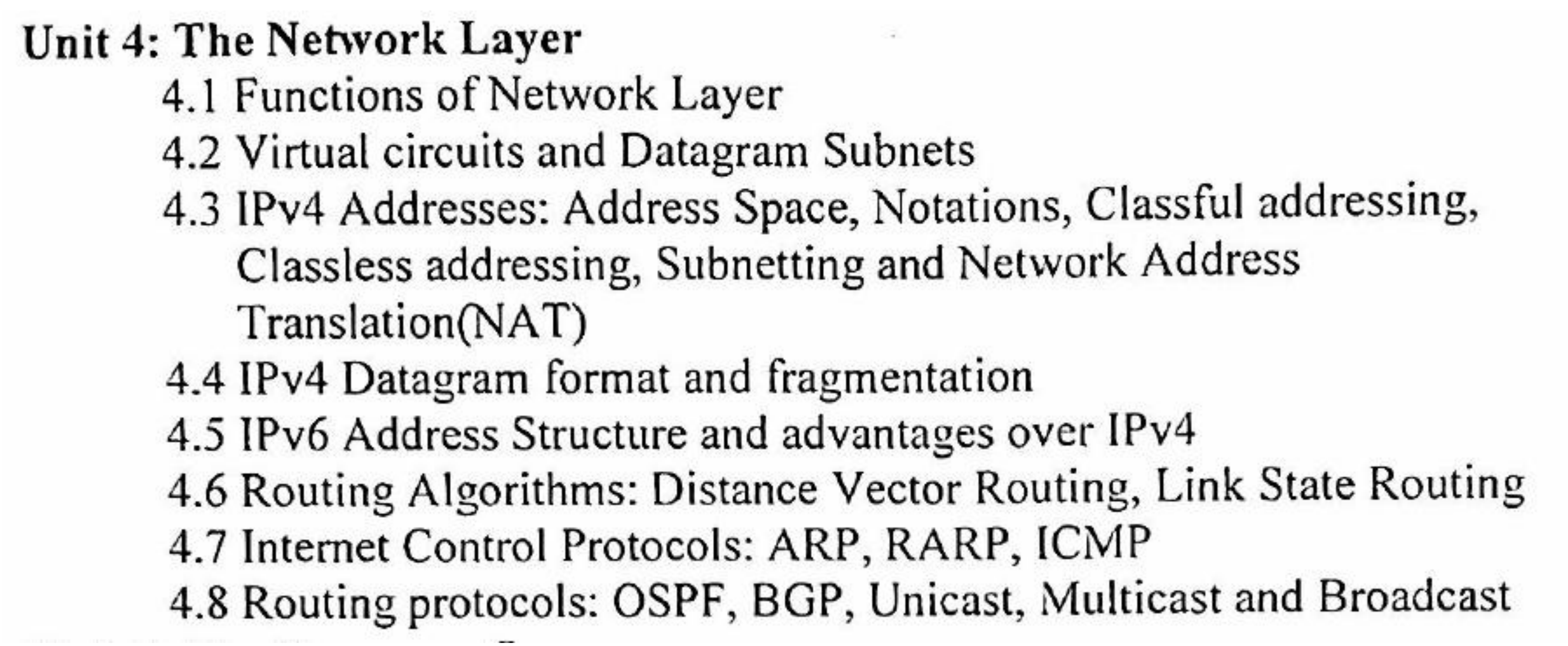
Computer Networking
BCA V SEM

The Network Layer

Kailash Karki
Kailash.karki@deerwalk.edu.np



Unit 4: The Network Layer

- 4.1 Functions of Network Layer
- 4.2 Virtual circuits and Datagram Subnets
- 4.3 IPv4 Addresses: Address Space, Notations, Classful addressing, Classless addressing, Subnetting and Network Address Translation(NAT)
- 4.4 IPv4 Datagram format and fragmentation
- 4.5 IPv6 Address Structure and advantages over IPv4
- 4.6 Routing Algorithms: Distance Vector Routing, Link State Routing
- 4.7 Internet Control Protocols: ARP, RARP, ICMP
- 4.8 Routing protocols: OSPF, BGP, Unicast, Multicast and Broadcast

Introduction

- Network layer works for the transmission of data from one host to the other located in different networks. It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender & receiver's IP address are placed in the header by the network layer.
- The network Layer controls the operation of the subnet. The main aim of this layer is to deliver packets from source to destination across multiple links (networks). If two computers (system) are connected on the same link, then there is no need for a network layer. It routes the signal through different channels to the other end and acts as a network controller.
- It also divides the outgoing messages into packets and to assemble incoming packets into messages for higher levels.

Functions

- **Internetworking:** It provides Internetworking.
- **Logical Addressing:** When packet is sent outside the network, N/W layer adds Logical (network) address of the sender & receiver to each packet. Network addresses are assigned to local devices by n/w administrator and assigned dynamically by special server called DHCP (Dynamic Host Configuration Protocol)
- **Routing:** When independent n/w are connected to create internetwork several routes are available to send the data from S to D. These n/w are interconnected by routers & gateways that route the packet to final destination. Protocols used to route the network traffic are known as Network layer protocols.
- **Packetizing:** Encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.

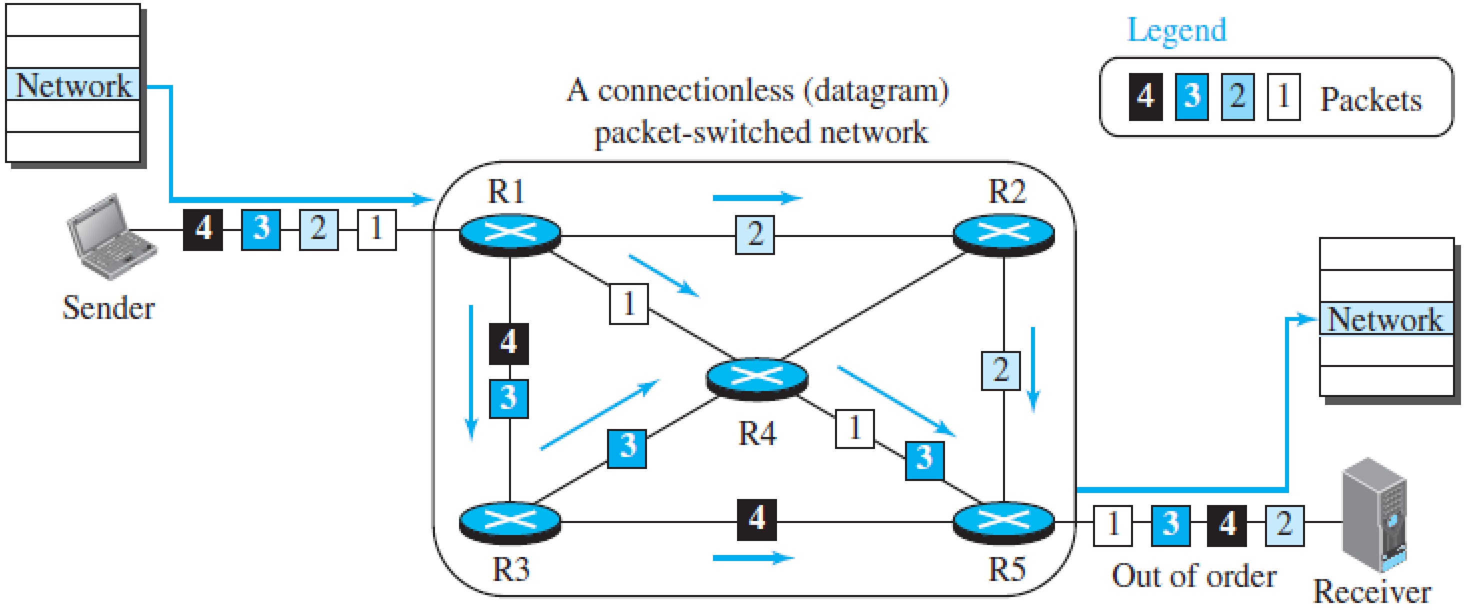
Packet Switching:

- At the network layer, a message from the upper layer is divided into manageable packets and each packet is sent through the network. The source of the message sends the packets one by one; the destination of the message receives the packets one by one. The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer. The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.

- A packet-switched network can use two different approaches to route the packets: the *datagram approach* and the *virtual circuit approach*.

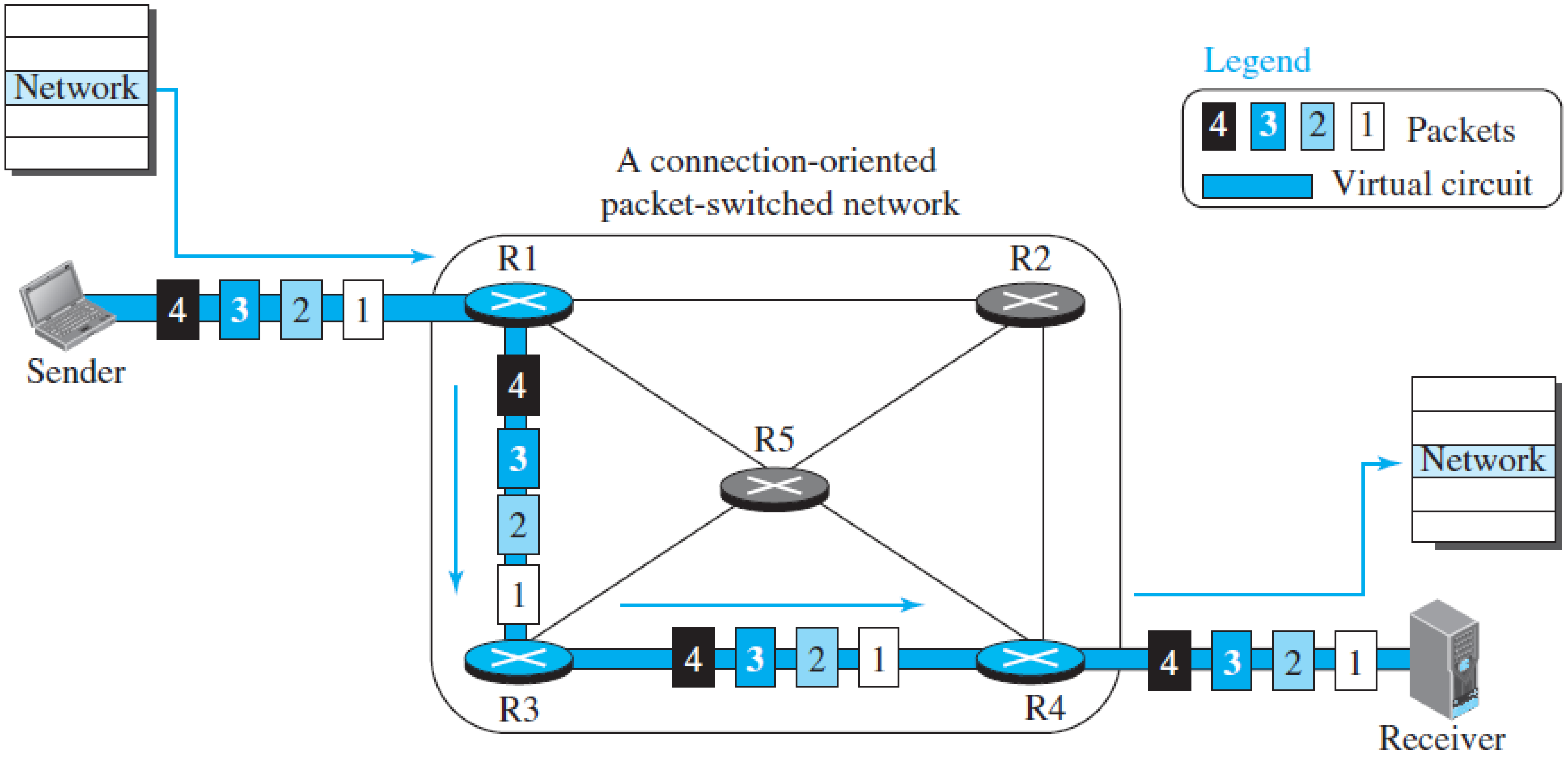
Datagram Approach: Connectionless Service

- When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message.

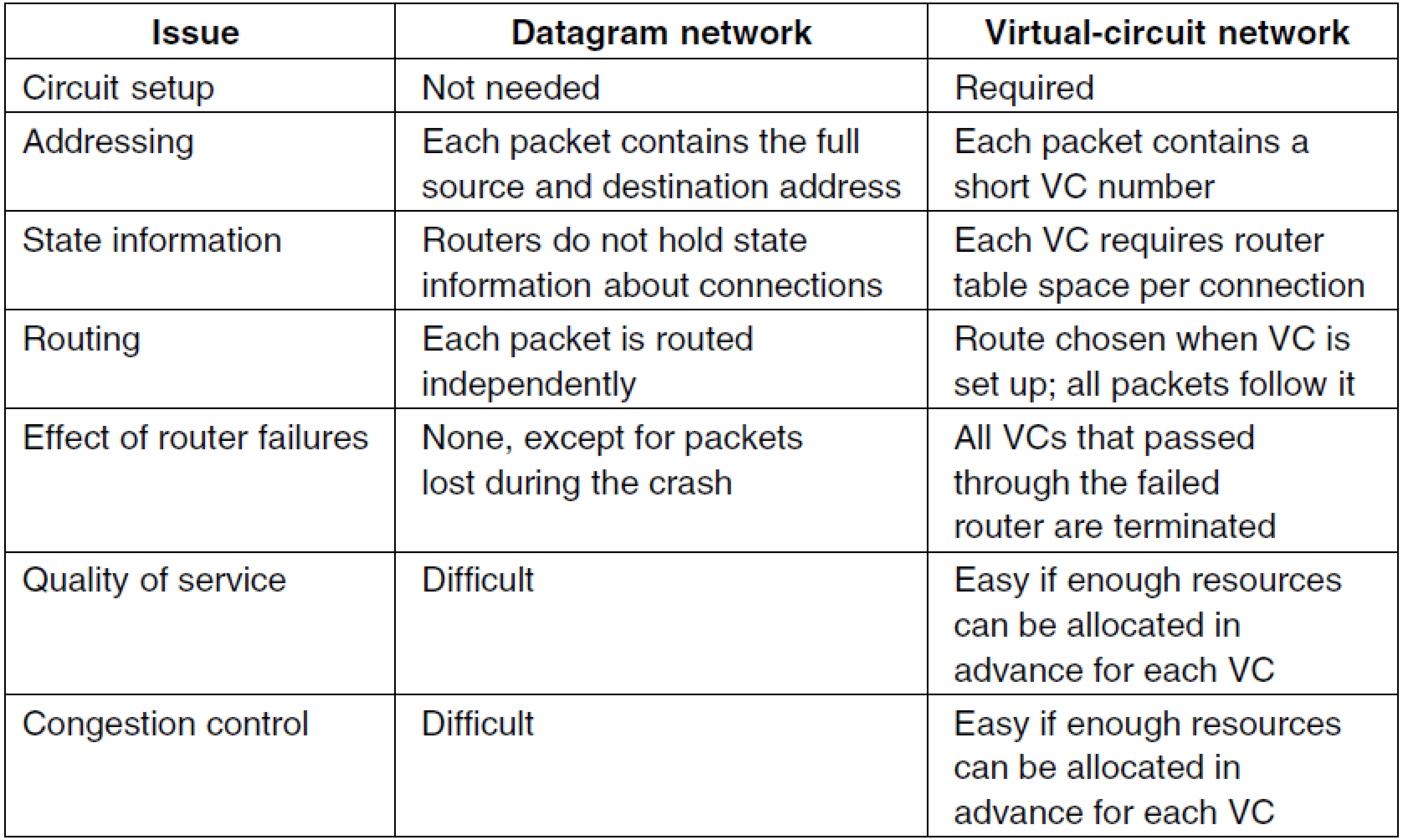


- Each packet is routed based on the information contained in its header: source and destination addresses. The destination address defines where it should go; the source address defines where it comes from. The router in this case routes the packet based only on the destination address. The source address may be used to send an error message to the source if the packet is discarded.

Virtual-Circuit Approach: Connection-Oriented Service



In a connection-oriented service (also called *virtual-circuit approach*), there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path.



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

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.
- An IPv4 address is a 32-bit address that 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.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet. If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- An IPv4 address is typically written in decimal digits, formatted as four 8-bit fields separated by periods. Each 8-bit field represents a byte of the IPv4 address. This form of representing the bytes of an IPv4 address is often referred to as the dotted-decimal format such as 192.168.123.132, 168.212. 226.204.

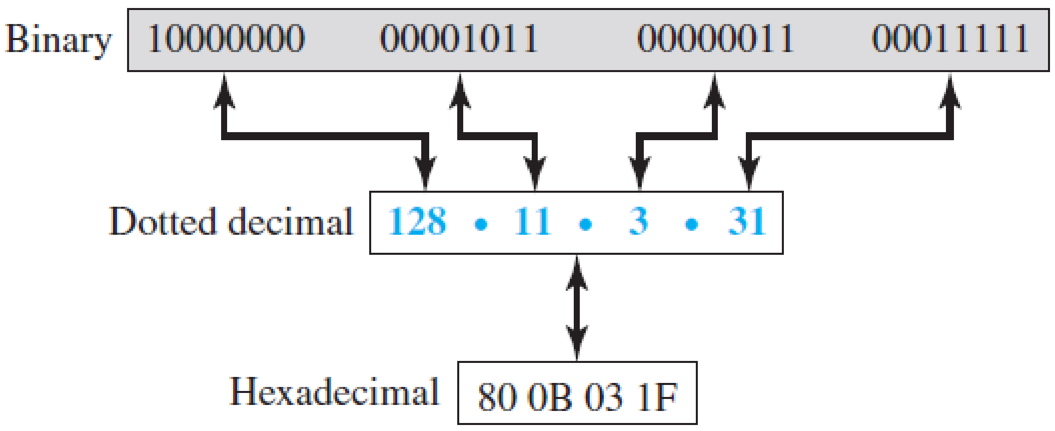
IPv4 Address : Address Space

- An **address space** is the total number of addresses used by the protocol. If a protocol uses *b* bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion)

Notation:

There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16).

Three different notations in IPv4 addressing



Change the following IP address from binary notation to dotted-decimal notation.

10000001 00001011 00001011 11101111

Change the following IP address from dotted-decimal notation to binary notation.

111.56.45.78

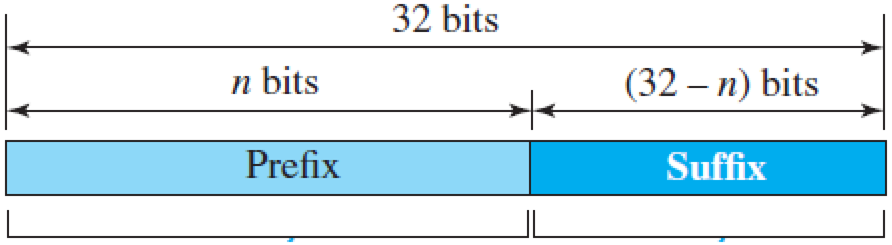
Find the error, if any, in the following IP address

a) 111.56.045.78
b) 221.34.7.8.20
c) 75.45.301.14
d) 11100010.23.14.67

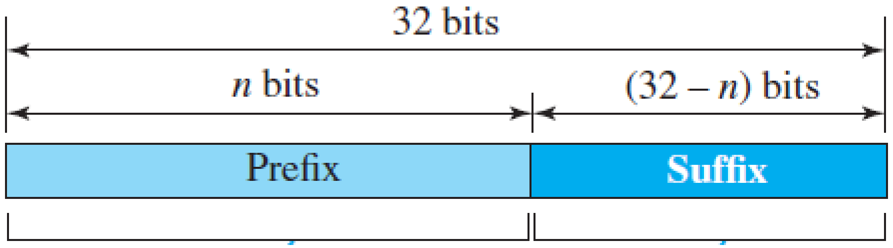
Hierarchy in Addressing

- In any communication network that involves delivery, such as a telephone network or a postal network, the addressing system is hierarchical. In a postal network, the postal address (mailing address) includes the country, state, city, street, house number, and the name of the mail recipient. Similarly, a telephone number is divided into the country code, area code, local exchange, and the connection.

- A 32-bit IPv4 address is also hierarchical, but divided only into two parts. The first part of the address, called the *prefix*, defines the network; the second part of the address, called the *suffix*, defines the host (connection of a device to the Internet).

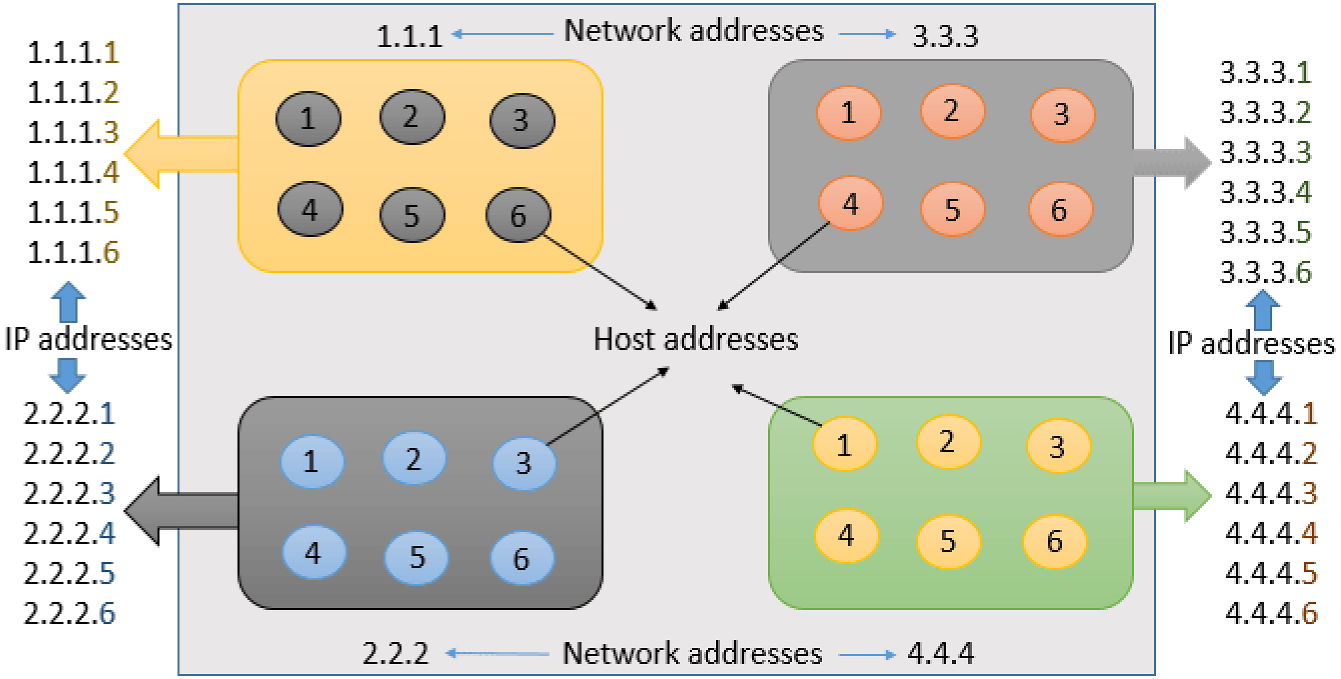


- A prefix can be fixed length or variable length. The network identifier in the IPv4 was first designed as a fixed-length prefix. This scheme, which is now obsolete, is referred to as classful addressing. The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.



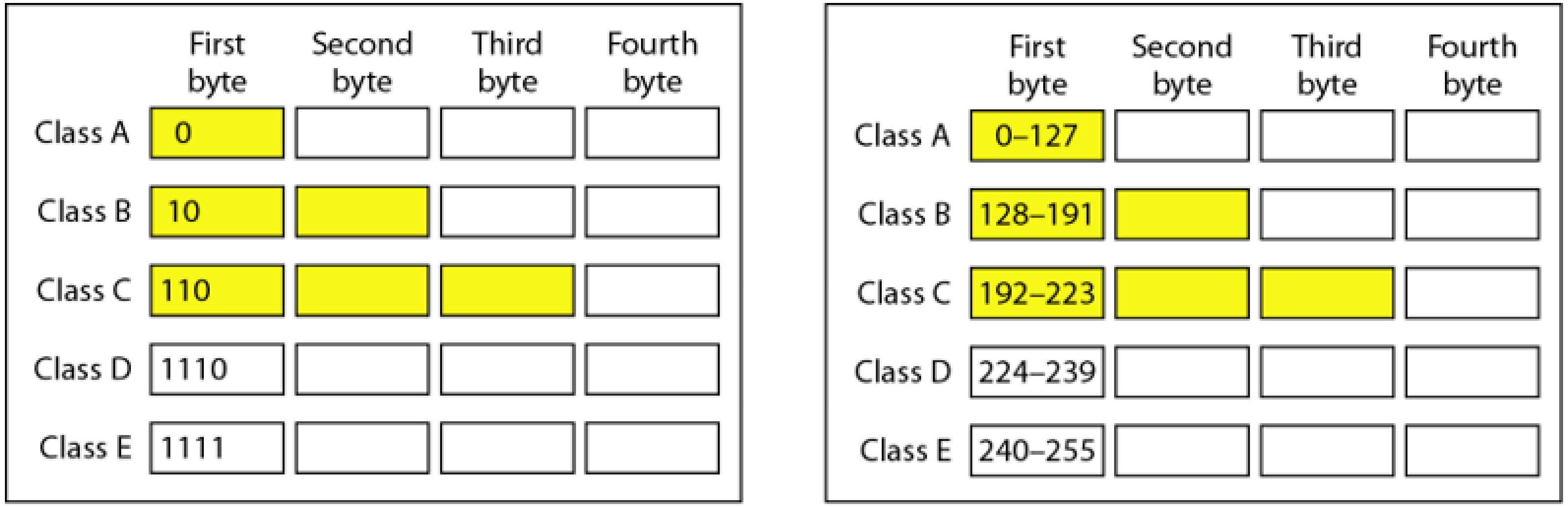
Network Address and Host Address:

The Network address identifies the specific network to which host is attached, and Host address uniquely identifies a host within a network.



Classful Addressing:

When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (*n* = 8, *n* = 16, and *n* = 24). The whole address space was divided into five classes (class A, B, C, D, and E). This scheme is referred to as **classful addressing.**



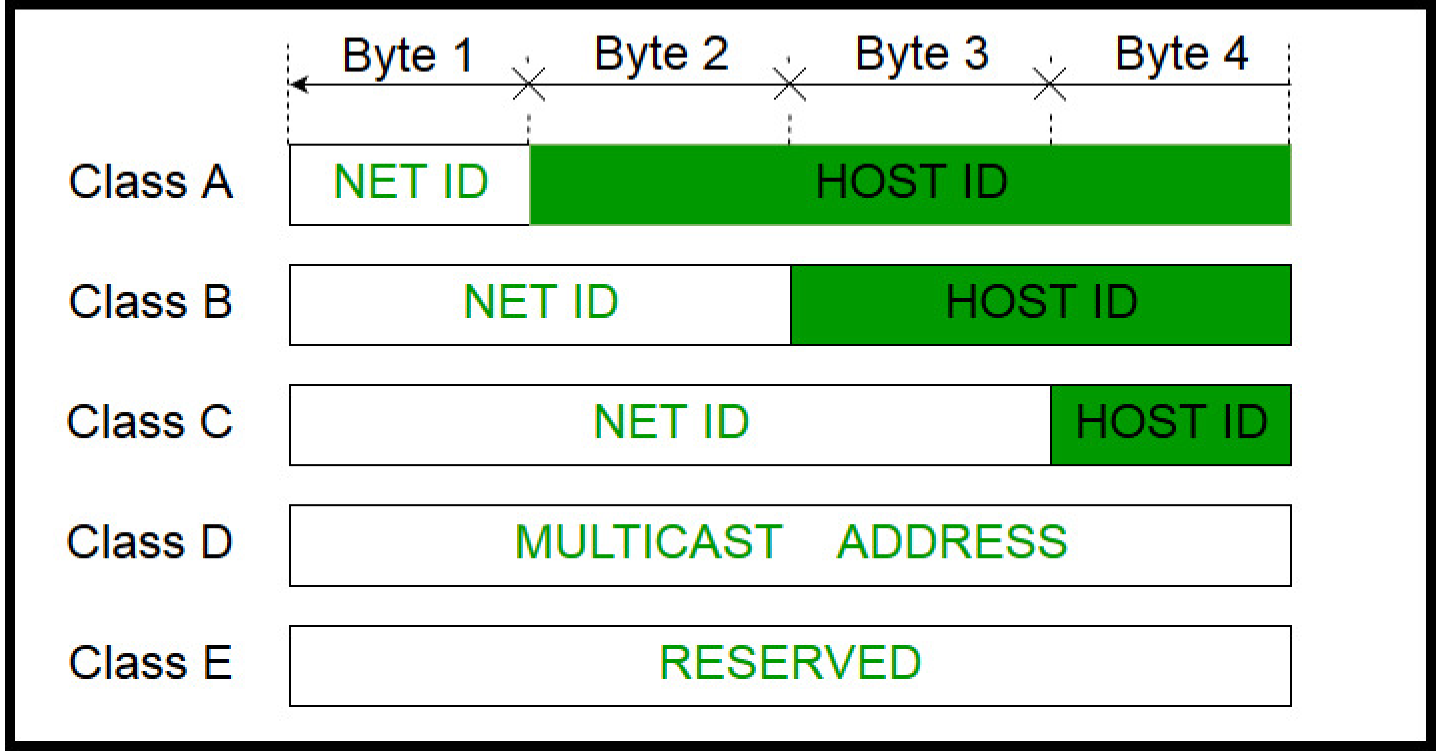
	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10	🟨		
Class C	110	🟨	🟨	
Class D	1110			
Class E	1111			

a. Binary notation

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

b. Dotted-decimal notation

In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier. This means there are only $2^7 = 128$ networks in the world that can have a class A address.



In class B, the network length is 16 bits, but since the first two bits, which are (10), define the class, we can have only 14 bits as the network identifier. This means there are only $2^{14} =$ 16,384 networks in the world that can have a class B address.

All addresses that start with (110) belong to class C. In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier. This means there are 2^21 = 2,097,152 networks in the world that can have a class C address.

Class D is not divided into prefix and suffix. It is used for multicast addresses. All addresses that start with 1111 in binary belong to class E. As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

Find the class of the address:

1. 00000001 00001011 00001011 11101111
2. 11000001 10000011 00011011 11111111
3. 227.12.14.87
4. 193.14.56.22

Class A

Class A: divided into 128 blocks

- First block: **0**.0.0.0 ~ **0**.255.255.255
- ….
- Last block: **127**.0.0.0 ~ **127**.255.255.255

Each block contains 16777216 addresses

Class B

Class B: divided into 16,384 blocks

First block: **128.0**.0.0 ~ **128.0**.255.255

….

Last block: **191**.**255**.0.0 ~ **191**.**255**.255.255

Each block contains 65,535 addresses

Class C

Class C: divided into 2.097,152 blocks

- First block: **192.0.0**.0 ~ **128.0.0**.255
- ….
- Last block: **223.255.255**.0 ~ **223.255.255**.255

Each block contains 256 addresses

Class D and Class E

- Class D
  - Just one block
  - Designed for multicasting and each address is used to identify one multicasting group
- Class E
  - Just one block
  - Designed for use as reserved addresses

Address Depletion:

The reason that classful addressing has become obsolete is address depletion. Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.

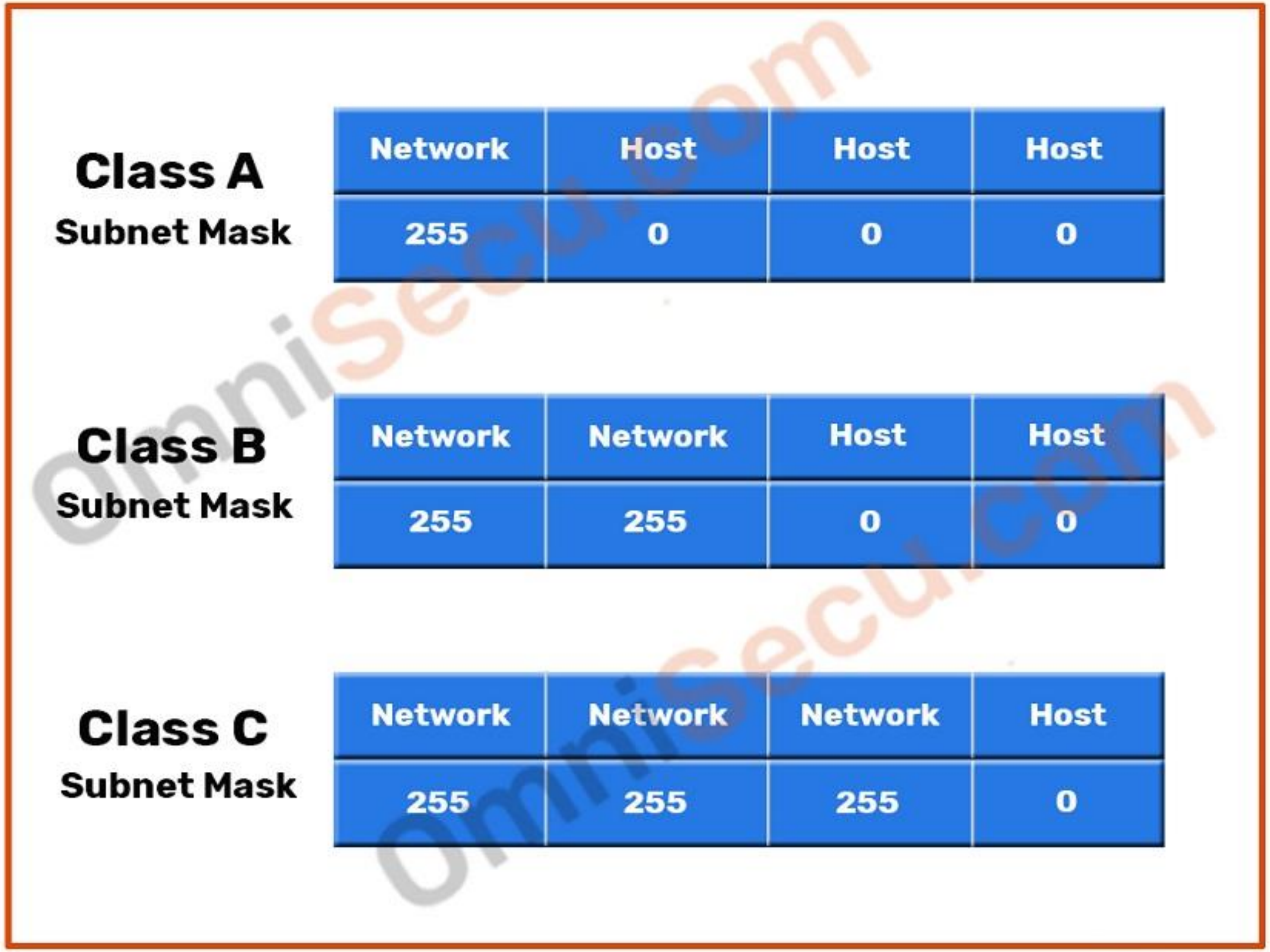
To understand the problem, let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network (seen by the rest of the world) with 16,777,216 nodes (computers in this single network). Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).

Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.

Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.

Subnet Mask

- A subnet mask is a 32-bit number created by setting host bits to all 0s and setting network bits to all 1s. In this way, the subnet mask separates the IP address into the network and host addresses.



Class A	Network	Host	Host	Host
Subnet Mask	255	0	0	0
Class B	Network	Network	Host	Host
Subnet Mask	255	255	0	0
Class C	Network	Network	Network	Host
Subnet Mask	255	255	255	0

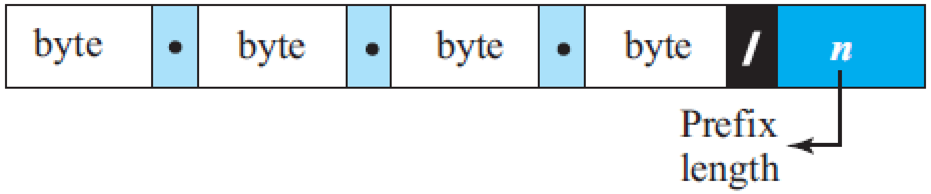
Classless Addressing

Subnetting and supernetting in classful addressing did not really solve the address depletion problem. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6 (discussed later), a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called *classless addressing*. In other words, the class privilege was removed from the distribution to compensate for the address depletion.

Unlike classful addressing, the prefix length in classless addressing is variable. We can have a prefix length that ranges from 0 to 32. The size of the network is inversely proportional to the length of the prefix. A small prefix means a larger network; a large prefix means a smaller network.

Prefix Length: Slash Notation

- The first question that we need to answer in classless addressing is how to find the prefix length if an address is given. Since the prefix length is not inherent in the address, we need to separately give the length of the prefix. In this case, the prefix length, *n*, is added to the address, separated by a slash. The notation is informally referred to as *slash notation* and formally as ***classless interdomain routing*** or ***CIDR*** strategy.



Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

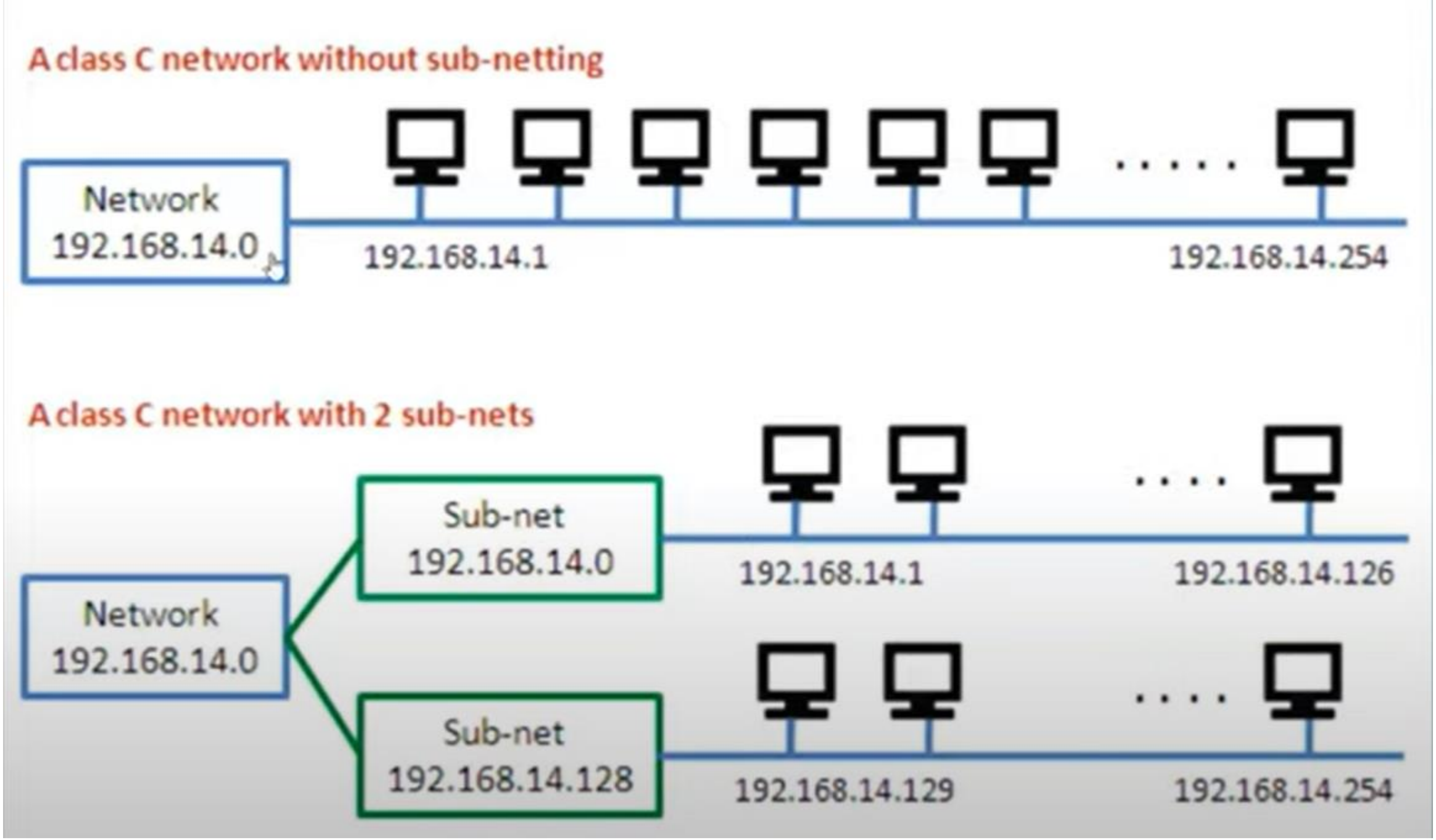
1. A classless address is given as 167.199.170.82/27.

Find

a) The number of addresses
b) First Address
c) Last Adress

2. Given the CIDR representation 20.10.30.35 / 27. Find the range of IP Addresses in the CIDR block.

3. Given the CIDR representation 100.1.2.35 / 20. Find the range of IP Addresses in the CIDR block



A class C network without sub-netting

A class C network with 2 sub-nets

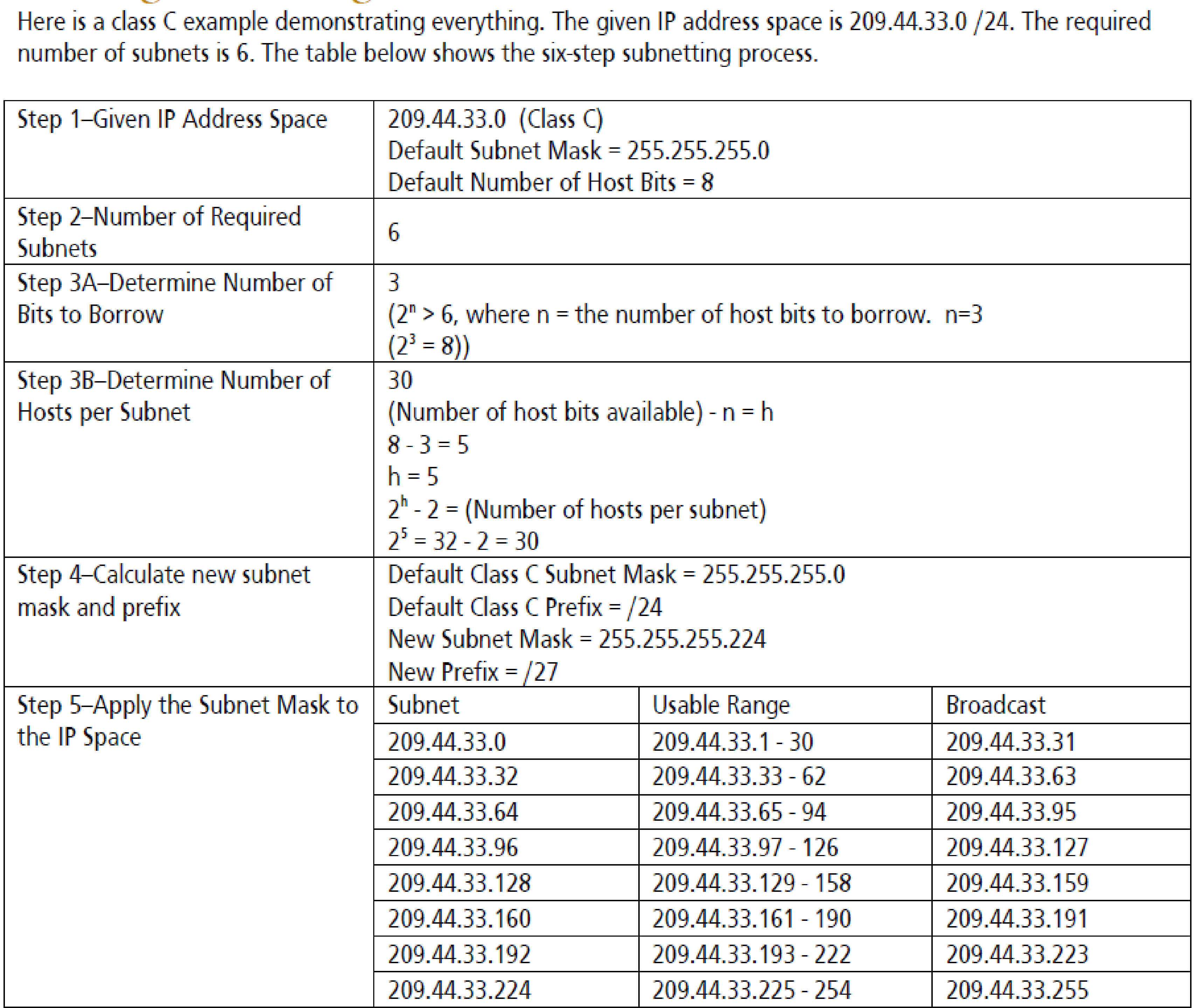
Subnetting:

- A subnetwork or subnet is a logical division of an IP network.
- An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet). Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks, and so on.

Suppose you are given network address: 192.168.10.0 and subnet mask: 255.255.255.240 then calculate total number of subnets and numbers of hosts per subnet.

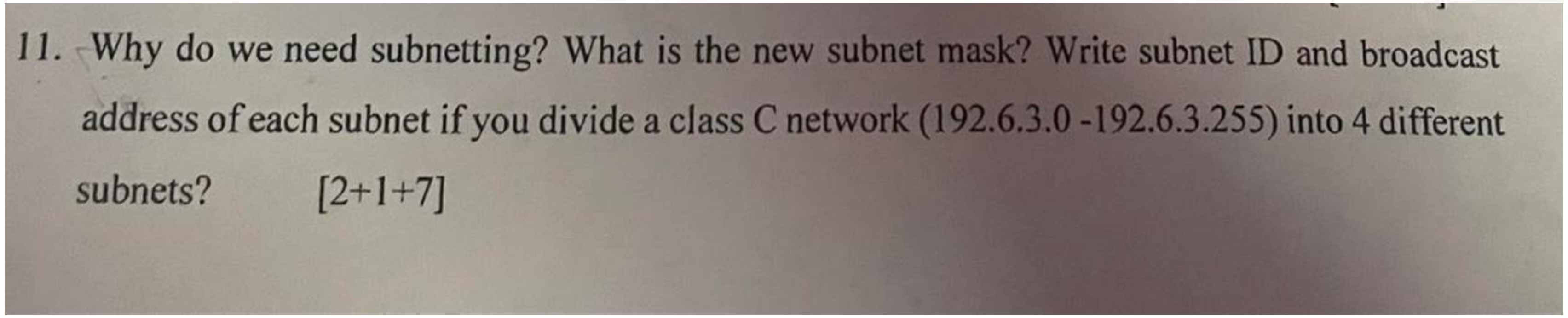
Subnet the IP address 216.215.5.0 into 30 hosts in each subnet

IP Address Space: 209.44.33.0/24
Required Number of Subnet = 6



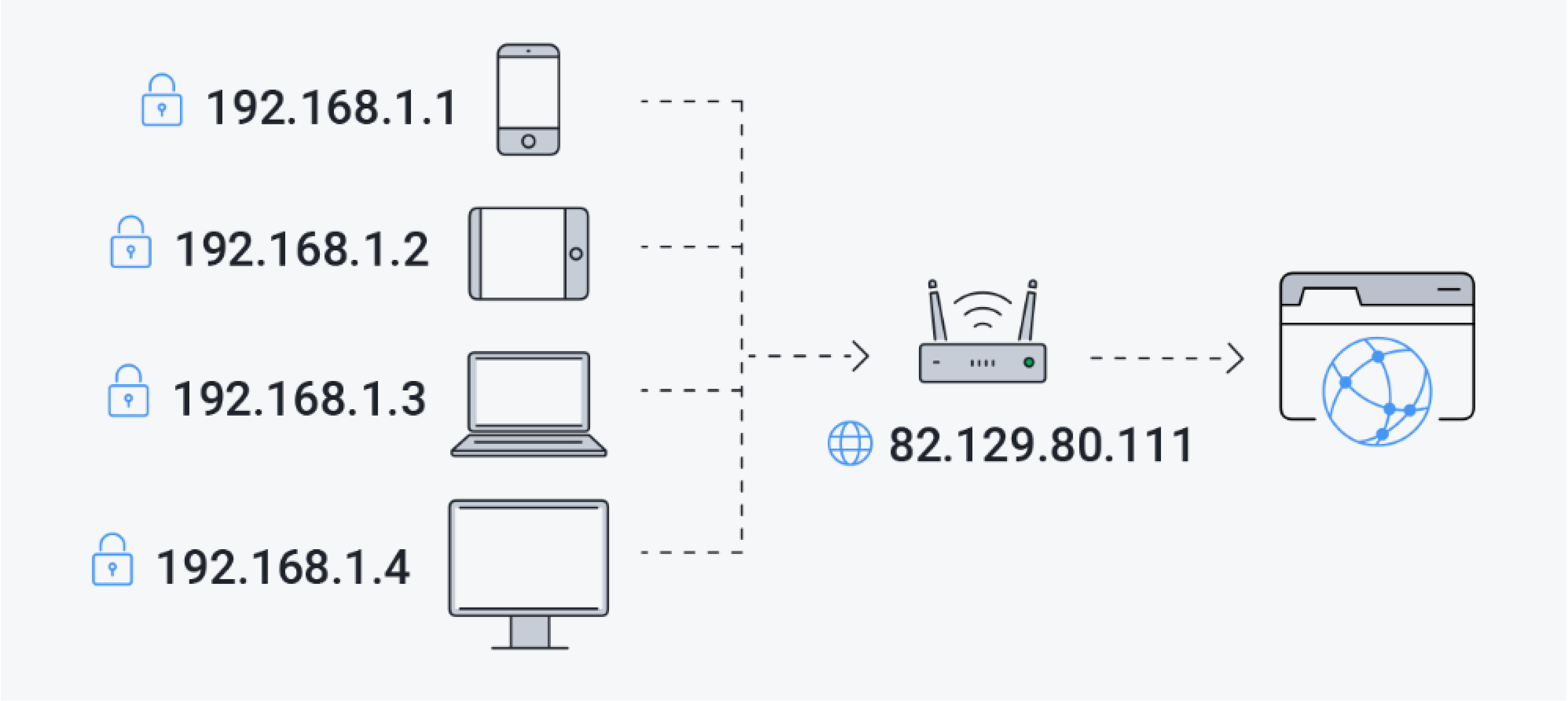
Here is a class C example demonstrating everything. The given IP address space is 209.44.33.0 /24. The required number of subnets is 6. The table below shows the six-step subnetting process.

Step 1–Given IP Address Space	209.44.33.0 (Class C) Default Subnet Mask = 255.255.255.0 Default Number of Host Bits = 8		
Step 2–Number of Required Subnets	6		
Step 3A–Determine Number of Bits to Borrow	3 ($2^n > 6$, where n = the number of host bits to borrow. n=3 ($2^3 = 8$))		
Step 3B–Determine Number of Hosts per Subnet	30 (Number of host bits available) - n = h 8 - 3 = 5 h = 5 $2^h - 2$ = (Number of hosts per subnet) $2^5 = 32 - 2 = 30$		
Step 4–Calculate new subnet mask and prefix	Default Class C Subnet Mask = 255.255.255.0 Default Class C Prefix = /24 New Subnet Mask = 255.255.255.224 New Prefix = /27		
Step 5–Apply the Subnet Mask to the IP Space	Subnet	Usable Range	Broadcast
	209.44.33.0	209.44.33.1 - 30	209.44.33.31
	209.44.33.32	209.44.33.33 - 62	209.44.33.63
	209.44.33.64	209.44.33.65 - 94	209.44.33.95
	209.44.33.96	209.44.33.97 - 126	209.44.33.127
	209.44.33.128	209.44.33.129 - 158	209.44.33.159
	209.44.33.160	209.44.33.161 - 190	209.44.33.191
	209.44.33.192	209.44.33.193 - 222	209.44.33.223
	209.44.33.224	209.44.33.225 - 254	209.44.33.255



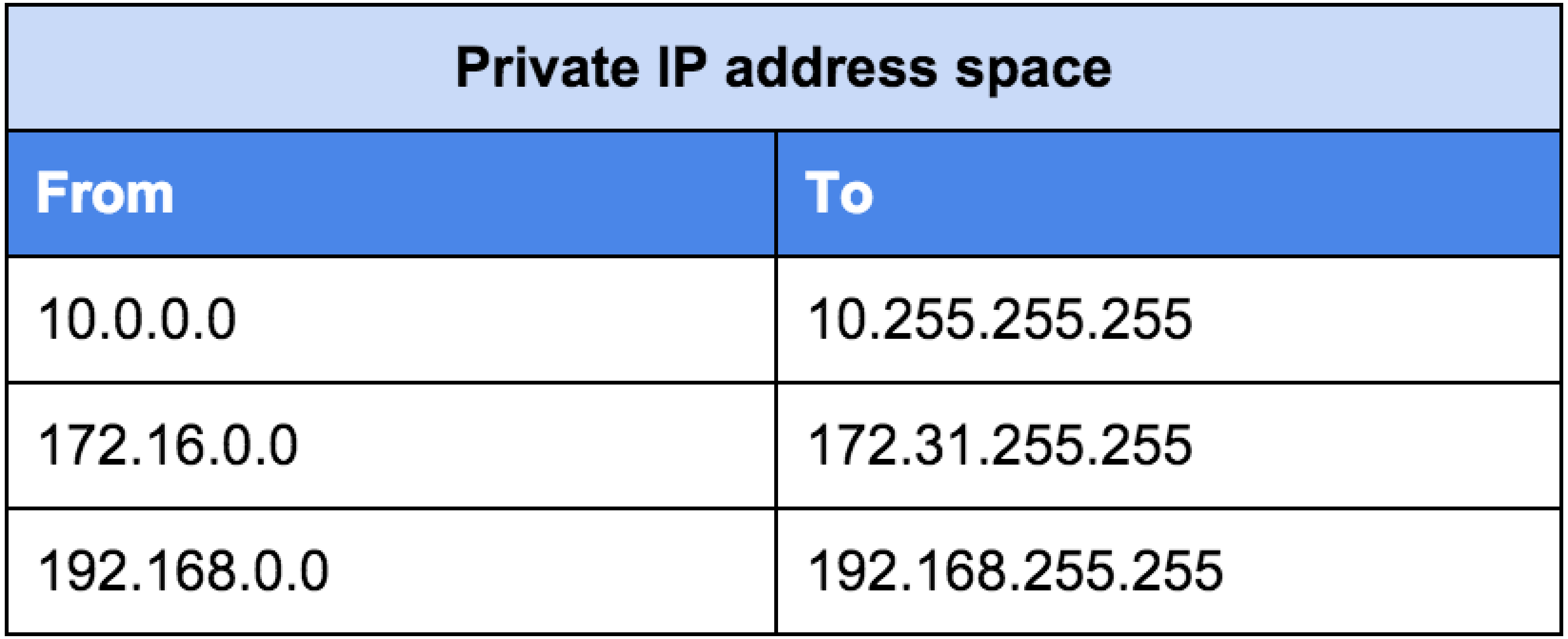
11. Why do we need subnetting? What is the new subnet mask? Write subnet ID and broadcast address of each subnet if you divide a class C network (192.6.3.0 -192.6.3.255) into 4 different subnets? [2+1+7]

Private and Public IP Addresses:



Each device on a network has a private IP address, and the router has a public IP address to communicate with the rest of the internet.

- A public IP address is the address that is when assigned to a device, it can be directly accesses over the internet. All servers and publicly known devices has public IP addresses. A public IP address is globally unique and it can be obtained from ISPs.
- IPv4 has limited number of addresses and if each device is given a public IP then the IPv4 addressing scheme is not sufficient, to solve this problem each organization has one public IP to represent itself and all internal networks and devices ha private IP addresses. These are the addresses which any one can use for their personal use without any permissions. So this way number of addresses can be saved as each organizations only uses private IPs to have inter networking



Private IP address space	
From	**To**
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Private IP	Public IP
Used with LAN or Network	Used on Public Network
Not recognized over Internet	Recognized over Internet
Assigned by LAN administrator	Assigned by Service provider / IANA
Unique only in LAN	Unique Globally
Free of charge	Cost associated with using Public IP
Range – Class A -10.0.0.0 to 10.255.255.255 Class B – 172.16.0.0 to 172.31.255.255 Class C – 192.168.0.0 – 192.168.255.255	Range – Class A -1.0.0.0 to 9.255.255.255 11.0.0.0 – 126.255.255.255 Class B -128.0.0.0 to 172.15.255.255 172.32.0.0 to 191.255.255.255 Class C -192.0.0.0 – 192.167.255.255 192.169.0.0 to 223.255.255.255

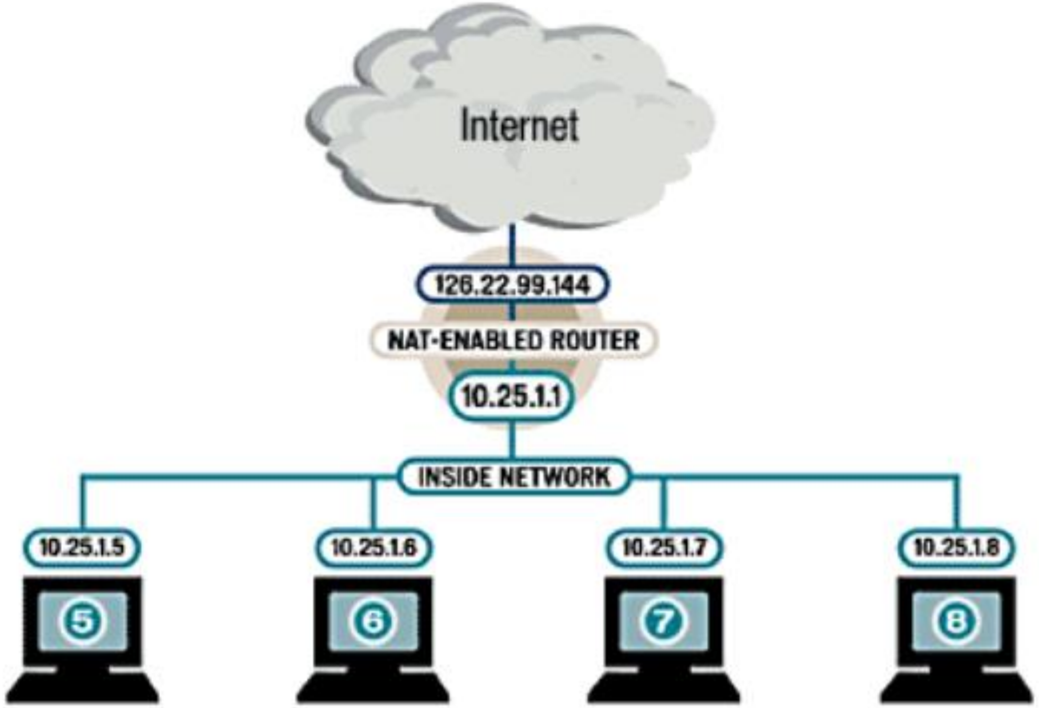
Network Address Translation (NAT)

- To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a public IP address to a private 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.
- Some benefits of NAT include:

❖Reuse of private IP addresses

❖Enhancing security for private networks by keeping internal addressing private from the external network

❖Connecting a large number of hosts to the global Internet using a smaller number of public (external) IP address, thereby conserving IP address space


Internet
126.22.99.144
NAT-ENABLED ROUTER
10.25.1.1
INSIDE NETWORK
10.25.1.5
10.25.1.6
10.25.1.7
10.25.1.8
5
6
7
8

Continued….

- There are three different types of NATs. People use them for different reasons, but they all still work as a NAT.

1. Static NAT

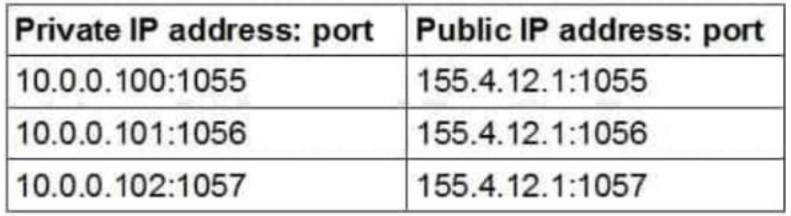
- In this, a single private IP address is mapped with a single Public IP address, i.e., a private IP address is translated to a public IP address. It is used in Web hosting.

2. Dynamic NAT

- Instead of choosing the same IP address every time, this NAT goes through a pool of public IP addresses. This results in the router or NAT device getting a different address each time the router translates the local address to a public address.

3. PAT

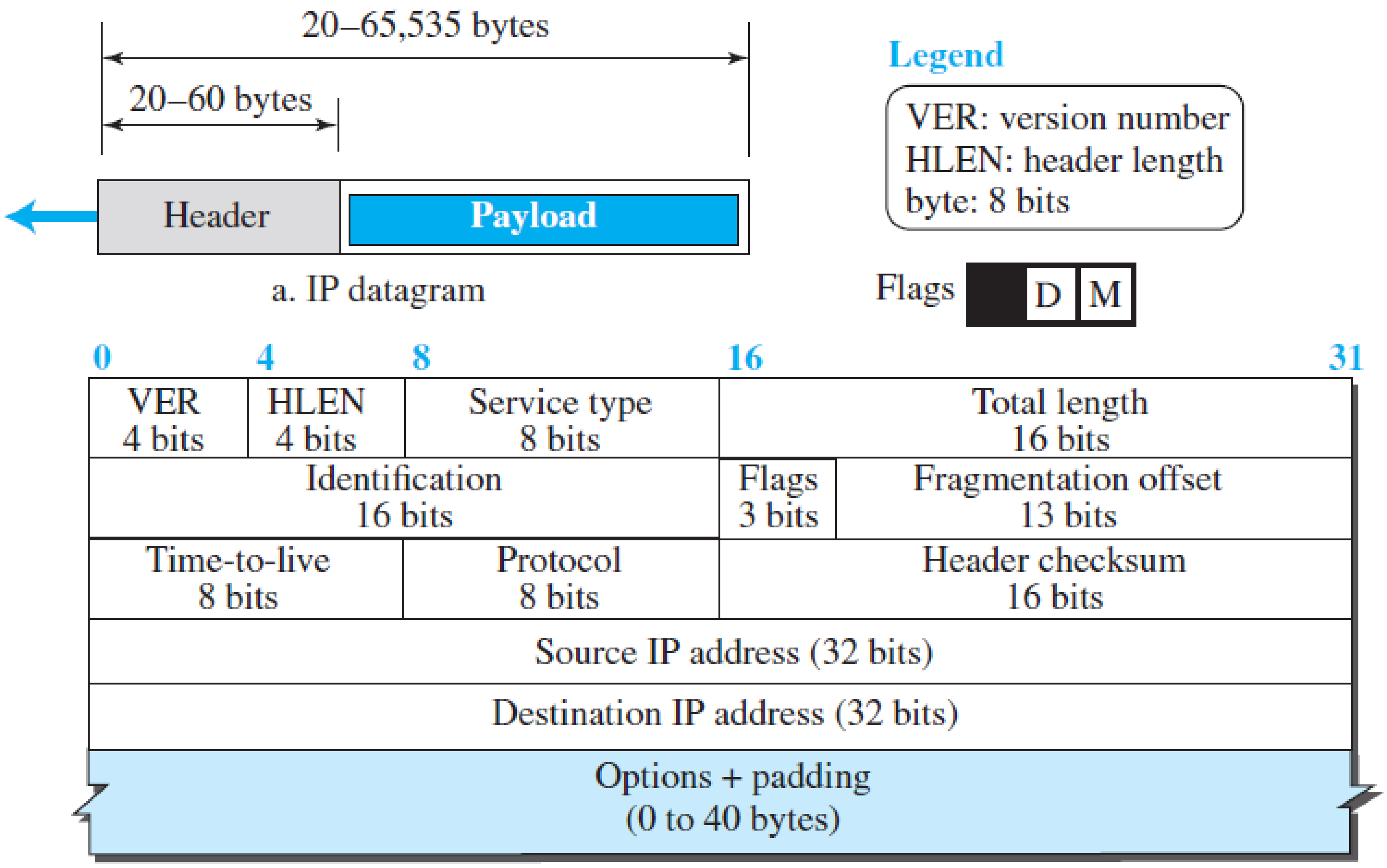
- In this, many local (private) IP addresses can be translated to a single public IP address. Port numbers are used to distinguish the traffic, i.e., which traffic belongs to which IP address. Also called NAT Overloading.



Private IP address: port	Public IP address: port
10.0.0.100:1055	155.4.12.1:1055
10.0.0.101:1056	155.4.12.1:1056
10.0.0.102:1057	155.4.12.1:1057

IPV4 Header Format:

- IPV4 header format is of 20 to 60 bytes in length, contains information essential to routing and delivery, consist of 13 fields where each has its own features and provides essential data required to transmit the data.



a. IP datagram

Legend

VER: version number
HLEN: header length
byte: 8 bits

0	4	8	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

b. Header

VER (Version): The first header field is a 4-bit version indicator. In the case of IPv4, the value of its four bits is set to 0100, which indicates 4 in binary.

HLEN (Header Length): 2nd field of an IPv4 header, and it is of 4 bits in size. This header component is used to show how many bytes are present in the header. This size can be between 20 bytes to 60 bytes. The receiver needs to multiply the value of this field by 4 to find the total length.

Service type: Also called Differentiated Services Code Point or DSCP. This field is used to provide features related to service quality, such as for data streaming or Voice over IP (VoIP) calls. It is used to specific how a datagram will be handled.

Total Length: This field's size is 16 bit, and it is used to denote the size of the entire datagram. The minimum size of an IP datagram is 20 bytes, and at the maximum, it can be 65,535 bytes.

Identification, Flags and Fragmentation Offset – Parts of Fragmentation.

Time to live: Time to live (or TTL in short) is an 8-bit field to indicate the maximum time the datagram will be live in the internet system. The time here is measured in seconds, and in case the value of TTL is zero, the datagram is erased. Every time a datagram is processed, it's Time to live is decreased by one second. These are used so that datagrams that are not delivered are discarded automatically.

Protocol: This is a filed in the IPv4 header reserved to denote which protocol is used in the later (data) portion of the datagram. For Example, number 6 is used to denote TCP and 17 is used to denote UDP protocol.

Header's checksum: The checksum field is of 16-bit length, and it is used to check the header for any errors. The header is compared to the value of its checksum at each hop, and in case the header checksum is not matching, the packet is discarded.

Source Address: It is a 32-bit address of the source of the IPv4 packet.

Destination Address: the destination address is also 32 bit in size, and it contains the receiver's address.

Options: The IP datagram may contain zero, one, or more options, which makes the total length of the Options field in the IPv4 header variable. Each of the options can be either a single byte long, or multiple bytes in length, depending on how much information the option needs to convey. These options contain values and settings for things related to security, Record route and time stamp etc.

1. An IPv4 packet has arrived with the first 8 bits as (01000010)2 The receiver discards the packet. Why?

Ans: There is an error in this packet. The 4 leftmost bits (0100)2 show the version, which is correct. The next 4 bits (0010)2 show an invalid header length (2 × 4 = 8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

IPv6 address format

- An IPv6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbols.
- For example, given below is a 128 bit IPv6 address represented in binary format and divided into eight 16-bits blocks:

  0010000000000001 0000000000000000 0011001000111000 1101111111100001
  0000000001100011 0000000000000000 0000000000000000 1111111011111011

- Each block is then converted into Hexadecimal and separated by ‘:’ symbol:

  2001:0000:3238:DFE1:0063:0000:0000:FEFB

- Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. The rules are as follows:
- **Rule.1:** Discard leading Zero(es):
- In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

Continued….

2001:0000:3238:DFE1:63:0000:0000:FEFB

- **Rule.2:** If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

2001:0000:3238:DFE1:63::FEFB

Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as (2nd block):

2001:0:3238:DFE1:63::FEFB

Assignment:

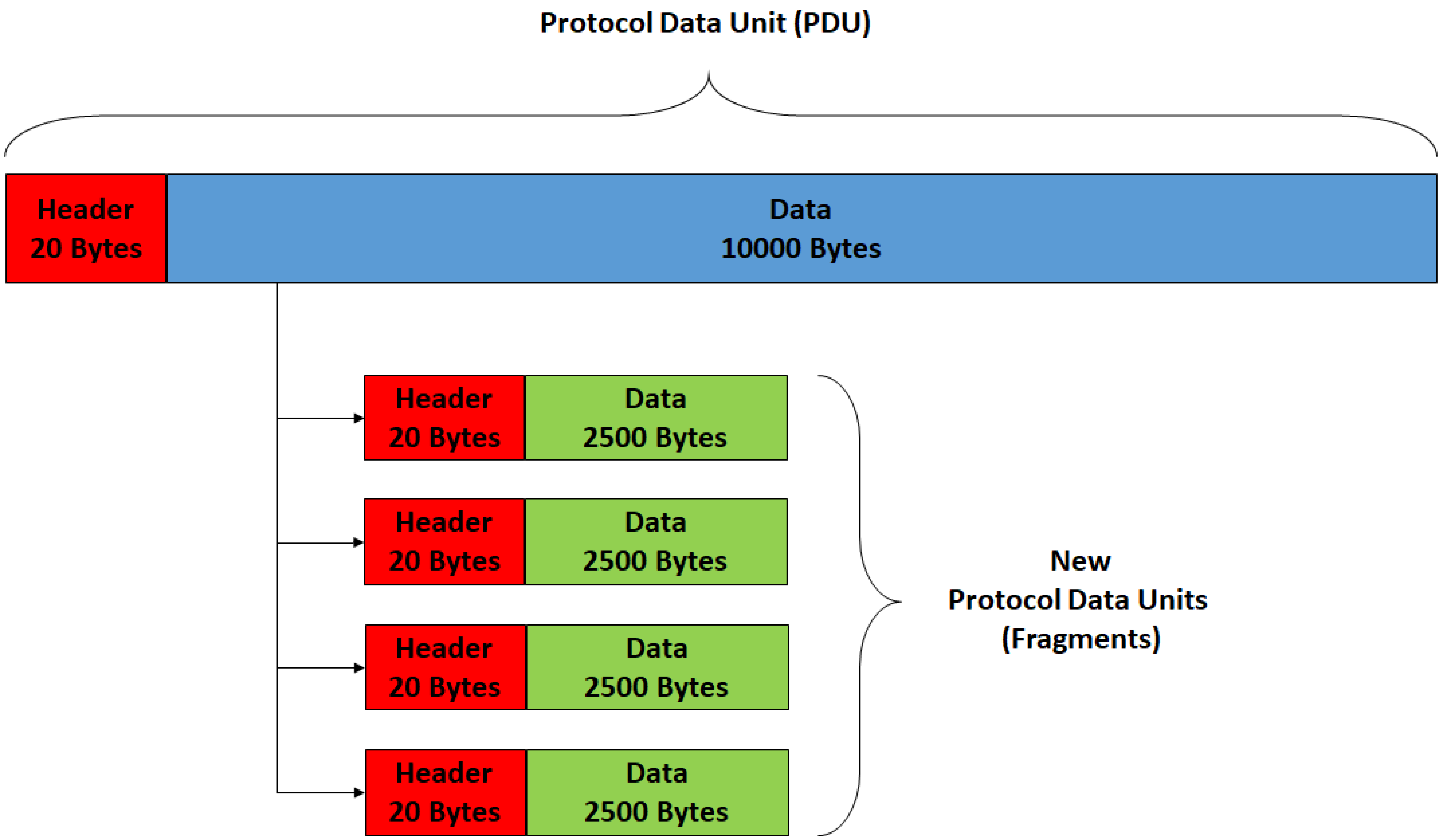
1. Difference between IPV6 and IPV4
2. Advantages of IPV6 over IPV4

Fragmentation:

- Different Networks may have different maximum transmission unit (MTU), for example due to differences in LAN technology. When one network wants to transmit datagrams to a network with a smaller MTU, the routers on path may fragment and reassemble datagrams.

- When a packet is received at the router, destination address is examined and MTU is determined. If size of the packet is bigger than the MTU, and the ‘Do not Fragment (DF)’ bit is set to 0 in header, then the packet is fragmented into parts and sent one by one.

- An important point to be noted here is that all fragments would be having same identification number, thus indicating that all the fragments belong to the same parent data packet.



Protocol Data Unit (PDU)

Header
20 Bytes

Data
10000 Bytes

Header
20 Bytes

Data
2500 Bytes

Header
20 Bytes

Data
2500 Bytes

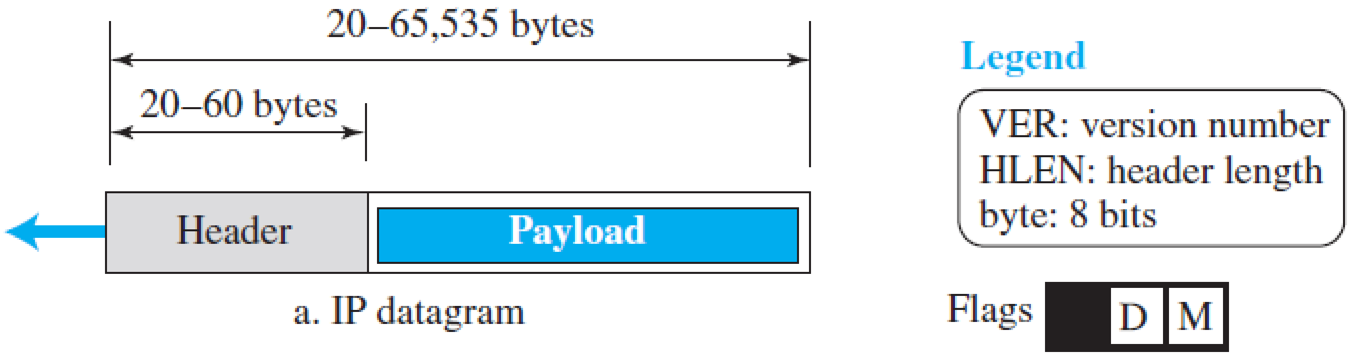
Header
20 Bytes

Data
2500 Bytes

Header
20 Bytes

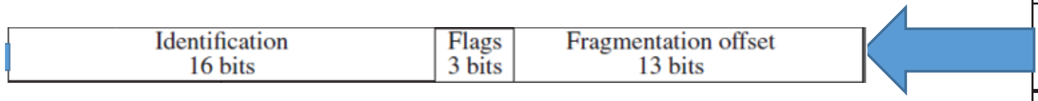
Data
2500 Bytes

New
Protocol Data Units
(Fragments)



a. IP datagram

Fields Related to Fragmentation



0	4	8	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

b. Header

Identification: The 16-bit *identification field* identifies a datagram originating from the source host. When a datagram is fragmented, the value in the identification field is copied into all fragments. In other words, all fragments have the same identification number, which is also the same as the original datagram. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value should be assembled into one datagram.

Flags: The 3-bit *flags field* defines three flags. The leftmost bit is reserved (not used). The second bit (D bit) is called the *do not fragment* bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host . If its value is 0, the datagram can be fragmented if necessary. The third bit (M bit) is called the *more fragment bit*. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation Offset: The 13-bit *fragmentation offset field* shows the relative position of this fragment with respect to the whole datagram

A datagram of 3000 bytes (20 bytes of IP header + 2980 bytes IP payload) reached the router and must be forwarded to link with MTU (maximum transmission unit) of 500 bytes. How many fragments will be granted? Also, write MF, offset, and total length values for all.

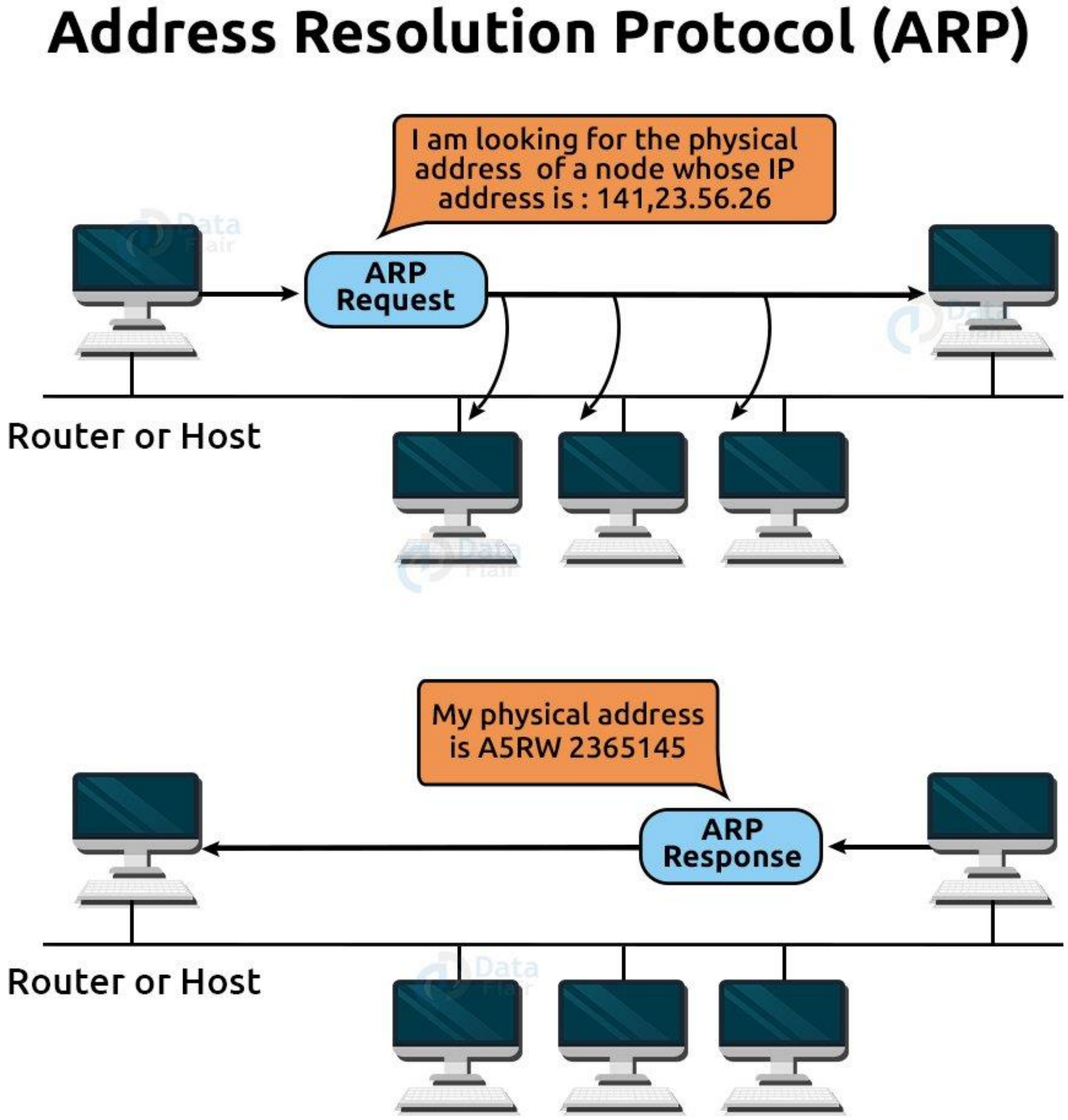
Refer: https://cstaleem.com/ipv4-datagram-header-numerical

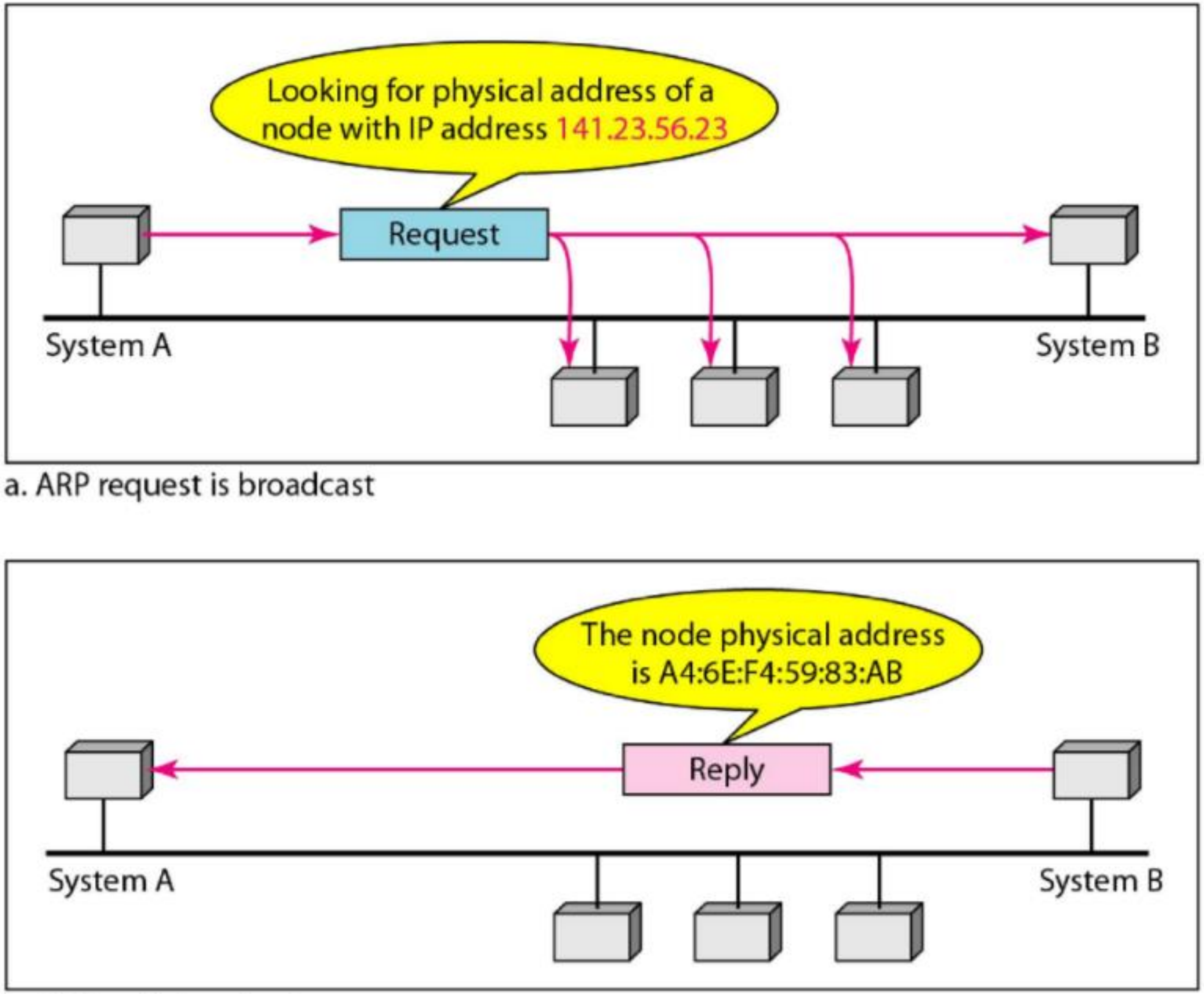
Internet Control Protocols

1. Address Resolution Protocol (ARP)
2. Reverse Address Resolution Protocol (RARP)
3. Internet Control Message Protocol (ICMP)

Address Resolution Protocol

- Address Resolution Protocol is a communication protocol used for discovering physical address(MAC Address) associated with given network address (IP Address). Typically, ARP is a network layer to data link layer mapping process, which is used to discover MAC address for given Internet Protocol Address. In order to send the data to destination, having IP address is necessary but not sufficient; we also need the physical address of the destination machine.


Address Resolution Protocol (ARP)
I am looking for the physical address of a node whose IP address is : 141,23.56.26
ARP Request
Router or Host
My physical address is A5RW 2365145
ARP Response
Router or Host

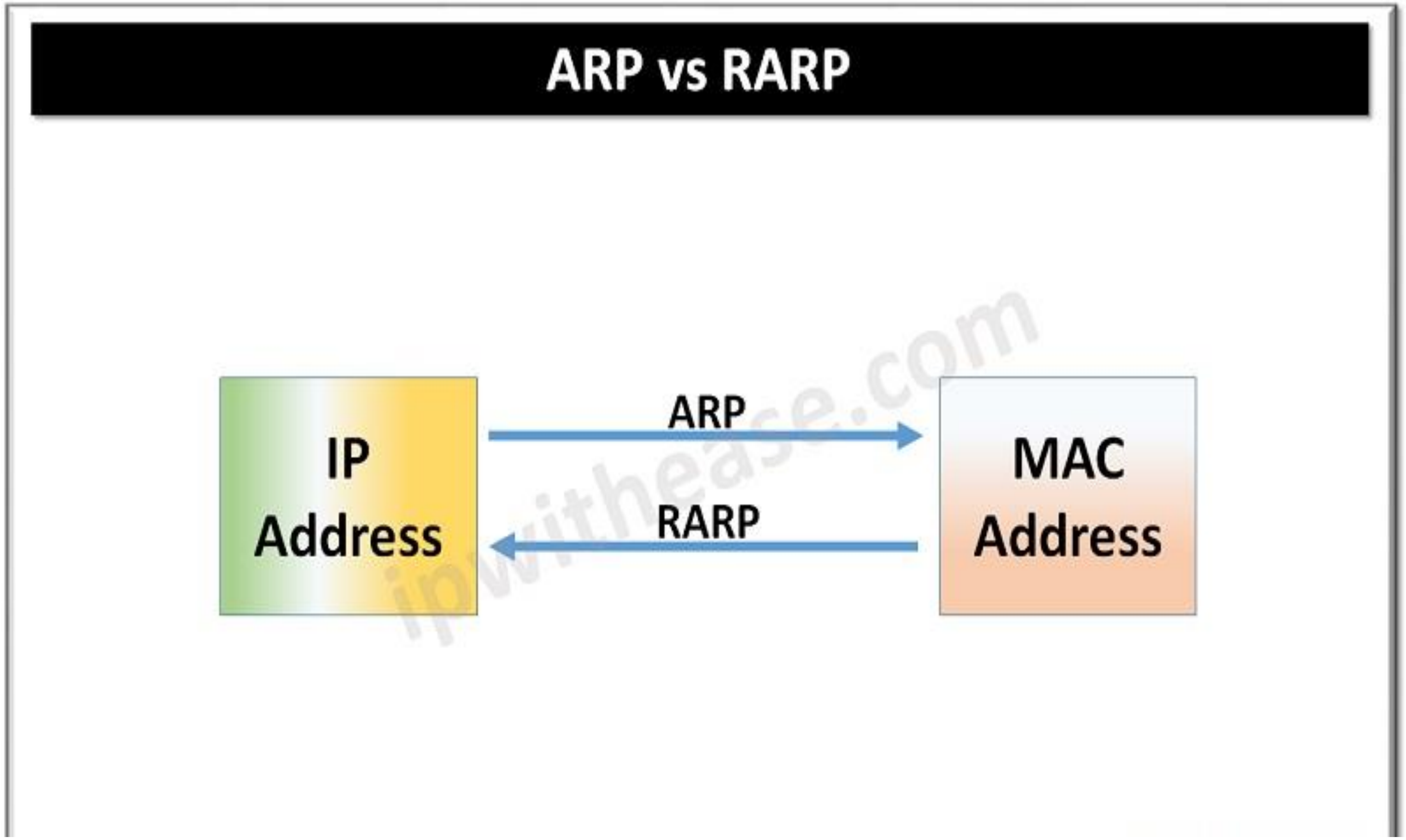

Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

a. ARP request is broadcast

b. ARP reply is unicast

Reverse ARP (RARP)

- It is a networking protocol used by the client system in a local area network (LAN) to request its IPv4 address from the ARP gateway router table. A table is created by the network administrator in the gateway-router that is used to find out the MAC address to the corresponding IP address.
- When a new system is set up or any machine that has no memory to store the IP address, then the user has to find the IP address of the device. The device sends a RARP broadcast packet, including its own MAC address in the address field of both the sender and the receiver hardware.



- A host installed inside of the local network called the RARP-server is prepared to respond to such type of broadcast packet. The RARP server is then trying to locate a mapping table entry in the IP to MAC address. If any entry matches the item in the table, then the RARP server sends the response packet along with the IP address to the requesting computer.

Internet Control Message Protocol (ICMP)

- Since IP does not have an inbuilt mechanism for sending error and control messages. It depends on Internet Control Message Protocol(ICMP) to provide an error control. It is used for reporting errors and management queries. It is a supporting protocol and is used by networks devices like routers for sending error messages and operations information. e.g. the requested service is not available or that a host or router could not be reached.
- ICMP messages are sent in several scenarios. For example, if one device sends a message that is too large for the recipient to process, the recipient will drop that message and send an ICMP message back to the source. Another example is when the network gateway finds a shorter route for the message to travel on. When this happens, an ICMP message is sent, and the packet is redirected to the shorter route.
- ICMP is also used for network diagnostics, specifically the ping and traceroute terminal utilities.

Continued…..

- The ICMP messages are usually divided into two categories:

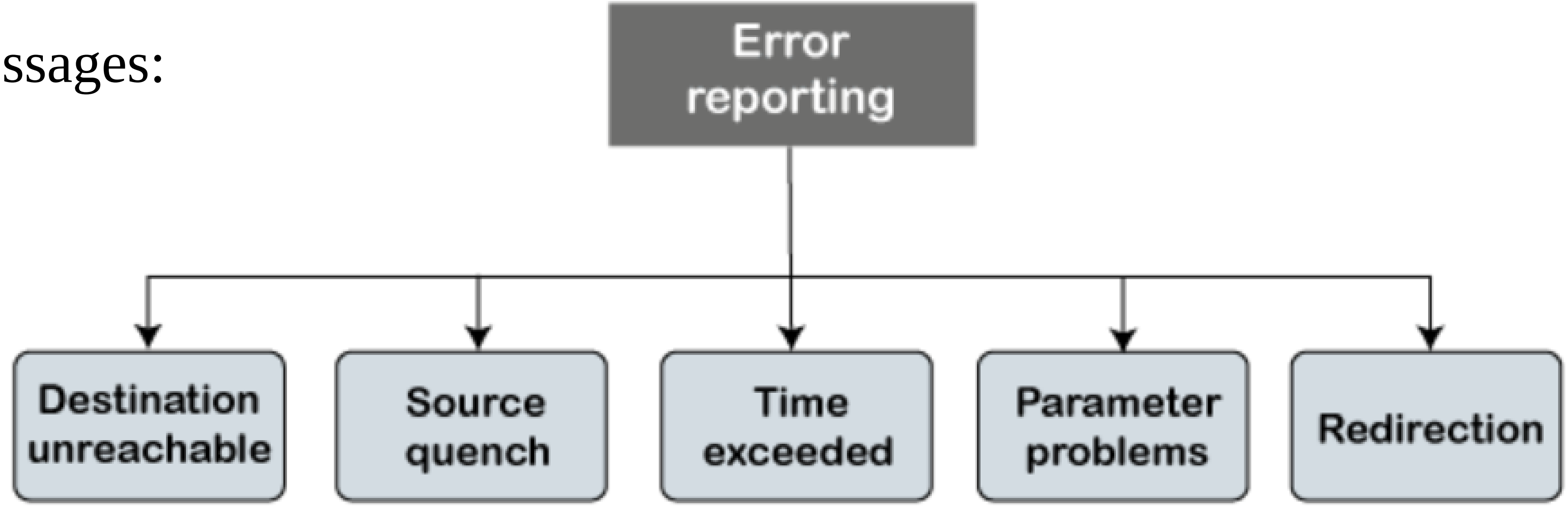
✓Error-reporting messages

- The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

✓Query messages

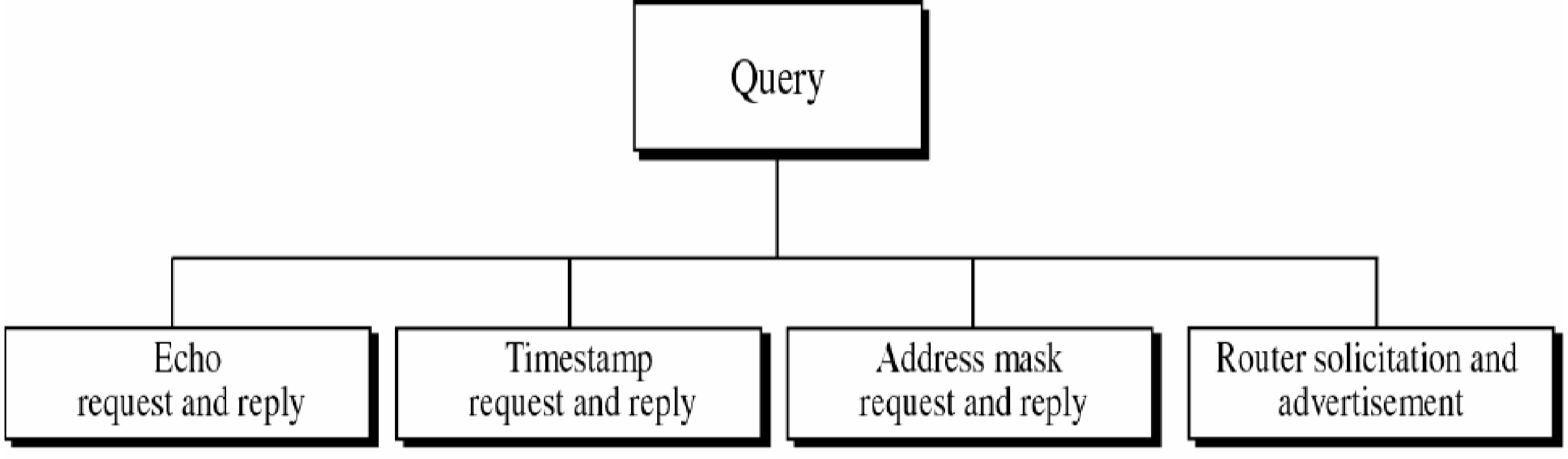
- The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Error Reporting Messages:



- **Destination un-reachable :** Destination unreachable is generated by the host or its inbound gateway to inform the client that the destination is unreachable for some reason.
- **Source quench message :** When receiving host detects that the rate of sending packets (traffic rate) to it is too fast, it sends the source quench message to the source to slow the pace down so that no packet can be lost.
- **Time Exceeded**: This message indicates that the Time-To-Live value of the datagram has reached zero but the datagram has not yet been reached the final destination.
- **Parameter Problem**: If a device finds a problem that is not covered in any ICMP message type, it sends a parameter problem message to the sender.
- **Redirect:** This error message is used when a router needs to tell a sender that it should use a different path for a particular destination. Usually, it happens when the router knows a shorter path to the destination.

ICMP Query Messages


Query
Echo request and reply
Timestamp request and reply
Address mask request and reply
Router solicitation and advertisement

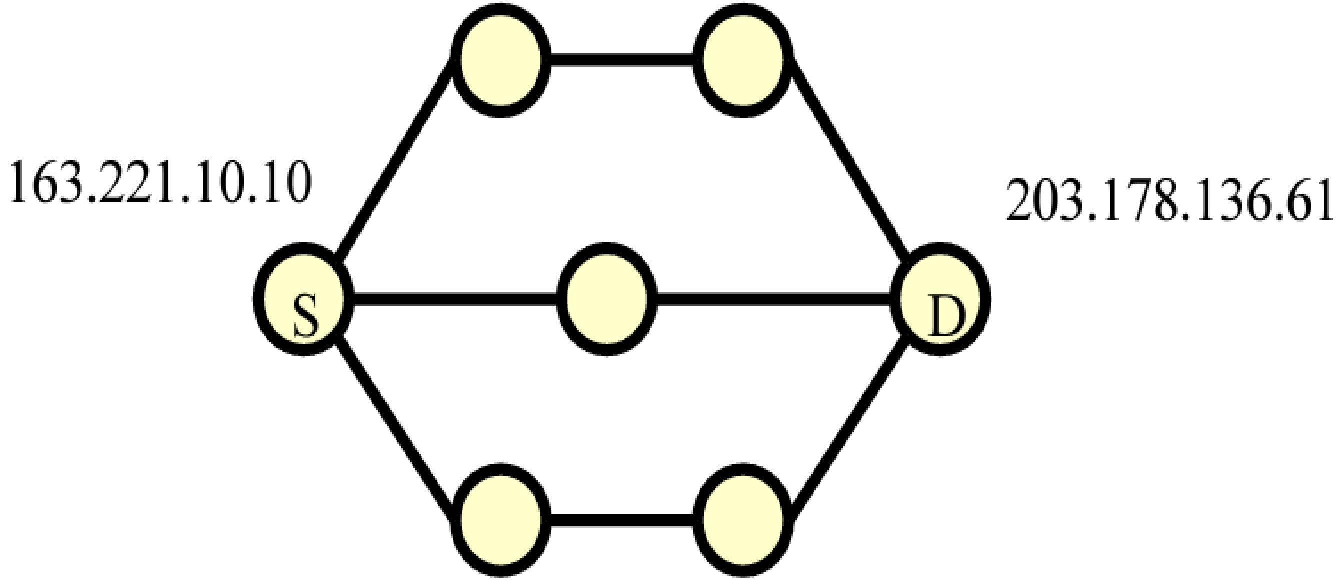
- **Echo Request and Reply:** Echo request or reply is the first step towards checking if the destination device is alive or not. To check it, the source device sends an ICMP Echo message to the destination. Upon receiving the Echo request, the destination device replies with “Echo Reply”.
- **Timestamp Request and Reply:** ICMP timestamp request and reply messages are used to sync up the timings.
- **Address Mask Request and Reply:** Address Mask request or reply is used for finding out the subnet address of the destination network where the packet has to be sent.
- **Router Solicitation and Advertisement:** If there is an update in the routing table of a router, those updates are communicated or advertised via the ICMP messages only. Not only the routing table update but messages are also communicated via ICMP messages between two routers.

Routing algorithm

- In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted.
- Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job.
- The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.
- Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.

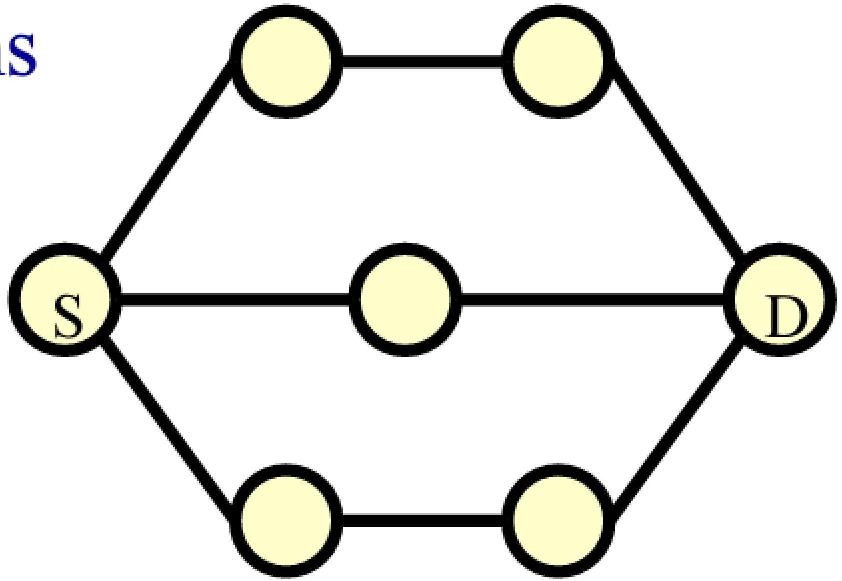
The Routing Problem

- How do I get from source to destination?



- Which path is best? In terms of:
  - Number of hops
  - Delay, bandwidth
  - Policy constraints, cost…
- Who will make decision?
  - Router?
  - Source?
- How can we detect failures?

Characterization of routing systems



- Static routing
  - Compute route *a priori*
- Dynamic routing
  - Reflect dynamic state of network

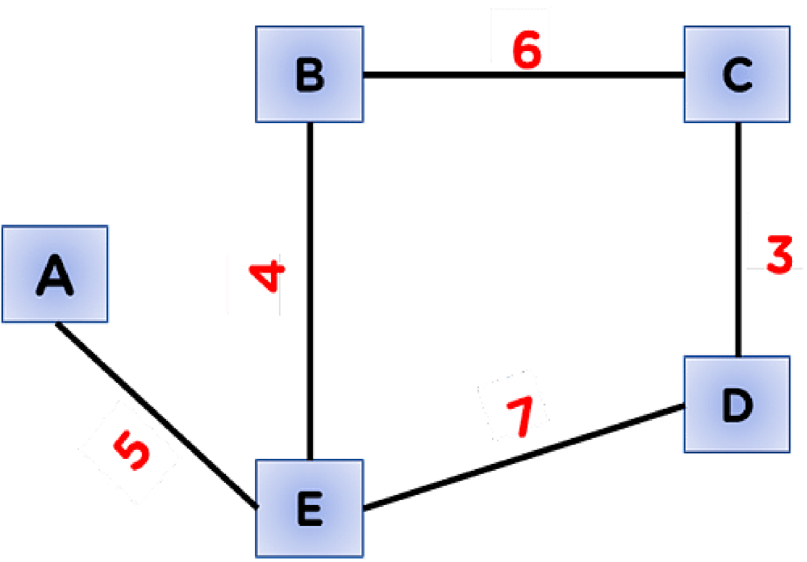
- Source-based routing
  - Source node computes path to destination
- Hop-by-hop routing
  - Every node computes next hop

Distance Vector Routing

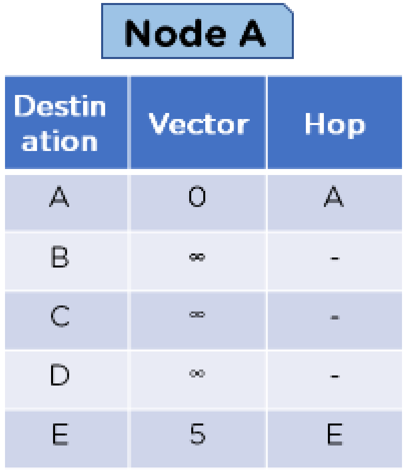
- The distance vector routing protocol is applied to assign the best and the shortest route for the data. In this network protocol, the distance refers to the distance (vector) between neighboring nodes, and the routing refers to the established route.

- Key Features

- Every router is responsible for sharing the network knowledge in the channel more responsibly with the neighboring nodes.
- Sharing of routing information is done only between directly connected network nodes in the channel.
- Each node in the connection is designed to share the updated routing data with each of the nodes in the network.

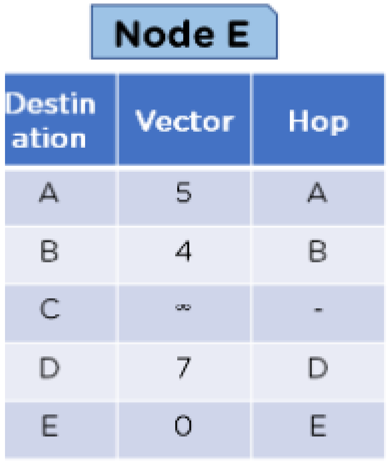


Initial Step



Node A

Destination	Vector	Hop
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	5	E

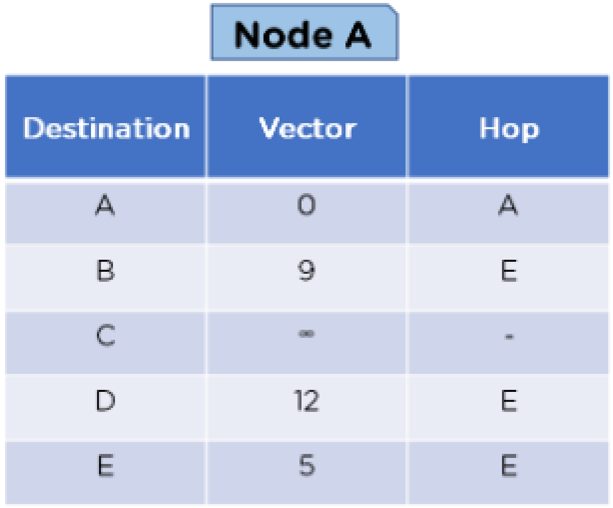


Node E

Destination	Vector	Hop
A	5	A
B	4	B
C	∞	-
D	7	D
E	0	E

Design the routing table for each of the nodes in the same way.

Update Step



Node A

Destination	Vector	Hop
A	0	A
B	9	E
C	∞	-
D	12	E
E	5	E

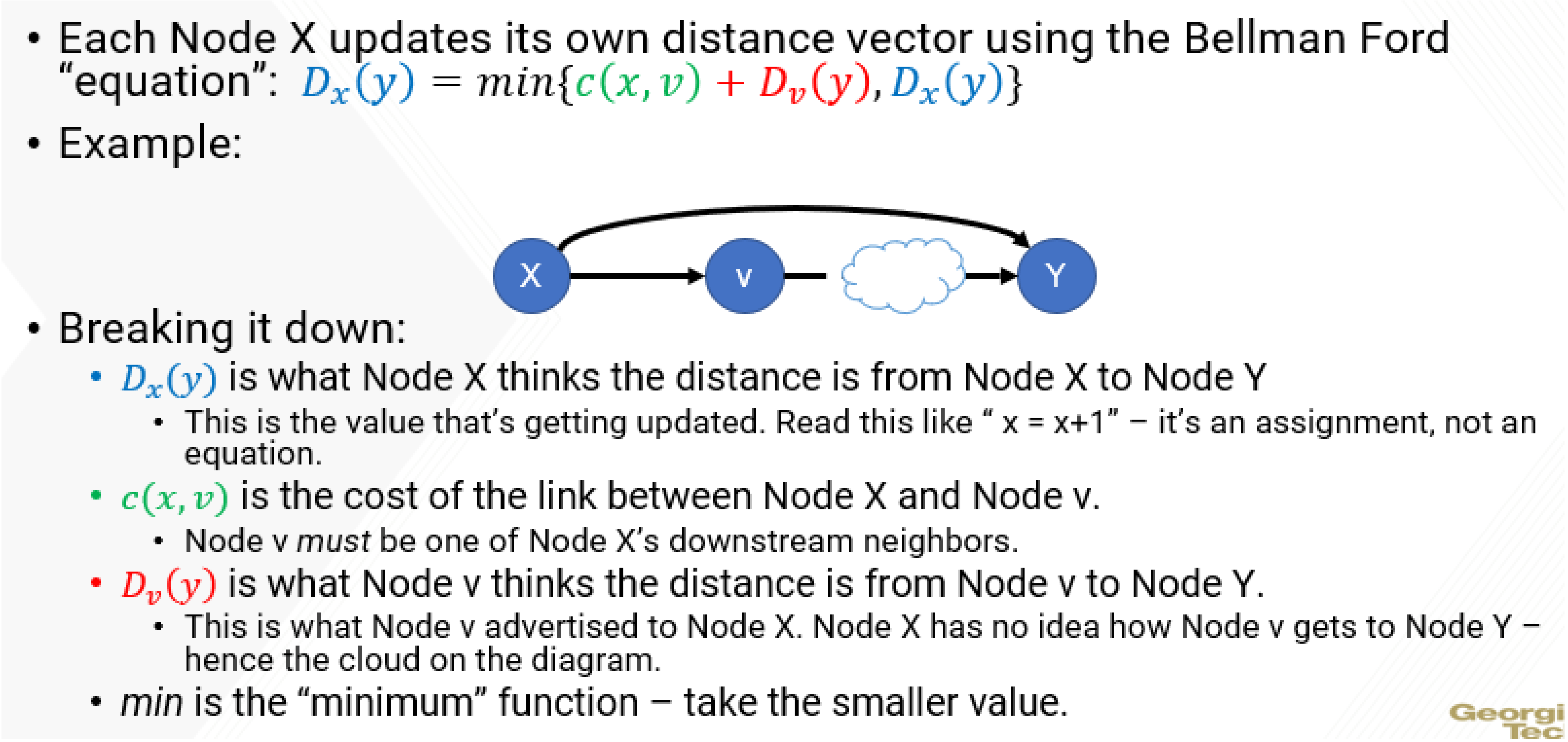
Similarly, we can perform the update step for all the nodes in the model, and this update step is to be followed for (n-1) iterations where n - Number of nodes. Going by our example model, we will perform the update step at least four times, i.e., (5-1) = 4. At the end of the update step, we will get the most efficient routing data for each node in the network model, where the sharing of routing data at regular intervals will still continue in the network.

Important terms of Distance Vector Routing Algorithm

- **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.
- **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The router sends whatever it has about the network through the ports. The information is received by the router and uses the information to update its own routing table.
- **Information sharing at regular intervals:** The router sends the information to the neighboring routers at regular intervals.

Bellman Ford Equation:

Distance Vector Routing uses the Bellman-Ford equation to update its own distance vector.

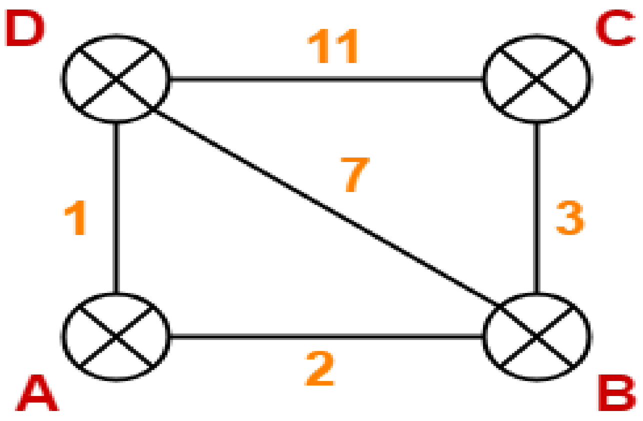


- Each Node X updates its own distance vector using the Bellman Ford "equation": $D_x(y) = min\{c(x,v) + D_v(y), D_x(y)\}$
- Example:

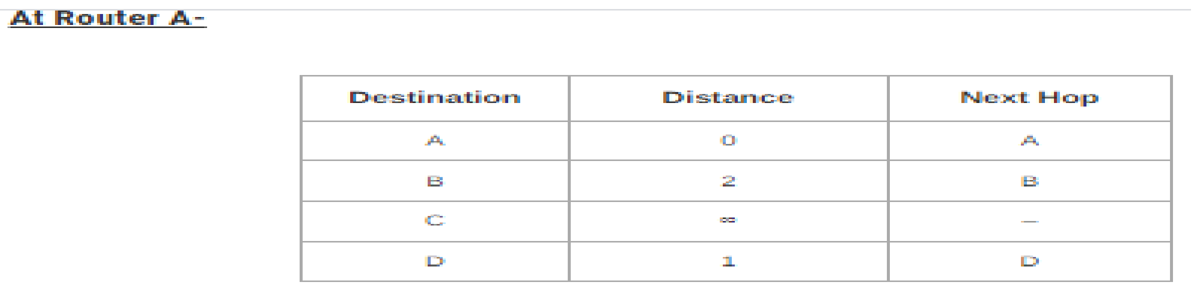
- Breaking it down:
  - $D_x(y)$ is what Node X thinks the distance is from Node X to Node Y
    - This is the value that's getting updated. Read this like " x = x+1" – it's an assignment, not an equation.
  - $c(x,v)$ is the cost of the link between Node X and Node v.
    - Node v *must* be one of Node X's downstream neighbors.
  - $D_v(y)$ is what Node v thinks the distance is from Node v to Node Y.
    - This is what Node v advertised to Node X. Node X has no idea how Node v gets to Node Y – hence the cloud on the diagram.
  - *min* is the "minimum" function – take the smaller value.

Example

- There is a network consisting of 4 routers. Weights could be distances or costs or delays.

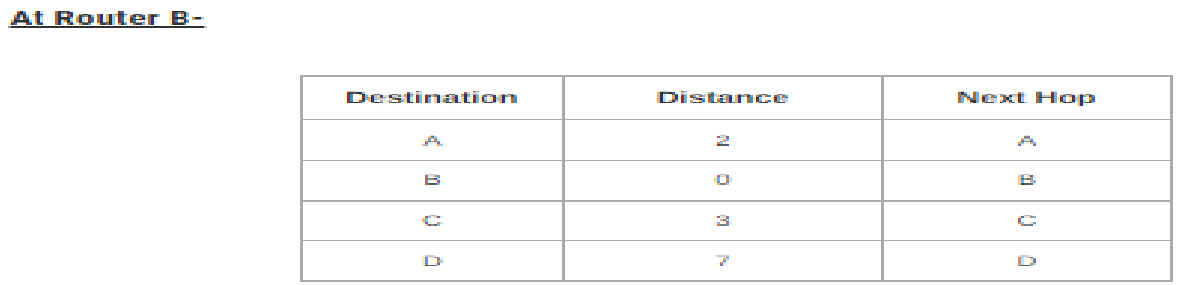


Step 1:
Each router prepares its routing table using its local knowledge.



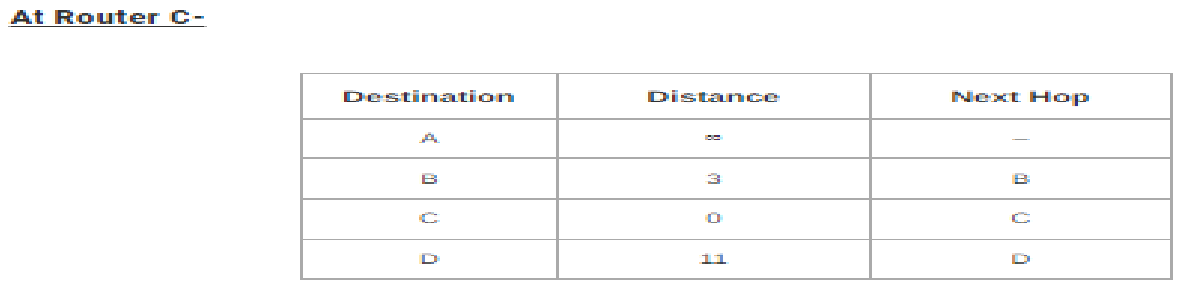
At Router A-

Destination	Distance	Next Hop
A	0	A
B	2	B
C	∞	–
D	1	D



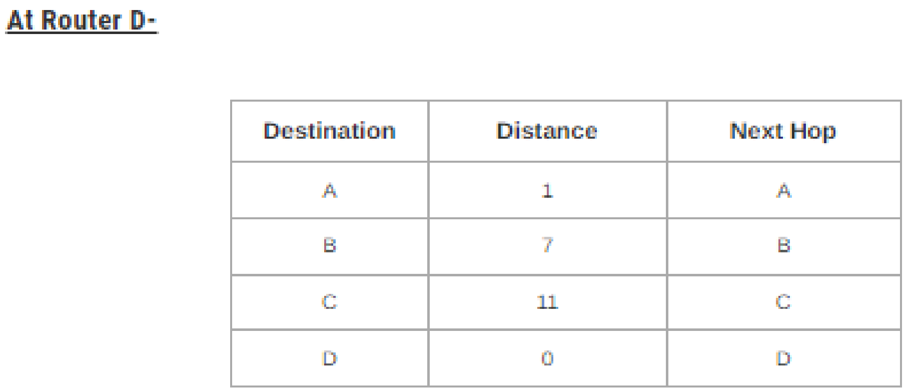
At Router B-

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	7	D



At Router C-

Destination	Distance	Next Hop
A	∞	–
B	3	B
C	0	C
D	11	D



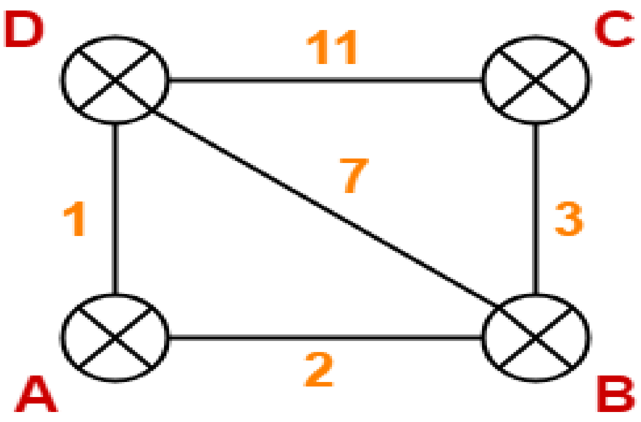
At Router D-

Destination	Distance	Next Hop
A	1	A
B	7	B
C	11	C
D	0	D

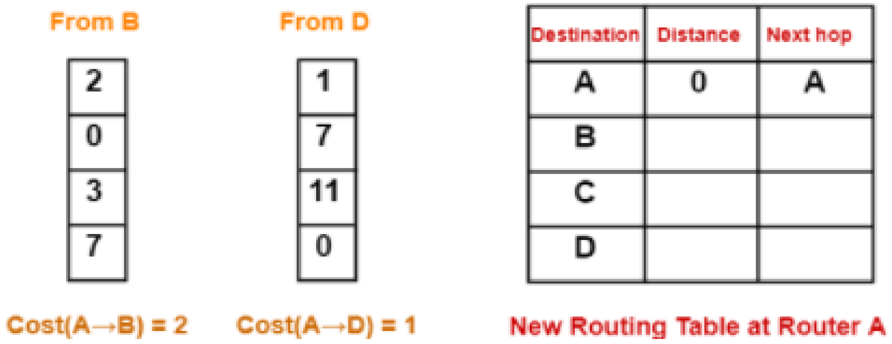
Continued….

Step 2:

Each router exchanges its distance vector obtained in Step-01 with its neighbors. After exchanging the distance vectors, each router prepares a new routing table.



- At Router A:
- Router A receives distance vectors from its neighbors B and D
- Router A prepares a new routing table as:



From B
2
0
3
7

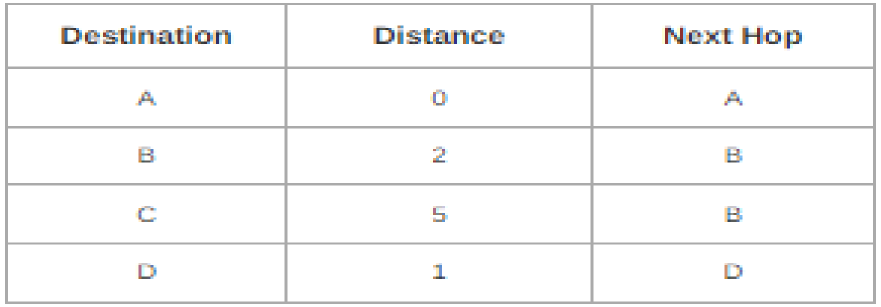
Cost(A→B) = 2

From D
1
7
11
0

Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A



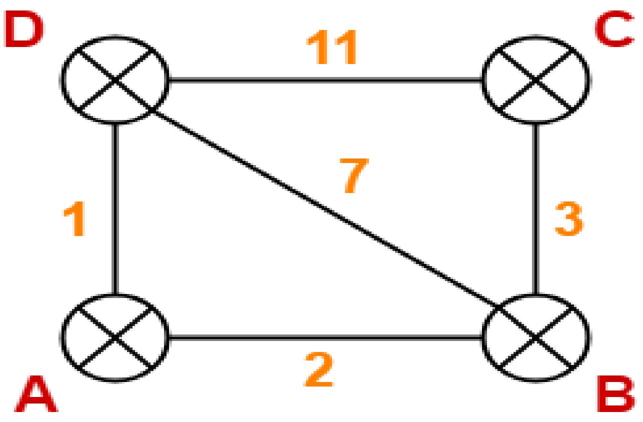
Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

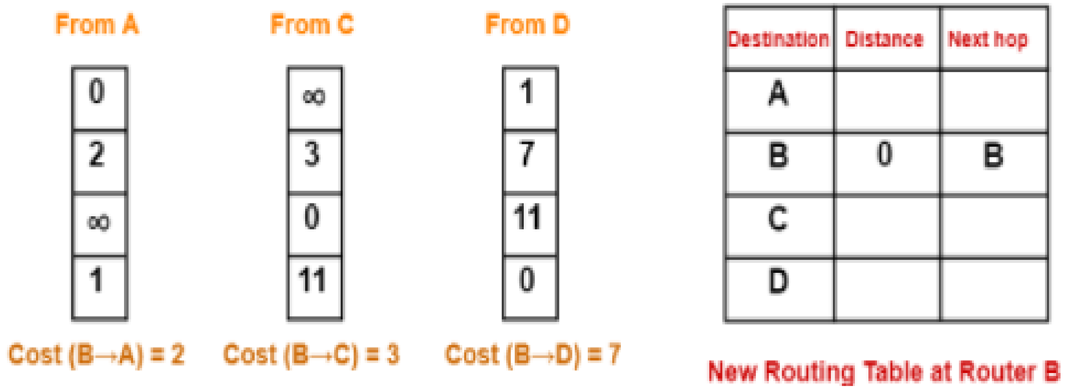
- Cost of reaching destination B from router A = min {2 + 0, 1 + 7} = 2 via B
- Cost of reaching destination C from router A = min {2 + 3, 1 + 11} = 5 via B
- Cost of reaching destination D from router A = min {2 + 7, 1 + 0} = 1 via D

Continued….

At Router B:

- Router B receives distance vectors from its neighbors A, C and D.

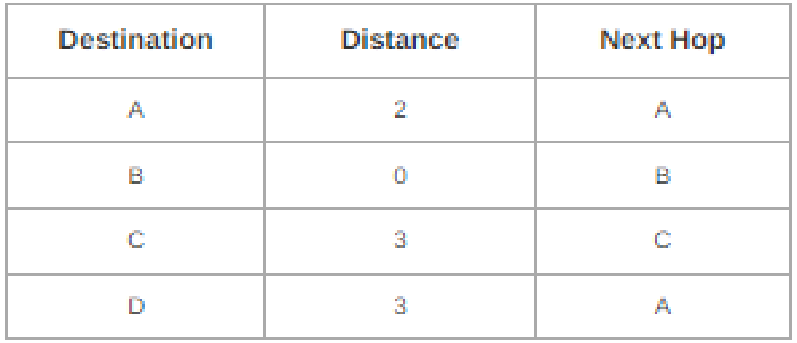




From A	From C	From D
0	∞	1
2	3	7
∞	0	11
1	11	0
Cost (B→A) = 2	Cost (B→C) = 3	Cost (B→D) = 7

Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B

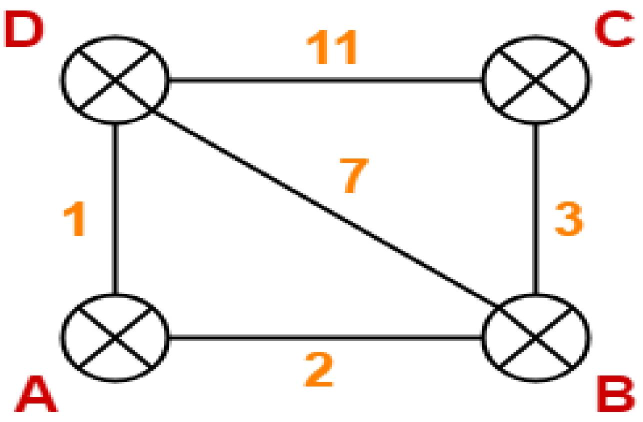


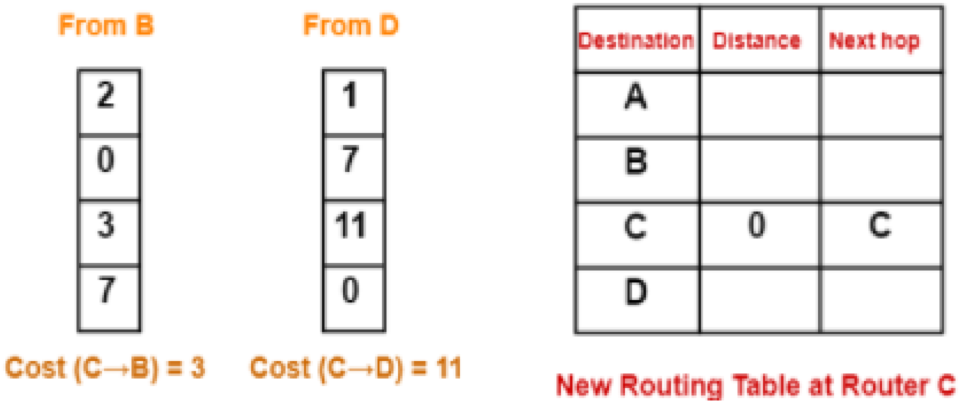
Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

Cost of reaching destination A from router B = min {2+0, 3 + ∞, 7+1} = 2 via A
Cost of reaching destination C from router B = min {2+ ∞, 3+0, 7+11} = 3 via C
Cost of reaching destination D from router B = min {2+1, 3+11, 7+0} = 3 via A

Continued….

- At Router C:
- Router C receives distance vectors from its neighbors B and D





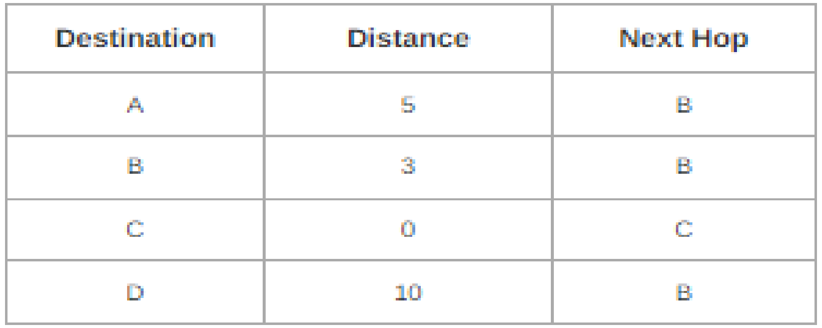
From B	From D
2	1
0	7
3	11
7	0

Cost (C→B) = 3

Cost (C→D) = 11

Destination	Distance	Next hop
A		
B		
C	0	C
D		

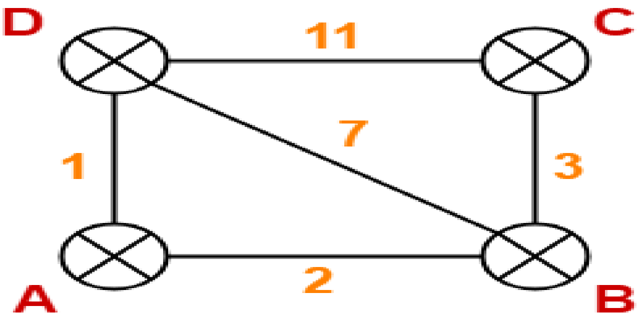
New Routing Table at Router C



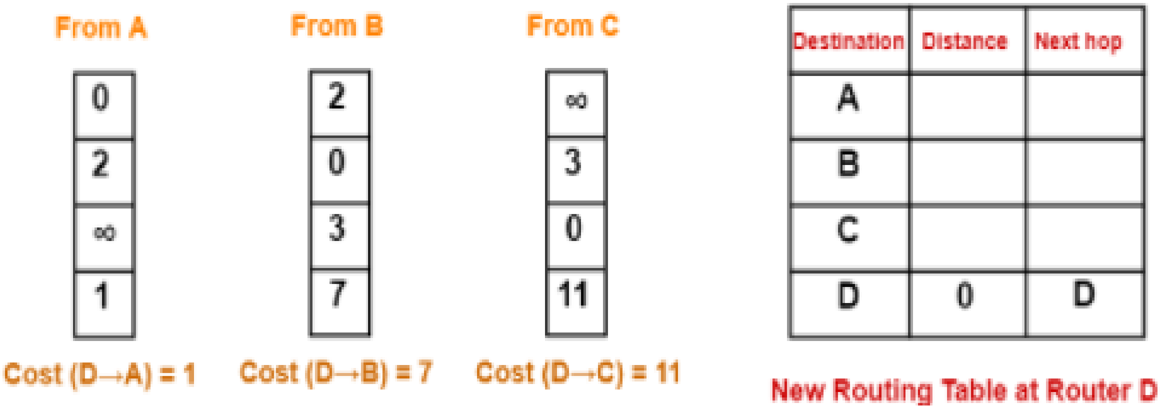
Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	10	B

- Cost of reaching destination A from router C = min {3+2, 11+1} = 5 via B
- Cost of reaching destination B from router C = min {3+0, 11+7} = 3 via B
- Cost of reaching destination D from router C = min {3+7, 11+0} = 10 via B

Continued….



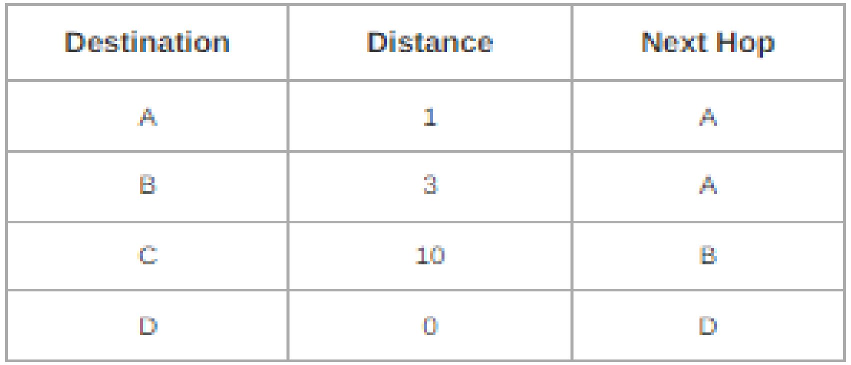
At Router D:
Router D receives distance vectors from its neighbors A, B and C.



From A	From B	From C
0	2	∞
2	0	3
∞	3	0
1	7	11
Cost (D→A) = 1	Cost (D→B) = 7	Cost (D→C) = 11

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D



Destination	Distance	Next Hop
A	1	A
B	3	A
C	10	B
D	0	D

- Cost of reaching destination A from router D = min {1+0, 7+2, 11+ ∞} = 1 via A
- Cost of reaching destination B from router D = min {1+2, 7+0, 11+3} = 3 via A
- Cost of reaching destination C from router D = min {1 + ∞, 7+3, 11+0} = 10 via B

Step 3:
Each router exchanges its distance vector obtained in Step-02 with its neighboring routers. After exchanging the distance vectors, each router prepares a new routing table.

Link State Routing

- Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.
- It is a dynamic routing algorithm in which each router shares knowledge of its neighbors with every other router in the network.
- A router sends its information about its neighbors only to all the routers through flooding.
- Information sharing takes place only whenever there is a change.
- It makes use of **Dijkstra's Algorithm** for making routing tables.
- **Problems:** Heavy traffic due to flooding of packets.
  – Flooding can result in infinite looping which can be solved by using the **Time to live (TTL)** field.
- **The three keys to understand the Link State Routing algorithm:**

❖**Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.

❖**Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.

❖**Information sharing:** A router sends the information to every other router only when the change occurs in the information.

Continued….

- Two phases

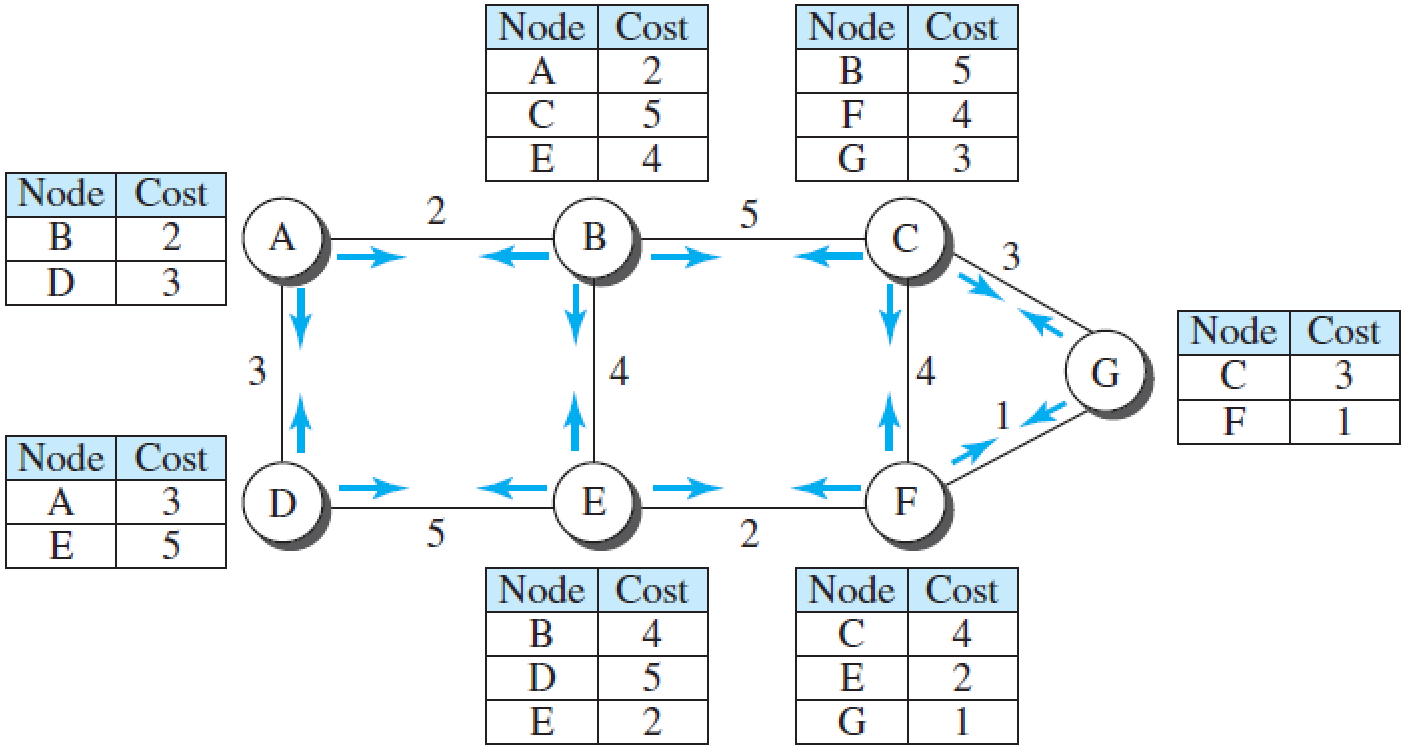
✓ Reliable flooding » Tell all routers what you know about your local topology.

✓ Path calculation (Dijkstra’s algorithm) » Each router computes best path over complete network

- Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes. The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.

After Flooding:

At Router A,



Node	Cost
A	2
C	5
E	4

Node	Cost
B	5
F	4
G	3

Node	Cost
B	2
D	3

Node	Cost
C	3
F	1

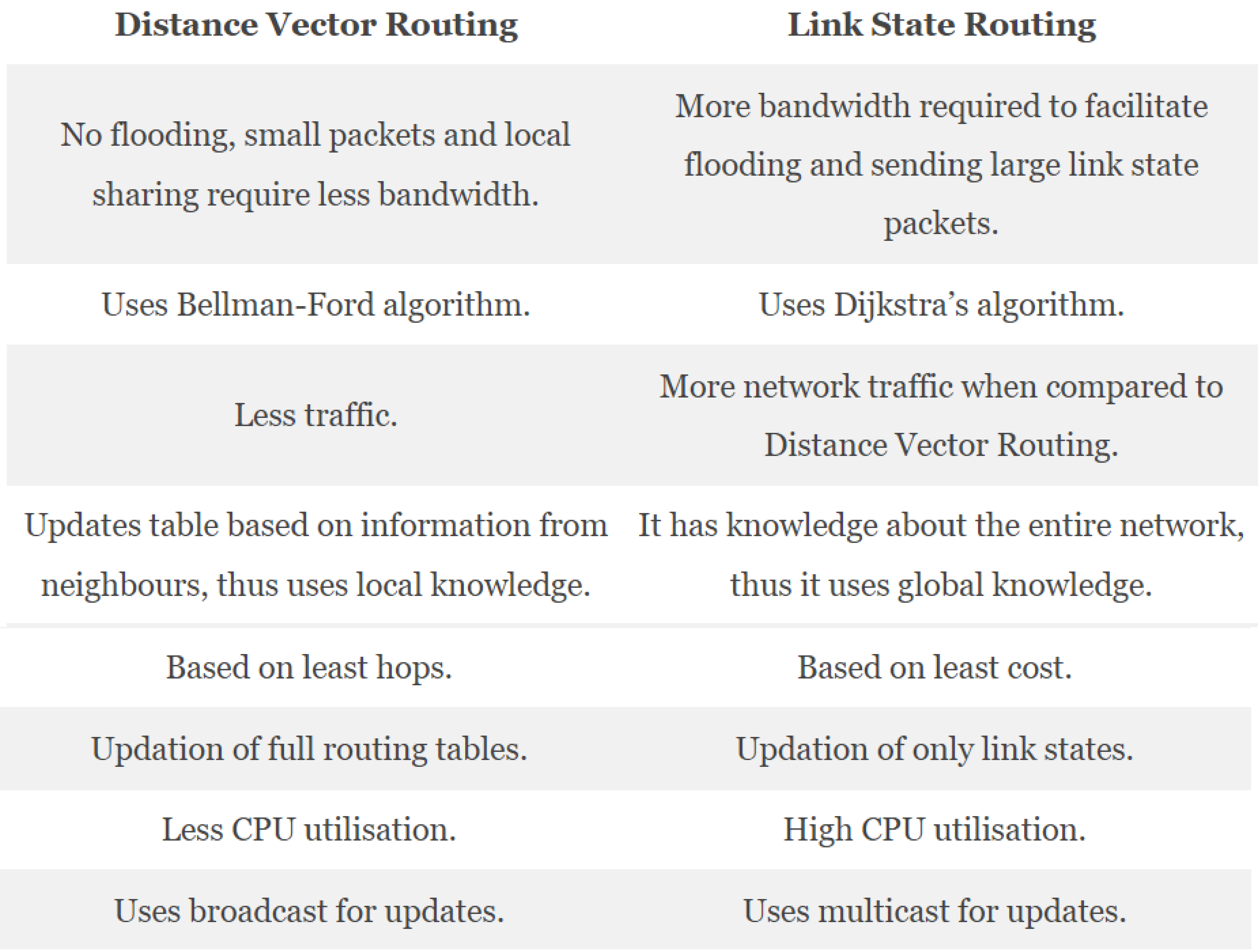
Node	Cost
A	3
E	5

Node	Cost
B	4
D	5
E	2

Node	Cost
C	4
E	2
G	1

A	0	Via A
B	2	Via A
C	7	Via B
D	3	Via A
E	6	Via B
F	8	Via BE
G	9	Via BEF

In the same way every router builds the database.



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.
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.

Open Shortest Path First (OSPF)

- OSPF is a Link State protocol that's considered may be the most famous protocol among the Interior Gateway Protocol (IGP) family, developed in the mid 1980's by the OSPF working group of the IETF.
- When configured, OSPF will listen to neighbors and gather all link state data available to build a topology map of all available paths in its network and then save the information in its topology database, also known as its **Link-State Database** (**LSDB**).
- From the information gathered, it will calculate the best shortest path to each reachable subnet/network using an algorithm called **Shortest Path First** (**SFP**) that was developed by the computer scientist *Edsger W. Dijkstra* in 1956.
- OSPF will then construct **three tables** to store the following information:

✓ **Neighbor Table:** Contains all discovered OSPF neighbors with whom routing information will be interchanged

✓ **Topology Table:** Contains the entire road map of the network with all available OSPF routers and calculated best and alternative paths.

✓ **Routing Table:** Contain the current working best paths that will be used to forward data traffic between neighbors.

Border Gateway Protocol (BGP)

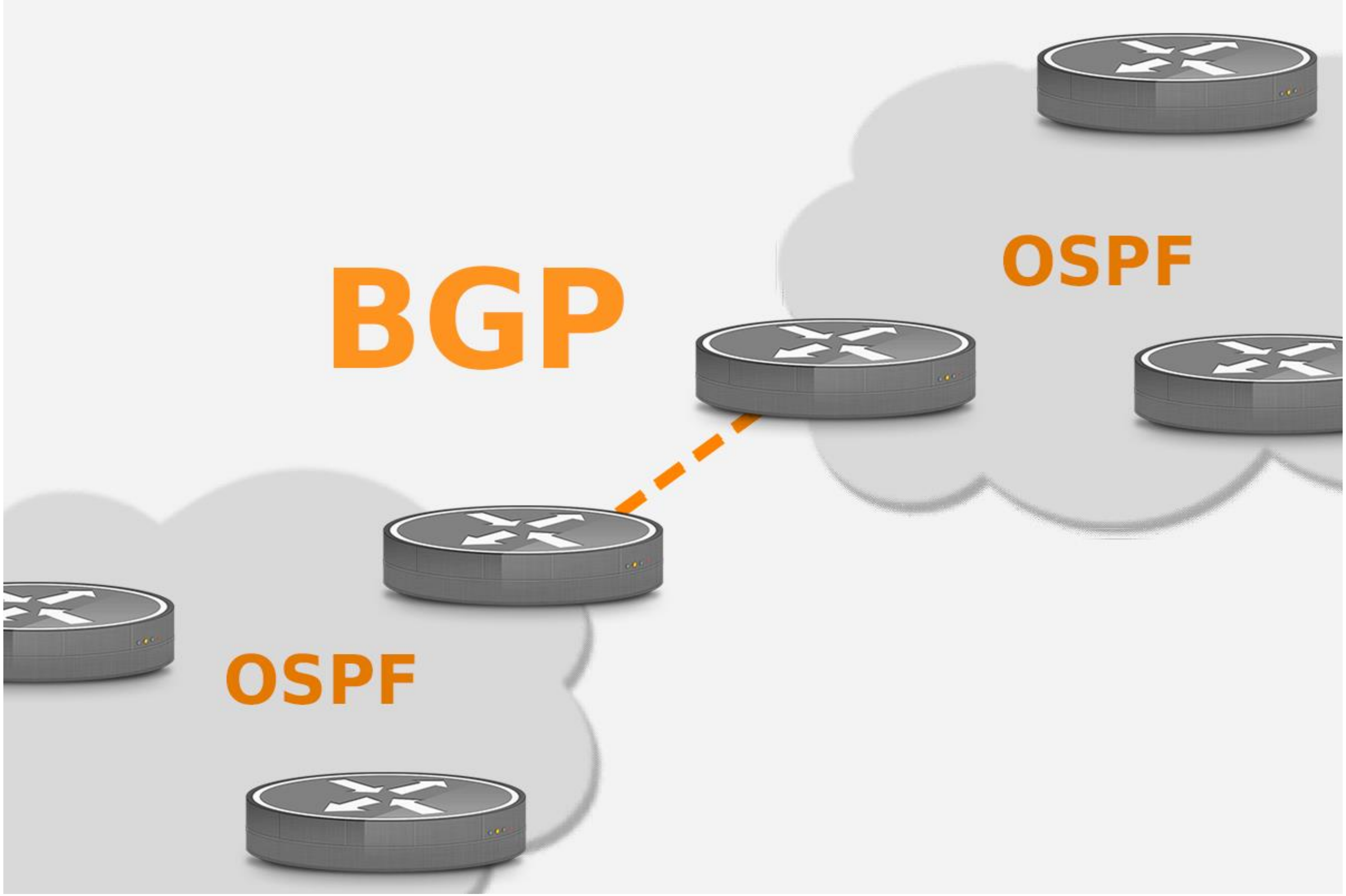
- Border Gateway Protocol (BGP) refers to a gateway protocol that enables the internet to exchange routing information between autonomous systems (AS).
- It is a Path vector protocol that uses a best-path algorithm to select the best path for a given IP prefix. It is based on a greedy algorithm that selects the shortest AS path as the route for a given IP prefix. BGP also has mechanisms to prevent routing loops.
- To automatically determine the best route, BGP references factors like:

  - Path length

  - Origin type

  - Router identification

  - Neighbor IP addresses
- BGP allows administrators to alter transfer routes depending on their needs and offers advanced security features so only authorized routers can exchange data and information with each other.


BGP
OSPF
OSPF

Unicast, Multicast and Broadcast Routing

- If a datagram is destined for only one destination (one-to-one delivery), we have *unicast routing*. If the datagram is destined for several destinations (one-to-many delivery), we have *multicast routing*. If the datagram is destined for all destinations (one to all), we have broadcast routing.

Unicast Routing:

- This type of information transfer is useful when there is a participation of single sender and single recipient. So, in short, you can term it as a one-to-one transmission. For example, a device having IP address 10.1.2.0 in a network wants to send the traffic stream(data packets) to the device with IP address 20.12.4.2 in the other network, then unicast comes into the picture. This is the most common form of data transfer over the networks.

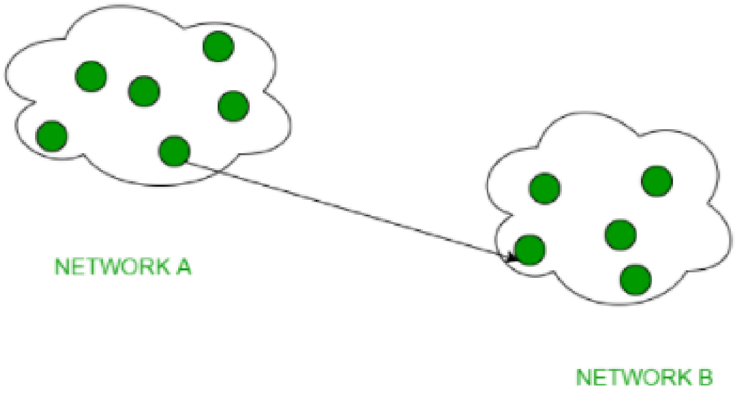


Figure: Unicast

Multicast

- Multicast is the term used to describe communication where a piece of information is sent from one or more points to a set of other points. In this case there is may be one or more senders, and the information is distributed to a set of receivers (theer may be no receivers, or any other number of receivers).
- One example of an application which may use multicast is a video server sending out networked TV channels. Simultaneous delivery of high quality video to each of a large number of delivery platforms will exhaust the capability of even a high bandwidth network with a powerful video clip server. This poses a major salability issue for applications which required sustained high bandwidth. One way to significantly ease scaling to larger groups of clients is to employ multicast networking.

Broadcast

- Broadcast is the term used to describe communication where a piece of information is sent from one point to all other points. In this case there is just one sender, but the information is sent to all connected receivers.
- Broadcast transmission is supported on most LANs (e.g. Ethernet), and may be used to send the same message to all computers on the LAN (e.g. the address resolution protocol (arp) uses this to send an address resolution query to all computers on a LAN, and this is used to communicate with an IPv4 DHC server).
- There are two types of broadcasting transfer:

  - Limited Broadcasting

  - Direct Broadcasting
- In Limited Broadcast the data reaches from a source to all the host in a same network. In Directed Broadcast a host in one network sends the message to all host in another network.