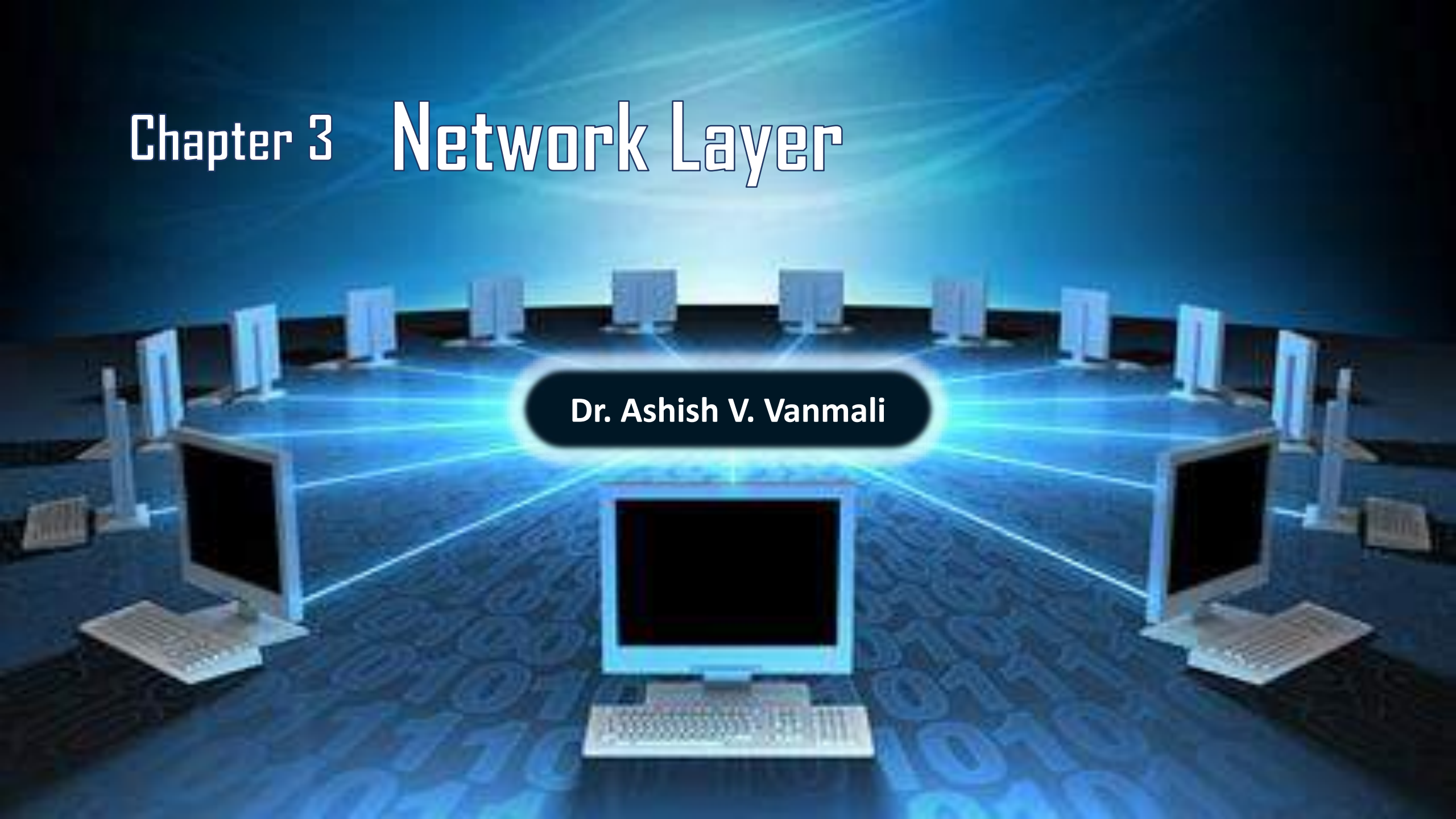
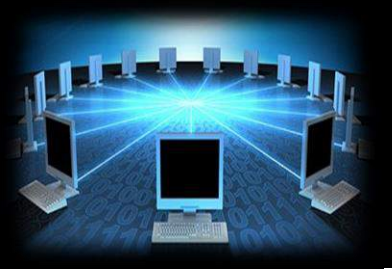
Chapter 3 Network Layer

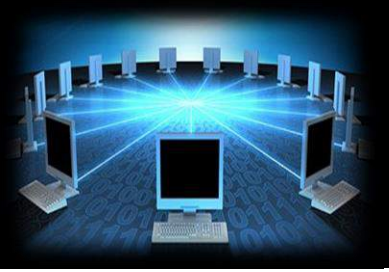


Dr. Ashish V. Vanmali



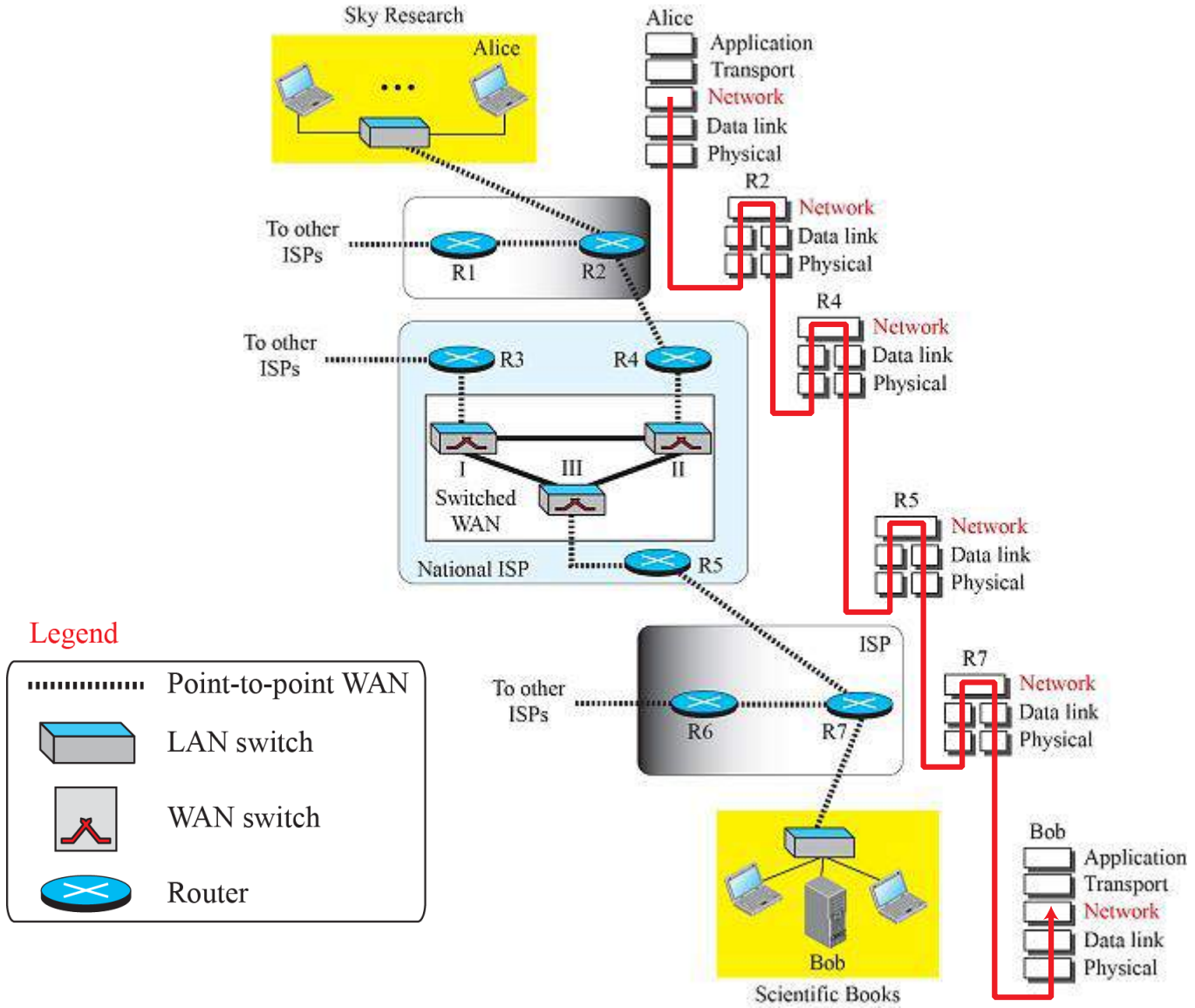
Outline of the Chapter

- Network Layer Services
- IPv4 Addressing
- Subnetting & Supernetting
- Classless Addressing
- IPv4 Protocol, DHCP Protocol, NAT Protocol
- Routing Algorithms (Distance Vector Routing, Link state routing, Path Vector Routing)
- Routing Protocols (RIP, OSPF, BGP)
- IPv6 (Addressing & Protocol)

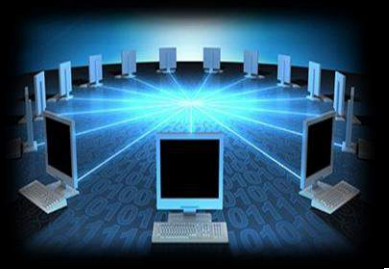


Network Layer Services

Communication at the Network Layer

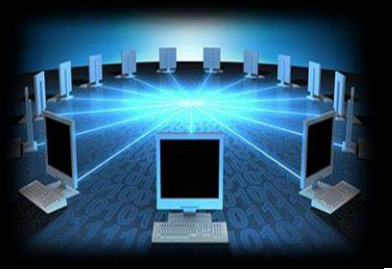


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Network Layer Services

- **Packetizing**
  - Encapsulating the payload (data received from upper layer) in a network layer packet at the source and decapsulating the payload at the destination.
- **Routing**
  - Routing is applying strategies and running some routing protocols to create the decision-making or routing tables for each router.
- **Forwarding**
  - Forwarding is the action applied by each router when packet arrives at one of its interface. The decision-making table used by router to perform this action is called forwarding table or routing table.



Network Layer Services

Error Control

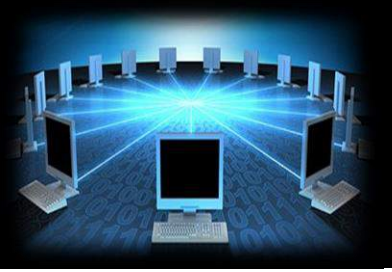
- Network layer does not provide error control mechanism.
- It has checksum field to control any corruption in the header, but not for the whole datagram.

Flow Control

- Network layer does not provide flow control.

Congestion Control

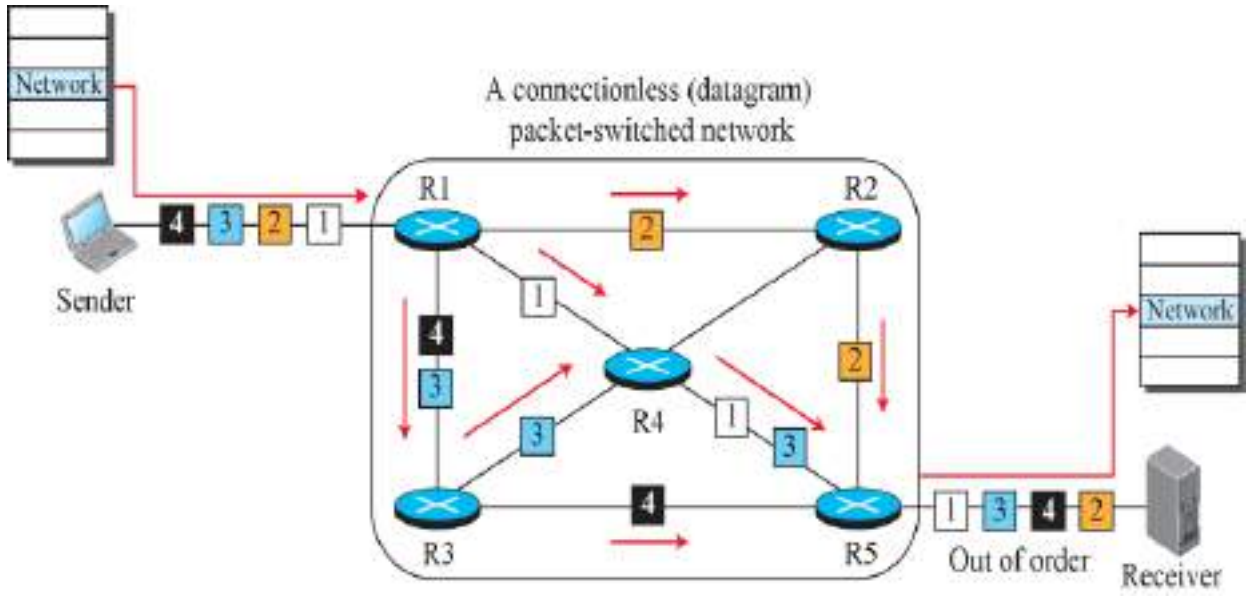
- Network layer provide congestion control.



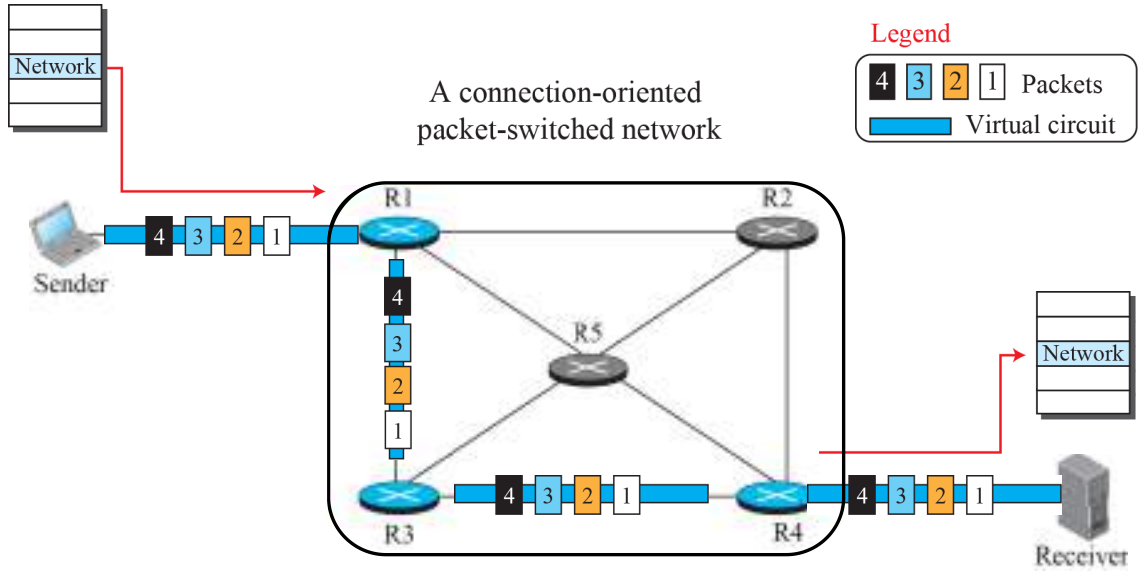
Packet Switching

- A router, in fact, is a switch that creates a connection between an input port and an output port (or a set of output ports).
- Packet switching can use either Datagram approach or Virtual-circuit approach.

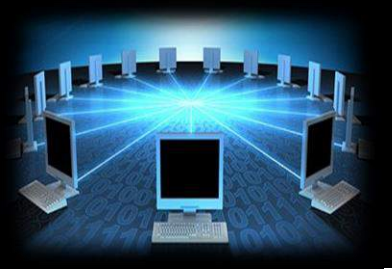
Datagram Packet Switching



Virtual-Circuit Packet Switching



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Network-Layer Performance

- The performance of a network can be measured in terms of delay, throughput, and packet loss.

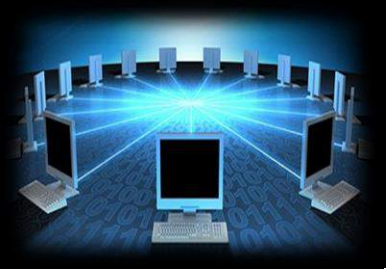
❑ Delay

- The total delay comprises of transmission delay, propagation delay, processing delay and queueing delay.

1. Transmission Delay

- It is the time taken by the transmitter to put the packet on the line.
- If the first bit of the packet is put on the line at time t_1 and last bit at time t_2, transmission delay is $(t_2 - t_1)$.

$$Delay_{tr} = \frac{Packet\ Length}{Transmission\ Rate}$$



Network-Layer Performance

Delay

2. Propagation Delay

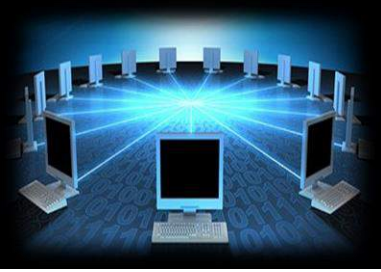
- It is the time taken the packet to travel from the sending node to the receiving node over the transmission media.

$$Delay_{pg} = \frac{Distance}{Propagation\ speed}$$

3. Processing Delay

- It is the time required for a router or a destination host to receive a packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port or to the upper layer protocol.

$$Delay_{pr} = Time\ required\ to\ process\ the\ packet\ in\ the\ router$$



Network-Layer Performance

❑ Delay

4. Queueing Delay

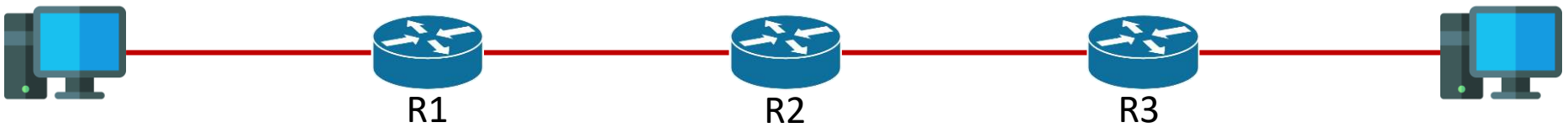
- It is the time a packet waits in the input queue and output queue of a router.

$$Delay_{qu} = Time\ a\ packet\ waits\ in\ the\ input\ and\ output\ queues\ in\ a\ router$$

Total Delay

- If there are n number of routers in the path, then

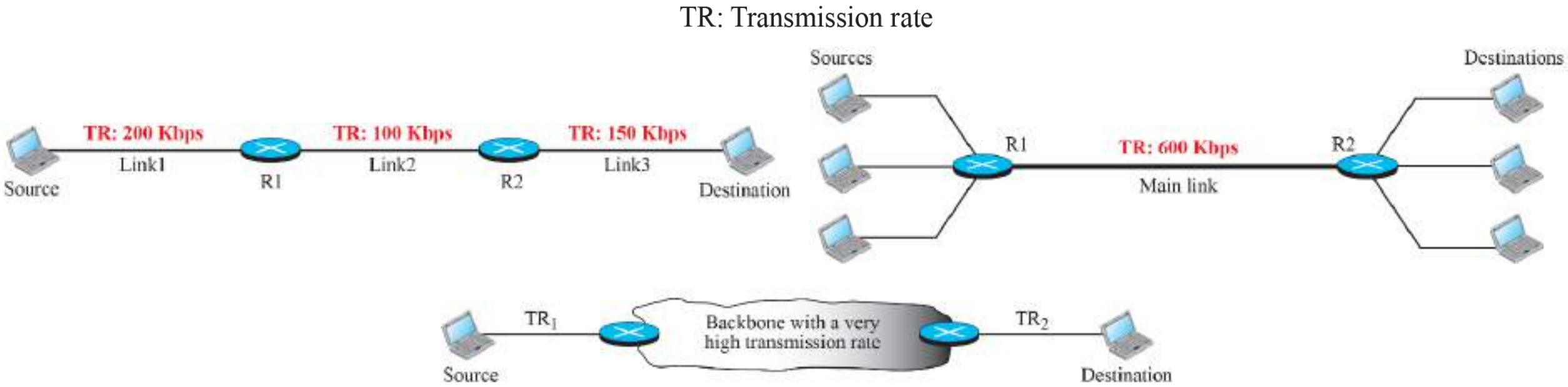
$$Total\ Delay = (n+1)\left[Delay_{tr} + Delay_{pg} + Delay_{pr}\right] + (n)\left[Delay_{qu}\right]$$



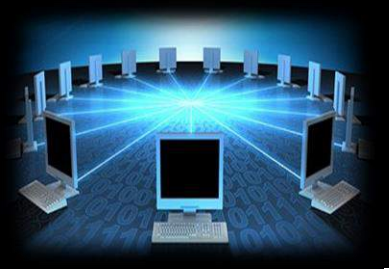
Network-Layer Performance

Throughput

- Throughput at any point in a network is defined as the number of bits passing through the point in a second, which is actually, the transmission rate of the data at that point.



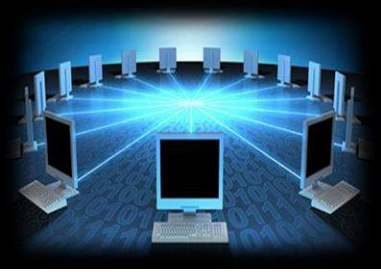
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



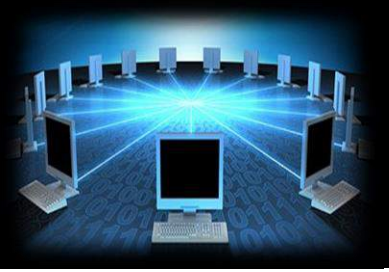
Network-Layer Performance

Packet Loss

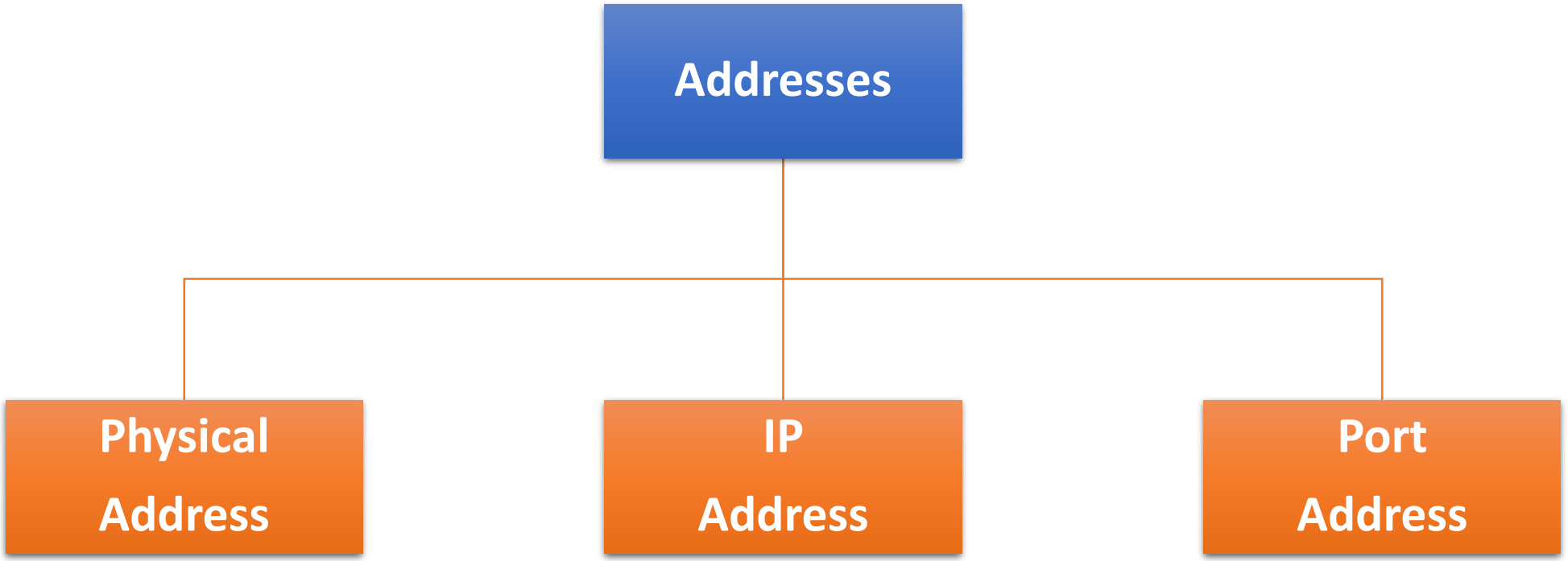
- Performance of communication is severely affected by the number of packets lost during transmission.
- If the input buffer of the router is full, some packets may be dropped results in loss of packets.
- The effect of packet loss on internet network layer is that the packet needs to be resent, which in turn may create overflow and cause more packet loss.



IPv4 Addressing

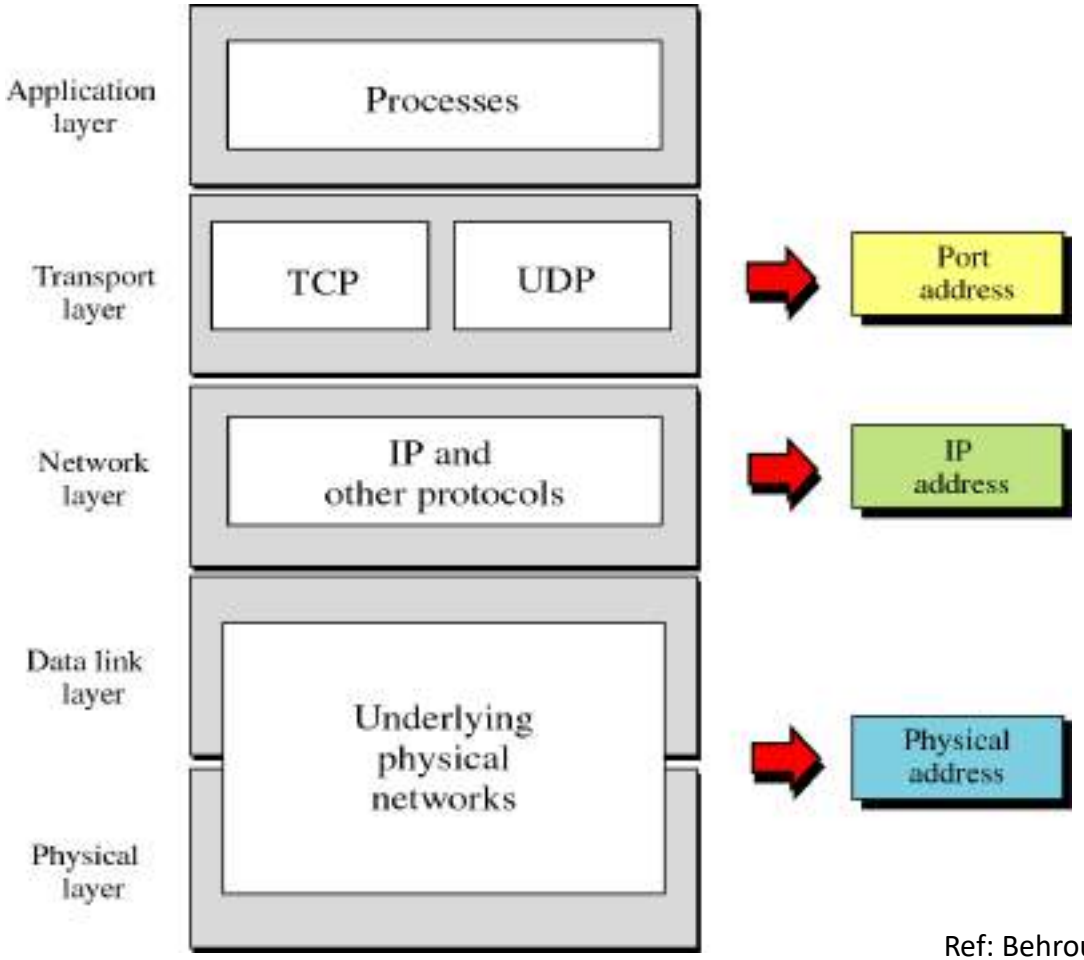


Addresses in TCP/IP

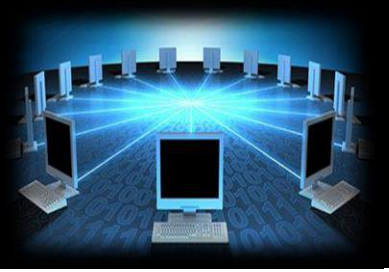


Addresses in TCP/IP

❑ Relationship of layers and addresses in TCP/IP

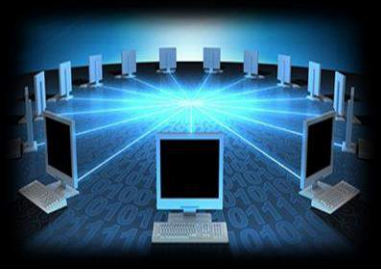


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



IPv4 Address

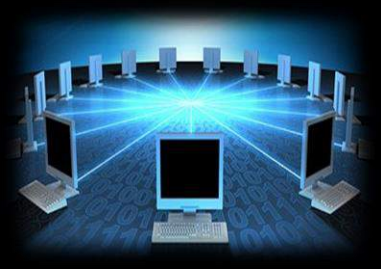
- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the **Internet address** or **IP address**.
- The IP addresses are 32 bits (4 bytes) in length. These addresses are referred to as **IPv4 (IP version 4) addresses** or simply **IP addresses**.
- IP address uniquely and universally defines the connection of a host or a router to the Internet.
- The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.
- The address space of IPv4 is 2^{32} or 4,294,967,296.



IPv4 Address

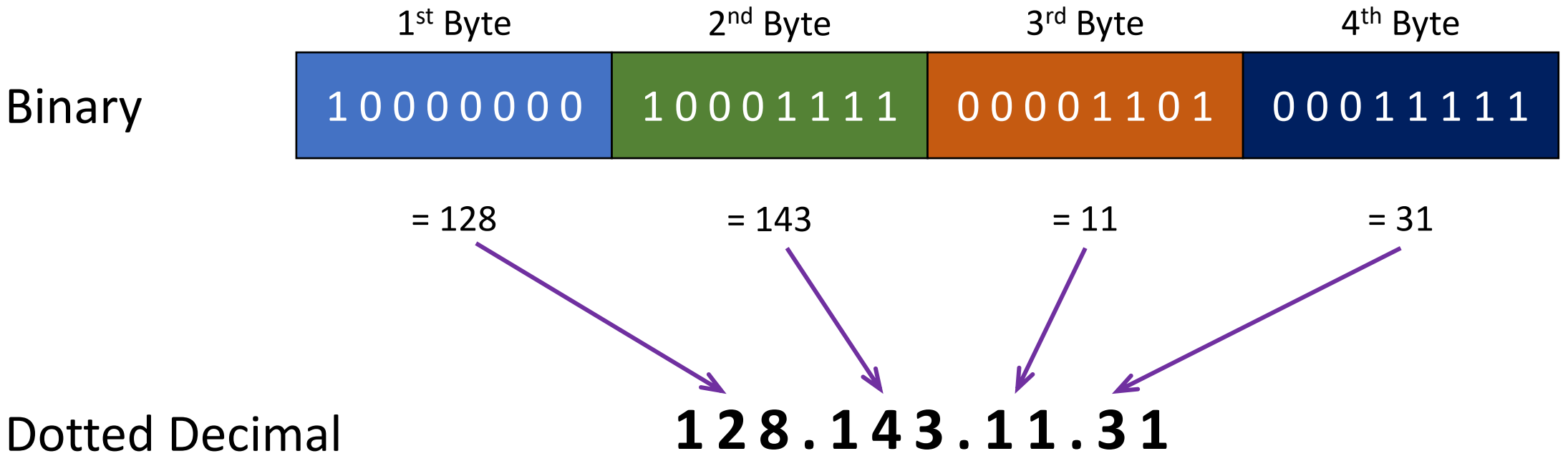
Notations

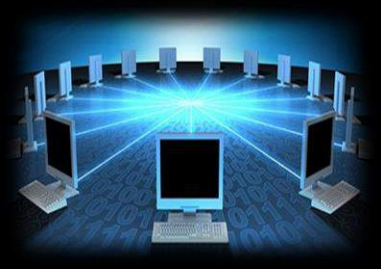
- There are two prevalent notations to show an IPv4 address: **binary notation** and **dotted decimal notation**.
- In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte.
- To make the IPv4 address more compact and easier to read, IP addresses are usually written in decimal form with a decimal point (dot) separating the bytes.
- Since, each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.



IPv4 Address

Notations

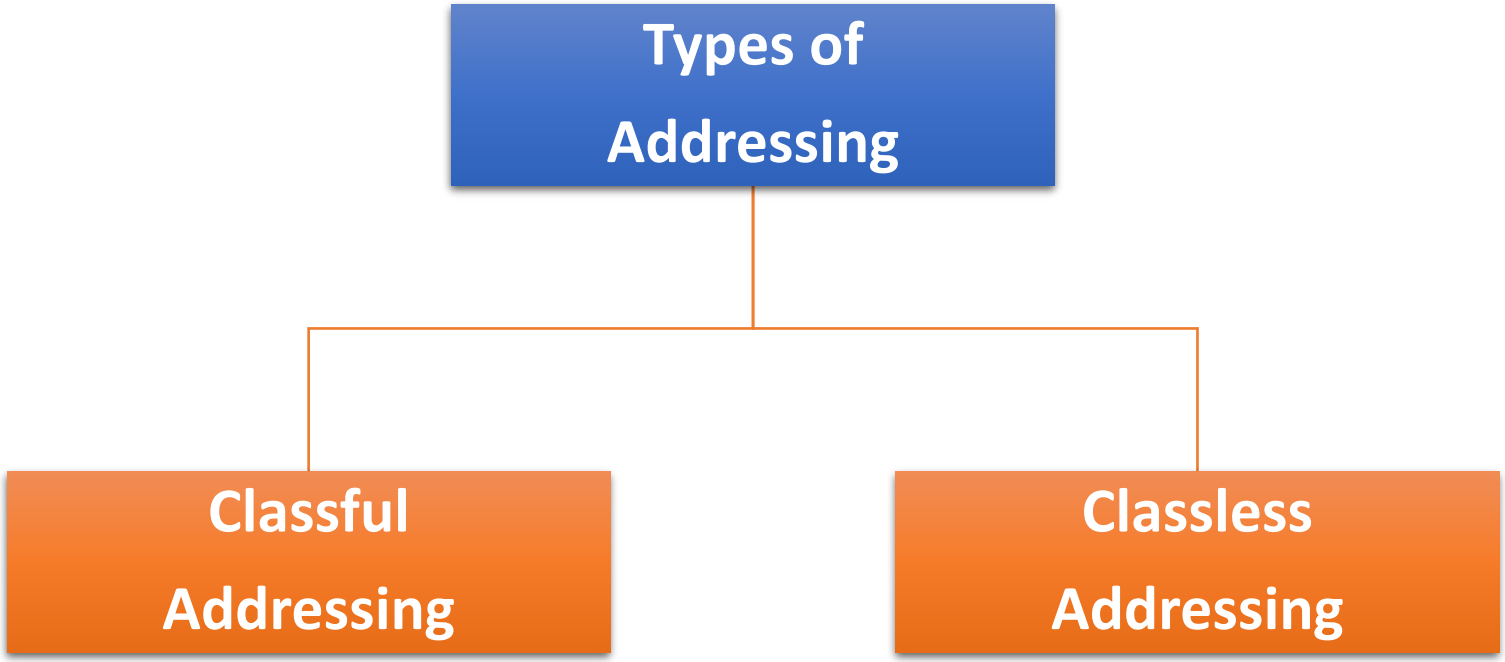


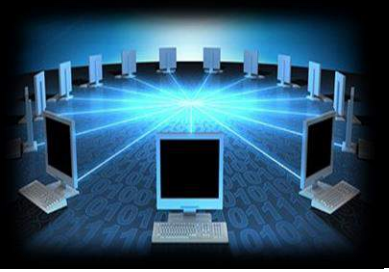


IPv4 Address

Types of Addressing

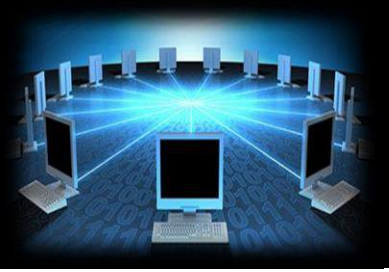
- IPv4 addressing, at its inception, used the concept of classes.





Classful Addressing

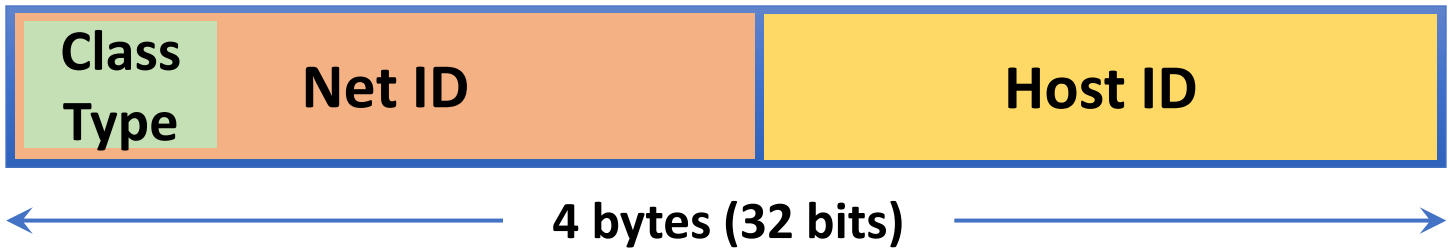
- In classful addressing, the address space is divided into five classes: A, B, C, D, and E.
- Each class occupies some part of the address space.
- We can find the class of an address when given the address in binary notation or dotted-decimal notation.
- If the address is given in binary notation, the first few bits can immediately tell us the class of the address.
- If the address is given in decimal-dotted notation, the first byte defines the class.



Classful Addressing

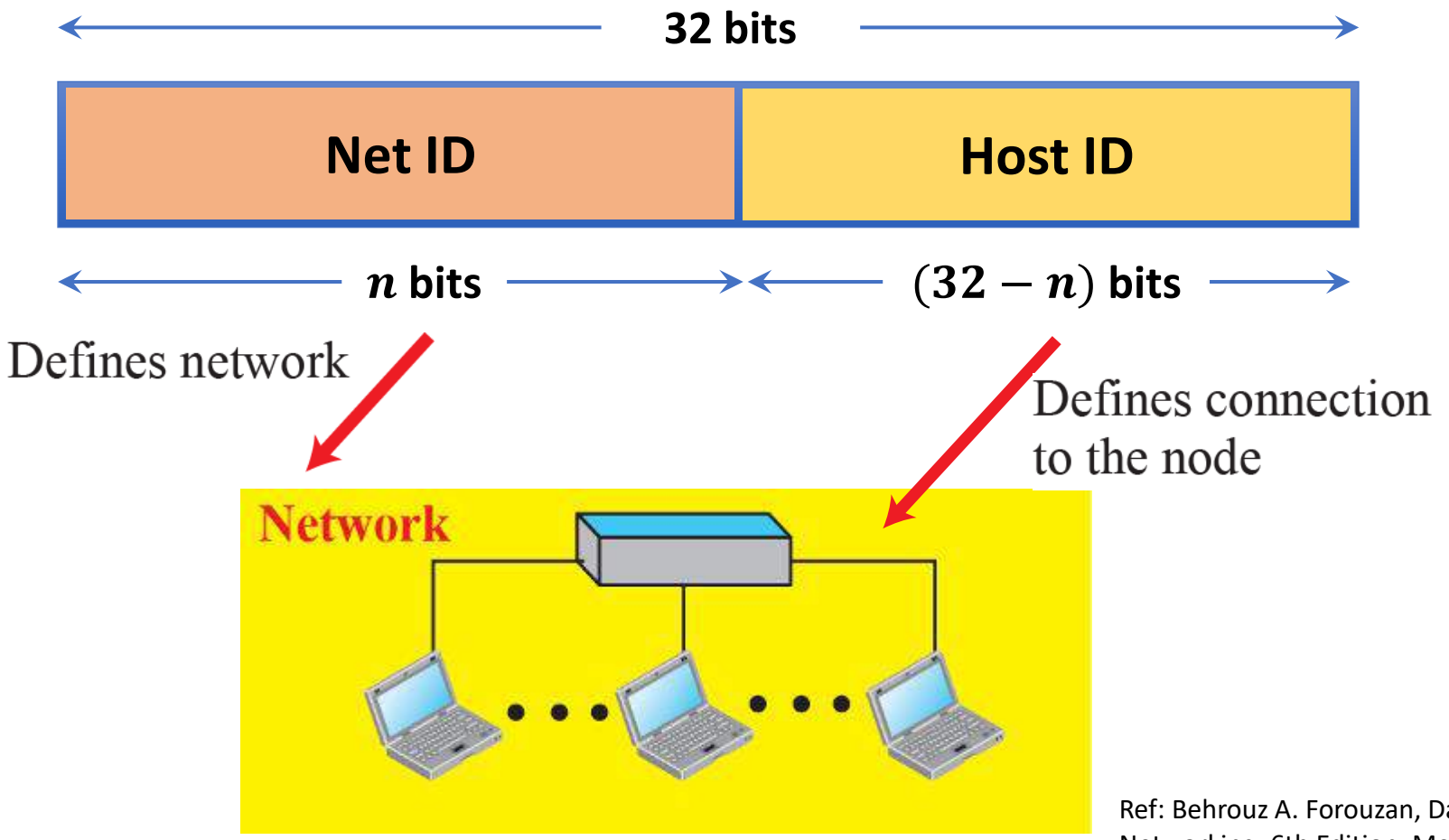
Fields of IP Address

- IP address structure has two level hierarchy:
  - Network ID (Net ID)
  - Host ID
- Net ID identifies **network** the host is connected to. All hosts connected to same network has same Net ID.
- Host ID identifies **network connection** to the host, rather than actual host.
- Router can forward packets based on Net ID only, thereby shorting size of routing table significantly.

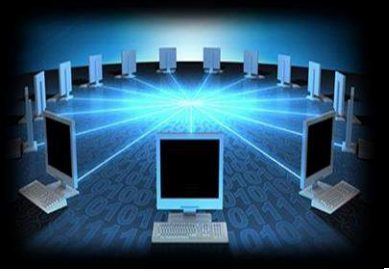


Classful Addressing

Fields of IP Address

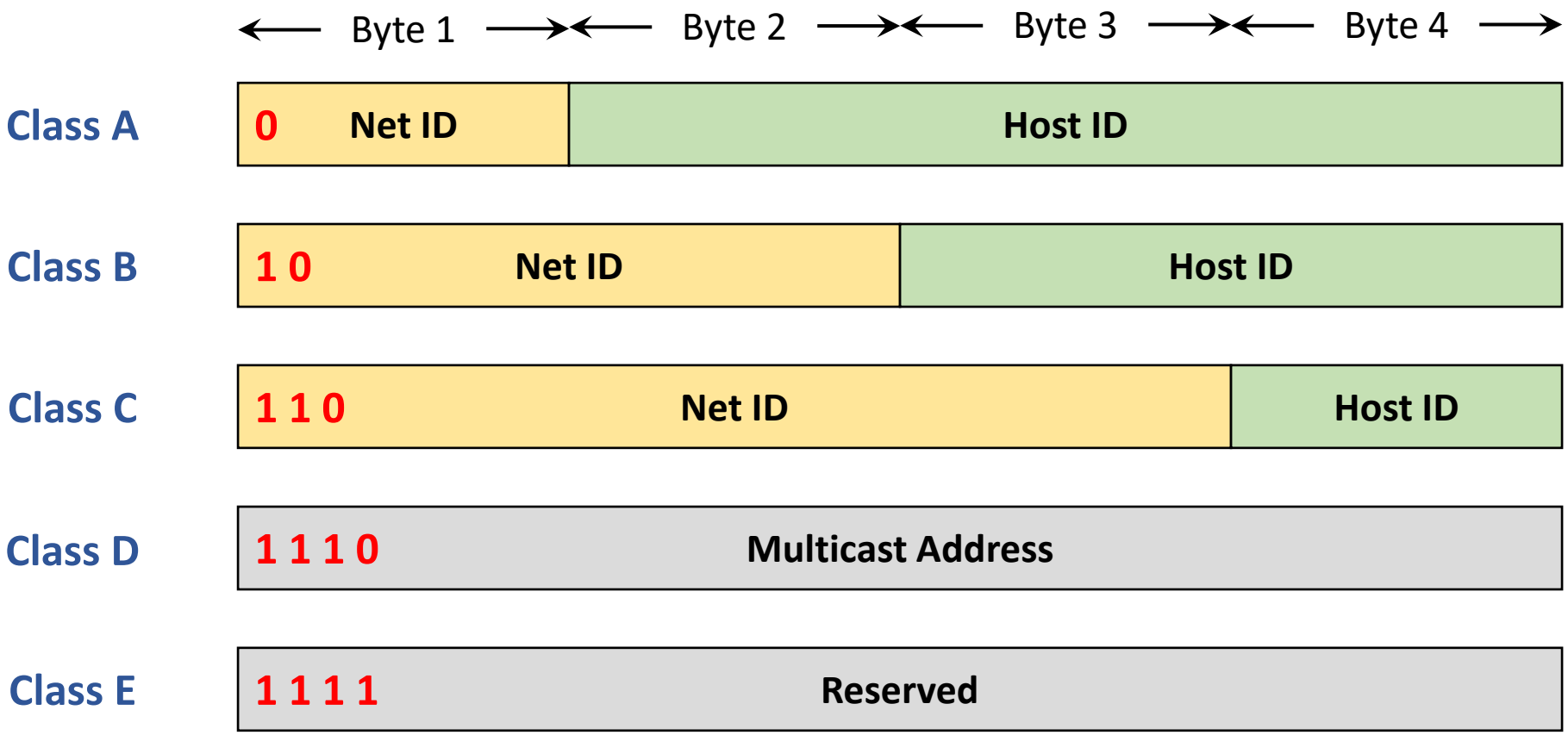


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



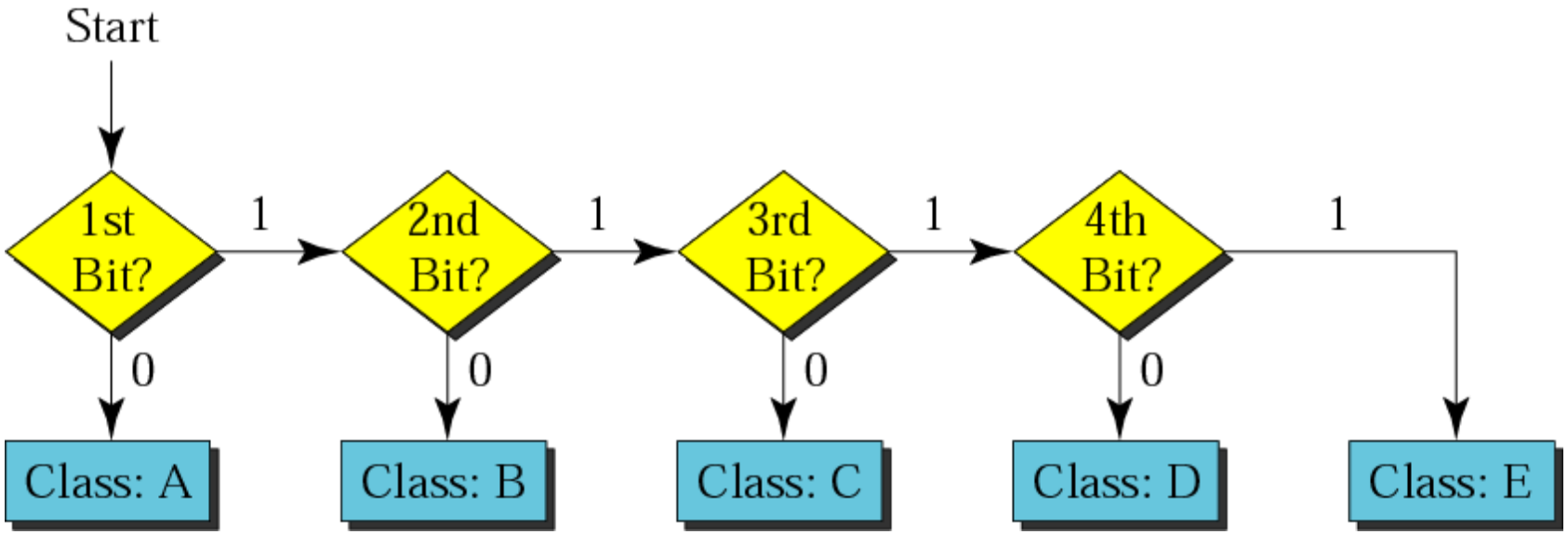
Classful Addressing

IP Address Classes

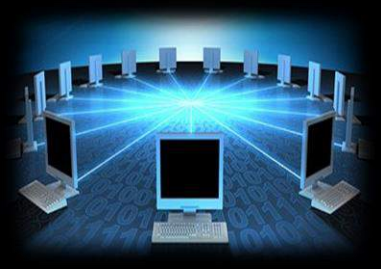


Classful Addressing

Finding the address class



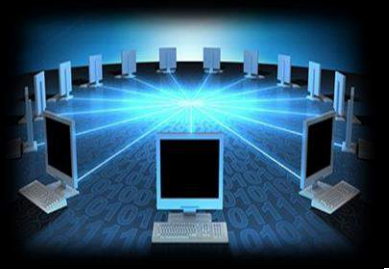
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Classful Addressing

- **Finding the class in decimal notation**

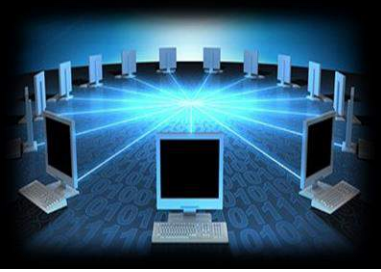
	First byte	Second byte	Third byte	Fourth byte
Class A	**0 to 127**			
Class B	**128 to 191**			
Class C	**192 to 223**			
Class D	**224 to 239**			
Class E	**240 to 255**			



Classful Addressing

Class range of IP addresses

Class	Range	Maximum no. of networks	Maximum no. of hosts
A	0.0.0.0 to 127.255.255.255	2^7	2^{24}
B	128.0.0.0 to 191.255.255.255	2^{14}	2^{16}
C	192.0.0.0 to 223.255.255.255	2^{21}	2^8
D	224.0.0.0 to 239.255.255.255	--	--
E	240.0.0.0 to 255.255.255.255	--	--



Classful Addressing

Example:

- Find the class of the address: 00000001 00001011 00001011 11101111

Ans: The first bit is 0. This is a class A address.

- Find the class of the address: 11000001 10000011 00011011 11111111

Ans: The first 2 bits are 1; the third bit is 0. This is a class C address.

- Find the class of the address: 227.12.14.87

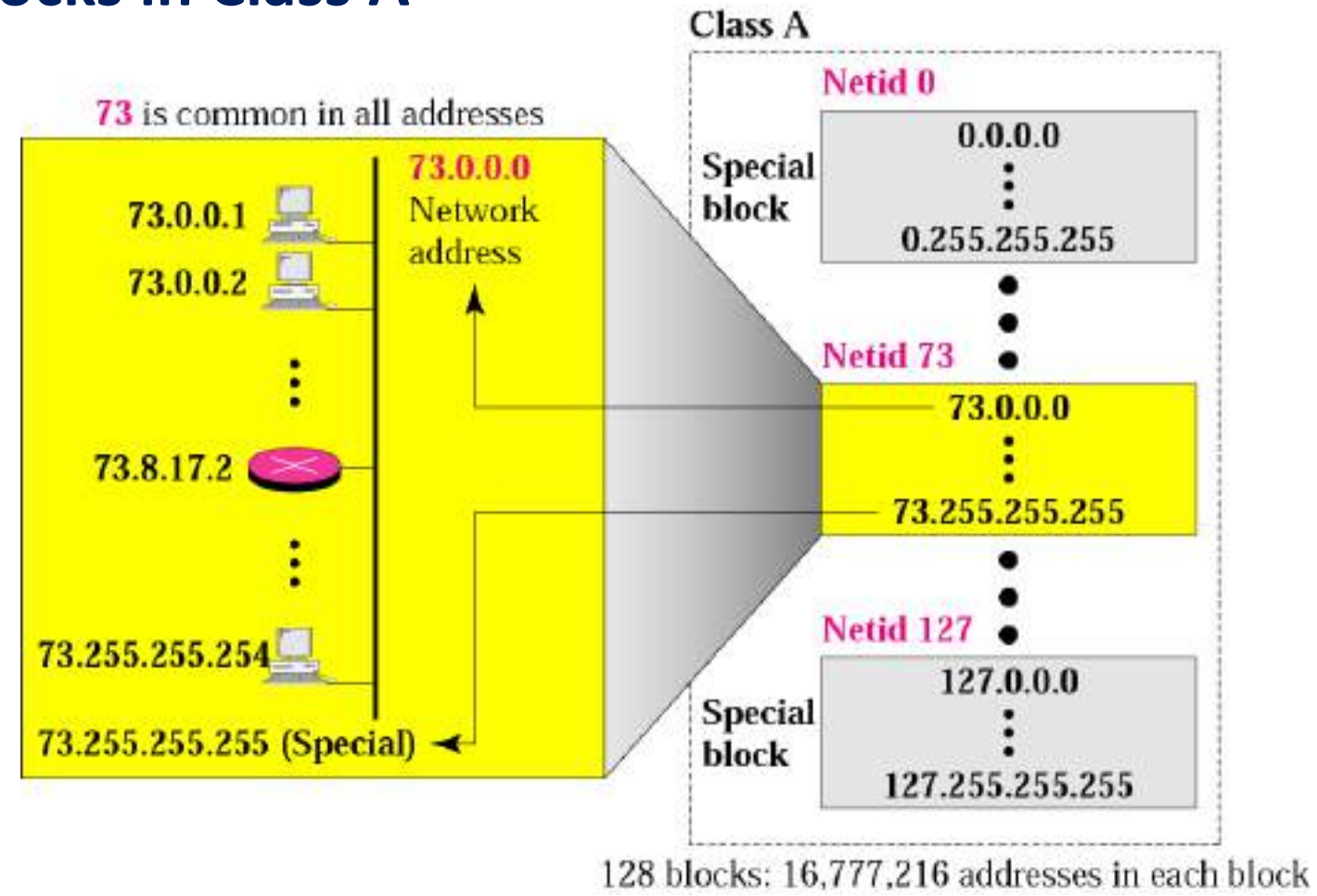
Ans: The first byte is 227 (between 224 and 239). This is a class D address.

- Find the class of the address: 193.14.56.22

Ans: The first byte is 193 (between 192 and 223). This is a class C address.

Classful Addressing

❑ Blocks in Class A

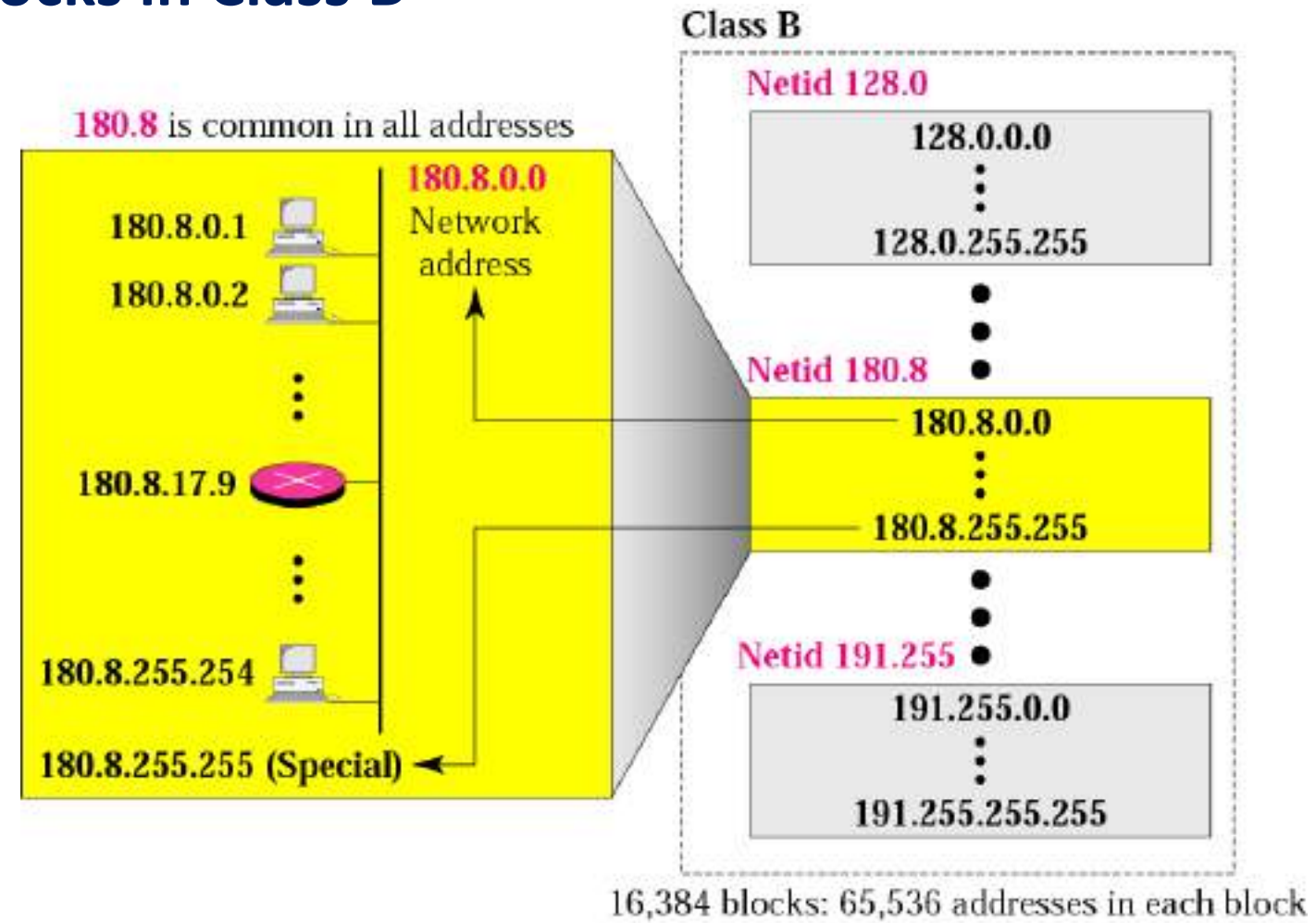


⇒ **Millions of class A addresses are wasted.**

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Classful Addressing

❑ Blocks in Class B

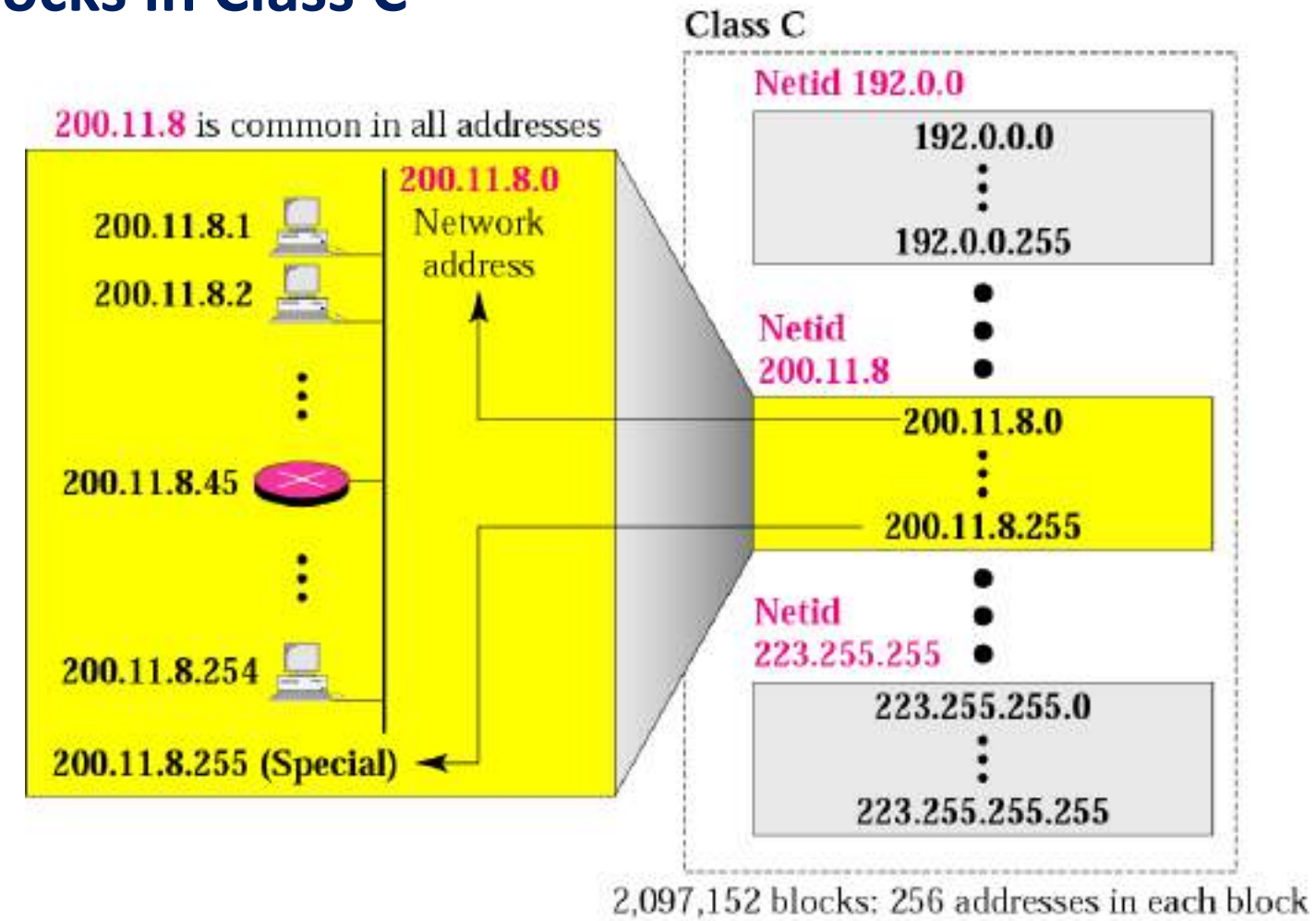


⇒ **Many class B addresses are wasted.**

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

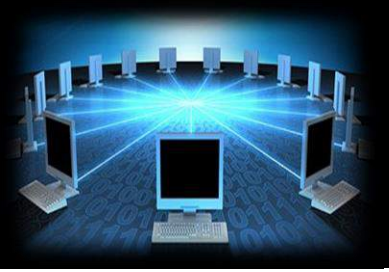
Classful Addressing

❑ Blocks in Class C



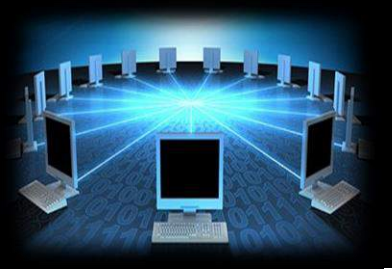
⇒ The number of addresses in a class C block is smaller than the needs of most organizations.

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Classful Addressing

- Millions of class A addresses are wasted.
- Many class B addresses are wasted.
- The number of addresses in a class C block is smaller than the needs of most organizations.
- Class D addresses are used for multicasting; there is only one block in this class.
- Class E addresses are reserved for special purposes; most of the block is wasted.



Classful Addressing

Example:

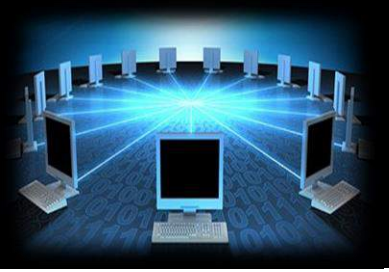
- Given the network address 132.21.0.0, find the class, the block, and the range of the addresses.

Ans:

The class is B because the first byte is between 128 and 191.

The block has a NetID of 132.21.

The addresses range from 132.21.0.0 to 132.21.255.255.



Classful Addressing

Example:

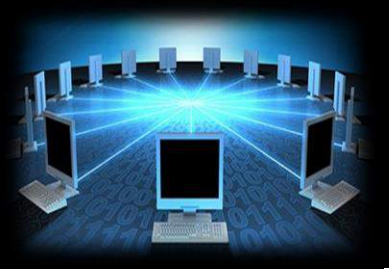
- Given the network address 220.34.76.0, find the class, the block, and the range of the addresses.

Ans:

The class is C because the first byte is between 192 and 223.

The block has a NetID of 220.34.76.

The addresses range from 220.34.76.0 to 220.34.76.255.



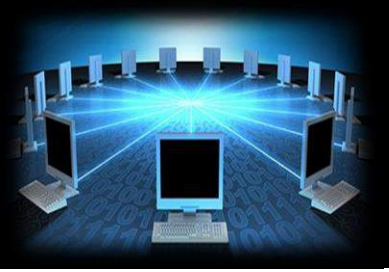
Classful Addressing

Mask

- Although the length of the NetID and HostID (in bits) is predetermined in classful addressing, we can also use a **mask** (also called the **default mask**), a 32-bit number made of contiguous 1's followed by contiguous 0's.

Default masks for classful addressing

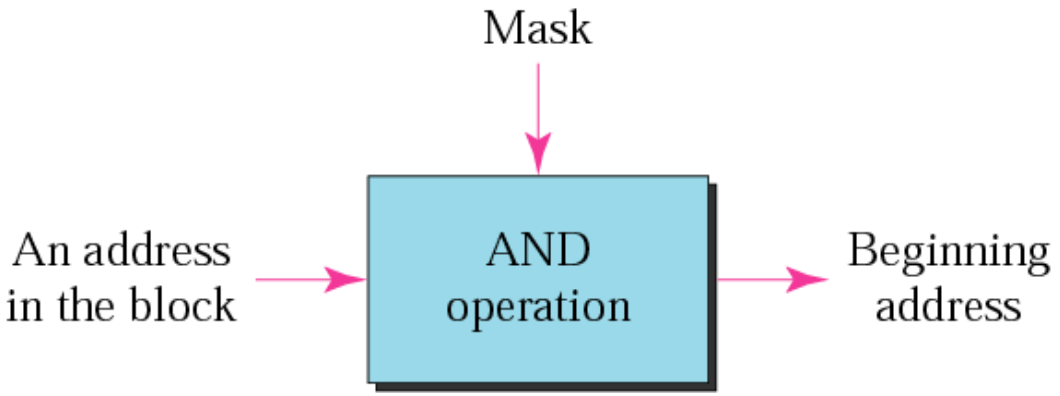
Class	Mask in binary	Mask in dotted-decimal
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0



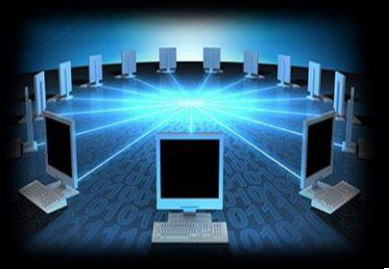
Classful Addressing

Mask

- The network address is the beginning address of each block.
- It can be found by applying the default mask to any of the addresses in the block (including itself).
- It retains the NetID of the block and sets the HostID to zero.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Classful Addressing

Example:

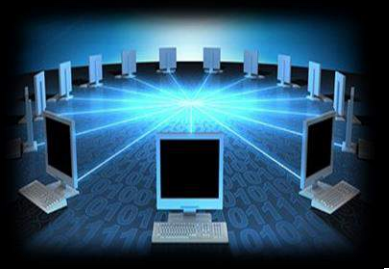
- Given the address 201.180.56.5 and the class C default mask, find the beginning address (network address).

Ans:

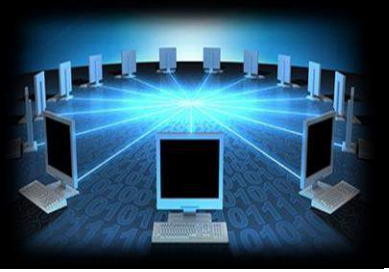
The default mask of Class C is 255.255.255.0

This means that the first 3 bytes are preserved, and the last byte is set to 0.

The network address is 201.180.56.0.



Subnetting & Supernetting

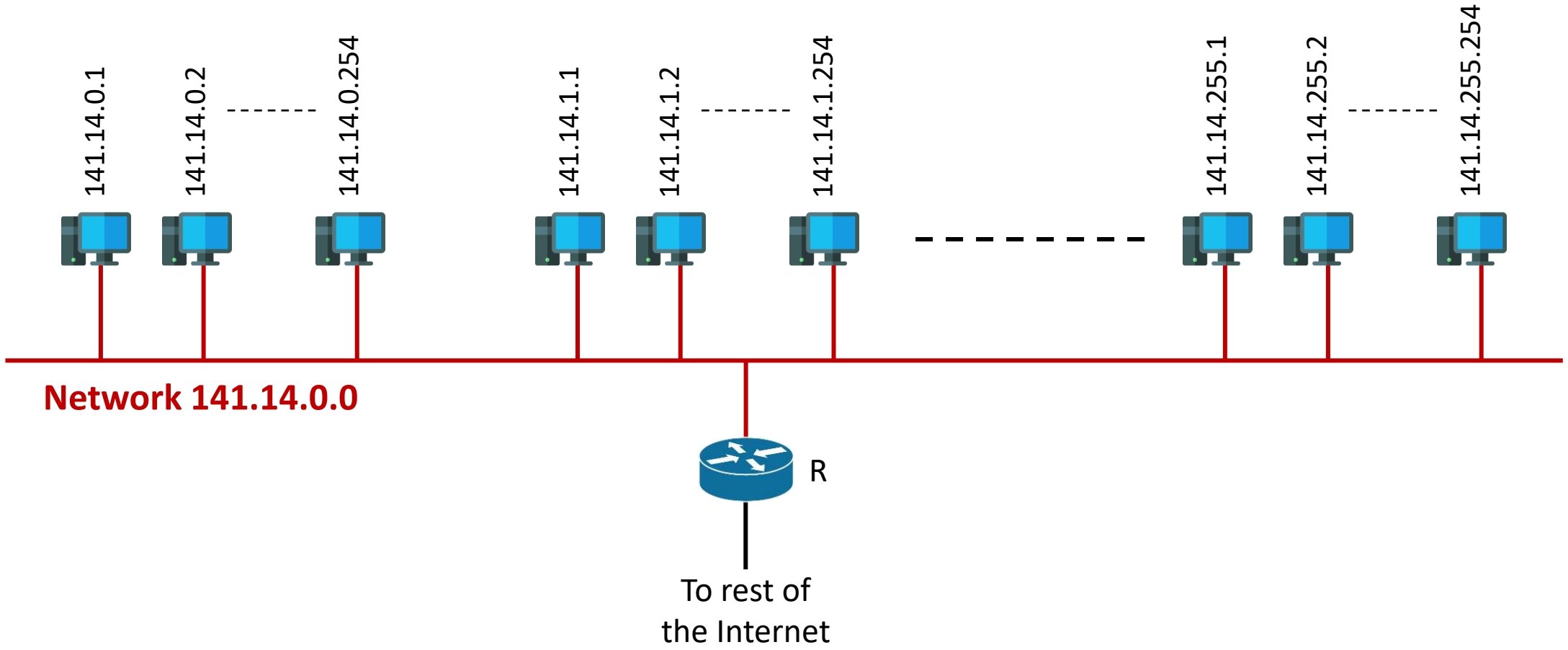


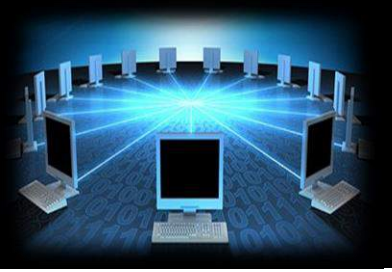
Subnetting

- IP addresses were originally designed with two levels of addressing.
- To reach a host on the Internet, we must first reach the network and then the host.
- However, organizations with class A or B needs to divide its large network into several subnetworks for better security and management.
- The idea of splitting a block to smaller blocks is referred to as **subnetting**.
- In subnetting, a network is divided into several smaller subnetworks (subnets) with each subnetwork having its own subnetwork address.

Subnetting

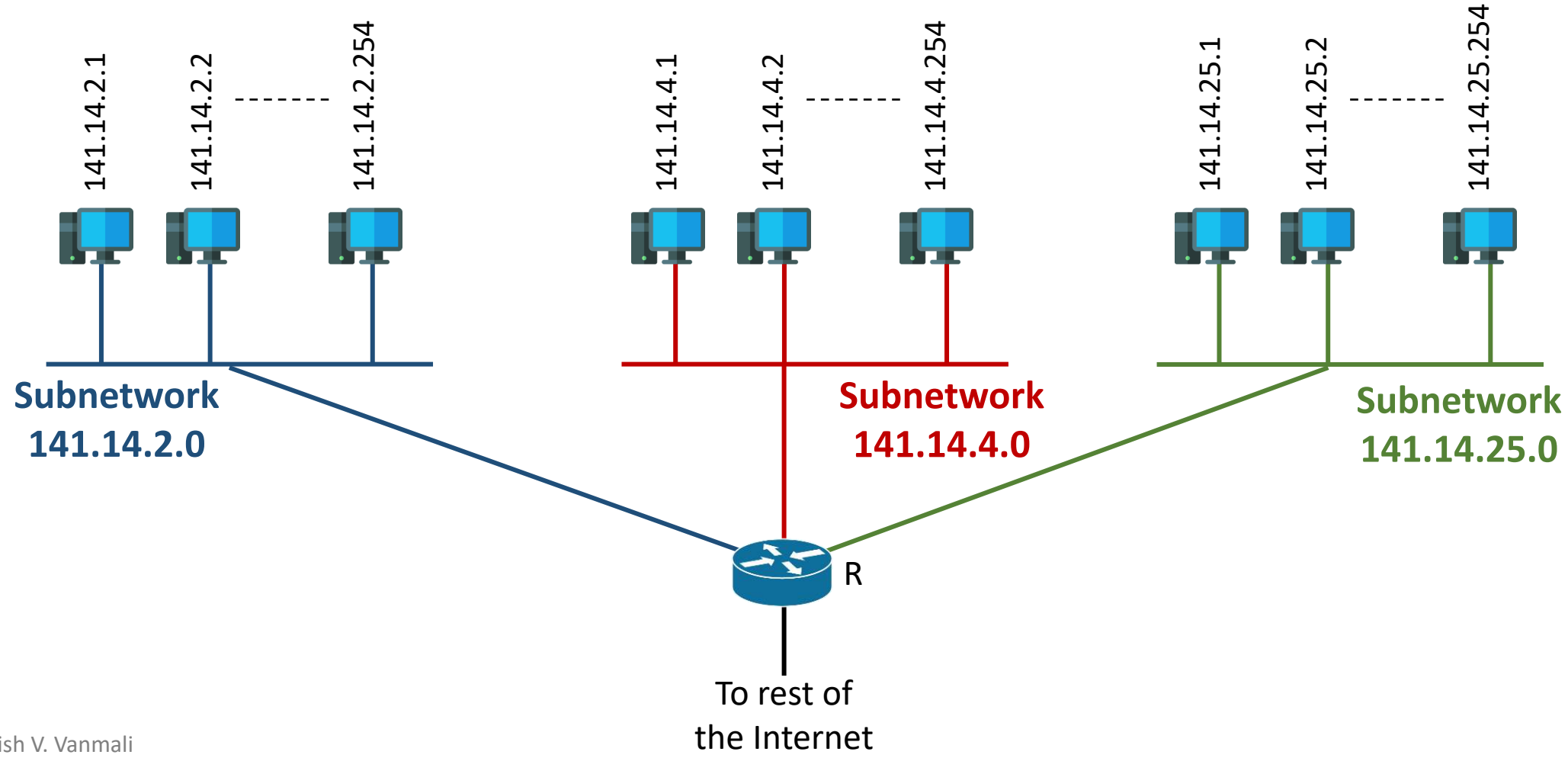
Network without Subnetting

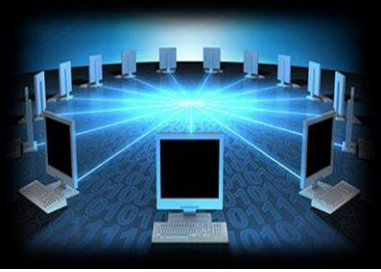




Subnetting

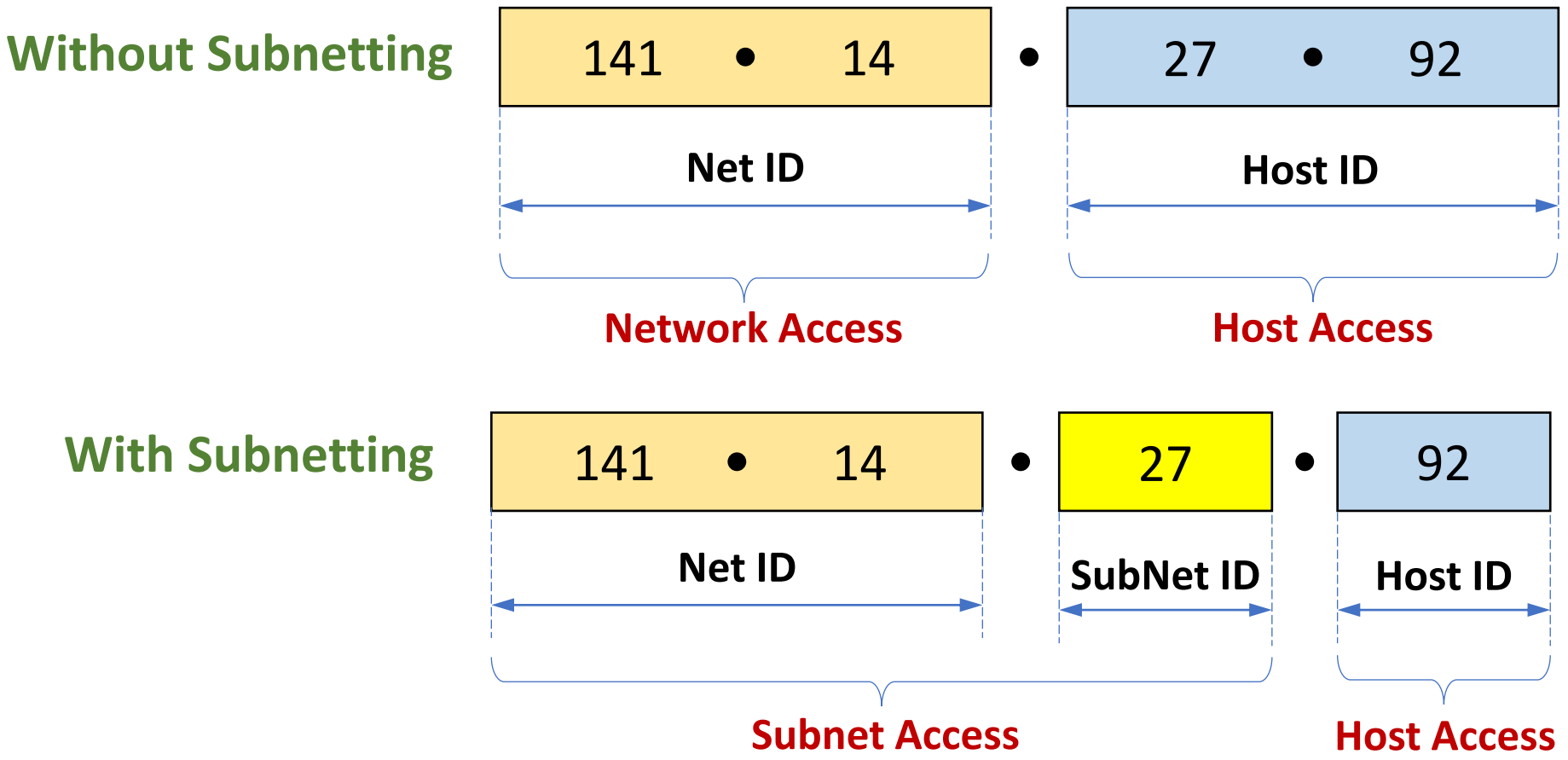
Network with Subnetting

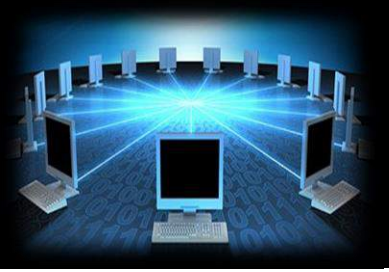




Subnetting

❑ Addresses in a network with and without subnetting

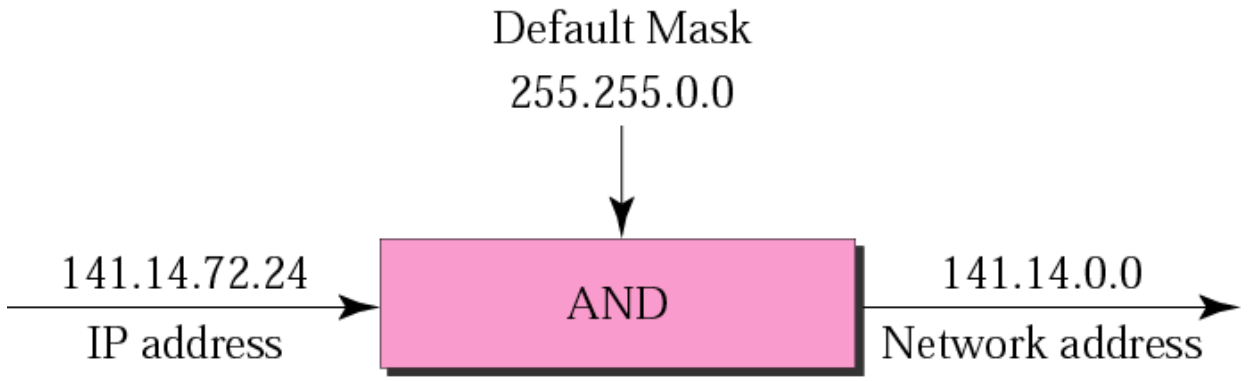




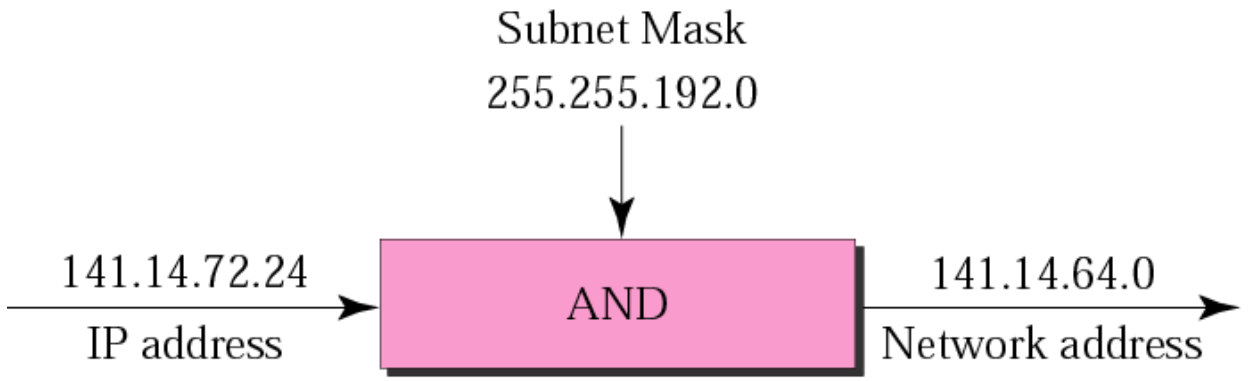
Subnetting

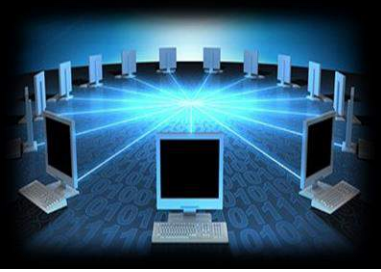
❑ Default mask and Subnet mask

Without Subnetting



With Subnetting





Subnetting

Finding the Subnet Address

- **Method 1:**
  - Use binary notation for both the address and the mask and then apply the AND operation to find the subnet address.
  - Example –

IP Address = 200.45.34.56

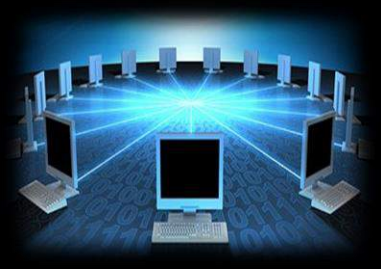
Subnet mask = 255.255.240.0

200.45.34.56 = 11001000 00101101 00100010 00111000

255.255.240.0 = AND 11111111 11111111 11110000 00000000

11001000 00101101 00100000 00000000

∴ Subnet address = 200.45.32.0.



Subnetting

Finding the Subnet Address

- **Method 2 (Short-cut method):**
  - If the byte in the mask is 255, copy the byte in the address.
  - If the byte in the mask is 0, replace the byte in the address with 0.
  - If the byte in the mask is neither 255 nor 0, we write the mask and the address in binary and apply the AND operation.

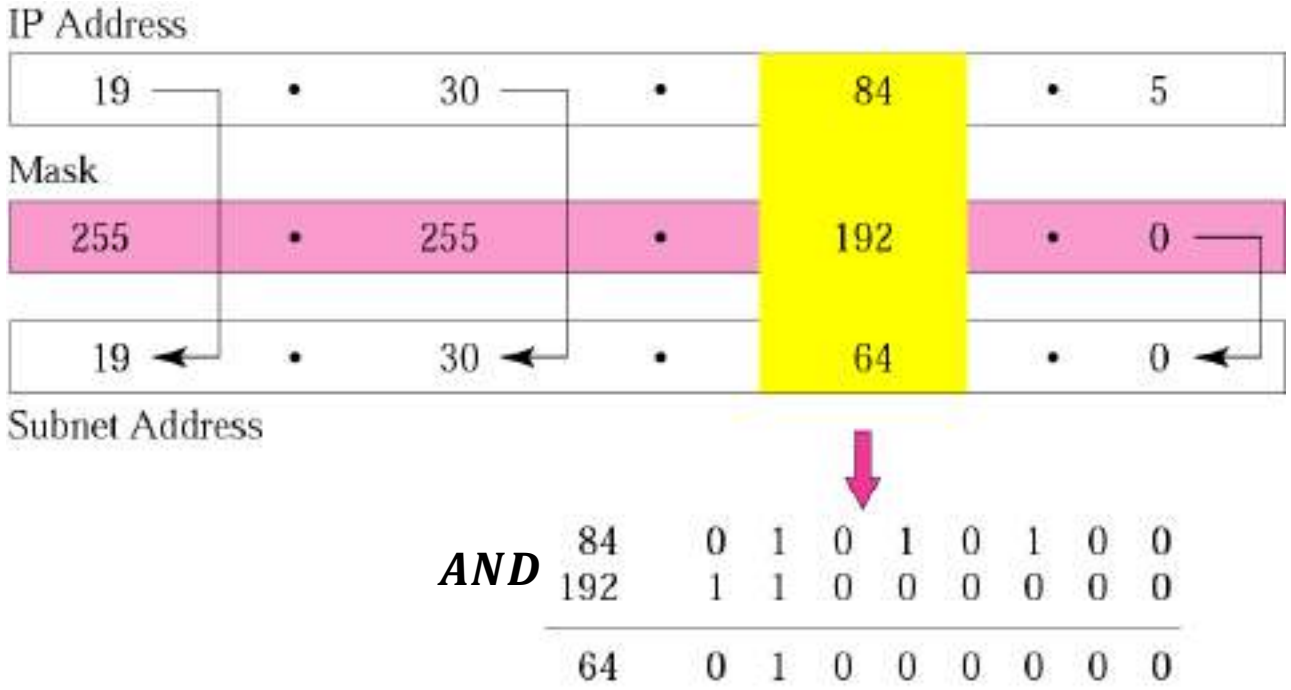
Subnetting

Finding the Subnet Address

- **Method 2 (Short-cut method):**
  - Example –

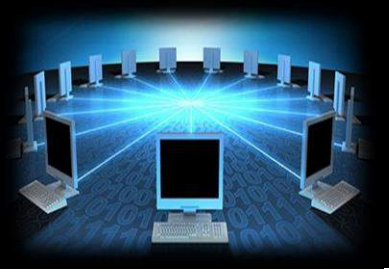
    IP Address = 19.30.80.5

    Subnet mask = 255.255.192.0



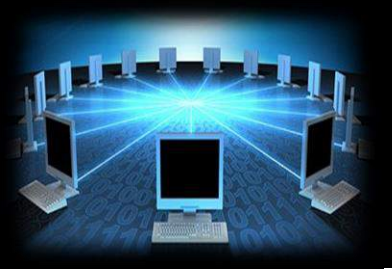
∴ Subnet address = 19.30.64.0

Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

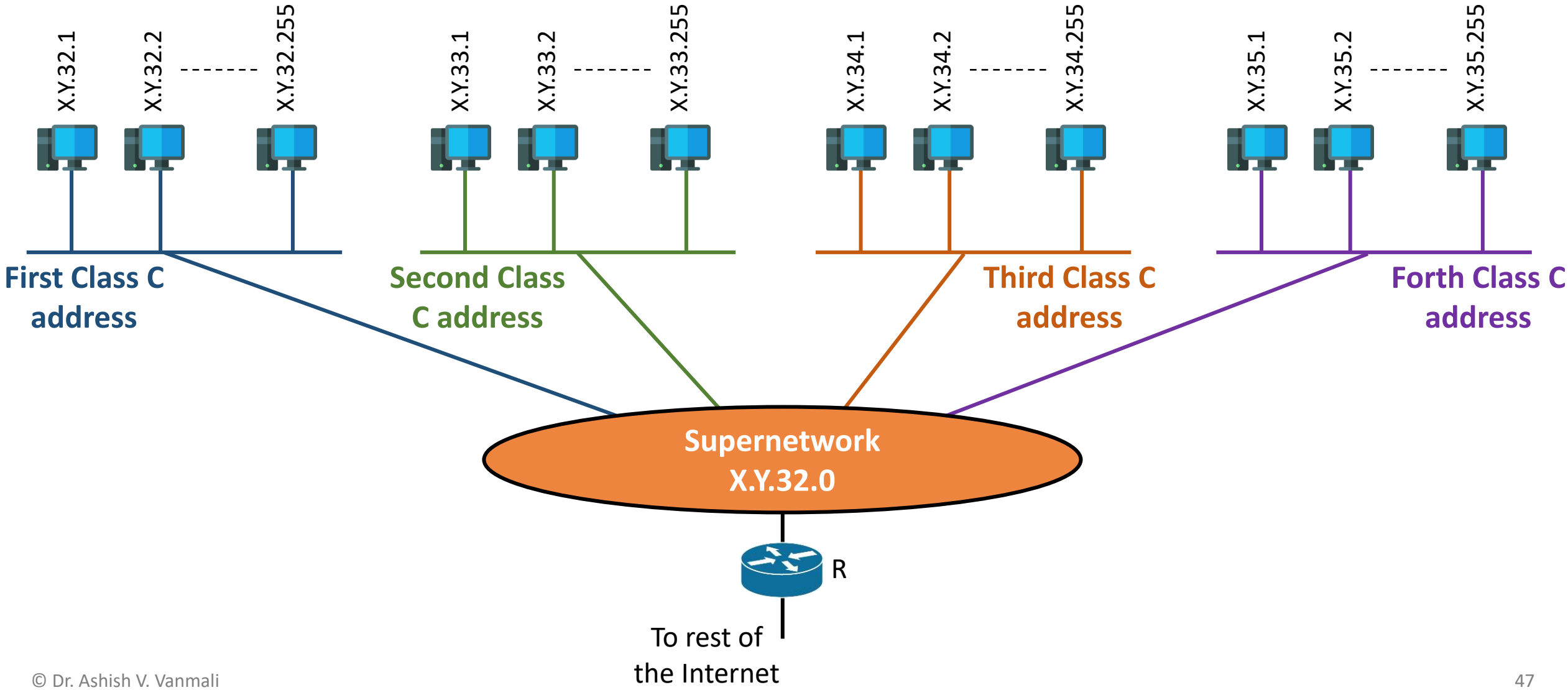


Supernetting

- Subnetting could not completely solve address depletion problems in classful addressing because most organizations did not want to share their granted blocks with others.
- Since class C blocks were still available but the size of the block did not meet the requirement of new organizations that wanted to join the Internet, one solution was supernetting.
- In **supernetting**, an organization can combine several class C blocks to create a larger range of addresses.
- In other words, several networks are combined to create a supernetwork.
- By doing this, an organization can apply for several class C blocks instead of just one.

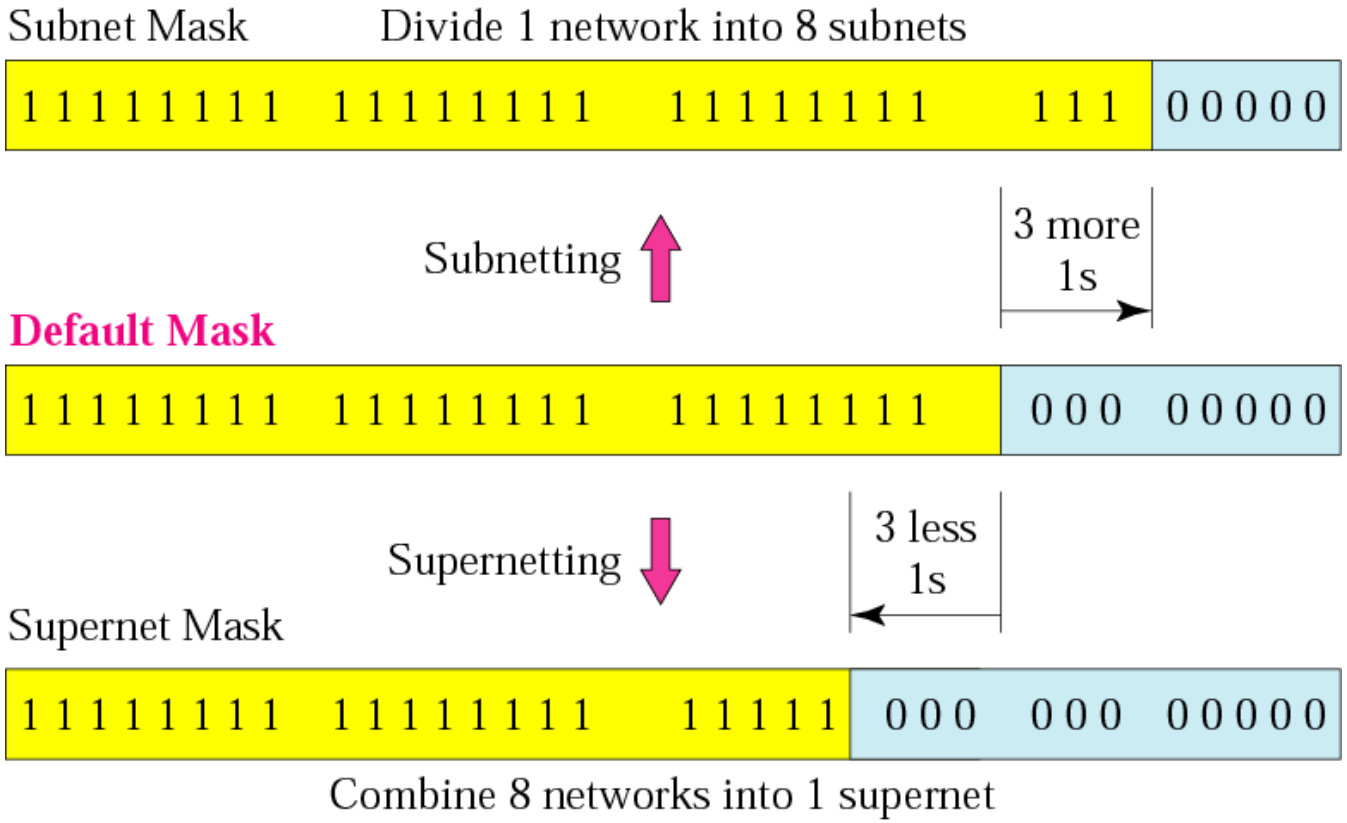


Supernetting

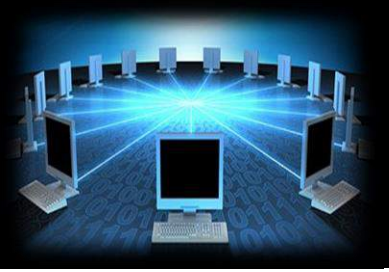


Supernetting

❑ Comparison of Subnet, Default, and Supernet masks



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Supernetting

Finding Supernet Address

- Example –

IP Address = 205.16.34.10

Supernet mask = 255.255.248.0

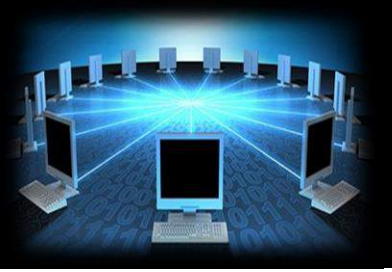
205.16.34.10 = 11001101 00010000 00100010 00001010

$\overset{AND}{}$

255.255.248.0 = 11111111 11111111 11111000 00000000

11001101 00010000 00100000 00000000

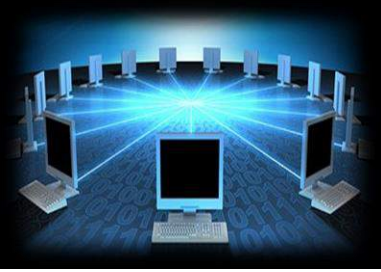
∴ Supernet address = 205.16.32.0



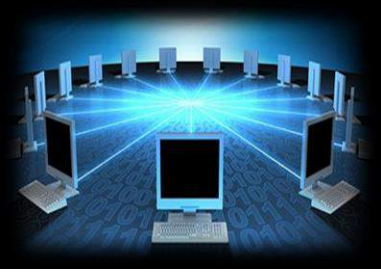
Subnetting v/s Supernetting

Basic of Comparison	Subnetting	Supernatting
Description	Subnetting is a technique of dividing a network into two or more sub-networks.	Supernetting is a technique of aggregating various networks to form to form one single large network.
Implementation	Subnetting is implemented via Variable-length subnet masking.	Supernetting is implemented via classless inter-domain routing.
Importance	Subnetting helps to reduce the address depletion.	Supernetting helps to simplify and fasten the routing process.
Mask Bits	In Subnetting, the mask bits are removed towards the right of the default mask.	In Supernetting, the movement of the masked bits is towards the left of the default mask.
Effect	In Subnetting, the network address's number of bits are significantly increased.	In Supernetting, the host address's number of bits are significantly increased.

Ref:
https://forum.huawei.com/enterprise/en/data/attachment/forum/202106/26/142324ech5x155u92vhk7j.png?10.PNG



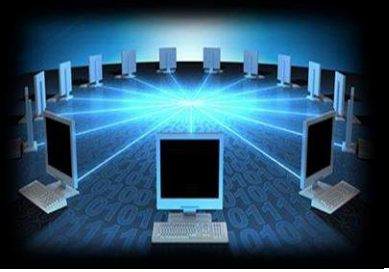
Classless Addressing



Classless Addressing

- Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult.
- In **Classless Addressing** the class privilege was removed from the distribution to compensate for the address depletion.
- In classless addressing, the whole address space is divided into variable length blocks.
- Theoretically, we can have a block of 20, 21, 22, . . . , 232 addresses.
- The only restriction is that the number of addresses in a block needs to be a power of 2.

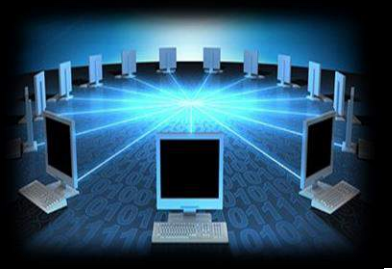
Classless Addressing is also referred as CIDR (Classless Inter-Domain Routing)



Classless Addressing

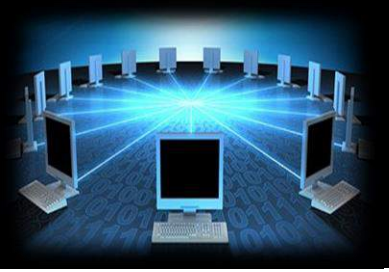
- In classful addressing, two-level addressing was provided by dividing an address into NetID and HostID.
- The NetID defined the network; the HostID defined the host in the network.
- In classless addressing, divided into two parts, the **prefix** and the **suffix**.
- The prefix plays the same role as the NetID; the suffix plays the same role as the HostID.
- All addresses in the block have the same prefix; each address has a different suffix.

Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Classless Addressing

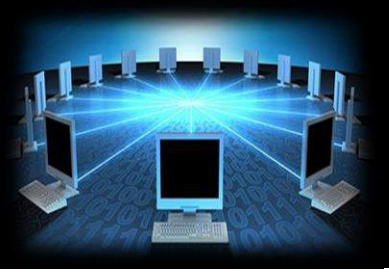
- In classful addressing, the length of the NetID, n, depends on the class of the address; it can be only 8, 16, or 24.
- In classless addressing, the length of the prefix, n, depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32.
- In classless addressing, the value of n is referred to as **prefix length**; the value of $(32 - n)$ is referred to as **suffix length**.



Classless Addressing

Slash Notation

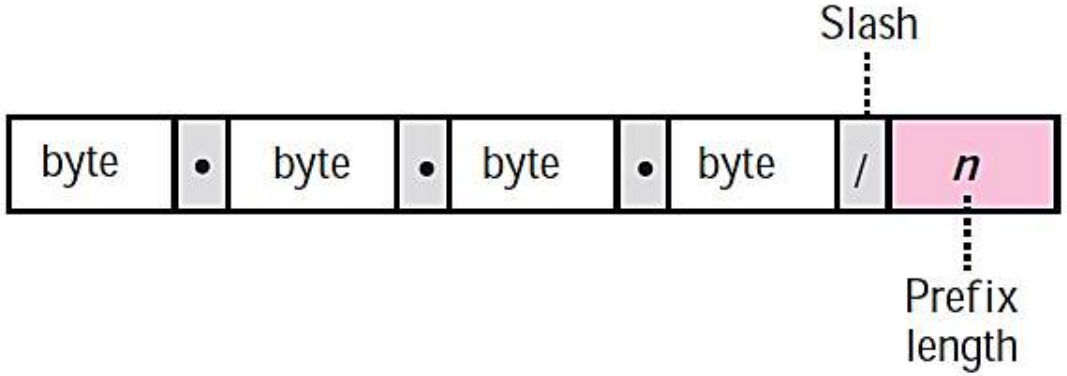
- The NetID length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block.
- However, there is a difference here in classful and classless addressing:
  - In classful addressing, the NetID length is inherent in the address. Given an address, we know the class of the address that allows us to find the NetID length (8, 16, or 24).
  - In classless addressing, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length.
- In classless addressing, we need to include the prefix length to each address if we need to find the block of the address.



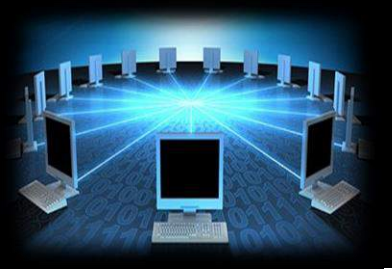
Classless Addressing

Slash Notation

- In classless addressing, the prefix length n is added to the address separated by a slash. The notation is informally referred to as **slash notation**.
- The slash notation is formally referred to as **classless inter-domain routing** or **CIDR** (pronounced cider) notation.
- The prefix length n indicates that first n bits of the Network Mask are 1 and remaining $(32 - n)$ bits are 0.



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Classless Addressing

Extracting Block Information

- The number of addresses in the block can be found as:

$$N = 2^{32-n}$$

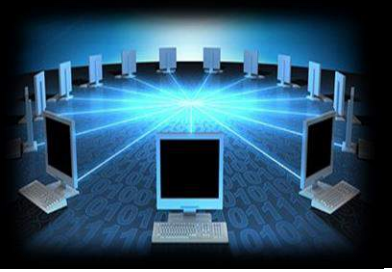
- The first address (network address) in the block can be found by ANDing the address with the network mask:

$$First\ address\ =\ (any\ address)\ AND\ (network\ mask)$$

Alternatively, we can keep the n leftmost bits of any address in the block and set the $(32 - n)$ bits to 0s to find the first address.

- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

$$Last\ address\ =\ (any\ address)\ OR\ [NOT\ (network\ mask)]$$



Classless Addressing

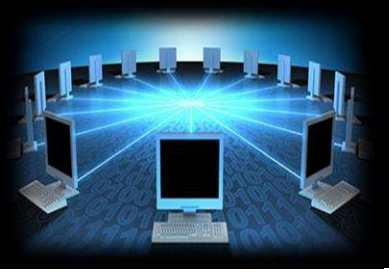
Example:

$$167.199.170.82/27$$

- The value of $n = 27$. The network mask has twenty-seven 1s and five 0s.
  i.e. $11111111\ 11111111\ 11111111\ 11100000 \implies 255.255.255.240$
- The number of addresses in the network
  $$N = 2^{32-n} = 2^{32-27} = 2^5 = 32$$
- First Address:

$$\begin{array}{lll} Address\ in\ binary: & & 10100111\ 11000111\ 10101010\ 01010010 \\ Network\ mask: & \boldsymbol{AND} & 11111111\ 11111111\ 11111111\ 11100000 \\ \hline First\ address: & & 10100111\ 11000111\ 10101010\ 01000000 \end{array}$$

$$\implies First\ Address = 167.199.170.64/27$$



Classless Addressing

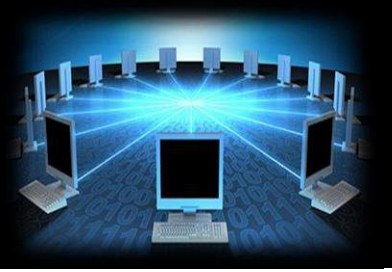
Example:

$$167.199.170.82/27$$

- Last Address:

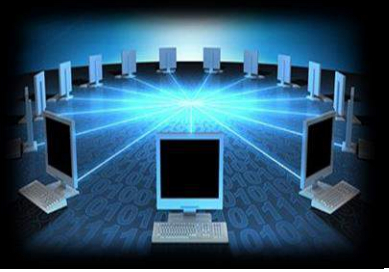
$$\begin{array}{ll} Address\ in\ binary: & 10100111\ 11000111\ 10101010\ 01010010 \\ Complement\ of\ Network\ mask: & \boldsymbol{OR}\quad 00000000\ 00000000\ 00000000\ 00011111 \\ \hline First\ address: & 10100111\ 11000111\ 10101010\ 01011111 \end{array}$$

$$\Longrightarrow Last\ Address = 167.199.170.95/27$$

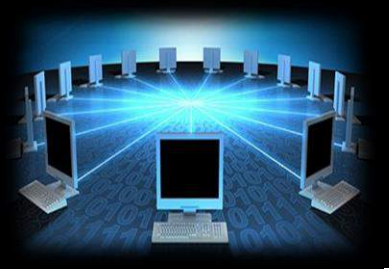


For problems on Subnetting, Supernetting, Classless Addressing refer:

Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill Education

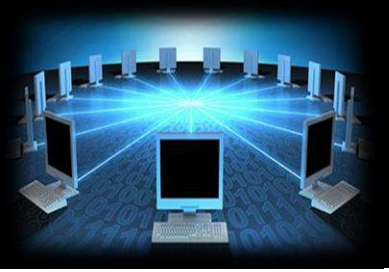


IPv4 Protocol



IPv4 Protocol

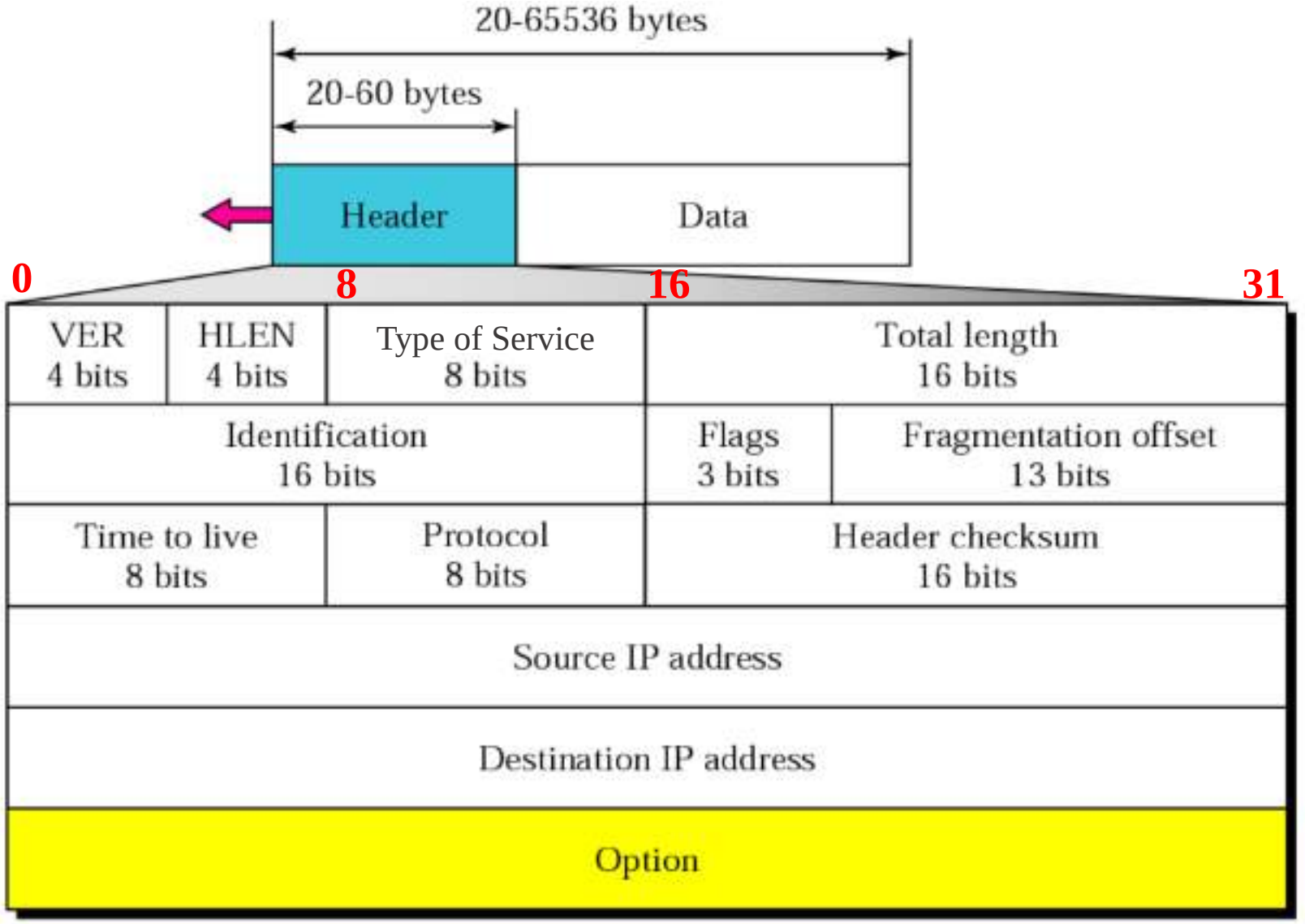
- **The Internet Protocol (IP)** is the transmission mechanism used by the TCP/IP protocols at the network layer.
- IP is an unreliable, connectionless datagram protocol with a best-effort delivery service.
- Unreliable $\Longrightarrow$ IP packets can be corrupted, lost, arrive out of order, or delayed and may create congestion for the network.
- Connectionless $\Longrightarrow$ No Need establish virtual circuit.
- Best-effort $\Longrightarrow$ IP will try its best to forward packets to the destination but does not guarantee.



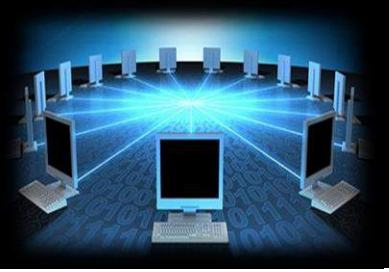
IP Datagram Format

- Packets in the network (internet) layer are called **datagrams**.
- A datagram is a variable-length packet consisting of two parts: header and payload (data).
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.
- It is customary in TCP/IP to show the header in 4-byte sections.

IP Datagram Format



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



IP Datagram Format

Version (VER)

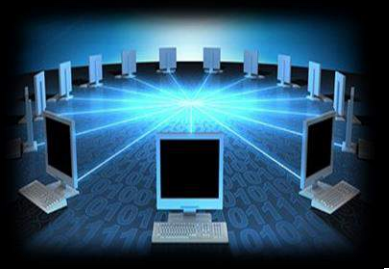
- Defines the version of IP protocol.
- Currently the version is 4. However, version 6 (or IPv6) may totally replace version 4 in the future.

Header length (HLEN)

- Defines the total length of the datagram header in multiples of 4-bytes.
- Maximum HLEN = $4 \times 15 = 60$

Type of Service

- Defines how the datagram should be handled.
- Includes bits that define priority of datagram.
- Also specifies type of service the sender desires such as level throughput, reliability, delay etc.



IP Datagram Format

Total Length

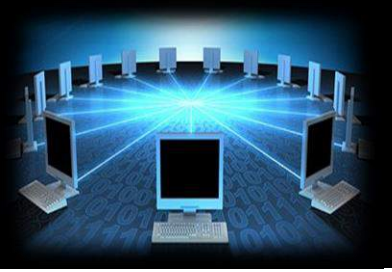
- Defines the total length of the datagram including the header.
- Length of data = Total length − Header length
- The total length of the IP datagram is limited to $65{,}535$ $(2^{16} - 1)$ bytes.

Identification

- This field is used in fragmentation.
- A packet when passing through different networks may be divided into fragments to match the network frame size. When this happens, each fragment is identified with sequence number in this field.

Flags

- This field is used in fragmentation (can or cannot be fragmented).



IP Datagram Format

Fragmentation Offset

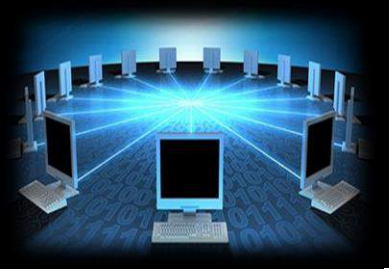
- It is a pointer that shows offset of data in original packet (if it is fragmented).

Time to Live

- Defines number of hops a datagram can travel before it is discarded.
- Source sets this field with initial value. As the packet travels through network, each router decrements its value by 1.
- If the value becomes zero before packet reaches its destination, the packet is discarded.

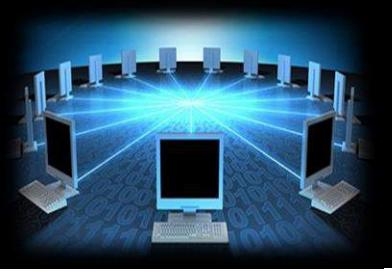
Protocol

- Defines the higher-level protocol that uses the services of the IP layer.
- E.g. ICMP =1 IGMP = 2
  TCP = 6 UDP = 17
  OSPF = 89

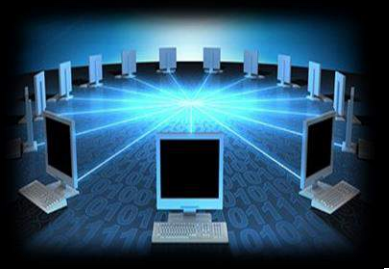


IP Datagram Format

- **Header Checksum**
  - Used to check the integrity of header (not rest of the packet).
  - When a router decrements the TTL, router must also recompute the header checksum.
- **Source and Destination Address**
  - 32-bit fields define the logical (IP) address of the source and destination.
- **Option**
  - Gives more functionalities to IP packet.
  - Can carry data that controls routing, timing, management etc.

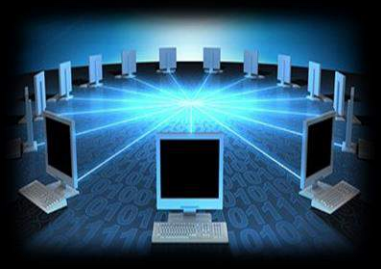


DHCP Protocol



DHCP Protocol

- **Dynamic Host Configuration Protocol (DHCP)** is a client/server protocol.
- It automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway.
- DHCP port number for server is 67 and for the client is 68.
- In DHCP, the client and the server exchange mainly 4 DHCP messages (discovery, offer, request, and ACK) in order to make a connection, also called DORA process

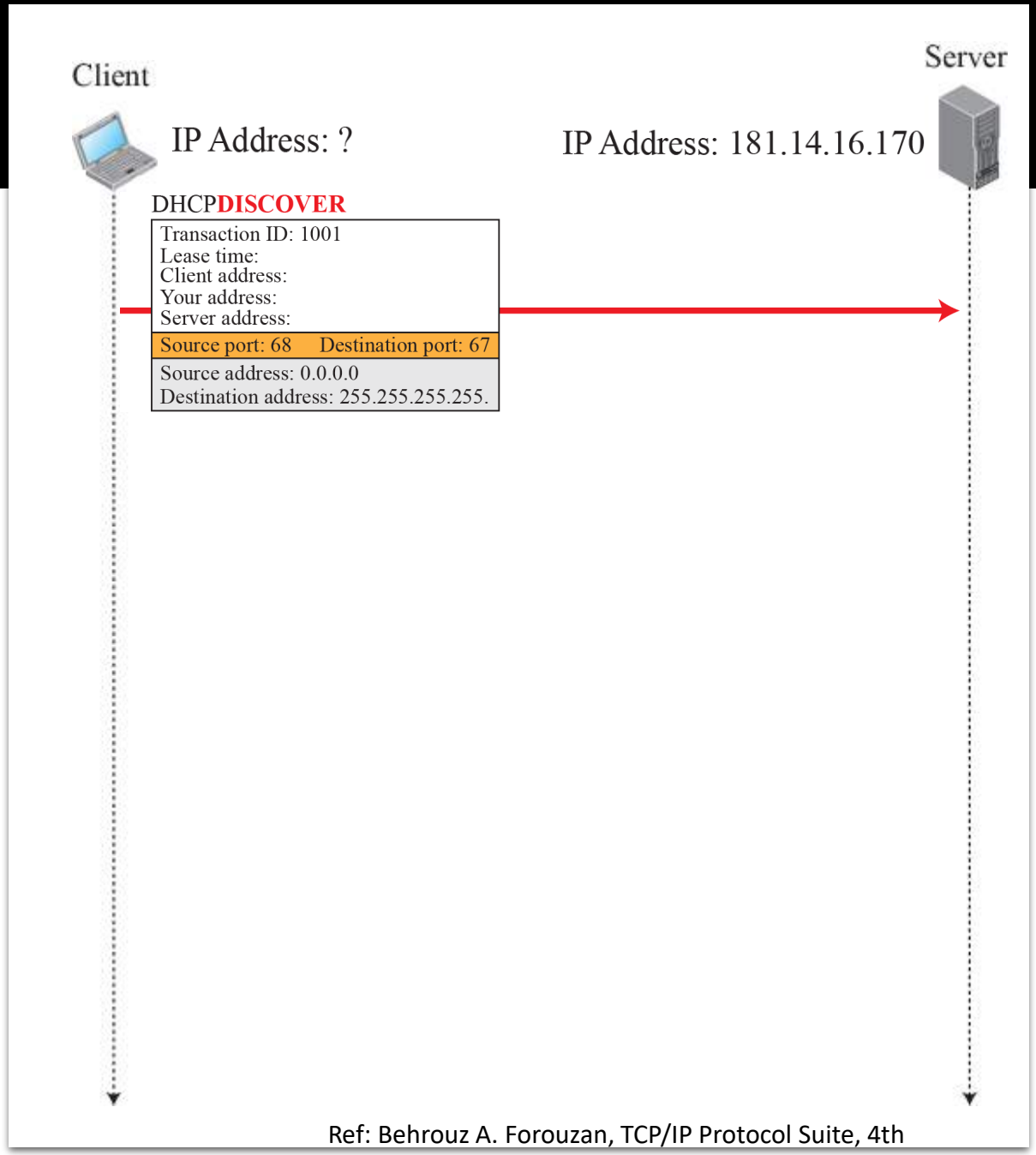


DHCP Protocol

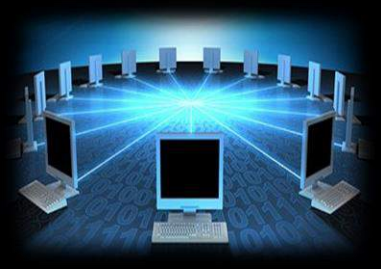
❑ Operation of DHCP

1. DHCP Discover message

- This is a first message generated in the communication process between server and client.
- This message is generated by Client host in order to discover if there is any DHCP server/servers are present in a network or not.
- This message is broadcasted to all devices present in a network to find the DHCP server.



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

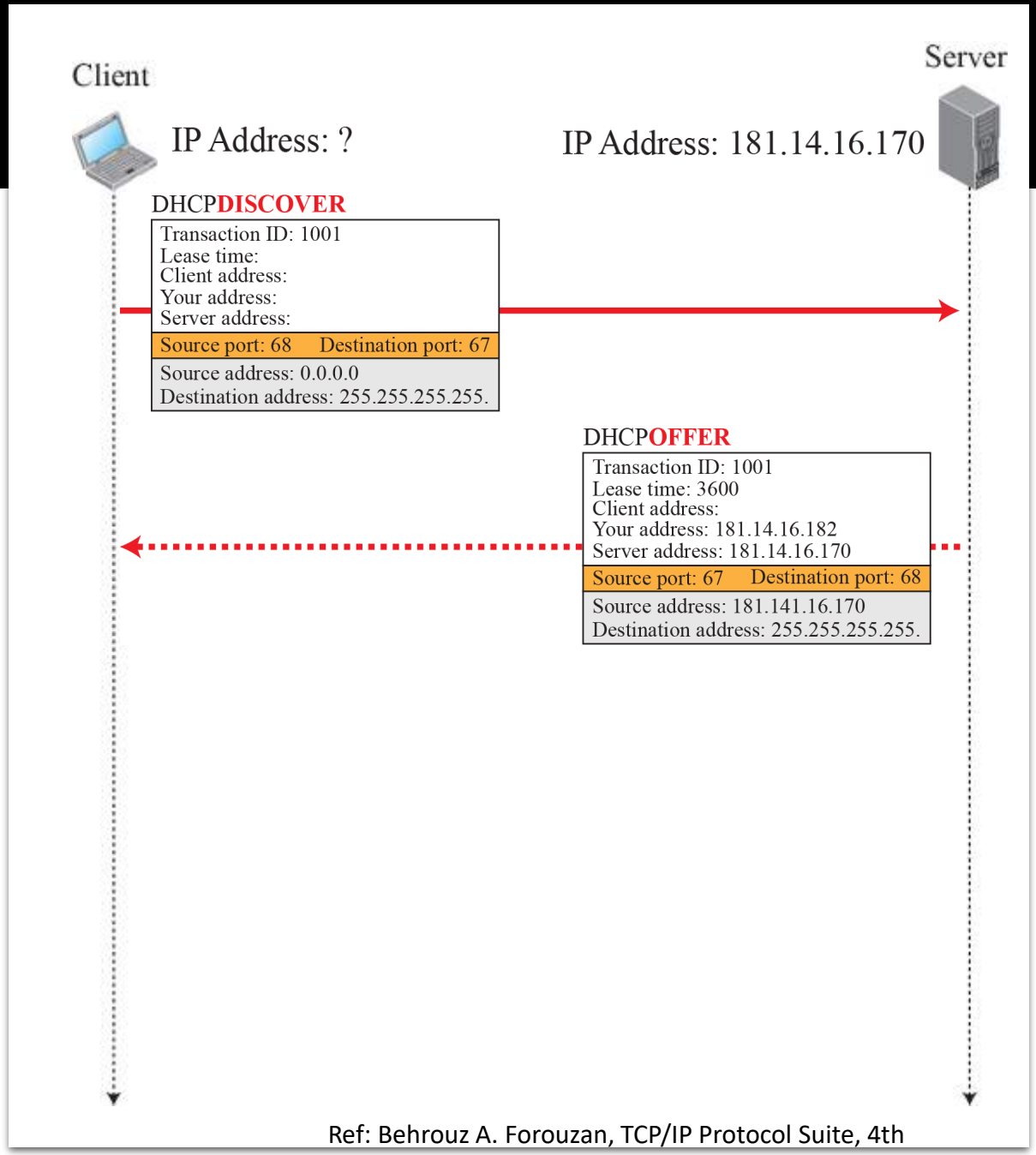


DHCP Protocol

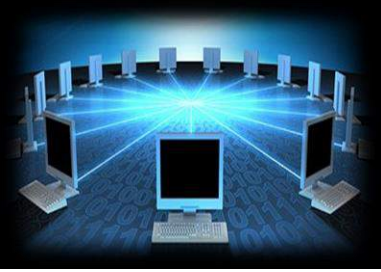
❑ Operation of DHCP

2. DHCP Offer message

- The server will respond to host in this message specifying the unleased IP address and other TCP configuration information.
- This message is broadcasted by server.
- If there are more than one DHCP servers present in the network then client host will accept the first DHCP OFFER message it receives.
- Also a server ID is specified in the packet in order to identify the server.



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

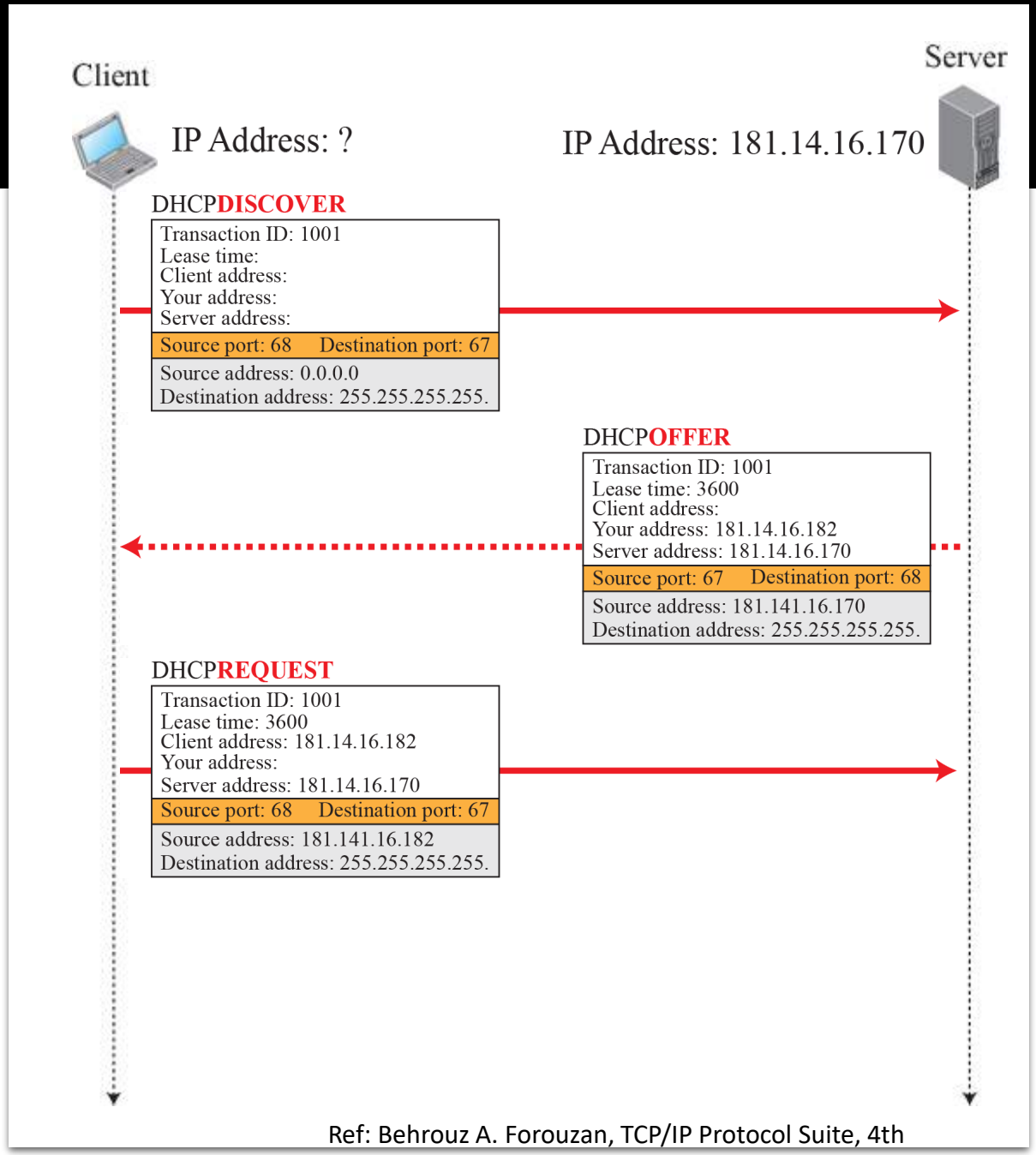


DHCP Protocol

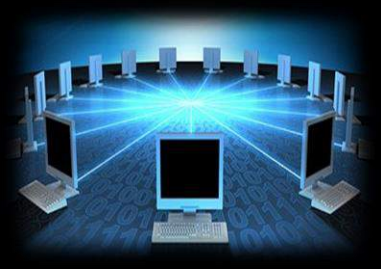
❑ Operation of DHCP

3. DHCP Request message

- When a client receives a offer message, it responds by broadcasting a DHCP request message.
- It is used to find if there is any other host present in the network with same IP address.
- If there is no reply by other host, then there is no host with same IP in the network and the message is broadcasted to server showing the acceptance of IP address.



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

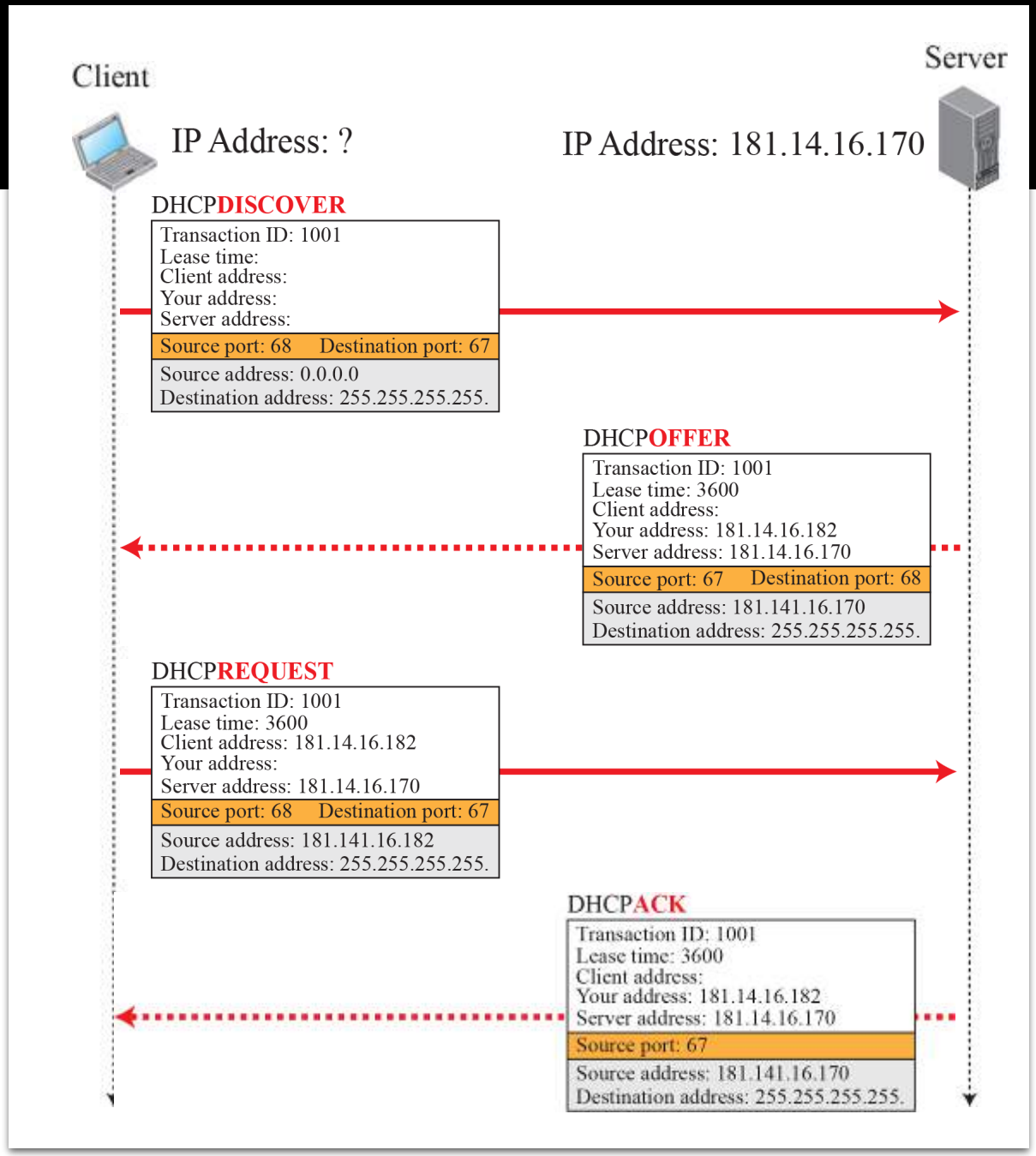


DHCP Protocol

❑ Operation of DHCP

4. DHCP ACK message

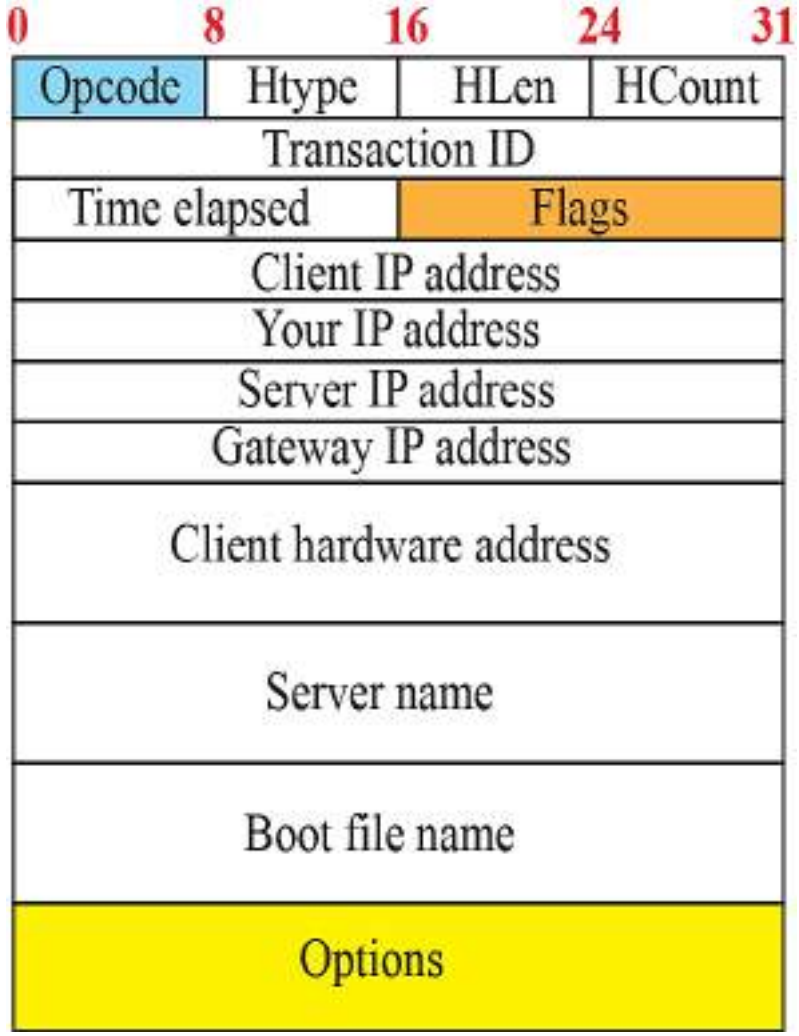
- In response to the request message received, the server will make an entry with specified client ID and bind the IP address offered with lease time.
- Now, the client will have the IP address provided by server.



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

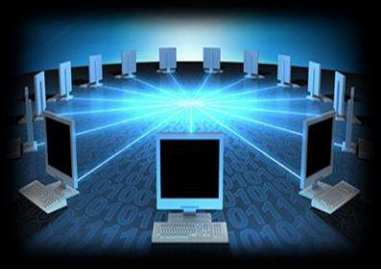
DHCP Protocol

DHCP message format

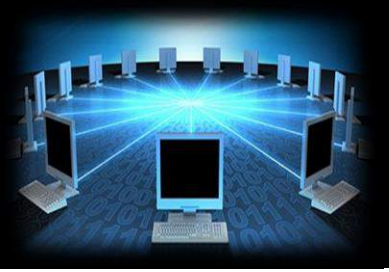


Fields:

Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Lengh of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

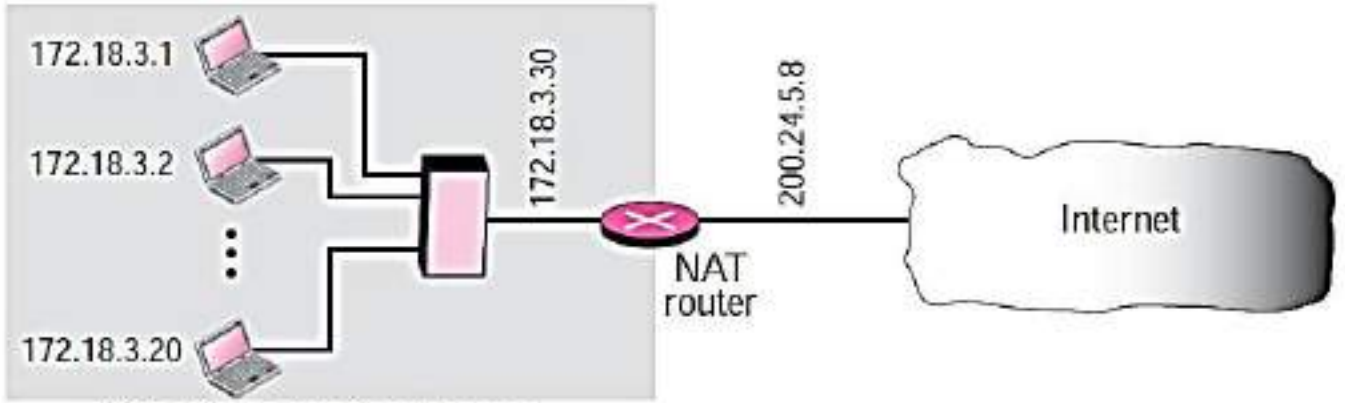


Network Address Translation (NAT)

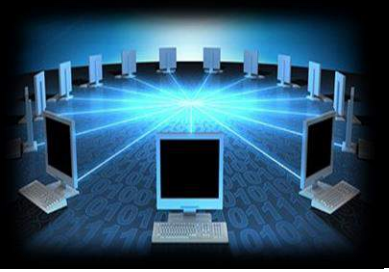


Network Address Translation (NAT)

- 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 Network Address Translation (NAT) is to allow multiple devices to access the Internet through a single public address.
- To achieve this, the translation of a 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.



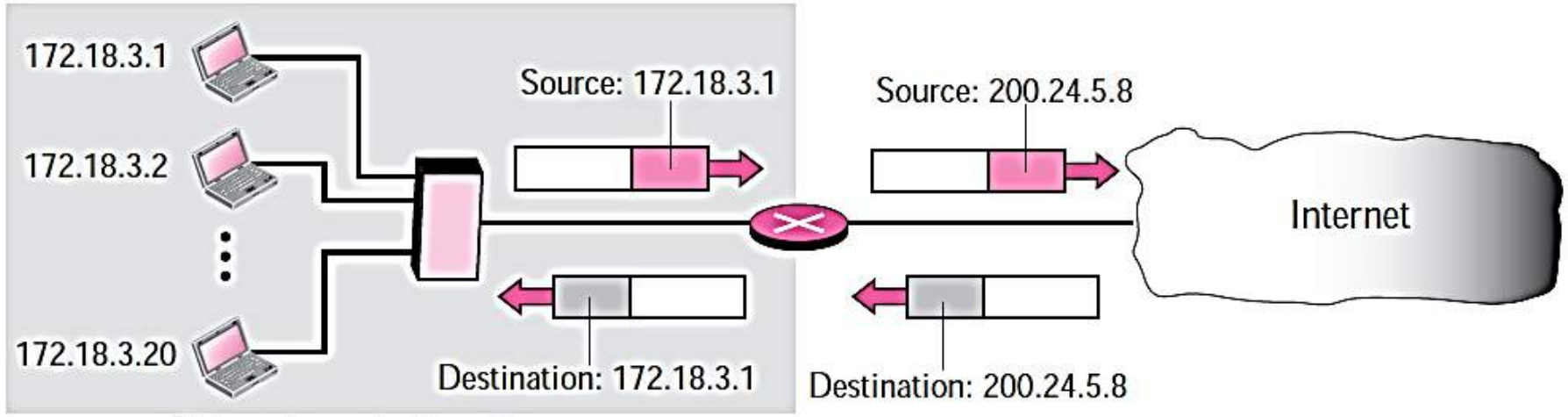
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Network Address Translation (NAT)

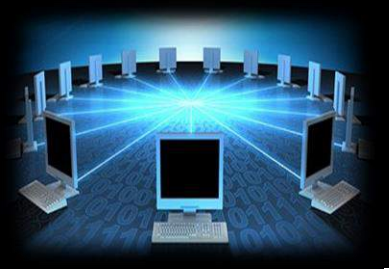
Address Translation

- All of the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.



Site using private addresses

Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Network Address Translation (NAT)

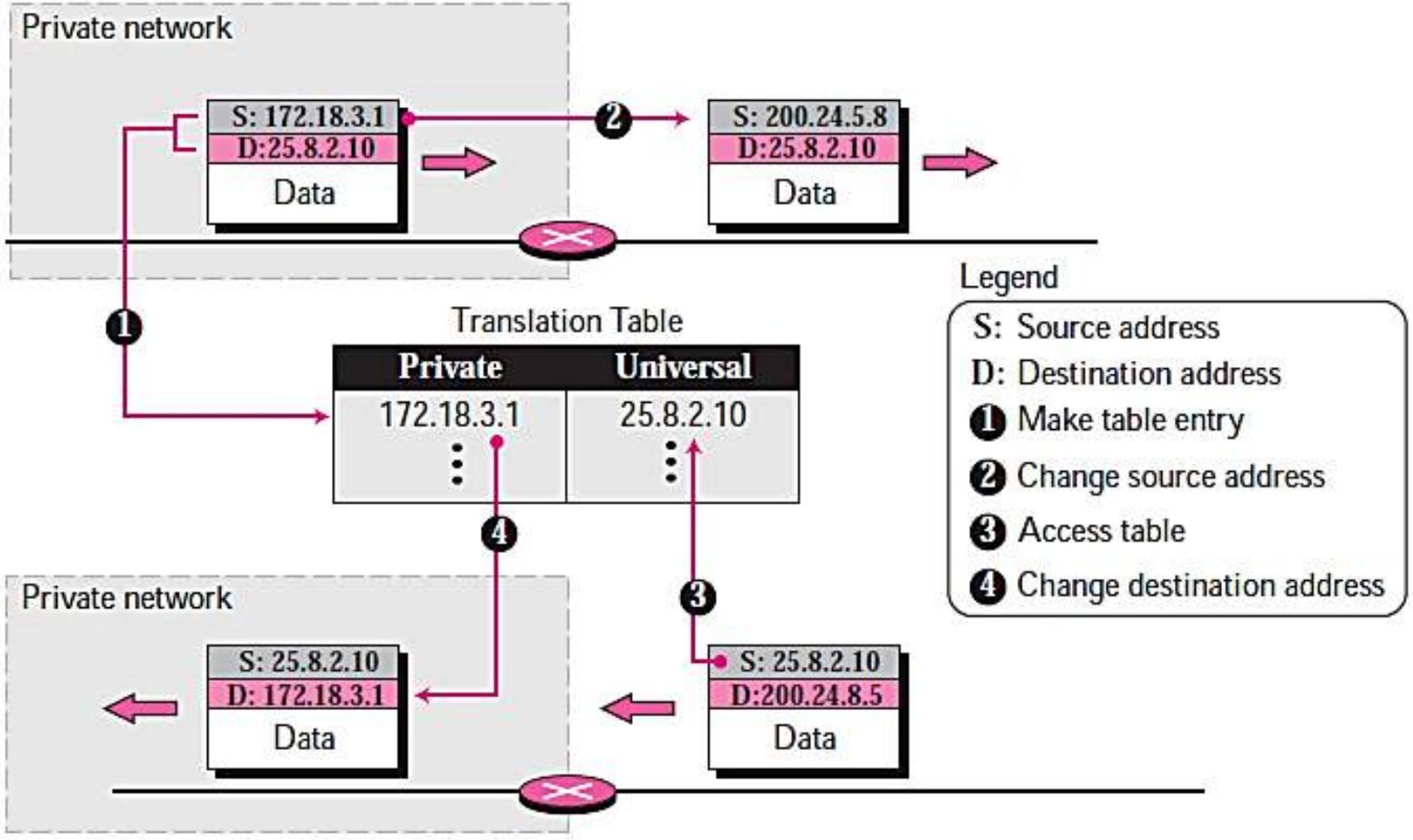
Translation Table

- **Using One IP Address**
  - In its simplest form, a translation table has only two columns: the private address and the external address (destination address of the packet).
  - When the router translates the source address of the outgoing packet, it also makes note of the destination address — where the packet is going.
  - When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.

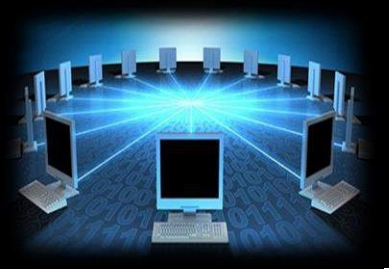
Network Address Translation (NAT)

Translation Table

- **Using One IP Address**



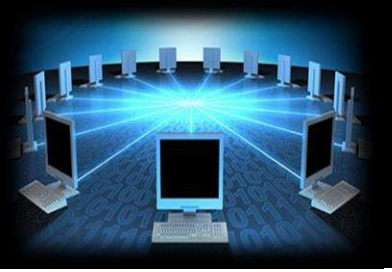
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Network Address Translation (NAT)

Translation Table

- **Using Both IP Addresses and Port Addresses**
  - To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table.
  - For example, suppose two hosts inside a private network with addresses 172.18.3.1 and 172.18.3.2 need to access the HTTP server on external host 25.8.3.2.
  - Translation table with one IP address will create ambiguity.
  - If the translation table has five columns, instead of two, that include the source and destination port addresses and the transport layer protocol, the ambiguity is eliminated.



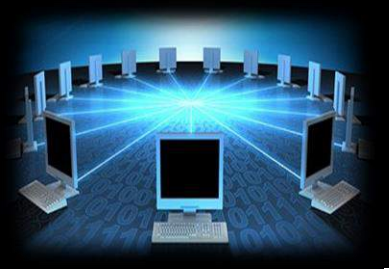
Network Address Translation (NAT)

Translation Table

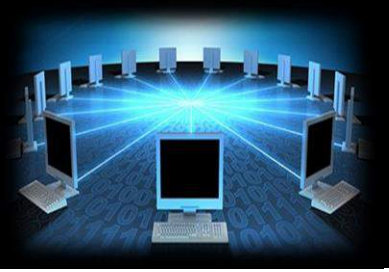
- **Using Both IP Addresses and Port Addresses**

Five-column translation table

Private Address	Private Port	External Address	External Port	Transport Protocol
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

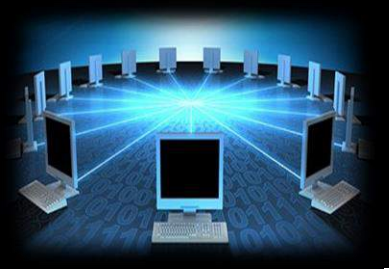


Routing Algorithms



Forwarding of IP Packets

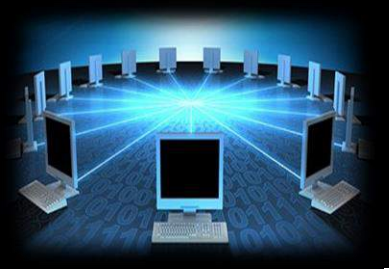
- Forwarding means to place the packet in its route to its destination.
- Since the Internet today is made of a combination of links (networks), forwarding means to deliver the packet to the next hop.
- Next hop can be the final destination or the intermediate connecting device.
- **Routing** is the process of moving packets from source to destination in a network.



Forwarding of IP Packets

Properties of Routing Algorithm

- Rapid and accurate delivery of packets.
- Adaptability to adapt changes in network topology due to failure of node/link.
- Adaptability to varying traffic conditions.
- Ability to route packets from temporarily congested links.
- Ability to determine the connectivity of the network.
- Ability to avoid routing loops.
- Low overheads.



Forwarding of IP Packets

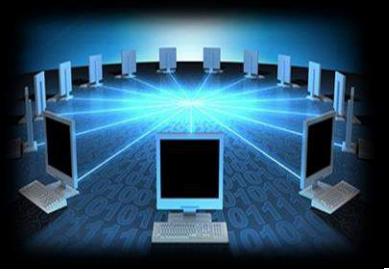
Routing Tables

- For Switch i.e. Virtual Circuit Packet Switching

Incoming		Outgoing	
Node	VCI	Node	VCI

- For Router i.e. Datagram Packet Switching

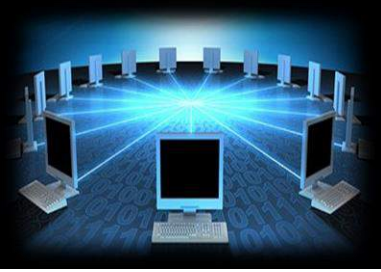
Destination	Next Node



Forwarding of IP Packets

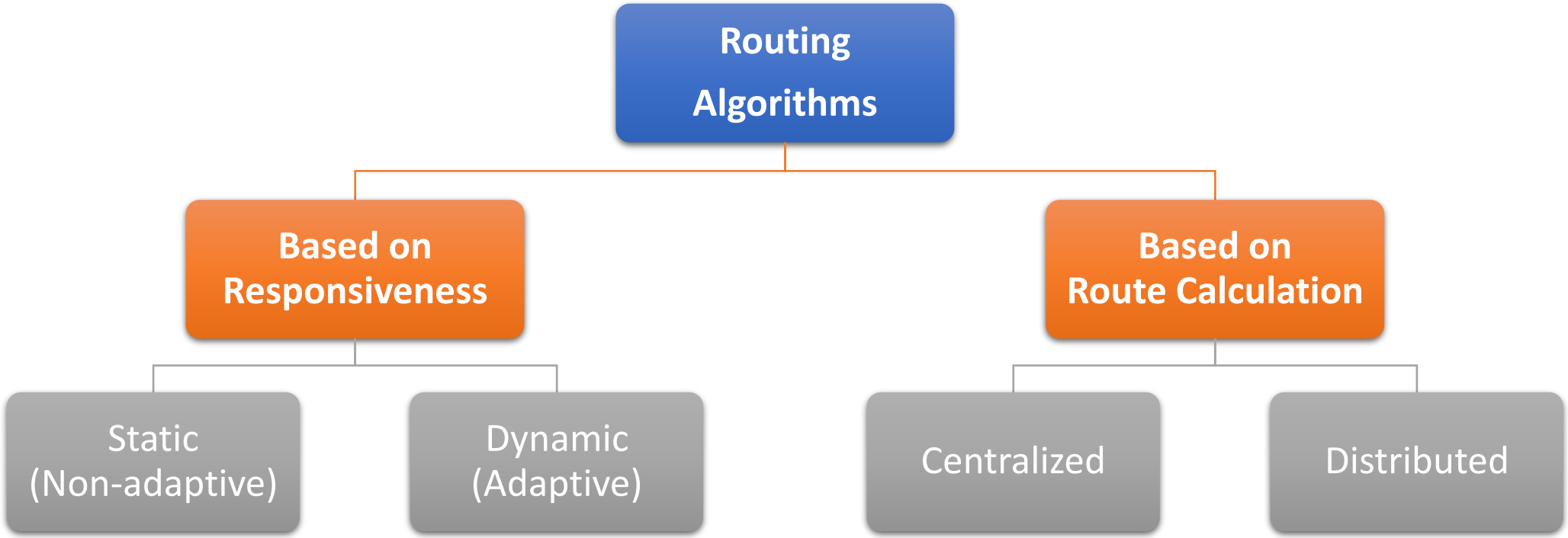
Metrics used to assign a cost to each link

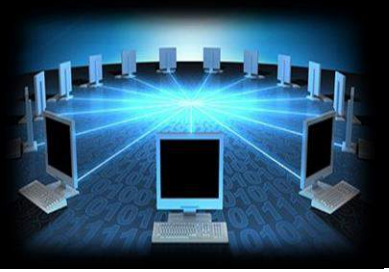
- **Cost ≈ 1/Capacity**
  - The cost is inversely proportional to the link capacity.
  - Higher cost is assigned to lower capacity links to allow packets to take a path with highest capacity link.
  - If each link has equal capacity, then path with minimum hops is chosen as shortest path.
- **Cost ≈ Packet Delay**
  - The cost is proportional to average packet delay.
  - The shortest path represents the fastest path to reach destination.
- **Cost ≈ Congestion**
  - The cost is proportional to congestion (traffic loading).
  - The shortest path tries to avoid congested link.



Routing

❑ Classification of Routing



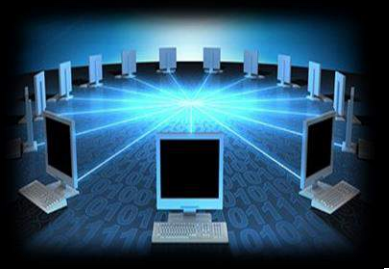


Routing

❑ Classification of Routing

Static Routing	Dynamic Routing
Path once computed remains fixed regardless traffic.	Path is computed depending on traffic conditions.
Less adaptive to network changes.	Better adaptability to network changes.
Less complex.	More complex.

Centralized Routing	Distributed Routing
One node calculates all paths and inform others.	Each node calculates next hop.
Less adaptive to network changes.	More adaptive to network changes.
Less probability of producing loops.	More probability of producing loops.

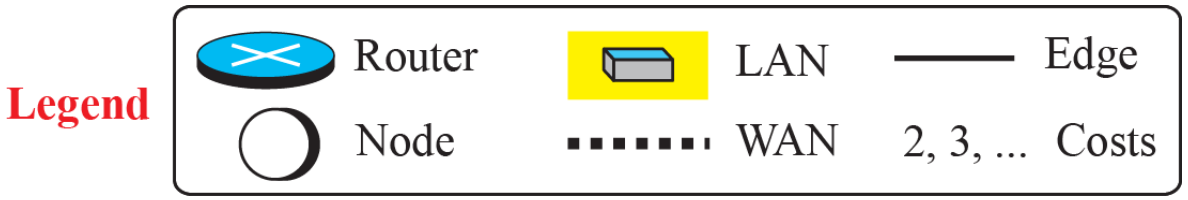


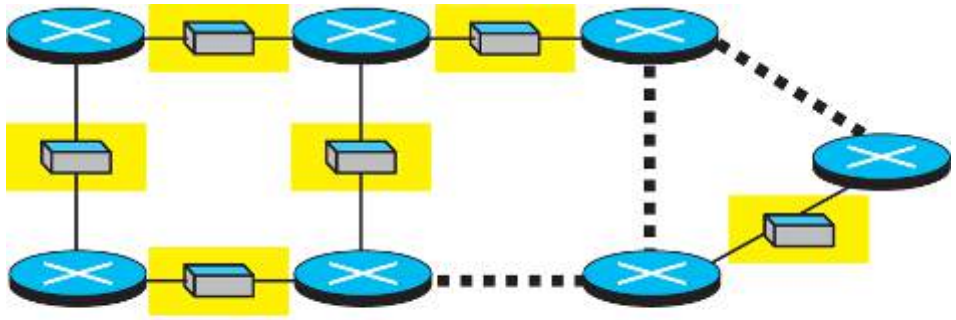
Routing

❑ Distance Vector Routing v/s Link State Routing

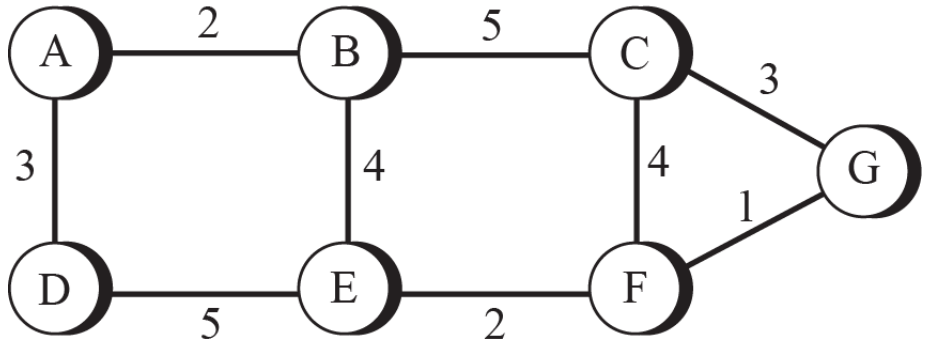
- **Distance Vector Routing**
  - Neighbouring routers exchange routing tables that state the set or vector of known distance.
  - After neighbouring routers exchange this information, they process it using **Bellman-Ford Algorithm**.
  - Slow convergence and less flexible in case of network changes.
- **Link State Routing**
  - Each router floods information about the state of links that connect it to its neighbours.
  - This process allows each router to construct a map of entire network.
  - From this map, it derives shortest path using **Dijkstra's Algorithm**.
  - Fast convergence and more flexible in case of network changes.

An internet and its graphical representation



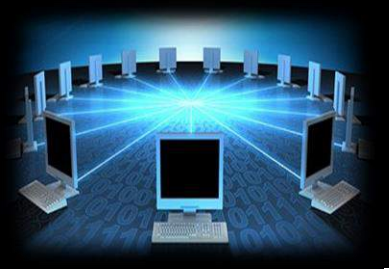


a. An internet



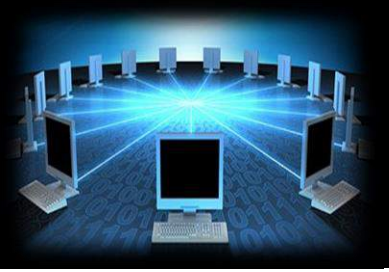
b. The weighted graph

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Bellman-Ford Algorithm

- If we know the cost between each pair of nodes, we can use Bellman-Ford Algorithm to find the least cost (shortest path) between any two nodes.
- The algorithm is based on the fact that if all neighbors of node i know the shortest distance to node j, then the shortest distance between node i and j can be found by adding the distance between node i and each neighbor to the neighbor's shortest distance to node j and then select the minimum.

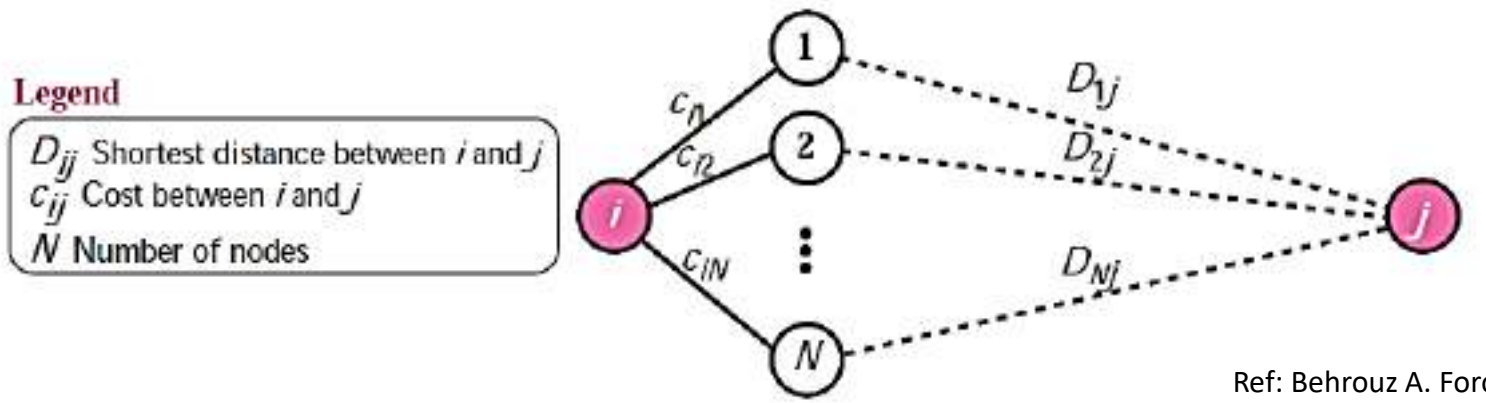


Bellman-Ford Algorithm

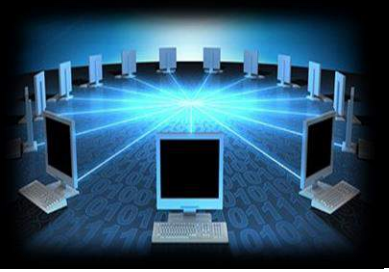
Steps to find the shortest path

1. The shortest distance and the cost between a node and itself is initialized to 0.
2. The shortest distance between a node and any other node is set to infinity. The cost between a node and any other node should be given (can be infinity if the nodes are not connected).
3. The algorithm repeats as shown below until there is no more change in the shortest distance vector.

$$D_{ij} = \text{minimum } \{(c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \ldots (c_{iN} + D_{Nj})\}$$



Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

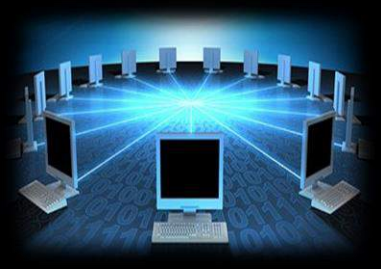


Bellman-Ford Algorithm

Pseudocode

```
1  Bellman_Ford ( )
2  {
3     // Initialization
4     for (i = 1 to N; for j = 1 to N)
5     {
6        if(i == j)   D_ij = 0   c_ij = 0
7        else         D_ij = ∞   c_ij = cost between i and j
8     }
9     // Updating
10    repeat
11    {
12       for (i = 1 to N; for j = 1 to N)
13       {
14          D_ij ← minimum [(c_i1 + D_1j)  ...  (c_iN + D_Nj)]
15       } // end for
16    } until (there was no change in previous iteration)
17 } // end Bellman-Ford
```

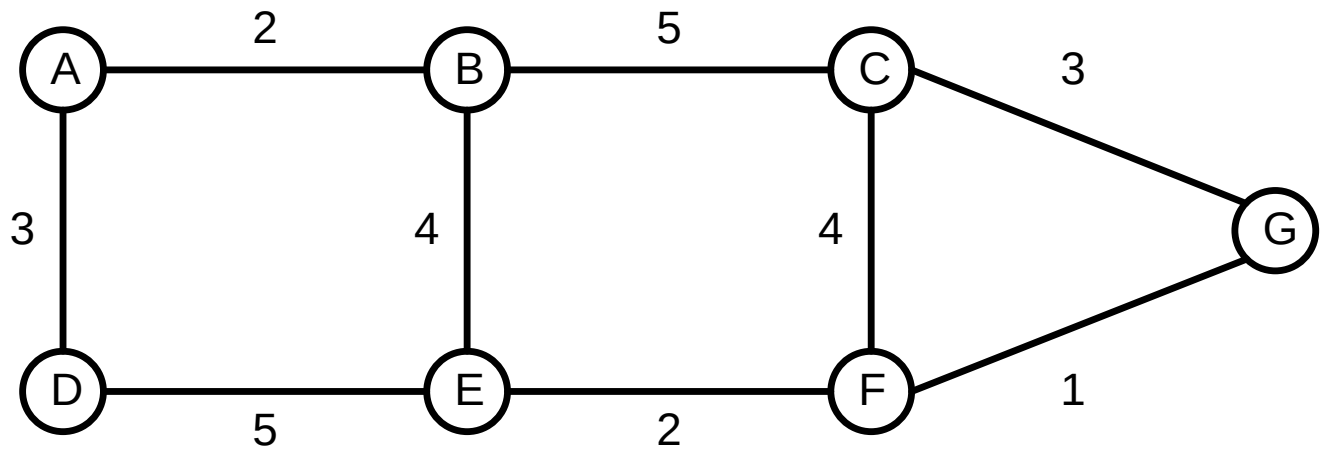
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

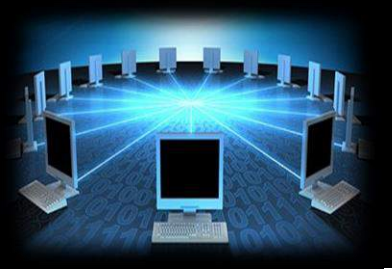


Bellman-Ford Algorithm

Example:

Find the shortest path to every node from node A



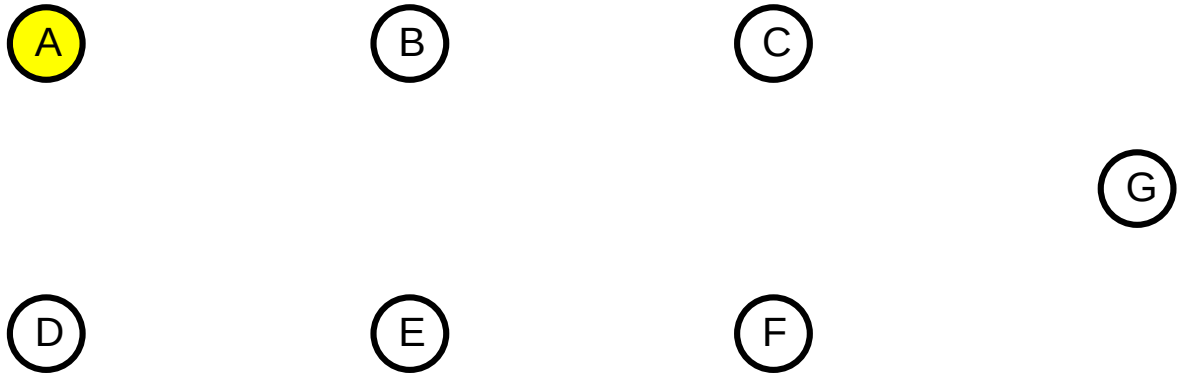


Bellman-Ford Algorithm

Example:

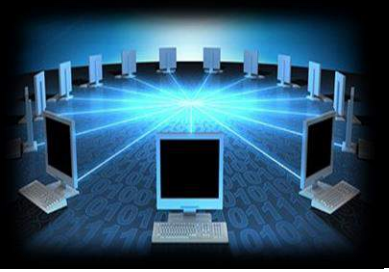
Find the shortest path to every node from node A

Initialization



Processing Table

Iteration	Previous Hop & distance to reach node					
	B	C	D	E	F	G
0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞

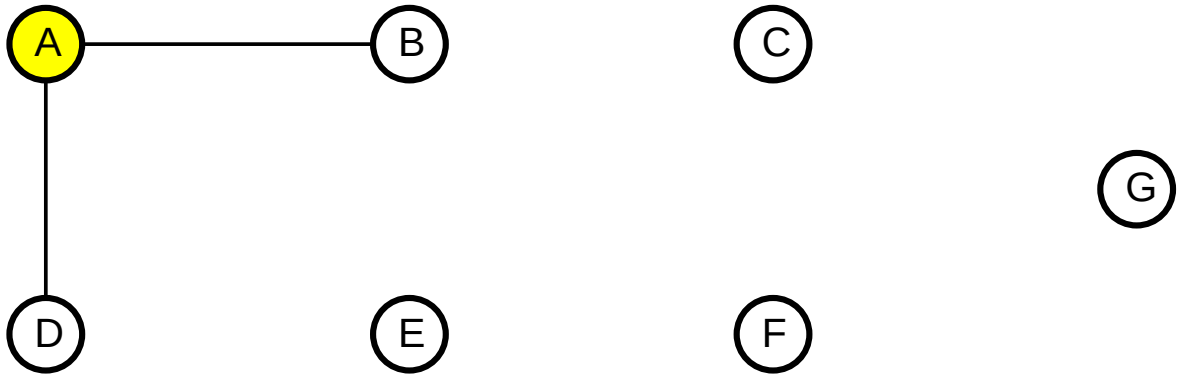


Bellman-Ford Algorithm

Example:

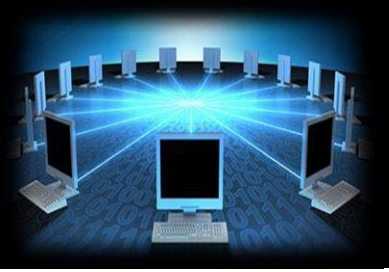
Find the shortest path to every node from node A

Iteration 1: Using 1 arc



Processing Table

Iteration	Previous Hop & distance to reach node					
	B	C	D	E	F	G
0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
1	A, 2	-1, ∞	A, 3	-1, ∞	-1, ∞	-1, ∞

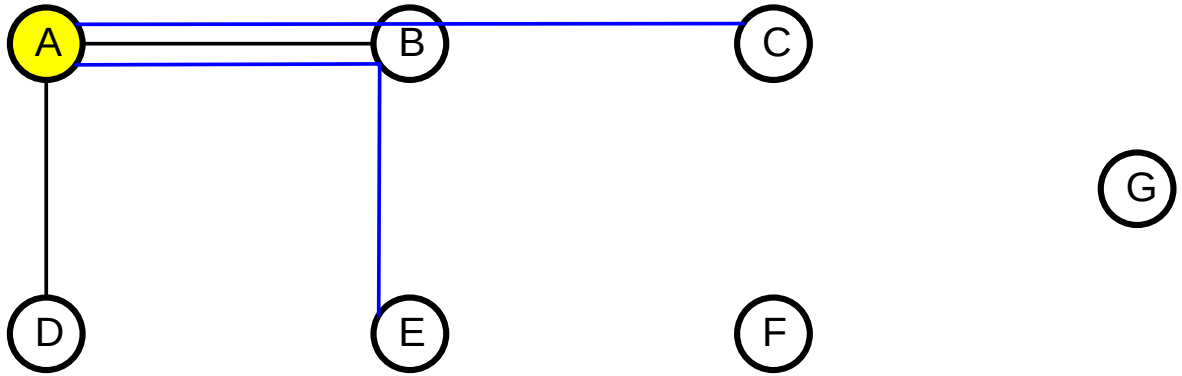


Bellman-Ford Algorithm

Example:

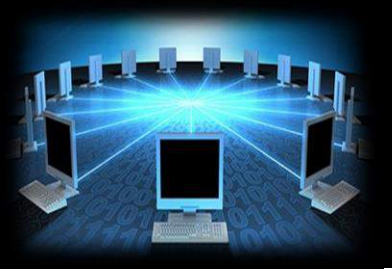
Find the shortest path to every node from node A

Iteration 2: Using 2 arc



Processing Table

Iteration	Previous Hop & distance to reach node					
	B	C	D	E	F	G
0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
1	A, 2	-1, ∞	A, 3	-1, ∞	-1, ∞	-1, ∞
2	A, 2	B, 7	A, 3	B, 6	-1, ∞	-1, ∞

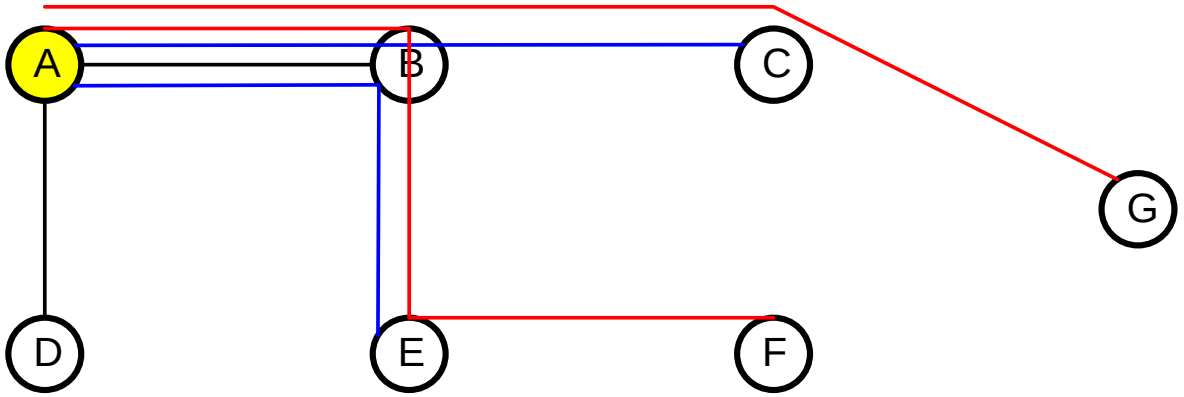


Bellman-Ford Algorithm

Example:

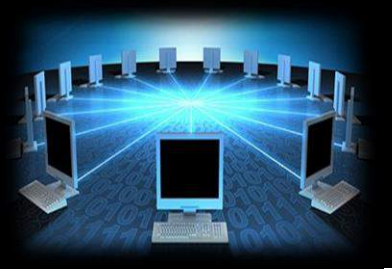
Find the shortest path to every node from node A

Iteration 3: Using 3 arc



Processing Table

Iteration	Previous Hop & distance to reach node					
	B	C	D	E	F	G
0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
1	A, 2	-1, ∞	A, 3	-1, ∞	-1, ∞	-1, ∞
2	A, 2	B, 7	A, 3	B, 6	-1, ∞	-1, ∞
3	A, 2	B, 7	A, 3	B, 6	E, 8	C, 10

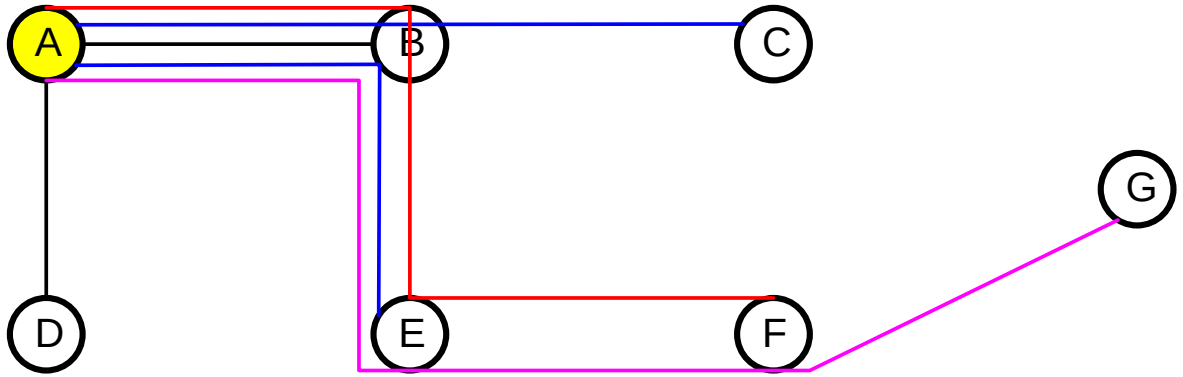


Bellman-Ford Algorithm

Example:

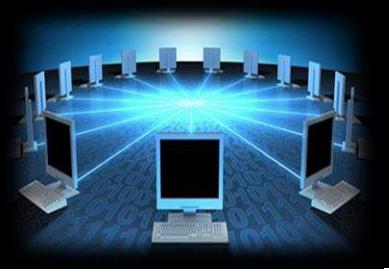
Find the shortest path to every node from node A

Iteration 4: Using 4 arc



Processing Table

Iteration	Previous Hop & distance to reach node					
	B	C	D	E	F	G
0	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞	-1, ∞
1	A, 2	-1, ∞	A, 3	-1, ∞	-1, ∞	-1, ∞
2	A, 2	B, 7	A, 3	B, 6	-1, ∞	-1, ∞
3	A, 2	B, 7	A, 3	B, 6	E, 8	C, 10
4	A, 2	B, 7	A, 3	B, 6	E, 8	F, 9

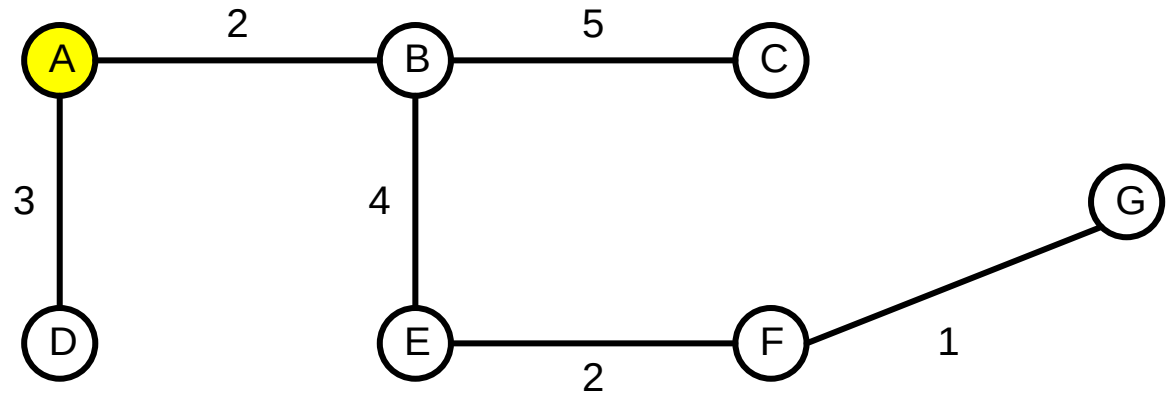


Bellman-Ford Algorithm

Example:

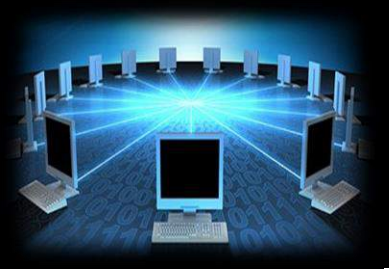
Find the shortest path to every node from node A

Final Tree



Routing Table at A

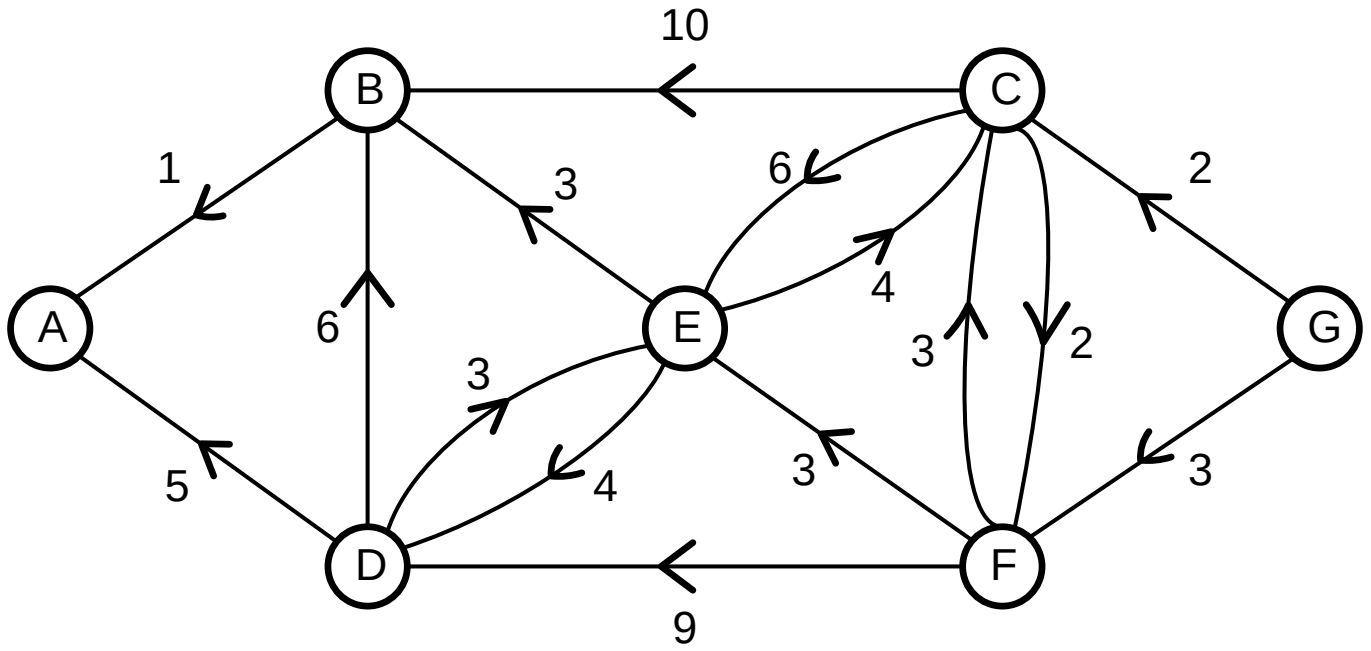
Destination	Cost	Next Hop
B	2	B
C	7	B
D	3	D
E	6	B
F	8	B
G	9	B

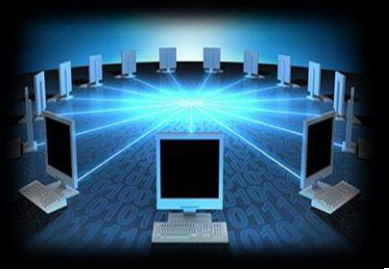


Bellman-Ford Algorithm

Example:

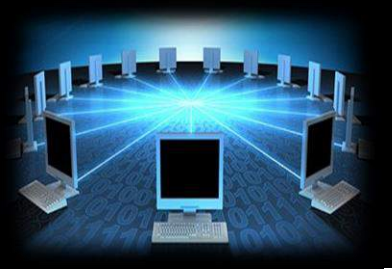
Find the shortest path from every node to node A





Dijkstra's Algorithm

- Link state routing has a different philosophy from that of distance vector routing.
- In link state routing, if each node in the domain has the entire topology of the domain—the list of nodes and links, how they are connected including the type, cost (metric), and the condition of the links (up or down).
- The node can use the **Dijkstra's algorithm** to build a routing table.
- In Dijkstra's algorithm, all other nodes can be reached from the root through only one single route.
- A shortest path tree is a tree in which the path between the root and every other node is the shortest.

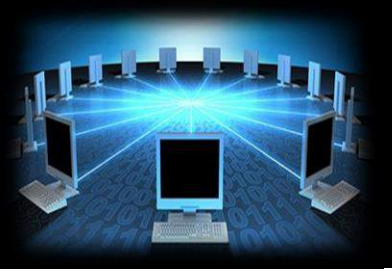


Dijkstra's Algorithm

Steps to find the shortest path

1. **Initialization:** Select the node as the root of the tree and add it to the path. Set the shortest distances for all the root's neighbors to the cost between the root and those neighbors. Set the shortest distance of the root to zero.
2. **Iteration:** Repeat the following two steps until all nodes are added to the path:
   a. **Adding the next node to the path:** Search the nodes not in the path. Select the one with minimum shortest distance and add it to the path.
   b. **Updating:** Update the shortest distance for all remaining nodes using the shortest distance of the node just moved to the path in step 2.

$$D_j = minimum\left(D_j, D_i + C_{ij} \right) \quad for\ all\ remaining\ nodes$$

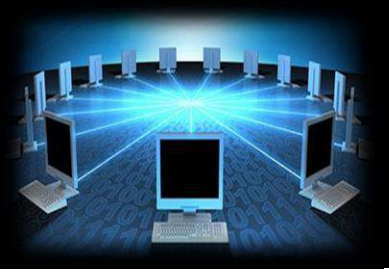


Dijkstra's Algorithm

Pseudocode

```
1  Dijkstra ( )
2  {
3     // Initialization
4     Path = {s}              // s means self
5     for (i = 1 to N)
6     {
7        if (i is a neighbor of s and i ≠ s)    D_i = c_si
8        if (i is not a neighbor of s)          D_i = ∞
9     }
10    D_s = 0
11
12 } // Dijkstra
13    // Iteration
14    Repeat
15    {
16       // Finding the next node to be added
17    Path = Path ∪ i   if D_i is minimum among all remaining nodes
18
19       // Update the shortest distance for the rest
20    for (j = 1 to M)     // M number of remaining nodes
21    {
22          D_j = minimum (D_j , D_j + c_ij)
23     }
24    } until (all nodes included in the path, M = 0)
25
```

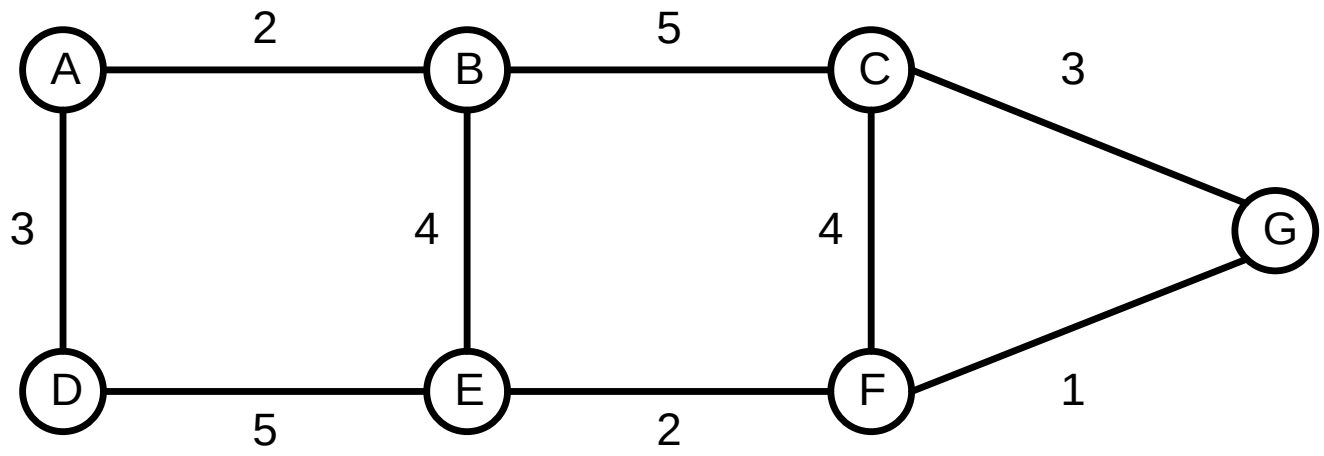
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



Dijkstra's Algorithm

Example:

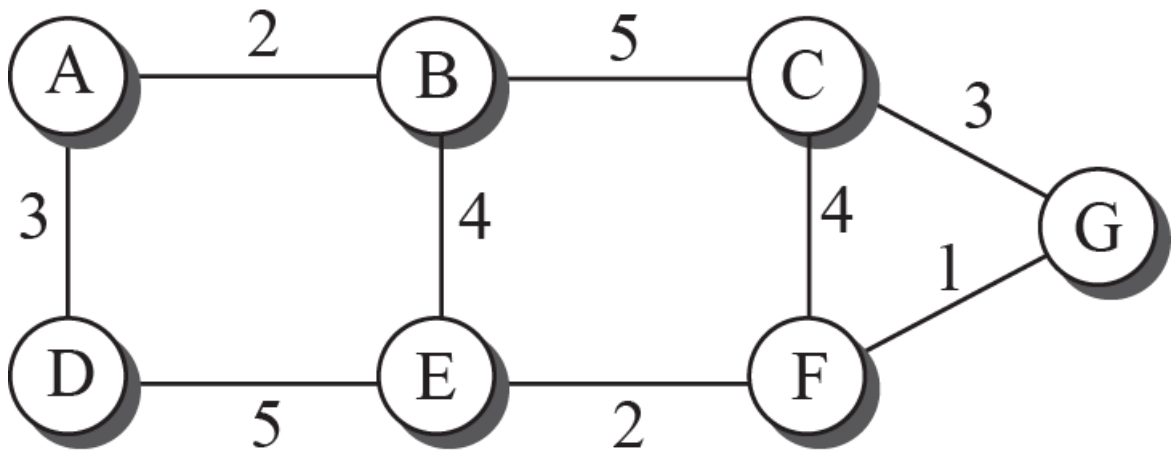
Find the shortest path to every node from node A



Dijkstra's Algorithm

Example:

Find the shortest path to every node from node A



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

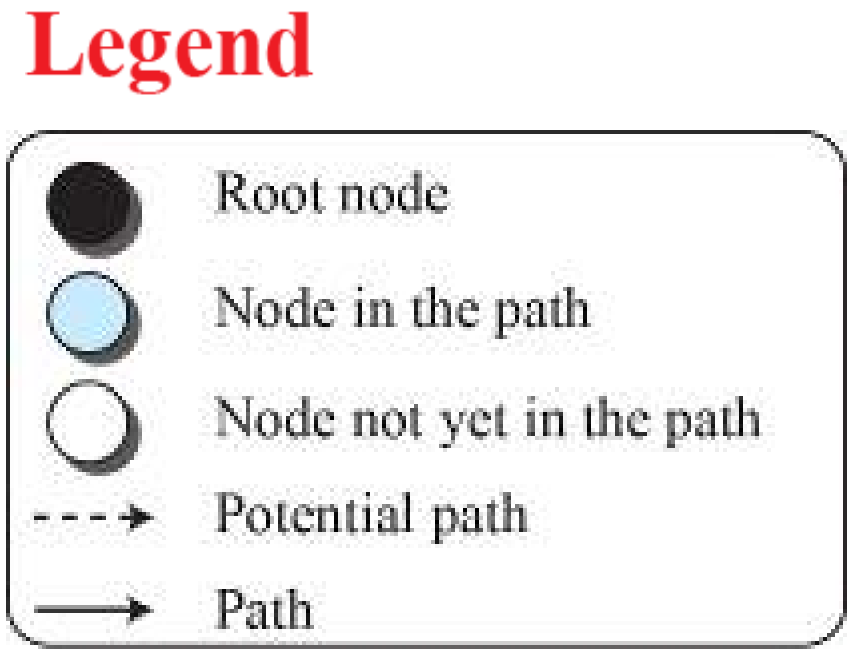
b. Link state database

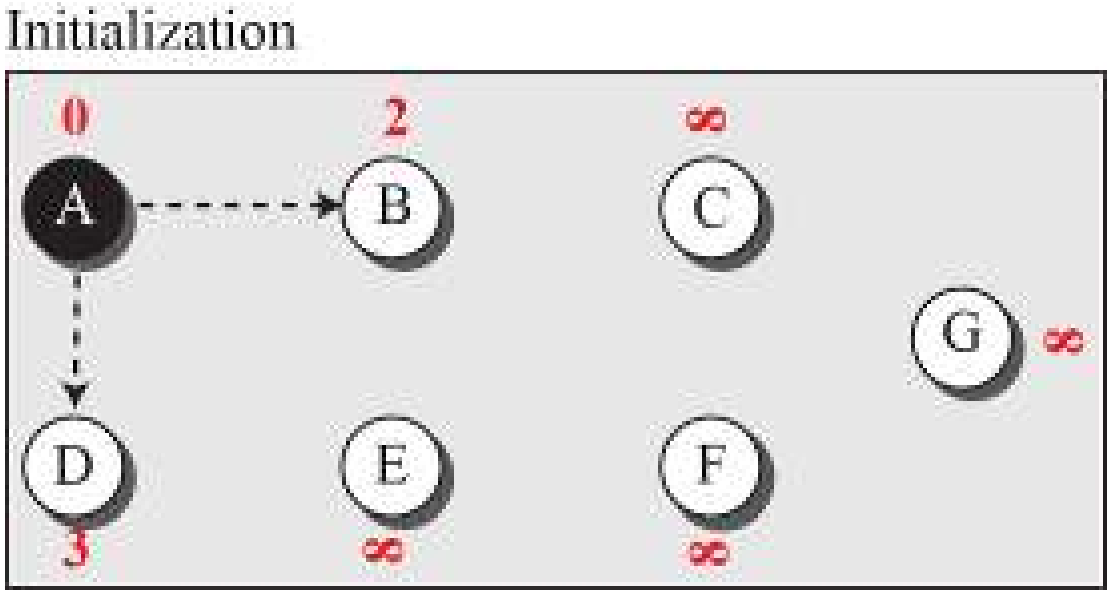
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

Dijkstra's Algorithm

Example:

Find the shortest path to every node from node A





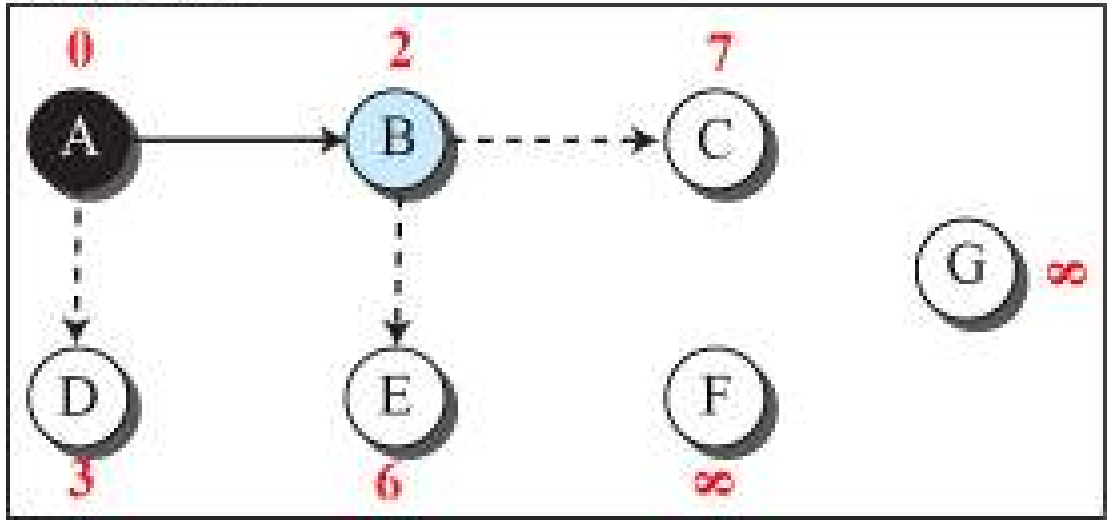
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

Dijkstra's Algorithm

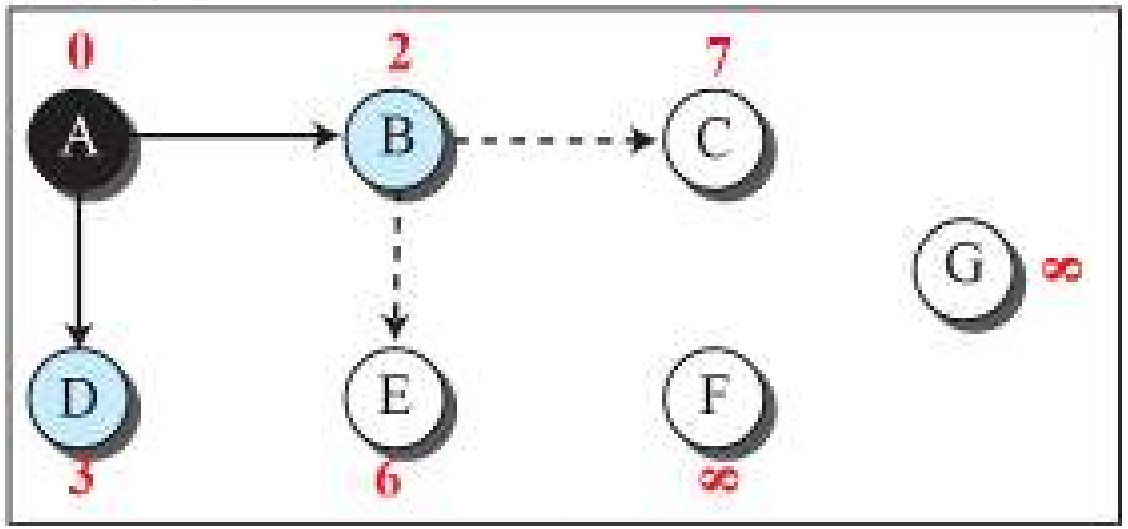
Example:

Find the shortest path to every node from node A

Iteration 1



Iteration 2



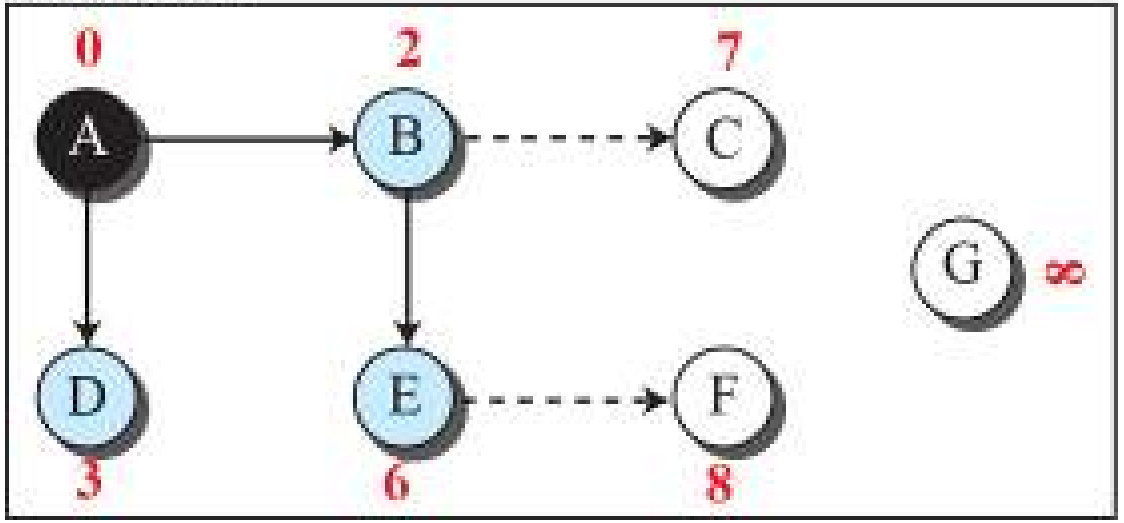
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

Dijkstra's Algorithm

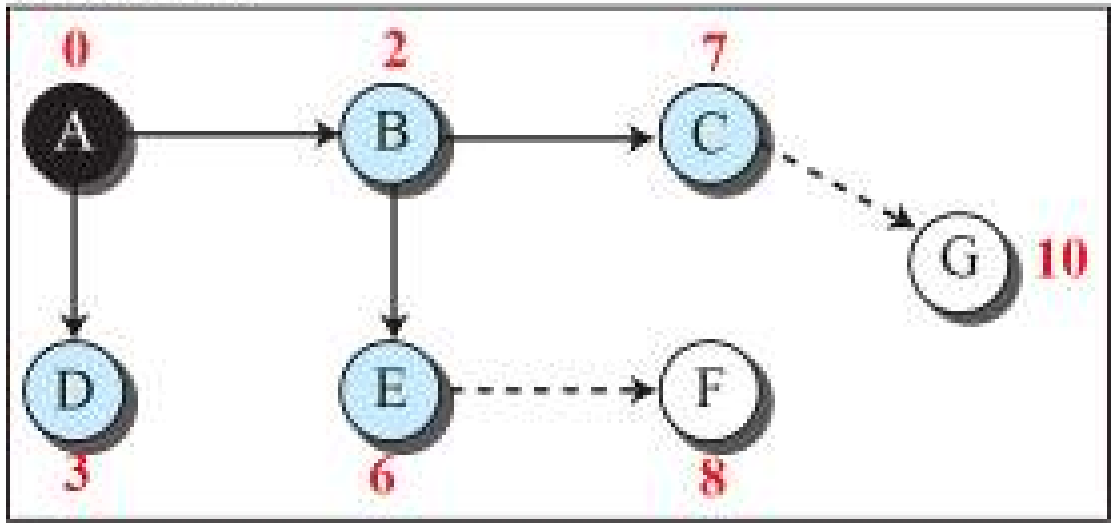
Example:

Find the shortest path to every node from node A

Iteration 3



Iteration 4



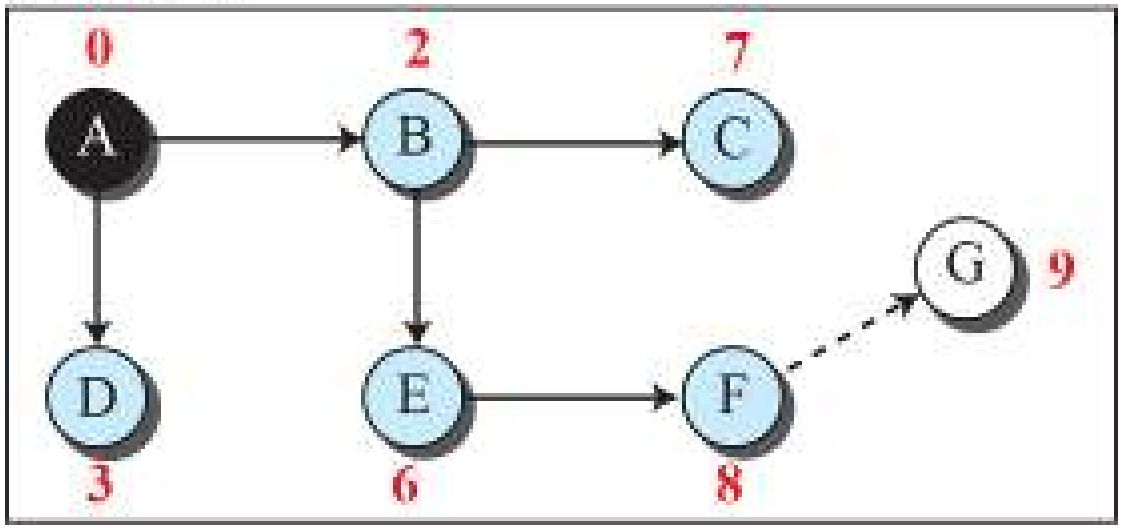
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

Dijkstra's Algorithm

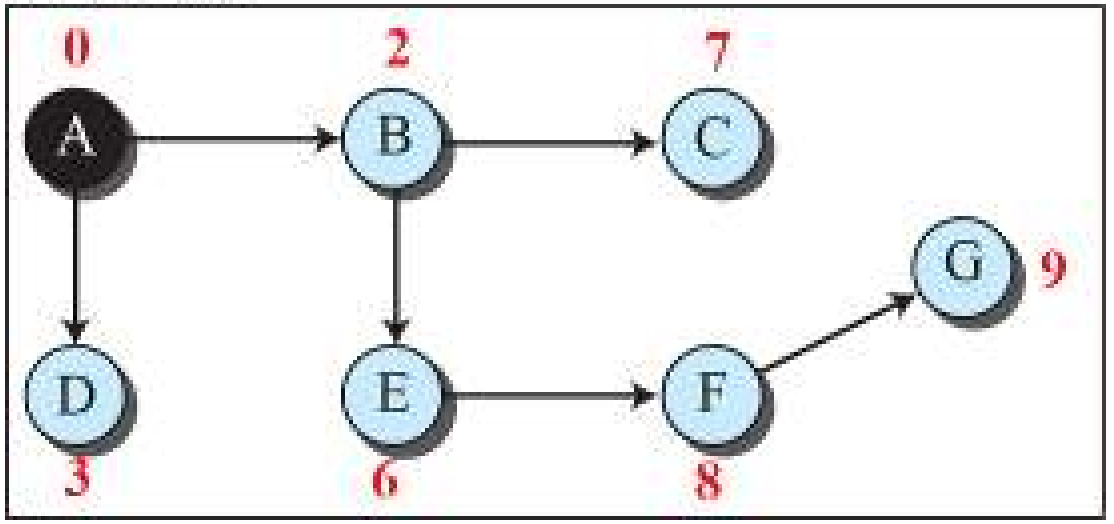
Example:

Find the shortest path to every node from node A

Iteration 5

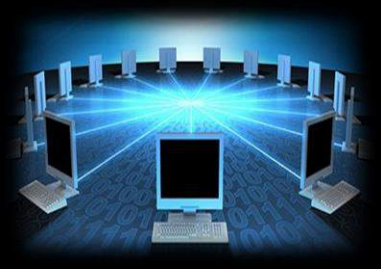


Iteration 6



No. of Iterations = No. of nodes - 1

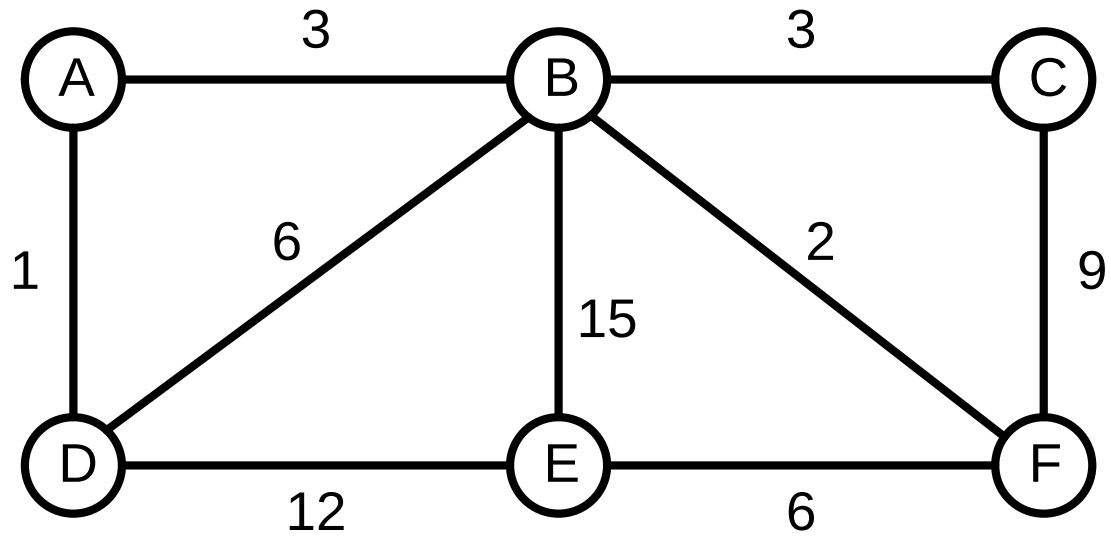
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.

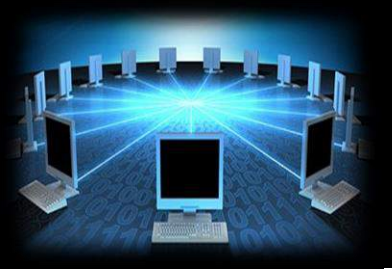


Dijkstra's Algorithm

Example:

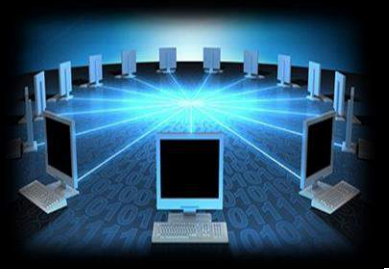
Find the shortest path to every node from node A





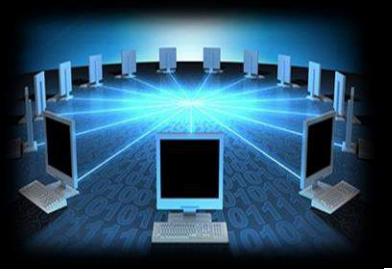
Distance Vector Routing v/s Link State Routing

Distance Vector Routing	Link State Routing
No flooding, small packets and local sharing require less bandwidth.	More bandwidth required to facilitate flooding and sending large link state packets.
Uses Bellman-Ford algorithm.	Uses Dijkstra's algorithm.
Less traffic.	More network traffic when compared to Distance Vector Routing.
Updates table based on information from neighbours, thus uses local knowledge.	It has knowledge about the entire network, thus it uses global knowledge.
Persistent looping problem exists.	Only transient loop problems.
Based on least hops.	Based on least cost.
Updation of full routing tables.	Updation of only link states.
Less CPU utilisation.	High CPU utilisation.
Uses broadcast for updates.	Uses multicast for updates.
Moderate convergence time.	Low convergence time.



Path-Vector Routing

- Distance-vector and link-state routing are based on the least-cost routing.
- However, there are some instances where least cost is not the priority.
- For e.g. assume that there are some routers in the internet that a sender wants to prevent its packets form going through. Least-cost routing does not prevent a packet from passing through an area when that area is in the least-cost path.
- To handle such situation, a third routing algorithm called **Path-vector routing** is used.
- Path-vector routing is not based on least-cost routing.
- The best route is determined by the source using the policy it imposes on the route i.e. source can control the path.

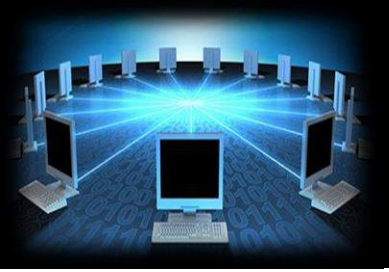


Path-Vector Algorithm

Spanning Tree

- The path from source to destinations is determined by the best spanning tree.
- It is determined by the policy imposed by the source. Some of the policies are –
  - Minimum no. of nodes to be visited.
  - Avoid some nodes as the middle node in a route.
- Spanning tree is updated using following equation

$$Path(x, y) = best\{Path(x, y), [x + path(v, y)]\}$$



Path-Vector Algorithm

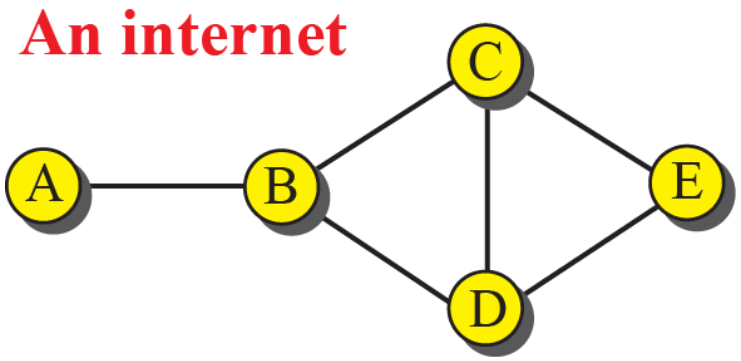
- **Pseudocode**

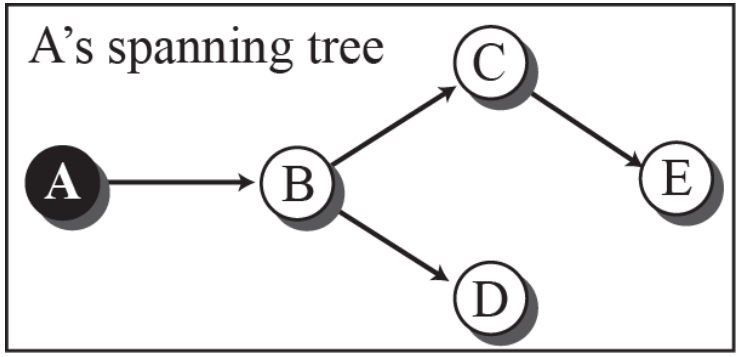
```
1  Path_Vector_Routing ( )
2  {
3      // Initialization
4      for (y = 1 to N)
5      {
6          if (y is myself)
7              Path[y] = myself
8          else if (y is a neighbor)
9              Path[y] = myself + neighbor node
10         else
11             Path[y] = empty
12     }
13     Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
14     // Update
15     repeat (forever)
16     {
17         wait (for a vector Path_w from a neighbor w)
18         for (y = 1 to N)
19         {
20             if (Path_w includes myself)
21                 discard the path                    // Avoid any loop
22             else
23                 Path[y] = best {Path[y], (myself + Path_w[y])}
24         }
25         If (there is a change in the vector)
26             Send vector {Path[1], Path[2], ..., Path[y]} to all neighbors
27     }
28 } // End of Path Vector
```

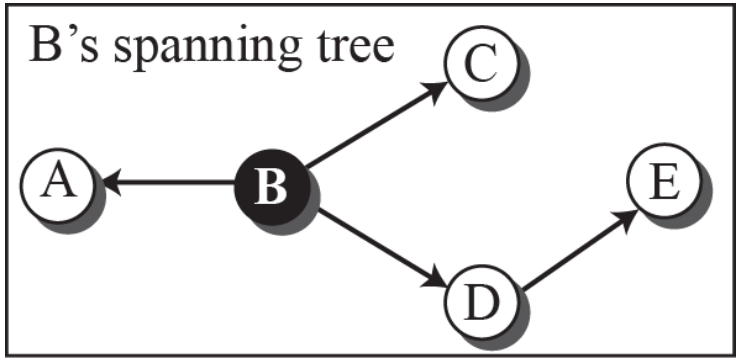
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

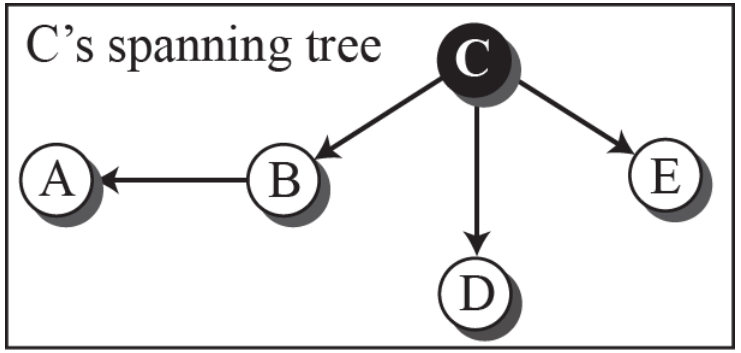
Path-Vector Algorithm

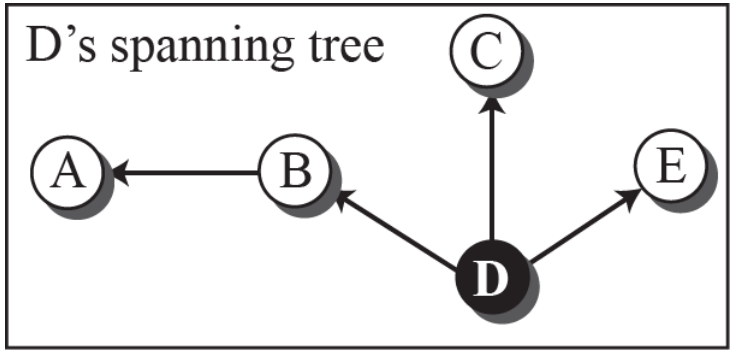
Example:

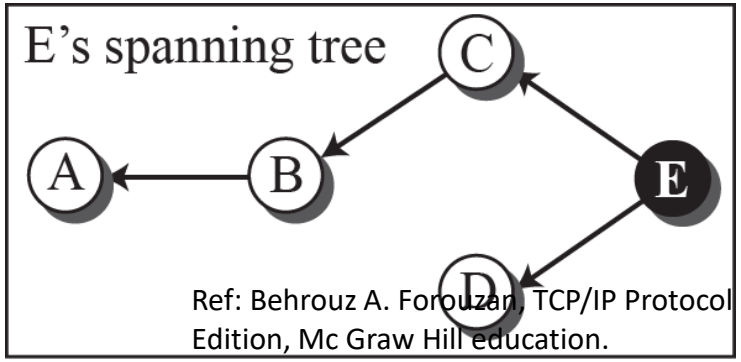






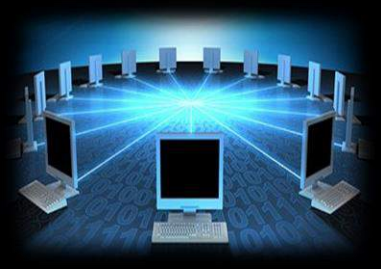






Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

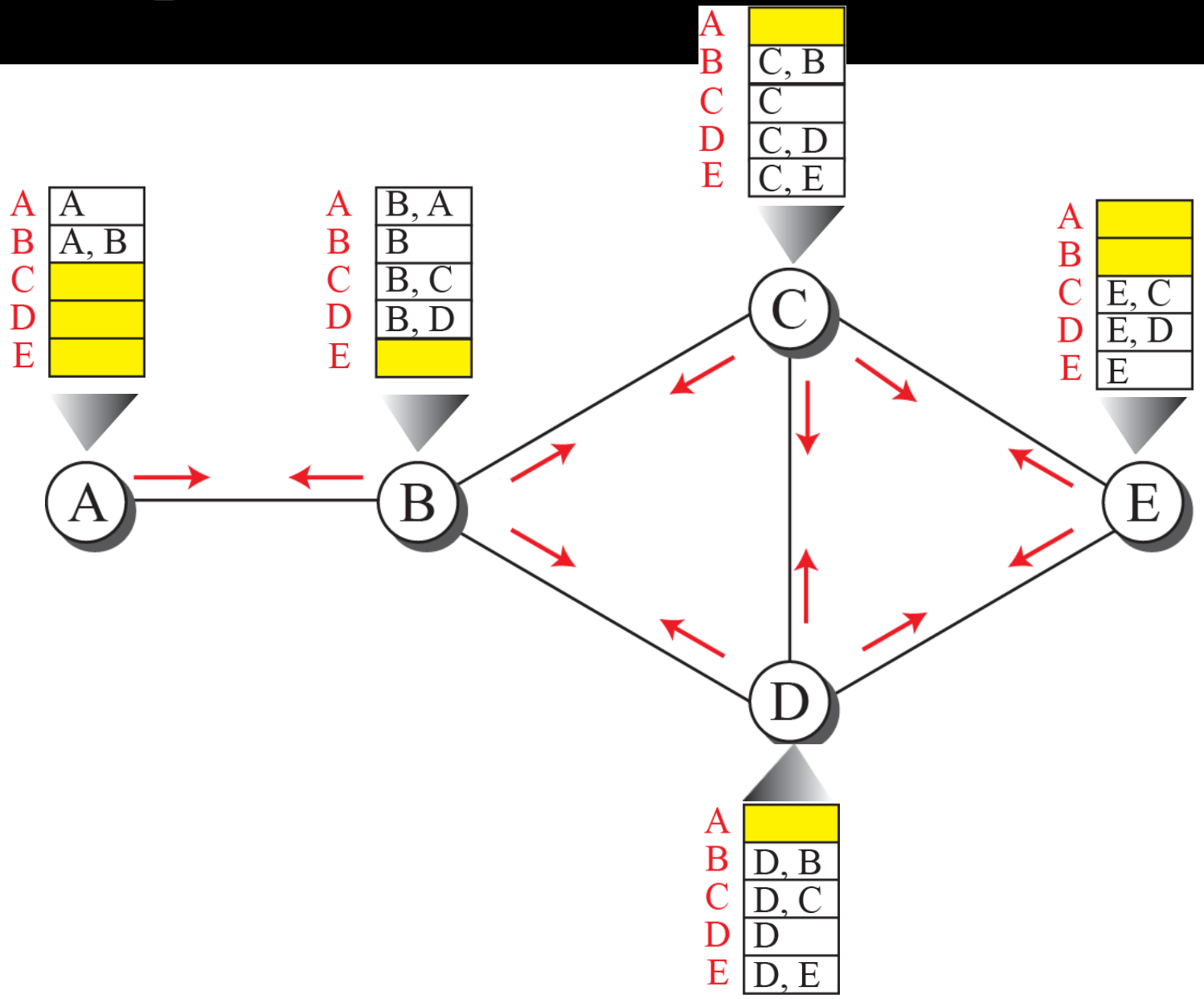
Ref: Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Mc Graw Hill education.



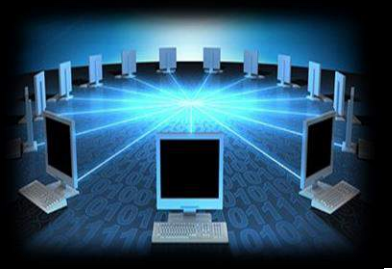
Path-Vector Algorithm

Example:

Path vectors made at booting time



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Path-Vector Algorithm

Example:

Updating path vectors

Note:
X [ ]: vector X
Y: node Y

	New C	Old C	B
A	C, B, A		B, A
B	C, B	C, B	B
C	C	C	B, C
D	C, D	C, D	B, D
E	C, E	C, E	

C[ ] = best (C[ ], C + B[ ])

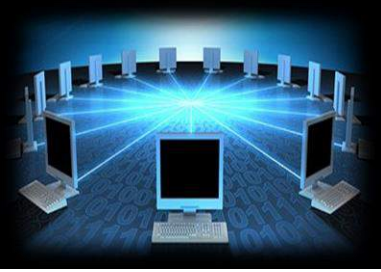
Event 1: C receives a copy of B's vector

	New C	Old C	D
A	C, B, A	C, B, A	
B	C, B	C, B	D, B
C	C	C	D, C
D	C, D	C, D	D
E	C, E	C, E	D, E

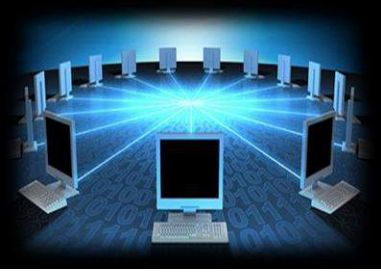
C[ ] = best (C[ ], C + D[ ])

Event 2: C receives a copy of D's vector

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



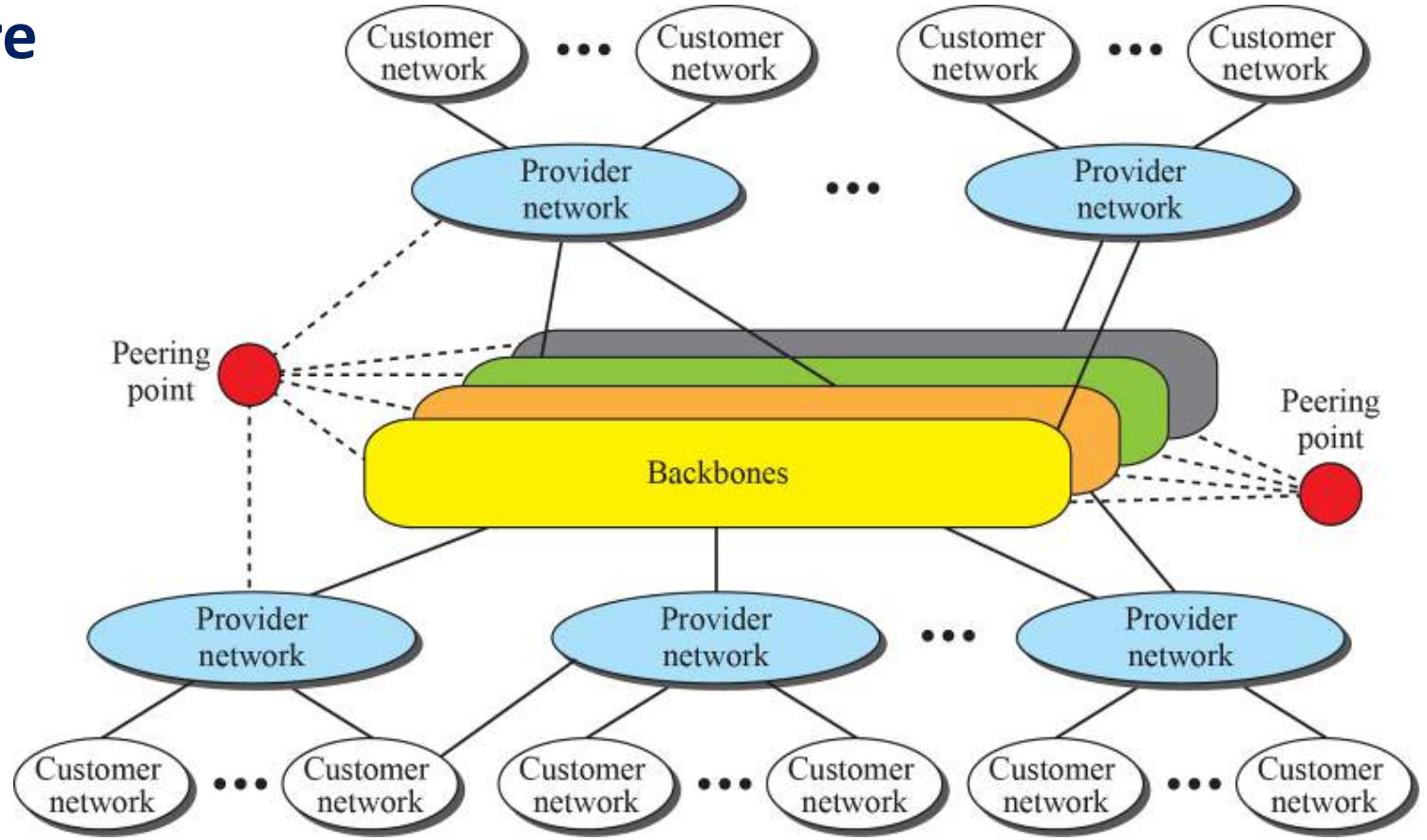
Routing Protocols



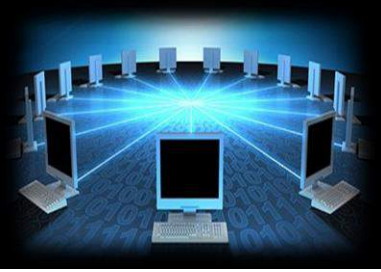
Routing Protocols

- A routing protocol needs to define its domain of operation, the messages exchanged, communication between routers, and interaction with protocols in other domains.

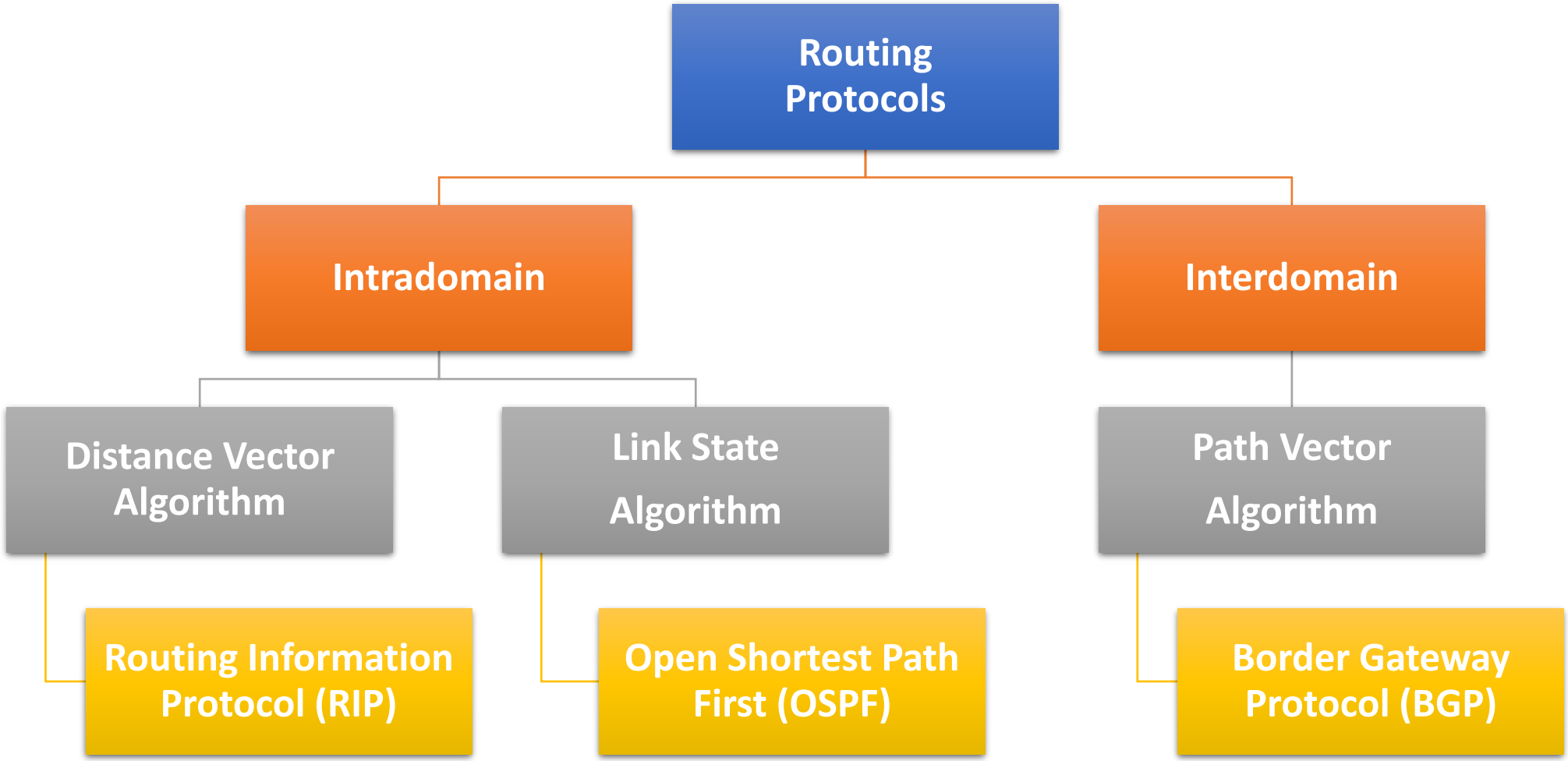
❑ Internet structure

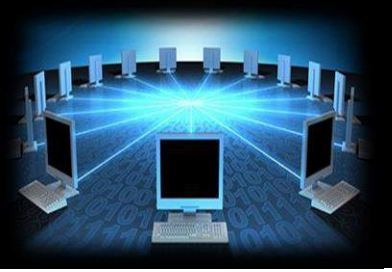


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Routing Protocols



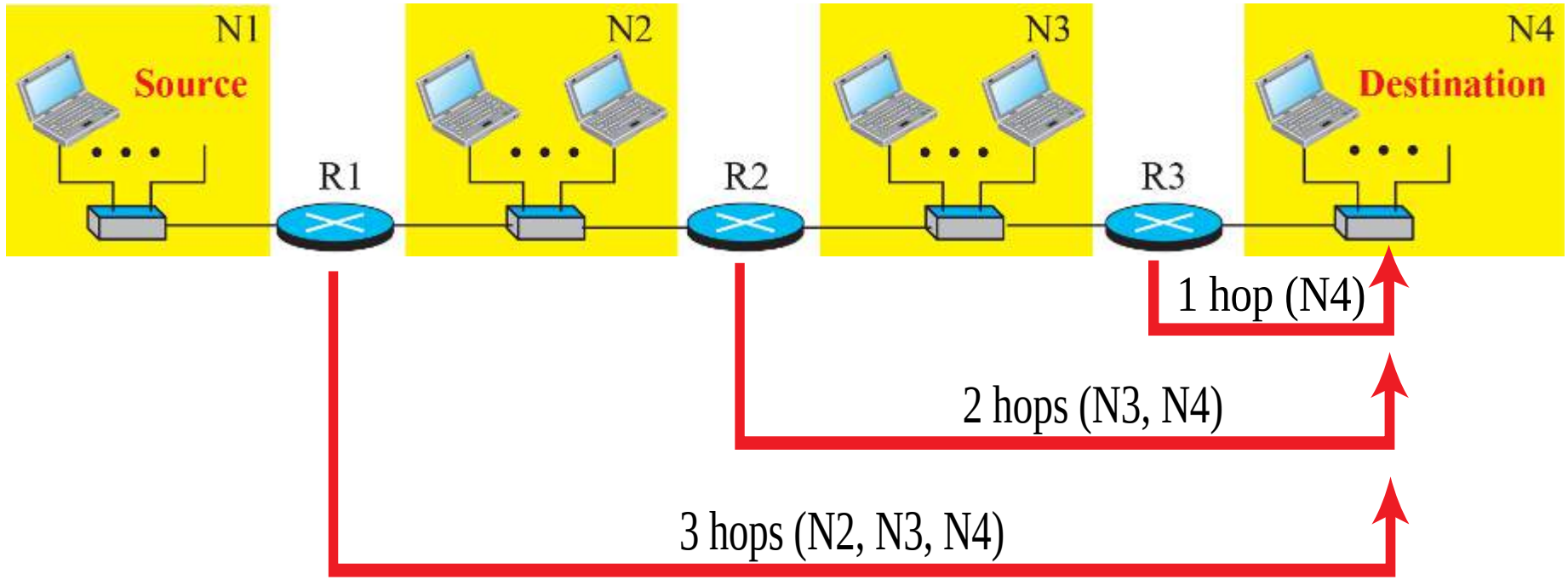


Routing Information Protocol (RIP)

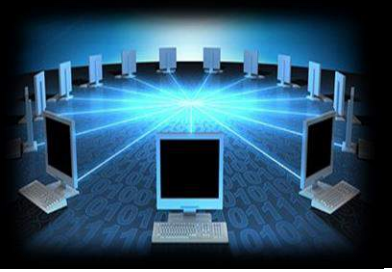
- The Routing Information Protocol (RIP) is an **intradomain (interior) routing protocol** used inside an **autonomous system (AS)**.
- It is a very simple protocol based on **distance vector routing**.
- RIP implements distance vector routing directly with some considerations:
  1. In an autonomous system, we are dealing with routers (nodes) and networks (links).
  2. The destination in a routing table is a network, which means the first column defines a network address.
  3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) that have to be used to reach the destination. For this reason, the metric in RIP is called a **hop count**.
  4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
  5. The next node column defines the address of the router to which the packet is to be sent to reach its destination.

Routing Information Protocol (RIP)

❑ Hop Count



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Routing Information Protocol (RIP)

Forwarding Tables

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

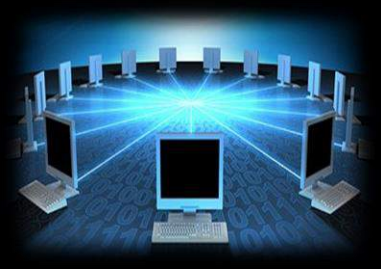
Forwarding table for R2

Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	——	1
N4	——	1

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



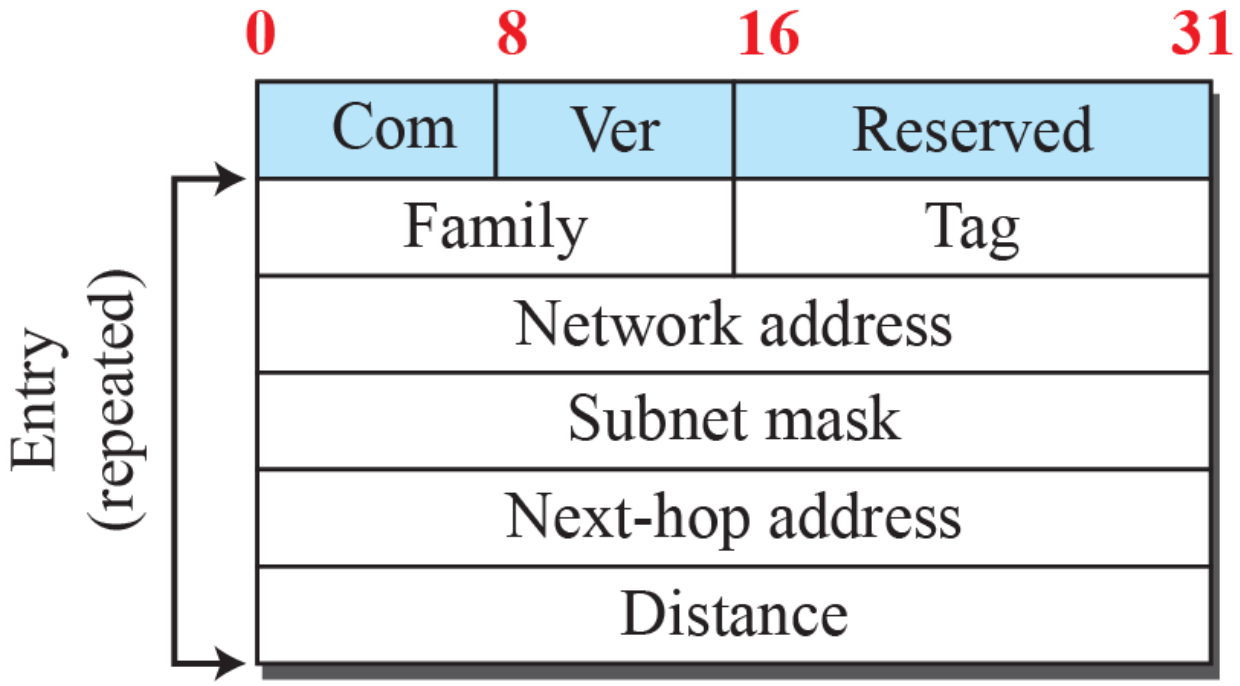
Routing Information Protocol (RIP)

RIP Implementation

- RIP is implemented as a **process** that uses the services of User Datagram Protocol (UDP) on the port number 520.
- RIP is a **daemon process** (a process running at the background).
- Although RIP is a routing protocol to help IP route its datagrams, the RIP messages are encapsulated inside UDP user datagrams, which in turn are encapsulated inside IP datagrams.
- It means that, RIP runs at the application layer, but creates forwarding tables for IP at the network layer.

Routing Information Protocol (RIP)

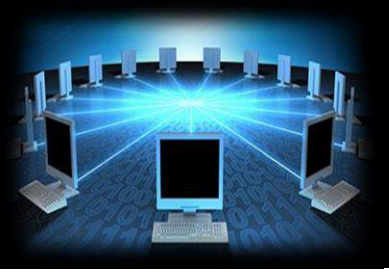
RIP Message Format



Fields

Com: Command, request (1), response (2)
Ver: Version, current version is 2
Family: Family of protocol, for TCP/IP value is 2
Tag: Information about autonomous system
Network address: Destination address
Subnet mask: Prefix length
Next-hop address: Address length
Distance: Number of hops to the destination

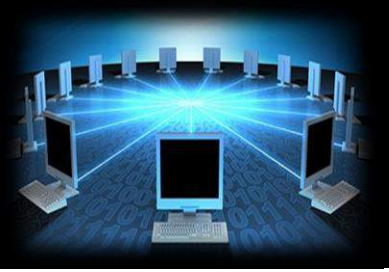
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Routing Information Protocol (RIP)

RIP Message Format

- RIP has two types of messages: request and response.
- **Request:**
  - A request message is sent by a router that has just come up or by a router that has some time-out entries.
  - A request can ask about specific entries or all entries.
- **Response:**
  - A response can be either **solicited** or **unsolicited**.
  - A **solicited** response is sent only in answer to a request. It contains information about the destination specified in the corresponding request.
  - An **unsolicited** response, on the other hand, is sent periodically, every 30 seconds or when there is a change in the routing table.

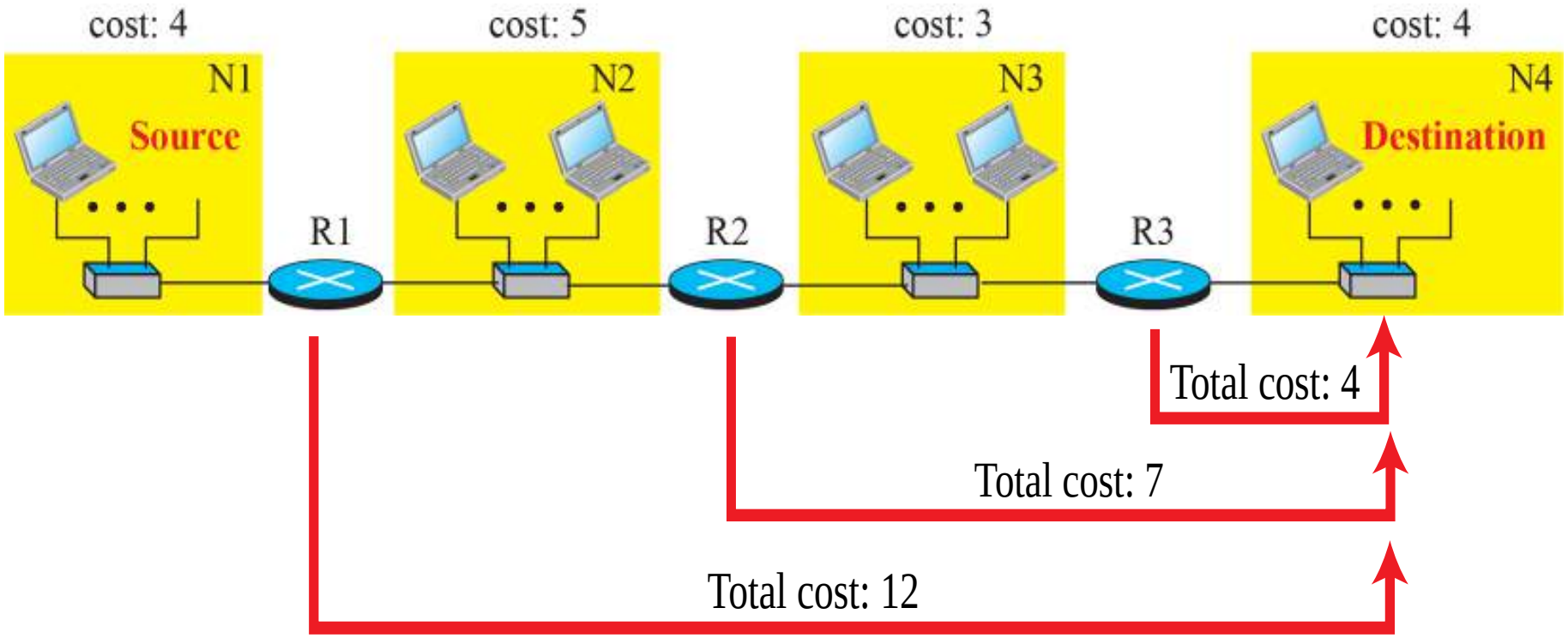


Open Shortest Path First (OSPF) Protocol

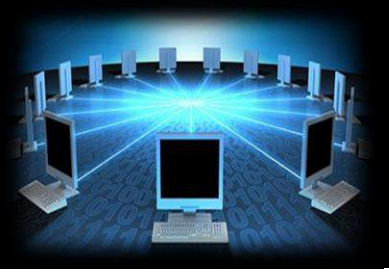
- The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on **link state routing**.
- Its domain is also an **autonomous system (AS)**.
- The OSPF protocol allows the administrator to assign a cost, called the **metric**, to each route.
- The metric can be based on a type of service (minimum delay, maximum throughput, and so on).

Open Shortest Path First (OSPF) Protocol

❑ Metric in OSPF



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Open Shortest Path First (OSPF) Protocol

❑ Forwarding Tables in OSPF

Forwarding table for R1

Destination network	Next router	Cost
N1	——	4
N2	——	5
N3	R2	8
N4	R2	12

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	——	5
N3	——	3
N4	R3	7

Forwarding table for R3

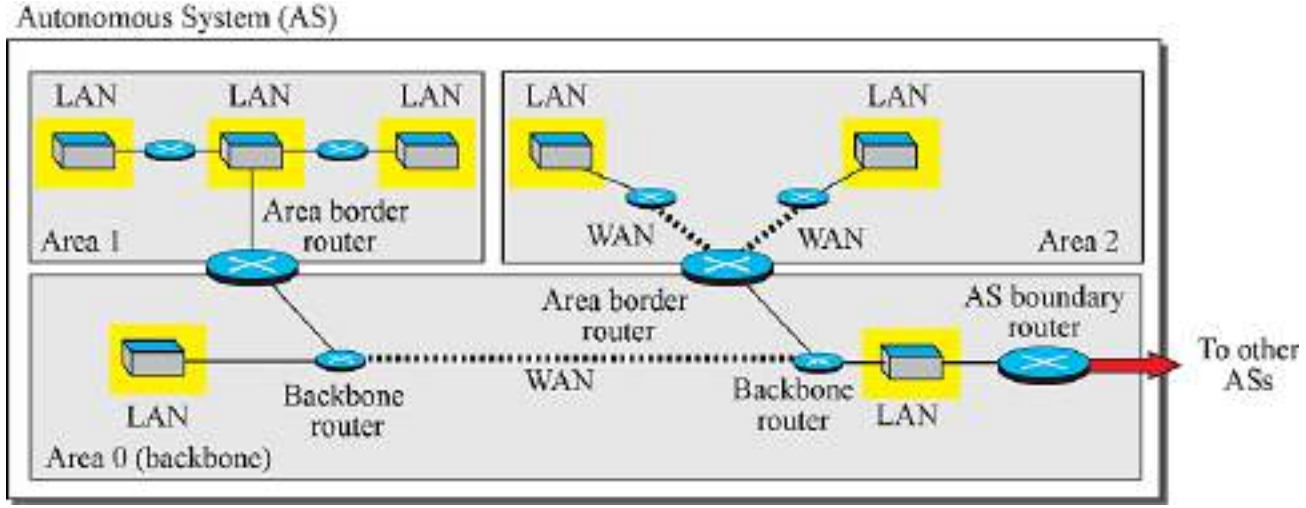
Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	——	3
N4	——	4

Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

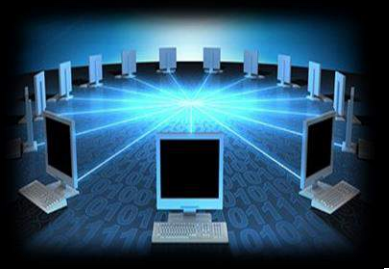
Open Shortest Path First (OSPF) Protocol

Areas in an autonomous system

- To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas.
- An **area** is a collection of networks, hosts, and routers all contained within an autonomous system.
- An autonomous system can be divided into many different areas.

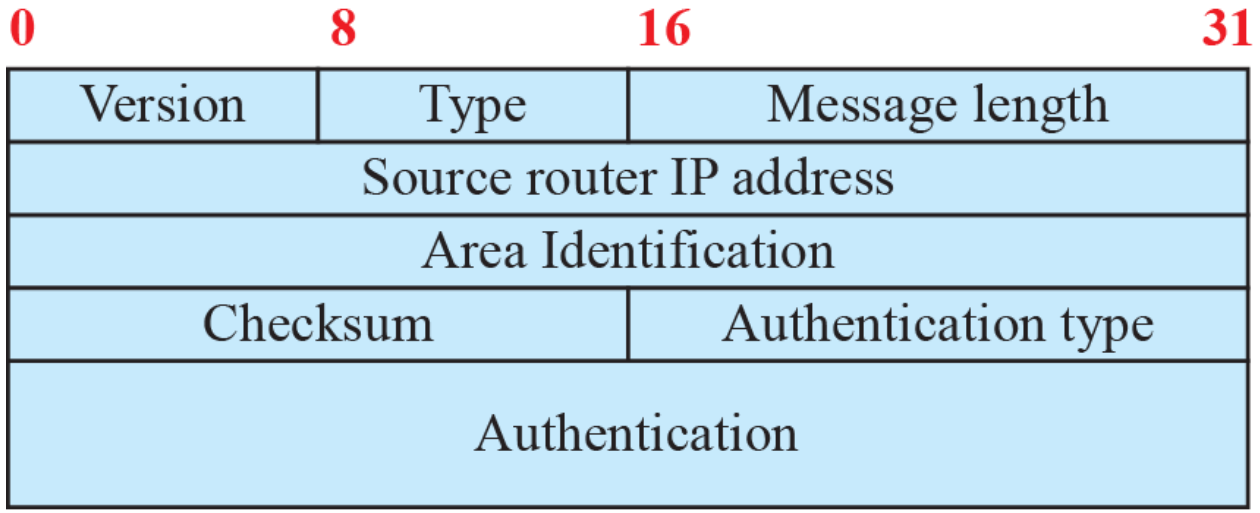


Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



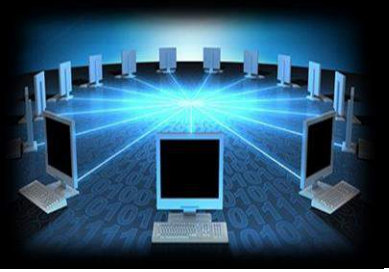
Open Shortest Path First (OSPF) Protocol

OSPF Common Header



- **Source router IP address:** This 32-bit field defines the IP address of the router that sends the packet.
- **Area identification:** This 32-bit field defines the area within which the routing takes place.

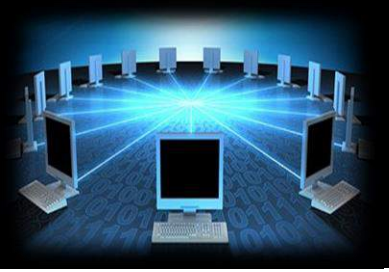
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



Open Shortest Path First (OSPF) Protocol

OSPF Common Header

- **Checksum:** This field is used for error detection on the entire packet excluding the authentication type and authentication data field.
- **Authentication type:** This 16-bit field defines the authentication protocol used in this area. At this time, two types of authentication are defined: 0 for none and 1 for password.
- **Authentication:** This 64-bit field is the actual value of the authentication data. In the future, when more authentication types are defined, this field will contain the result of the authentication calculation. For now, if the authentication type is 0, this field is filled with 0s. If the type is 1, this field carries an eight-character password.

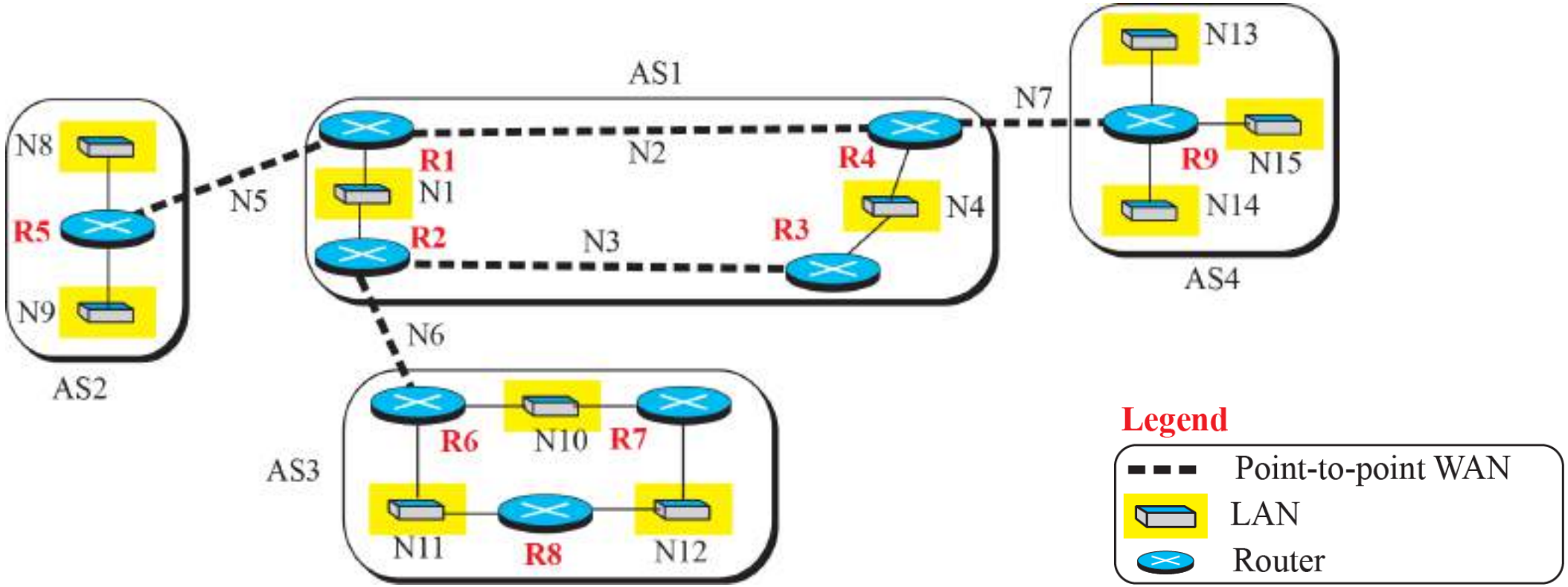


Border Gateway Protocol (BGP)

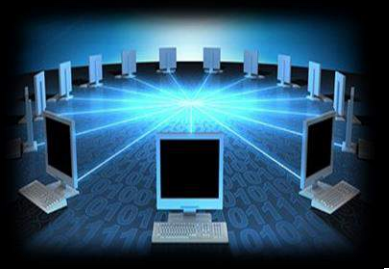
- Border Gateway Protocol (BGP) is the only **interdomain routing protocol** used in the Internet today.
- It first appeared in 1989 and has gone through four versions.
- BGP4 is based on the **path vector routing**.
- The protocol can connect together any internetwork of autonomous system (AS) using an arbitrary topology.
- The only requirement is that each AS have at least one router that is able to run BGP and that is router connect to at least one other AS's BGP router.

Border Gateway Protocol (BGP)

A sample internet with four ASes



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



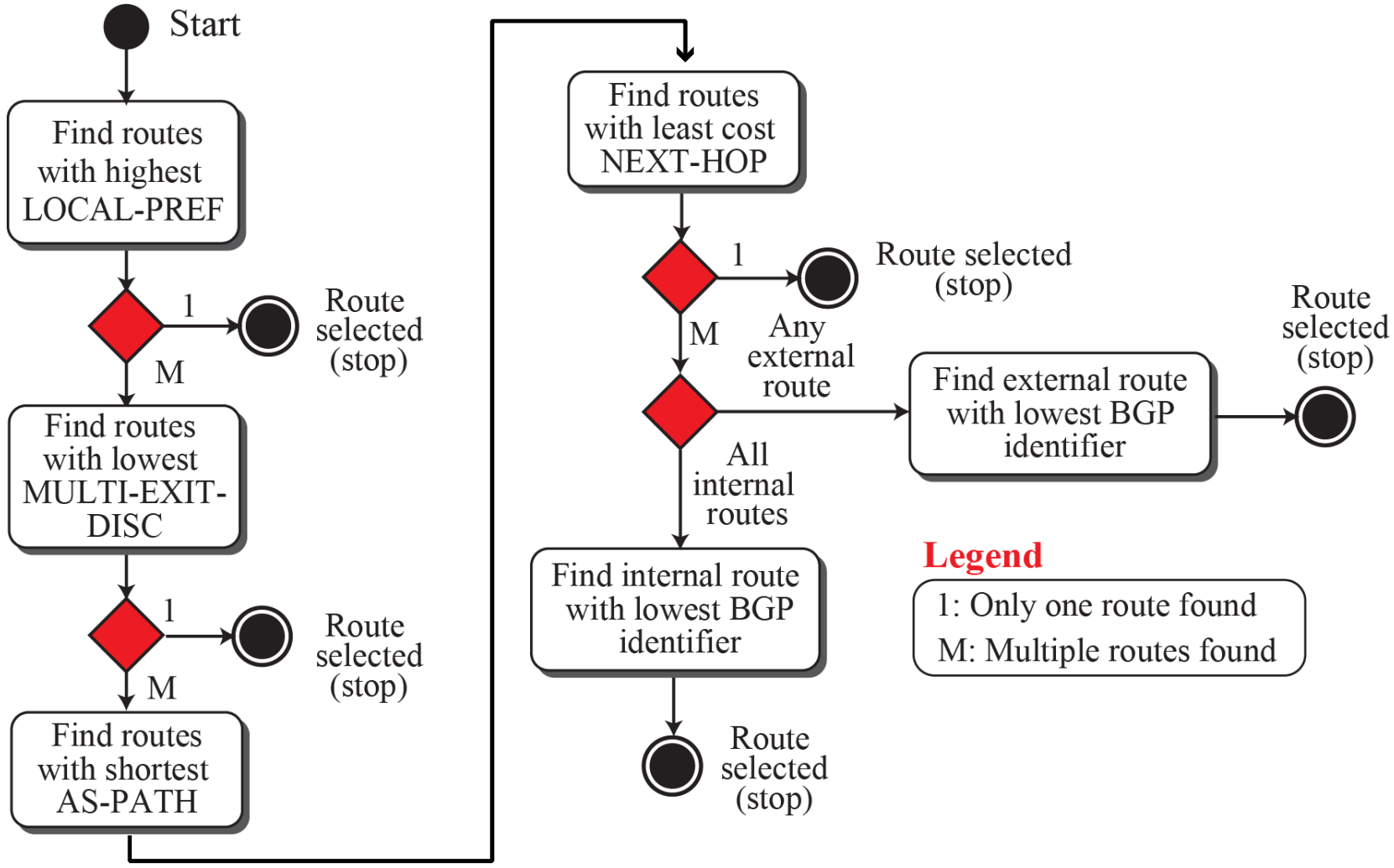
Border Gateway Protocol (BGP)

External BGP and internal BGP

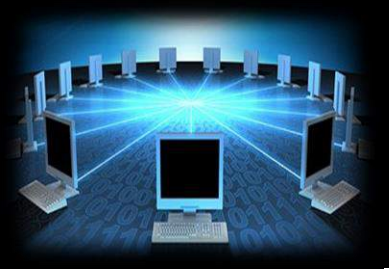
- There are two variations of BGP – eBGP and iBGP.
- Routes are exchanged and traffic is transmitted over the Internet using external BGP (eBGP).
- Autonomous systems can also use an internal version of BGP to route through their internal networks, which is known as internal BGP (iBGP).
- eBGP is installed on each border router (one at the edge of each AS which is connected to a router at another AS).
- iBGP is installed on all routers.

Border Gateway Protocol (BGP)

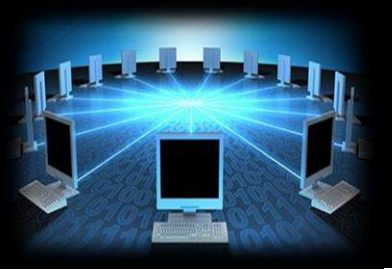
❑ Flow Diagram For Route Selection



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

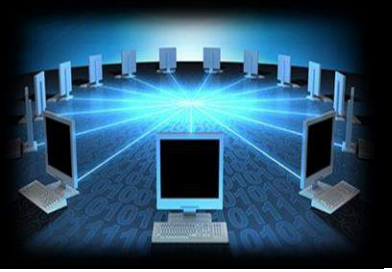


IPv6
(Addressing & Protocol)



IPv6

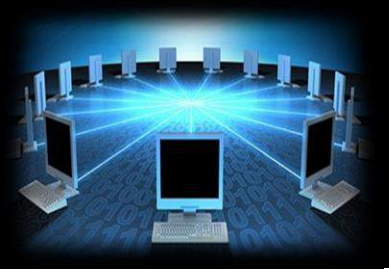
- The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP protocol in the early 1990s.
- The new version, which is called Internet Protocol version 6 (IPv6), or IP new generation (IPng).
- It was a proposal to augment the address space of IPv4 and at the same time redesign the format of the IP packet and revise some auxiliary protocols such as ICMP.



IPv6

Features of IPv6

- **Longer Address Field:** Address field extended from 32 bits to 128 bits with more levels of hierarchy.
- **Simplified Header Format:** Simpler header format compared to IPv4. Fields like Checksum, IHL, Identification, Flags and Fragment Offset are absent.
- **Flexible support for options:** The options in IPv6 appear in optional extension headers that are encoded more efficiently and flexibly.
- **Flow label capacity:** Adds flow label to identify a certain packet flow that requires a certain QoS.
- **Security:** Supports built in authentication and confidentiality.
- **Large Packets:** Supports payloads that are longer than 64 kbytes, called jumbo payloads.



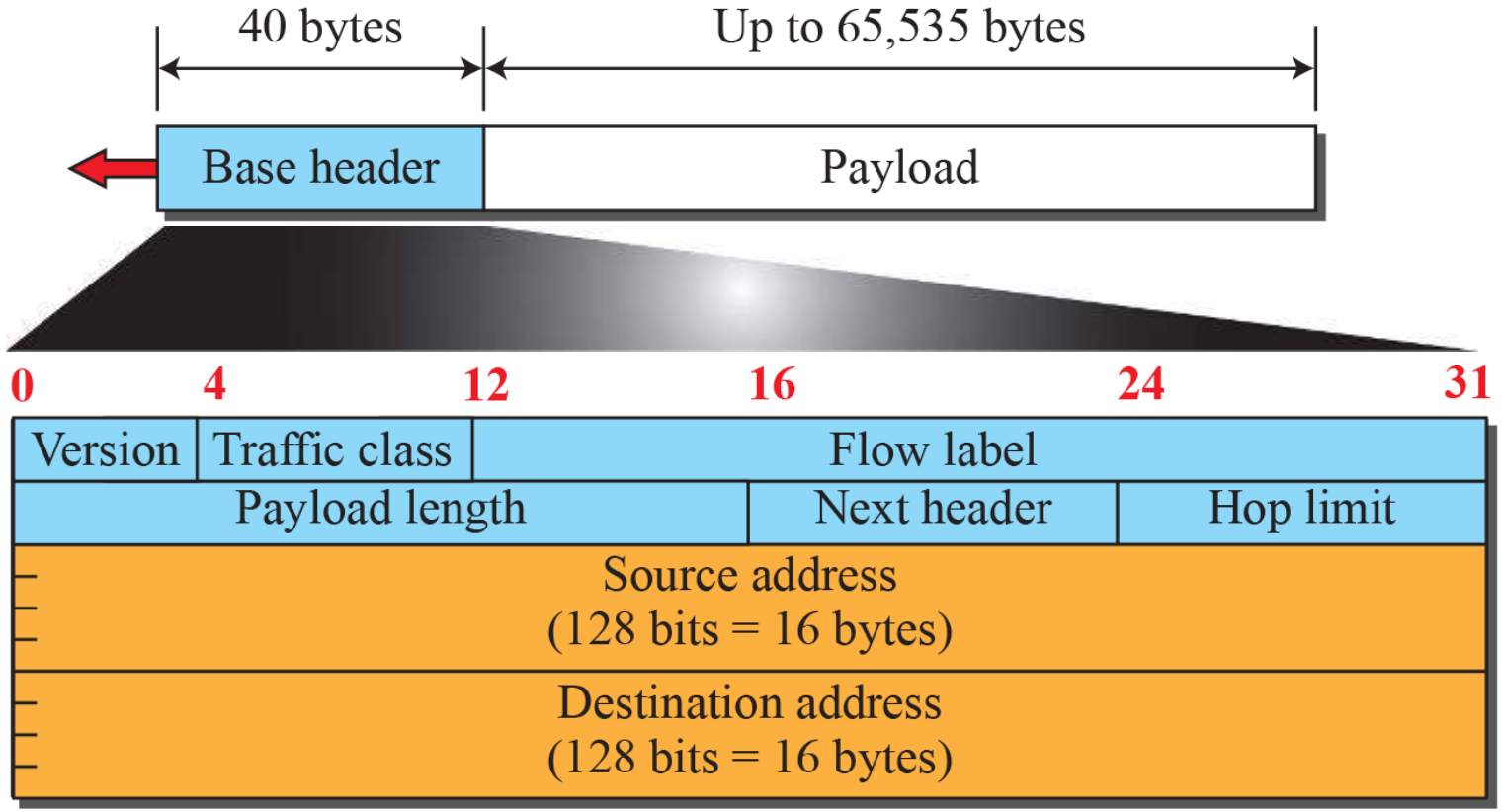
IPv6

Features of IPv6

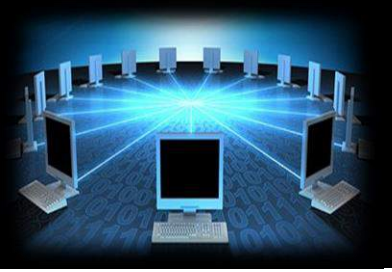
- **Fragmentation at source only:** Routers do not perform fragmentation. If a packet needs to be fragmented, the source should check minimum MTU along the path and perform the necessary fragmentation.
- **No checksum field:** The checksum field is removed to reduce the packet processing time at a router.

Packet Format of IPv6

- Each packet is composed of a base header followed by the payload. The base header occupies 40 bytes, whereas payload can be up to 65,535 bytes of information.



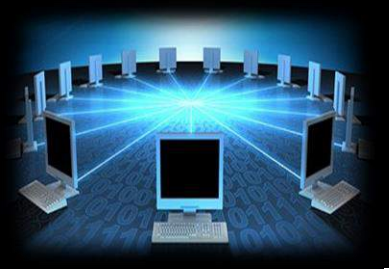
Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.



IPv6

Packet Format of IPv6

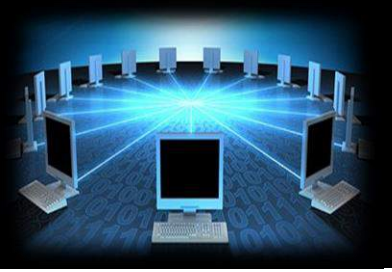
- **Version:** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- **Traffic Class:** This 8-bit field is used to distinguish different payloads with different delivery requirements. It replaces the service class field in IPv4.
- **Flow label:** The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data. It identifies the QoS requested by the packet.
- **Payload length:** The 16-bit payload length field defines the length of the IP datagram excluding the base header. With 16-bits, payload length is limited to 65535 bytes. To send larger payloads, we can use options in the extension header.
- **Next header:** The 8-bit field identifies the type of extension header that follows the base header. Gives more flexibility and efficiency to IPv6.



IPv6

Packet Format of IPv6

- **Hop limit:** This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- **Source address:** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address:** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

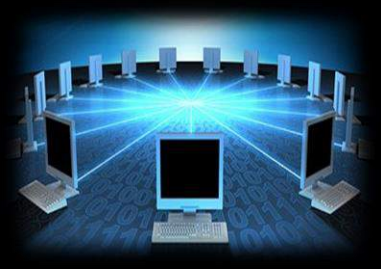


IPv6

IPv6 Address Notation

- A computer normally stores the address in binary, but is clear that 128 bits cannot easily be handled by humans.
- To make addresses more readable, IPv6 specifies **colon hexadecimal notation** (or colon hex for short).
- In this notation, 128 bits are divided into eight sections, each 2 bytes in length.
- Two bytes in hexadecimal notation require four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.
- E.g.

FDEC : BA98 : 7654 : 3210 : ADBF : BBFF : 2922 : FFFF



IPv6

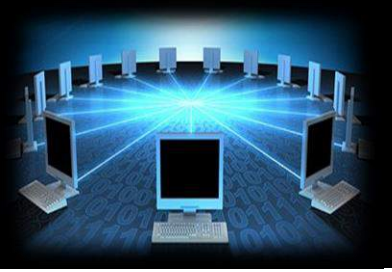
IPv6 Address Notation

- **Zero compression:**
- Zero compression can be applied to colon hex notation if there are consecutive sections consisting of zeros only.
- E.g.

FDEC : 0000 : 0000 : 0000 : 0DBF : BBFF : 2922 : 00FF

FDEC : 0 : 0 : 0 : DBF : BBFF : 2922 : FF

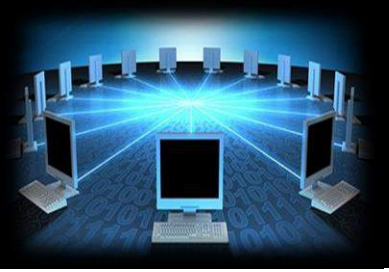
FDEC : : DBF : BBFF : 2922 : FF



IPv6

Types of Addresses

- In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast.
- **Unicast Address:**
  - A unicast address defines a single interface (computer or router).
  - The packet sent to a unicast address will be routed to the intended recipient.
- **Anycast Address:**
  - An anycast address defines a group of computers that all share a single address.
  - A packet with an anycast address is delivered to only one member of the group, the most reachable one.
  - An anycast communication is used, for example, when there are several servers that can respond to an inquiry. The request is sent to the one that is most reachable.

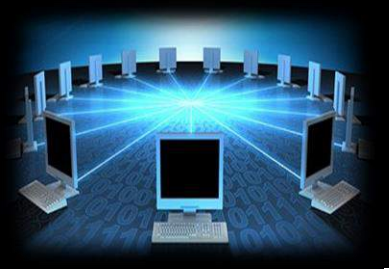


IPv6

Types of Addresses

- **Multicast Address:**
  - A multicast address also defines a group of computers.
  - However, there is a difference between anycasting and multicasting. In multicasting, each member of the group receives a copy.

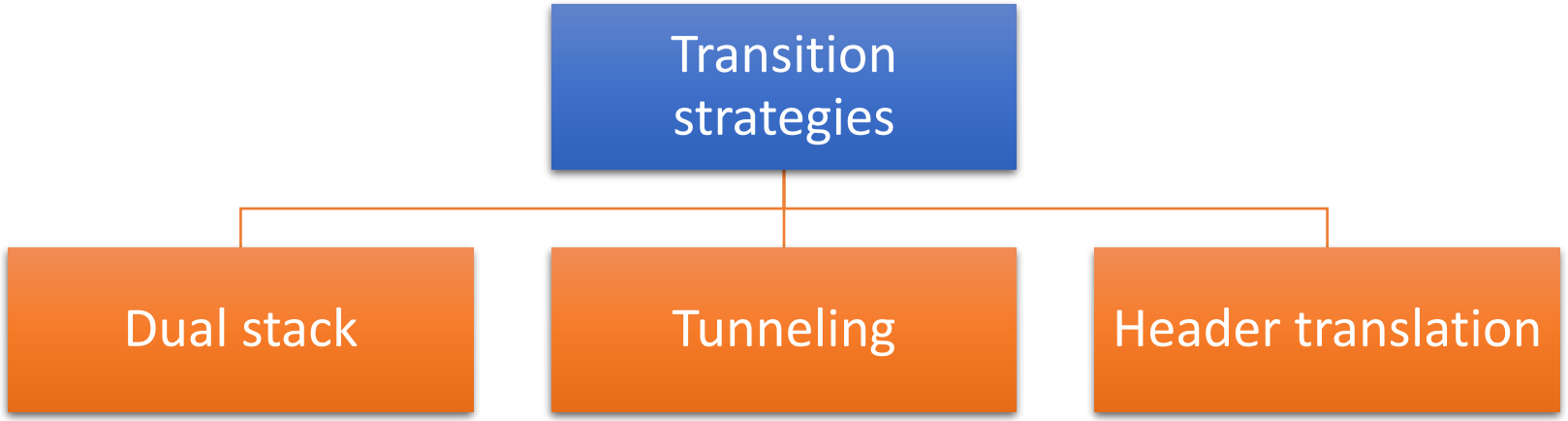
IPv6 does not define broadcasting. IPv6 considers broadcasting as a special case of multicasting.

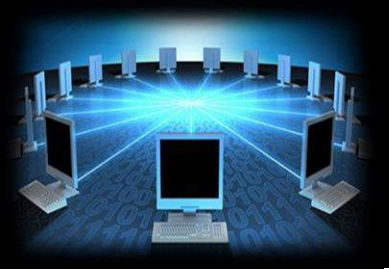


IPv6

TRANSITION FROM IPv4 TO IPv6

- Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.
- The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.
- Three strategies have been devised by the IETF to help the transition

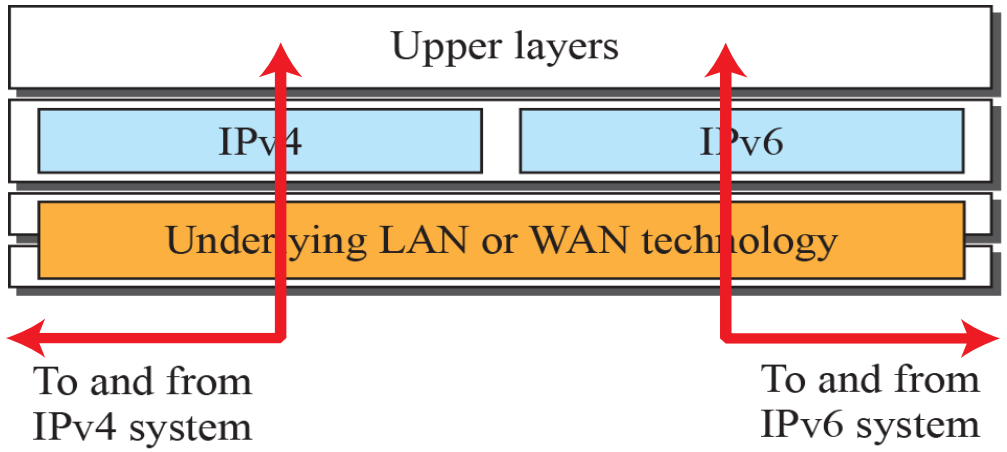




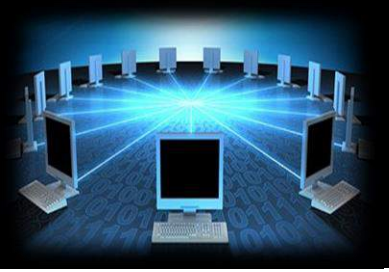
IPv6

Dual Stack

- In **dual stack**, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.
- To determine which version to use when sending a packet to a destination, the source host queries the DNS.
- If the DNS returns an IPv4 address, the source host sends an IPv4 packet. If the DNS returns an IPv6 address, the source host sends an IPv6 packet.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

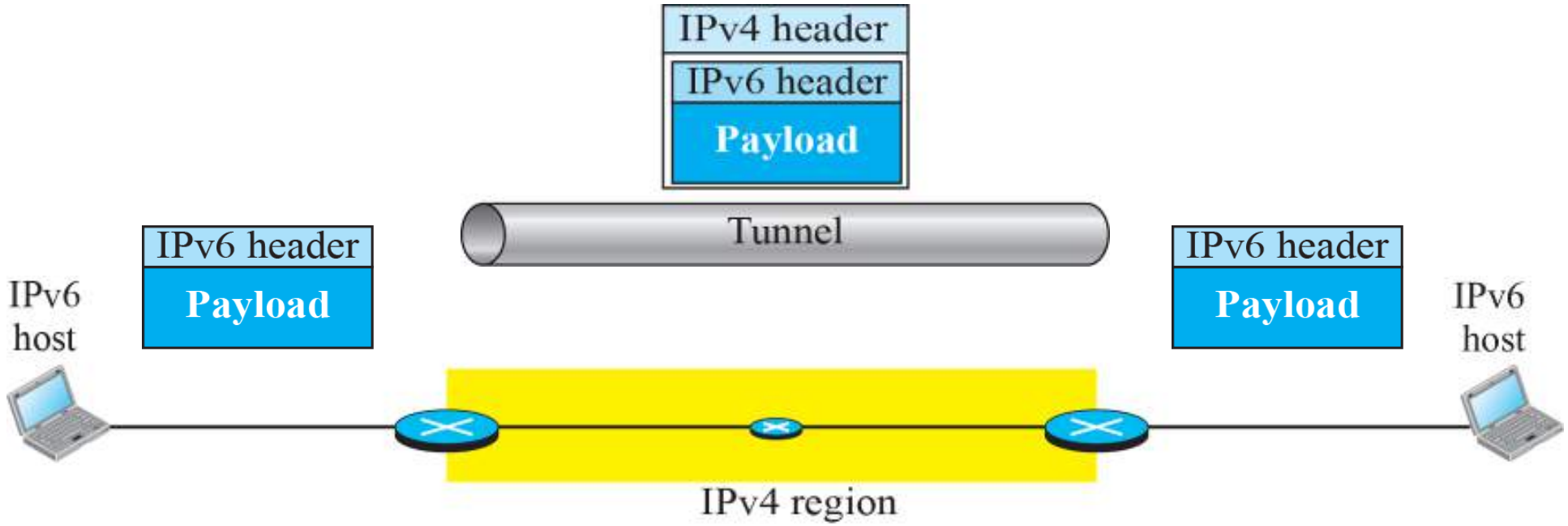


IPv6

Tunneling

- **Tunneling** is a strategy used when two computers using IPv6 want to communicate with each other, and the packet must pass through a region that uses IPv4.
- To pass through this region, the packet must have an IPv4 address.
- So, the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region.
- It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end.
- To make it clear that the IPv4 packet is carrying an IPv6 packet as data, the protocol value is set to 41.

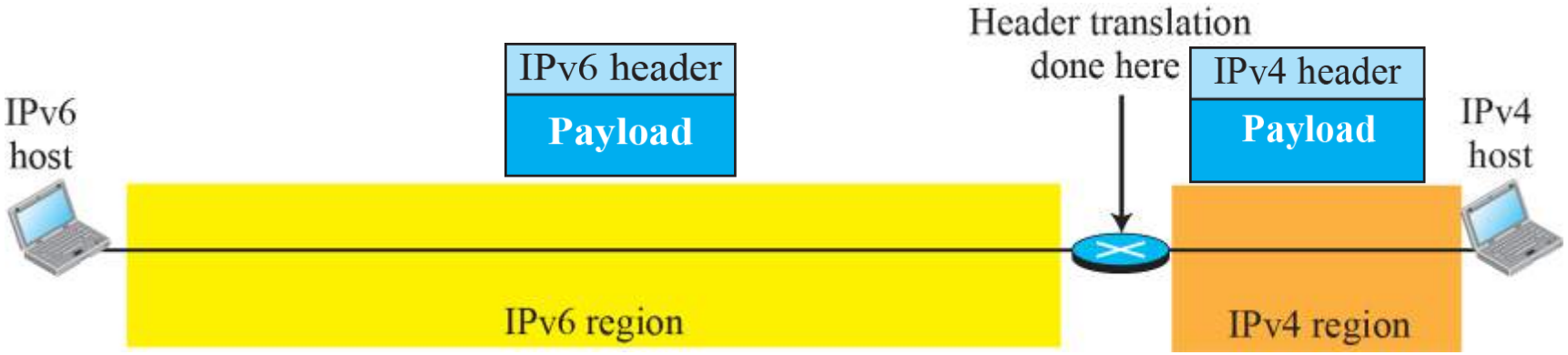
Tunneling



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.

Header Translation

- **Header translation** is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4.
- The sender wants to use IPv6, but the receiver does not understand IPv6.
- In this case, the header format must be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header.



Ref: Behrouz A. Forouzan, Data Communications and Networking, 6th Edition, Mc Graw Hill education.