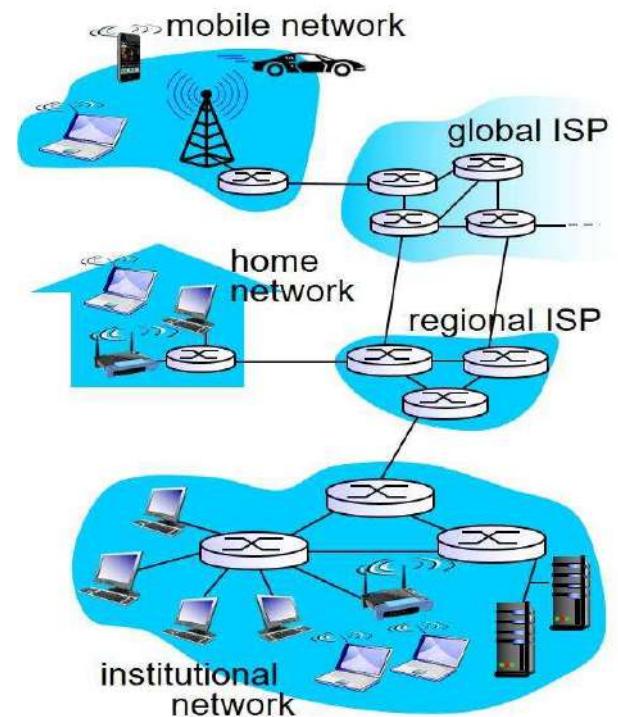
IIIT Sri City

Content

- Introduction
- Communication Link
 - Guided
 - Unguided
- Multiplexing
 - Frequency division multiplexing (FDM)
 - Time division multiplexing (TDM)

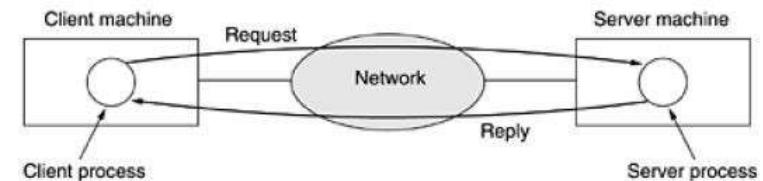
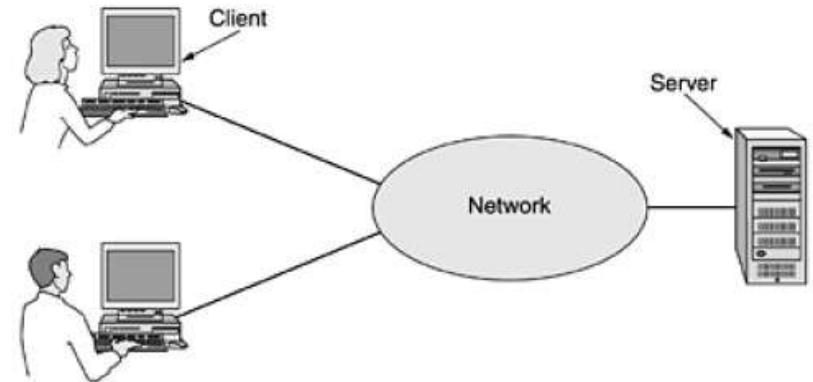
What is a Network?

- A network is an interconnection of devices.
- The computers/laptops connected to the network are known as end systems or hosts.
- The digital data is fragmented into packets.



Uses of Computer network

- Business applications
 - Resource sharing
 - powerful medium of communication (email and online document preparation)
 - Video conferencing
 - Doing business electronically with other companies (ex: Isuzu).
 - Doing business with consumer (online market).



Uses of Computer network

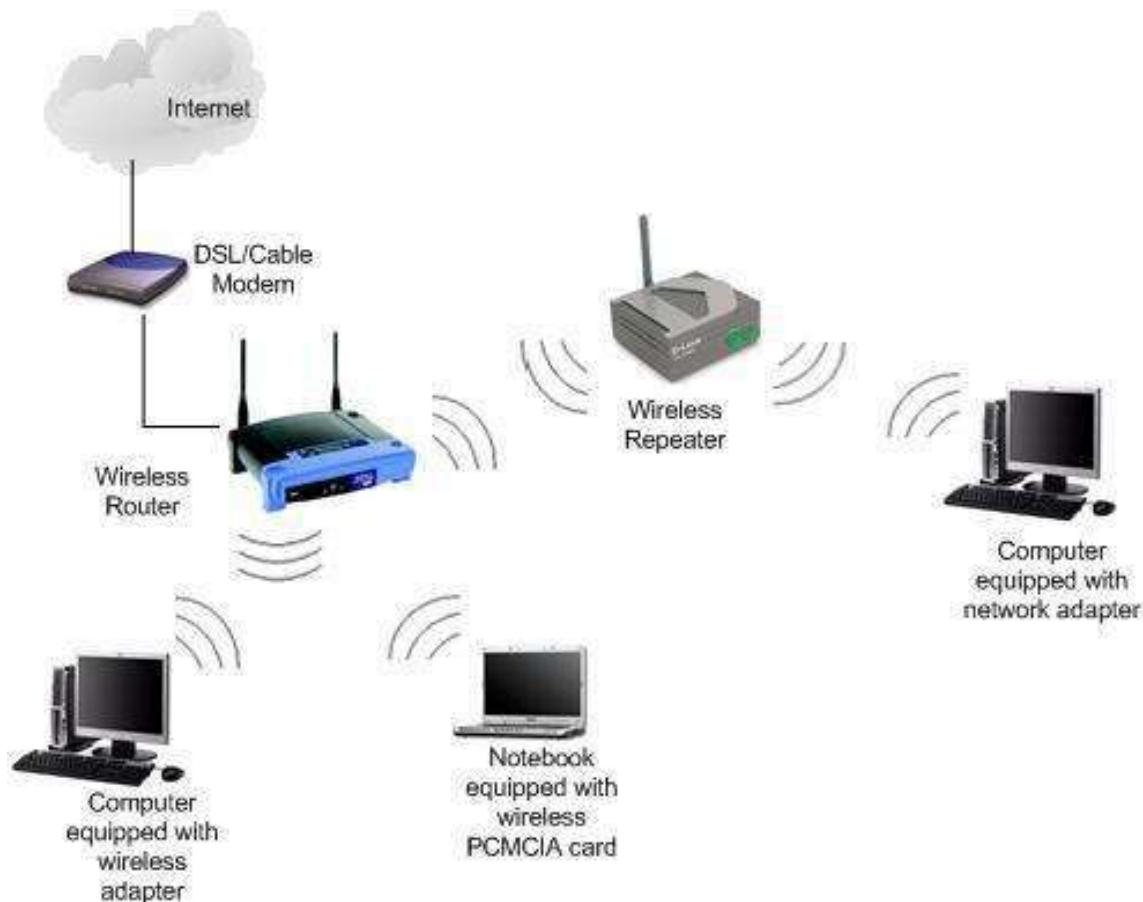
- Home applications
 - Why do people buy computer for home use?
 - Earlier days it is for word processing and gaming , now for “Internet access”
 - Internet provides access to **remote information**, **person- to-person communication**, **entertainment**, **e-commerce**.

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books on-line
B2B	Business-to-business	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products on line
P2P	Peer-to-peer	File sharing

Network Essentials

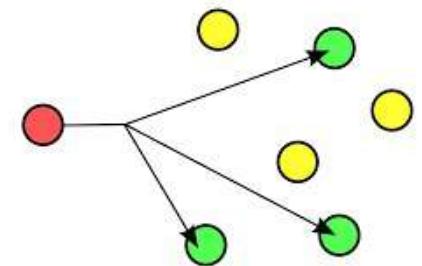
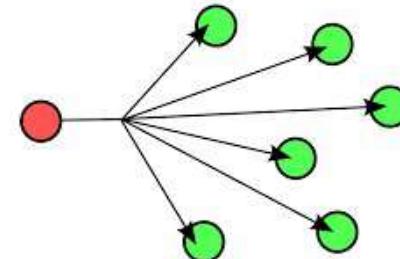
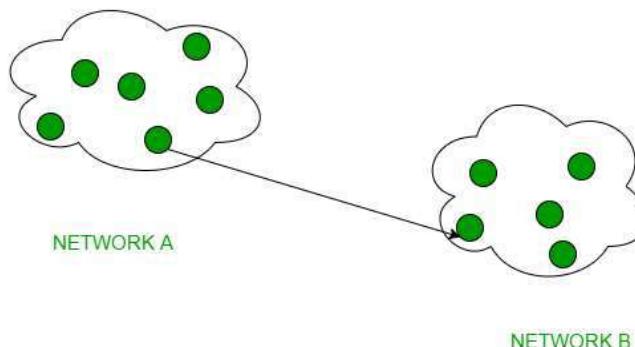
- **Modem**
- “**Modulator and demodulator**”, is a hardware device that converts data into a format suitable for a transmission medium so that it can be transmitted from one computer to another.
- **Ethernet**
- System for connecting the number of computers to form a LAN.
- **Router**
- A router is a device that forwards data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network.
- **Repeater**
- A **network** device used to regenerate or replicate a signal. **Repeaters** are used in transmission systems to regenerate analog or digital signals distorted by transmission loss. Analog **repeaters** frequently can only amplify the signal while digital **repeaters** can reconstruct a signal to near its original quality.

Wireless Network



Classification of Networks

- Transmission technology:
 - Unicasting : transmission with exactly one sender and exactly one receiver
 - Broadcasting : information is intended to all hosts
 - Multicasting : information is intended for a subset of hosts in the network



Network Hardware: Classification

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

Classification of Networks:

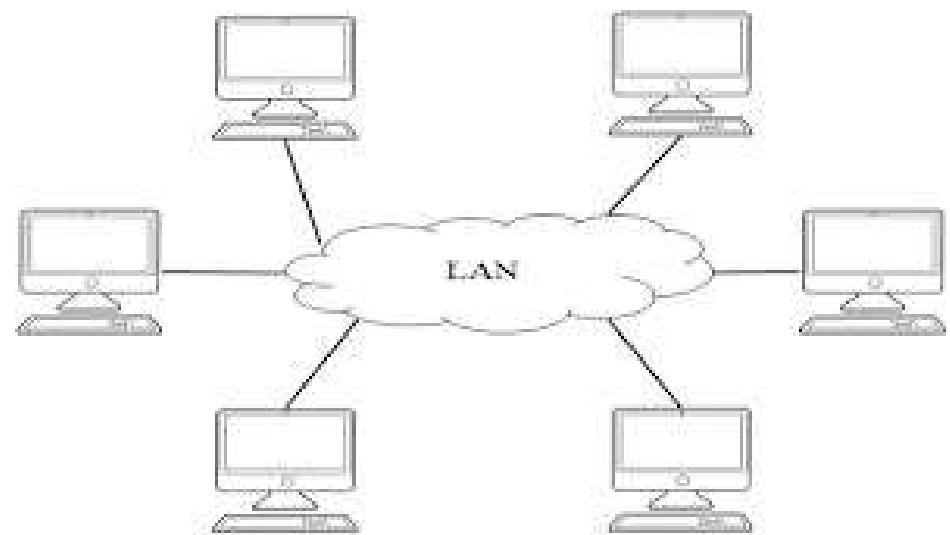
- Personal area networks (PANs)
 - Organized around an individual person, within a small office or residence.
 - Within the range of few meters
 - Notable example is Bluetooth
 - Watching movies on online streaming service to TV
 - With multiple uses within a same residence then, referred as Home Area Network (HAN).



Connecting peripherals to computer via Bluetooth

Classification of Networks:

- Local area networks (LANs)
 - Typically an individual office building: suitable for sharing resources (data storage and printers).
 - Range: It can reach few hundred meters, can be increased further using wireless repeaters.
 - Wireless LAN: WLAN



Privately owned network: wireless/wired connections.

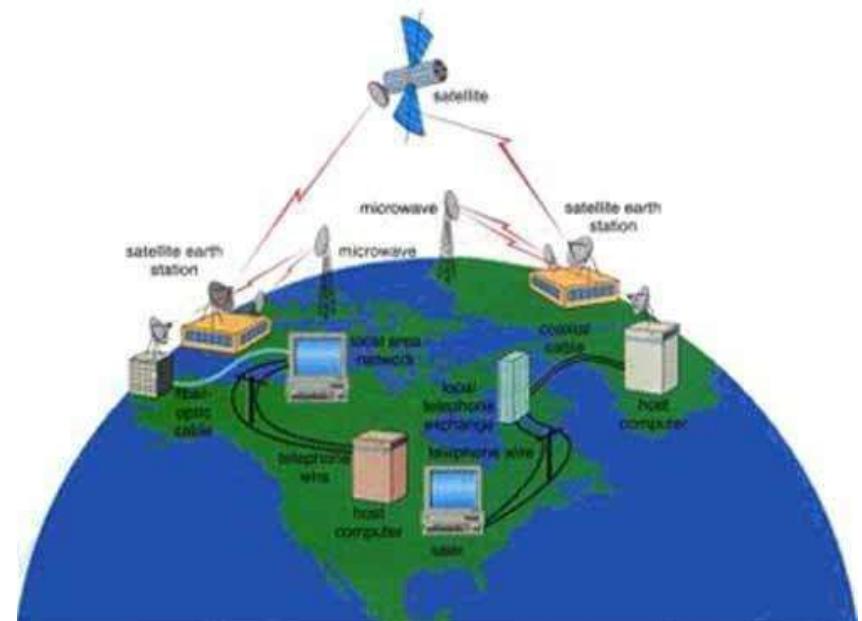
Classification of Networks:

- Metropolitan area networks (MANs)
 - Computer network across entire city, college campus or small region.
 - Referred as Campus Area Network (CAN).
 - Range: from several miles to tens of miles.
 - Connect several LANs together to form a bigger network.



Classification of Networks:

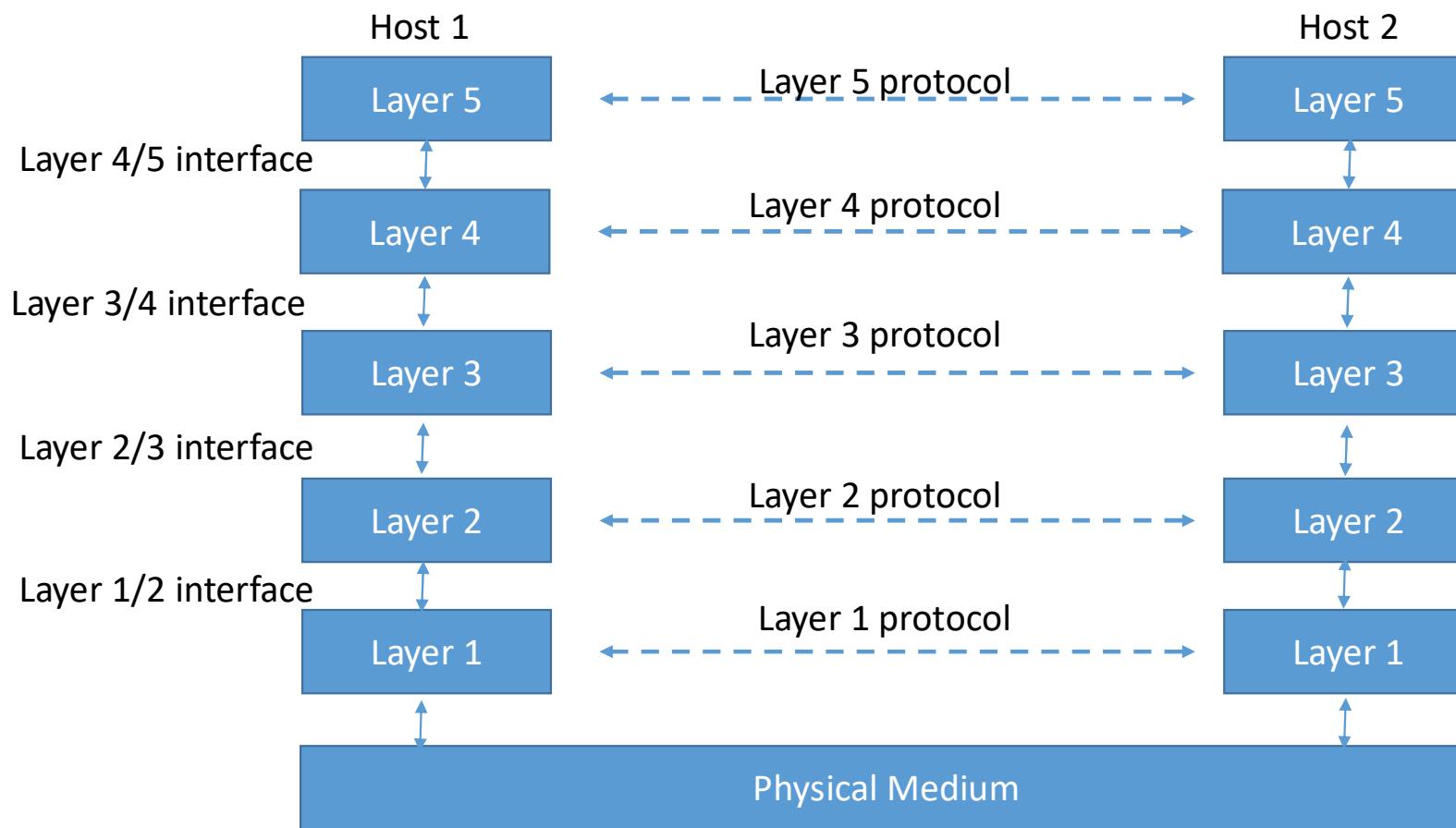
- Wide area networks (WANs)
 - Occupies a very large area, such as an entire country or the entire world
 - can contain multiple smaller networks, such as LANs or MANs
 - The most well-known WAN is the “**Internet**”



Network Software

- Protocol
 - Is an agreement between the communicating parties on how communication is to proceed.
 - Violation of protocol will make communication more difficult, if not completely impossible.

Layers, protocols, and interfaces

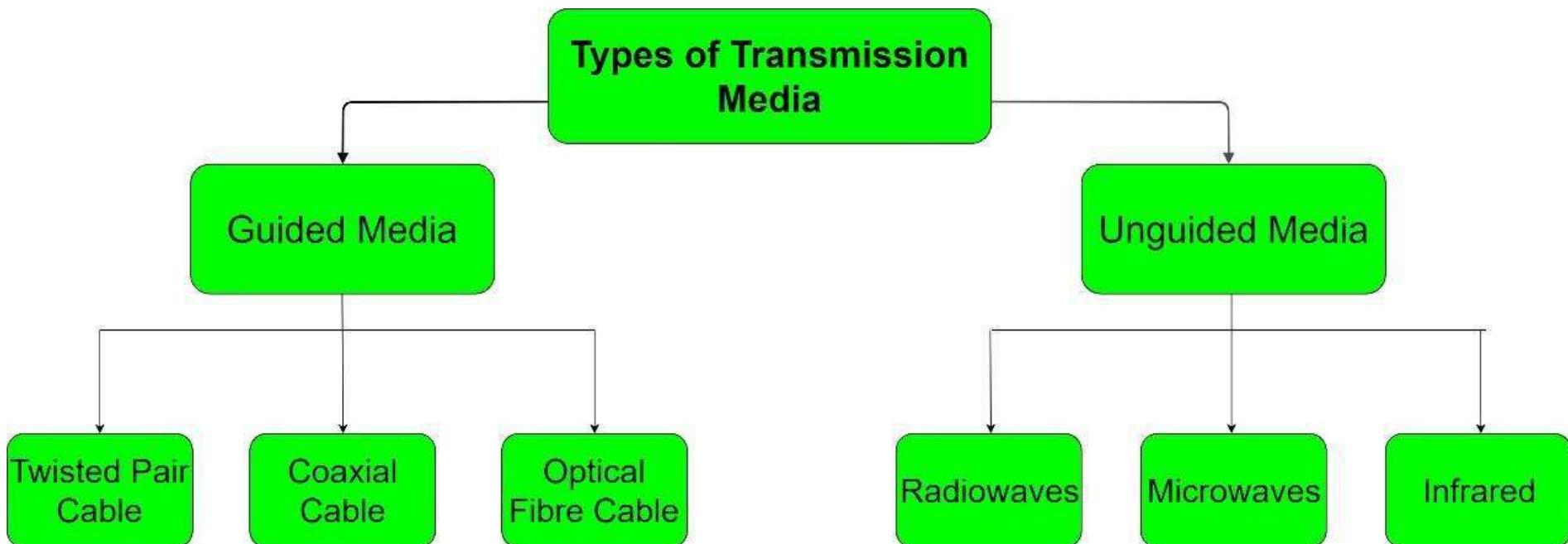


The Physical Layer

- Lowest of our protocol model
- Defines the electrical, timing and other interfaces by which bits are sent as signals over channels.
- The properties of different kinds of physical channels determine the performance.
- Kinds of transmission media: Guided and Unguided.

Communication Link?

Communication link : provides a way for information to move between physically separated components



Magnetic Media

- One of the most common ways to transport data from one computer to another is to write them onto magnetic tape or removable media.
- It is often more cost effective, especially for applications in which high bandwidth or cost per bit transported is the key factor.



Floppy disc.



Hard Disc



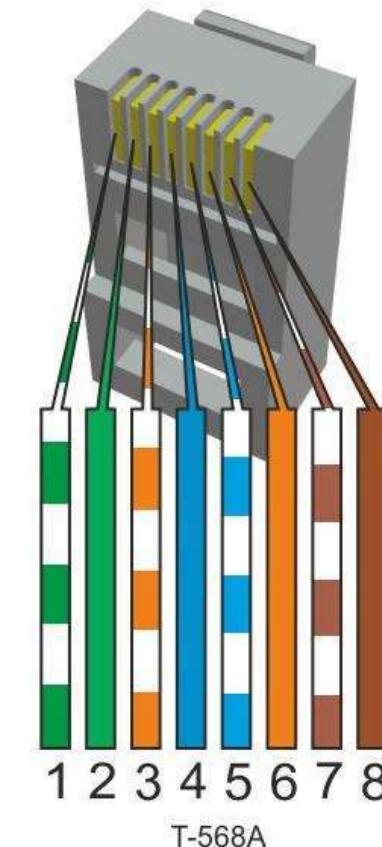
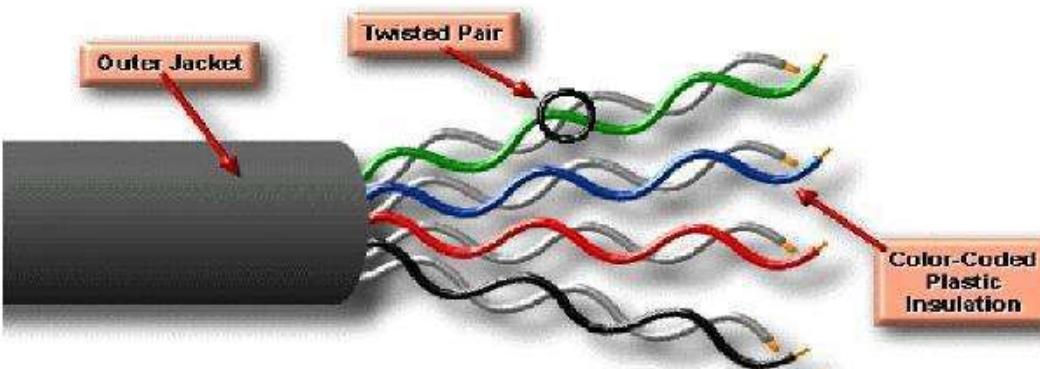
External hard Disc

Twisted pair

- Used for telephone communications and most modern Ethernet networks.
- A pair of wires forms a circuit that can transmit data.
- The pairs are twisted to provide protection against *crosstalk*, the noise generated by adjacent pairs.
- When electrical current flows through a wire, it creates a small, circular magnetic field around the wire (Ampere's Law).
- When two wires in an electrical circuit are placed close together, their magnetic fields are the exact opposite of each other. Thus, the two magnetic fields cancel each other out.
- Twisting the wires can enhance this *cancellation effect*.
- *Two types: Unshielded, and shielded.*

UTP (unshielded)

- is a medium that is composed of pairs of wires (4 pairs for network medium)
- UTP cable often is installed using a Registered Jack 45 (RJ-45) connector



Pin	Description	10base-T	100Base-T	1000Base-T
1	Transmit Data+ or BiDirectional	TX+	TX+	BI_DA+
2	Transmit Data- or BiDirectional	TX-	TX-	BI_DA-
3	Receive Data+ or BiDirectional	RX+	RX+	BI_DB+
4	Not connected or BiDirectional	n/c	n/c	BI_DC+
5	Not connected or BiDirectional	n/c	n/c	BI_DC-
6	Receive Data- or BiDirectional	RX-	RX-	BI_DB-
7	Not connected or BiDirectional	n/c	n/c	BI_DD+
8	Not connected or BiDirectional	n/c	n/c	BI_DD-

UTP

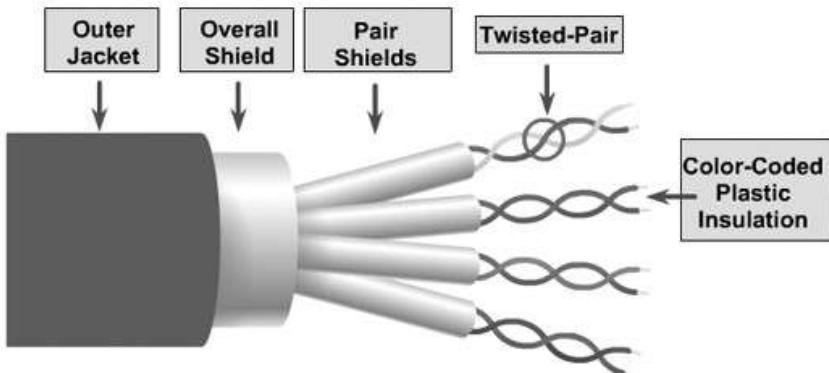
- Advantages: Smaller size (external diameter), easy to install and less expensive.
- Disadvantages: UTP cable is more prone to electrical noise and interference than other types of networking media, and the distance between signal boosts is shorter for UTP .
- The following summarizes the features of UTP cable:
 - Speed and throughput—10 to 1000 Mbps
 - Average cost per node—Least expensive
 - Media and connector size—Small
 - Maximum cable length—100 m (short)

UTP cabling

- **Category 1 (1 pair)**—Used for telephone communications. Not suitable for transmitting data.
- **Category 2 (2 pairs)**—Capable of transmitting data at speeds up to 4 megabits per second (Mbps).
- **Category 3 (4 pairs)**—Used in 10BASE-T networks, Can transmit data at speeds up to 10 Mbps.
- **Category 4 (4 pairs)**—Used in Token Ring networks, Can transmit data at speeds up to 16 Mbps.
- **Category 5 (4 pairs)**—Can transmit data at speeds up to 100 Mbps.
- **Category 5e (4 pairs)** —Used in networks running at speeds up to 1000 Mbps (1 gigabit per second [Gbps]).
- **Category 6 (4 pairs)**—Typically, Category 6 cable consists of four pairs of 24 American Wire Gauge (AWG) copper wires. Category 6 cable is currently the fastest standard for UTP.

STP (Shielded Twisted Pair)

- Combines the techniques of shielding, cancellation, and wire twisting.
- Each pair of wires is wrapped in a metallic foil. The four pairs of wires then are wrapped in an overall metallic braid or foil.
- Reduces electrical noise both within the cable (pair-to-pair coupling, or crosstalk) and from outside the cable (EMI and RFI)



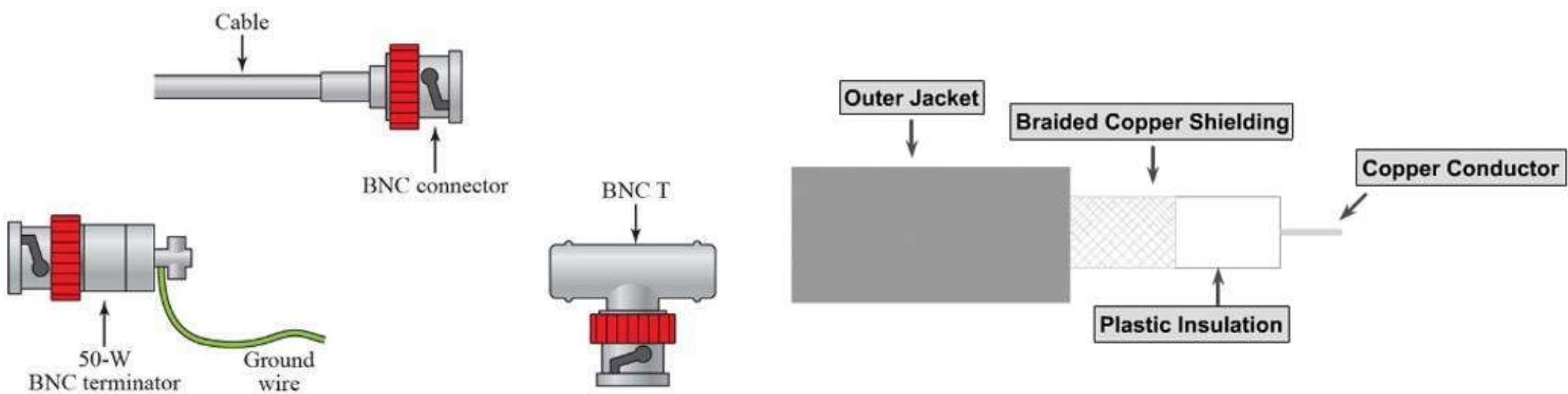
- Speed and throughput: 10-100 Mbps
- Cost per node: Moderately expensive
- Media and connector size: Medium to Large
- Maximum cable length: 100m (short)

STP comparison with UTP

- Although STP prevents interference better than UTP, it is more expensive and difficult to install.
- the metallic shielding must be grounded at both ends. If it is improperly grounded, the shield acts like an antenna and picks up unwanted signals.
- Because of its cost and difficulty with termination, STP is rarely used in Ethernet networks.
- The speed of both types of cable is usually satisfactory for local-area distances.

Coaxial Cable

- Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and a braided metal shield.
- The metal shield helps to block any outside interference from fluorescent lights, motors, and other computers.
- The most common type of connector used the Bayonet Neill-Concelman (BNC) connector



Categories of Coax.

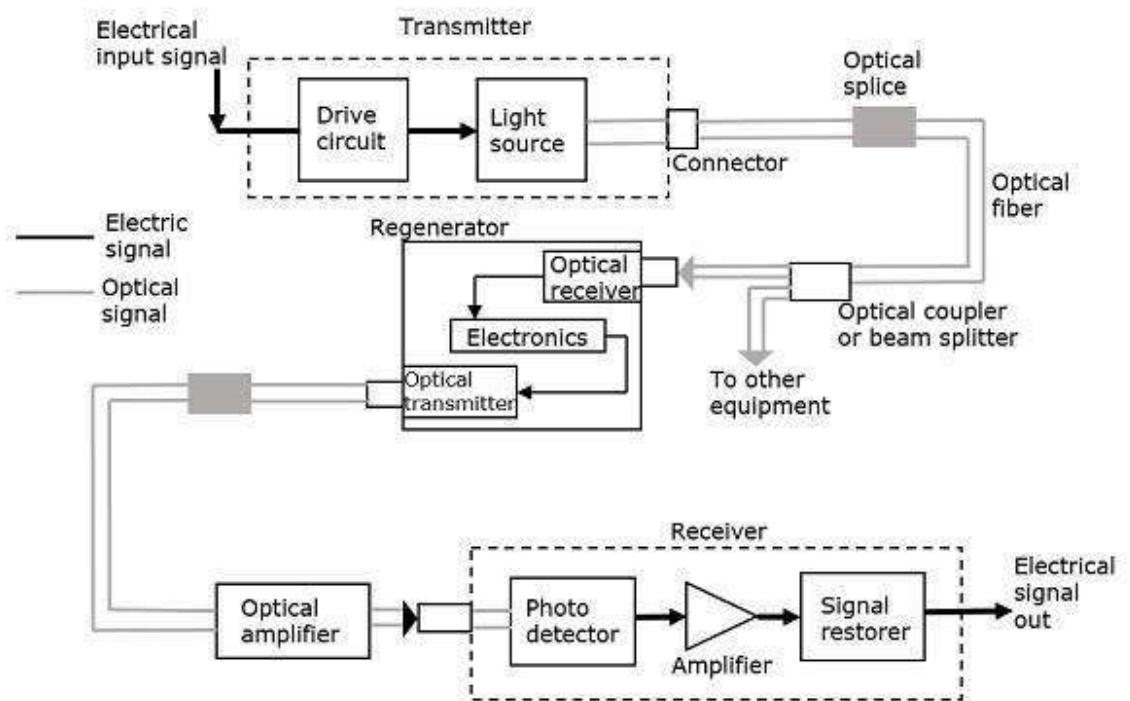
- Base band
- For digital transmission, a 50 ohm (Ω) coaxial cable is used. It defines a process of transmitting a single signal at a time with a very high speed. It is generally used for LAN's.
- Broadband
- Analog transmission on standard cable television 75 ohm (Ω) cabling is used by this. It defines a process of transmitting multiple signals simultaneously with very high speed. It covers a large area as compared to Baseband Coaxial Cable.

Advantages and Disadvantages

- It can be used for both analog and digital transmission.
- It offers higher bandwidth as compared to twisted pair cable and can span longer distances.
- Because of better shielding in coaxial cable, loss of signal or attenuation is less.
- Better shielding also offers good noise immunity.
- It is relatively inexpensive as compared to optical fibers.
- It has lower error rates as compared to twisted pair.
- It is not as easy to tap as twisted pair because copper wire is contained in plastic jacket.
- It is usually more expensive than twisted pair.

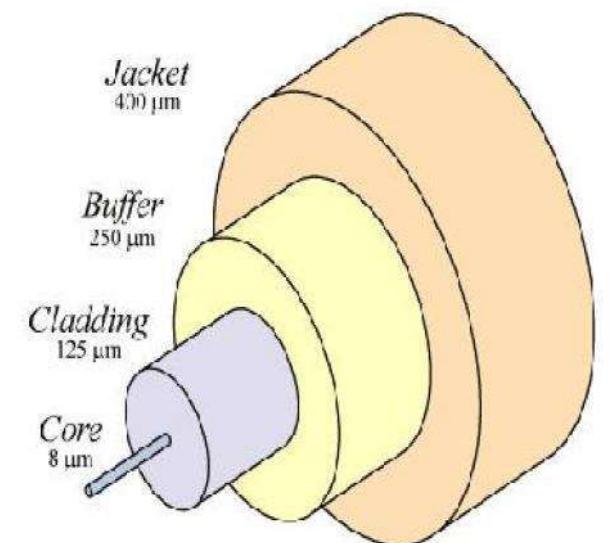
Fiber optics

- An optical transmission system has three key components:
 - the light source,
 - the transmission medium, and
 - the detector



Fiber Optics: Construction

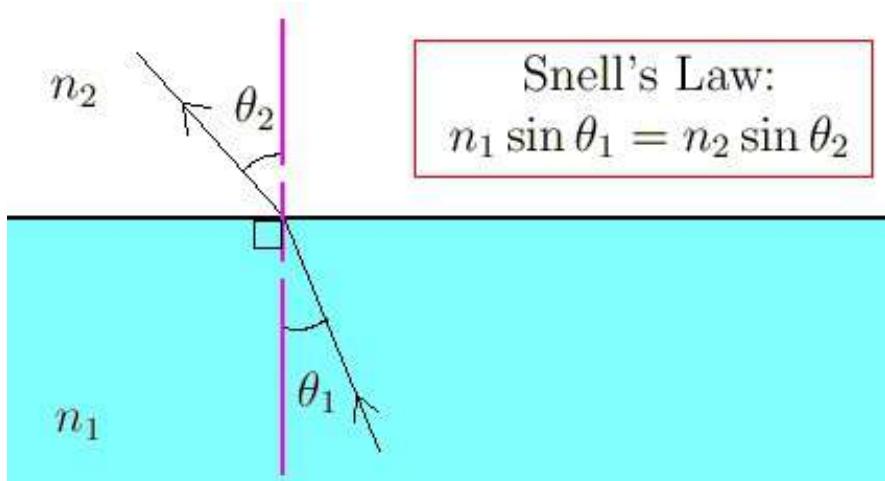
- Core: The core of a fiber cable is a cylinder of plastic that runs all along the fiber cable's length. The diameter of the core depends on the application used.
- Cladding: Cladding is an outer optical material that protects the core. The main function of the cladding is that it reflects the light back into the core.
- Buffer: The main function of the buffer is to protect the fiber from damage and thousands of optical fibers arranged in hundreds of optical cables.
- Jacket: These bundles are protected by the cable's outer covering that is called jacket.



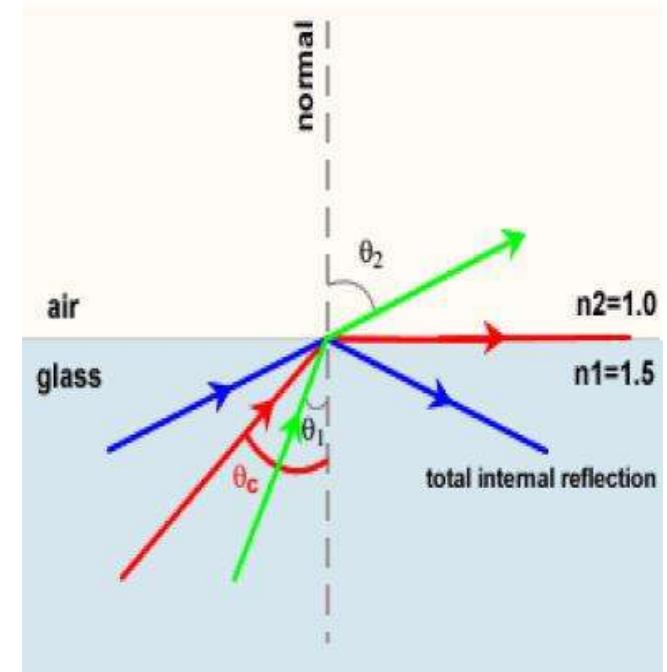
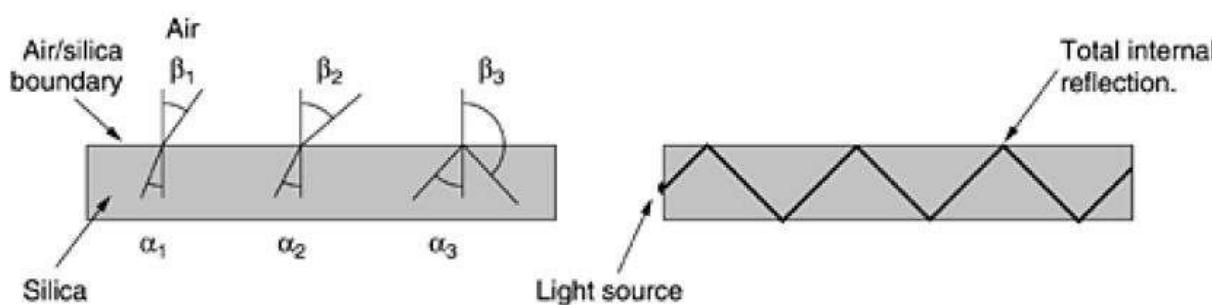
Fiber Optics: Working Principle

- A hair-thin Fiber consist of two concentric layers of high-purity silica glass the core and the cladding, which are enclosed by a protective sheath.
- Core and cladding have different refractive indices, with the core having a refractive index, n_1 , which is slightly higher than that of the cladding, n_2 .
- When light enters the fiber made of material with higher refractive index than the cladding surrounding it, it stays inside the material due to total internal reflection and is thus transmitted forward.
- **Index of refraction:** Index of refraction is a measurement of speed of light in material.

Snell's Law: Law of Refraction, Critical Angle, total internal Reflection

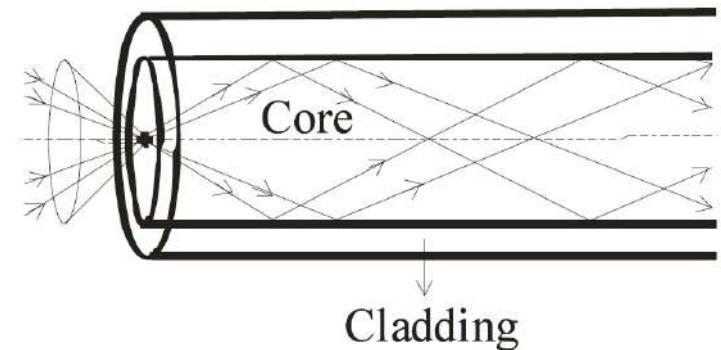
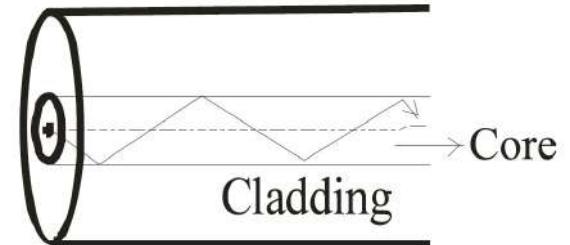


$$\text{Snell's Law: } n_1 \sin \theta_1 = n_2 \sin \theta_2$$



Fiber Optics: Modes

- Mode is the one which describes the nature of propagation of electromagnetic waves (light) in a wave guide (Fiber).
- Single mode fiber: In a fiber, if only one mode is transmitted through it, then it is said to be a single mode fiber.
- If more than one mode is transmitted through optical fiber, then it is said to be a multimode fiber.
- The larger core radii of multimode fibers make it easier to launch optical power into the fiber and facilitate the end to end connection of similar powers.



Types of Fibers

- **Step-index fiber** – The refractive index of the core is uniform throughout and undergoes an abrupt change (or step) at the cladding boundary.
- **Graded-index fiber** – The core refractive index is made to vary as a function of the radial distance from the center of the fiber.
- Further divided into:
- **Single-mode fiber** – These are excited with laser.
- **Multi-mode fiber** – These are excited with LED.

Item	LED	Semiconductor laser
Data rate	Low	High
Fiber type	Multimode	Multimode or single mode
Distance	Short	Long
Lifetime	Long life	Short life
Temperature sensitivity	Minor	Substantial
Cost	Low cost	Expensive

Fiber Optics: Advantages and Disadvantages

- **Advantages:**

- The transmission bandwidth of the fiber optic cables is higher than the metal cables.
- The amount of data transmission is higher in fiber optic cables.
- The power loss is very low and hence helpful in long-distance transmissions.
- Fiber optic cables provide high security and cannot be tapped.
- Fiber optic cables are the most secure way for data transmission.
- Fiber optic cables are immune to electromagnetic interference.
- These are not affected by electrical noise.

- **Disadvantages:**

- Though fiber optic cables last longer, the installation cost is high.
- The number of repeaters are to be increased with distance.
- They are fragile if not enclosed in a plastic sheath. Hence, more protection is needed than copper ones

Media Type	Maximum Segment Length	Speed	Cost	Advantages	Disadvantages
UTP	100 m	10 Mbps to 1000 Mbps	Least expensive	Easy to install; widely available and widely used	Susceptible to interference; can cover only a limited distance
STP	100 m	10 Mbps to 100 Mbps	More expensive than UTP	Reduced crosstalk; more resistant to EMI than Thinnet or UTP	Difficult to work with; can cover only a limited distance
Coaxial	500 m (Thicknet) 185 m (Thinnet)	10 Mbps to 100 Mbps	Relatively inexpensive, but more costly than UTP	Less susceptible to EMI interference than other types of copper media	Difficult to work with (Thicknet); limited bandwidth; limited application (Thinnet); damage to cable can bring down entire network
Fiber-Optic	10 km and farther (single-mode) 2 km and farther (multimode)	100 Mbps to 100 Gbps (single mode) 100 Mbps to 9.92 Gbps (multimode)	Expensive	Cannot be tapped, so security is better; Difficult to terminate can be used over great distances; is not susceptible to EMI; has a higher data rate than coaxial and twisted-pair cable	

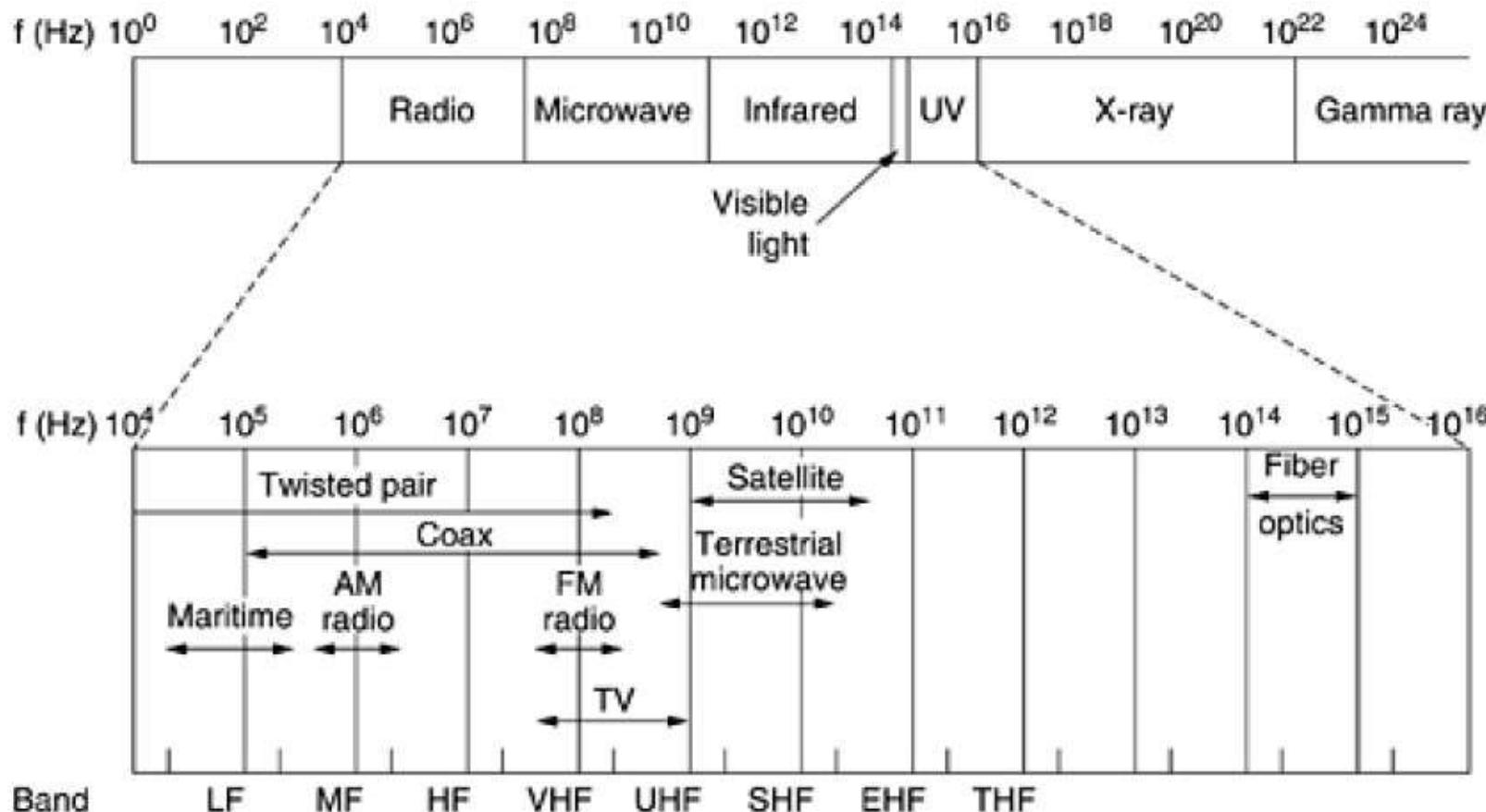
Unguided Media: Wireless Transmission

- Electromagnetic Spectrum
- Radio transmission
- Microwave Transmission
- Infrared Transmission
- Light Transmission

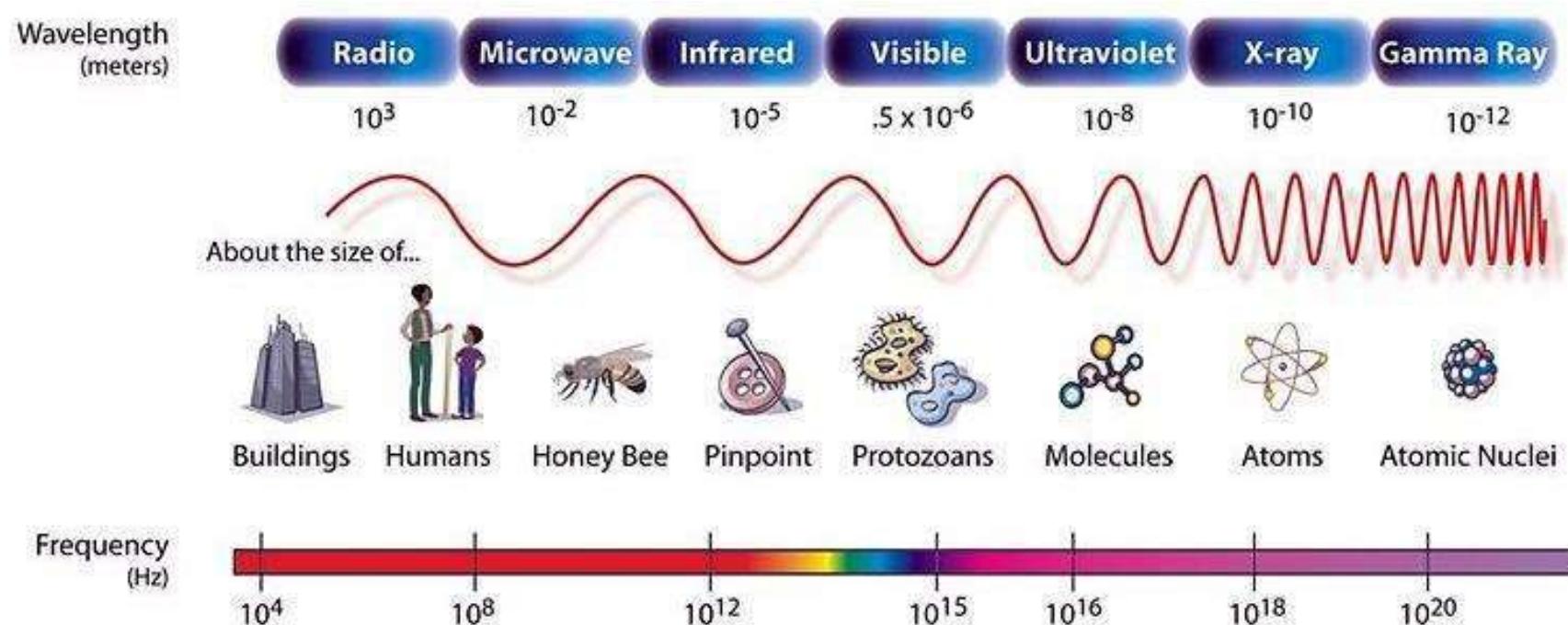
Electromagnetic Spectrum

- Electromagnetic waves, can propagate through space, were predicted by J C Maxwell in 1865 and observed by Heinrich Hertz in 1887.
- When an antenna of the appropriate size is attached to an electrical circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away.
- In vacuum, all electromagnetic waves travel at the same speed, no matter what their frequency.
- In copper or fiber the speed slows to about 2/3 of this value and becomes slightly frequency dependent.

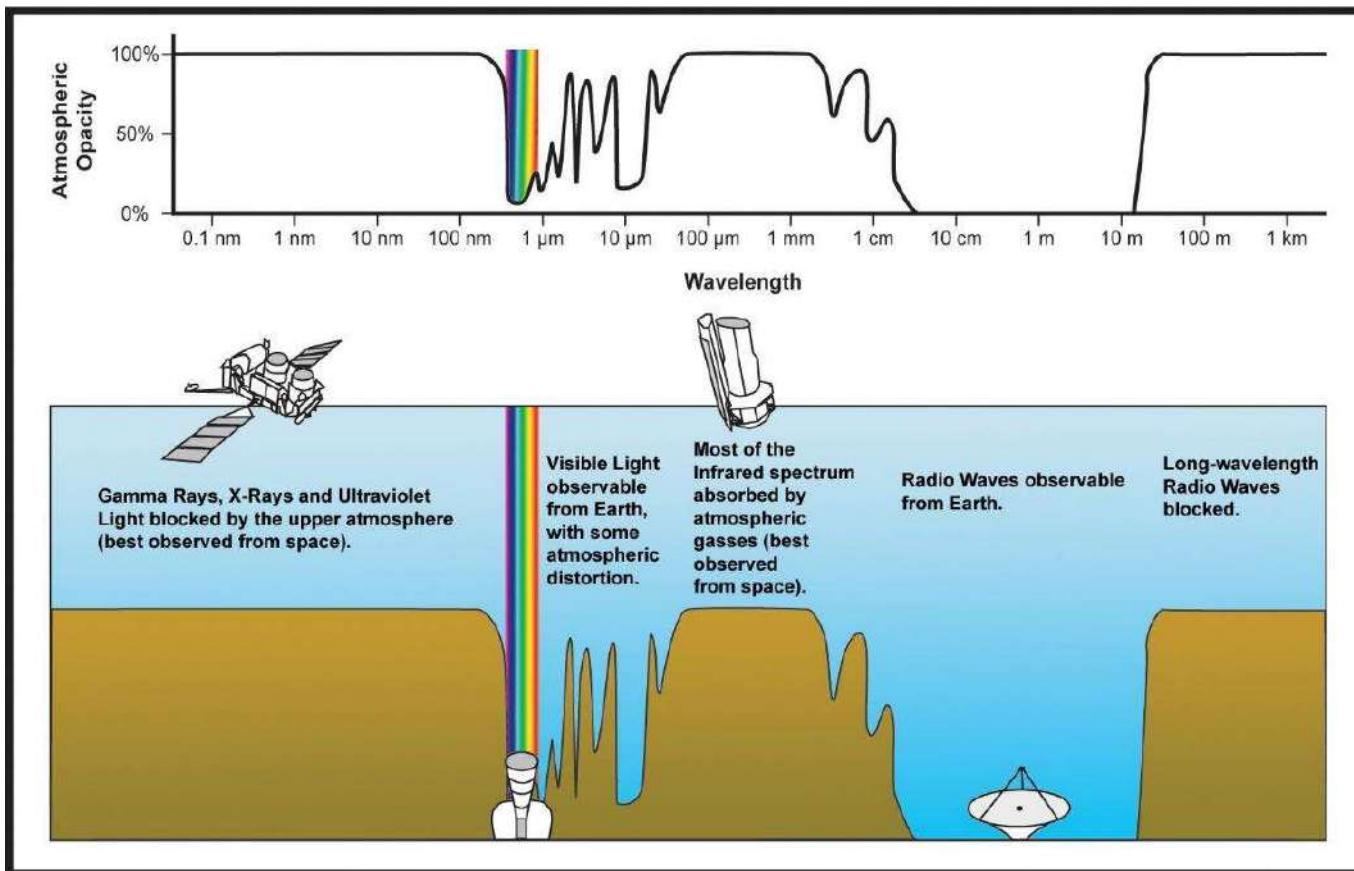
Electromagnetic spectrum and its uses for communication



Electromagnetic spectrum and its uses for communication



Electromagnetic spectrum and its uses for communication



Electromagnetic spectrum and its uses for communication

- The **radio**, **microwave**, **infrared**, and **visible light** portions of the spectrum can all be **used for transmitting information**
- by modulating the amplitude, frequency, or phase of the waves.
- **Ultraviolet light**, **X-rays**, and **gamma rays** would be even better, due to their higher frequencies.
- but they are **hard to produce and modulate**, do not propagate well through buildings, and are **dangerous to living things**.

Radio Transmission (10 kHz – 300 MHz)

- easy to generate, can travel long distances, and can penetrate buildings easily, so they are widely used for communication, both indoors and outdoors.
- Radio waves also are omnidirectional, meaning that they travel in all directions from the source, so the transmitter and receiver do not have to be carefully aligned physically.
- Due to radio's ability to travel long distances, interference between users is a problem. For this reason, all governments tightly license the use of radio transmitters.
- There is a wide range of subcategories contained within radio including AM and FM radio

Radio Transmission

- **AM radio waves:** commercial radio signals (540 and 1600 kHz), information is carried by amplitude variation, while the frequency remains constant.
- **FM radio waves:** commercial radio signals (88 and 108 MHz), information is carried by frequency modulation, while the signal amplitude remains constant.
- TV broadcast: (174 – 216 MHz).

Microwave Transmission (300 MHz – 300 GHz)

- Microwaves are “small” compared to waves used in typical radio broadcasting.
- The microwave portion of the electromagnetic spectrum can be subdivided into:
 - **Extremely High Frequency (30 to 300 GHz):** wavelength range of 10 to 1 mm, so it is sometimes called the millimeter band.
 - **Super High Frequency (3 to 30 GHz):** ten to one centimeters, used for wireless LANs, cell phones, satellite communication, microwave radio relay links, and numerous short range terrestrial data links
 - **Ultra-High Frequency (300 MHz to 3 GHz):** 10 centimeters to 1 meter, used for television broadcasting, cordless phones, walkie-talkies, satellite communication, and numerous other applications

Infrared and Millimeter Wave

- Unguided infrared and millimeter waves are widely used for short-range communication (The remote controls used on televisions, VCRs, and stereos all use infrared communication).
- They are relatively directional, cheap, and easy to build but have a major drawback: they do not pass through solid objects.
- In general, as we go from long-wave radio toward visible light, the waves behave more and more like light and less and less like radio.
- On the other hand, infrared system in one room of a building will not interfere with a similar system in adjacent rooms or buildings.
- Infrared communication has a limited use on the desktop, for example, connecting notebook computers and printers, it is not a major player in the communication.

Light Wave Transmission

- A more modern application is to connect the LANs in two buildings lasers mounted on their rooftops.
- Coherent optical signaling using lasers is inherently unidirectional, so each building needs its own laser and its own photodetector. This scheme offers very high bandwidth and very low cost.
- It is also relatively easy to install and, unlike microwave, does not require an FCC license.

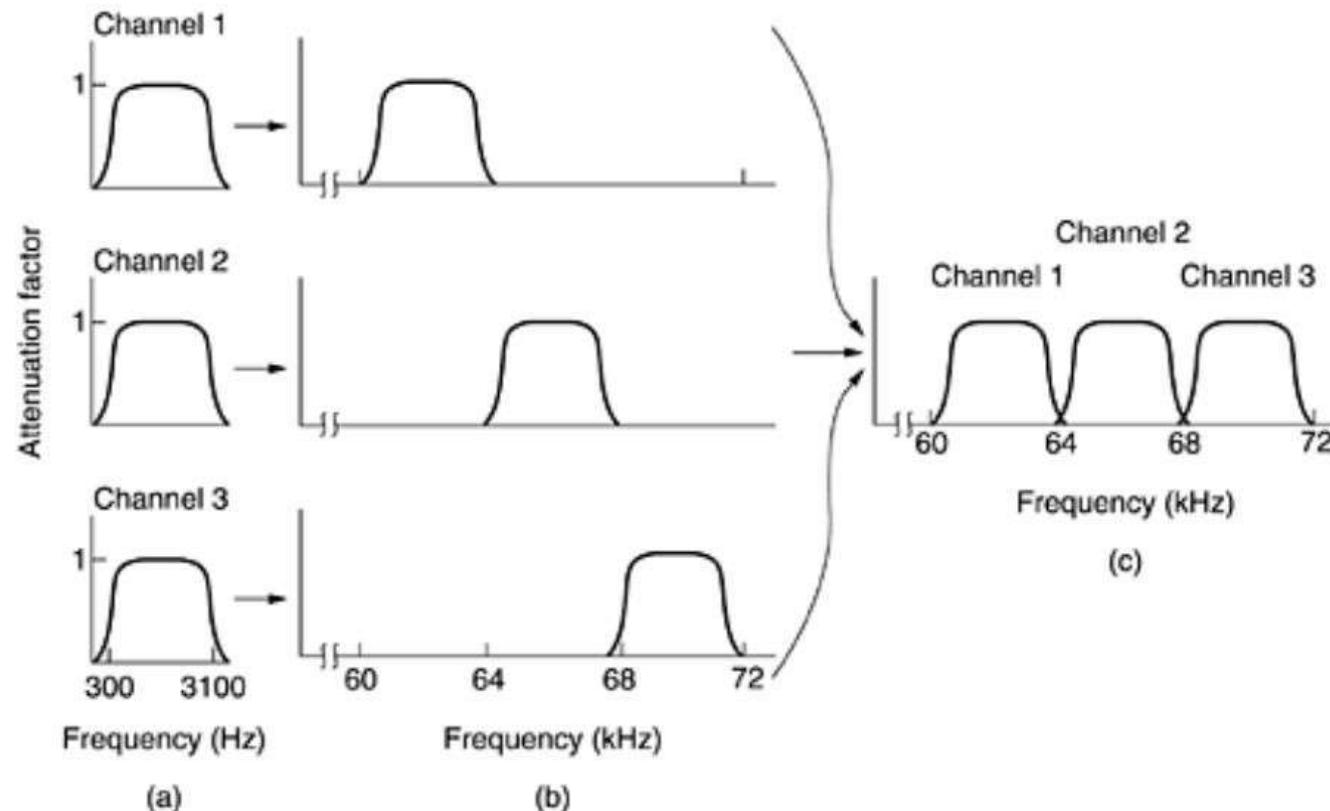
Multiplexing

- **Definition:** Multiplexing is a technique which combines multiple signals into one signal, suitable for transmission over a communication channel such as coaxial cable or optical fiber.
- **By doing multiplexing,** large amount bandwidth can be saved, cost can be reduced, circuit complexity can be reduced and multiple signals can be sent simultaneously over a single communication channel.
- **Analog:** Frequency Division Multiplexing and Wavelength Division Multiplexing
- **Digital:** Time Division Multiplexing

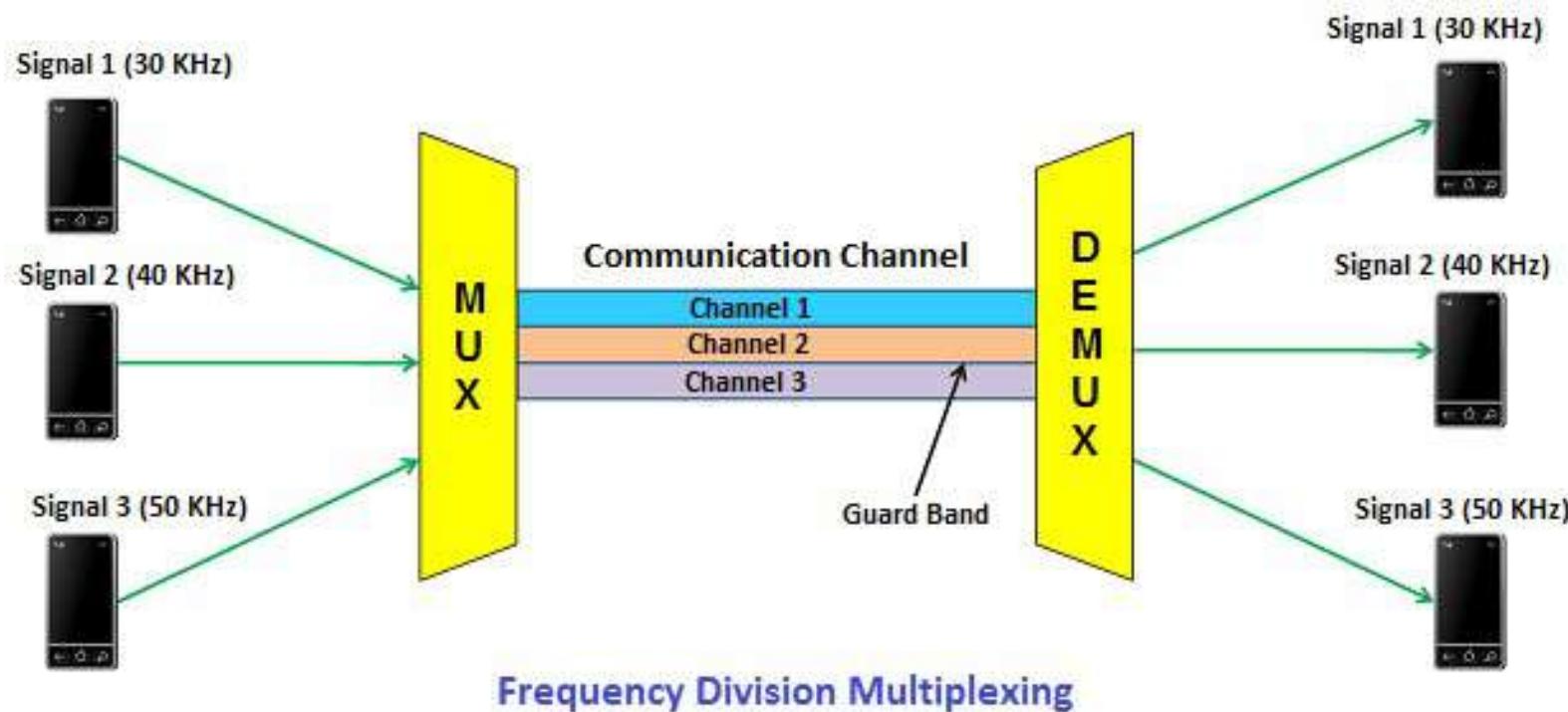
Frequency Division Multiplexing (FDM)

- popular multiplexing technique in TV and radio.
- combines multiple signals into one signal → transmitted over the communication channel.
- bandwidth of the communication channel should be greater than the combined bandwidth of individual signals.
- divides the bandwidth of a **channel into several logical sub-channels** and each logical sub-channel is separated by an unused bandwidth called Guard Band to prevent overlapping of signals.
- A guard band is a narrow frequency range that separates two signal frequencies.

FDM Operation



FDM Operation



FDM

- **Advantages of Frequency Division Multiplexing (FDM)**

- It transmits multiple signals simultaneously.
- In frequency division multiplexing, the demodulation process is easy.
- It does not need Synchronization between transmitter and receiver.

- **Disadvantages of Frequency Division Multiplexing (FDM)**

- It needs a large bandwidth communication channel.

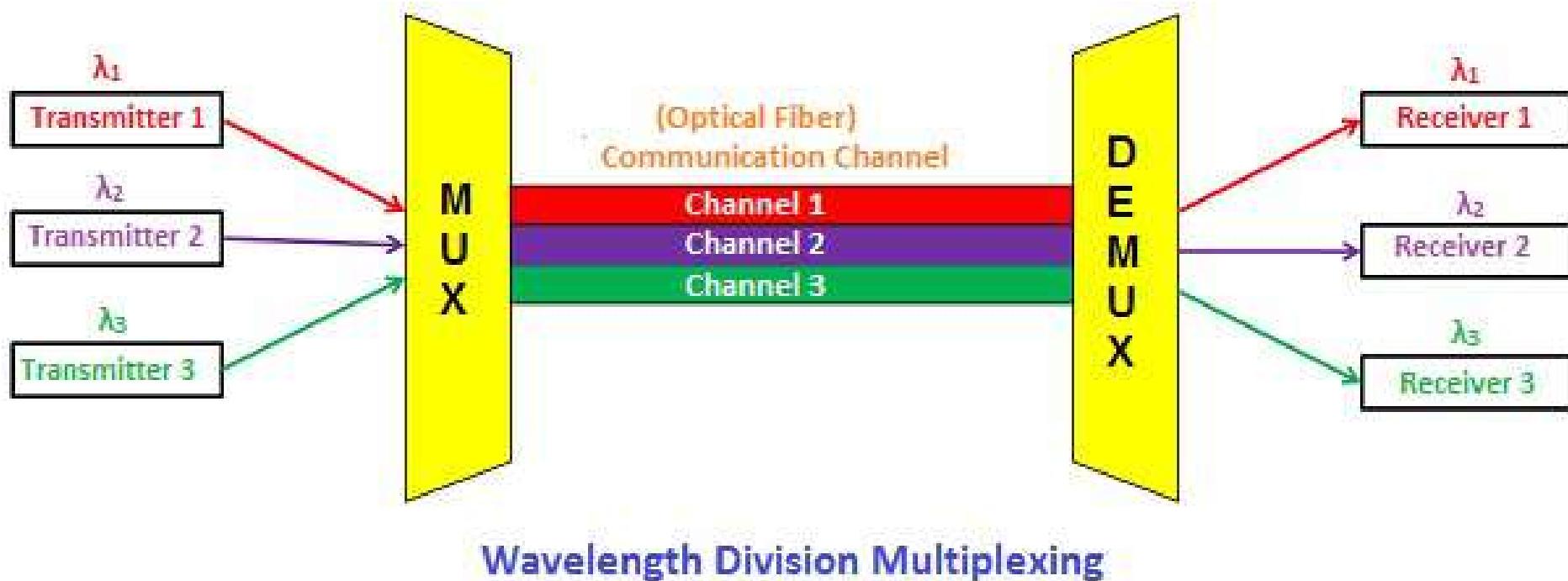
- **Applications of Frequency Division Multiplexing (FDM)**

- Frequency division multiplexing is used for FM and AM radio broadcasting.
- It is used in first generation cellular telephone.
- It is used in television broadcasting.

Wavelength Division Multiplexing (WDM)

- Wavelength division multiplexing is a technology that increases the bandwidth of a communication channel (optical fiber) by simultaneously allowing multiple optical signals through it.
- the working principle of wavelength division multiplexing is similar to frequency division multiplexing. The only difference is in wavelength division multiplexing optical signals are used instead of electrical signals.
- The main advantage of WDM system is that only need to upgrade the multiplexer and demultiplexer at each end; no need to buy more fibers which are more expensive.

WDM



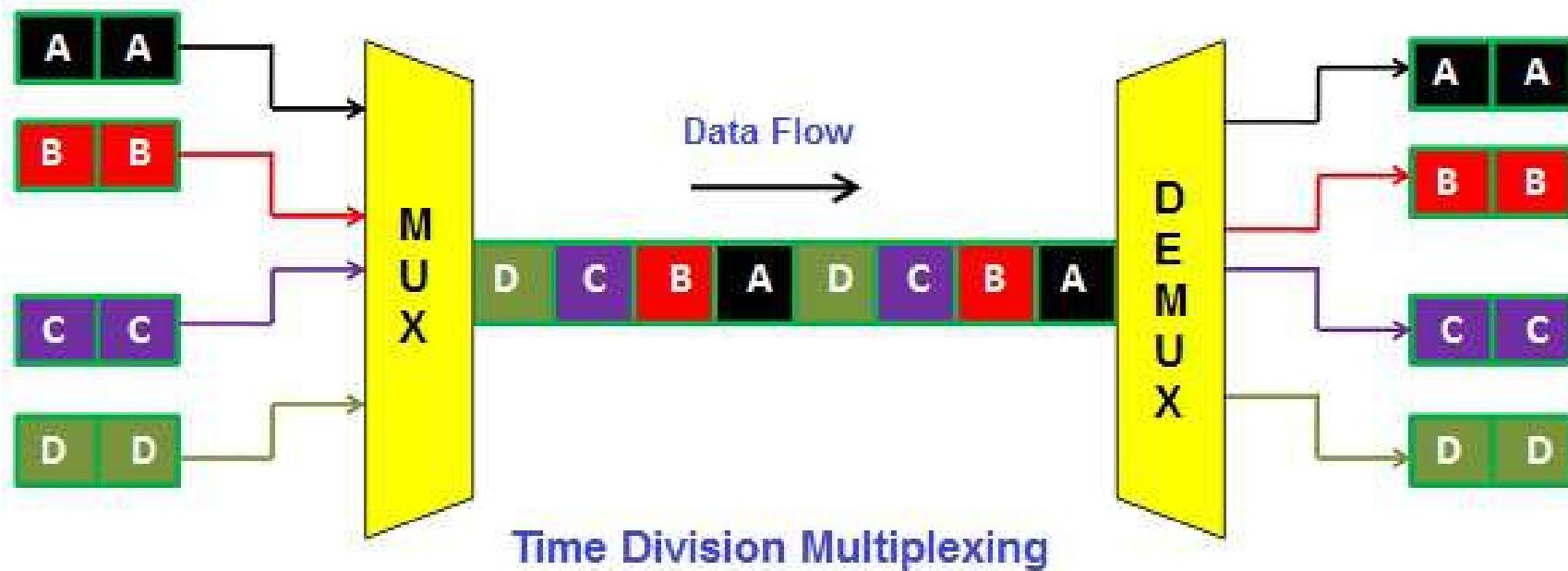
WDM

- WDM techniques are of two types:
 - Dense Wavelength Division Multiplexing (longer distances)
 - Coarse Wavelength Division Multiplexing (Shorter distances)
- **Advantages of Wavelength Division Multiplexing (WDM)**
 - WDM allows transmission of data in two directions simultaneously
 - Low cost
 - Greater transmission capacity
 - High security
 - Long distance communication with low signal loss

Time Division Multiplexing (digital)

- multiple signals are combined and transmitted one after another on the same communication channel.
- in time division multiplexing, all signals operate with the same frequency are transmitted at different times.

TDM

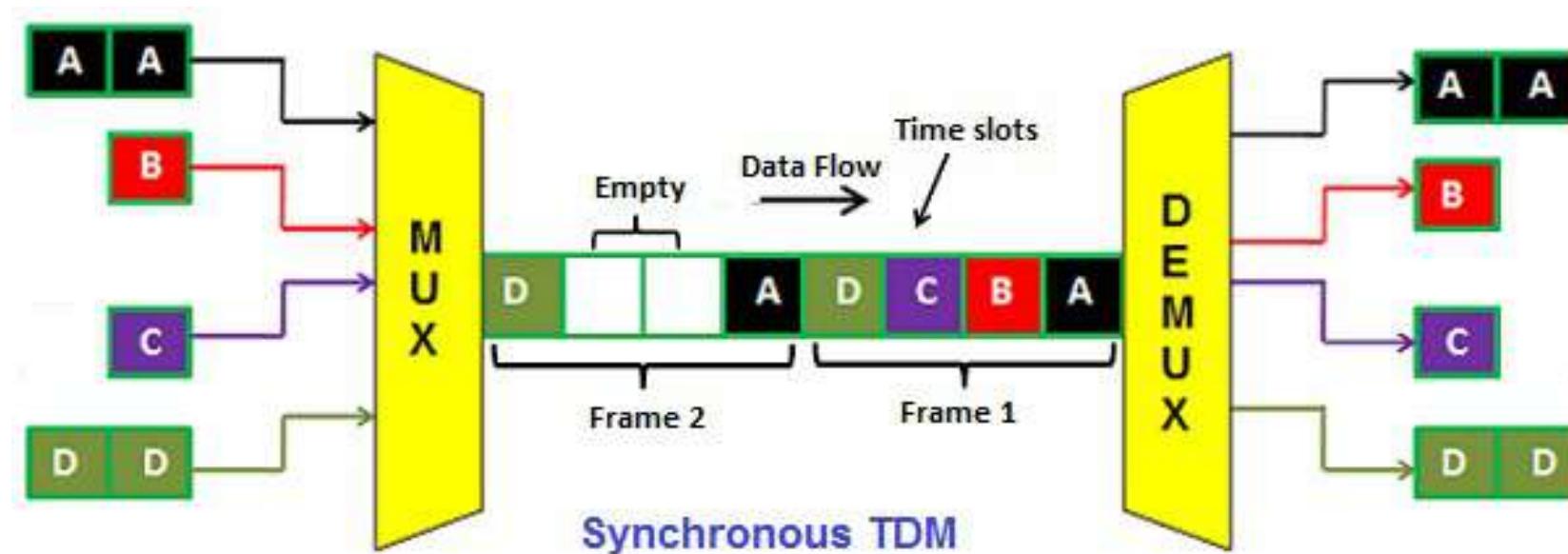


TDM

- Time Division Multiplexing is mainly classified into two types:
 - Synchronous TDM (fixed time slots)
 - Asynchronous TDM (no fixed time slots they are flexible).
- **Advantages of Time Division Multiplexing (TDM)**
 - Full bandwidth is utilized by a user at a particular time.
 - The time division multiplexing technique is more flexible than frequency division multiplexing.
 - In time division multiplexing, the problem of crosstalk is very less.
- **Disadvantages of Time Division Multiplexing (TDM)**
 - In time division multiplexing, synchronization is required.

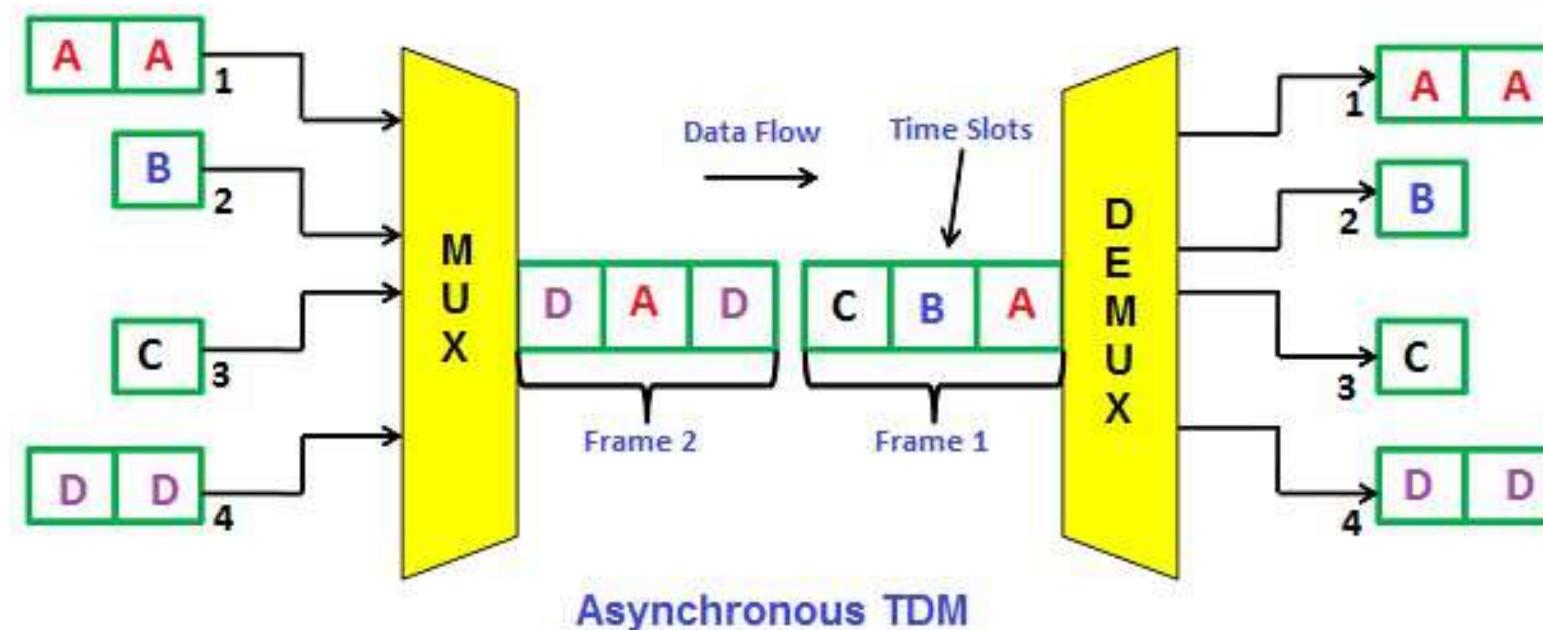
Synchronous TDM

synchronous TDM, the number of time slots is equal to the number of transmitters.



Asynchronous TDM

in Asynchronous TDM, the number of time slots is not equal to the number of devices (transmitters). The time slots in asynchronous TDM are always less than the number of devices (transmitter)





Computer Communication Networks

Introduction, Communication link, Multiplexing

Dr. Raja Vara Prasad
Assistant Professor
IIIT Sri City

CDMA—Code Division Multiple Access

Figure 2-45. (a) Binary chip sequences for four stations. (b) Bipolar chip sequences. (c) Six examples of transmissions. (d) Recovery of station C's signal.

A: 0 0 0 1 1 0 1 1
B: 0 0 1 0 1 1 1 0
C: 0 1 0 1 1 1 0 0
D: 0 1 0 0 0 0 1 0

(a)

A: (-1 -1 -1 +1 +1 -1 +1 +1)
B: (-1 -1 +1 -1 +1 +1 +1 -1)
C: (-1 +1 -1 +1 +1 +1 -1 -1)
D: (-1 +1 -1 -1 -1 +1 -1)

(b)

Six examples:

-- 1 -	C	$S_1 = (-1 +1 -1 +1 +1 +1 -1 -1)$
- 1 1 -	B + C	$S_2 = (-2 \ 0 \ 0 \ 0 +2 +2 \ 0 -2)$
1 0 --	A + B	$S_3 = (\ 0 \ 0 -2 +2 \ 0 -2 \ 0 +2)$
1 0 1 -	A + B + C	$S_4 = (-1 +1 -3 +3 +1 -1 -1 +1)$
1 1 1 1	A + B + C + D	$S_5 = (-4 \ 0 -2 \ 0 +2 \ 0 +2 -2)$
1 1 0 1	A + B + C + D	$S_6 = (-2 -2 \ 0 -2 \ 0 -2 +4 \ 0)$

(c)

two stations, A and C, both transmit a 1 bit at the same time that B transmits a 0 bit.

$$\begin{aligned}S_1 \cdot C &= (1 +1 +1 +1 +1 +1 +1)/8 = 1 \\S_2 \cdot C &= (2 +0 +0 +0 +2 +2 +0 +2)/8 = 1 \\S_3 \cdot C &= (0 +0 +2 +2 +0 -2 +0 -2)/8 = 0 \\S_4 \cdot C &= (1 +1 +3 +3 +1 -1 +1 -1)/8 = 1 \\S_5 \cdot C &= (4 +0 +2 +0 +2 +0 -2 +2)/8 = 1 \\S_6 \cdot C &= (2 -2 +0 -2 +0 -2 -4 +0)/8 = -1\end{aligned}$$

(d)

If the received chip sequence is S and the receiver is trying to listen to a station whose chip sequence is

$$S \bullet C = (A + \bar{B} + C) \bullet C = A \bullet C + \bar{B} \bullet C + C \bullet C = 0 + 0 + 1 = 1$$

CDMA—Code Division Multiple Access

two stations, A and C, both transmit a 1 bit at the same time that B transmits a 0 bit.

If the received chip sequence is S and the receiver is trying to listen to a station whose chip sequence is C

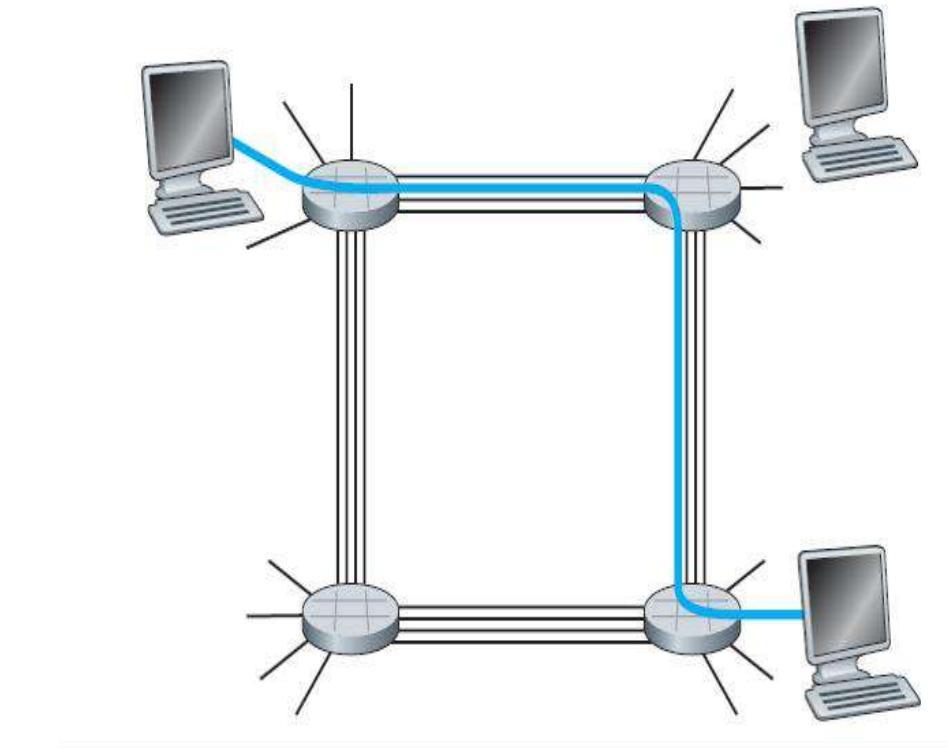
$$S \bullet C = (A + \bar{B} + C) \bullet C = A \bullet C + \bar{B} \bullet C + C \bullet C = 0 + 0 + 1 = 1$$

How are the end systems connected

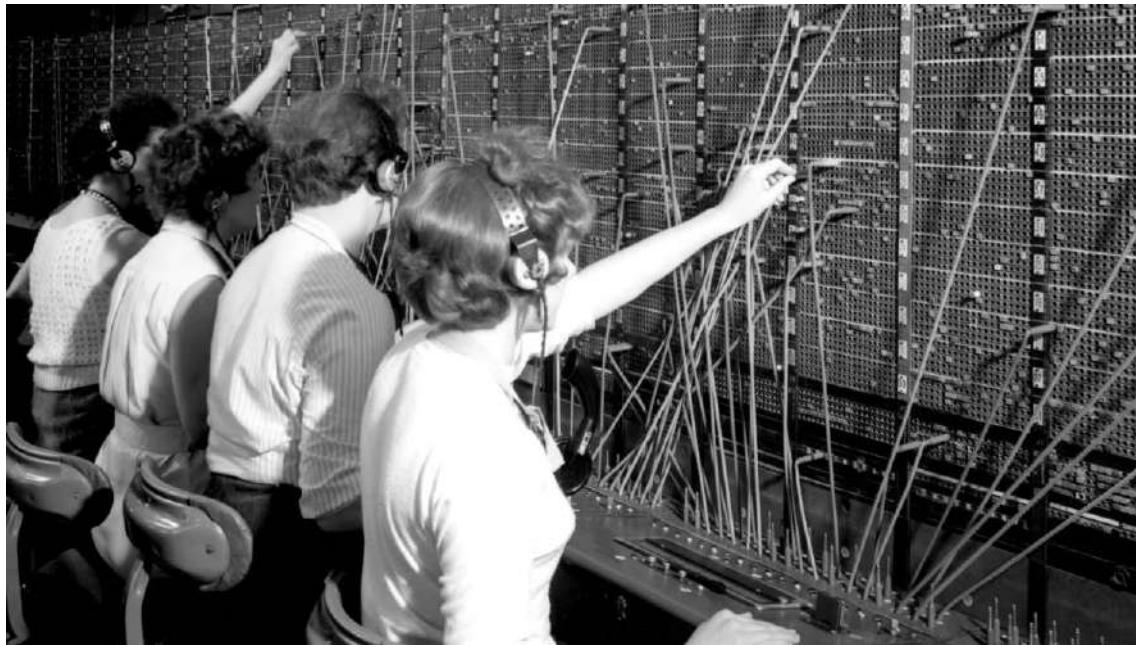
- Circuit switching
 - A dedicated path from source to destination
 - Resources on the path are reserved for the source-destination pair
- Packet switching
 - No dedicated path from source to destination
 - A switch/router forwards packets to another router / destination on the path.

Circuit switching

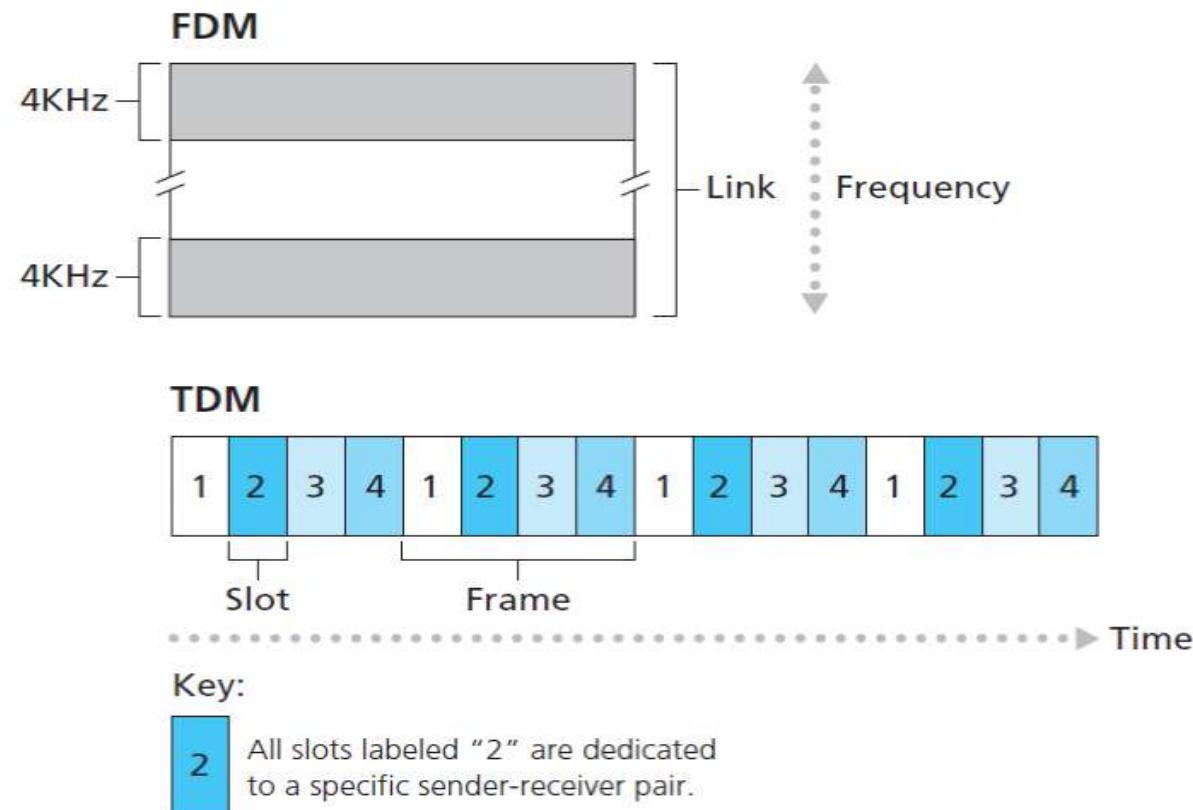
- The network establishes a connection from source to its destination. This connection is called **circuit**.
- Resources such as bandwidth, buffers on the circuit are blocked for the duration of communication.
- Telephone network is a circuit switching network.
- Links are finite, so very few users can be supported simultaneously.



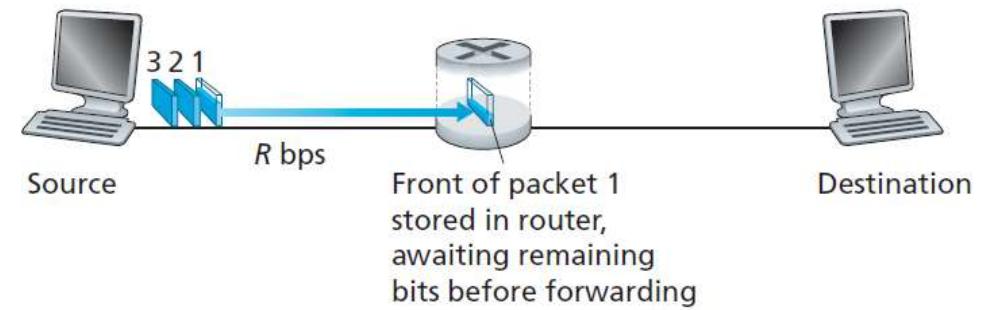
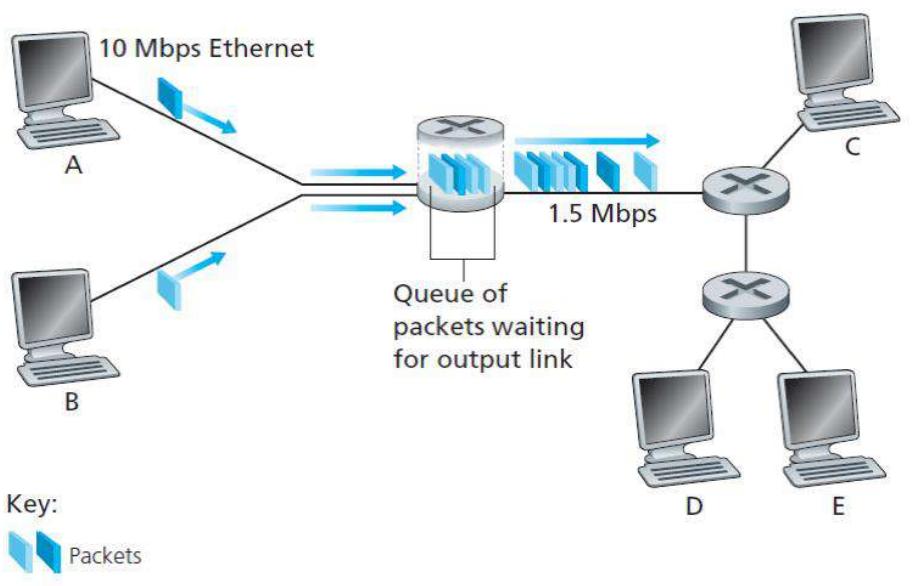
Circuit switching



Multiplexing in circuit switching



Packet switching



Statistical multiplexing

- Suppose users share a 1Mbps link.
- A user can be active or inactive. User will generate 100Kbps when active and we assume that a user is active for 10% of the time.
- Circuit switching : 100Kbps must be reserved for each user all the time, can support 10 users simultaneously!
- Circuit switching with TDM:
 - Say, one-second frame is divided into 10 frames each of 100ms.
 - Only 10 simultaneous connections are supported!!!

Statistical multiplexing

- **Packet switching:** Let there be 35 users in the system. What is the probability that 11 or more users are active simultaneously?
 - Approximately 0.0004
- As the probability of more than 10 users being active simultaneously is small, **Packet switching can support 35 users!**
- Packet switching allocate links on demand
- On demand allocation of resources is referred to as **Statistical multiplexing**.

Circuit switching vs Packet switching

Circuit switching

- Waste of bandwidth in silent periods
- Expensive
- Supports less number of connections
- Suitable for real-time services (video conferencing, etc)

Packet switching

- Effective use of bandwidth
- Cheaper than circuit switched network
- Supports more simultaneous connections
- Queuing delays
- Packet loss
- Not suitable for delay constrained applications

Layered Network Architecture

Why Layered Architecture?

- Organizing a network is a **big and complicated task**.
- Divide and conquer
- Example: Organization of an institute
 - academic section
 - finance section
 - administration section
 - procurement section

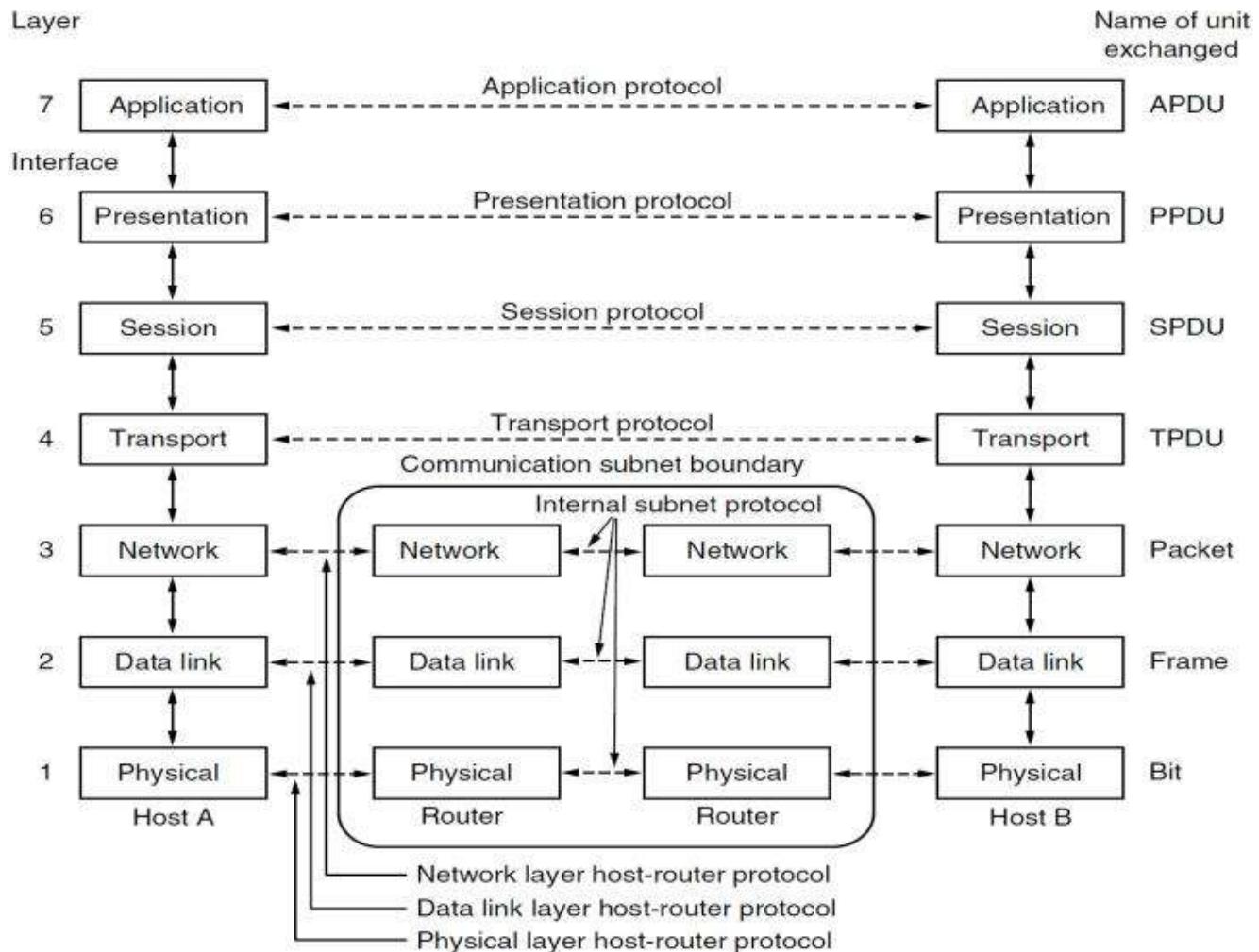
Advantages of Layered Architecture

- Divide the design issues into **small pieces**.
- A layer provides a **service** (set of actions) to the immediate higher layer.
- New technologies can be adopted in a layer without affecting other layers.
- Each layer can be analysed and tested independently.

Open System Interconnection (OSI) Reference Model

- Developed by International Organization for Standardization (ISO)
- 7-layer model:
 - Application layer
 - Presentation layer
 - Session layer
 - Transport layer
 - Network layer
 - Data-link layer
 - Physical layer

Layers



Application Layer

- Consists of user programs, network applications that does work at hand
- Examples:
 - File transfer, Remote login, Mail, Web access
- Protocols: FTP, Telnet, Simple Mail Transfer Protocol(SMTP), HTTP.

Presentation Layer

- Concerned with syntax and semantics of information transmitted
- Translation
- Encoding data: Data compression/conversion, encryption and decryption

Session Layer

- Allows to establish a session between peers
- Dialogue control: Session can allow bidirectional traffic or only unidirectional traffic.
- Token management: In some protocols, it is required that both sides do not attempt same operation at same time.
Session layer provides tokens to perform such actions
- Synchronization: Pausing and resuming a download.

Transport Layer

- Connection-oriented services to applications
 - flow control
 - guaranteed delivery of messages to destination
- Ensures data delivery is
 - error-free
 - in sequence
 - no loss, duplication and corruption of packets

Network Layer

- Interface between host and network
- Routing
- Congestion and deadlock
- Internetworking

Data-Link Layer and Physical Layer

- **Data-link layer**
 - Takes packet from network layer and moves it to the next router
 - error-free delivery: computes error detection information
- **Physical layer**
 - Controls transmission into the network cable.
 - Defines electrical signals.

Internet Protocol Stack

- Application layer
- Transport layer
- Network layer
- Data-link layer
- Physical layer

Encapsulation

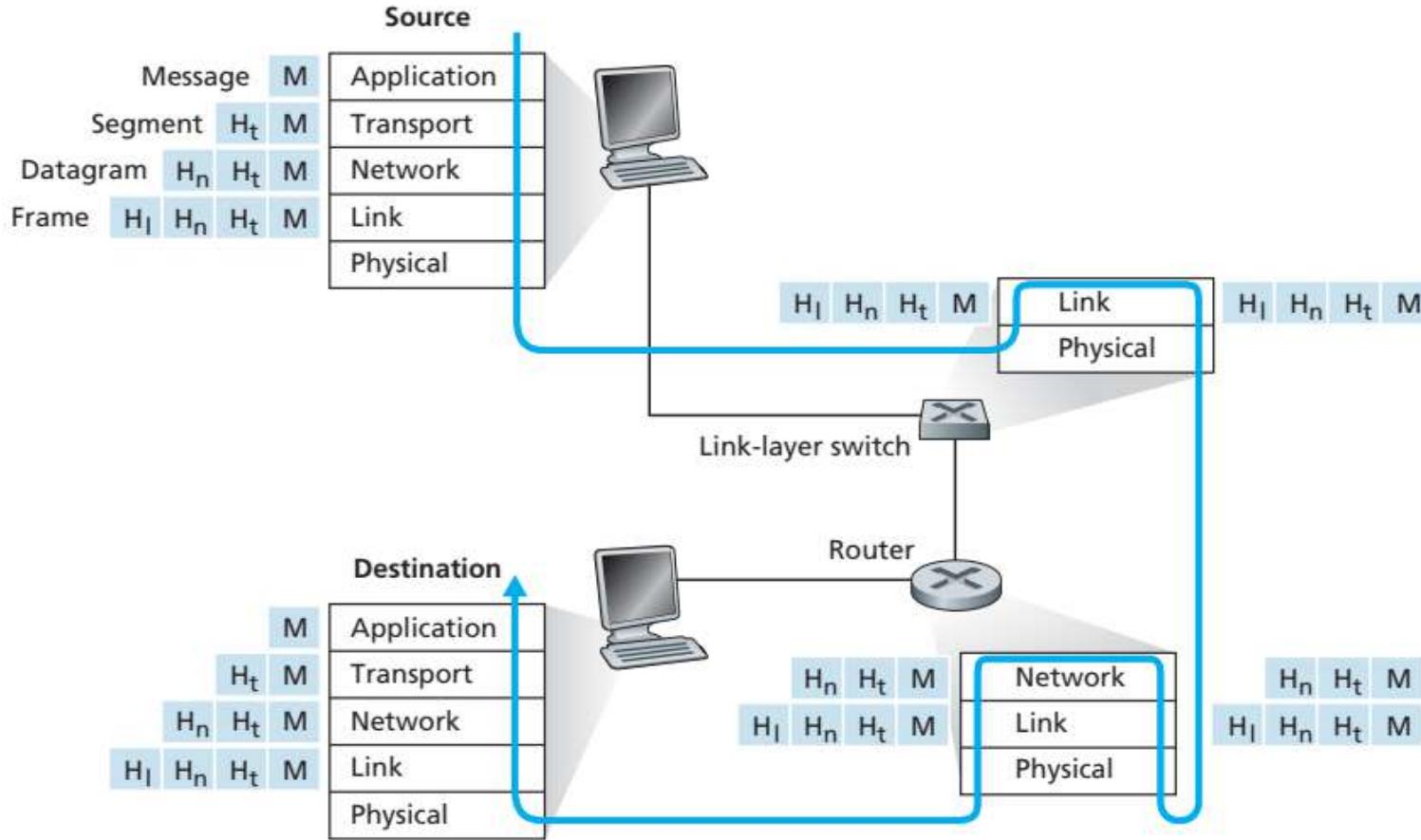


Figure 1.24 ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality



Computer Communication Networks

Introduction, Communication link, Multiplexing

Dr. Raja Vara Prasad
Assistant Professor
IIIT Sri City

Delays in Packet Switched Networks

- Packets travel from source to destination via intermediate routers/switches.
 - Processing delay
 - Queueing delay
 - Transmission delay
 - Propagation delay
- **Nodal delay** = Processing delay + Queuing delay + Transmission delay + Propagation delay

Processing Delay

- Time required to **examine** the packets header
 - Determines where to direct the packet
 - Check for errors
- Order of microseconds

Queuing Delay

- If a router is **busy** in processing and transmitting a packet, a freshly arrived packet has to wait in **queue** (buffer) for its turn.
- No queuing delay if the router is idle.
- Queuing delay varies with time and location. In general, it is a random variable.
- Order of microseconds to milliseconds.

Transmission Delay

- Time required to **push** the packet into the link
- If the length of the packet is L bits and transmission rate of the link is R bps, then

$$\text{Transmission delay} = \frac{L}{R}$$

- Order of microseconds to milliseconds

Propagation Delay

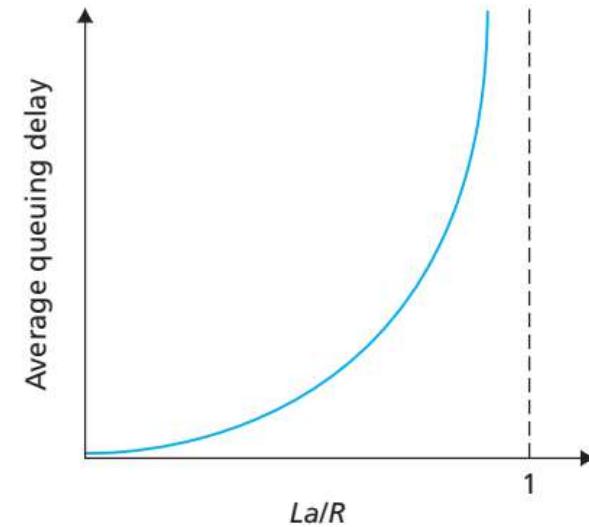
- Time required to **propagate** from one end of the link to the other end
- The propagation speed depends on the physical link between the routers
- In general, propagation speed s , is in the order of $2 \times 10^8 - 3 \times 10^8 \text{ m/s}$.
- Propagation speed depends on the distance bewteen the routers, d
- Propagation delay = $\frac{d}{s}$

Traffic Intensity

- Queuing delays are **random** in nature
- Arrivals to a queue are also **random** in nature
- Traffic intensity is an indication of queuing delay
- Let a be the average number of packets arriving at a queue
- Each packet is of length L bits adn transmission rate is R bps
- Traffic intensity** = $\frac{La}{R}$

Traffic Intensity

- If traffic intensity > 1 , the *queuelength* increases to ∞
- It is desirable to have traffic intensity < 1 .
- If traffic intensity **close to 1**, there will be a significant queuing delay

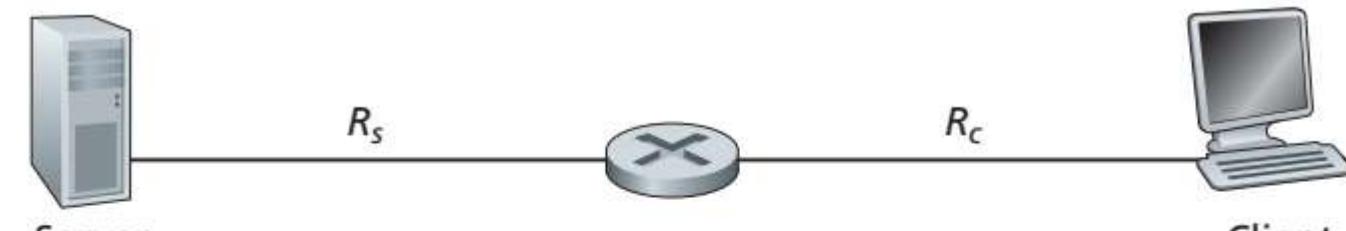


► Dependence of average queuing delay on traffic intensity

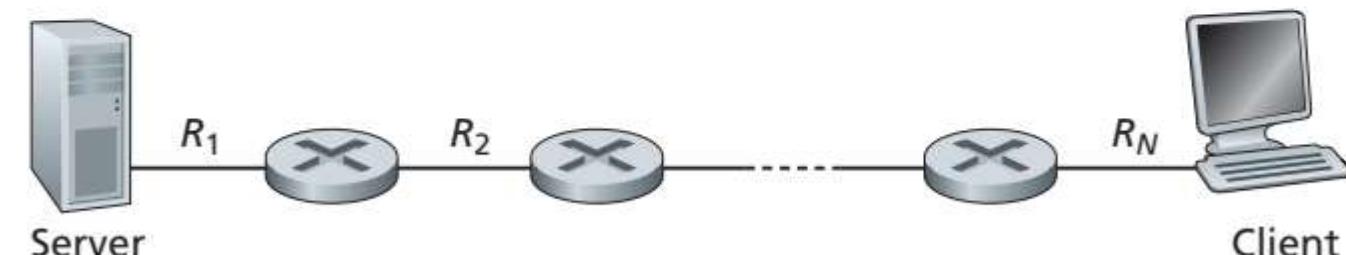
Throughput

- Suppose Host A is sending data to Host B across a computer network
- Instantaneous throughput is the rate at which Host B is receiving data
- Suppose it takes T seconds to transfer F bits from Host A to Host B, then average throughput = $\frac{F}{T}$ bps.

Throughput



a.



b.

Figure 1.19 ♦ Throughput for a file transfer from server to client

Throughput - Challenges

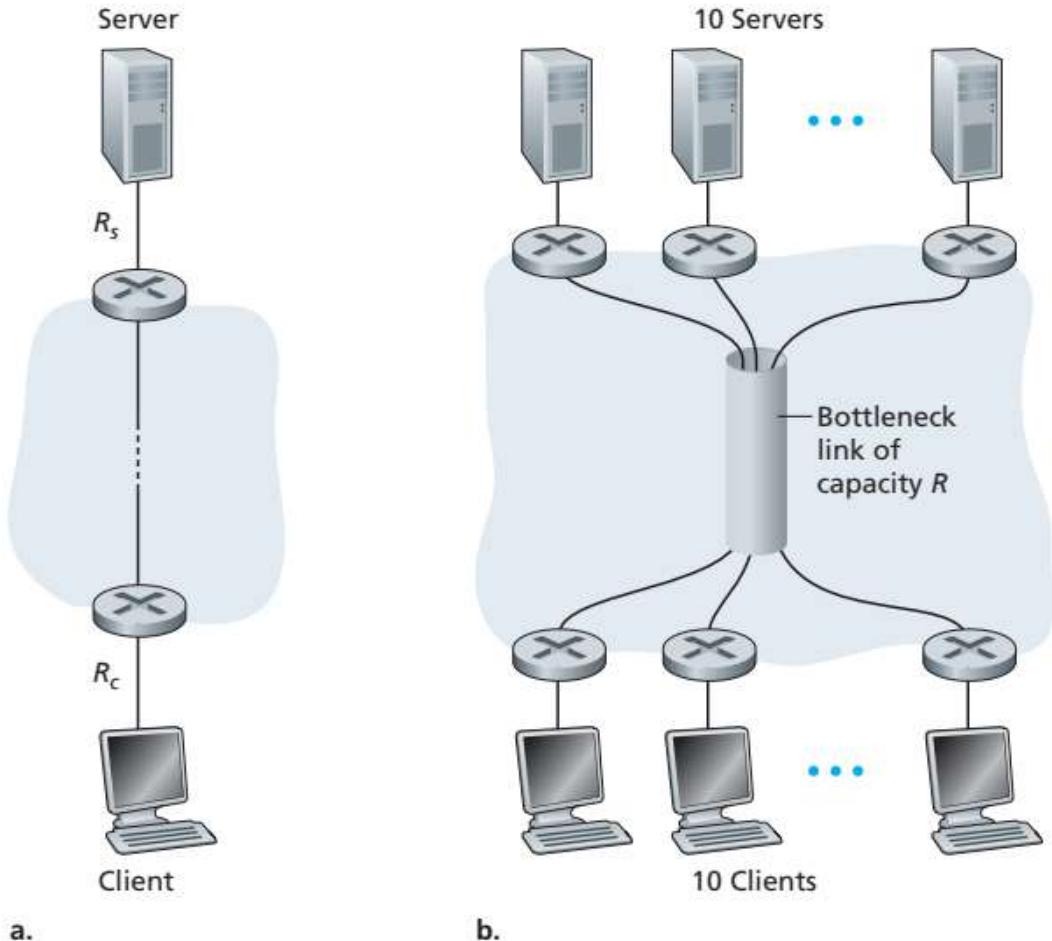


Figure 1.20 ♦ End-to-end throughput: (a) Client downloads a file from server; (b) 10 clients downloading with 10 servers

Case-a:

R_s is large—say a hundred times larger than both R_s and R_c —then the throughput for each download will once again be $\min\{R_s, R_c\}$.

Case-b:

Suppose $R_s = 2$ Mbps, $R_c = 1$ Mbps, $R = 5$ Mbps,

- common link divides its transmission rate equally among the 10 downloads.
- Then the bottleneck for each download is no longer in the access network
- instead the shared link in the core, which only provides each download with 500 kbps of throughput.

the end-to-end throughput for each download is now reduced to 500 kbps

Tutorial – Problems

1. Suppose users share a 2 Mbps link. Also suppose each user transmits continuously at 1 Mbps when transmitting, but each user transmits only 20 percent of the time.
- a. When circuit switching is used, how many users can be supported?
 - b. For the remainder of this problem, suppose packet switching is used. Why will there be essentially no queuing delay before the link if two or fewer users transmit at the same time? Why will there be a queuing delay if three users transmit at the same time?
 - c. Find the probability that a given user is transmitting.
 - d. Suppose now there are three users. Find the probability that at any given time, all three users are transmitting simultaneously. Find the fraction of time during which the queue grows.

Problem 3

Suppose N packets arrive simultaneously to a link at which no packets are currently being transmitted or queued. Each packet is of length L bits and the link has a transmission rate of R bits/sec. What is the average queueing delay for the N packets ?

Problems

- 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 $R_1 = 500\text{ kbps}$, $R_2 = 2\text{ Mbps}$, $R_3 = 1\text{ Mbps}$.
 - Assuming no other traffic, what is the throughput for the file transfer
 - Suppose the file size is 4 million bytes, how long will it take to transfer the file from A to B?
- How long does it take for a packet of length 1000 bytes to propagate over a link of propagation speed $2.5 \times 10^8 \text{ m/s}$. Length of the link is 2,500 Km and transmission rate is 2Mbps.

- P6. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B.
- Express the propagation delay, d_{prop} , in terms of m and s .
 - Determine the transmission time of the packet, d_{trans} , in terms of L and R .
 - Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.
 - Suppose Host A begins to transmit the packet at time $t = 0$. At time $t = d_{\text{trans}}$, where is the last bit of the packet?
 - Suppose d_{prop} is greater than d_{trans} . At time $t = d_{\text{trans}}$, where is the first bit of the packet?
 - Suppose d_{prop} is less than d_{trans} . At time $t = d_{\text{trans}}$, where is the first bit of the packet?
 - Suppose $s = 2.5 \cdot 10^8$, $L = 120$ bits, and $R = 56$ kbps. Find the distance m so that d_{prop} equals d_{trans} .

- P12. A packet switch receives a packet and determines the outbound link to which the packet should be forwarded. When the packet arrives, one other packet is halfway done being transmitted on this outbound link and four other packets are waiting to be transmitted. Packets are transmitted in order of arrival. Suppose all packets are 1,500 bytes and the link rate is 2 Mbps. What is the queuing delay for the packet? More generally, what is the queuing delay when all packets have length L , the transmission rate is R , x bits of the currently-being-transmitted packet have been transmitted, and n packets are already in the queue?

Traceroute

- program that can run in any Internet host
- When the user specifies a destination hostname, the program in the source host sends multiple, special packets toward that destination
- packets work their way toward the destination, they pass through a series of routers
- router receives one of these special packets, it sends back to the source a short message that contains the name and address of the router
- source will send N special packets into the network, with each packet addressed to the ultimate destination
- source records the time that elapses between when it sends a packet and when it receives the corresponding return message
- the source can reconstruct the route taken by packets flowing from source to destination, and the source can determine the round-trip delays to all the intervening routers

```
1 cs-gw (128.119.240.254) 1.009 ms 0.899 ms 0.993 ms
2 128.119.3.154 (128.119.3.154) 0.931 ms 0.441 ms 0.651 ms
3 border4-rt-gi-1-3.gw.umass.edu (128.119.2.194) 1.032 ms 0.484 ms 0.451 ms
4 acr1-ge-2-1-0.Boston.cw.net (208.172.51.129) 10.006 ms 8.150 ms 8.460 ms
5 agr4-loopback.NewYork.cw.net (206.24.194.104) 12.272 ms 14.344 ms 13.267 ms
6 acr2-loopback.NewYork.cw.net (206.24.194.62) 13.225 ms 12.292 ms 12.148 ms
7 pos10-2.core2.NewYork1.Level3.net (209.244.160.133) 12.218 ms 11.823 ms 11.793 ms
8 gige9-1-52.hsipaccess1.NewYork1.Level3.net (64.159.17.39) 13.081 ms 11.556 ms 13.297 ms
9 p0-0.polyu.bbnplanet.net (4.25.109.122) 12.716 ms 13.052 ms 12.786 ms
10 cis.poly.edu (128.238.32.126) 14.080 ms 13.035 ms 12.802 ms
```

Networks Under Attack

- “attempt to wreak havoc in our daily lives by damaging our Internet-connected computers, violating our privacy, and rendering inoperable the Internet services on which we depend”

malicious stuff—collectively known as **malware**—that can enter and infect devices

- deleting our files.
- installing spyware that collects our private information, such as social security numbers, passwords, and keystrokes
- sends this over the Internet back to attacker

compromised host may also be enrolled in a network of thousands of similarly compromised devices, collectively known as a **botnet**

- ✓ **self-replicating**
- ✓ **Viruses** are malware that require some form of user interaction to infect the user’s device
- ✓ **Worms** are malware that can enter a device without any explicit user interaction

Networks Under Attack

Denial-of-service (DoS) attacks:

- renders a network, host, or other piece of infrastructure unusable by legitimate users
- *Vulnerability attack*: sending a few well-crafted messages to a vulnerable application or operating system running on a targeted host. The service can stop or, worse, the host can crash.
- *Bandwidth flooding*: sends a deluge of packets to the targeted host, so many packets that the target's access link becomes clogged, preventing legitimate packets from reaching the server.
- *Connection flooding*. The attacker establishes a large number of half-open or fully open TCP connections at the target host. The host stops accepting legitimate connections due to the bogus connections.
- **Distributed DoS (DDoS) attack**: leveraging botnets with thousands of comprised hosts; much harder to detect and defend against than a DoS attack from a single host.

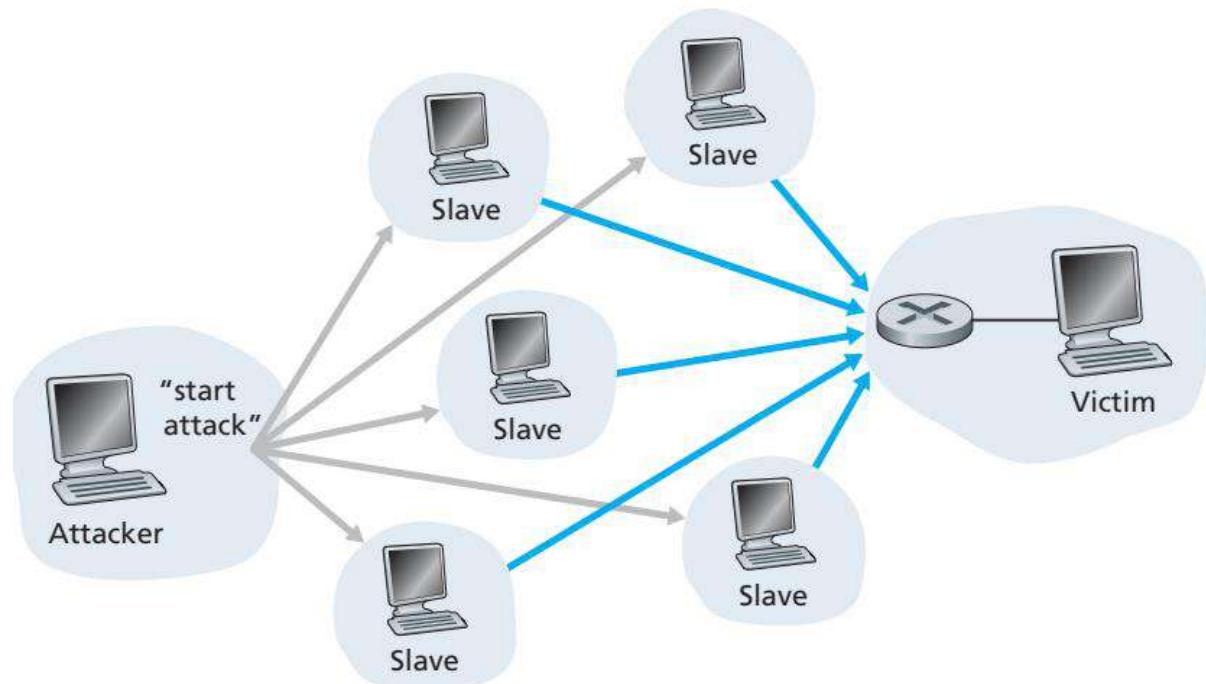
Networks Under Attack

Distributed DoS (DDoS) attack: leveraging botnets with thousands of comprised hosts

- much harder to detect and defend against than a DoS attack from a single host.

Packet Sniffers:

- placing a passive receiver in the vicinity of the wireless transmitter, that receiver can obtain a copy of every packet that is transmitted
- packets can contain all kinds of sensitive information, including passwords, social security numbers, trade secrets, and private personal messages.
- Sniffed packets can then be analyzed offline for sensitive information
- Wireshark: a packet sniffer
- packet sniffers are passive—do not inject packets into the channel—difficult to detect
- defenses against packet sniffing involve cryptography





Indian Institute of Information Technology, Sri City, Chittoor
(An Institute of National Importance under an Act of Parliament)

Computer Communication Networks

Application Layer

Dr. Raja Vara Prasad

Assistant Professor

IIIT Sri City

Application Layer

Network Applications

Network application development -- writing programs that run on different end systems and communicate with each other over the network

Example:

Web application → two distinct programs that communicate with each other:

- the browser program running in the user's host (desktop, laptop, tablet, smartphone, and so on);
- the Web server program running in the Web server host.
- in P2P file-sharing system there is a program in each host that participates in the file-sharing community

Network Applications

- do not need to write software that runs on network core devices, such as routers or link-layer switches
- Network core devices do not function at the application layer
- function at lower layers— specifically at the network layer and below

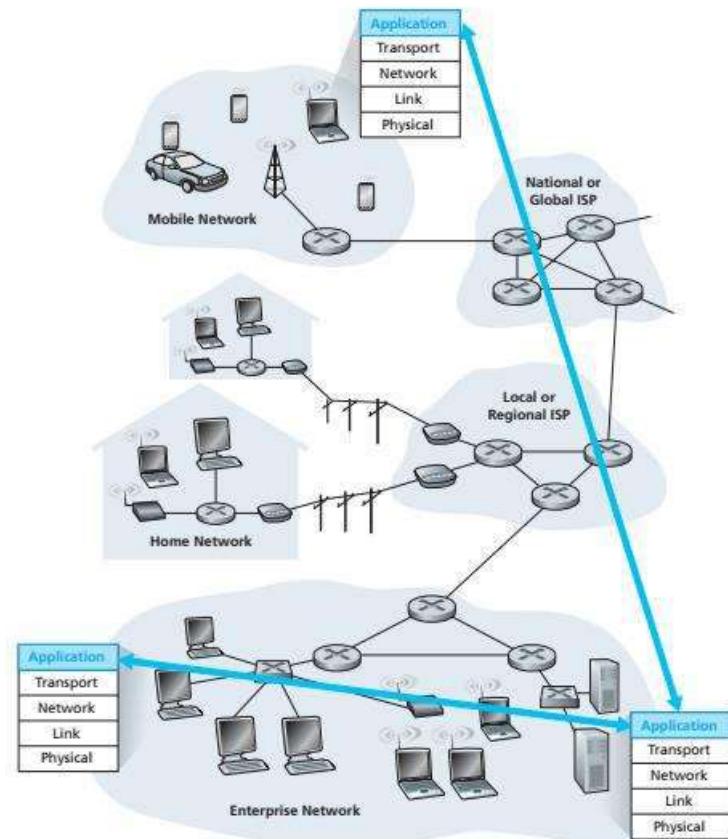


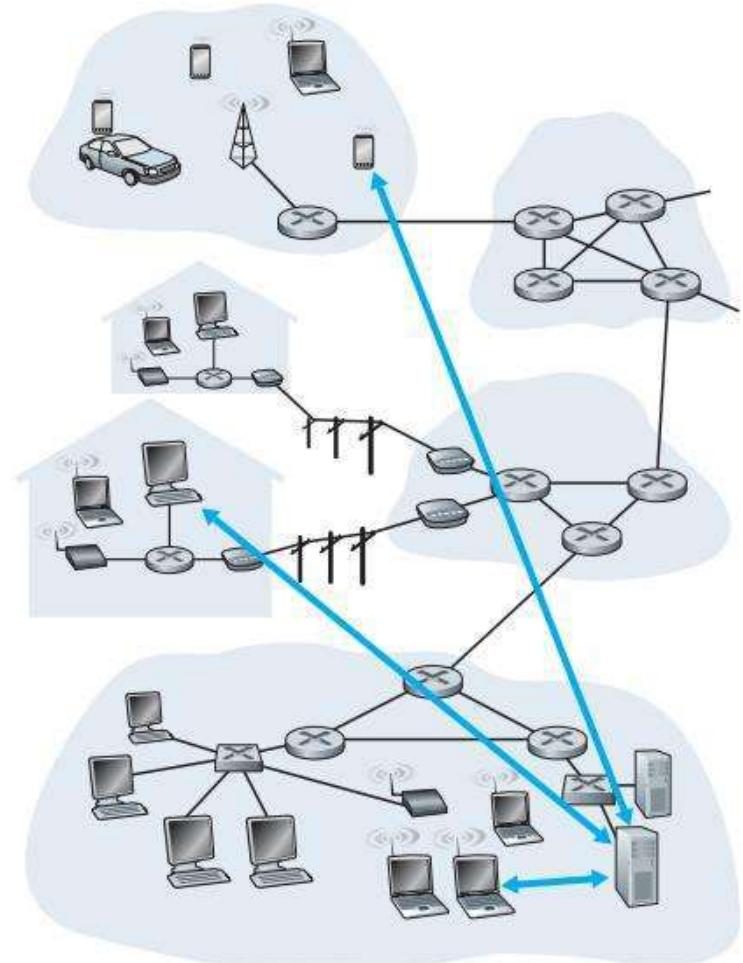
Figure 2.1 • Communication for a network application takes place between end systems at the application layer

Network Applications

- Applications use the services of network (Transport layer)
- For an application developer, architecture and services of network are fixed
- Architectures of applications:
 - Client-Server architecture
 - Peer-to-Peer (P2P) architecture
- Application developer decides on the architecture and services of transport layer to be used.

Client-Server Architecture

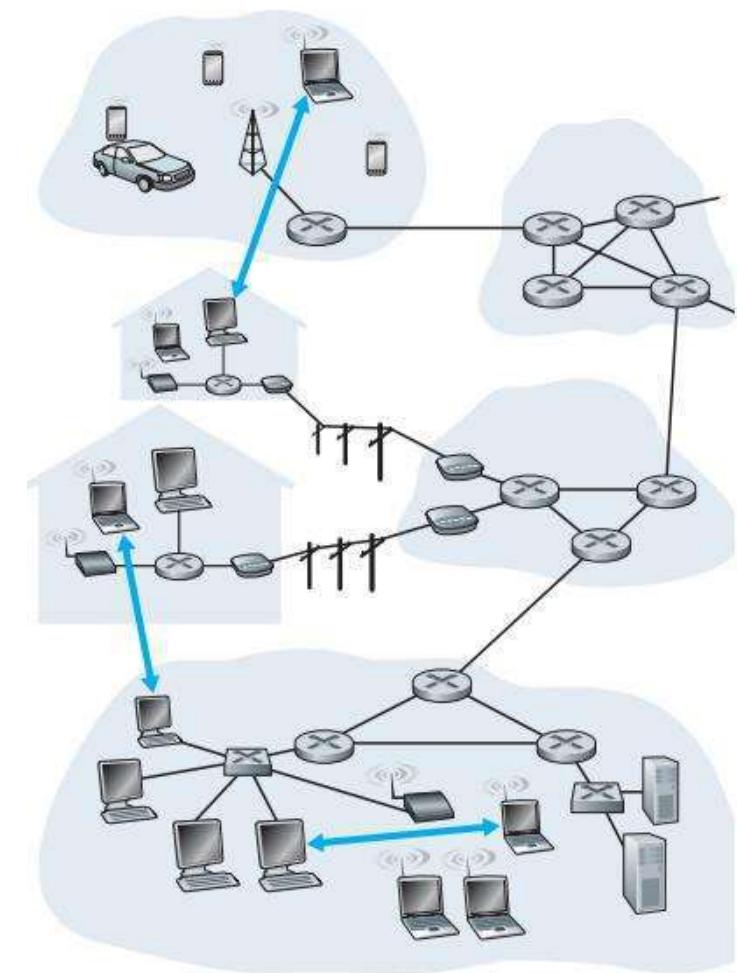
- Server: An end system that **serves the requests** from various hosts.
- A server is always **ON**.
- Client: An end system that **requests** a server for content.
- A client can be either **ON-OFF** or always **ON**.
- Example applications using this architecture: web, e-mail, file transfer, etc.



a. Client-server architecture

Peer-to-Peer Architecture

- End systems communicate by a direct connection.
- The end systems are called peers.
- Example applications: skype, internet telephony, torrents, etc
- Advantages:
 - File distribution
 - Self-scalable: can handle growth in traffic
 - Cost effective: no server infrastructure and server bandwidth.
- Challenges in P2P Architecture:
 - ISP friendly: asymmetric data traffic.
 - Security
 - Incentives: Peers should share bandwidth.



b. Peer-to-peer architecture

Processes Communicating

- A process is a program that is running within an end system.
- A client process is a process running on a client and a server process is process running on a server.
- It is the client process and server processes that are actually communicating.
- A process sends and receives messages to and from transport layer through a software interface known as **socket**.
- A socket is also known as **Application Programming Interface (API)**.

Interface Between the Process: API

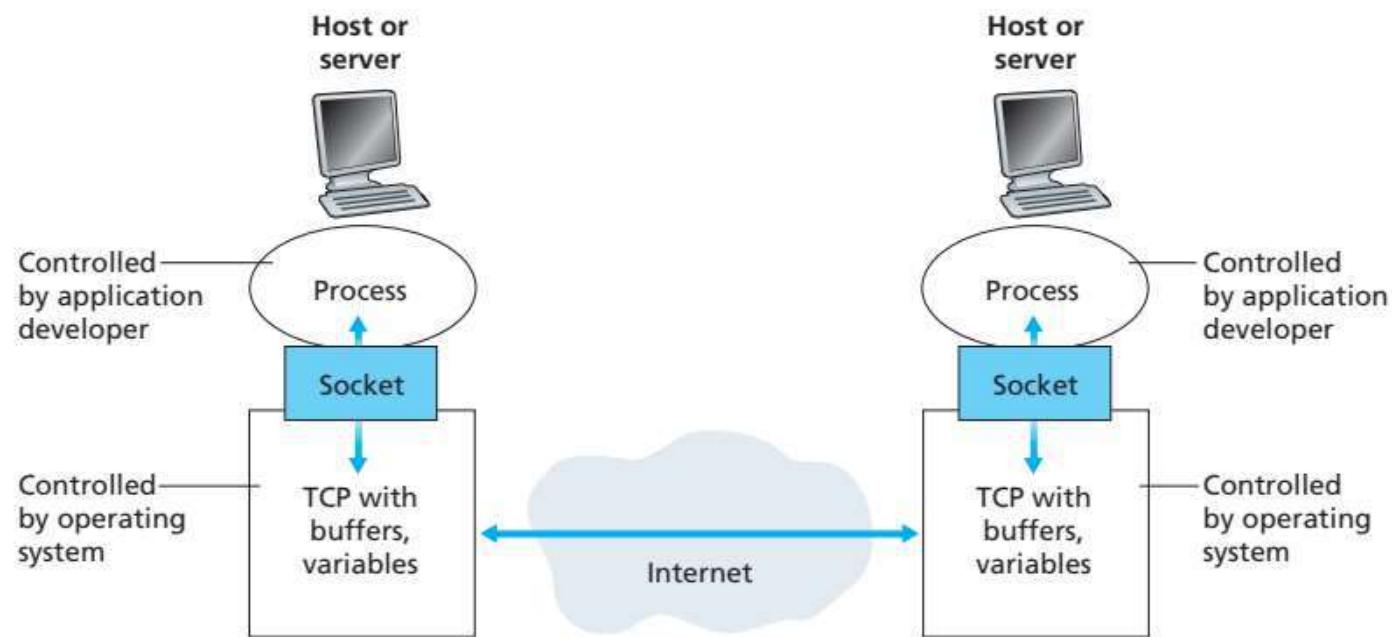


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

Services of Transport Layer

- **Reliable data transfer:** Guaranteed data delivery service.
- **Throughput**
- **Timing:** for example, it is guaranteed that a packet will be delivered no more than 100 msec later.
- **security:** end-point authentication, encryption and decryption.

Requirements of Applications

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–1 Mbps 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

Transport protocols

- Transmission Control Protocol (TCP)
 - Connection oriented service: handshaking, full-duplex connection
 - Reliable data transfer service: packets get delivered without error and in proper order.
 - Congestion control
- User Datagram Protocol (UDP)
 - Connectionless
 - Unreliable data transfer service.
 - No congestion control
- can often provide satisfactory service to time-sensitive applications,
- cannot provide any timing or throughput guarantees

Applications

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

Addressing Processes

- There are many processes running on a host, how to identify the destination process?
- We identify host by **IP address**.
- We identify processes by **port numbers!**
- For example, web server is identified by port number 80, mail server is identified by port number 25.

Application Layer - Introduction

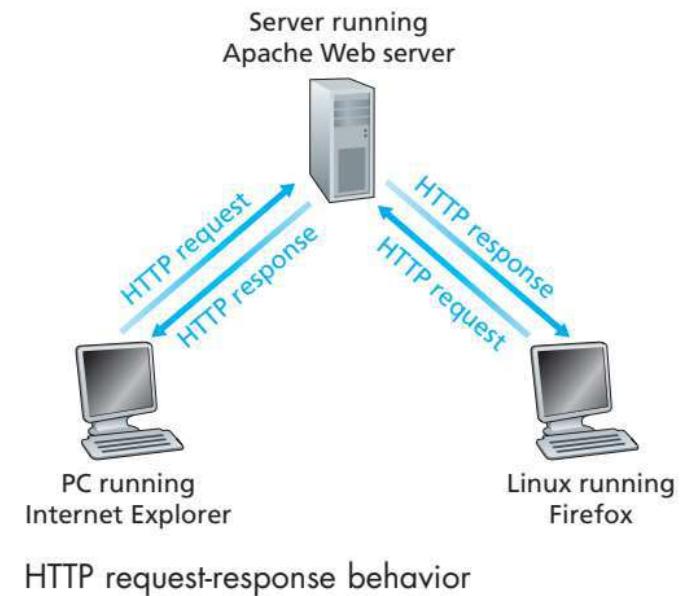
Application-layer protocol defines:

- The types of messages exchanged, for example, request messages and response messages
- The syntax of the various message types, such as the fields in the message and how the fields are delineated
- The semantics of the fields - meaning of the information in the fields
- Rules for determining when and how a process sends messages and responds to messages

Application-layer protocol is only one piece of a network application

Web and HTTP

- A web page is a document and consists of objects
- An object is nothing but a file such as HyperText Markup Language (HTML) file, an image file, applet or video clip.
- If a web page contains a basic html file and ten images, we say the web page contains 11 objects.
- HyperText Transfer Protocol (HTTP) is the web's application layer protocol
- HTTP uses client-server architecture with TCP.
- The client program and server program talk to each other by exchanging HTTP messages.



Uniform Resource Locator

- An object should be addressable by a URL.
- Each URL consists of hostname and objects path name
- For example, <http://www.iiits.ac.in/wp-content/uploads/2017/05/Untitled-design-15.png> is url for an image.
- www.iiits.ac.in is host name
- [wp-content/uploads/2017/05/Untitled-design-15.png](http://www.iiits.ac.in/wp-content/uploads/2017/05/Untitled-design-15.png) is path name.
- Client side of HTTP is implemented in Web browser and server side is implemented in Web server.
- Examples: Apache and Microsoft Internet Information server.

- ✓ base HTML file plus objects
- ✓ base HTML file references the other objects in the page with the objects' URLs.

- HTTP client initiates a connection with HTTP server (**handshaking**).
- Once the connection is established, client and server exchange messages through socket interface.
- Client sends an HTTP request and receives HTTP messages through its socket
- Server receives HTTP requests and sends HTTP responses through its socket interface.
- Client/server need not worry about packets (does not have any control) after sending through their socket.
- Server sends requested files without storing state information of client. Thus HTTP is a **stateless** protocol.

HTTP Connection

- Let us say, a web page has one html file and 10 images.
- How does client retrieve the web page?
- Nonpersistent and Persistent**
- Nonpersistent: one TCP connection for **each** file
- Persistent: one TCP connection for **all** files

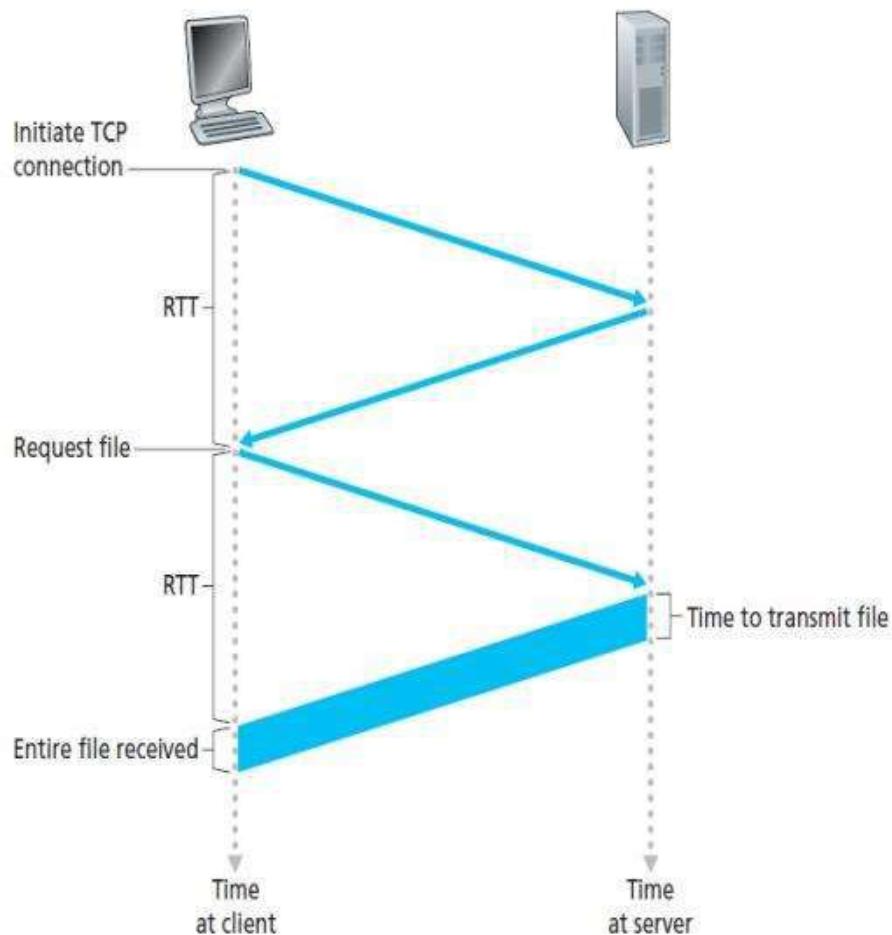
Nonpersistent Connection

- For each file:
 - HTTP client initiates a TCP connection to the server on port number 80
 - Client sends its HTTP request and it includes the path name to the file
 - HTTP server receives the request and retrieves the file and sends the HTTP response to the client
 - HTTP server tells TCP to close the connection.
- TCP connections can be **serial or parallel** depending on browser's configuration

Example: Non-Persistent

- steps of transferring a Web page from server to client for the case of non-persistent connections.
 - page consists of a base HTML file and 10 JPEG images → 11 objects reside on the same server.
 - URL for the base HTML file is:
<http://www.someSchool.edu/someDepartment/home.index>
1. The HTTP client process initiates a TCP connection to the server `www.someSchool.edu` on port number 80, which is the default port number for HTTP. Associated with the TCP connection, there will be a socket at the client and a socket at the server.
 2. The HTTP client sends an HTTP request message to the server via its socket. The request message includes the path name `/someDepartment/home.index`. (We will discuss HTTP messages in some detail below.)
 3. The HTTP server process receives the request message via its socket, retrieves the object `/someDepartment/home.index` from its storage (RAM or disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.
 4. The HTTP server process tells TCP to close the TCP connection. (But TCP doesn't actually terminate the connection until it knows for sure that the client has received the response message intact.)
 5. The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file. The client extracts the file from the response message, examines the HTML file, and finds references to the 10 JPEG objects.
 6. The first four steps are then repeated for each of the referenced JPEG objects.

Round-Trip Time



Persistent Connection

- Server leaves the connection after sending the HTTP response
- **Pipelining:** A browser can request for files without waiting for the reception of pending requests.
- TCP closes after some idle period
- Default mode HTTP: Persistent connection with pipelining.

HTTP Request Format

- HTTP request message:

```
GET /somedir/page.html HTTP/1.1
```

```
Host: www.iitm.ac.in
```

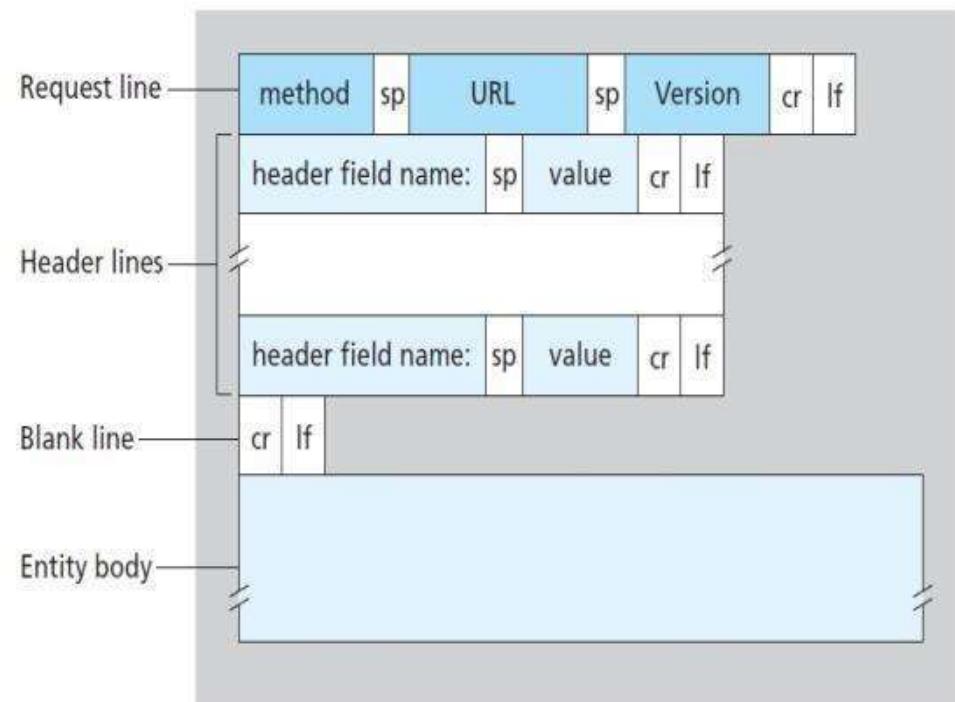
```
Connection: close
```

```
User-agent: Mozilla/4.0
```

```
Accept-language: En
```

- Methods: GET, PUT, POST, HEAD, DELETE

HTTP Request





Computer Communication Networks

Application Layer

Dr. Raja Vara Prasad

Assistant Professor

IIIT Sri City

Application Layer

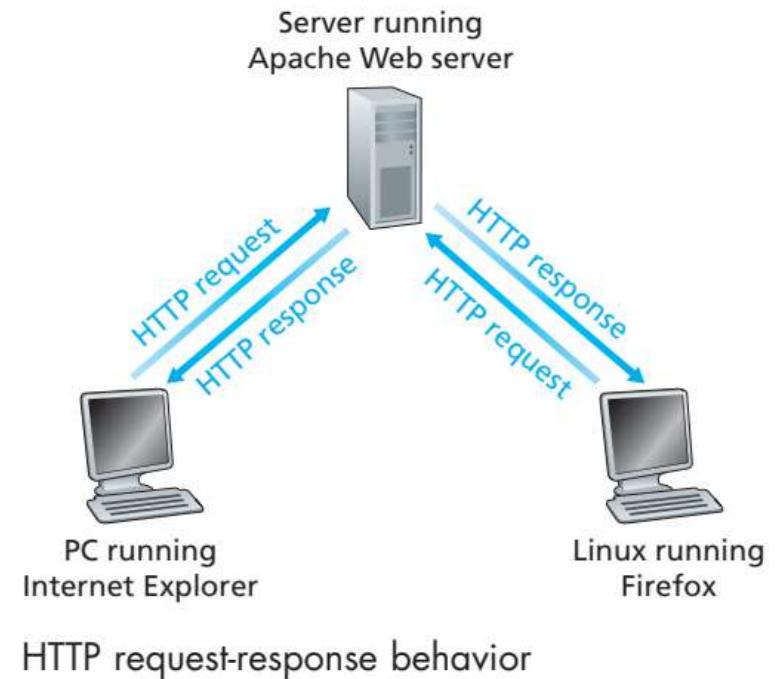
Application Layer - Introduction

Application-layer protocol defines:

- The types of messages exchanged, for example, request messages and response messages
- The syntax of the various message types, such as the fields in the message and how the fields are delineated
- The semantics of the fields - meaning of the information in the fields
- Rules for determining when and how a process sends messages and responds to messages

Application-layer protocol is only one piece of a network application

- A web page is a document and consists of objects
- An object is nothing but a file such as HyperText Markup Language (HTML) file, an image file, applet or video clip.
- If a web page contains a basic html file and ten images, we say the web page contains 11 objects.
- HyperText Transfer Protocol (HTTP) is the web's application layer protocol
- HTTP uses client-server architecture with TCP.
- The client program and server program talk to each other by exchanging HTTP messages.



Uniform Resource Locator

- An object should be addressable by a URL.
- Each URL consists of hostname and objects path name
- For example, <http://www.iiits.ac.in/wp-content/uploads/2017/05/Untitled-design-15.png> is url for an image.
- www.iiits.ac.in is host name
- [wp-content/uploads/2017/05/Untitled-design-15.png](http://www.iiits.ac.in/wp-content/uploads/2017/05/Untitled-design-15.png) is path name.
- Client side of HTTP is implemented in Web browser and server side is implemented in Web server.
- Examples: Apache and Microsoft Internet Information server.

- ✓ base HTML file plus objects
- ✓ base HTML file references the other objects in the page with the objects' URLs.

- HTTP client initiates a connection with HTTP server (**handshaking**).
- Once the connection is established, client and server exchange messages through socket interface.
- Client sends an HTTP request and receives HTTP messages through its socket
- Server receives HTTP requests and sends HTTP responses through its socket interface.
- Client/server need not worry about packets (does not have any control) after sending through their socket.
- Server sends requested files without storing state information of client. Thus HTTP is a **stateless** protocol.

- Let us say, a web page has one html file and 10 images.
- How does client retrieve the web page?
- Nonpersistent** and **Persistent**
- Nonpersistent: one TCP connection for **each** file
- Persistent: one TCP connection for **all** files

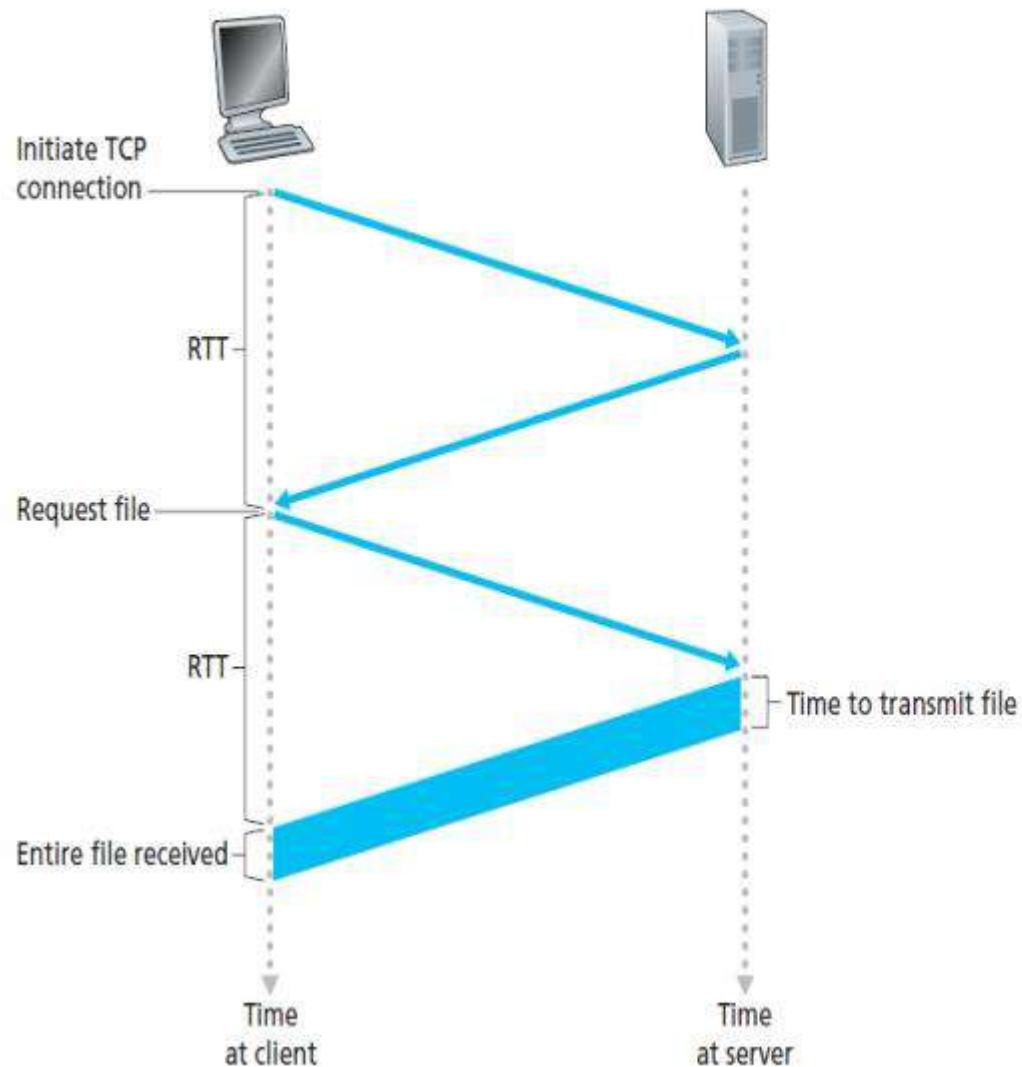
- For each file:
 - HTTP client initiates a TCP connection to the server on port number 80
 - Client sends its HTTP request and it includes the path name to the file
 - HTTP server receives the request and retrieves the file and sends the HTTP response to the client
 - HTTP server tells TCP to close the connection.
- TCP connections can be **serial or parallel** depending on browser's configuration

Example: Non-Persistent

- steps of transferring a Web page from server to client for the case of non-persistent connections.
- page consists of a base HTML file and 10 JPEG images → 11 objects reside on the same server.
- URL for the base HTML file is:
<http://www.someSchool.edu/someDepartment/home.index>

1. The HTTP client process initiates a TCP connection to the server `www.someSchool.edu` on port number 80, which is the default port number for HTTP. Associated with the TCP connection, there will be a socket at the client and a socket at the server.
2. The HTTP client sends an HTTP request message to the server via its socket. The request message includes the path name `/someDepartment/home.index`. (We will discuss HTTP messages in some detail below.)
3. The HTTP server process receives the request message via its socket, retrieves the object `/someDepartment/home.index` from its storage (RAM or disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.
4. The HTTP server process tells TCP to close the TCP connection. (But TCP doesn't actually terminate the connection until it knows for sure that the client has received the response message intact.)
5. The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file. The client extracts the file from the response message, examines the HTML file, and finds references to the 10 JPEG objects.
6. The first four steps are then repeated for each of the referenced JPEG objects.

Round-Trip Time



- Server leaves the connection after sending the HTTP response
- **Pipelining:** A browser can request for files without waiting for the reception of pending requests.
- TCP closes after some idle period
- Default mode HTTP: Persistent connection with pipelining.

HTTP Request Format

- HTTP request message:

```
GET /somedir/page.html HTTP/1.1
```

```
Host: www.iitm.ac.in
```

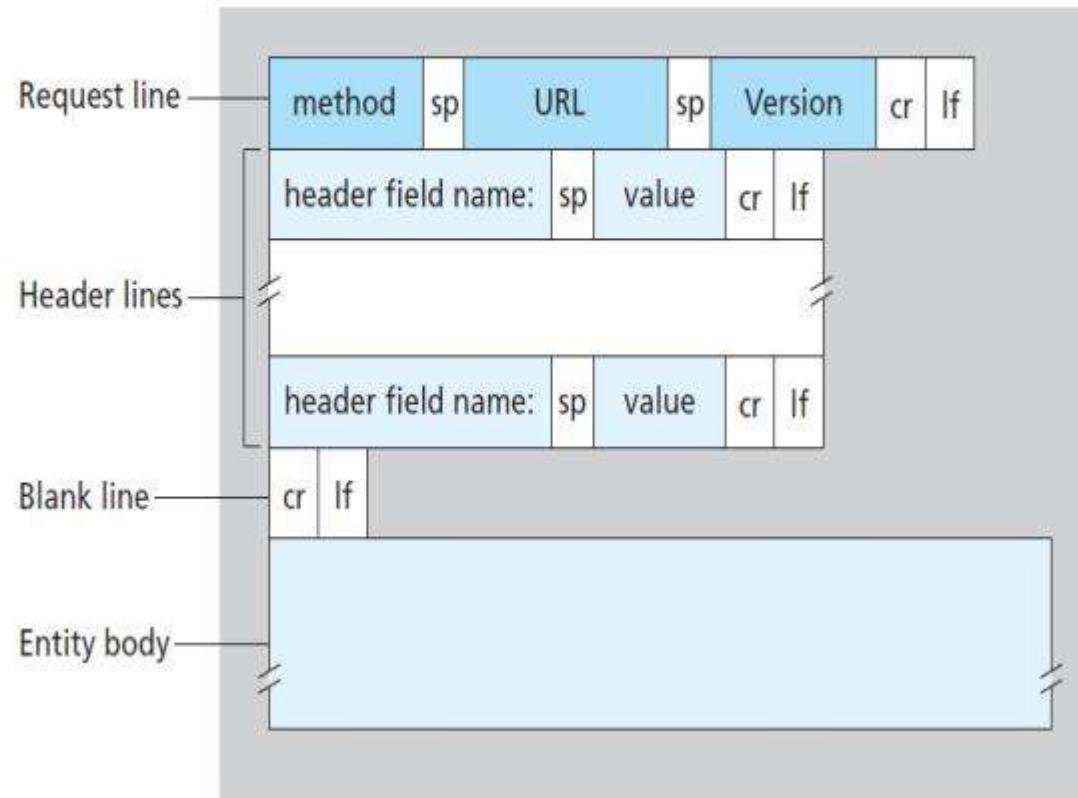
```
Connection: close
```

```
User-agent: Mozilla/4.0
```

```
Accept-language: En
```

- Methods: GET, PUT, POST, HEAD, DELETE

HTTP Request



- HTTP response message:

HTTP/1.1 200 OK

Connection: close

Date: Sat, 07 Jul 2007 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Sun, 6 May 2007 09:23:24 GMT

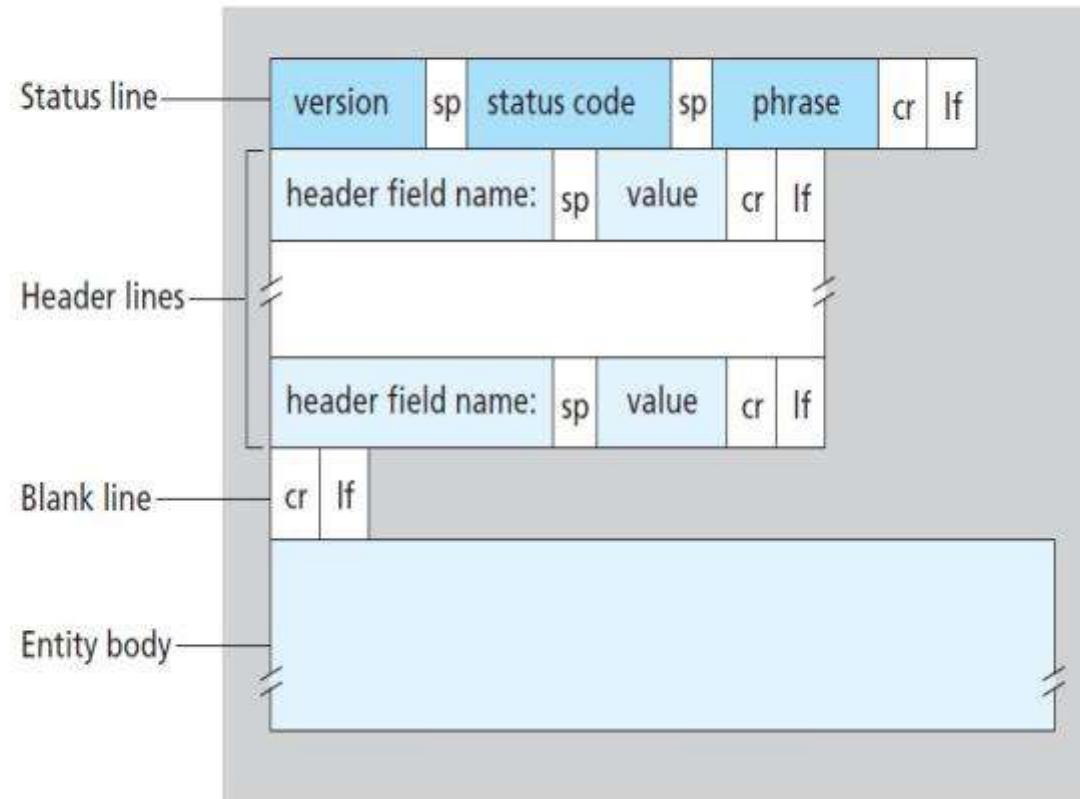
Content-length: 6821

Content-Type: text/html

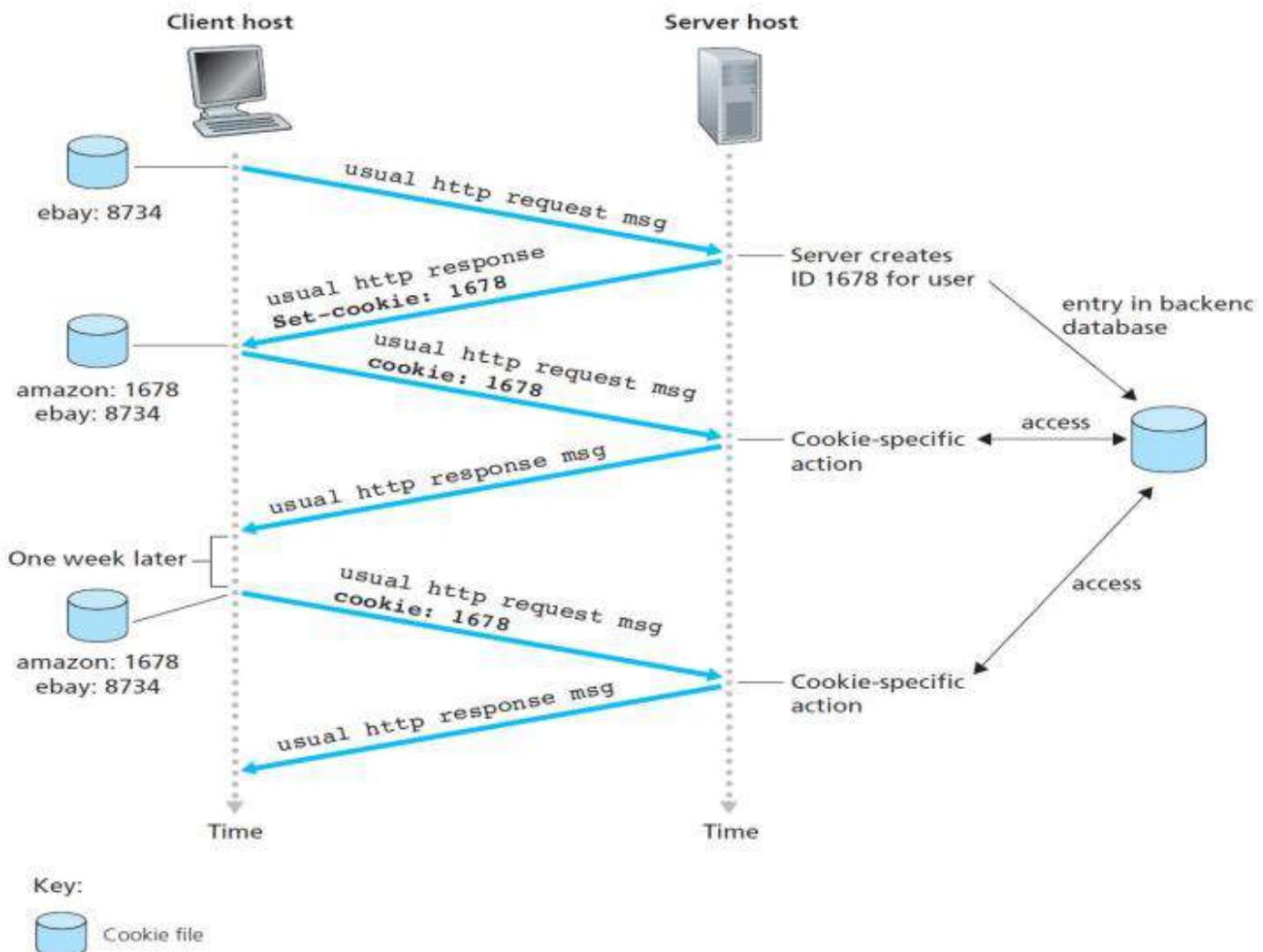
(data data ... data)

- 200 OK
- 301 Moved Permanently
- 404 Not Found

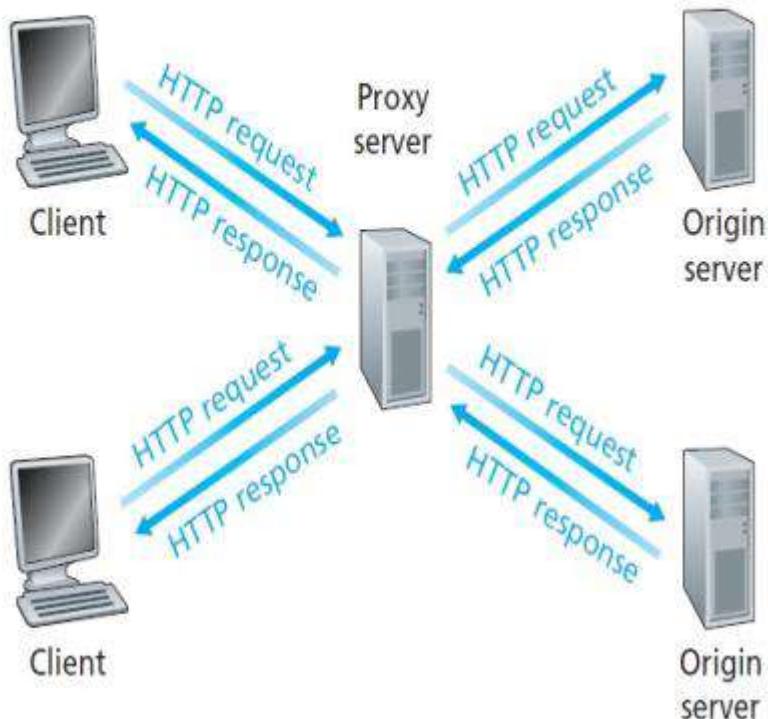
HTTP Response



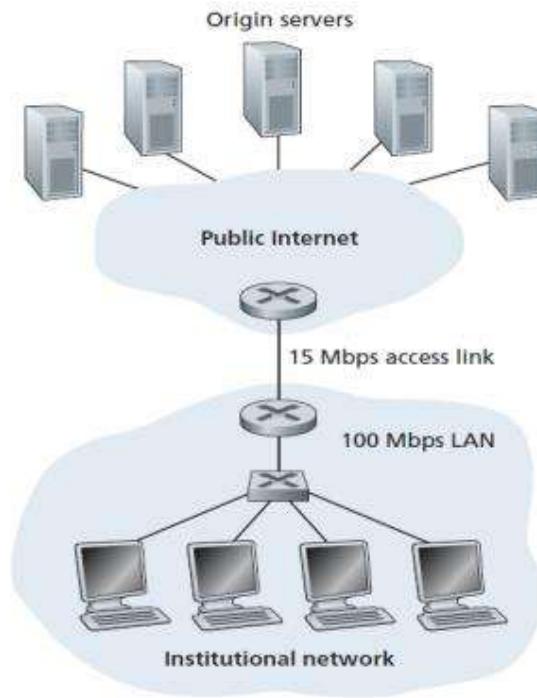
Cookies



Web Caching



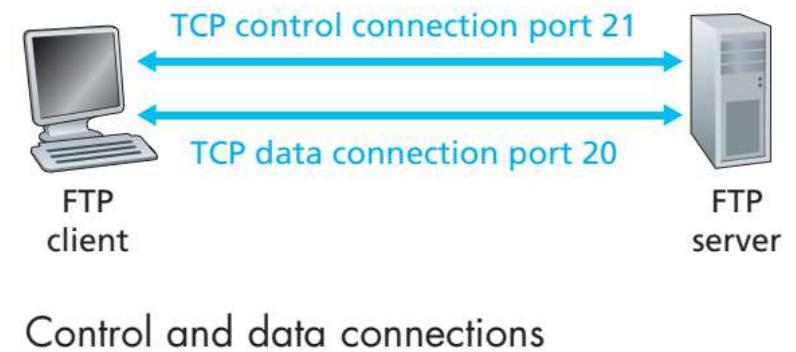
Problem



- Average object size is 1Mbits
- Average request rate 15 objects per sec.
- Average response time from internet is 2 sec.

- Traffic intensity on the LAN
- 0.15
- Traffic intensity on the access link
- 1
- Suppose the access link is upgraded to 100Mbps, find traffic intensity on the access link
- Find the average resposne time
- Expensive solution

- Similar to HTTP: client-server architecture, transmission control protocol
- Two parallel TCP connections to transfer a file: **TCP control connection** and **TCP data connection**
- Control information:
 - User identification
 - Change remote directory
 - Commands to **put** and **get** files
- FTP is said to control information **out-of-band** whereas HTTP is said to control information **in-band**.

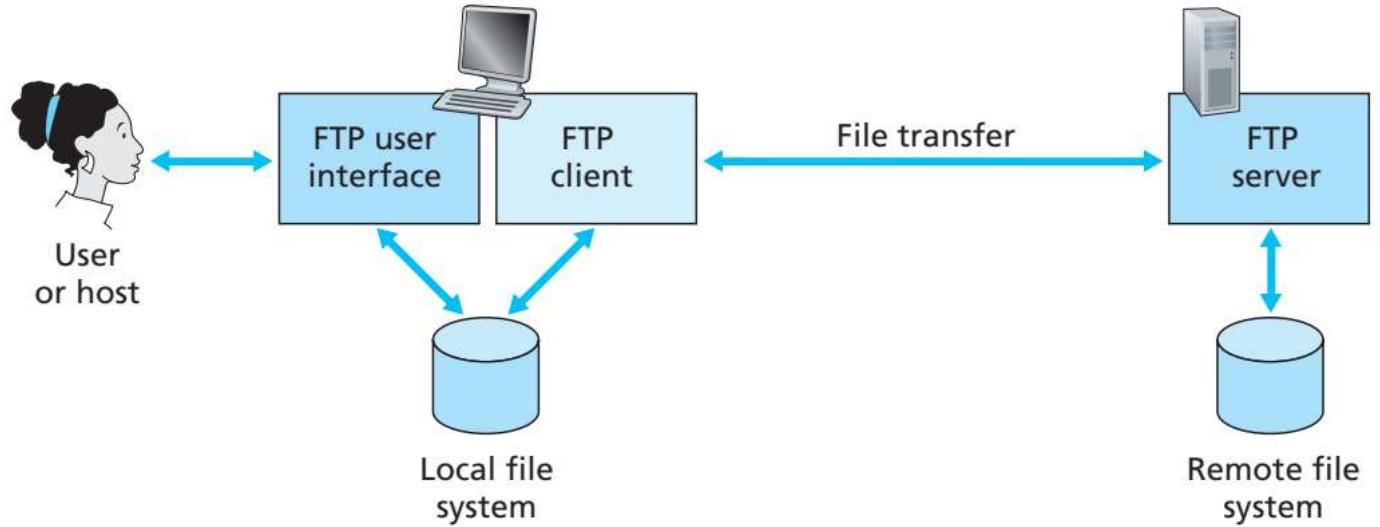


- Commands:

- **USER** username
- **PASS** password
- **LIST**
- **RETR** filename
- **STOR** filename

- Replies:

- **331** username OK, password required
- **125** data connection already open; transfer starting
- **425** can not open data connection
- **452** error writing file



- Asynchronous communication medium
- Major components of e-mail system:
 - **User agent**: allows users to read, forward, save and compose messages
 - **Mail server**
 - **SMTP**
- Examples of user agents: Microsoft Outlook, Mozilla Thunderbird, Apple Mail

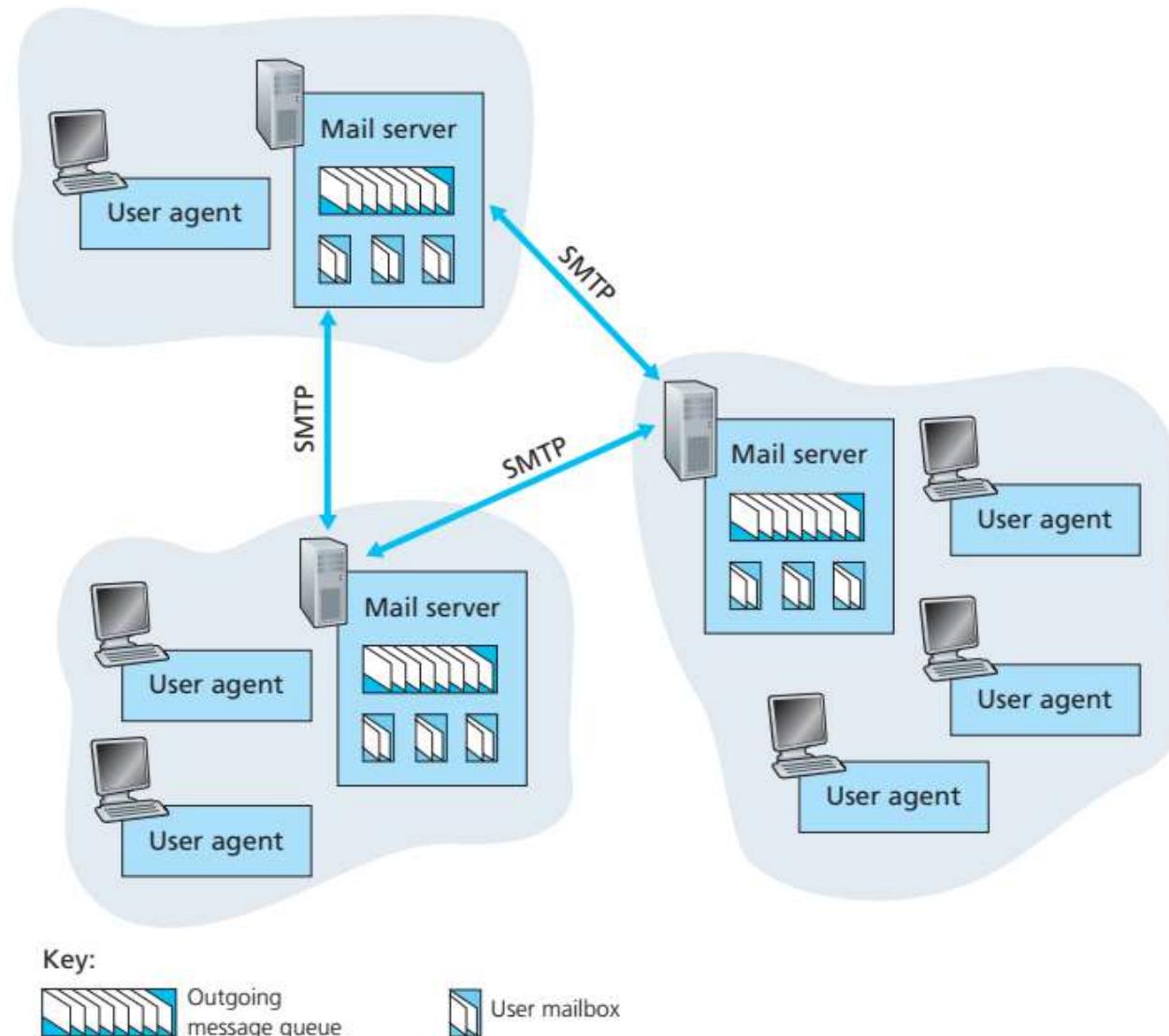


Figure 2.16 ♦ A high-level view of the Internet e-mail system

- User agent sends message to user's mail server.
- SMTP transfers message from user's mail server to recipient's mail server.
- Client side of SMTP is running on sender's mail server and server side of SMTP is running on recipient's mail server.
- Recipient's mail server delivers the message in recipient's mail box.

SMTP Sequence of Operations

- Alice composes message using her user agent. Provides Bob's mail address and instructs to send the message.
- User agent sends the message to her mail server and message waits in the queue of the server.
- SMTP client sees the message in the mail server and it opens a TCP connection to an SMTP server running on Bob's mail server.
- SMTP transfers the message from client to server.
- SMTP server receives the message. Bob's mail server places the message in Bob's mail box.
- Bob invokes his user agent to read the message.

- If recipient's mail server is down, SMTP client **reattempts** to send the message (say for every 30 minutes)
- If the delivery is not successful after some duration, it will be notified to the sender and message will be dropped.

SMTP:

- restricts the body of all mail messages to simple 7-bit ASCII
- Valid when transmission capacity was scarce and no one was e-mailing large attachments or large image, audio, or video files.
- message does not get placed in some intermediate mail server

Client-Server Conversation

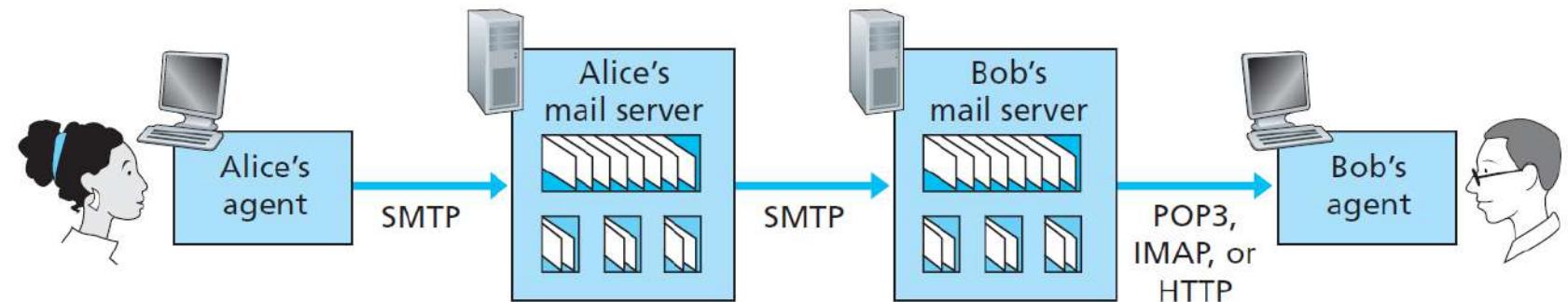
```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

- Header lines similar to those in HTTP messages
- Header must have **From:**, **To:**
- Optional header lines include **Subject:**

- HTTP is a **pull protocol**
- SMTP is **push protocol**
- SMTP requires each message to be 7-bit ASCII format.
HTTP does not have this restriction
- HTTP encapsulates each object in its own HTTP response message. Internet mail places all of its objects into one message.

- In early days of internet, Bob reads mail by logging onto mail server and executing a mail reader on that host
- Client-server architecture
- Reads e-mail by running a client on the user's end system
- Mail access protocol transfers message from Bob's mail server to his local PC.
- Popular mail access protocols: Post Office Protocol - version 3 (**POP3**), Internet Mail Access Protocol (**IMAP**) and HTTP

- Begins when a user agent opens a TCP connection with mail server on port 110.
- POP3 progresses in three phases:
 - Authorization
 - Transaction
 - Update
- Authorization: `user <username>` and `pass <password>`
- Transaction: user agent sends commands and server responds with `+OK` and `-ERR`



POP3 Transaction

- Two modes:
 - download and delete
 - download and keep
- Download and delete:

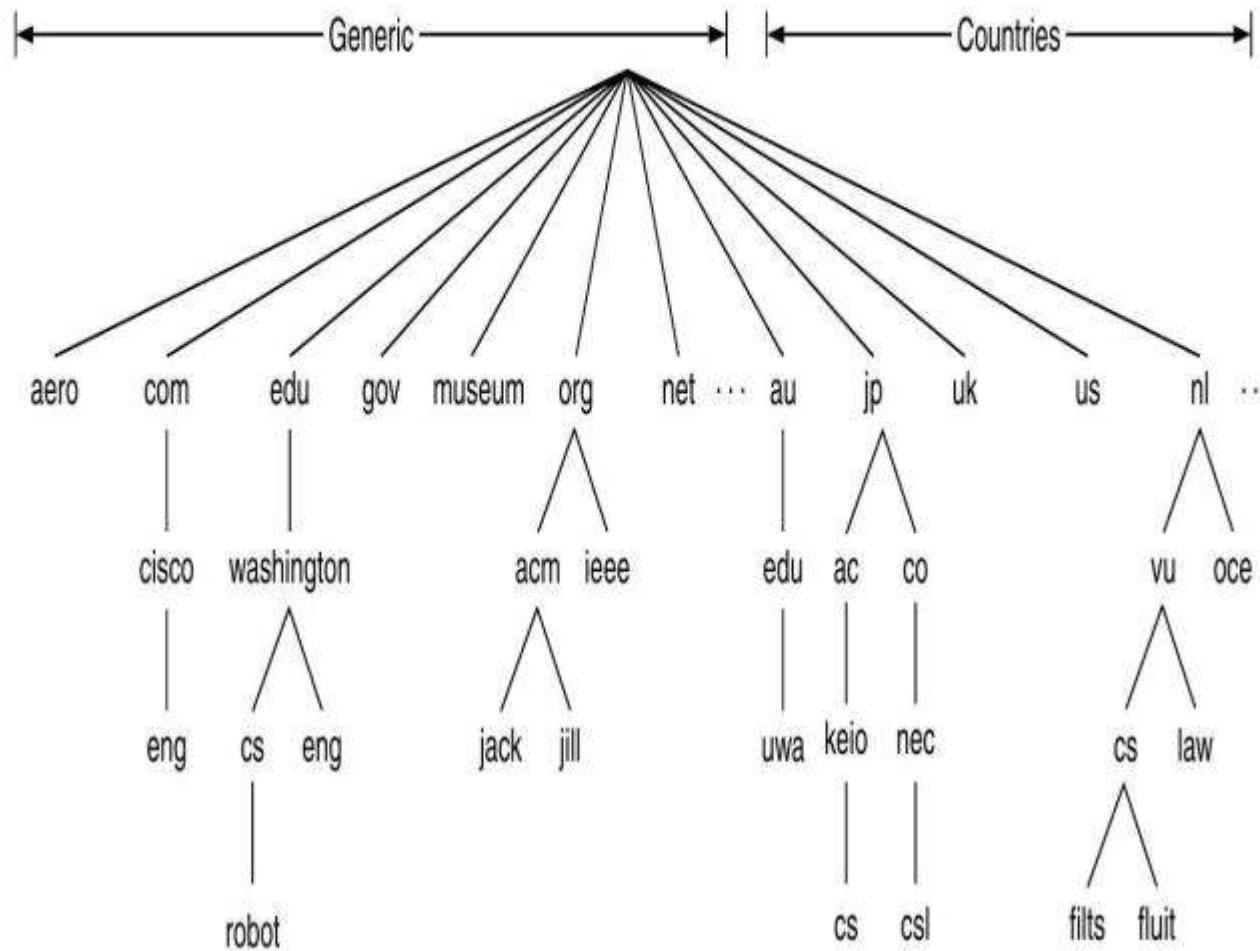
```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: dele 1
C: retr 2
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

- IMAP associates each message with a folder
- Provides commands to allow users to **create folder** and **move messages across folders**
- Provides commands to search for a message
- Maintains user **state information** across IMAP sessions
- Components of messages can be retrieved
- HTTP:
 - e-mail access through web browser
 - web browser communicates to the mail server via HTTP

What is a Domain Name

- Consider www.iiits.in
- Domain: **in**
- What is the domain name of www.iitm.ac.in
- Domain: **in**, subdomain: **ac**
- **250** top-level domains; examples: com, org, edu.

Domain Name Space



Examples of Domains

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No

Who Manages Domains

- **ICANN**: Internet Corporation for Assigned Names and Numbers
- **Registrars** of ICANN check for uniqueness
- Domain names can be **absolute** or **relative**
- Absolute domain names end with .
- Relative domain names have to be interpreted based on the context

- We identify hosts by hostnames. For example, www.amazon.in
- For a network, there is a very little information about the host. Network needs **IP address** for processing
- Domain name servers (**DNS**) provides the necessary mapping from hostname to IP address
- DNS is an application layer protocol used by other applications
- Client-Server architecture; uses UDP at its transport layer

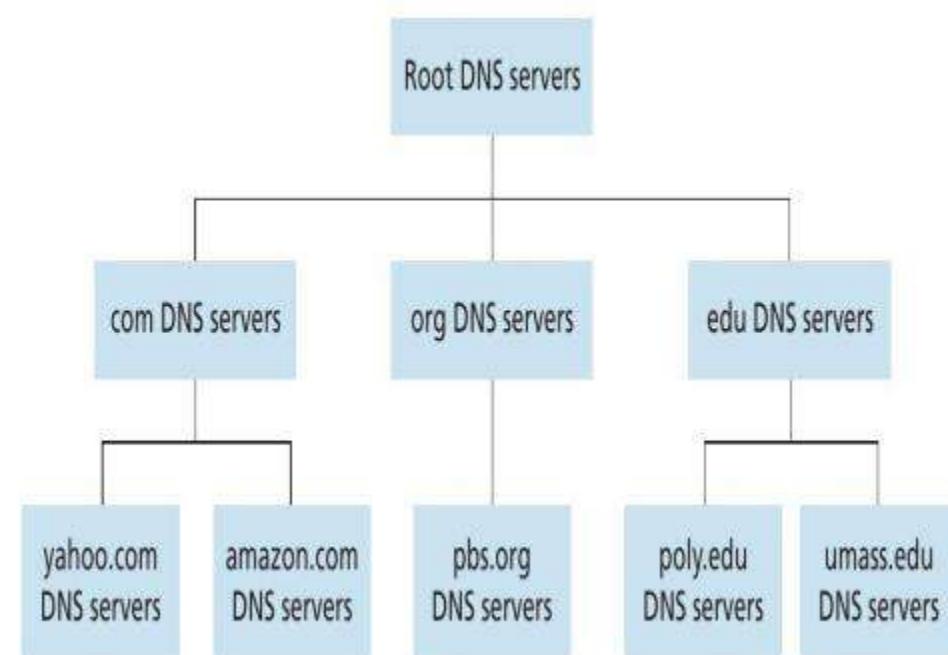
DNS Example:

1. The same user machine runs the client side of the DNS application.
2. The browser extracts the hostname, `www.someschool.edu`, from the URL and passes the hostname to the client side of the DNS application.
3. The DNS client sends a query containing the hostname to a DNS server.
4. The DNS client eventually receives a reply, which includes the IP address for the hostname.
5. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.

DNS adds an additional delay—sometimes substantial—to the Internet applications that use it

- We typically memorize **alias** hostnames but the actual hostnames are very complicated
- The **canonical** hostnames are not mnemonic. Canonical: *according to the rules*
- Example: *www.timesofindia.com* is the alias but the actual host name or canonical name is *timesofindia.indiatims.com*
- Different canonical names might have the same alias
- Many hosts can be installed within a domain or subdomain.
Example: *www.ee.iitm.ac.in*, *www.cse.iitm.ac.in*,
smail.iitm.ac.in

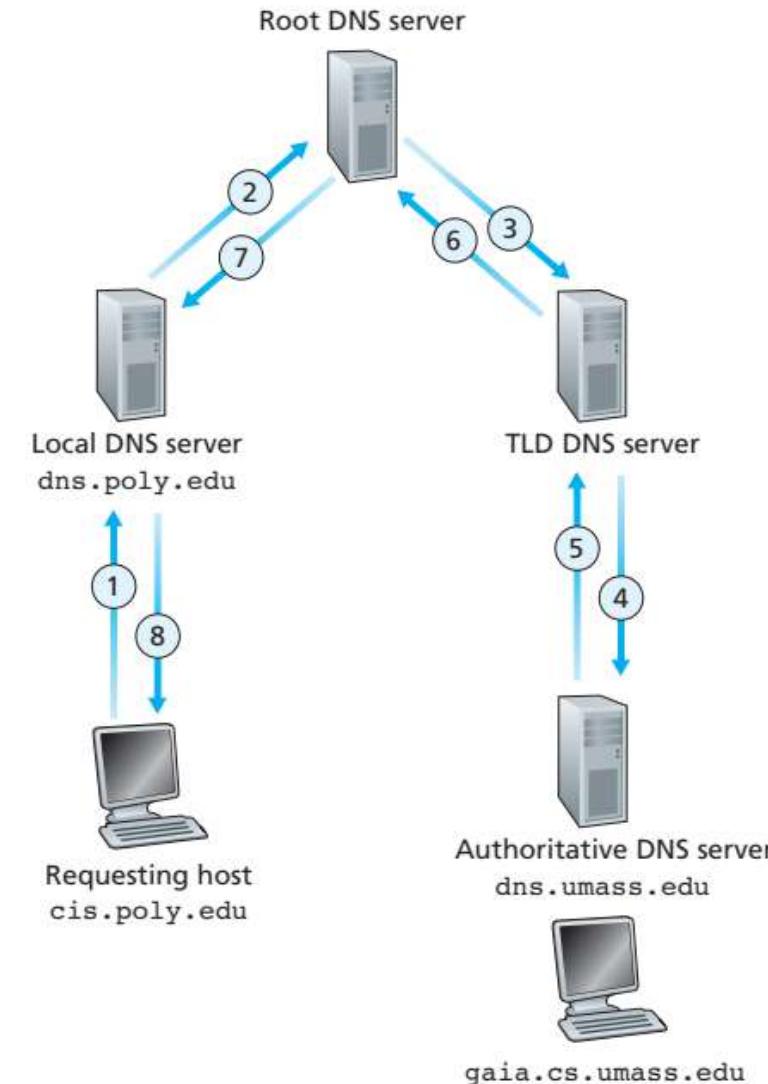
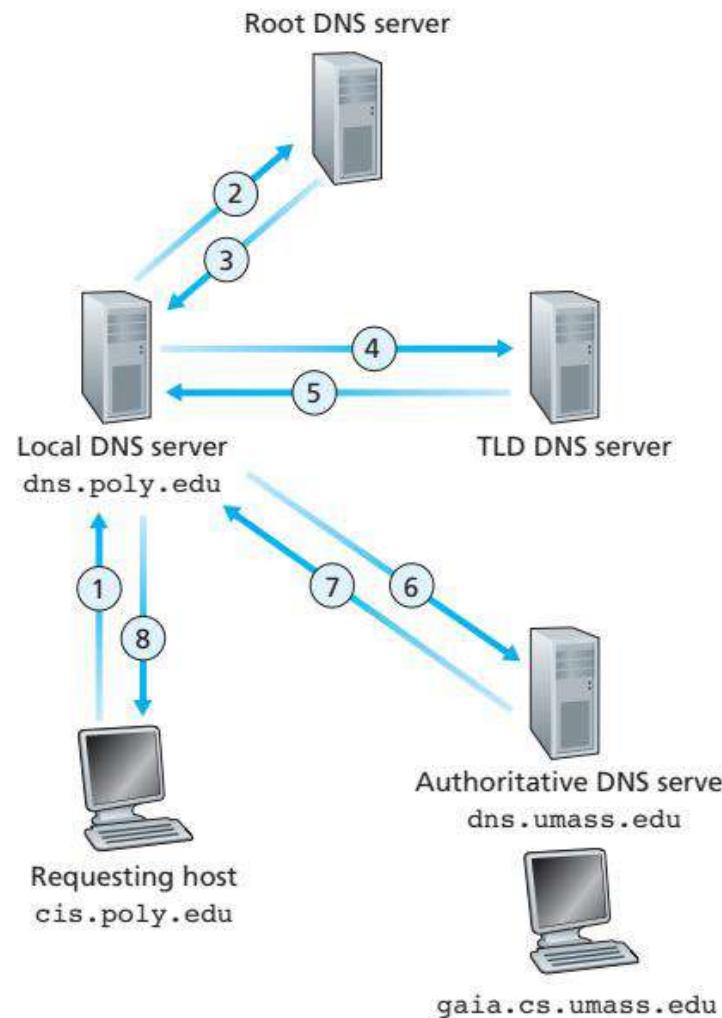
Hierarchy of DNS



Problems of Single DNS:

- A single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

How does DNS Work: Recursive and Iterative Query



- An ISP can provide local DNS
- Host will query the local DNS and that takes it forward to the root DNS
- Cache DNS replies

DNS Resource Records:

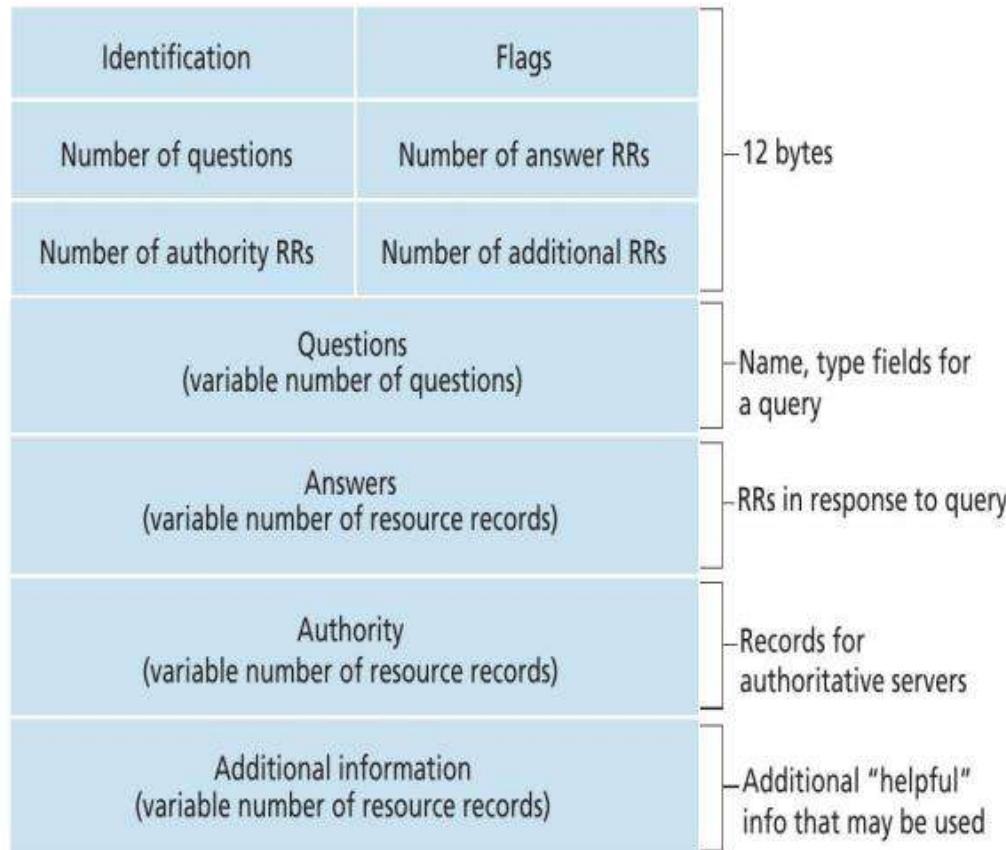
- DNS distributed database store resource records
- Resource Record is four tuple: (Name, Value, Type, TTL)
- TTL: Time-to-Live
- The interpretation of Name and Value files change based on Type

Types in RR

- **Type = A:** *Name* is a **hostname** and *Value* is the **IP address** of the host
- **Type = NS:** *Name* is a **domain** and *Value* is the **hostname** of the authoritative DNS server
- **Type = CNAME:** *Name* is an **alias** and *Value* is the **canonical hostname** of the alias.
- **Type = MX:** *Name* is an **alias hostname** and *Value* is the **canonical hostname of a mail server** of the alias.

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

DNS Message Format



Flags:

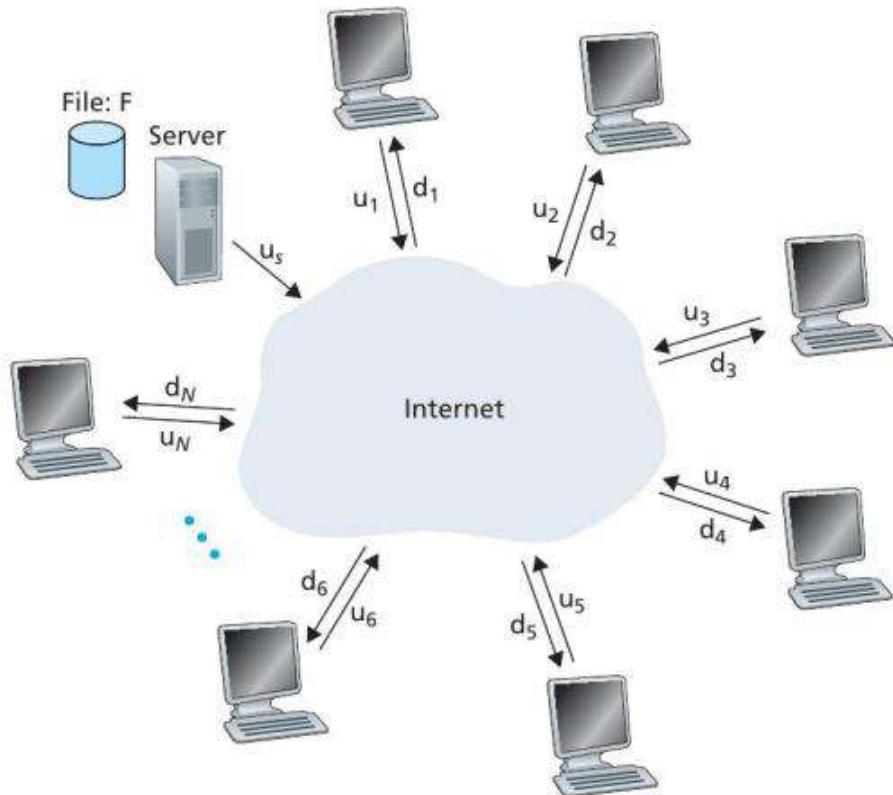
- 1-bit flag to indicate its a query/reply
- 1-bit recursion flag is set if the DNS supports recursion

- **File distribution**: application that transfers a file from a single source to multiple peers.
- **Database distributed** over a large community of peers.
- **Internet telephony** : Skype.

File Distribution:

- Each peer can **redistribute** any portion of the file to any other peer
- Popular file distribution protocol : BitTorrent, developed by Bram Cohen
- Scalability

Scalability



- N peers
- **Distribution time:** the time required to distribute a file to all peers.

- Internet has abundant bandwidth and all bottlenecks are in the network access
- All the server and client bandwidth is available for file distribution

Distribution Time for Client-Server Architecture

- Let D_{cs} denote the distribution time for client-server architecture for a file size of F bits
- The server has to transmit a total of NF bits at an upload rate of $u_s \text{ bps}$.
- Minimum time required for distribution is $\frac{NF}{u_s}$ seconds
- Let $d_{min} = \min\{d_1, \dots, d_N\}$
- Minimum distribution time is $\frac{F}{d_{min}}$ seconds
- Thus,

$$D_{cs} \geq \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{min}} \right\}$$

- Show that

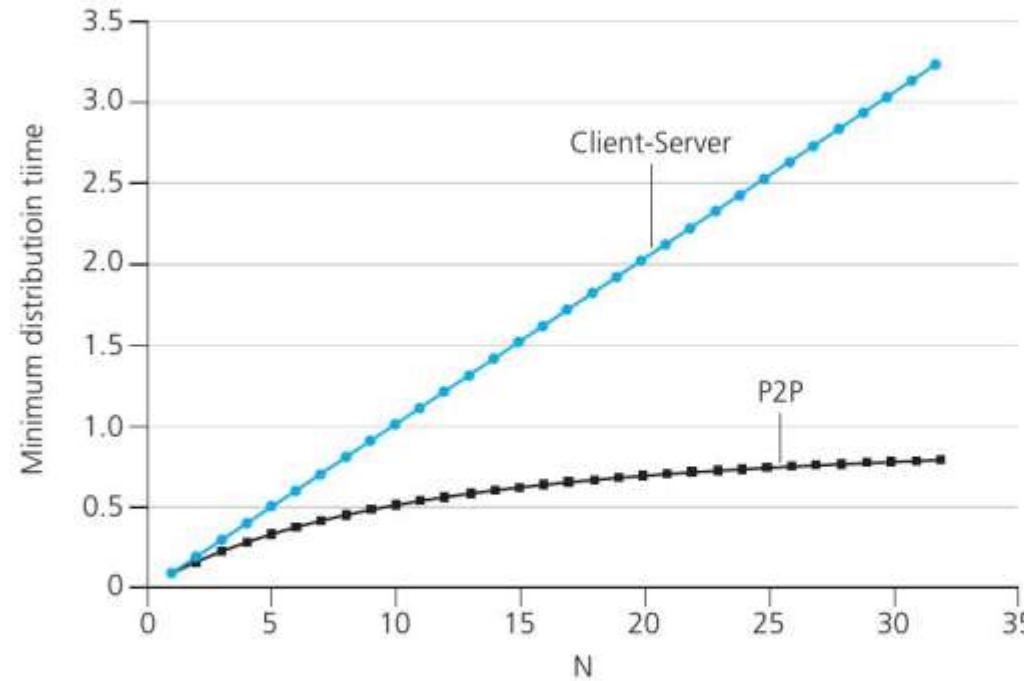
$$D_{cs} = \max \left\{ \frac{NF}{u_s}, \frac{F}{d_{min}} \right\}$$

- The server has to send each bit of the file at least once:
Minimum distribution time is at least $\frac{F}{u_s}$ seconds
- The peer with lowest download rate can not obtain F bits in less than $\frac{F}{d_{min}}$ seconds
- The total upload rate $u_{total} = u_s + u_1 + \dots + u_N$. The system must deliver F bits to each of the N peers: Minimum distribution time is $\frac{NF}{u_{total}}$
- Thus, minimum distribution time D_{P2P} is at least

$$\max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_{total}}\right\}$$

- Assumption: each peer can redistribute a bit as soon as it receives the bit.
- There is a scheme that actually achieves this lower bound.

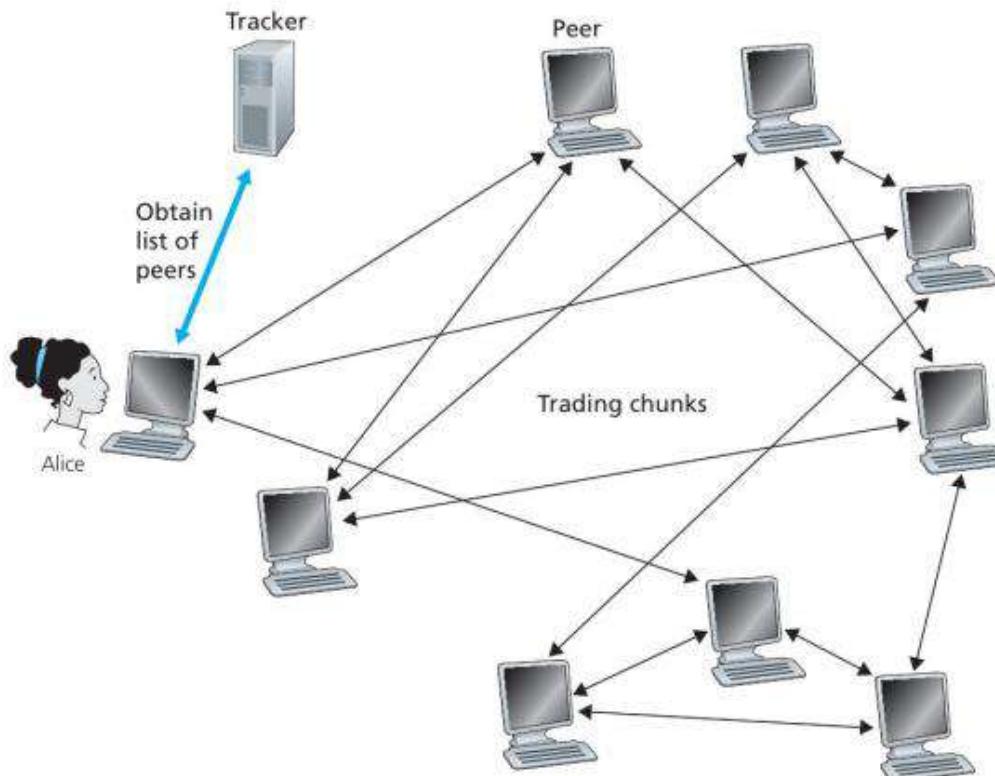
Distribution Time for P2P Architecture



- All peers upload at a rate of u bps.
- $\frac{F}{u} = 1$ hour, $u_s = 10u$ and $d_{min} \geq u_s$.

- Collection of peers participating in the distribution of a file is called a **torrent**
- Peers in a torrent download equal-size **chunks** of the file (typically 256 KBytes)
- A peer accumulates more and more chunks over time
- Once a peer has acquired complete file, it may leave the torrent or continue to participate in the torrent
- Peers may leave torrents with subsets of chunks

- Each torrent has a node called **tracker**.
- When a peer joins the torrent, it registers with the tracker
- Each peer in the torrent **periodically updates the tracker** about its presence.



- Alice receives a subset of participating peers in the torrent
- She establishes TCP connection with some of the peers and we call them as **neighboring peers of Alice**
- Neighboring peers may vary over time
- Each peer will have some subset of chunks from the file, with different peers having different subsets
- Alice maintains a list of chunks that her neighbors have.

- Alice will issue requests for chunks she currently does not have
- Which chunks should be requested first?
- Rarest first: finds the chunks that are rarest among her neighbors
 - Alice will issue requests for chunks she currently does not have
 - To which of her neighbors should she send requested chunks?
 - Tit-for-tat

- Alice gives priority to the neighbors that are currently supplying her data at the highest rate
- Typically four neighbors are chosen. These peers are said to be **unchoked**
- Every 30 seconds, she also picks one additional neighbor at random and sends it chunks. Let it be Bob.
- Bob is said to be **optimistically unchoked**.
- In due course of time, Alice, may become one of the top uploaders in which case Bob could start sending data to Alice.

Distributed Hash Tables (DHT)

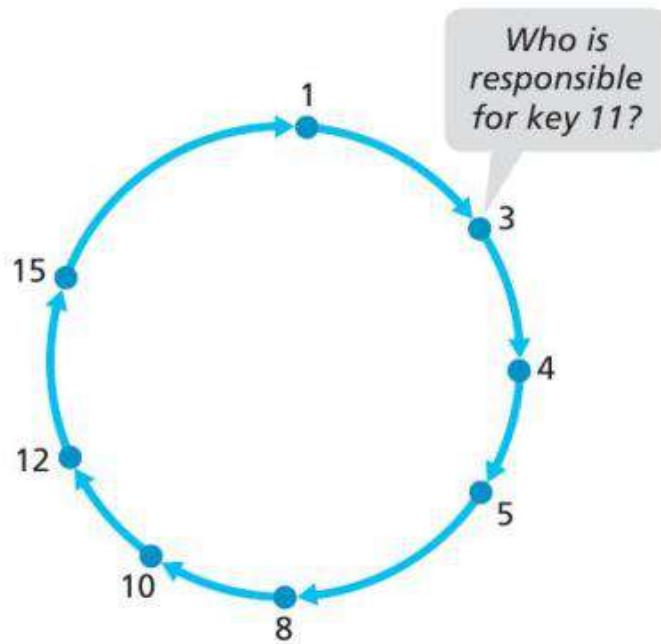
- Huge database to be stored among number of peers in a distributed way
- Database consists of (key, value) pairs. For Example, (PAN No., Aadhar No.), (Content Name, IP), etc.
- Peers query the database by supplying the key and database replies the matching pairs to the querying peer
- **How to store database among the peers**

- Assign an **identifier** to each peer.
- An identifier is an integer in $[0, 2^n - 1]$ for some fixed n
- (key, value) pairs are also identified by integers using **hash functions**
- Hash function is available to all peers.

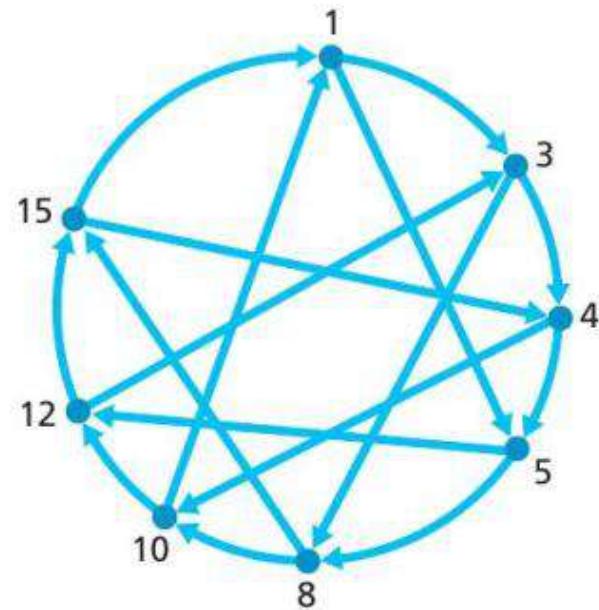
The index for a specific string will be equal to sum of ASCII values of characters multiplied by their respective order in the string after which it is modulo with 2069 (prime number).

String	Hash function	Index
abcdef	$(971 + 982 + 993 + 1004 + 1015 + 1026) \% 2069$	38
bcdefa	$(981 + 992 + 1003 + 1014 + 1025 + 976) \% 2069$	23
cdefab	$(991 + 1002 + 1013 + 1024 + 975 + 986) \% 2069$	14
defabc	$(1001 + 1012 + 1023 + 974 + 985 + 996) \% 2069$	11

- Define a rule for assigning keys to peers
- **Closest to the key:**
- For example, $n = 4$, with eight peers: 1,3,4,5,8,10,12 and 15.
Store (11, 0123-4567-8910) in one of the eight peers
- By closest convention, peer 12 is the **immediate successor** for key 11. Store in peer 12.
- If the key is larger than all the peer identifiers, we use modulo- 2^n convention.

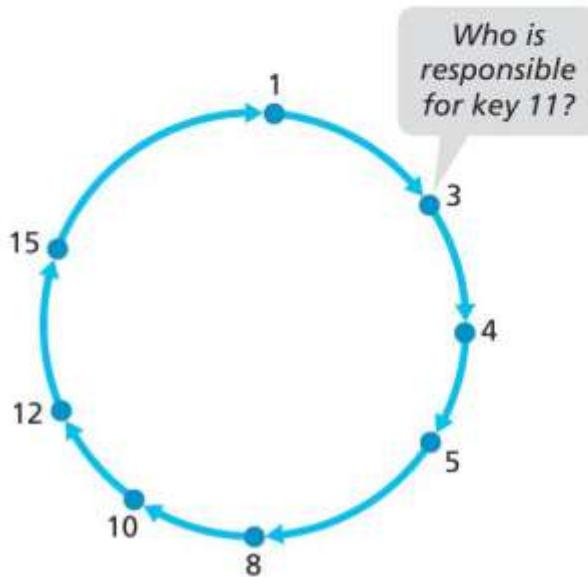


- Each peer is aware of only its immediate predecessor and successor
- N messages at most



- Number of shortcuts are relatively small in number
 - How many shortcut neighbors and which peers should be these shortcut neighbors? Research problem: $O(\log(N))$

- Peers can come and go without warning
- Peers keep track to two immediate predecessor and successors.
- When a peer abruptly leaves, its predecessor and successor learn that a peer has left and **updates the list of its predecessor and successor.**





Computer Communication Networks

Transport Layer

Dr. Raja Vara Prasad

Assistant Professor

IIIT Sri City

Transport Layer

Transport Layer

how two entities can communicate reliably over a medium that may lose and corrupt data ?

controlling the transmission rate of transport-layer entities in order to avoid

Or

recover from, congestion within the network.

Transport Layer Services

- **logical communication**
- Transport-layer **segments**
- Transport-layer protocol provides logical communication between *processes* running on different hosts
- a network-layer protocol provides logical communication between *hosts*
- services that a transport protocol can provide are often constrained by the service model of the underlying network-layer protocol
- a transport protocol can offer reliable data transfer service to an application even when the underlying network protocol is unreliable
- can use encryption

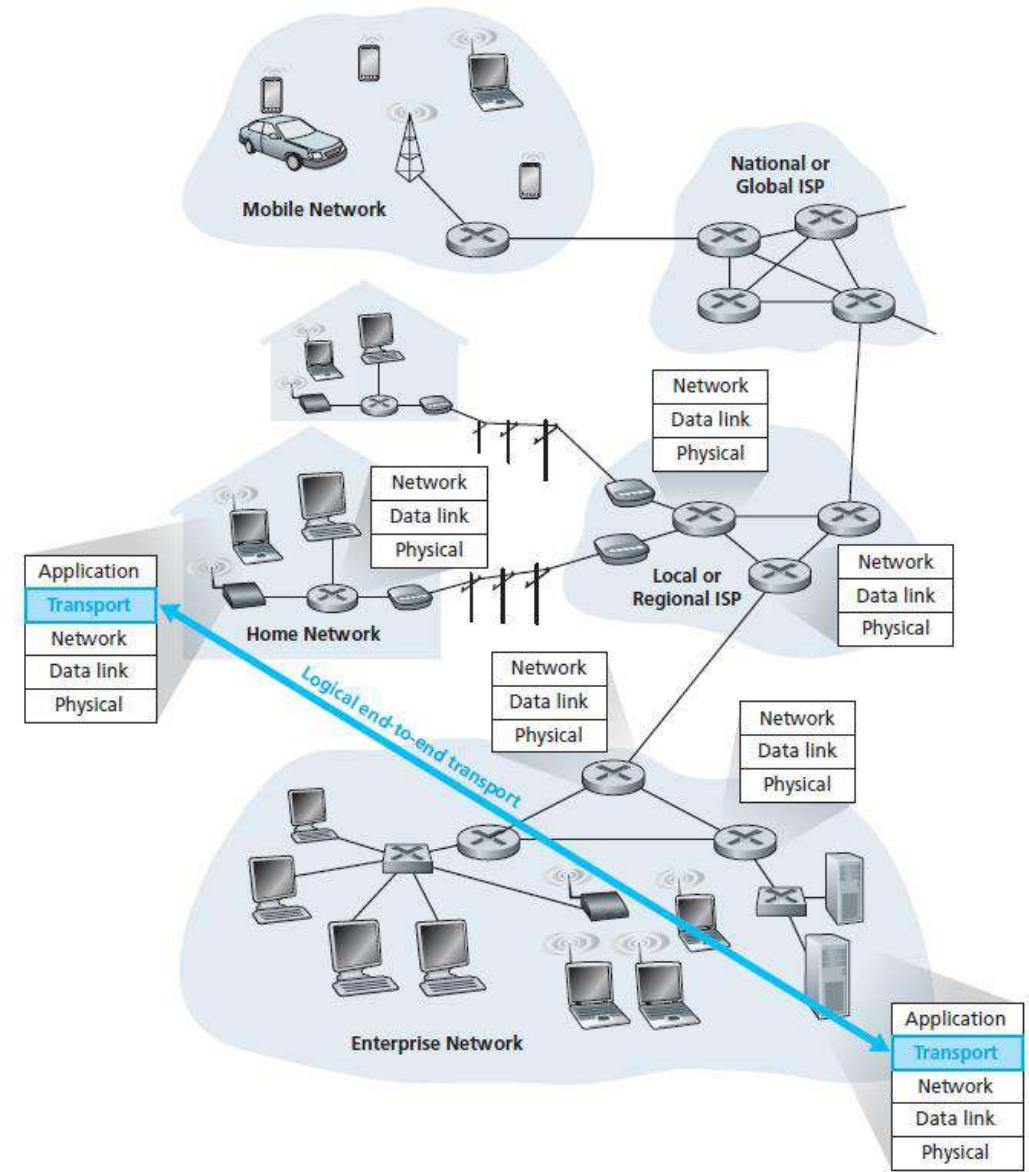


Figure 3.1 • The transport layer provides logical rather than physical communication between application processes

Transport Layer in the Internet

- Internet Protocol. IP provides logical communication between hosts.
- IP service model is a **best-effort delivery service**
- “best effort” to deliver segments between communicating hosts → *makes no guarantees*.
- not guarantee segment delivery
- it does not guarantee orderly delivery of segments
- does not guarantee the integrity of the data in the segments

UDP Services:

process-to-process data delivery and error checking

TCP:

- reliable data transfer
- correct and in order → using flow control, sequence numbers, acknowledgments, and timers
- **congestion control**

Multiplexing and Demultiplexing

- host-to-host delivery service provided by the network layer
- process-to-process delivery service for applications running on the hosts – Transport Layer
- a process can have one or more **sockets**
- transport layer in the receiving host does not deliver data directly to a process → to an intermediary socket
- more than one socket in the receiving host → each socket → unique identifier
- Each transport-layer segment has a set of fields

Demultiplexing:

- receiving end → the transport layer examines these fields to identify the receiving socket
- directs the segment to that socket
- **Multiplexing**
 - gathering data chunks at the source host from different sockets
- encapsulating each data chunk with header information to create segments
- passing the segments to the network layer is called

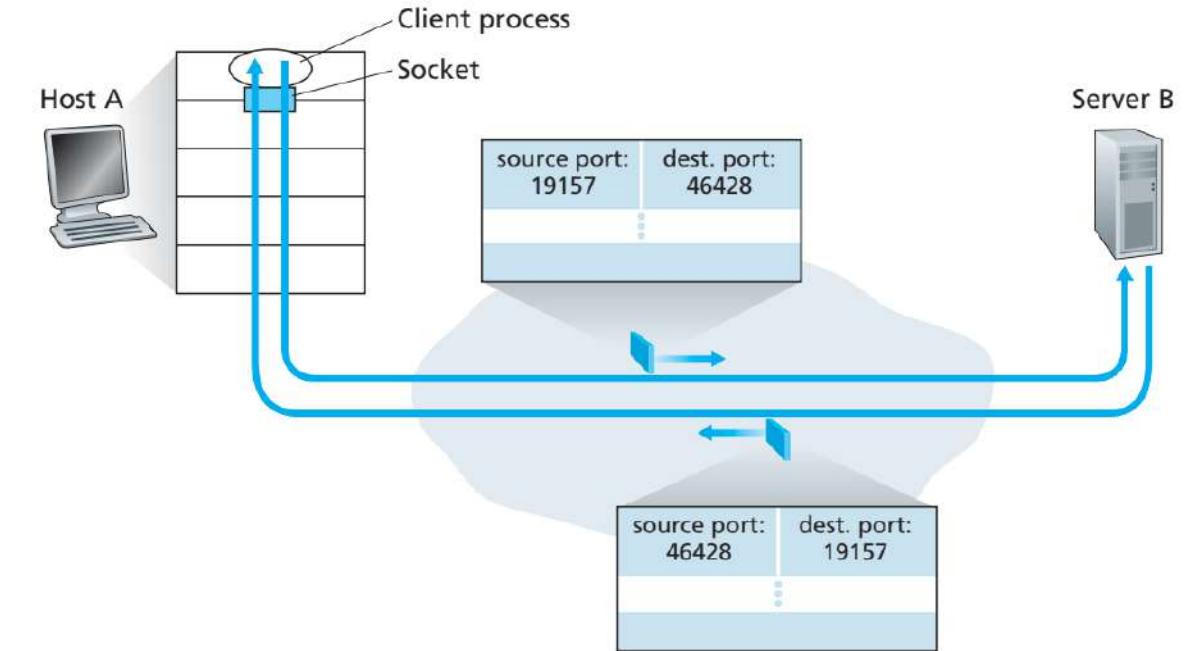
Connectionless Multiplexing and Demultiplexing

UDP socket:

- transport layer assigns a port number in the range 1024 to 65535 that is currently not being used by any other UDP port in the host

Ex: A process in Host A, with UDP port 19157 → to send a chunk of application data to a process with UDP port 46428 in Host B.

- UDP socket: identified by a two-tuple → a destination IP address and a destination port number



if two UDP segments have different source IP addresses and/or source port numbers, but have the same *destination* IP address and *destination* port number ?

Connection Oriented Multiplexing and Demultiplexing

TCP socket:

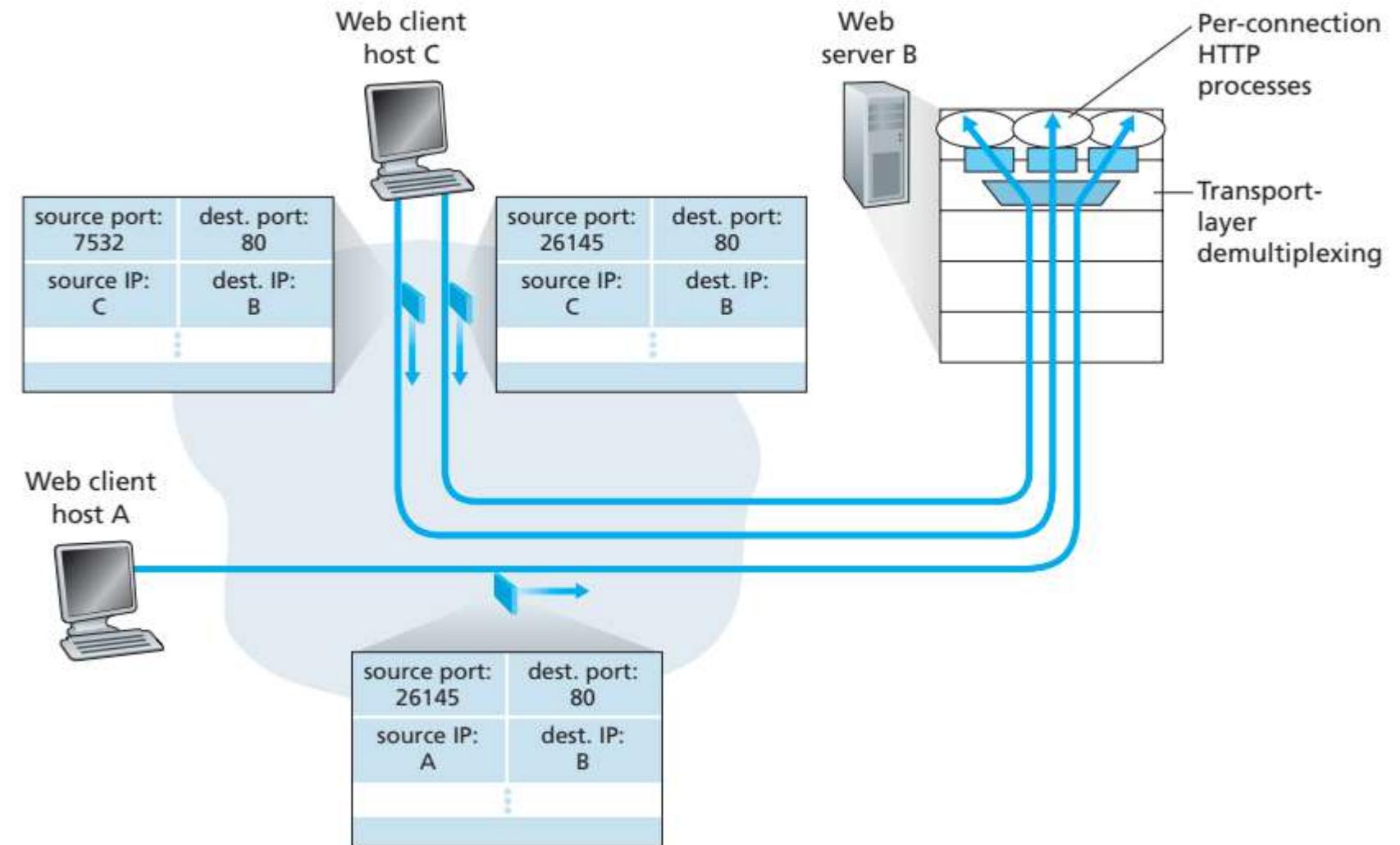
- TCP socket is identified by a four-tuple
source IP , source port number, destination IP, destination port number
- host uses all four values to direct the segment to the appropriate socket

server host may support many simultaneous TCP connection sockets, with each socket attached to a process, and with each socket identified by its own four tuple.

Web Servers and TCP:

- all segments will have destination port 80.
- Web servers often use only one process, and create a new thread with a new connection socket for each new client connection .
- client and server using persistent HTTP → same server socket
- non-persistent HTTP → a new TCP connection is created and closed for every request/response
- frequent creating and closing of sockets --- severely impact the performance of a busy Web server

Connection Oriented Multiplexing and Demultiplexing



Connectionless - UDP

- UDP → no handshaking between sending and receiving transport-layer → UDP is said to be *connectionless*
Example: DNS → a query → DNS query message and passes the message to UDP
- many applications are better suited for UDP for the following reasons:
 - ***Finer application-level control over what data is sent, and when***
→ TCP will also continue to resend a segment until the receipt of the segment has been acknowledged by the destination, regardless of how long reliable delivery takes → real-time applications
 - ***No connection establishment***
 - ***No connection state*** : Connection state includes receive and send buffers, congestion-control parameters, and sequence and acknowledgment number parameters
“can typically support many more active clients when the application runs over UDP rather than TCP”
 - ***Small packet header overhead*** : TCP segment has 20 bytes of header : UDP: 8 bytes

Connectionless - UDP

- UDP is used for RIP routing table updates
- carry network management data

Multimedia applications, such as Internet phone, real-time video conferencing, and streaming of stored audio and video.

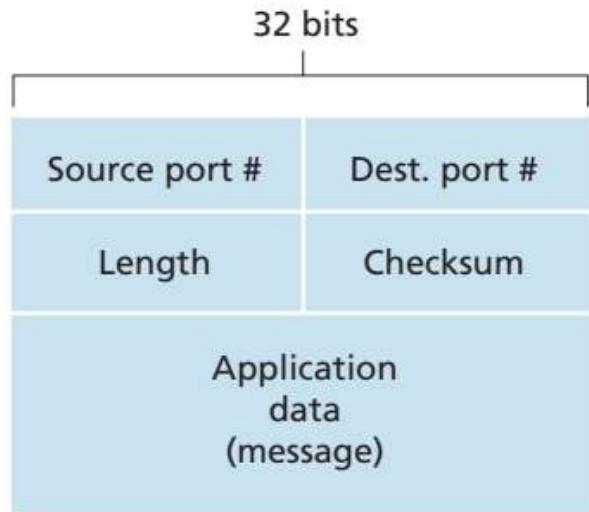
- Internet phone and video conferencing, react very poorly to TCP's congestion control
- When packet loss rates are low and some organizations blocking UDP traffic for security reasons TCP becomes an increasingly attractive protocol for streaming media transport.
- lack of congestion control in UDP can result in high loss rates between a UDP sender and receiver, and the crowding out of TCP sessions

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	typically proprietary	UDP or TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Routing protocol	RIP	Typically UDP
Name translation	DNS	Typically UDP

Connectionless - UDP

- **UDP Segment Structure**
- **UDP Checksum**
provides for error detection

0110011001100000
0101010101010101
100011100001100



The sum of first two of these 16-bit words is

0110011001100000
01010101010101
1011101110110101

Adding the third word to the above sum gives

1011101110110101
100011100001100
0100101011000010

I's complement 101101010011101

At the receiver, all four 16-bit words are added, including the checksum.

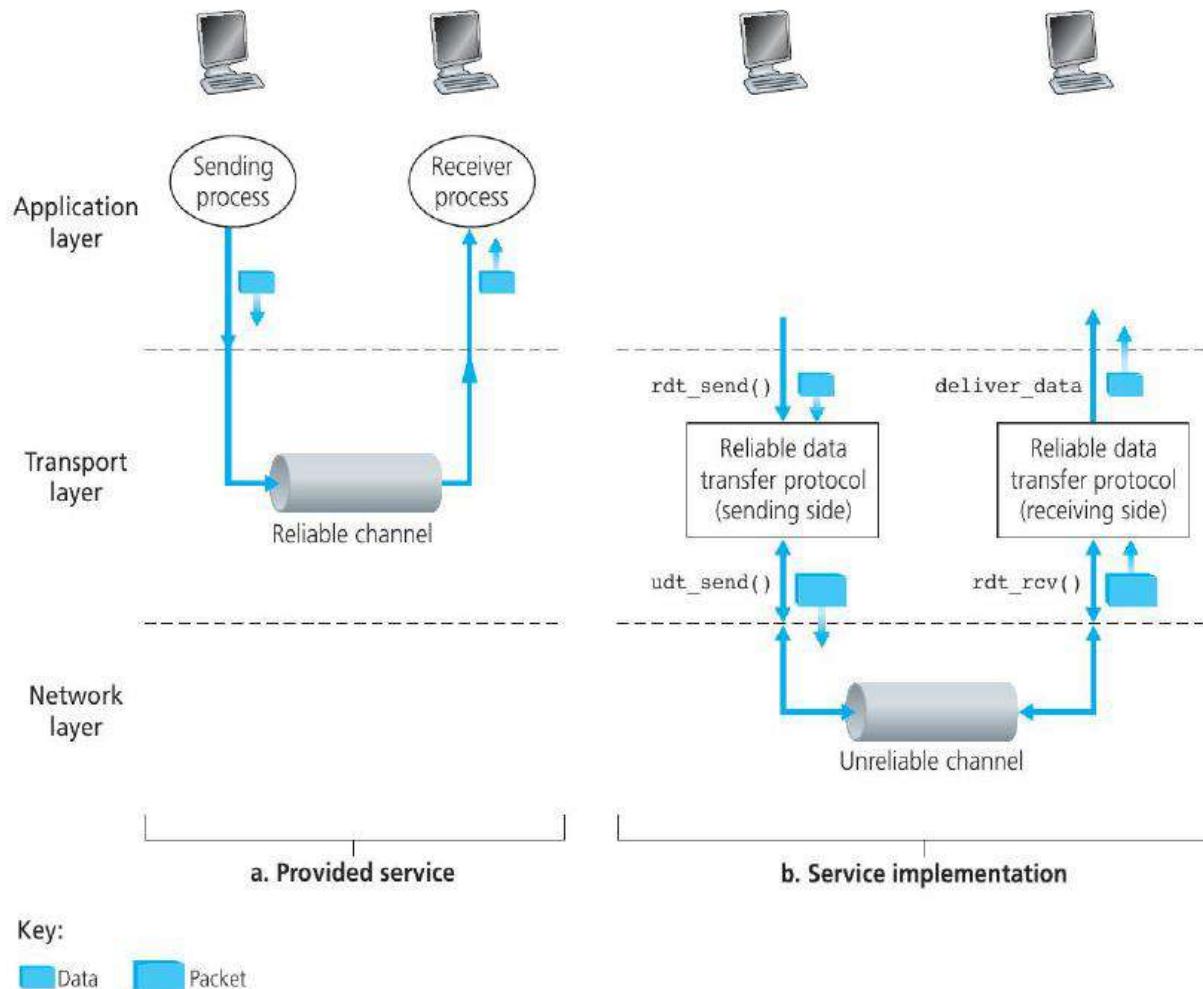
If no errors are introduced into the packet -- 1111111111111111

Link layer protocols also provide error checking. Then why UDP again ?

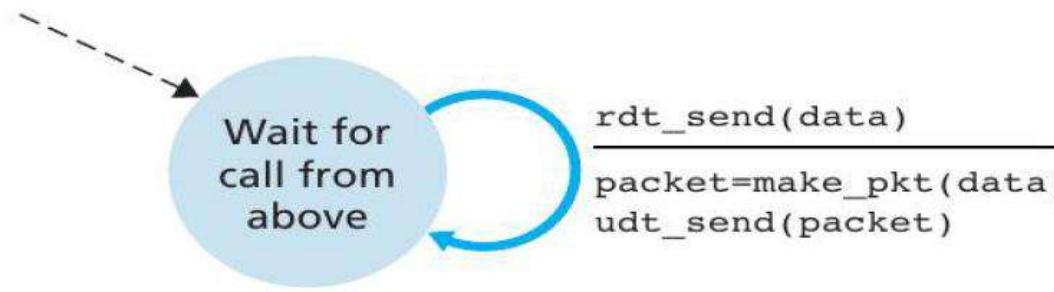
Provides error checking, it does not do anything to recover from an error

Principles of Reliable Data Transfer

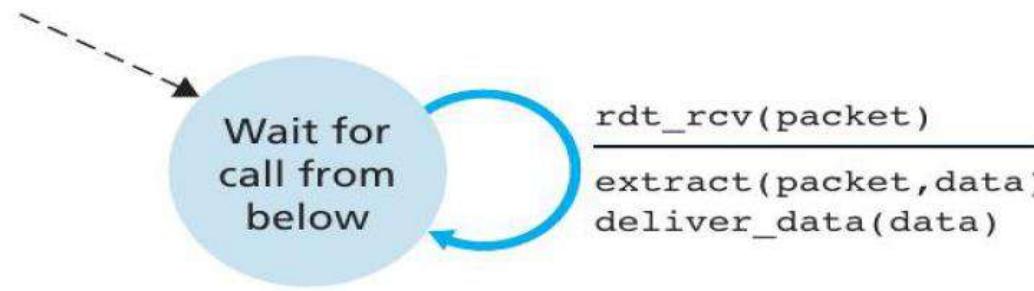
Reliable Data Transfer



RDT1.0: Perfectly Reliable Channel

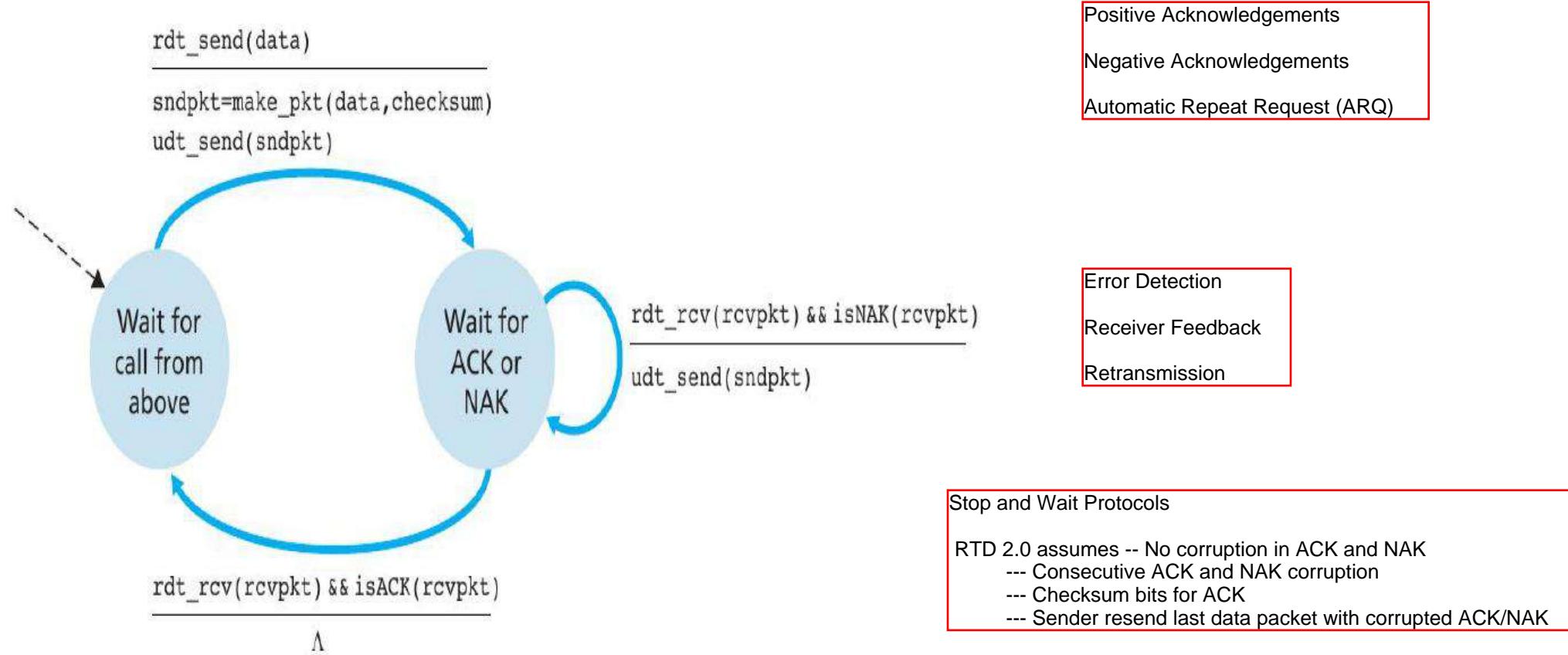


a. rdt1.0: sending side



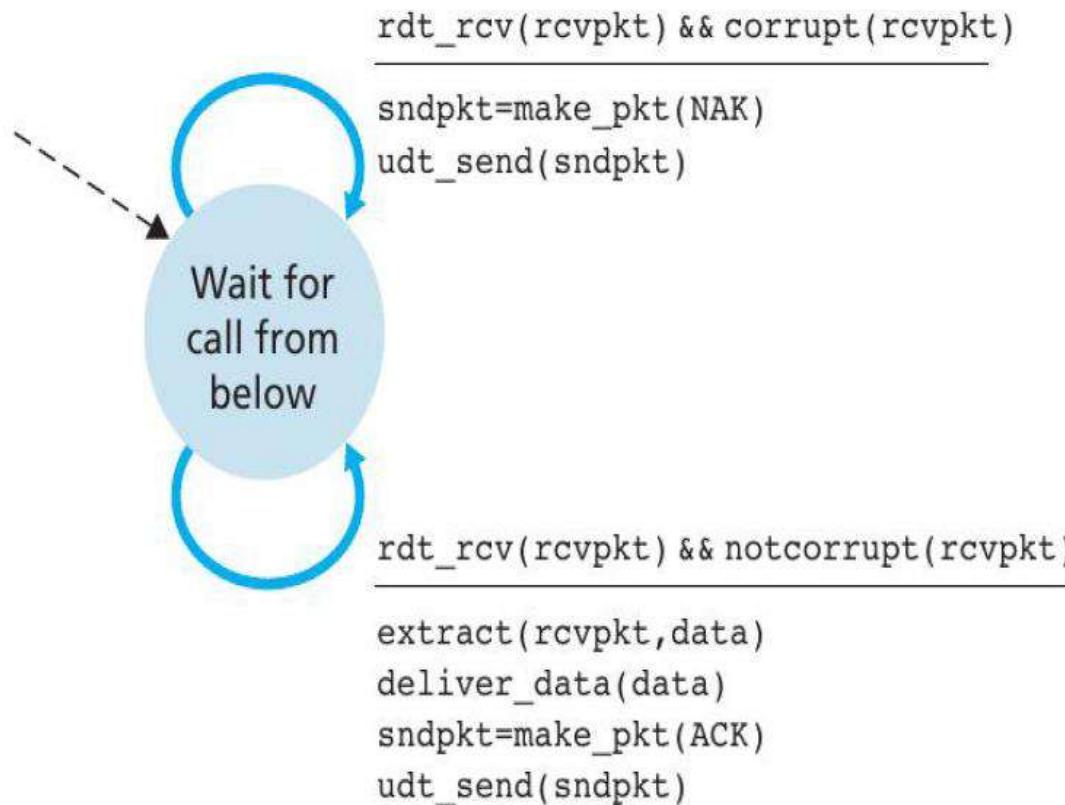
b. rdt1.0: receiving side

RDT Over a Channel with Bit Errors: rdt 2.0 sender



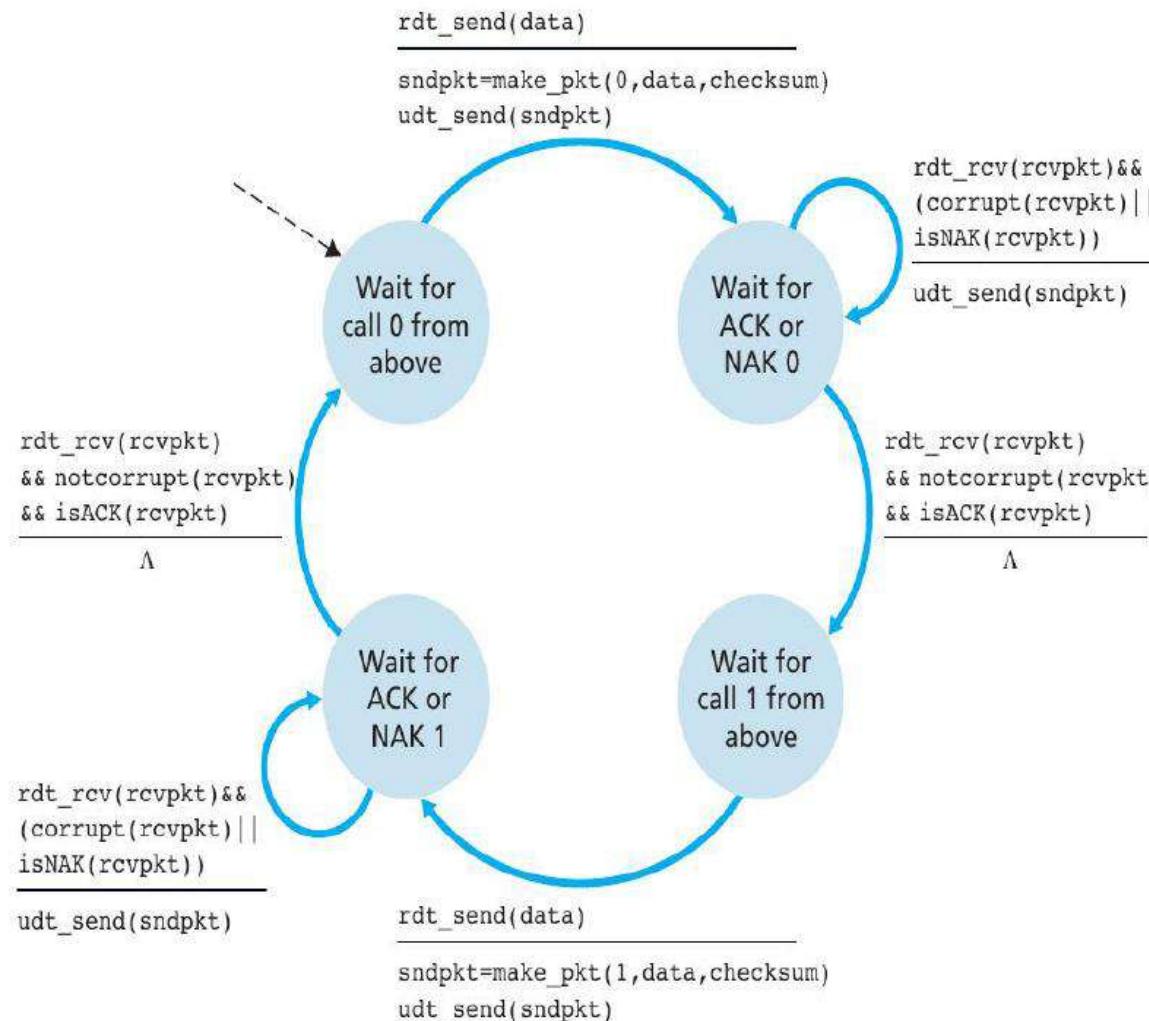
a. rdt2.0: sending side

RDT Over a Channel with Bit Errors: rdt 2.0 receiver

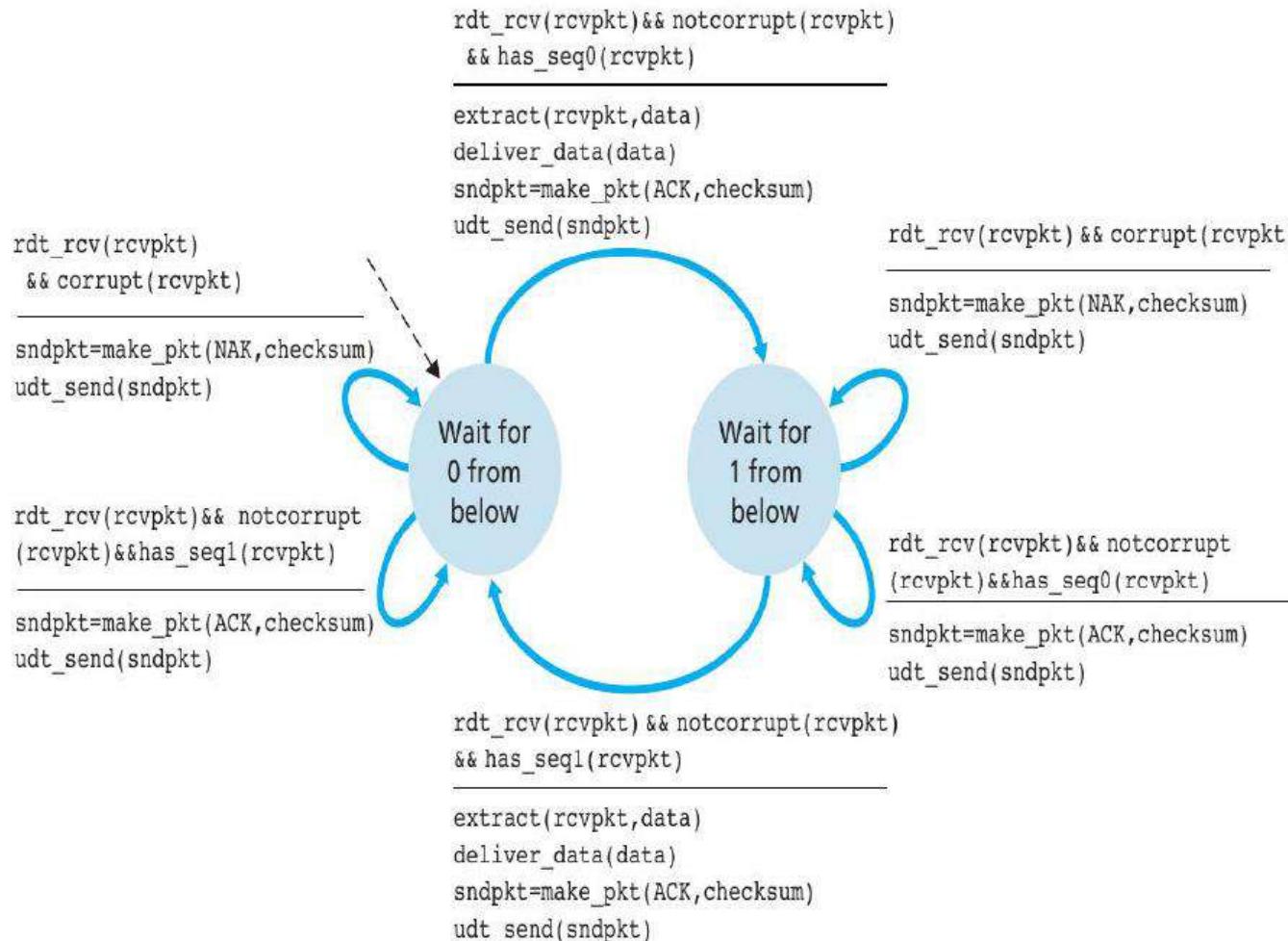


b. rdt2.0: receiving side

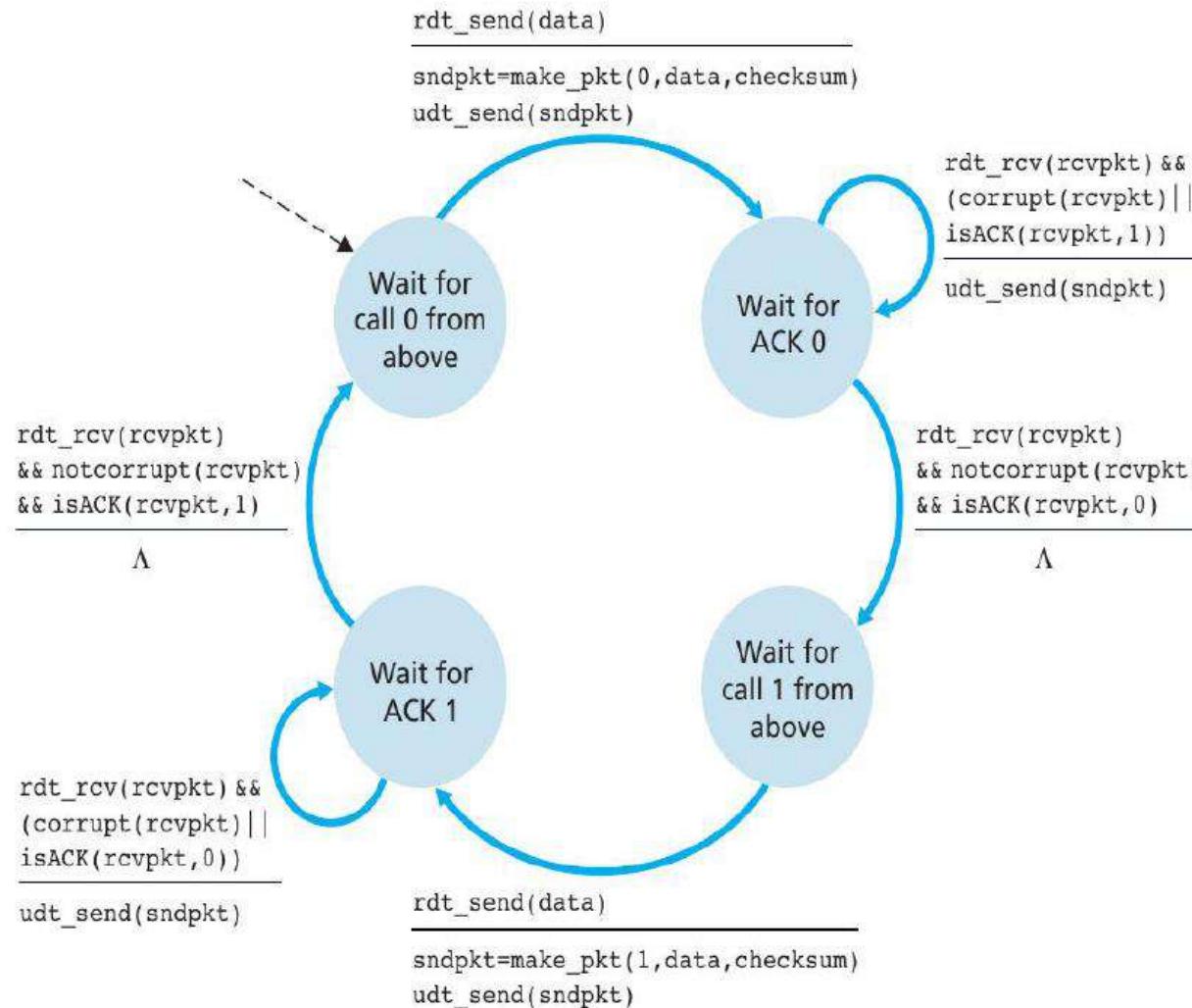
RDT 2.1 Sender



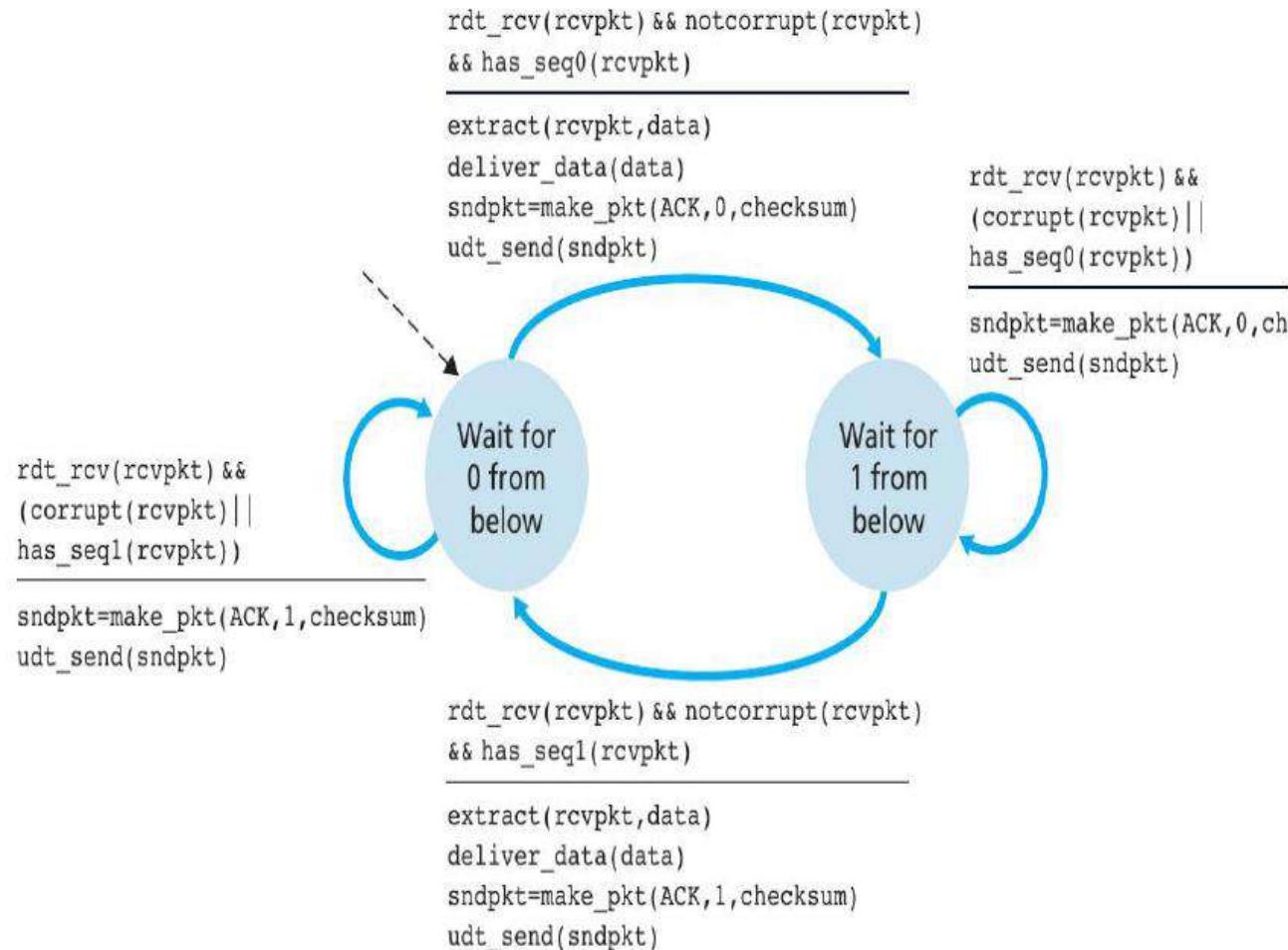
RDT 2.1 Receiver



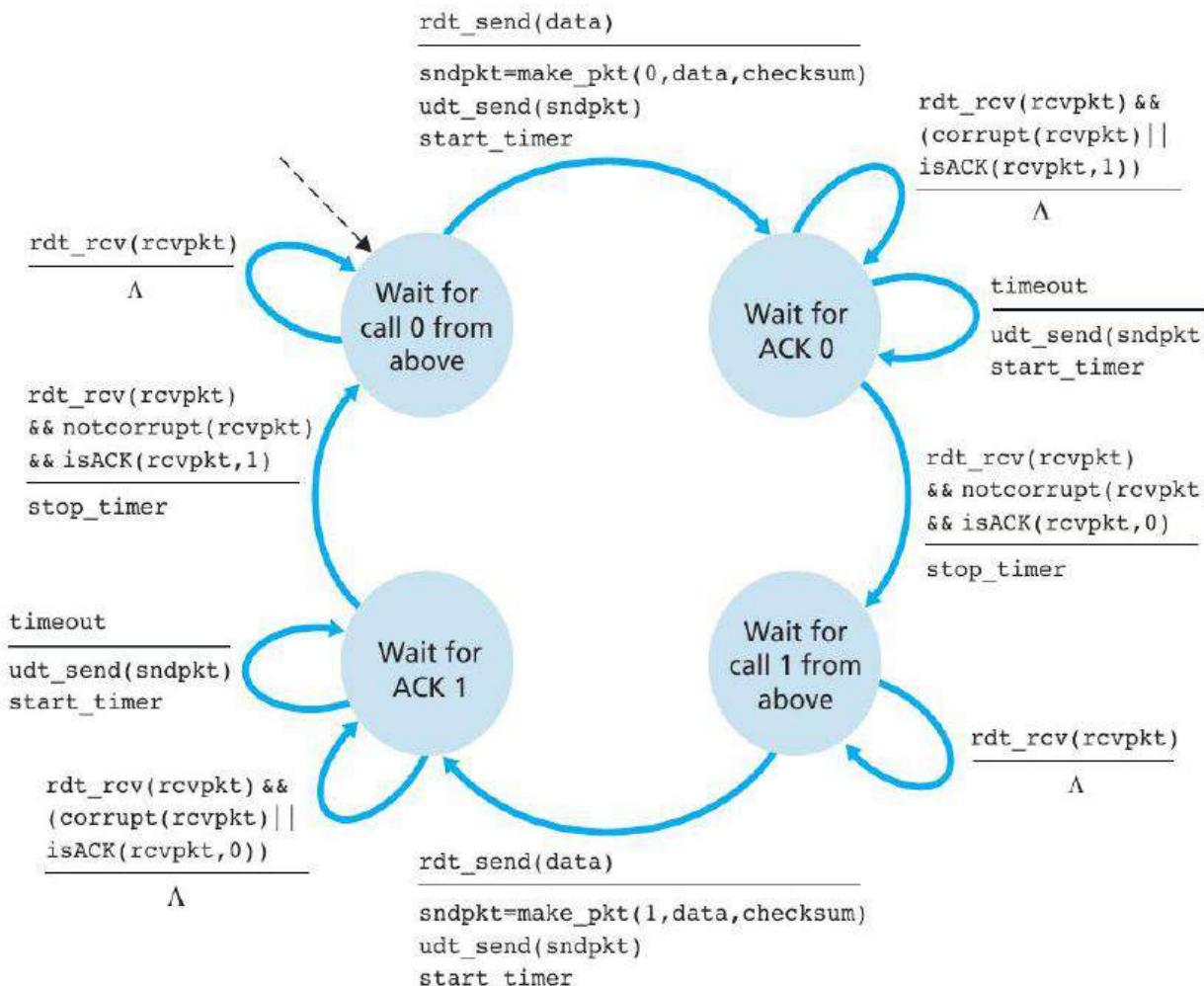
RDT Over a Lossy Channel with Bit Errors: rdt 2.2 sender



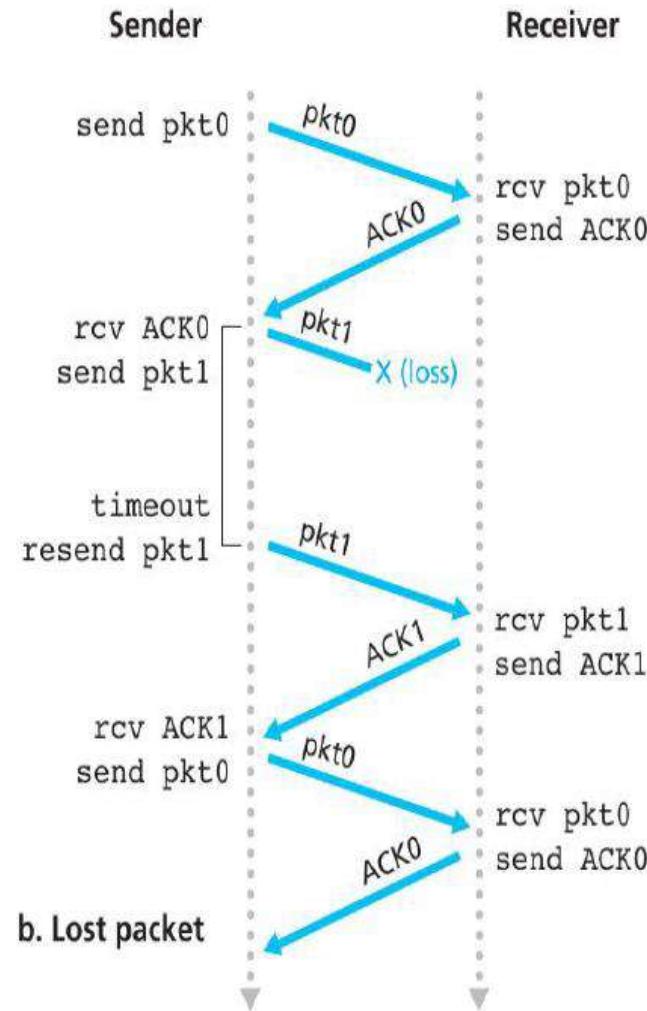
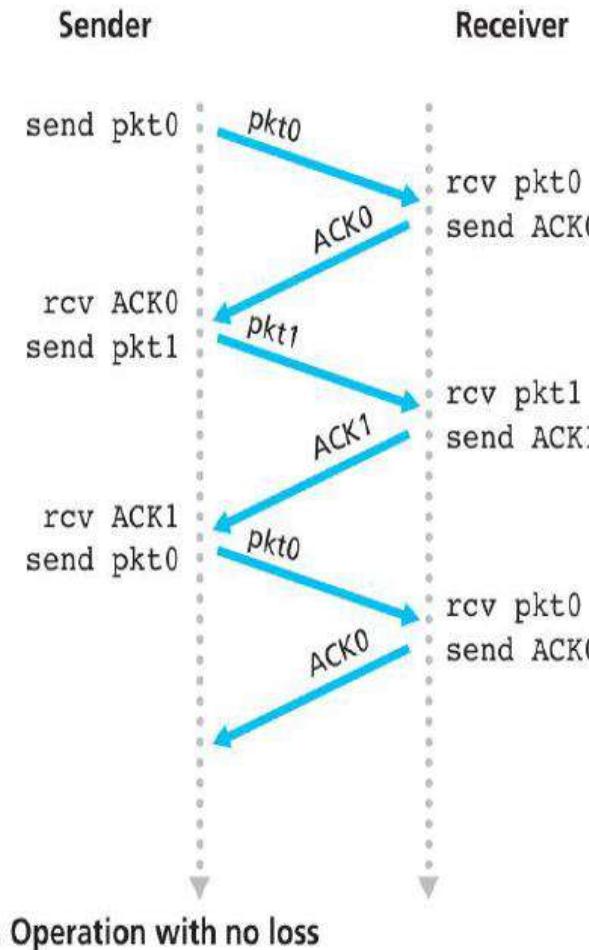
RDT Over a Lossy Channel with Bit Errors: rdt 2.2 receiver



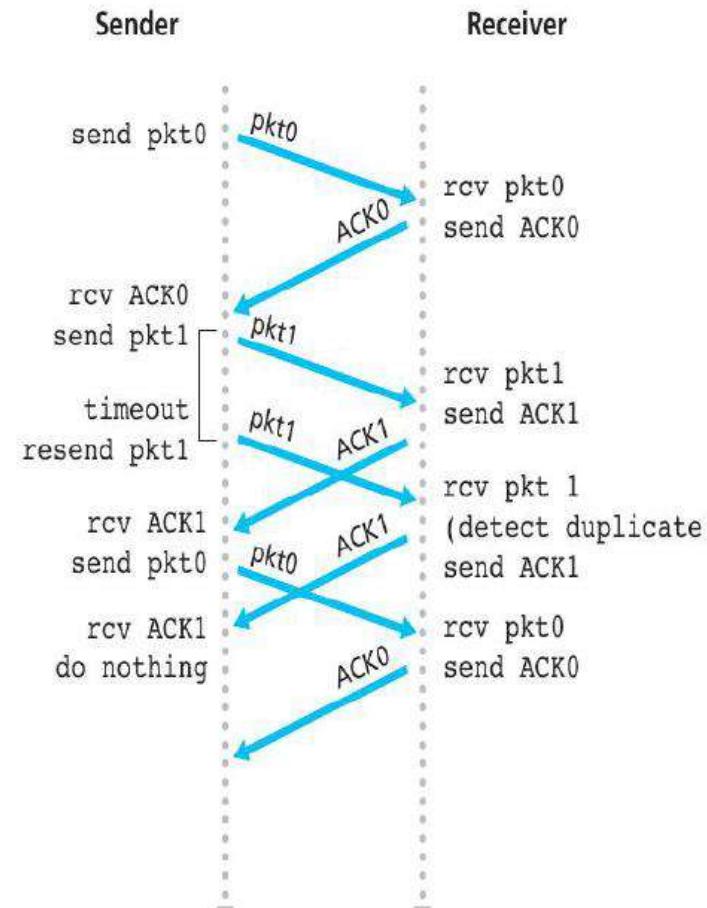
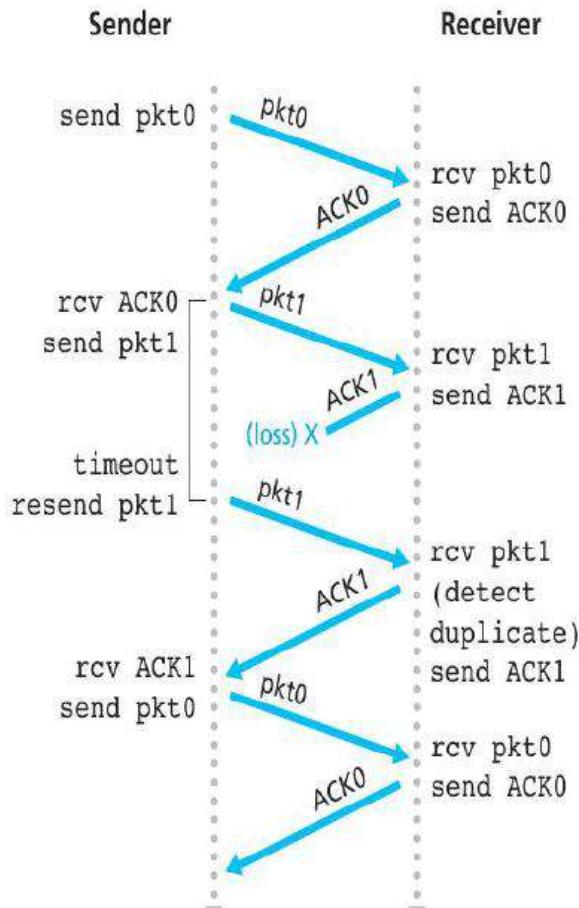
RDT 3.0: NAK-Free



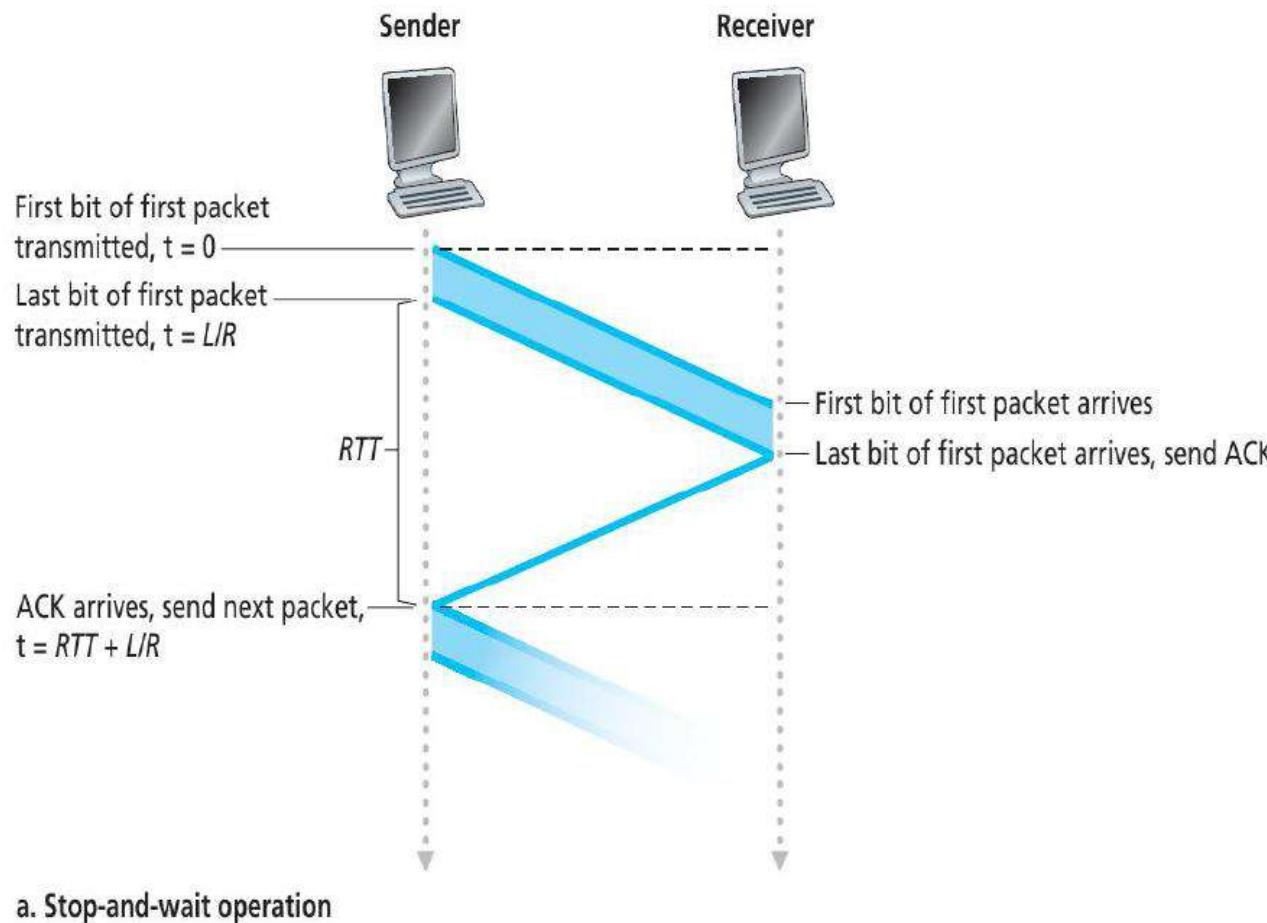
RDT 3.0-Alternating-bit Protocol: Operation



RDT 3.0-Alternating-bit Protocol: Operation



Stop-and-Wait Operation



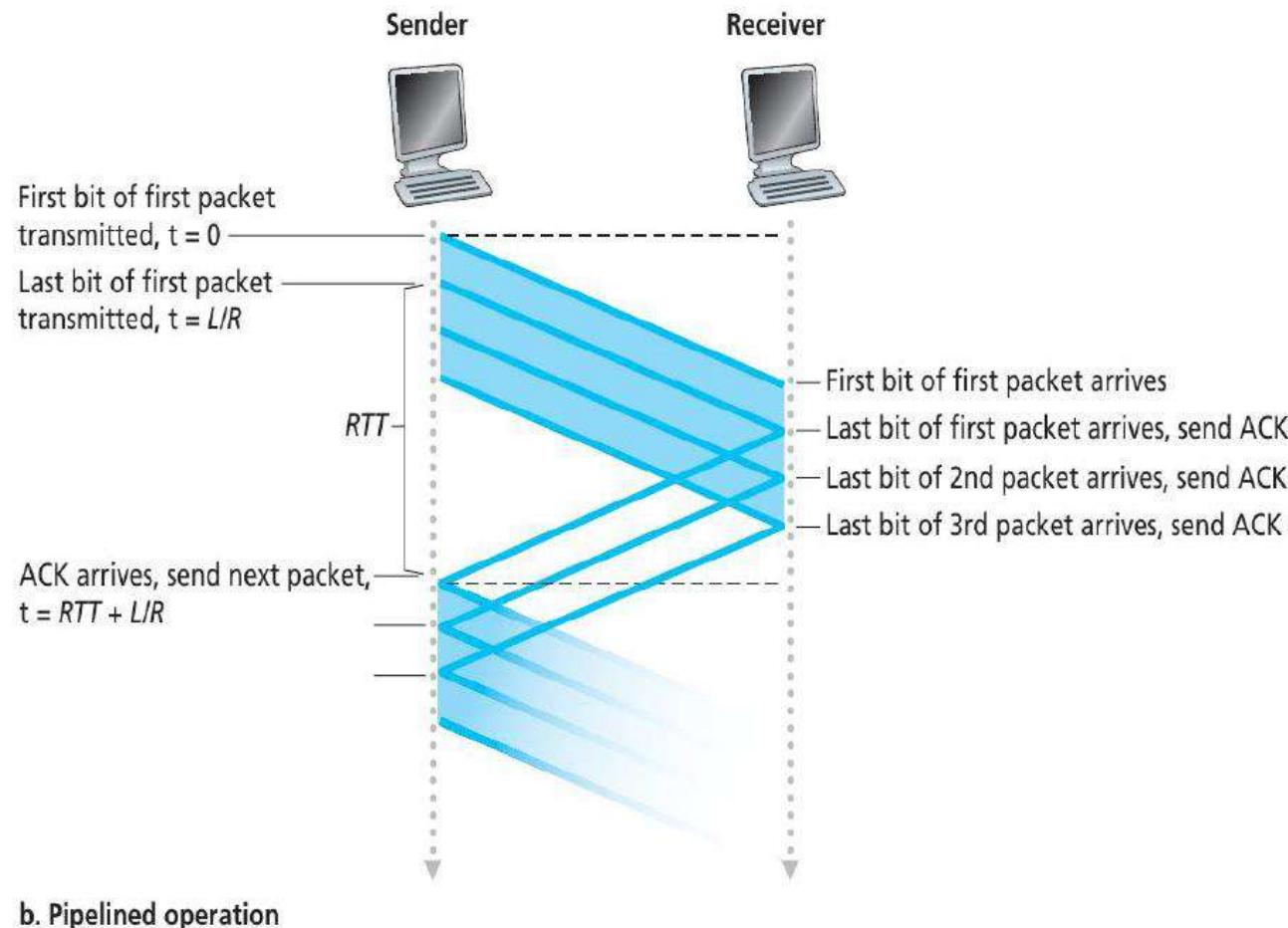
round-trip propagation delay between these two end systems, RTT, is approximately 30ms. connected by a channel with a transmission rate, R , of 1 Gbps

packet size L of 1,000 bytes

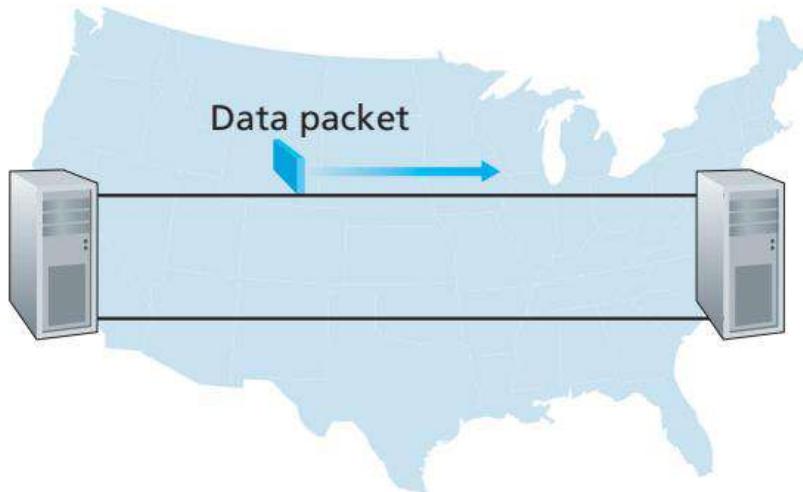
$D_{td} = ?$

sender utilization $U_{\{\text{sender}\}}$?

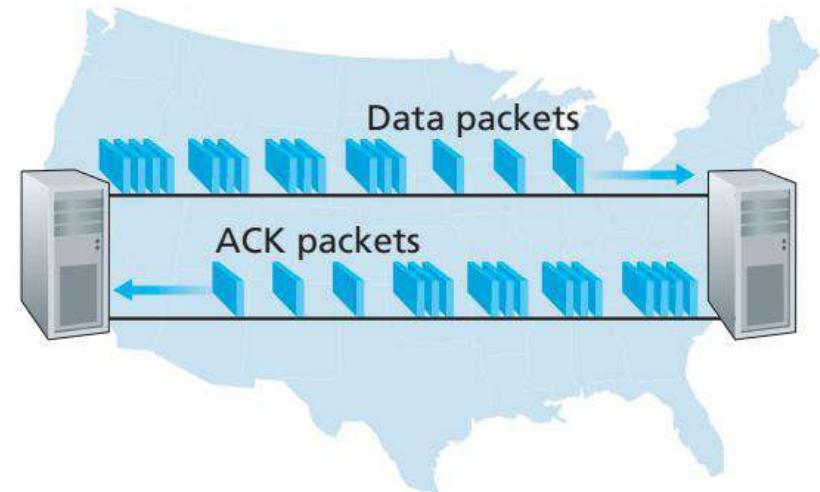
Pipelining



Pipelining



a. A stop-and-wait protocol in operation



b. A pipelined protocol in operation

The range of sequence numbers must be increased: each in-transit packet must have a unique sequence number

- sender and receiver sides of the protocols may have to buffer more than one packet.
- The range of sequence numbers needed and the buffering requirements will depend on the manner in which a data transfer protocol responds to lost, corrupted, and overly delayed packets.
 - Go-Back-N
 - Selective Repeat(SR)

GBN

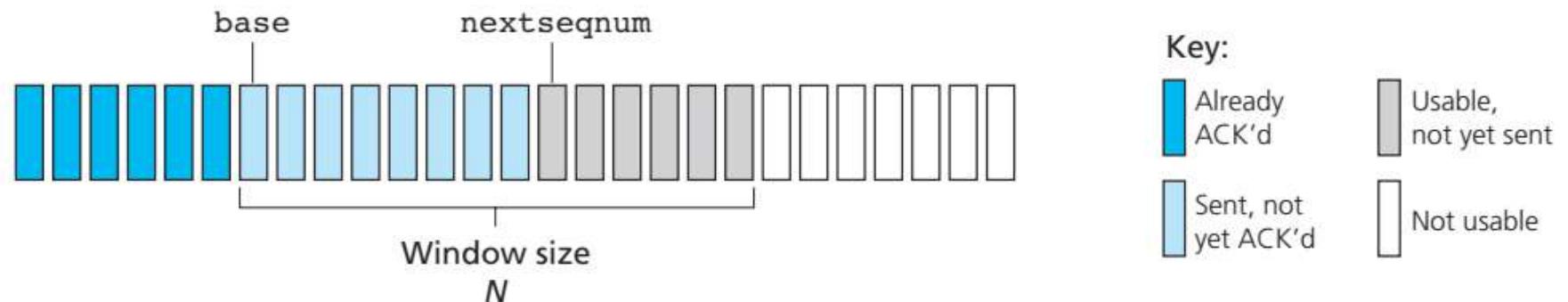
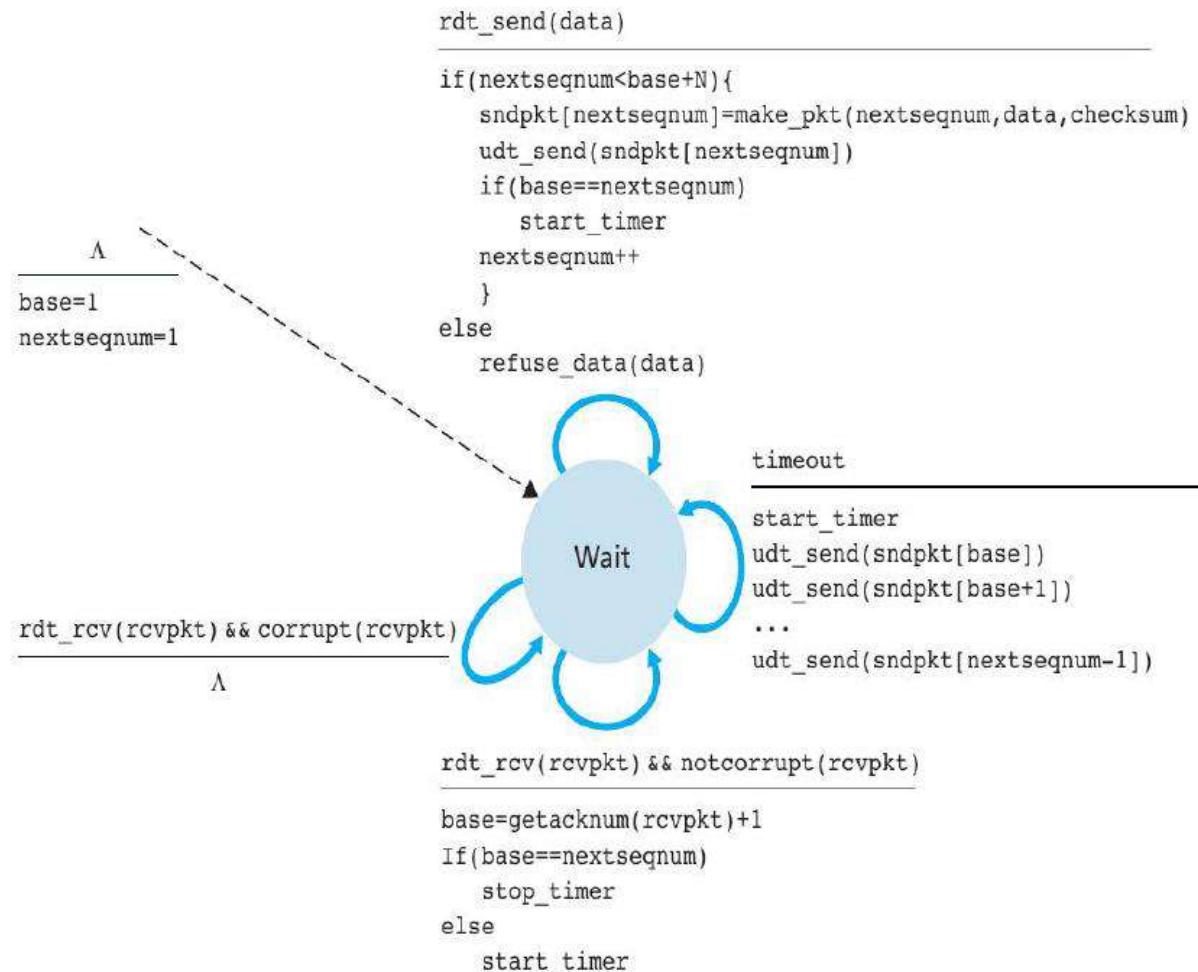


Figure 3.19 ♦ Sender's view of sequence numbers in Go-Back-N

k is the number of bits in the packet sequence number field, range of sequence numbers is $[0, 2^k - 1]$

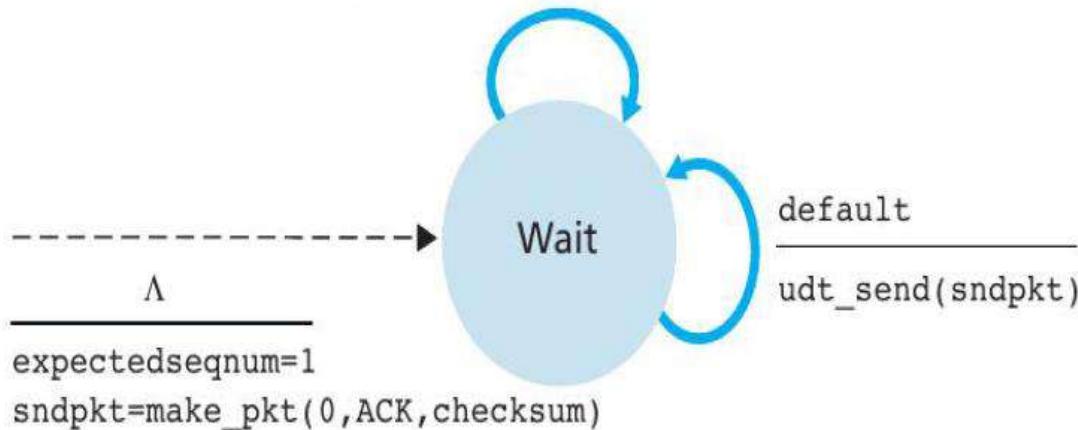
- Sequence numbers in the interval $[0, \text{base}-1]$ → transmitted and acknowledged.
- The interval $[\text{base}, \text{nextseqnum}-1]$ → sent but not yet acknowledged.
- Interval $[\text{nextseqnum}, \text{base}+N-1]$ → packets that can be sent immediately; data from the upper layer.
- Greater than or equal to $\text{base}+N$ cannot be used until an unacknowledged packet currently in the pipeline

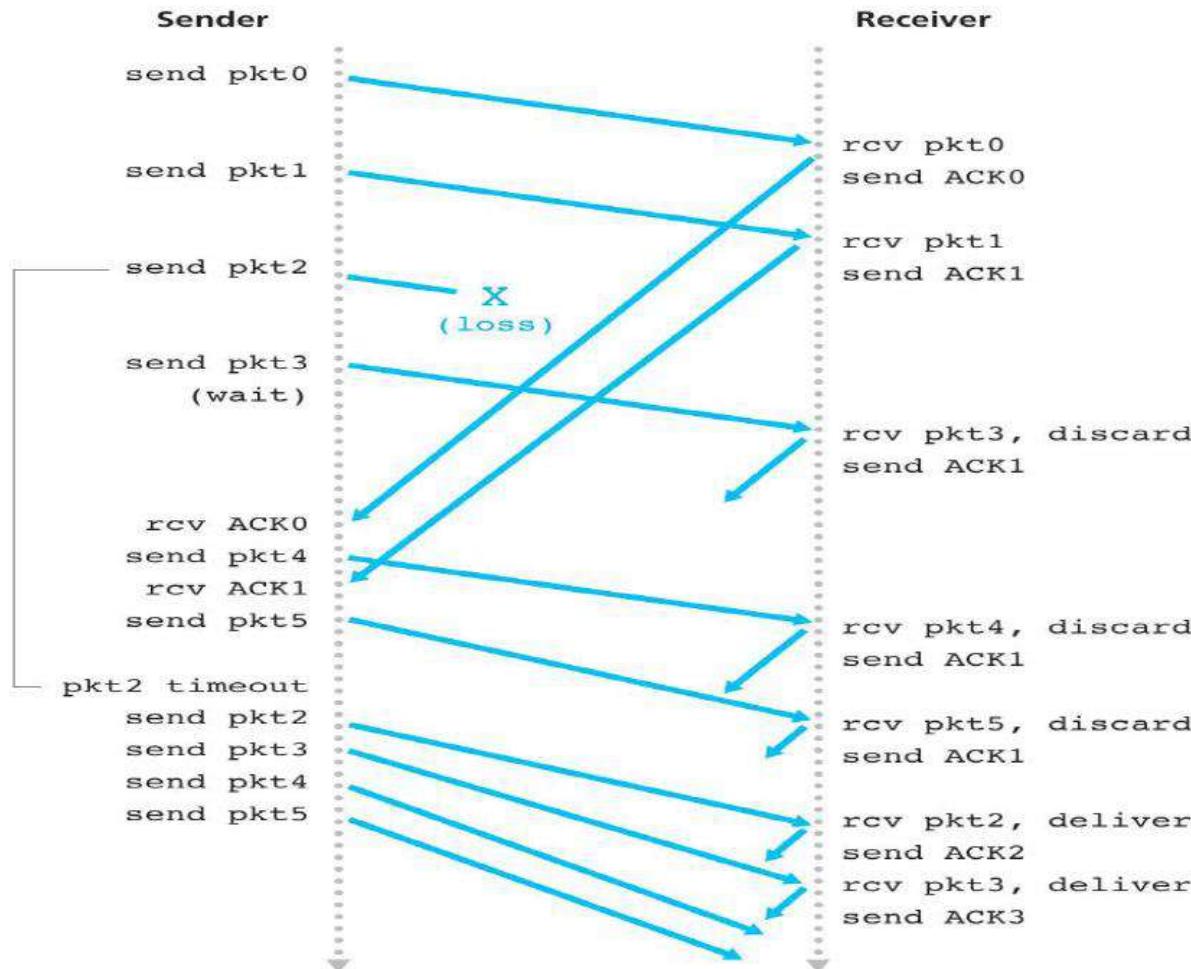
GBN Sender



```
rdt_rcv(rcvpkt)
  && notcorrupt(rcvpkt)
  && hasseqnum(rcvpkt,expectedseqnum)
```

```
extract(rcvpkt,data)
deliver_data(data)
sndpkt=make_pkt(expectedseqnum,ACK,checksum)
udt_send(sndpkt)
expectedseqnum++
```

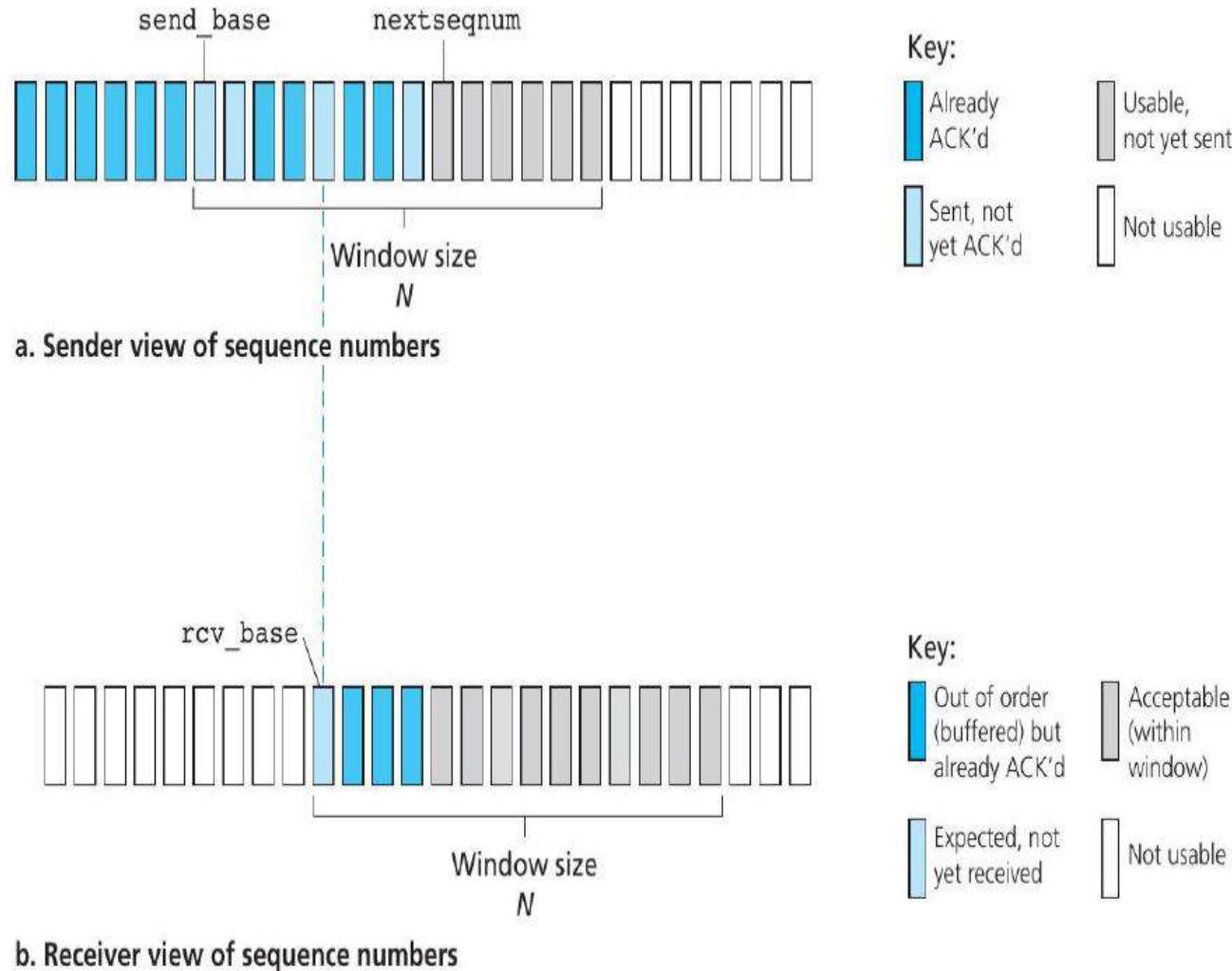




Drawbacks of GBN:

- A single packet error can thus cause GBN to retransmit a large number of packets, many unnecessarily
- probability of channel errors increases, the pipeline can become filled with unnecessary retransmissions

Selective-Repeat

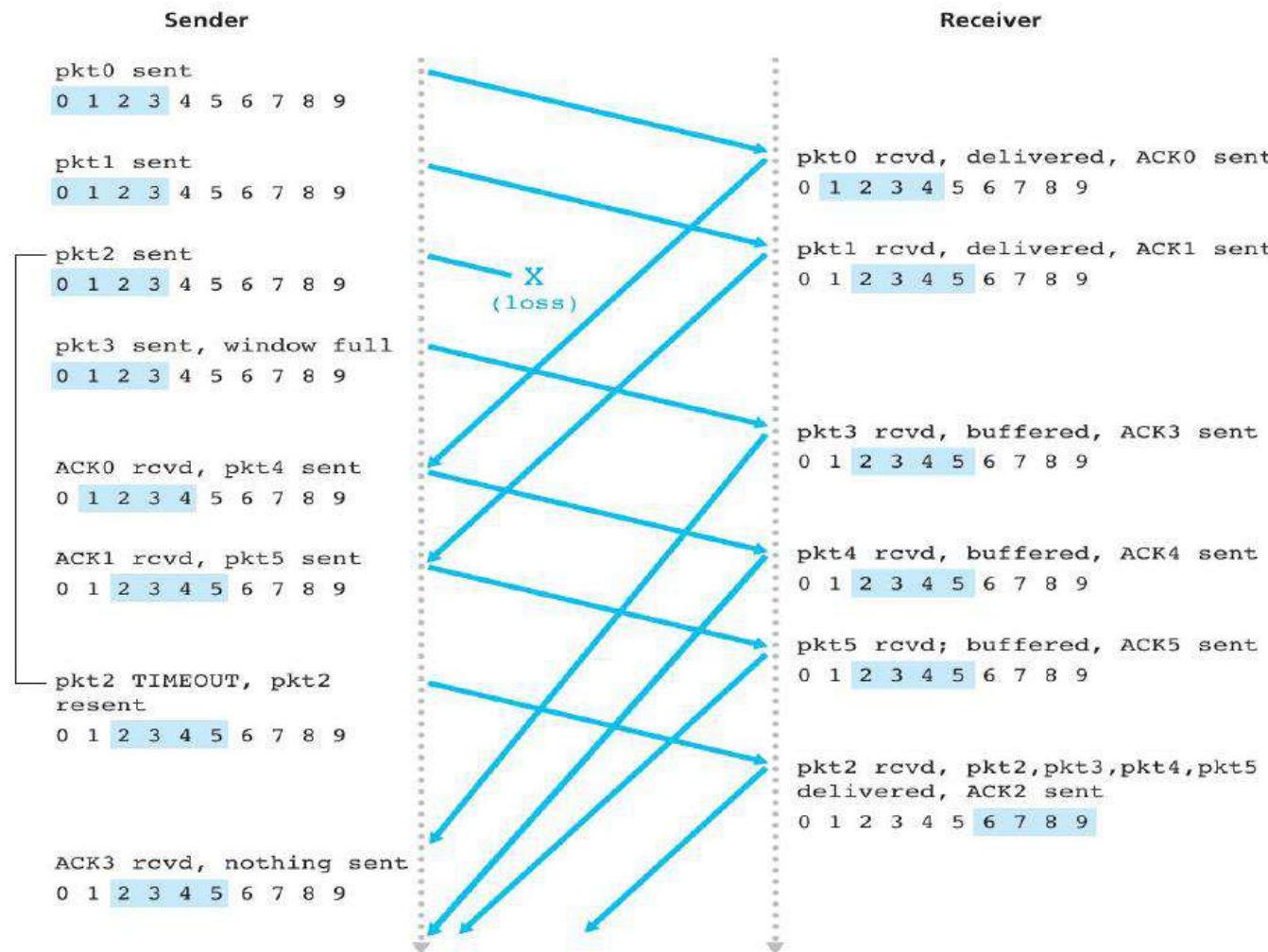


1. *Data received from above.* When data is received from above, the SR sender checks the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent; otherwise it is either buffered or returned to the upper layer for later transmission, as in GBN.
2. *Timeout.* Timers are again used to protect against lost packets. However, each packet must now have its own logical timer, since only a single packet will be transmitted on timeout. A single hardware timer can be used to mimic the operation of multiple logical timers [Varghese 1997].
3. *ACK received.* If an ACK is received, the SR sender marks that packet as having been received, provided it is in the window. If the packet's sequence number is equal to `send_base`, the window base is moved forward to the unacknowledged packet with the smallest sequence number. If the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these packets are transmitted.

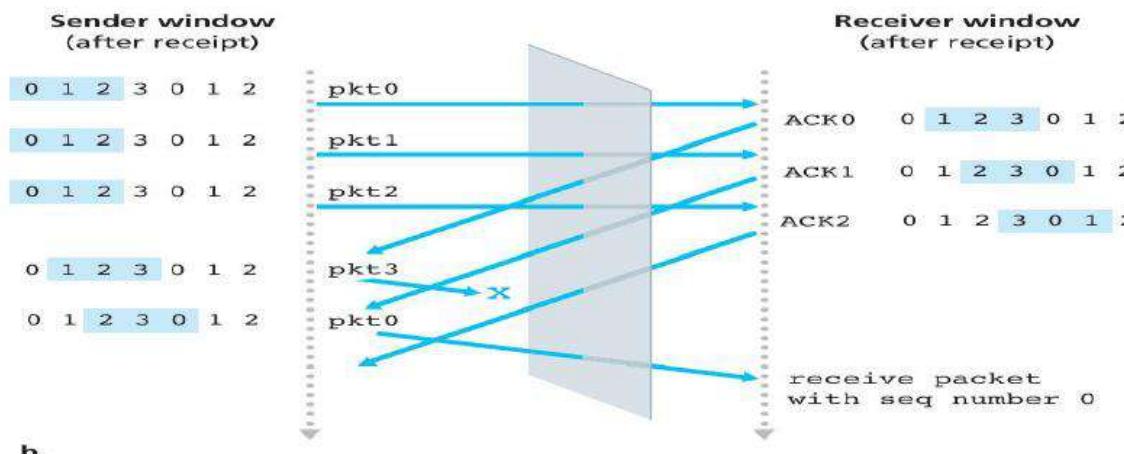
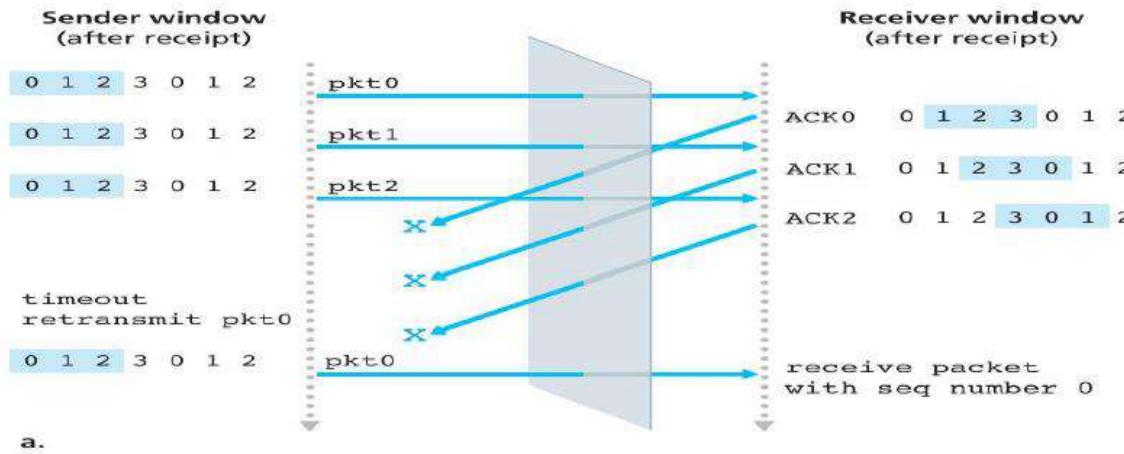
SR receiver events and actions

1. *Packet with sequence number in [`rcv_base`, `rcv_base+N-1`] is correctly received.* In this case, the received packet falls within the receiver's window and a selective ACK packet is returned to the sender. If the packet was not previously received, it is buffered. If this packet has a sequence number equal to the base of the receive window (`rcv_base` in Figure 3.22), then this packet, and any previously buffered and consecutively numbered (beginning with `rcv_base`) packets are delivered to the upper layer. The receive window is then moved forward by the number of packets delivered to the upper layer. As an example, consider Figure 3.26. When a packet with a sequence number of `rcv_base=2` is received, it and packets 3, 4, and 5 can be delivered to the upper layer.
2. *Packet with sequence number in [`rcv_base-N`, `rcv_base-1`] is correctly received.* In this case, an ACK must be generated, even though this is a packet that the receiver has previously acknowledged.
3. *Otherwise.* Ignore the packet.

SR Operation



Window Size in SR





Computer Communication Networks

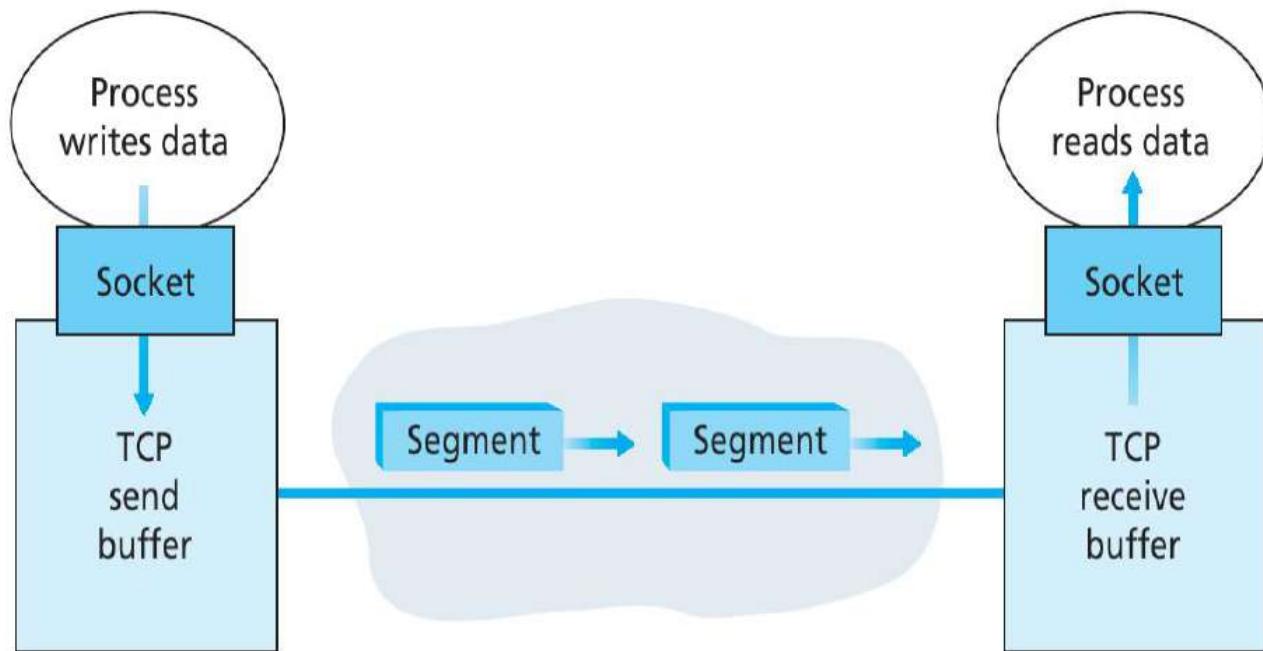
Transport Layer

Dr. Raja Vara Prasad

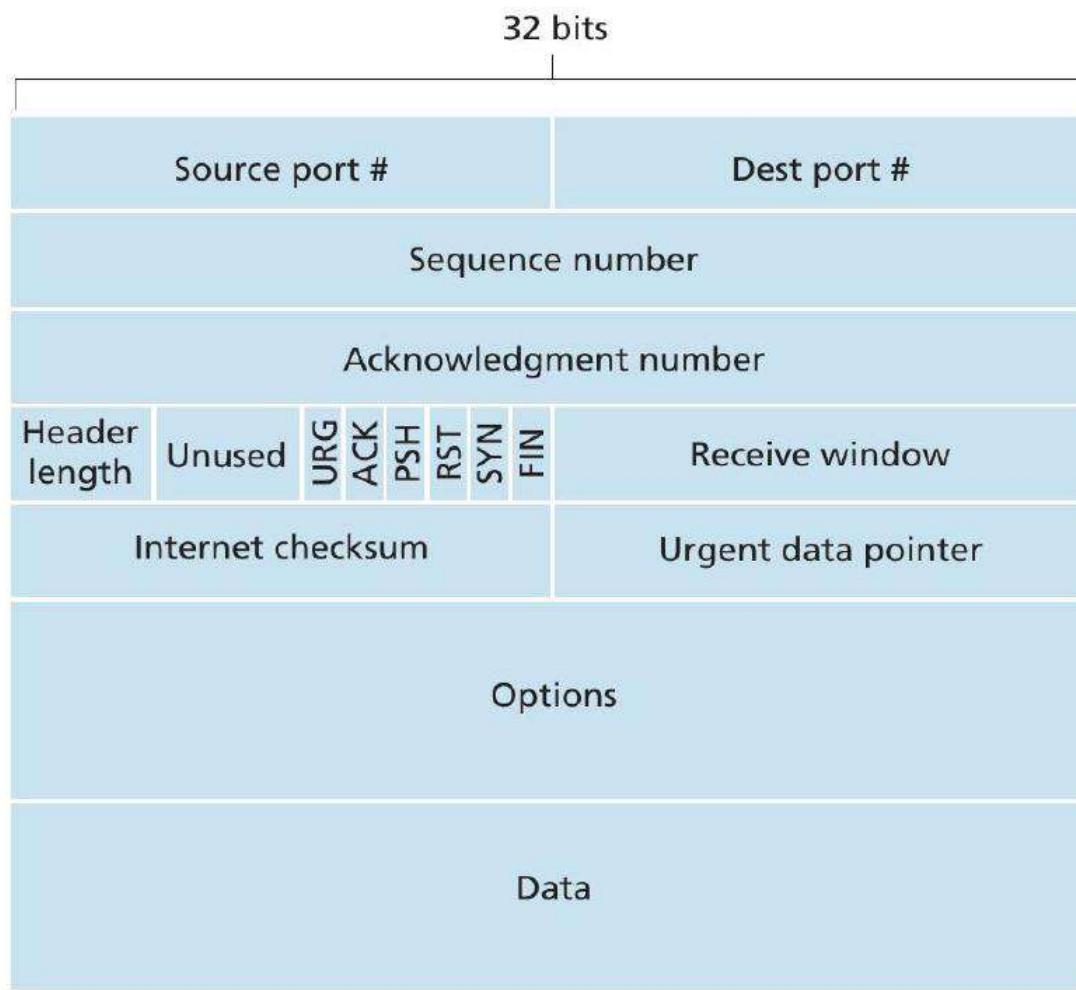
Assistant Professor

IIIT Sri City

- TCP is a **full duplex** service
- No **multicasting**
- Maximum segment size (MSS) is the maximum amount of **data** that a TCP segment can contain.



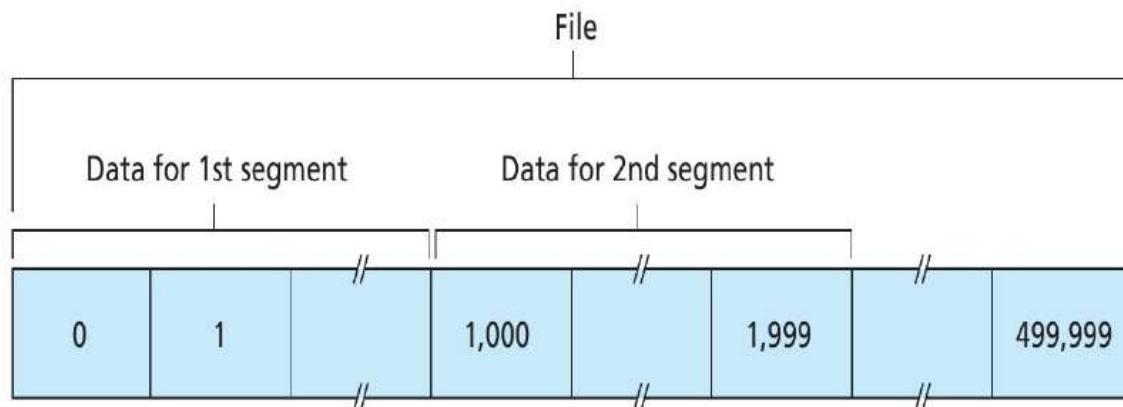
TCP Segment



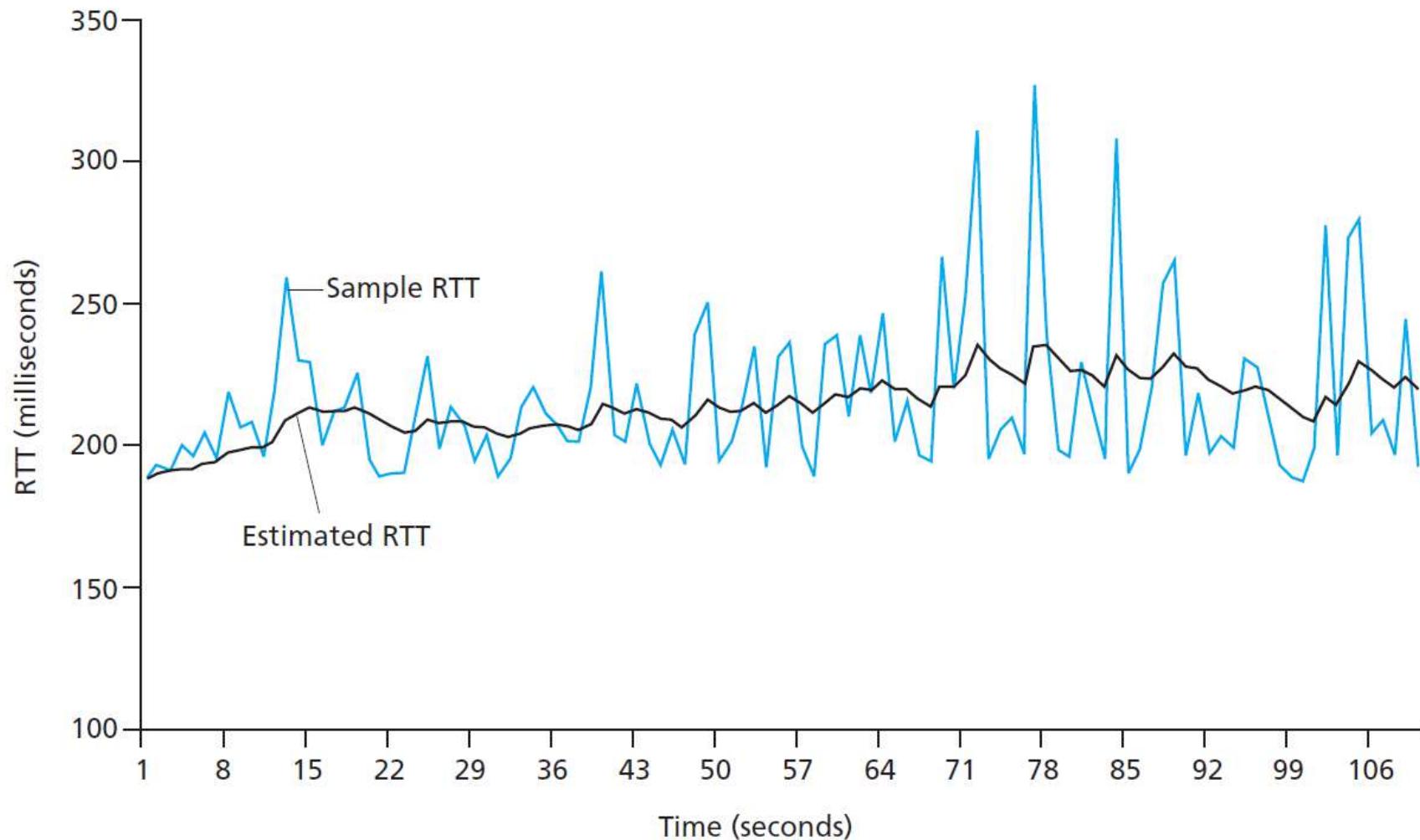
- The 16-bit **receive window** indicates the number of bytes that a receiver is willing to accept
- **Header length** field is 4-bytes, specifies the length of the TCP header in **32-bit words**.
- **Options** are used to negotiate MSS, include time-stamping, etc.
- The **flag field** contains 6 bits, **RST**, **SYN**, **FIN** are used for connection setup and teardown.
- **PSH** indicates that data has to be sent to upper layers immediately.
- **URG** is used to mark the segment as urgent, when it is on there will be a 16-bit **urgent data pointer** field at the end of urgent data.

TCP Sequence Numbers

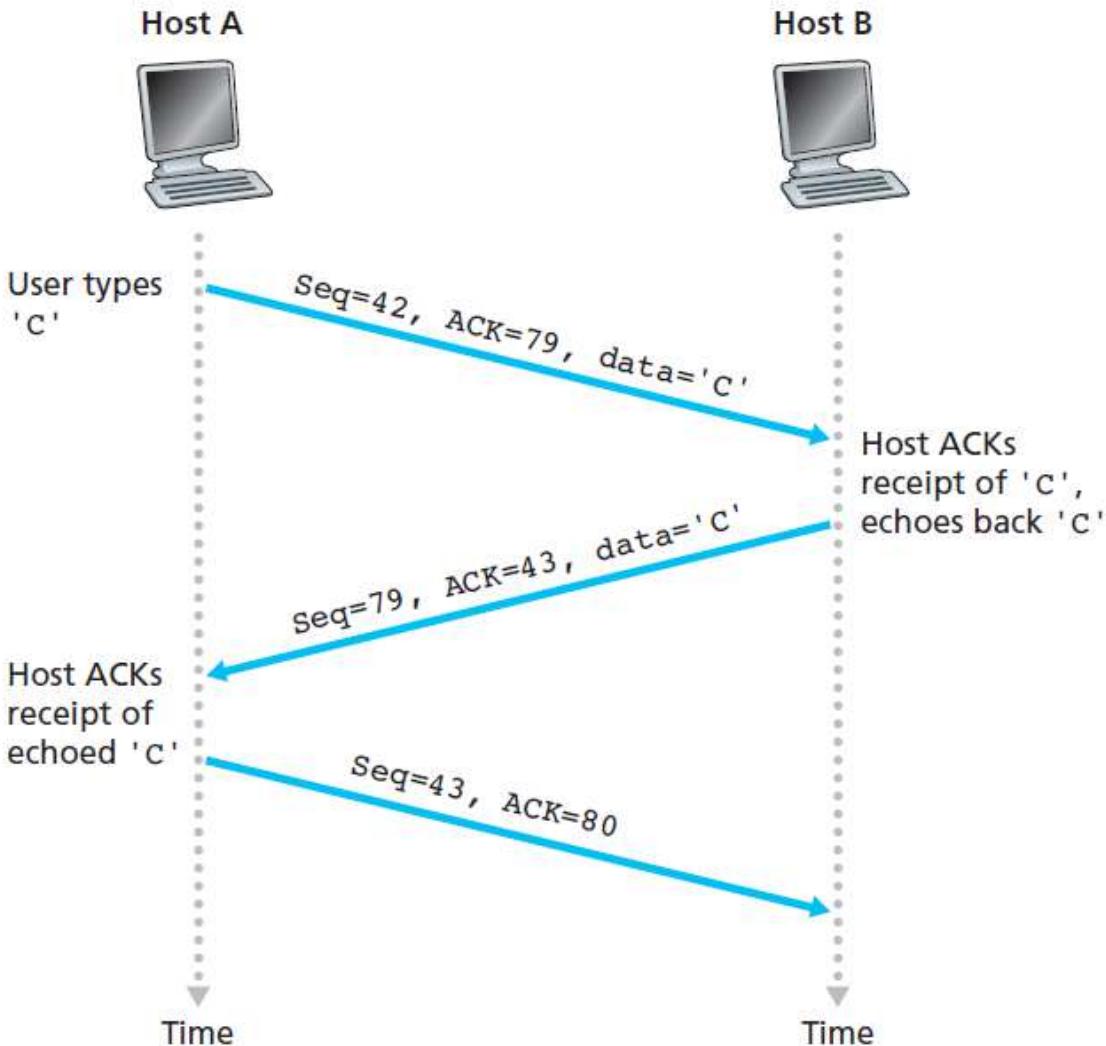
- The sequence number of a segment is the **byte-stream number** of the first byte of data.
- The acknowledge number is the **sequence number of the next byte** that receiver is expecting from source.
- TCP provides **cumulative acknowledgments**; Out-of-order **segements?**
- Sequence numbers may not always start from '0'.



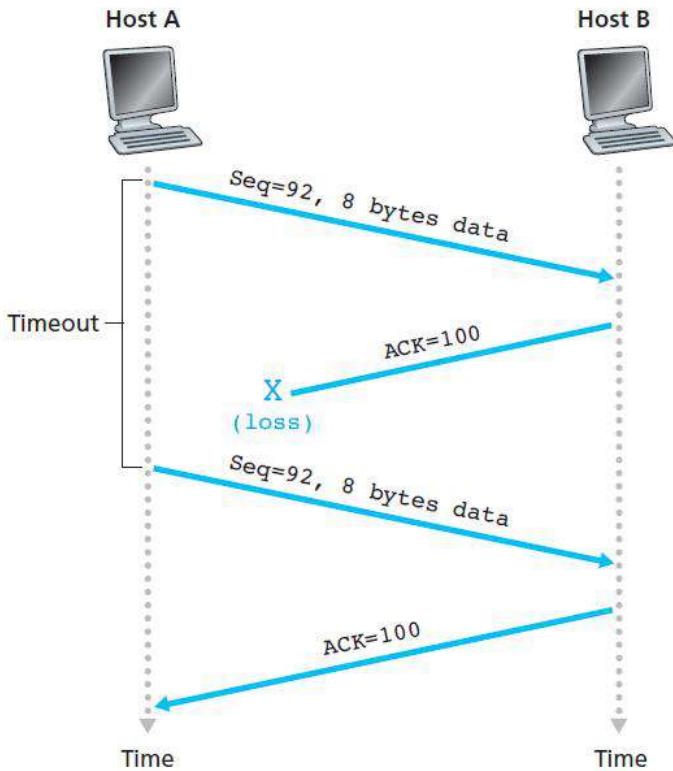
- **SampleRTT**: RTT of a freshly transmitted packet. Computed for each RTT.
- Exponentially weighted moving average: $\text{EstimatedRTT} = (1 - \alpha)\text{EstmiatedRTT} + \alpha \text{ SampleRTT}$
- $\alpha = 0.125$
- $\text{DevRTT} = (1 - \beta) \text{ DevRTT} + \beta | \text{SampleRTT} - \text{EstimatedRTT} |$
- $\beta = 0.25$
- $\text{Timeout} = \text{EstimatedRTT} + 4. \text{ DevRTT}$



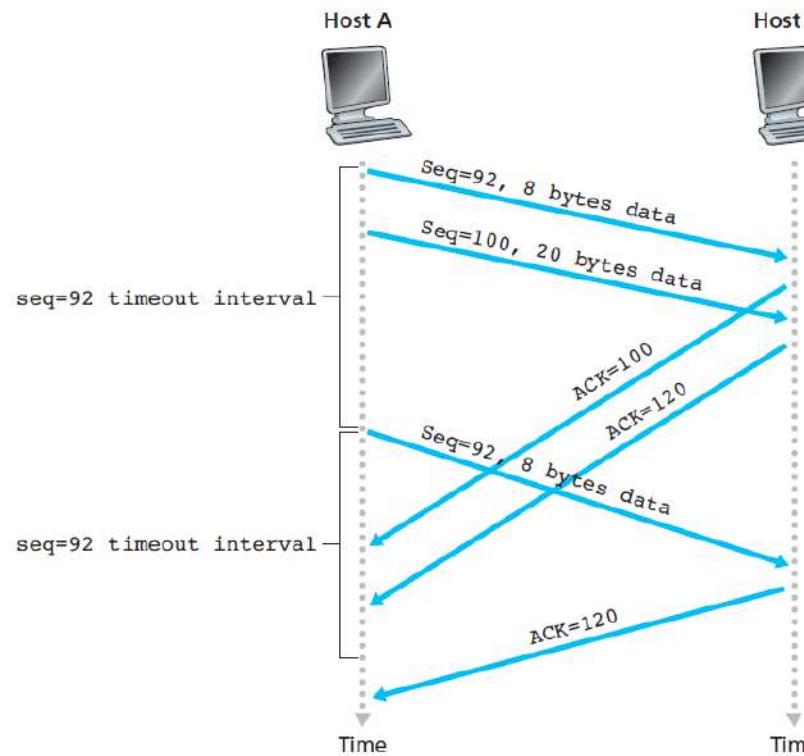
Example: TELNET



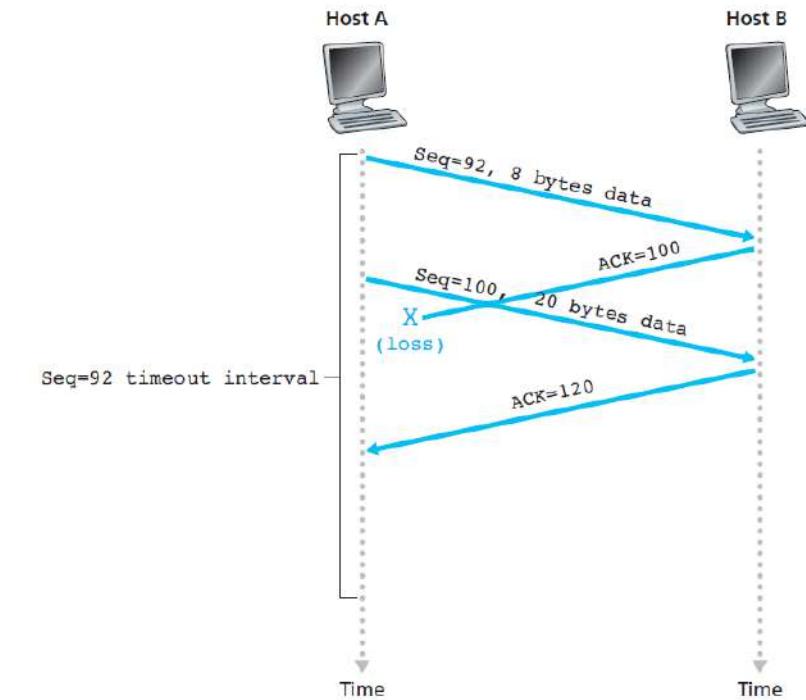
Example: Different scenarios of Reliability aspects



a



b



c

Doubling the Timeout interval:

intervals grow exponentially after each retransmission

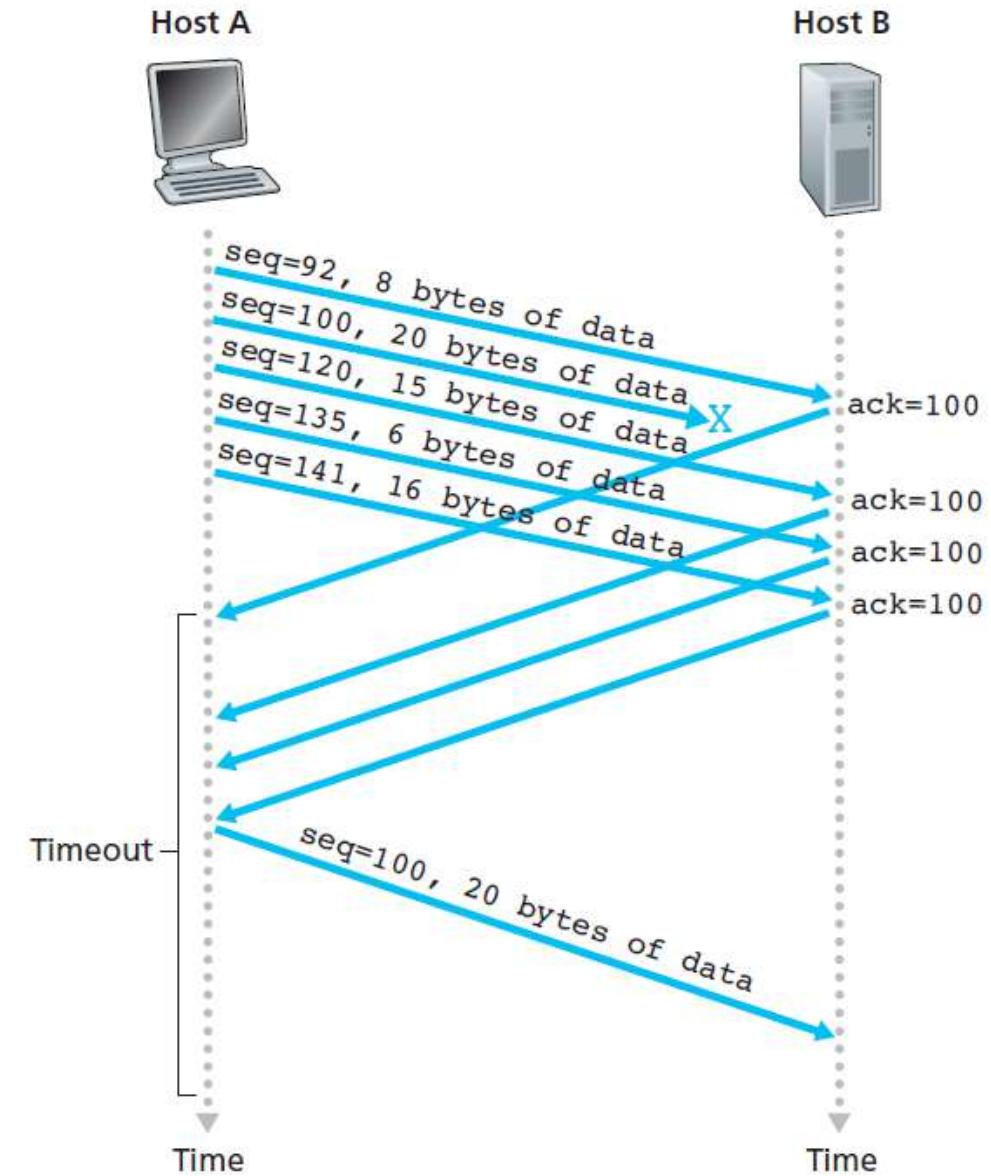
Fast Retransmit:

duplicate ACK multiple times

Example: three duplicate ACKs are received → the TCP sender performs a **fast retransmit**, retransmitting the missing segment *before* that segment's timer expires

Selection: Go-Back-N or Selective Repeat?

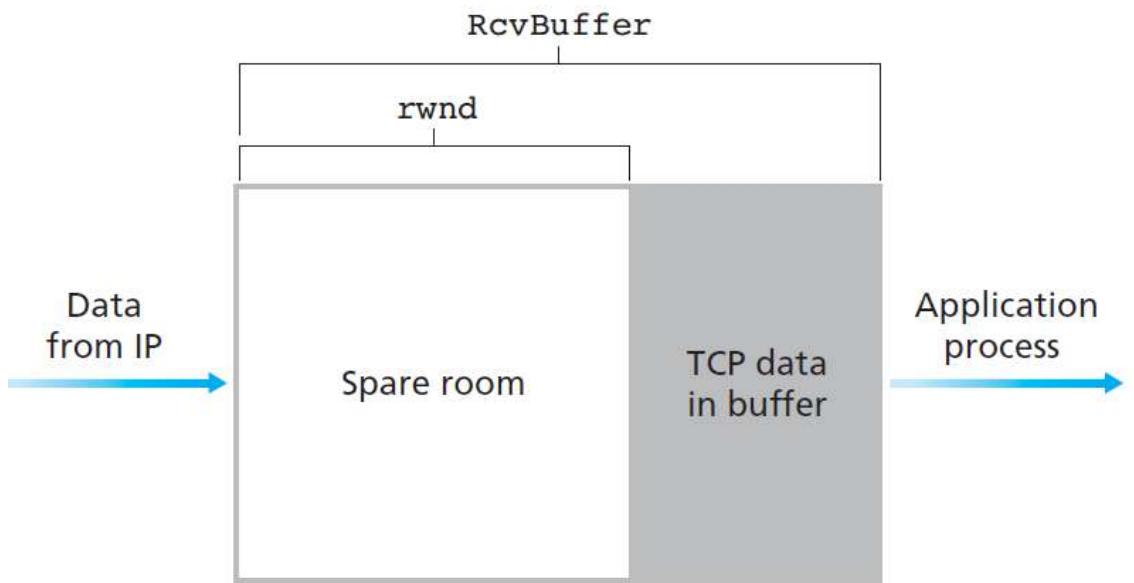
TCP's error-recovery mechanism is probably best categorized as a hybrid of GBN and SR protocols



Flow Control

Flow control is thus a speed-matching service → matching the rate at which the sender is sending against the rate at which the receiving application is reading

receive window: give the sender an idea of how much free buffer space is available at the receiver



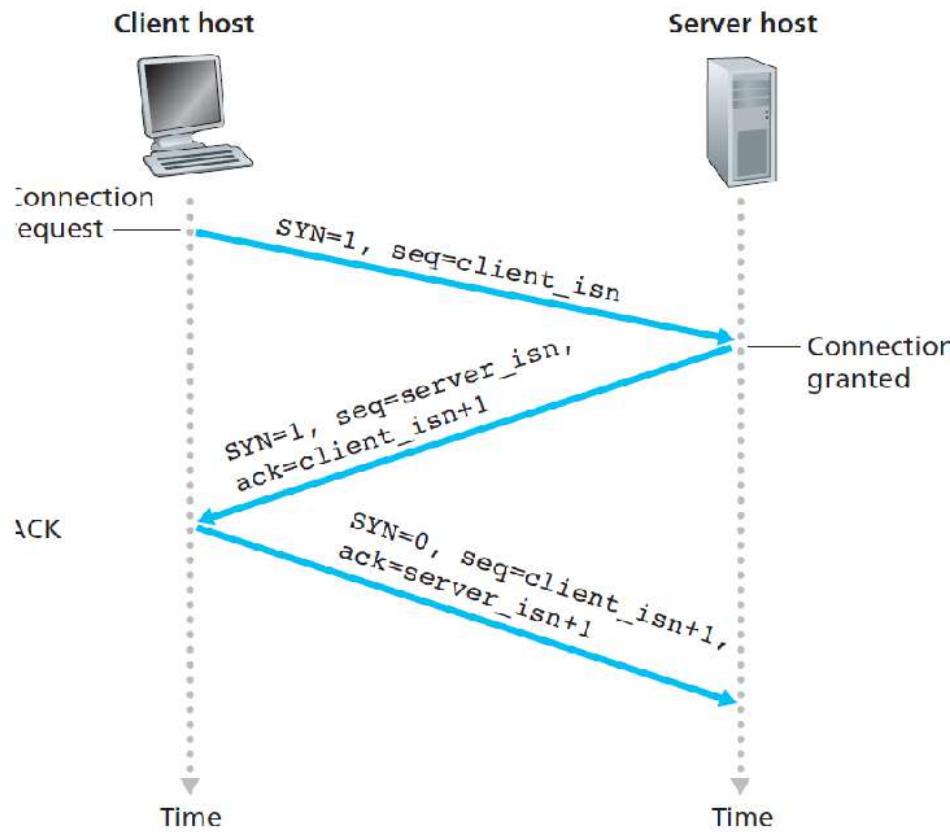
$$\text{LastByteRcvd} - \text{LastByteRead} \leq \text{RcvBuffer}$$

The receive window, denoted `rwnd` is set to the amount of spare room in the buffer:

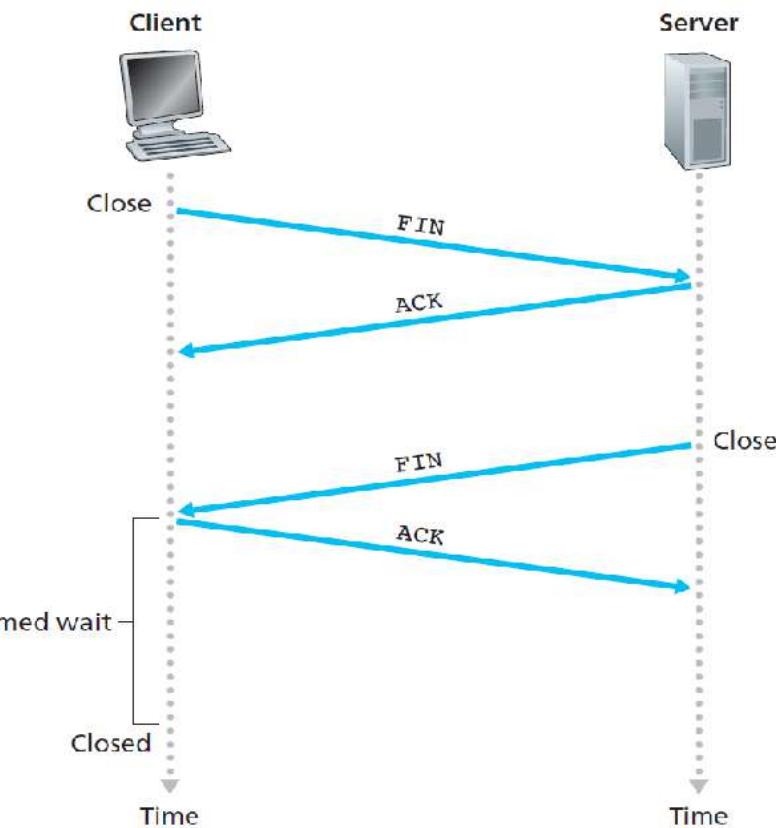
$$\text{rwnd} = \text{RcvBuffer} - [\text{LastByteRcvd} - \text{LastByteRead}]$$

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{rwnd}$$

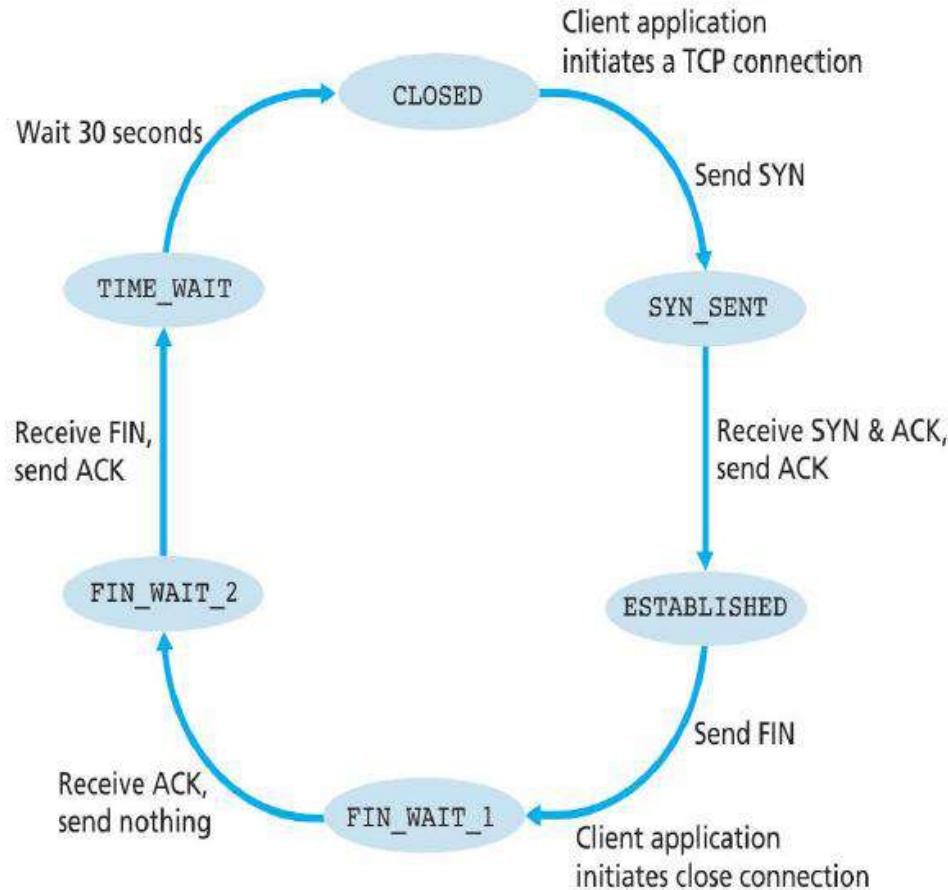
TCP Connection Establishment



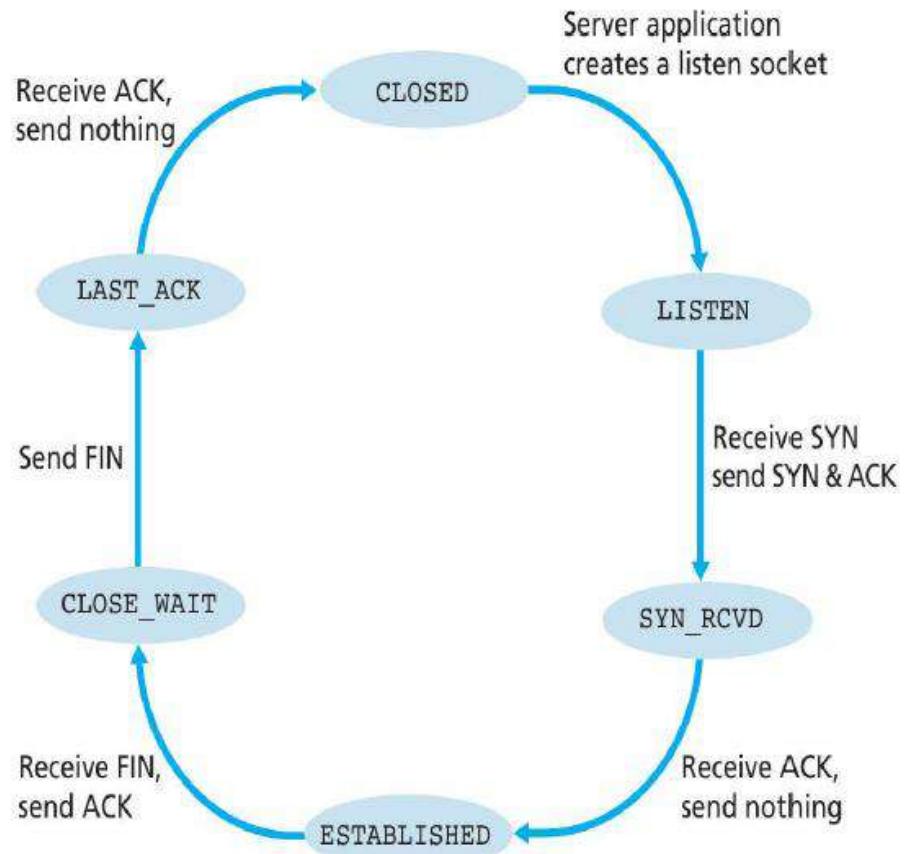
TCP Connection Termination



TCP States at Cleint

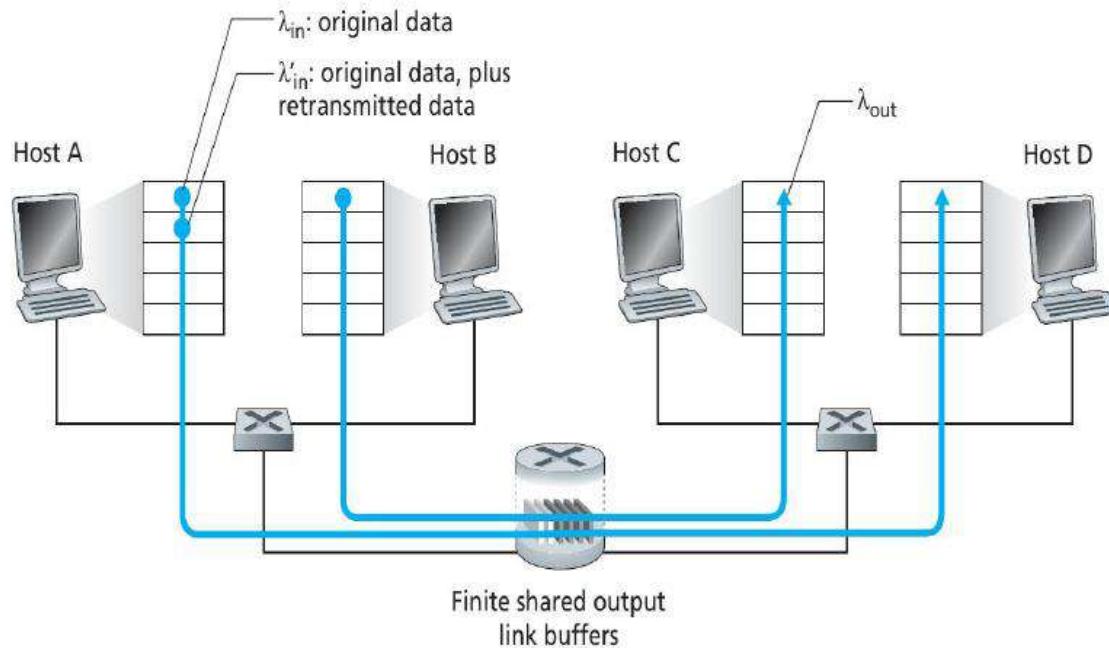


TCP States at Server



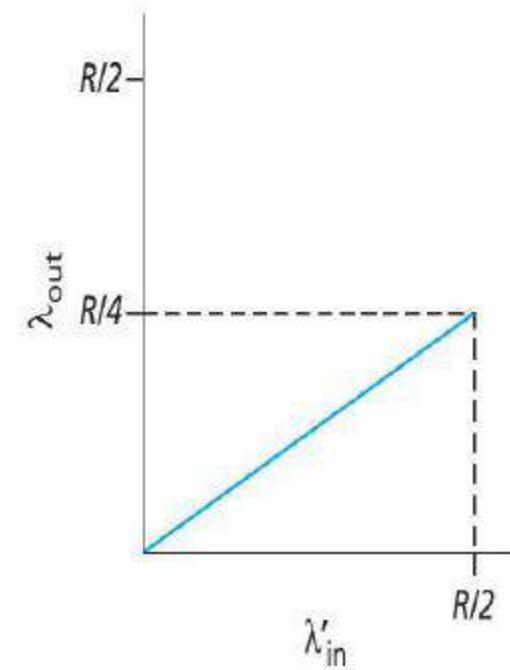
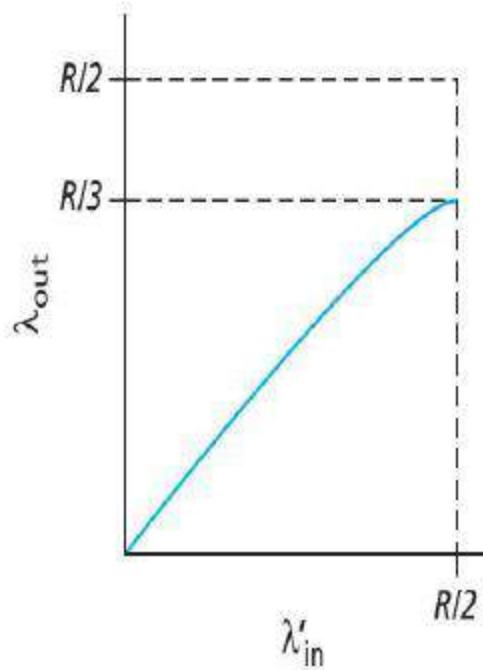
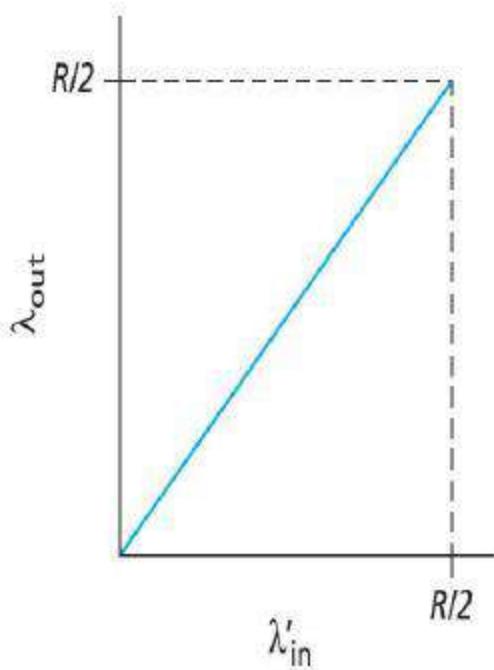
- Why does **Congestion** occur?
- Packet arrival rate at a router is **near or higher** than the **output link capacity**.
- Consequences?
- Buffer overflows, retransmissions to compensate for lost packets
- **Unneeded retransmissions**

Congestion Scenario - 1

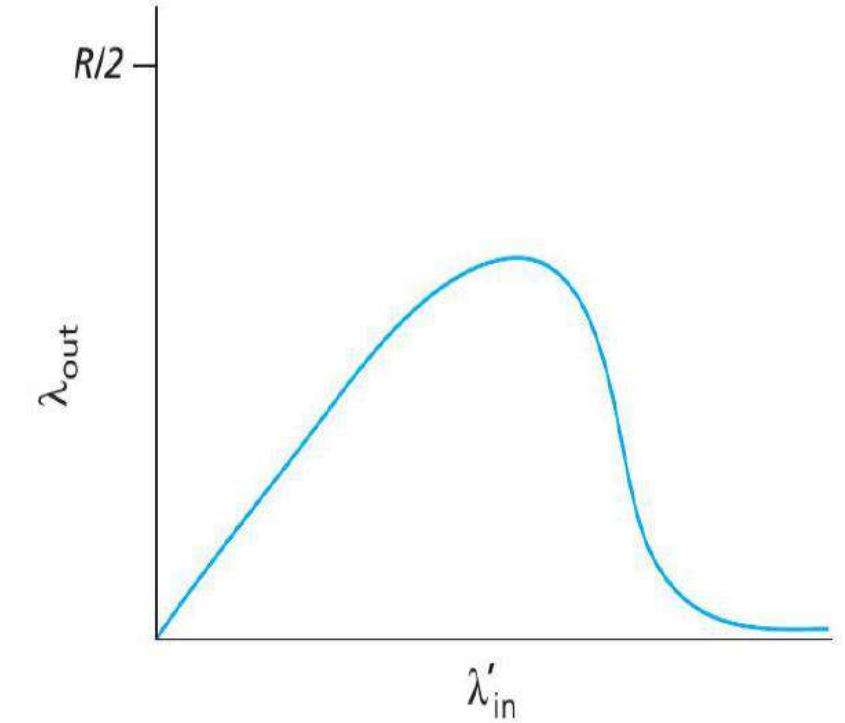
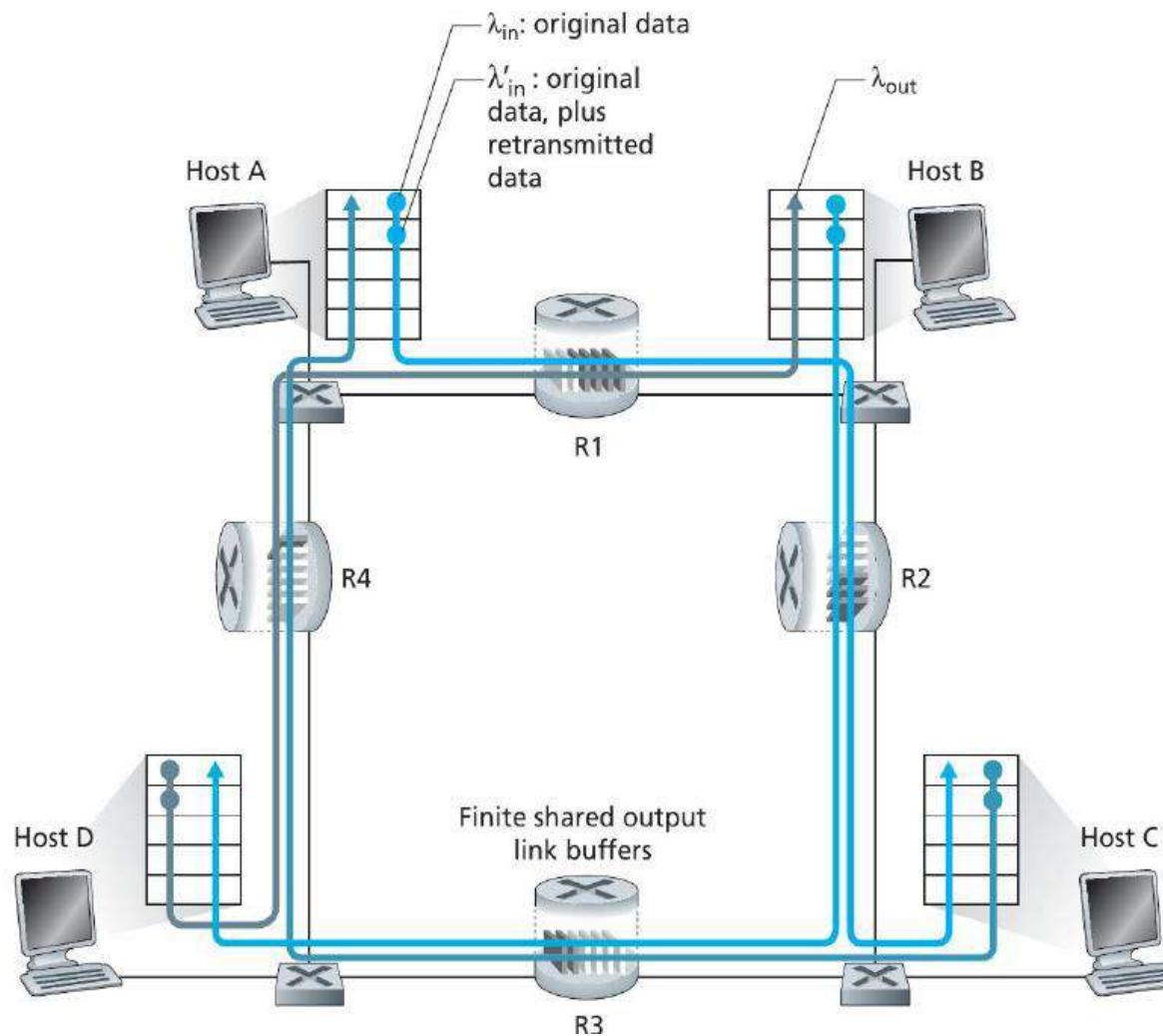


- (a) Host A knows whether buffer in the router has free space or not (**Magic!**)
- (b) Host A retransmits only if it is sure that packet is **lost** (**Someone has to give this information**)
- (c) Host A retransmits on timeouts!

Congestion Scenario - 1



Congestion Scenario - 2



Congestion Control

- End-to-end congestion control: no **explicit** information about congestion the network
- Network-assisted Congestion Control: **Choke packet**
- How does TCP identify congestion?
- No assistance form IP
- Identify congestion through **timeouts** and **duplicate ACKs**

Congestion Control

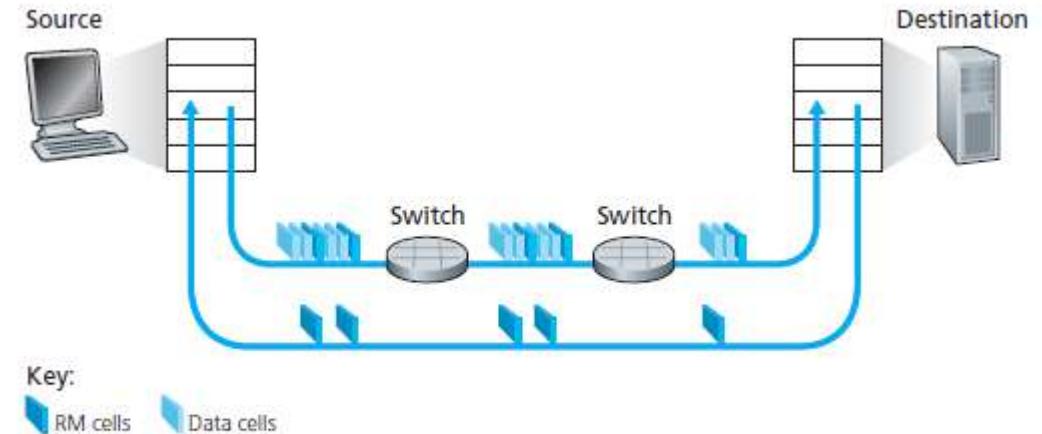
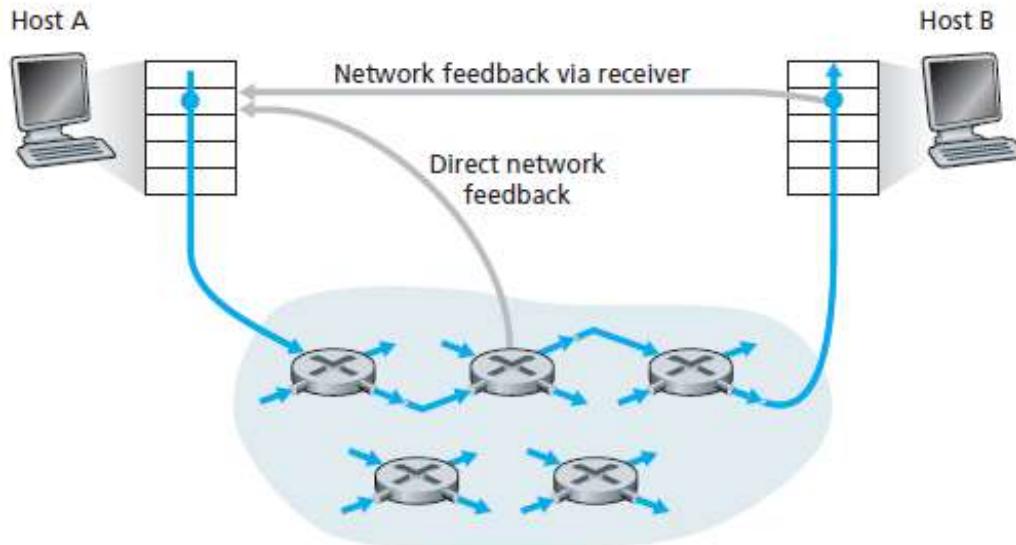


Figure 3.50 Congestion-control framework for ATM ABR service

EFCI bit. Each *data cell* contains an **explicit forward congestion indication (EFCI) bit**.

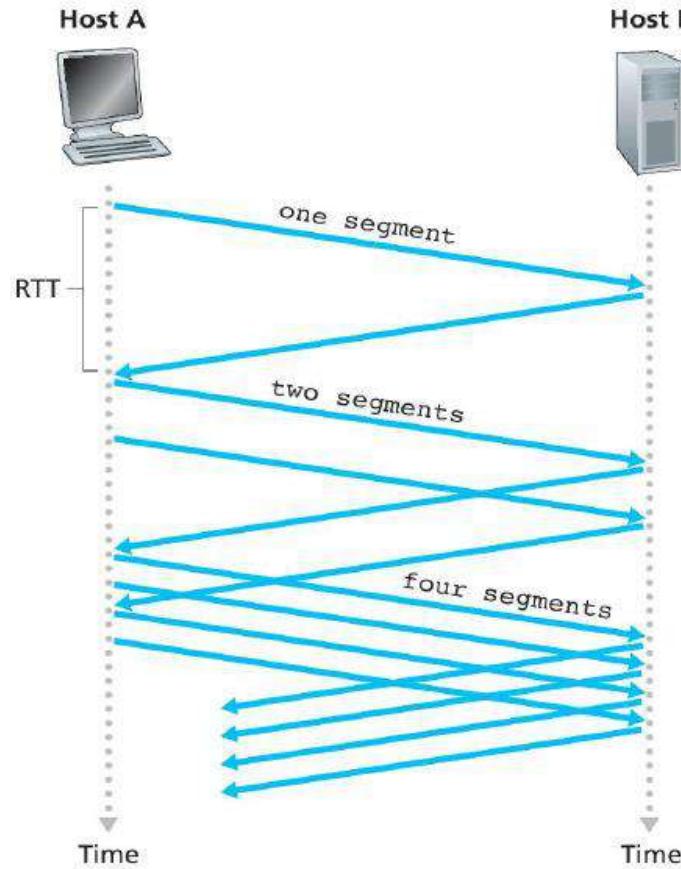
CI and NI bits: **congestion indication (CI) bit** and a **no increase (NI) bit**

ER setting. Each RM cell also contains a 2-byte **explicit rate (ER) field**.

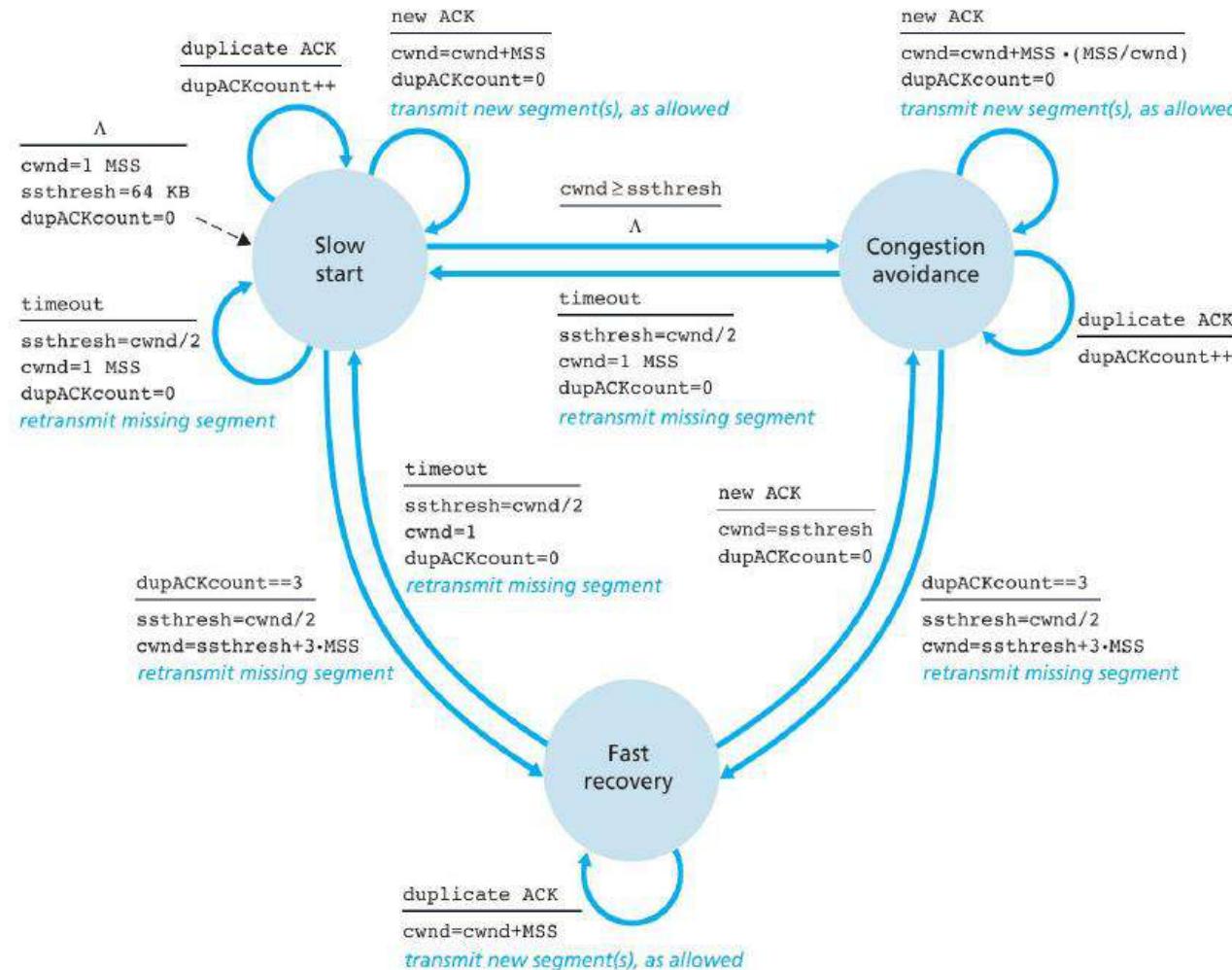
TCP's Congestion Control

- How does TCP control the sending rate?
- Defines a new variable called **cwnd**.
- $\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{cwnd}, \text{rwnd}\}$
- Sends cwnd bytes of information per RTT (approximately)
- Can we adjust the speed? **Slef-clocking**
 - A lost segment triggers the sender to reduce rate of transmission
 - An acknowledgment indicates **all is well!** Increase the rate
 - Bandwidth Probing

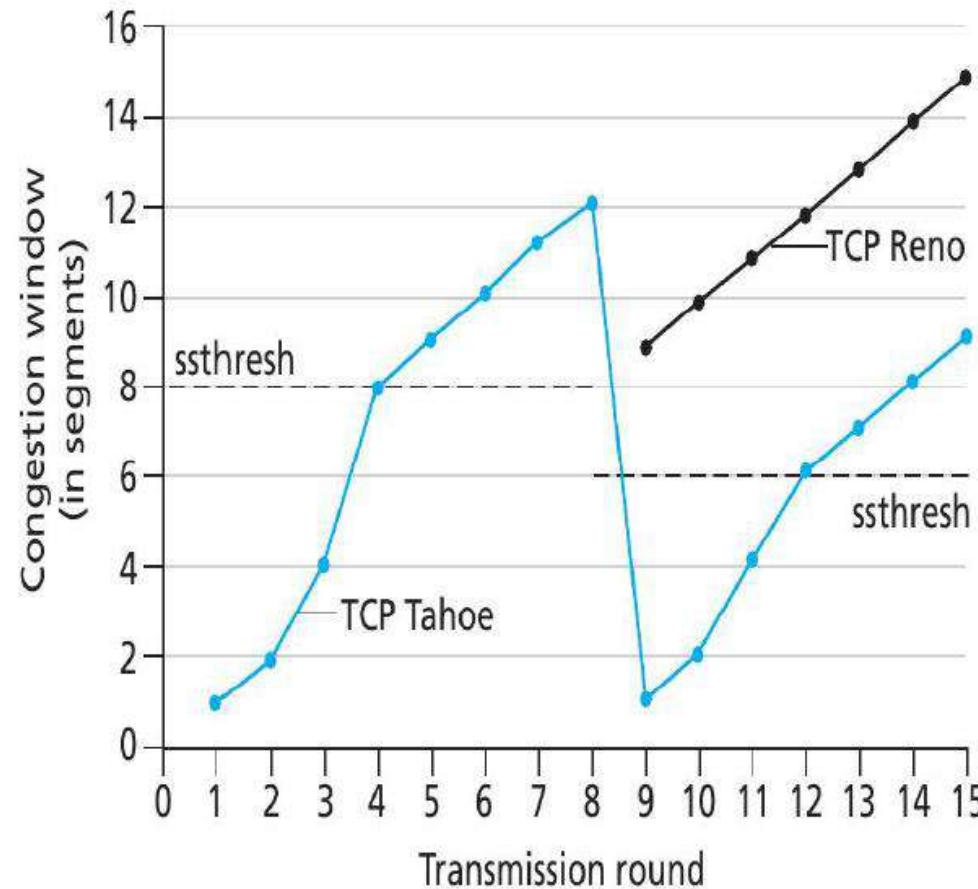
TCP Slow Start



TCP Congestion Control



TCP Congestion Window



AIMD

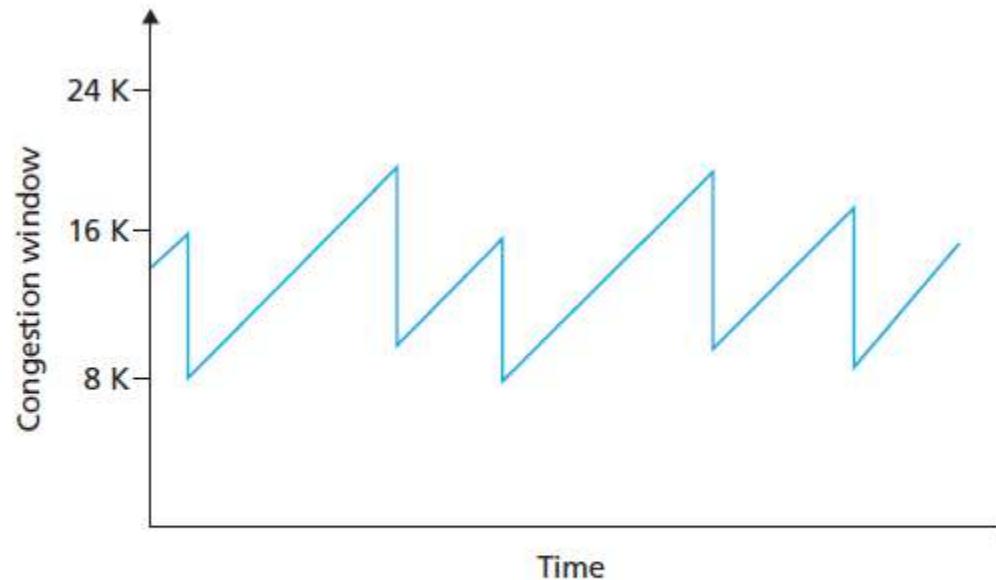


Figure 3.54 ♦ Additive-increase, multiplicative-decrease congestion control

additive-increase, multiplicative decrease (AIMD) form of congestion control:

TCP linearly increases its congestion window size (and hence its transmission rate) until a triple duplicate-ACK event occurs.

Network Layer

Dr. Raja Vara Prasad,
IIIT Sri City, Chittoor

Network Layer

Functionalities

- forwarding
- routing
- connection setup

Services

- guaranteed delivery
- guaranteed delivery with bounded delay
- in-order packet delivery
- guaranteed maximum jitter

Topics:

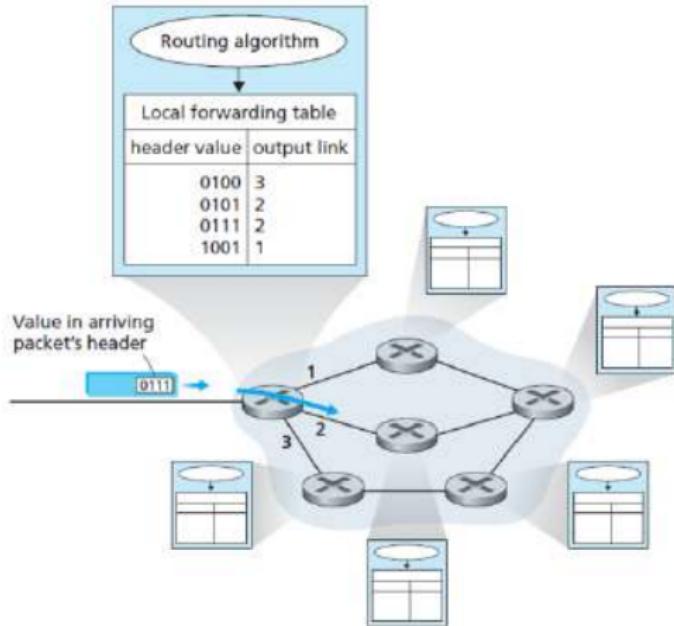
- Inside a router
- Packet forwarding
- Internet Protocol
- Addressing and IPV4
- Network Address Translation (NAT)
- Datagram Fragmentation
- Internet Control Message protocol and IPV6

Network Layer

Functionalities

Forwarding

- transfer of a packet from an incoming link to an outgoing link within a *single* router
- Every router has a **forwarding table**.
- forwards a packet by examining the value of a field in the arriving packet's header
- header value to index into the router's forwarding table.
- value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded.



Functionalities

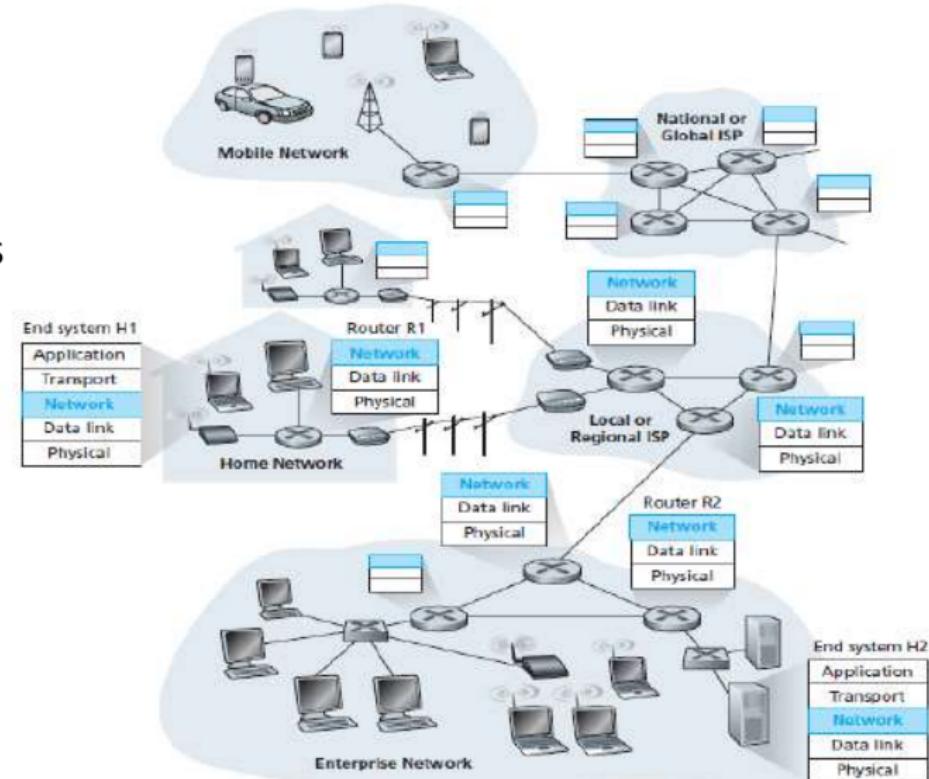
Routing

- involves *all* of a network's routers,
- collective interactions via routing protocols
- determine the paths that packets take on their trips from source to destination node
- must determine the route or path taken by packets ---from a sender to a receiver.
- algorithms to calculate these paths -- **routing algorithms**
- Routing algorithms : Centralized or Decentralized
- Human configuration without routing – response to changes in the network is slow
- Link layer or L2 switches: forwarding decision on values in the fields of the link layer
- Router or L3 switches: forwarding decision on the value in the networklayer
- Field.
- L3 require L2 services also

Network Layer

Routing:

- Primary role of the routers is to forward datagrams from input links to output links
- Routers do not run application- and transport-layer protocols



Network Layer

Connection setup

- Some network-layer architectures
- require the routers along the chosen path from source to destination to handshake with each other
- to set up state before network-layer data packets within a given source-to-destination connection can begin to flow

Network Services Models:

Transport layer at a sending host transmits a packet into the network

- can the transport layer rely on the network layer to deliver the packet to the destination?
- When multiple packets are sent, will they be delivered to the transport layer in the receiving host in the order in which they were sent?

Network Layer

Network Services Models:

- Will the amount of time between the sending of two sequential packet transmissions be the same as the amount of time between their reception?
- Will the network provide any feedback about congestion in the network?
- What is the abstract view (properties) of the channel connecting the transport layer in the sending and receiving hosts?

Network service model defines the characteristics of end-to-end transport of packets between sending and receiving end systems

- guaranteed delivery
- guaranteed delivery with bounded delay
- in-order packet delivery
- *Guaranteed minimal bandwidth*
- *Security services*

Network Layer

- Guaranteed delivery
 - guarantees that the packet
- guaranteed delivery with bounded delay
 - not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound
- in-order packet delivery
 - guarantees that packets arrive at the destination in the order that they were sent
- *Guaranteed minimal bandwidth*
 - sending host transmits bits at a rate below the specified bit rate, no packet is lost
 - each packet arrives within a pre-specified host-to-host delay
- *Guaranteed maximum jitter*
 - time between the transmission and reception of two successive packets at the sender and receiver
- *Security services*
 - source host could encrypt the payloads of all datagrams
 - destination host responsible for decrypting the payloads

Network Layer

Network Architecture	Service Model	Bandwidth Guarantee	No-Loss Guarantee	Ordering	Timing	Congestion Indication
Internet	Best Effort	None	None	Any order possible In order	Not maintained Maintained	None Congestion will not occur
ATM	CBR	Guaranteed constant rate	Yes	In order	Maintained	Congestion indication provided
ATM	ABR	Guaranteed minimum	None	In order	Not maintained	Congestion indication provided

CBR: real-time, constant bit rate audio and video traffic
virtual pipe for a dedicated fixed-bandwidth transmission

Available bit rate (ABR) ATM network service.

- a minimum transmission rate (MCR) is guaranteed
- If the network has enough free resources at a given time → higher rate than the MCR

Virtual-Circuit and Datagram Networks

Transport layer provides a choice between two services:
 UDP or TCP

- Network layer → connectionless service or connection service
- Connection begins with handshaking between the source and destination
- connectionless service → no handshaking preliminaries.
- connection service at the network layer → **virtual-circuit (VC) networks;**
- A connectionless service at the N/W layer → **datagram networks.**

Virtual-Circuit

virtual-circuit networks, use connections at the network layer.

--- connections known as **virtual circuits (VCs)**

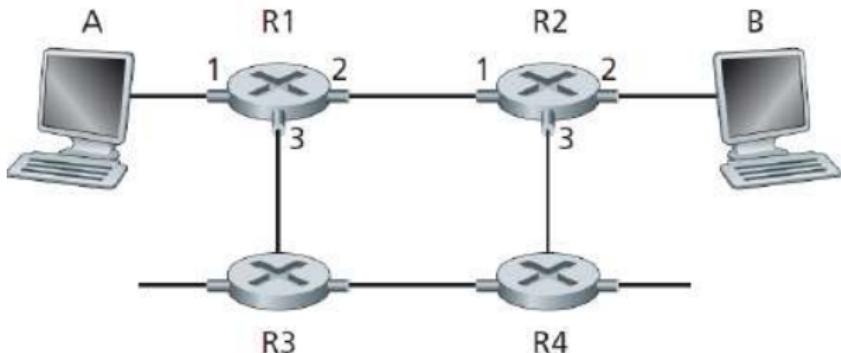
Ex: ATM and Frame Relay

A VC consists:

- (1) a path (series of links and routers) between the source and destination hosts
- (2) VC numbers, one number for each link along the path,
- (3) entries in the forwarding table in each router along the path.

- packet will carry a VC number in its header
- a virtual circuit may have a different VC number on each link,
- each intervening router must replace the VC number of each traversing packet with a new VC number.
- The new VC number is obtained from the forwarding table.

Virtual-Circuit



Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Virtual-Circuit

- new VC across a router → entry is added to the forwarding table.
- whenever a VC terminates → entries in each table along its path are removed.

why a packet doesn't just keep the same VC number on each of the links along its route ?

- replacing the number from link to link reduces the length of the VC field in the packet header
- with multiple VC numbers, each link in the path can choose a VC number independently of the VC numbers chosen at other links along the path.
- If a common VC number were required for all links along the path, the routers would have to exchange and process a substantial number of messages to agree on a common VC number

Virtual-Circuit

- In a VC network, the network's routers must maintain **connection state information** for the ongoing connections.

three phases in a virtual circuit:

1. **VC setup:**

transport layer contacts the network layer, specifies the receiver's address and waits for the network to set up the VC

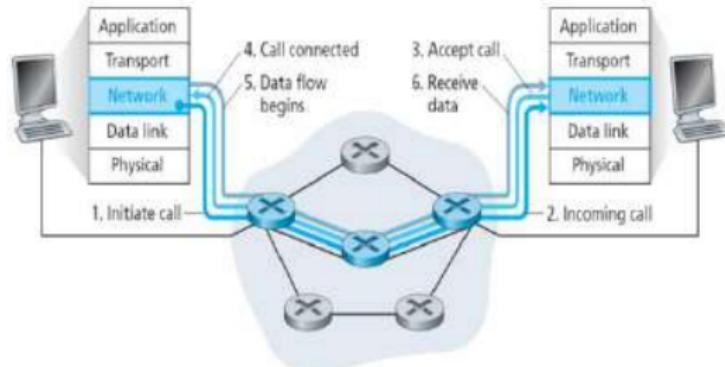
Network layer →

- determines the path,
- series of links and routers through which all packets of the VC will travel
- determines the VC number for each link along the path
- adds an entry in the forwarding table in each router along the path
- may reserve resources like bandwidth along the path of the VC

Virtual-Circuit

2. Data transfer:

3. VC teardown: initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC.

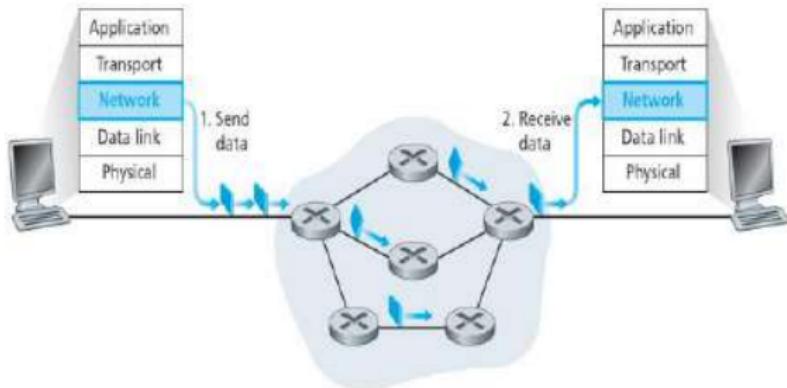


The network layer →

- inform the end system on the other side of the network of the call termination
- update the forwarding tables in each of the packet routers on the path to indicate that the VC no longer exists.
- **Signaling Messages:**
 - messages that the end systems send into the network to initiate or terminate a VC
 - messages passed between the routers to set up the VC
- **Signaling Protocols**

Datagram Network

- each time an end system wants to send a packet,
 - it stamps the packet with the address of the destination end system and then pops the packet into the network
 - passes through a series of routers
 - routers uses the packet's destination address to forward the packet
 - each router has a forwarding table that maps destination addresses to link interfaces
 - router uses the packet's destination address to look up the appropriate output link interface in the forwarding table
-
- all destination addresses are 32 bits
 - forwarding table would have one entry for every possible destination address
 - more than 4 billion possible addresses



Datagram Network

- Router matches a prefix
- if there's a match, the router forwards the packet to a link associated with the match.
- Multiple matches:
longest prefix matching rule

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

11001000 00010111 00010110 10100001

11001000 00010111 00011000 10101010

Datagram Network

Comparison of VC & Datagram: update intervals of forwarding tables

- time scale at which this forwarding state information changes is **relatively slow:** every 1-5 minutes
- In a VC network: forwarding table is modified: a new connection is set up or an existing connection is torn down. happen **at a microsecond** in a backbone router.

@ Datagram

- **forwarding tables in datagram networks can be modified at any time,**
- a series of packets sent from one end system to another may follow different paths through the network
- **may arrive out of order**

Datagram

Example of Datagram:

Internet is a datagram network

- more sophisticated end-system devices -- network-layer service model as simple as possible.
- in-order delivery, reliable data transfer, congestion control etc implemented at a higher layer, in the end systems.
- Internet network-layer service model makes minimal service guarantees, it imposes minimal requirements on the network layer
- makes it easier to interconnect networks that use very different link-layer technologies:
 - ex: satellite, Ethernet, fiber, or radio with different transmission rates and loss characteristics
- to add a new service simply by attaching a host to the network and defining a new application-layer protocol has allowed new Internet applications in a short period of time.

Inside a Router

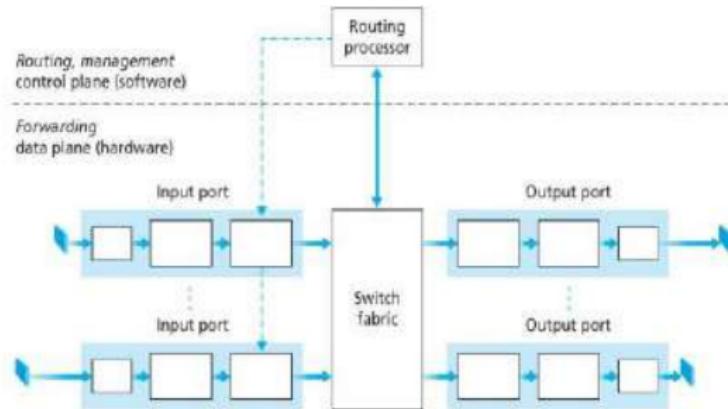


Figure 4.6 • Router architecture

- Input port
- Switching fabric
- Output port
- Routing processor

Inside a Router

Input port:

performs the physical layer function

performs link-layer functions

lookup function is also performed

Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor

Switching fabric.

connects the router's input ports to its output ports.

is completely contained within the router: a network inside of a network router!

Output ports:

stores packets received from the switching fabric

transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions

Routing processor.

- executes the routing protocols
- maintains routing tables and attached link state information,
- computes the forwarding table for the router.
- performs the network management functions

Inside a Router

Router forwarding plane:

router's input ports, output ports, and switching fabric together → forwarding function → always implemented in hardware: **router forwarding plane**

Ex:

- a 10 Gbps input link and a 64-byte IP datagram,
- the input port has only **51.2 ns** to process the datagram before another datagram may arrive.
- If N ports are combined on a line card (as is often done in practice), the datagram-processing **pipeline must operate N times faster**
- far too fast for software implementation

Router's control functions

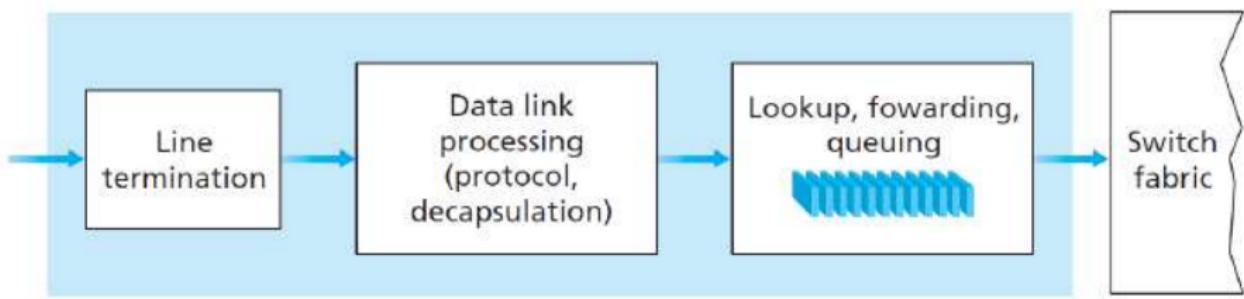
- executing the routing protocols, responding to attached links that go up or down, and performing management functions
- operate at the **millisecond** or second timescale
- **router control plane** functions are usually implemented in **software** and execute on the **routing processor**

Inside a Router

1. Input processing
2. Switching
3. Output processing

1. Input processing:

- input port's line termination function and link-layer processing
implement the physical and link layers for that individual input link
- lookup performed → Central to the router's operation → to look up the output port → Arriving packet will be forwarded via the switching fabric.



1. Input processing:

- a shadow copy typically stored at each input port
- forwarding table is copied from the routing processor to the line cards
- forwarding decisions → locally, at each input port --> without invoking the centralized routing processor on a per-packet basis
→ avoiding a centralized processing bottleneck.
- search through the forwarding table → for the longest prefix match
- at Gigabit transmission rates → lookup → nanoseconds
- techniques beyond a simple linear search through a large table
- Special attention: memory access times → embedded on-chip DRAM, faster SRAM, Ternary Content Address Memories (TCAMs) using the fabric

Inside a Router

1. Input processing:

- a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric
- will be queued at the input port and then scheduled to cross the fabric at a later point in time

Other important aspects of Input processing:

- (1) physical- and link-layer processing must occur, as discussed above;
- (2) the packet's version number, checksum and time-to-live field must be checked and the latter two fields rewritten;
- (3) counters used for network management (number of IP datagrams received) must be updated.

Inside a Router

Switching:
through this fabric that the
packets are actually switched
from an input port to an output
port.

Three types of switching:

- *Switching via memory*
- *Switching via a bus.*
- *Switching via an interconnection network*

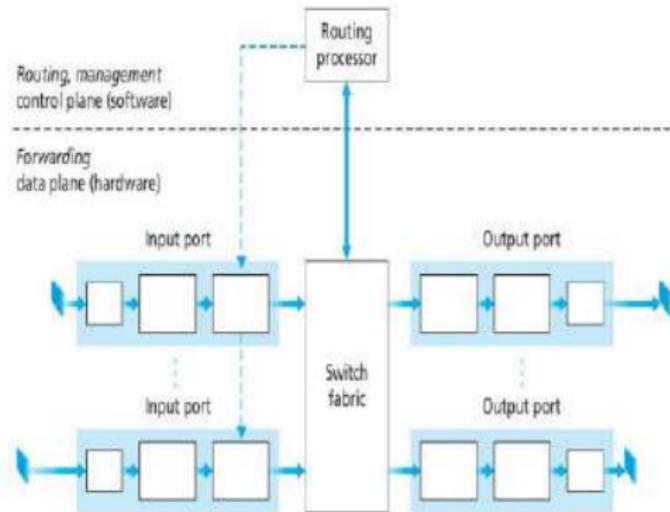


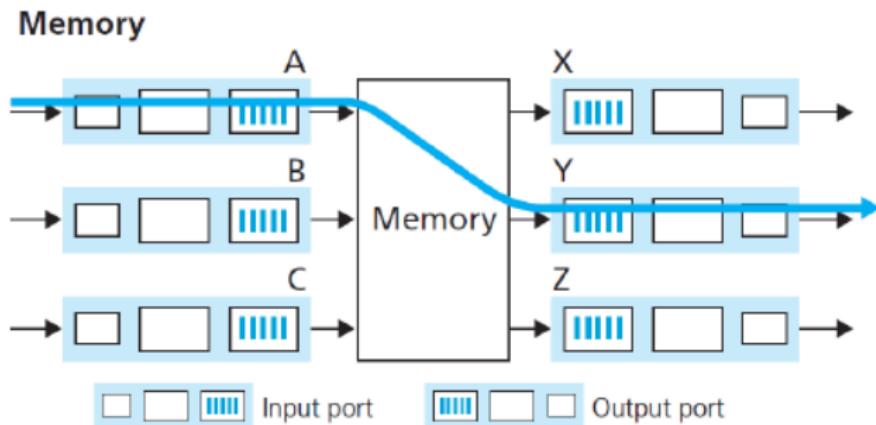
Figure 4.6 • Router architecture

Inside a Router: Switching

Switching via memory

- earliest routers – switching → under direct control of CPU
- *Input port signals the arrival of a packet → routing processor → Interrupt*
- *Processor completes lookup → copies packet → output buffer*
- *Present day routers → processing on a line card*
- shared-memory multiprocessors

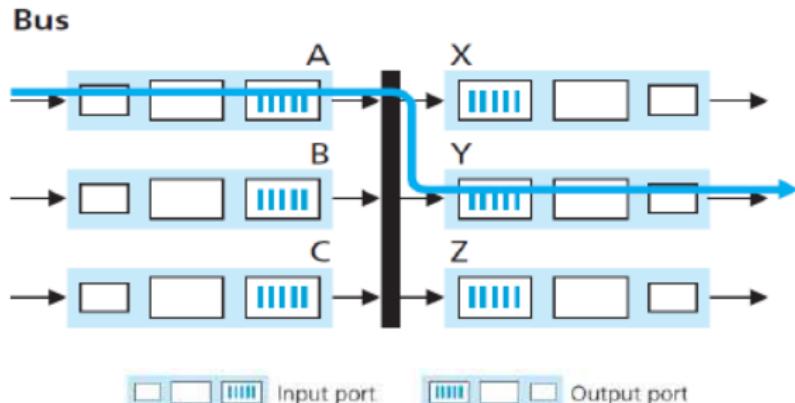
Ex: Cisco's Catalyst 8500 series switches



Inside a Router: Switching:

Switching via a bus:

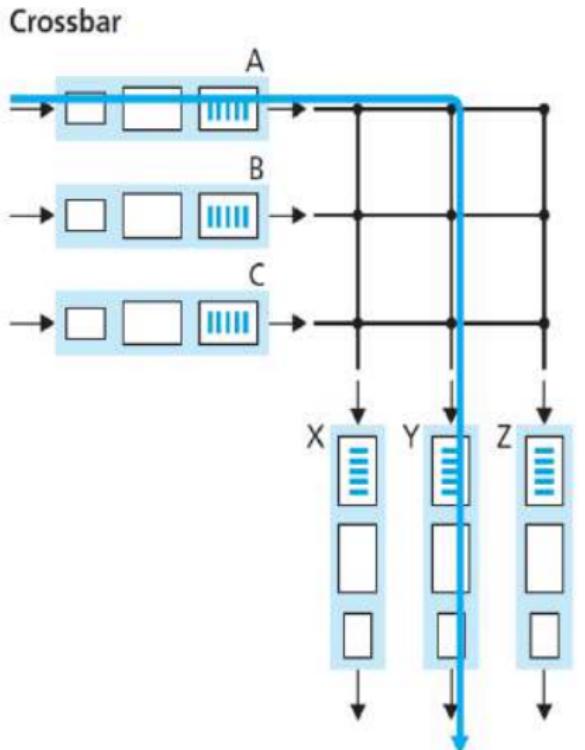
- an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
- input port pre-pend a switch-internal label → indicating the local output port → transmitting the packet onto the bus
- received by all output ports, but only the port that matches the label
- Even multiple packets at input ports → one packet on bus
- switching speed of the router is limited to the bus speed



Inside a Router: Switching

Switching via an interconnection network:

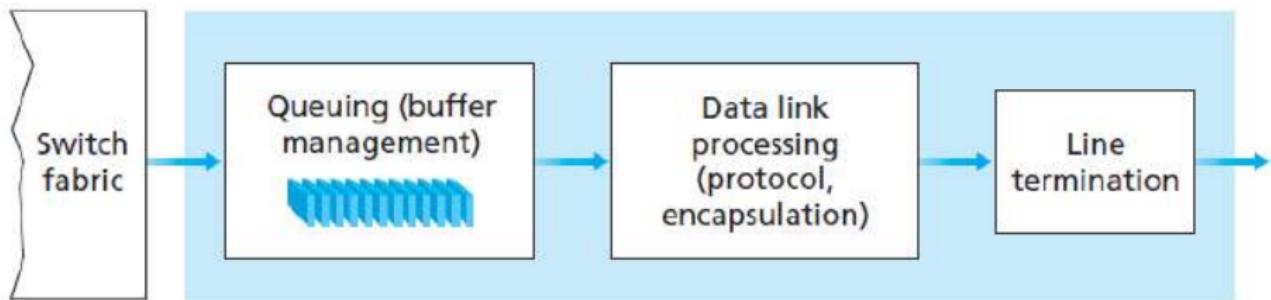
- more sophisticated interconnection network
- crossbar switch is an interconnection network consisting of $2N$ buses that connect N input ports to N output ports
- Each vertical bus intersects each horizontal bus at a crosspoint → can be opened or closed at any time by the switch fabric controller
- crossbar networks are capable of forwarding multiple packets in parallel
- if two packets from two different input ports → to the same output port → one will have to wait at the input



Inside a Router

Output processing:

- takes stored packets in the output port's memory → transmits them over the output link
- selecting and de-queueing packets for transmission
- performing the needed link layer and physical-layer transmission functions



Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

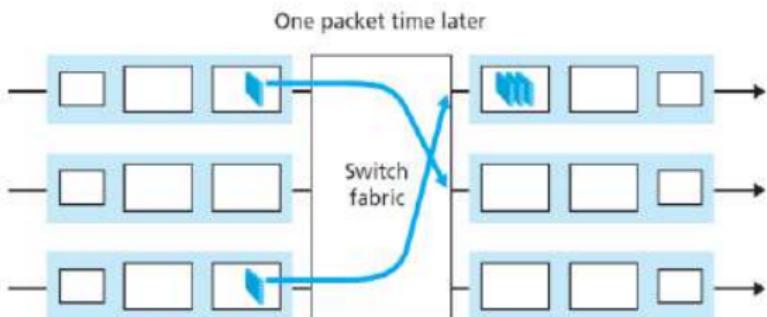
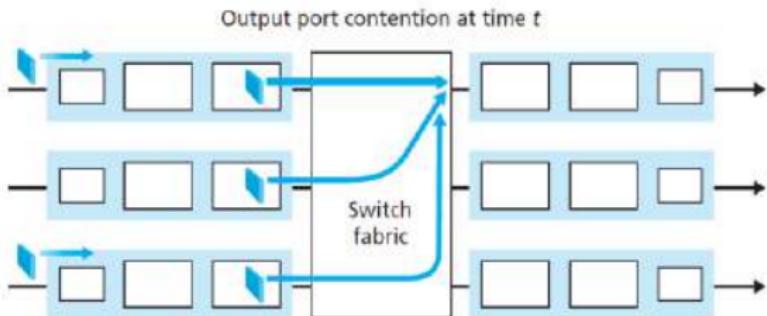
Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

Inside a Router: Output processing:

- **packet scheduler** at the output port must choose one packet among those queued for transmission
- first-come-first-served (FCFS)
- weighted fair queuing (WFQ) → shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission
- no enough memory to buffer an incoming packet either drop the arriving packet or remove one or more already-queued packets
- **Random Early Detection** - probabilistic marking/dropping functions



Inside a Router: Output processing:

what if the Switch fabric is not fast enough: → packet queuing at the input ports

Assume:

- (1) all link speeds are identical
- (2) one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link
- (3) packets are moved from a given input queue to their desired output queue in an FCFS manner

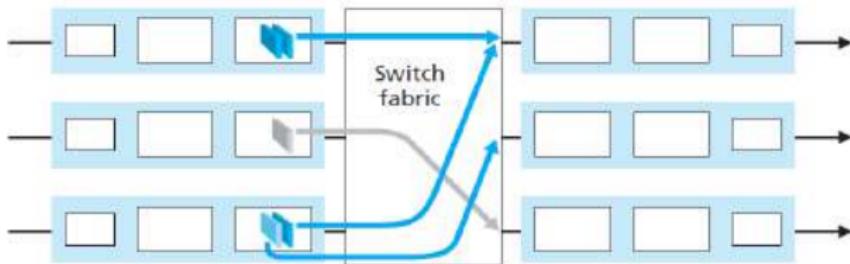
→ Multiple packets can be transferred in parallel, as long as their output ports are different

→ if two packets of two input queues are destined for the same output queue one of the packets will be blocked and must wait at the input queue.

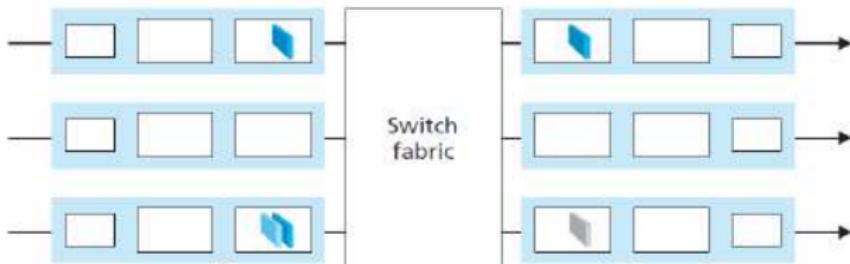
→ **head-of-the-line (HOL) blocking** → queue will grow to unbounded length

Inside a Router: Output processing:

Output port contention at time t—
one dark packet can be transferred



Light blue packet experiences HOL blocking



Key:

destined for upper output port

destined for middle output port

destined for lower output port

Inside a Router: Routing plane

- fully resides and executes in a routing processor within the router
 - network-wide routing control plane → decentralized
- with different pieces executing at different routers and interacting by sending control messages to each other

New router control plane architectures

- part of the control plane is implemented in the routers along with the data plane
- part of the control plane can be implemented externally to the router
- A well-defined API dictates how these two parts interact and communicate with each other

Software Defined Networking (SDN)

- separating the software control plane from the hardware data plane
- allowing different customized control planes to operate over fast hardware data planes

Network Layer

Dr. Raja Vara Prasad,
IIIT Sri City, Chittoor

Inside a Router

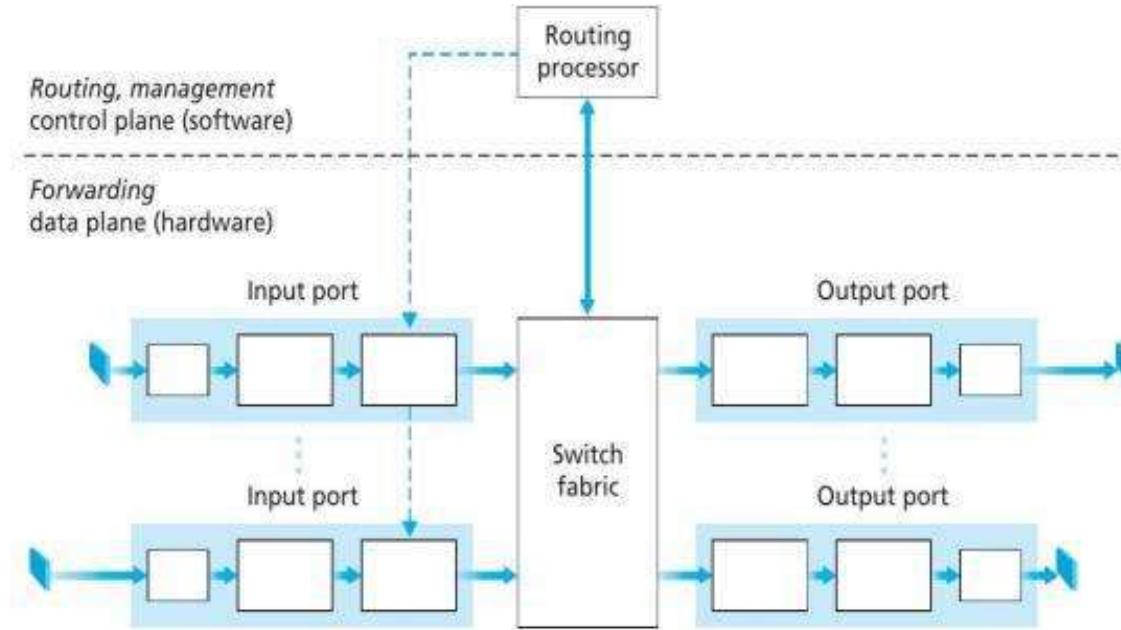


Figure 4.6 • Router architecture

- Input port
- Switching fabric
- Output port
- Routing processor

Inside a Router

Input port:

performs the physical layer function

performs link-layer functions

lookup function is also performed

Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor

Switching fabric.

connects the router's input ports to its output ports.

is completely contained within the router: a network inside of a network router!

Output ports:

stores packets received from the switching fabric

transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions

Routing processor.

- executes the routing protocols
- maintains routing tables and attached link state information,
- computes the forwarding table for the router.
- performs the network management functions

Inside a Router

Router forwarding plane:

router's input ports, output ports, and switching fabric together → forwarding function → always implemented in hardware: **router forwarding plane**

Ex:

- a 10 Gbps input link and a 64-byte IP datagram,
- the input port has only **51.2 ns** to process the datagram before another datagram may arrive.
- If N ports are combined on a line card (as is often done in practice), the datagram-processing **pipeline must operate N times faster**
- far too fast for software implementation

Router's control functions

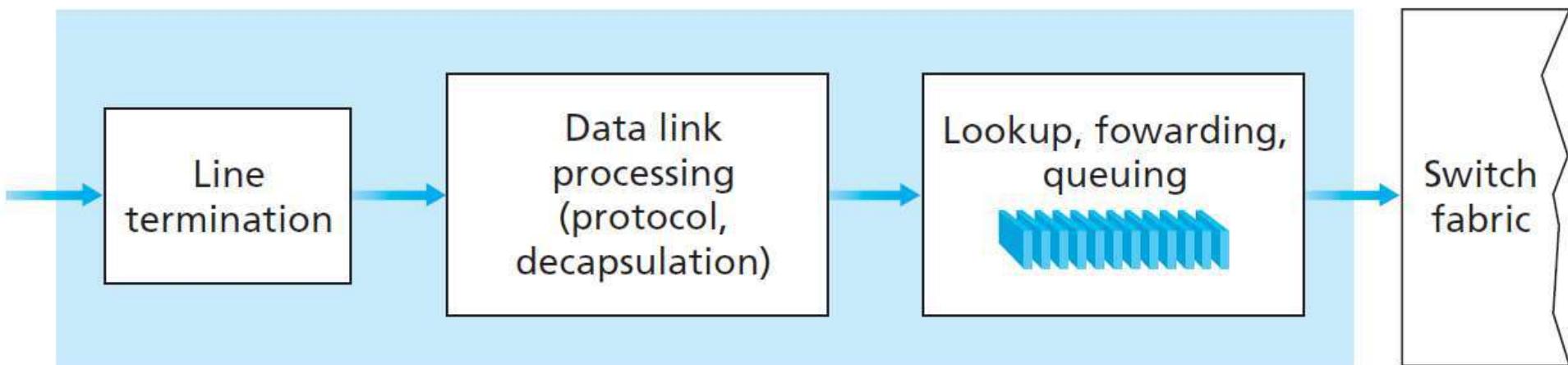
- executing the routing protocols, responding to attached links that go up or down, and performing management functions
- operate at the **millisecond** or second timescale
- **router control plane** functions are usually implemented in **software** and execute on the **routing processor**

Inside a Router

1. Input processing
2. Switching
3. Output processing

1. Input processing:

- input port's line termination function and link-layer processing implement the physical and link layers for that individual input link
- lookup performed → Central to the router's operation → to look up the output port → Arriving packet will be forwarded via the switching fabric.



1. Input processing:

- a shadow copy typically stored at each input port
- forwarding table is copied from the routing processor to the line cards
- forwarding decisions → locally, at each input port --> without invoking the centralized routing processor on a per-packet basis
→ avoiding a centralized processing bottleneck.
- search through the forwarding table → for the longest prefix match
- at Gigabit transmission rates → lookup → nanoseconds
- techniques beyond a simple linear search through a large table
- Special attention: memory access times → embedded on-chip DRAM, faster SRAM, Ternary Content Address Memories (TCAMs) using the fabric

1. Input processing:

- a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric
- will be queued at the input port and then scheduled to cross the fabric at a later point in time

Other important aspects of Input processing:

- (1) physical- and link-layer processing must occur, as discussed above;
- (2) the packet's version number, checksum and time-to-live field must be checked and the latter two fields rewritten;
- (3) counters used for network management (number of IP datagrams received) must be updated.

Inside a Router

Switching:
through this fabric that the
packets are actually switched
from an input port to an output
port.

Three types of switching:

- *Switching via memory*
- *Switching via a bus.*
- *Switching via an interconnection network*

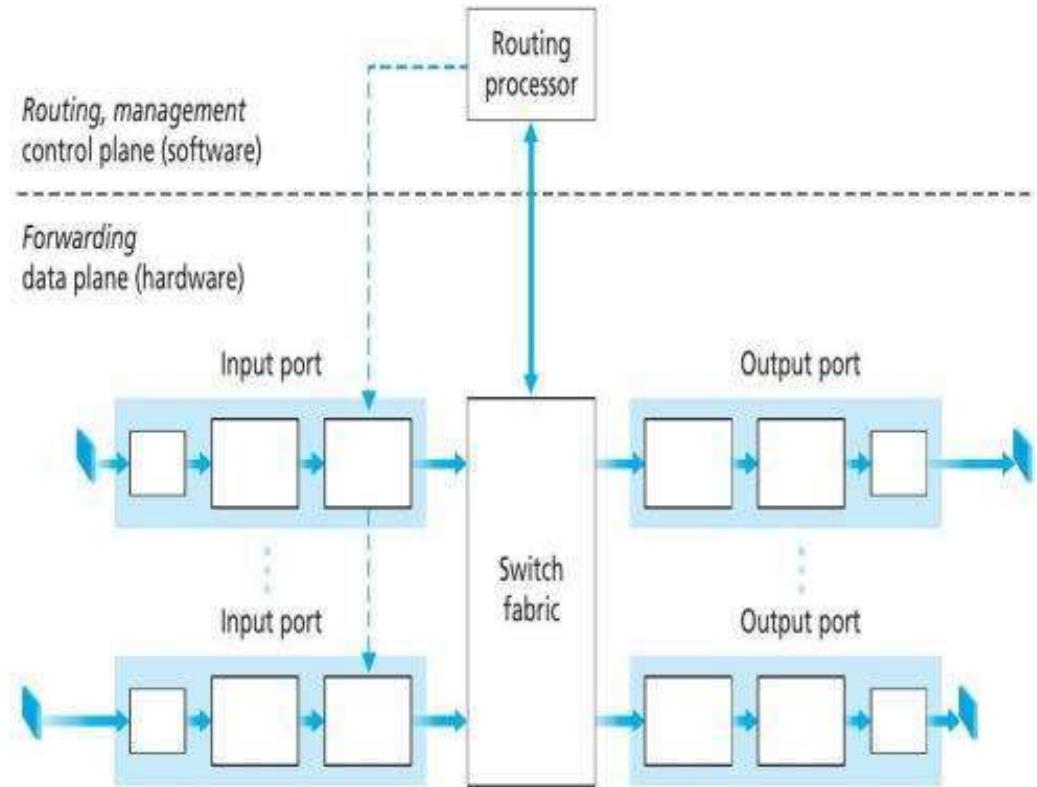


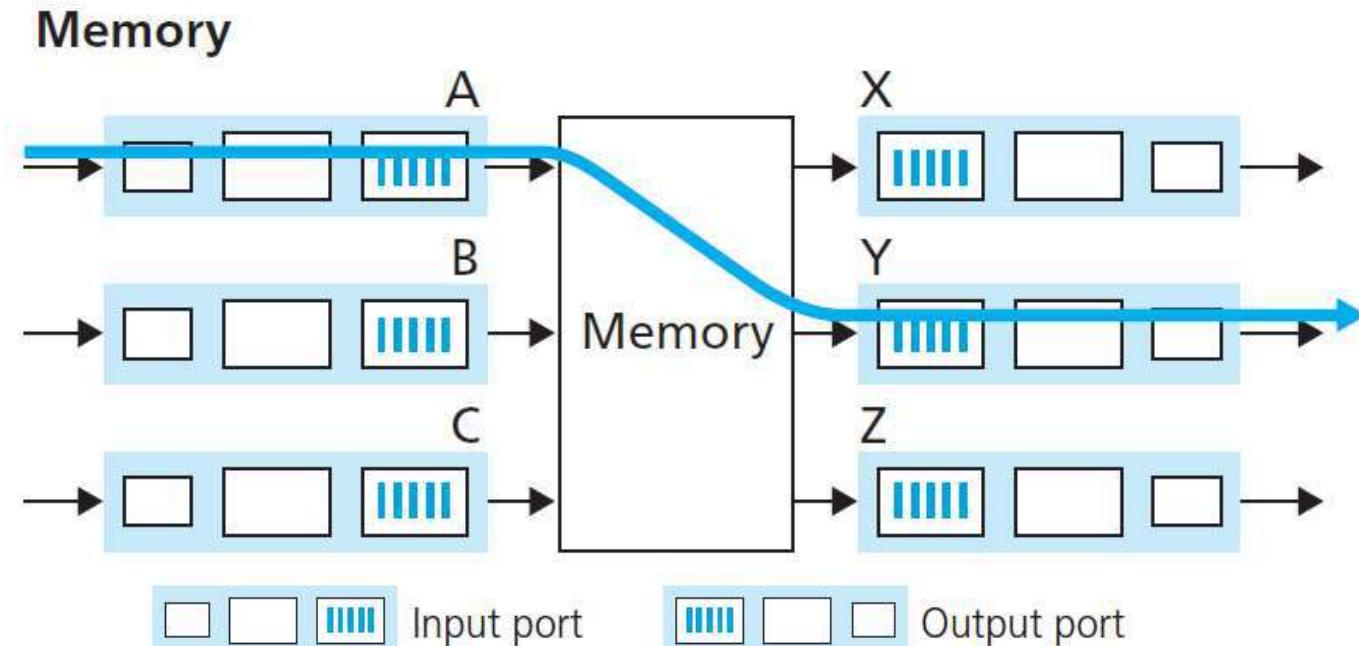
Figure 4.6 ♦ Router architecture

Inside a Router: Switching

Switching via memory

- earliest routers – switching → under direct control of CPU
- *Input port signals the arrival of a packet → routing processor → Interrupt*
- *Processor completes lookup → copies packet → output buffer*
- *Present day routers → processing on a line card*
- shared-memory multiprocessors

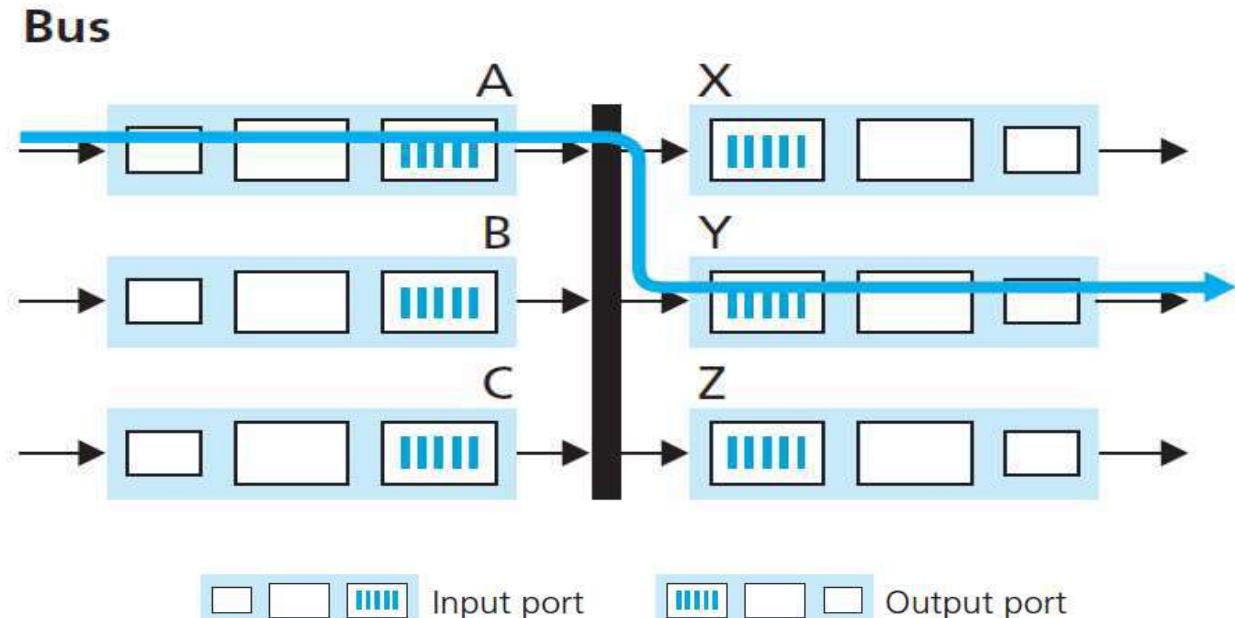
Ex: Cisco's Catalyst 8500 series switches



Inside a Router: Switching:

Switching via a bus:

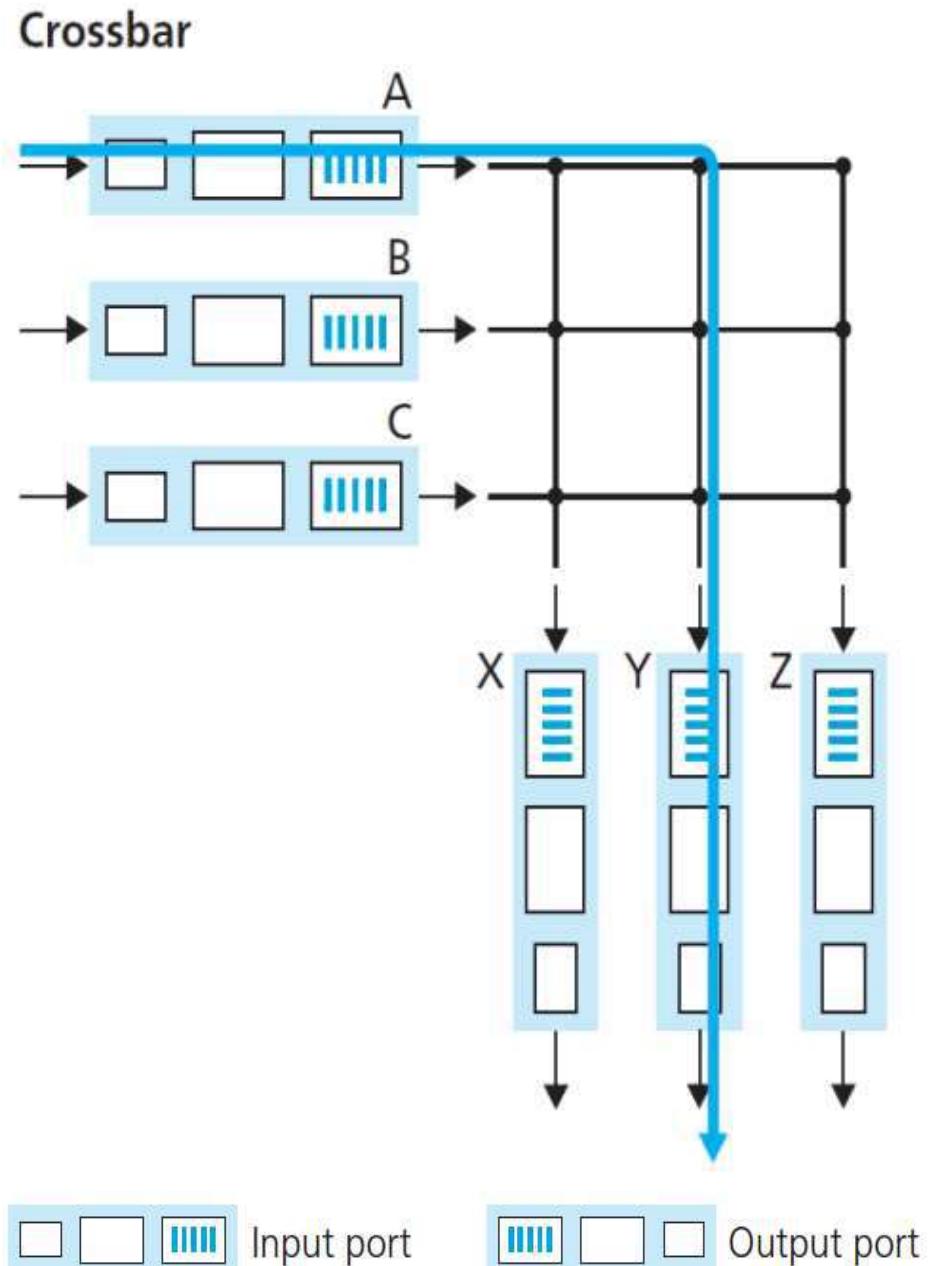
- an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
- input port pre-pend a switch-internal label → indicating the local output port → transmitting the packet onto the bus
- received by all output ports, but only the port that matches the label
- Even multiple packets at input ports → one packet on bus
- switching speed of the router is limited to the bus speed



Inside a Router: Switching

Switching via an interconnection network:

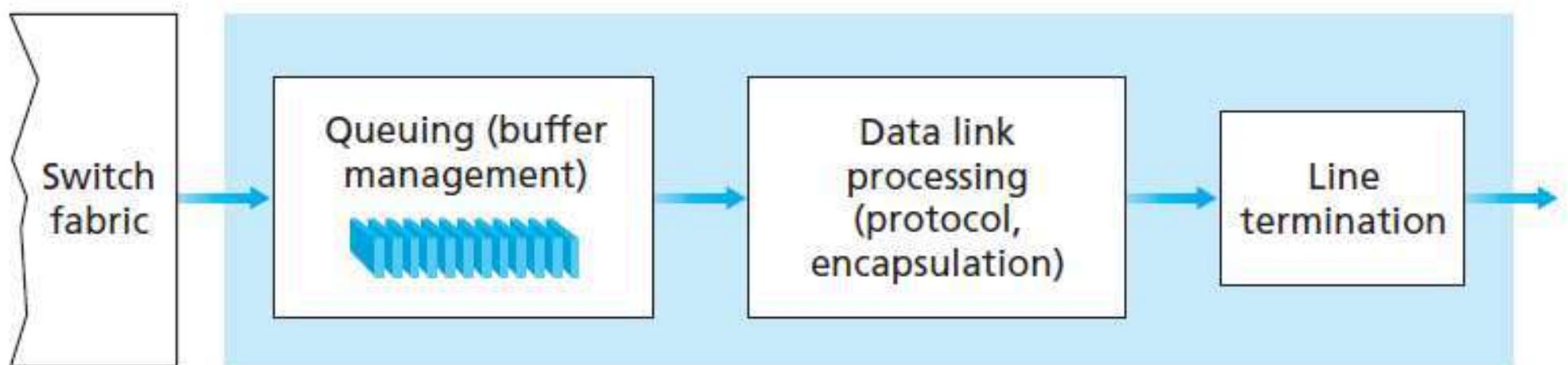
- more sophisticated interconnection network
- crossbar switch is an interconnection network consisting of $2N$ buses that connect N input ports to N output ports
- Each vertical bus intersects each horizontal bus at a crosspoint → can be opened or closed at any time by the switch fabric controller
- crossbar networks are capable of forwarding multiple packets in parallel
- if two packets from two different input ports → to the same output port → one will have to wait at the input



Inside a Router

Output processing:

- takes stored packets in the output port's memory → transmits them over the output link
- selecting and de-queueing packets for transmission
- performing the needed link layer and physical-layer transmission functions



Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

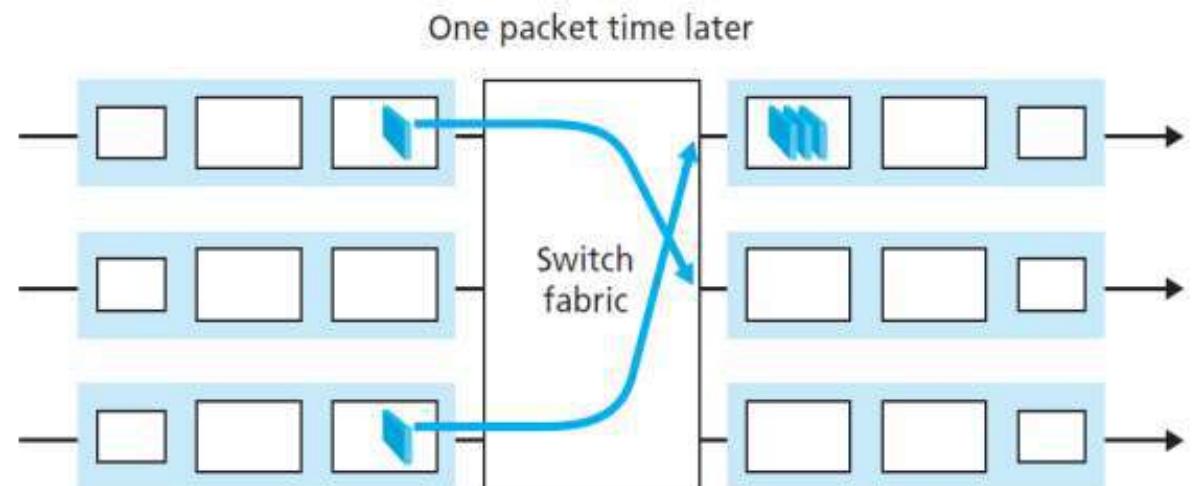
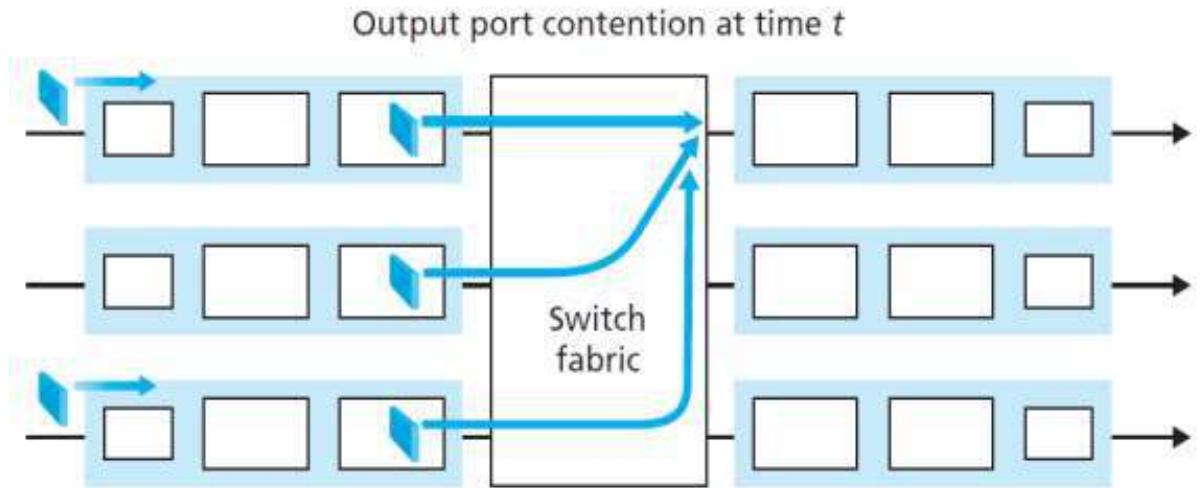
Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

Inside a Router: Output processing:

- **packet scheduler** at the output port must choose one packet among those queued for transmission
- first-come-first-served (FCFS)
- weighted fair queuing (WFQ) → shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission
- no enough memory to buffer an incoming packet either drop the arriving packet or remove one or more already-queued packets
- **Random Early Detection** - probabilistic marking/dropping functions



Inside a Router: Output processing:

what if the Switch fabric is not fast enough: → packet queuing at the input ports

Assume:

- (1) all link speeds are identical
- (2) one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link
- (3) packets are moved from a given input queue to their desired output queue in an FCFS manner

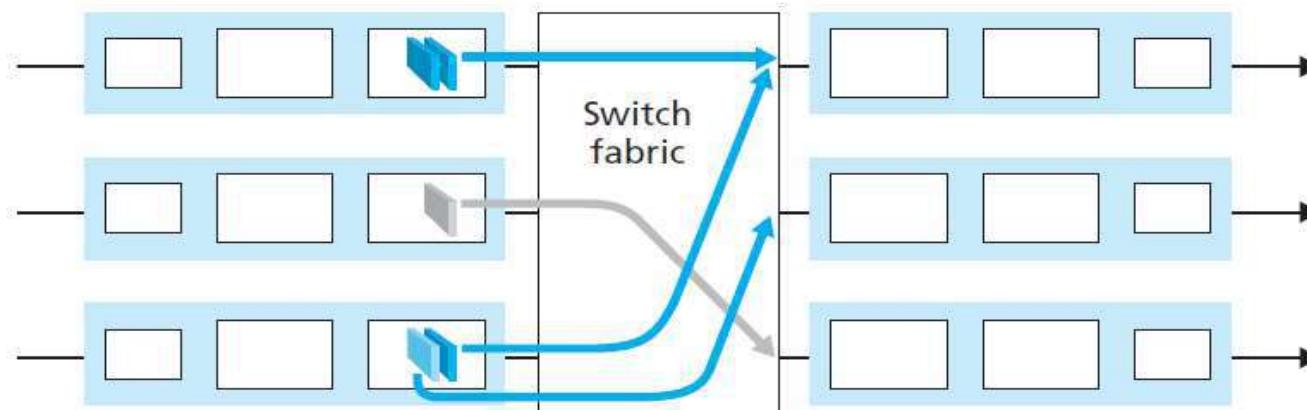
→ Multiple packets can be transferred in parallel, as long as their output ports are different

→ if two packets of two input queues are destined for the same output queue one of the packets will be blocked and must wait at the input queue.

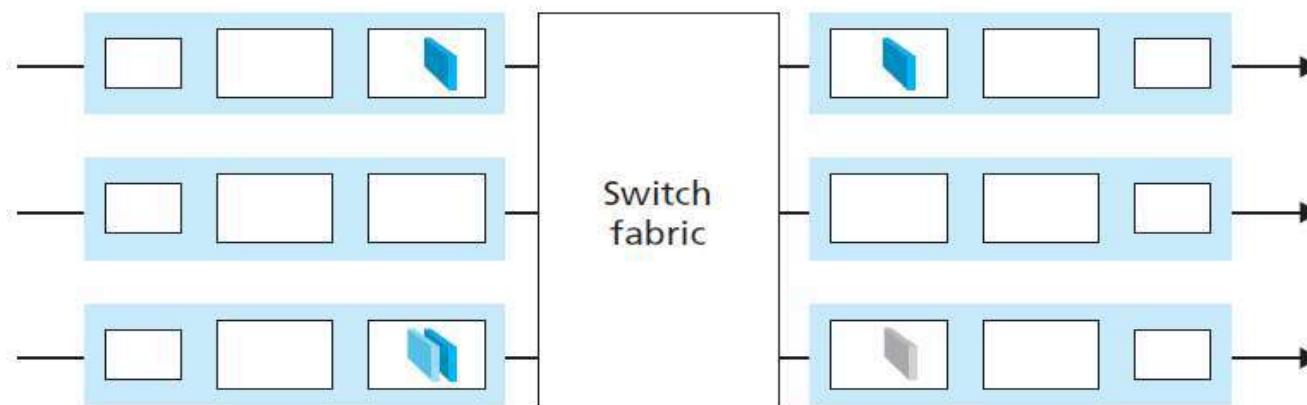
→ **head-of-the-line (HOL) blocking** → queue will grow to unbounded length

Inside a Router: Output processing:

Output port contention at time t —
one dark packet can be transferred



Light blue packet experiences HOL blocking



Key:

destined for upper output port

destined for middle output port

destined for lower output port

Inside a Router: Routing plane

- fully resides and executes in a routing processor within the router
- network-wide routing control plane → decentralized
→ with different pieces executing at different routers and interacting by sending control messages to each other

New router control plane architectures

- part of the control plane is implemented in the routers along with the data plane
- part of the control plane can be implemented externally to the router
- A well-defined API dictates how these two parts interact and communicate with each other

Software Defined Networking (SDN)

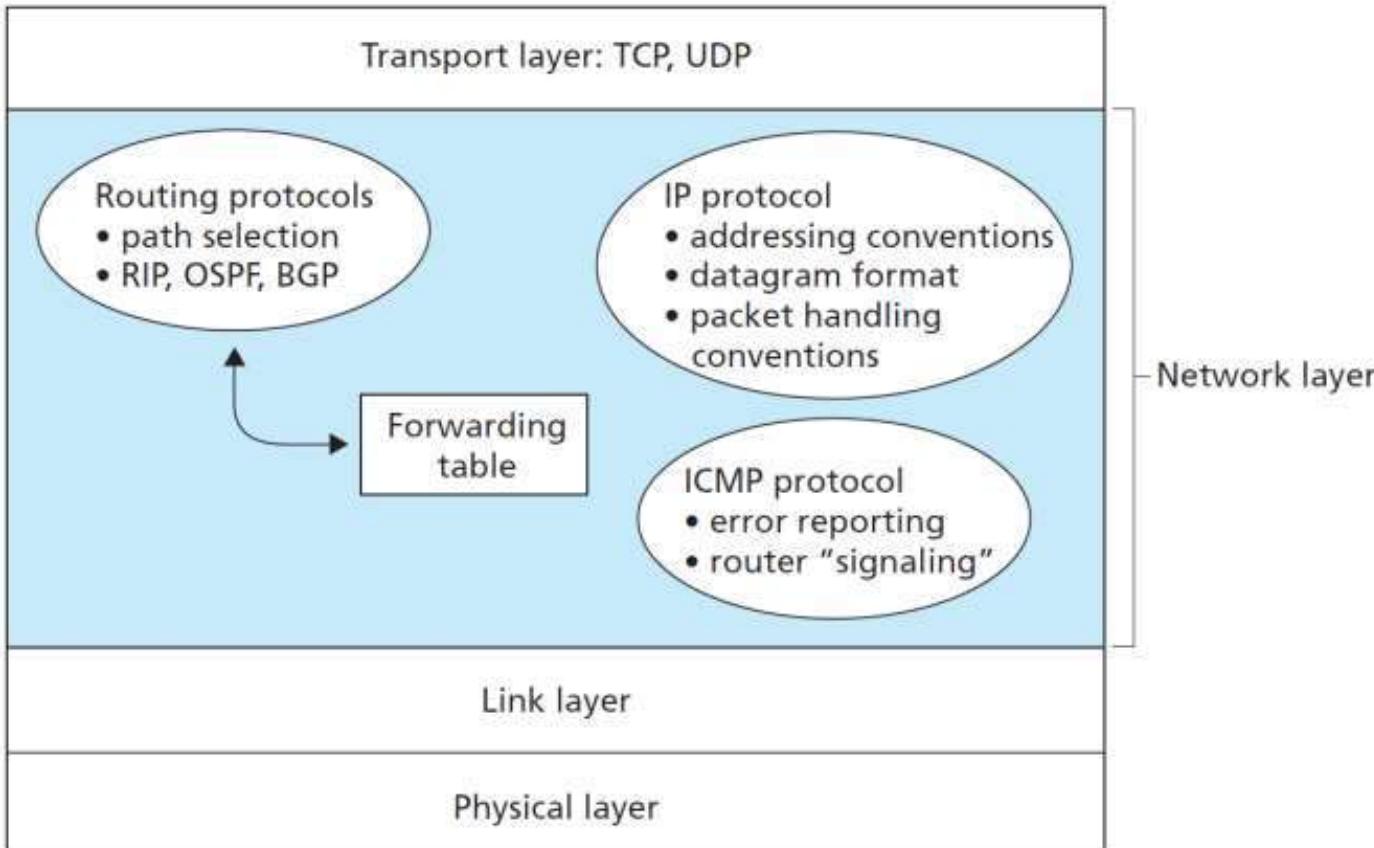
- separating the software control plane from the hardware data plane
- allowing different customized control planes to operate over fast hardware data planes

Internet Protocol

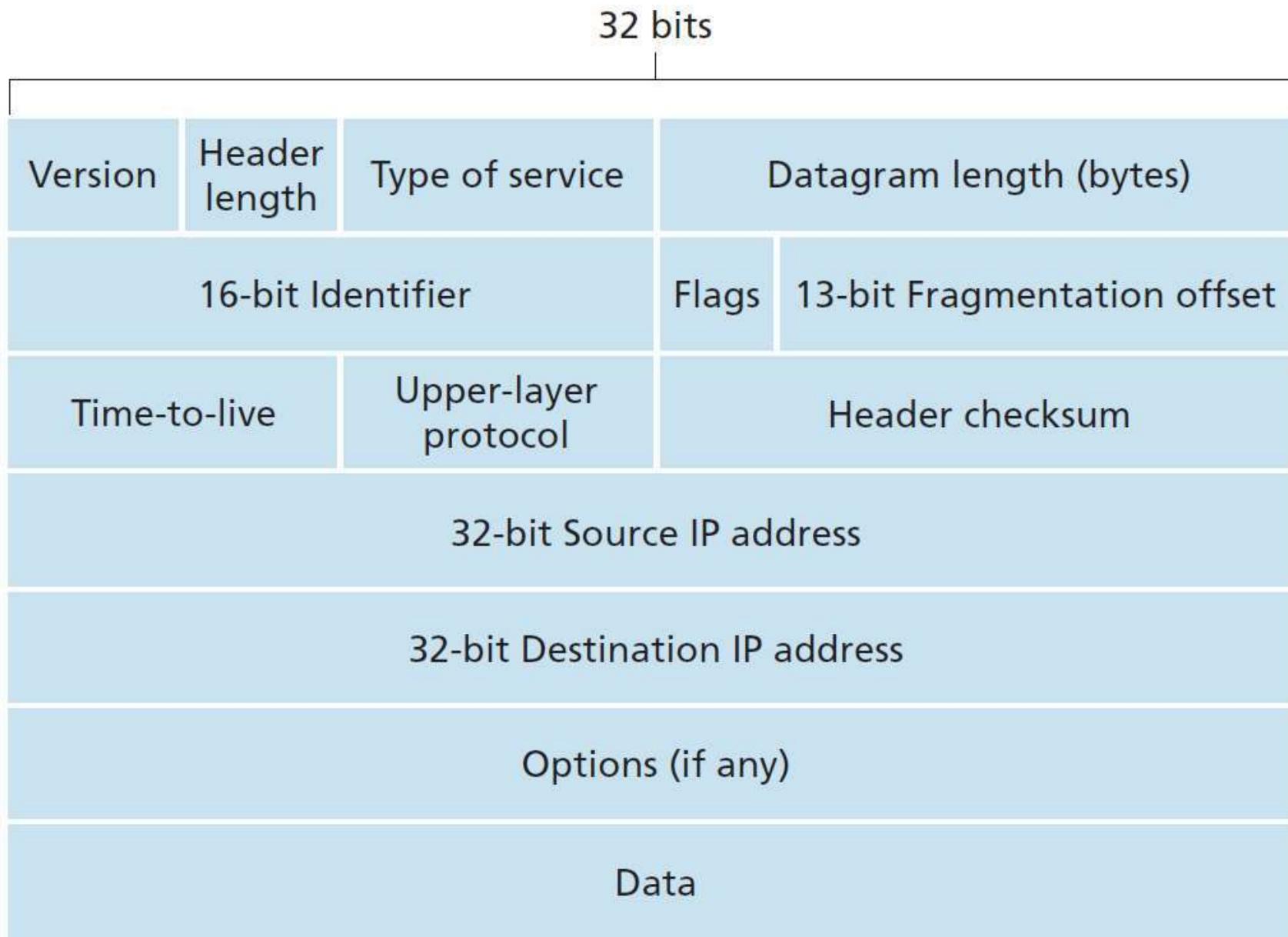
Important modules:

1. IP Protocol
2. Routing component
3. Internet Control Message Protocol

IPV4 and IPV6



Internet Protocol



Internet Protocol

Version number: 4 bits specify the IP protocol version
router can determine how to interpret the remainder of the IP datagram

Header length: IPv4 datagram can contain a variable number of options
→ 4 bits are needed → where in the IP datagram the data actually begins
→ Most of IP datagrams do not contain options → 20-byte header

Type of service: to allow different types of IP datagrams
→ differentiating datagrams requiring low delay, high throughput, or reliability
→ distinguish real-time and non-real time datagrams

Datagram length: total length of the IP datagram → header + data
→ 16 bits long → theoretical maximum size: 65,535 bytes → rarely > than 1500 bytes

Identifier, flags, fragmentation offset: fields required for IP fragmentation

Time to Live:
→ to ensure that datagrams do not circulate forever: long-lived routing loop
→ field is decremented by one each time

Internet Protocol

Protocol:

field is used when an IP datagram reaches its final destination.
value indicates → specific transport-layer protocol to which the data portion of this IP datagram should be passed.

Ex:value-6 indicates → data portion is passed to TCP, 17 indicates to UDP.

Header checksum:

- aids a router in detecting bit errors in a received IP datagram.
- computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic.
- stored in the checksum field and compares with router computed checksum
- detects an error condition → Routers typically discard datagrams
- checksum must be recomputed and stored again at each router → TTL field, and possibly the options field as well, may change.

TCP already has checksum, why do datagrams need checksum?

Source and destination IP addresses:

it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field

Internet Protocol

Options:

- allow an IP header to be extended.
- existence of options does complicate
- datagram headers can be of variable length,
- cannot determine a priori where the data field will start
- amount of time needed to process an IP datagram at a router can vary greatly
- IP options were dropped in the IPv6 header

Data (payload):

- IP datagram contains the transport-layer segment to be delivered to the destination.
 - can carry other types of data → ICMP messages
- ***datagram carrying a TCP segment → each nonfragmented datagram carries a total of 40 bytes of header → 20 bytes of IP header plus 20 bytes of TCP header along with the application-layer message.

Internet Protocol

IP Datagram Fragmentation:

- Not all link-layer protocols can carry network-layer packets of the same size.
 - Some protocols can carry big datagrams → other protocols carry only little packets.
 - Ex: Ethernet frames up to 1,500 bytes
 - wide-area links no more than 576bytes.
 - **maximum amount of data that a link-layer frame can carry
 - maximum transmission unit (MTU).
 - IP datagram is encapsulated within the link-layer frame for transport from one router to the next router
 - MTU link-layer protocol places a hard limit on the length of an IP datagram.
 - problem: each of the links along the route between sender and destination can use different link-layer protocols
 - each of these protocols can have different MTUs.
 - Router with interconnects several links with different link layer MTU's may receive a datagram and outgoing link MTU my be smaller
 - ***squeeze this oversized IP datagram into the payload field of the link-layer frame***
- Solution:** Fragment the IP datagram into two or more smaller datagrams

Internet Protocol - Fragmentation

Fragment:

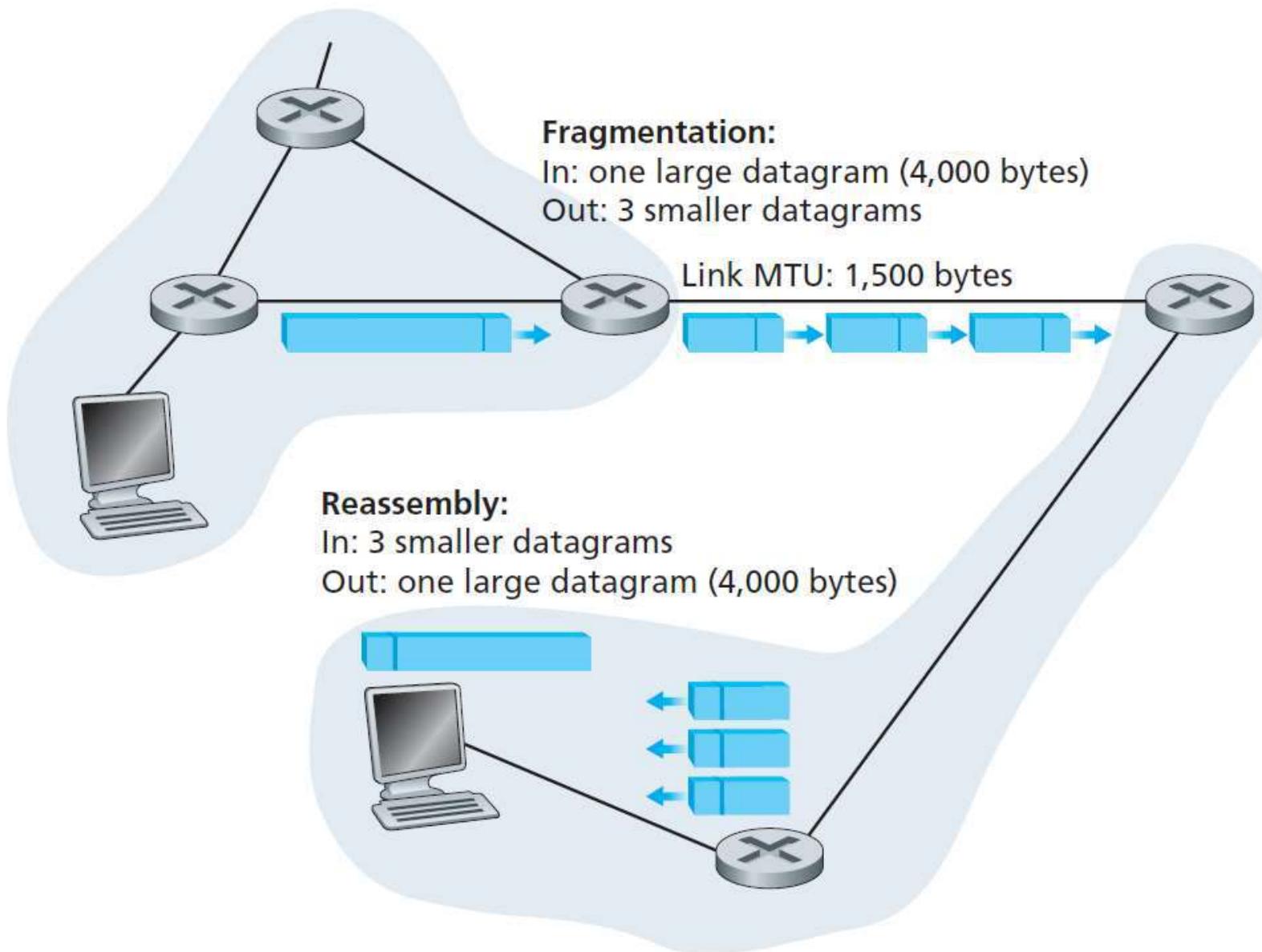
- IP datagram into two or more smaller IP datagrams
 - encapsulate each smaller IP datagrams in a separate link-layerframe;
 - send these frames over the outgoing link. Each smaller datagrams is a **fragment**.
-
- Fragments need to be reassembled before they reach destination.
 - TCP and UDP: expecting to receive complete, unfragmented segments from nwk layer
 - reassembling datagrams in the routers → significant complication damp router performance.
 - datagram reassembly in the end systems rather than in network routers.

Internet Protocol - Fragmentation

Identification flag, and fragmentation offset fields in the IP datagram header:

- destination host to determine whether it has received the last fragment
- how the fragments it has received should be pieced back together to form the original datagram
- sending host stamps the datagram **with an identification number** When a router needs to fragment a datagram
- each resulting datagram is stamped with the **source address, destination address, and identification number** of the original datagram.
- destination receives a series of datagrams
- can examine the identification numbers of the datagrams
- determine which of the datagrams are actually fragments of the same larger datagram
- to make sure, if Destination host has received last fragment of the original datagram,
 - **the last fragment has a flag bit set to 0
 - **other fragments have this flag bit set to 1
- a fragment is missing or reassemble the fragments in their proper order
 - *** the offset field is used to specify where the fragment fits within the original IP datagram.

Internet Protocol - Fragmentation



Internet Protocol - Fragmentation

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that $185 \cdot 8 = 1,480$)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= $3,980 - 1,480 - 1,480$) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that $370 \cdot 8 = 2,960$)	flag = 0 (meaning this is the last fragment)

** payload of the datagram is passed to the transport layer only after the IP layer has fully reconstructed the original IP datagram.

** If one or more of the fragments does not arrive at the destination, the incomplete datagram is discarded and not passed to the transport layer

** TCP will recover from this loss by having the source retransmit the data in the original datagram

Internet Protocol - Fragmentation

Challenges:

complicates routers and end systems → to accommodate datagram fragmentation and reassembly

→ fragmentation can be used to create lethal DoS attacks, attacker sends a series of bizarre and unexpected fragments.

Ex: Jolt2 attack → attacker sends a stream of small fragments to the target host, none of which has an offset of zero. The target can collapse as it attempts to rebuild datagrams out of the degenerate packets.

→ Another class of exploits:

sends overlapping IP fragments, fragments whose offset values are set so that the fragments do not align properly.

IPv6, does away with fragmentation altogether, thereby streamlining IP packet processing and making IP less vulnerable to attack.

Network Layer

Dr. Raja Vara Prasad,
IIIT Sri City, Chittoor

Inside a Router

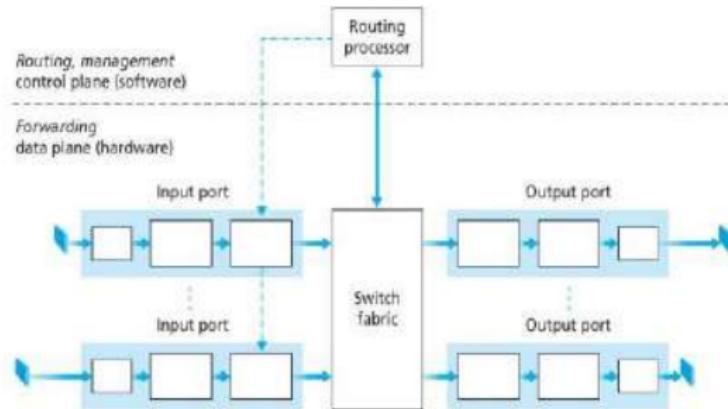


Figure 4.6 • Router architecture

- Input port
- Switching fabric
- Output port
- Routing processor

Inside a Router

Input port:

performs the physical layer function

performs link-layer functions

lookup function is also performed

Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor

Switching fabric.

connects the router's input ports to its output ports.

is completely contained within the router: a network inside of a network router!

Output ports:

stores packets received from the switching fabric

transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions

Routing processor.

- executes the routing protocols
- maintains routing tables and attached link state information,
- computes the forwarding table for the router.
- performs the network management functions

Inside a Router

Router forwarding plane:

router's input ports, output ports, and switching fabric together → forwarding function → always implemented in hardware: **router forwarding plane**

Ex:

- a 10 Gbps input link and a 64-byte IP datagram,
- the input port has only **51.2 ns** to process the datagram before another datagram may arrive.
- If N ports are combined on a line card (as is often done in practice), the datagram-processing **pipeline must operate N times faster**
- far too fast for software implementation

Router's control functions

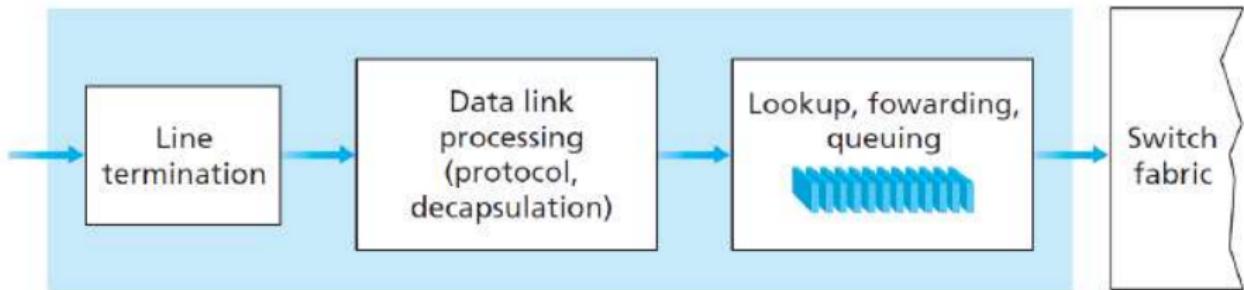
- executing the routing protocols, responding to attached links that go up or down, and performing management functions
- operate at the **millisecond** or second timescale
- **router control plane** functions are usually implemented in **software** and execute on the **routing processor**

Inside a Router

1. Input processing
2. Switching
3. Output processing

1. Input processing:

- input port's line termination function and link-layer processing implement the physical and link layers for that individual input link
- lookup performed → Central to the router's operation → to look up the output port → Arriving packet will be forwarded via the switching fabric.



1. Input processing:

- a shadow copy typically stored at each input port
- forwarding table is copied from the routing processor to the line cards
- forwarding decisions → locally, at each input port --> without invoking the centralized routing processor on a per-packet basis
→ avoiding a centralized processing bottleneck.
- search through the forwarding table → for the longest prefix match
- at Gigabit transmission rates → lookup → nanoseconds
- techniques beyond a simple linear search through a large table
- Special attention: memory access times → embedded on-chip DRAM, faster SRAM, Ternary Content Address Memories (TCAMs) using the fabric

Inside a Router

1. Input processing:

- a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric
- will be queued at the input port and then scheduled to cross the fabric at a later point in time

Other important aspects of Input processing:

- (1) physical- and link-layer processing must occur, as discussed above;
- (2) the packet's version number, checksum and time-to-live field must be checked and the latter two fields rewritten;
- (3) counters used for network management (number of IP datagrams received) must be updated.

Inside a Router

Switching:
through this fabric that the
packets are actually switched
from an input port to an output
port.

Three types of switching:

- *Switching via memory*
- *Switching via a bus.*
- *Switching via an interconnection network*

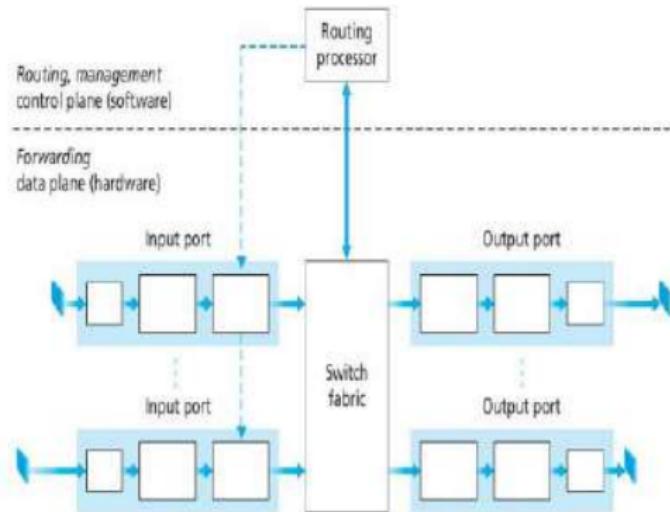


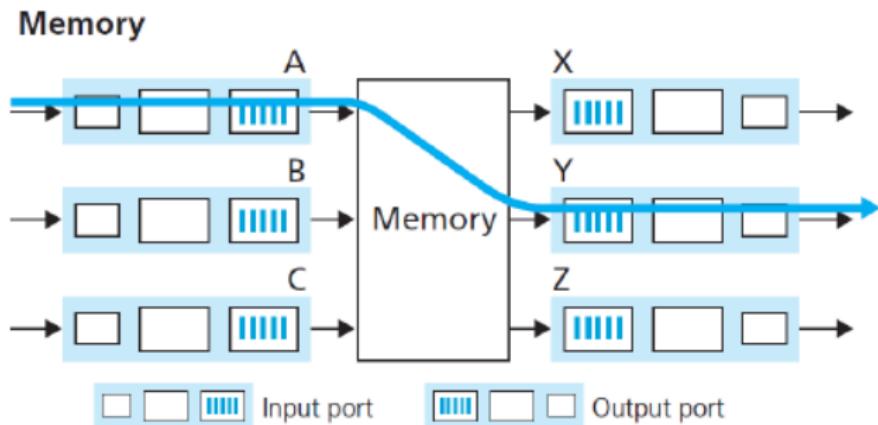
Figure 4.6 • Router architecture

Inside a Router: Switching

Switching via memory

- earliest routers – switching → under direct control of CPU
- *Input port signals the arrival of a packet → routing processor → Interrupt*
- *Processor completes lookup → copies packet → output buffer*
- *Present day routers → processing on a line card*
- shared-memory multiprocessors

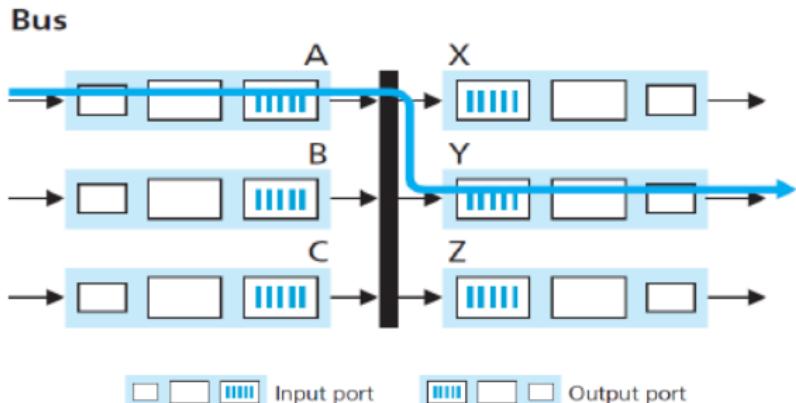
Ex: Cisco's Catalyst 8500 series switches



Inside a Router: Switching:

Switching via a bus:

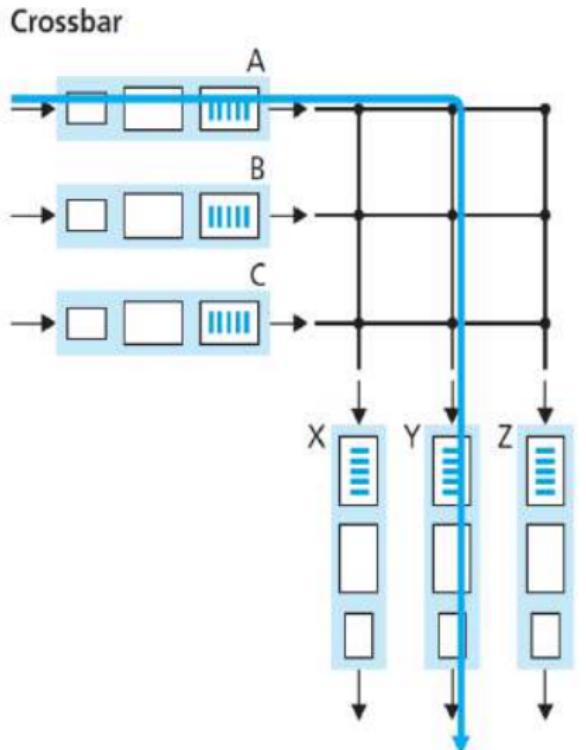
- an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
- input port pre-pend a switch-internal label → indicating the local output port → transmitting the packet onto the bus
- received by all output ports, but only the port that matches the label
- Even multiple packets at input ports → one packet on bus
- switching speed of the router is limited to the bus speed



Inside a Router: Switching

Switching via an interconnection network:

- more sophisticated interconnection network
- crossbar switch is an interconnection network consisting of $2N$ buses that connect N input ports to N output ports
- Each vertical bus intersects each horizontal bus at a crosspoint → can be opened or closed at any time by the switch fabric controller
- crossbar networks are capable of forwarding multiple packets in parallel
- if two packets from two different input ports → to the same output port → one will have to wait at the input



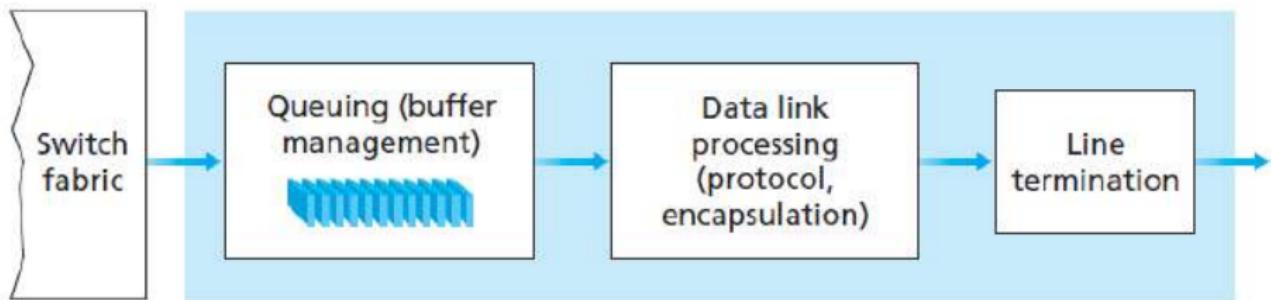
□ □ ■■■ Input port

■■■ □ □ Output port

Inside a Router

Output processing:

- takes stored packets in the output port's memory → transmits them over the output link
- selecting and de-queueing packets for transmission
- performing the needed link layer and physical-layer transmission functions



Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

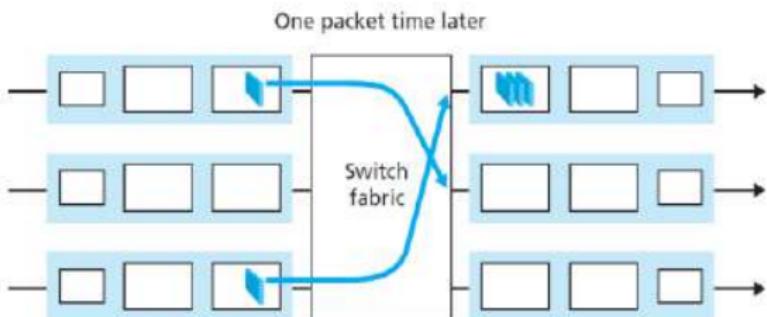
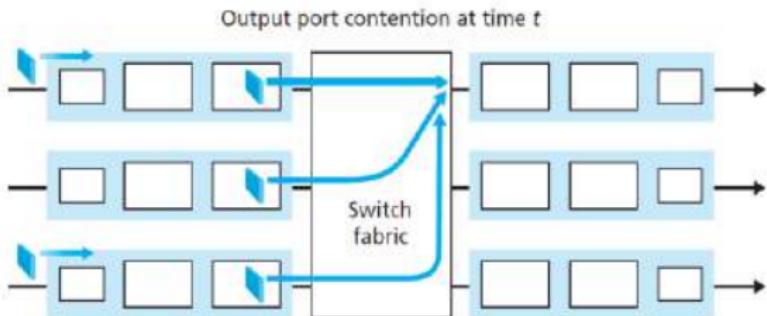
Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
 - the traffic load → the relative speed of the switching fabric,
 - the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N * R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- N arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

Inside a Router: Output processing:

- **packet scheduler** at the output port must choose one packet among those queued for transmission
- first-come-first-served (FCFS)
- weighted fair queuing (WFQ) → shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission
- no enough memory to buffer an incoming packet either drop the arriving packet or remove one or more already-queued packets
- **Random Early Detection** - probabilistic marking/dropping functions



Inside a Router: Output processing:

what if the Switch fabric is not fast enough: → packet queuing at the input ports

Assume:

- (1) all link speeds are identical
- (2) one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link
- (3) packets are moved from a given input queue to their desired output queue in an FCFS manner

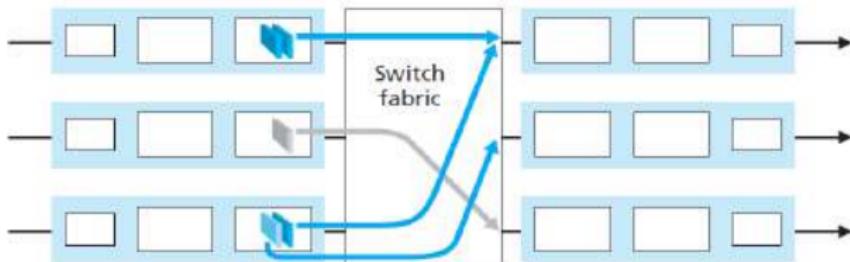
→ Multiple packets can be transferred in parallel, as long as their output ports are different

→ if two packets of two input queues are destined for the same output queue one of the packets will be blocked and must wait at the input queue.

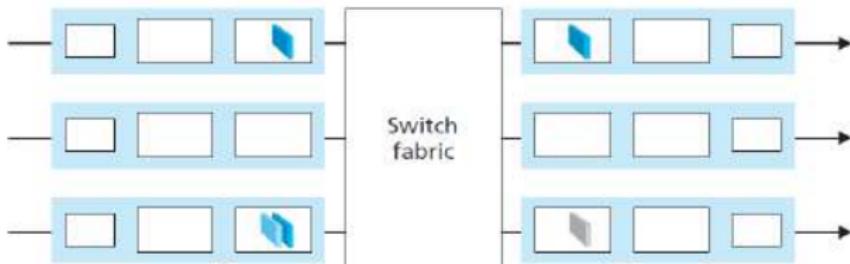
→ **head-of-the-line (HOL) blocking** → queue will grow to unbounded length

Inside a Router: Output processing:

Output port contention at time t—
one dark packet can be transferred



Light blue packet experiences HOL blocking



Key:

destined for upper output port

destined for middle output port

destined for lower output port

Inside a Router: Routing plane

- fully resides and executes in a routing processor within the router
 - network-wide routing control plane → decentralized
- with different pieces executing at different routers and interacting by sending control messages to each other

New router control plane architectures

- part of the control plane is implemented in the routers along with the data plane
- part of the control plane can be implemented externally to the router
- A well-defined API dictates how these two parts interact and communicate with each other

Software Defined Networking (SDN)

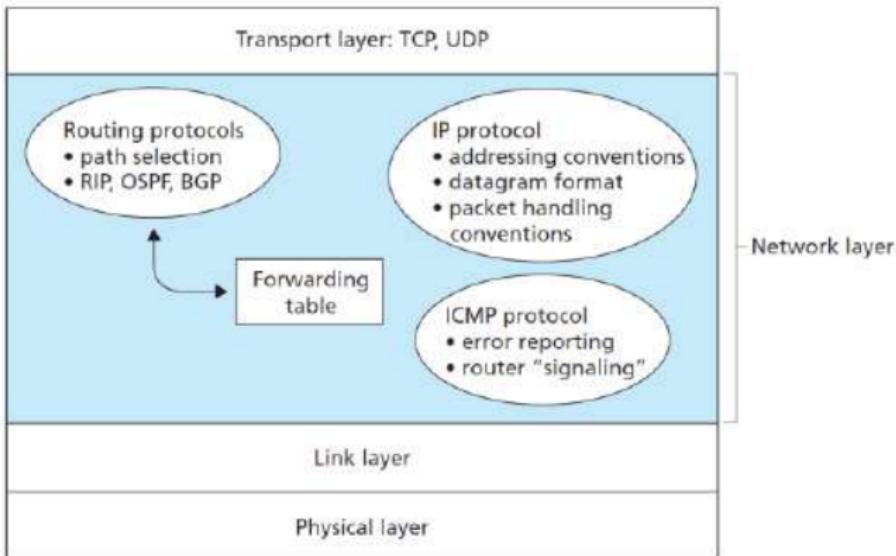
- separating the software control plane from the hardware data plane
- allowing different customized control planes to operate over fast hardware data planes

Internet Protocol

Important modules:

1. IP Protocol
2. Routing component
3. Internet Control Message Protocol

IPV4 and IPV6



Internet Protocol

32 bits					
Version	Header length	Type of service	Datagram length (bytes)		
	16-bit Identifier	Flags	13-bit Fragmentation offset		
Time-to-live	Upper-layer protocol	Header checksum			
32-bit Source IP address					
32-bit Destination IP address					
Options (if any)					
Data					

Internet Protocol

Version number: 4 bits specify the IP protocol version

router can determine how to interpret the remainder of the IP datagram

Header length: IPv4 datagram can contain a variable number of options

→ 4 bits are needed → where in the IP datagram the data actually begins

→ Most of IP datagrams do not contain options → 20-byte header

Type of service: to allow different types of IP datagrams

→ differentiating datagrams requiring low delay, high throughput, or reliability

→ distinguish real-time and non-real time datagrams

Datagram length: total length of the IP datagram → header + data

→ 16 bits long → theoretical maximum size: 65,535 bytes → rarely > than 1500 bytes

Identifier, flags, fragmentation offset: fields required for IP fragmentation

Time to Live:

→ to ensure that datagrams do not circulate forever: long-lived routing loop

→ field is decremented by one each time

Internet Protocol

Protocol:

field is used when an IP datagram reaches its final destination.

value indicates → specific transport-layer protocol to which the data portion of this IP datagram should be passed.

Ex: value-6 indicates → data portion is passed to TCP, 17 indicates to UDP.

Header checksum:

- aids a router in detecting bit errors in a received IP datagram.
- computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic.
- stored in the checksum field and compares with router computed checksum
- detects an error condition → Routers typically discard datagrams
- checksum must be recomputed and stored again at each router → TTL field, and possibly the options field as well, may change.

TCP already has checksum, why do datagrams need checksum?

Source and destination IP addresses:

it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field

Internet Protocol

Options:

- allow an IP header to be extended.
- existence of options does complicate
- datagram headers can be of variable length,
- cannot determine a priori where the data field will start
- amount of time needed to process an IP datagram at a router can vary greatly
- IP options were dropped in the IPv6 header

Data (payload):

- IP datagram contains the transport-layer segment to be delivered to the destination.
 - can carry other types of data → ICMP messages
- ***datagram carrying a TCP segment → each nonfragmented datagram carries a total of 40 bytes of header → 20 bytes of IP header plus 20 bytes of TCP header along with the application-layer message.

Internet Protocol

IP Datagram Fragmentation:

- Not all link-layer protocols can carry network-layer packets of the same size.
- Some protocols can carry big datagrams → other protocols carry only little packets.
 - Ex: Ethernet frames up to 1,500 bytes
 - wide-area links no more than 576bytes.
- **maximum amount of data that a link-layer frame can carry
 - maximum transmission unit (MTU).
 - IP datagram is encapsulated within the link-layer frame for transport from one router to the next router
 - MTU link-layer protocol places a hard limit on the length of an IP datagram.
 - problem: each of the links along the route between sender and destination can use different link-layer protocols
 - each of these protocols can have different MTUs.

→ Router with interconnects several links with different link layer MTU's
may receive a datagram and outgoing link MTU my be smaller

***squeeze this oversized IP datagram into the payload field of the link-layer frame**

Solution: Fragment the IP datagram into two or more smaller datagrams

Internet Protocol - Fragmentation

Fragment:

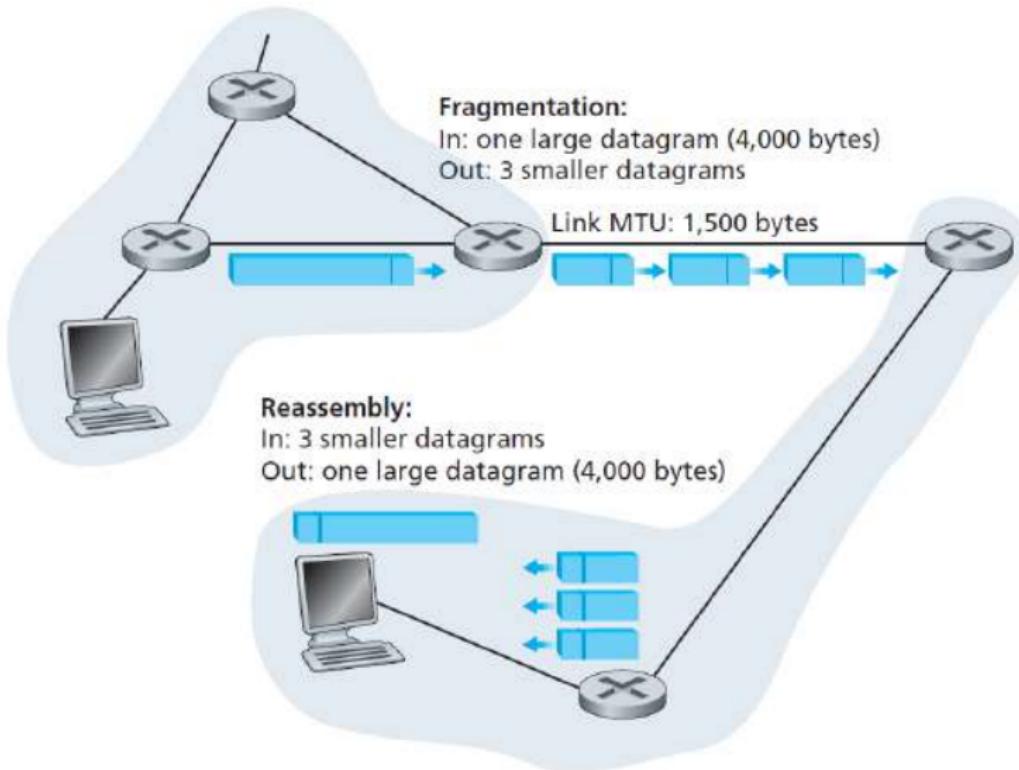
- IP datagram into two or more smaller IP datagrams
 - encapsulate each smaller IP datagrams in a separate link-layerframe;
 - send these frames over the outgoing link. Each smaller datagrams is a **fragment**.
-
- Fragments need to be reassembled before they reach destination.
 - TCP and UDP: expecting to receive complete, unfragmented segments from nwk layer
 - reassembling datagrams in the routers → significant complication damp router performance.
 - datagram reassembly in the end systems rather than in network routers.

Internet Protocol - Fragmentation

Identification flag, and fragmentation offset fields in the IP datagram header:

- destination host to determine whether it has received the last fragment
- how the fragments it has received should be pieced back together to form the original datagram
- sending host stamps the datagram **with an identification number** When a router needs to fragment a datagram
- each resulting datagram is stamped with the **source address, destination address, and identification number** of the original datagram.
- destination receives a series of datagrams
- can examine the identification numbers of the datagrams
- determine which of the datagrams are actually fragments of the same larger datagram
- to make sure, if Destination host has received last fragment of the original datagram,
 - **the last fragment has a flag bit set to 0
 - **other fragments have this flag bit set to 1
- a fragment is missing or reassemble the fragments in their proper order
 - *** the offset field is used to specify where the fragment fits within the original IP datagram.

Internet Protocol - Fragmentation



Internet Protocol - Fragmentation

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that $185 \cdot 8 = 1,480$)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes. $(= 3,980 - 1,480 - 1,480)$ of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that $370 \cdot 8 = 2,960$)	flag = 0 (meaning this is the last fragment)

- ** payload of the datagram is passed to the transport layer only after the IP layer has fully reconstructed the original IP datagram.
- ** If one or more of the fragments does not arrive at the destination, the incomplete datagram is discarded and not passed to the transport layer
- ** TCP will recover from this loss by having the source retransmit the data in the original datagram

Internet Protocol - Fragmentation

Challenges:

complicates routers and end systems → to accommodate datagram fragmentation and reassembly

→ fragmentation can be used to create lethal DoS attacks, attacker sends a series of bizarre and unexpected fragments.

Ex: Jolt2 attack → attacker sends a stream of small fragments to the target host, none of which has an offset of zero. The target can collapse as it attempts to rebuild datagrams out of the degenerate packets.

→ Another class of exploits:

sends overlapping IP fragments, fragments whose offset values are set so that the fragments do not align properly.

IPv6, does away with fragmentation altogether, thereby streamlining IP packet processing and making IP less vulnerable to attack.

Internet Protocol - Addressing

- host typically has only a single link into the network
 - boundary between the host and the physical link is called an **interface**
 - Router to receive a datagram on one link and forward the datagram on some other link → two or more links
 - Boundary between the router and any one of its links: interface: multiple interfaces
 - IP requires each host and router interface to have its own IP address
 - IP address is technically associated with an interface, rather than with the host or router containing that interface
- IP address → 32 bits → 2^{32} → approx 4 billion addresses
- Address are in dotted decimal notation

Ex: 193.32.216.9

11000001 00100000 11011000 00001001

Each interface on every host and router in the global Internet must have an IP address that is globally unique

Internet Protocol - Addressing

IP address of the form 223.1.1.xxx → same leftmost 24 bits in their IP address
four interfaces are also interconnected to each other by a network *that contains no routers*

interfaces would be interconnected by an Ethernet switch

223.1.1.0/24, where the /24 notation → **subnet mask**

→ three host interfaces 223.1.1.1, 223.1.1.2, & 223.1.1.3

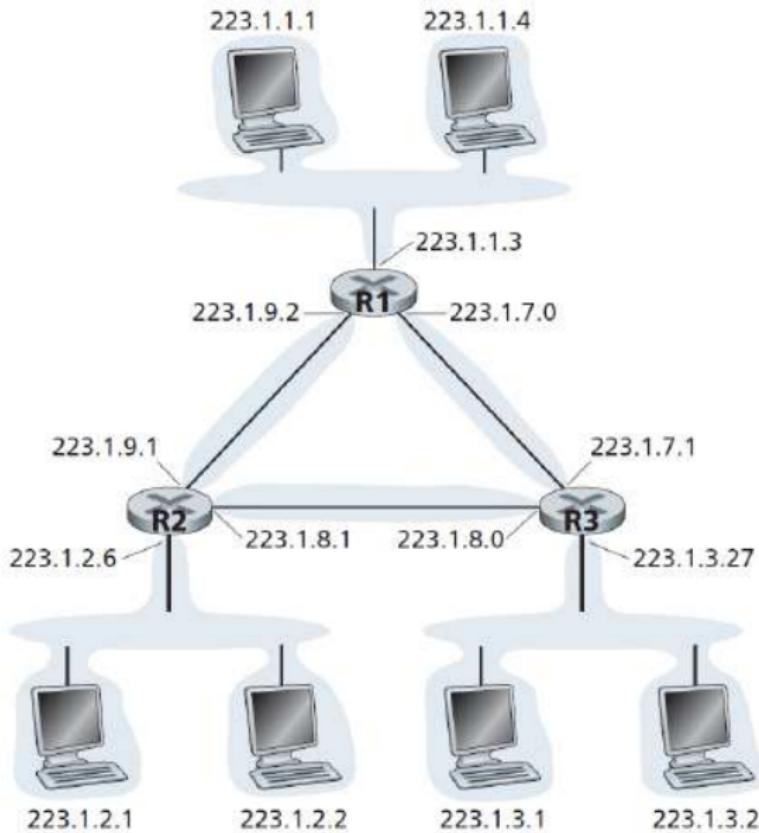
subnet-1: 223.1.1.0/24 , subnet-2: 223.1.2.0/24 , subnet-3: 223.1.3.0/24



Internet Protocol - Addressing

To determine the subnets,
detach each interface from its
host or router,
*Creating islands of isolated
networks,*
with interfaces terminating the
end points of the isolated
networks.
Each of these isolated networks
is called a **subnet**.

How many subnets ? In the
given example network



Internet Protocol - Addressing

Internet's address assignment strategy: **Classless Interdomain Routing (CIDR)**

generalizes the notion of subnet addressing

the 32-bit IP address is divided into two parts

→ dotted-decimal form $a.b.c.d/x$, where x indicates the number of bits in the first part of the address.

The x most significant bits of an address of the form $a.b.c.d/x$ constitute the network portion of the IP address, and are often referred to as the **prefix** (or *network prefix*) of the address

organization is typically assigned a block of contiguous addresses, that is, a range of addresses with a common prefix

IP addresses of devices within the organization will share the common prefix
 x leading prefix bits are considered by routers outside the organization's network

Internet Protocol - Addressing

a router outside the organization → forwards a datagram whose destination address is inside the organization, **only the leading x bits of the address need be considered**

considerably **reduces the size of the forwarding table** in these routers

→ a **single entry** of the form $a.b.c.d/x$ will be sufficient to forward packets to **any destination** within the organization.

→ remaining $32-x$ bits of an address can be thought of as distinguishing among the devices *within* the organization → with **same network prefix**

→ Remaining bits that will be considered when forwarding packets at routers *within* the organization

→ lower-order bits may or may not have an additional subnetting structure

Classful addressing:

the network portions of an IP address constrained to 8, 16, or 24 bits

subnets with 8, 16, and 24-bit subnet addresses → class A, B, and C networks

class C (/24) subnet accommodates: $2^8 - 2 = 254$ hosts

class B (/16) subnet $2^{16} - 2 = 65,634$ hosts

***** broadcast address 255.255.255.255 → message from a host is delivered to all hosts on the same subnet.

Internet Protocol - Addressing

- ability to use a single prefix to advertise multiple networks is often referred to as **address aggregation** or **route aggregation** or **route summarization**.
- works extremely well when addresses are allocated in blocks to ISPs and then from ISPs to client organizations
- what happens when addresses are not allocated in such a hierarchical manner ?

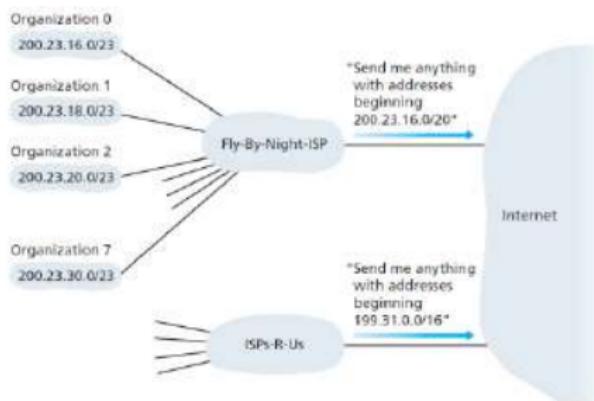


Figure 4.18 • Hierarchical addressing and route aggregation



Figure 4.19 • ISPs-R-U's has a more specific route to Organization 1

Internet Protocol - Addressing

Obtaining a Block of Addresses:

network administrator might first contact its ISP

provide addresses from a larger block of addresses that had already been allocated

ISP may itself have been allocated the address block 200.23.16.0/20

IP addresses are managed under the authority

*****Internet Corporation for Assigned Names and Numbers (ICANN)

→ to allocate IP addresses, to manage the DNSroot servers. It also has the very

→ assigning domain names and resolving domain name disputes

ISP's block	200.23.16.0/20	11001000 00010111 00010000 00000000
Organization 0	200.23.16.0/23	11001000 00010111 00010000 00000000
Organization 1	200.23.18.0/23	11001000 00010111 00010010 00000000
Organization 2	200.23.20.0/23	11001000 00010111 00010100 00000000
...
Organization 7	200.23.30.0/23	11001000 00010111 00011110 00000000

Internet Protocol - Addressing

Host addresses can also be configured
Manually

Dynamic Host Configuration Protocol

(DHCP): allows a host to obtain an IP address automatically

network admin can configure DHCP

→ host receives the same IP address each time it connects to the network,
or

→ A host may be assigned a **temporary II address** that will be different each time the host connects to the network.

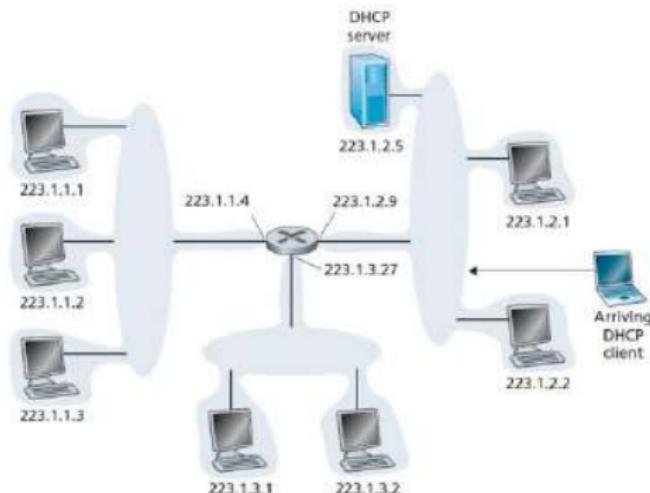


Figure 4.20 • DHCP client-server scenario

- also allows a host to learn additional information → subnet mask, the address of its first-hop router and the address of its local DNS server.
- **plug-and-play protocol**
- in residential Internet access networks and in wireless LANs, where hosts join and leave the network frequently

Internet Protocol - Addressing

Suited for many users coming and going, and addresses are needed for only a limited amount of time

Ex:

In Campus moving from Library to classroom to labs → one subnet to other

residential ISP access networks:

Ex: 2,000 customers, but no more than 400 customers are ever online at the same time

Instead of a block of 2,048 addresses, a DHCP server assigns dynamically needs only 512 addresses : block → a.b.c.d/23.

As the hosts join and leave,

→ DHCP server update its list of available IP addresses.

→ Each time a host joins, the DHCP server allocates an arbitrary address from its current pool of available addresses; each time a host leaves, its address is returned to the pool.

Internet Protocol - Addressing

DHCP is a client-server protocol

- client is typically a newly arriving host wanting to obtain network configuration information
- each subnet will have a DHCP server or DHCP relay agent : typically a router

a four-step process:

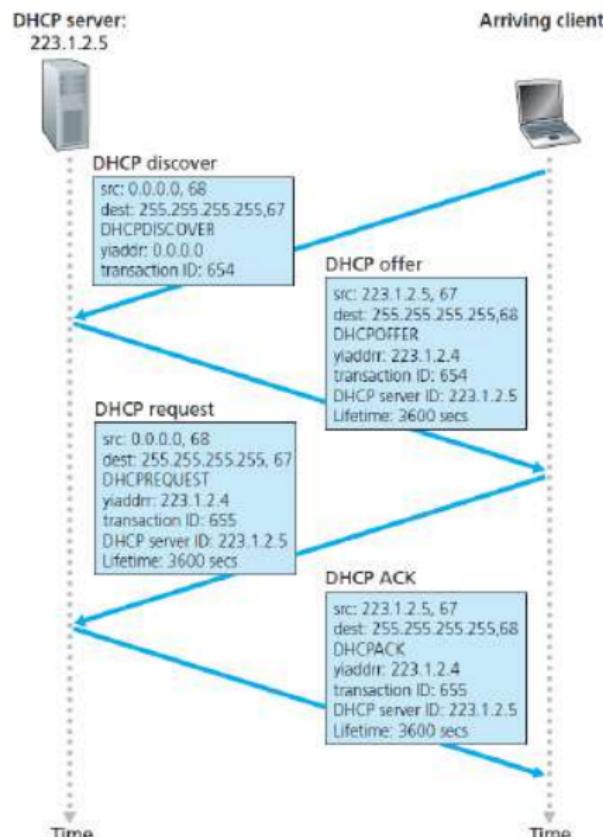
- **DHCP server discovery**
- **DHCP server offer(s)**
- **DHCP request.**
- **DHCP ACK.**

Once the client receives the DHCP ACK, client can use the DHCP-allocated IP address for the lease duration.

Challenges:

→ mobility aspect: each time a node connects to a new subnet, a TCP connection to a remote application cannot be maintained as mobile node moves between subnets

→ single permanent address as it moves between subnets



Internet Protocol - NAT

Network Address Translation (NAT)

- small office, home office (SOHO) subnets
- what if the ISP had already allocated the contiguous portions of the SOHO network's current address range?
- for a private network or a **realm** with private addresses
- A *realm with private addresses* refers to a network whose addresses only have meaning to devices within that network
- Devices within a given home network can send packets to each other using 10.0.0.0/24 addressing
- packets forwarded *beyond* the home network into the larger global Internet clearly cannot use these addresses
- if private addresses only have meaning within a given network, how is addressing handled when packets are sent to or received from the global Internet, where addresses are necessarily unique?

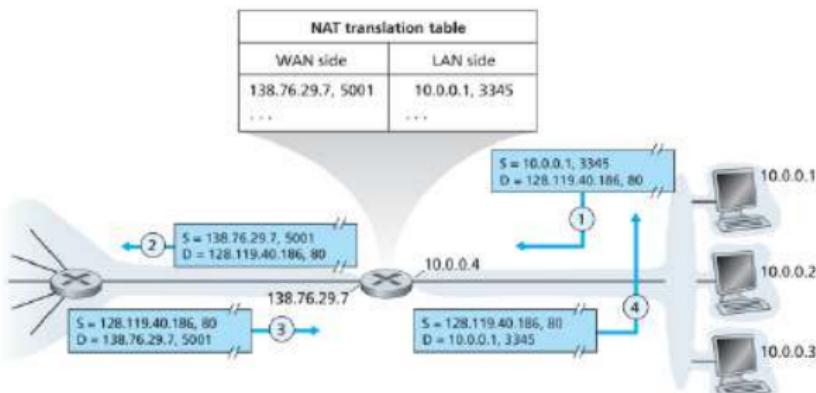


Figure 4.22 • Network address translation

Internet Protocol - NAT

Network Address Translation (NAT)

- NAT router behaves to the outside world as a *single* device with a *single* IP address
- NAT-enabled router is hiding the details of the home network from the outside world
- The router gets its address from the ISP's DHCP server
- the router runs a DHCP server to provide addresses to computers within the NAT-DHCP-router-controlled home network's address space.
- **NAT translation table** at the NAT router → to include port numbers as well as IP addresses in the table entries

Example: Host in Home network requests a web server (port 80) with IP – 128.119.40.186
assigns source port: 3345 and sends to datagram.

- NAT router receives the datagram, generates a new source port number 5001 for the datagram, replaces the source IP address with its WAN-side IP address 138.76.29.7 --
- replaces the original source port number 3345 with the new source port number 5001
- NAT router can select any source port number that is not currently in the NAT translation table
- Web server unaware → the arriving datagram containing the HTTP request has been manipulated by the NAT router
- responds with a datagram whose destination address is the IP address of the NAT router, and whose destination port number is 5001

Internet Protocol - NAT

Network Address Translation (NAT)

- port numbers are meant to be used for addressing processes, not for addressing hosts → violation can indeed cause problems for servers running on the home network
- NAT protocol violates the so-called end-to-end argument; that is, hosts should be talking directly with each other, without interfering nodes modifying IP addresses and port numbers
- we should use IPv6 (see Section 4.4.4) to solve the shortage of IP addresses, rather than recklessly patching up the problem with a stopgap solution like NAT. But like it or not, NAT has become an important component of the Internet
- NAT interferes with P2P applications, including P2P file-sharing applications and P2P Voice-over-IP applications

Internet Protocol - ICMP

Internet Control Message Protocol (ICMP)

- used by hosts and routers to communicate network-layer information to each other
- most typical use of ICMP is for error reporting.
- For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as “Destination network unreachable.” → origins in ICMP
- IP router was unable to find a path to the host specified in your Telnet, FTP, or HTTP application. That router created and sent a type-3 ICMP message to your host indicating the error
- ICMP is often considered part of IP → lies just above IP,
 - as ICMP messages are carried inside IP datagrams. ICMP messages are carried as IP payload, just as TCP or UDP segments are carried as IP payload.
 - when a host receives an IP datagram with ICMP specified as the upper-layer protocol, it demultiplexes the datagram’s contents to ICMP, just as it would demultiplex a datagram’s content to TCP or UDP.
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram that caused the ICMP message to be generated in the first place
- well-known ping program sends an ICMP type 8 code 0 message to the specified host. The destination host, seeing the echo request, sends back a type 0 code 0 ICMP echo reply.

Internet Protocol - ICMP

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 4.23 • ICMP message types

Internet Protocol - ICMP

Internet Control Message Protocol (ICMP)

- ICMP : **source quench message**
 - to perform congestion control—to allow a congested router to send an ICMP source quench message to a host to force that host to reduce its transmission rate. TCP has its own congestion-control mechanism that operates at the transport layer, without the use of network-layer feedback such as the ICMP source quench message.

Traceroute program:

- to trace a route from a host to any other host in the world is implemented with ICMP messages
- source sends a series of ordinary IP datagrams to the destination. Each of these datagrams carries a UDP segment with an unlikely UDP port number
- The first of these datagrams has a TTL of 1, the second of 2, the third of 3, and so on. The source also starts timers for each of the datagrams.
- When the n th datagram arrives at the n th router, the n th router observes that the TTL of the datagram has just expired. Rules of the IP protocol, the router discards the datagram and sends an ICMP warning message to the source (type 11 code 0).
- warning message includes the name of the router and its IP address. When this ICMP message arrives back at the source, the source obtains the round-trip time from the timer and the name and IP address of the n th router from the ICMP message.

Internet Protocol - ICMP

Internet Control Message Protocol (ICMP)

Traceroute program:

How does a Traceroute source know when to stop sending UDP segments?

- source increments the TTL field for each datagram it sends.
- one of the datagrams will eventually make it all the way to the destination host.
- this datagram contains a UDP segment with an unlikely port number, the destination host sends a port unreachable ICMP message (type 3 code 3) back to the source.
- the source host receives this particular ICMP message, it knows it does not need to send additional probe packets.
- source host learns the number and the identities of routers that lie between it and the destination host and the round-trip time between the two hosts.

Internet Protocol – IPV6

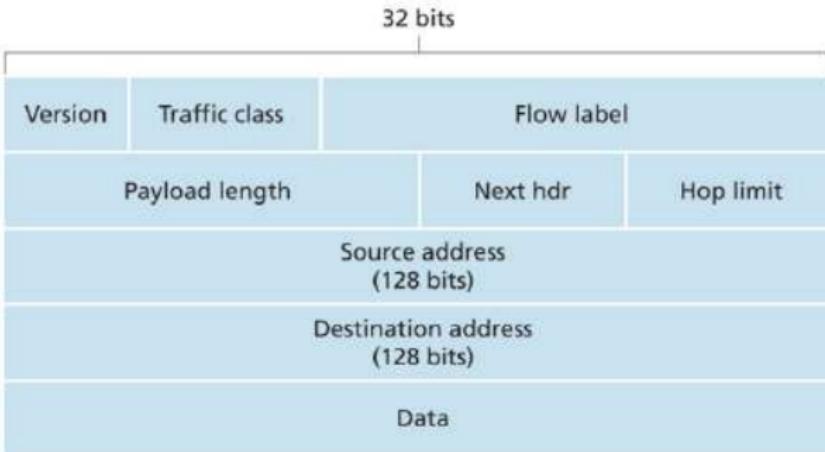
- 32-bit IP address space was beginning to be used up
- need for a large IP address space, a new IP protocol: IPv6

IPv6 Datagram Format *streamlined 40-byte*

header

- 40-byte fixed-length header for faster processing of the IP datagram

- → 4-bit field identifies the IP version number



Traffic class : 8-bit field similar to the TOS field in IPv4

Payload length: 16-bit value → number of bytes in the IPv6 datagram + 40-byte datagram header.

Next header: field identifies the protocol to which the contents

example: to TCP or UDP, uses same fields as IPV4 header

Internet Protocol – IPV6

Hop limit: field decremented by one by each router that forwards the datagram.
→ hop limit count reaches zero, the datagram is discarded.

Source and destination addresses: The various formats of the IPv6 128-bit address

Data: This is the payload portion of the IPv6 datagram.

Unique features of IPV6:

Expanded addressing capabilities: size of the IP address from 32 to 128 bits

→ IPv6 new type of address → **anycast address** → allows a datagram to be delivered to any one of a group of hosts.

Flow labeling and priority: IPv6 definition of a **flow**.

- labeling of packets belonging to particular flows for which the sender requests special handling → non-default quality of service or real-time service.
example, audio and video transmission might likely be treated as a flow. On the traditional applications → file transfer and e-mail, might not be treated as flows.
- traffic carried by a high-priority user might also be treated as a flow.
- The IPv6 header also has an 8-bit traffic class field.
- This field, like the TOS field in IPv4, can be used to give priority to certain datagrams within a flow
- it can be used to give priority to datagrams from certain applications
ex: ICMP over datagrams from other applications

Internet Protocol – IPV6

Fields dropped from IPv4 :

Fragmentation/Reassembly:

- does not allow for fragmentation and reassembly at intermediate routers;
- these operations can be performed only by the source and destination.
- If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram
- sends a “Packet Too Big” ICMP error message back to the sender.
- The sender can then resend the data, using a smaller IP datagram size.
- removing this functionality from the routers and placing it squarely in the end systems considerably speeds up IP forwarding within the network.

Header checksum.

- Transport-layer and link-layer protocols in the Internet layers perform check sum
- functionality is redundant in the network layer → removed.
- fast processing of IP packets was a central concern → the IPv4 header contains a TTL field (similar to the hop limit field in IPv6), the IPv4 header checksum needed to be recomputed at every router.

Options: no longer a part of the standard IP header

New version of ICMP for IPv6 : ICMPv6: added new types and codes required by IPv6

- “Packet Too Big” type, and an “unrecognized IPv6 options” error code.
- ICMPv6 subsumes the functionality of the Internet Group Management Protocol (IGMP)

Internet Protocol – IPv6

Transitioning from IPv4 to IPv6 :

- public Internet, which is based on IPv4, be transitioned to IPv6
- new IPv6-capable systems can be made backward compatible
- can send, route, and receive IPv4 datagrams, already deployed IPv4-capable systems are not capable of handling IPv6 datagrams

Two ways :

Dual-stack approach:

- IPv6 nodes also have a complete IPv4 implementation
- an IPv6/IPv4 → node ability to send and receive both IPv4 and IPv6 datagrams
- interoperating with an IPv4 node → IPv6/IPv4 node can use IPv4 datagrams → when interoperating with an IPv6 node, it can speak IPv6
- if either the sender or the receiver is only IPv4- capable, an IPv4 datagram must be used.
- As a result, it is possible that two IPv6- capable nodes can end up, in essence, sending IPv4 datagrams to each other.
- in performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram that have no counterpart in IPv4 → information in these fields will be lost.

Internet Protocol – IPv6

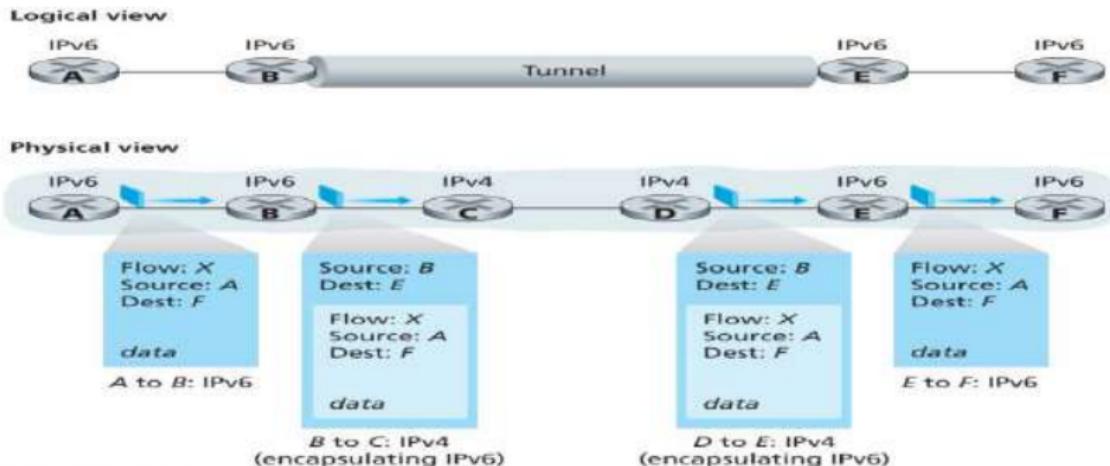


Figure 4.26 • Tunneling

Tunneling:

- two IPv6 nodes want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers
- the intervening set of IPv4 routers between two IPv6 routers as a **tunnel**
- IPv6 node on the sending side of the tunnel takes the *entire* IPv6 datagram and puts it in the data field of an IPv4 datagram.
- This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel and sent to the first node in the tunnel.

Internet Protocol – IPv6

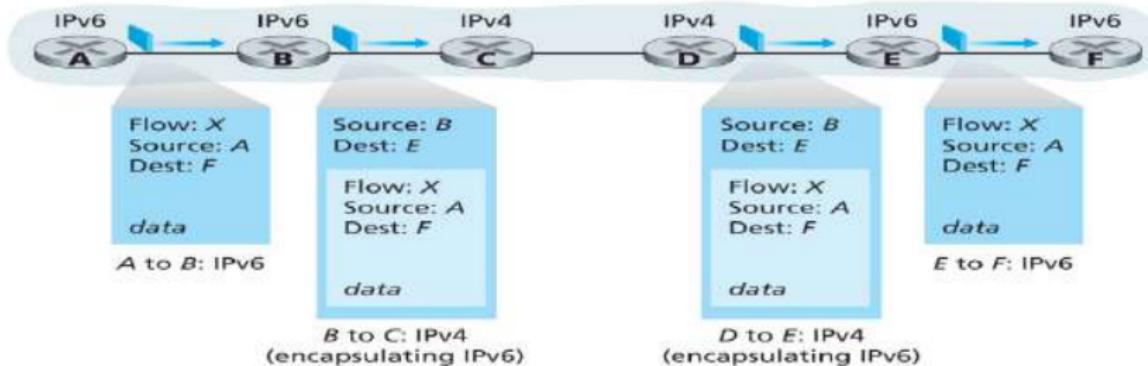


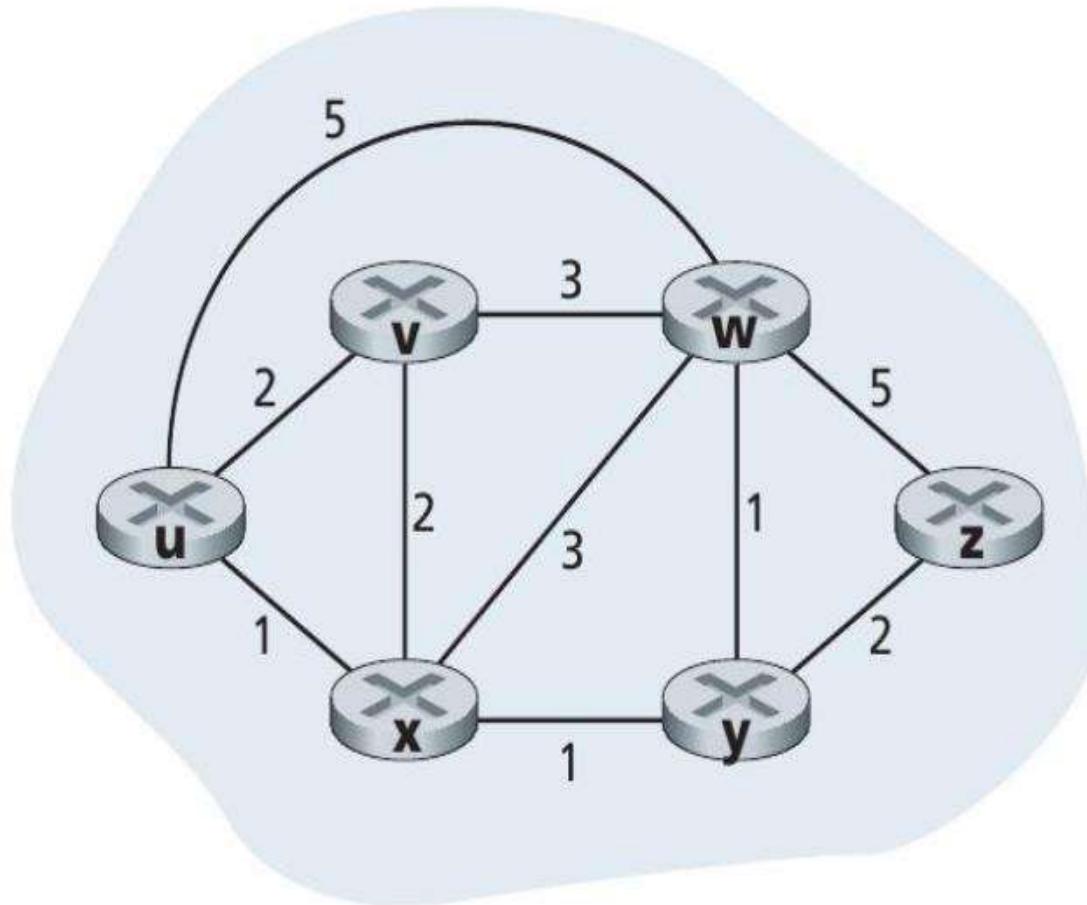
Figure 4.26 • Tunneling

Tunneling:

- The intervening IPv4 routers in the tunnel route this IPv4 datagram as any other datagram
- IPv4 datagram itself contains a complete IPv6 datagram.
- The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram
- determines that the IPv4 datagram contains an IPv6 datagram → extracts the IPv6 datagram
- routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor.

Routing

Notation



Notation

- We represent a network by an **undirected graph** $G = (N, E)$
- N is the set of **Nodes (routers)**
- E is the set of edges connecting nodes (**links**)
- $c(x, y)$ is the cost of the edge between x and y .
- Cost of a path (x_1, \dots, x_p) is sum of costs of edges along the path: $c(x_1, x_2) + \dots + c(x_{p-1}, x_p)$
- We aim to find paths with **least cost**.

Classification

- **Global vs Decentralized:**
 - Global routing algorithm: requires global information about links and costs at every router. Also known as Link-State algorithm
 - Decentralized routing algorithm: no node has complete information
- **Static vs Dynamic** routing
- **Load-sensitive vs Load-insensitive** routing

Link-State Routing Algorithm

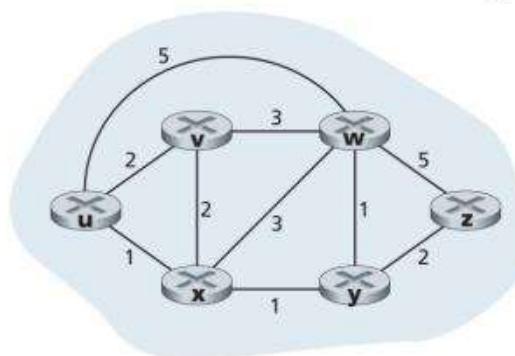
- We study Dijkstra's algorithm
- $D(v)$: cost of the least cost path from source to destination v as of this iteration
- $p(v)$: previous node along the current least cost path from the source to v
- N' : subset of N . If $v \in N'$, then least cost path to v from source is definitely known.

LS Algorithm

```
1 Initialization:  
2   N' = {u}  
3   for all nodes v  
4     if v is a neighbor of u  
5       then D(v) = c(u,v)  
6     else D(v) = ∞  
7  
8 Loop  
9   find w not in N' such that D(w) is a minimum  
10  add w to N'  
11  update D(v) for each neighbor v of w and not in N':  
12    D(v) = min( D(v), D(w) + c(w,v) )  
13  /* new cost to v is either old cost to v or known  
14    least path cost to w plus cost from w to v */  
15 until N' = N
```

LS Routing Algorithm: Example

step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyw		3,y			4,y
4	uxywv					4,y
5	uxywvwz					



Distance-Vector (DV) Routing Algorithm

- Decentralized, asynchronous
- Iterative process
- $d_x(y)$ denotes cost of least cost path from x to y
- **Bellman-Ford** equation

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\},$$

v is a neighbor of x .

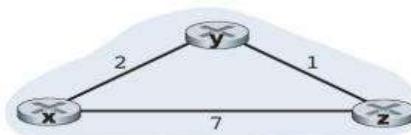
- Each node x maintains the following **routing information**:
 - For each neighbor v , the cost $c(x, v)$
 - Node x 's distance vector, $\mathbf{D}_x = [D_x(y) : y \in N]$
 - Distance vectors of each of its neighbors \mathbf{D}_v

DV Algorithm

At each node, x :

```
1 Initialization:  
2   for all destinations y in N:  
3      $D_x(y) = c(x,y)$  /* if y is not a neighbor then  $c(x,y) = \infty$  */  
4   for each neighbor w  
5      $D_w(y) = ?$  for all destinations y in N  
6   for each neighbor w  
7     send distance vector  $D_x = [D_x(y): y \text{ in } N]$  to w  
8  
9 loop  
10  wait (until I see a link cost change to some neighbor w or  
11    until I receive a distance vector from some neighbor w)  
12  
13  for each y in N:  
14     $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$   
15  
16  if  $D_x(y)$  changed for any destination y  
17    send distance vector  $D_x = [D_x(y): y \text{ in } N]$  to all neighbors  
18  
19 forever
```

DV Example



$$D_x(x) = 0$$

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2 + 0, 7 + 1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2 + 1, 7 + 0\} = 3$$

Node x table

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
z	∞	∞	∞	

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z	7	1	0	

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
z	3	1	0	

Node v table

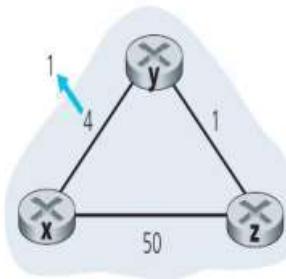
		cost to			cost to			cost to		
		x	y	z	x	y	z	x	y	z
from	x	∞	∞	∞	0	2	7	0	2	3
	y	2	0	1	2	0	1	2	0	1
	z	∞	∞	∞	7	1	0	3	1	0

Node z table

		cost to			cost to			cost to		
		x	y	z	x	y	z	x	y	z
from	x	∞	∞	∞	0	2	7	0	2	3
	y	∞	∞	∞	2	0	1	2	0	1
	z	7	1	0	3	1	0	3	1	0

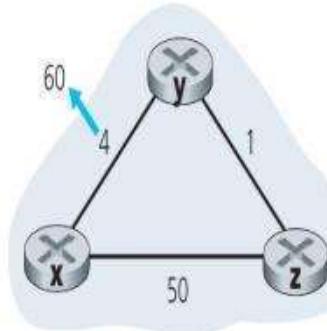
Time

DV Algorithm: Link Cost Changes



- Focus on **distance tables entries of y and z to x**
- At t_0 , cost has changed to 1 from 4. y updates its table with $D_y(x) = 1$ and informs z
- At t_1 , z receives update from y and updates its table $D_z(x) = 2$
- At t_2 , y receives update from z and no changes in table.

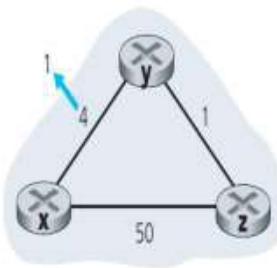
DV Algorithm: Link Cost Changes



- Before link cost changes:

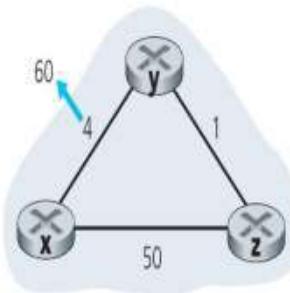
$$D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$$

DV Algorithm: Link Cost Changes



- Focus on distance tables entries of y and z to x
- At t_0 , cost has changed to 1 from 4. y updates its table with $D_y(x) = 1$ and informs z
- At t_1 , z receives update from y and updates its table $D_z(x) = 2$
- At t_2 , y receives update from z and no changes in table.

DV Algorithm: Link Cost Changes



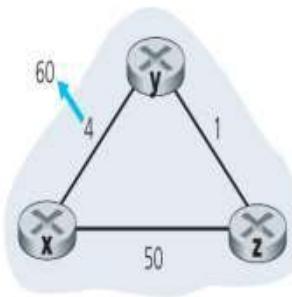
- Before link cost changes:

$$D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$$

- At t_0 , cost has changed to 60 from 4. y updates its table with

$$D_y(x) = \min\{c(y, x) + D_x(x), c(y, z) + D_z(x)\} \quad (1)$$

DV Algorithm: Link Cost Changes

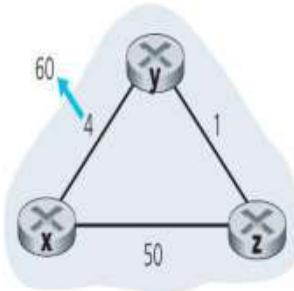


- Before link cost changes:
 $D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$
 - At t_0 , cost has changed to 60 from 4. y updates its table with

$$D_y(x) = \min\{c(y, x) + D_x(x), c(y, z) + D_z(x)\} \quad (1)$$

- $D_y(x) = \min\{60 + 0, 1 + 5\} = 6$

DV Algorithm: Link Cost Changes



- Before link cost changes:

$$D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$$

- At t_0 , cost has changed to 60 from 4. y updates its table with

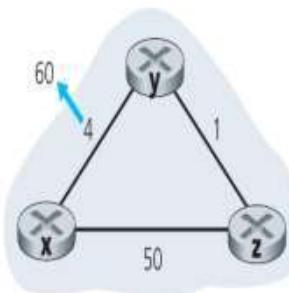
$$D_y(x) = \min\{c(y, x) + D_x(x), c(y, z) + D_z(x)\} \quad (1)$$

- $D_y(x) = \min\{60 + 0, 1 + 5\} = 6$

- At t_1 , z receives update from y and updates its table

$$D_z(x) = \min\{50 + 0, 1 + 6\} = 7$$

DV Algorithm: Link Cost Changes



- Before link cost changes:

$$D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$$

- At t_0 , cost has changed to 60 from 4. y updates its table with

$$D_y(x) = \min\{c(y, x) + D_x(x), c(y, z) + D_z(x)\} \quad (1)$$

- $D_y(x) = \min\{60 + 0, 1 + 5\} = 6$

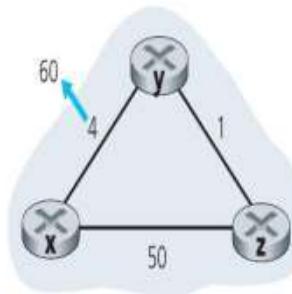
- At t_1 , z receives update from y and updates its table

$$D_z(x) = \min\{50 + 0, 1 + 6\} = 7$$

- At t_2 , y receives update from z and updates table as

$$D_y(x) = 8. \text{ and this process repeats.}$$

DV Algorithm: Link Cost Changes



- Before link cost changes:

$$D_y(x) = 4, D_y(z) = 1, D_z(y) = 1, D_z(x) = 5$$

- At t_0 , cost has changed to 60 from 4. y updates its table with

$$D_y(x) = \min\{c(y, x) + D_x(x), c(y, z) + D_z(x)\} \quad (1)$$

- $D_y(x) = \min\{60 + 0, 1 + 5\} = 6$

- At t_1 , z receives update from y and updates its table

$$D_z(x) = \min\{50 + 0, 1 + 6\} = 7$$

- At t_2 , y receives update from z and updates table as

$$D_y(x) = 8. \text{ and this process repeats.}$$

- Count-to-infinity!

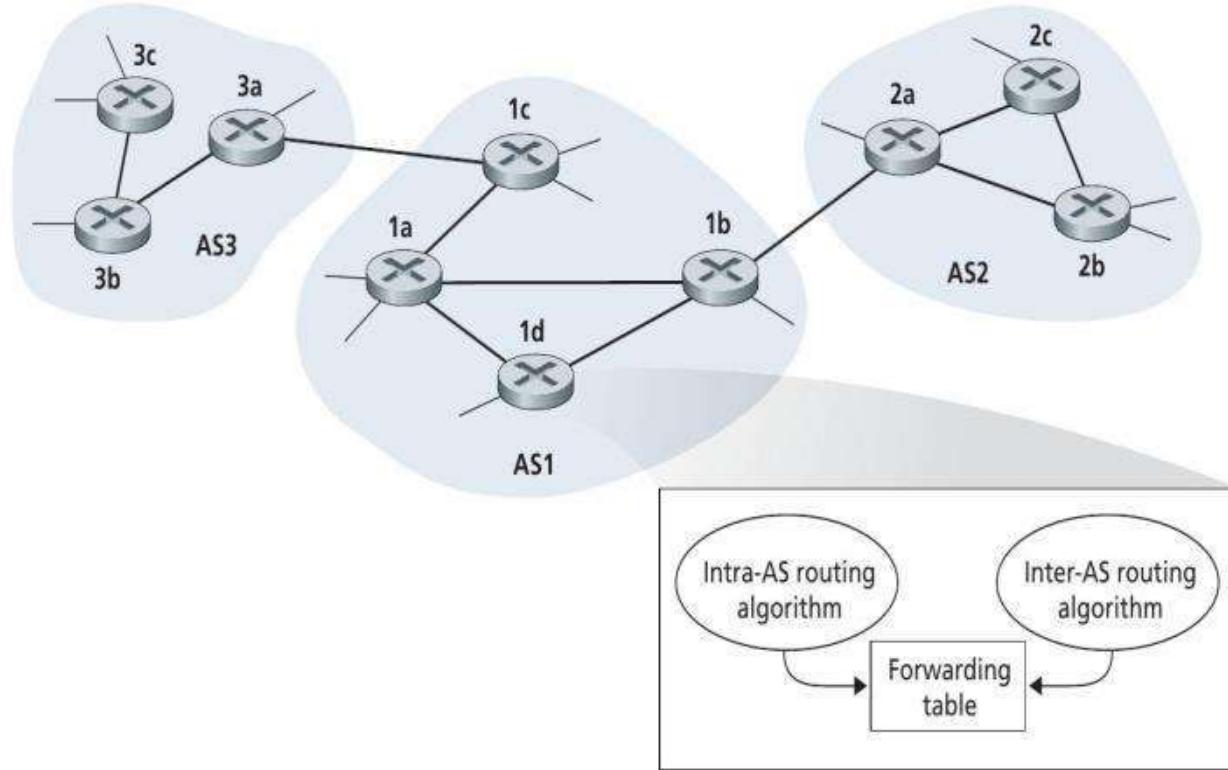
Poisoned Reverse

- If z routes through y , z will inform y that $D_z(x) = \infty$
- y cannot route to x via z as there is **no path!**
- When $c(x, y) = 60$, y updates its table with $D_y(x) = 60!$
- After receiving an update z routes to x via direct path and updates its table with $D_z(x) = 50$
- After receiving update from z , y recomputes route to x via z and informs z with $D_y(x) = \infty$ (**infact it is 51!**)

Hierarchical Routing

- **Scale**: number of routers in internet is very large. Which algorithm to use?
- **Administrative autonomy**: an organization should be able to run and administer its network as it wishes.
- These problems can be solved by organizing routers into **autonomous systems (AS)**
- Each AS will have a **gateway router**

Hierarchical routing



- Hot-potato routing

Routing in Internet

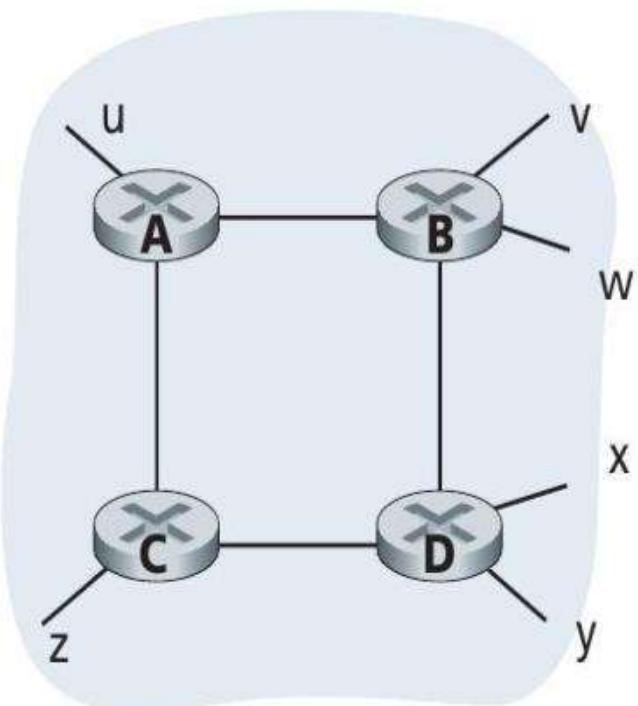
- Intra-AS routing
 - Routing information protocol (RIP): based on DV algorithm
 - Open shortest path first (OSPF): based on LS algorithm
- Inter-AS routing
 - Border Gateway Protocol (BGP)

BGP (Section 4.6.3) is left for self study! Its part of our CCN course

Routing Information Protocol

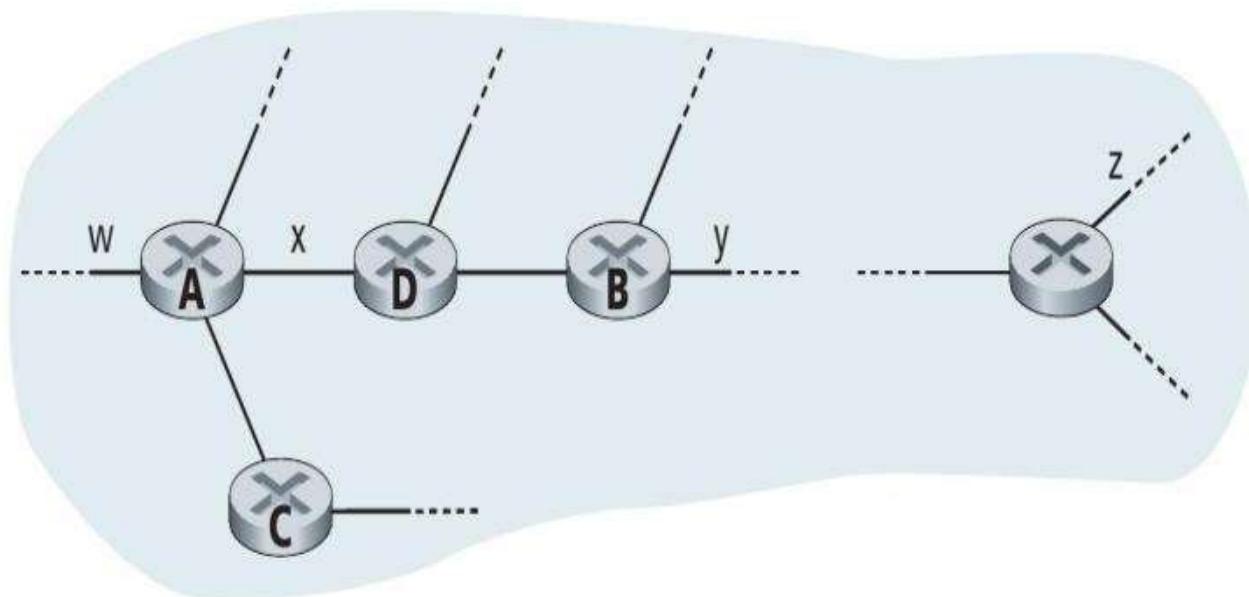
- In RIP, a **hop** means a **subnet**
- Cost of a path from source router to destination subnet is the number of hops (**subnets**) along the path including the destination subnet
- The maximum cost of a path is **limited to 15**
- RIP uses distance vector algorithm: the routers need to exchange distance vectors or routing updates every **30 seconds**
- The **RIP response messages** or **RIP advertisements** can contain a list up to 25 destination subnets within the AS.

Example at A



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

A portion of AS



Routing Table at D

Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
z	B	7
x	-	1
....

Advertisement from A

Destination Subnet	Next Router	Number of Hops to Destination
Z	C	4
W	-	1
X	-	1
...

Updated Routing Table at D

Destination Subnet	Next Router	Number of Hops to Destination
W	A	2
Y	B	2
Z	A	5
....

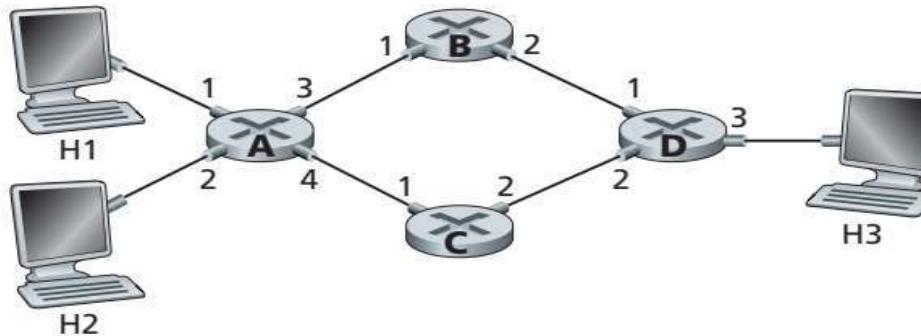
If a router does not hear from its neighbor once every 180 seconds, that neighbor is considered **dead**. The router propagates about this information to its neighboring routers that are **alive!**

Open Shortest Path First

- OSPF uses Dijkstra's shortest-path algorithm
- Choice of link cost is left to the administrator.
- A router broadcasts routing information to **all other routers** in the AS.
- A router broadcasts link's state whenever **there is a change** and **periodically** every 30 seconds.
- OSPF provides features such as **security**, multiple same-cost paths
- RIP and OSPF are in wide use: regional ISPs use RIP, top-tier ISPs use OSPF.

Tutorial_Network_Layer

- a. Suppose that this network is a datagram network. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.
- b. Suppose that this network is a datagram network. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (Hint: this is a trick question.)
- c. Now suppose that this network is a virtual circuit network and that there is one ongoing call between H1 and H3, and another ongoing call between H2 and H3. Write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4.
- d. Assuming the same scenario as (c), write down the forwarding tables in nodes B, C, and D.

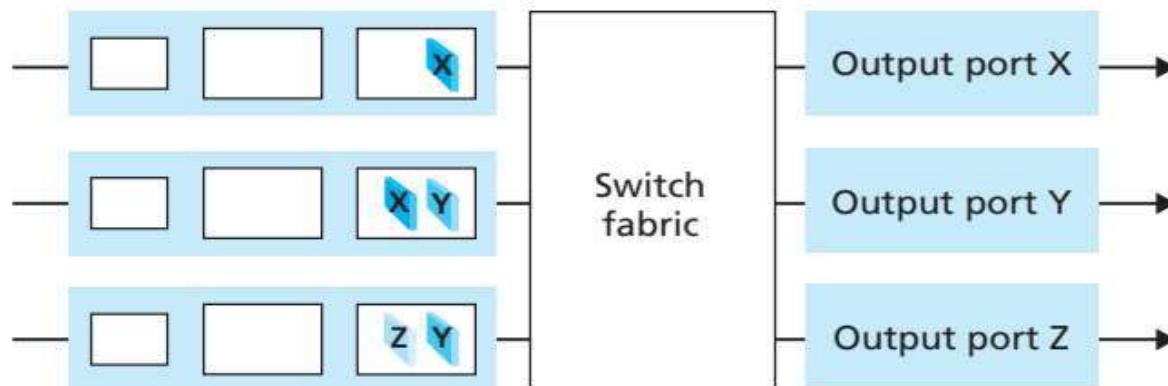


Tutorial_Network_Layer

- P8. In Section 4.3, we noted that the maximum queuing delay is $(n-1)D$ if the switching fabric is n times faster than the input line rates. Suppose that all packets are of the same length, n packets arrive at the same time to the n input ports, and all n packets want to be forwarded to *different* output ports. What is the maximum delay for a packet for the (a) memory, (b) bus, and (c) crossbar switching fabrics?

Tutorial_Network_Layer

P9. Consider the switch shown below. Suppose that all datagrams have the same fixed length, that the switch operates in a slotted, synchronous manner, and that in one time slot a datagram can be transferred from an input port to an output port. The switch fabric is a crossbar so that at most one datagram can be transferred to a given output port in a time slot, but different output ports can receive datagrams from different input ports in a single time slot. What is the minimal number of time slots needed to transfer the packets shown from input ports to their output ports, assuming any input queue scheduling order you want (i.e., it need not have HOL blocking)? What is the largest number of slots needed, assuming the worst-case scheduling order you can devise, assuming that a non-empty input queue is never idle?



Link Layer

Dr. Raja Vara Prasad,
IIIT Sri City, Chittoor

Link Layer

- Moves datagrams node-to-node
- Link layer protocols: Ethernet, IEEE 802.11 (Wireless LAN/Wifi), Token-ring, PPP
- Services
 - Framing
 - Link-access
 - Reliable-delivery
 - Flow Control
 - Error detection
 - Error correction
 - Half-duplex and Full-duplex

Link Layer

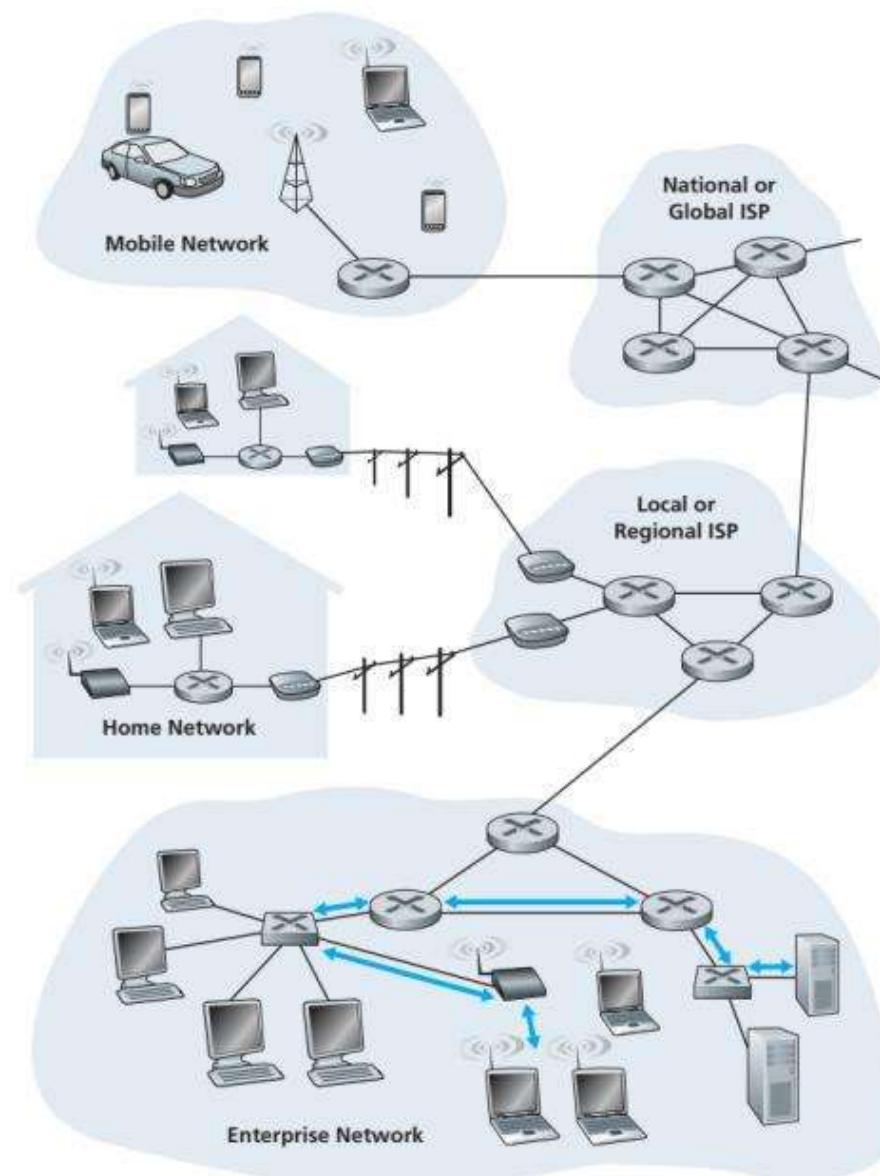
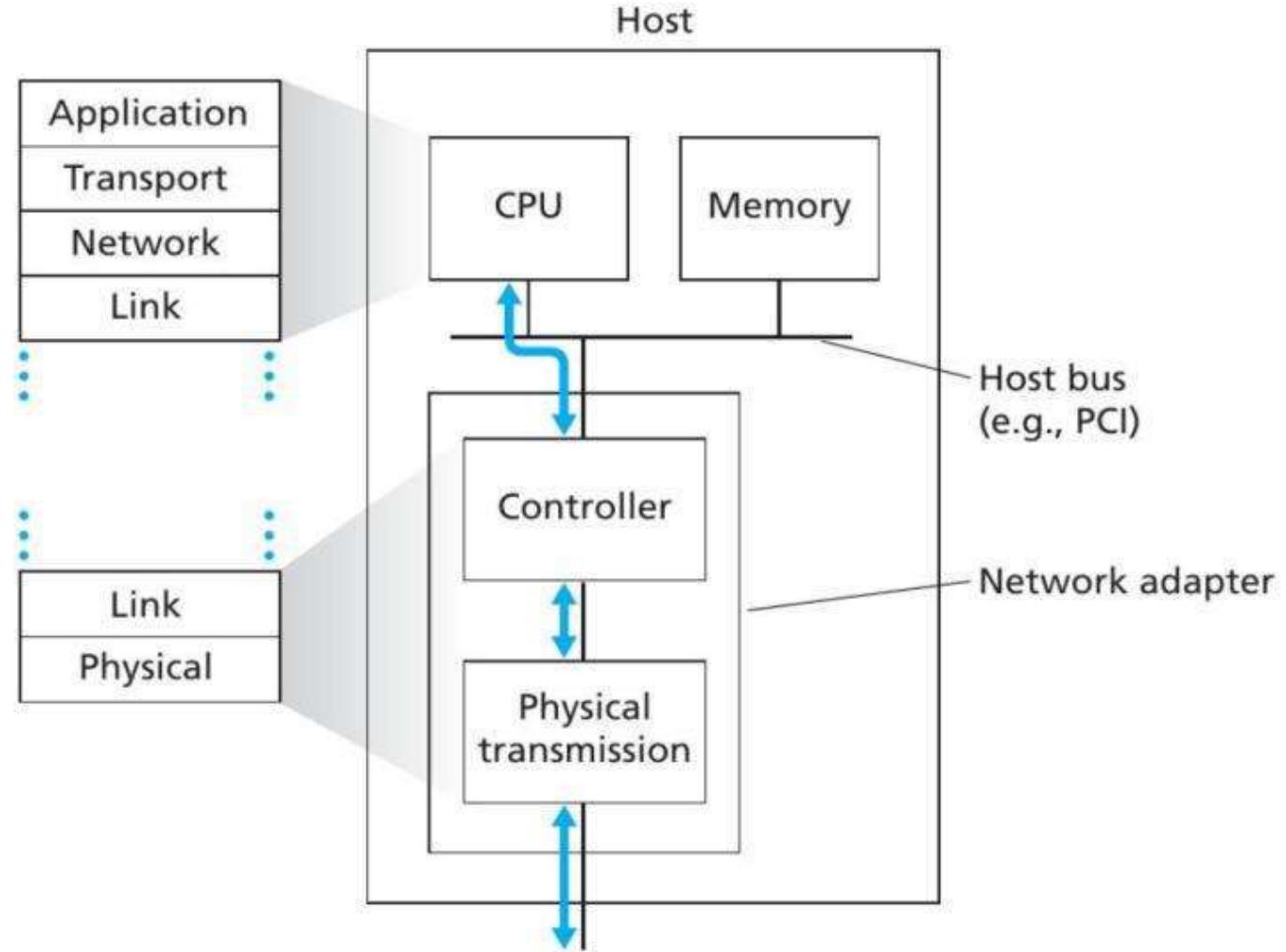


Figure 5.1 • Six link-layer hops between wireless host and server

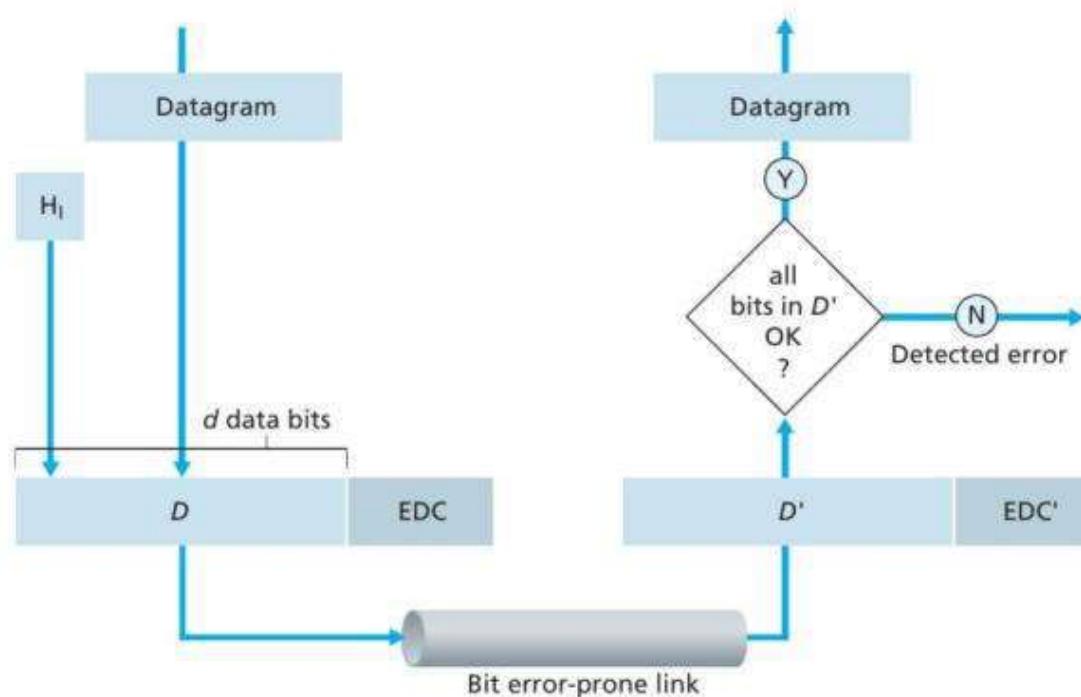
Where is the Link layer implemented



Link Layer Implementation

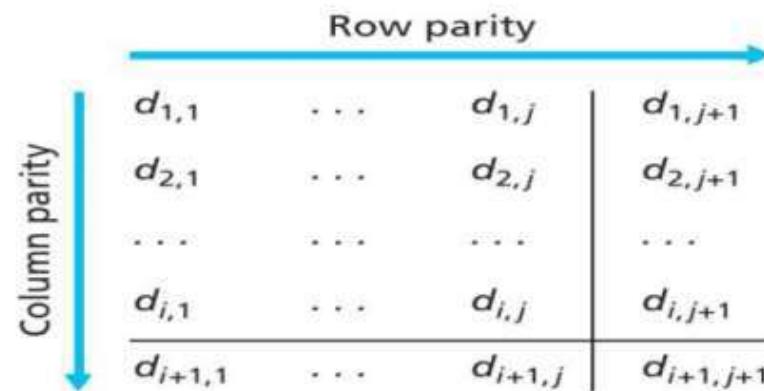
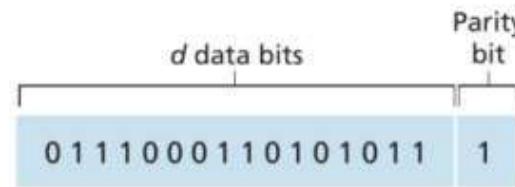
- **Software components**
 - receiving datagram from network layer
 - assembling link-layer addressing information
 - activating the controller hardware
- **Hardware components**
 - transfer frame from one adapter to another adapter
 - error detection and correction

Error Detection and Correction



- **EDC**: error detection and correction bits
- Parity checks
- Checksumming methods
- Cyclic redundancy checks (CRC)

Parity Checks



Two tables illustrating error detection and correction. Both tables have a vertical line separating data from parity.

No errors:

1	0	1	0	1	1	1
1	1	1	1	0	0	0
0	1	1	1	0	1	1
					0	0

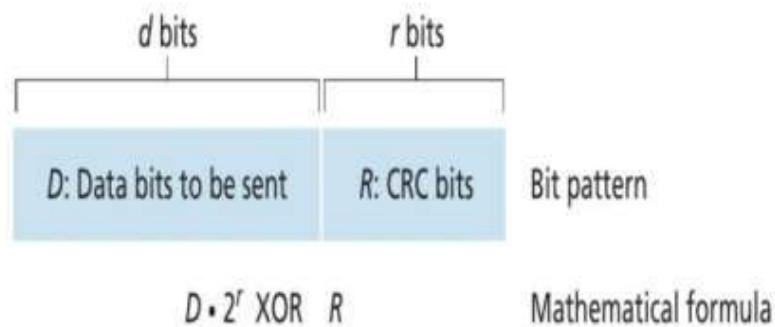
Correctable single-bit error:

1	0	1	0	1	1	1
1	0	1	1	0	0	0
0	1	1	1	0	1	1
					0	0

Annotations for the second table:

- A blue vertical line highlights the first column, labeled "Parity error".
- A blue horizontal line highlights the fourth row, labeled "Parity error".

Cyclic Redundancy Check (CRC)

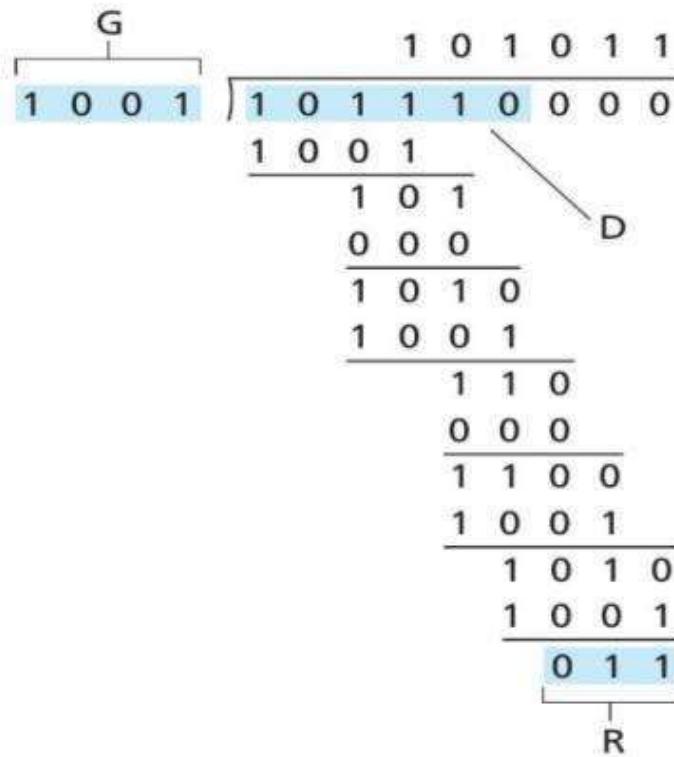


- Bit string can be viewed as a polynomial
- Sender and receiver **agree on** $r + 1$ bit pattern known as **generator G**
- Most significant bit of G should be **1**
- Given data D , sender will choose additional r bits, R and append them to D
- The resulting $d + r$ bit pattern should be **divisible** by G .
- CRC calculations are done in **modulo-2** arithmetic without carries and borrows (**XOR** operations)

- Find R such that there exists n that satisfies

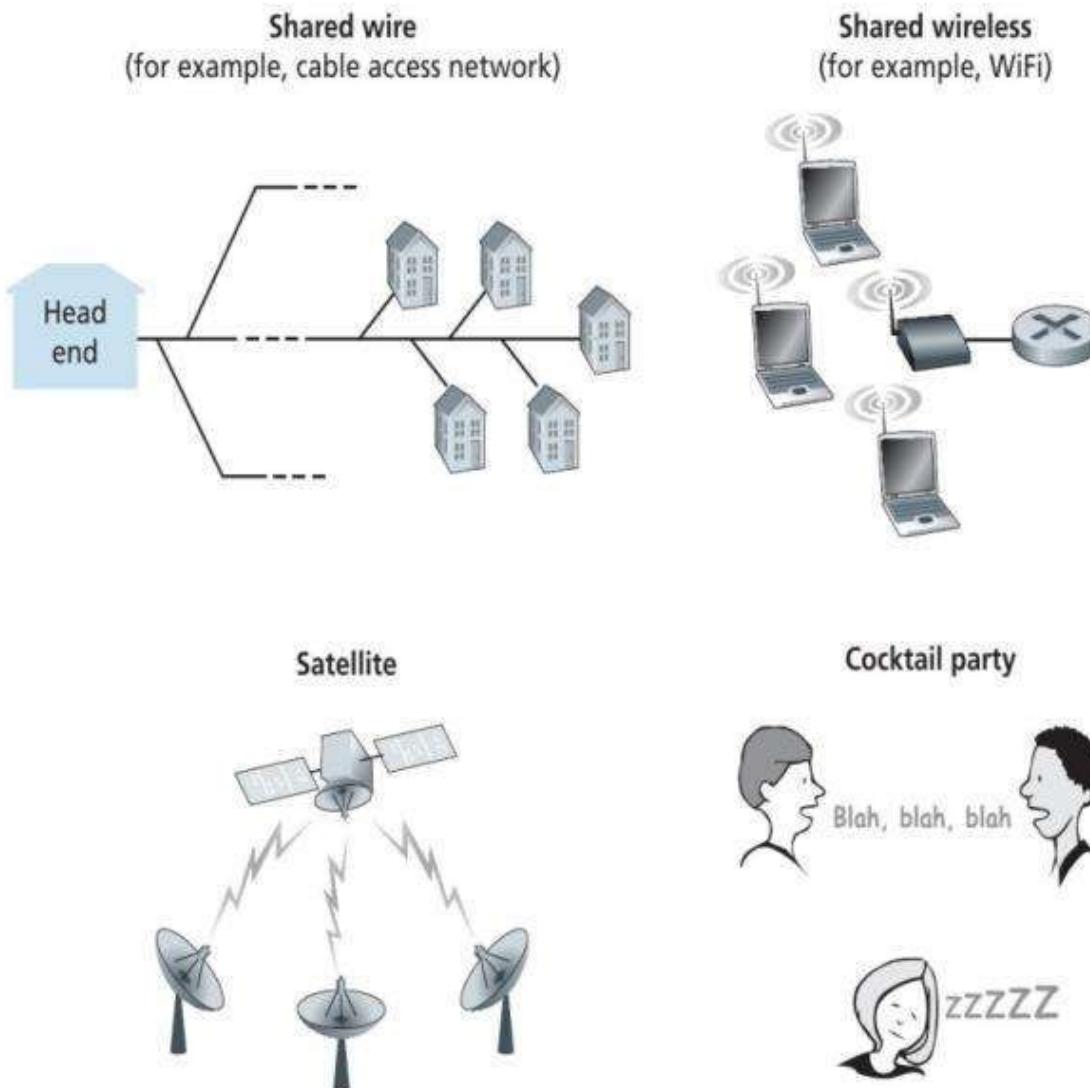
$$D \cdot 2^r \text{XOR } R = nG$$

- $R = \text{remainder } \frac{D \cdot 2^r}{G}$
- Example:



- International standards define 8-, 16-, 24-, 32-bit generators.

Multiple Access Channels



Multiple Access Protocols

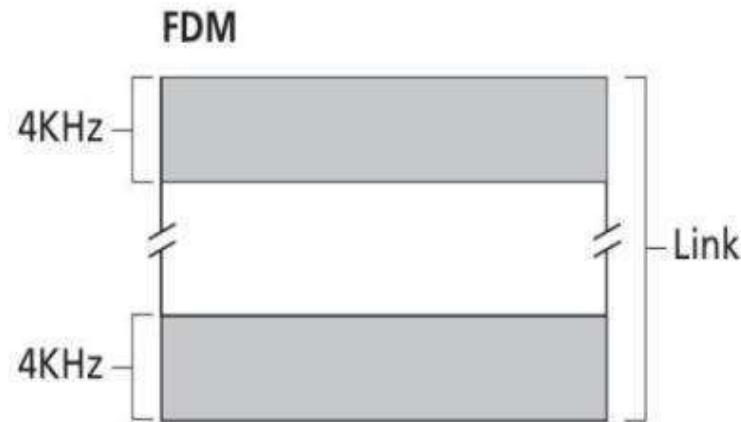
- If multiple nodes transmit frames at same time, packets **collide!**
- Channel partitioning protocols
 - TDM
 - FDM
- Random access protocols
 - Pure ALOHA
 - Slotted ALOHA
 - Carrier sense multiple access (CSMA), CSMA/CD
- Taking-turns Protocols
 - Polling protocol
 - Token-passing Protocol

Multiple Access Protocols

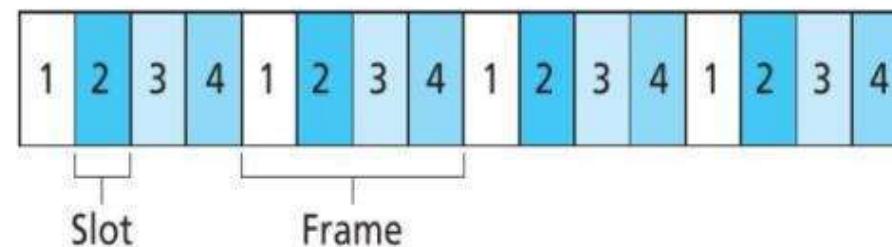
Desirable characteristics of MAC protocols on a broadcast channel of rate R bps:

- When only one node is has frames to send, that node should have throughput of R bps
- When M nodes have frames to send, each node should have throughput of R/M bps
- Protocol is decentralized
- Protocol is simple and inexpensive to implement.

Channel Partitioning Protocol



TDM



Key:



All slots labeled "2" are dedicated
to a specific sender-receiver pair.

Drawbacks of TDM and FDM

- When only one node is active, it gets throughput of R/N bps.
- Node has to wait for its turn!
- **Code division multiple access (CDMA)**

Slotted ALOHA

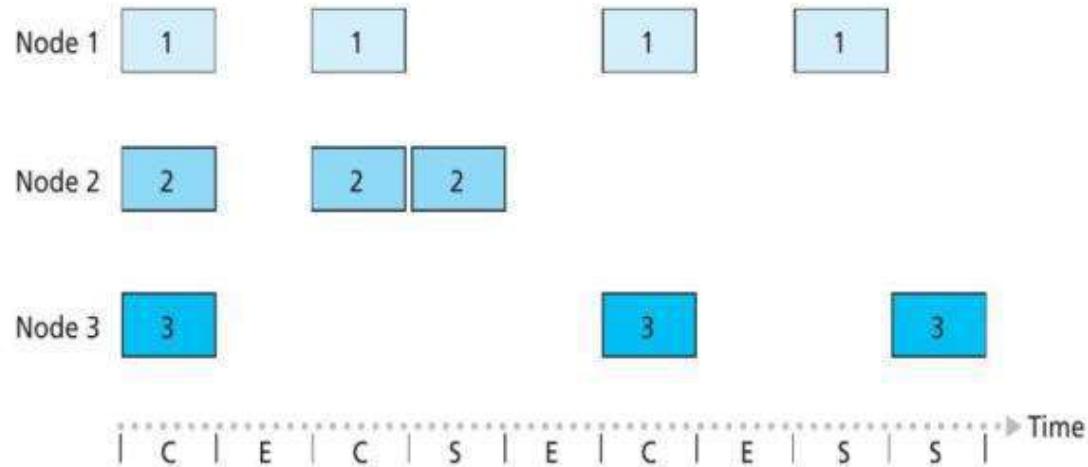
Model and assumptions

- All frame consists of exactly L bits
- Time is divided into slots of size L/R seconds
- Nodes start to transmit frames only at the beginning of slots
- Nodes are synchronized
- If two or more frames collide in a slot, then all nodes can detect the collision before the slot ends

Slotted ALOHA

- When a node has **fresh frame** to send, it waits for beginning of the next slot and transmits the frame in the slot
- If there is no collision, the node has successfully transmitted the packet and no need to retransmit
- If there is a collision, the node detects it before end of the slot. The node **retransmits the frame in each subsequent slot with probability p** until the frame is transmitted without a collision

Slotted ALOHA: Drawbacks



Key:

C = Collision slot

E = Empty slot

S = Successful slot

- Collisions
- Empty spaces
- Efficiency: fraction of successful slots

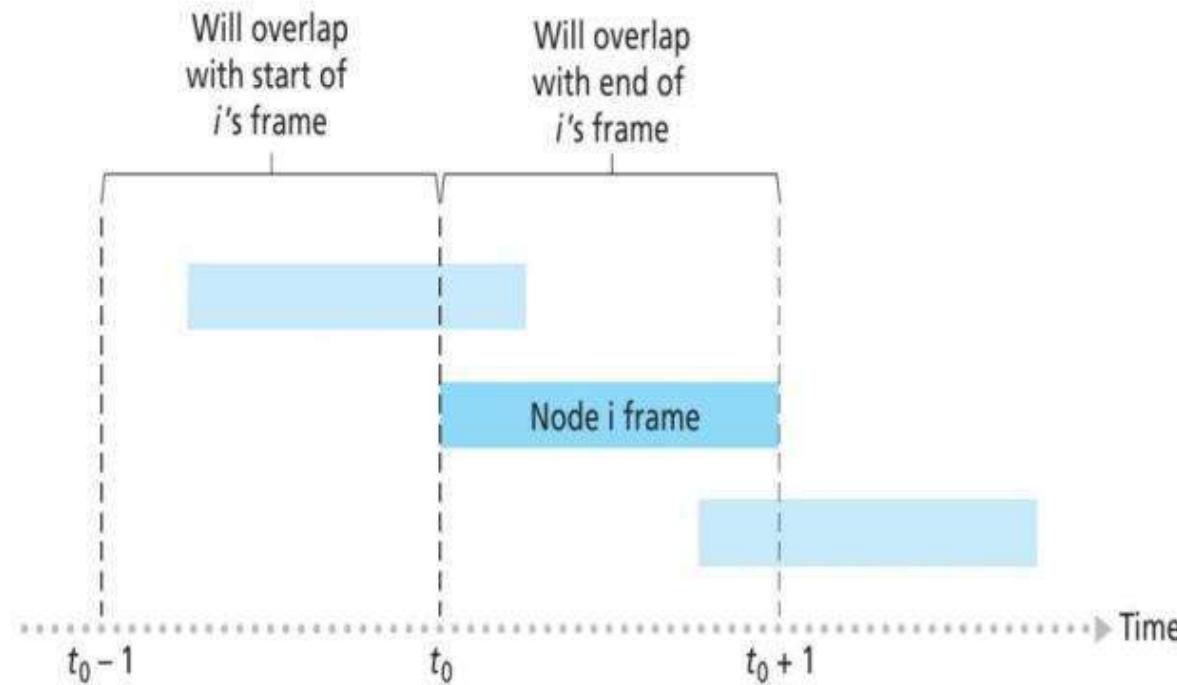
Efficiency of Slotted ALOHA

- Assume that each node always has a frame to send
- Probability that only one node (out of N) transmits
- $Np(1 - p)^{N-1}$
- Efficiency = $Np(1 - p)^{N-1}$
- Find p that maximizes efficiency, let it be p^*
- As $N \rightarrow \infty$, Efficiency $\rightarrow \frac{1}{e}$
- Only 37% of slots are used for successful transmission! a similar analysis show that 37% slots are empty and remaining slots have collisions.

Pure ALOHA

- **Unslotted** time axis
- Transmit a frame as soon as it arrives
- If there is a collision, node retransmits the frame **immediately** with probability p . Otherwise, wait for frame transmission time.
- After this wait, it then retransmits the frame with probability p or waits for another frame time with probability $1 - p$.

Pure ALOHA

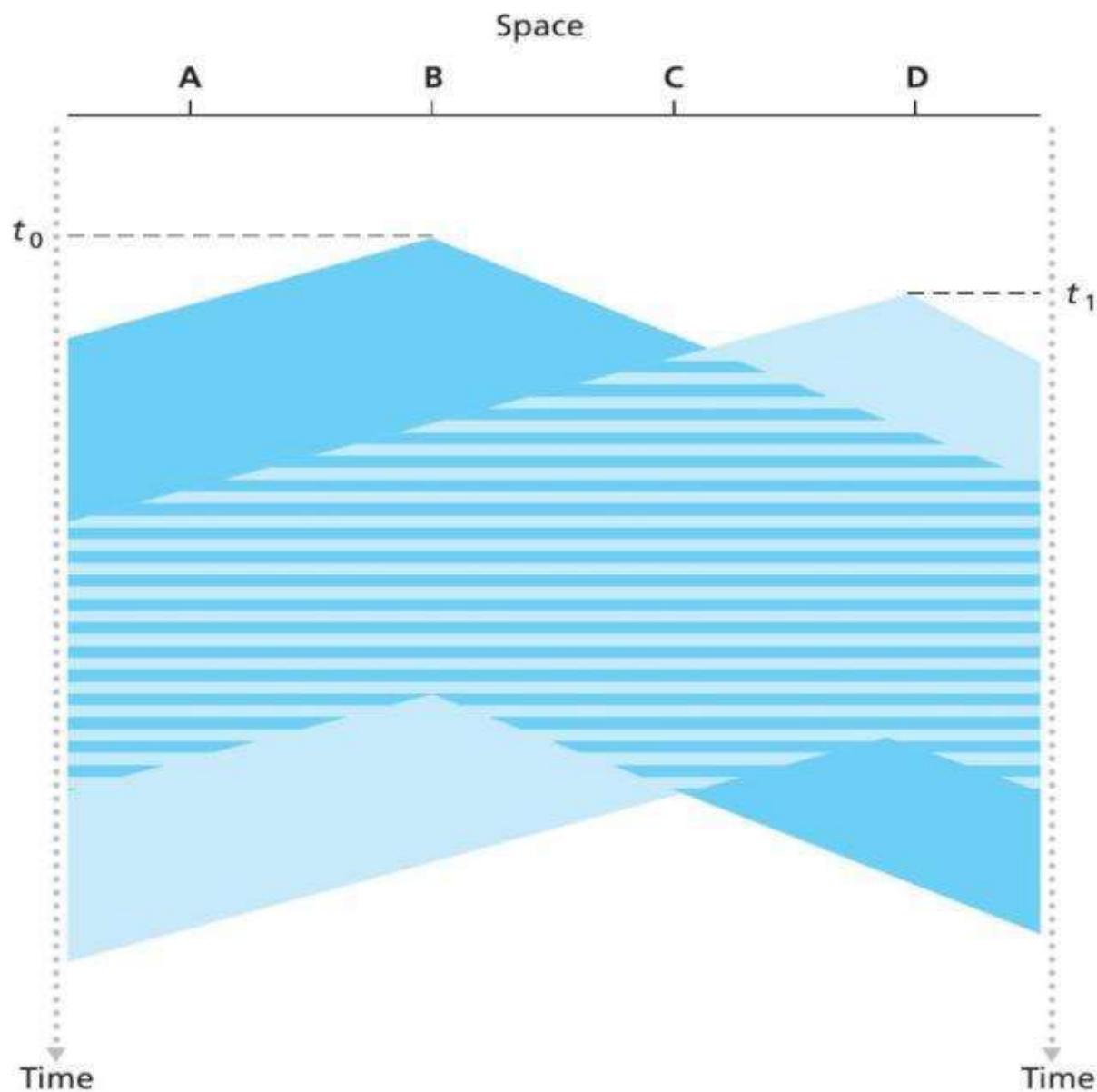


- Efficiency : $\frac{1}{2e}$

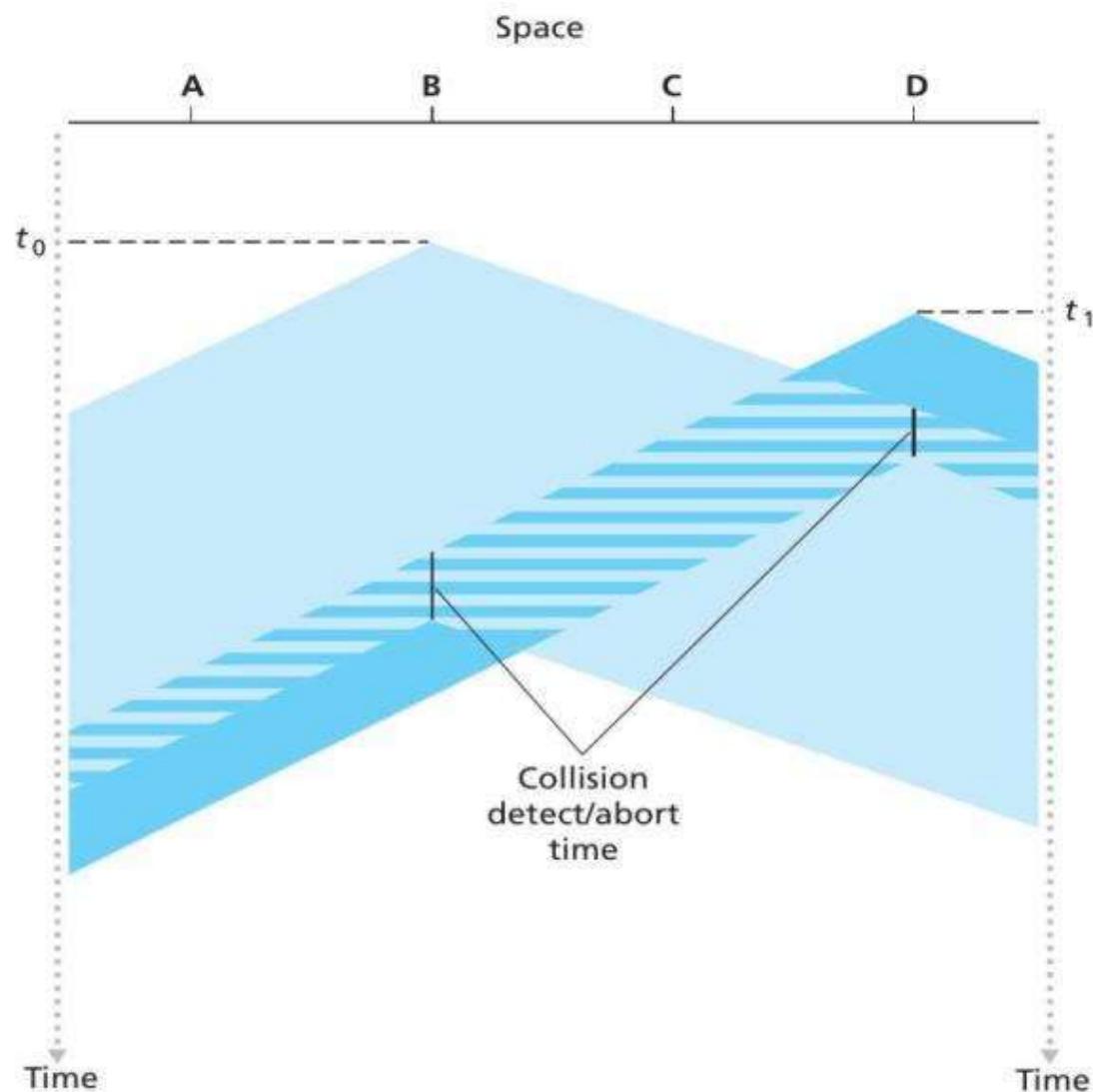
Carrier Sense Multiple Access

- Listen before speaking: carrier sensing
- If channel is busy, nodes 'backs off' a random amount of time and then senses again.
- If the channel is idle, node transmits the frame
- collision detection: If someone else begins talking at the same time, stop talking

CSMA



CSMA/CD



Taking-Turns Protocol

- Polling Protocol

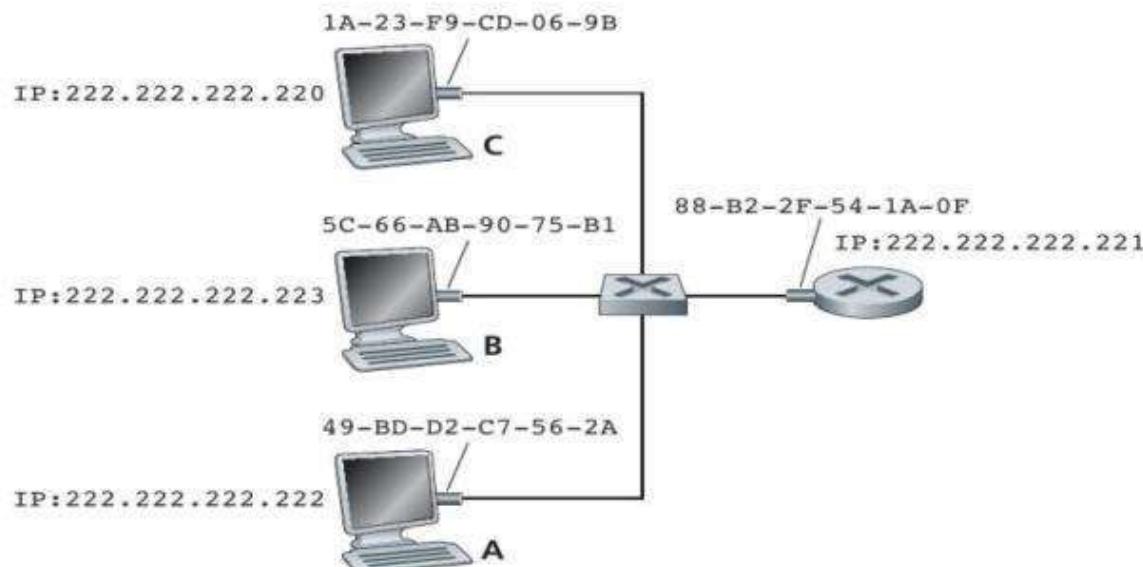
- Master node polls each of the nodes in a **round-robin** fashion
- Polling delay
- Master node may fail!

- Token-passing Protocol

- A special-purpose frame known as a **token** is exchanged among the nodes
- A node with token can transmit a maximum number of frames and send the token to next node
- A node holds token only if it has frames to transmit
- Very efficient.

Link-Layer Addressing

- Are IP addresses really unique?
- Node's adapter has a link-layer address
- Also known as **LAN address** or **Physical address** or **MAC address**
- MAC address
 - Managed by IEEE
 - Flat structure
 - Broadcast address : **FF-FF-FF-FF-FF-FF**



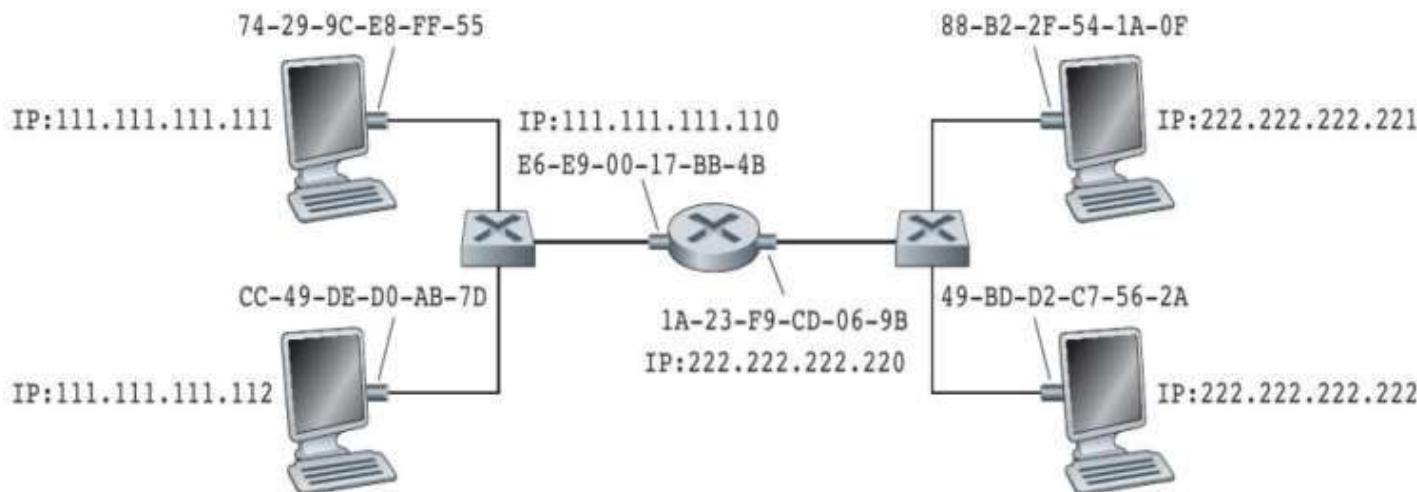
Address Resolution Protocol

- Sending node has to provide it's link layer not only IP address of destination but also **destination's MAC address**
- How does the source node determines the MAC address of it's destination?
- **Address Resolution Protocol (ARP)**
- Analogous to DNS

IP Address	MAC Address	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

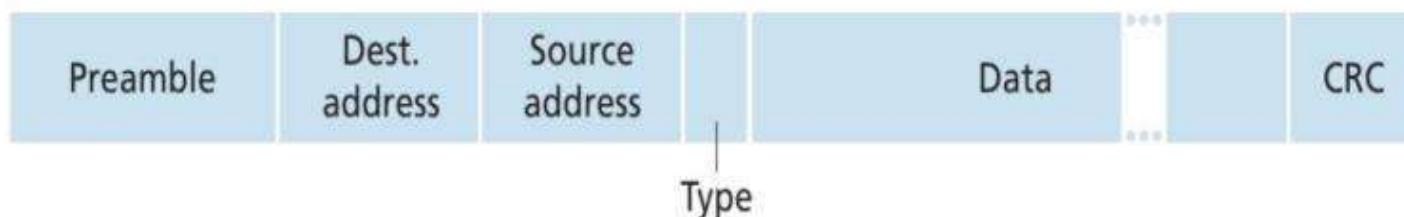
- Suppose node 222.222.222.220 wants to send a datagram to 222.222.222.222
- If ARP does not have an entry about the destination, it first constructs an ARP **query** packet.
- Query:
 - IP addresses of sender and receiver
 - sender's MAC address
 - Broadcast MAC address
- Encapsulated in a link-layer frame and sent in to the subnet
- Each node checks to see if its IP address matches with the destination address
- The one node with a match sends a ARP **response** packet with desired mapping

Sending a Datagram Off the Subnet



- Destination MAC address should be that of **router's MAC** on subnet 1
- The router determines the correct interface based on destination IP address
- After processing, router encapsulates the datagram in a frame with **destination MAC address**.

Ethernet



- Popular wired LAN technology
- Data field: **minimum length** is 46 bytes and **maximum length** is 1500 bytes
- Type field: specifies the protocol at network layer
- **Preamble:**
 - 8-bytes
 - first byte has value **10101010**
 - last byte has value **10101011**
 - to synchronize the clocks

Ethernet's MAC Protocol: CSMA/CD

- The adapter takes datagram from network layer and prepares Ethernet frame and keeps in adapter's buffer
- If the channel is idle for **96 bit times**, it starts to transmit the frame
- If the channel is busy, it waits until it senses no signal energy plus 96 bit times and then starts to transmit the frame
- If the adapter transmits the entire frame without detecting collision, the adapter is finished with the frame
- If the adapter detects a collision, it stops transmitting its frame and instead transmits a **48-bit jam signal**

Ethernet's MAC Protocol: CSMA/CD

- After transmitting the jam signal, the adapter enters an **exponential backoff** phase.
- Exponential Backoff: After experiencing n th collision in a row for a frame, the adapter chooses a value for K at random from $\{0, 1, 2, \dots, 2^m - 1\}$ for $m = \min(n, 10)$. The adapter then waits $K.512$ bit times and then returns to step 2.
- d_{prop} is maximum time it takes signal energy to propagate between any two adapters
- d_{trans} is the time to transmit a maximum size Ethernet frame.
- Efficiency = $\frac{1}{1+5d_{prop}/d_{trans}}$