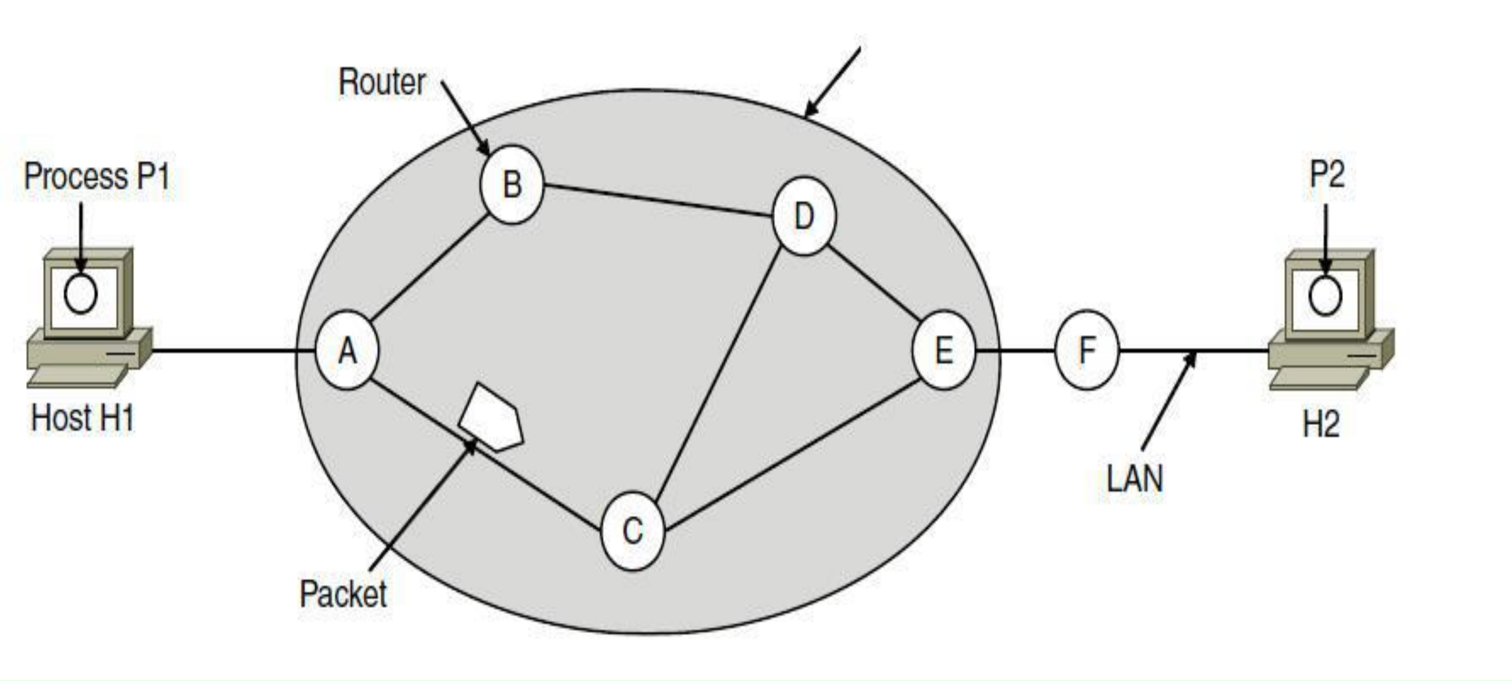
Module 4
Network Layer

Prof. Sachin Chavan
Assistant Professor
MGM'S College of Engineering and Technology
Navi Mumbai

Network Layer Design Issues:

1. Store-and-forward packet switching
2. Services provided to transport layer
3. Implementation of connectionless service
4. Implementation of connection-oriented service
5. Comparison of virtual-circuit and datagram networks (ie Connection Oriented and Connectionless Networks)

1. Store-and-forward packet switching



- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

2. Services provided to transport layer

The network layer provides services to the transport layer at the network layer/transport layer interface. The services need to be carefully designed with the following goals in mind:

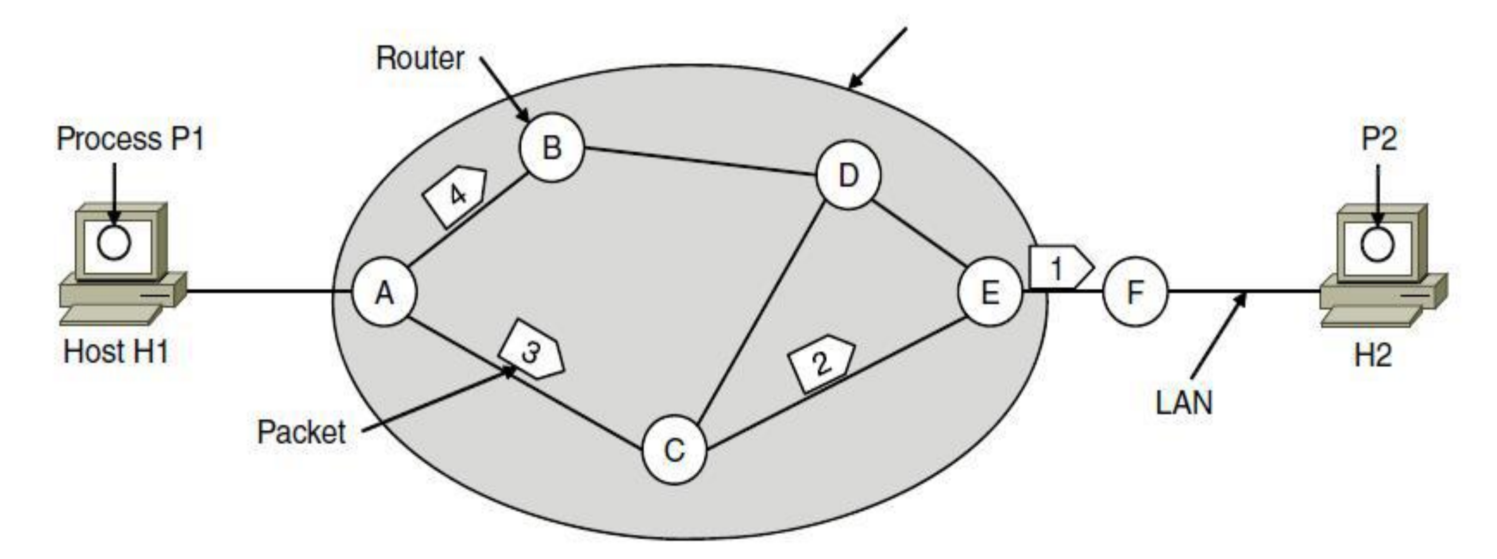
1. Services independent of router technology.

2. Transport layer shielded from number, type, topology of routers.

3. Network addresses available to transport layer use uniform numbering plan

– even across LANs and WANs

3 Implementation of connectionless service



If connectionless service is offered, packets are injected into the network individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams and the network is called a **datagram network.**

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A.

Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.

A’s initial routing table is shown in the figure under the label ‘‘initially.’’

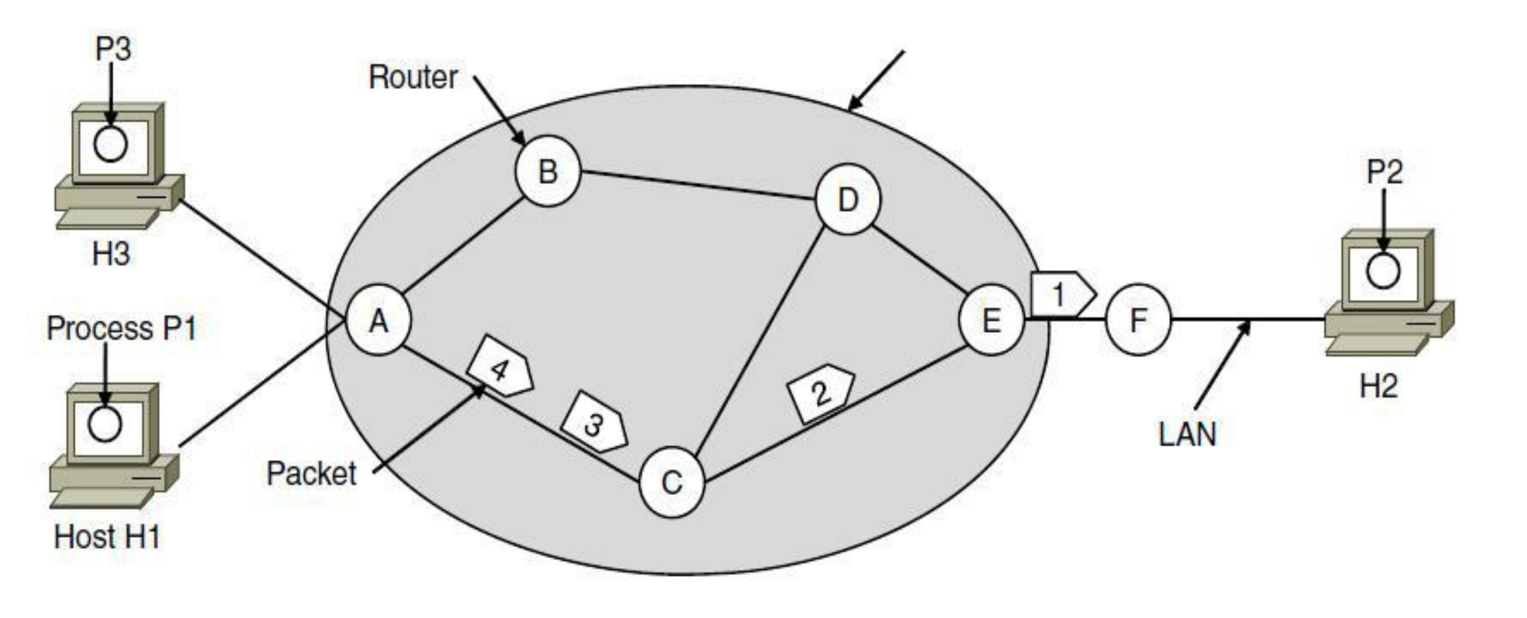
At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packet is forwarded according to A’s table, onto the outgoing link to C within a new frame.

Packet 1 is then forwarded to E and then to F.

However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F. For some reason (traffic jam along ACE path), A decided to send packet 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label ‘‘later.’’

The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

4. Implementation of connection-oriented service



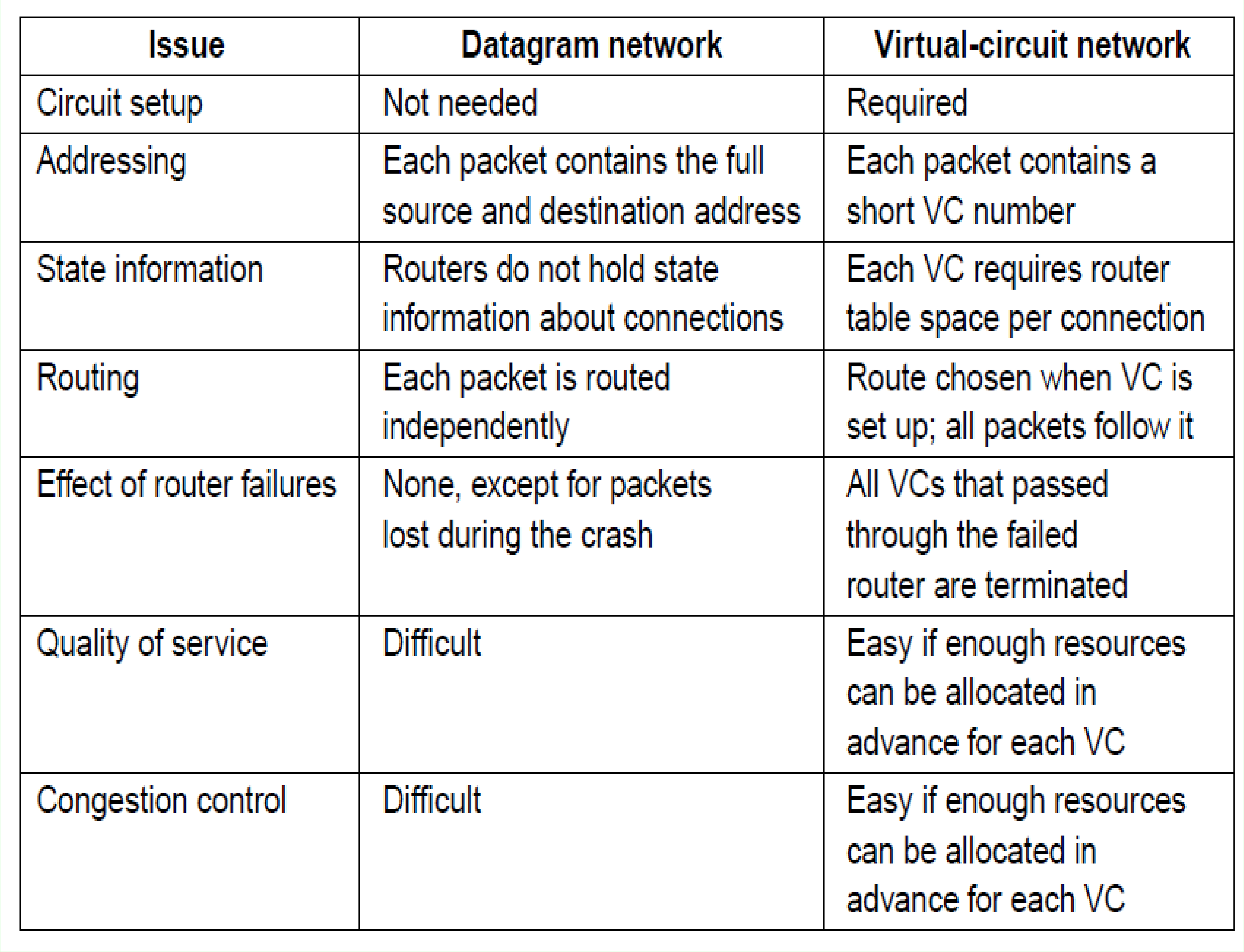
If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit), and the network is called **a virtual-circuit network**

When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation shown in Figure. Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables. The first line of A’s table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.

Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit. This leads to the second row in the tables. Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this process is called label switching. An example of a connection-oriented network service is MPLS (Multi Protocol Label Switching).

5. Comparison of virtual-circuit and datagram networks



Issue	Datagram network	Virtual-circuit network
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

Communication Primitives:

Data is transported over a network by three simple methods,

1.Unicast:-from one source to one destination i.e. One-to-One

(Example: such as a website server, to a single endpoint such as a client PC)

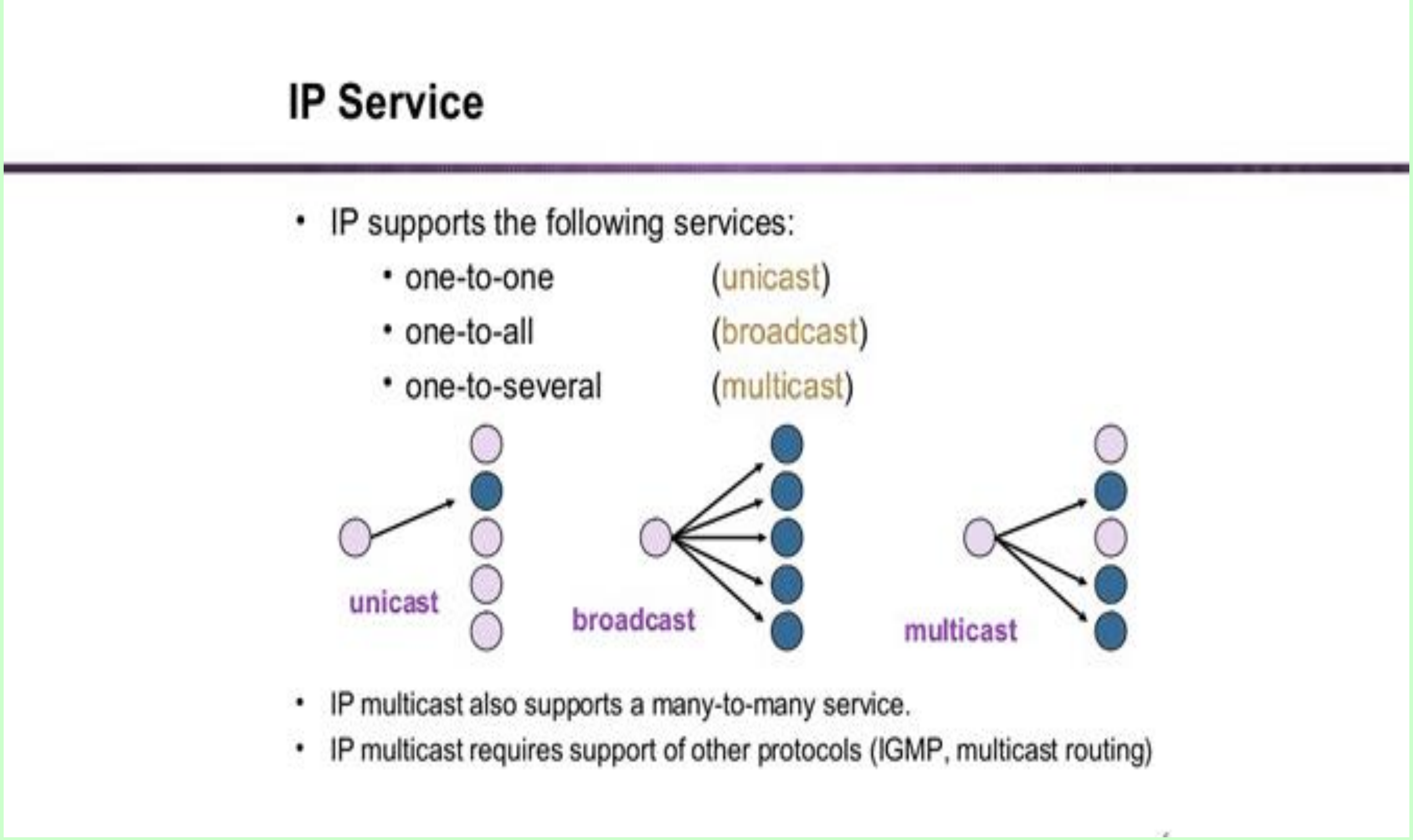
2. Broadcast:- from one source to all possible destinations i.e. One-to-All

(Example:-Cable Television System)

3. Multicast:-from one source to multiple destinations stating an interest in receiving the traffic i.e. One-to-Many

*(Example:-Multicast lets server's direct single copies of data streams that are then simulated and routed to hosts that request it. IP multicast requires support of some other protocols like **IGMP (Internet Group Management Protocol), Multicast routing** for its working. Also in Classful IP addressing **Class D** is reserved for multicast groups.)*

Note:- There is no separate classification for Many-to-Many applications, for example, video conferencing or online gaming, where multiple sources for the same receiver and where receivers often are double as sources. This service model works on the basis of one-to-many multicast and for that reason requires no unique protocol.



IP Service

- IP supports the following services:
  - one-to-one (unicast)
  - one-to-all (broadcast)
  - one-to-several (multicast)

unicast
broadcast
multicast

- IP multicast also supports a many-to-many service.
- IP multicast requires support of other protocols (IGMP, multicast routing)

IPv4 Addressing (classfull and classless)

IP (Internet Protocol) Addressing

An **IP address** does not identify a specific computer. Instead, each IP address identifies a connection between a computer and a network.

A computer with multiple network connections (e.g., a router) must be assigned one IP address for each connection.

IPv4 addresses are:

- Virtual (they are only understood by software)
- Used for all communication in TCP/IP
- 32-bit integers*
- Unique for each host

***Note:**

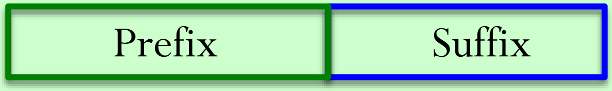
- IPv4 uses 32-bit IP addresses.
- IPv6 uses 128-bit IP addresses.

IP Address Details

IP addresses are divided into two parts

Prefix **--** which identifies the network

Suffix **--** which identifies the host



Prefix Suffix

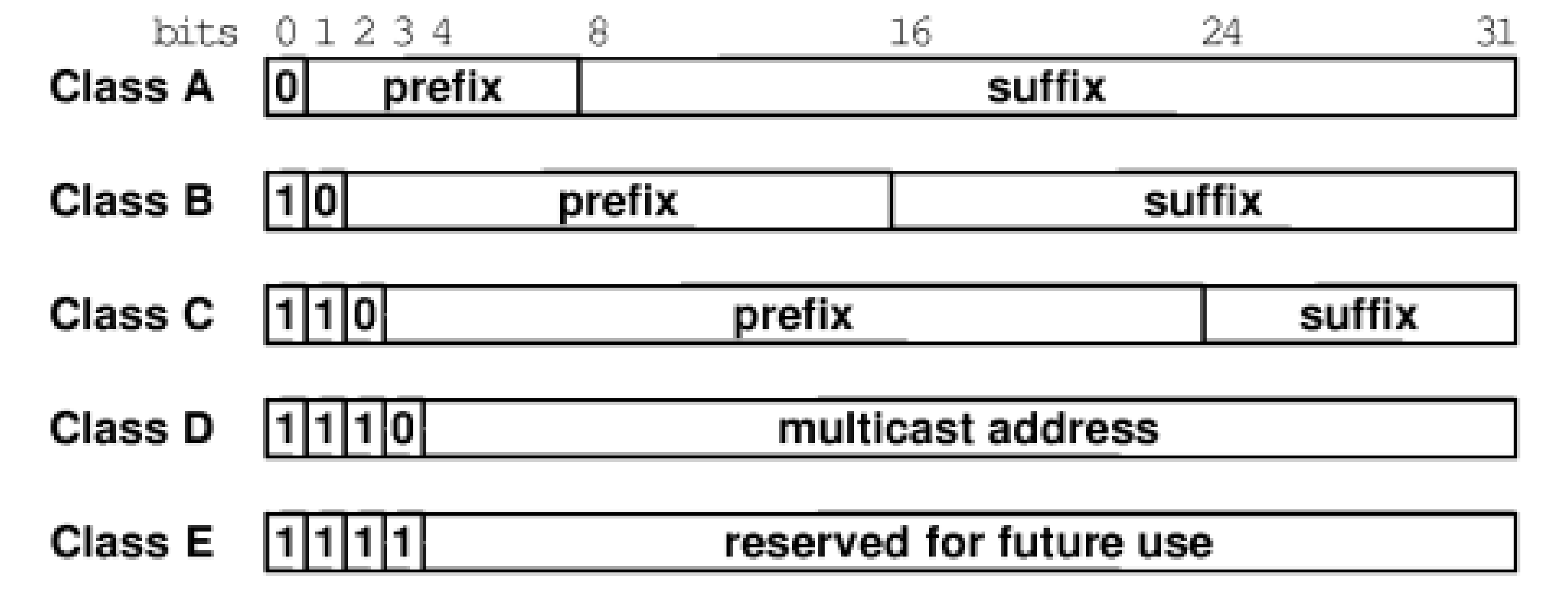
The *Internet Assigned Number Authority* is the global authority that has control over the assignment a unique prefix to each network.

A *local administrator* assigns a unique suffix to each host.

The IP hierarchy guarantees that:

- Each computer is assigned a unique address.
- Suffixes can be assigned locally without global coordination.

Original Classes of Addresses



The initial bits determine the class of the address.

The class determines the boundary between prefix and suffix.

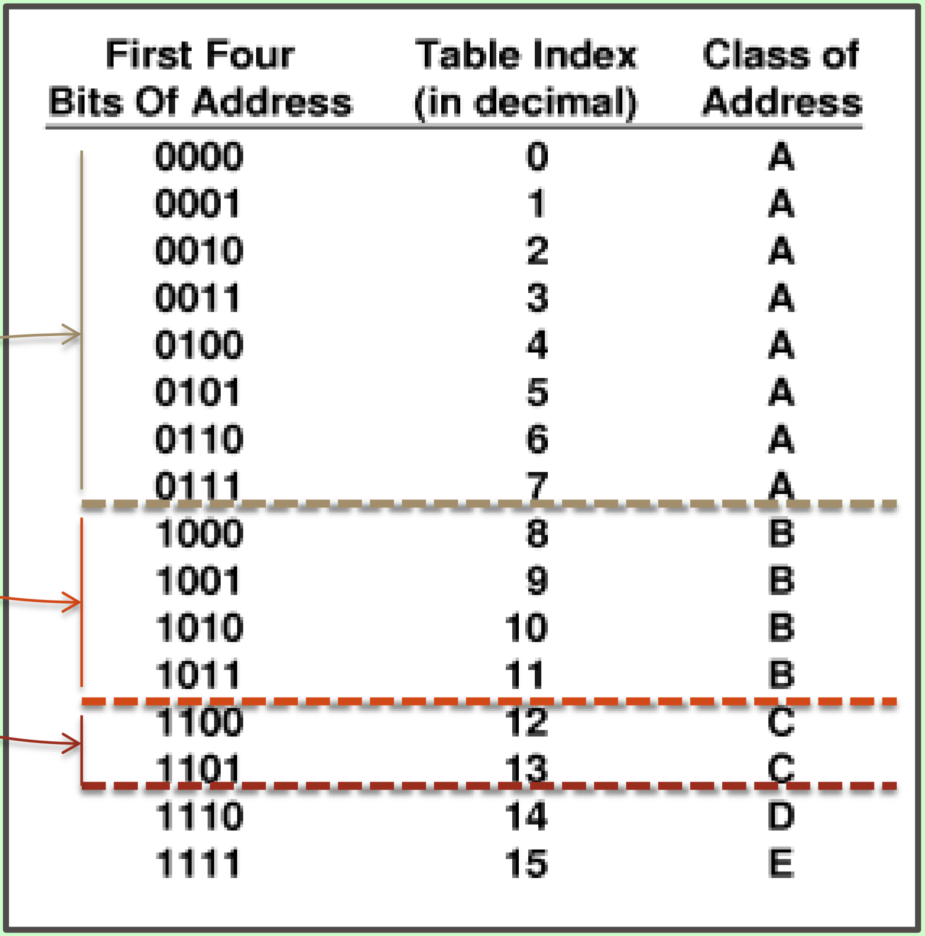
Classes of Addresses (Cont’d)

The maximum network size is determined by the class of the address:

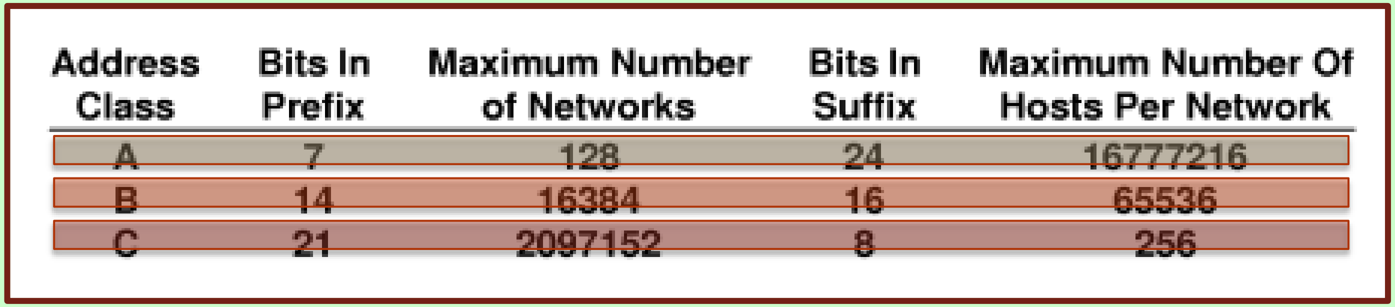
Class A -- large

Class B -- medium

Class C -- small



First Four Bits Of Address	Table Index (in decimal)	Class of Address
0000	0	A
0001	1	A
0010	2	A
0011	3	A
0100	4	A
0101	5	A
0110	6	A
0111	7	A
1000	8	B
1001	9	B
1010	10	B
1011	11	B
1100	12	C
1101	13	C
1110	14	D
1111	15	E



Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

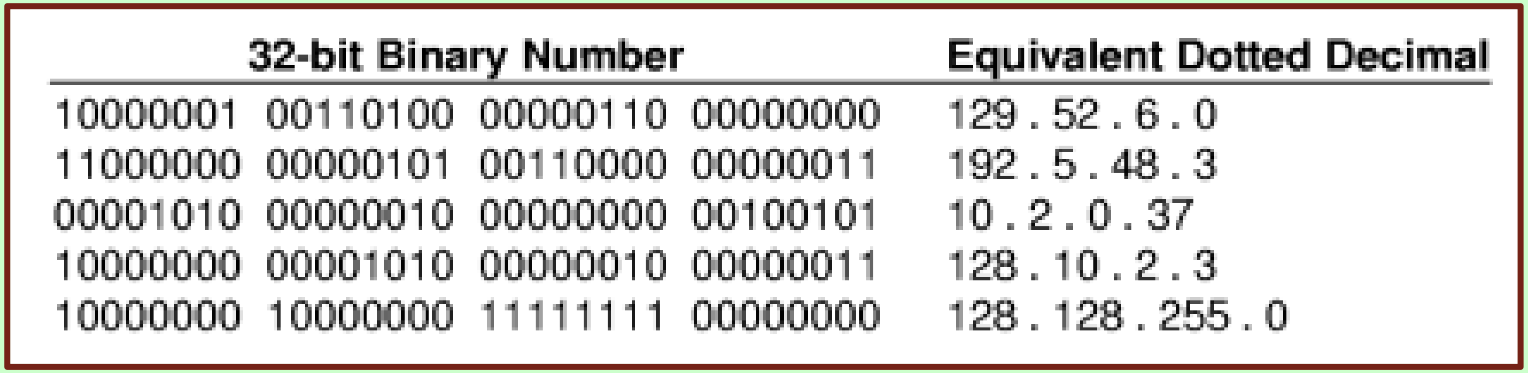
Dotted Decimal Notation

octet
= byte
= 8-bits

Dotted decimal notation is used:

- as shorthand for IP addresses.
- to let humans avoid binary numbers.

Dotted decimal notation represents each octet in decimal separated by dots.
(Note: This is not the same as domain names like www.mgmcet.edu)



32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129 . 52 . 6 . 0
11000000 00000101 00110000 00000011	192 . 5 . 48 . 3
00001010 00000010 00000000 00100101	10 . 2 . 0 . 37
10000000 00001010 00000010 00000011	128 . 10 . 2 . 3
10000000 10000000 11111111 00000000	128 . 128 . 255 . 0

For dotted decimal notation:

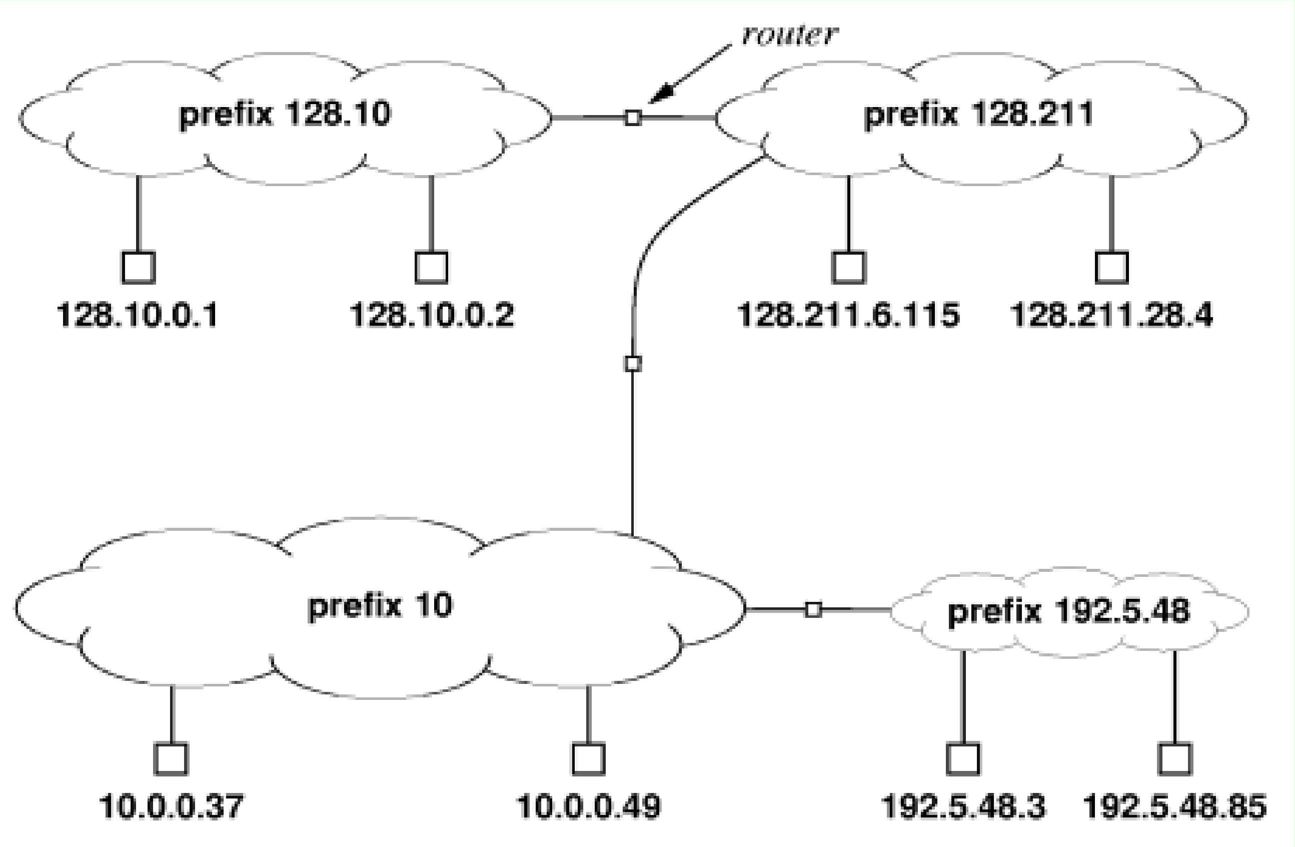
There are four decimal values per 32-bit address.

Each decimal number:

-- Represents eight bits

-- Has a value between 0 and 255

Addressing Example



Subnetting:-

- Creates multiple logical networks that exist within a single Class A, B, or C network.
- If you do not subnet, you will only be able to use one network from your Class A, B, or C network, which is unrealistic
- Each data link on a network must have a unique network ID, with every node on that link being a member of the same network

Benefits of Subnetting

1) Reduced network traffic
2) Optimized network performance
3) Simplified management
4) Facilitated spanning of large geographical distances

Subnets and Classless Addressing (cont'd)

Example:

Consider an ISP that hands out prefixes and a customer of the ISP that requests a prefix for a network that contains 55 hosts.

Classful addressing, would require a complete class C prefix.

8-bits of suffix = 256 possible values = 0..255

Note: *We do not use 0 (0000 0000) or 255 (1111 1111) for hosts*

So Class C gives us 254 possible addresses.

⇒ that means 199 of the 254 possible suffixes would never be assigned

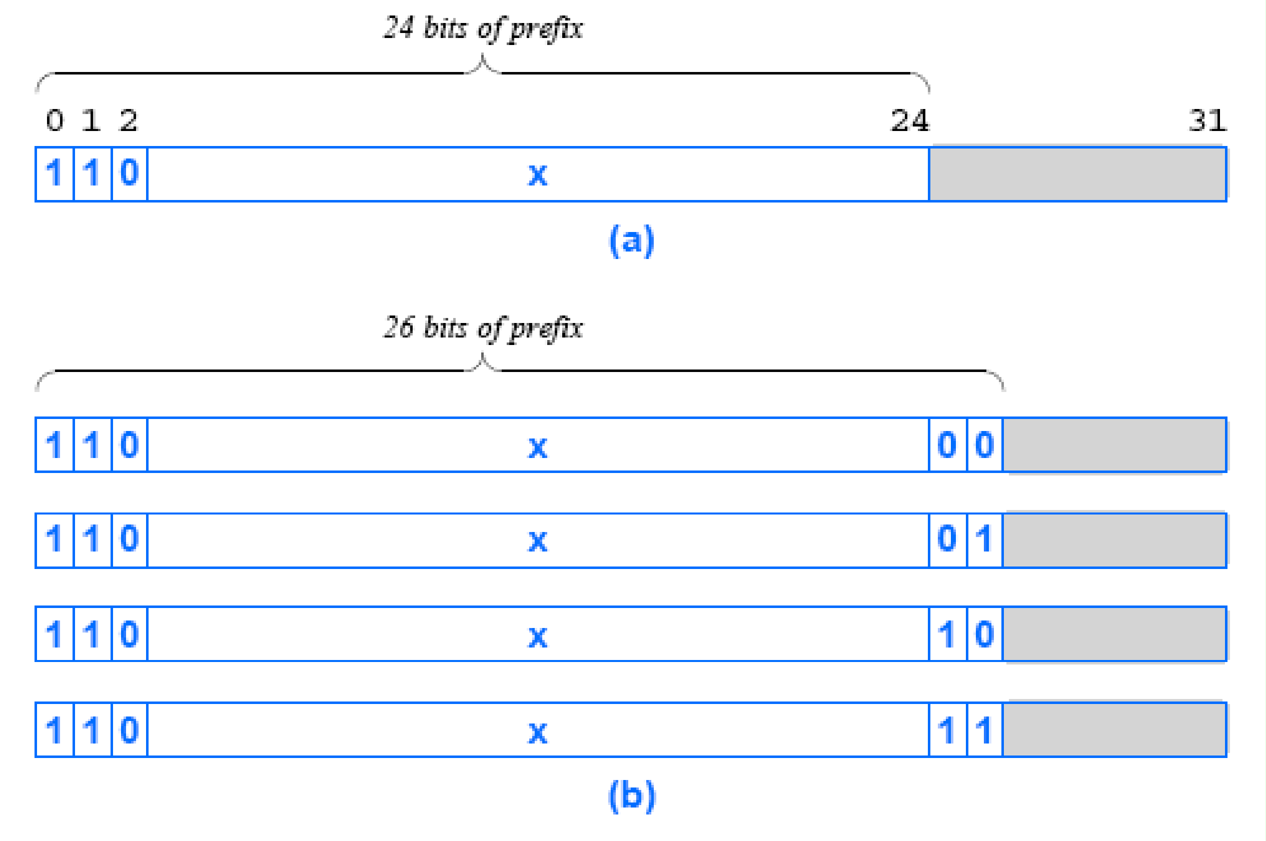
⇒ most of the class C address space is wasted

With Classless addressing, the ISP can assign:

- a prefix that is 26 bits long
- a suffix that is 6 bits long

6-bits of suffix = 2^6 possible values = 64 (minus 0 and 255) = 62 addresses

Subnets and Classless Addressing (cont'd)



This figures illustrates the way classless addressing can be used by an ISP to divide a class C prefix into four (4) longer prefixes:

- each one can accommodate a network of up to 62 hosts
- the host portion of each prefix is shown in gray

Address (or Subnet) Masks

The ***classless*** and ***subnet addressing*** schemes require hosts and routers to store an additional piece of information: ***a value that specifies the exact boundary between the network prefix and the host suffix.***

To mark the boundary, IPv4 uses a 32-bit value known as an address mask, also called a ***subnet mask.***

Why store the boundary size as a bit mask?

- Hosts and routers need to compare the network prefix portion of the address to a value in their forwarding tables.

- The bit-mask representation makes the comparison efficient by making bitwise operations.

Address (or Subnet) Masks (cont'd)

Subnetting Example 1:

Consider the following 32-bit network prefix:

10000000 00001010 00000000 00000000 = 128.10.0.0

Consider a 32-bit mask:

11111111 11111111 00000000 00000000 = 255.255.0.0

Consider a 32-bit destination address on the network which has address:

10000000 00001010 00000010 00000011 = 128.10.2.3

A logical AND (&) between the destination address and the address mask extracts the high-order 16-bits:

10000000 00001010 00000000 00000000 = 128.10.0.0

Classless Inter-Domain Routing (CIDR)

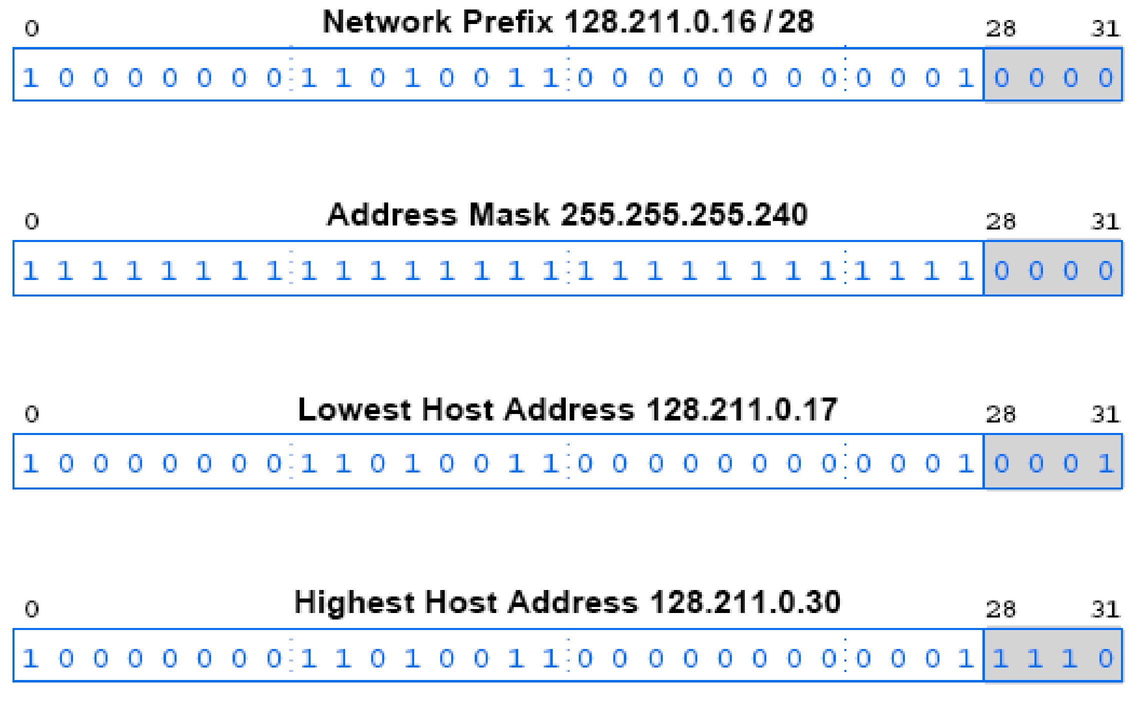
The general form of CIDR notation is: **ddd.ddd.ddd.ddd/m**

- **ddd** is the decimal value for an octet of the address
- **m** is the number of one bits in the mask

Consider the mask needed for a network with 28 bits of prefix:

- It has 28-bits of 1s followed by 4-bits of 0s
- In dotted decimal, the mask is: **255.255.255.240**

In CIDR notation,
the mask is written:
128.211.0.16/28
which specifies
a mask with 28 bits
of prefix and 4 bits
of suffix.



Network Prefix 128.211.0.16 / 28

```
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
```

Address Mask 255.255.255.240

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
```

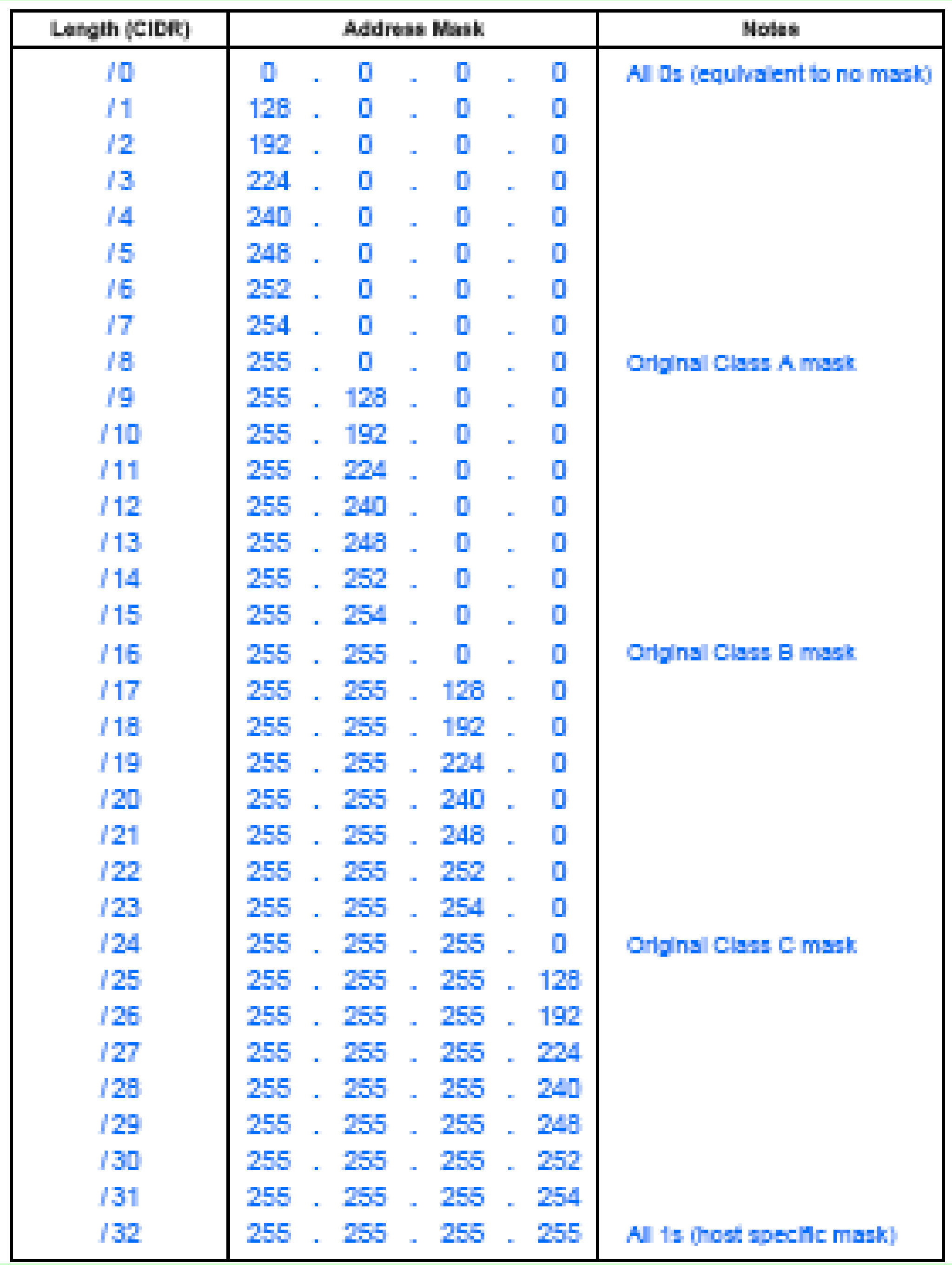
Lowest Host Address 128.211.0.17

```
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
```

Highest Host Address 128.211.0.30

```
1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
```

Figure:
A list of address masks in CIDR notation and in dotted decimal



Length (CIDR)	Address Mask	Notes
/ 0	0 . 0 . 0 . 0	All 0s (equivalent to no mask)
/ 1	128 . 0 . 0 . 0	
/ 2	192 . 0 . 0 . 0	
/ 3	224 . 0 . 0 . 0	
/ 4	240 . 0 . 0 . 0	
/ 5	248 . 0 . 0 . 0	
/ 6	252 . 0 . 0 . 0	
/ 7	254 . 0 . 0 . 0	
/ 8	255 . 0 . 0 . 0	Original Class A mask
/ 9	255 . 128 . 0 . 0	
/ 10	255 . 192 . 0 . 0	
/ 11	255 . 224 . 0 . 0	
/ 12	255 . 240 . 0 . 0	
/ 13	255 . 248 . 0 . 0	
/ 14	255 . 252 . 0 . 0	
/ 15	255 . 254 . 0 . 0	
/ 16	255 . 255 . 0 . 0	Original Class B mask
/ 17	255 . 255 . 128 . 0	
/ 18	255 . 255 . 192 . 0	
/ 19	255 . 255 . 224 . 0	
/ 20	255 . 255 . 240 . 0	
/ 21	255 . 255 . 248 . 0	
/ 22	255 . 255 . 252 . 0	
/ 23	255 . 255 . 254 . 0	
/ 24	255 . 255 . 255 . 0	Original Class C mask
/ 25	255 . 255 . 255 . 128	
/ 26	255 . 255 . 255 . 192	
/ 27	255 . 255 . 255 . 224	
/ 28	255 . 255 . 255 . 240	
/ 29	255 . 255 . 255 . 248	
/ 30	255 . 255 . 255 . 252	
/ 31	255 . 255 . 255 . 254	
/ 32	255 . 255 . 255 . 255	All 1s (host specific mask)

Subnet Masks (Cont’d)

The Mask field in a routing table is used to extract the network part of an address during lookup.

A bit mask makes prefix extraction efficient, using Boolean *AND*.

Example 2:

A datagram is destined for 192.4.10.3

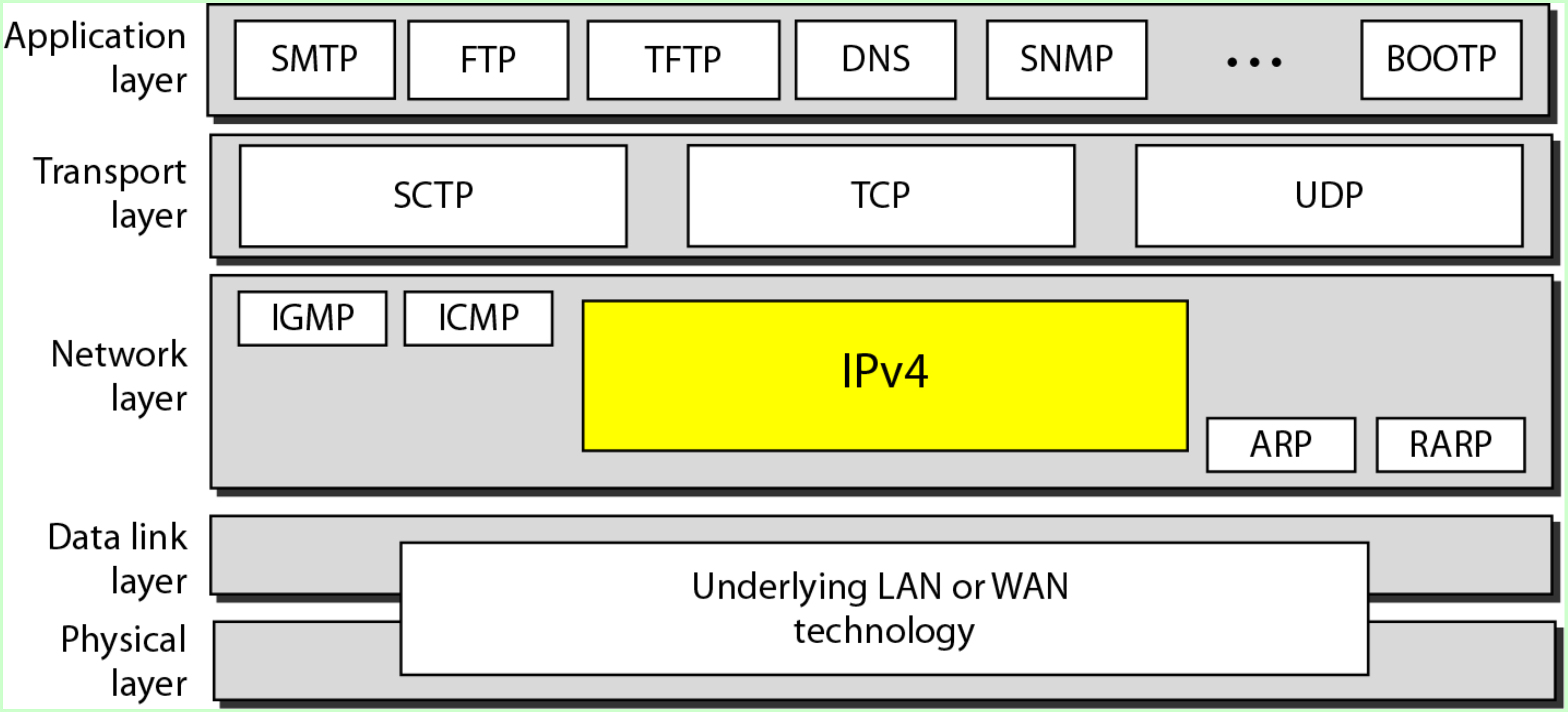
If 192.4.10.3 is a class C[*] network, the subnet mask will be 255.255.255.0.

192.4.10.3 & 255.255.255.0 = 192.4.10.0

[*] Why is 192.4.10.3 considered a class C network?

(Hint: see previous slide.)

Position of IPv4 in TCP/IP protocol suite



IGMP:-Internet Group Management Protocol

SCTP:- Stream Control Transmission Protocol

SMTP:-Simple Mail Transfer Protocol

TFTP:-Trivial File Transfer Protocol

BOOTP:-Bootstrap Protocol

ICMP:-Internet Control Message Protocol

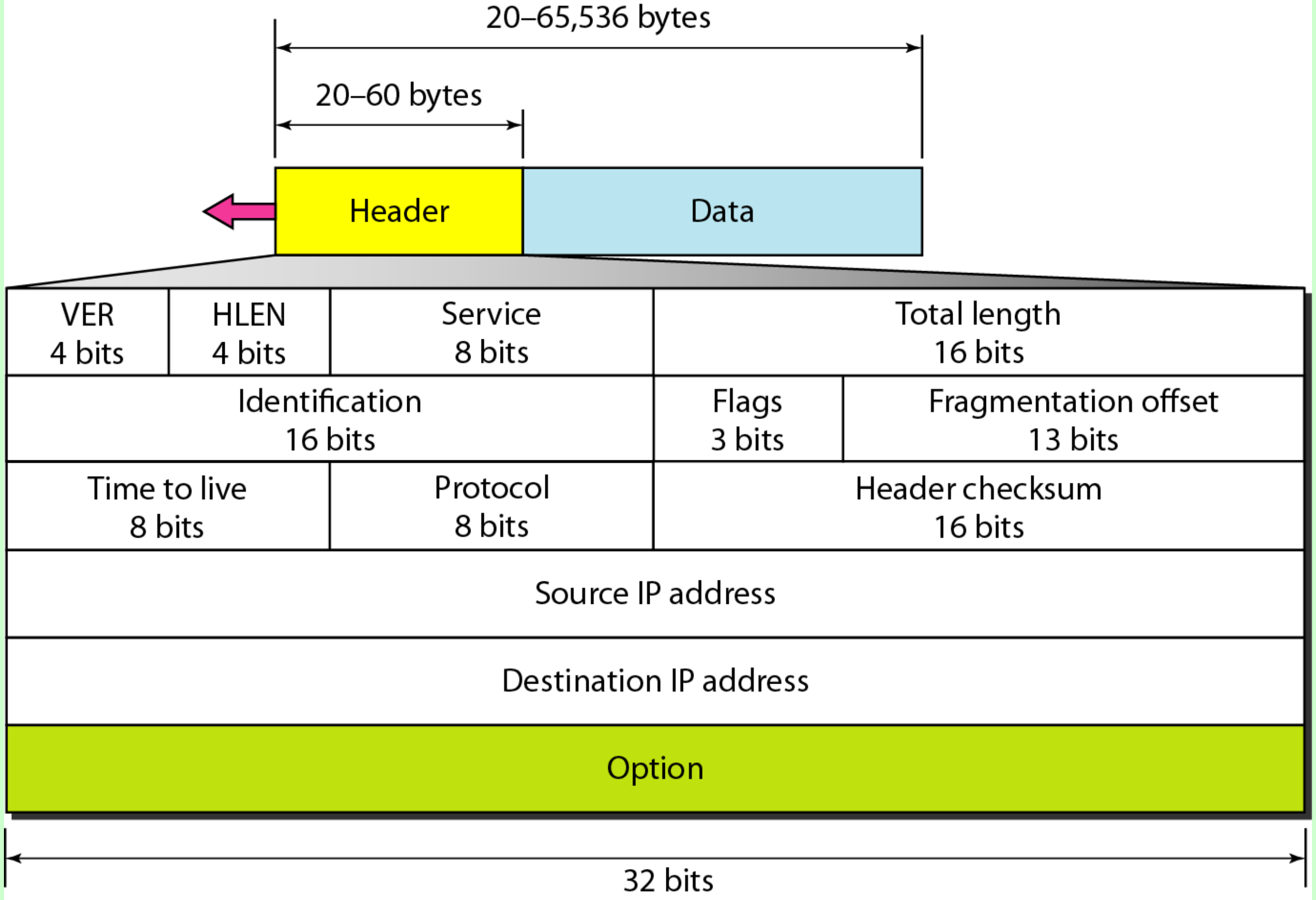
ARP:-Address Resolution Protocol

RARP:-Reverse Address Resolution Protocol

SNMP:-Simple Network Management Protocol

DNS:-Domain Name System

IPv4 datagram format



20–65,536 bytes

20–60 bytes

Header

Data

VER
4 bits | HLEN
4 bits | Service
8 bits | Total length
16 bits

Identification
16 bits | Flags
3 bits | Fragmentation offset
13 bits

Time to live
8 bits | Protocol
8 bits | Header checksum
16 bits

Source IP address

Destination IP address

Option

32 bits

Here is a description of each field:

Version – the version of the IP protocol. For IPv4, this field has a value of 4.

Header length – the length of the header in 32-bit words. The minumum value is 20 bytes, and the maximum value is 60 bytes.

Priority and Type of Service – specifies how the datagram should be handled. The first 3 bits are the priority bits.

Total length – the length of the entire packet (header + data). The minimum length is 20 bytes, and the maximum is 65,535 bytes.

Identification – used to differentiate fragmented packets from different datagrams.

Flags – used to control or identify fragments.

Fragmented offset – used for fragmentation and reassembly if the packet is too large to put in a frame.

Time to live – limits a datagram's lifetime. If the packet doesn't get to its destination before the TTL expires, it is discarded.

Protocol – defines the protocol used in the data portion of the IP datagram. For example, TCP is represented by the number 6 and UDP by 17.

Header checksum – used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.

Source IP address – the IP address of the host that sent the packet.

Destination IP address – the IP address of the host that should receive the packet.

Options – used for network testing, debugging, security, and more. This field is usually empty.

Network Address Translation (NAT):-

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of private IP address to a public IP address is required. **Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on router or firewall.

Network Address Translation (NAT) working –

Generally, the border router is configured for NAT i.e the router which has one interface in local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

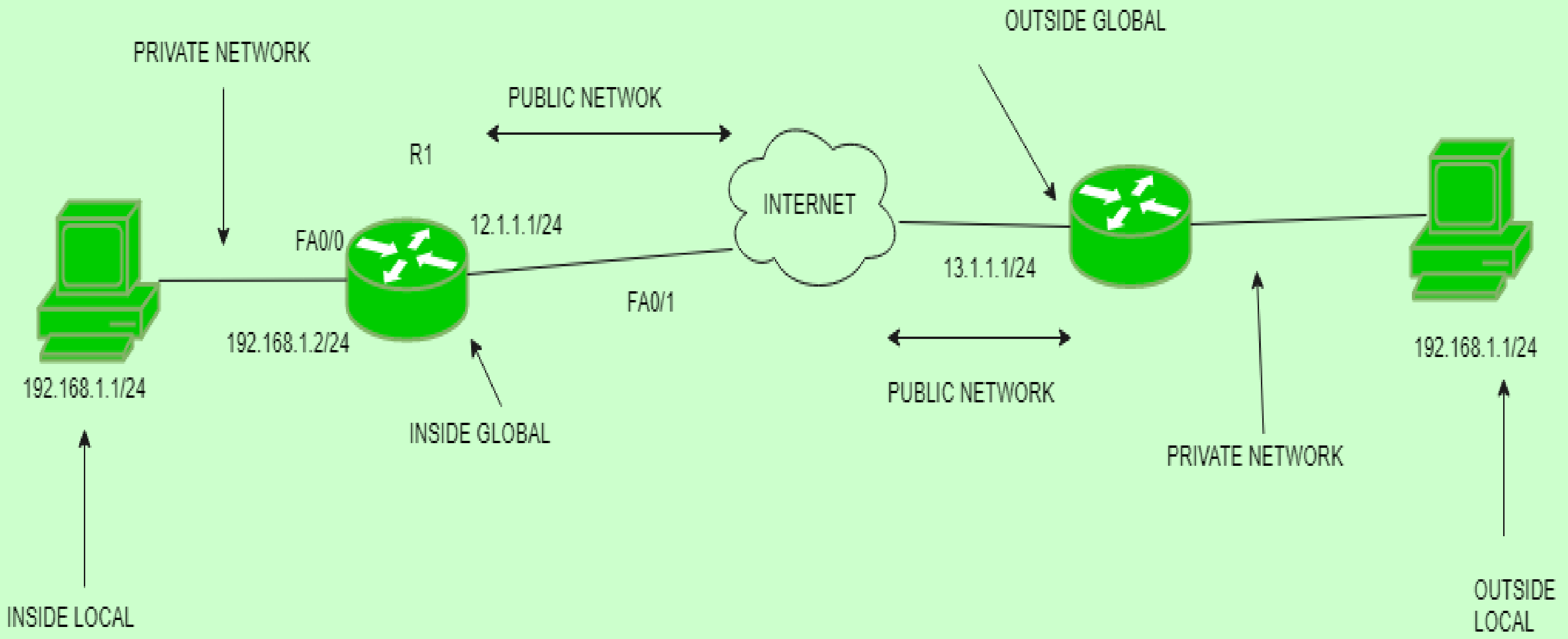
If NAT run out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

Why mask port numbers?

Suppose, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time. If NAT does an only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination. Destination will send replies on the public IP address of the router. Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are same). Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

NAT inside and outside addresses –

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organization. These are the network Addresses in which the translation of the addresses will be done.



Inside local address – An IP address that is assigned to a host on the Inside (local) network. The address is probably not a IP address assigned by the service provider i.e., these are private IP address. This is the inside host seen from the inside network.

Inside global address – IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.

Outside local address – This is the actual IP address of the destination host in the local network after translation.

Outside global address – This is the outside host as seen form the outside network. It is the IP address of the outside destination host before translation.

Network Address Translation (NAT) Types –

There are 3 ways to configure NAT:
1.Static NAT – In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global address. This is generally used for Web hosting. These are not used in organisations as there are many devices who will need Internet access and to provide Internet access, the public IP address is needed.

Suppose, if there are 3000 devices who need access to the Internet, the organisation have to buy 3000 public addresses that will be very costly.

2.Dynamic NAT – In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP address. If the IP address of pool is not free, then the packet will be dropped as an only a fixed number of private IP address can be translated to public addresses.

Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access Internet then the packet will be dropped therefore many private IP addresses are mapped to a pool of public IP addresses. NAT is used when the number of users who wants to access the Internet is fixed. This is also very costly as the organisation have to buy many global IP addresses to make a pool.

3.Port Address Translation (PAT) – This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

Advantages of NAT –

- NAT conserves legally registered IP addresses
- It provides privacy as the device IP address, sending and receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

Disadvantage of NAT –

- Translation results in switching path delays.
- Certain applications will not function while NAT is enabled.
- Complicates tunneling protocols such as IPsec.
- Also, router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

Routing Algorithms

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.

There are two processes inside router:

a) One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is forwarding.

b) The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is routing.

Routing algorithms can be grouped into two major classes:
1) nonadaptive (Static Routing)
2) adaptive. (Dynamic Routing)

1)Nonadaptive algorithm:-

It do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing

2)Adaptive algorithm:-

in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Adaptive algorithms differ in

1) Where they get their information (e.g., locally, from adjacent routers, or from all routers),
2) When they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and
3) What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

This procedure is called dynamic routing

Different Routing Algorithms:-

1)Shortest path algorithm (Dijkastra‘s)

2) Link state routing

3)Distance vector routing

1)Shortest Path Routing (Dijkstra’s) :-

The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph

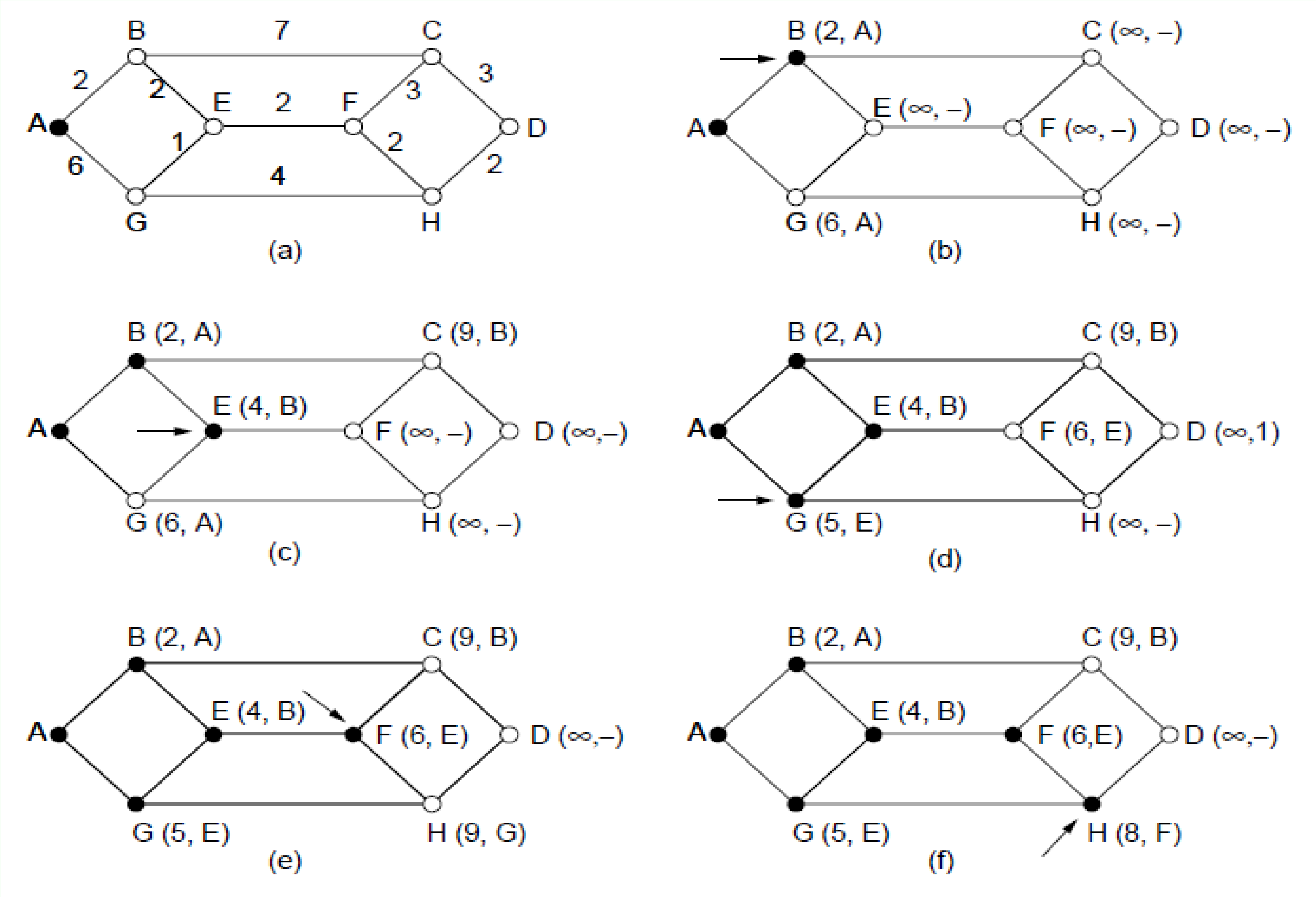
1.Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.

2. Examine each neighbor of the node that was the last permanent node.
3. Assign a cumulative cost to each node and make it tentative
4. Among the list of tentative nodes

a. Find the node with the smallest cost and make it Permanent

b. If a node can be reached from more than one route then select the route with the shortest cumulative cost.

5. Repeat steps 2 to 4 until every node becomes permanent 1)


B
7
C
2
2
3
3
E
2
F
A
D
1
2
6
4
2
G
H
B (2, A)
C (∞, –)
E (∞, –)
A
F (∞, –)
D (∞, –)
G (6, A)
H (∞, –)
B (2, A)
C (9, B)
E (4, B)
A
F (∞, –)
D (∞,–)
G (6, A)
H (∞, –)
B (2, A)
C (9, B)
E (4, B)
A
F (6, E)
D (∞,1)
G (5, E)
H (∞, –)
B (2, A)
C (9, B)
E (4, B)
A
F (6, E)
D (∞,–)
G (5, E)
H (9, G)
B (2, A)
C (9, B)
E (4, B)
A
F (6,E)
D (∞,–)
G (5, E)
H (8, F)

(a)

(b)

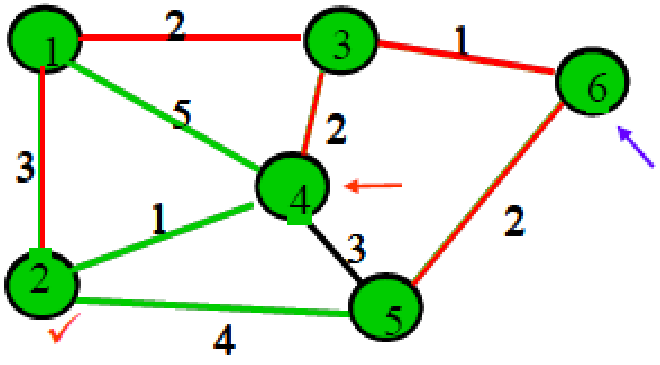
(c)

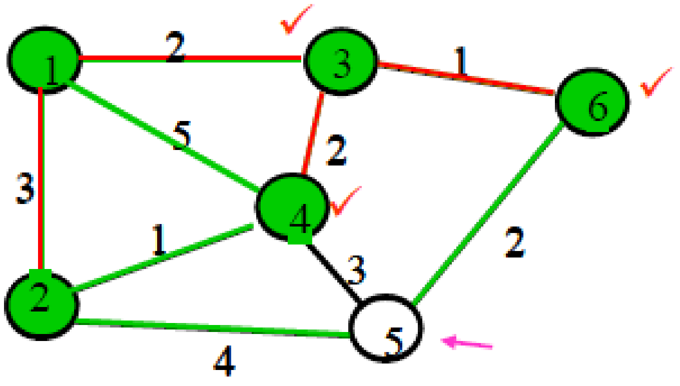
(d)

(e)

(f)

Execution of Dijkstra’s algorithm



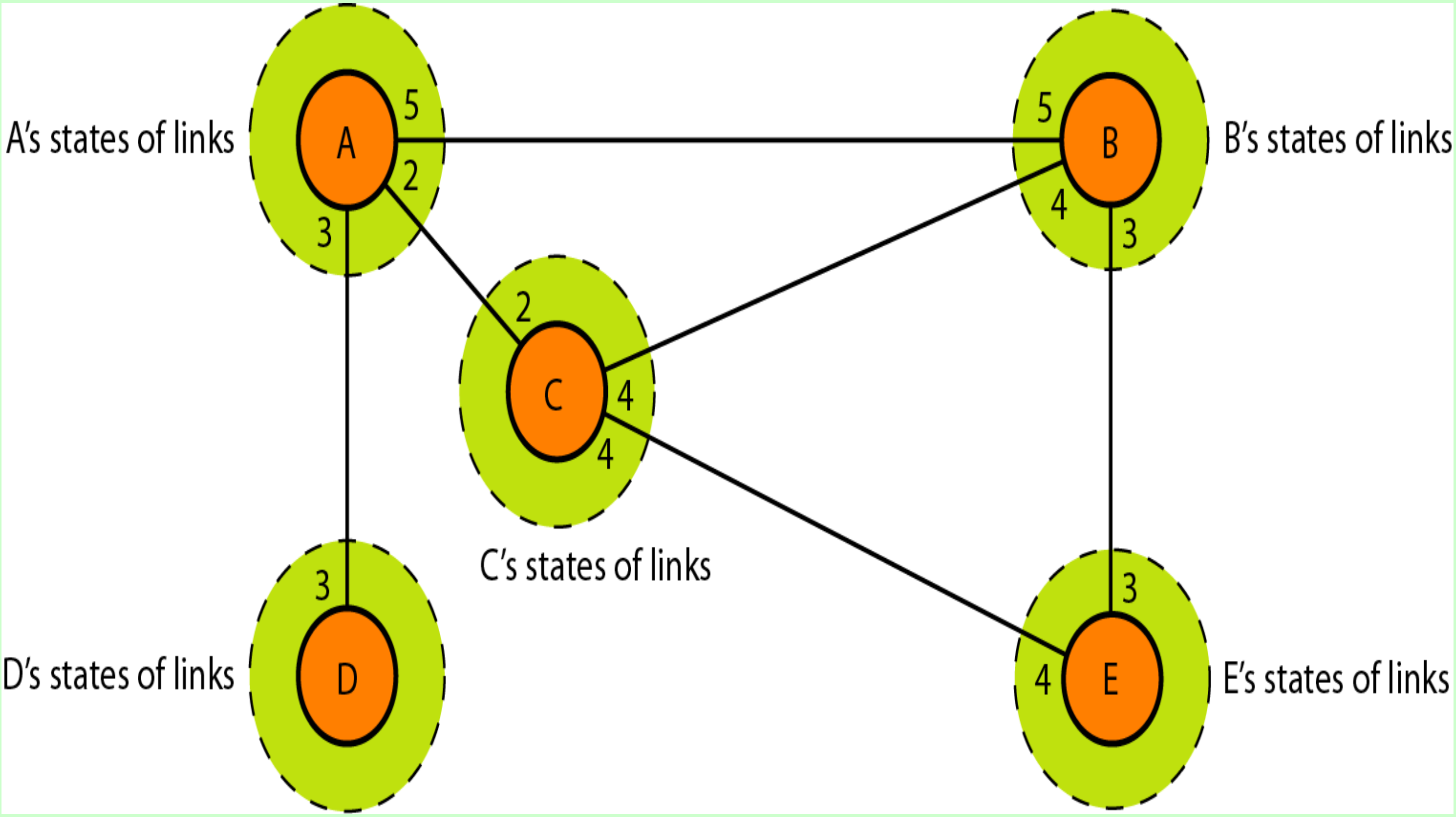


Iteration	Permanent	tentative	D_2	D_3	D_4	D_5	D_6
Initial	{1}	{2,3,4}	3	2 ✓	5	∝	∝
1	{1,3}	{2,4,6}	3 ✓	2	4	∝	3
2	{1,2,3}	{4,6,5}	3	2	4	7	3 ✓
3	{1,2,3,6}	{4,5}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	{5}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	{}	3	2	4	5	3

2) Link State Routing:-

Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links.

In other words, the whole topology can be compiled from the partial knowledge of each node



Building Routing Tables:

1. Creation of the states of the links by each node, called the link state packet (LSP).

2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.

3. Formation of a shortest path tree for each node.

4. Calculation of a routing table based on the shortest path tree

1.Creation of Link State Packet (LSP):-

A link state packet can carry a large amount of information. For the moment, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

LSPs are generated on two occasions:
1. When there is a change in the topology of the domain
2. on a periodic basis: The period in this case is much longer compared to distance vector.
The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

2. Flooding of LSPs:

After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following

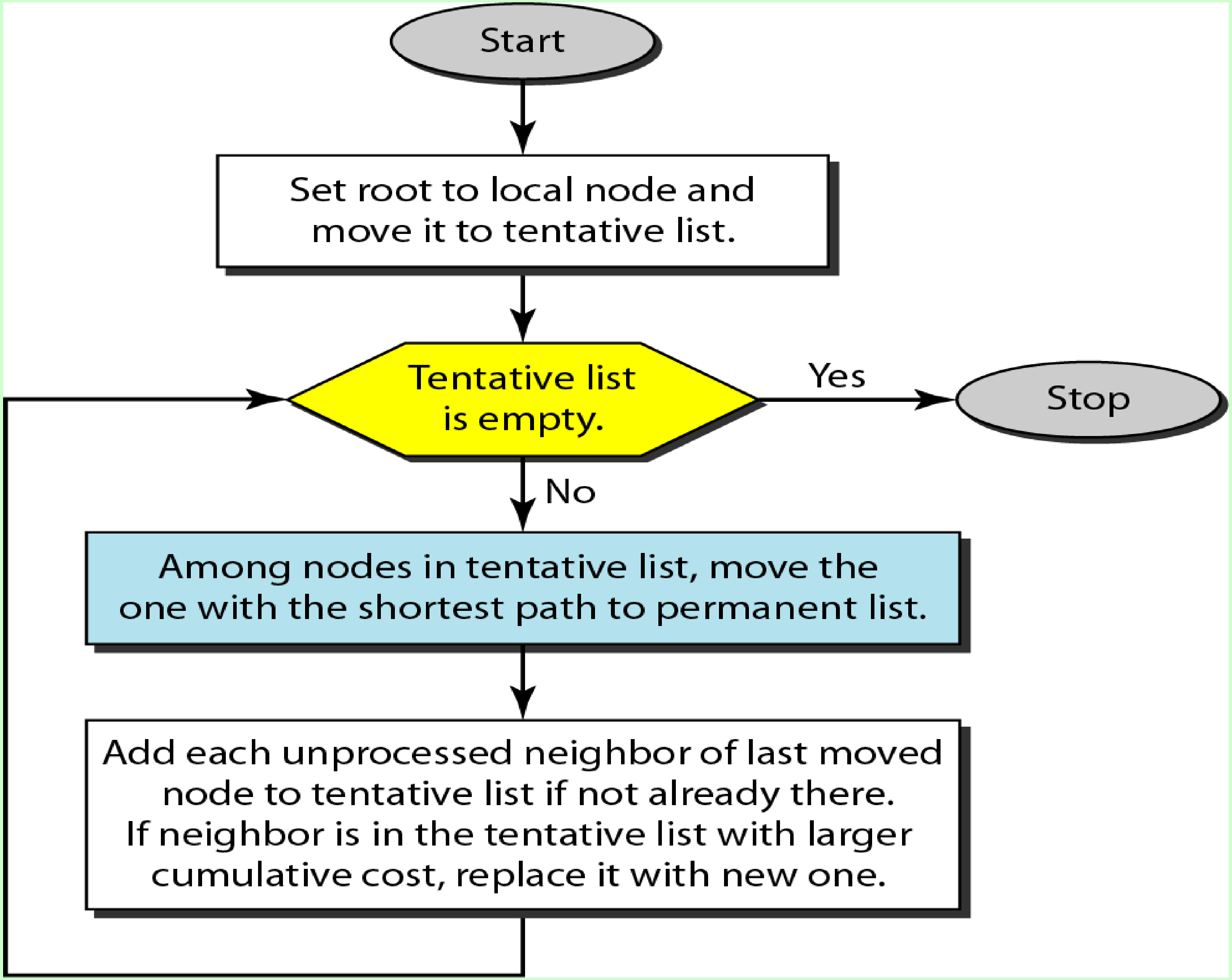
1.The creating node sends a copy of the LSP out of each interface 2.
A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number),it discards the LSP. If it is newer, the node does the following:
a. It discards the old LSP and keeps the new one.
b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node
has only one interface).

3.Formation of Shortest Path Tree:

Dijkstra Algorithm

A shortest path tree is a tree in which the path between the root and every other node is the shortest.
The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: **tentative and permanent.**

It finds the neighbors of a current node, makes
them tentative, examines them, and if they pass the criteria, makes them permanent.



Start

Set root to local node and move it to tentative list.

Tentative list is empty.

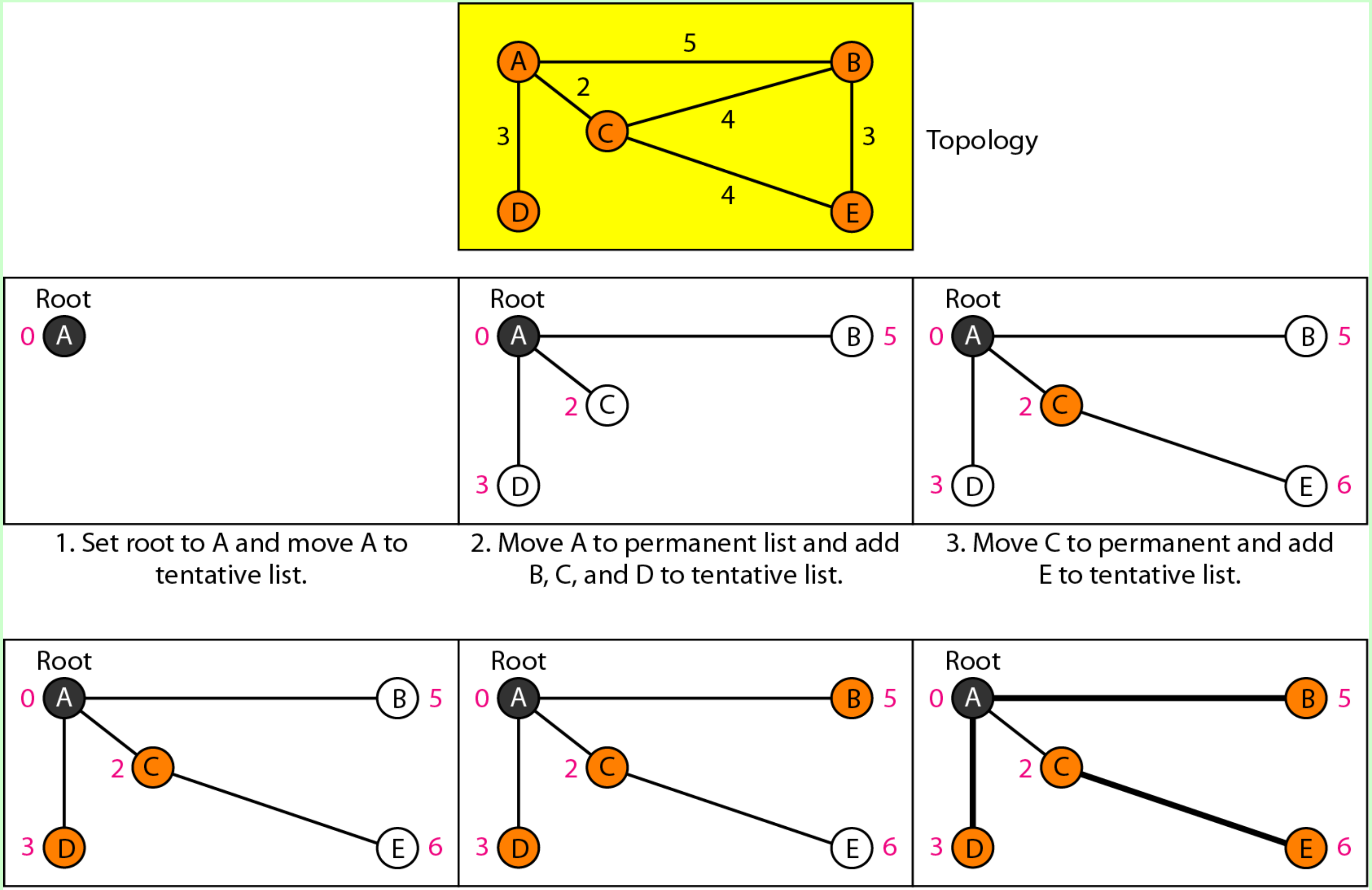
Yes

Stop

No

Among nodes in tentative list, move the one with the shortest path to permanent list.

Add each unprocessed neighbor of last moved node to tentative list if not already there. If neighbor is in the tentative list with larger cumulative cost, replace it with new one.



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.

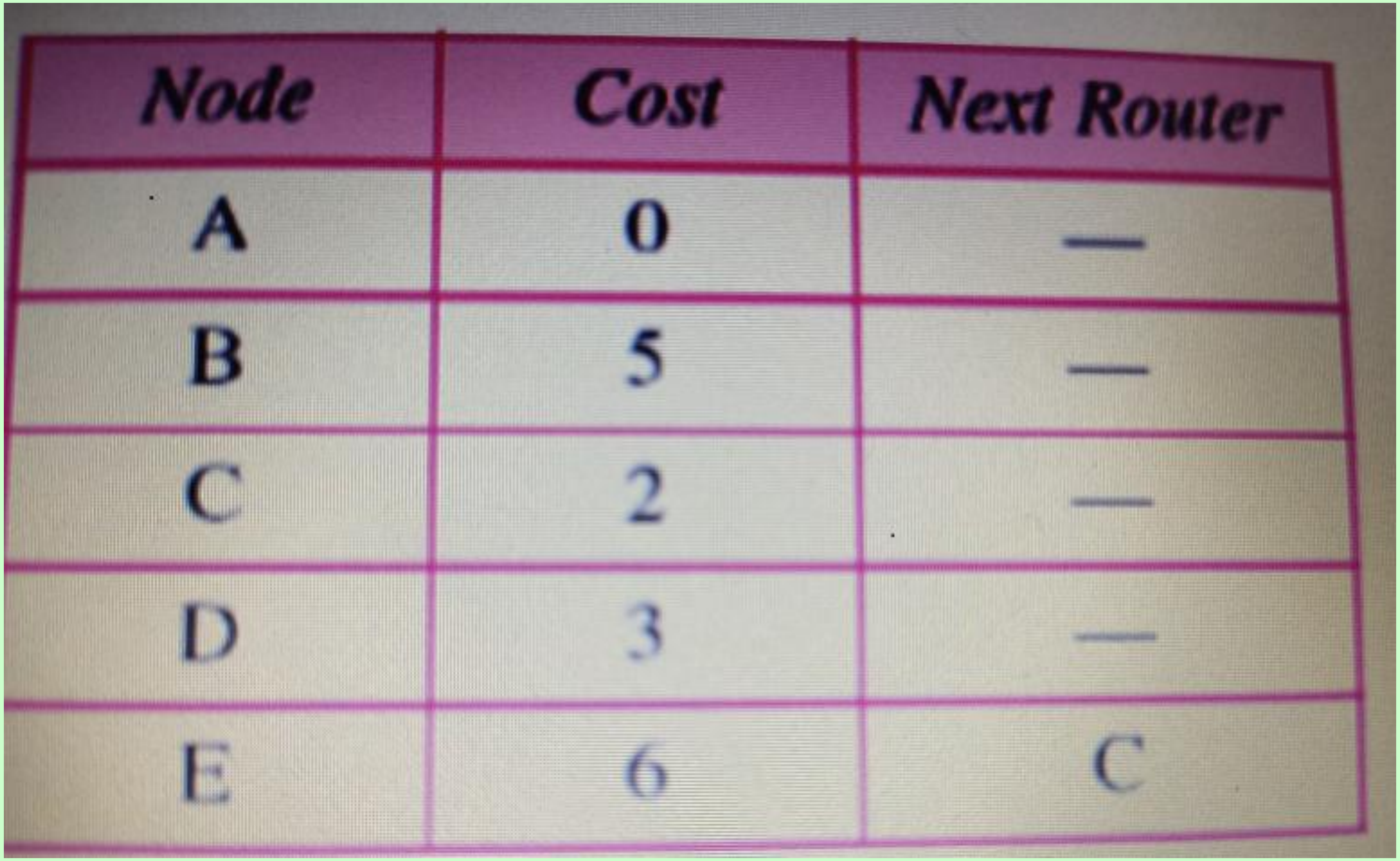
4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

4.Calculation of a routing table :-

-routing table for node A



Node	Cost	Next Router
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

3)Distance Vector Routing algorithm:-

In distance vector routing, the least -cost route between any two nodes is the route with minimum distance.

In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

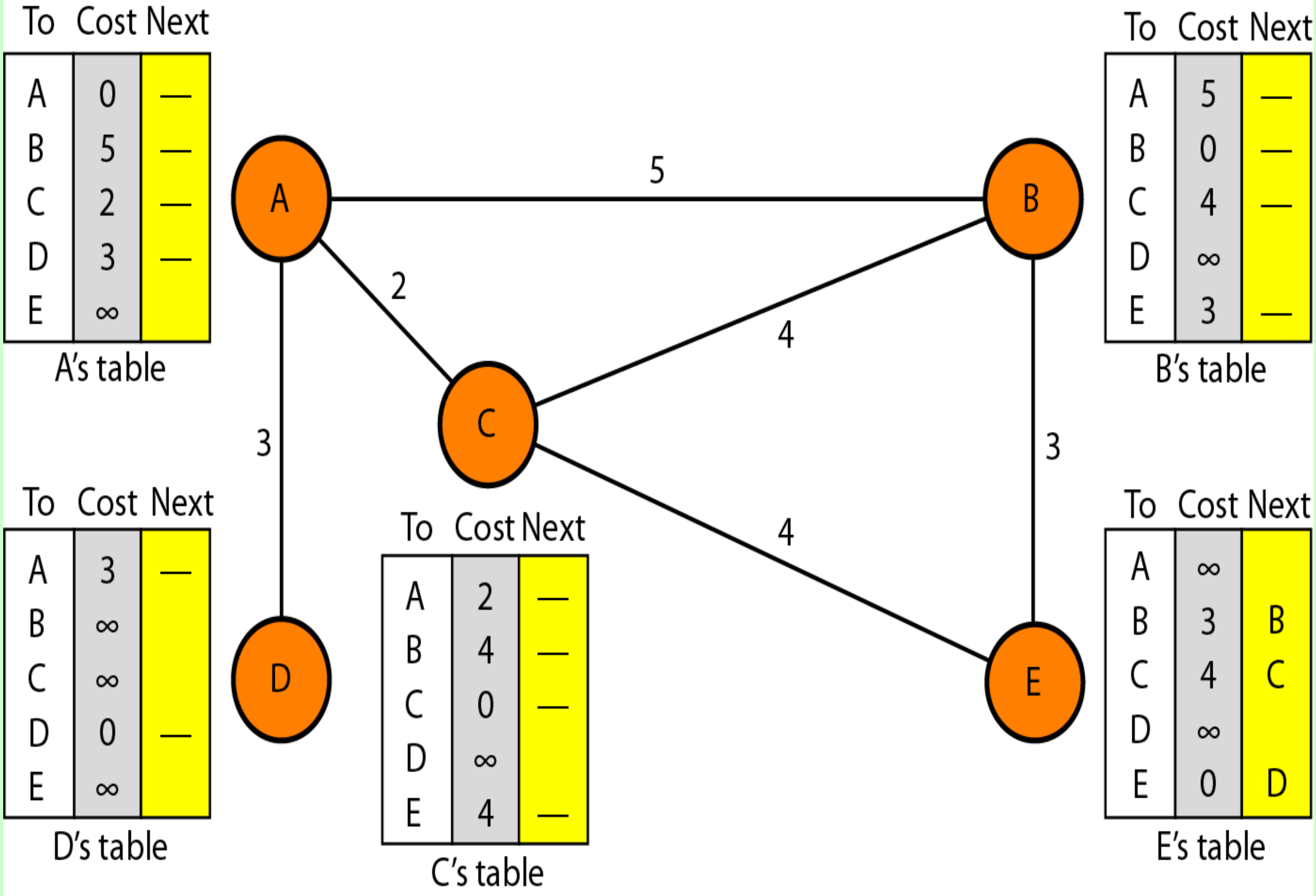
Mainly 3 things in this

Initialization
Sharing
Updating

1)Initialization

Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.

Below fig shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's table

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

E's table

Initialization of tables in distance vector routing

2)Sharing

The whole idea of distance vector routing is the sharing of information between neighbors.

Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C,
node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

NOTE:- In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change

3)Updating

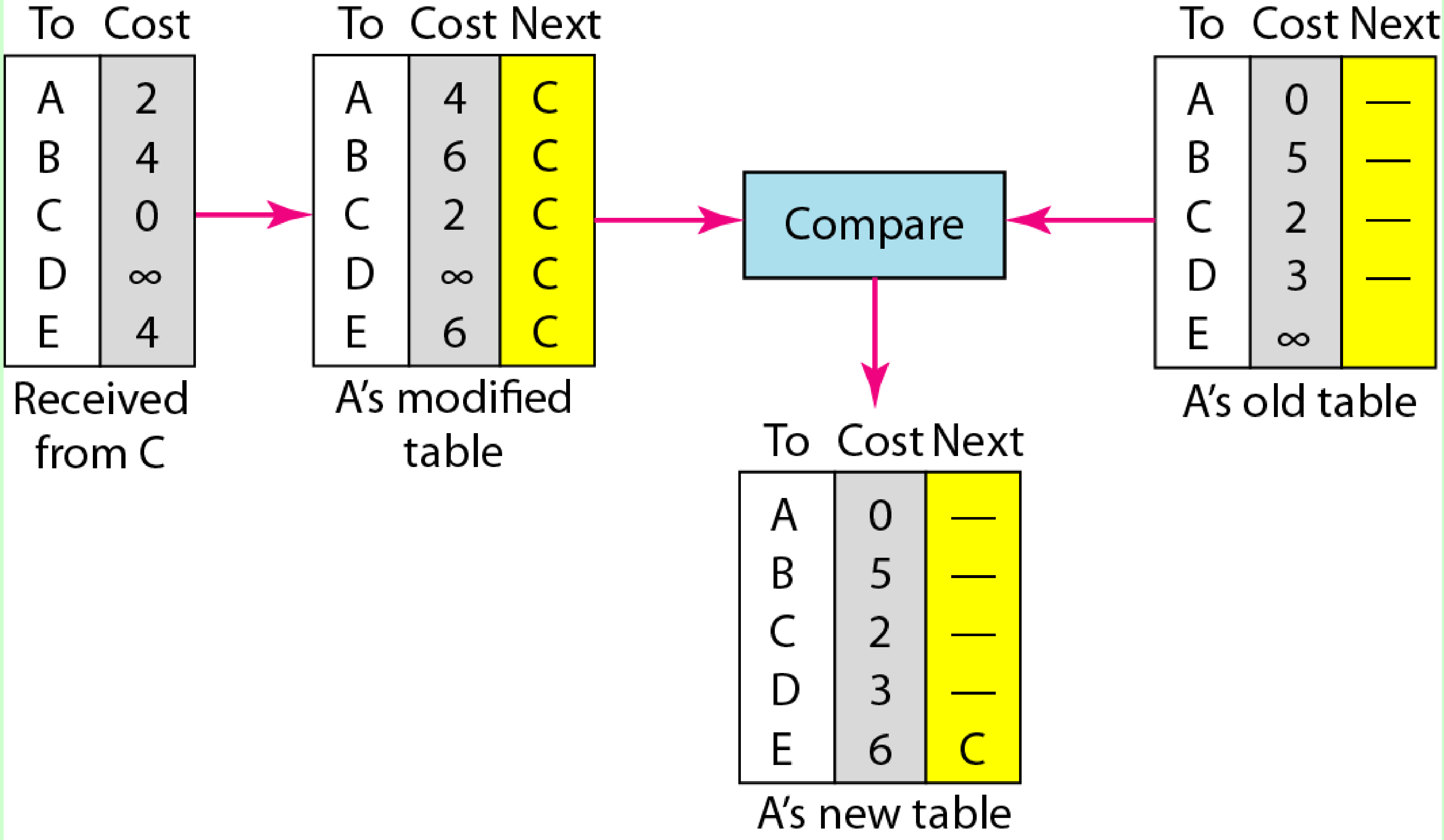
When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. (x+y)
2. If the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b. If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.

Updating in distance vector routing



To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's modified table

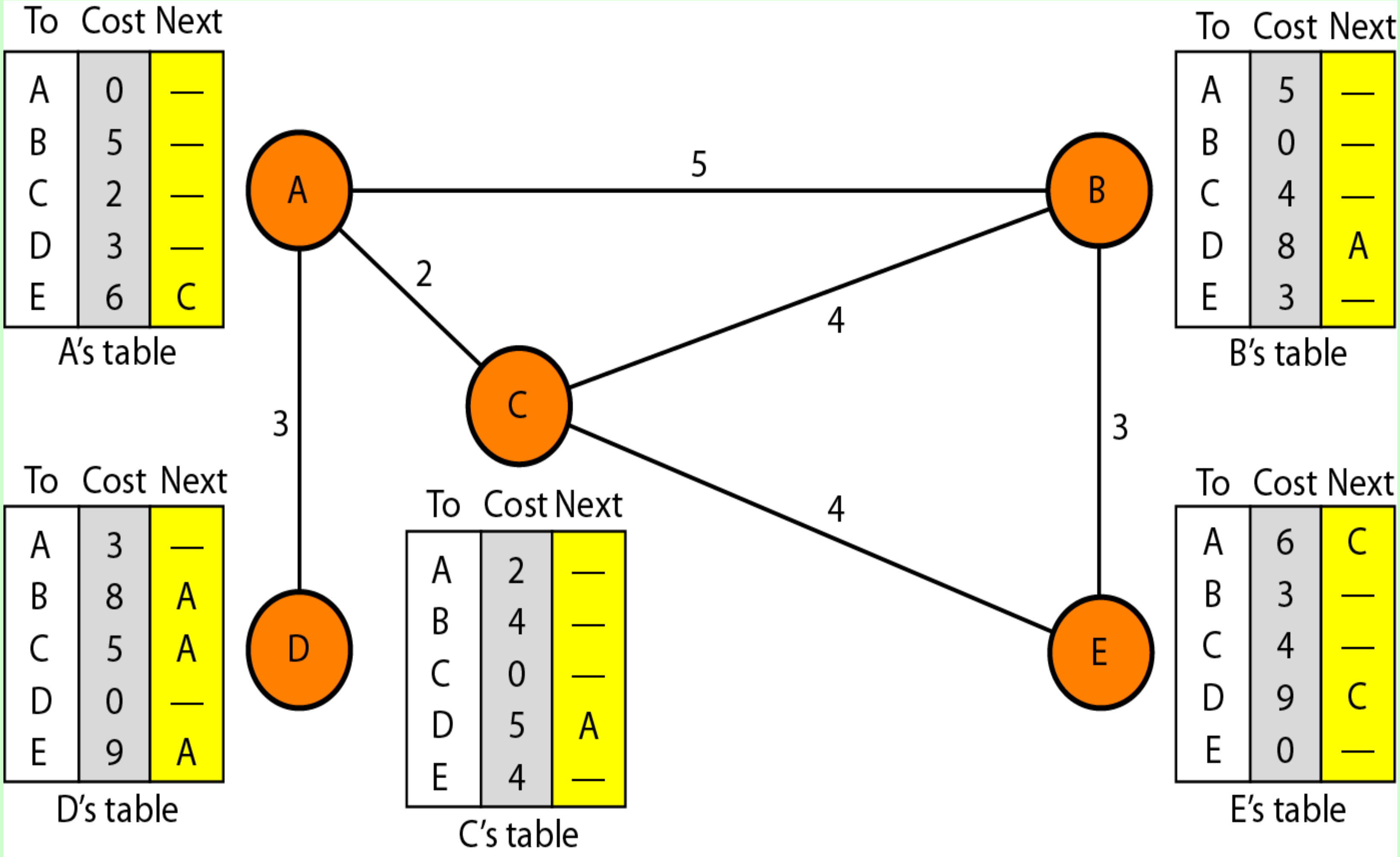
To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's old table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's new table

Final Diagram



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

D's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

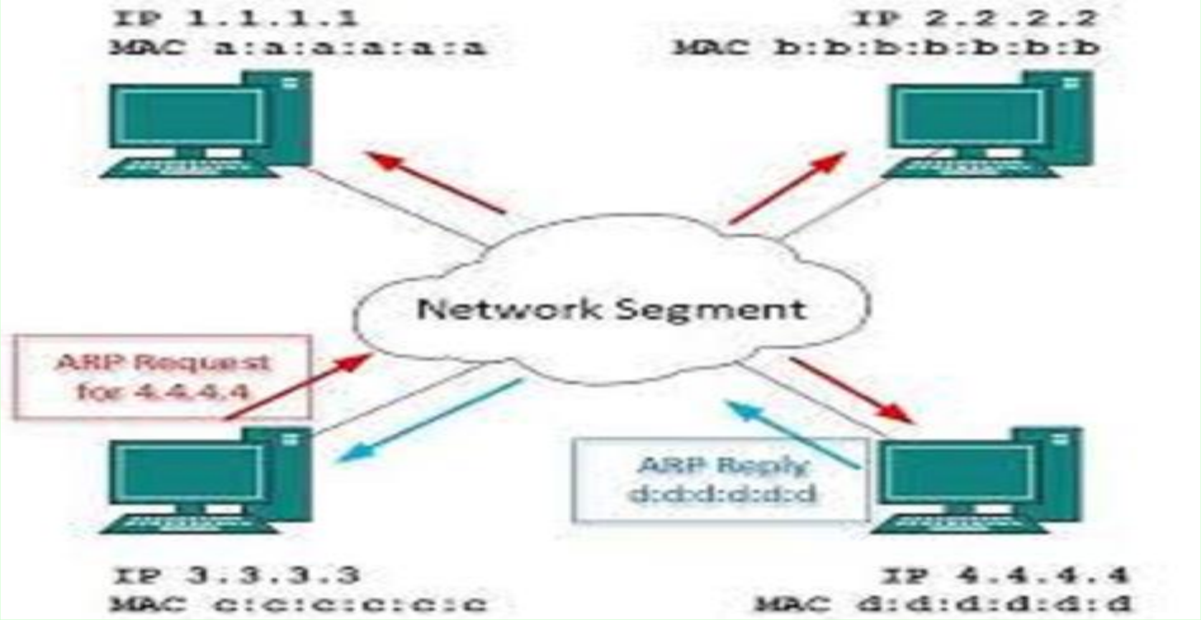
E's table

Protocol:-

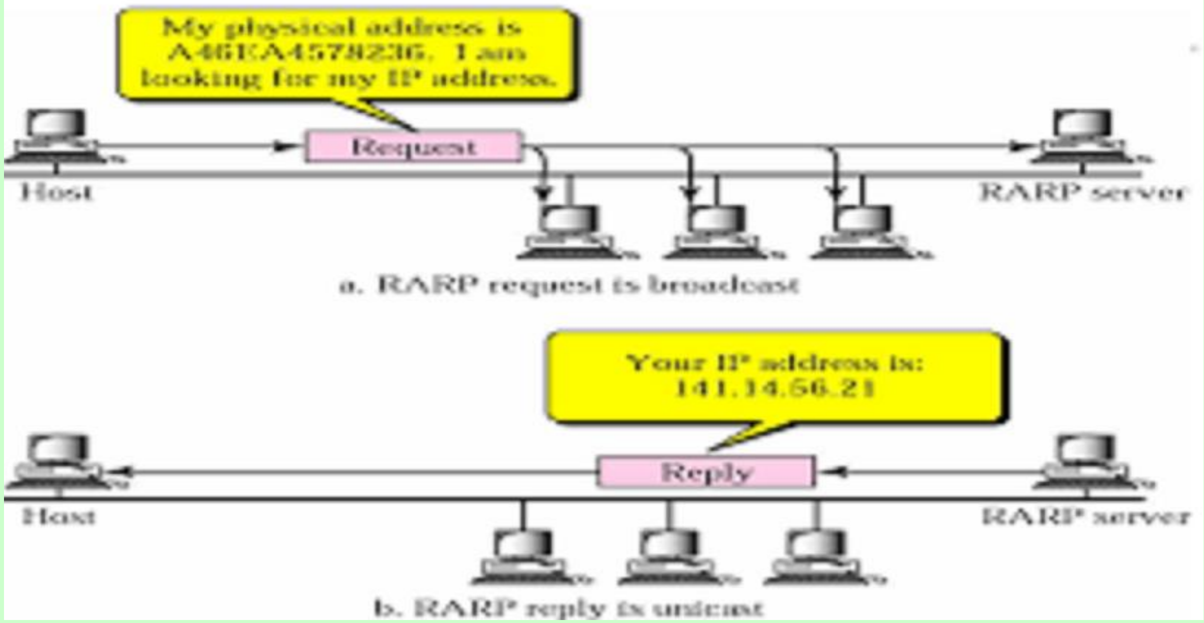
ARP,RARP, ICMP, IGMP

The **Address Resolution Protocol** (**ARP**) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. This mapping is a critical function in the Internet protocol suite.

The Reverse Address Resolution Protocol (RARP) is an obsolete computer networking protocol used by a client computer to request its Internet Protocol (IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address.



RARP Operation

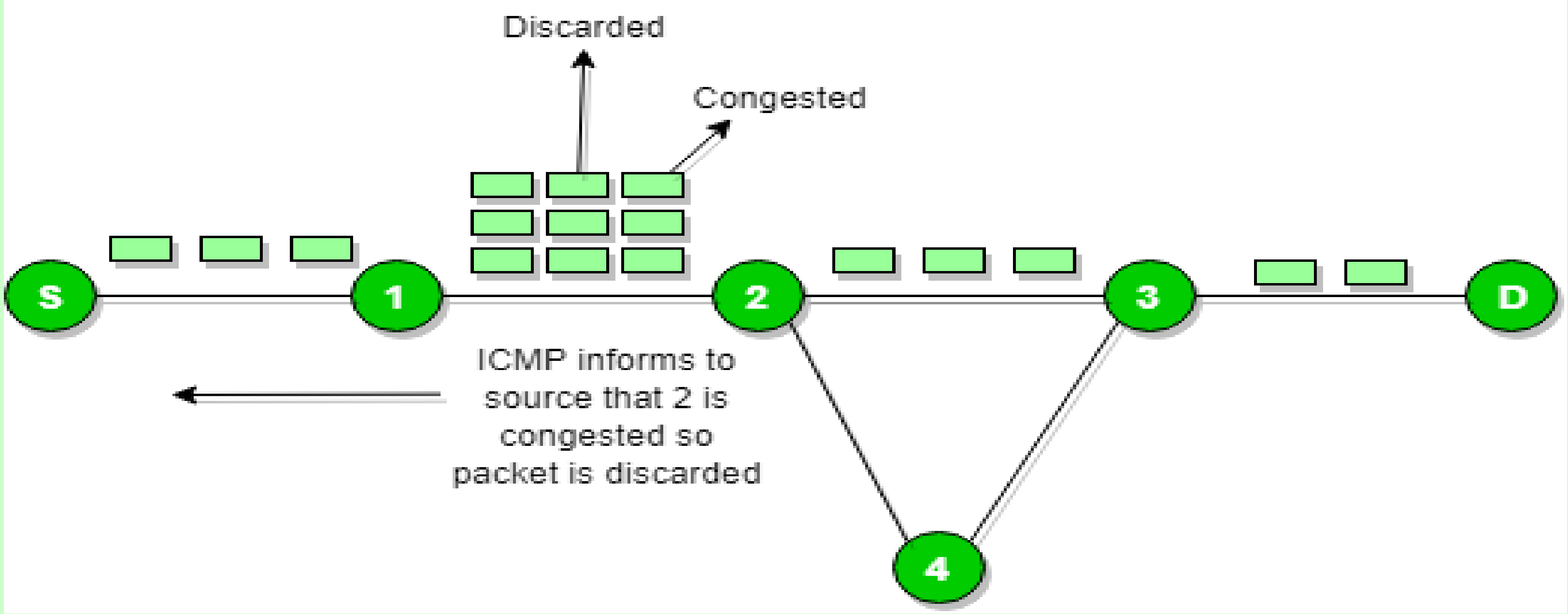

My physical address is A46EA4578236. I am looking for my IP address.
Request
Host
RARP server
Your IP address is: 141.14.56.21
Reply
Host
RARP server

a. RARP request is broadcast

b. RARP reply is unicast

ICMP(Internet Control Message Protocol):-

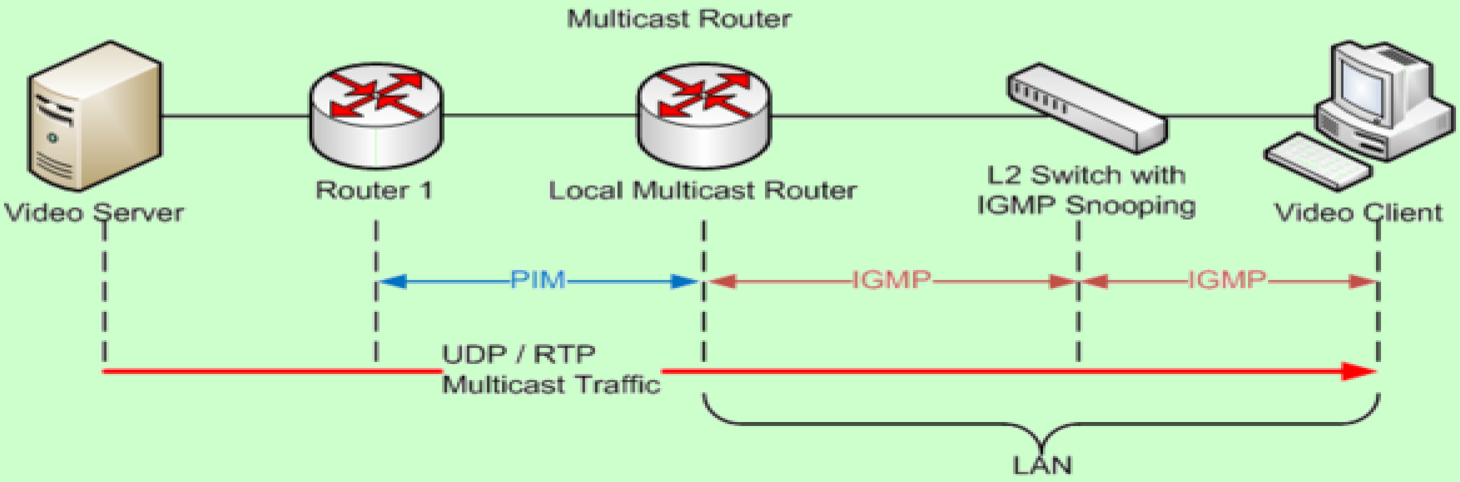
ICMP is a transport level **protocol** within TCP/IP which communicates information about **network** connectivity issues back to the source of the compromised transmission. It sends control messages such as destination **network** unreachable, source route failed.


Discarded
Congested
S
1
2
3
D
4
ICMP informs to source that 2 is congested so packet is discarded

IGMP(Internet Group Management Protocol):-

The Internet Group Management Protocol (IGMP) is a communications protocol used by hosts and adjacent routers on IPv4 networks to establish multicast group memberships. IGMP is an integral part of IP multicast and allows the network to direct multicast transmissions only to hosts that have requested them.

IGMP can be used for one-to-many networking applications such as online streaming video and gaming, and allows more efficient use of resources when supporting these types of applications.



Congestion :

It is an important issue in a packet-switched network.

It may occur, if the load on the network – the number of packets sent to the network – is greater than the capacity of the network – the number of packets a network can handle.

Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

What are the reasons for the congestions in a network ?

Congestions happens in any system that involves waiting.

For eg, congestion happens on a freeway because any abnormality in the flow, such as an accident during rush hour, creates blockage.

Congestions in a network or internetwork occurs because routers and switches have queues - buffers that hold the packets before and after processing .

A router, for example, has an input queue and an output queue for each interface.

When a packet arrives at the incoming interface, it undergoes three steps before departing.

1. The packet is put at the end of the input queue while waiting to be checked.

2. The processing module of the router removes the packet from the input queue once it reaches the front of the queue and uses its routing table and the destination address to find the route.

3. The packet is put in the appropriate output queue and waits its turn to be sent.

It needs to be aware of two issues : First, if the rate of packet arrival is higher than the packet processing rate, the input queues become longer and longer.

Second, if the packet departure rate is less than the packet processing rate, the output queues become longer and longer.

Network Performance :

Congestion control involves two factors that measure the performance of a network delay and throughput (throughput in a network is the number of packets passing through the network in a unit of time).

Congestion Control :

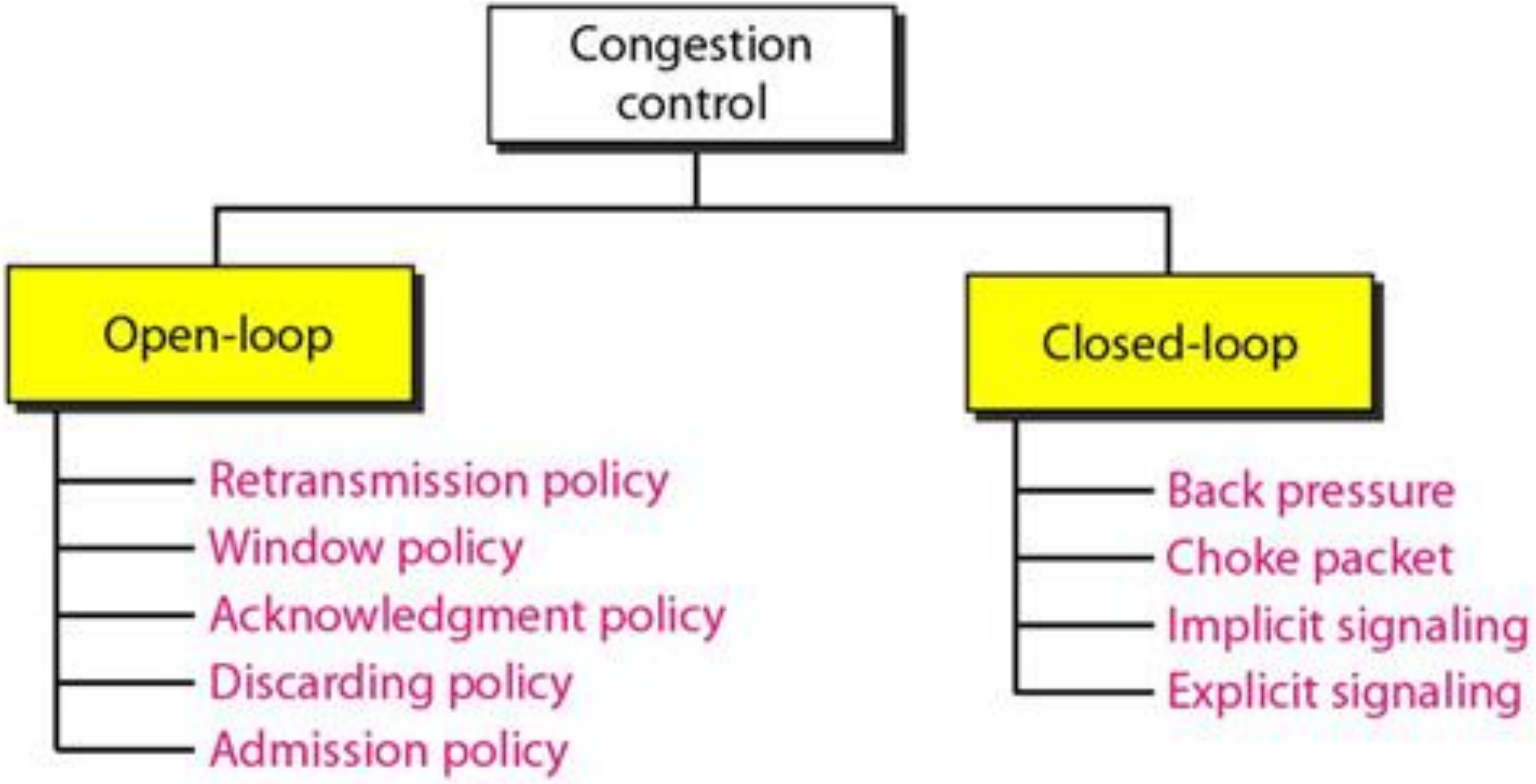
It refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.

It can be classified under two broad categories :

Open-loop congestion control (prevention) and

Closed-loop congestion control (removal).

Congestion Control Categories



Congestion control

Open-loop

- Retransmission policy
- Window policy
- Acknowledgment policy
- Discarding policy
- Admission policy

Closed-loop

- Back pressure
- Choke packet
- Implicit signaling
- Explicit signaling

Open-loop congestion control (prevention) :

These policies are applied to prevent congestion before it happens. In these, congestions is handled by either the sourced or the destination.

1. **Retransmission policy** : A good retransmission policy can prevent congestion.

2. **Window Policy** : The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control.

3. **Acknowledgment Policy** : If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.

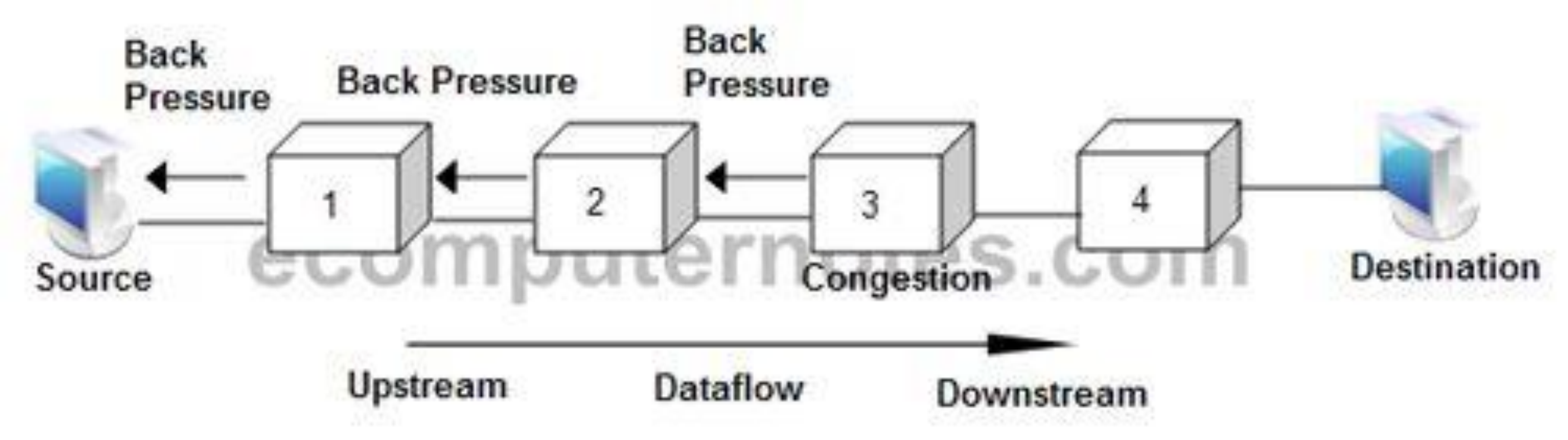
4. **Discarding Policy** : A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. Eg. In audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented.

5. **Admission Policy** : An admission policy which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network.

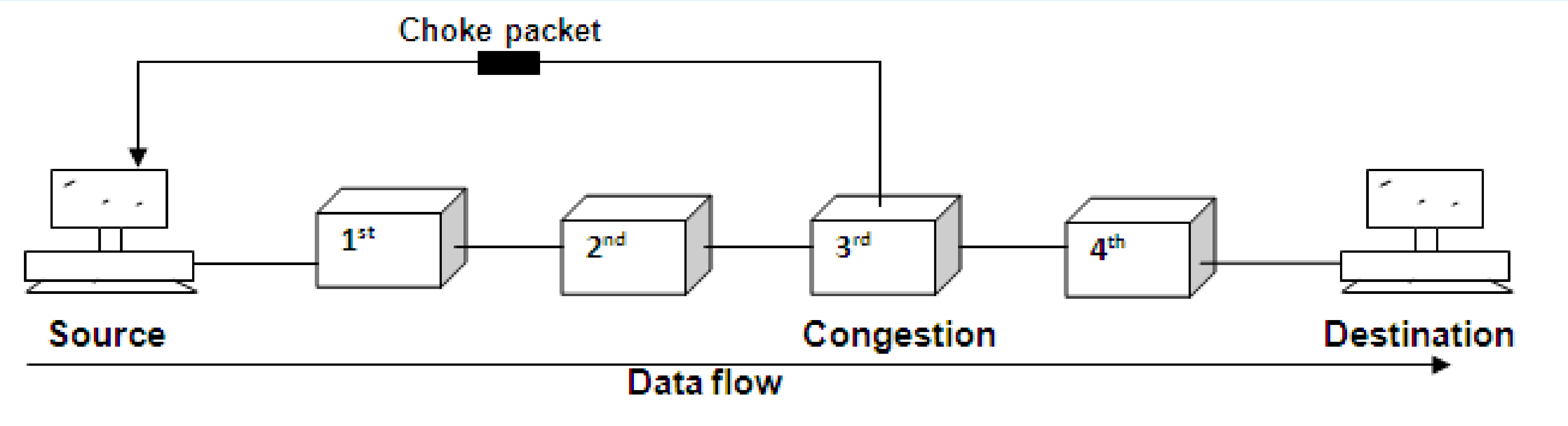
Closed-loop congestion control (removal).

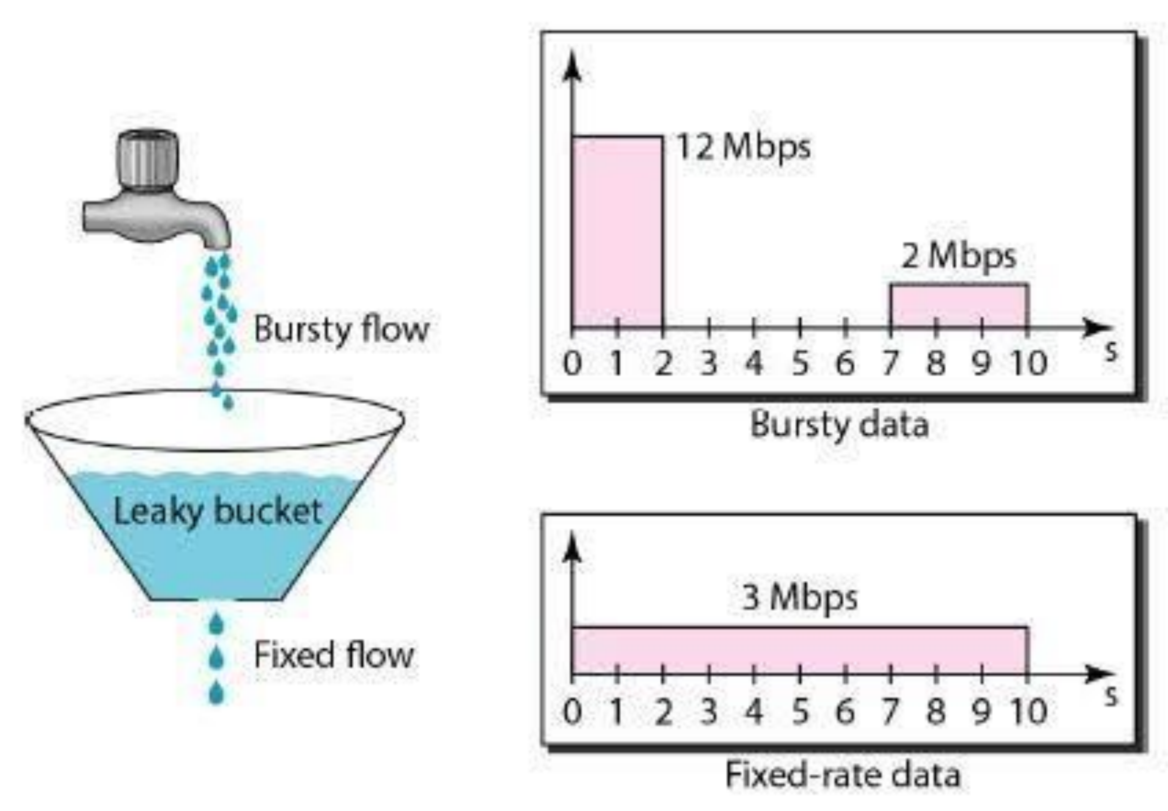
Closed loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. They are :

1. **Back Pressure** : When a router is congested, it can inform the previous upstream router to reduce the rate of outgoing packets. The action can be recursive all the way to the router before the source. This mechanism is called back pressure. { [The further routers may be congested (it may have large data to handle), so it informs the previous routers to slow down the output flow of data. It continues up to the source machine. This process is the back pressure]. So the pressure on the last router is moved backward to the source to remove the congestion. It was implemented in first virtual-circuit network, X.25. It cannot be implemented in a datagram network because in this type of network a node (router) does not have the slightest knowledge of the upstream router. }



Backpressure Method


Choke packet
1st
2nd
3rd
4th
Source
Congestion
Destination
Data flow


Bursty flow
Leaky bucket
Fixed flow
12 Mbps
2 Mbps
0 1 2 3 4 5 6 7 8 9 10
s
Bursty data
3 Mbps
0 1 2 3 4 5 6 7 8 9 10
s
Fixed-rate data

2. **Choke Point** : It is a packet sent by a router to the source to inform it of congestion. This type of control is similar to ICMP’s source quench packet.

3. **Implicit Signaling** : The source can detect an implicit signal concerning congestion and slow down its sending rate. [the delay in receiving an acknowledgment can be a signal that the network is congested.].

4. **Explicit signaling** : The routers that experience congestion can send an explicit signal, the setting of a bit in a packet, to inform the sender or the receiver of congestion.

5. **Backward Signaling** : the bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

6. **Forward Signaling** : This bit be set in the packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgment, to alleviate (reduce / ease / relive) the congestion.

QUALITY OF SERVICE [QoS] : It is internetworking issue that has been discussed more than defined. It can be defined as something a flow seeks to attain.

Flow Characteristics : There are four types of characteristics attributed to aflow : ***reliability, delay, jitter, and bandwidth***.

Reliability : It is a characteristics that a flow needs. Lack of reliability means losing a packet or acknowledgement, which entails retransmission. However the sensitivity of application programs to reliability is not the same.

Delay : Source to Destination delay is another flow characteristic. Again application can tolerate delay in different degrees. In this case telephony audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer of e-mail is less important.

Jitter: It is the variation in delay for packets belonging to the same flow. Eg. For application like audio and video, if four packets are sent at times 0, 1, 2, 3 and they arrive at 20, 21, 22, 23, all have the same delay time of 20 unites of time. On the other hand, if the above four packets arrive at 21, 23, 21, 28 units of time, they have different delays of 21, 22, 19, 25. The first case is completely acceptable; the second case is not. For this application it doesn’t matter, whether the delay is short or long, but with same delay for all packets. So the second one is not acceptable.

Jitter is defined as the variation in the packet delay. High jitter means the difference between delays is large; low jitter means the variation is small.

Bandwidth :   Different applications need different bandwidth. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an email may not reach even a million.

Flow Classes : Based on the flow characteristics, we can classify flows into groups, with each group having similar levels of characteristics. ATM (Asynchronous Transfer Mode) Forum defines four services : CBR (constant-bit-rate), VBR (variable-bit-rate), ABR (available-bit-rate), UBR (unspecified-bit-rate).

TECHNIQUES TO IMPROVE QoS : Four common methods .

1. **Scheduling** : Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Some of the scheduling techniques are given below:

1. **FIFO Queuing** : In First-in, First-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.

2. **Priority Queuing**: Packets are assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last. Note that the system does not stop serving a queue until it is empty. It can provide better QoS than FIFO queuing.

3. **Weighted Fair Queuing** : In this technique, the packets are still assigned to different classes are admitted to different queues. The queues are weighted based on the priority of the queues.

2. **Traffic Shaping** : It’s a mechanism to control the amount and the rate of the traffic sent to the network. **LEAKY BUCKET** : The input rate can vary, but the output rate remains constant.

3. **Resource Reservation** : A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The QoS is improved if these resources are reserved beforehand.

4. **Admission Control** : It refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specification. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

Traffic Shaping

1. Another method of congestion control is to “shape” the traffic before it enters the network.

2. Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.

3. At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).

Two traffic shaping algorithms are:

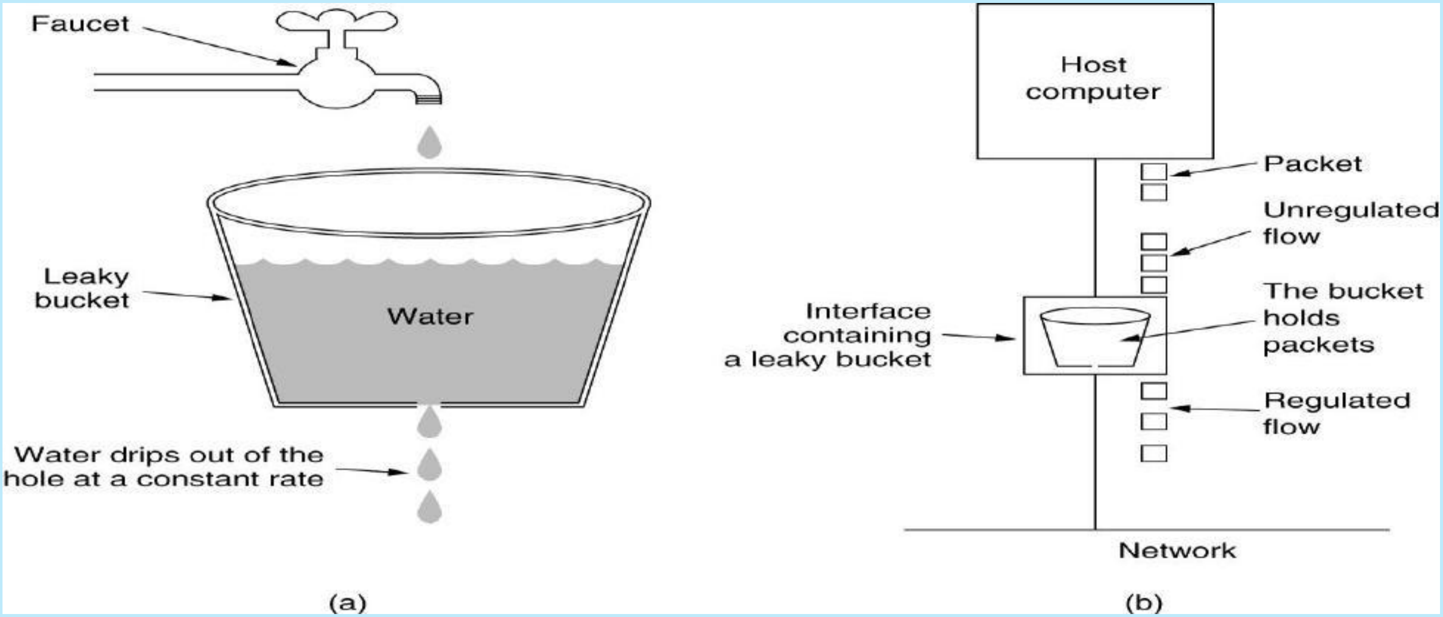
Leaky Bucket

Token Bucket

The **Leaky Bucket Algorithm**

used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are

discarded.



(a) A leaky bucket with water.

(b) A leaky bucket with packets.

1. The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.

2. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.

3. When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick.

E.g.

1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick

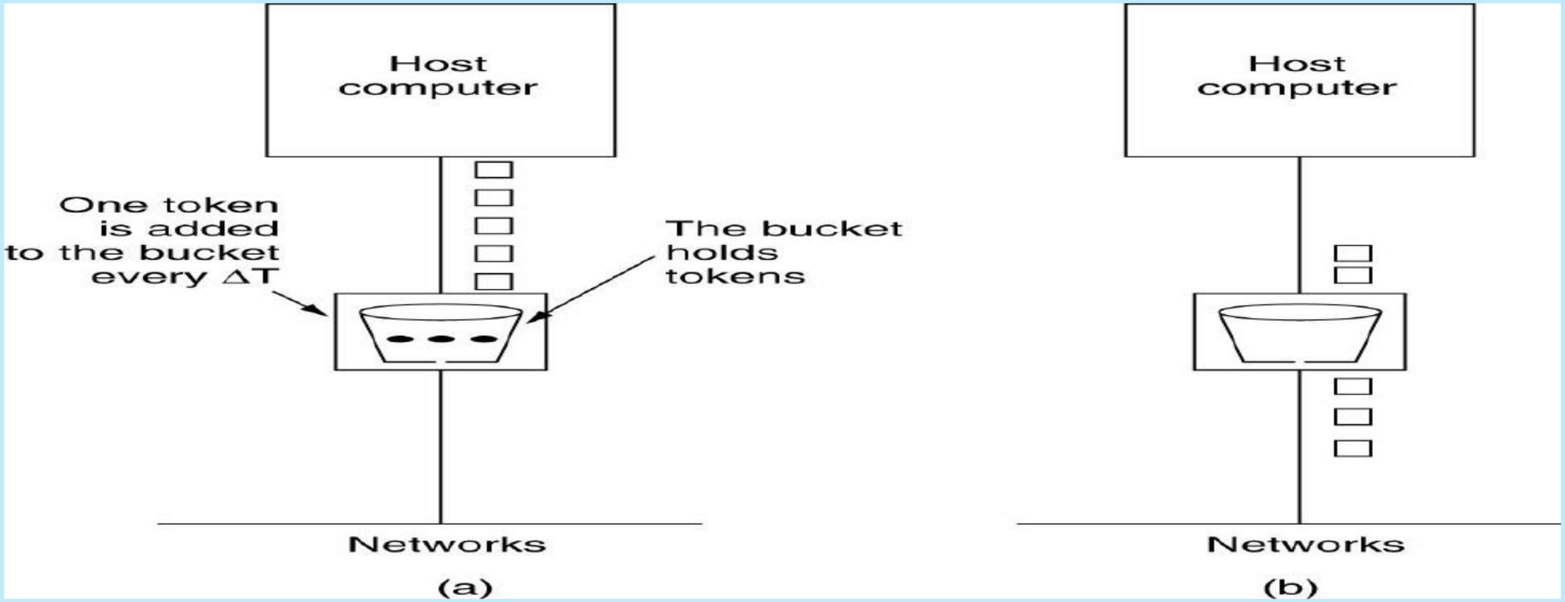
Token Bucket Algorithm

In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.

2. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.

3. Tokens are generated by a clock at the rate of one token every △ t sec.

4. Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



(a) Before.

(b) After.

Leaky Bucket vs. Token Bucket

1.LB discards packets; TB does not. TB discards tokens.
2.With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3. LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
4. TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.