

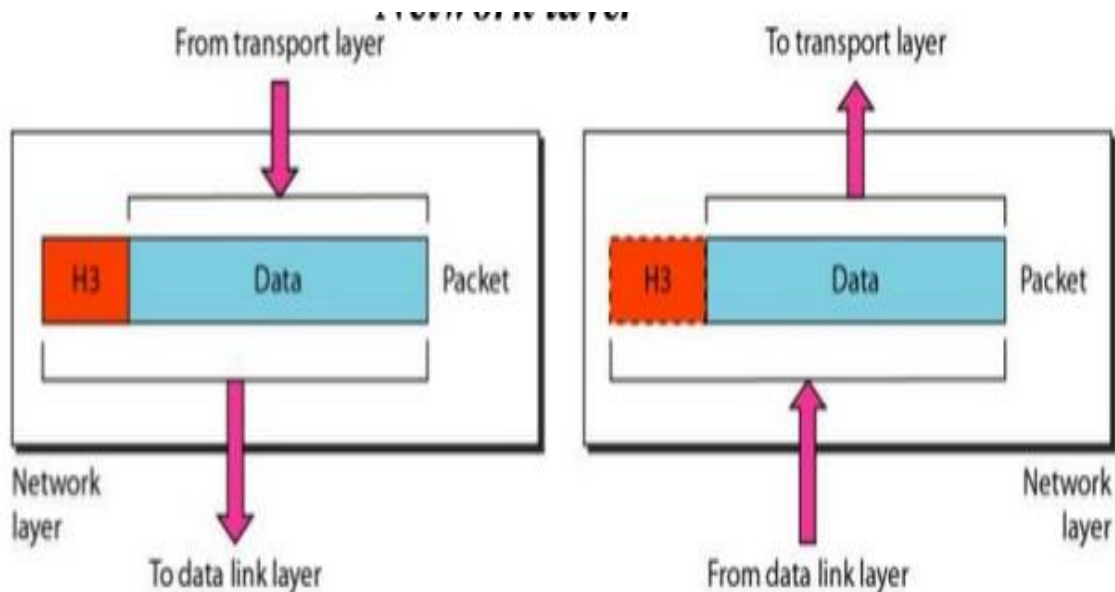
# Unit 3

## COMPUTER NETWORKS RCS-601

# Unit 3

- **Network Layer:** Network Layer - Point - to Pont Networks, routing, Congestion control  
Internetworking -TCP / IP, IP packet, IP address, IPv6.

# Network Layer -Revision



- The network layer controls the operation of the **subnet**.
- The network layer is responsible for the delivery of individual **packets** from the source host to the destination host.
- The network layer controls the operation of the subnet. A key design issue is determining how **packets are routed from source to destination**.

The network layer is responsible for the delivery of individual packets from the source host to the destination host.

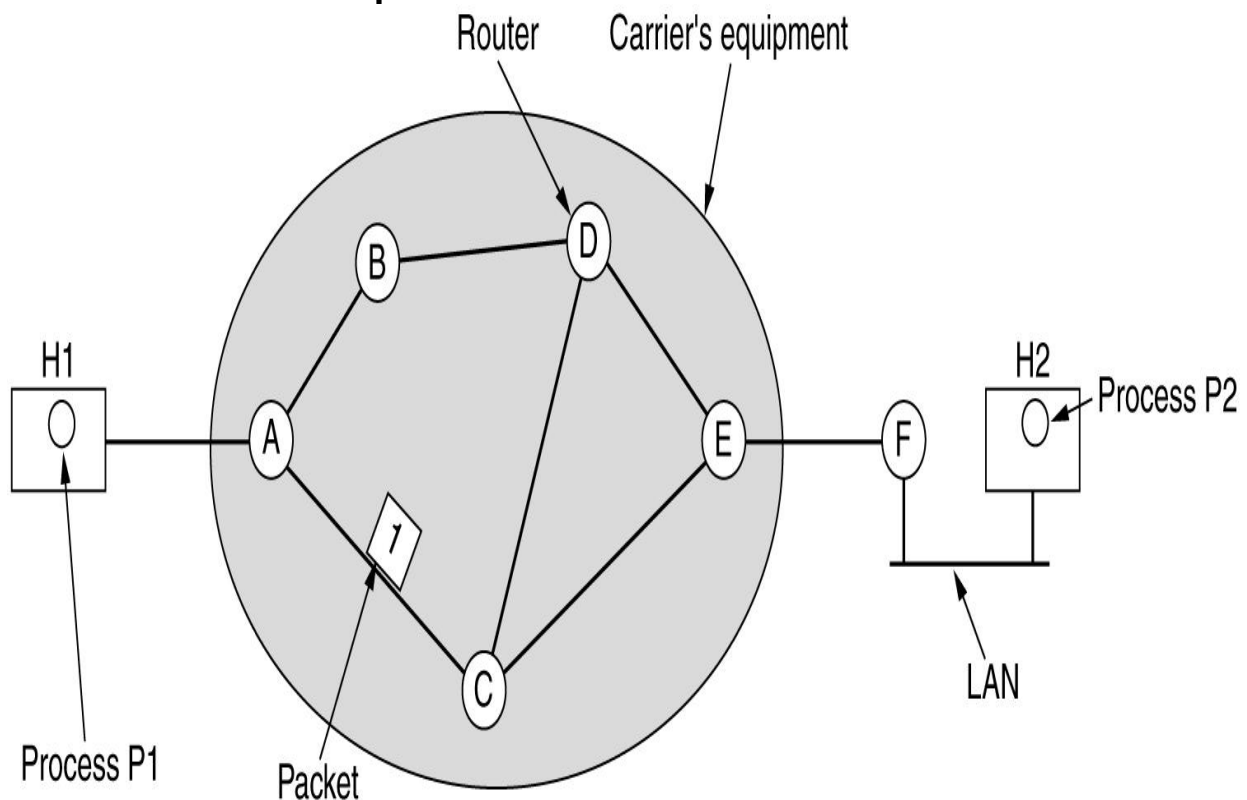
- Routes can be based on **static tables** that are "wired into" the network and rarely changed. They can also be determined at the **start of each conversation**.
- If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. The **control of such congestion** also belongs to the network layer.
- When a packet has to travel from one network to another to get to its destination, many problems can arise. The **addressing** used by the second network **may be different** from the first one. The second one may not accept the packet at all because it is **too large**. The **protocols may differ**, and so on. It is up to the network layer to overcome all these problems

# Design issues

- Goal of layer: **get packets from source host to destination host**
  - **Routing**: should know about topology of subnet
  - **Congestion**: should avoid overloading some communication lines and routers
  - **Quality of service**: offer the appropriate service
  - **Internetworking**: deal with network differences, if source and destination are connected to different networks

# Design issues

- Store-and-forward packet switching
  - Equipment of: carrier <> customer
  - Algorithm at router
    - Receive packet



# Design issues: services

- Interface
  - Important: = interface between carrier and customer
  - Designed with following goals in mind:
    - Services should be independent of the subnet technology
    - Transport layer should be shielded from the number, type, topology of the subnets
    - Network addresses should use a uniform numbering plan, even across LANs and WANs
- Connections?
  - Connection-oriented <> Connectionless!

# Design issues: services

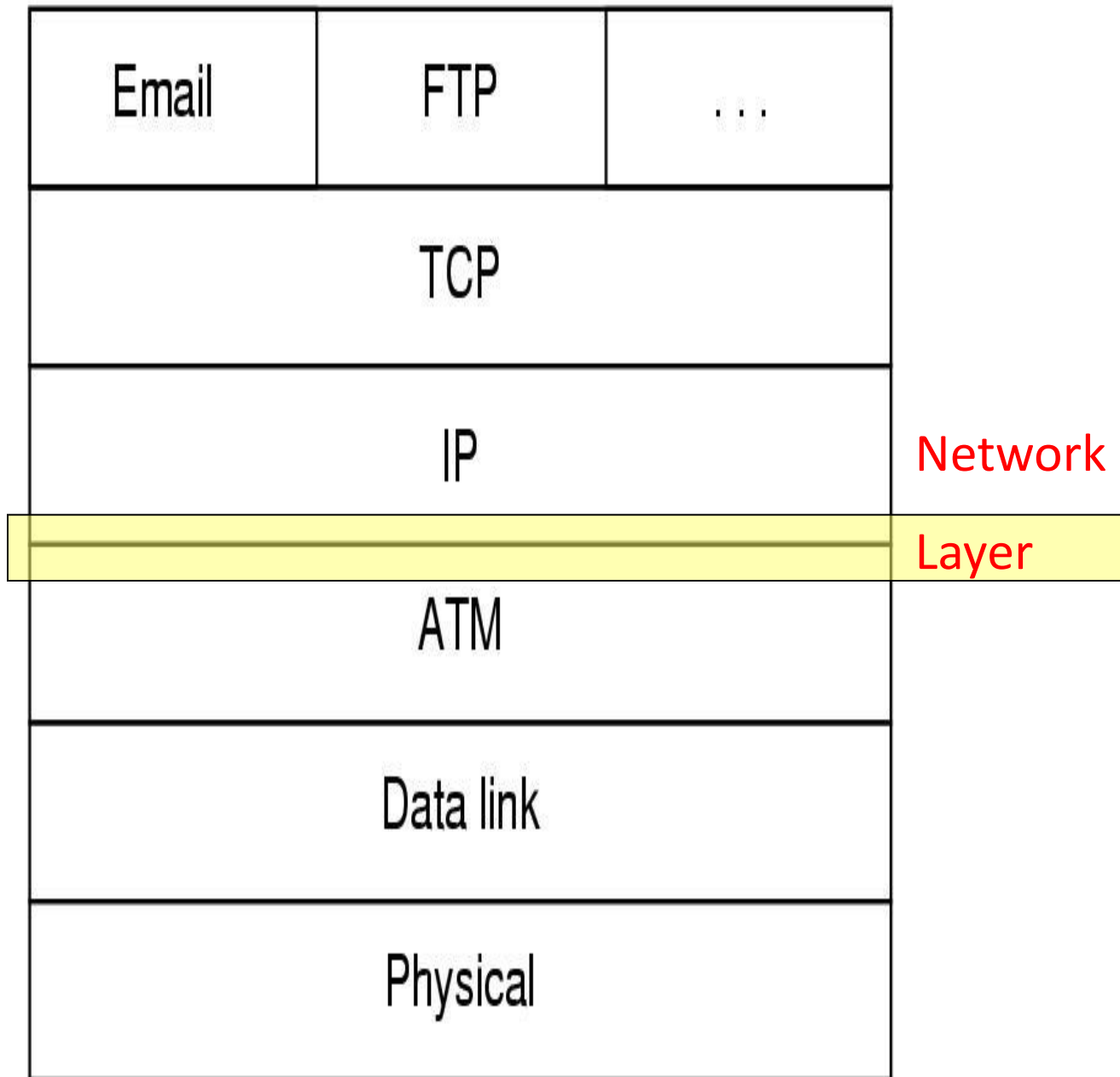
- Connection-oriented <> Connectionless
  - PTTs: connection-oriented
    - 100 years of experience with the world-wide telephone system
    - Connection time → billing!
  - Internet
    - Subnets are inherently unreliable
  - Real issue: **where** to put the **complexity** as some/many applications require reliable transfer (~ connection-oriented service)
    - Network layer
    - Transport layer



# Design issues: services

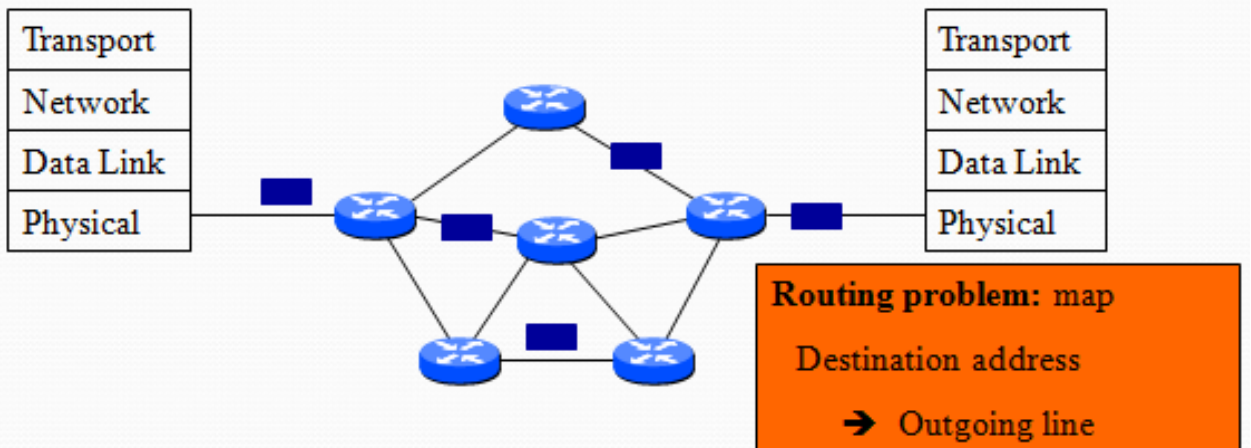
- Connection-oriented <> Connectionless (*cont.*)
  - In favour of connectionless service
    - Computing power is cheap: hosts can handle the complexity
    - Subnet is a large, long lasting investment: keep it simple
    - For some applications speedy delivery (low, constant delay) is important
  - In favour of connection-oriented service
    - Users want a reliable trouble-free service
    - Some services are easier to provide on top of connection-oriented service
  - Examples
    - ATM: connection oriented
    - IP: connection-less
    - IP on top of ATM

# Design issues: services



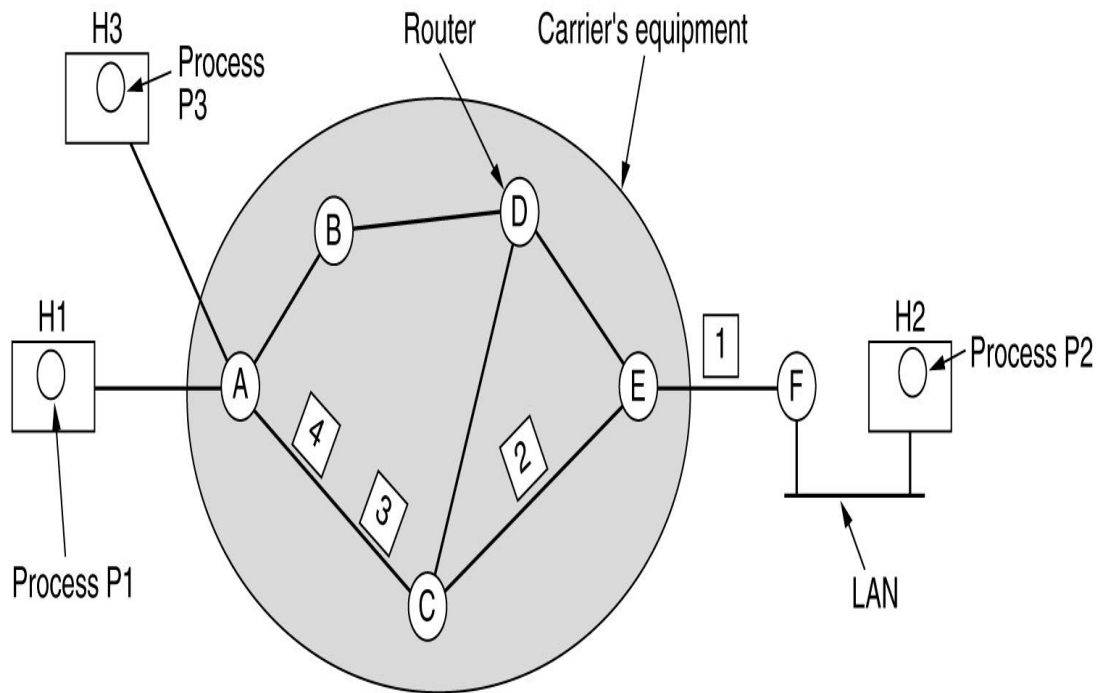
# Design: internal organisation

- Virtual circuits
  - Routes chosen at connection time
  - Connection identified by a virtual circuit number (VCn)
  - Primary service of subnet is connection-oriented



# Design: internal organisation

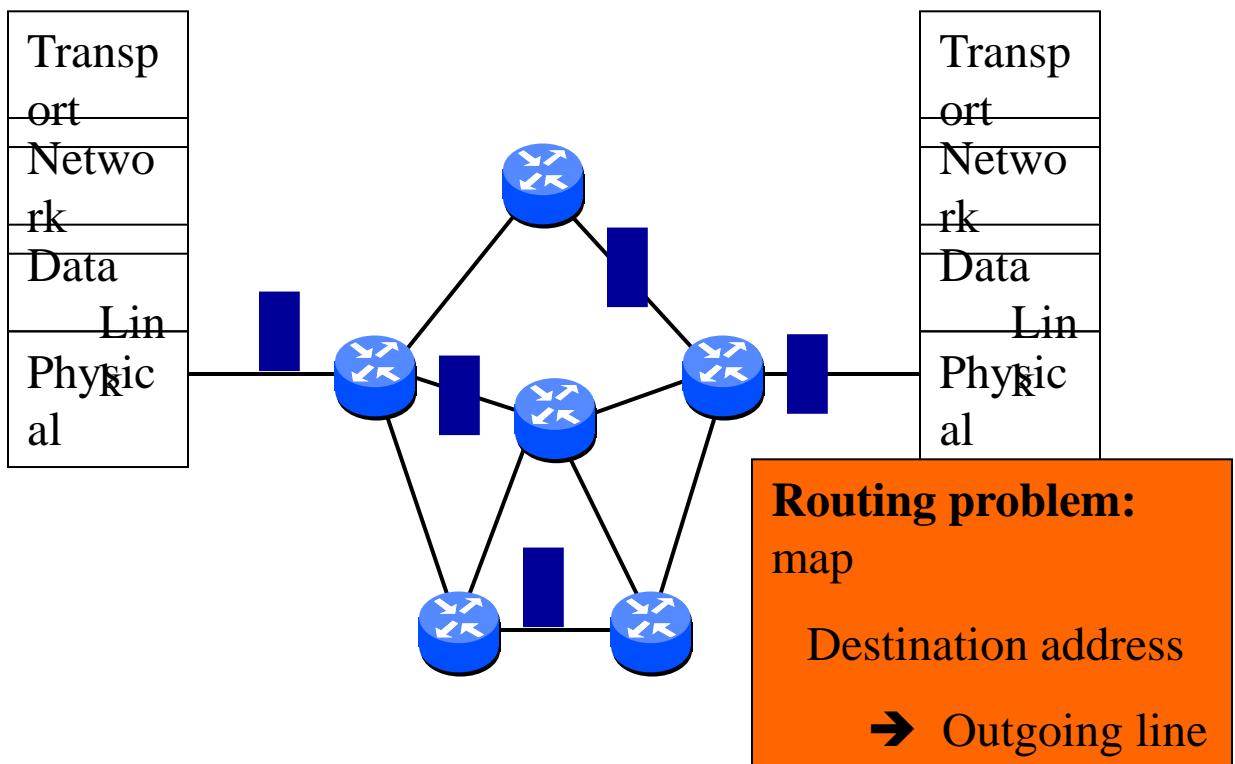
- Virtual circuits



A's table				C's table				E's table			
H1	1	C	1	A	1	E	1	C	1	F	1
H3	1	C	2	A	2	E	2	C	2	F	2
In											

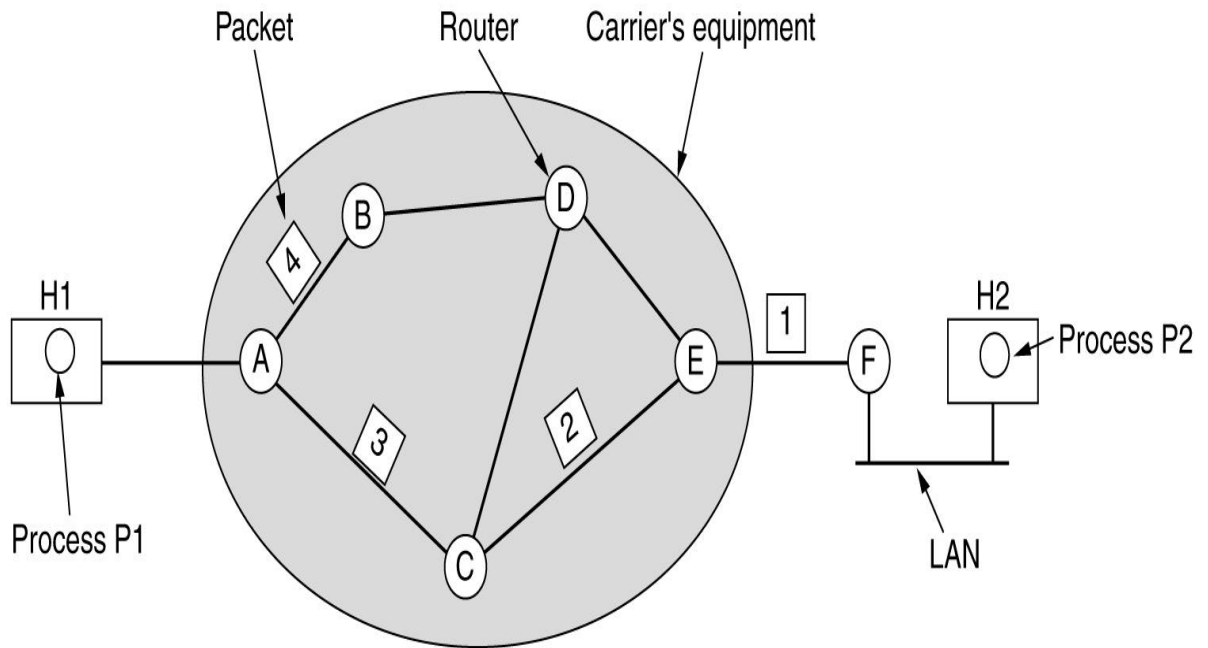
# Design: internal organisation

- Datagram subnet
  - Each packet is routed independently
  - Subnet has more work to do
  - More robust, easier to adapt to failures and congestion



# Design: internal organisation

- Datagram subnet



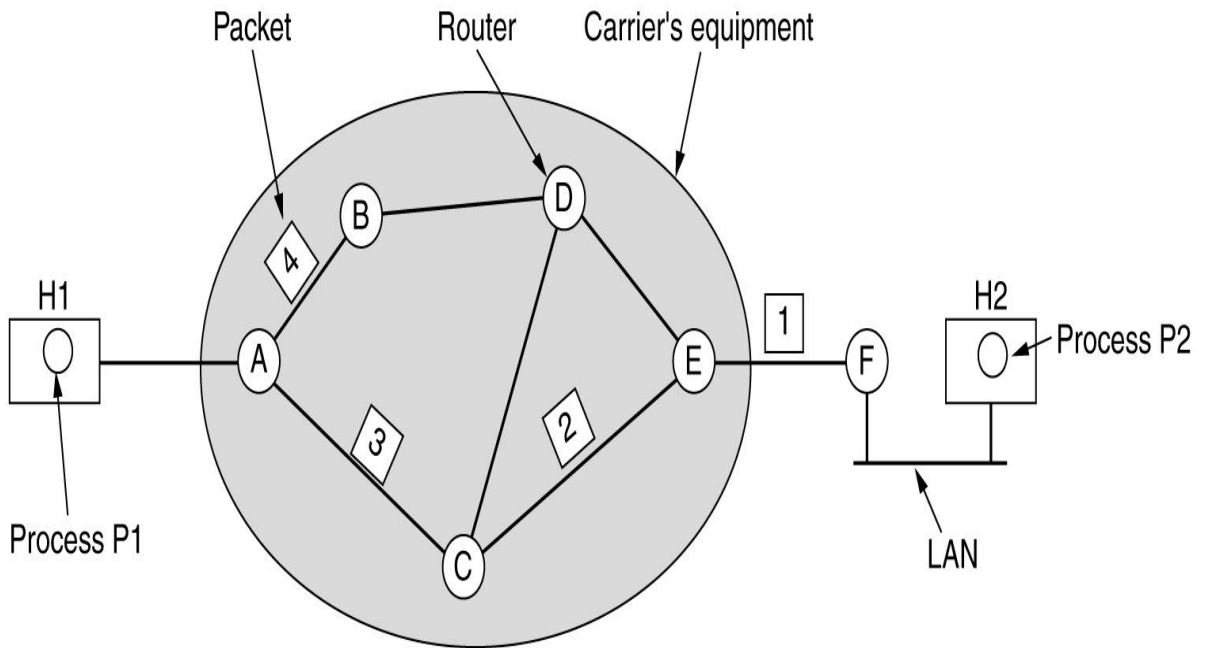
A's table

initially	later	C's table	E's table
A   -	A   -	A   A	A   C
B   B	B   B	B   A	B   D
C   C	;   C	C   -	C   C
D   B	D   B	D   D	D   D
E   C	E   B	E   E	E   -
F   C	F   B	F   E	F   F

Dest. Line

# Design: internal organisation

- Datagram subnet



A's table

initially	later
A   -	A   -
B   B	B   B
C   C	C   C
D   B	D   B
E   C	E   B
F   C	F   B

Dest. Line

C's table

A   A
B   A
C   -
D   D
E   E
F   E

E's table

A   C
B   D
C   C
D   D
E   -
F   F

**Change of routing table**

# Design: internal organisation

Issue	Datagram subnet	VC subnet
Circuit setup	Not needed	required
Addressing	Full addresses (source + destination) in each packet	Short VC number in each packet
State information	No state held in subnet	State held for each VC
Routing	Done for each packet independently	Route chosen at connection setup
Effect of router failures	None, except for packet losses	All VCs passing failed router are terminated
Congestion control	difficult	Easy if enough buffers can be allocated in advance



# Design: internal organisation

	Type of	subnet
Service to upper layer	Datagram	Virtual Circuit
Connectionless	UDP IP	UDP IP ATM
Connection-oriented	TCP IP	ATM AAL1 ATM

# Network Layer

- Design issues
- Routing
- Congestion
- Internetworking
- Internet Protocols
- Multimedia or Qos

# Routing

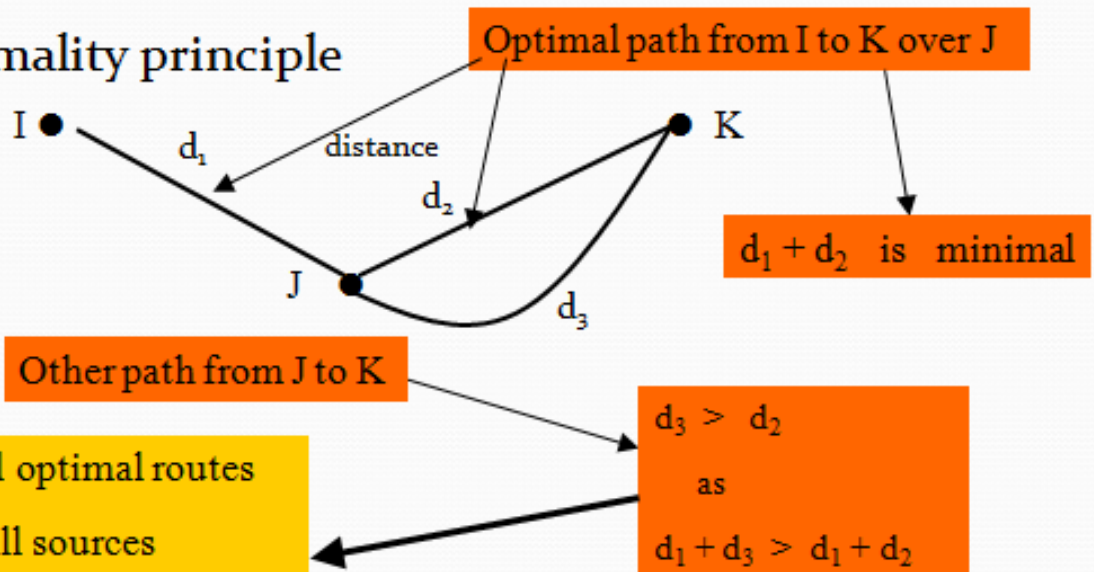
*Routing algorithm*:: that part of the Network Layer responsible for deciding on which output line to transmit an incoming packet.

- Remember: For virtual circuit subnets the routing decision is made ONLY at set up.

**Algorithm properties**::  
correctness, simplicity,  
robustness, stability, fairness,  
optimality, and scalability.

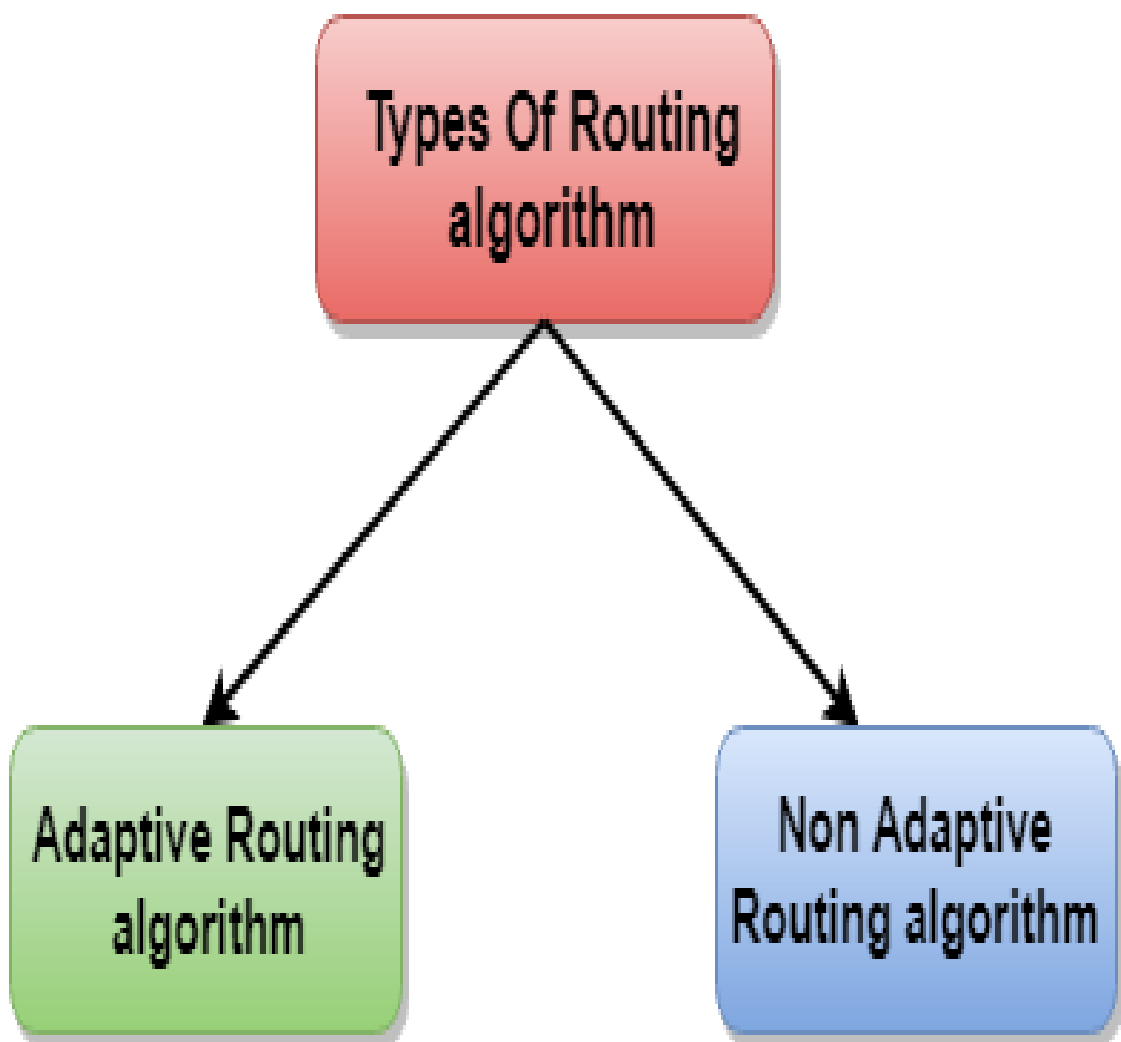
# Routing algorithms

- Optimality principle



The Routing algorithm is divided into two categories:

- Adaptive Routing algorithm
- Non-adaptive Routing algorithm



# Routing Classification

## Adaptive Routing

based on current measurements of traffic and/or topology.

1. centralized
2. isolated
3. distributed

## Non-Adaptive Routing

1. flooding
2. static routing using shortest path algorithms

# Adaptive Routing algorithm

An adaptive routing algorithm is also known as **dynamic routing algorithm**.

This algorithm makes the routing decisions based on the topology and network traffic.

The main parameters related to this algorithm are hop count, distance and estimated transit time.

# An adaptive routing algorithm can be classified into three parts:

**Centralized algorithm:** It is also known as global routing algorithm as it computes the least-cost path between source and destination by using complete and global knowledge about the network. **Link state algorithm** is referred to as a centralized algorithm since it is aware of the cost of each link in the network.

**Isolation algorithm:** It is an algorithm that obtains the routing information by using local information rather than gathering information from other nodes.

**Distributed algorithm:** It is also known as decentralized algorithm as it computes the least-cost path between source and destination in an iterative and distributed manner. In the decentralized algorithm, no node has the knowledge about the cost of all the network links



# Non-Adaptive Routing algorithm

Non Adaptive routing algorithm is also known as a **static routing algorithm**.

When booting up the network, the routing information stores to the routers.

Non Adaptive routing algorithms do not take the routing decision based on the network topology or network traffic.

# The Non-Adaptive Routing algorithm is of two types:

- **Flooding:** In case of flooding, every incoming packet is sent to all the outgoing links except the one from it has been reached. The disadvantage of flooding is that node may contain several copies of a particular packet.
- **Random walks:** In case of random walks, a packet sent by the node to one of its neighbors randomly. An advantage of using random walks is that it uses the alternative routes very efficiently.

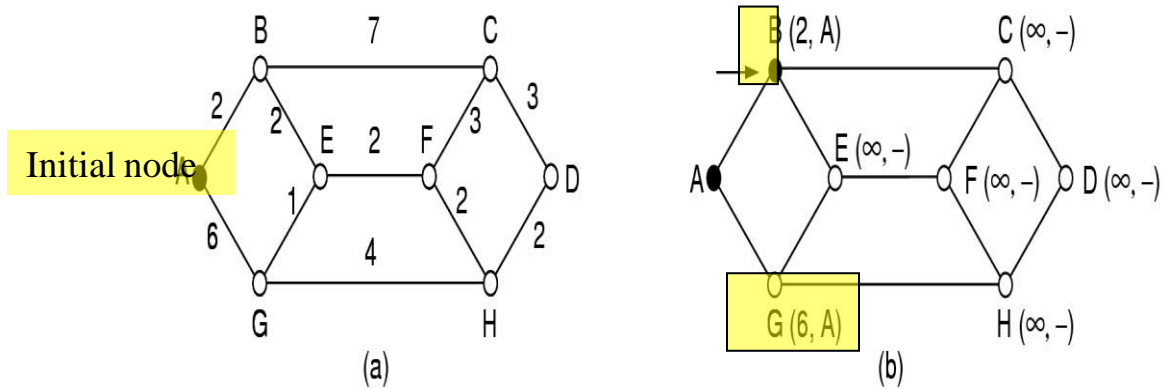
## Differences b/w Adaptive and Non-Adaptive Routing Algorithm

Basis Of Comparison	Adaptive Routing algorithm	Non-Adaptive Routing algorithm
Define	Adaptive Routing algorithm is an algorithm that constructs the routing table based on the network conditions.	The Non-Adaptive Routing algorithm is an algorithm that constructs the static table to determine which node to send the packet.
Usage	Adaptive routing algorithm is used by dynamic routing.	The Non-Adaptive Routing algorithm is used by static routing.
Routing decision	Routing decisions are made based on topology and network traffic.	Routing decisions are the static tables.
Categorization	The types of adaptive routing algorithm, are Centralized, isolation and distributed algorithm.	The types of Non Adaptive routing algorithm are flooding and random walks.
Complexity	Adaptive Routing algorithms are more complex.	Non-Adaptive Routing algorithms are simple.

# Routing: shortest path

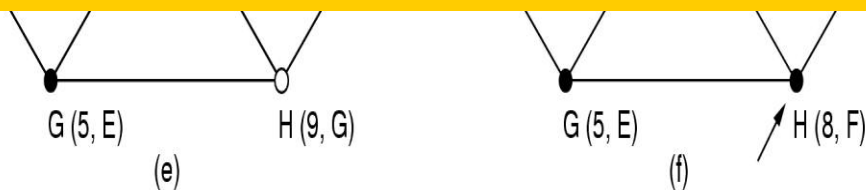
- Algorithm of Dijkstra: shortest path in graph
  - Graph
    - Node = router
    - Arc = communication line
  - Metric
    - Number of hops
    - Geographic distance
    - Mean queueing and transmission delay

# Routing: shortest path

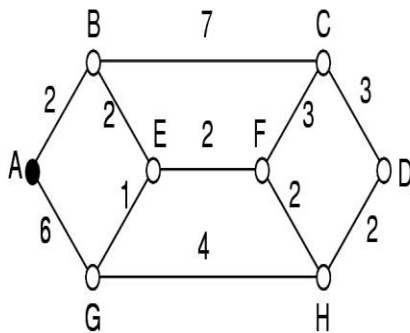


## Elements of algorithm:

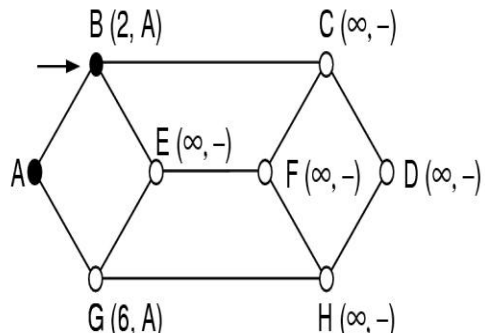
- Mark all nodes as free: ○
- Mark initial node as selected: ●
- repeat till destination is selected:
  - Label all free nodes reachable from selected nodes with shortest distance to a selected node
  - Select free node with shortest distance to a selected node and mark it as selected



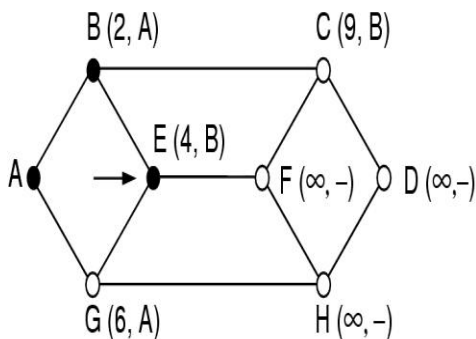
# Routing: shortest path



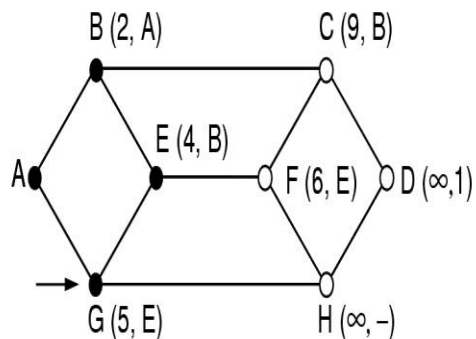
(a)



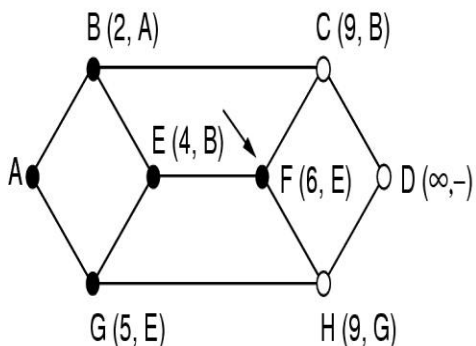
(b)



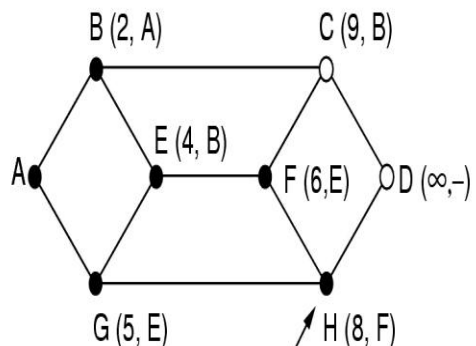
(c)



(d)



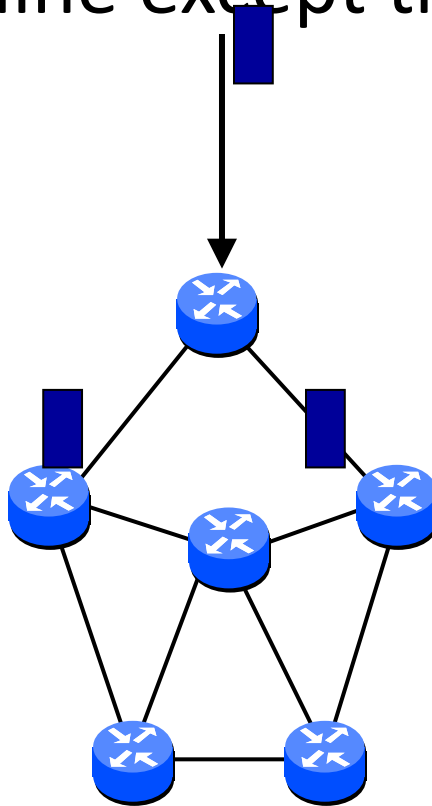
(e)



(f)

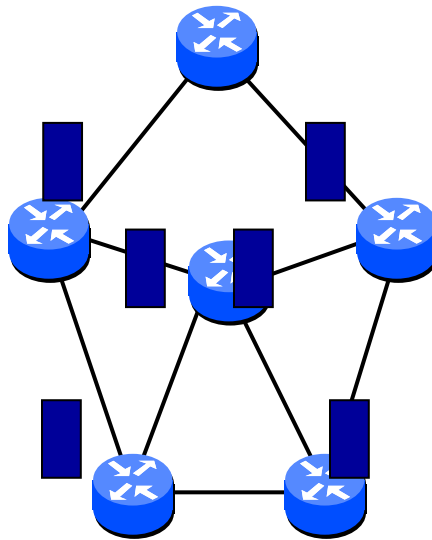
# Routing: flooding

- Every packet is sent out on every outgoing line except the one it arrived at



# Routing: flooding

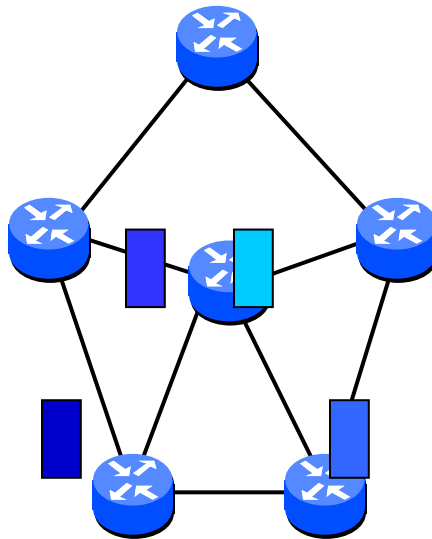
- Every packet is sent out on every outgoing line except the one it arrived at





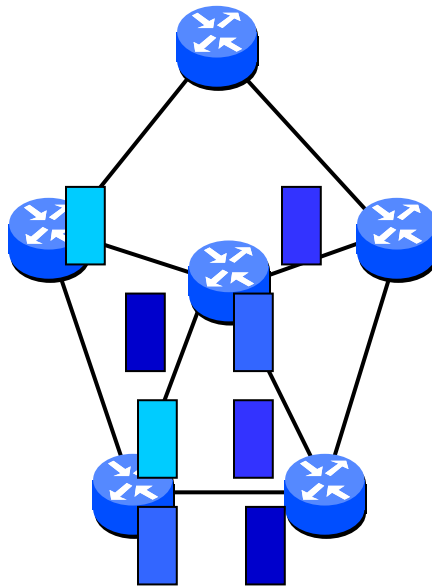
# Routing: flooding

- Every packet is sent out on every outgoing line except the one it arrived at



# Routing: flooding

- Every packet is sent out on every outgoing line except the one it arrived at



# Routing: flooding

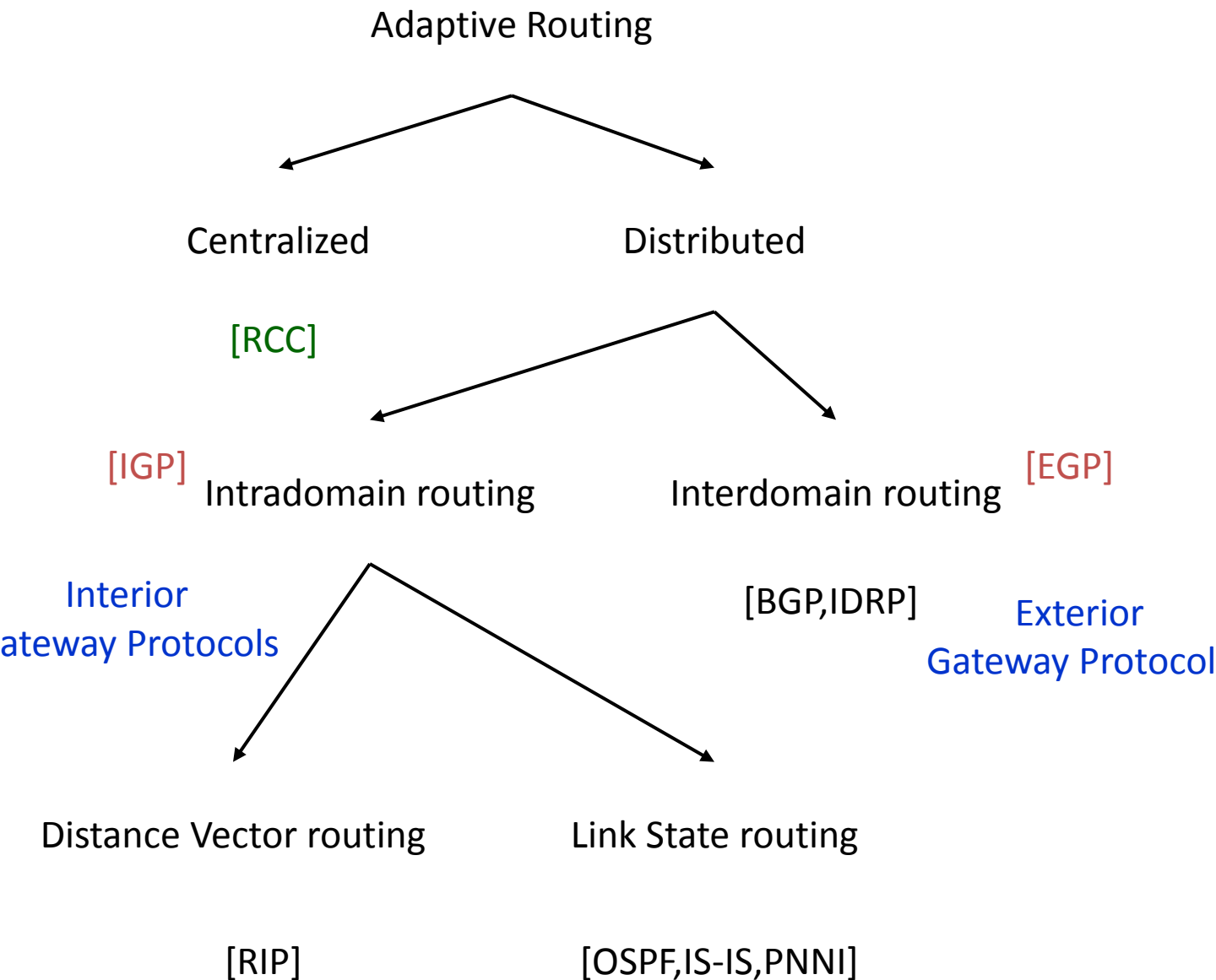
- Every packet is sent out on every outgoing line except the one it arrived at
- Duplicates!! How to limit?
  - Hop counter
    - Decrement in each router
    - Discard packet if counter is 0
    - Initialisation?
  - Sequence number in packet
    - Avoid sending the same packet a second time
    - Keep in each router per source a list of packets already seen
- *Useful?*

# Routing: flooding

- Every packet is sent out on every outgoing line except the one it arrived at
- Sometimes useful
  - Robust algorithm: e.g. military applications
  - Broadcast
  - Comparison purposes: always shortest path
- Selective flooding
  - Use only those lines that are going approximately in right direction
  - Still working?

# Internetwork Routing

## [Halsall]



# Distance Vector Routing

- Historically known as the *old* ARPANET routing algorithm {or known as *Bellman-Ford algorithm*}.

Basic idea: each network node maintains a Distance Vector table containing the *distance* between itself and ALL possible destination nodes.

- Distances are based on a chosen metric and are computed using information from the **neighbors'** distance vectors.

Metric: *usually hops or delay*

# Distance Vector Routing

## Information kept by DV router

1. each router has an ID
2. associated with each link connected to a router, there is a link cost (static or dynamic) **the metric issue!**

## Distance Vector Table Initialization

Distance to itself = 0

Distance to ALL other routers = infinity  
number

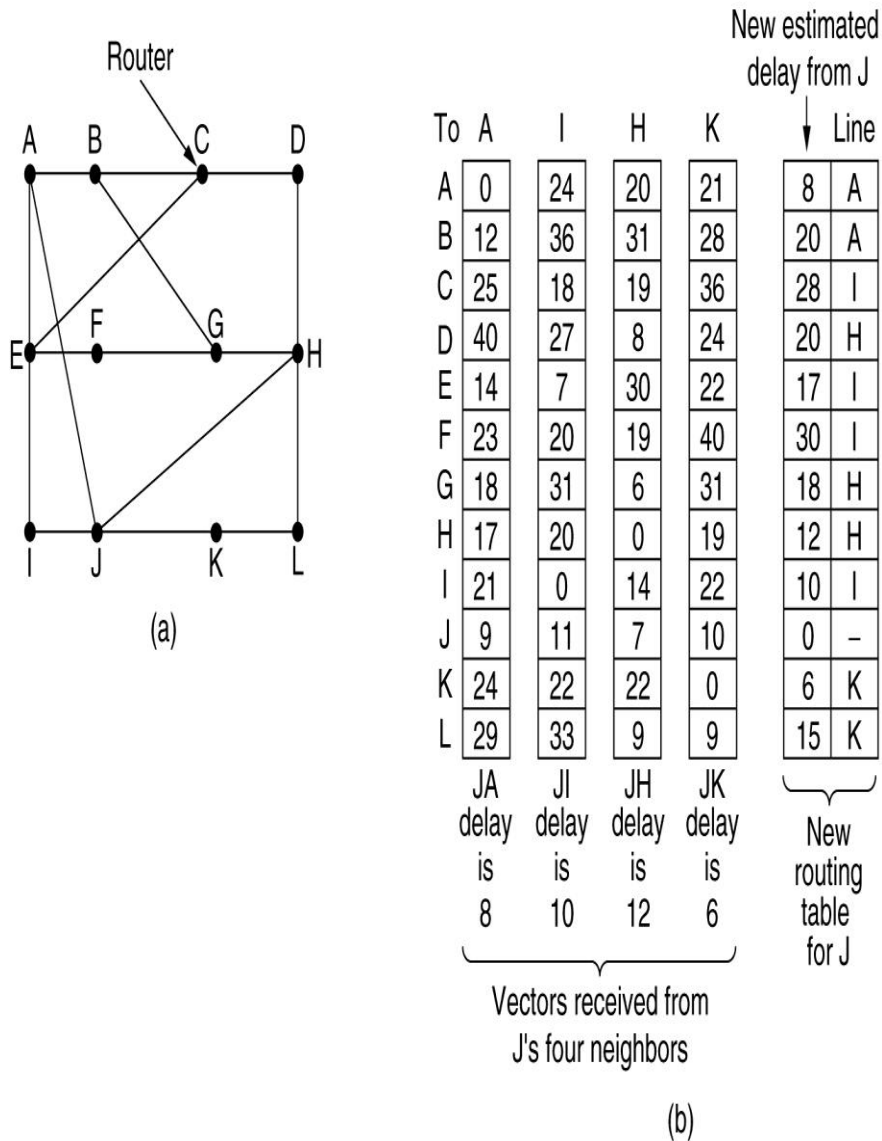
# Distance Vector Algorithm

## [Perlman]

1. Router transmits its *distance vector* to each of its neighbors.
2. Each router receives and saves the most recently received *distance vector* from each of its neighbors.
3. A router **recalculates** its distance vector when:
  - a. It receives a *distance vector* from a neighbor containing different information than before.
  - b. It discovers that a link to a neighbor has gone down (i.e., a topology change).
- The DV calculation is based on minimizing the cost to each destination.



# Distance Vector Routing



**Figure 5-9.** (a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

# Routing Information Protocol (RIP)

- RIP had widespread use because it was distributed with BSD Unix in *“routed”, a router management daemon.*
- **RIP** is the most used Distance Vector protocol.
- RFC1058 in June 1988.
- Sends packets every 30 seconds or faster.
- Runs over UDP.
- Metric = hop count
- BIG problem is max. hop count =16  
➔ RIP limited to running on small networks!!
- Upgraded to RIPv2

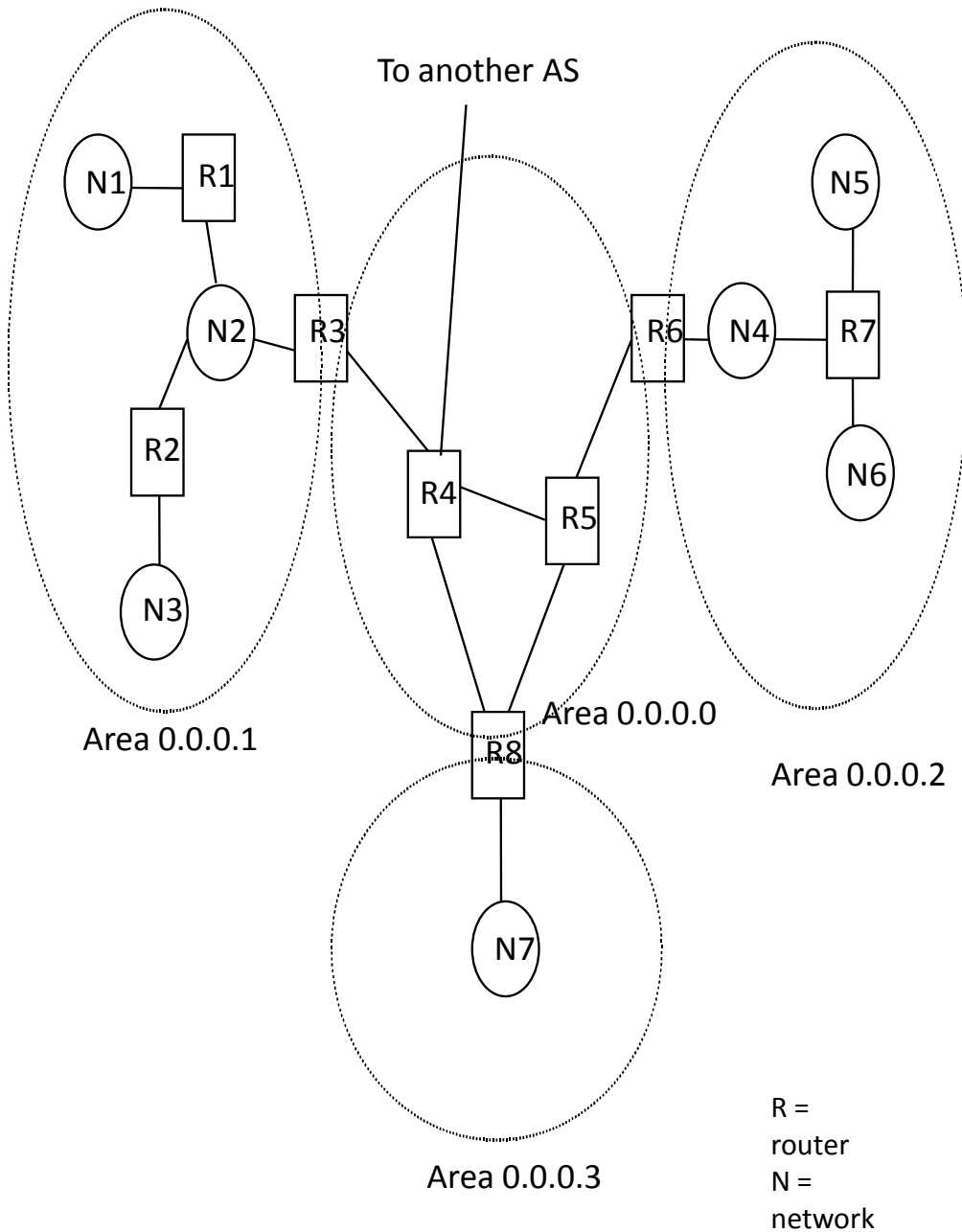
# Link State Algorithm

1. Each router is responsible for meeting its neighbors and learning their names.
2. Each router constructs a **link state packet (LSP)** which consists of a list of names and cost to reach each of its neighbors.
3. The **LSP** is transmitted to ***ALL other routers***. Each router stores the most recently generated **LSP** from each other router.
4. Each router uses complete information on the network topology to compute the ***shortest path route*** to each destination node.

# Open Shortest Path First (OSPF)

- OSPF runs *on top of* IP, i.e., an OSPF packet is transmitted with IP data packet header.
- Uses Level 1 and Level 2 routers
- Has: backbone routers, area border routers, and AS boundary routers
- LSPs referred to as **LSAs (Link State Advertisements)**
- Complex algorithm due to **five** distinct LSA types.

## OSPF Areas



# OSPF

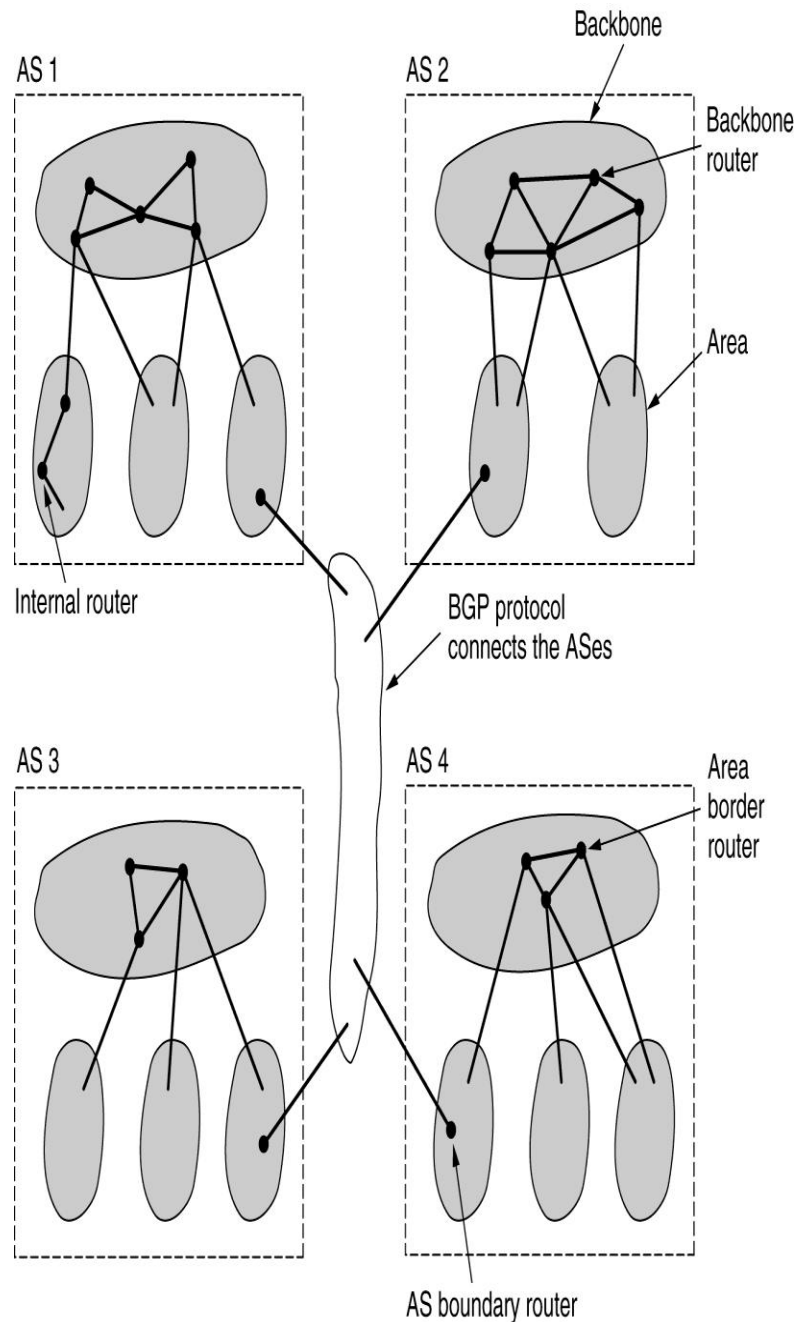


Figure 5-65. The relation between ASes, backbones, and areas in OSPF.

# Border Gateway Protocol (BGP)

- The replacement for EGP is BGP. Current version is BGP-4.
- BGP assumes the Internet is an arbitrary interconnected set of AS's.
- In *interdomain routing* the goal is to find ANY path to the intended destination that is loop-free. The protocols are more concerned with **reachability** than optimality.

# What is **congestion**?

A state occurring in network layer when the message traffic is so heavy that it slows down network response time.

## **Effects of Congestion**

As delay increases, performance decreases.

If delay increases, retransmission occurs, making situation worse.

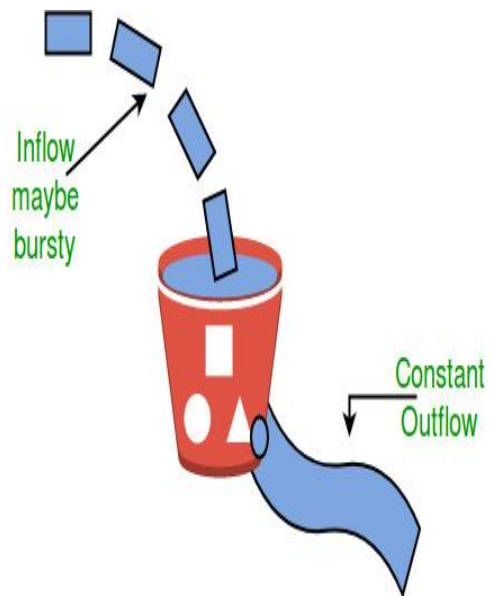


# Congestion control algorithms

## Leaky Bucket Algorithm

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following **steps** are involved in leaky bucket algorithm:

- When host wants to send packet, packet is thrown into the bucket.
- The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
- Bursty traffic is converted to a uniform traffic by the leaky bucket.
- In practice the bucket is a finite queue that outputs at a finite rate.

# Token bucket Algorithm

## **Need** of token bucket Algorithm:-

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

## **Steps** of this algorithm can be described as follows:

- In regular intervals tokens are thrown into the bucket.  $f$
- The bucket has a maximum capacity.  $f$
- If there is a ready packet, a token is removed from the bucket, and the packet is sent.
- If there is no token in the bucket, the packet cannot be sent.

Let's understand with an example,

In figure (A) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In figure (B) We see that three of the five packets have gotten through, but the other two are stuck waiting for more tokens to be generated.

# Ways in which token bucket is superior to leaky bucket:

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in the token bucket algorithm. In the token bucket, algorithm tokens are generated at each tick (up to a certain limit). For an incoming packet to be transmitted, it must capture a token and the transmission takes place at the same rate. Hence some of the busy packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system.

**Formula:**  $M * s = C + \rho * s$

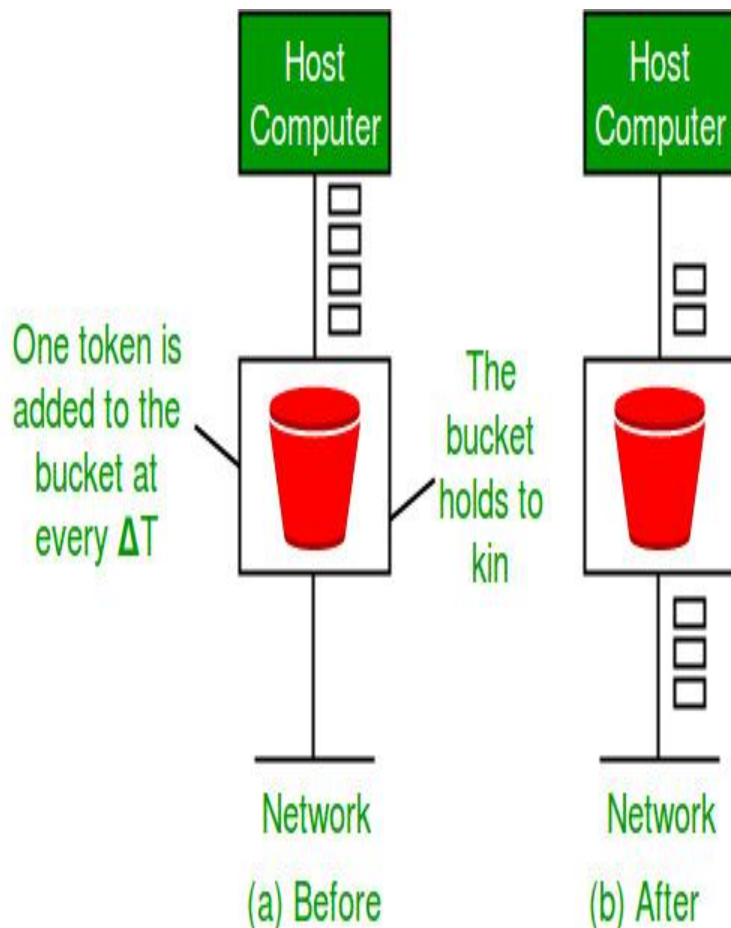
where  $S$  – is time taken

$M$  – Maximum output rate

$\rho$  – Token arrival rate

$C$  – Capacity of the token bucket in byte

# Let's understand with an example,



# The TCP/IP reference model

The TCP/IP reference model was developed prior to OSI model. The major design goals of this model were,

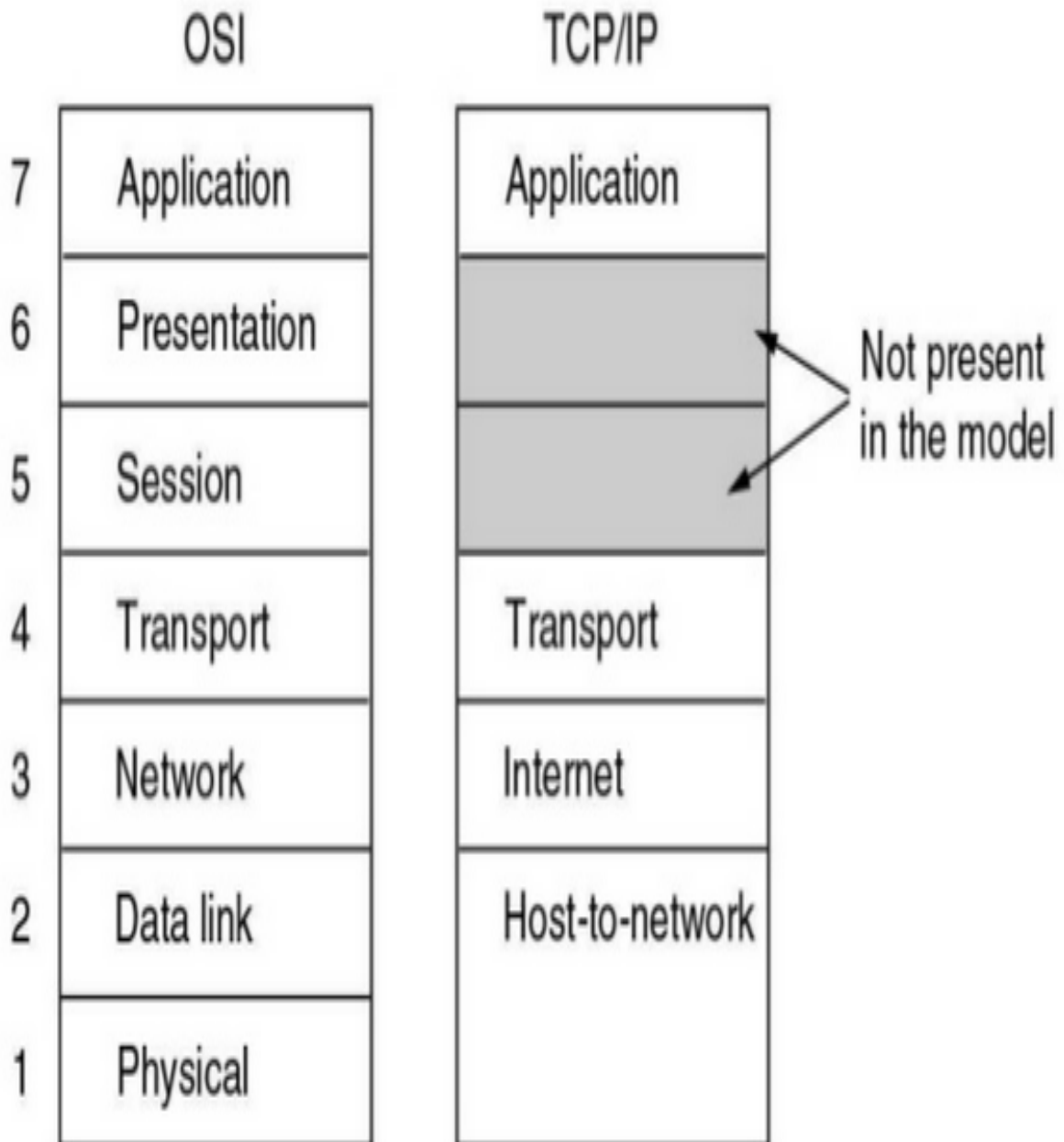
1. To connect multiple networks together so that they appear as a single network.
2. To survive after partial subnet hardware failures.
3. To provide a flexible architecture.

## Transmission control protocol/ information protocol

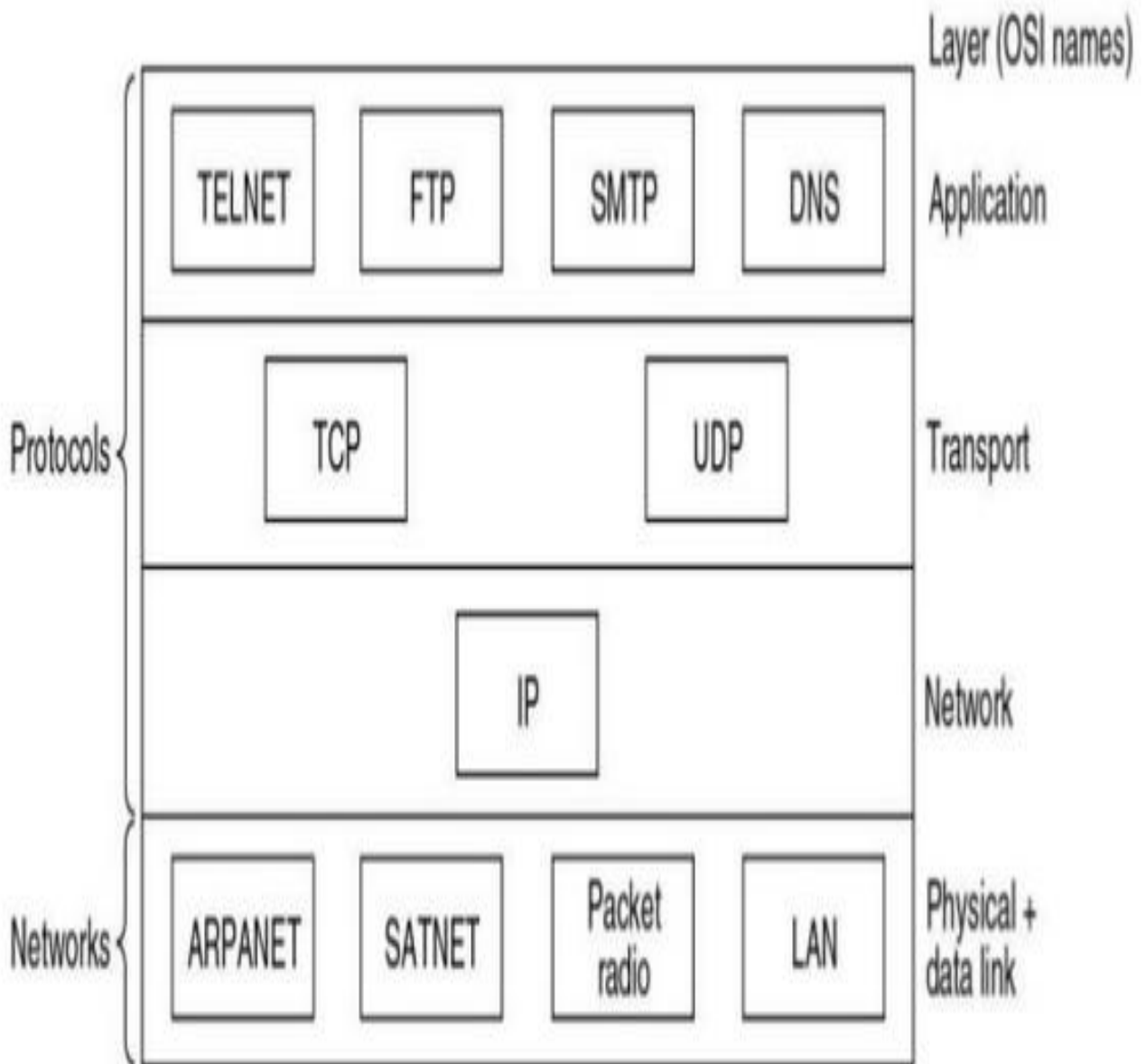
Unlike OSI reference model, TCP/IP reference model has only 4 layers. They are,

1. Host-to-Network Layer
2. Internet Layer
3. Transport Layer
4. Application Layer

- The TCP/IP reference model.



# Protocols and networks in the TCP/IP model initially





# Internet Layer

- Its job is to permit hosts to inject packets into any network and have they travel independently to the destination (potentially on a different network).
- They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired.
- The internet layer defines an official packet format and protocol called **IP (Internet Protocol)**.
- The job of the internet layer is to deliver IP packets where they are supposed to go.
- Packet routing is clearly the major issue here, as is avoiding congestion.

# Transport Layer

- It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer.
- Two end-to-end transport protocols have been defined here.
- **TCP** (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet.
  - It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream.

TCP also handles flow control

- **UDP** (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own.
  - It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery(transmitting speech or video.)

# Application Layer

- The TCP/IP model does not have session or presentation layers. On top of the transport layer is the application layer. It contains all the higher-level protocols.
- The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP).
- The virtual terminal protocol allows a user on one machine to log onto a distant machine and work there.
- The file transfer protocol provides a way to move data efficiently from one machine to another.
- Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

# Telnet

- **Telnet** is a network protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.
- The term *telnet* may also refer to the software that implements the client part of the protocol. Telnet client applications are available for virtually all computer platforms. *To telnet* means to establish a connection with the Telnet protocol, either with command line client or with a programmatic interface.
- The Telnet program runs on your computer and connects your PC to a server on the network. You can then enter commands through the Telnet program and they will be executed as if you were entering them directly on the server. This enables you to control the server and communicate with other servers on the network.
- To start a Telnet session, you must log in to a server by entering a valid username and password. Telnet is a common way to remotely control Web servers.



# File Transfer Protocol (FTP)

- The **File Transfer Protocol (FTP)** is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.
- FTP is built on a client-server architecture and uses separate control and data connections between the client and the server.
- The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as Web page editors

- **Simple Mail Transfer Protocol (SMTP)** is an Internet standard for electronic mail (e-mail) transmission
- Short for *Simple Mail Transfer Protocol*, a protocol for sending e-mail messages between servers.
- Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client using either POP or IMAP.
- In computing, the **Post Office Protocol (POP)** is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.
- Virtually all modern e-mail clients and servers support POP and IMAP (**Internet Message Access Protocol**) are the two most prevalent Internet standard protocols for e-mail retrieval, with many webmail service providers such as Gmail, Outlook.com and Yahoo! Mail also providing support for either IMAP or POP3 to allow mail to be downloaded.



- The **Domain Name System (DNS)** is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.
- The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain. Authoritative name servers are assigned to be responsible for their supported domains. This mechanism provides distributed and fault tolerant service and was designed to avoid the need for a single central database.

# IP Address

- An **Internet Protocol address (IP address)** is a numerical label assigned to each device (e.g., computer, printer) participating in a computer network that uses the Internet Protocol for communication.
- An IP address serves two principal functions:
  - host or network interface identification
  - location addressing.



# IP Address

- IP addresses format
  - binary numbers,
  - usually stored in text files and displayed in human-readable notations, such as 172.16.254.1 (IPv4)
- IPv4
  - 32-bit number
  - still in use today.
- IPv6
  - 128-bits number
  - developed in 1995.

# IP Address

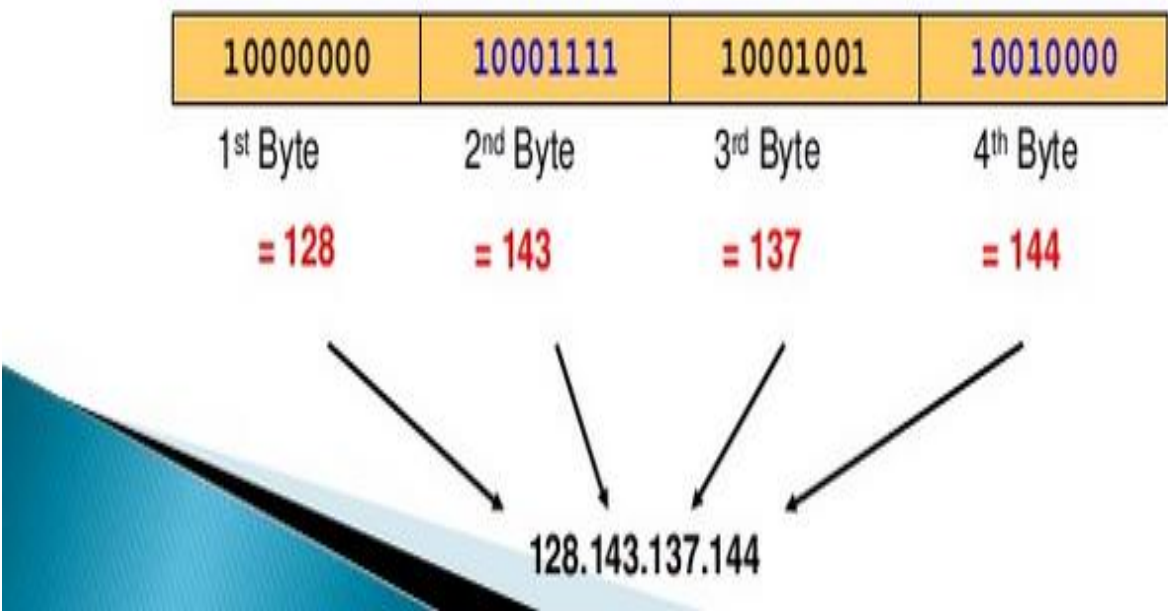
- ▶ What is an IP address...?

- An IP address is a unique global address for a network interface

- is a **32 bit long** identifier

- encodes a network number (**network prefix**) and a **host number**

10000000	10001111	10001001	10010000
1 <sup>st</sup> Byte	2 <sup>nd</sup> Byte	3 <sup>rd</sup> Byte	4 <sup>th</sup> Byte
= 128	= 143	= 137	= 144



128.143.137.144

# Class Ranges of Internet Addresses

	From	To
Class A	<div><div>0.0.0.0</div><div>Netid Hostid</div></div>	<div><div>127.255.255.255</div><div>Netid Hostid</div></div>
Class B	<div><div>128.0.0.0</div><div>Netid Hostid</div></div>	<div><div>191.255.255.255</div><div>Netid Hostid</div></div>
Class C	<div><div>192.0.0.0</div><div>Netid Hostid</div></div>	<div><div>223.255.255.255</div><div>Netid Hostid</div></div>
Class D	<div><div>224.0.0.0</div><div>Group address</div></div>	<div><div>239.255.255.255</div><div>Group address</div></div>
Class E	<div><div>240.0.0.0</div><div>Undefined</div></div>	<div><div>255.255.255.255</div><div>Undefined</div></div>

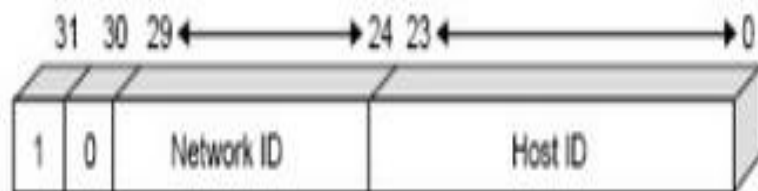
# Class A

- ▶ Class A addresses are assigned to networks with a **very large number of hosts**
- ▶ The high-order bit in a class A address is always set to zero.
- ▶ The next seven bits (completing the first octet) complete the network ID.
- ▶ The remaining 24 bits represent the host ID.



# Class B

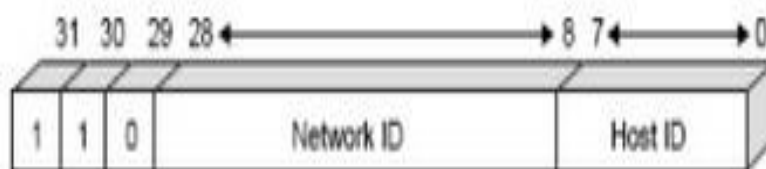
- ▶ Class B addresses are assigned to medium-sized to large-sized networks.
- ▶ The two high-order bits in a class B address are always set to binary 1 0.
- ▶ The next 14 bits complete the network ID.
- ▶ The remaining 16 bits represent the host ID.





# Class C

- ▶ Class C addresses are used for small networks.
- ▶ The three high-order bits in a class C address are always set to binary 1 1 0.
- ▶ The next 21 bits complete the network ID.
- ▶ The remaining 8 bits represent the host ID.



# Class D & E

- ▶ Class D addresses are reserved for IP multicast addresses.
  - The four high-order bits in a class D address are always set to binary 1 1 1 0.
  - The remaining bits are for the address that interested hosts recognize.
- ▶ Class E is an experimental address that is reserved for future use
  - The high-order bits in a class E address are set to 1111.

# Class Ranges of Network IDs...

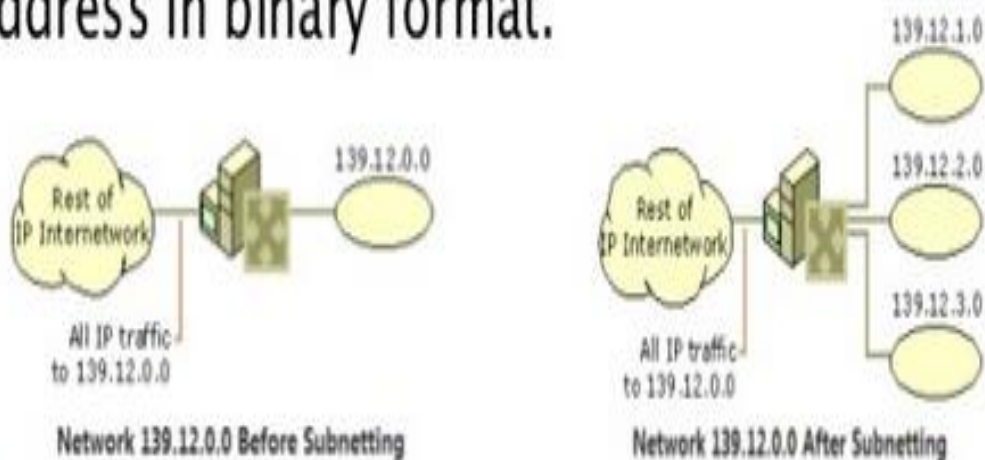
Address Class	First Network ID	Last Network ID
Class A	1.0.0.0	126.0.0.0
Class B	128.0.0.0	191.255.0.0
Class C	192.0.0.0	223.255.255.0

- ▶ The network ID cannot begin with the number 127. The number 127 in a class A address is reserved for internal loopback functions.
- ▶ All bits within the network ID cannot be set to 1. All 1's in the network ID are reserved for use as an IP broadcast address.



# Subnetting....

- ▶ Subnetting enables the network administrator to further divide the host part of the address into two or more subnets.
- ▶ In this case, a part of the host address is reserved to identify the particular subnet.
- ▶ This is easier to see if we show the IP address in binary format.



# Subnet Mask

- A subnet mask is a number that defines a range of [IP addresses](#) available within a [network](#). A single subnet mask limits the number of valid IPs for a specific network. Multiple subnet masks can organize a single network into smaller networks (called subnetworks or subnets). Systems within the same subnet can communicate directly with each other, while systems on different subnets must communicate through a [router](#).

# Example

- subnet mask 255.255.255.192, which limits the number of IP addresses to 64.
- Large networks with several thousand machines may use a subnet mask of 255.255.0.0. This is the default subnet mask used by Class B networks and provides up to 65,536 IP addresses ( $256 \times 256$ ).
- The largest Class A networks use a subnet mask of 255.0.0.0, allowing for up to 16,777,216 IP addresses ( $256 \times 256 \times 256$ ).

# Subnet Mask....

- ▶ Subnet masks are frequently expressed in dotted decimal notation.
- ▶ Subnet mask is not an IP address.
- ▶ Each host on a TCP/IP network requires a subnet mask even on a single segment network.

Address Class	Bits for Subnet Mask	Subnet Mask
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

# Example

- **255.255.255.0**
- In the example above, the first three sections are full (255 out of 255), meaning the IP addresses of devices within the subnet mask must be identical in the first three sections. The last section of each computer's IP address can be anything from 0 to 255. If the subnet mask is defined as 255.255.255.0, the IP addresses 10.0.1.99 and 10.0.1.100 are in the same subnet, but 10.0.2.100 is not.
- A subnet mask of 255.255.255.0 allows for close to 256 unique hosts within the network (since not all 256 IP addresses can be used).

# Subnet Mask Hierarchy

Subnet Mask	CIDR	Binary Notation	Available Addresses Per Subnet
255.255.255.255	/32	11111111.11111111.11111111.11111111	1
255.255.255.254	/31	11111111.11111111.11111111.11111110	2
255.255.255.252	/30	11111111.11111111.11111111.11111100	4
255.255.255.248	/29	11111111.11111111.11111111.11111000	8
255.255.255.240	/28	11111111.11111111.11111111.11110000	16
255.255.255.224	/27	11111111.11111111.11111111.11100000	32
255.255.255.192	/26	11111111.11111111.11111111.11000000	64
255.255.255.128	/25	11111111.11111111.11111111.10000000	128
255.255.255.0	/24	11111111.11111111.11111111.00000000	256
255.255.254.0	/23	11111111.11111111.11111110.00000000	512
255.255.252.0	/22	11111111.11111111.11111100.00000000	1024
255.255.248.0	/21	11111111.11111111.11111000.00000000	2048
255.255.240.0	/20	11111111.11111111.11110000.00000000	4096
255.255.224.0	/19	11111111.11111111.11100000.00000000	8192
255.255.192.0	/18	11111111.11111111.11000000.00000000	16384
255.255.128.0	/17	11111111.11111111.10000000.00000000	32768
255.255.0.0	/16	11111111.11111111.00000000.00000000	65536
255.254.0.0	/15	11111111.11111110.00000000.00000000	131072
255.252.0.0	/14	11111111.11111100.00000000.00000000	262144
255.248.0.0	/13	11111111.11111000.00000000.00000000	524288
255.240.0.0	/12	11111111.11110000.00000000.00000000	1048576
255.224.0.0	/11	11111111.11100000.00000000.00000000	2097152
255.192.0.0	/10	11111111.11000000.00000000.00000000	4194304
255.128.0.0	/9	11111111.10000000.00000000.00000000	8388608
255.0.0.0	/8	11111111.00000000.00000000.00000000	16777216

## Subnet Blocks

Binary	Decimal
$2^8-2^0$	255
$2^8-2^1$	254
$2^8-2^2$	252
$2^8-2^3$	248
$2^8-2^4$	240
$2^8-2^5$	224
$2^8-2^6$	192
$2^8-2^7$	128

Number of valid hosts is always two less than the subnet block

# Journey to IP Versions...

- ▶ IPV(1–3) : were not formally assigned.
- ▶ IPV4 : TCP/IP , 32bit IP address currently used.
- ▶ IPV5 : Internet Stream Protocol (SP)
  - Experimental Protocol
  - Never Introduced for public use.
- ▶ IPV6 : Designed to replace IPV4 , 128bit IP address

# Features of IPV4...

- ▶ Connectionless protocol and best effort based.
- ▶ Simplicity
  - It is simpler and easy to remember
  - Require less memory
- ▶ Familiarity
  - Millions of devices are already knowing it
  - Existing infrastructure already support it



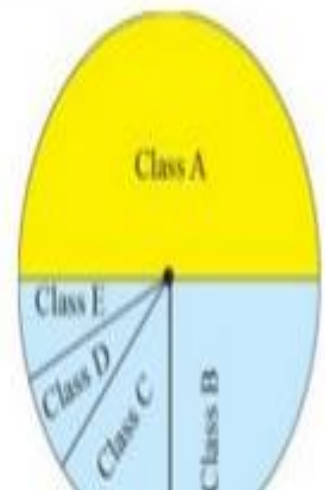
# Benefits of IPV4....

- ▶ Widely support
- ▶ Shorter & Sweeter (header)
- ▶ Support of all Operating Systems
- ▶ All commonly used protocols are supported



# Shortcoming of IPV4....

- ▶ IPV4 specification didn't identify any security mechanism.
- ▶ Millions of class A addresses are wasted.
- ▶ Many class B addresses also wasted.
- ▶ Not so many organizations are so small to have a class C block.
- ▶ Class E addresses were reserved for future purposes.



# IPv4 Supporting Devices..

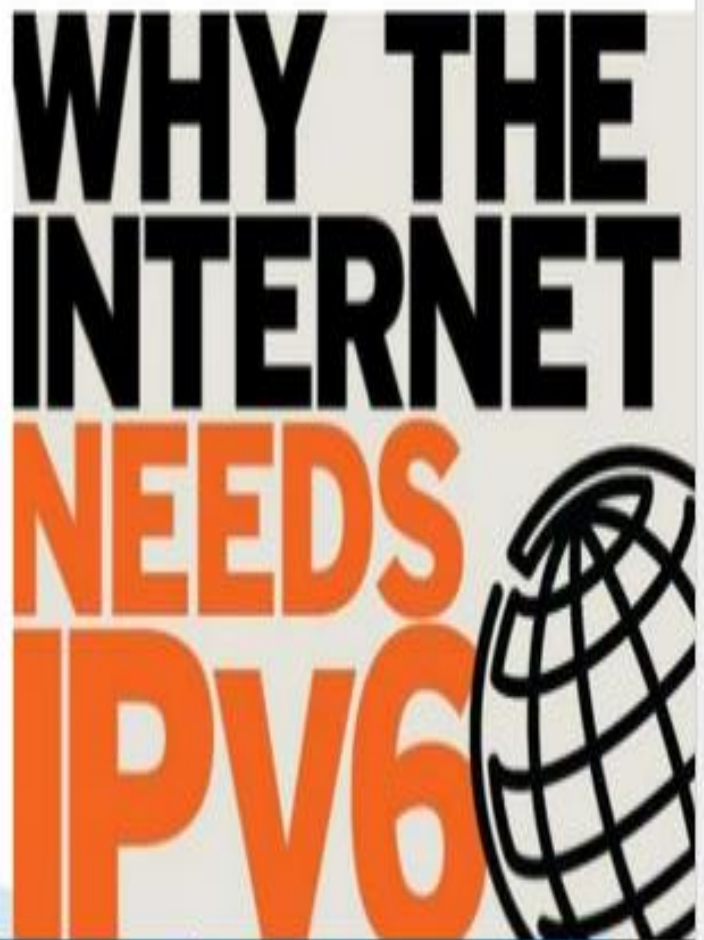
- ▶ PCs
- ▶ Servers
- ▶ Modems
- ▶ Routers
- ▶ Printers
- ▶ Cameras
- ▶ Smart Phones
- ▶ Tablets & Gaming Systems
- ▶ Just about anything else connecting to the Internet



# Why IPV6.....?

IPV6 provides a platform on new internet functionality that will be needed in the immediate future and provide

**flexibility** for future growth and **expansion.**



# Benefits of IPV6.....



---

# IP Based Technologies..

- Internet
- VoIP
- IP – TV
- IP-VPN
- Wireless Mobile Technology
- Internet Broadcasting
- Multihoming


**Q- If a class B network on the Internet has a subnet mask of 255.255.248.0, what is the maximum number of hosts per subnet?**

- - (A) 1022
  - (B) 1023
  - (C) 2046
  - (D) 2047
- Answer (C)

The binary representation of subnet mask is 11111111.11111111.11110000.00000000. There are 21 bits set in subnet. So 11 (32-21) bits are left for host ids. Total possible values of host ids is  $2^{11} = 2048$ . Out of these 2048 values, 2 addresses are reserved. The address with all bits as 1 is reserved as broadcast address and address with all host id bits as 0 is used as network address of subnet.

In general, the number of addresses usable for addressing specific hosts in each network is always  $2^N - 2$  where N is the number of bits for host id.

# Second Explanation -

- **255.255.248.0**  

- $256 - 248 = 8$
- 256
- $256 \times 8 = 2048$
- Excluding the reserved Network and Broadcast addresses, there can be a maximum number of 2046 hosts.



The ip network 200.198.160.0 is using subnet mask 255.255.255.224. Design the subnets

$$256-224 = 32$$

A subnet mask of 255.255.255.224 allows for close to 32 unique hosts within the network.