

## CCN: Network Layer-IP Addressing

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Goals:

- ❖ IP Addressing
- ❖ Types of IP Addresses
  - ✓ Classful Addressing
  - ✓ Sub netting
  - ✓ CIDR—Classless Inter Domain Routing

## IP Addresses

## IP Addressing

- ❖ To send a packet from a source node to a destination node correctly through a network,
- ✓ The packet must contain enough information about the destination.
- ✓ The packet has to be retransmitted packet must contain enough information about the source address,
- ❖ The addressing scheme used for this purpose has considerable effect on routing.

---



---



---



---



---



---

## IP Addressing

- ❖ The addressing used in the IP layer
- ✓ To identify each device connected to the Internet is called the **Internet address or IP address**.
- ❖ An IPv4 address is a 32-bit address that
- ✓ Uniquely defines the connection of a host or a router to the Internet;
- ❖ The address space of IPv4 is  $2^{32}$  or 4,294,967,296.

---



---



---



---

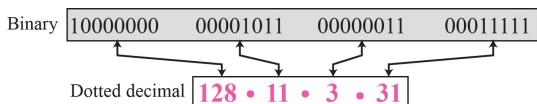


---



---

## IP Addressing : Dotted-Decimal Notation




---



---



---



---



---



---

## Types of IP Addresses

---

---

---

---

---

---

### Types of IP Addresses

- ❖ There are two types of IP addresses
  1. Classful Addressing.
  2. Classless Addressing

---

---

---

---

---

---

### Classful Addressing

---

---

---

---

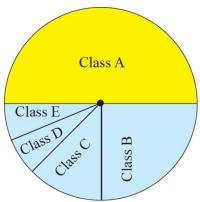
---

---

## Classful Addressing

- ❖ IP addresses started a few decades ago which uses the concept of classes.
- ❖ This architecture is called **classful addressing**.
- ❖ In classful addressing, the IP address space is divided into five classes: **A, B, C, D, and E**.
- ✓ Each class occupies some part of the whole address space

## Classes



Class A: $2^{31} = 2,147,483,648$ addresses, 50%
Class B: $2^{30} = 1,073,741,824$ addresses, 25%
Class C: $2^{29} = 536,870,912$ addresses, 12.5%
Class D: $2^{28} = 268,435,456$ addresses, 6.25%
Class E: $2^{28} = 268,435,456$ addresses, 6.25%

## Recognizing the Classes

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0.....			
Class B	10.....			
Class C	110....			
Class D	1110....			
Class E	1111....			

Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

## Recognizing the Classes

- ❖ We can find the class of an address when the address is given either in binary or dotted decimal notation.
- ❖ In the binary notation,
  - ✓ The first few bits can immediately tell us the class of the address;
- ❖ In the dotted-decimal notation,
  - ✓ The value of the first byte can give the class of an address

---



---



---



---

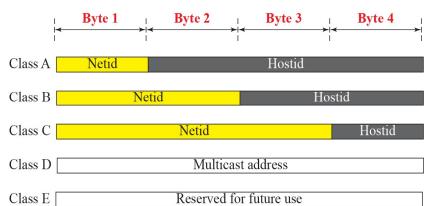


---



---

## Assignment Formats of Address: Netid and Hostid




---



---



---



---



---



---

## Class Address

- ❖ Each Class A format allows up to **126 networks** with **16 million hosts**.
- ❖ Each Class B format allows up to **16,382 networks** with up to **64 K hosts**.
- ❖ Each Class C format allows **2 million networks** with up to **254 hosts**.
- ❖ Each Class D is used for **multicasting** in which a **datagram is directed to multiple hosts**.
- ❖ Class E Addresses beginning with 11110 are reserved for future use.

---



---



---



---

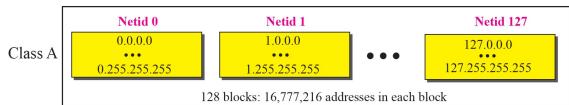


---



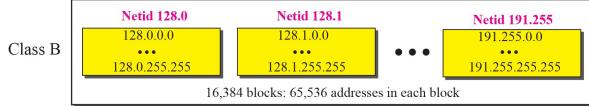
---

### Blocks in Class A



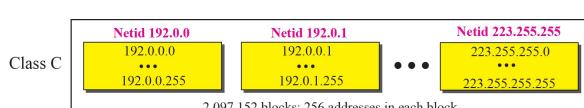
- ❖ Millions of class A addresses are wasted.

### Blocks in Class B



- ❖ Many class B addresses are wasted.

### Blocks in Class C



- ❖ Not so many organizations are so small to have a class C block.

### The Single Block in Class D

Class D

224.0.0.0     ...     239.255.255.255

One block: 268,435,456 addresses

- ❖ Class D addresses are made of one block, used for multicasting.

### The Single Block in Class E

Class E

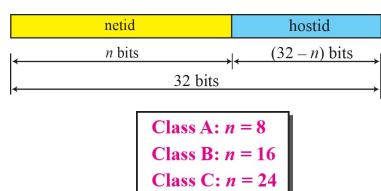
240.0.0.0     ...     255.255.255.255

One block: 268,435,456 addresses

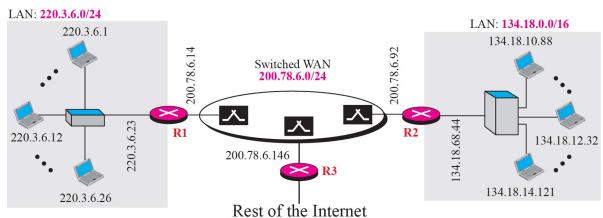
- ❖ The only block of class E addresses was reserved for future purposes.

### Classful Addressing

- ❖ The range of addresses allocated to an organization in classful addressing was a block of addresses in Class A, B, or C.



## Sample Internet



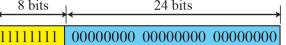
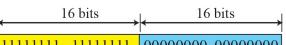
## Classful Addressing

## **Special IP addresses.**

## Network Mask

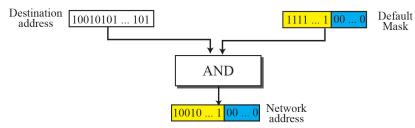
- ❖ Network Mask is mainly to extract the network address from the destination address of a packet by the router
  - ❖ A network mask or a default mask in classful addressing is a 32-bit number with **n leftmost bits all set to 1s** and **(32 - n) rightmost bits all set to 0s.**
    - ✓ Since n is different for each **class in classful addressing,**

## Network Mask

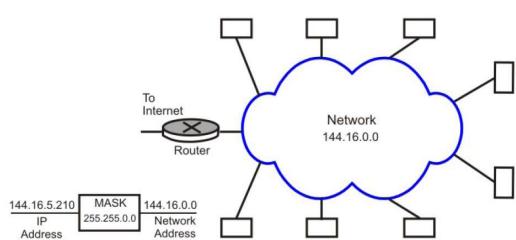
Mask for class A		255.0.0.0
Mask for class B		255.255.0.0
Mask for class C		255.255.255.0

## How to get Network Address using Network Mask

- To filter packets for a particular network, a router **filters out the net id part (by ANDing with all 1's)** by **removing the host id part (by ANDing with all 0's)**.



## Network Address using Network Mask



## Subnetting

---



---

---

---

---

---

---

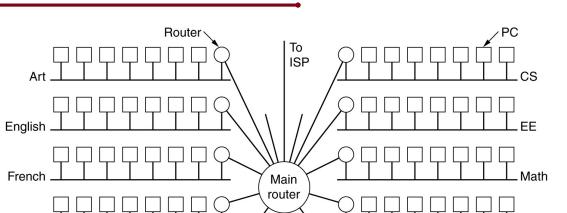
### Subnetting

---

- ❖ Usually all the hosts in a network must have the same network number.
  - ✓ This property of IP addressing causes **problem as the network grows.**
  - ❖ To overcome this problem, a concept known as **subnets** is used, which splits a network into several parts for internal use,
- 
- 
- 
- 
- 
- 

### Subnetting

---



**A campus network consisting of LANs for various departments.**

---

---

---

---

---

---

## Subnetting

- Even enough a **network** splits into several parts, but still **look like a single network** to the outside world.
- To facilitate **routing in subnet**, a concept known as **subnet mask is used**
- Subnetting reduces **router table space** by creating a **three-level hierarchy**; net id, subnet id followed by hosted.

---



---



---



---

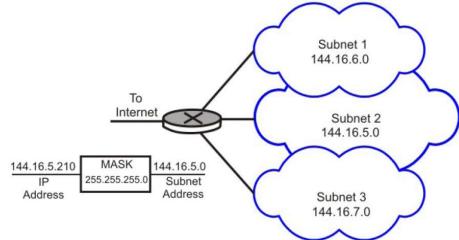


---



---

## Subnetting




---



---



---



---

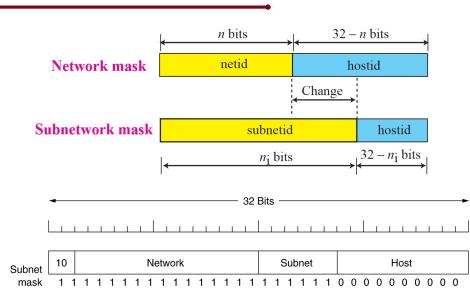


---



---

## Subnetting: Network mask and Subnet mask




---



---



---



---



---



---

## CIDR—Classless InterDomain Routing

---



---

---

---

---

---

---

## CIDR—Classless InterDomain Routing

---

- ❖ Classful addressing did not really solve the address space problem and made the distribution of addresses and the routing process more difficult.
- ❖ With the growth of the Internet,
  - ✓ it was clear that a larger address space was needed as a long-term solution.

---

---

---

---

---

---

## CIDR—Classless InterDomain Routing

---

- ❖ In 1996, the Internet authorities announced a new architecture called classless addressing.
- ❖ In classless addressing, variable-length blocks are used that belong to no classes.
- ❖ In classless addressing can have a **block of 1 address, 2 addresses, 4 addresses, 128 addresses**, and so on.

---

---

---

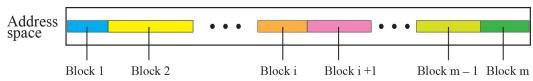
---

---

---

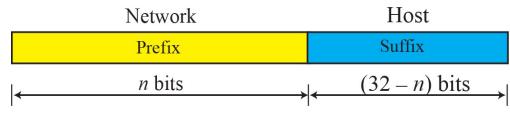
## Variable-Length Blocks

- ❖ In classless addressing, the **whole address space** is divided into **variable length blocks**.
- ✓ we can have a **block of  $2^0, 2^1, 2^2, \dots, 2^{32}$  addresses**



## Variable-Length Blocks : Prefix And Suffix

- ❖ In classless addressing, The **block** is actually divided into **two parts**, the **prefix** and the **suffix**. the **prefix defines the network** and the **suffix defines the host**.



## Variable-Length Blocks : Prefix And Suffix

- ✓ All **addresses** in the **block** have the **same prefix**; each address has a **different suffix**.
- ✓ The **length of the prefix, n**, depends on the **size of the block**; it can be **0, 1, 2, 3, ..., 32**.

## Slash Notation

- Classless addressing play a very important role, when we need to extract the information about the block from a given address in the block
  - The prefix length, n, is added to the address separated by a slash.
- The notation is informally referred to as **slash notation**

---



---



---



---

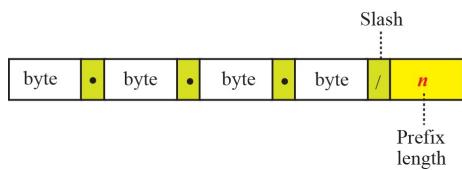


---



---

## Slash Notation




---



---



---



---



---



---

## Slash Notation: Example

- In the address **12.23.24.78/8**,
- The Network Mask Is **255.0.0.0**. The mask has **eight 1s and twenty-four 0s**. The prefix length is 8; the suffix length is 24.
- In the address **130.11.232.156/16**,
- the Network Mask is **255.255.0.0**. The mask has **sixteen 1s and sixteen 0s**. The prefix length is 16; the suffix length is 16.

---



---



---



---



---

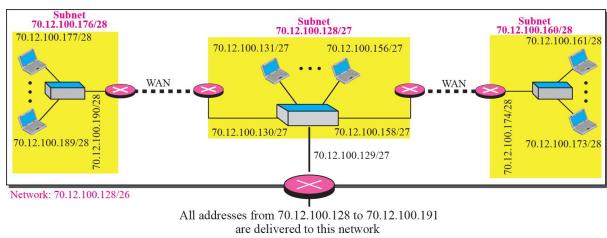


---

## Slash Notation: Example

- ❖ In the address **167.199.170.82/27**,
- ✓ The **network mask** is **255.255.255.224**. The mask has **twenty-seven 1s and five 0s**. The **prefix length** is **27**; the **suffix length** is **5**.

## Subnet in Classless Addressing



## Network Mask

- ❖ Find the Network Mask for the Following IP Address

**167.199.170.82/27.**

Address in binary: 10100111 11000111 10101010 01010010  
Network mask: 11111111 11111111 11111111 11100000

**Goals:**

- ❖ IP Addressing
- ❖ Types of IP Addresses
  - ✓ Classful Addressing
  - ✓ Sub netting
  - ✓ CIDR—Classless Inter Domain Routing

---

---

---

---

---

---

**Thank You**

---

---

---

---

---

---

## CCN: Network Layer- IP Protocol.

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Goals:

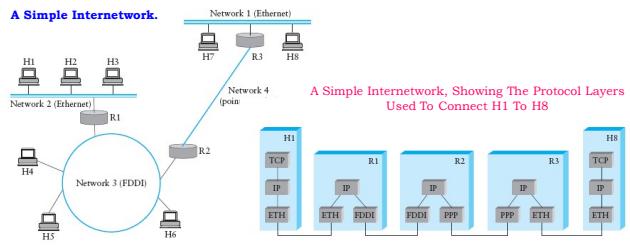
- ❖ Internetworking: Introduction
- ❖ The Network Layer in the Internet
  - ✓ IP Protocol
  - ✓ IP Header

## Internetworking

## Internetworking

- ❖ Until now, we have implicitly assumed that there is a single homogeneous network
  - ✓ Each machine using the same protocol in each layer.
- ❖ Internetwork or “internet” refer to an arbitrary collection of networks interconnected to provide some sort of host-to- host packet delivery service

## Internetworking



## How Networks Can Be Connected

## Internetworking Devices

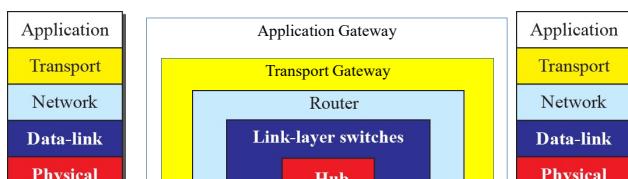
- ❖ Hosts and networks operates with coordination.
- ❖ Connecting devices makes a network or networks together to make an internet.
- ❖ Connecting devices can operate in different layers of the Network model.



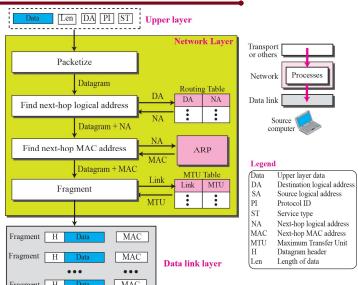
## Internetworking Devices

- ❖ IEEE802.1 identified the following possible internetworking scenarios.
- ✓ A single LAN
- ✓ Two LANs connected together (LAN-LAN)
- ✓ A LAN connected to a WAN (LAN-WAN)
- ✓ Two LANs connected through a WAN (LAN-WAN-LAN)

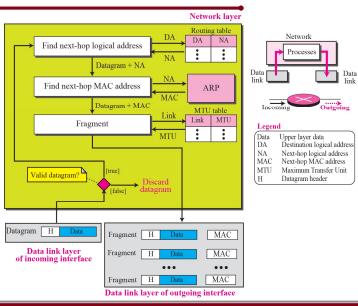
## Internetworking Devices



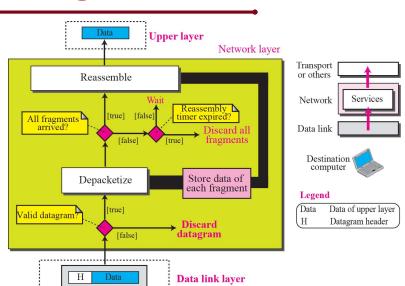
## Services provided at the Source Computer



## Processing at each router



## Processing At The Destination Computer



## IPv4 Protocol

---



---

---

---

---

---

---

### IP Protocol: Introduction

- ❖ The **Internet Protocol (IP)** is the **transmission mechanism** used by the TCP/IP protocols at the network layer.
- ❖ Several **protocols** are required to provide necessary **functionality** for **internetworking**.
- ❖ TCP/IP acts as a link of **different types of LAN and WAN** to provide Internet, a single integrated **network** for seamless communication.

---

---

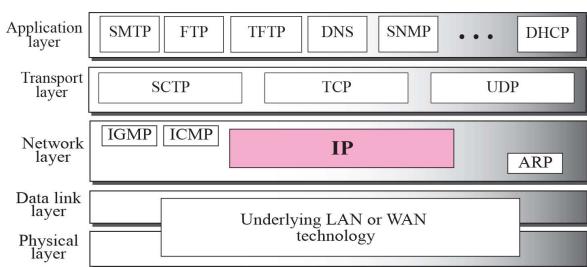
---

---

---

---

### IP Protocol: Introduction




---

---

---

---

---

---

## IP Protocol

- ❖ IP Protocol is an **unreliable and connectionless best-effort delivery service protocol.**
  - ✓ Best effort mean there is no error and flow control.
- ❖ IP Protocol performs **error detection and discards a packet**, if it is **corrupted**.

---



---



---



---



---



---

## IP Protocol

- ❖ To achieve reliability, it is necessary to **combine IP with a reliable protocol such as TCP.**
- ❖ Packets in the network (internet) layer are called **Datagrams.**

---



---



---



---



---



---

## IP Header

- ❖ A **datagram** is a **variable-length packet** consisting of two parts:
  - Header and Data.**
- ✓ The Header is **20 to 60 bytes in length**
- ✓ It contains **information essential to routing and delivery.**

---



---



---



---



---



---

## IP Header

- ❖ The IP header has a **number of fields** to provide:
 

<ul style="list-style-type: none"> <li>✓ Source And Destination IP Addresses</li> <li>✓ Non Transparent Fragmentation</li> <li>✓ Error Checking</li> <li>✓ Priority</li> <li>✓ Security</li> </ul>	<ul style="list-style-type: none"> <li>✓ Source Routing Option</li> <li>✓ Route Recording Option</li> <li>✓ Stream Identification</li> <li>✓ Time Stamping</li> </ul>
--	---

---



---



---



---



---

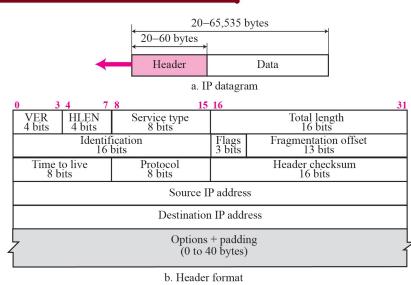


---



---

## IP Datagram with Header Format




---



---



---



---



---



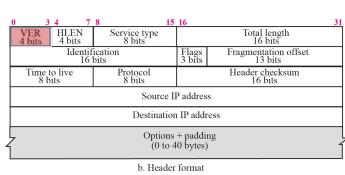
---



---

## IP Protocol : Version

- ❖ Version of the **IP protocol in use typically 4** which is usually called as **IPv4**
- ❖ Version Field contains **4-bits**




---



---



---



---



---



---

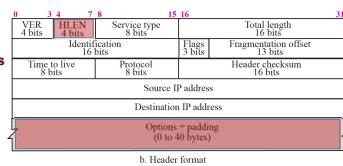


---

## IP Protocol : Header Length

- Length of the header expressed as the number of **32-bit words**.
- Minimum size is **5**, which applies when no options are present.
- Maximum size is **15** which limits the header to **60 bytes**, and thus the Options field to **40 bytes**.

Length of the header contains **4-bits**



b. Header format

\_\_\_\_\_

\_\_\_\_\_

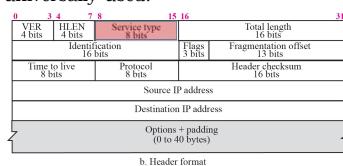
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## IP Protocol : Service Type

- Service Type contains **8-bits**
- Service Type allows **packet to be assigned a priority**. Router can use this field to route packets. Not universally used.



b. Header format

\_\_\_\_\_

\_\_\_\_\_

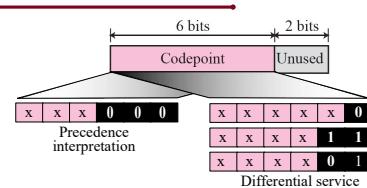
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## IP Protocol : Service Type

*The precedence subfield was designed, but never used in version 4.*



Values for codepoints

Category	Codepoint	Assigning Authority
1	XXXX0	Internet
2	XXXX1	Local
3	XXXX01	Temporary or experimental

\_\_\_\_\_

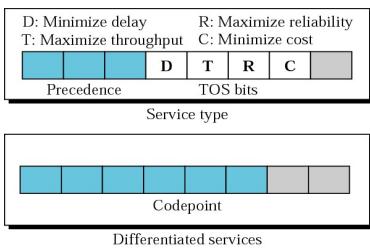
\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## IP Protocol : Differentiated services



## IP Protocol : Service Type

- ❖ Originally, the **6-bit field contained (from left to right)**,
  - ✓ A **three-bit Precedence field** and
  - ✓ The **Three flags, D, T, and R.**
- ❖ The Precedence field was a **priority**, from 0 (normal) to 7 (network control packet).
- ❖ The **three flag bits allowed the host** to specify what it cared most about from the set **(Delay, Throughput, Reliability)**.

## IP Protocol : Service Type

Default types of service

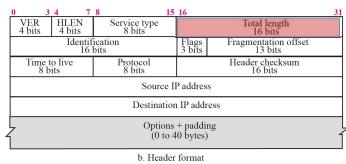
TOS Bits	Description
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Types of service

Protocol	TOS Bits	Description
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

## IP Protocol : Total Length

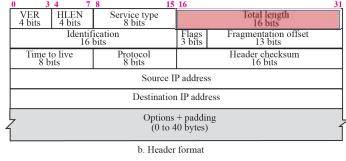
- ❖ Total Length contains 16-bits
- ❖ The total length field defines the **total length of the datagram including the header.**



## IP Protocol : Total Length

- ❖ Total Length contains 16-bits
- ❖ The total length field defines the **total length of the datagram including the header.**

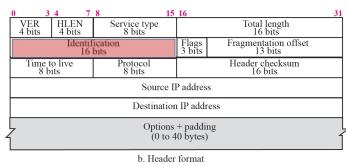
❖ Maximum datagram size is  
**( $2^{16}$ ) 65536 bytes.**



## IP Protocol : Identification

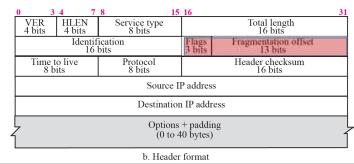
- ❖ The Identification field is needed to allow the **destination host to determine which datagram a newly arrived fragment belongs to** which source.

❖ All the fragments of a datagram contain the same **Identification value.**



## IP Protocol : Fragmentation

- ❖ A datagram can travel through different networks.
- ✓ The format and size of a frame depend on the protocol used by the physical network. A datagram may have to be fragmented to fit the protocol regulations.



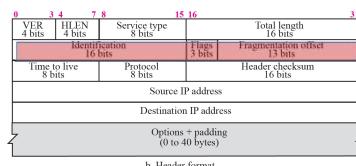
## IP Protocol : Fragmentation

- ❖ Divide the datagram to make it possible to pass through various networks. This is called **Fragmentation**.

## IP Protocol : Fragmentation

### ❖ Fields Related to Fragmentation

#### 1. Identification



#### 2. Flags

#### 3. Fragment Offset

## IP Protocol : Fragmentation

### ❖ Fields Related to Fragmentation

- 1. Identification:** Identifies a datagram originating from the source host. The combination of the **identification** and **source IP address** **must uniquely** define a datagram as it leaves the source host.

---



---



---



---



---



---

## IP Protocol : Fragmentation

### ❖ Fields Related to Fragmentation

- 2. Flags. :** This is a three-bit field.
- ✓ The **first bit** is **Reserved (Not Used)**.
  - ✓ The **second bit** is called the **do not fragment bit**.
  - ✓ The **third bit** is called the **More fragment**

---



---



---



---



---



---

## IP Protocol : Fragmentation: Flag

- ❖ The second bit is called the **Do Not Fragment Bit**.
- ❖ It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again
  - **If its value is 1**, the machine must not fragment the datagram.
  - **If its value is 0**, the datagram can be fragmented if necessary.

D: Do not fragment  
M: More fragments




---



---



---



---



---



---

## IP Protocol : Fragmentation: Flag

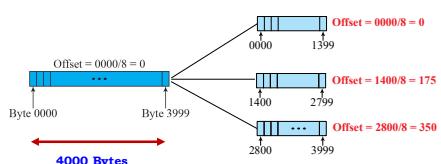
- ❖ The third bit is called the **more fragment bit**.
- ❖ It will give information to the **destination** all fragments of a datagram have arrived or not. Otherwise Node knows that other fragments are still arriving
  - If its value is **1**, it means the datagram is not the last fragment; there are more fragments after this one.
  - If its value is **0**, it means this is the **last or only fragment**

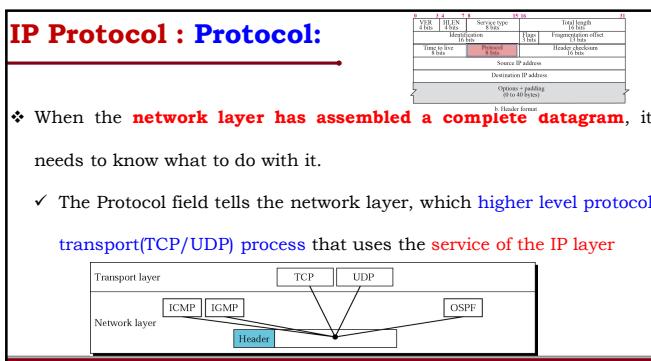
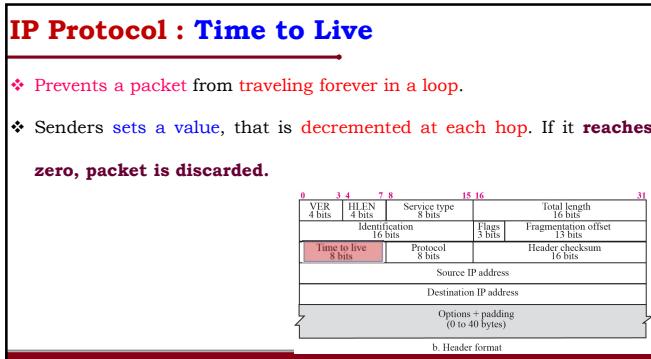
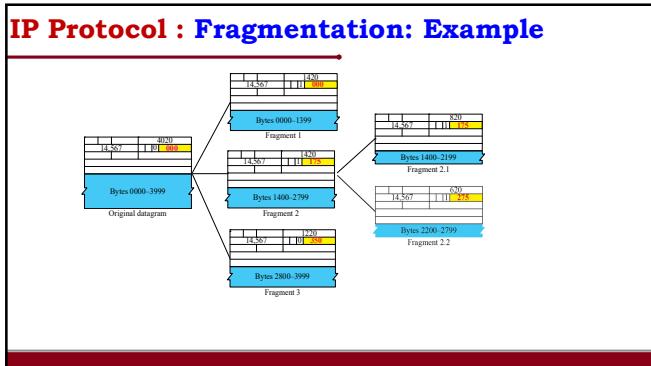
## IP Protocol : Fragmentation

- ❖ **Fields Related to Fragmentation**
- 3. **Fragment Offset:** Fragment offset tells where in the **current datagram this fragment belongs**
- ❖ Fragment offset is a **13-bit field** shows the relative position of this fragment with respect to the whole datagram.
- ❖ The **offset of the data** in the **original datagram** measured in **units of 8 bytes**.

## IP Protocol : Fragmentation: Example

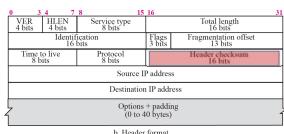
- ❖ All fragments in a **datagram** must be a multiple of 8 bytes except the



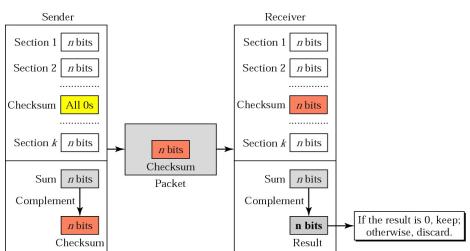


## IP Protocol : Checksum

- The **Error Detection method** used by most network protocols is called the **checksum**.
- The checksum **protects against the corruption** that may occur during the **transmission of a packet**.
- Checksum in IP covers only the header, not the data**



## IP Protocol : Checksum



## IP Protocol : Checksum: Example- Sender Side

5	0	28
1	0	0
4	17	0
10.12.14.5		
12.6.7.9		
4, 5, and 0	→ 01000101	00000000
28	→ 00000000	00011100
1	→ 00000000	00000001
0 and 0	→ 00000000	00000000
4 and 17	→ 00000100	00010001
0	→ 00000000	00000000
10.12	→ 00001010	00001100
14.5	→ 00001110	00000101
12.6	→ 00001100	00000110
7.9	→ 00000111	00001001
Sum	→ <b>01101000</b>	<b>01001110</b>
Checksum	→ <b>10001011</b>	<b>10110001</b>

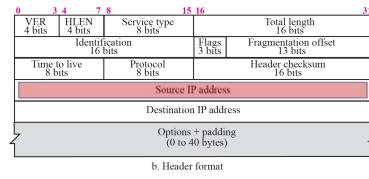
## IP Protocol : Checksum Calculation -Receiver

4	5	0	28
1	0	0	
4	17	35761	
		10.12.14.5	
		12.6.7.9	

4, 5, and 0 → 01000101 00000000  
 28 → 00000000 0001100  
 1 → 00000000 00010001  
 0 and 0 → 00000000 00000000  
 4 and 17 → 00000100 00010001  
 Checksum → 10001011 10110001  
 10.12 → 00001100 00001100  
 14.5 → 00001110 00000101  
 12.6 → 00001100 00000110  
 7.9 → 00000111 00001001  
 Sum → 1111 1111 1111 1111  
 Checksum → 0000 0000 0000 0000

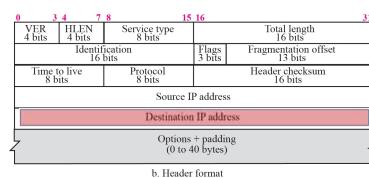
## IP Protocol : Source IP address

- Internet address of the sender which is **32-bit Address**



## IP Protocol : Destination IP address

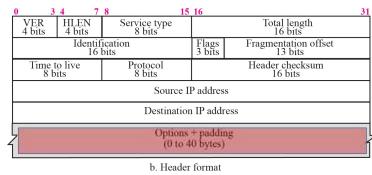
- Internet address of the destination which is **32-bit Address**



## IP Protocol : Options

- ❖ Options can be used to provide more functionality to the IP datagram

**datagram**



## IP Protocol : Options

- ❖ The header of the IP datagram is made of two parts:

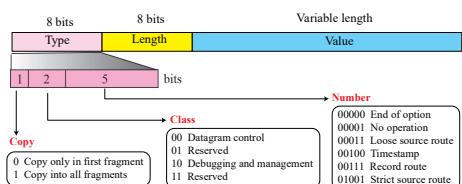
**Fixed part** : The fixed part is 20 bytes long

**Variable Part** : The variable part comprises the options, which can be a maximum of 40 bytes.

## IP Protocol : Options

- ❖ Options, as the name implies, are not required for a datagram.
- ✓ Used for Network Testing and Debugging.

### IP Protocol : Options Format



### IPv6 Protocol

### The Success Of IPv4 Protocol

- ❖ The current version of IP has been **extremely successful**
- ❖ The success of the current version of IP is incredible
- ✓ The protocol has accommodated changes in **hardware technologies, heterogeneous networks, and extremely large scale**

## Why IPv6 is Required

- ❖ If IP works so well, why change?
  - ✓ When IP was defined, **only a few computer networks existed.**
  - ✓ The designers decided to use **32 bits for an IP address**, which allows to include **over a million networks.**
- ❖ The global **Internet is growing exponentially**, with the **size doubling in less than a year.**

---



---



---



---



---



---

## Need of IPv6 Protocol

- ❖ The primary motivation for defining a new version of IP to ascend from the **address space limitation**
  - ✓ **Larger addresses** are necessary to accommodate continued growth of the Internet.
- ❖ IP is **central to all Internet communication**, changing **IP requires a change to the entire Internet.**

---



---



---



---



---



---

## A Name And A Version Number

- ❖ The **new version of IP received 6** as its official version number, and the protocol became known as **IPv6**.
- ❖ To distinguish **IPv4 from IPv6**,
  - ✓ The **current version of IP became known as IPv4**

---



---



---



---



---



---

### IPv6 Changes from IPv4

- ❖ IPv6 uses **larger addresses**
- ❖ Entirely **new datagram header format**.
- ❖ The **IPv6 header is always a variable size**, While IPv4 uses fixed-length headers for storing key information

---



---



---



---



---



---



---

### IPv6 retention Features

- ❖ **IPv6 retains** many of the **design features that have made IPv4** so successful.
- ✓ Like IPv4, **IPv6 is connectionless** — each datagram contains a destination address, and each datagram is routed independently.

---



---



---



---



---



---



---

### IPv6 Features

- ❖ The **new features in IPv6** can be grouped into **five broad categories**:

  1. **Address Size.**
  2. **Header Format.**
  3. **Extension Headers.**
  4. **Support For Real-Time Traffic.**
  5. **Extensible Protocol**

---



---



---



---



---



---



---

### Address Size.

- ❖ Each IPv6 address contains **128 bits**.
- ❖ The resulting **address space** is **large enough** to accommodate continued **growth** of the world-wide Internet for many decades.

---



---



---



---



---



---

### Header Format.

- ❖ The **IPv6 datagram header** is completely different than the IPv4 header.
- ❖ Almost **every field** in the header has been **changed**; some have been replaced.

---



---



---



---



---



---

### Extension Headers.

- ❖ Unlike IPv4, which uses a **single header format** for all datagrams, IPv6 encodes information into **separate headers**.
- ❖ A **datagram** consists of the **base IPv6 header** followed by **zero or more extension headers**, followed by **data**.

---



---



---



---



---



---

### Support For Real-Time Traffic

- ❖ IPv6 includes a mechanism that allows a sender and receiver to establish a **high-quality path** through the underlying network and to associate **datagrams** with that path.

---

---

---

---

---

---

### Extensible Protocol.

- ❖ The **extension scheme** makes **IPv6 more flexible** than **IPv4**, and means that **new features** can be added to the design as needed.

---

---

---

---

---

---

### IPv6 Datagram Format

---

---

---

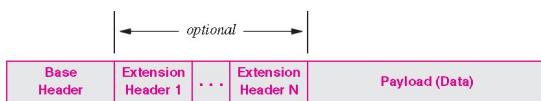
---

---

---

## IPv6 Datagram Format

- An IPv6 datagram contains a **series of headers**.
  - Each **datagram** begins with a **base header**, which is followed by **zero or more extension headers** followed by the **payload**.

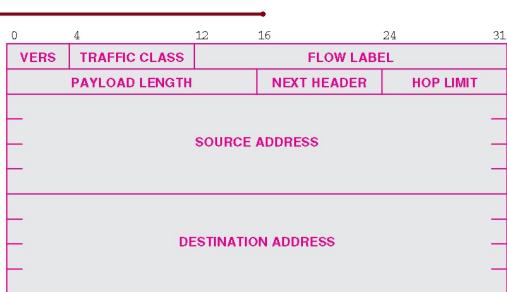


The general form of an IPv6 datagram.

## IPv6 Datagram Format : Observation

- Some extension headers are **larger than** the base header, and
- others extension headers are **smaller than** the base header.
- In many datagrams, the **size of the payload** is much **larger than** the size of the headers.

## IPv6 Base Header Format



### IPv6 Base Header Format

- ❖ Although, IPv6 is **twice as large** as an IPv4 header, the IPv6 base header contains **less information**.
- ❖ **Most of the space in the header** is devoted to the **SOURCE ADDRESS** and **DESTINATION ADDRESS** fields,
- ✓ Each of which occupies **sixteen octets**, four times more than an IPv4 address.

---



---



---



---



---



---

### IPv6 Base Header Format

- ❖ The base header contains **SIX fields**.
- 1. The **VERS field** identifies the **protocol as version 6**.
- 2. The **TRAFFIC CLASS field** specifies the traffic class using a definition of **traffic types known as differentiated services** to specify general characteristics that the datagram needs.
- 3. The **PAYLOAD LENGTH field** corresponds to **IPv4's datagram length field**.

---



---



---



---



---



---

### IPv6 Base Header Format

- 3. The **PAYLOAD LENGTH** specifies only the **size of the data being carried** (i.e., the payload); the **size of the header is excluded**.
- 4. The **HOP LIMIT** corresponds to the **IPv4 TIME-TO-LIVE field**.
  - ✓ IPv6 interprets the **HOP LIMIT strictly**
  - ✓ The datagram will be discarded if the **HOP LIMIT counts down to zero** before the datagram arrives at its destination.

---



---



---



---



---

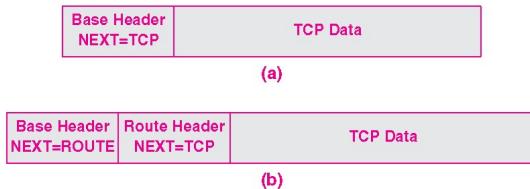


---

### IPv6 Base Header Format

5. Field **FLOW LABEL** was originally intended to associate a **datagram** with a particular **underlying network path**.
6. The **NEXT HEADER** field is used to specify the **type of information** that follows the **current header**.
- ✓ if the **datagram includes an extension header**, the **NEXT HEADER** field specifies the type of the extension header.
  - ✓ If **no extension header exists**, the **NEXT HEADER** field specifies the type of data being carried in the payload.

### IPv6 Base Header Format : Next Header



**Thank You**

## CCN: Network Layer- Internet Control Message

---

**Dr. E.SURESH BABU**

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



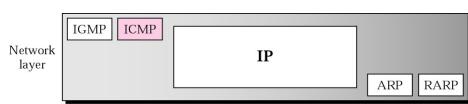
## Goals:

---

- ❖ Internet Control Message
  - ✓ ICMP
  - ✓ ARP and RARP
  - ✓ BOOTP and DHCP

## Internet Control Protocols

---



## Internet Control Protocols

- ❖ The Internet has **several control protocols** used in the **network layer** along with IP
- 1. Internet Control Message Protocol (ICMP)**
- 2. Address Resolution Protocol(ARP)**
- 3. Reverse Address Resolution Protocol (RARP)**
- 4. DHCP (Dynamic Host Configuration Protocol).**

---



---



---



---



---



---

## Internet Control Message Protocol (ICMP)

---



---



---



---



---



---

## Internet Control Message Protocol (ICMP)

- ❖ To make **efficient use of the network resources**,
- ✓ IP was designed to provide **unreliable and connectionless best-effort datagram delivery service.**
- ✓ However, IP **does not handle**
  - **Error-control mechanism**
  - **Lacks mechanism for host and management queries.**

---



---



---



---



---



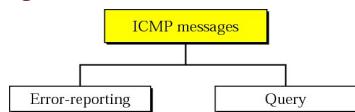
---

## Internet Control Message Protocol (ICMP)

- ❖ A **Protocol** known as Internet Control Message Protocol (ICMP) has been designed to **compensate the above two deficiencies**.
- ❖ ICMP messages can be broadly divided into **two broad categories**:

### 1. Error Reporting Messages

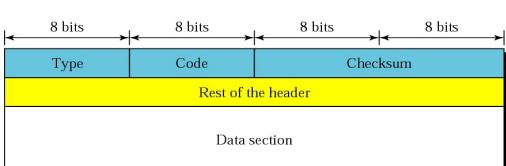
### 2. Query Messages.



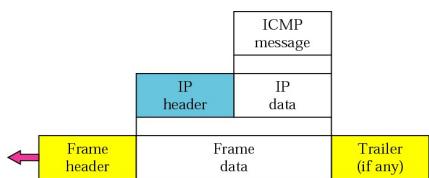
## Internet Control Message Protocol (ICMP)

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

## General format of ICMP messages

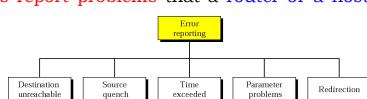


## ICMP Encapsulation



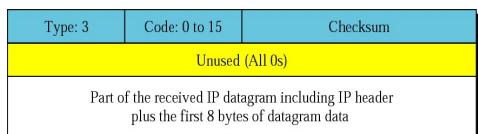
## Error Reporting Messages

- The error-reporting messages report problems that a router or a host (destination) may encounter

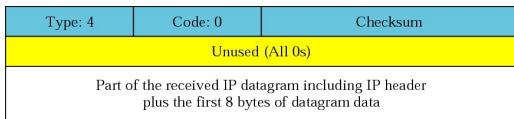


Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection

## Destination-Unreachable Format

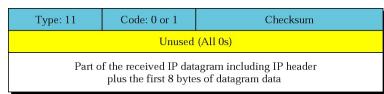


### Source-Quench format (Choke Packets)



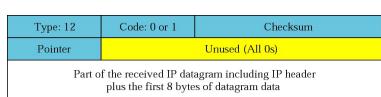
- ❖ A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.
- ❖ The source must slow down the sending of datagrams until the congestion is relieved.

### Time-Exceeded message format



- ❖ The TIME EXCEEDED message is sent when a packet is dropped because its counter has reached zero.

### Parameter-Problem message format

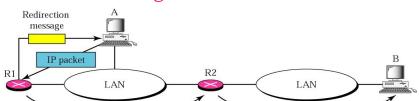


- ❖ The PARAMETER PROBLEM message indicates that an illegal value has been detected in a header field.

## Parameter-Problem message format

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

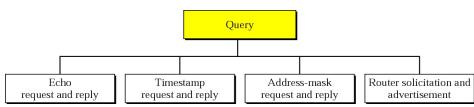
- The **REDIRECT message** is used when a **router notices that a packet seems to be routed wrong.**



## Query Messages.

- In this type of ICMP message, a **node sends a message that is answered in a specific format by the destination node.**

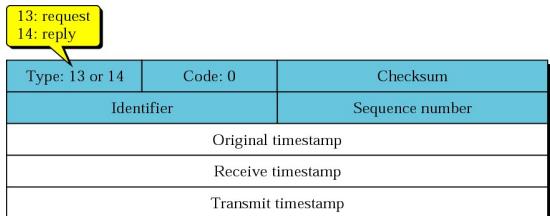
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement



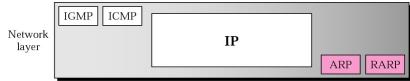
## Echo-request and Echo-reply messages

8: Echo request 0: Echo reply	Type: 8 or 0	Code: 0	Checksum
Identifier			Sequence number
Optional data Sent by the request message; repeated by the reply message			

### **Timestamp-Request And Timestamp-Reply Message Format**



### **Address Resolution Protocol(ARP)**



### **Address Resolution Protocol(ARP)**

- ❖ It may be noted that the knowledge of hosts' IP address is not sufficient for sending packets,
- ✓ Data link hardware does not understand internet addresses.

## Address Resolution Protocol(ARP)

- ❖ In an **Ethernet** network,
  - ✓ The **Ethernet controller card** can send and receive using **48-bit Ethernet addresses**.
  - ✓ The **32-bit IP addresses** are unknown to these cards.
- ❖ Requires a **mapping of the IP addresses to the corresponding Ethernet addresses**.

---



---



---



---



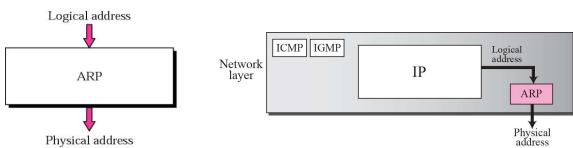
---



---

## Address Resolution Protocol(ARP)

- ❖ Address Resolution Protocol (ARP) is a mapping technique that **maps IP addresses to the corresponding Ethernet addresses**.




---



---



---



---



---



---

## Address Resolution Protocol(ARP)

- ❖ ARP is a dynamic mapping approach for **finding a physical address for a known IP address**.
- ❖ In this **elegant approach**
  - ✓ ARP broadcast packet onto the Ethernet asking "**who owns the destination IP address?**".
  - ✓ The **destination node** responds with its **Ethernet address** after hearing the request.

---



---



---



---



---

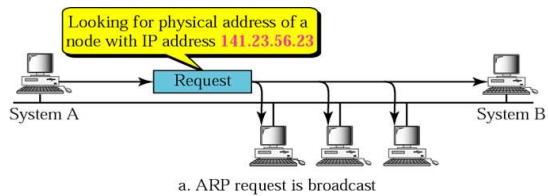


---

## Address Resolution Protocol(ARP)

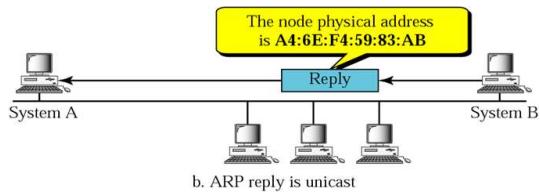
- ❖ ARP involves following two basic steps
  - An ARP request is broadcast to all stations in the network
  - An ARP reply is an unicast to the host requesting the mapping

### ARP request is broadcast to all stations in the network



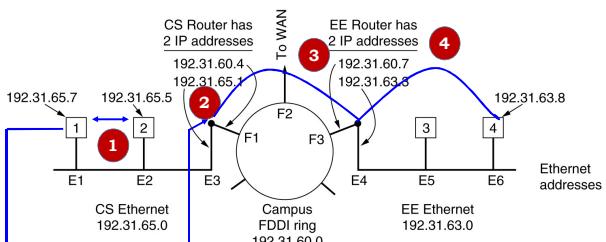
a. ARP request is broadcast

### An ARP reply is an unicast to the host requesting the mapping

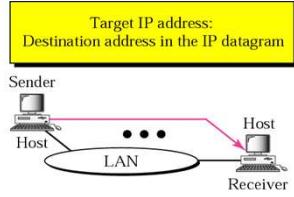


b. ARP reply is unicast

### Four Cases using ARP

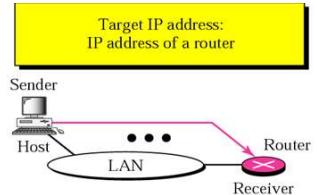


### First Case using ARP



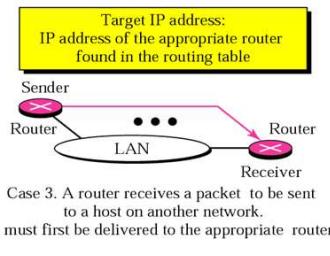
Case 1. A host has a packet to send to another host on the same network.

### Second Case using ARP



Case 2. A host wants to send a packet to another host on another network.  
It must first be delivered to a router.

### Third Case using ARP




---

---

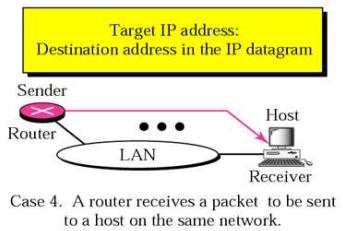
---

---

---

---

### Fourth Case using ARP




---

---

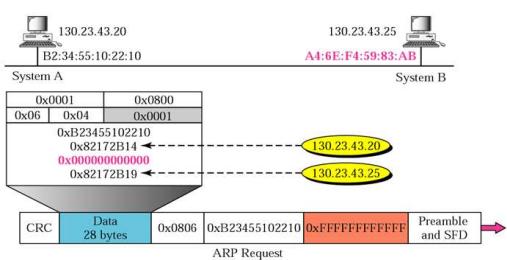
---

---

---

---

### ARP Example : Request




---

---

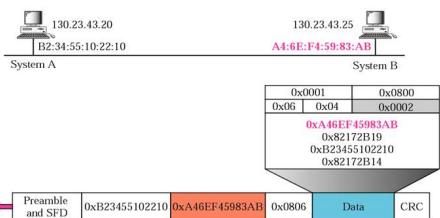
---

---

---

---

### ARP Example : Reply



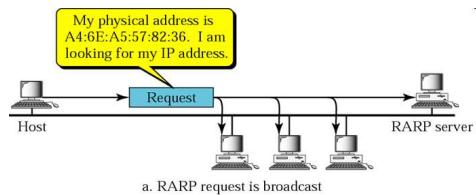
### Reverse Address Resolution Protocol (RARP)



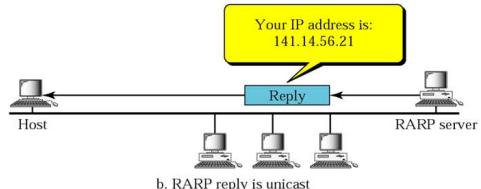
### Reverse Address Resolution Protocol(RARP)

- ❖ RARP finds the **logical address for a machine that only knows its physical address.**

### RARP Operation : Request



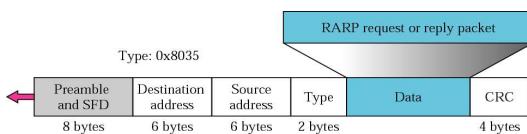
### RARP Operation



### RARP Packet

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

### Encapsulation of RARP packet



### Limitations of RARP

- ❖ A disadvantage of RARP is that it uses a destination address of all 1s (limited broadcasting) to reach the RARP server.
- ✓ Such broadcasts are not forwarded by routers,
- ❖ Therefore, RARP server is needed on each network.

### DHCP (Dynamic Host Configuration Protocol).

### DHCP (Dynamic Host Configuration Protocol).

- ❖ DHCP allows both **manual IP address assignment** and **automatic assignment**.

---



---



---



---



---



---

### DHCP (Dynamic Host Configuration Protocol).

- ❖ DHCP is based on the idea of a **special server that assigns IP addresses** to hosts asking for one.
- ✓ This server need not be on the **same LAN as the requesting host**.  
Since
- ❖ The **DHCP server** may not be **reachable by broadcasting**, a **DHCP relay agent** is needed on each LAN,

---



---



---



---

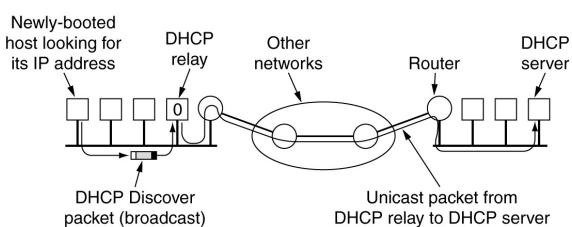


---



---

### DHCP (Dynamic Host Configuration Protocol).




---



---



---



---



---



---

### DHCP (Dynamic Host Configuration Protocol).

- ❖ To find its IP address,
  - ✓ A newly-booted machine broadcasts a DHCP DISCOVER packet.
  - ✓ The DHCP relay agent on its LAN intercepts all DHCP broadcasts.
    - When it finds a DHCP DISCOVER packet, it sends the packet as a unicast packet to the DHCP server, possibly on a distant network.
    - The only piece of information the relay agent needs is the IP address of the DHCP server.

---



---



---



---



---



---

### Problem with Automatic assignment of IP addresses

- ❖ An issue that arises with automatic assignment of IP addresses from a pool
  - ✓ how long an IP address should be allocated.
  - ✓ If a host leaves the network and does not return its IP address to the DHCP server,
    - IP address will be permanently lost.
    - After a period of time, many addresses may be lost.

---



---



---



---



---



---

### Leasing Concept

- ❖ A technique called leasing is used to overcome the problem
  - ✓ IP address assignment may be for a fixed period of time,
  - ✓ Just before the lease expires, the host must ask the DHCP server for a renewal.
  - ✓ If it fails to make a request or the request is denied, the host may no longer use the IP address it was given earlier.

---



---



---



---



---



---

**Goals:**

- ❖ Internet Control Message
  - ✓ ICMP
  - ✓ ARP and RARP
  - ✓ BOOTP and DHCP

---

---

---

---

---

---

**Thank You**

---

---

---

---

---

---

## CCN: Network Layer- Routing Algorithm

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Goals:

- ❖ What is Routing
- ❖ Desirable Properties of a Router
- ❖ The Optimality Principle
- ❖ Routing Algorithm Metrics

### What is Routing

## What is Routing

- ❖ A famous quotation

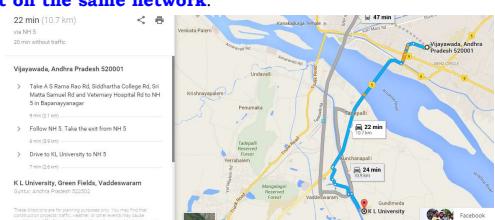
**"A name indicates what we seek.  
An address indicates where it is.  
A route indicates how we get there."**

-- Jon Postel



# What is Routing

- ❖ Routing is an issue needed between source and destination that are **not on the same network**.



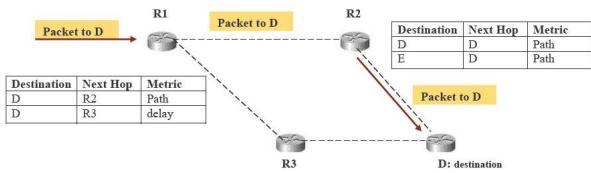
## What is Routing

- ❖ Routing is the act of moving information across an inter-network from a source to a destination.
  - ❖ Routing is referred as the process of choosing a path which is used to send the packets.
  - ❖ The main function of the network layer is routing the packets from the source machine to the destination machine.

## Why Routing Algorithm are Needed

- ❖ The routing algorithm is the part of the network layer software responsible
  - ✓ Deciding which output line an incoming packet should be transmitted on,
  - ✓ What should be the next intermediate node for the packet.
- ❖ The Routing algorithms uses proper data structures for choosing the routes between source and destination which is major area of network design

## Why Routing Algorithm are Needed



## Why Routing Algorithm are Needed

- ❖ Routing Algorithms initialize and maintain routing tables, which contain route information.
- ✓ Route information varies depending on the routing algorithm used.
- ❖ Routing algorithms fill routing tables with a variety of information.
- ❖ When a router receives an incoming packet, it checks the destination address

## The Optimality Principle

---



---

---

---

---

---

---

### The Optimality Principle

- ❖ Optimality Principle provides optimal routes without regard to network topology or traffic.
- ❖ To achieve Optimality Principle
  - ✓ if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

---

---

---

---

---

---

### The Optimality Principle: Sink Tree

- ❖ In optimality principle,
- ✓ The set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a sink tree.
- ✓ Note that a sink tree is not necessarily unique;
- ✓ The goal of all routing algorithms is to discover and use the sink trees for all routers.

---

---

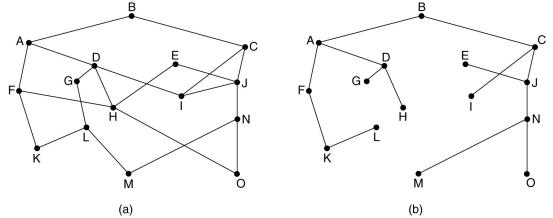
---

---

---

---

### The Optimality Principle: Sink Tree



### Routing Algorithm Metrics

### Routing Algorithm Metrics

- ❖ To determine the **best route** and evaluate the performance of routing algorithms, All the following **different metrics** have been used:

- ✓ **Path Length**
- ✓ **Delay**
- ✓ **Bandwidth**
- ✓ **Load**
- ✓ **Reliability**

## Path Length

- ❖ Path length is the most common routing metric.
- ✓ Some routing protocols allow network administrators to assign arbitrary costs to each network link.
- ❖ Path length is the sum of the costs associated with each link traversed.
- ❖ Some other routing protocols define hop count,
- ✓ A metric that specifies the number of passes through routers, that a packet must pass through in a route from a source to a destination.

---



---



---



---



---



---

## Routing Delay

- ❖ Routing delay refers to the length of time required to move a packet from source to destination through the internet.
- ❖ Delay depends on many factors,
- ✓ Bandwidth of intermediate network links,
- ✓ The port queues (receive and transmit queues) at each router,
- ✓ Network congestion on all intermediate network links

---



---



---



---



---



---

## Bandwidth

- ❖ Bandwidth refers to the available traffic capacity of a link.
- ✓ A 10-Mbps Ethernet link would be preferable to a 64-kbps leased line.
- ❖ Bandwidth is a rating of the maximum attainable throughput on a link

---



---



---



---



---



---

## Routing Load

- ❖ **Load** refers to the degree to which a network resource, such as a router, is busy.
- ❖ **Load** can be calculated in a variety of ways,
  - ✓ **CPU utilization and**
  - ✓ **Packets processed per second.**

---



---



---



---



---



---

## Reliability

- ❖ Reliability, in the context of routing algorithms, refers to the dependability (usually described in terms of the bit-error rate) of each network link.
- ❖ Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links.

---



---



---



---



---



---

## Routing Algorithm

---



---



---



---



---



---

## Routing Algorithm

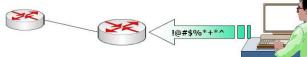
- ❖ Shortest Path Routing
- ❖ Flooding
- ❖ Distance Vector Routing
- ❖ Link State Routing
- ❖ Hierarchical Routing
- ❖ Broadcast Routing
- ❖ Multicast Routing

## Routing Algorithm

- ❖ Routing algorithms can be classified based on the following criteria:
  1. Non-adaptive(static) Versus Adaptive Routing
  2. Single-path Versus Multi-path Routing
  3. Intra-domain Versus Inter-domain Routing
  4. Flat Versus Hierarchical Routing
  5. Link-state Versus Distance Vector Routing

### Non-adaptive Versus Adaptive Routing

## Non-Adaptive Routing



- ❖ Non-adaptive is also known as Static Routing Algorithms.
  - ❖ In Static routing algorithms,
    - ✓ The routing table mappings are established by the network administrator before the beginning of routing.
    - ✓ These mappings of the routing table do not change unless the network administrator alters them

## Non-Adaptive Routing

- ❖ **Routing Algorithms** that use static routes are
    - ✓ Simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.
  - ❖ Routing decisions in these algorithms does not depend upon on current topology or traffic.

### **Limitation: Non-Adaptive Routing**

- ❖ Static routing systems cannot **react to network changes**, they generally are considered **unsuitable for today's Network**

## Adaptive Routing

- ❖ Adaptive routing is also referred as dynamic routing
- ❖ Most of the dominant routing algorithms today are dynamic routing algorithms,
- ✓ Adjust to changing network circumstances by analyzing incoming routing update messages.



## Adaptive Routing

- ❖ If the message indicates that a network change has occurred
  - ✓ The routing software recalculates routes and sends out new routing update messages.
- ❖ Dynamic routing algorithms can be supplemented with static routes where appropriate.

## Static Routing Algorithm

## Some of the Static Routing Algorithm

- ❖ Shortest Path Routing
- ❖ Flooding

---

---

---

---

---

---

### Shortest Path Routing

---

---

---

---

---

---

## Shortest Path Routing

- ❖ Shortest Path Routing is a **feasible routing algorithms** that is simple and easy to understand.
- ❖ The idea of this **shortest path routing** is to **build a graph of the subnet**
  - ✓ Each node of the **graph representing a router** and each arc of the **graph representing a communication link**.

---

---

---

---

---

---

## Shortest Path Routing

- ❖ To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- ❖ Shortest Path Routing is a Path-selection model
- ❖ It is a Destination-based Routing Algorithm
- ❖ Shortest Path Routing maintains static link weights, therefore it is Load-insensitive
- ❖ It provides Minimum hop count or sum of link weights

---



---



---



---



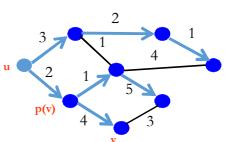
---



---

## Shortest-Path Problem

- ❖ Given: Network Topology with link costs
  - $c(x,y)$ : Link cost from node x to node y
  - Infinity if x and y are not direct neighbors
- ❖ Compute: least-cost paths to all nodes
  - From a given source u to all other nodes
  - $p(v)$ : predecessor node along path from source to v




---



---



---



---



---



---

## Dijkstra's Shortest-Path Algorithm

- ❖ Dijkstra's Shortest-Path is a Iterative algorithm
- ✓ After k iterations, know least-cost path to k nodes
- ❖ S: nodes whose least-cost path definitively known
- ✓ Initially,  $S = \{u\}$  where u is the source node
- ✓ Add one node to S in each iteration

---



---



---



---



---



---

## Dijkstra's Shortest-Path Algorithm

- ❖ D(v): current cost of path from source to node v
- ✓ Initially,  $D(v) = c(u,v)$  for all nodes v adjacent to u
- ✓  $D(v) = \infty$  for all other nodes v
- ✓ Continually update  $D(v)$  as shorter paths are learned

---



---



---



---



---



---

## Dijkstra's Algorithm

```

1 Initialization:
2   S = {u}
3   for all nodes v
4     if v adjacent to u {
5       D(v) = c(u,v)
6     else D(v) = ∞
7
8 Loop
9   find w not in S with the smallest D(w)
10  add w to S
11  update D(v) for all v adjacent to w and not in S:
12    D(v) = min{D(v), D(w) + c(w,v)}
13 until all nodes in S

```

---



---



---



---

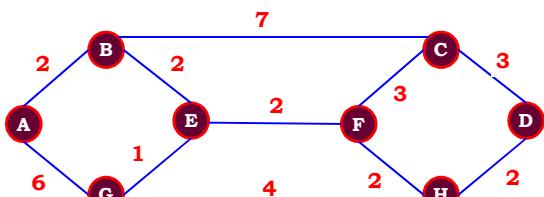


---



---

## Dijkstra's Algorithm Example




---



---



---



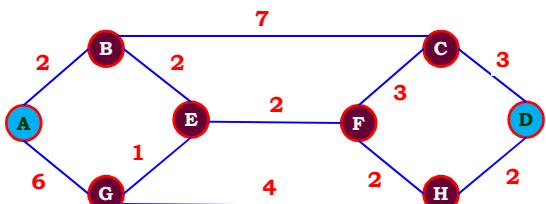
---



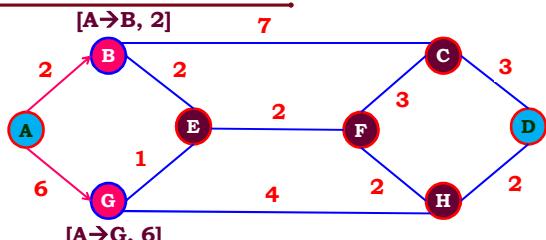
---



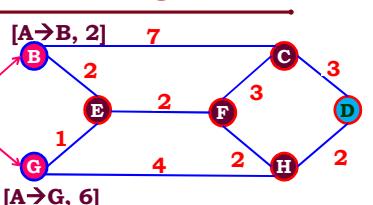
---

**Dijkstra's Algorithm Example**

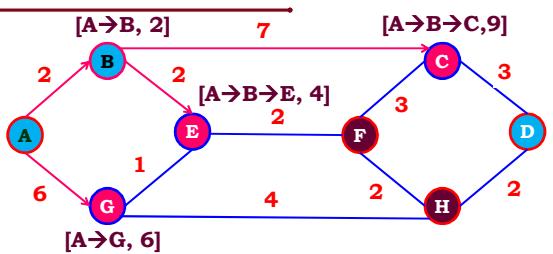
**Source Node : 'A' and Destination Node : 'D'**

**Dijkstra's Algorithm Example**

**Make 'A' Permanent and B,G are Temporary**

**Dijkstra's Algorithm Example**

	A	B	C	D	E	F	G	H
A	2	∞	∞	∞	∞	6	∞	

**Dijkstra's Algorithm Example**

Make 'B' Permanent and E,C are Temporary

---



---



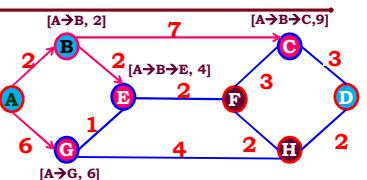
---



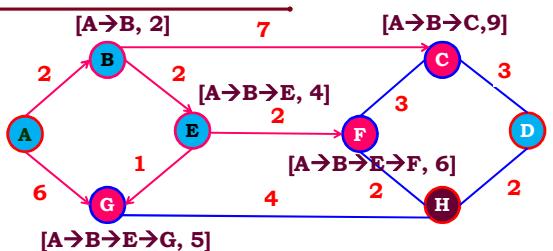
---



---

**Dijkstra's Algorithm Example**

	B	C	D	E	F	G	H
A	2	$\infty$	$\infty$	$\infty$	$\infty$	6	$\infty$
$A \rightarrow B$		7	$\infty$	2	$\infty$	$\infty$	$\infty$

**Dijkstra's Algorithm Example**

Make 'E' Permanent and G,F are Temporary

---



---



---



---



---

**Dijkstra's Algorithm Example**

	B	C	D	E	F	G	H
A	2	∞	∞	∞	∞	6	∞
A→B		7	∞	2	∞	∞	∞
A→B→E	2	∞	∞	∞	2	1	∞

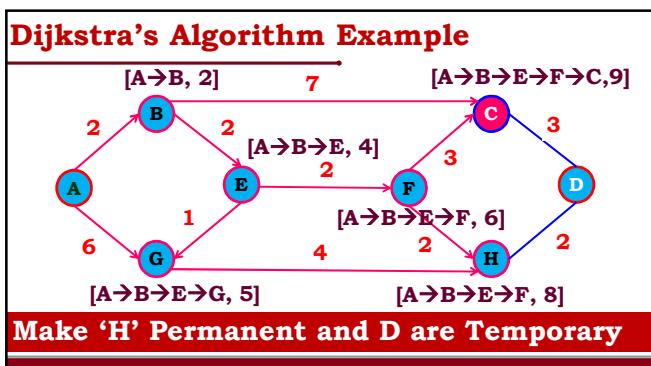
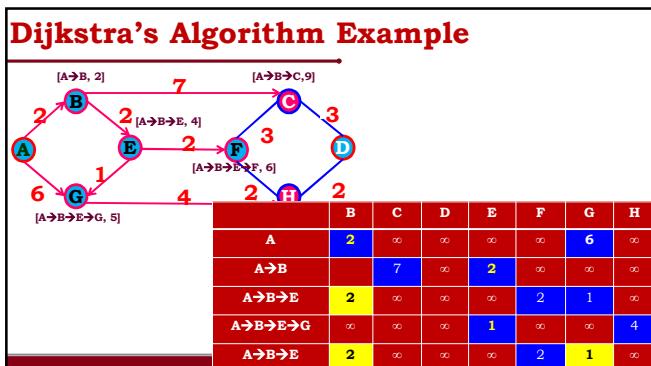
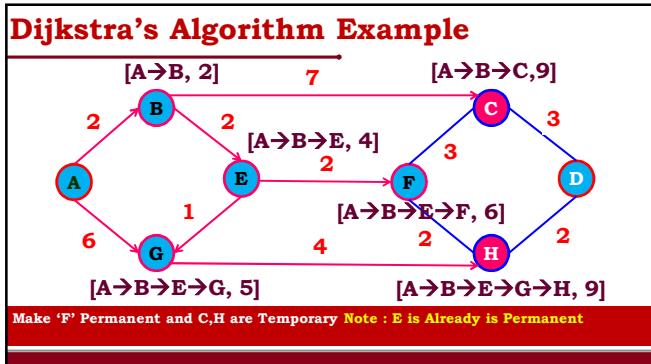
**Dijkstra's Algorithm Example**

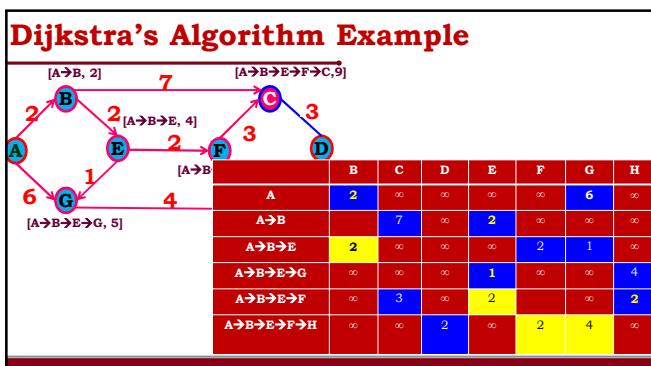
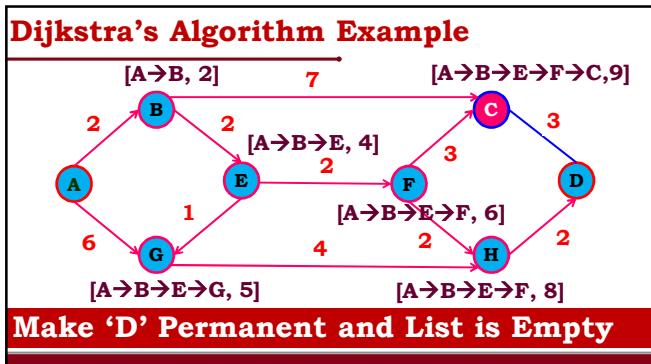
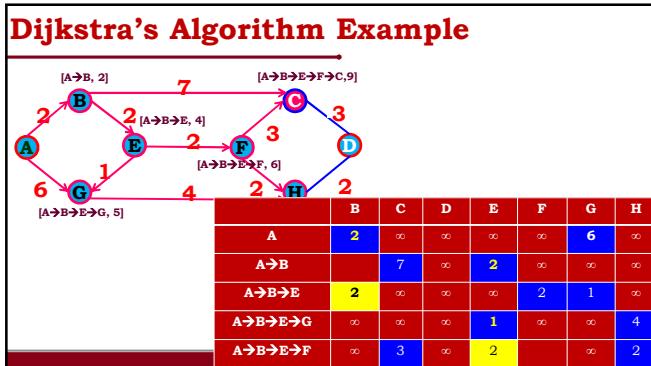
Make 'G' Permanent and A,H are Temporary Note : A is Already is Permanent

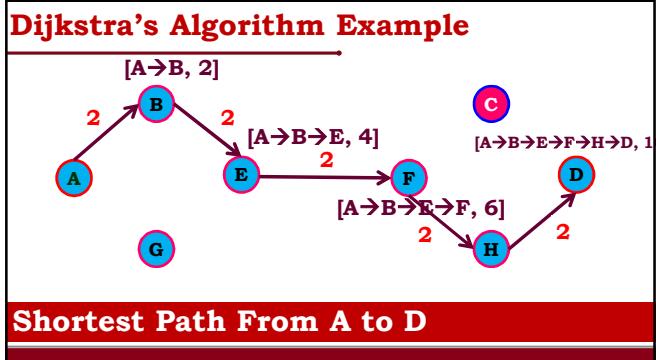
	B	C	D	E	F	G	H
A	2	∞	∞	∞	∞	6	∞
A→B		7	∞	2	∞	∞	∞
A→B→E	2	∞	∞	∞	2	1	∞
A→B→E→G	∞	∞	∞	1	∞	∞	4
A→B→E→G→H	2	∞	∞	∞	2	1	∞

**Dijkstra's Algorithm Example**

	B	C	D	E	F	G	H
A	2	∞	∞	∞	∞	6	∞
A→B		7	∞	2	∞	∞	∞
A→B→E	2	∞	∞	∞	2	1	∞
A→B→E→G	∞	∞	∞	1	∞	∞	4
A→B→E→G→H	2	∞	∞	∞	2	1	∞








---



---



---



---



---



---




---



---



---



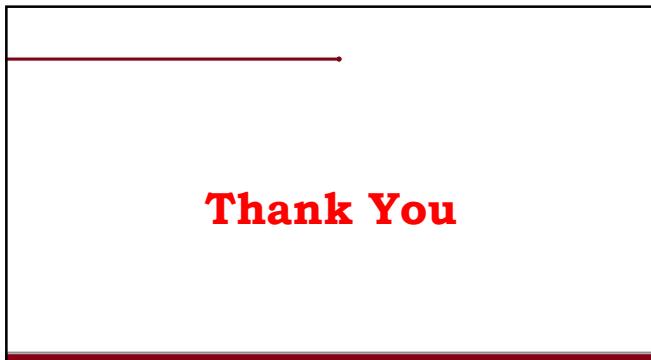
---



---



---




---



---



---



---



---



---

## CCN: Network Layer- Dynamic Routing Algorithm

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Types of Dynamic Routing Algorithm

#### Types of Dynamic Routing

- ❖ Two dynamic algorithms

1. Distance Vector Routing and

2. Link State Routing,

## Distance Vector Routing

---

---

---

---

---

---

## Distance Vector Routing

- ❖ The **distance vector routing algorithm** are also known as
  - ✓ **Distributed Bellman-Ford routing algorithm (1957)** or
  - ✓ **Ford-Fulkerson algorithm (1962)**

---

---

---

---

---

---

## Distance Vector Routing

- ❖ In this routing algorithms
- ✓ Each **router** maintain a table (i.e, a vector) giving the **best known distance to each destination**
- ✓ These tables are updated by exchanging information with the neighbors.

---

---

---

---

---

---

## Distance Vector Routing

- ❖ This routing algorithm uses
  - ✓ Destination and ports addresses,
  - ✓ Distance vector routing tables that contain the distance (in hops) to each network
  - ✓ Routing tables in distance vector (DV) routing typically contain a number of routes to a given network address

---



---



---



---



---



---

## Distance Vector Routing

- ❖ In distance vector routing, The routing table entry contains two parts
  - ✓ The Preferred outgoing line to use for that destination
  - ✓ Estimate of the time or distance to that destination.

---



---



---



---



---



---

## Bellman-Ford routing algorithm

- ❖ Define Distances at each node 'x'
- $d_x(y) = \text{cost of least-cost path from } x \text{ to } y$
- ❖ Update distances based on neighbors
- $d_x(y) = \min \{c(x,v) + d_v(y)\} \text{ over all neighbors } v$

---



---



---



---



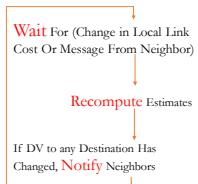
---



---

## Distance Vector Routing

Each Node:




---

---

---

---

---

---

## Distance Vector Routing

- ❖ In distance vector routing, The **routing metric** might be used as
  - ✓ **Number of hops,**
  - ✓ **Time Delay in milliseconds,**
  - ✓ **Total number of packets queued along the path etc**

---

---

---

---

---

---

## Distance Vector Routing: Number of Hops

---

---

---

---

---

---

### Distance Vector Routing: Number of Hops

- The router is assumed to know the "distance" to each of its neighbors.

If the metric is hops, the distance is just one hop.

---



---



---



---

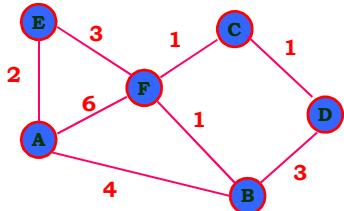


---



---

### Distance Vector Routing : Example




---



---



---



---



---



---

### Distance Vector Routing : Example

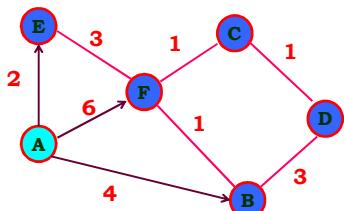


Table for A		
Dst	Cst	Hop
A	0	A
B	4	B
C	$\infty$	-
D	$\infty$	-
E	2	E
F	6	F

---



---



---



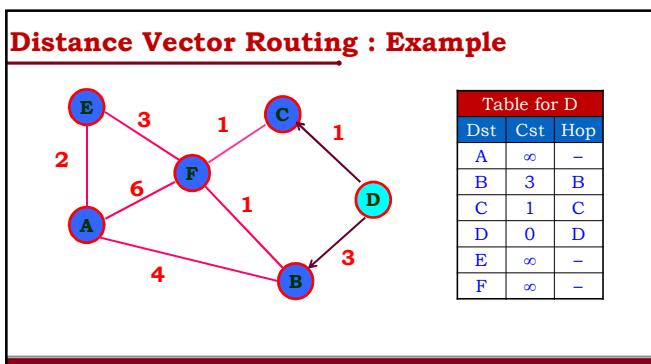
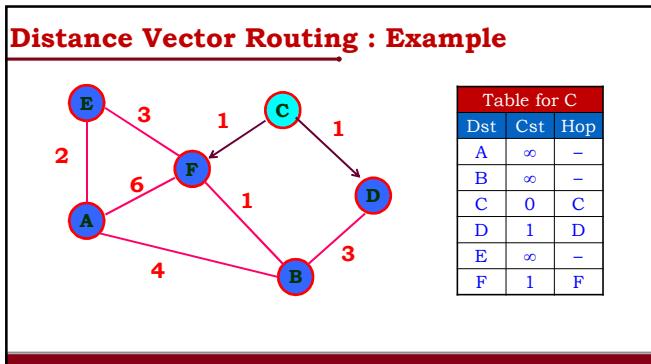
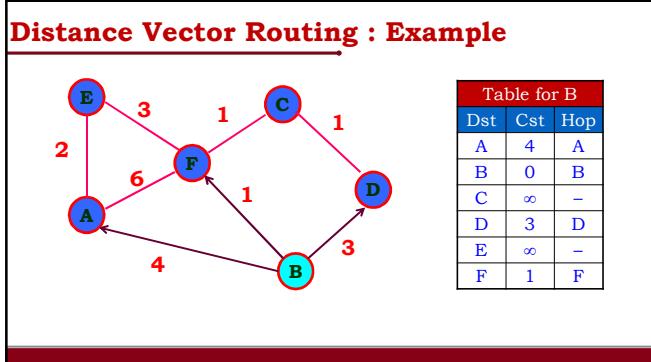
---

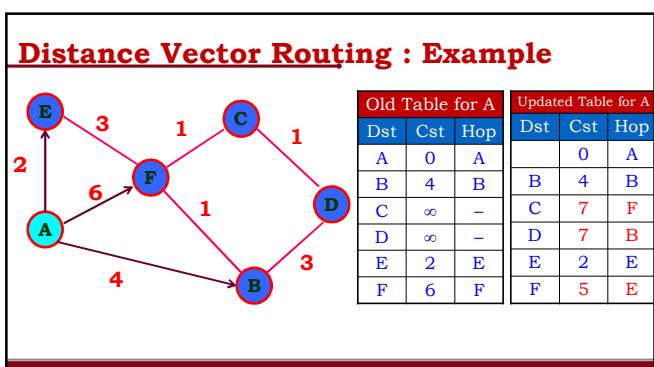
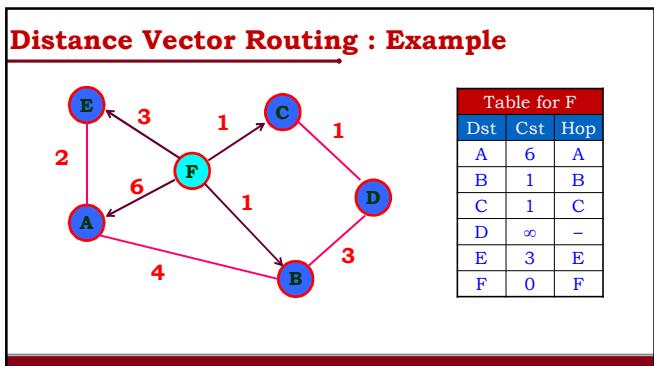
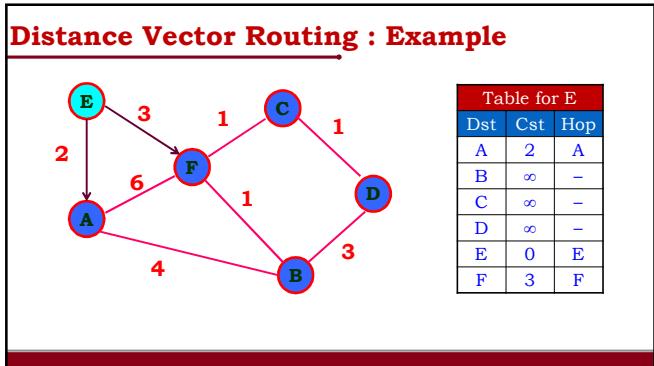


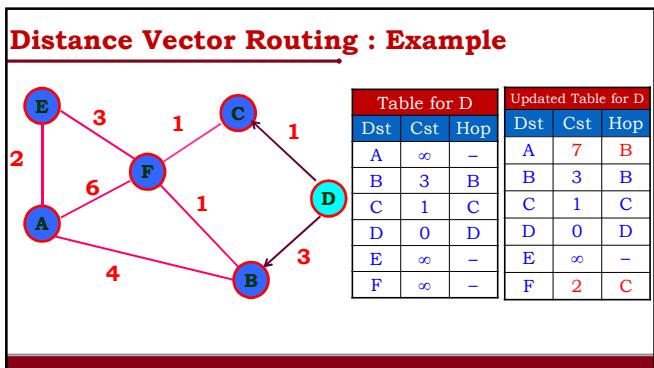
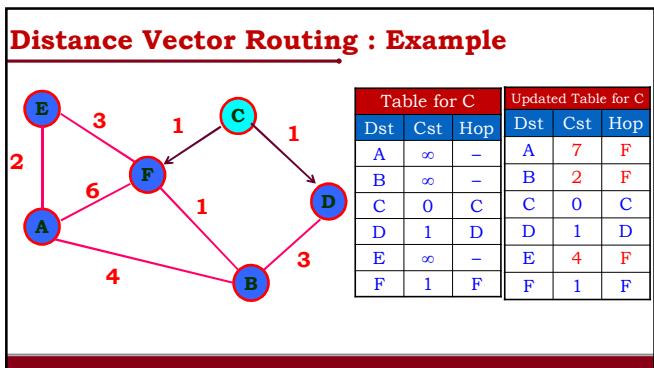
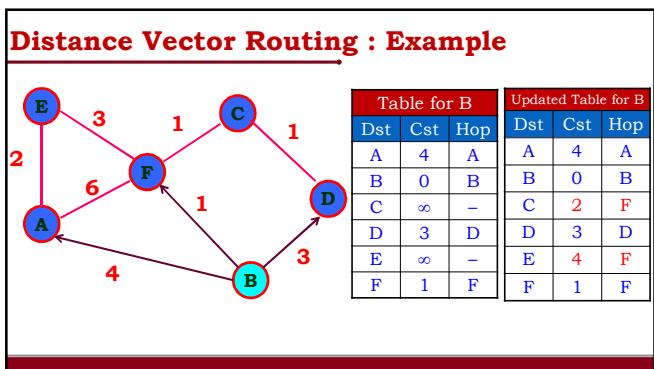
---

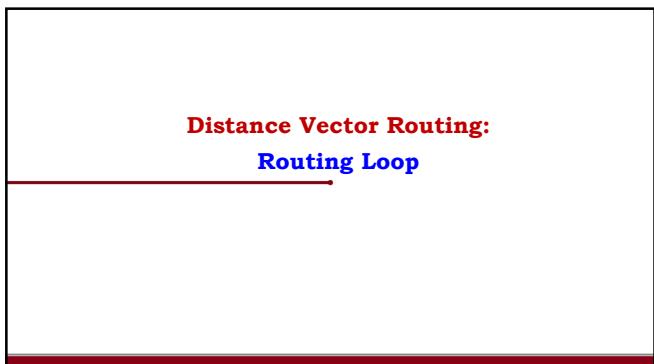
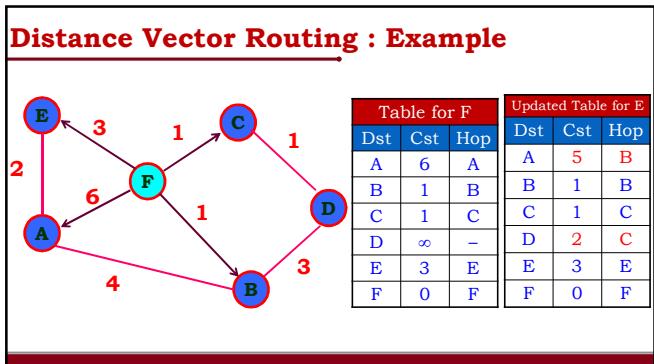
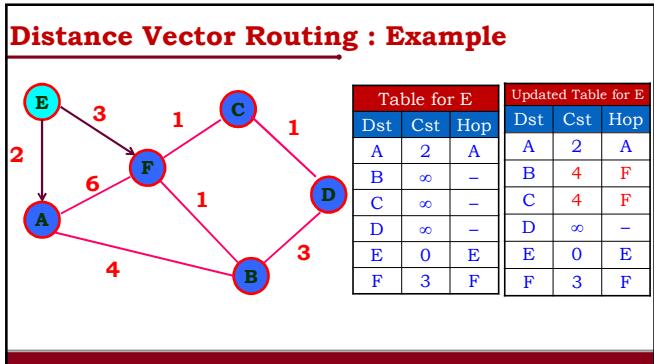


---



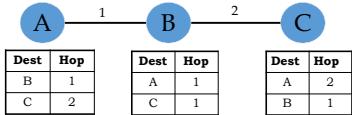






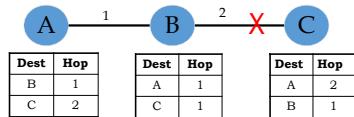
## Routing Loop

- ❖ Consider the network shown below:



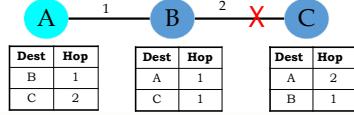
## Routing Loop

- ❖ What happens if we have link failure between B & C?
- ✓ B will realize there is a problem, and tries to determine another route to C



## Routing Loop

- ❖ B will ask its neighbours (i.e. A), if they can reach C
- ❖ How will A respond?
- ✓ Look at A's routing table to find the answer

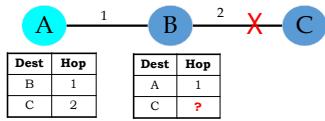


## **Slow Convergence**

- ❖ B will ask its **neighbours** (i.e. A), if they can reach C

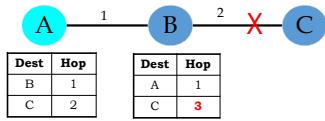
### ❖ How will A respond?

- ✓ Look at A's routing table to find the answer



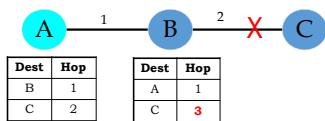
## Routing Loop

- ❖ A thinks that it can reach C in 2 hops
  - ❖ This path is through B, but that is not known to A
  - ❖ B will get this information and (falsely) think it can access C through A
  - ❖ B believes that the path is 3 hops



## Routing Loop

- ❖ Now suppose **B has a message for C**:
  - ❖ It will **send the message to A**, Just as its **routing table suggests**
  - ❖ **A will send the message back to B**
  - ❖ This will **put the packet into a continuous loop**



**Distance Vector Routing:**  
**Slow Convergence**

---

---

---

---

---

---

**Slow Convergence**

- ❖ Distance vector routing works in theory
- ✓ It is not suitable in practice
- ❖ Distance vector routing reacts rapidly for good news,
- ✓ Responds slowly to bad news.

---

---

---

---

---

---

**Slow Convergence**

- ❖ Convergence is getting correct routing tables after a failure in a router or link
- ✓ May take minutes
- ✓ During that time, many packets may be lost

---

---

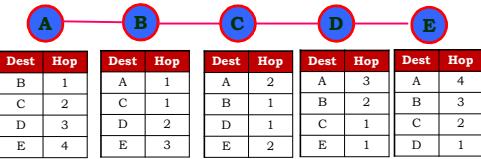
---

---

---

---

### Fast Good News Propagates : Example



Dest	Hop								
B	1	A	1	A	2	A	3	A	4
C	2	C	1	B	1	B	2	B	3
D	3	D	2	D	1	C	1	C	2
E	4	E	3	E	2	E	1	D	1

---

---

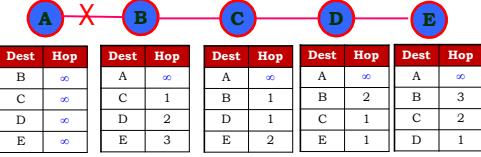
---

---

---

---

### Fast Good News Propagates: Example



Dest	Hop								
B	$\infty$	A	$\infty$	A	$\infty$	A	$\infty$	A	$\infty$
C	$\infty$	C	1	B	1	B	2	B	3
D	$\infty$	D	2	D	1	C	1	C	2
E	$\infty$	E	3	E	2	E	1	D	1

---

---

---

---

---

---

- ❖ Delay metric is the number of hops.
- ❖ Suppose A is down initially and all the other routers know this.
- ❖ All the Routers are recorded the delay to A as infinity.

### Fast Good News Propagates: First Exchange



Dest	Hop	Dest	Hop	Dest	Hop	Dest	Hop
A	1	A	$\infty$	A	$\infty$	A	$\infty$

---

---

---

---

---

---

- ❖ When A comes up with routing neighbor exchanges.
- ❖ All the routers exchange simultaneously.
- ❖ First exchange, B learns that its left neighbor has zero delay to A.

### Fast Good News Propagates: Second Exchange



Dest	Hop
A	1

Dest	Hop
A	2

Dest	Hop
A	$\infty$

Dest	Hop
A	$\infty$

- ❖ B now makes an entry in its routing table that A is one hop away to the left
- ❖ On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2,
- ❖ However, D and E do not hear the good news until later.

### Fast Good News Propagates: Third Exchange



Dest	Hop
A	1

Dest	Hop
A	2

Dest	Hop
A	3

Dest	Hop
A	$\infty$

- ❖ Clearly, the good news is spreading at the rate of one hop per exchange.
- ❖ In a subnet whose longest path is of length N hops, within N exchanges everyone will know about newly-revived lines and routers.

### Fast Good News Propagates: Fourth Exchange



Dest	Hop
A	1

Dest	Hop
------	-----

- ❖ Clearly, the good news is spreading at the rate of one hop per exchange.
- ❖ In a subnet whose longest path is of length N hops, within N exchanges everyone will know about newly-revived lines and routers.

**Distance Vector Routing:**  
**The Count-to-Infinity Problem**

---



---



---



---



---



---

**Bad News Propagates : Example**

---



---



---



---



---



---



**Bad News Propagates : Example**

---



---



---



---



---

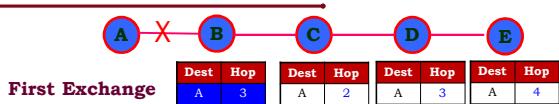


---



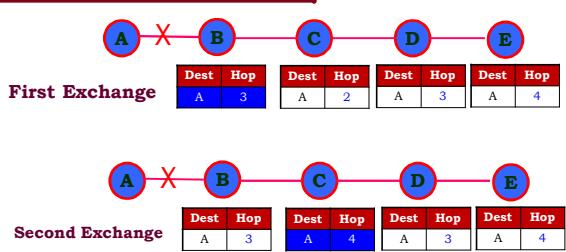
- ❖ Suddenly A goes down, The line between A and B is cut,
  - ❖ At the first packet exchange, B does not hear anything from A.
  - ❖ Fortunately, C says: Do not worry; I have a path to A of length 2.
  - ❖ **C does not know that C's path runs through B itself.**
- 
- 
- 
- 
- 
-

### Bad News Propagates : Example

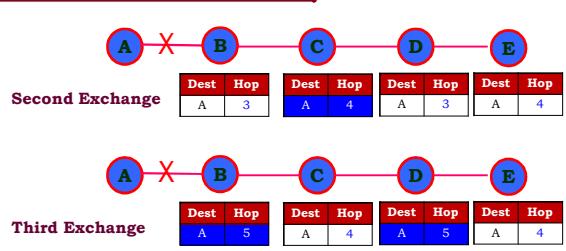


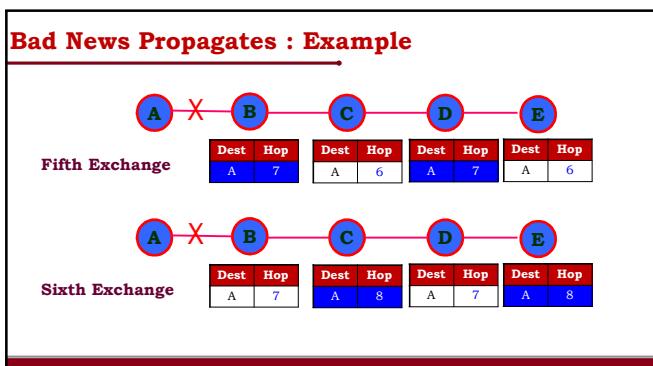
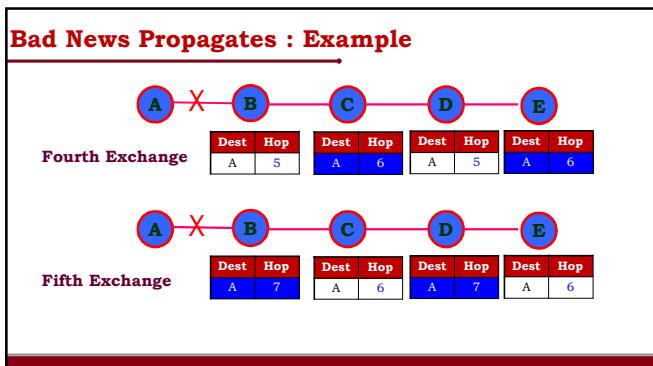
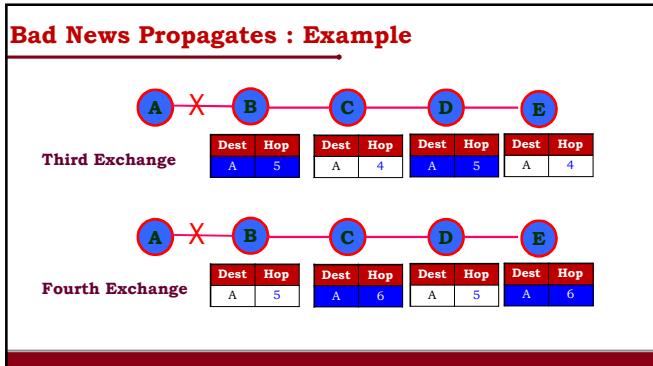
- ❖ Suddenly A goes down, The line between A and B is cut,
- ❖ At the **first packet exchange**, B does not hear anything from A.
- ❖ Fortunately, C says: Do not worry; I have a path to A of length 2.
- ❖ **C does not know that C's path runs through B itself.**

### Bad News Propagates : Example

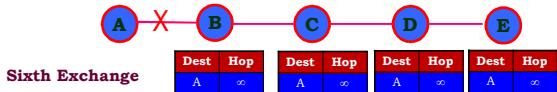


### Bad News Propagates : Example





### Finally : The Count-to-Infinity Problem



- ❖ Gradually, all routers work their way up to infinity
- ❖ If the metric is time delay, Long delay from being treated as down.
- ❖ This problem is known as the **count-to-infinity** problem.

### Distance Vector Summary

- ❖ Good
  - ✓ Only need communicate with neighbors (so little bandwidth is wasted on protocol overhead)
  - ✓ Relatively little processing of info
- ❖ Bad
  - ✓ Count to infinity problem
  - ✓ Slow convergence (the real issue)

### Link-State Routing

## **Link-State Routing**

- ❖ Link-state algorithms is also known as **shortest path first algorithms**
- ❖ This algorithms is named after **Dijkstra's algorithm (1959)** which it uses to compute routes
- ❖ Variants of link state routing are now **widely used.**

---



---



---



---



---



---

## **Link-State Routing : Idea**

- ❖ The idea behind link state routing is simple and can be stated as **five parts**. Each router must do the following:

  1. Discover its neighbors and learn their network addresses.
  2. Measure the delay or cost to each of its neighbors.
  3. Construct a Link State Packet
  4. Send this Link State Packet to all other routers.
  5. Compute the shortest path to every other router.

---



---



---



---



---



---

### **Step-1 Discover the Neighbors & Learn the Neighbors Information**

- ❖ All routers will **maintain the routing tables**
- ✓ Contains **Information of entire network topology**
  - Lists of routers,
  - Information about each router's neighbors and
  - The connection between the neighbors
- ❖ Each router keeps track of its **neighbors links**
  - ✓ Whether the **link is up or down**
  - ✓ The **cost on the link**

---



---



---



---



---



---

**Step-1 Discover the Neighbors & Learn the Neighbors Information**

- ❖ Each router is responsible for
  - ✓ Meeting its neighbors
  - ✓ Learning their names (Network Addresses)
- ❖ The task to accomplish the above goal by sending a special **HELLO packet** on each point-to-point line.
- ❖ Receiving router on the other side is expected to send back a reply back by telling who it is.

---



---



---



---



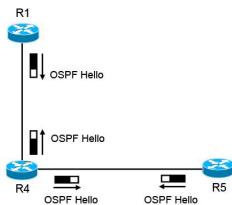
---



---

**Neighbors Discovery and Learning the Neighbors Information**

- ❖ These names(Network Addresses) must be globally unique




---



---



---



---



---



---



---

**Step-2 Measure The Delay Or Cost To Each Of Its Neighbors.**

- ❖ The link state routing algorithm requires each router to know the delay to each of its neighbors.
- ❖ To determine the delay
  - ✓ Intermediate Node/ Source Node will sent special ECHO packet to the neighbor node
  - ✓ On the other side neighbor node will reply with special ECHO packet back immediately.

---



---



---



---



---



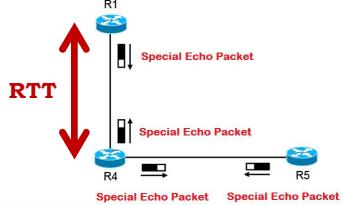
---



---

### Step-2 Measure The Delay Or Cost To Each Of Its Neighbors.

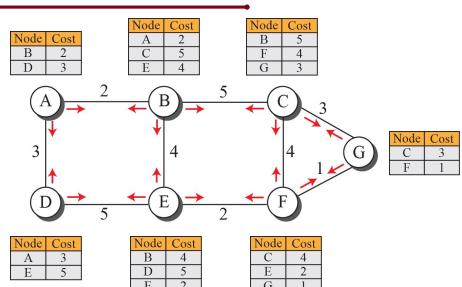
- By measuring the round-trip time and dividing it by two,
- The sending router can get a reasonable estimate of the delay.



### Step-3 Construct a Link State Packet

- The next step is for each router to build a **Link State Packet** containing all the data.
- The **Link State Packet** Contains
  - Sender ID**,
  - Sequence Number and**
  - Age or Time-to-Live (TTL) for this packet**
  - List of Neighbors and Cost of the link**

### Step-3 Construct a Link State Packet



### Step-3 Construct a Link State Packet

- ❖ Each router broadcasts the link state Packet
  - ✓ To give every router a **complete view of the graph**

---



---



---



---

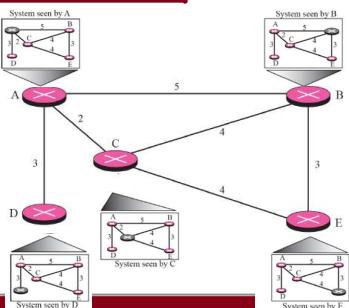


---



---

### Complete Topological Information




---



---



---



---



---



---

### How to Distribute the Link State Packet

- ❖ The Important part of this algorithm is **distributing the link state packets reliably**
  - ✓ Distribution of Link State Packet (LSPs) can be difficult
- ❖ The fundamental idea to **distribute the link state packets** is to use **Flooding Technique**

---



---



---



---



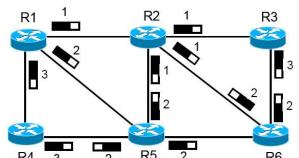
---



---

### Flooding

- ❖ Each Node sends link-state information to all other links and then the next node sends out all of its links
- ✓ Except the one(s) where the information arrived



### Reliable Flooding

- ❖ Reliable Flooding ensure all nodes receive link-state information that has latest version
- ❖ Challenges to Reliable Flooding
  - ✓ Packet loss
  - ✓ Out-of-order arrival

### Solutions: Reliable Flooding

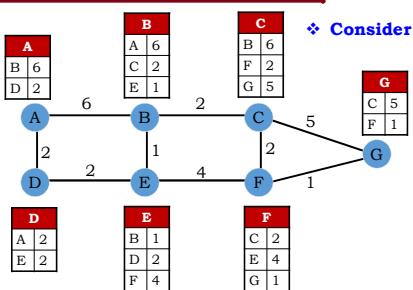
- ❖ Sequence numbers
- ❖ Time-to-live for each packet
- ❖ Acknowledgments and Retransmissions

**Step-4 : Compute The Shortest Path To Every Other Router.**

**Already Discussed**

**Dijkstra's LSR Algorithm:**

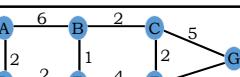
❖ Consider the following network:



**Dijkstra's LSR Algorithm:**

❖ Now, if we want to generate a PATH for C:

C(0)



**Dijkstra's LSR Algorithm:**

- Examine C's LSP
- Add F, G, and B to Temp

---

---

---

---

---

---

**Dijkstra's LSR Algorithm:**

- Place F in PATH (shown as solid line)
- Add G and E to Temp (adding costs)

---

---

---

---

---

---

**Dijkstra's LSR Algorithm:**

- G exists in Temp twice, keep only the best
- The new G is a better path than the old ( $3 < 5$ )

---

---

---

---

---

---

**Dijkstra's LSR Algorithm:**

- Put B into path (shown as solid line)
- Add A and E to Temp

The graph shows the state of the algorithm after putting node B into the path. Node B is highlighted in cyan with a value of 2. Node A is also highlighted in cyan with a value of 0, indicating it is being considered for addition to the temporary set. Other nodes are shown in their original colors (red or blue) with their current shortest distances.

---



---



---



---



---



---

**Dijkstra's LSR Algorithm:**

- E exists in Temp twice, keep only the best
- The new E is better than the old ( $3 < 6$ )

The graph shows the state of the algorithm after updating node E's distance to 3, as it is now closer via node C. The previous distance of 6 is crossed out.

---



---



---



---



---



---

**Dijkstra's LSR Algorithm:**

- Place E in PATH (shown as solid line)
- Add D to Temp

The graph shows the state of the algorithm after placing node E into the path and adding node D to the temporary set. Node E is highlighted in cyan with a value of 3. Node D is highlighted in blue with a value of 5.

---



---



---



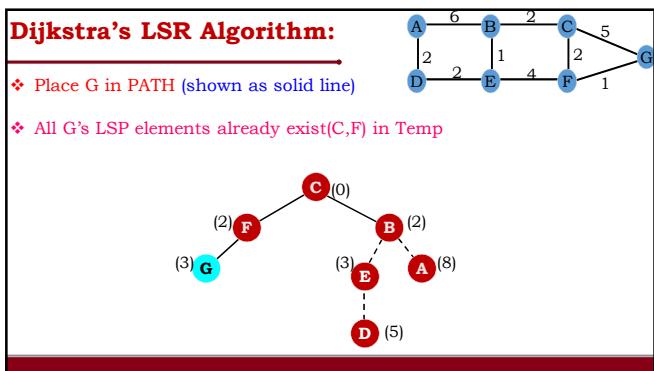
---



---



---




---



---



---



---



---



---



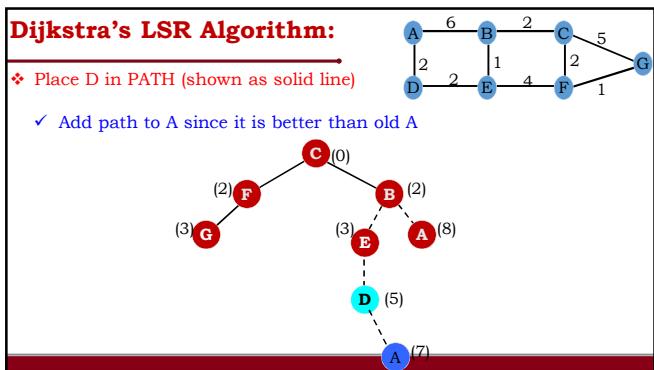
---



---



---




---



---



---



---



---



---



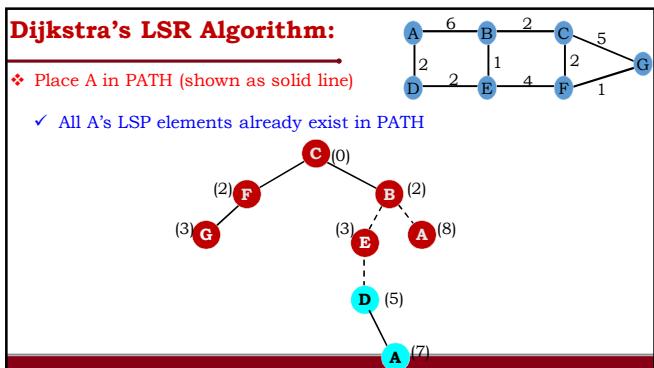
---



---



---




---



---



---



---



---



---



---



---

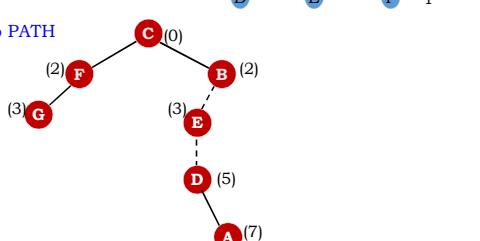


---

**Dijkstra's LSR Algorithm:**

- ❖ We are done since all routes from Temp

were placed into PATH




---

---

---

---

---

---

### **When LSPs Are Generated And Distributed**

---

---

---

---

---

---

### **When LSPs Are Generated And Distributed**

- ❖ A time period passes
- ❖ New neighbors connect to the router
- ❖ The link cost of a neighbor has changed
- ❖ A link to a neighbor has failed (link failure)
- ❖ A neighbor has failed (node failure)

---

---

---

---

---

---

---

### Link-State Summary

❖ Good

- ✓ Converges relatively quickly

❖ Bad

- ✓ Lots of information stored at each node because LSP for each node in network must be stored at each node (**scalability problem**)
- ✓ Flooding of LSPs uses bandwidth
- ✓ Potential security issue (if false LSP propagates)

Thank You

## CCN: Data Link Layer: Introduction

---

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



## Goals:

---

- ❖ Introduction to the data link layer
  - ✓ Framing- Determining message boundaries✓
  - ✓ Error control ✓
  - ✓ Error detection ✓
    - Parity ✓
    - Cyclic redundancy checks (CRC) ✓
  - ✓ Error correction ✓
    - Row/column parity ✓
    - Hamming codes ✓

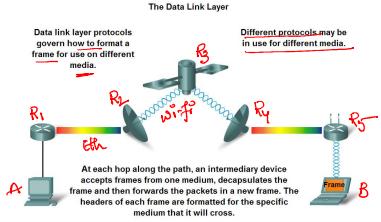
## Introduction

---

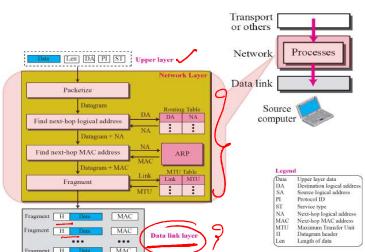


## Data Link Layer

- ❖ **Service:** To reliably deliver packets between two physically connected machines.



## Data Link Layer



## Data Link Layer

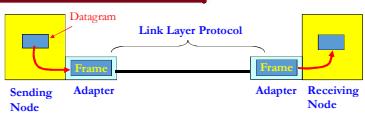
- ❖ The **Data Link layer** performs **two basic services:**
  - ✓ Allows the upper layers to access the media using techniques such as framing
  - ✓ **Controls** how data is placed onto the media and is received from the media using techniques such as media access control and error detection



## Functions of Data Link Layer

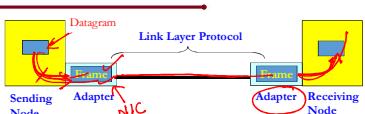
- ❖ Providing a well-defined service interface to the network layer
- ❖ Determining how the bits of the physical layer are grouped into frames
- ❖ Dealing with transmission errors
- ❖ Regulating the flow of frames so that slow receivers are not flooded by fast senders

## Digital Adaptors Communicating



- ❖ Data Link layer implemented in **Adaptor (Network Interface Card)**
- ✓ Ethernet card, PCMCIA card, 802.11 card

## Digital Adaptors Communicating



- ❖ Sending side:
  - ✓ Encapsulates datagram in a frame
  - ✓ Adds error checking bits, flow control, etc.
- ❖ Receiving side
  - ✓ Looks for errors, flow control, etc.
  - ✓ Extracts datagram and passes to receiving node

## Data Link Layer: Design Issues

❖ Services provided to network layer:

- ✓ Unacknowledged connectionless service.
- ✓ Acknowledged connectionless service.
- ✓ Acknowledged connection-oriented service.

❖ Framing:

- ✓ How to determine the start and end of a bit stream on the physical link?

---



---



---



---



---



---

## Data Link Layer : Design Issues

❖ Error control:

- ✓ Error correcting codes.
- ✓ Error detecting codes.

❖ Flow control:

- ✓ Ensuring that receiver can keep up with transmission.

---



---



---



---



---



---

## DLL: Services provided to Network Layer

---



---



---



---



---



---

## Framing and Synchronization

---



---

---

---

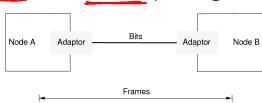
---

---

---

### Why Framing?

- ❖ The **physical layer** provides bit synchronization to ensure that the sender and receiver use the **same bit duration and timing**.
- ❖ **Framing** is necessary for the purpose of synchronization and data control functions.
- ❖ Framing is the process of grouping the bits into frames (messages or packets)




---

---

---

---

---

---

### Why Framing?

- ❖ How to interpret a continuous stream of bits as a series of frames?

**From Continuous Stream Of Bits**  
 → 011110000101110011110111000100010  
 ↑  
 To Series Of Frames  
 01111000  
**Frame-1**

---

---

---

---

---

---

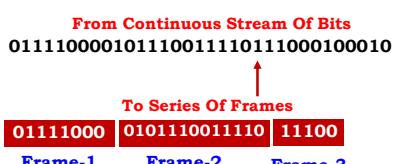
## Why Framing?

- ❖ How to interpret a continuous stream of bits as a series of frames?



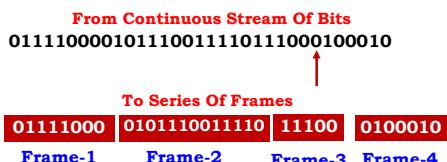
## Why Framing?

- ❖ How to interpret a continuous stream of bits as a series of frames?



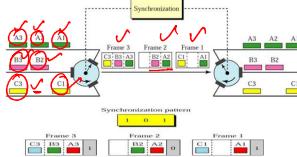
## Why Framing?

- ❖ How to interpret a continuous stream of bits as a series of frames?



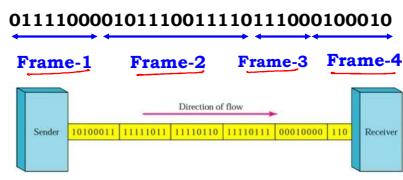
## Synchronization

- ❖ Data sent by a sender in bit-serial form through a medium must be correctly interpreted at the receiving end.
  - Requires the beginning, the end, logic level and duration of each bit as sent at the transmitting end must be recognized at the receiving end.



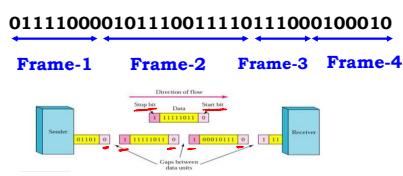
## Synchronization

- ❖ One basic question is how the receiver knows the Beginning and End of a Frame.



## Synchronization

- ❖ One basic question is how the receiver knows the Beginning and End of a Frame.



## Solution

- ❖ Framing will encapsulate the packets
- ✓ Frame are separated from one another
- ✓ Identifies the idle period of a data link transmission.
- ✓ Adds checksum at the end of each frame such that the receiver can detect errors

---



---



---



---



---



---

## Error Control

---



---



---



---



---

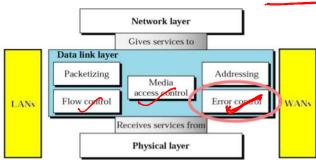


---

## Data Link Control

- ❖ The most important responsibilities of the data link layer are flow control and error control.

- ✓ Collectively, these functions are known as Data Link Control.




---



---



---



---



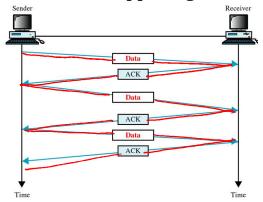
---



---

## Error Control

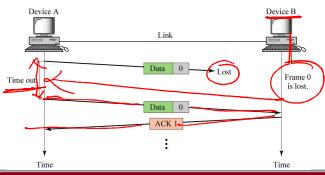
- The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the destination side.



## Error Control : Timers

- What happens if frame to vanish completely due to hardware troubling
- The receiver will not react at all.

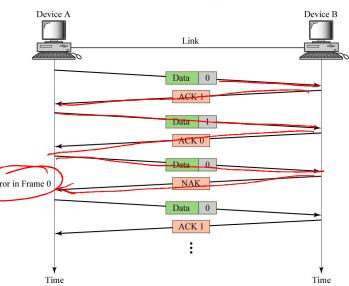
- Data Link Layer should introduce the TIMERS



## Error Control

- When will happen when the frames are transmitted multiple times by the sender and that the receiver will accept the same frame two or more times
  - Assign SEQUENCE NUMBERS to outgoing frames, so that the receiver can distinguish retransmissions from originals.

## Error Control: Sequence Number



## Error Detection And Error Correction

## Error Detection And Error Correction

- ❖ Data can be corrupted during transmission.
- ❖ For reliable communication
- ✓ Errors must be detected and corrected.



Some basic responsibilities of the data link layer is to **provide a reliable communication for the upper layer**

## Why Error Detection/Error Correction

- ❖ Environmental interference and physical defects in the communication medium
- ✓ Cause **RANDOM BIT ERRORS** during data transmission.
- ❖ Some applications require that errors be detected and corrected.
- ✓ **Detection:** determining if an error has occurred. ✓
- ✓ **Correction:** actually fixing errors. ✓

---



---



---



---



---



---

## Types of Errors

---



---



---



---



---

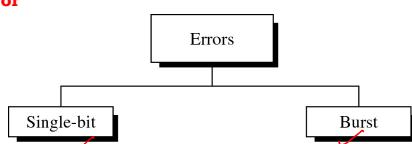


---

## Types of Errors

- ❖ There are TWO types of Bit Errors in a Link

1. Single-Bit Error
2. Burst Error




---



---



---



---



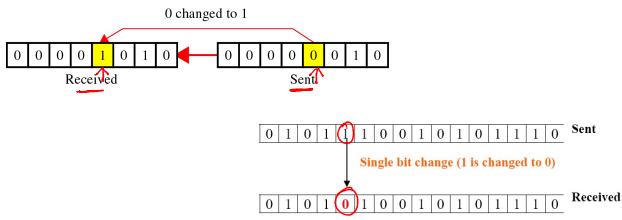
---



---

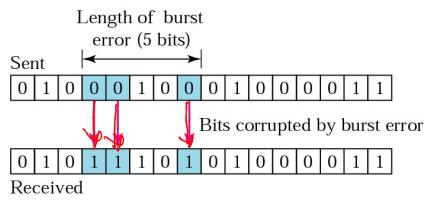
## Single-Bit Error

- In a single-bit error, only **one bit** in the **data unit** has changed.



## Burst Error

- A burst error means that **2 or more bits** in the **data unit** have changed



## Dealing with Errors

- Receiver must be aware that an **error occurred** in a frame
- Need to have an **ERROR DETECTION MECHANISM**
- Add some **information to correct errors**
  - ERROR CORRECTING MECHANISM**
- Ask **sender to re-send frame (retransmission strategies)**

**In practice all are employed**

## Goals:

- ❖ Introduction to the data link layer
  - ✓ Framing- Determining message boundaries
  - ✓ Error control
  - ✓ **Error detection**
    - Parity
    - **Cyclic redundancy checks (CRC)**
  - ✓ **Error correction**
    - Row/column parity
    - Hamming codes

---



---



---



---



---



---

## Error Detection Mechanism

---



---



---



---



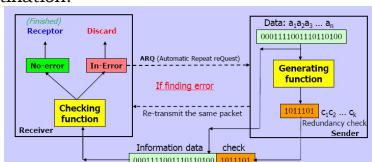
---



---

## Error Detection Mechanism

- ❖ Error detection uses the **Concept Of Redundancy**,
- ✓ where **additional bits are added** to facilitate to detect the errors at the destination.




---



---



---



---

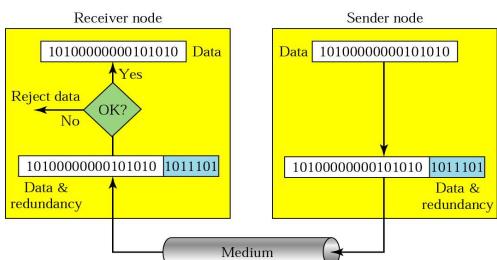


---



---

## Redundancy



## Error Detection Mechanism

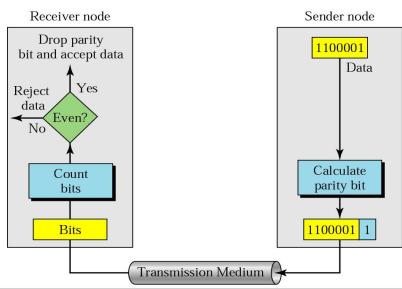
- ❖ Some of the popular techniques to detect the errors.
1. Simple Parity Check
  2. Two-dimensional Parity Check
  3. Checksum
  4. Cyclic Redundancy Check

### Simple Parity Check

## Simple Parity Check

- The most common and least expensive mechanism for error-detection is the Simple Parity Check.
- A redundant bit called parity bit, is appended to every data unit
- The number of 1s in the data unit including the parity becomes EVEN. it is called as Even Parity Checking.

## Simple Parity Check



## Example-1:

- Suppose the sender wants to send the word WORLD. In ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

- The following shows the actual bits sent

11101110 11011110 11100100 11011000 11001001

### Example-2:

- Now suppose the word WORLD in Example-1 is received by the receiver without being corrupted in transmission.

11101110 11011110 11100100 11011000 1100100

- The receiver counts the 1s in each character and comes up with even numbers {6, 6, 4, 4, 3}. The data are accepted.

---



---



---



---



---



---

### Example-3:

- Now suppose the word WORLD in Example-1 is corrupted during transmission.

11111110 11011110 11101100 11011000 11001001

- The receiver counts the 1s in each character and comes up with even and odd numbers {7, 6, 5, 4, 4}. The receiver knows that the data are corrupted, discards them, and asks for retransmission.

---



---



---



---



---



---

### Limitations.

- Errors more than one bit (Burst) cannot be detected.
- Single parity check code can detect only odd number of errors in a code word.

---



---



---



---



---



---

## Two-Dimensional Parity Check

---



---

---

---

---

---

---

## Two-Dimensional Parity Check

---

- ❖ Two-dimensional Parity Check which organizes the **block of bits** in the **form of a table**.
    - Parity check bits are calculated for **each row** which is called as **Vertical Parity Check**
    - Parity check bits are also calculated **for all columns** which is called as **Longitudinal Parity Check**
- 

---

---

---

---

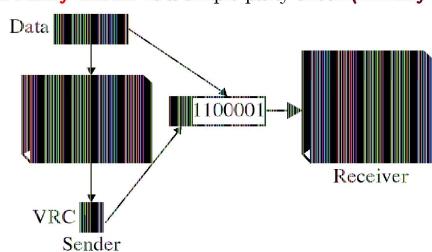
---

---

## Vertical Parity Check

---

- ❖ **Vertical Parity Check** is a simple parity check (**Already Discussed**)




---

---

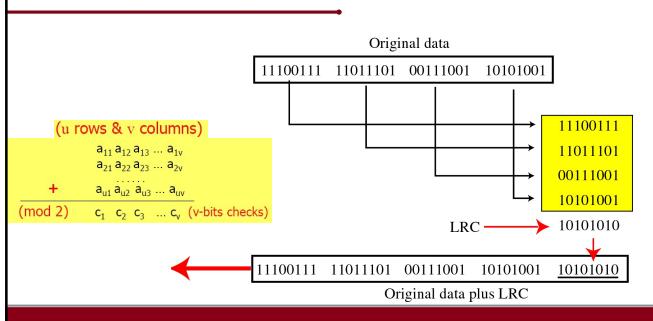
---

---

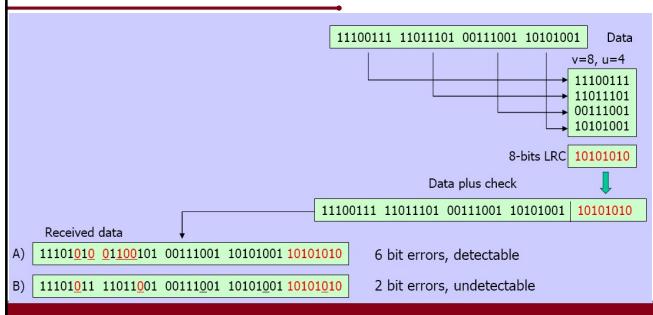
---

---

## Longitudinal Parity Check



## Longitudinal Parity Check: Example



## Limitations

- ❖ LRC can detect bit errors of odd number : 1, 3, 5, ...
- ❖ Sometimes, LRC can detect many bit errors of even number: 2, 4, 6, ...
- ❖ In Certain Cases, LRC **can't detect some bit errors of even number:**  
2, 4, 6, ...

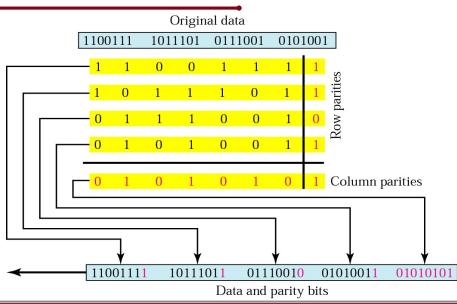
## Two-Dimensional Parity Check

- ❖ Two-dimensional Parity Check which makes use of both **Vertical Parity Check and Longitudinal Parity Check** finally both are sent along with the data.

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

a. Design of row and column parities

## Two-Dimensional Parity Check



## Advantages..

- ❖ Two- Dimension Parity Checking increases the probability of detecting burst errors

## Disadvantages..

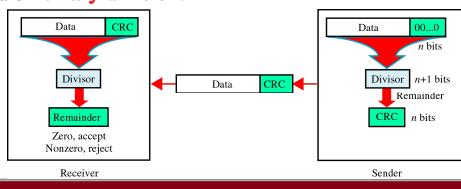
- ❖ If **two bits in one data unit are damaged** and **two bits in exactly same position in another data unit are also damaged**
- ✓ 2-D Parity check checker will not detect an error.

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

## CRC (Cyclic Redundancy Code)

### CRC (Cyclic Redundancy Code)

- ❖ The **Cyclic Redundancy Check** is the most powerful and easy to implement technique.
- ❖ CRC is based on **binary division**.



## CRC (Cyclic Redundancy Code)

- ❖ In CRC,
  - A **sequence of redundant bits**, called **cyclic redundancy check bits**, are appended to the end of data unit
  - The **resulting data unit** becomes **exactly divisible by predetermined binary number**.

---



---



---



---



---



---

## CRC (Cyclic Redundancy Code)

- ❖ At the **destination side**,
  - The **incoming data unit** is divided by the same number.
  - If there is **no remainder**, the **data unit is assumed to be correct** and is therefore accepted.
  - A **remainder** indicates that the **data unit has been damaged** in transit and therefore must be rejected.

---



---



---



---



---



---

## CRC Algorithm

- ❖ If a **k** bit message is to be transmitted,
  - The transmitter generates an **r-bit sequence**, known as **Frame Check Sequence (FCS)**.
  - **(k+r)** bits are actually being transmitted.
  - The **r-bit FCS** is generated by **dividing the original number, appended by r zeros**, by a predetermined number.

---



---



---



---



---



---

## CRC Algorithm

- ❖ This number  $(r+1)$  bit in length considered as the coefficients of a polynomial, called Generator Polynomial.
- ❖ The remainder of this division process generates the  $r$ -bit FCS.

---



---



---



---



---



---

## CRC Algorithm

- ❖ On receiving the packet, the receiver divides the  $(k+r)$  bit frame by the same predetermined number
  - if it produces no remainder, it can be assumed that no error has occurred during the transmission.

---



---



---



---



---



---

## A Polynomial

$$x^7 + x^6 + x^4 + x^3 + x + 1$$

---



---



---



---

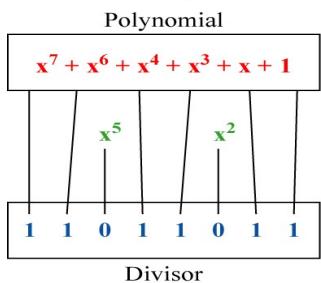


---



---

## A Polynomial Representing a Divisor



## Standard Polynomials

CRC-12

$$x^{12} + x^{11} + x^3 + x^2 + x + 1$$

CRC-16

$$x^{16} + x^{15} + x^2 + 1$$

CRC- ITU-T

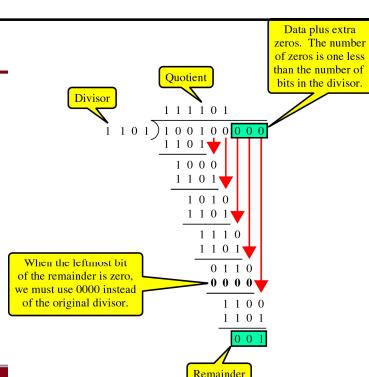
$$x^{16} + x^{12} + x^5 + 1$$

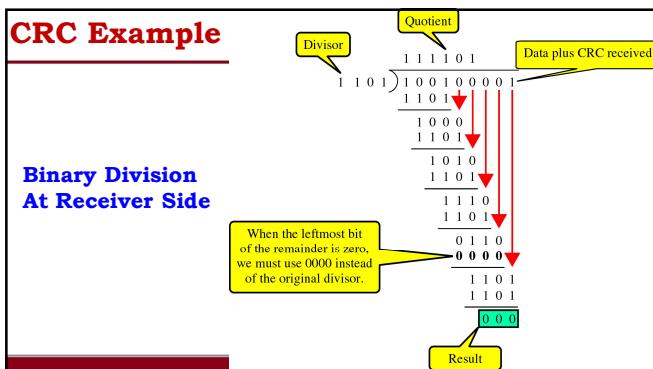
CRC-32

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

## CRC Example

Binary Division  
At Sender Side





Type of Error	Accuracy
single-bit error	100%
odd-number of error bits	100%
bursts of length < 17	100%
bursts of length = 17	100%
bursts of length > 17	$1 - (\frac{1}{2})^{15} = 0.9999695$
	$1 - (\frac{1}{2})^{16} = 0.9999848$

**Advantages**

- ❖ CRC is a **very effective error detection technique**
- ❖ CRC can **detect all single-bit errors**
- ❖ CRC can **detect all double-bit errors (three 1's)**
- ❖ CRC can **detect any odd number of errors (X+1)**
- ❖ CRC can **detect all burst errors** of less than the **degree of the polynomial**.
- ❖ CRC detects most of the **larger burst errors with a high probability**.
- ❖ For example **CRC-12** detects **99.97% of errors** with a length 12 or more.

**Goals:**

- ❖ **Introduction to the data link layer**
- ✓ **Framing- Determining message boundaries**
- ✓ **Error control**
- ✓ **Error detection**
  - **Parity**
  - **Cyclic redundancy checks (CRC)**
- ✓ **Error correction**
  - **Row/column parity**
  - **Hamming codes**

**Thank You**

**CCN: Data Link Layer: Elementary Data Link Protocols**

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.

**Goals:**

- ❖ Elementary Data Link Protocols
  - ✓ Flow Control
  - ✓ Error Control
- ❖ Data Link Protocols
  - ✓ Simplex Stop-and-Wait Protocol
  - ✓ ARQ Protocols

**Elementary Data Link  
Protocols**

## Introduction..

- ❖ For Reliable and Efficient Data Communication
  - ✓ There is need of coordination between two machines.
- ❖ Some of the following constraints are to be addressed:
  - ✓ Both sender and receiver have limited speed
  - ✓ Both sender and receiver have limited memory

---



---



---



---



---



---

## Introduction..

- ❖ It is necessary to satisfy the following requirements:
  - A fast sender should not overwhelm a slow receiver
    - ✓ Must perform a certain amount of processing before passing the data on to the higher-level software.
  - If error occur during transmission
    - ✓ it is necessary to devise mechanism to correct it

---



---



---



---



---



---

## Data Link Control

- ❖ The most important functions of Data Link layer to satisfy the above two requirements
- ❖ The above two requirements are controlled with **FLOW CONTROL** and **ERROR CONTROL**.
- ❖ Collectively, these two functions are known as **data link control**,

---



---



---



---



---



---

## FLOW CONTROL

---

<http://www.ccs-labs.org/teaching/rn/animations/flow/index.htm>

---

---

---

---

---

---

### **Flow Control : Introduction**

- ❖ Flow Control is a technique in which
  - Sender and receiver can communicate with each other with different speed characteristics.
- ❖ Flow control ensures
  - Sender with higher processing capability does not overwhelm a Receiver with lesser processing capability.

---

---

---

---

---

---

### **Flow Control : Introduction**

- ❖ Flow control achieves
  - orderly flow of transmitted data between the source and the destination.
- ❖ Finally, Flow control refers to the **set of procedures** used
  - To restrict the amount of data that the sender can send before waiting for acknowledgment.

---

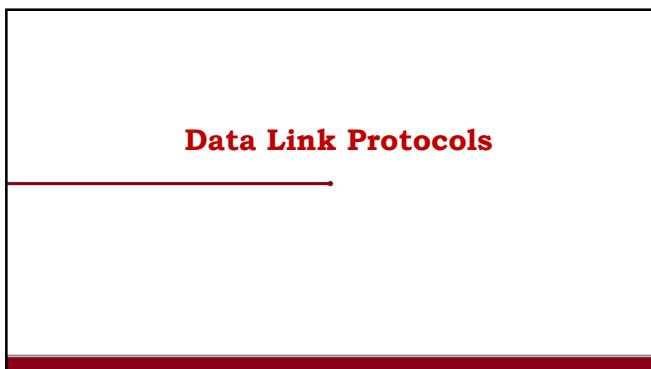
---

---

---

---

---



## Data Link Protocols

---

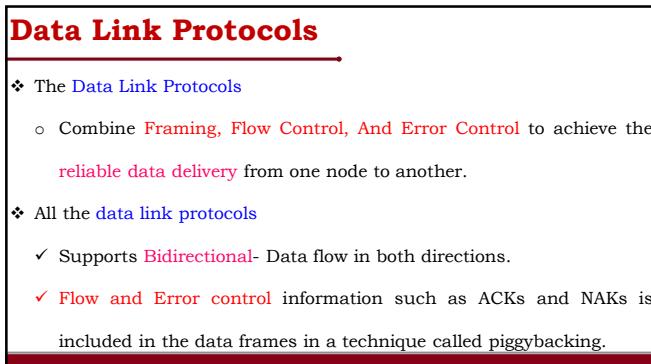
---

---

---

---

---



### Data Link Protocols

- ❖ The Data Link Protocols
  - Combine **Framing, Flow Control, And Error Control** to achieve the reliable data delivery from one node to another.
- ❖ All the **data link protocols**
  - ✓ Supports **Bidirectional**- Data flow in both directions.
  - ✓ **Flow and Error control** information such as ACKs and NAKs is included in the data frames in a technique called piggybacking.

---

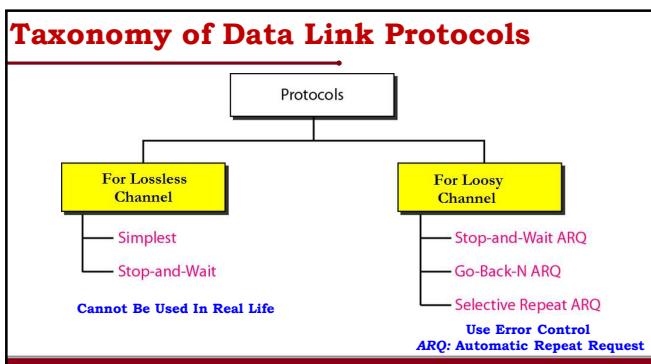
---

---

---

---

---




---

---

---

---

---

---

## Data Link Layer: Flow Control **ARQ Protocols**

---

---

---

---

---

---

### **ARQ Protocols**

- ❖ ARQ stands for **Automatic Repeat reQuest**.
- ❖ In ARQ, an **error control** method is incorporated with **flow control** protocols

---

---

---

---

---

---

### **Types of ARQ Protocols**

- ❖ We will study **three ARQ strategies**:

  1. **Stop And Wait ARQ**,
  2. **Go Back N ARQ**
  3. **Selective Repeat ARQ**

---

---

---

---

---

---

## Stop And Wait ARQ Protocols

---



---

---

---

---

---

---

### Simplex Stop and Wait with ARQ

- ❖ For Noiseless Link,
  - ✓ Pure Stop and Wait protocol will Work
- ❖ For Noisy link,
  - ✓ Pure Stop and Wait protocol will break down
  - ✓ Incorporate **some error control mechanism** to have better solution

---

---

---

---

---

---

### Simplex Stop and Wait with ARQ

- ❖ In this protocol,
- ✓ Sender ensures that **each packet is received correctly** before framing the next packet sending it.
- ✓ Sender sends a **packet in frame with CRC**, then **waits** for an **ACK (acknowledgment)** or **NAK (negative acknowledgment)**.

---

---

---

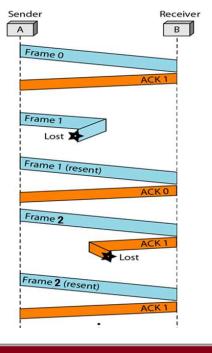
---

---

---

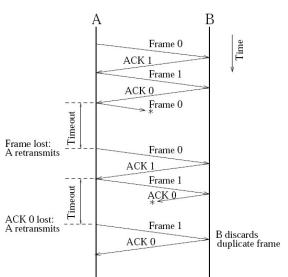
## Stop and Wait with ARQ

- ❖ If error is detected by receiver,
    - ✓ It discards the frame and send a negative ACK (NAK), causing sender to re-send the frame



## **Stop and Wait with ARQ : Timer**

- ❖ What happens if a **frame** never received by the receiver,
    - ✓ Sender will maintain **Timer**
    - ✓ Each time a **frame is sent**, **Timer** is set
    - ✓ If **no ACK or NAK is received** during **timeout period**, it **re-sends the frame**

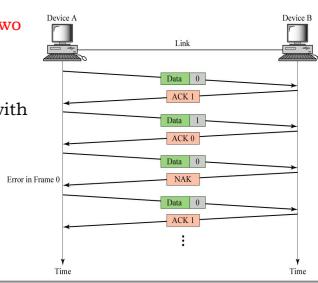


### **Timer introduces a problem**

- ❖ Suppose timeout and sender retransmits a frame but receiver actually received the previous transmission
    - ✓ Receiver has **duplicated copies**

## Stop and Wait with ARQ : Sequence Numbers

- ❖ To avoid receiving and accepting two copies of same frame,
- ✓ Frames and ACKs are labeled with sequence numbers



## Limitation

- ❖ Only one frame can be in transmission at a time, **Link Utilization is clearly wasteful**
- ✓ Leads to inefficiency if propagation delay is much longer than the transmission delay.

**Transmission time:** The time it takes for a station to transmit a frame  
**Propagation delay:** The time it takes for a bit to travel from sender to receiver

## Data Link Layer: Flow Control Sliding Window Protocols

[http://www.ccs-labs.org/teaching/rn/animations/gbn\\_sr/](http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/)

## Sliding Window Protocols

- ❖ To improve the link utilization,
- ✓ we can use the **Sliding-window Protocol** which is a universally accepted flow control protocol

---



---



---



---



---



---

## Sliding Window Protocols

- ❖ Sliding Window Protocols provides
- ✓ Suppose a single message may contain multiple frames. There is a need to transit the multiple frames at the same time.
- ✓ Efficiency can be improved by making use of the full-duplex line effectively .

---



---



---



---



---



---

## Sliding Window Protocols

- ❖ Frames and Acknowledgements are numbered using Sequence Numbers
- ❖ Sender maintains a list of sequence numbers (frames) it is allowed to transmit, called **SENDING WINDOW**
- ❖ Receiver maintains a list of sequence numbers it is prepared to receive, called **RECEIVING WINDOW**

---



---



---



---

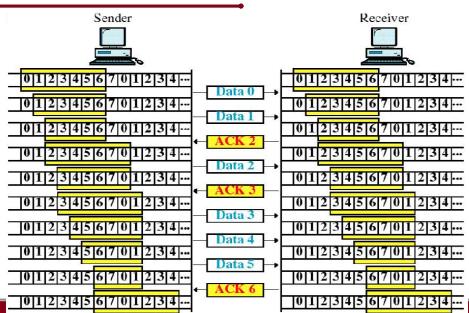


---



---

## Sliding Window Protocols



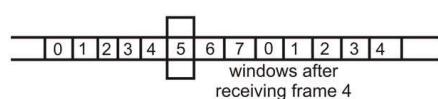
## Sending Window

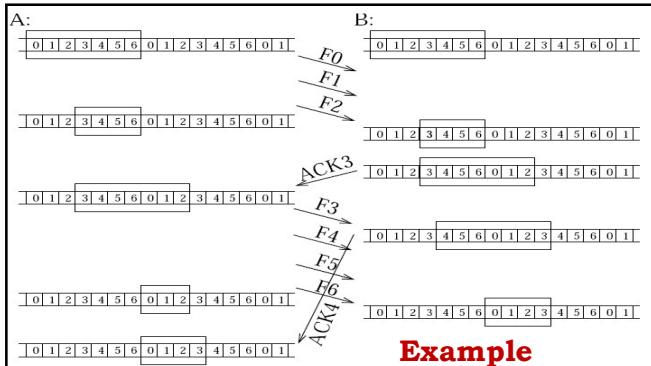
- ❖ A sending window of size N
  - ✓ Sender can send up to N frames without the need for an ACK
- ❖ A window size of N implies buffer space for N frames



## Receiving Window

- ❖ The receiver always maintains a window of size 1
- ❖ ACK5 means that receiver has received frame 0 to frame 4 correctly, ready to receive frame 6





### Example

- ❖ Consider the case of 3-bit sequence number with maximum window size  $N = 7$
- ❖ Sending and receiving windows can shrink or grow during operation
- ❖ The receiver do not need to acknowledge every frames
- ❖ If both sending and receiving window sizes are  $N = 1$ , the sliding window protocol reduces to the stop-and-wait

### Sliding Window Protocols

- ❖ In practice, error control must be incorporated with flow control,
- ❖ Two common error control mechanisms in ARQ

1. Go back N ARQ
2. Selective Repeat ARQ

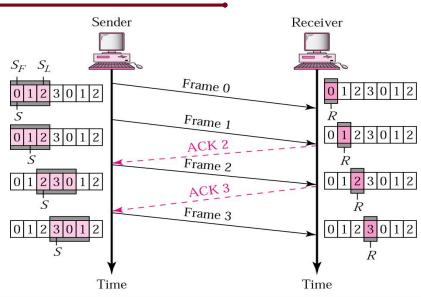
## Sliding Window Protocols Go Back and N Protocol

[http://www.ccs-labs.org/teaching/rn/animations/gbn\\_sr/](http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/)

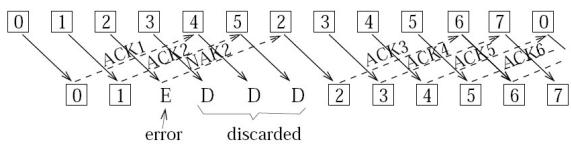
### Go back N ARQ Protocol

- ❖ The most popular ARQ protocol is the go-back-N ARQ. This Go back N ARQ is also called as continuous ARQ.
- ✓ Sender sends the frames continuously without waiting for acknowledgement.
- ✓ As the receiver receives the frames, it keeps on sending ACKs or a NACK(incorrectly received).

### Go back N ARQ Protocol

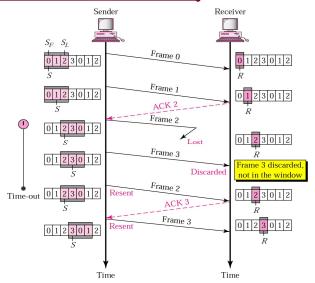


## Go back N ARQ Protocol

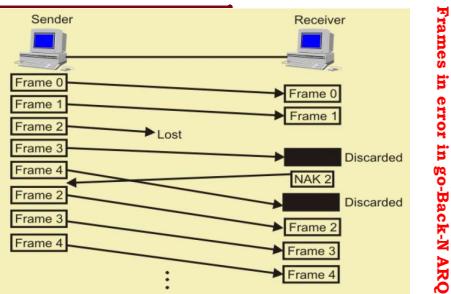


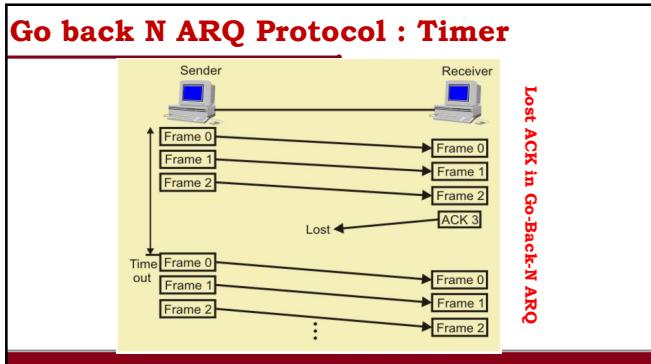
- ❖ When the sender receives a NACK,
    - ✓ it retransmits the frame in error plus all the succeeding frames as shown in Fig.

## Go back N ARQ Protocol



### **Go back N ARQ Protocol : Timer**





### Limitation.

- ❖ One potential weakness of the go back n ARQ protocol
- ✓ When one packet in a window is dropped by the link, subsequent **packets are not accepted by the receiver.**

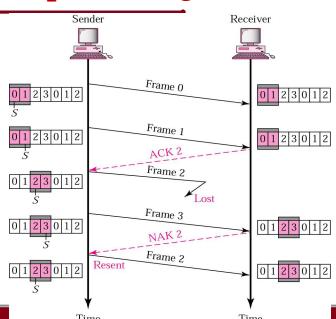
## Sliding Window Protocols Selective-Repeat ARQ

[http://www.ccs-labs.org/teaching/rn/animations/gbn\\_sr/](http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/)

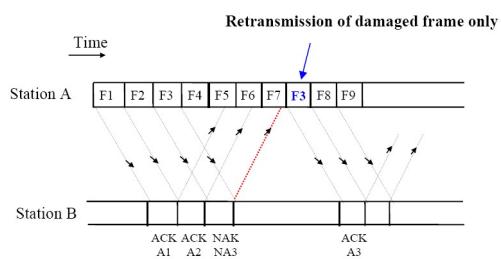
## Selective-Repeat ARQ

- The selective-repetitive ARQ scheme retransmits only NAKs or timer has expired.
- This is the **most efficient among the ARQ schemes**,

## Selective-Repeat ARQ



## Selective-Repeat ARQ



## Comparison

❖ **Go-back-N ARQ:**

- ✓ Simplicity in buffering and protocol processing at sender and receiver.
- ✓ If receiver discards bad or out of order messages, no receiver buffering is required.
- ✓ More retransmission traffic with unreliable network.

❖ **Selective repeat ARQ:**

- ✓ More receiver buffering required.
- ✓ More complex buffer management for both sender and receiver.
- ✓ Saves bandwidth: avoids retransmission of correct frames.

---



---



---



---



---



---

## Trade-Off

- ❖ Go-back-n and selective-Repeat can be seen as trade-offs between **link bandwidth (data rate)** and **data link layer buffer space**
- If link bandwidth is large but buffer space is limited , **Go-Back-N is preferred**
  - If link bandwidth is small but buffer space is pretty, **selective-Repeat is preferred**

---



---



---



---



---



---

## Goals:

❖ **Elementary Data Link Protocols**

- ✓ Flow Control
- ✓ Error Control

❖ **Data Link Protocols**

- ✓ Simplex Stop-and-Wait Protocol
- ✓ ARQ Protocols

---



---



---



---



---



---

**Thank You**

## CCN: Data Link Layer: Medium Access Control

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Goals:

- ❖ The Medium Access Sublayer
- ❖ Types of MAC Protocols
  - ✓ Channel Partitioning MAC Protocols
  - ✓ Multiple Access Protocols

### The Medium Access Sublayer

## Data Link Layer – Accessing the Media

- ❖ The Data Link layer performs two basic services:
  - ✓ Allows the upper layers to access the media using techniques such as framing
  - ✓ Controls how data is placed onto the media and is received from the media
    - Using **Media Access Control Technique**

---



---



---



---



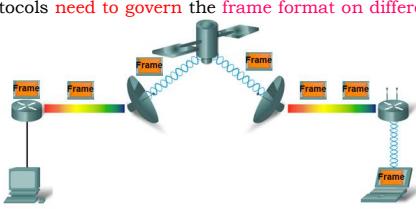
---



---

## Data Link Layer – Accessing the Media

- ❖ Data Link layer protocols are required to control media access
- ❖ Data Link layer protocols need to govern the frame format on different Media




---



---



---



---



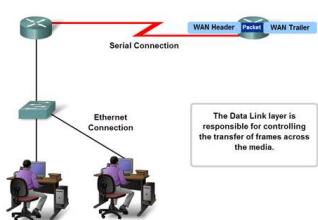
---



---

## Data Link Layer – Accessing the Media

- ❖ Role of framing in preparing a packet for transmission on a given media




---



---



---



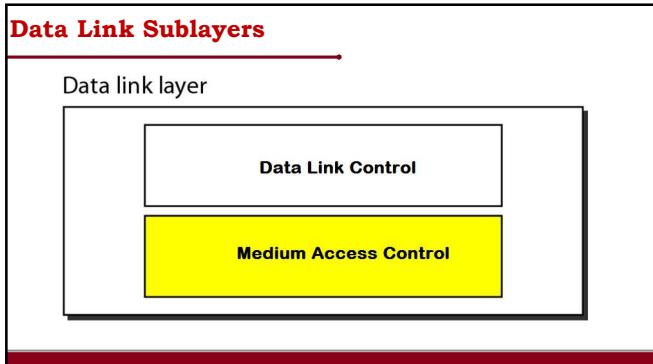
---



---



---



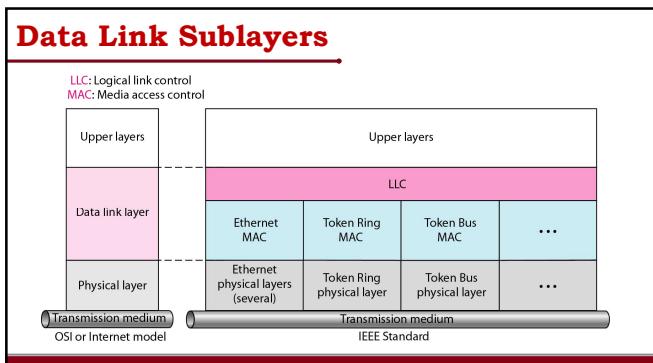

---

---

---

---

---



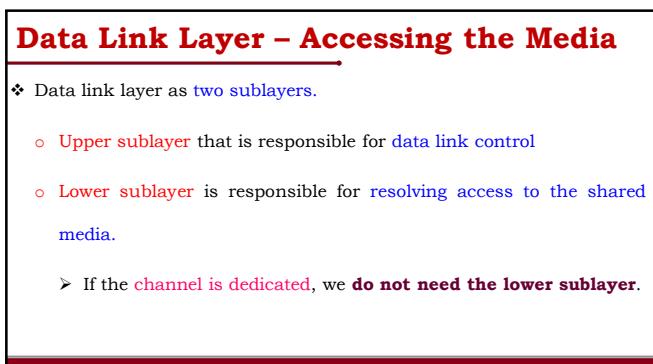

---

---

---

---

---




---

---

---

---

---

## Data Link Layer – Accessing the Media

❖ **Logical Link Control**

- LLC places information in the frame that identifies which Network layer protocol is being used for the frame.
- This sublayer is responsible for flow and error control;

---



---



---



---



---



---

## Data Link Layer – Accessing the Media

❖ **Media Access Control** provides

- Data Link layer addressing
- Delimiting of data according to the physical signaling requirements of the medium
- Type of Data Link layer protocol in use.

❖ MAC is responsible for **multiple access**

---



---



---



---



---



---

## Introduction to MAC Sublayer

❖ As we discussed already in **Transmission Technology**,

- Usually networks can be divided into two categories:

- 1. Point-to-Point connections**
- 2. Broadcast channels.**

❖ Here we mainly concentrates with **broadcast networks and their protocols.**

---



---



---



---



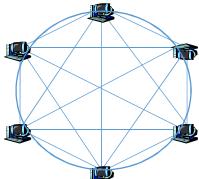
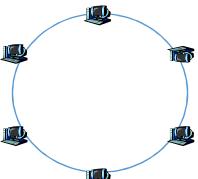
---



---

**Observation.** It's all about resource allocation

- ❖ Why to share access to a common medium

Fully-connected links   Shared Broadcast Medium

---



---



---



---



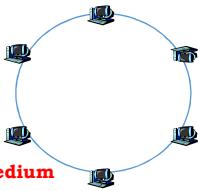
---



---

**Introduction to MAC Sublayer**

- ❖ In any broadcast network, the key issue is
  - How to determine who gets to use the channel when there is competition for it.



Shared Broadcast Medium

---



---



---



---



---



---

**Introduction to MAC Sublayer**

- ❖ In many broadcast network environments,
  - ✓ The transmission links are **shared by multiple users.**
- ❖ Broadcast channels are sometimes referred to as **Multi-Access Channels Or Random Access Channels.**
- ❖ Examples of multiple-access environments
  - ✓ Ethernet, Token Ring and Wireless Networks

---



---



---



---



---



---

## Introduction to MAC Sublayer

- ❖ MAC (Medium Access Control) sublayer protocols are responsible for determining
  - ✓ who gets next on a multi-Access channel
- ❖ The MAC sublayer is especially important in LANs, However, WANs uses point-to-point links,

---



---



---



---



---



---

## Types of MAC Protocols

- ❖ How to share access to a common medium
  - ✓ Two Ways to Share the Media
    1. Channel Partitioning MAC protocols
    2. Random access MAC protocols

---



---



---



---



---



---

## Channel Partitioning MAC Protocols

---



---



---



---



---



---

## The Channel Allocation Problem

- ❖ To allocate a **single broadcast channel** among competing users.

1. **Static Channel Allocation**

2. **Dynamic Channel Allocation**

---

---

---

---

---

---

### Static Channel Allocation

---

---

---

---

---

---

## Static Channel Allocation

- ❖ **Static Channel Allocation** are applied both in **LANs** and **MANs**.
- ❖ **Static Channel** allocates a **single channel** among **multiple competing users** using
  - **Frequency Division Multiplexing (FDM)**.

---

---

---

---

---

---

## Static Channel Allocation

- ❖ If there are  $N$  users, the bandwidth is divided into  $N$  equal-sized portions
  - Each user is assigned one portion.
  - Each user has a private frequency band, there is no interference between users.

---



---



---



---



---



---

## Static Channel Allocation

### Advantages

- ❖ Suitable for fixed number of users with constant traffic.

### Disadvantage:

- ❖ when the number of users is large and continuously varying, or the traffic is bursty
- ✓ FDM presents some problems such as Wastage of Bandwidth and Lack of Bandwidth of some users

---



---



---



---



---



---

## Dynamic Channel Allocation

---



---



---



---



---



---

## Dynamic Channel Allocation

- ❖ Dynamic channel allocation methods work with five key assumptions
  1. Station Model.
  2. Single Channel Assumption.
  3. Collision Assumption
  4. Continuous Vs Slotted Time.
  5. Carrier Sense Vs Non-Carrier Sense

---



---



---



---



---



---

## Station Model

- ❖ This model consists of N independent stations,
- ❖ Each Station generates frames for transmission.
- ❖ Once a frame has been generated,
  - The station is blocked and does nothing until the frame has been successfully transmitted.

---



---



---



---



---



---

## Single Channel Assumption

- ❖ A single channel is available for all communication.
- ❖ All stations can transmit on it and all can receive from it.
- ❖ As far as the hardware is concerned,
  - All stations are equivalent,
- ❖ As far as the software is concerned,
  - Some protocol may assign priorities

---



---



---



---



---



---

### Collision Assumption

- ❖ If two frames are transmitted simultaneously,
  - Frame will be overlap in time and the resulting signal is garbled.
  - This event is called a **COLLISION**.
- ❖ All stations can detect collisions.
  - ✓ A collided frame must be transmitted again later.

---



---



---



---



---



---

### Continuous Vs Slotted Time

- ❖ In **Continuous Time**.
  - Frame transmission can begin at any instant.
  - No Division of time
- ❖ In **Slotted Time**.
  - Time is divided into discrete intervals (slots).
  - Frame transmissions always begin at the start of a slot.
  - A slot may contain 0, 1, or more frames, depending upon idle slot

---



---



---



---



---



---

### Carrier Sense Vs No Carrier Sense

- ❖ In **Carrier Sense**.
  - Stations can tell if the channel is in use before trying to use it.
  - If the channel is sensed as busy,
    - ✓ no station will attempt to use it until it goes idle.

---



---



---



---



---



---

## Carrier Sense Vs No Carrier Sense

- ❖ In **No Carrier Sense**.
  - Stations cannot sense the channel before trying to use it.
  - Stations just follows **go ahead and transmit**.
  - Only later can they determine whether or not the transmission was successful.

---



---



---



---



---



---

## Multiple Access Protocols

Schemes for Sharing a Communication Medium

---



---



---



---



---



---

## Multiple Access Protocols

- ❖ Some applications want to **broadcast messages to all stations** on the LAN
- ❖ Shared communication channel can make broadcast efficient message is delivered to all stations

---



---



---



---



---



---

## Multiple Access Protocols

- ❖ In Multiple Access/random access/Contention methods
  - ✓ No station is **superior** to another station
  - ✓ **None is assigned** the control over another.
  - ✓ No station **permits/does not permit** another station to send.
- ❖ A **station that has data to send** uses a **procedure defined by the protocol** to make a **decision on whether to send or not.**

---



---



---



---



---



---

## Multiple Access Protocols

- ❖ Multiple Access Protocols does **not have scheduled time** for a station to transmit.
  - Transmission is **random among the stations**. Therefore, these methods are called **Random Access**.

---



---



---



---



---



---

## Multiple Access Protocols

- ❖ Multiple Access Protocols **does not specify which station** should send next.
  - Stations **compete with one another** to access the medium. Therefore, these methods are also called **contention methods**.

---



---



---



---

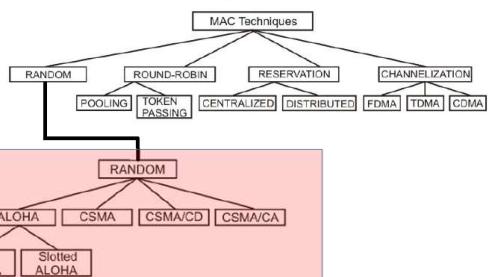


---



---

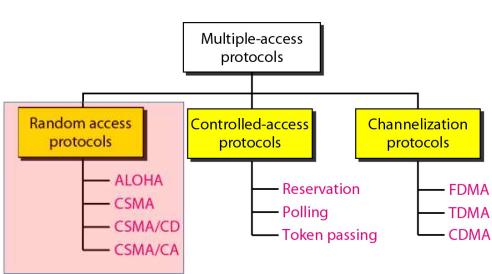
## Various Multiple Access Protocols



## Goals:

- ❖ The Medium Access Sublayer
- ❖ Types of MAC Protocols
  - ✓ Channel Partitioning MAC Protocols
  - ✓ Multiple Access Protocols**
  - ✓ Types of Multiple Access Protocols**

## Types of Multiple Access Protocols



## ALOHA

---



---

---

---

---

---

---

### ALOHA

---

- ❖ In the 1970s, a protocol called ALOHA was developed for a wireless system at the University of Hawaii.
- ❖ ALOHA was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- ❖ The original ALOHA protocol is called pure ALOHA which is a simple and elegant protocol
- ❖ Pure ALOHA is truly a free-for-all scheme

---

---

---

---

---

---

### Pure ALOHA

---

- ❖ In Pure ALOHA
  - ✓ Multiple Remote Stations and
  - ✓ One Base (Central) Station.
- ❖ The idea of Pure ALOHA
  - ✓ Each station sends a frame whenever it has a frame to send

---

---

---

---

---

---

## ALOHA

- ❖ Frame transmissions are made at one frequency from a remote station to the base station;
- ❖ Next, the base station re-broadcasts frames on another frequency.

---



---



---



---

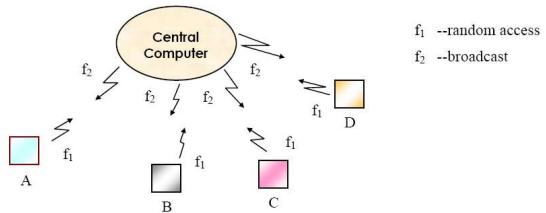


---



---

## Pure ALOHA




---



---



---



---



---



---

## Pure ALOHA

- ❖ Frames are transmitted at completely arbitrary times.
- ❖ Stations transmit the fixed length frames on a common channel
- ❖ when two transmissions overlap, they collide each other
- ❖ The central Station acknowledges the correct frames, as it receives from the station
- ❖ when a Station does not get an acknowledgment within a specific timeout, it assumes that its frame collided.

---



---



---



---

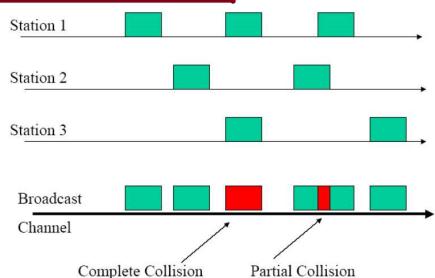


---

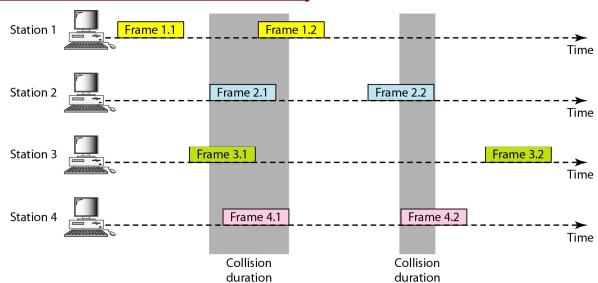


---

### Pure ALOHA



### Pure ALOHA-Another Example



### Limitation with Pure ALOHA

- ❖ The best channel utilization of **18 percent only**.
- ❖ This method is **inefficient**, When **everyone transmitting at will**,
- ❖ **High Throughput** was also not expected.

## Slotted ALOHA

---

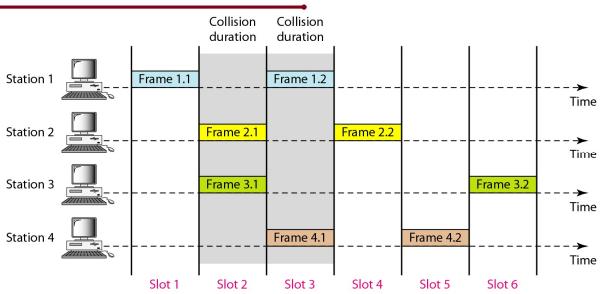
### Slotted ALOHA

---

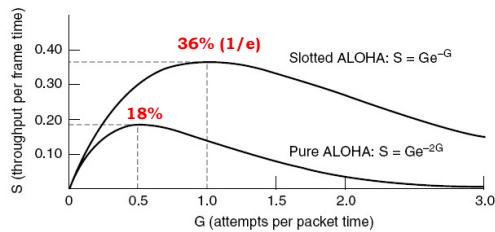
- ❖ Slotted ALOHA was suggested by Roberts(1972) to improve upon the efficiency of pure ALOHA.
  - ❖ In slotted ALOHA
    - ✓ we divide the **time into slots** and **force the station to send only at the beginning of the time slot**.
    - ✓ This approach requires the **users to agree on slot boundaries**.
    - ✓ **Synchronization** can be achieved with a clock
- 

### Slotted ALOHA

---



### Channel Utilization of Pure and Slotted Aloha



### Limitation

- ❖ The poor efficiency of the ALOHA scheme is that
  - ✓ A node starts transmission without paying any attention to what others are doing

### Carrier Sense Multiple Access (CSMA)

## Introduction

- ❖ To minimize the chance of collision and increase the performance
  - ✓ CSMA method was developed.
- ❖ The chance of collision can be reduced
  - ✓ if a station senses the medium before trying to use it.

---



---



---



---



---



---

## Introduction

- ❖ In Carrier sense multiple access (CSMA)
  - ✓ Each station first listen to the medium (or check the state of the medium) before sending.
- ❖ CSMA Works is based on the principle "Sense Before Transmit" or "Listen Before Talk."
- ❖ CSMA can reduce the possibility of collision, but it cannot eliminate it.

---



---



---



---

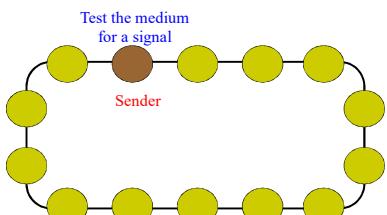


---



---

## Carrier Sensing




---



---



---



---

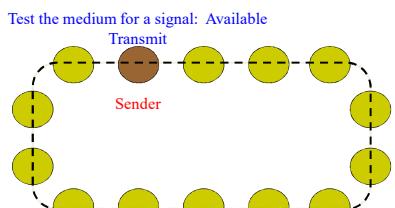


---



---

### Carrier Sensing



---

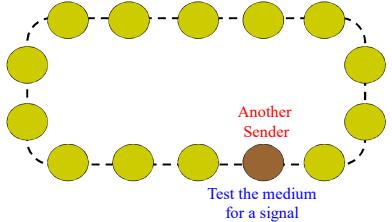
---

---

---

---

### Carrier Sensing



---

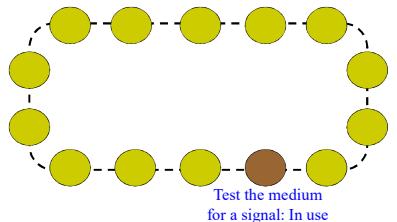
---

---

---

---

### Carrier Sensing



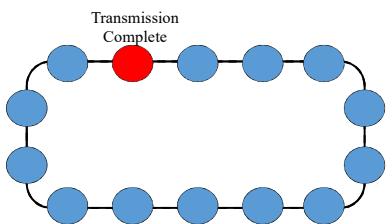
---

---

---

---

---

**Carrier Sensing**

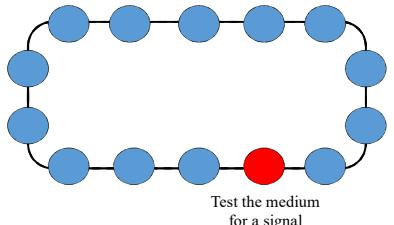
---

---

---

---

---

**Carrier Sensing**

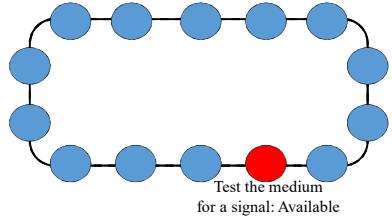
---

---

---

---

---

**Carrier Sensing**

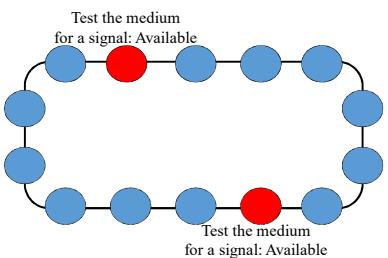
---

---

---

---

---

**Carrier Sensing: Collisions**

---

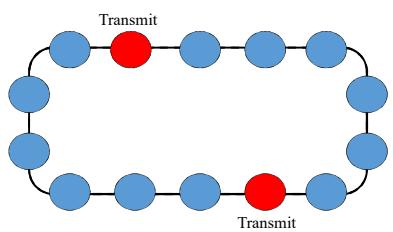
---

---

---

---

---

**Carrier Sensing: Collisions**

---

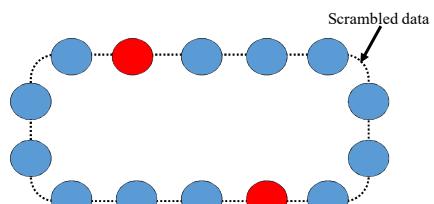
---

---

---

---

---

**Carrier Sensing: Collisions**

---

---

---

---

---

---

### Observation

- ❖ What should a station do if the channel is busy?
- ❖ What should a station do if the channel is idle?

---

---

---

---

---

---

### Types of CSMA (Persistence Methods)

---

---

---

---

---

---

### Types of CSMA

- ❖ There are three variations of this CSMA scheme
- 1. 1-Persistent CSMA
- 2. Non-Persistent CSMA
- 3. P-Persistent CSMA

---

---

---

---

---

---

## 1-Persistent CSMA

❖ In **1-Persistent CSMA**

- ✓ A **node having data to send**. It **starts sending** only,
  - if the **channel is sensed free**.
- ✓ If the **medium is busy**,
  - The **node continues to monitor** until the **channel is idle**.  
Then it **starts sending data**.

---



---



---



---



---

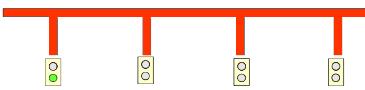


---



---

## Monitoring the Transmission Medium



- ❖ When the **medium is detected to be in use at each station**, that station cannot transmit.

---



---



---



---



---

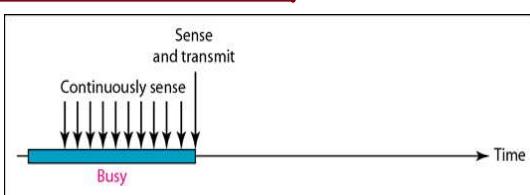


---



---

## 1-Persistent CSMA



- ❖ Stations listen to the **channel** continuously  
❖ If **channel is busy** wait till free

---



---



---



---



---



---



---

### 1-Persistent CSMA

- ❖ The protocol is called **1-persistent**
  - ✓ The station **transmits with a probability of 1** when it finds the **channel idle**.
- ❖ If a **collision occurs**,
  - ✓ The station waits a **random amount of time** and **starts all over again**.

---



---



---



---



---

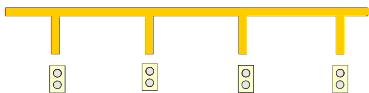


---



---

### Collision due to Propagation Delay




---



---



---



---



---



---



---

### Observations

- ❖ This protocol is far better than **pure ALOHA**.
- ❖ Intuitively, this approach will **lead to a higher performance** than pure ALOHA.
- ❖ Exactly the same holds for **slotted ALOHA**.

---



---



---



---



---



---



---

## Non-Persistent CSMA

- ❖ A second carrier sense protocol is **Non-Persistent CSMA**.
- ❖ In this **protocol**,
- ✓ Before sending, a **station** **senses** the channel.
- ✓ If **no one else is sending**, the station **begins sending the frames**

---



---



---



---

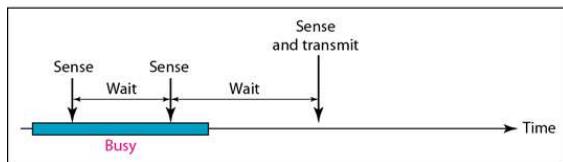


---



---

## Non-Persistent CSMA




---



---



---



---



---



---

## Non-Persistent CSMA

- ❖ if the **channel** is already in use
  - ✓ the station does **not continually sense** it
  - ✓ Instead, it waits a **random period of time** and then repeats the algorithm.
- ❖ Consequently, this algorithm leads to **better channel utilization** but **longer delays** than 1-persistent CSMA.

---



---



---



---



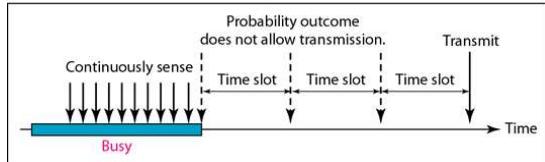
---



---

## P-Persistent CSMA

- ❖ In this P-persistent CSMA applies to slotted channels
  - ❖ If the channel is free,
    - ✓ a node starts sending the packet.
    - ✓ Otherwise the node continues to monitor until the channel is free and then it sends with probability p.



## P-Persistent CSMA

- ❖ With a probability  $q = 1 - p$ ,
    - ✓ It waits until the next slot.
    - ✓ If that slot is also idle, it either transmits or waits again, with probabilities  $p$  and  $q$ .
  - ❖ This process is repeated until either the frame has been transmitted or another station has begun transmitting.

## CSMA with Collision Detection

### Introduction to CSMA/CD

- ❖ CSMA/CD protocol is a refinement to CSMA scheme.
  - ❖ CSMA/CD is used to overcome **one inefficiency of CSMA**.
    - In CSMA scheme,
    - ✓ when **two packets collide the channel remains unutilized for the entire duration of transmission time** of both the packets.
- $a = \frac{\text{Propagation delay}}{\text{Packet transmission time.}}$

### Introduction to CSMA/CD

- ❖ If the **propagation time is small compared to the packet transmission time**,
  - Wasted channel capacity can be **considerable**.
  - This **wastage of channel capacity** can be **reduced**
    - ✓ if the **nodes continue to monitor the channel** while transmitting a packet

## Introduction to CSMA/CD

- ❖ If a collision is detected during transmission of a packet,
  - The node immediately terminate transmission
  - Same Node transmits jamming signal to ensure that all stations know that collision has occurred.
  - After transmitting the jamming signal, the node waits for a random amount of time and then transmission is resumed.

---



---



---



---

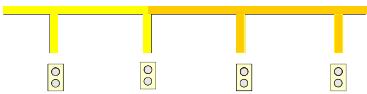


---



---

## Ensuring Collision Detection




---



---



---



---



---



---

## Introduction to CSMA/CD

- ❖ This refined scheme is known as Carrier Sensed Multiple Access with Collision Detection (CSMA/CD) or Listen-While-Talk.

---



---



---



---

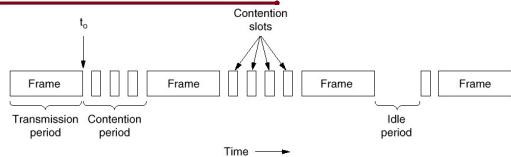


---



---

## A Conceptual Model for CSMA/CD



- ❖ Two types of slots, **packet slots and contention slots**
- ❖ Each **node tries to transmit** at a **contention slot** with probability  $p$
- ❖ If collision, it is **detected at the end of contention slot**
- ❖ If success, **no transmission tries till end of packet slot**

## Limitations

- ❖ However, Collisions do not occur with CSMA/CD
- ❖ Collisions can still occur during the contention period which **affect the system performance**
- ✓ When the **cable is long and the frames are short**.

## Goals:

- ❖ The Medium Access Sublayer
- ❖ Types of MAC Protocols
  - ✓ Channel Partitioning MAC Protocols
  - ✓ Multiple Access Protocols

**Thank You**

## CCN: Introduction

Dr. E.SURESH BABU  
Assistant Professor

Computer Science and Engineering Department  
National Institute of Technology, Warangal.  
Warangal, TS, India.



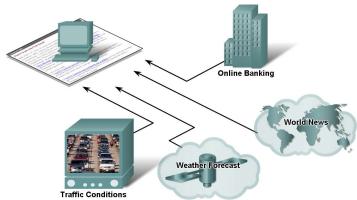
### Goals:

- ❖ How Networks Impact Daily Life
- ❖ Networking is Relevant
- ❖ Why Learn about Networking (Internet)?
- ❖ Basics of Computer Network?
- ❖ Computer Network Applications
- ❖ Internet Usage Report
- ❖ Internet Structure: Network of Networks

**Networking is Today's World**

## How Networks Impact Daily Life

- Benefits of **instantaneous communication** and how it supports and improves our lives.




---

---

---

---

---

---

## How Networks Impact Daily Life....

- Characteristics and **purpose of popular communication** media such as, **IM, Blogs, Podcasting, and Collaboration Tools**

- Instant messaging** : Real time communication between two or more people based on typed text
- Weblogs (Blogs)**: Web pages created by an individual
- Podcasting**: Website that contains audio files available for downloading




---

---

---

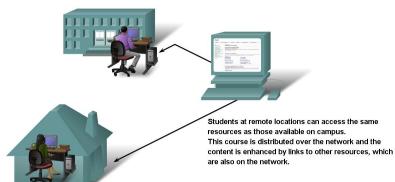
---

---

---

## How Networks Impact Daily Life....

- Using **information networks** to share and collaborate improves **teaching and learning**




---

---

---

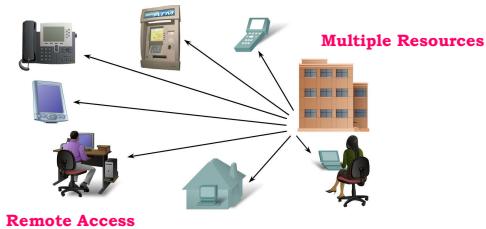
---

---

---

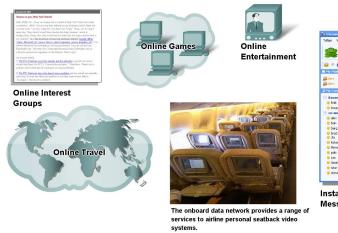
## How Networks Impact Daily Life....

- Ways of communication over a network changes the way we work



## How Networks Impact Daily Life....

- Ways communication over a network supports the way we play



**Networking is Relevant**

## Current Trend in Networking

- ❖ Indispensable part of modern society
- ✓ **Commercial:** E-commerce, Banking, Inventorying, Telecommunications , Archiving, Health
- ✓ **Social:** Critical Infrastructure, Homeland Security, Policing
- ✓ **Human Interaction/Communication:** Email, Chat, Videoconferencing, Social Networking, Entertainment

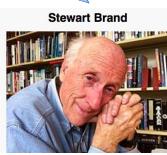
## Current Trend in Networking

Information wants to be free because it has become so cheap to distribute, copy, and recombine... It wants to be expensive because it can be immeasurably valuable to the recipient.

(1985)



WIKIPEDIA





---

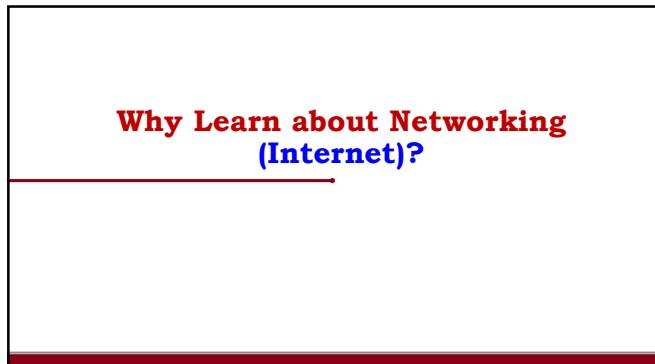
---

---

---

---

---



---

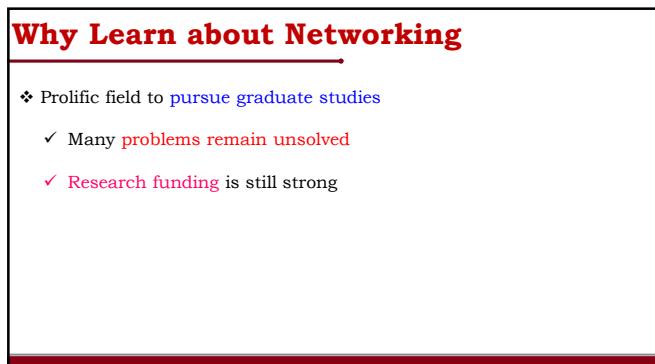
---

---

---

---

---



---

---

---

---

---

---

## Basics of Computer Network




---

---

---

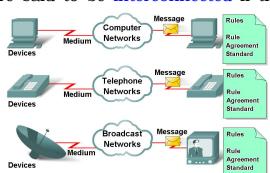
---

---

---

### Definition of a Computer Network

- ❖ A System that carries information between Two or more entities, in the form of electric signals
- ❖ Two computers are said to be interconnected if they are able to exchange information




---

---

---

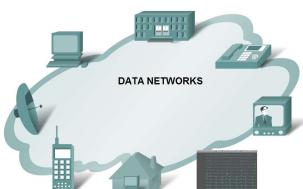
---

---

---

### Definition of a Computer Network.....

- ❖ A Computer Network can provide a powerful communication medium among Various Devices.



✓ Imagine your life with out any of these convenience

---

---

---

---

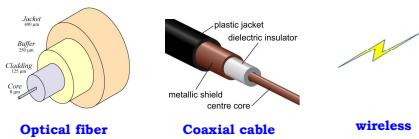
---

---

### Network Elements.....



### Network Elements.....



### Advantages of Computer Networks

- ❖ Important benefits of computer networks are:
- ✓ Resource sharing
- ✓ Powerful communication medium
- ✓ Higher reliability
- ✓ Higher flexibility
- ✓ Lower cost
- ✓ Incremental expansion

## Computer Network Applications




---

---

---

---

---

## Computer Network Applications

- ❖ The main area of applications can be broadly classified into following categories:
  - ❖ **Scientific and Technical Computing**
    - ✓ Client Server Model, Distributed Processing
    - ✓ Parallel Processing, Communication Media
  - ❖ **Commercial**


    - ✓ Advertisement, Telemarketing, Teleconferencing
    - ✓ Worldwide Financial Services



---

---

---

---

---

## Computer Network Applications

- ❖ **Network for the People (Currently used applications)**
  - ✓ Telemedicine, Distance Education,
  - ✓ Access to Remote Information,
  - ✓ Person-to-Person Communication,
  - ✓ Interactive Entertainment



---

---

---

---

---

## **Other Computer Network Applications**

#### ❖ Marketing and Sales:

- ✓ Marketing professionals use them to collect, exchange, and analyze data related to customer needs and product development cycles.
  - ✓ Sales application includes teleshopping, online-reservation services for hotels, airlines and so on.



## **Other Computer Network Applications**

#### ❖ Financial Services:

- ✓ Application includes credit history searches, foreign exchange and investment services, and electronic fund transfer.



## **Other Computer Network Applications**

#### ❖ Manufacturing:

- ✓ Computer networks essentially provides two services are computer-aided design (CAD) and computer-assisted manufacturing (CAM)
  - ✓ Allow multiple users to work on a project simultaneously



## Other Computer Network Applications

### ❖ Directory services:

- ✓ Allow **list of files to be stored** in **central location** to speed worldwide search operations.



### ❖ Information Services:

- ✓ Bulletin boards and Data banks.
- ✓ A **World Wide Web site** offering technical specification for a **new product** is an information service.

## Other Computer Network Applications

### ❖ Electronic mail:

- ✓ Probably **E-mail** the most widely used computer network application.

### ❖ Teleconferencing:

- ✓ Allows conference to occur without the participants being in the same place.



## Other Computer Network Applications

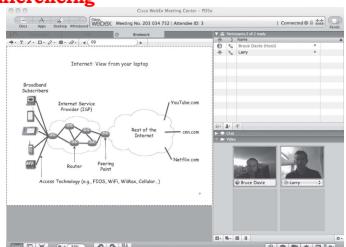
### ❖ Voice over IP:

- ✓ Computer networks are also used to provide **voice communication**.
- ✓ Pretty cheap as compared to the **normal telephonic conversation**.



## Other Computer Network Applications

### ❖ Video Conferencing



## Other Computer Network Applications

### ❖ Video on demand:

- ✓ Future services may include [video on request](#)
- ✓ where a person can [request the cable television networks](#) for a particular [movie or any clip at anytime](#)



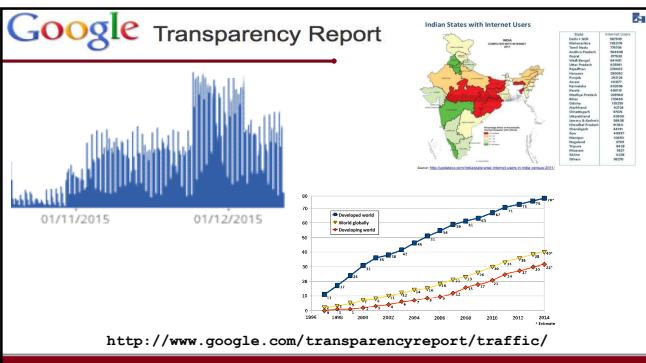
## More Computer Network Applications

### ❖ Appears in every facet of engineering

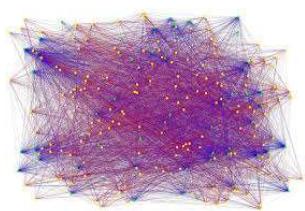
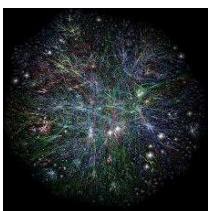
- ✓ **Modern Trend :** Network every (electronic) device (computers, phones, sensors, planes, cars, TVs, appliances, heart monitors, ...)



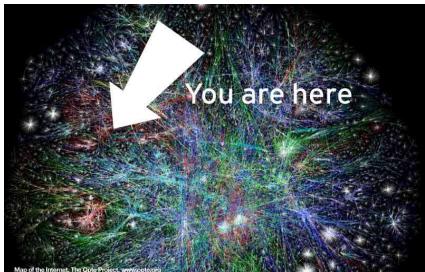
## Internet Usage Report:



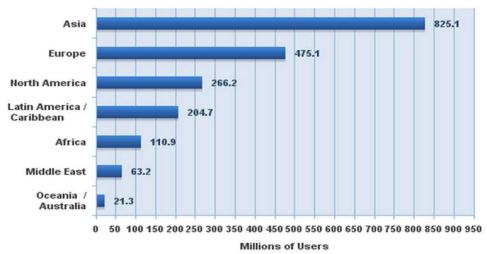
## How does the Internet Look Like?



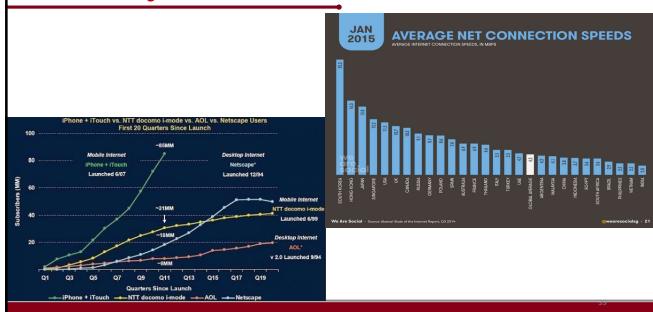
## How does the Internet Look Like?



## How Many Users?



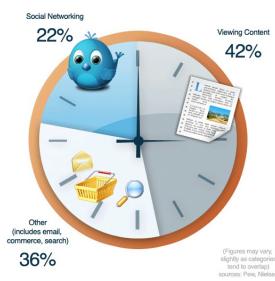
## How many more Users?



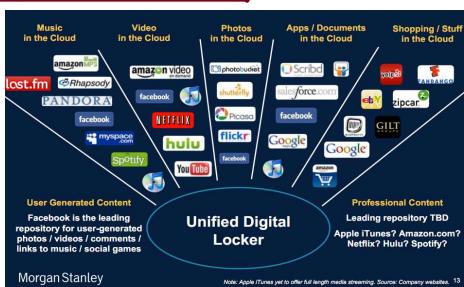
## How much Traffic?



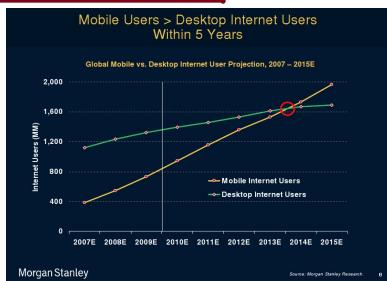
## How is Time Spent?



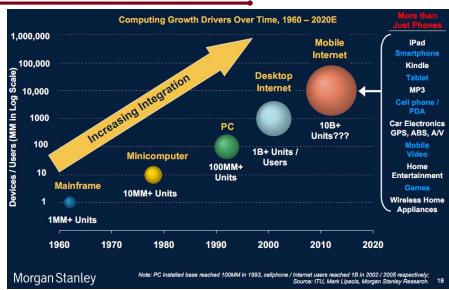
## What Do Users Expect?



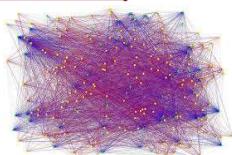
## How do they get it?



## Where are we headed?



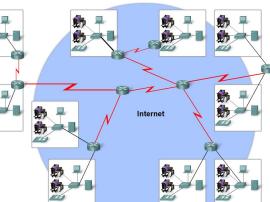
## Internet Structure: Network of Networks



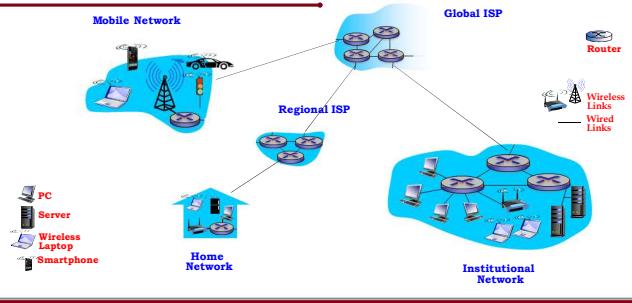
## The Internet

- Internet is a **collection of networks or network of networks**.

- It allows various applications such as **E-mail, File Transfer, Remote Log-in, World Wide Web, Multimedia**, etc run across the internet.



## The Internet



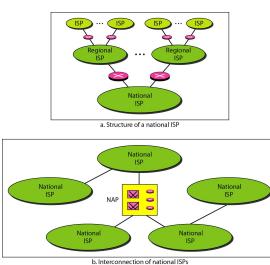
## The Internet

- End systems** connect to **Internet via access ISPs** (Internet Service Providers)

- Residential, Company And University ISPs**

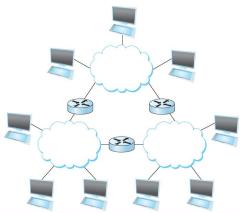
- Access ISPs** must be interconnected.

- So that any **two hosts can send packets to each other**



## Internet Structure: Network of Networks

- Resulting **network of networks** is very complex
- Evolution was driven by **economics and national policies**

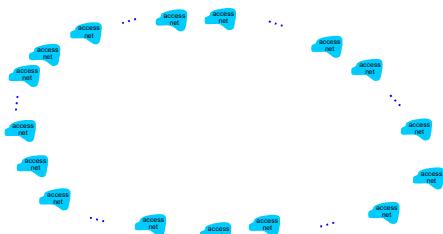


## Internet Structure: Network of Networks

- Let's take a **stepwise approach** to describe **current Internet structure**

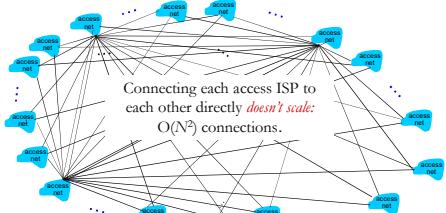
## Internet Structure

*Question:* given *millions* of access ISPs, how to connect them together?



## Internet Structure

*Option:* connect each access ISP to every other access ISP?



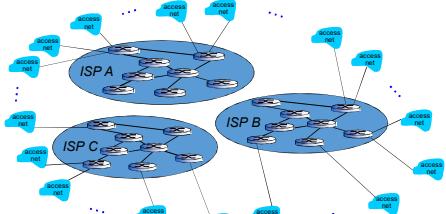
## Internet Structure

*Option:* connect each access ISP to a global transit ISP? *Customer and provider ISPs have economic agreement.*



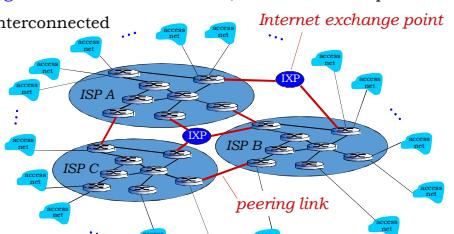
## Internet Structure

♦ But if **one global ISP** is viable business, there will be competitors ....



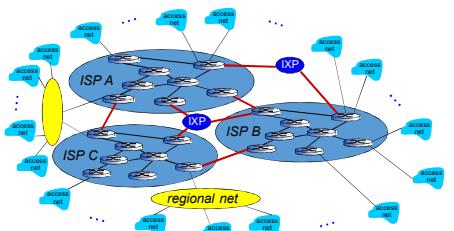
### Internet Structure

- But if one global ISP is viable business, there will be competitors .... which must be interconnected



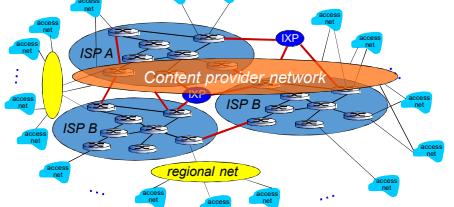
### Internet Structure

- ... and regional networks may arise to connect access nets to ISPS



### Internet Structure

- ... and content provider networks (e.g., Google, Microsoft) may run their own network, to bring services, content close to end users



**Goals:**

- ❖ How Networks Impact Daily Life
- ❖ Networking is Relevant
- ❖ Why Learn about Networking (Internet)?
- ❖ Basics of Computer Network?
- ❖ Computer Network Applications
- ❖ Internet Usage Report
- ❖ Internet Structure: Network of Networks

---

---

---

---

---

---

---

**Thank You**

---

---

---

---

---

---

---

## CCN: Design and Issues

**Dr. E.SURESH BABU**  
**Assistant Professor**

**Computer Science and Engineering Department  
National Institute of Technology, Warangal.  
Warangal, TS, India.**



---

---

---

---

---

---

---

---

---

---

## Goals:

- ❖ Computer Network Design
  - ❖ Network Issues
    - ✓ Communication Problem
    - ✓ Identification Problem
    - ✓ Connection Problem

---

---

---

---

---

---

# Computer Network Design



---

---

---

---

---

---

---

## Computer Network Design

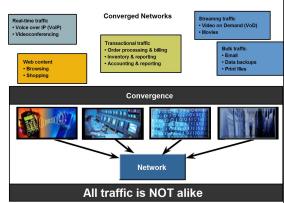
- ❖ Need to define the **Network Architecture, Protocols, Applications, Interfaces, Policies, Usages.**

### ❖ Who deploys the network

- Enterprise, government, end-user

### ❖ Where is the network deployed

- Home, building, campus, state, country, continent, globe



## Network Issues



## Computer Network Issues

- ❖ Some of the **Network issues** to be known while interconnecting with collection of autonomous computers

- **Communication Problem**
- **Identification Problem**
- **Connection Problem**

## Communication Problem



---

---

---

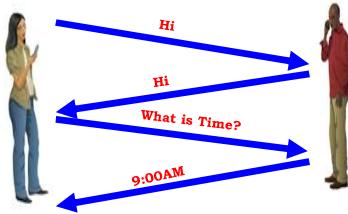
---

---

---

## Communication Problem (A Human Protocol )

- ❖ How communication can take place between Human Beings



---

---

---

---

---

---

## Communication Problem can be overcome in Computer Networks



---

---

---

---

---

---

## Solution....

- ❖ Communication between two computers done through the PROTOCOLS
- ❖ Protocols takes two (or more) communicating entities running the same protocol in order to accomplish a task
- ❖ Protocols that control the sending and receiving of information within the network




---

---

---

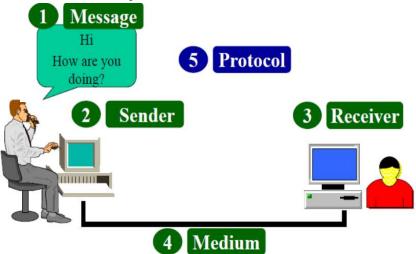
---

---

---

## Communication Problem

- ❖ Network Protocols used by Machines rather than Humans




---

---

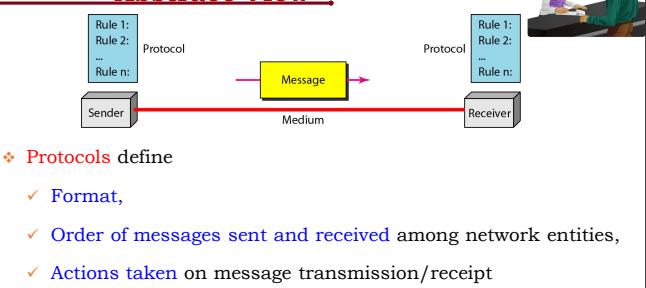
---

---

---

---

## Communication Problem: Abstract View




---

---

---

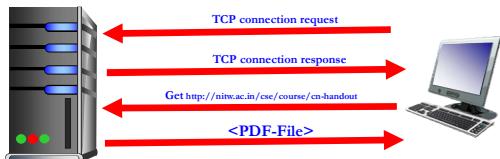
---

---

---

## Communication Problem(A Network Protocol)

- ❖ How **communication** can take place between **Network Entities**



## Various Protocols Used in Networking

- ❖ **Protocols** are **set of rules**. These **protocol standards** are proposed by **RFC (Request for Comments)**.
  - ✓ **FTP** → File Transfer Protocol
  - ✓ **HTTP** → Hyper Text Transfer Protocol
  - ✓ **SSH** → Secure Shell
  - ✓ **POP3** → Post Office Protocol
  - ✓ **SMTP** → Simple Mail Transfer Protocol
  - ✓ **TFTP** → Trivial File Transport Protocol
  - ✓ **Telnet** → Remote Login, etc...



## Identification Problem



## Identification Problem

- ❖ General **identification problems** that occur in networks
  - ✓ **Identification of the network**
  - ✓ **Identification of the system** with in the **network**
  - ✓ **Identification of the process** with in the **system**

---



---



---



---



---



---

## Identification of the Network

---



---



---



---



---



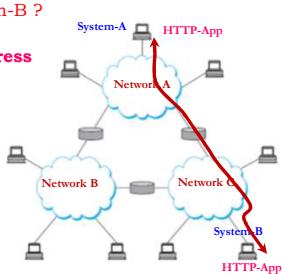
---

## Identification of the Network

- ❖ How to identify the network of System-B ?

- Need to know the **Network Address**

(ID)




---



---



---



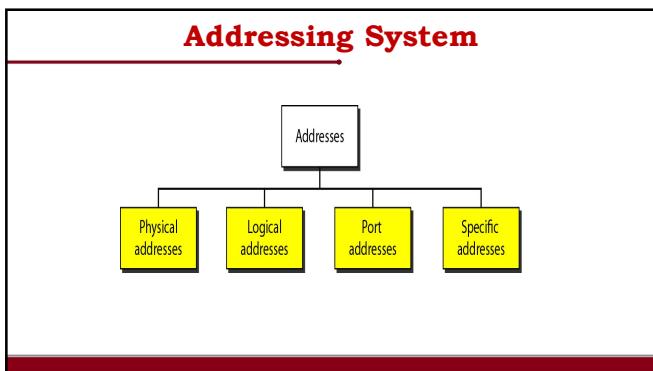
---



---



---




---

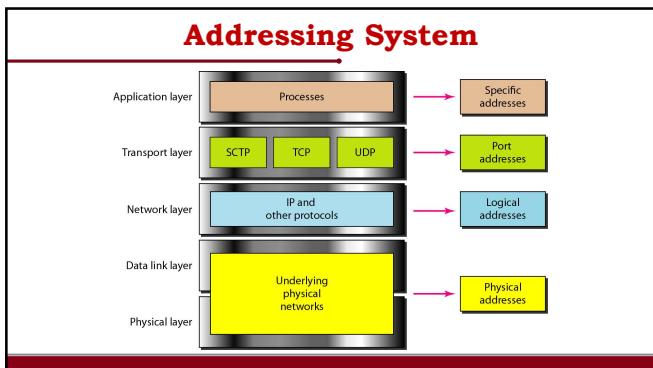
---

---

---

---

---




---

---

---

---

---

---




---

---

---

---

---

---

## Application Layer Addressing

### ❖ Uniform Resource Locator (URL)

- ✓ HTTP protocol : <http://www.nitw.ac.in>
- ✓ FTP protocol : <http://ftp.nitw.ac.in>
- ✓ SMTP protocol : <http://webmail.nitw.ac.in>

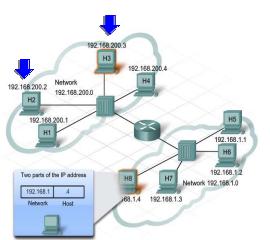
### ❖ Generally URL resolves IP Address using DNS servers (Domain Name Servers)

### ❖ How to get the IP address for the particular URL ?

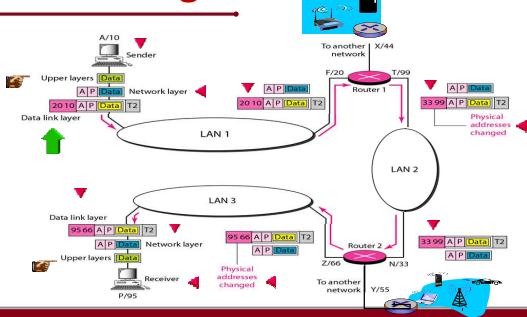
## Logical Addressing:

### ❖ Logical Addressing:

- ✓ Logical Addressing is mainly used for Identification of Network
- ✓ Logical Addressing used in Network Layer Addressing
- ✓ Example of Network Layer Addressing is Internet Protocol Addressing (IP Addressing)



## Logical Addressing:



## **Identification of the System with in the Network**

---



---



---



---



---



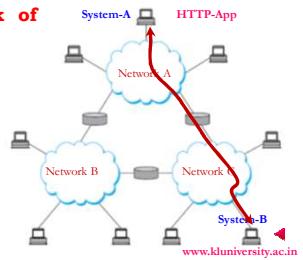
---

### **Identification of the System with in the Network**

- ❖ How to **identify the network of**

**System-B ?**

- ✓ Need to **know the host ID**




---



---



---



---



---



---



---

### **Identification of the System with in the Network**

- ❖ **Physical Layer Addressing** is used to **identify the System within network**

✓ **ARP addressing schema**

✓ **MAC address (Medium access control)**

---



---



---



---



---



---

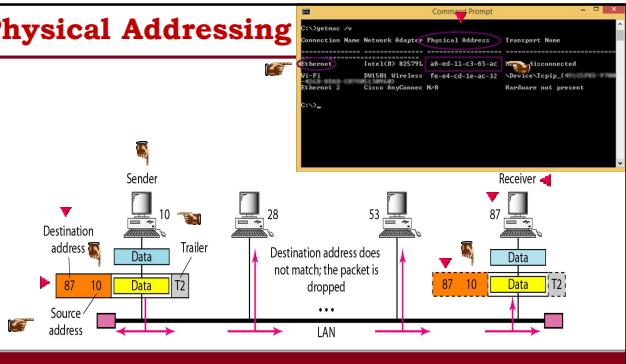


---

## Physical Addressing

- ❖ **Physical Addressing System** are **Permanent Addressing system** because **physical address** remains the **same** regardless of where the **host is placed on the network**.
- ✓ On a Host, the **MAC address** does not **change**; it is physically assigned to the **host NIC** and is known as the **physical address**.
- ✓ A **Physical address is a 48-bit flat address** burned into the **NIC card**

## Physical Addressing



## Physical Addressing

```

Windows IP Configuration

C:\>ipconfig /all

Windows IP Configuration

Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . : JudgeXP
  Link Layer Protocol : Microsoft TCP/IPv4
  IP Address . . . . . : 192.168.1.10
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.1.1
  DHCP Server . . . . . : 192.168.1.1
  DNS Servers . . . . . : 192.168.1.1

Ethernet adapter Wireless Network Connection:
  Connection-specific DNS Suffix . : JudgeXP
  Link Layer Protocol : Microsoft TCP/IPv4
  IP Address . . . . . : 192.168.1.10
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.1.1
  DHCP Server . . . . . : 192.168.1.1
  DNS Servers . . . . . : 192.168.1.1

```

## Identification of the Process with in the system

---



---



---



---



---



---

## Identification of the process with in the system

- ❖ Service(Port) Point Addressing System is mainly used to identify the process with in the system

- ✓ Use of **Port Number**
- ✓ For Ex: **Port 80 for HTTP**




---



---



---



---

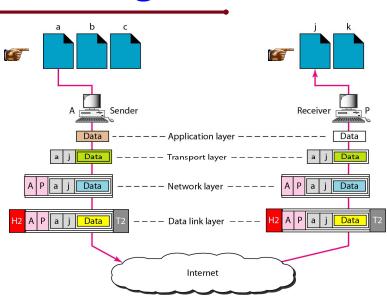


---



---

## Port Addressing




---



---



---



---



---



---

## Connection Problem




---

---

---

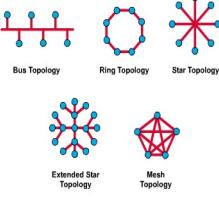
---

---

---

## Connection Problem

- ❖ General Connection problems can be solved using Network Topologies
  - ✓ BUS, RING, MESH, STAR etc.....
- ❖ Topology refers to the way a network is laid out, either physically or logically
  - ✓ Two or more devices connect to a link, two or more links form a **topology**




---

---

---

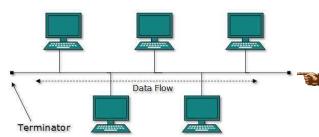
---

---

---

## Bus Topology

- ❖ This topology is commonly referred to as a **Linear Bus**,
- ✓ All the devices on a bus topology are connected by **one single cable**.
- ✓ In other words, A **long cable** acts as a **backbone to link** all the devices




---

---

---

---

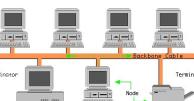
---

---

## Bus Topology

❖ **Advantages:**

- ✓ Ease of installation, Less cabling



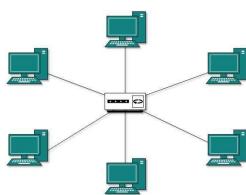
❖ **Disadvantages:**

- ✓ Fault isolation difficult,
- ✓ A fault or break in the cable stops all transmissions

## Star Topology

❖ The **star topology** is the most commonly used architecture in **Ethernet LANs**.

- ✓ The **star topology** resembles spokes in a bicycle wheel.
- ✓ When used with **network devices** that filter frames or packets



- ❖ This topology significantly **reduces the traffic** on the wires by **sending packets only** to the wires of the destination host.

## Star Topology

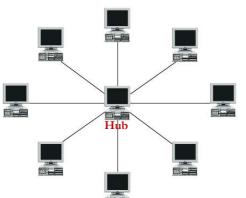
- ❖ Each device has a **dedicated point-to-point link** only to a **central controller**, usually **called a hub**. **No direct traffic between devices**

❖ **Advantages:**

- ✓ Less expensive, Less cabling and Robust

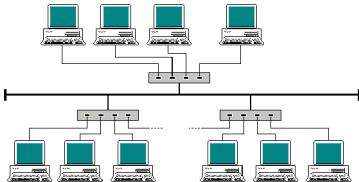
❖ **Disadvantages:**

- ✓ More cabling than Bus



## Tree Topology

- ❖ Larger networks use the **Extended Star Topology** also called **Tree Topology**.



## Tree Topology

❖ **Advantages:**

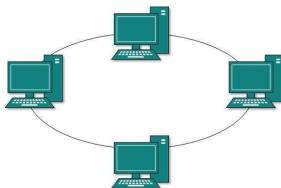
- ✓ It scales well
- ✓ Expansion of Network is possible and easy.
- ✓ Managing and maintaining is easy
- ✓ Error detection and correction is easy.

❖ **Disadvantages**

- ✓ It relies heavily on the main bus cable, if it breaks whole network fails
- ✓ As more and more nodes and segments are added, the maintenance becomes difficult.
- ✓ Scalability of the network depends on the type of cable used.

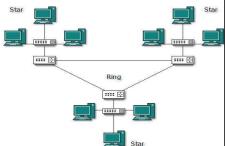
## Ring Topology

- ❖ A Frame travels around the ring, stopping at each node.
- ✓ If a node wants to transmit data, it adds the data as well as the destination address to the frame.
- ❖ The frame then continues around the ring until
  - ✓ it finds the destination node, which takes the data out of the frame.



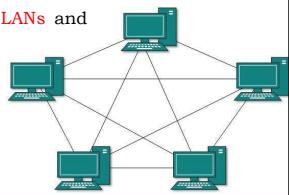
## Ring Topology

- ❖ **Single ring** – All the devices on the network share a single cable
- ❖ **Dual ring** – The **dual ring topology** allows data to be sent in both directions.
- ❖ **Advantages:**
  - ✓ Installation and reconfiguration relatively easy,
  - ✓ fault isolation simple
- ❖ **Disadvantages:**
  - ✓ A break in the ring can disable the entire network



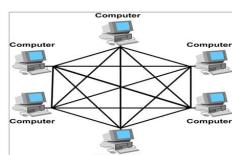
## Mesh Topology

- ❖ The **mesh topology** connects all devices (nodes) to each other for redundancy and fault tolerance.
- ❖ It is used in **WANs** to interconnect **LANs** and for **mission critical networks**
- ✓ Banks and financial institutions.



## Mesh Topology

- ❖ **Advantages:**
  - ✓ Dedicated connection,
  - ✓ Robust privacy/security
  - ✓ Fault identification/isolation easy
- ❖ **Disadvantages:**
  - ✓ Amount of cabling and I/O ports
  - ✓ installation and reconfiguration is difficult
  - ✓ Implementing the mesh topology is expensive and difficult.



## Devices

- 1. Hub:** A distributor that has a lot of ports which connected to computers.
- 2. Switches:** like a hub but it transmit packets to its destination
- 3. Bridge:** it is used to connect two similar LANs.
- 4. Routers:** choose the best path to transmit the packet.
- 5. Gateway:** it is used to connect two different LANs.
- 6. Repeaters:** repeats signals that travels via long distance

---



---



---



---



---



---

## Considerations When Choosing a Topology

- ❖ **Cost:** A linear bus network may be the **least expensive** way to install a network;
- ❖ **Infrastructure:** **Length of cable needed.** The linear bus network uses shorter lengths of cable.
- ❖ **Future growth:** With a star topology, **expanding a network is easily done**
- ❖ **Cable type:** The most common cable is **unshielded twisted pair**, which is most often used with star topologies.

---



---



---



---



---



---

## Goals:

- ❖ **Computer Network Design**
- ❖ **Network Issues**
- ✓ **Communication Problem**
- ✓ **Identification Problem**
- ✓ **Connection Problem**

---



---



---



---



---



---

**Thank You**

## CCN: End-To-End Communication: Switching



Dr. E.SURESH BABU  
Assistant Professor

Computer Science and Engineering Department  
National Institute of Technology, Warangal.  
Warangal, TS, India.



### Goals:

- ❖ Why is Switching is Required
  - Circuit Switching
    - ✓ Public Switched Telephone Networks
  - Message Switching
  - Packet Switching
    - ✓ Virtual Circuit Packet Switching
    - ✓ Datagram Packet Switching

### Why is Switching is Required

### Why is Switching is required

- ❖ When two computers are located close to each other that need to communicate,
  - ✓ it is often easiest just to run a cable between them.
  - ✓ LANs work this way.

---



---



---



---



---



---

### Limitations With Directly Connected Networks

- ❖ Directly connected networks limit the geographical area covered and number of hosts
  - ✓ Enable communication between hosts not directly connected

---



---



---



---



---



---

### Suitable Mechanism....

- ❖ When the distances are large or there are many computers
  - ✓ it is necessary to develop suitable mechanism for communication between any two devices

---



---



---



---



---



---

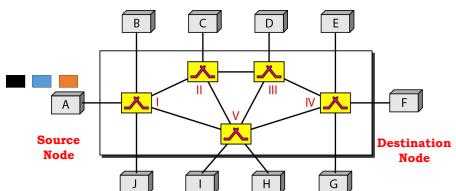
## Switching Mechanism

---

### Switching Mechanism

---

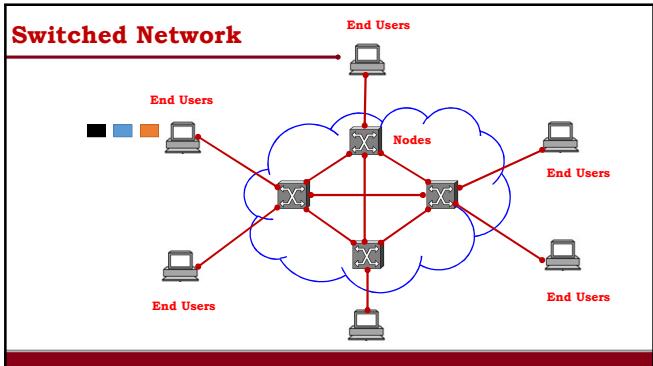
- ❖ A **Switching techniques** is used to perform communication between any two devices



### Switching Process

---

- ❖ In the **Switched Network Methodology**,
- ✓ Network consists of a **set of interconnected nodes**
- ✓ Information is transmitted from source to destination via **different routes**, which is **controlled by the switching mechanism**.




---

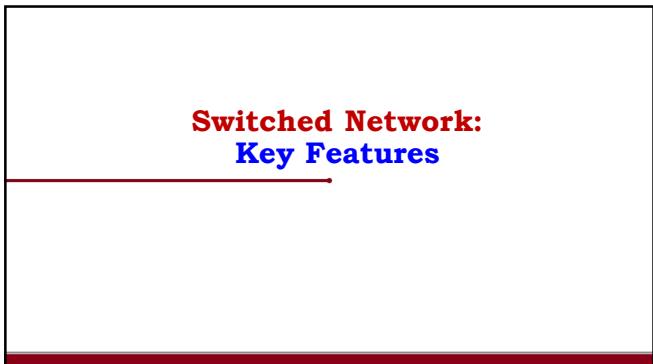
---

---

---

---

---




---

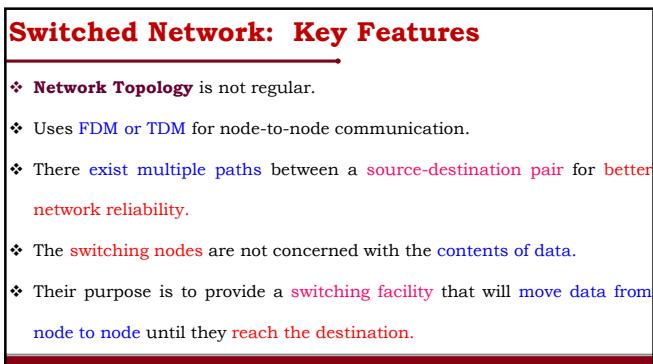
---

---

---

---

---




---

---

---

---

---

---

## Taxonomy of Switched Networks

---



---

---

---

---

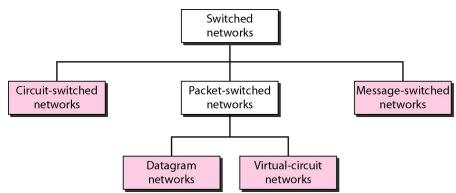
---

---

### Taxonomy of Switched networks

---

- ❖ The switching performed by different nodes can be categorized into the following three types:




---

---

---

---

---

---

## Circuit Switching

---



---

---

---

---

---

---

## Circuit Switching

- ❖ A **Circuit-switched Network** consists of a set of switches connected by physical links.
- ✓ A connection between two stations is a **dedicated path** made of **one or more links**.
- ✓ Each connection uses only **one dedicated channel** on each link.
- ✓ Each link is normally divided into **n channels** by using **FDM or TDM**.

---



---



---



---



---



---

## Circuit Switching

- ❖ Circuit switching is commonly used technique in **Telephony**,
- ✓ where the caller sends a special message with the **address of the callee** (i.e. by dialling a number) to **state its destination**.




---



---



---



---



---



---

## Circuit Switching

- ❖ Circuit Switching involved the following **three distinct steps**
- **Circuit Establishment**
- **Data transfer**
- **Circuit disconnect**

---



---



---



---



---



---

## Circuit Switching

### ❖ Circuit Establishment:

✓ To establish an **end-to-end connection** before any transfer of data.

✓ Circuit may be a **dedicated link/Shared**.

### ❖ Data transfer:

✓ **Transfer data** is from the source to the destination.

✓ The **data may be analog or digital**, depending on the nature of the network.

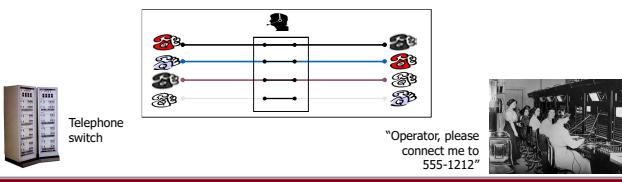
✓ The connection is generally **full-duplex**.

### ❖ Circuit disconnect:

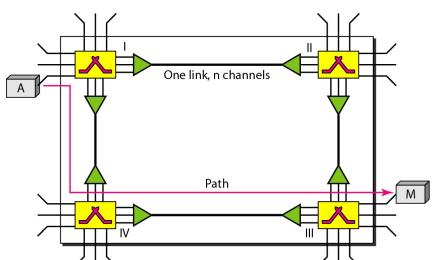
✓ Terminate connection at the end of data transfer.

## Circuit Switching

**A Circuit-switched Network is made of a set of switches connected by physical links, in which each link is divided into n channels.**



## Scenario



### Disadvantages of Circuit-Switched Network

- ❖ A **dedicated connection** that has **no transmission** means **wasted bandwidth**
- ❖ A **connection is time consuming** if short, infrequent, or sporadic communication is to occur

---



---



---



---



---



---

### Message Switching

---



---



---



---



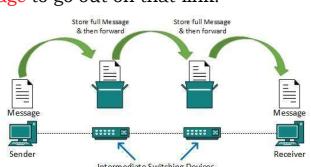
---



---

### Message Switching

- ❖ Each **network node receives and stores** the message
- ❖ Determines the **next Node of the route**
- ❖ **Queues the message** to go out on that link.




---



---



---



---

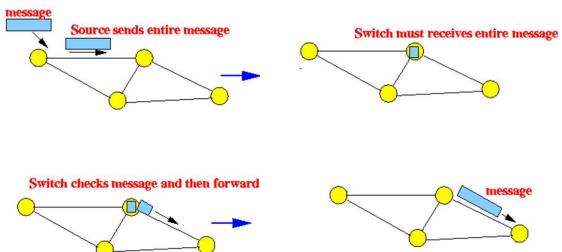


---



---

### Message Switching

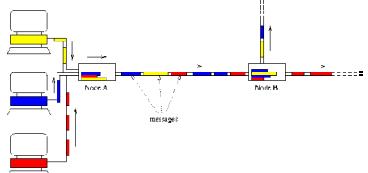


### Message Switching

- ❖ In Message Switching Method,
  - ✓ The message is sent to the nearest switching node directly.
  - ✓ The nearest node stores the message, checks for errors, selects the best available route and forwards the message to the next intermediate node.

### Message Switching

- ❖ The Data Link line becomes free again for other messages, while the process is being continued in some other nodes.
- ❖ Due to the mode of action, this method is also known as **store-and-forward technology**



### Message Switching: Disadvantages

- ❖ Message of large size dominates the link and storage

---



---



---



---



---



---

### Packet Switching

---



---



---



---



---



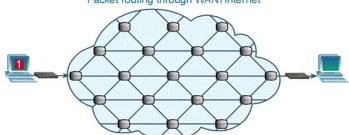
---

### Packet Switching

- ❖ Packet Switching approach was developed for **long-distance data communication** purpose

- ✓ To overcome the **limitations of message switching**

Packet routing through WAN/Internet.




---



---



---



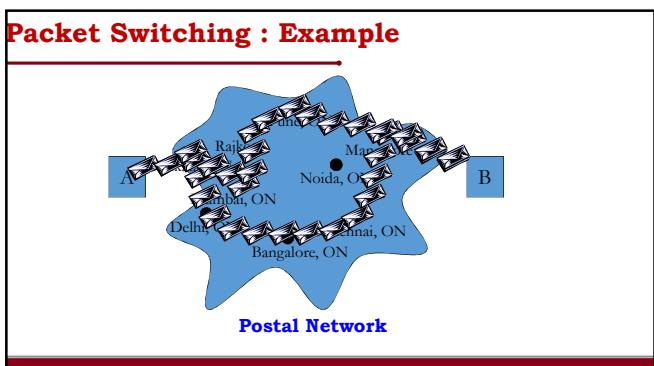
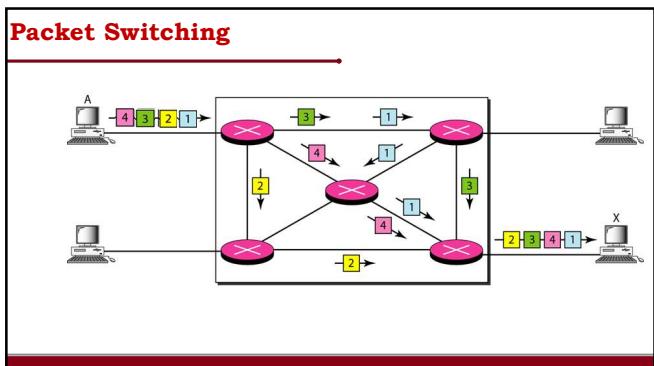
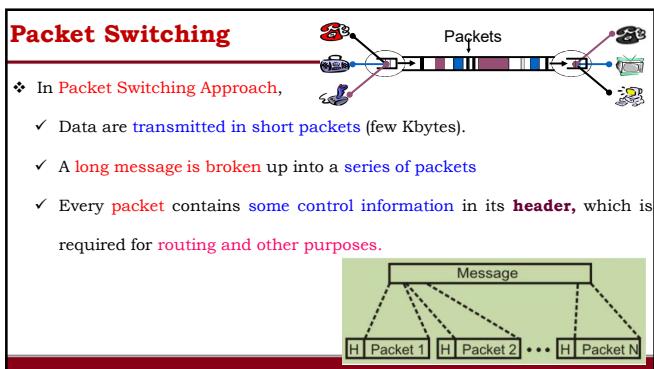
---



---



---



## Why Packet Switching

- ❖ The **Directly connected networks** suffer from two limitations
  - ✓ Limits with # of hosts attached.
    - Ex : An **Ethernet** can connect up to only **1024 hosts**.
  - ✓ Limits the geographic area with a single network
    - Ex : An **Ethernet** can span only **2500m**

---



---



---



---

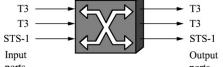


---



---

## Why Packet Switching

- ❖ To build the **networks** that scales well and **covers the globally**
  - ✓ we need to **enable communication between hosts** that are **not directly connected**.
- ❖ A **Switch** is a **multi-input, multi-output device** that allows us to **interconnect links** to form **larger networks**.
- ❖ A **switch** transfers **packets** from an **input** to **one or more outputs**


The diagram shows a central switch node with four input ports labeled 'Input ports' and four output ports labeled 'Output ports'. Arrows indicate bidirectional traffic flow between the ports. One input port is labeled 'T3' and one output port is also labeled 'T3'. Another input port is labeled 'STS-1' and its corresponding output port is also labeled 'STS-1'.

---



---



---



---



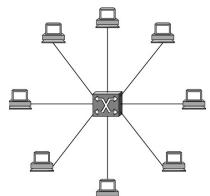
---



---

## Why Packet Switching

- ❖ The **inputs and outputs** of a switch are referred to as **Ports**




---



---



---



---



---



---

## Packet Switching Works

- ❖ How does the **Packet Switch** decide which output port to place each packet on?

---

---

---

---

---

---

## Packet Switching Works

- ❖ Generally Switch looks at the **header of the packet addresses** to make decisions.
- ❖ There are **two common approaches** to know the details of how it uses the header of the packet addresses,
  1. **Datagram Or Connectionless Packet Switching Approach**
  2. **Virtual Circuit Or Connection-oriented Packet Switching Approach.**

---

---

---

---

---

---

## Virtual Circuit Packet Switching

---

---

---

---

---

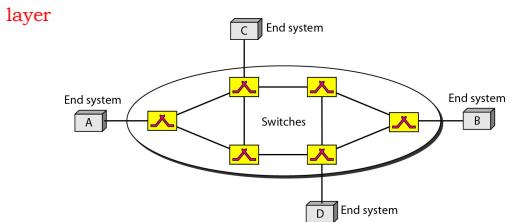
---

## Virtual Circuit Packet Switching

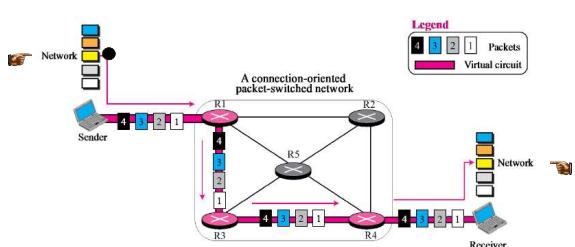
- ❖ Virtual Circuit Packet Switching is a widely used technique that uses the concept of a **virtual circuit(VC)**.
- ❖ This approach, which is also called **connection-oriented model** that requires
  - ✓ **Virtual Connection Setup** between the Source and Destination Host
  - ✓ **Data Transfer**

## Virtual Circuit Packet Switching

- ❖ A virtual-circuit network is normally implemented in the **data link layer**

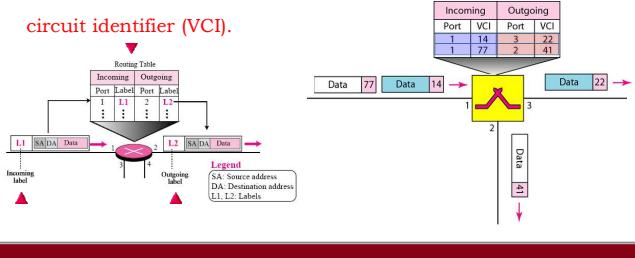


## Virtual Circuit Packet Switching



## Virtual Circuit Packet Switching

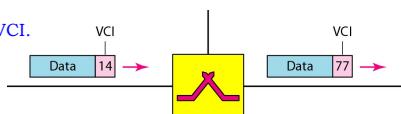
- Each **Switch** has a **VC table**. Every VCs are identified by a **virtual circuit identifier (VCI)**.



## Virtual Circuit Packet Switching

- The **identifier** that is actually used for **data transfer** is called the **virtual-circuit identifier (VCI)**.
- A **VCI** is used by a **frame** between two switches.

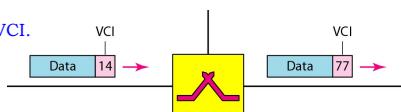
When a **frame arrives at a switch**, it has a **VCI**; when it leaves, **it has a different VCI**.



## Virtual Circuit Packet Switching

- The **identifier** that is actually used for **data transfer** is called the **virtual-circuit identifier (VCI)**.
- A **VCI** is used by a **frame** between two switches.

When a **frame arrives at a switch**, it has a **VCI**; when it leaves, **it has a different VCI**.

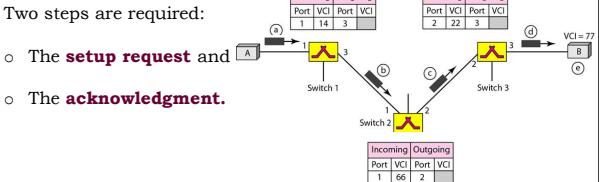


## Phases...

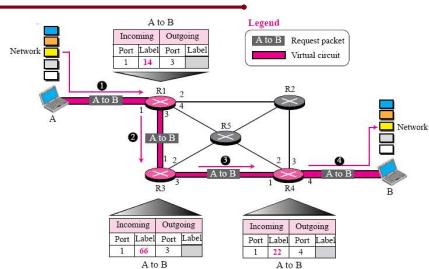
- ❖ Virtual-circuit Network will go through **three phases** between source and destination
- ❖ **Setup,**
- ❖ **Data Transfer,**
- ❖ **Disconnect**

## Setup Phase: Request

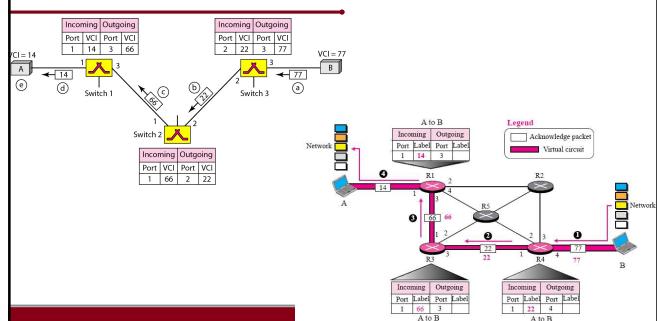
- ❖ In the setup phase,
- ✓ A switch creates an entry for a virtual circuit.
- ✓ Two steps are required:
  - The **setup request** and
  - The **acknowledgment.**



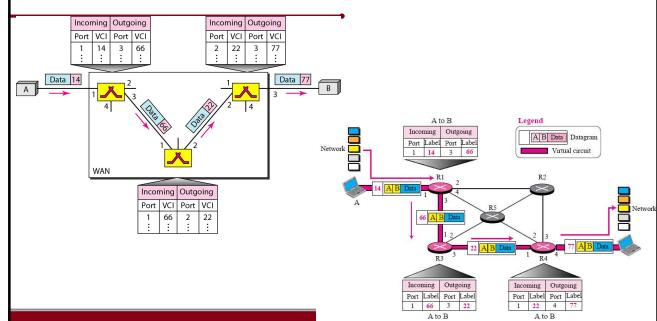
## Setup Phase: Request



## Setup Phase: Acknowledgment



## Data Transfer Phase



## Popular VC Networks

- Two popular networks are **X.25** and **Frame Relay**, which are commonly used for public data networks (PDN).

### Advantages.....

- ❖ Routing is faster
  - ✓ A route must only be determined once, for the first message
  - ✓ Once the route has been determined, the path used by the router is reused for all messages
  - ✓ As a result, routing tables are much smaller (and can be searched more quickly)

---



---



---



---



---



---

### Advantages.....

- ❖ Messages do not arrive out of order
  - ✓ As a result, receivers do not need to reorder the cells

---



---



---



---



---



---

### Disadvantages.....

- ❖ Connections take some time to create
- ❖ The connection may be lost after a timeout, and will have to be recreated again and again
- ❖ Routing tables will be dynamic, and routing algorithms are more complex

---



---



---



---



---



---

## Datagram Packet Switching

---



---

---

---

---

---

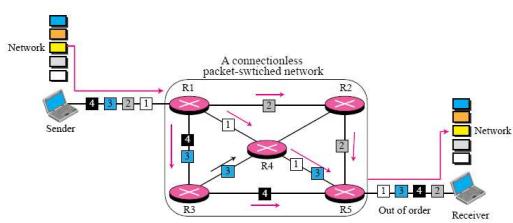
### Datagram Packet Switching

---

- ❖ A switch in a datagram network uses a **routing table** that is based on the **destination address**.
  - ❖ The **destination address** in the **header** of a **packet** in a datagram network remains the **same during the entire journey of the packet**.
    - ✓ Each **packet** is routed individually
- 
- 
- 
- 
- 

### Datagram Packet Switching

---




---

---

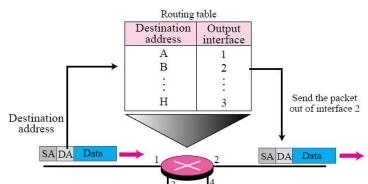
---

---

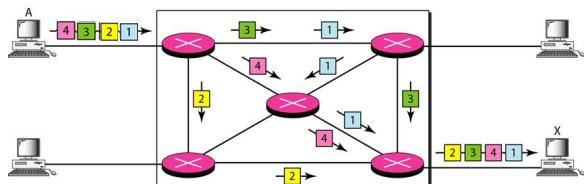
---

## Datagram Packet Switching

- A switch in this network consults a routing table which is stored inside the switch to decide **how to forward a packet**.



## Datagram Packet Switching



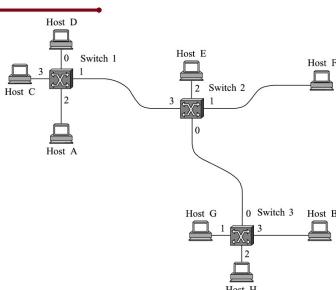
## Datagram Packet Switching

- No dedicated connection between communicating hosts
- Packets are sent to the switch at any time
- Source is not aware of the state of the destination
- Packets may follow independent paths to the destination (out-of-order delivery, larger delays, etc.)
- Less prone to switch failures if alternative paths exist

## Datagram Packet Switching

Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Routing table of Switch 2



## Advantages.....

- ❖ Connections need not be created
- ❖ Infrequent messaging is perfect for **connectionless messaging**
- ❖ Routing each message separately allows for **load balancing**

## Disadvantages.....

- ❖ Each message takes a certain amount of time to transmit (including transmission, routing, reception, etc.)
- ❖ Messages may arrive out of order

## Comparison of Virtual-Circuit and Datagram Approaches

---



---



---



---



---



---

### **Comparison of Virtual-Circuit and Datagram Approaches**

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

---



---



---



---



---



---

### **Goals:**

- ❖ Why is Switching is Required
  - Circuit Switching
    - ✓ Public Switched Telephone Networks
  - Message Switching
  - Packet Switching
    - ✓ Virtual Circuit Packet Switching
    - ✓ Datagram Packet Switching

---



---



---



---



---



---

**Thank You**

## CCN: Networks Software Model



Dr. E.SURESH BABU  
Assistant Professor



Computer Science and Engineering Department  
National Institute of Technology, Warangal.  
Warangal, TS, India.



### Goals:

- ❖ Networks Software Model
- ❖ Layering Concept
  - ✓ Open System Interconnection
    - Specific Responsibilities of each layer
  - ✓ TCP/IP Reference Model
- ❖ Comparison OSI and TCP/IP
- ❖ Analyzing the TCP/IP Using Wireshark (Active Learning)

## Data Communication

## Network Technology

- ❖ Network Technology is coordinated with set of software and hardware
- ✓ For Example: Drivers, Network adapters, Cables and Connectors
- ❖ Mechanism of Data transmission across the communications links

---



---



---



---



---



---

## Data Communication

- ❖ Direct communication is exchange of data between two or more devices via some transmission medium
- ❖ Transmission medium between devices : Medium through Which we transmit our data.
- ✓ Eg. Transmission of data from laptop to PC via cable, wireless etc.




---



---



---



---

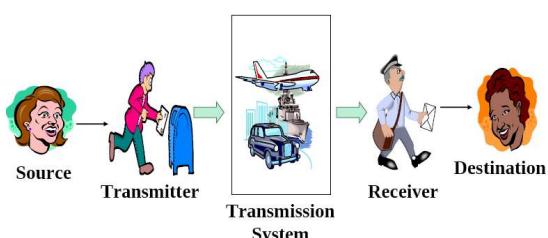


---



---

## Physical Communication Model




---



---



---



---



---



---

## Physical Communication Model

- ❖ **Communication Model** has five components
- ✓ **Source** : Generates the data to be transmitted
- ✓ **Transmitter** : Converts data into transmittable signals
- ✓ **Transmission system** : Carries the data
- ✓ **Receiver** : Converts the received signals into data
- ✓ **Destination** : Accepts the incoming data.

---



---



---



---

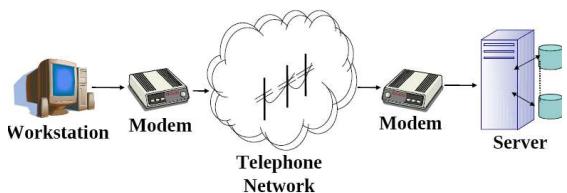


---



---

## Electronics Communication Model




---



---



---



---

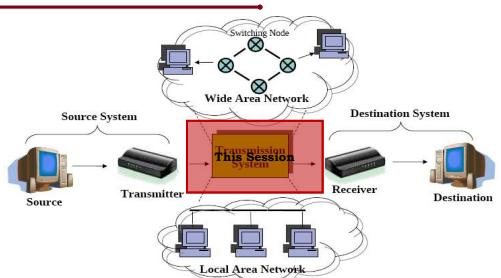


---



---

## Network Model




---



---



---



---



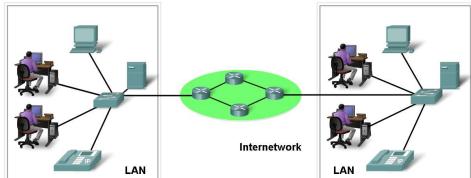
---



---

## Communication Technology

- All computer networks are in widespread use based on **Scaling**.



## Scaling on Communication Technology

- The another criteria for classifying of computer networks is **scale**.
- They are divided into **three categories**:

  - Local Area Network (LAN)**.
  - Metropolitan Area Network(MAN)**.
  - Wide Area Networks (WAN)**.

## Networks Software Model

- Computer Networks is a **very complex systems** with many “pieces”
- ✓ **Hosts**,
- ✓ **Routers**,
- ✓ **Links of various media**,
- ✓ **Applications**,
- ✓ **Protocols**,
- ✓ **Hardware, Software etc.** which needs a **Overall Plan**

## Networks Software Model

- ❖ Overall Plan of Computer Networks requires an **Architecture**
- ❖ Network Architectures define the
  - ✓ Standards and techniques for designing and building communication systems for computers and other devices.

---



---



---



---



---



---



---

## Basic Concept of Layering

- ❖ To reduce the **design complexity** and to **flexible implementation**, the **concept of layering** is introduced
- ✓ Most **computer networks** are organized as a **Stack of Layers**

---



---



---



---



---

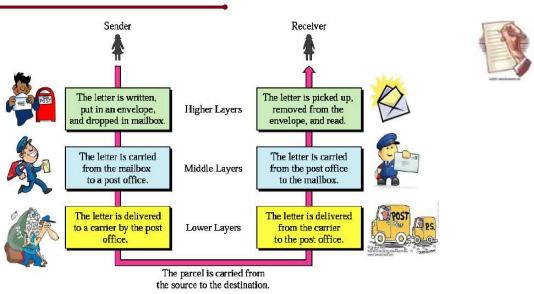


---



---

## Layering Task involved in sending a letter




---



---



---



---



---

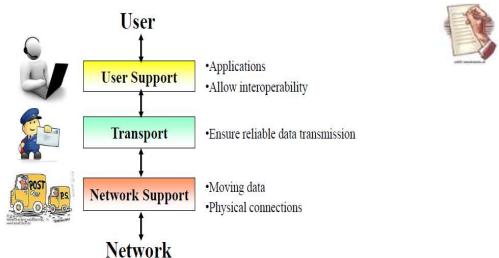


---



---

## Layering Task




---

---

---

---

---

---

## Layering Concept

---

---

---

---

---

---

## Layering System

- ❖ Layering System provides
- ✓ Modularity and Abstracts out of implementation details.
- ✓ Enables different components and implemented independently.
- ✓ Changes in one layer does not necessarily changes in other layers.
- ✓ Once a certain service is implemented,
  - Several upper layer programs may use the same service.

---

---

---

---

---

---

## Network Architecture

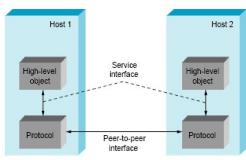
- ❖ The basic idea of a **layered architecture** is to divide the design into small pieces called **Layers**.
- ❖ A set of layers, protocols and interfaces between two consecutive layers is known as **network architecture**

## Layering Architecture

- ❖ The basic elements of a **layered model** are
  - ✓ **Services**,
  - ✓ **Protocols** and
  - ✓ **Interfaces**.

## Services, Protocols and Interfaces...

- ❖ Interfaces are defined between the layers that provides the service.
- ✓ The messages from one layer to another are sent only through interfaces.
- ❖ Two modules in the same layer on distinct network entities, that communicate with each other are referred to as **peers**.



## Layering Architecture

- ❖ Interfaces between layers (Physical)
- ❖ Peer-to-Peer process (Logical)

---



---



---



---

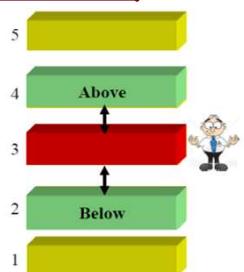


---



---

## Interfaces between layers (Physical)




---



---



---



---



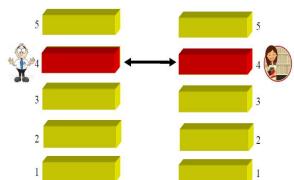
---



---

## What is Peer-to-Peer Communication

- ❖ Two modules in the same layer on distinct network entities, that communicate with each other are referred to as **peers**.




---



---



---



---



---



---

## Open System Interconnection (OSI) Reference Model

---



---

---

---

---

---

---

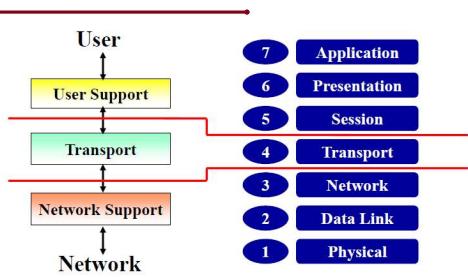
### Open System Interconnection(OSI) Reference Model

---

- ❖ The OSI reference model is a **layered architecture** developed by
    - ✓ **International Standards Organization(ISO).**
    - ✓ which as an **international standard** for data networks
  - ❖ The **OSI reference model** has a **clean structure** that helps in the understanding of layering.
- 
- 
- 
- 
- 
- 

### Open System Interconnection (OSI) Reference Model

---




---

---

---

---

---

---

---

### Open System Interconnection(OSI) Reference Model

- ❖ The OSI model is a framework into which the various networking standards can fit.
- ❖ The OSI model specifies
  - ✓ what aspects of a network's operation can be addressed by various network standards.
- ❖ Finally, the OSI model is sort of a standard of standards.

---



---



---



---



---

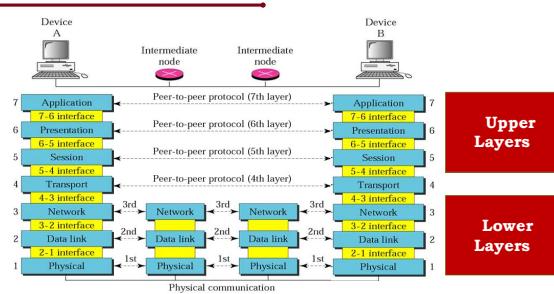


---



---

### Open System Interconnection (OSI) Reference Model




---



---



---



---



---



---



---

### OSI Model: Conceptual Picture

- ❖ The first three layers are sometimes called the **lower layers**.
  - ✓ They deal with the mechanics of how information is sent from one computer to another over a network.
- ❖ Layers 4 through 7 are sometimes called the **upper layers**.
  - ✓ They deal with how applications programs relate to the network through application programming interfaces.

---



---



---



---



---



---



---

## Layers of the OSI model

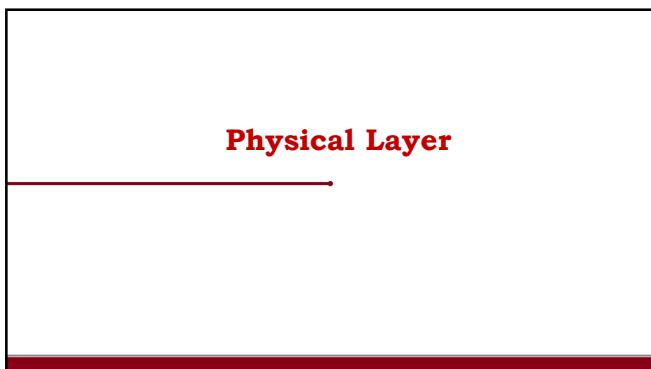
<b>Physical Layer</b>	: Transmission/reception of raw bits
<b>Data Link Layer</b>	: Maps bits into frames, dictates sharing of common medium, corrects/detects errors , re-orders frames
<b>Network Layer</b>	:Routes packets to destination, may perform fragmentation and re-assembly.
<b>Transport Layer:</b>	Flow (congestion) control, error control, transparent transport to upper layers

## Layers of the OSI model

<b>Session Layer</b>	: Establishes connection among hosts, duplex, half duplex, graceful connection termination, combination of streams
<b>Presentation Layer</b>	: Negotiation of format of data exchanged between hosts
<b>Application layer:</b>	Application services such as FTP, X.400 (mail), HTTP

## Layers of the OSI model

7	<b>Application</b>	User service
6	<b>Presentation</b>	Translate format, encrypt
5	<b>Session</b>	Session manage, checkpoints
4	<b>Transport</b>	Reliable end-to-end (whole message)
3	<b>Network</b>	Packet end-to-end (across network)
2	<b>Data Link</b>	Node-to-node (same network segment)
1	<b>Physical</b>	Physical



## Physical Layer

---

---

---

---

---

---

### Specific Responsibilities: Physical Layer

- ❖ **Representation of bits:** 0/1 encoded into signals (electrical or optical)
- ❖ **Data Rate:** duration of a bit, how long a bit lasts
- ❖ **Synchronization of bits:** synchronization at the bit level
- ❖ **Line configuration:** point-to-point or multipoint
- ❖ **Physical Topology:** mesh, star, ring, or bus
- ❖ **Transmission Mode:** simplex, half-duplex, or full-duplex

---

---

---

---

---

---

## Physical Layer

- ❖ The **Bottom layer of the OSI model** is the **Physical layer**.
- ❖ The physical layer is concerned with **transmission of raw bits** over a **communication channel**.
- ✓ It specifies the **mechanical, electrical and procedural network interface specifications** and
- ✓ The **physical transmission of bit streams** over a **transmission medium** connecting **two pieces of communication equipment (Topology)**.

---

---

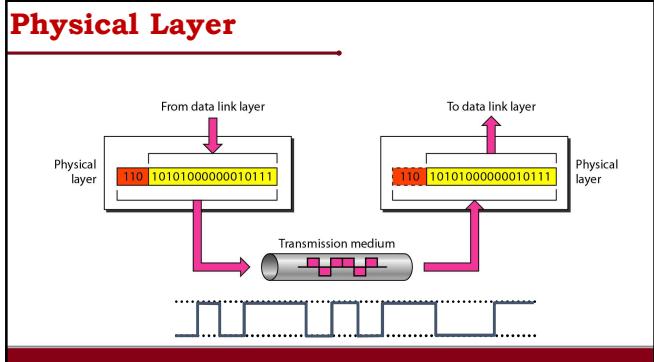
---

---

---

---

## Physical Layer



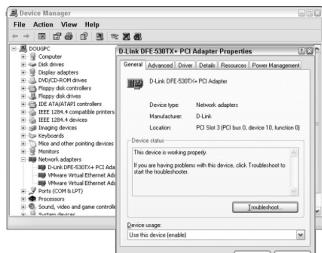
## Physical Layer

- ❖ **Physical layer** addresses the **physical characteristics** of the network
    - ✓ **Types of cables** used to connect devices,
    - ✓ The **types of connectors** used,
    - ✓ How **long the cables** can be and so on.
  - ❖ The **star, bus, ring, and mesh** network topologies are used in the **Physical layer**.

## **Physical Layer**

## Network Adaptors(NIC) Properties

- The *network adapter* (also called a *network interface card* or *NIC*) that's installed in each computer on the network is a **Physical layer device**



## Layers of the OSI model

7	<b>Application</b>	User service
6	<b>Presentation</b>	Translate format, encrypt
5	<b>Session</b>	Session manage, checkpoints
4	<b>Transport</b>	Reliable end-to-end (whole message)
3	<b>Network</b>	Packet end-to-end (across network)
2	<b>Data Link</b>	Node-to-node (same network segment)
1	<b>Physical</b>	Physical

## Data Link Layer

## Specific Responsibilities: Data Link Layer

- ❖ Layer 2 is Responsible of:
  - ✓ Moving frames from one hop (node) to the next.
  - ✓ **Framing:** divided the stream of bits received from the network layer manageable data units called frames.
  - ✓ **Physical address (MAC address).**
  - ✓ **Flow control.**
  - ✓ **Error control:** added trailer to the end of frame.
  - ✓ **Access control.**
  - ✓ **Hop-to-Hop** ( node-to-node).

---



---



---



---



---



---



---

## Data Link Layer

- ❖ The Data Link layer is the lowest layer at which meaning is assigned to the bits that are transmitted over the network.
- ❖ Data link layer provides service to the Network layer.
- ✓ The network layer wants to be able to send packets to its neighbors without worrying about the details of getting it there in one piece.

---



---



---



---



---



---



---

## Data Link Layer

- ❖ Data link protocols address things
  - ✓ Size of each packet of data to be sent,
  - ✓ The address of each packet so that it's delivered to the intended recipient.

---



---



---



---



---



---



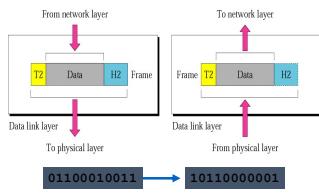
---

## Data Link Layer

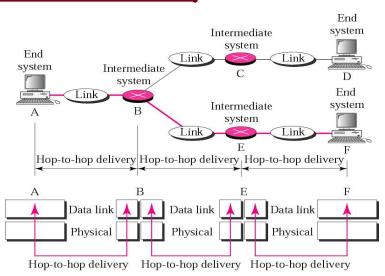
- ❖ **Error control protocol** returns a **positive or negative acknowledgment** to the sender.
- ✓ A **positive acknowledgment** indicates the **frame was received without errors**,
- ✓ While a **negative acknowledgment** indicates the opposite.

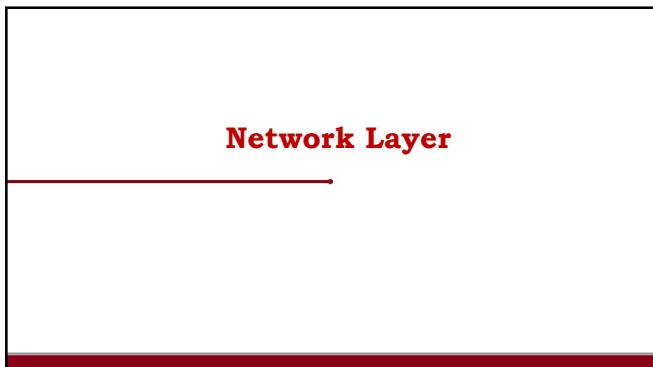
## Data Link Layer

- ❖ **Flow control** prevents a fast sender from overwhelming a slower receiver.
- ✓ For example, a **supercomputer** can easily generate data faster than a **PC can consume it**.



## Layer-2: Data Link layer Hop-to-Hop delivery





## Network Layer

---

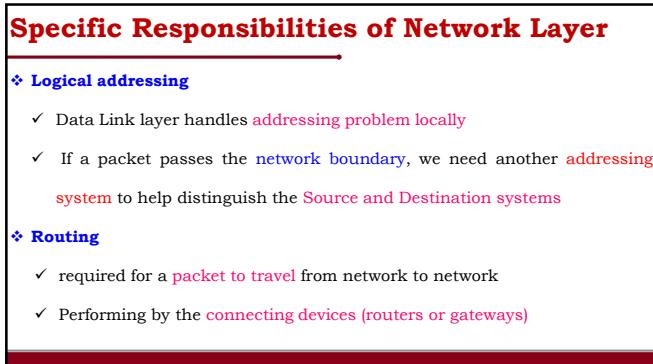
---

---

---

---

---



- Specific Responsibilities of Network Layer**
- ❖ **Logical addressing**
    - ✓ Data Link layer handles addressing problem locally
    - ✓ If a packet passes the **network boundary**, we need another addressing system to help distinguish the **Source** and **Destination** systems
  
  - ❖ **Routing**
    - ✓ required for a **packet to travel** from network to network
    - ✓ Performing by the **connecting devices (routers or gateways)**

---

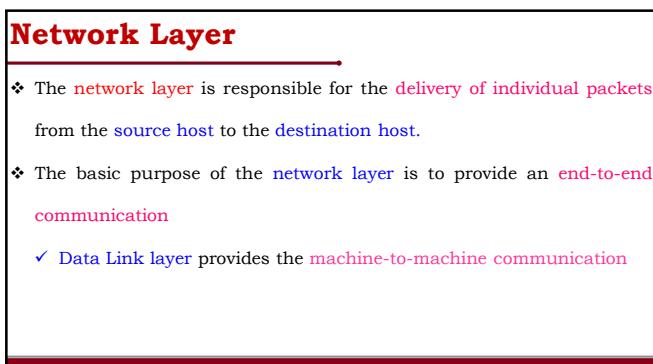
---

---

---

---

---



- Network Layer**
- ❖ The **network layer** is responsible for the **delivery** of individual packets from the **source host** to the **destination host**.
  - ❖ The basic purpose of the **network layer** is to provide an **end-to-end communication**
  - ✓ **Data Link layer** provides the **machine-to-machine communication**

---

---

---

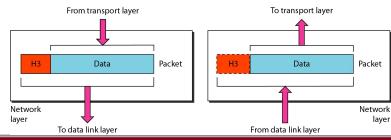
---

---

---

## Network Layer

- ❖ The network layer logically concatenates a set of links to provide
  - ✓ the abstraction of an end-to-end connection (to the transport layer).
- ❖ The network layer is handed a piece of data by the transport layer
  - ✓ Delivering the packets to the correct destination.



## Network Layer

- ❖ The Network Layer is responsible for
  - ✓ Sending the packets, which includes **data plus source and destination addresses**
- ❖ The **end-to-end connection** is performed using two basic approaches
  1. Connection-oriented Network-layer Services.
  2. Connectionless Network-layer Services.

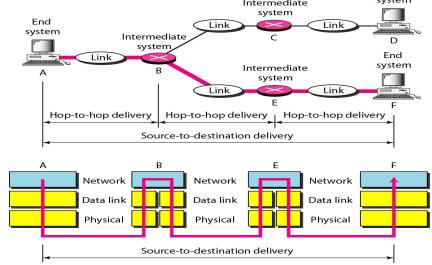
## Connection-Oriented Network-layer Services.

- ❖ In **Connection-Oriented** approach is also known as **virtual circuit approach**,
- ✓ A **route** consists of logical connection that establishes **first between two users**.

### Connection-Less Network-layer Services.

- The Connection-Less is also called as datagram approach which is a self-contained message unit.
- This approach contains sufficient information for routing from the source node to the destination node.
- This approach does not dependent on previous message that interchanges between them.

### Network Layer: Source-to-Destination Delivery



### Transport Layer

## Specific Responsibilities of Transport Layer

- ❖ Service-point addressing (port addresses used in TCP)
- ❖ Segmentation and reassembly
- ❖ Control connection: connectionless or connection-oriented
- ❖ Flow control (end to end, not just a single link)
- ❖ Error control (end to end, not just a single link)

---

---

---

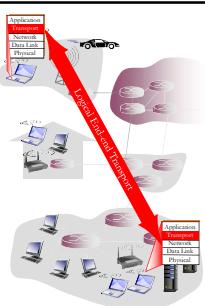
---

---

---

## Transport Layer

- ❖ Transport Layer provide logical end-to-end communication between application processes running on different hosts
- ❖ The Transport Layer is responsible for the delivery of a message from one process to another.




---

---

---

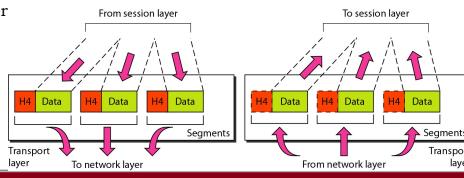
---

---

---

## Transport Layer

- ❖ Transport protocols run in end systems
- ✓ **Sender side:** breaks application messages into segments, passes to network layer
- ✓ **Receiver side:** reassembles segments into messages, passes to application layer




---

---

---

---

---

---

## Transport Protocols

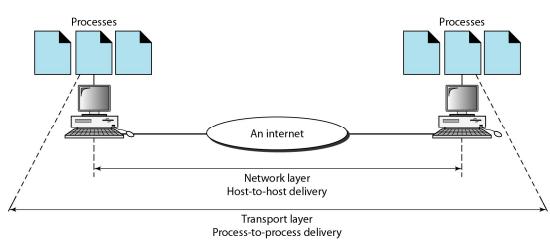
- ❖ Two Popular transport protocol available one is **TCP** and other is **UDP**
- ✓ **Transport Control Protocol (TCP)** is a connection-oriented protocol which provides error recovery, flow control, and congestion control.
- ✓ **User Datagram Protocol (UDP)** does not provide any of these services.

## Transport Protocols

```
Windows\system32\cmd.exe
C:\netstat
Active Connections

  Proto  Local Address          Foreign Address          State
  TCP    0.0.0.0:145             192.168.1.1:1027      ESTABLISHED
  TCP    0.0.0.0:2865            192.168.1.1:1027      CLOSE_WAIT
  TCP    0.0.0.0:2869            192.168.1.1:1027      CLOSE_WAIT
G:\>~
```

## Reliable Process-to-Process Delivery of a Message



## Session Layer

---



---

---

---

---

---

### Specific Responsibilities

- ❖ **Dialog control :** Allows two systems to enter into a dialog communication between two processes:
  - ✓ half-duplex or full-duplex
- ❖ **Synchronization:** add check points into a stream of data

---

---

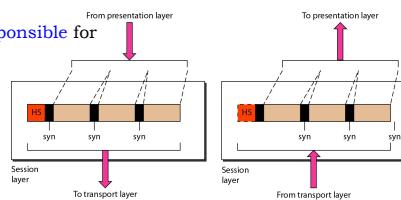
---

---

---

### Session Layer

- ❖ The Session layer allows users on different machines to establish session between them.
- ❖ The session layer is responsible for
  - ✓ **Dialog Control**
  - ✓ **Token Management**
  - ✓ **Synchronization.**




---

---

---

---

---

## Synchronization.

- ❖ Session layer provides checkpoints into data streams,
- ✓ if any crash, only the **data transferred** after the **last checkpoint** have to be repeated.

---



---



---



---



---



---

## Session Layer

- ❖ The session layer allows **three types of transmission modes:**
  - ◆ **Simplex:** In this mode, data flows in only one direction.
  - ◆ **Half-duplex:** In this mode, data flows in both directions, but only in one direction at a time.
  - ◆ **Full-duplex:** In this mode, data flows in both directions at the same time.

---



---



---



---



---



---

## Presentation Layer

---



---



---



---



---



---

## Presentation Layer: Summary

- ❖ The presentation layer is responsible for
  - ✓ **Translation,**
  - ✓ **Compression,**
  - ✓ **Encryption**

---



---



---



---



---



---

## Presentation Layer

- ❖ This layer was intended to **provide the mechanism**
  - ✓ to **convert of the data** being transmitted to the **appropriate format** depending on the machine architecture.
- ❖ This layer is concerned with **Syntax and Semantics** of the information transmitted

---



---



---



---



---



---

## Presentation Layer

- ❖ The Presentation layer also apply sophisticated **compression techniques**
  - ✓ Using **fewer bytes of data are required** to represent the information when it's **sent over the network.**
  - ✓ At the other **end of the transmission**, the Presentation layer then **uncompresses the data.**

---



---



---



---



---



---

## Presentation Layer

- ❖ The Presentation layer also support **sophisticated encryption technique**
- ✓ Scramble the data before it is **transmitted** and **unscramble it** at the other end

---



---



---



---



---



---

## Application Layer

---



---



---



---



---



---

## Application Layer:

- ❖ The **application layer** consists of the **set of applications running** on the end hosts.
- ✓ Examples are the **World Wide Web (the HTTP protocol)**, **Email**, **Telnet**, **FTP**, and **Streaming**.
- ❖ The **application layer** is responsible for **providing services to the user**.

---



---



---



---

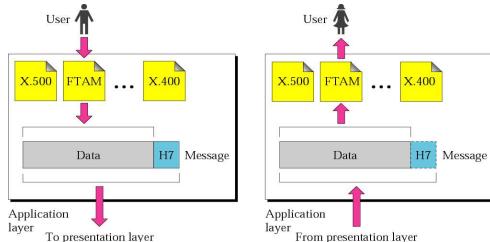


---



---

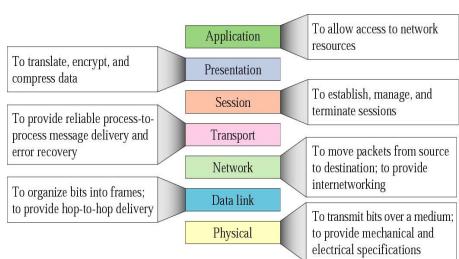
## Application Layer:



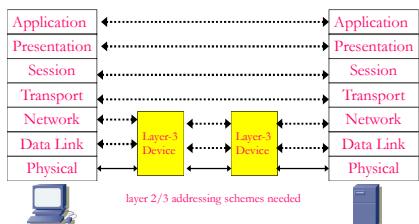
## Application Layer:

- ❖ Some of the better-known **Application layer protocols** are
  - ◆ **DNS (Domain Name System)** for resolving Internet domain names.
  - ◆ **FTP (File Transfer Protocol)** for file transfers.
  - ◆ **SMTP (Simple Mail Transfer Protocol)** for e-mail.
  - ◆ **SMB (Server Message Block)** for file sharing in Windows networks.
  - ◆ **NFS (Network File System)** for file sharing in UNIX networks.
  - ◆ **Telnet** for terminal emulation.

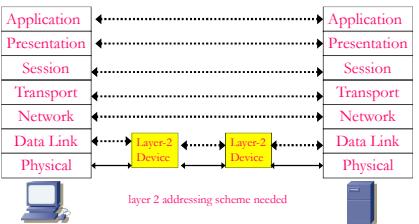
## Summary



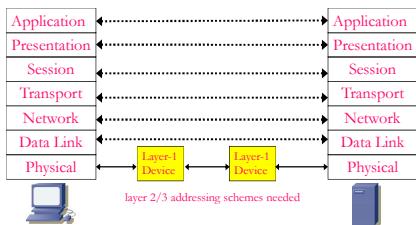
### Connection via LAYERS-3



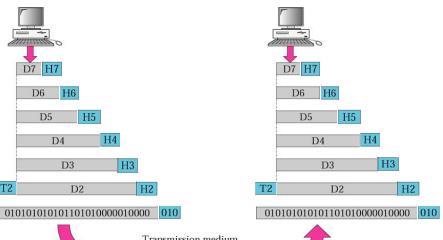
### Connection via LAYERS-2



### Connection via LAYERS-1



### An Exchange using the OSI model



### Limitation :OSI model “It's just a model”



### TCP/IP Reference Model

## TCP/IP Reference Model

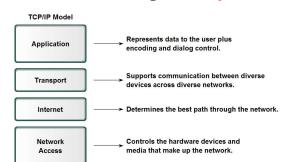
- ❖ The **TCP/IP Model** are also called as **Internet Protocol Suite**
- ✓ Describes a set of general design guidelines and implementations of specific networking protocols to enable computers to communicate over a network.
- ❖ **TCP/IP** provides **end-to-end connectivity** specifying
  - ✓ how data should be **formatted, addressed, transmitted, routed and received** at the destination.

## TCP/IP Reference Model

- ❖ TCP/IP Model is the **set of computer network communications protocols** and a **description framework used for the Internet** and other similar networks.
- ❖ The **TCP/IP Model** was created in the **1970s by DARPA**, an agency of the United States Department of Defense (DOD).

## TCP/IP Layers

- ❖ The **layers in the TCP/IP protocol suite do not exactly match** those in the OSI model.
- ❖ The **original TCP/IP protocol suite** was defined as having **four layers**:
  - ✓ **Host-to-network Layer**,
  - ✓ **Internet Layer**,
  - ✓ **Transport Layer**,
  - ✓ **Application Layer**.

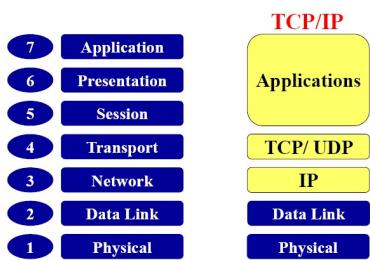


## TCP/IP Layers

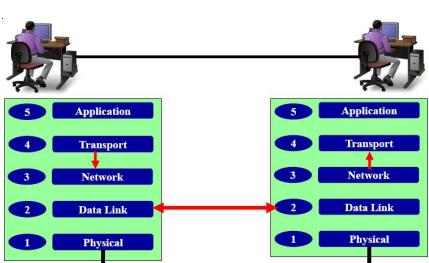
- When TCP/IP is compared to OSI, we can say that the **TCP/IP protocol suite is made of five layers:**

✓ <b>Application Layer.</b>	5	<b>Applications</b>	User service and interface
✓ <b>Transport Layer</b>	4	<b>Transport</b>	Process delivery + Error (TCP/UDP) Reliable end-to-end (whole message)
✓ <b>Network Layer</b>	3	<b>Network</b>	Move packets from source to destination Packet end-to-end (across network)
✓ <b>Data link Layer</b>	2	<b>Data Link</b>	Provide frames Node-to-node (same network segment)
✓ <b>Physical Layer</b>	1	<b>Physical</b>	Transmission bit streams (mechanical and electrical spec)

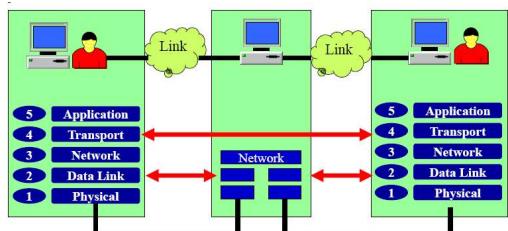
## Comparison of OSI TCP/IP Reference Model



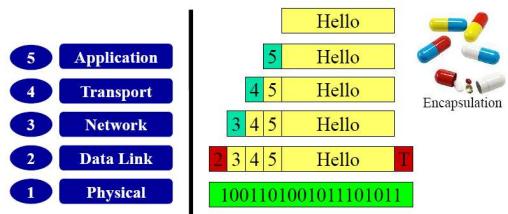
## Direct Connection



### Connection via Intermediate nodes

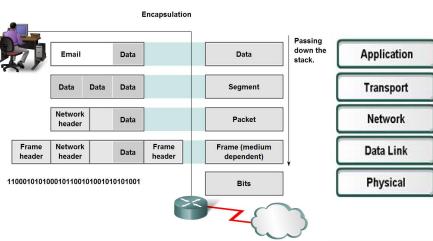


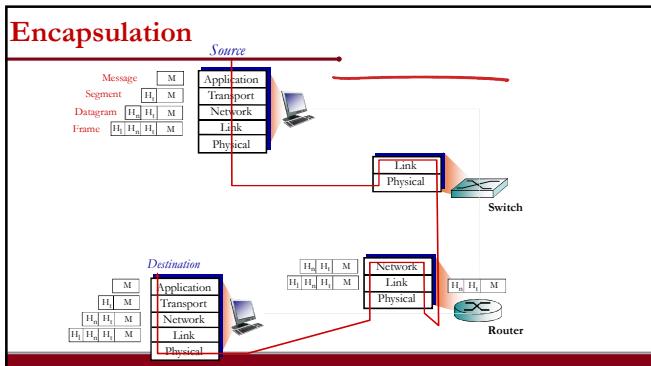
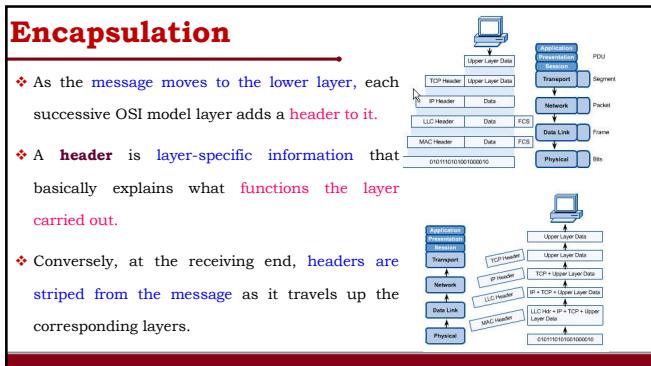
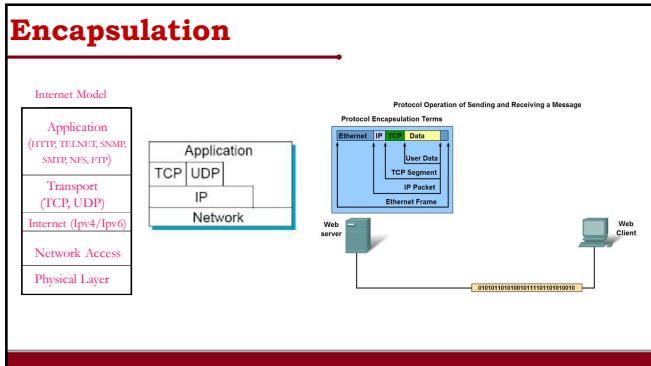
### Data Flow in a station



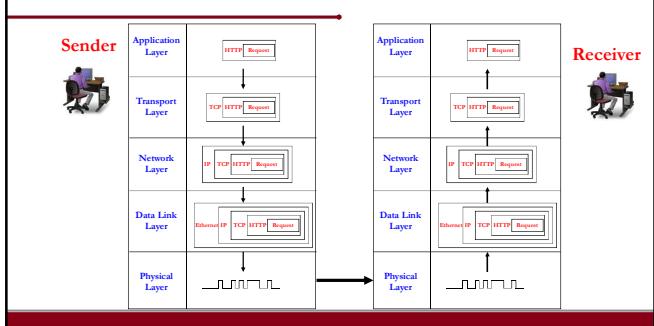
### Encapsulation

- ❖ Encapsulation is the process of embedding a header or trailer



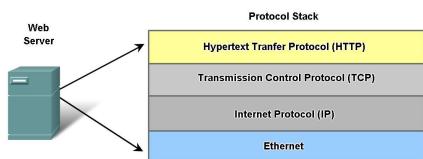


## Encapsulation



## Internet Protocol Stack

- A list of protocols used by a certain system, **one protocol per layer**, is called a **protocol stack**.



## TCP/IP and the OSI

OSI Layers	TCP/IP Layers	TCP/IP Protocols					
Application Layer	Application Layer	HTTP	FTP	Telnet	SMTP	DNS	
Presentation Layer		TCP		UDP			
Session Layer		IP					
Transport Layer	Transport Layer	Ethernet	Token Ring	Other Link-Layer Protocols			
Network Layer	Network Layer						
Data Link Layer	Network Interface Layer						
Physical Layer							



## Concept : OSI and TCP/IP

---

### Concept : OSI and TCP/IP

- ❖ Open System Interconnection(OSI) was developed by ISO as a first step toward international standardization of the protocol used in various layers.
- ❖ It deals with connecting open system..

- ❖ Transport Control Protocol /Internet Protocol (TCP/IP):
  - ✓ TCP is used in connection with IP and operates at the transport layer.
  - ✓ IP is the set of convention used to pass packets from one host to another.

### Similarities OSI and TCP/IP

- ❖ Both the **Models** are based on the **concept** of a stack of independent protocols.
- ❖ The **functionality of the layers** is roughly similar.
- ✓ They **share similar architecture**. i.e Both of them are constructed with layers.
- ✓ They **share a common application layer**. Both of the models share a common "application layer". However, Differs in services depending upon application

## Comparison OSI and TCP/IP

---



---

---

---

---

---

---

OSI	TCP/IP
<ul style="list-style-type: none"> <li>❖ OSI makes the distinction between Services, Interfaces, and Protocol.</li> <li>❖ The OSI model was devised before the protocols were invented. It can be made to work in diverse heterogeneous networks.</li> </ul>	<ul style="list-style-type: none"> <li>❖ TCP/IP does not originally clearly distinguish between services, interface, and protocol.</li> <li>❖ TCP/IP model was just a description of the existing protocols. The model and the protocol fit perfectly.</li> </ul>

---



---

---

---

---

---

---

OSI	TCP/IP
<ul style="list-style-type: none"> <li>❖ The OSI model supports both connectionless and connection-oriented communication in the network layer,</li> <li>✓ Only connection-oriented communication in the transport layer.</li> </ul>	<ul style="list-style-type: none"> <li>❖ TCP/IP model has only connectionless mode in the network layer</li> <li>✓ Supports both connectionless and conn-oriented comm.. modes in the transport layer, giving the user choice.</li> </ul>

---



---

---

---

---

---

---

<b>OSI</b>	<b>TCP/IP</b>
<ul style="list-style-type: none"> <li>❖ OSI emphasis on providing a <b>reliable data transfer service</b>,</li> <li>✓ Each layer of the OSI model detects and handles errors, all data transmitted includes checksums.</li> <li>✓ The transport layer <b>checks source-destination reliability</b>.</li> </ul>	<ul style="list-style-type: none"> <li>❖ TCP/IP treats <b>reliability</b> as an <b>end-to-end Problem</b>.</li> <li>✓ The transport layer handles all <b>error detection and recovery</b>,</li> <li>✓ it uses <b>checksums, timeouts and acknowledgments</b> to control transmissions and provides end-to-end verification.</li> </ul>

<b>OSI</b>	<b>TCP/IP</b>
<ul style="list-style-type: none"> <li>❖ OSI has seven layers</li> <li>❖ Host on OSI implementations do not handle <b>network operations</b>.</li> </ul>	<ul style="list-style-type: none"> <li>❖ TCP/IP has four layers</li> <li>❖ TCP/IP hosts participate in most network protocols.</li> </ul>

<b>Goals:</b>
<ul style="list-style-type: none"> <li>❖ Networks Software Model</li> <li>❖ Layering Concept <ul style="list-style-type: none"> <li>✓ Open System Interconnection <ul style="list-style-type: none"> <li>• Specific Responsibilities of each layer</li> </ul> </li> <li>✓ TCP/IP Reference Model</li> </ul> </li> <li>❖ Comparison OSI and TCP/IP</li> <li>❖ Analyzing the TCP/IP Using Wireshark (Active Learning)</li> </ul>

## CCN: Network Applications

**Dr. E.SURESH BABU**

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Outline

- ❖ Conceptual And Implementation Aspects Of Network Applications
- ❖ Application-layer Concepts
  - ✓ Network services required by applications,
  - ✓ Clients And Servers Concepts
  - ✓ Transport-layer Interfaces.
- ❖ Background Information- Network Programming
- ❖ Brief Look At Network Programming Using Sockets.
- ❖ Applying Sockets using Python,
- ❖ Building the Python's modules for Networked Applications.

### User Oriented Level: Application Layer

## Network Application

- ❖ The **network applications** have been the **driving force** behind the **Internet's success** motivating people in
  - ✓ **Homes, Schools, Governments, and Businesses** to make the **Internet an integral part** of their daily activities.

The **Internet** does not provide **services**. Instead, the **Internet** only provides **communication**, and **application programs** provide all **services**.

---



---



---



---



---



---



---

## Network Applications

- ❖ **Internet Applications** include the
  - ✓ **Classic Text-based Applications(1970s -1980s): Text Email, Remote Access To Computers, File Transfers**
  - ✓ **Killer application (mid-1990s): The World Wide Web, Web Surfing, Search, and Electronic Commerce.**

---



---



---



---



---



---



---

## Network Applications

- ❖ Internet Applications include the
  - ✓ **Voice And Video Applications(2000) :**
    - **Voice-over-IP (VoIP) and video conferencing over IP** such as **Skype**;
    - **User-generated video distribution** such as **YouTube**;
    - **Movies on demand** such as **Netflix**.

---



---



---



---



---



---



---

## Network Applications

- ❖ The emergence of a new generation (2010)
- ✓ Social Networking Applications, such as **Facebook and Twitter**,

---



---



---



---



---



---

## Creating a Network Applications

---



---



---



---



---



---

## Application Layer:

- ❖ The application layer is responsible for **creating** set of applications running on the **End Hosts**.
- ✓ User **write Applications Programs** that run on (different) end systems
- ✓ Communicate over network
- ✓ E.g., **Web Server software** communicates with **Browser Software**

---



---



---



---



---



---

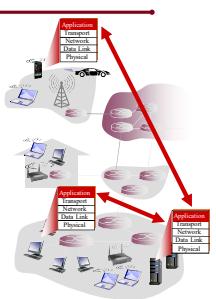
## Application Layer:

---

- ❖ The **application layer** is responsible for **providing services to the user**.
  - ❖ No need to write software for network-core devices
    - ✓ Network-core devices do not run user applications
- 
- 
- 
- 
- 
- 

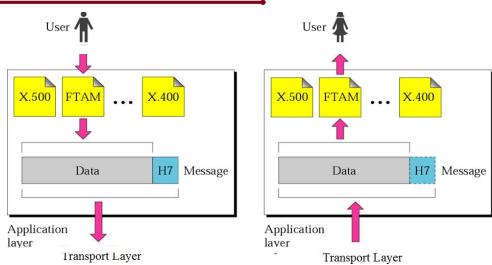
## Application Layer:

---



## Application Layer:

---



## Application Layer:

- ❖ The **Application Layer** is a particularly good place to start **our study of protocols**.
- ❖ Some of the better-known **Application layer protocols** are
  - ♦ **DNS (Domain Name System)** for resolving Internet domain names.
  - ♦ **HTTP (Hypertext Transfer Protocol)** which is Web's application layer protocol
  - ♦ **FTP (File Transfer Protocol)** for file transfers.
  - ♦ **SMTP (Simple Mail Transfer Protocol)** for e-mail.
  - ♦ **NFS (Network File System)** for file sharing in UNIX networks.
  - ♦ **Telnet** for terminal emulation.

---



---



---



---



---



---



---



---

## Application Architectures

---



---



---



---



---



---



---



---

## Application Architecture

- ❖ The **Network Architecture** is fixed and provides a **specific set of services to applications**. (e.g., **the five-layer Internet architecture** )
- ❖ The **application architecture** are designed by the **application developer** and dictates
  - ✓ how the **application is structured** over the various end systems.

---



---



---



---



---



---



---



---

## Application Architecture

- ❖ In **Application Architecture**, There are **Two predominant architectural paradigms** used in modern network applications:

  1. **The Client-Server Architecture**
  2. **The Peer-to-Peer (P2P) Architecture (Self Study)**

---



---



---



---



---



---

## The Client-Server Architecture

---



---



---



---



---



---

## Client/Server Model

- ❖ **Client-Server Model** used by applications to **establish the communication**
- ❖ **One application acts as a SERVER**
  - ✓ Starts execution first
  - ✓ Awaits contact from the client
- ❖ **The other application becomes a CLIENT**
  - ✓ Starts after server is running
  - ✓ Initiates contact

---



---



---



---

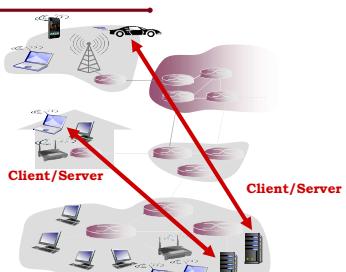


---



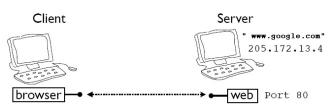
---

## Client/Server Model



## Client/Server Model

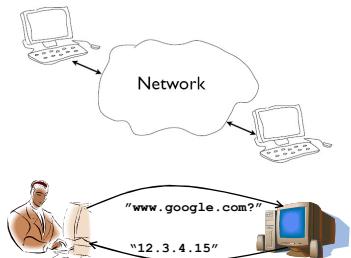
- ❖ **Client-Server** means doing **different things** with **different people**
- ✓ **Servers** wait for **incoming connections** and **provide a service**  
(e.g., web, mail, etc.)
- ✓ **Clients** make **connections to servers**



## Client/Server Model

- ❖ Important Concept: **Once communication** has been established, **data (e.g., requests and responses) can flow** in either direction between a **client and server**

### Logical Communication : Client/Server Model



### Client/Server Concepts : Example

- Suppose Client sends a request message (e.g., HTTP)

**GET /index.html HTTP/1.0**

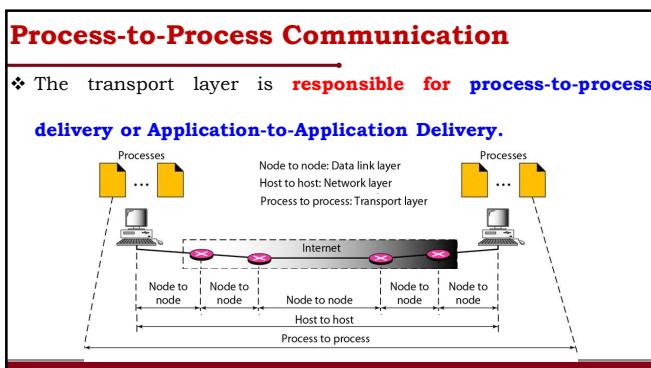
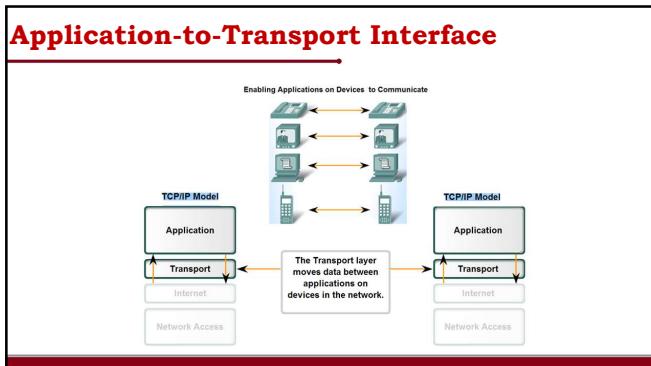
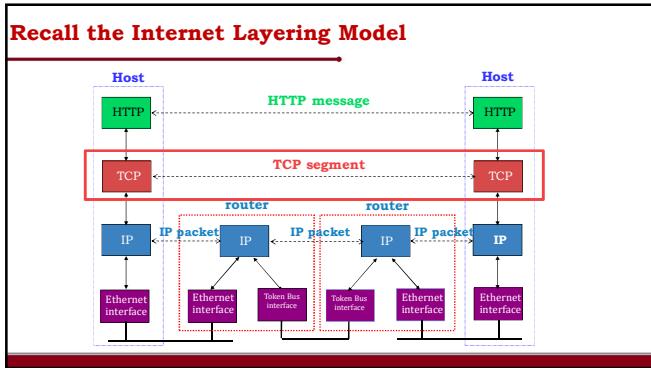
- Server sends back a response message

**HTTP/1.0 200 OK**  
**Content-type: text/html**  
**Content-length: 48823**  
**<HTML>**  
**...**

- The **exact format** depends on the application

### A Summary Of The Client-server Model.

Server Application	Client Application
Starts first	Starts second
Does not need to know which client will contact it	Must know which server to contact
Waits passively and arbitrarily long for contact from a client	Initiates a contact whenever communication is needed
Communicates with a client by both sending and receiving data	Communicates with a server by sending and receiving data
Stays running after servicing one client, and waits for another	May terminate after interacting with a server



## Two Basic Internet Communication Paradigms

- ❖ The Internet supports **two basic communication paradigms**

  1. **A Stream Paradigm (TCP)**
  2. **A Message Paradigm (UDP).**

---



---



---



---



---



---

## The Two Paradigms That Internet Applications Use.

Stream Paradigm	Message Paradigm
Connection-oriented	Connectionless
1-to-1 communication	Many-to-many communication
Sequence of individual bytes	Sequence of individual messages
Arbitrary length transfer	Each message limited to 64 Kbytes
Used by most applications	Used for multimedia applications
Built on TCP protocol	Built on UDP protocol

---



---



---



---



---



---

## Addressing: Port Numbers

---



---



---



---



---



---

## Addressing: Port Numbers

- ❖ One of the specific responsibilities of transport layer protocol is to **create a process-to-process communication**;
- ❖ To accomplish the process-to-process communication
  - ✓ Transport layer protocol uses **port numbers**.
- ❖ **Port numbers** provide **end-to-end addresses** at the transport layer

---



---



---



---



---



---

## Addressing: Port Numbers

- ❖ Port Numbers play an important role in the **TCP and UDP protocols**.

---



---



---



---

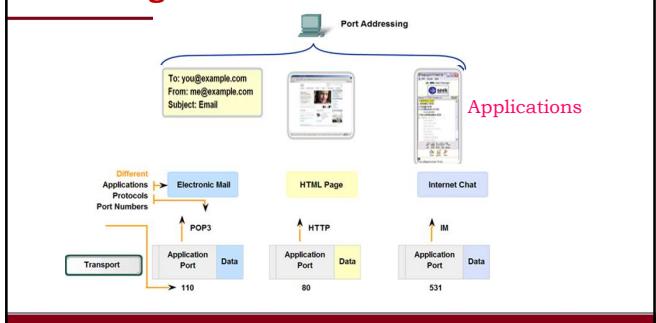


---



---

## Addressing: Port Numbers




---



---



---



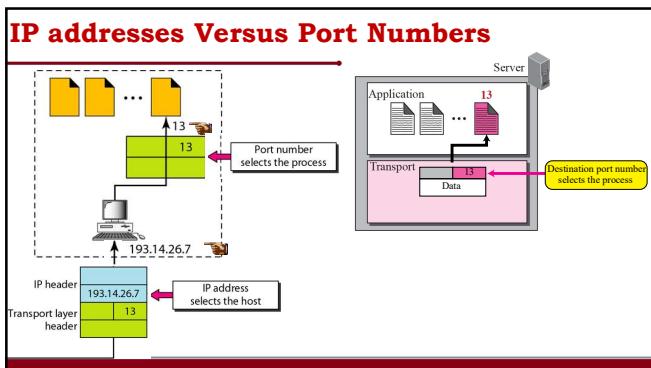
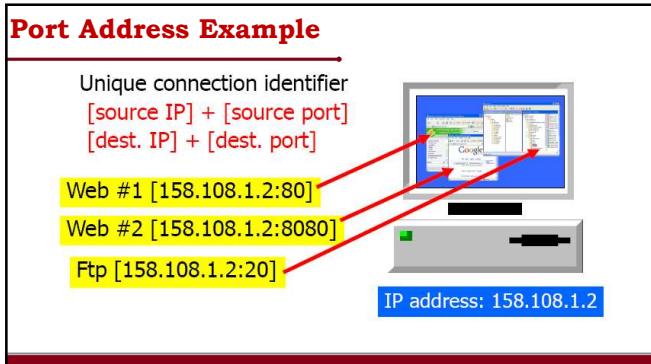
---



---



---



**Some Well-Known Ports**

Port	Protocol	UDP	TCP	Description
7	Echo	✓		Echoes back a received datagram
9	Discard	✓		Discards any datagram that is received
11	Users	✓	✓	Active users
13	Daytime	✓	✓	Returns the date and the time
17	Quote	✓	✓	Returns a quote of the day
19	Chargen	✓	✓	Returns a string of characters
20, 21	FTP		✓	File Transfer Protocol
23	TELNET		✓	Terminal Network
25	SMTP		✓	Simple Mail Transfer Protocol
53	DNS	✓	✓	Domain Name Service
67	DHCP	✓	✓	Dynamic Host Configuration Protocol
69	TFTP	✓		Trivial File Transfer Protocol
80	HTTP		✓	HyperText Transfer Protocol
111	RPC	✓	✓	Remote Procedure Call
123	NTP	✓	✓	Network Time Protocol
161, 162	SNMP		✓	Simple Network Management Protocol

## Addressing: Socket Address

---



---

---

---

---

---

### What is Socket?

---

- ❖ Sockets are **computer networking data structures** which embody the concept of the "**Communication Endpoint**"
- ❖ Each **Endpoint of a network connection** is always represented by a **host and port #**




---

---

---

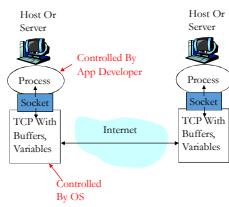
---

---

### Purpose...

---

- ❖ Socket API is the **programming interface** you need to use for **transmitting and receiving messages**
- ❖ Socket API help you
  - ✓ To **send and receive the messages**,
  - ✓ **Act as a buffer**,
  - ✓ Provides an interface between **user** and **lower layers**.




---

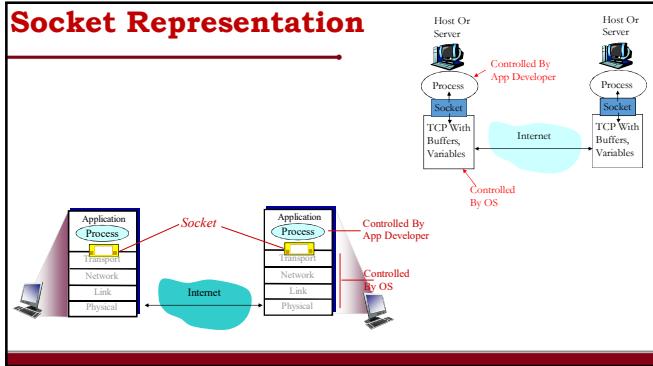
---

---

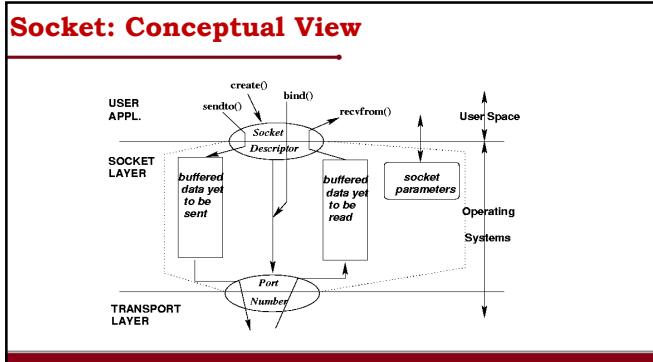
---

---

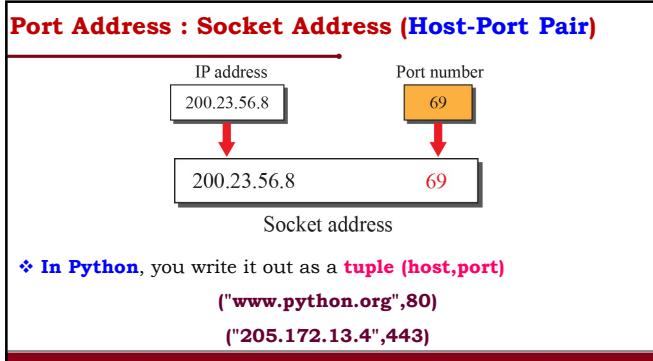
## Socket Representation



## Socket: Conceptual View



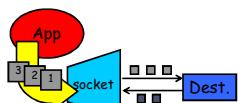
## Port Address : Socket Address (Host-Port Pair)



## Two Essential Types of Sockets

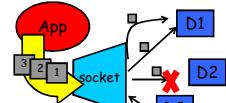
### ♦ SOCK\_STREAM

- ✓ TCP
- ✓ Reliable Delivery
- ✓ In-order Guaranteed
- ✓ Connection-oriented
- ✓ Bidirectional



### ♦ SOCK\_DGRAM

- ✓ UDP
- ✓ Unreliable Delivery
- ✓ No Order Guarantees
- ✓ Connection Less



## Outline

- ♦ Conceptual And Implementation Aspects Of Network Applications
- ♦ Application-layer Concepts
  - ✓ Network services required by applications,
  - ✓ Clients And Servers Concepts
  - ✓ Transport-layer Interfaces.
- ♦ Background Information- Network Programming
- ♦ Brief Look At Network Programming Using Sockets.
- ♦ Applying Sockets using Python,
- ♦ Building the Python's modules for Networked Applications.



**Thank You**

## CCN: Network Applications Protocols

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

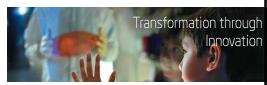
National Institute of Technology, Warangal.

Warangal, TS, India.



### Outline

- ❖ Application Layer Protocols
- ❖ Web Protocols
- ✓ Overview of HTTP



## Application Layer Protocols

### Why is Application Layer Protocols are Required

- ❖ Usually, Network Processes communicate with each other by sending **MESSAGES** into sockets.
- ✓ **How these messages are structured?**
- ✓ **What are the meanings of the various fields in the messages?**
- ✓ **When do the processes send the messages?**
- ❖ **Application-layer Protocol** will define the answer to all the above questions

---



---



---



---



---



---

### Purpose of Application Layer Protocols

- ❖ Specifically, **Application-layer Protocol** specifies the details, such as:
- ✓ The **syntax and semantics of messages** that can be exchanged
- ✓ Whether the **client or server initiates interaction**
- ✓ **Actions to be taken** if an **error arises**
- ✓ How the **two sides know when to terminate communication**

---



---



---



---



---



---

### Application Layer Protocols

- ❖ An **Application-layer Protocol** is only one piece of a **network application**
- ❖ **Application-layer Protocols** specify two aspects of interaction
  - 1. Representation**
  - 2. Transfer.**

---



---



---



---



---



---

### Two Key Aspects of an Application-layer Protocol.

Aspect	Description
Data Representation	Syntax of data items that are exchanged, specific form used during transfer, translation of integers, characters, and files between computers
Data Transfer	Interaction between client and server, message syntax and semantics, valid and invalid exchange error handling, termination of interaction

- ❖ Web uses separate protocols to describe web page syntax and web page transfer.

### Application Protocol Examples

- ❖ Web Browsing
- ❖ Email
- ❖ File Transfer
- ❖ Remote Login And Remote Desktop
- ❖ Domain Name System (Name Lookup)

### Application Layer Protocols : Web Protocols

### Introduction to Web

- ❖ In Early 1990s, a major new application **World Wide Web** had arrived as the first Internet application.
- ❖ The **World Wide Web** is one of the most widely used **services** in the Internet.
- ❖ Web servers as a **platform** for many killer applications emerging after 2003, including **YouTube, Gmail, and Facebook**.

---



---



---



---



---



---

### Advantages of Web

- ❖ Using Web,
- ✓ It is relatively easy for any individual to make information available over the Web
- ✓ Everyone can become a publisher at extremely low cost.
- ✓ Hyperlinks and search engines help us navigate through an ocean of Web sites.
- ✓ Graphics stimulate our senses.
- ✓ Forms, JavaScript, Java applets, and many other devices enable us to interact with pages and sites.

---



---



---



---



---



---

### Three Key Standards- World Wide Web Service Uses'

- ❖ Web is complex, many **protocol standards** have been devised to specify various aspects and details

Standard	Purpose
HyperText Markup Language (HTML)	A representation standard used to specify the contents and layout of a web page
Uniform Resource Locator (URL)	A representation standard that specifies the format and meaning of web page identifiers
HyperText Transfer Protocol (HTTP)	A transfer protocol that specifies how a browser interacts with a web server to transfer data

---



---



---



---



---



---

## Hyper Text Markup Language (HTML)

- ❖ Representation standard for **multimedia documents**
- ❖ Specifies document is entirely in **printable text**
- ❖ Document contains **markup guidelines** rather than precise, detailed formatting or typesetting instructions

---

---

---

---

---

---

## Uniform Resource Locator (URL)

**protocol:// computer\_name : port / document\_name ? parameters**

**http://nitw.ac.in/cse/faculty.html**

**http://10.45.10.1 : 80/cse/faculty.html**

---

---

---

---

---

---

## World Wide Web- Review

- ❖ A **Web page** (also called a document) consists of **objects**.
- ✓ An object can be **HTML file**, a **JPEG image**, a **Java applet**, or a **video clip**
- ✓ All the **objects** are addressable by a **single URL**.
- ✓ Most Web pages consist of a **base HTML file** and **several referenced objects**.

---

---

---

---

---

---

## World Wide Web- Review

www.someschool.edu/someDept/pic.gif

Host Name

Path Name

---

---

---

---

---

---

## Overview of HTTP

---

---

---

---

---

---

## Hyper Text Transfer Protocol (HTTP)

- ❖ The **Hyper Text Transfer Protocol (HTTP)** is a Web's application-layer protocol.
- ❖ HTTP is the **heart of the Web**.
- ❖ The HTTP is the **primary transfer protocol** that a **browser uses to interact with a web server**.

---

---

---

---

---

---

### Hyper Text Transfer Protocol (HTTP)

- ❖ HTTP usually implemented in two programs: a **client program** and a **server program**, which are two different end systems that talk to each other by **exchanging HTTP messages**.
- ❖ In the client-server model, a **browser** is a client that **extracts a server name from a URL** and **contacts the server**.

---



---



---



---



---



---

### Hyper Text Transfer Protocol (HTTP)

- ❖ **HTTP can be characterized as follows:**
- ✓ **Specifies format and meaning of messages**
- ✓ **Each message represented as text**
- ✓ **Transfers arbitrary binary data**
- ✓ **Can download or upload data**

---



---



---



---



---



---

### How HTTP Works ?

- ❖ HTTP defines
- ✓ How **Web clients** request **Web pages** from **Web servers** and
- ✓ How **servers** transfer **Web pages** to **clients**.

---



---



---



---



---



---

## HTTP used Client/Server Model



## HTTP used Client/Server Model

- ❖ **Client:** Web Browser requests, receives, (using HTTP protocol) and “displays” Web objects
- ❖ **Server:** Web server sends (using HTTP protocol) objects in response to requests

## Underlying Protocol

- ❖ HTTP uses **TCP** as its underlying **transport protocol**
- ❖ The **HTTP client** first initiates a **TCP connection** with the **server**.
- ❖ Once the **connection is established**,
- ✓ The **browser and the server processes** access **TCP through their socket interfaces**.

### HTTP uses TCP as a underlying protocol

- ❖ Client initiates **TCP connection** (creates socket) to server, port 80
- ❖ Server accepts **TCP connection** from client
- ❖ **HTTP messages** (application-layer protocol messages) **exchanged** between **browser (HTTP client)** and **Web server (HTTP server)**
- ❖ **TCP connection closed**

---



---



---



---



---



---

### HTTP uses TCP as a underlying protocol

- ❖ TCP provides a **reliable data transfer service** to HTTP.
- ✓ **HTTP need not worry** about **lost data or reordering of data** within the network.
- ❖ It is job of **TCP and the other protocols** in the lower layers of the protocol stack.

---



---



---



---



---



---

### Observation

- ❖ HTTP is “**Stateless**” Protocol
- ✓ **Server maintains no information about past client requests**
- ❖ If the HTTP protocols that **maintain past history (state)**
- ✓ **It is complex !**
- ✓ It is required, when **server/client crashes**, their views of “**state**” **may be inconsistent**, must be **reconciled**

---



---



---



---



---



---

### Non-Persistent and Persistent Connections

- ❖ When this **client-server interaction** is taking place over **TCP**,
  1. **Each request/response pair** be sent over a **separate TCP connection** (**Non-persistent Connections**).
  2. **All the request/response pair** should be sent over the **same TCP connection**. (**Persistent Connections**)

---



---



---



---



---



---

### Observation

- ❖ **HTTP** uses both **non-persistent connections** and **persistent connections**.
- ✓ By **default**, **HTTP uses persistent connections**
- ✓ **HTTP clients and servers must be configured** to use **non-persistent connections**

---



---



---



---



---



---

### Non-Persistent Connections

- ❖ **At most one object** sent over TCP connection
  - ✓ **connection then closed**
- ❖ Downloading **multiple objects** required **multiple connections**

---



---



---



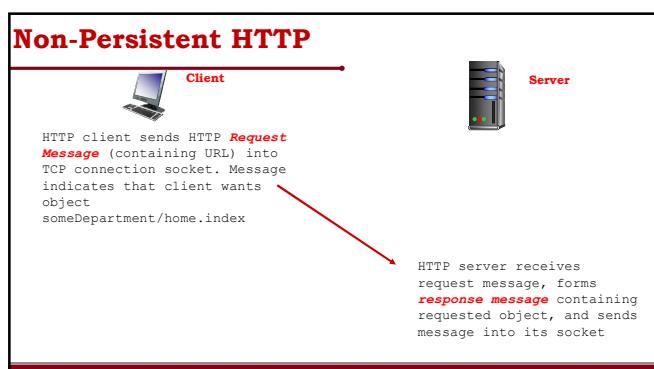
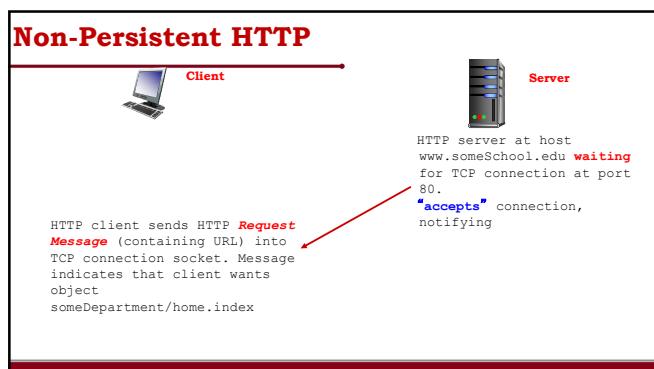
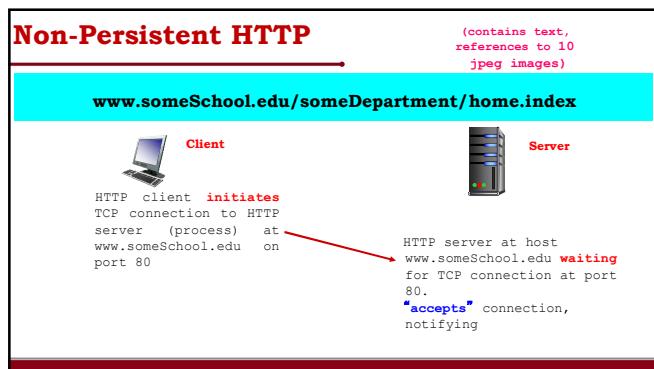
---

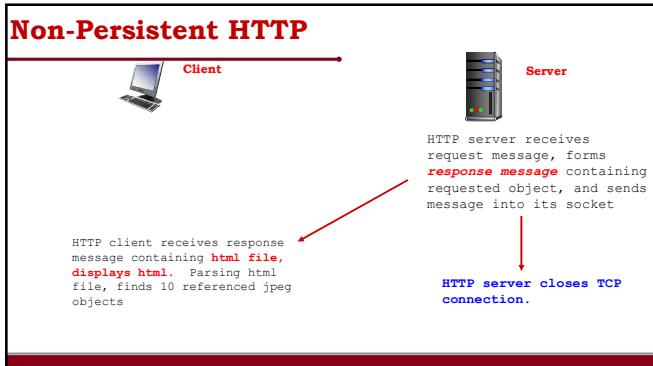


---



---






---

---

---

---

---

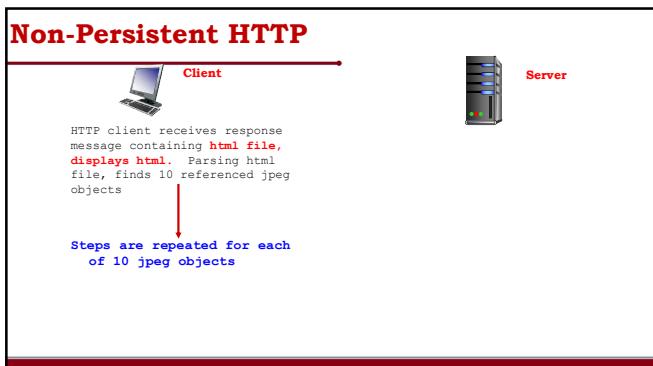
---

---

---

---

---




---

---

---

---

---

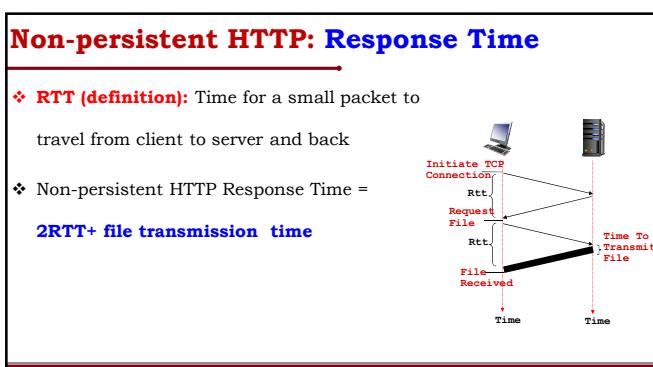
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

### Non-Persistent HTTP : Limitations

- ❖ Requires **2 RTTs per object**
- ❖ **OS overhead** for each TCP connection
- ❖ Browsers often **open parallel TCP connections** to fetch referenced objects

---



---



---



---



---



---

### Persistent HTTP

- ❖ **Multiple Objects** can be sent over **single TCP connection** between client, server.
- ✓ Server leaves **connection open** after sending response
- ✓ Subsequent **HTTP messages** between same client/server sent over **open connection**
- ✓ Client **sends requests** as soon as it encounters a referenced object

---



---



---



---



---



---

### Persistent HTTP

- ❖ Requires only **one RTT** for all the referenced objects

---



---



---



---



---



---

## HTTP Message Format

- ❖ Two types of **HTTP messages: Request, Response**
- ❖ HTTP request message: **ASCII (human-readable format)**

---

---

---

---

---

---

## The Four Major HTTP Request Types

- ❖ Once it establishes a connection, a browser sends an **HTTP request to the server**

Request	Description
GET	Requests a document; server responds by sending status information followed by a copy of the document
HEAD	Requests status information; server responds by sending status information, but does not send a copy of the document
POST	Sends data to a server; the server appends the data to a specified item (e.g., a message is appended to a list)
PUT	Sends data to a server; the server uses the data to completely replace the specified item (i.e., overwrites the previous data)

---

---

---

---

---

---

## HTTP Message Format

request line  
(GET, POST,  
HEAD commands)

Header Lines

carriage return,  
line feed at start  
of line indicates  
end of header lines

```

GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n

```

---

---

---

---

---

---

## HTTP Response Message

```

Status Line
(Protocol
Status Code
Status Phrase) HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
ETag: "17dc6-a5c-be716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
1\r\n
\r\n
Data, E.G.,
Requested
HTML File
data data data data data ...

```

## HTTP Response Status Codes

- ❖ **Status Code Appears** In 1st Line In Server-to-client Response Message.

Status Code	Corresponding Status String
200	OK
400	Bad Request
404	Not Found

## Outline

- ❖ Application Layer Protocols
- ❖ Web Protocols
- ✓ Overview of HTTP



Transformation through Innovation

**Thank You**

## CCN: Network Applications- Electronic Mail

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



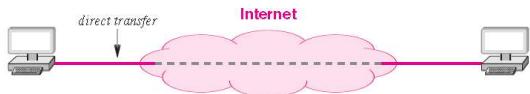
### Outline

- ❖ Electronic Mail
- ❖ Mail Access Protocols
  - ✓ Post Office Protocol—Version 3 (POP3)
  - ✓ Internet Mail Access Protocol (IMAP)
  - ✓ Web based Mail

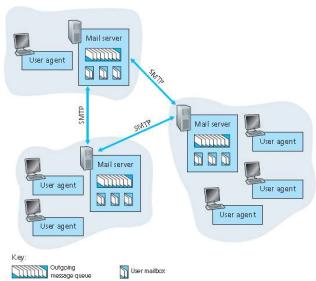
## Electronic Mail

## Electronic Mail in the Internet

- ❖ **Electronic Mail** has been around since the beginning of the Internet.
- ❖ **E-Mail** was the most popular application and remains one of the most widely used **Internet applications**.



## Electronic Mail in the Internet



## Electronic Mail in the Internet

- ❖ **Internet Mail System** has three **Major Components**:

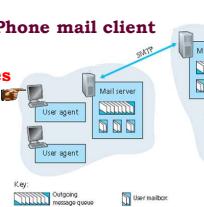
1. User Agents,
2. Mail Servers,
3. Simple Mail Transfer Protocol (SMTP).

## User Agents

- ❖ **User Agent** is usually a “**Mail Reader**”
- ✓ **Composing, Editing, Reading** the **Mail Messages**
- ✓ **EX:** **Outlook, Thunderbird, iPhone mail client**

### ❖ Outgoing, Incoming Messages

Stored On **Server**




---

---

---

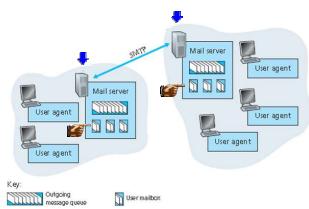
---

---

---

## Mail Servers:

- ❖ **Mailbox** contains **incoming messages for user**




---

---

---

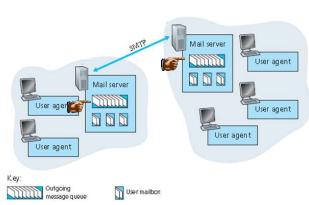
---

---

---

## Mail Servers:

- ❖ **Message Queue** of **outgoing (to be sent) mail messages**




---

---

---

---

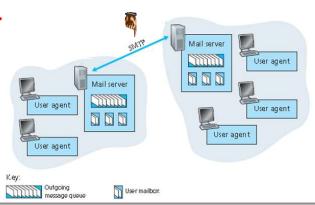
---

---

## SMTP Protocol

- ❖ SMTP Protocol between mail servers to send email messages

- ✓ Client: Sending mail server
- ✓ Server: Receiving mail server

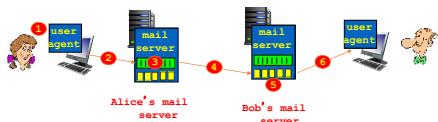


## SMTP Protocol

- ❖ SMTP is the **principal application-layer protocol** for Internet electronic mail.
- ❖ SMTP uses the **reliable data transfer service of TCP, port 25**
- ✓ **Transferring the mail** from the sender's mail server to the recipient's mail server.
- ❖ SMTP is much **older than HTTP**.

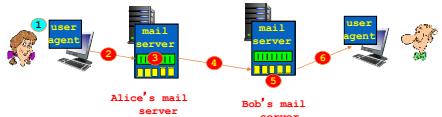
## SMTP Protocol Scenario:

- ❖ Alice sends message to Bob



### SMTP Protocol Scenario:

❖ Alice sends message to Bob



1) Alice uses User Agent to compose message "to" bob@someschool.edu

---



---



---



---



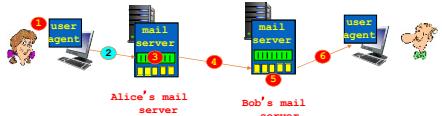
---



---

### SMTP Protocol Scenario:

❖ Alice sends message to Bob



2) Alice's User Agent sends message to her **Mail Server**; Message placed in **Message Queue**

---



---



---



---



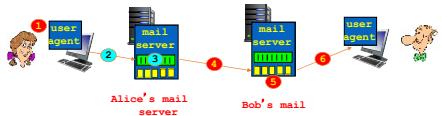
---



---

### SMTP Protocol Scenario:

❖ Alice sends message to Bob



3) Client side of SMTP opens **TCP connection with Bob's mail server**

---



---



---



---



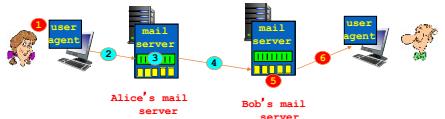
---



---

### SMTP Protocol Scenario:

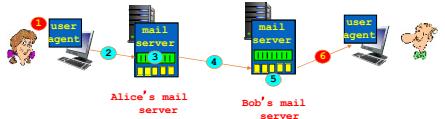
- ❖ Alice sends message to Bob



4.) SMTP client sends **Alice's message over the TCP connection**

### SMTP Protocol Scenario:

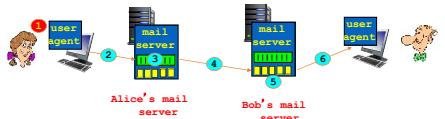
- ❖ Alice sends message to Bob



5.) Bob's mail server places the **message in Bob's mailbox**

### SMTP Protocol Scenario:

- ❖ Alice sends message to Bob



5.) Bob invokes his **user agent to read message**

## Sample SMTP interaction

- ❖ Messages exchanged between an **SMTP client (C)** and an **SMTP server (S)**. The **hostname of the client** is **crepes.Fr** and the **hostname of the server** is **hamburger.Edu**.

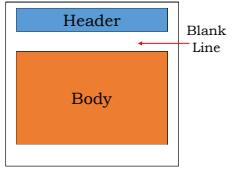
```

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

```

## Mail Message Formats

- ❖ Every header must have a
  - ✓ **From:** header line
  - ✓ **To:** header line;
  - ✓ **Subject:** Header line as well as other optional header lines.

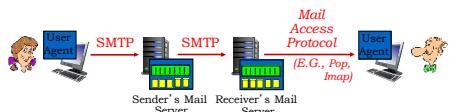


From: alice@crepes.fr  
 To: bob@hamburger.edu  
 Subject: Searching for the meaning of life.

## Mail Access Protocols

## Mail Access Protocols

- Once **SMTP delivers the message** from Alice's mail server to Bob's mail server, the **message is placed in Bob's mailbox**.



❖ **SMTP** only delivery to Receiver's Mail Server

❖ **Mail Access Protocol** are used to retrieval from Mail Server

## Mail Access Protocols

- Access Protocols** have the following characteristics:
- ✓ Provide access to a **user's mailbox**
  - ✓ Permit a user to **view headers, download, delete, or send individual messages**
  - ✓ Client runs on **user's personal computer**
  - ✓ Server runs on a computer that stores **user's mailbox**

## Mail Access Protocols

- There are **popular mail access protocols**
- ✓ **Post Office Protocol—Version 3 (POP3)**
  - ✓ **Internet Mail Access Protocol (IMAP)**
  - ✓ **HTTP**

### Post Office Protocol-3 (POP3)

- ❖ POP3 is an **extremely simple mail access protocol**.
- ❖ POP3 **begins**
  - ✓ when the user agent (the client) **opens a TCP connection** to the mail server (the server) on port 110.
  - ✓ With the **TCP connection established**.
- ❖ POP3 **progresses through three phases:** Authorization, Transaction, And Update.

### POP3: Authorization Phase

- ❖ During this phase, the user agent sends a **username and a password** to **authenticate the user**.
- ❖ **Client commands:**
  - ✓ **User:** Declare username
  - ✓ **Pass:** password
- ❖ **Server Responses**
  - ✓ **+OK**
  - ✓ **-ERR**

```

S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
  
```

### POP3: Transaction Phase

- ❖ During this phase,
- ✓ The user agent **retrieves messages**;
- ✓ The user agent can also **mark messages for deletion, remove deletion marks, and obtain mail statistics**.

### POP3: Transaction Phase -Example

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK pop3 server signing off
```

### POP3: Update Phase

- ❖ This phase occurs **after the client has issued the quit command, ending the POP3 session;**
- ✓ The **mail server deletes the messages** that were marked for deletion.

### POP3: Conclusion

- ❖ POP3 uses “**download and delete**” mode
  - ✓ Bob cannot **re-read e-mail** if he changes client
- ❖ POP3 also uses “**download-and-keep**”:
  - ✓ **Maintains** copies of messages on different clients
- ❖ POP3 is stateless across sessions
  - ✓ Do not carry **state information** across POP3 sessions

### Internet Mail Access Protocol (IMAP)

- ❖ The POP3 protocol does not provide to the user to create remote folders and assign messages to folders.
- ❖ IMAP is also a mail access protocol, which has many more features than POP3.
- ✓ But, it is also significantly more complex

---



---



---



---



---



---

### Internet Mail Access Protocol (IMAP)

- ❖ An IMAP server will associate each message with a folder;
- ✓ When a message first arrives at the server, it is associated with the recipient's INBOX folder.
- ✓ The recipient can then move the message into a new, user-created folder,
- ✓ The recipient can read the message, delete the message, and so on.

---



---



---



---



---



---

### Internet Mail Access Protocol (IMAP)

- ❖ The IMAP protocol provides commands to allow users to create folders and move messages from one folder to another.
- ❖ IMAP also provides commands that allow users to search remote folders for messages matching specific criteria.
- ❖ IMAP keeps user State Across Sessions

---



---



---



---



---



---

## Web based Mail

- ❖ One of the **Web based Mail** is **HTTP**
  - ✓ **Gmail, Hotmail, Yahoo! Mail**, etc.
- ❖ Suppose,
  - ✓ The **user agent is an ordinary Web browser**, want to communicate with its **remote mailbox via HTTP**.
  - ✓ When a recipient, wants to **access a message** in his **mailbox**,

---



---



---



---



---



---

## Web based Mail

- ✓ The **e-mail message is sent from recipient mail server to Bob's browser** using the **HTTP protocol** rather than the POP3 or IMAP protocol.
- ✓ When a sender, wants to send an e-mail message,
  - The **e-mail message is sent from her browser to her mail server over HTTP**
- ✓ Sender Alice's **mail server still sends messages** to, and receives messages from, other mail servers using **SMTP**.

---



---



---



---



---



---

## Outline

- ❖ **Electronic Mail**
- ❖ **Mail Access Protocols**
  - ✓ **Post Office Protocol—Version 3 (POP3)**
  - ✓ **Internet Mail Access Protocol (IMAP)**
  - ✓ **Web based Mail**

---



---



---



---



---



---



---



---

**Thank You**

## CCN: Transport Layer Introduction

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

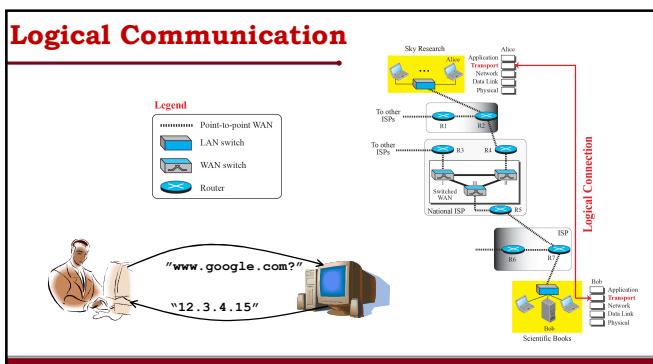
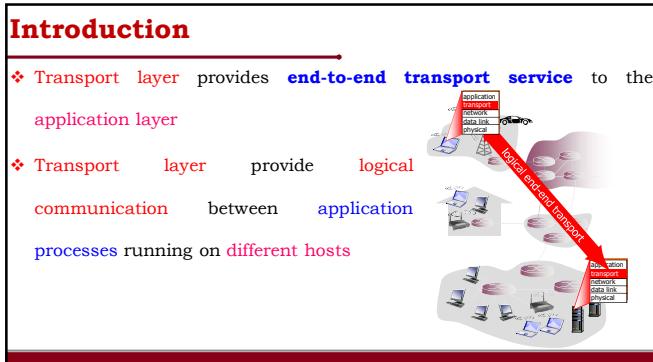
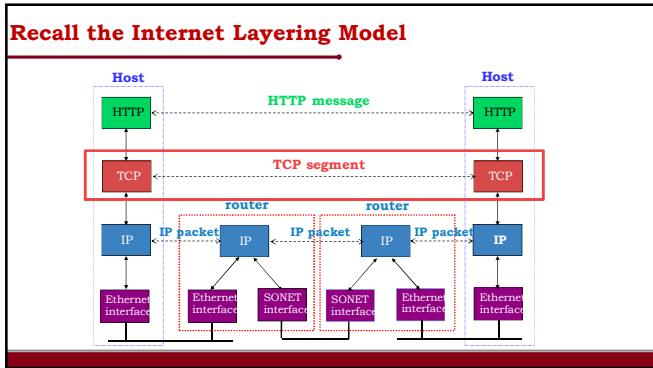
Warangal, TS, India.



### Goals:

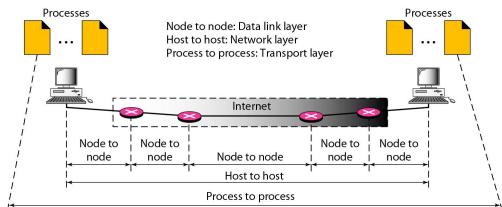
- ❖ Why do we need a Transport Layer
- ❖ Transport Layer: Introduction
  - ✓ Services Provided to the Upper Layers
  - ✓ Addressing: Port Numbers
  - ✓ Connection Establishment And Termination
  - ✓ Flow Control and Buffering
  - ✓ Multiplexing and De-Multiplexing
  - ✓ Segmentation and Reassembly

## Transport Layer: Introduction

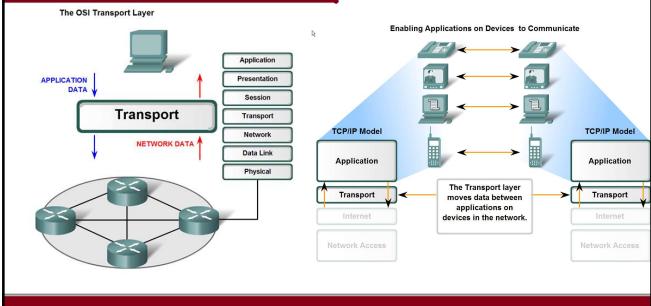


## Introduction

- The transport layer is responsible for process-to-process delivery.



## Introduction : Application-to-Application



## Introduction

- The transport layer is located between the network layer and the application layer.
- The transport layer is responsible for providing services to the application layer;
- it receives services from the network layer.

### Specific Responsibilities of Transport Layer

- ❖ Services Provided to the Upper Layers
- ❖ Control connection: connectionless or connection-oriented
- ❖ Service-point addressing (port addresses used in TCP)
- ❖ Segmentation and reassembly
- ❖ Flow control (end to end, not just a single link)
- ❖ Error control (end to end, not just a single link)
- ❖ Crash Recovery

---



---



---



---



---



---



---

### Transport Layer Services

- ❖ Guaranteed message delivery (end-to-end)
- ❖ Ordered delivery
- ❖ Rejection of duplicate messages
- ❖ Messages of arbitrary length
- ❖ Congestion control to handle network overloading
- ❖ Running of multiple application processes at the same host

---



---



---



---



---



---



---

### Services Provided to the Upper Layers

---



---



---



---



---



---

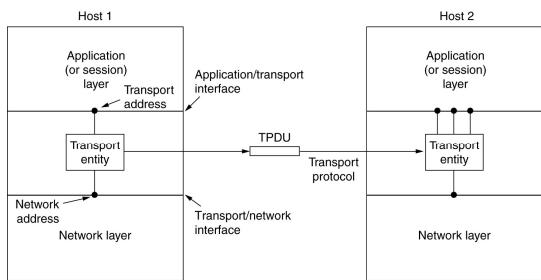


---

### Services Provided to the Upper Layers

- ❖ The **ultimate goal of the transport layer** is to provide
  - ✓ Efficient, reliable, and cost-effective service to its users
- ❖ To achieve the above goal, the **transport layer makes use** of the services provided by the **network layer**.
- ❖ The **hardware/software within the transport layer** are usually represented by with **transport entity**.
- ❖ The **transport entity** are located in the **operating system kernel**

### Services Provided to the Upper Layers



### Addressing: Port Numbers

## Addressing: Port Numbers

- ❖ One of the specific responsibilities of transport layer protocol is to **create a process-to-process communication**;
- ❖ To accomplish the process-to-process communication
  - ✓ Transport layer protocol uses **port numbers**.
- ❖ **Port numbers** provide **end-to-end addresses** at the transport layer

---



---



---



---



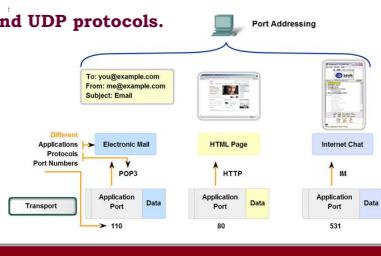
---



---

## Addressing: Port Numbers

- ❖ Identify how a **port number is represented** and describe the **role port numbers play in the TCP and UDP protocols.**




---



---



---



---



---



---

## Some Well-Known Ports

Port	Protocol	UDP	TCP	Description
7	Echo	✓		Echoes back a received datagram
9	Discard	✓		Discards any datagram that is received
11	Users	✓	✓	Active users
13	Daytime	✓	✓	Returns the date and the time
17	Quote	✓	✓	Returns a quote of the day
19	Chargen	✓	✓	Returns a string of characters
20, 21	FTP		✓	File Transfer Protocol
23	TELNET		✓	Terminal Network
25	SMTP		✓	Simple Mail Transfer Protocol
53	DNS	✓	✓	Domain Name Service
67	DHCP	✓	✓	Dynamic Host Configuration Protocol
69	TFTP	✓		Trivial File Transfer Protocol
80	HTTP		✓	HyperText Transfer Protocol
111	RPC	✓	✓	Remote Procedure Call
123	NTP	✓	✓	Network Time Protocol
161, 162	SNMP		✓	Simple Network Management Protocol

---



---



---



---

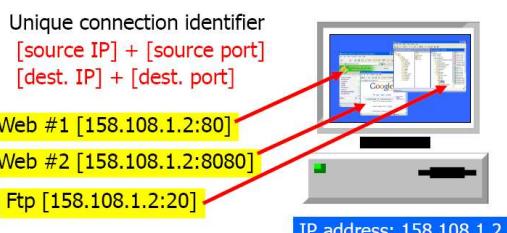


---

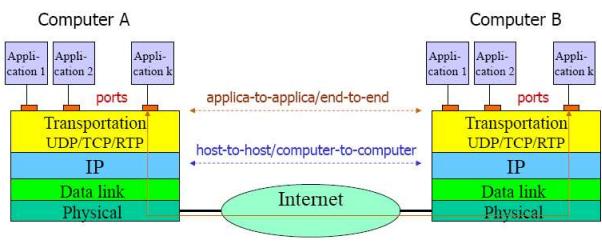


---

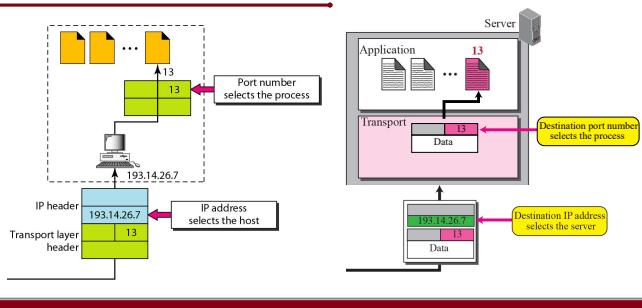
## Port Address Example



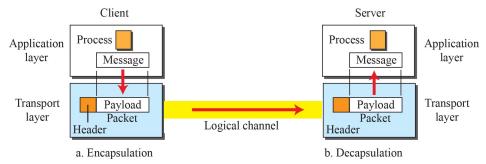
## Addressing



## IP addresses Versus Port Numbers



## Encapsulation and Decapsulation



## Connection Establishment And Termination

## Transport Protocols

- The **transport service** is implemented by a **transport protocol** used between the **two transport entities**.

### Types of Services offered by Transport Layers

- ❖ Just Like network layer service, Transport layers also supports two types of services
- ✓ Connection-Oriented Transport Service and
- ✓ Connectionless Transport Service

---



---



---



---



---



---

### Connection-Oriented Transport Service

- ❖ The connection-oriented transport service is similar to the connection-oriented network service in many ways.
- ✓ Connections have three phases: establishment, data transfer, and release.
- ✓ Addressing and flow control are also similar in both layers.

---



---



---



---



---



---

### Connection Establishment

- ❖ Establishing a connection from one transport entity to just send a CONNECTION REQUEST TPDU to the destination and wait for a CONNECTION ACCEPTED reply.
- ❖ The problem occurs when the Network can Lose, Store, and Duplicate Packets which causes serious complications.

---



---



---



---



---

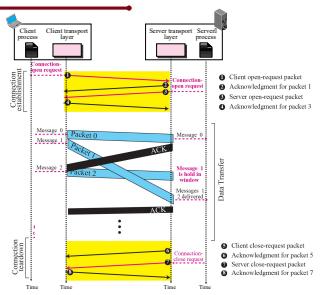


---

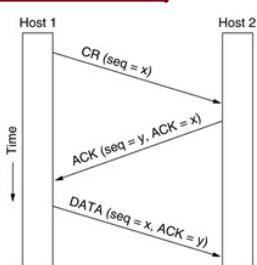
## Connection Establishment

- Connection Establishment can be achieved using three-way handshake.
  - Three protocol scenarios are used for establishing a connection using a CR denotes CONNECTION REQUEST.
- Normal Operation
  - Old CONNECTION REQUEST appearing out of nowhere.
  - Duplicate CONNECTION REQUEST and duplicate ACK.

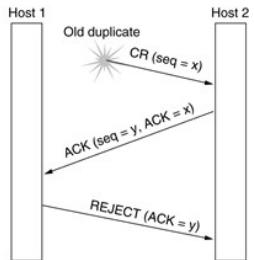
## Connection Establishment



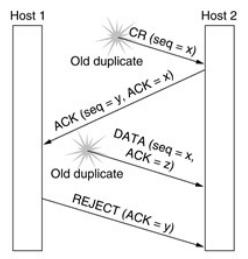
## Normal Operation



### Old Connection Request



### Duplicate Connection Request and Duplicate ACK



### Connection Release

## Connection Release

- ❖ Releasing a connection after establishing the connection.
- ❖ There are two styles of terminating a connection:
  1. Asymmetric Release
  2. Symmetric Release.

---



---



---



---



---



---

## Asymmetric Release

- ❖ Asymmetric release is the way the telephone system works
- ✓ when one party hangs up, the **connection is broken**.
- ❖ Asymmetric release is abrupt and may result in data loss.

---



---



---



---

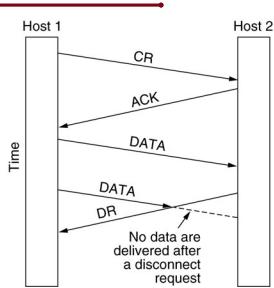


---



---

## Asymmetric Release




---



---



---



---



---



---

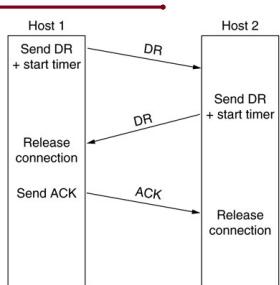
## Symmetric Release

- ❖ Symmetric Release is mainly used to **avoid data loss**.
- ❖ Symmetric release is **independently released** at each direction by sending DISCONNECT TPDU.

## Connection Release: Three-Way Handshake

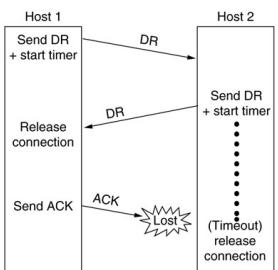
- ❖ Four protocol scenarios for releasing a connection using **three-way handshake**.
  - 1. Normal Case.**
  - 2. Final ACK lost.**
  - 3. Response Lost.**
  - 4. Response lost and subsequent DRs lost.**

## Connection Release: Normal Case.

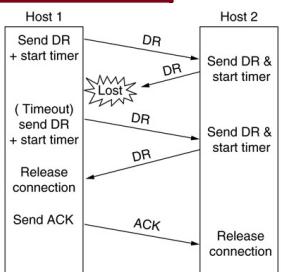


### Response lost and subsequent DRs lost.

- If the final ACK TPDU is lost, which is saved by the timer.
- When the timer expires, the connection is released anyway

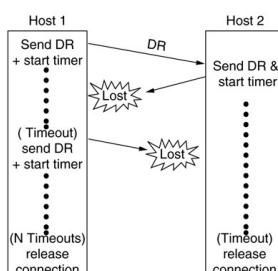


### Connection Release: Response Lost.



### Response lost and subsequent DRs lost.

- When the timer expires, the connection is released anyway



### Connection-Less Transport Service

- The connectionless transport service is also very similar to the connectionless network service.

---



---



---



---

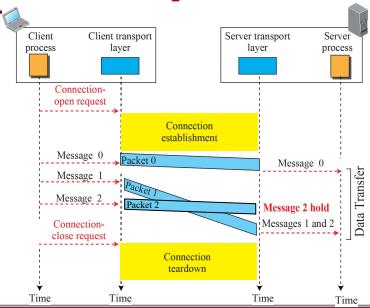


---



---

### Connection-Less Transport Service




---



---



---



---



---



---

### Goals:

- Why do we need a Transport Layer
- Transport Layer: Introduction
  - Services Provided to the Upper Layers
  - Addressing: Port Numbers
  - Connection Establishment And Termination
  - Flow Control and Buffering**
  - Multiplexing and De-Multiplexing**
  - Segmentation and Reassembly**

---



---



---



---



---



---

## Flow Control and Buffering

---

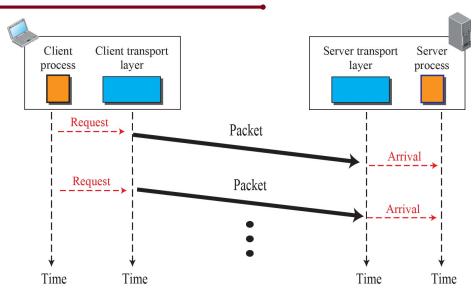
### Flow Control

---

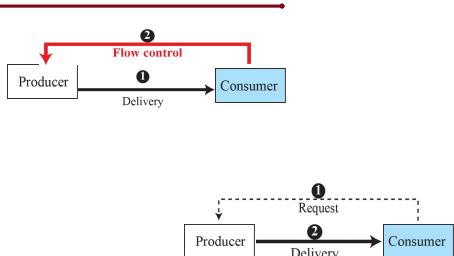
- ❖ The **flow control problem** in the **transport layer** is the same as in the **data link layer**,
  - ❖ The **basic similarity**
    - ✓ Sliding window or other scheme is needed on each connection to keep a fast transmitter from overrunning a slow receiver.
  - ❖ The main difference is that a **router** usually has relatively few lines, whereas a **host** may have numerous connections.
- 

### Flow Diagram

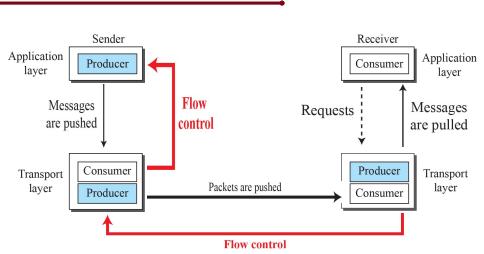
---



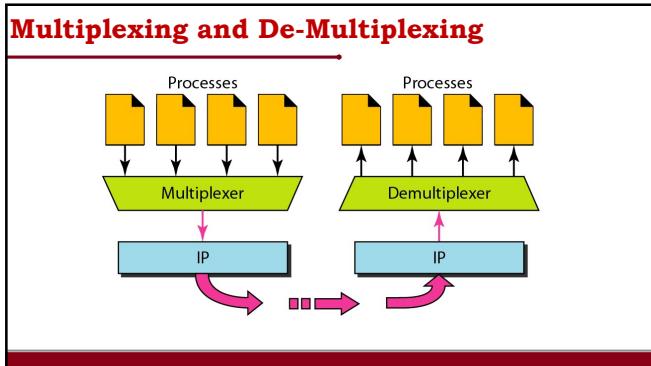
## Flow Control



## Flow Control



## Multiplexing and Demultiplexing




---

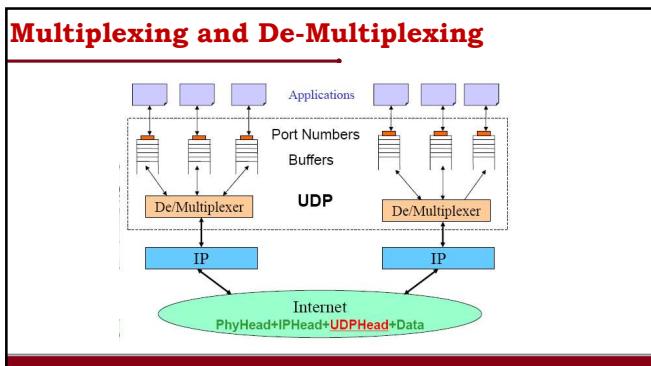
---

---

---

---

---




---

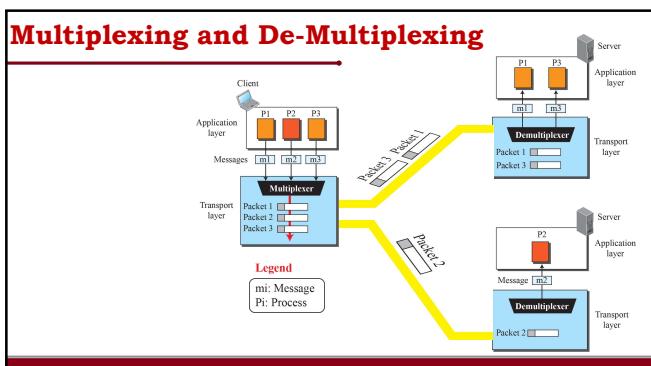
---

---

---

---

---




---

---

---

---

---

---




---

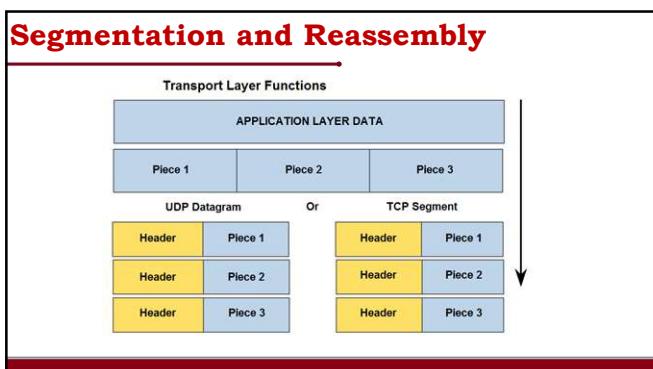
---

---

---

---

---




---

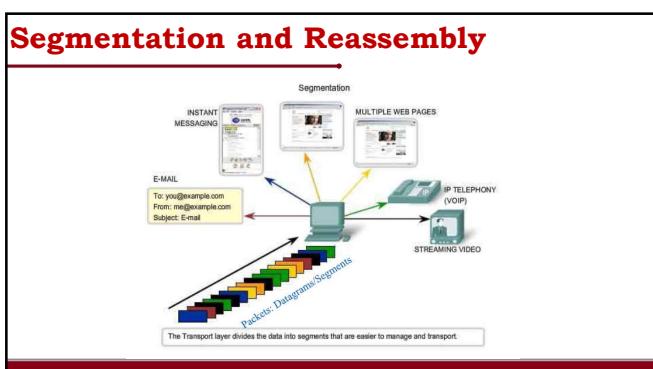
---

---

---

---

---




---

---

---

---

---

---

**Goals:**

- ❖ Why do we need a Transport Layer
- ❖ Transport Layer: Introduction
  - ✓ Services Provided to the Upper Layers
  - ✓ Addressing: Port Numbers
  - ✓ Connection Establishment And Termination
  - ✓ Flow Control and Buffering
  - ✓ Multiplexing and De-Multiplexing
  - ✓ Segmentation and Reassembly

---

---

---

---

---

---

---

**Thank You**

---

---

---

---

---

---

---

## CCN: User-Datagram Protocol

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.



### Goals:

- ❖ The Internet Transport Protocols
- ❖ User-Datagram Protocol
  - ✓ User Datagram Format
  - ✓ UDP Checksum
  - ✓ Encapsulation and Decapsulation
  - ✓ Segmentation and Reassembly

### The Internet Transport Protocols

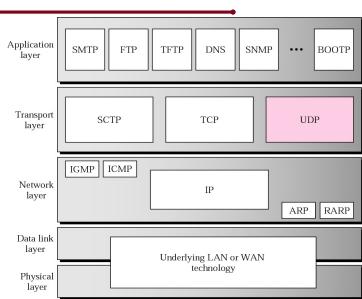
## The Internet Transport Protocols

- We can create a **transport-layer protocol** by combining a **set of services** described in the previous sections.
- The **Internet transport layer** has two main protocols

- 1. User-Datagram Protocol (Connectionless protocol)**
- 2. Transmission Control Protocol (Connection-oriented Protocol)**

## User-Datagram Protocol

### User Datagram Protocol (UDP)



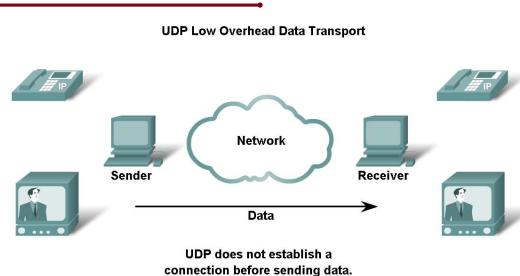
## User Datagram Protocol (UDP)

- ❖ The User Datagram Protocol (UDP) is called a **connectionless, unreliable transport protocol.**
- ❖ It provide **process-to-process communication** services to the IP layer.
- ❖ UDP is available to **network programmers** who wish to send datagrams

## Connectionless Service

- ❖ In Connectionless Service, a **host just sends packets**
  - ✓ **No connection** needs to be established
  - ✓ **No virtual circuit** needs to be created
- ❖ A **connectionless socket** can send **messages to multiple recipients**

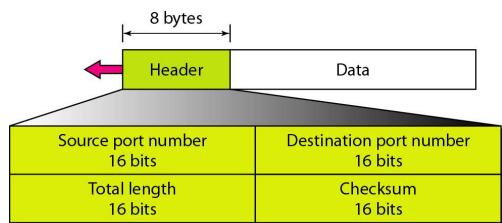
## Connectionless Service



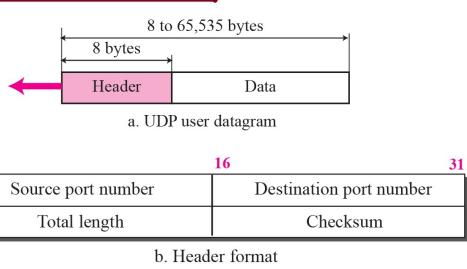
## User Datagram Protocol (UDP)

- ❖ UDP datagrams are called user datagrams,
- ✓ Allow user-level creation & transmission of datagrams
- ❖ UDP has a small, fixed-size (8 byte) header

## User Datagram Format



## User Datagram Format



## User Datagram

Bytes	Name	Description
2	Source Port	The port used to send the message
2	Dest. Port	The port to be used to receive the message
2	Length	The length of the data in the message
2	Checksum	The checksum of the message data
?	Data	The data of the message

8 bytes

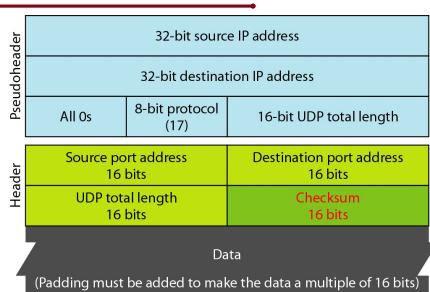
## UDP Length

$$\text{UDP length} = \text{IP length} - \text{IP header's length}$$

## UDP Checksum

- ❖ UDP checksum calculation is different from the one from IP
- ❖ Checksum includes three sections: a **pseudoheader, the UDP header, and the data coming** from the application layer.

## UDP Checksum



## Checksum calculation of a simple UDP user datagram

153.18.8.105	171.2.14.10	All 0s	17	15	
1087	13				
15	All 0s				
T	E	S	T		
I	N	G	All 0s		

Binary values and their decimal equivalents for the header fields:

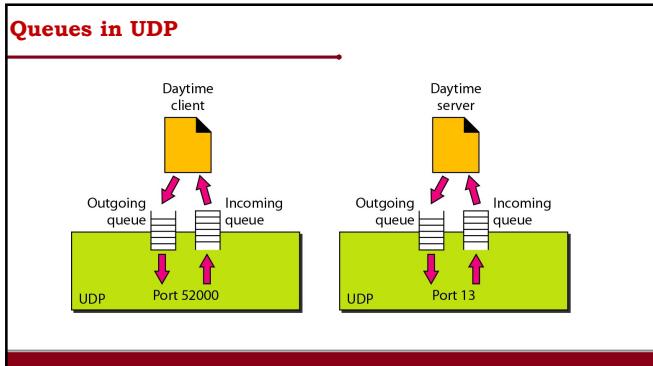
- 10011001 00010010 → 153.18
- 00001000 01101001 → 8.105
- 10101011 00000010 → 171.2
- 00001110 00001010 → 14.10
- 00000000 00010001 → 0 and 17
- 00000000 00001111 → 15
- 00000100 00111111 → 1087
- 00000000 00001101 → 13
- 00000000 00001111 → 15
- 00000000 00000000 → 0 (checksum)
- 01010100 01000101 → T and E
- 01010011 01010100 → S and T
- 01001001 01001110 → I and N
- 01000111 00000000 → G and 0 (padding)

Final steps:

- 10010110 11101011 → Sum
- 01101001 00010100 → Checksum

## Well-known ports used with UDP

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpt	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)



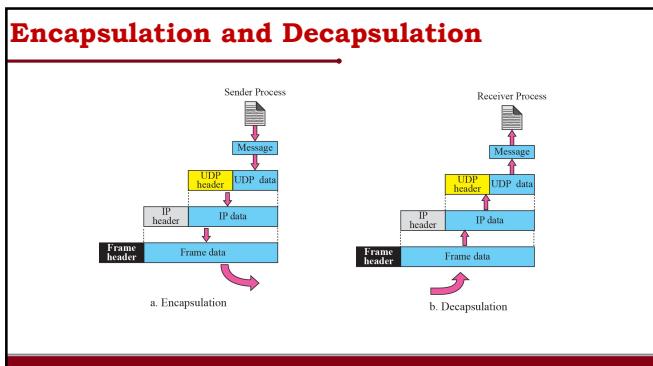

---

---

---

---

---



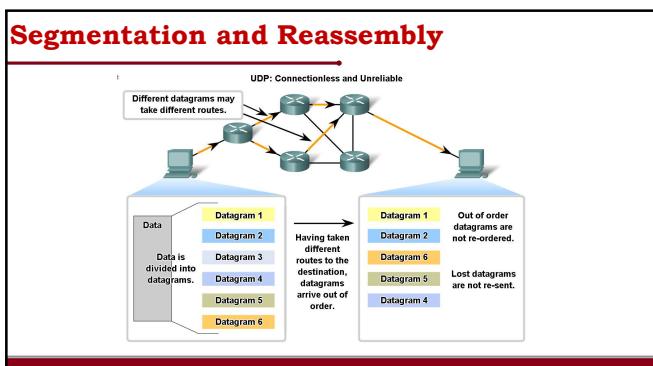

---

---

---

---

---




---

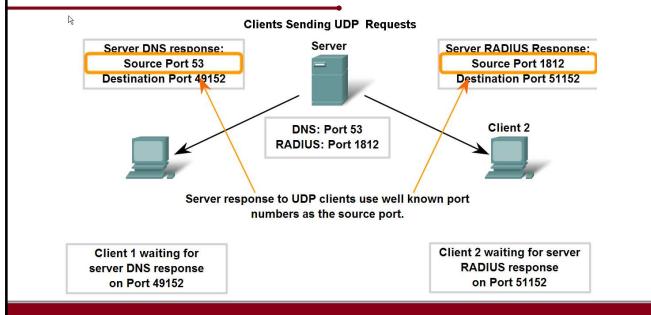
---

---

---

---

**UDP Protocol utilizes the port numbers in the client-server communication**



**Goals:**

- ❖ The Internet Transport Protocols
- ❖ User-Datagram Protocol
  - ✓ User Datagram Format
  - ✓ UDP Checksum
  - ✓ Encapsulation and Decapsulation
  - ✓ Segmentation and Reassembly

**Thank You**