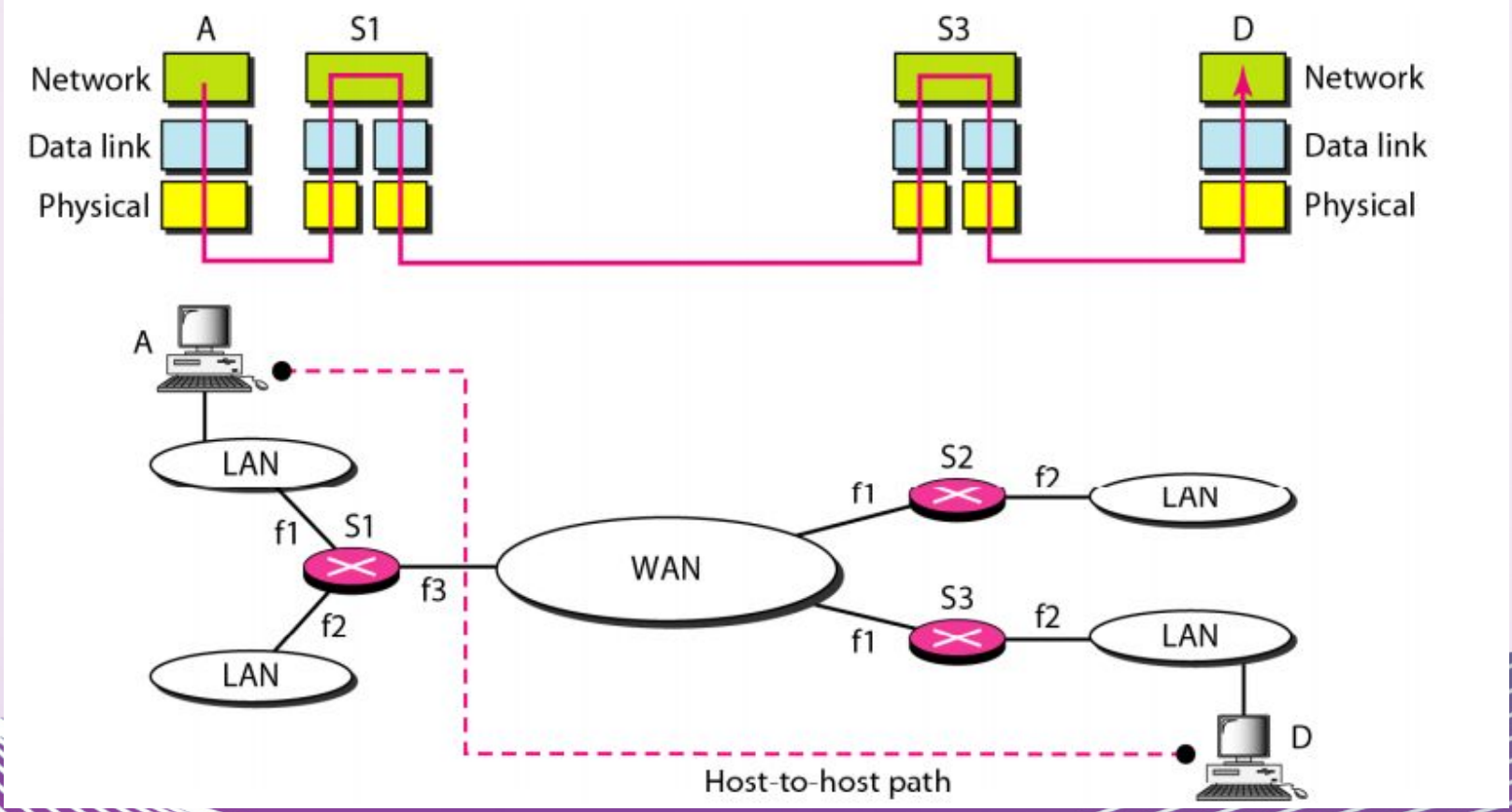
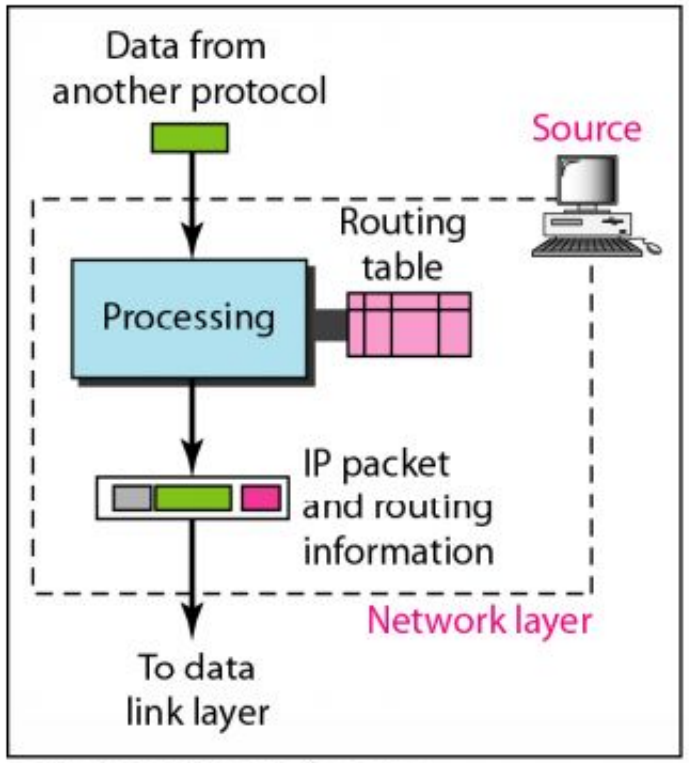
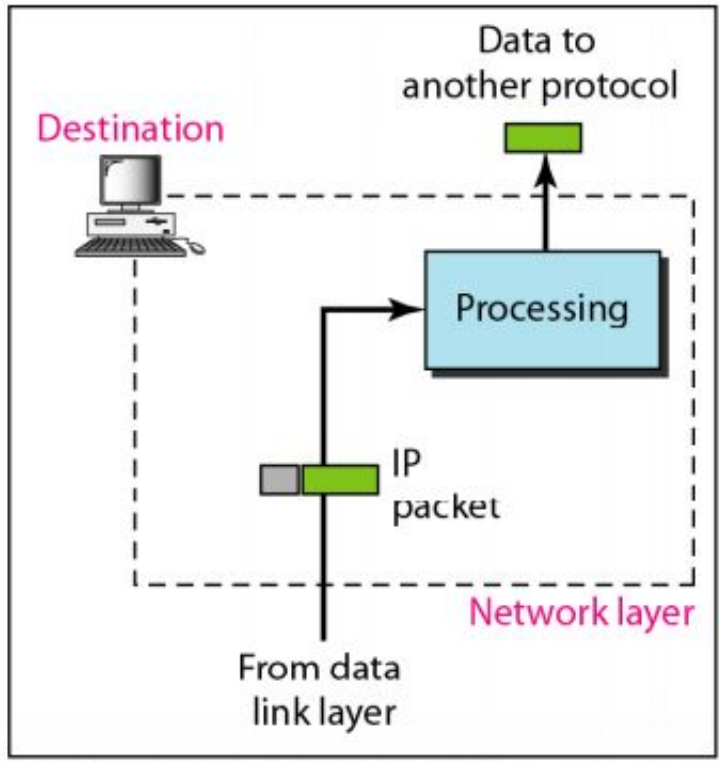
- Network layer in an internetwork



- Network layer at the source, router, and destination



a. Network layer at source



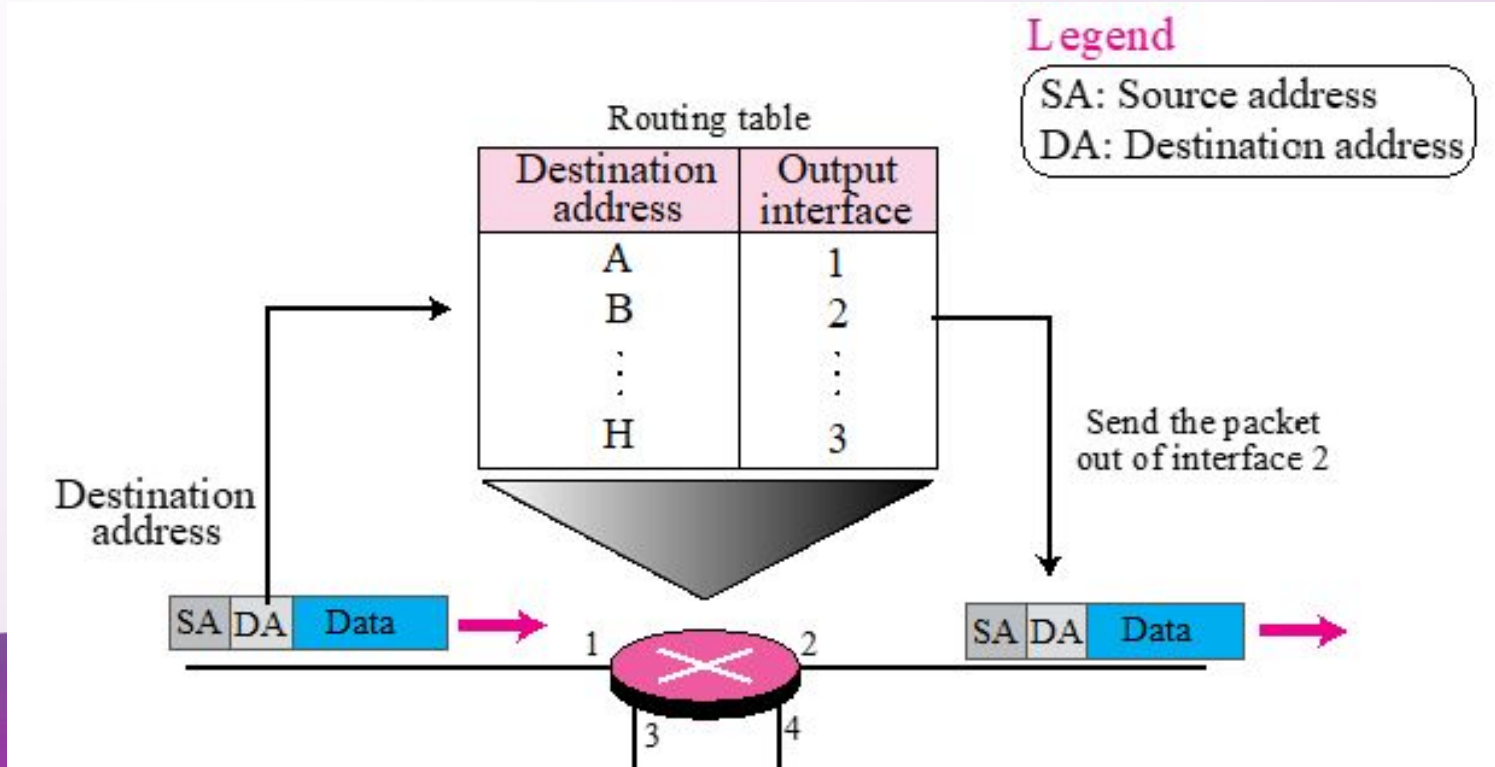
b. Network layer at destination

- Network layer is involved in source host, destination host and all routers
- At source the network layer ***accepts a packet from transport layer***
- Encapsulates the ***packet in datagram and delivers it to data link layer***

Network layer Services

1. Packetizing
   - Encapsulating the payload at source / Encapsulating the payload at destination
   - Network layer does the job of carrier, without using or changing content
   - Source and destination addresses are added along with addition protocol info
   - Only if the data size too large for delivery, it is fragmented
   - It re-assembles it if required at destination end
   - Router only see the header info, and if at all copies header to fragmented packets

2. Routing
   - Find the best possible path from source to destination
   - Routing protocols
3. Forwarding
   - The action the router applies when a packet appears at one of it's interface
   - transferring packet from an input link interface to the appropriate output link interface.
   - ***Routing table*** : decision making table



4. Error Control
   - Is skipped, as fragmented data cannot checked for error
   - A checksum is added in header for checking corrupt data in Header (not the while datagram)
   - ICMP, indirect error control
5. Flow Control
   - no overflow of data at receiver

4. Congestion Control
   - Congestion Control
   - Avoidance and counter step
   - 2 broad categories : open loop congestion control (prevention)
     closed loop congestion control (removal)

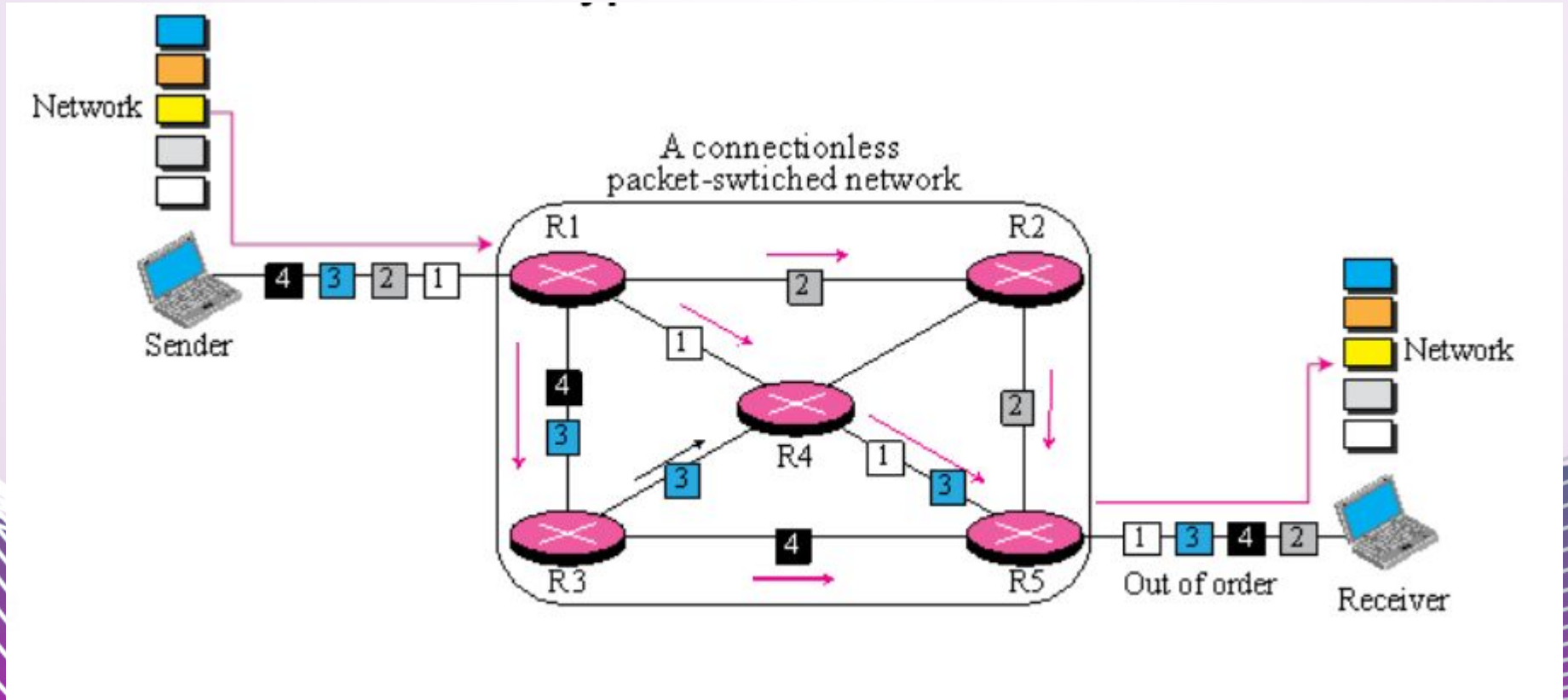
4. Security
   - No there,But implemented as IPSec (virtual level for connectionless)

Packet switching

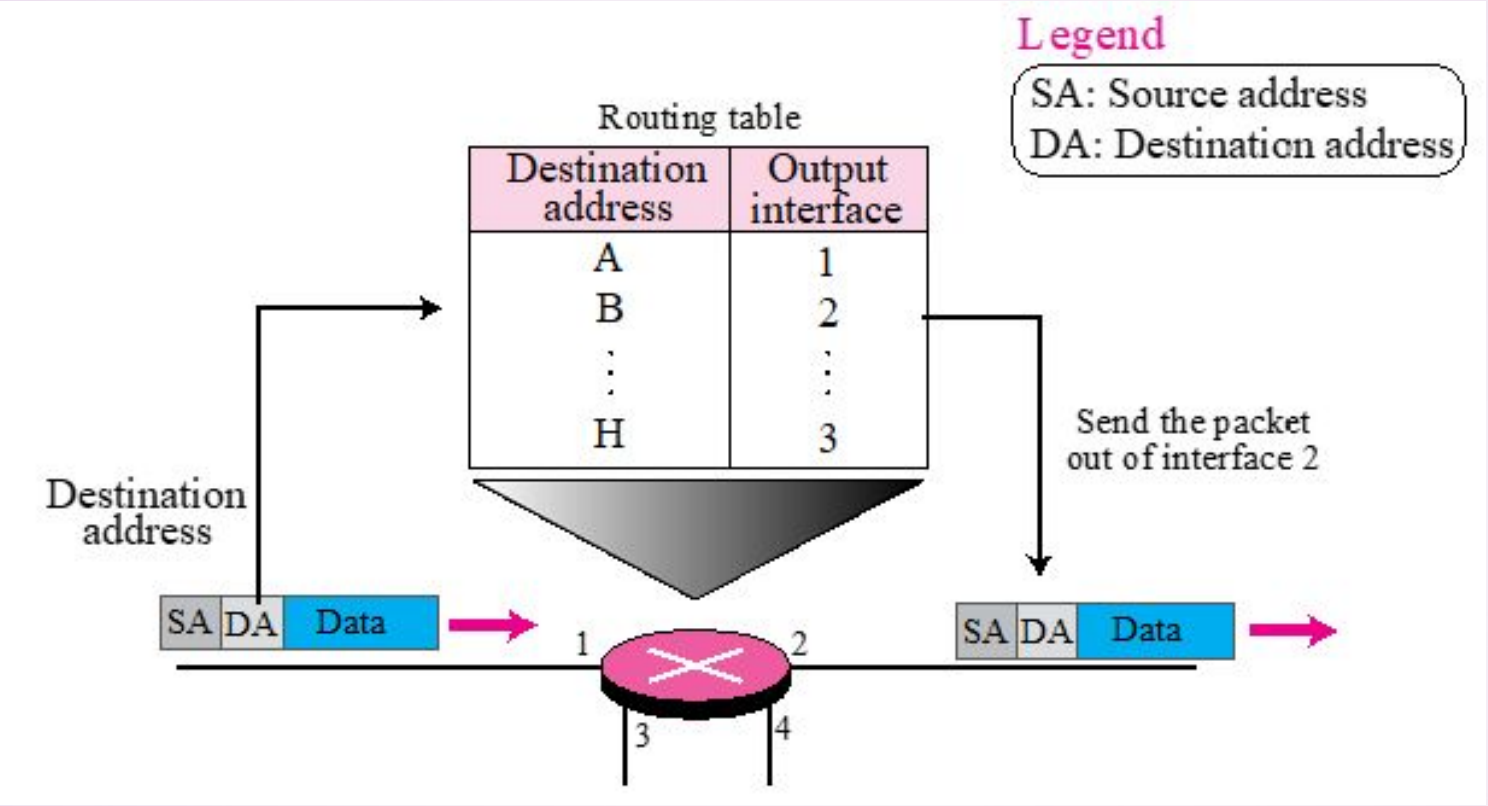
- Data communication switching happens as circuit switching and packet switching
- Network layer does packet switching
- Messages from upper layer is divided into manageable ***packets, called datagram***
- *Packets are reassembled at the destination*
- *2 approaches for routing*
  - *Datagram Approach (Connectionless)*
  - *Virtual approach (Connection oriented)*

Connectionless

- Network layer is only responsible for delivery of packets from the source to the destination
- All datagrams are ***independent***
- Receive ***out of order***
- *The switched here are called* ***routers.***



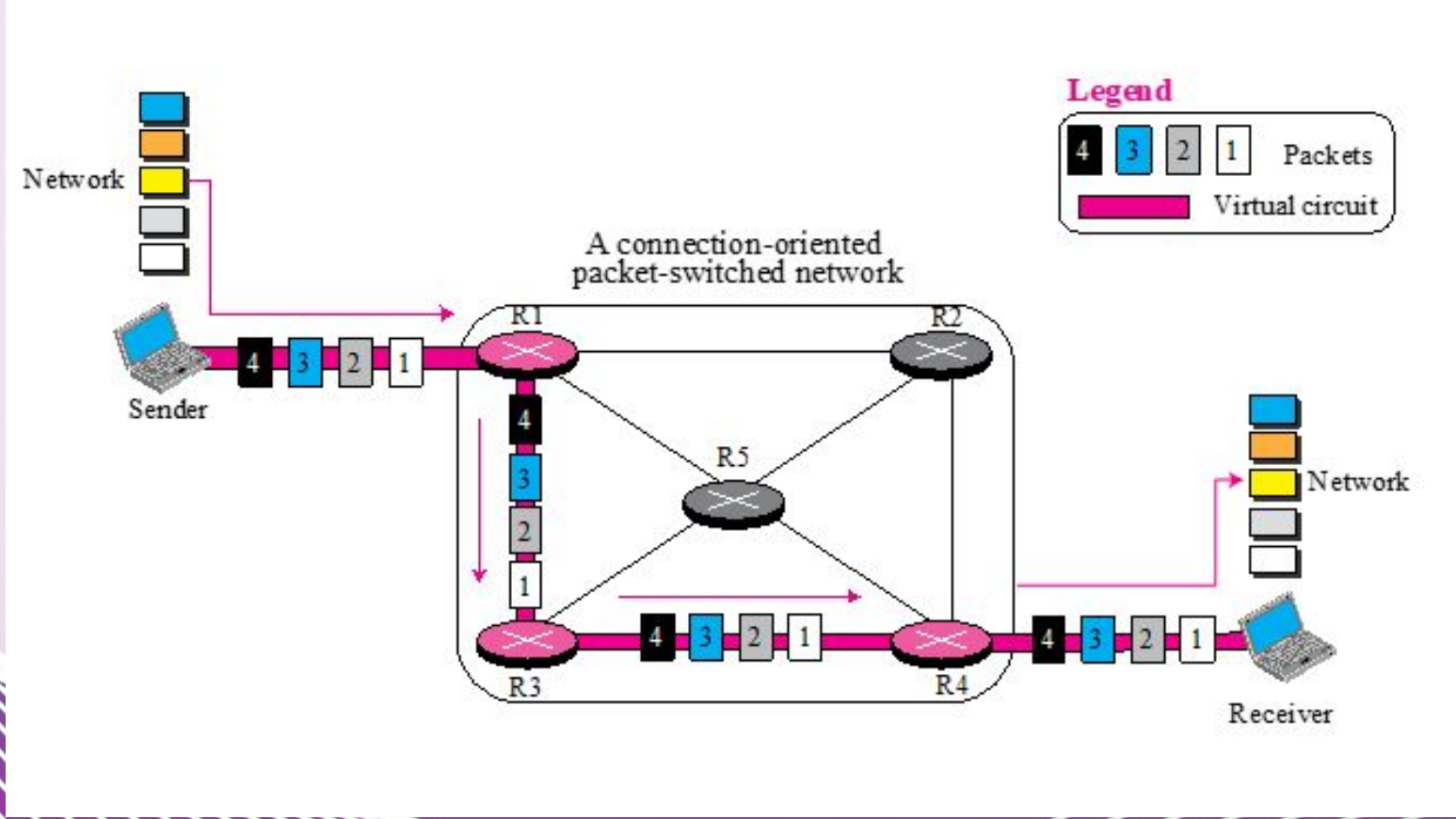
- Forwarding process in a router when used in connection-less network



- Packets are routed based on info in the header
- Source address is need for error message

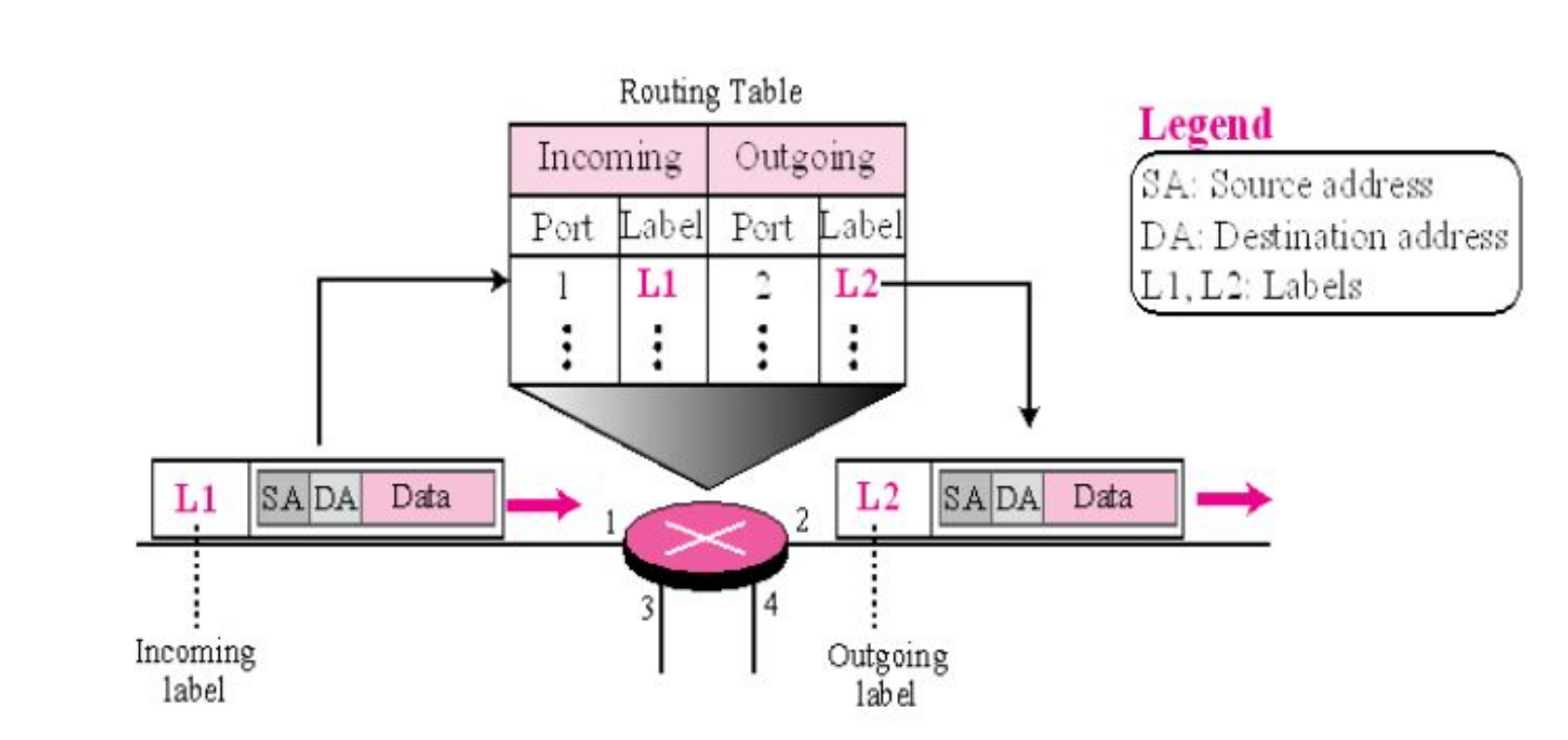
Virtual Circuit approach: Connection oriented

- *In a connection-oriented packet switched network, the forwarding decision is based on the label of the packet.*



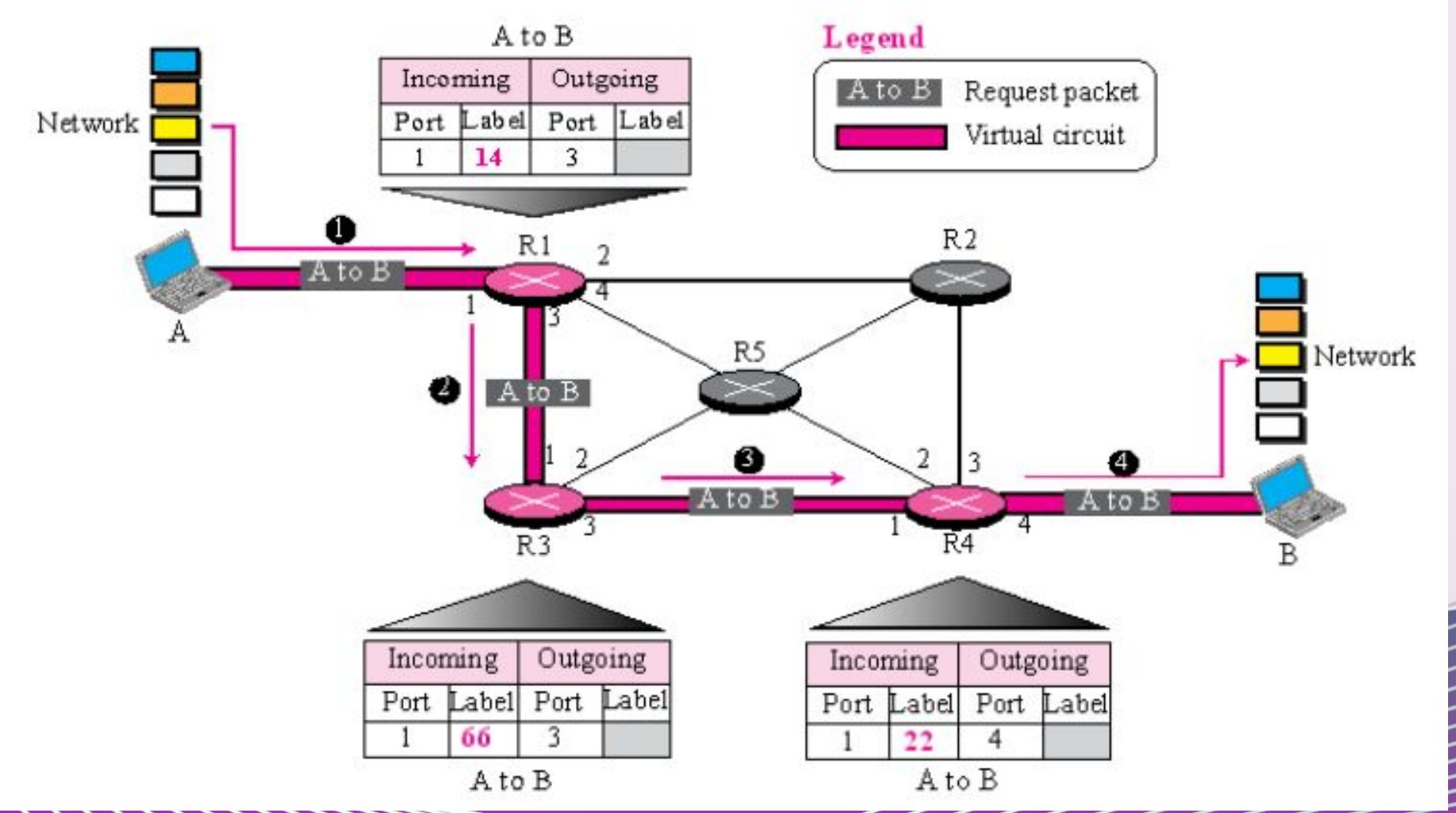
- The packets share a relationship
- A virtual connection needs to be set up before transmission starts
- All the datagrams flow the same path
- Along with source and destination address, it has a ***flow label*** which is a virtual circuit identifier
- Each packet is forwarded based on label.
- 3 phase process
  - Setup
    1. A *request packet* carries the source and destination address
    2. An *acknowledgement packet* complete the entries in the switching tables
  - Data transfer
  - Teardown

- Forwarding (uses labeling)

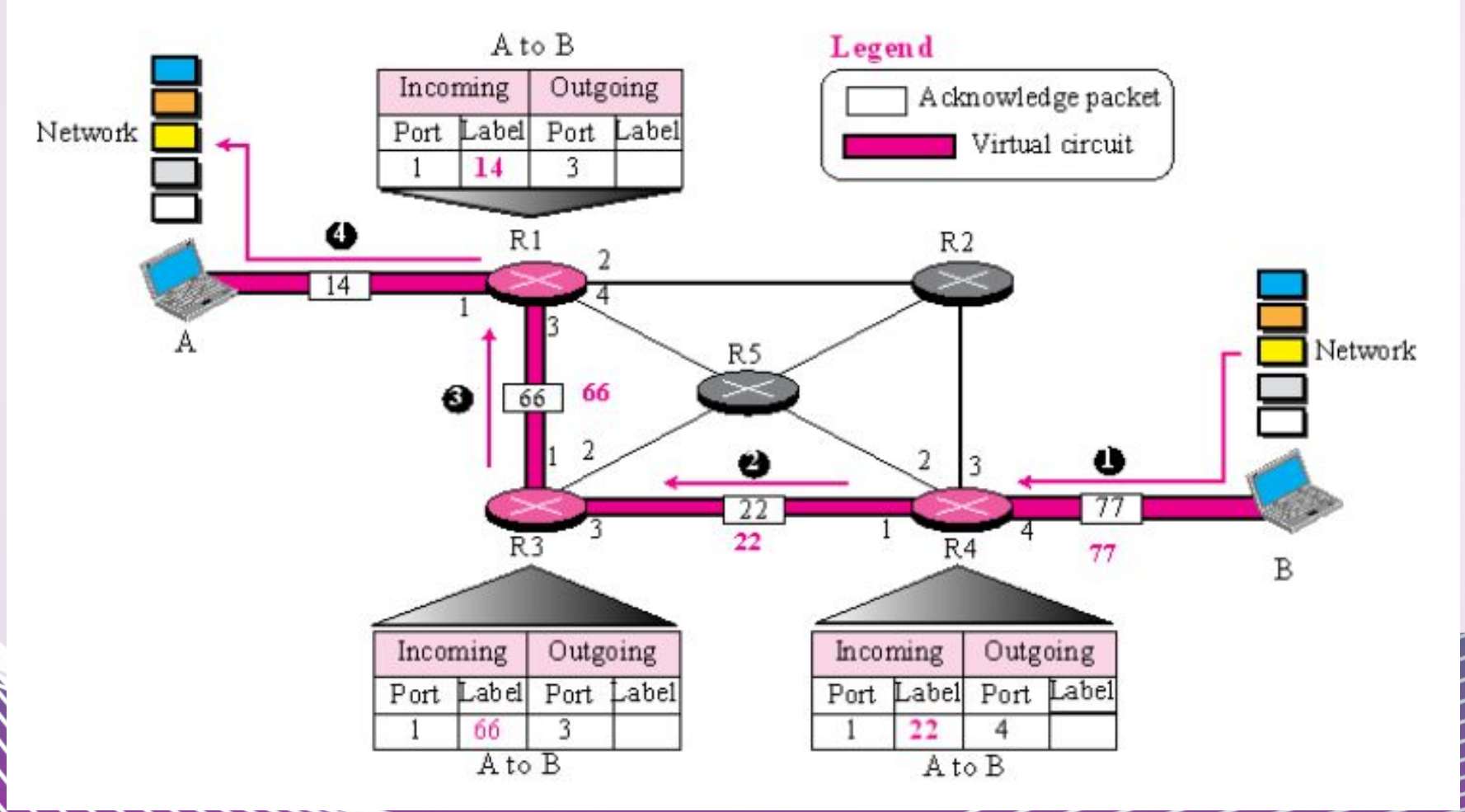


Setup phase

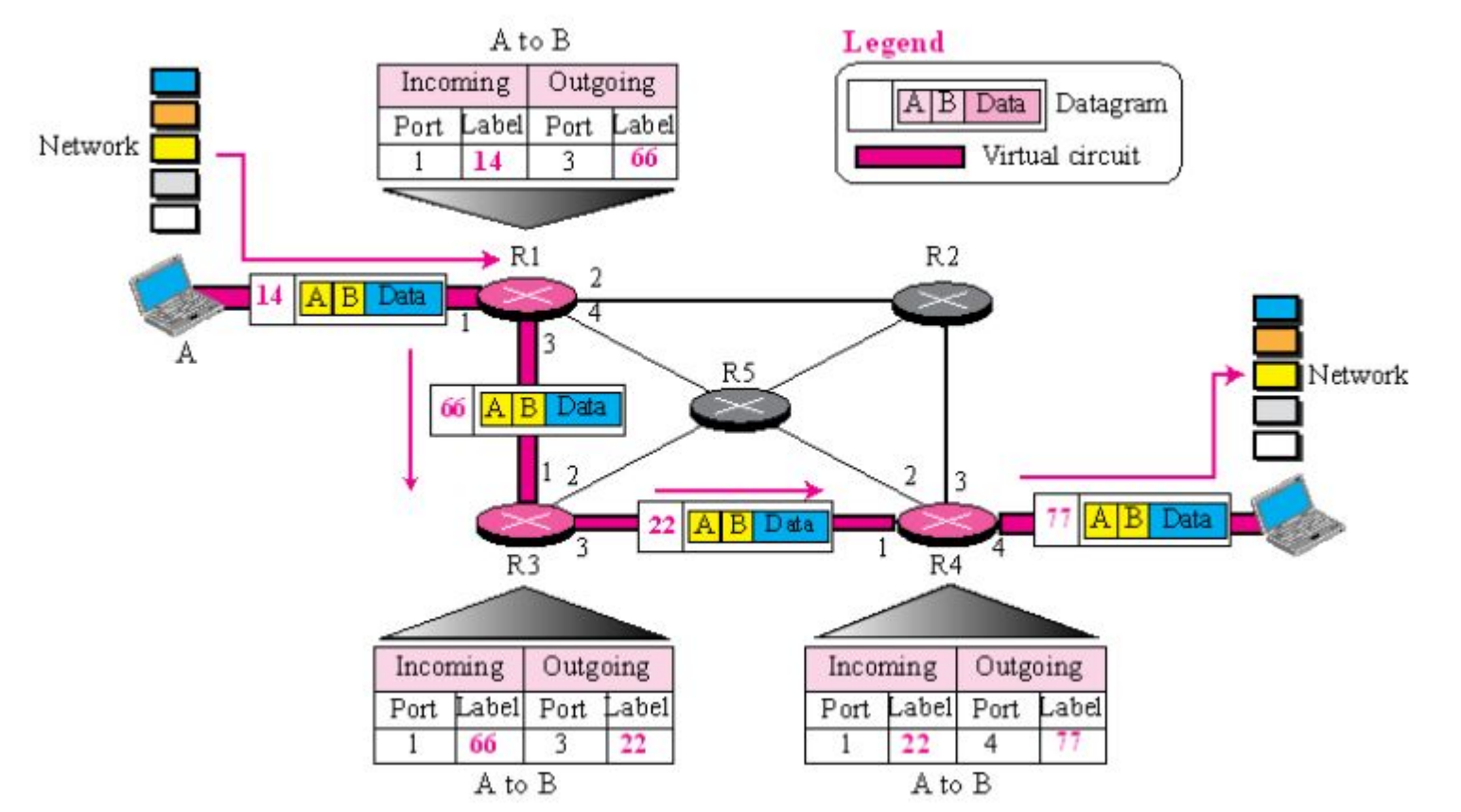
1. Sending Request Packet in a Virtual-Circuit Network



2. Setup Acknowledgment in a Virtual-Circuit Network



Data transfer



Tear down

- After sending all packets to destination, source sends a special packet called a ***teardown packet***
- A destination responds with a confirmation packet. All routers delete the corresponding entry from their tables

Network Performance Measures

Two Performance Measures

– Quantity of Service , QoS:Throughput)

- How much data travels across the net?
- How long does it take to transfer long files?

– Quality of Service, QoS:Average packet delay)

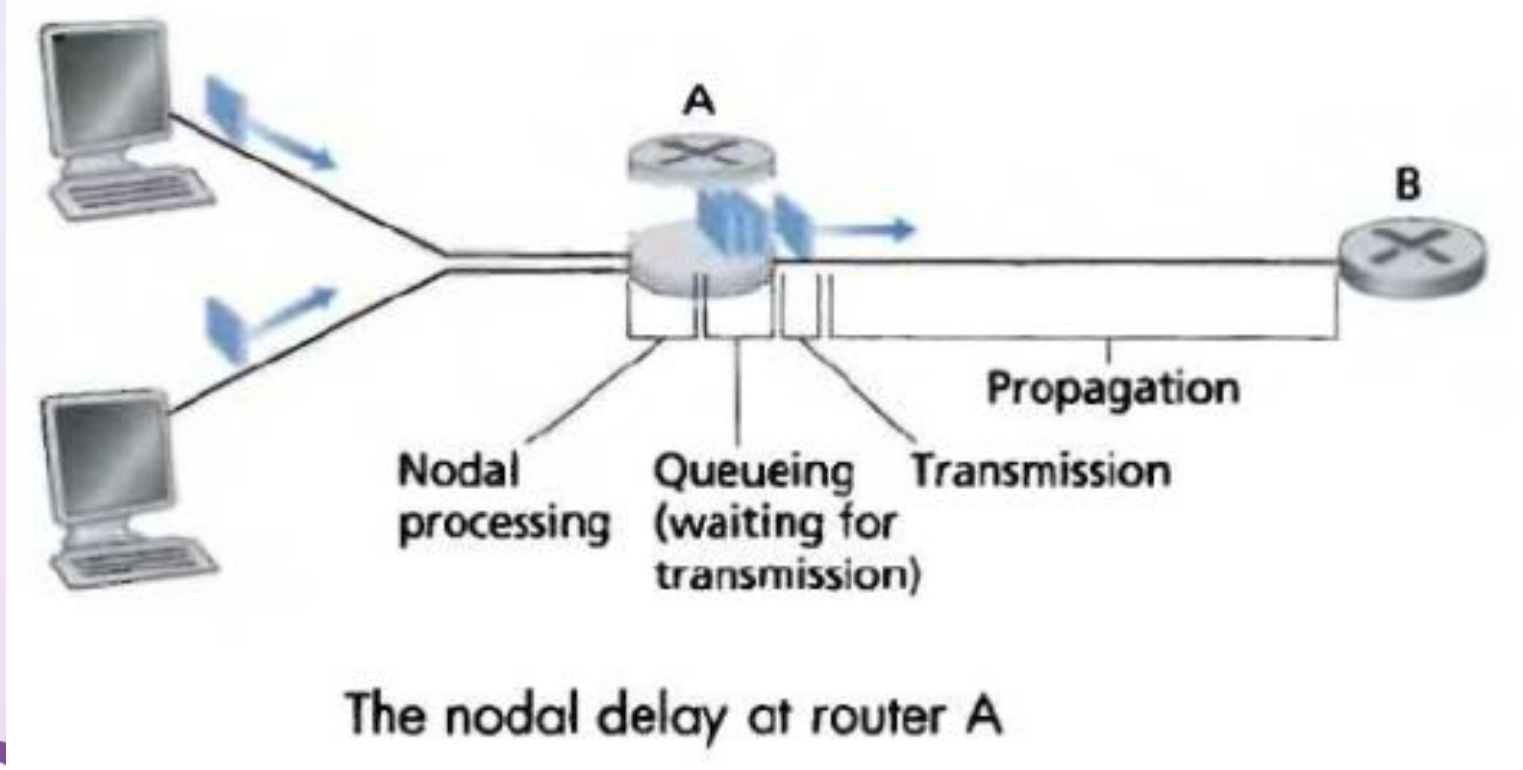
- How long does it take for a packet to arrive at its destination?
- How responsive is the system to user commands?
- Can the network support real-time delivery such as audio and video?

Network Layer Performance

The performance of Network layer is measured in terms of

1. Delay
2. Throughput
3. Packet loss

1. Delay



The nodal delay at router A

1. **Processing Delay -** The time required to examine the ***packet's header*** and decide where to direct the packet is part of the processing delay.
2. **Queuing Delay -** At the queue, the packet experiences a queuing delay as it ***waits to be transmitted onto the link***. The length of the ***queuing delay*** of a particular packet will depend on the number of earlier-arriving packets that are queued and waiting for transmission across the link.
3. **Transmission Delay -** time it takes to push the ***packet's bits onto the link***

   Transmission Delay =(packetlength/transmission rate).

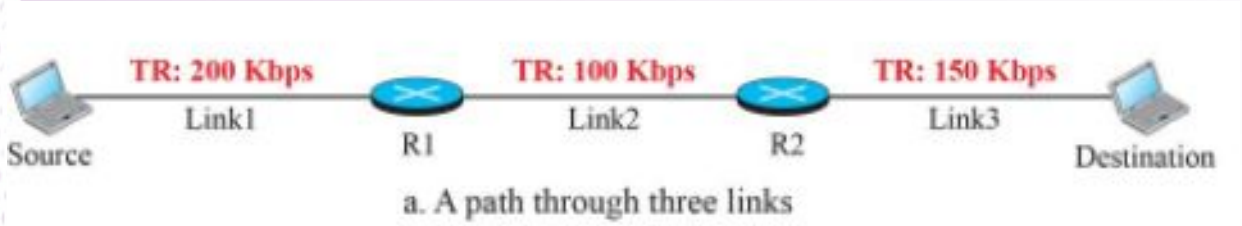
4. **Propagation Delay -** time for a signal ***to reach its destination***. Once a bit is pushed into the link, it needs to propagate to router B (in previous fig)

   The bit propagates at the propagation speed of the link. The propagation speed depends on the physical medium of the link (that is, fiber optics, twisted-pair copper wire)

   Propagation Delay =(Distance/propogation speed).

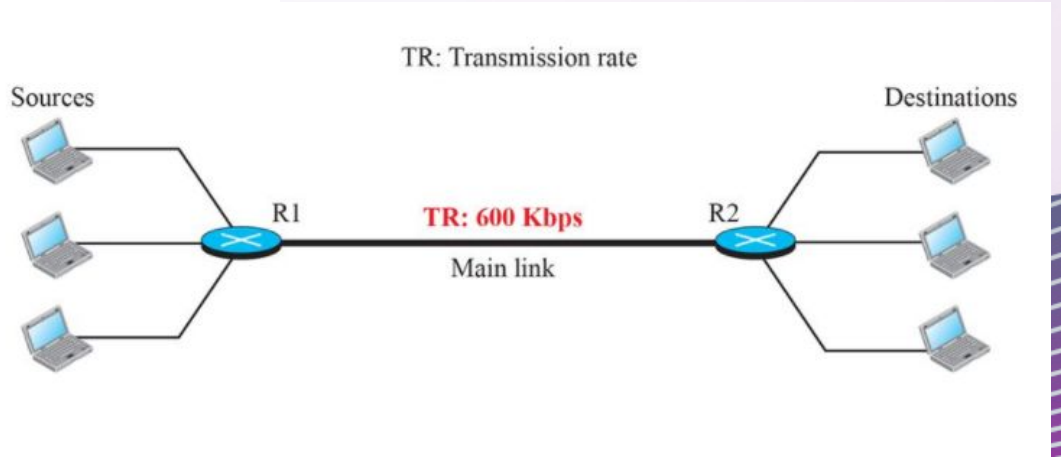
2. Throughput

- The amount of traffic a **network** can carry
- Minimum transmission rate out of all they links



a. A path through three links

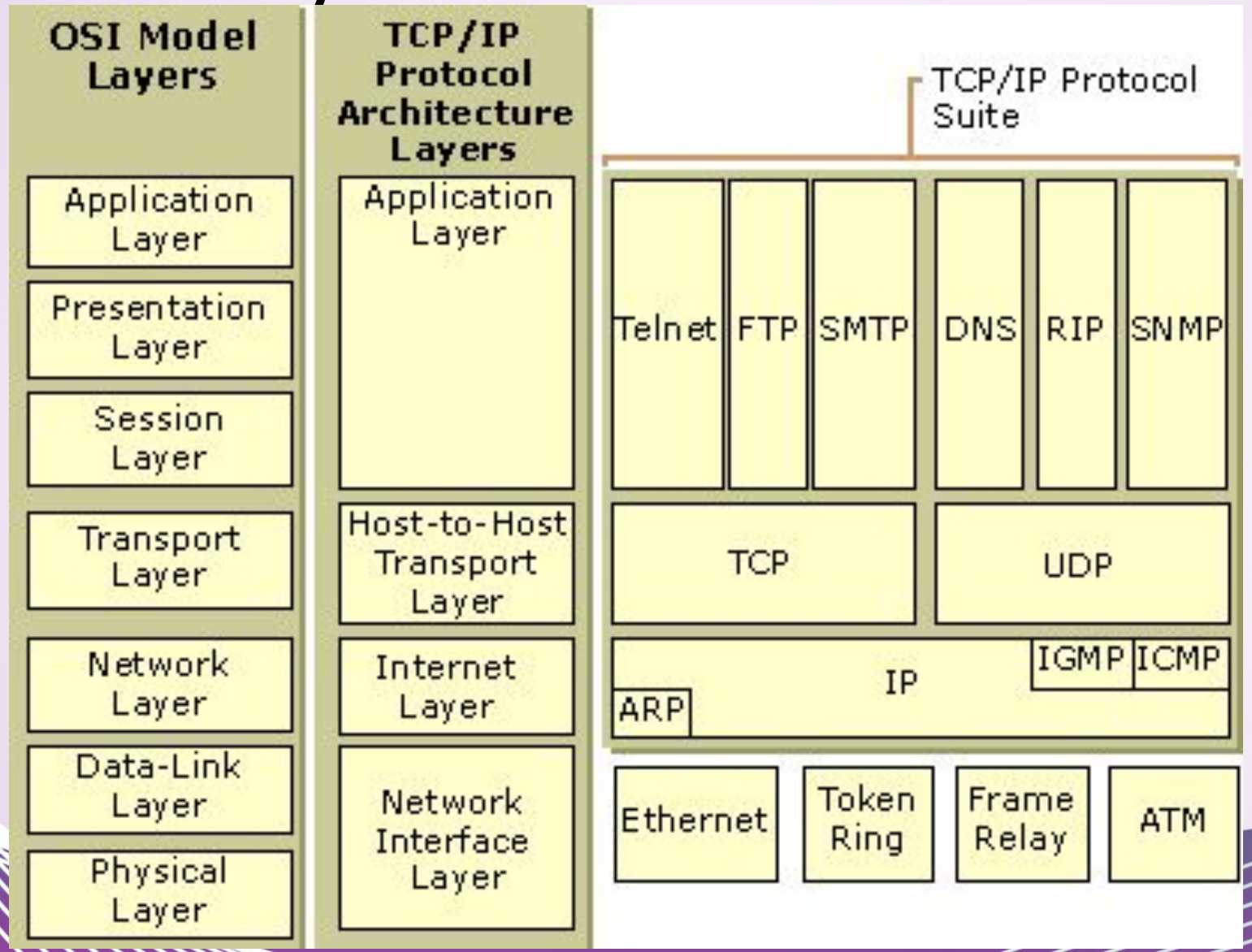
b. Simulation using pipes



3. Packet Loss

- reflects the number of packets lost
- Reasons
  - Link Congestion
  - Device (Router/Switch/Firewall/etc.) Performance
  - Faulty Hardware or Cabling

Network Layer Protocol



- **IPv4** is responsible for packetizing, forwarding, and delivery of a packet.
- **ICMPv4** helps IPv4 to handle some errors that may occur in delivery.
- **IGMP** is used to help IPv4 in multicasting.
- **ARP** is used in network layer address to link layer mapping

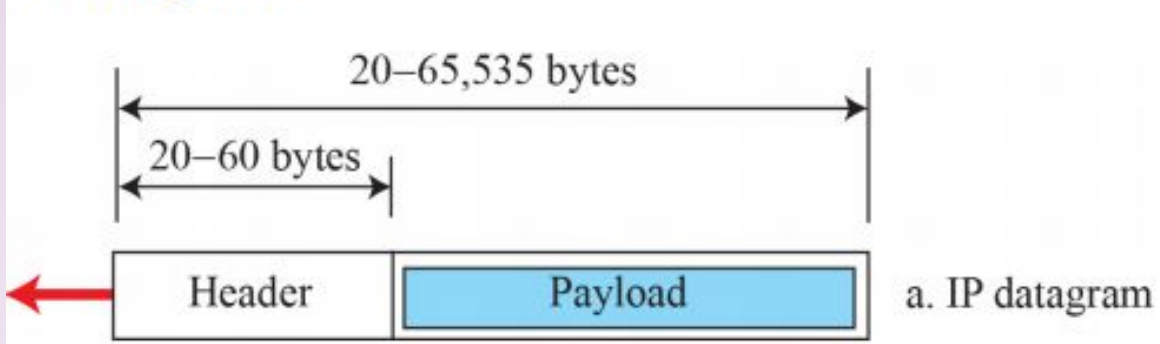
IPv4

- The Internet protocol offers a ***best-effort service of delivering*** datagrams between hosts.
- IPv4 is a ***connectionless*** internet protocol for packet switched network
- IPv4 datagrams may be lost, arbitrarily delayed, corrupted, or duplicated.
- Needs to be paired with TCP for reliability.

IPv4 Datagram Format

- Packets used by IP are called datagrams.
- A datagram is a variable-length packet consisting of two parts:
  - Header
  - payload (data)
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

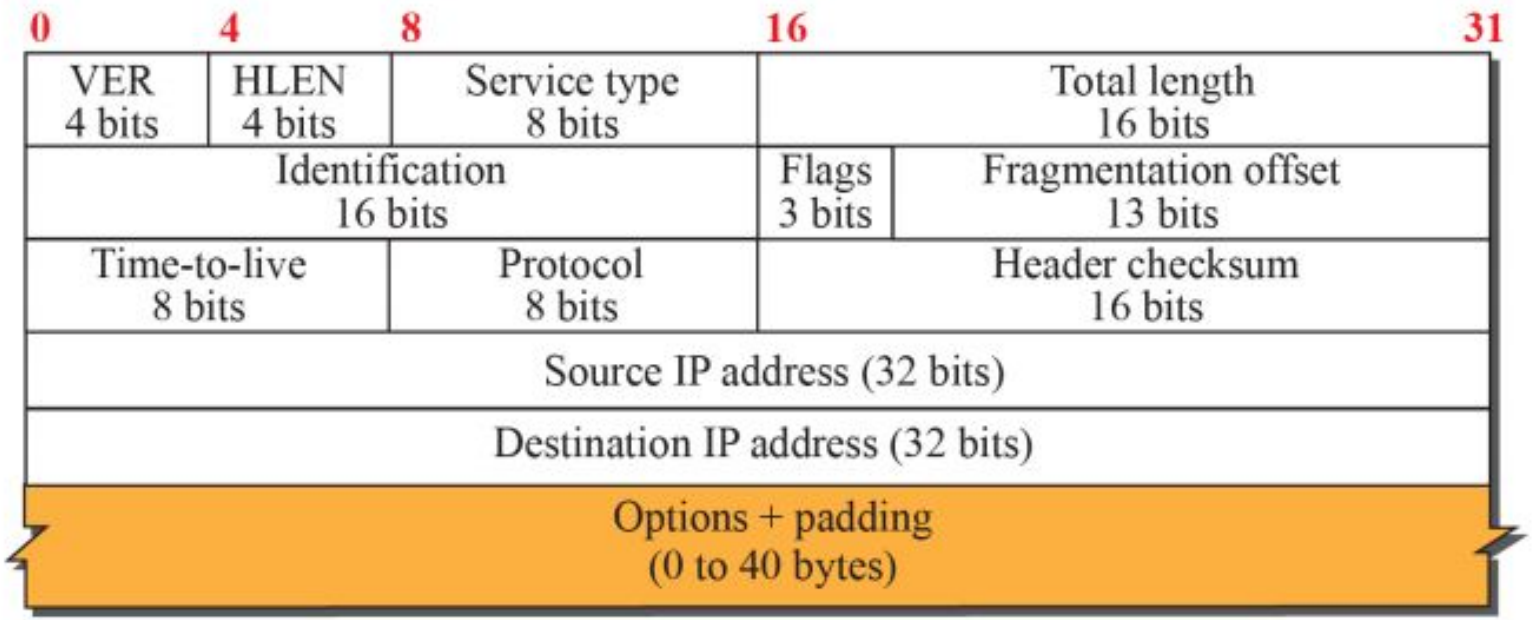
IP datagram



a. IP datagram

Legend

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



b. Header format

- **Version**

Version of the IP protocol which determines how to interpret the header. Currently the only permitted values are 4 (0100) or 6 (0110). The header format shown here is valid for IPv4 only.

- **HLEN**

Length of header as a number of 32-bit words

- **Type of service**

This field is often ignored by current routers but is meant to allow traffic to be prioritised (among other things).

- **Total Length**

The length of the entire datagram including header and data: maximum permitted it 65,535 bytes or 64K

- **Identification, Flags and Fragment Offset**

These values allow datagrams to be fragmented for transmission and reassembled at the destination

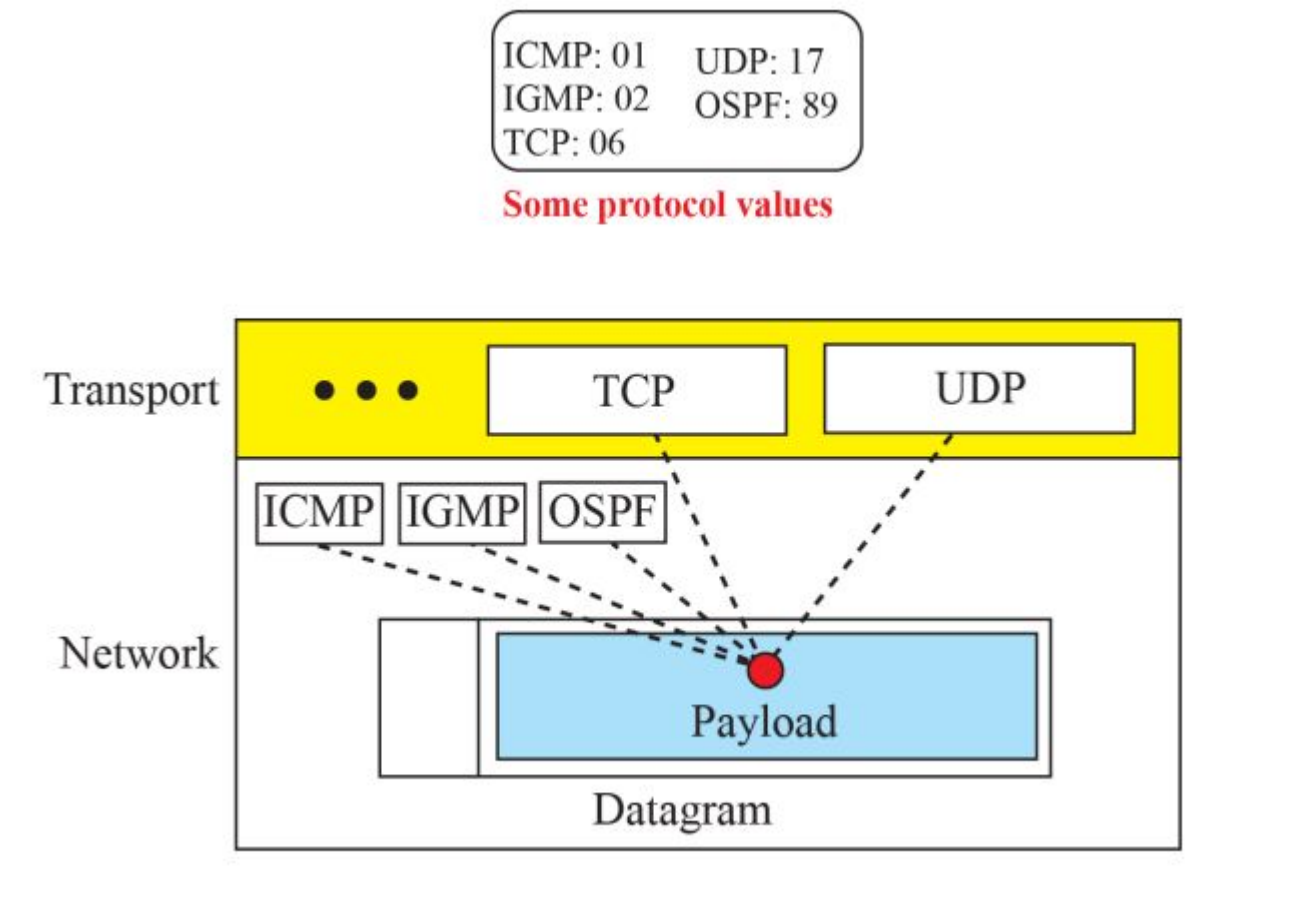
- **Time to live**

An integer which is decremented at each router "hop"; supposed to be interpreted as a number of seconds but more often treated as a "hop count". If the value reaches zero the datagram is discarded and an ICMP message is sent to the source host.

- **Protocol**

Identifies the transport-layer protocol which will interpret the Data section. This will typically be TCP or UDP but other values are possible. Protocols are identified by a unique number as listed in an online database at www.iana.org.

- Multiplexing and demultiplexing using the value of the protocol field
- Protocol plays the same role as the ports play in transport layer



- **Header checksum**

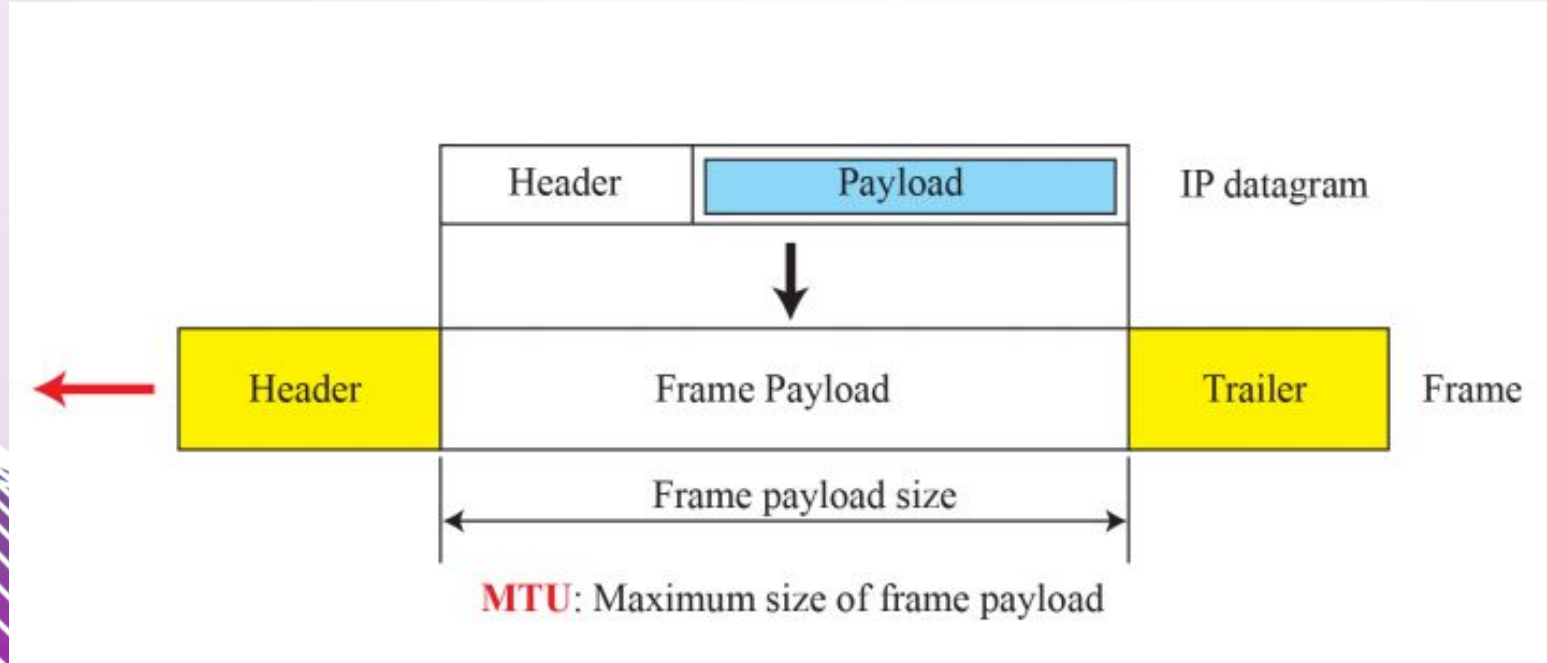
This is used to verify the header, and is recomputed at each router hop. This field is left out of IPv6 which relies on the transport layer for verification.

- **Addresses and Options**

These are 32-bit IP addresses which identify the network and host address. Note that IP does not have to specify addresses of any intermediate nodes; this can be left to the router. Routing requirements can also be specified in the Options field, along with options to do with security and debugging.

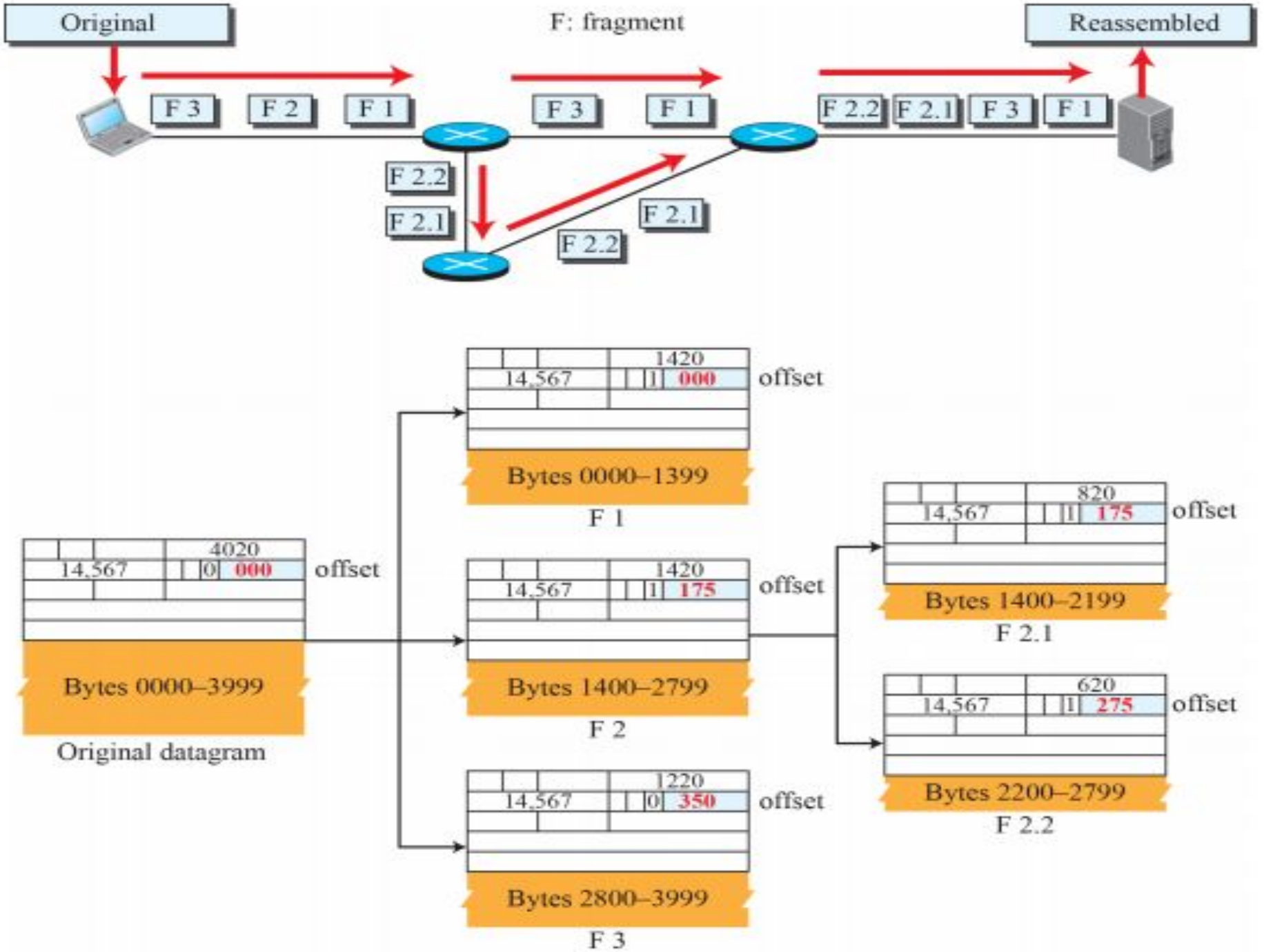
Fragmentation

- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.
- **Maximum transfer unit (MTU)**
  - Max size of the payload that can be encapsulated
  - Size of datagram < max size, else fragmented



- A datagram can be fragmented ***by the source host or any router*** in the path.
- When a datagram is fragmented, each fragment has its ***own header*** with most of the fields repeated, but some have been changed.
- The ***reassembly of the datagram***, however, is done only by the destination host, because each fragment becomes an independent datagram.

- Example


Original
F: fragment
Reassembled
F 3
F 2
F 1
F 3
F 1
F 2.2
F 2.1
F 3
F 1
F 2.2
F 2.1
F 2.1
F 2.2
4020
14,567
0
000
offset
Bytes 0000–3999
Original datagram
1420
14,567
1
000
offset
Bytes 0000–1399
F 1
1420
14,567
1
175
offset
Bytes 1400–2799
F 2
1220
14,567
0
350
offset
Bytes 2800–3999
F 3
820
14,567
1
175
offset
Bytes 1400–2199
F 2.1
620
14,567
1
275
offset
Bytes 2200–2799
F 2.2

Logical Addressing

- we need a global addressing scheme; we called this ***logical addressing***
- term ***IP address*** to mean a logical address in the network layer
- Version popularly in use IPv4 (IP version 4)
- IPv4 Internet addresses are ***32 bits*** in length
- this gives us a maximum of $\mathbf{2^{32}}$ addresses.
- The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called the new generation of IP or IPv6 (IP version 6).
- IPv6 uses ***128***-bit addresses that give much greater flexibility in address allocation.

IPv4 Addresses

- IPv4 addresses are universal.
- An IPv4 address is ***32 bits*** long.
- IPv4 addresses are ***unique / dynamic*** also possible
- ***IP is the address of the connection*** and not the host or router
- Every host would have an Unique address for every connection
- **Address Space**
  - ***total number of addresses*** used by the protocol
  - If a protocol uses ***N*** bits to define an address, the address space is 2^N
  - IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (minus few)

Notations

- There are two prevalent notations to show an IPv4 address:
  - ✔ binary notation (base 2)
  - ✔ Dotted decimal notation. (base 256)
  - ✔ Hexadecimal notation (base 16)

❖ Binary Notation

- is displayed as 32 bits.
- Each ***octet*** is often referred to as a ***byte***. So it is common to hear an IPv4 address referred to as a ***32-bit address or a 4-byte*** address.
- The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

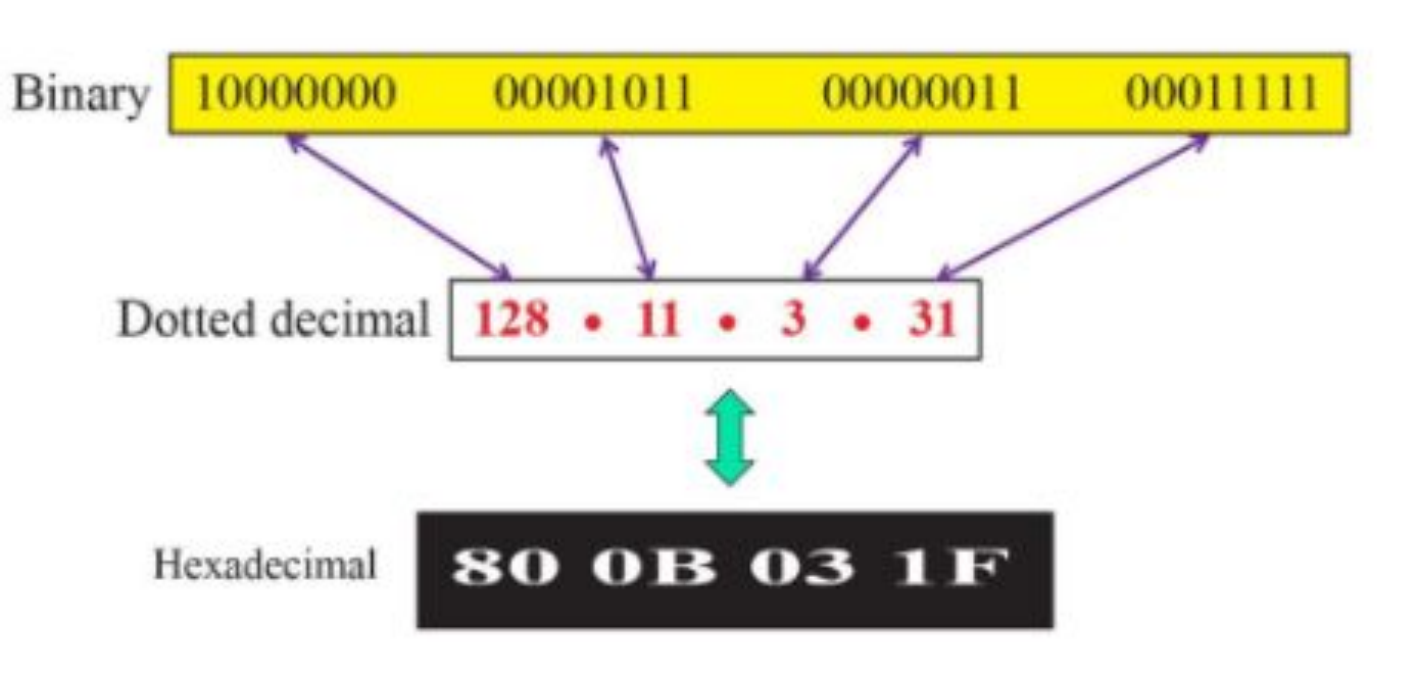
❖ Dotted-Decimal Notation

- To make the IPv4 address more compact and easier to read
- Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes.
- The following is the dotted-decimal notation of the above address:

117.149.29.2

❖ Hexadecimal Notation

- 4 bit



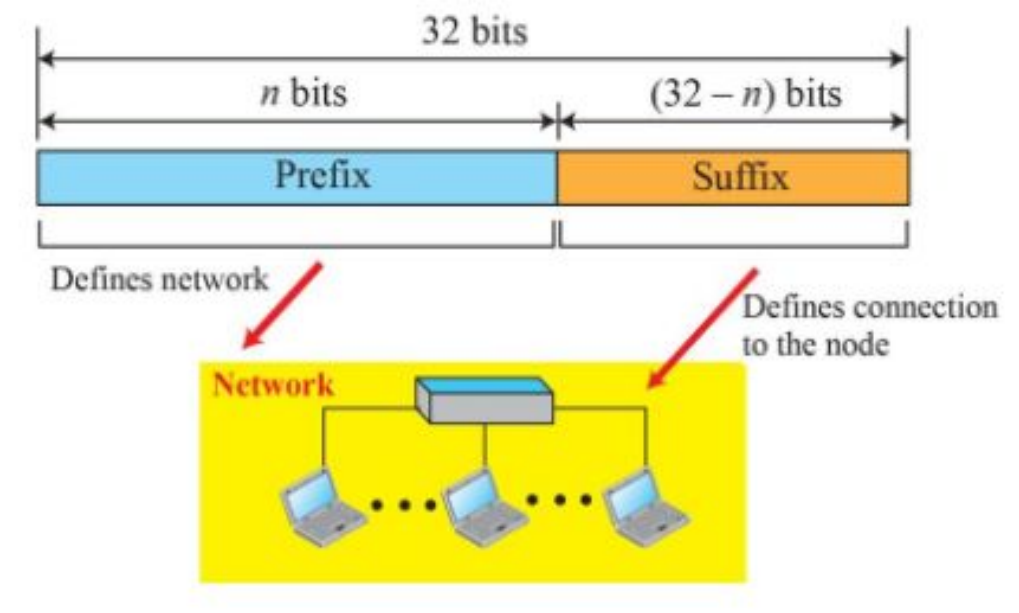
2001:0db8:85a3:0:0:8A2E:0370:7334

Identify valid IP address?

- 172.16.254.1
- 256.5.254.2
- 00011001101110011

Hierarchy in Addressing

- 32 bit IPv4 is hierarchical
- Divided into 2 parts :
  - prefix - defines the network
    - length is n bits
    - fixed (classfull) or variable length (classless)
  - Suffix – defines the node
    - length is 32-n bits

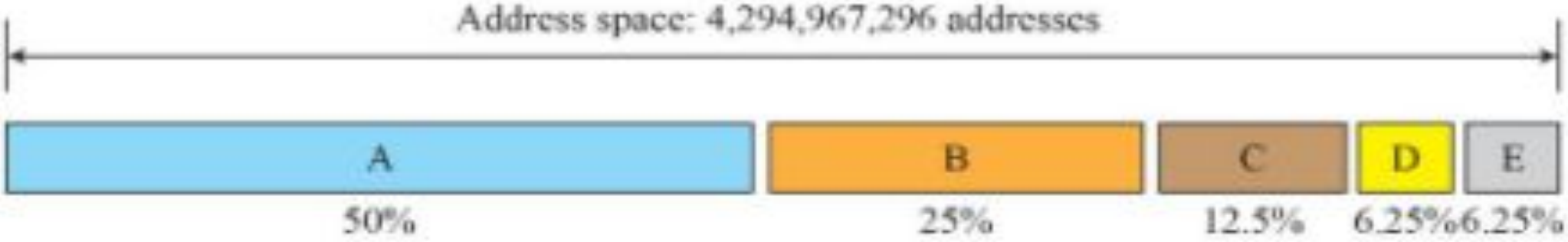


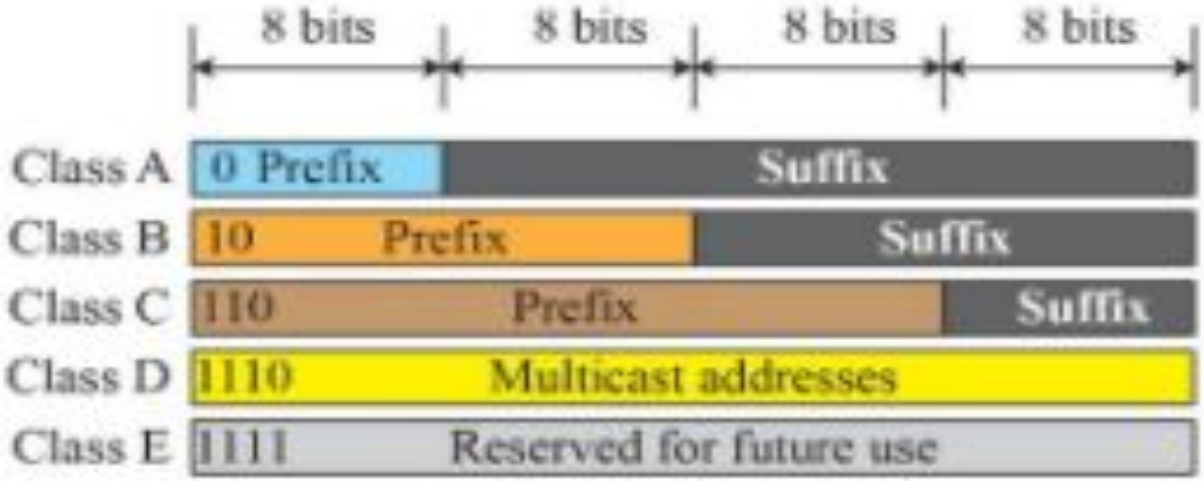
Classful Addressing

- Devices examines the ***first octet of the address*** and determines the address range.
- ***The high order bits never change for each class.***
- For instance in Classful Addressing:
  - 192.168.23.2 is in the Class C range
  - Therefore – 24 network bits and 8 hosts bits.
- The subnet mask allows networks to be subdivided or subnetted.
- Each class is assigned a **default subnet *mask***.

Class	High Order Bits	First Octet Range	Number of Network Bits	Number of Host Bits	Number of Networks	Number of Hosts per Network	Default Subnet Mask
A	0	0-127	8	24	128 (2^7)	16,777,216 (2^{24})	255.0.0.0
B	10	128-191	16	16	16,384 (2^{14})	65,536 (2^{16})	255.255.0.0
C	110	192-223	24	8	2,097,152 (2^{21})	256 (2^8)	255.255.255.0

- Fixed length prefix (3 fixed length prefixes are designed n=8,16,24)
- Address space is divided into 5 classes (A,B,C,D,E)
- occupation address space in classful addressing



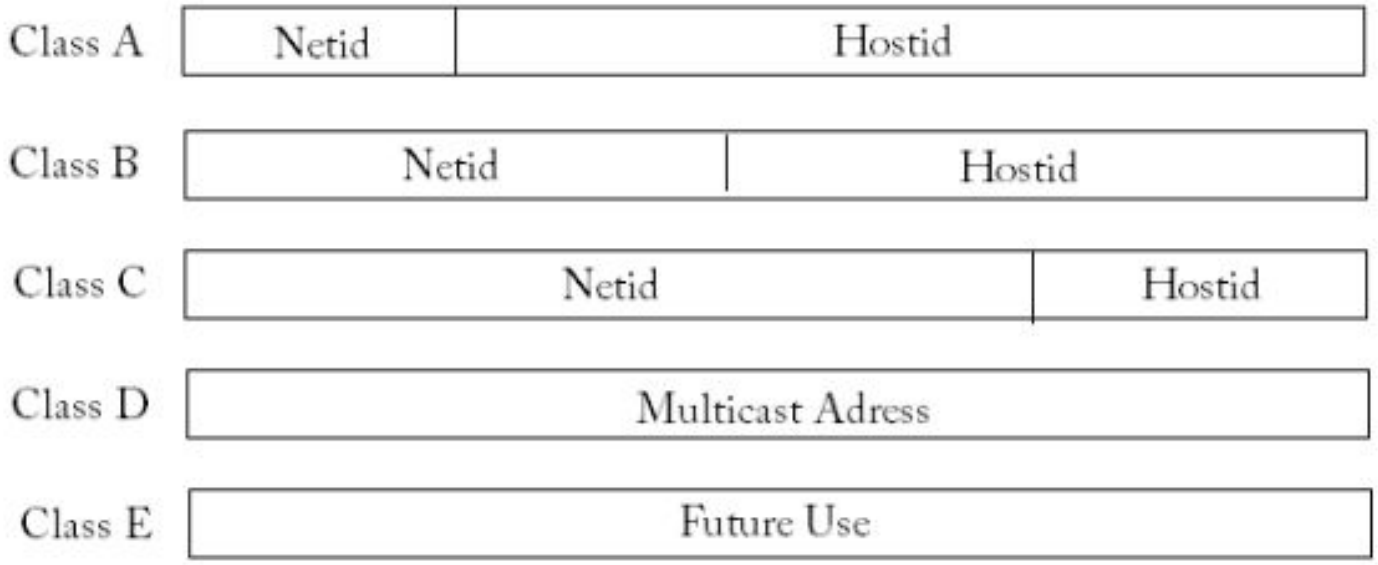


Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

Decimal format

Class A	0-127	0-255	0-255	0-255
Class B	128-191	0-255	0-255	0-255
Class C	192-223	0-255	0-255	0-255
Class D	224-239	0-255	0-255	0-255
Class E	240-255	0-255	0-255	0-255

Netid and hostid



Class A

- contained all addresses in which the most significant bit is zero.
- The network number for this class is given by the next 7 bits, therefore accommodating 128 networks in total, including the zero network, and including the IP networks already allocated.

Class B

- network was a network in which all addresses had the two most-significant bits set to 10
- For these networks, the network address was given by the next 14 bits of the address
- leaving 16 bits for numbering host on the network
- total of 65536 addresses per network.

Class C

- 3 high-order bits/ most-significant bits are set to 110
- designating the next 21 bits to number the networks
- leaving each network with 256 local addresses.

- In order to function properly with network devices, every IP network must contain three types of addresses:
    - **Network Address:**
        - All HOST BITS are set to 0.
    - **Host Address:**
        - HOST BITS will vary.
    - **Broadcast Address:**
        - All HOST BITS are set to 1.
- For a host to communicate directly with another host on the same network, they must have the same network portion.

- Addresses per class

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

- For every IP address range that we assign to a network segment, we automatically lose two addresses….
  - One for the network address (sometimes called the wire address or subnetwork address)
  - One for the broadcast address for that network.

$$(2^{number\_of\_bits} - 2 \text{ or } 2^n - 2)$$

Class	Number of Network Bits	Number of Host Bits	Number Hosts Per Network	Number of Useable Hosts per Network
A	8	24	$2^{24} = 16,777,216$	$2^{24} - 2 = 16,777,214$
B	16	16	$2^{16} = 65,536$	$2^{16} - 2 = 65,534$
C	24	8	$2^{8} = 256$	$2^{8} - 2 = 254$

- **Problems**

  Ip address: 134.35.78.2

  Class: B

  Netid bits:2 Byte Netid : 134.35

  Network address: 134.35.0.0

  Default Subnet mask : 255.255.0.0

- **TRY**

1.The network address: 17.0.0.0.

Find i) the Block (net id) ii) the range of addresses.

2. The network address : 132.21.0.0.

Find i) the Block ii) the range of addresses.

3. Given the network address 220.34.76.0

Find i) the Block ii) the range of addresses.

4. The given IP address is 23.56.7.91. find the network address.

- Solution

1. The class is A because the first byte is between 0 and 127. The block has a netid of 17. The addresses range from 17.0.0.0 to 17.255.255.255.
2. 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.
3. 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.
4. The default mask is 255.0.0.0, which means that only the first byte is preserved and the other 3 bytes are set to 0s. The network address is 23.0.0.0.

Special IPv4 Addresses

NETWORK ADDRESS

- A network address is an address where all host bits in the IP address are set to zero (0).
- In every subnet there is a network address. This is the first and lowest numbered address in the range because the address is always the address where all host bits are set to zero. The network address is defined in the RFC's as as the address that contains all zeroes in the host portion of the address and is used to communicate with devices that maintain the network equipment.

BROADCAST ADDRESS

- A broadcast address is an address where all host bits in the IP address are set to one (1).
- This address is the last address in the range of addresses, and is the address whose host portion is set to all ones. All hosts are to accept and respond to the broadcast address. This makes special services possible.

LOOPBACK ADDRESS (127.0.0.1)

- The 127.0.0.0 class 'A' subnet is used for special local addresses, most commonly the loopback address 127.0.0.1.
- This address is used to test the local network interface device's functionality. All network interface devices should respond to this address from the command line of the local host.
- If you ping 127.0.0.1 from the local host, you can be assured that the network hardware is functioning and that the network software is also functioning. The addresses in the **127.0.0.0 - 127.255.255.255** range cannot be reached from outside the host, and so cannot be used to build a LAN.

PRIVATE IP ADDRESSES

- RFC 1918 defines a number of IP blocks which were set aside by the American Registry of Internet Numbers (ARIN) for use as private addresses on private networks that are not directly connected to the Internet.
- The private addresses are:

Class	Start	End
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

Multicast IP Addresses

- There are a number of addresses that are set aside for special purposes, such as the IP's used in OSPF, Multicast, and experimental purposes that cannot be used on the Internet.

Class	Start	End
D	224.0.0.0	239.255.255.255

Subnetting

- In general a two level hierarchy is followed –
  - **Two-levels of Hierarchy:**
    - Net-id
    - Host-id
- **Disadvantage:**
  - host cannot be organized into groups.
  - All the hosts are at the same level (fixed no of hosts)

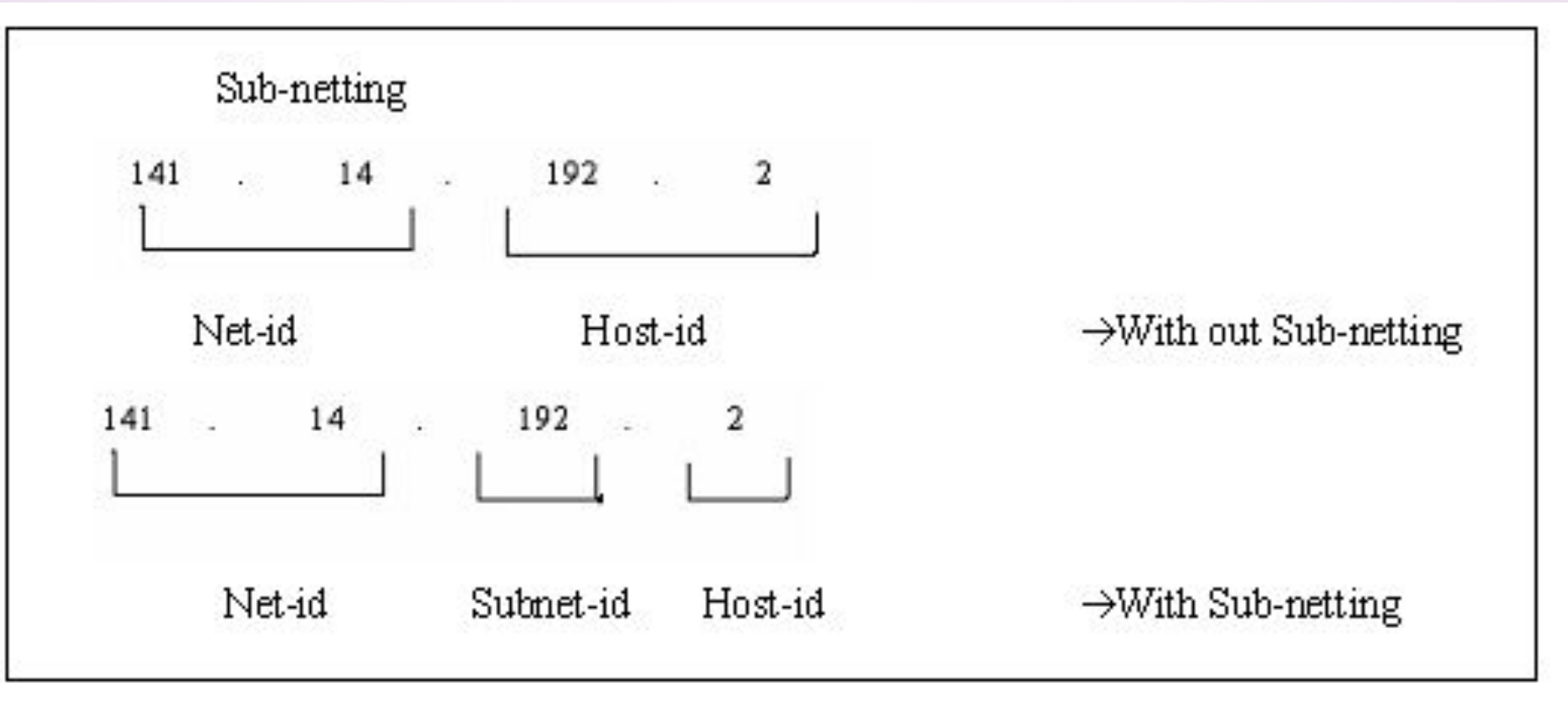
- A subnetwork or ***subnet*** is a logical subdivision of an IP network.
- The practice of dividing a network into two or more networks is called ***subnetting***.
- Computers that belong to a **subnet** are addressed with a common, identical, most-significant bit-group in their IP address called ***subnet address***

Masking

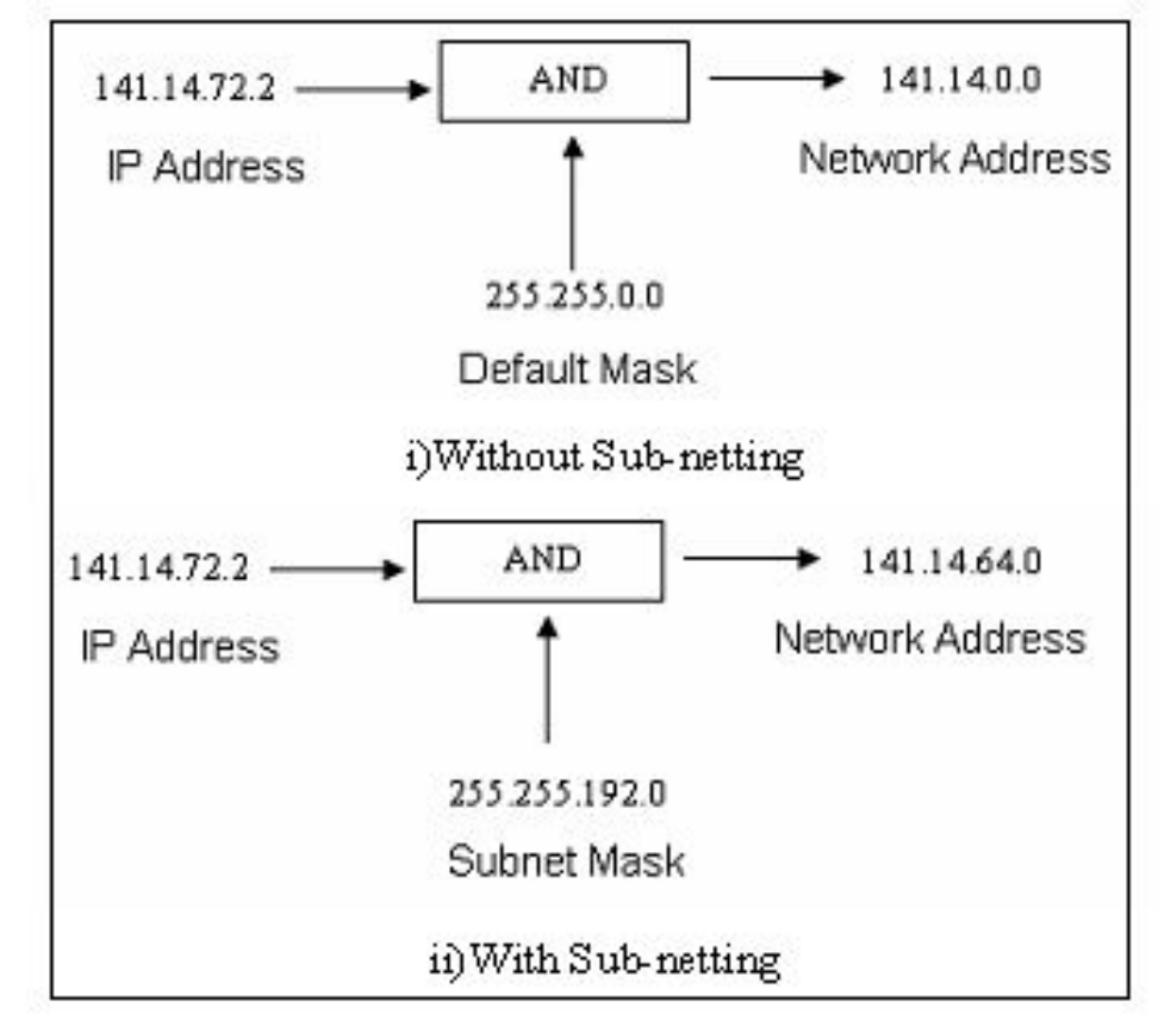
- A **subnet mask** hides, or "**masks**," the network part of a system's IP address
- leaves only the host part as the machine identifier.
- also called the default mask

Class	*Binary*	*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

Sub-netting



Subnet mask



How to find subnet Address

Example 1:

Given:

- IP Address: 200.45.34.56
- Subnet Mask: 255.255.240.0

Find:

- The subnet address

IP Address:	11001000	00101101	00100010	00111000
Subnet Mask:	11111111	11111111	11111111	11110000
	11001000	00101101	00100010	00110000
	200	. 45	. 34	. 48

Comparison of Default mask and subnet mask

- In the subnet mask the number of 1's is more than the no. of 1's in the default mask.

Default Mask: for class B address							
11111111	.	11111111	.	00000000	.	00000000	
255	.	255	.	0	.	0	
Sub net Mask:							
11111111	.	11111111	.	**1111**0000	.	00000000	
255	.	255	.	240	.	0	

Default mask and subnet mask

How to find number of subnets in a network:

- The no. of subnets can be found by counting the extra 1's that are added to the default mask.
- In our above example, the no. of extra 1's is **4,** so the no. of subnet is: 2^4 = **16** subnets.

How to find number of addresses per subnet:

- The no. of addresses per subnet can be found by counting the 0's in the subnet mask.
- In our above example, the no. of 0's is **12,** so the no. of addresses per subnet is: 2^{12} = **4096** addresses per subnets.

Condition to design a subnet

- Deciding the no. of subnets
  - No. of subnet to be power of **2**
- Finding the subnet mask
  - Find the No. of 1's in the default Mask
  - Find the No. of 1's that defines the subnet.
  - Add the No. of 1's from step **a** & **b**
  - Find the No. of 0's by subtracting the No. of 1's in step **c** from 32
- Finding the range of addresses in each subnet
  - The 1st address in the 1st subnet is the 1st address in the block.
  - When we address the No. of address in each subnet to get the last address.

A company is granted a site address 201.70.64.0. The company needs 6 subnets Design the subnet

Solution:

- We need 6 subnet ,which is of power of 2, now take the next power of 2 ie., 2^3
- So now 3 extra 1's will be there in the subnet mask
- No. of 0's is 5, ie., 2^5 host for each subnet
- Range of address
  - The 1st address in the subnet is 201.70.64.0
  - The last address in the 1st subnet is found by adding the No. of host minus 1 in each subnet

201 . 70 . 64 . 0

31

201 . 70 . 64 . 31

201 . 70 . 64 . 32

31

201 . 70 . 64 . 63

201 . 70 . 64 . 64

31

201 . 70 . 64 . 95

201 . 70 . 64 . 96

31

201 . 70 . 64 . 127

201 . 70 . 64 . 128

31

201 . 70 . 64 . 159

201 . 70 . 64 . 160

31

201 . 70 . 64 . 191

201 . 70 . 64 . 192

31

201 . 70 . 64 . 223

201 . 70 . 64 . 224

31

201 . 70 . 64 . 255

The range of address for each subnet are :

- Subnet 1: 201.70.64.0 to 201.70.64.31
- Subnet 2: 201.70.64.32 to 201.70.64.63
- Subnet 3: 201.70.64.64 to 201.70.64.95
- Subnet 4: 201.70.64.96 to 01.70.64.127
- Subnet 5: 201.70.64.128 to 201.70.64.159
- Subnet 6: 201.70.64.160 to 201.70.64.191

Super-netting

Why Super-netting

Suppose an organization is started with 250 hosts now they need to increase the No. of hosts to 500 then they can use 2 class C blocks ie., super-netting.

Conditions for super-netting:

1. The No. of blocks must be a power of **2.**
2. The blocks must be contiguous in the address space
3. The 3rd byte of the 1st address in the super block must be evenly divisible by the number of blocks.

Example

A company needs 600 addresses which of the following set of class C blocks can be used to form a super-net for this company?

a) 198.47.32.0 198.47.33.0 198.47.34.0

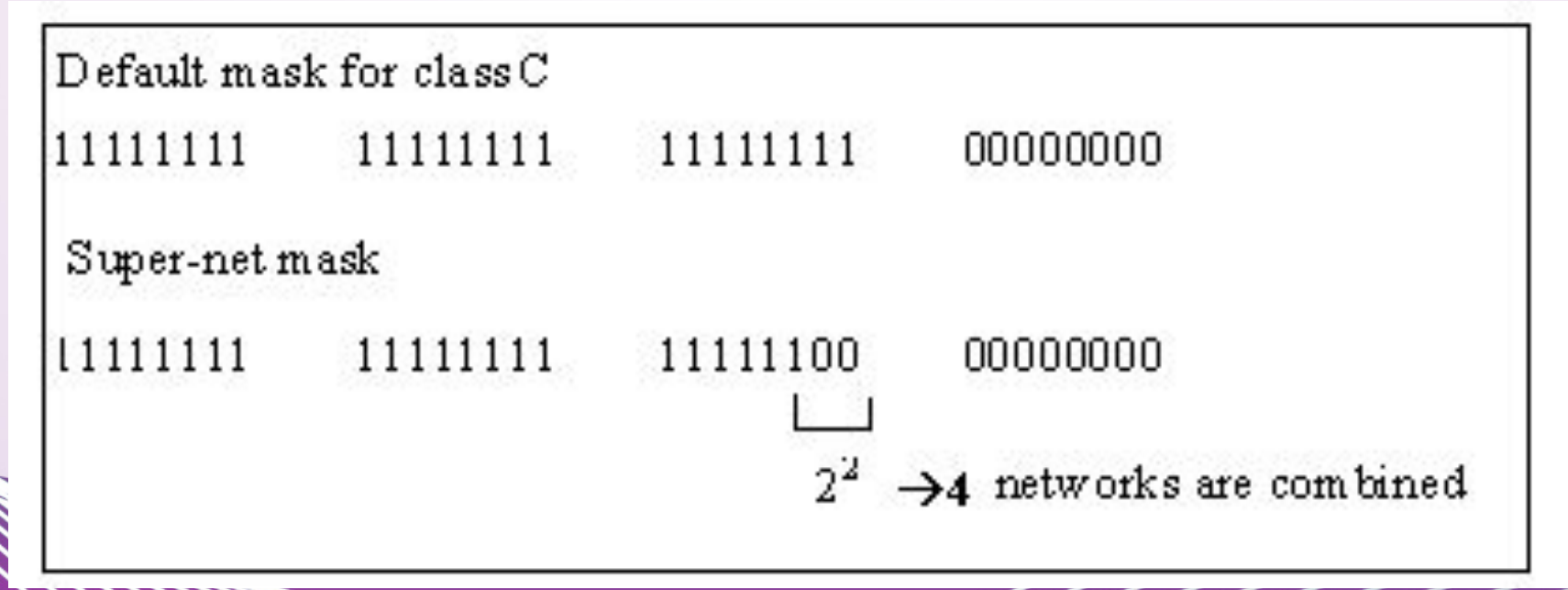
b) 198.47.32.0 198.47.42.0 198.47.52.0 198.47.62.0

c) 198.47.31.0 198.47.33.0 198.47.33.0 198.47.52.0

d) 198.47.32.0 198.47.33.0 198.47.34.0 198.47.35.0

Super-net mask

- In the subnet mask the no. of 1's will be more than the default mask
- In super-netting the no. of 1'swill be less than the default mask



Using super-net mask to find the range of address

- Compare the super-net mask and the default mask to get the no. of blocks.
- When we know that the default mask has **24** 1’s then, subtract the number of 1’s in the super-net mask from 24 gives us the number of blocks.

Example

A super-net has a 1st address of 205.16.32.0 & a super-net mask of 255.255.248.0. How many blocks are in this super-net and what is the range of address.

201.16.32.0 →Class C address							
The default mask has 24 1's (201.16.32.0)	11111111	.	11111111	.	11111111	.	00000000
The super-net has 21 1's (255.255.248.0)	11111111	.	11111111	.	11111000	.	00000000

Example(Cont…)

1. The super-net has 21 1’s
2. The default mask has 24 1’s (Default mask for class
3. Difference=3
4. No. of blocks =2^3 8
5. The blocks are 205.16.32.0 to 205.16.39.0
6. The 1^{st} address=205.16.32.0
7. The last address=205.16.39.255
8. The total address= 8 * 256 2048

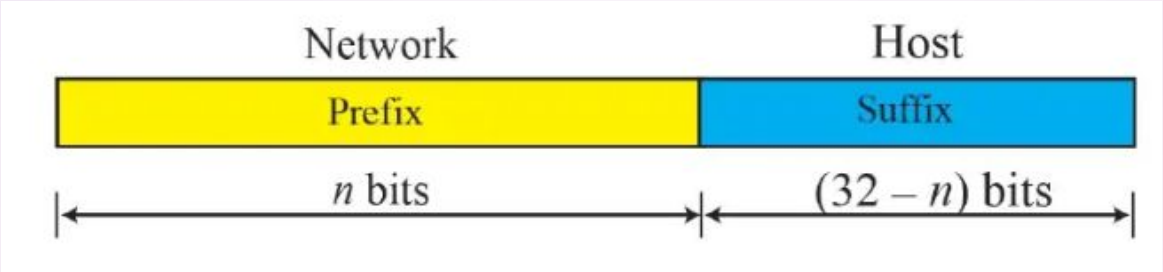
Classless Addressing

- To reduce the wastage of IP addresses in blocks we subnetting. But in **Classless addressing** wastage of IP addresses in a block is more reduced than Classful subnetting.
- Disadvantage of Classful Addressing:
  - Class A with a mask of 255.0.0.0 can support 16, 777, 214 addresses
  - Class B with a mask of 255.255.0.0 can support 65, 534 addresses
  - Class C with a mask of 255.255.255.0 can support 254 addresses
- But what if someone requires 2000 addresses ?
  - One way to address this situation would be to provide the person with class B network. But that would result in a waste of so many addresses.

- Another possible way is to provide multiple class C networks, but that too can cause a problem as there would be too many networks to handle.

- To resolve problems **CIDR** or **Class Inter-Domain Routing** was introduced: variable-length blocks. In this variable length, blocks are used that belongs to no class.

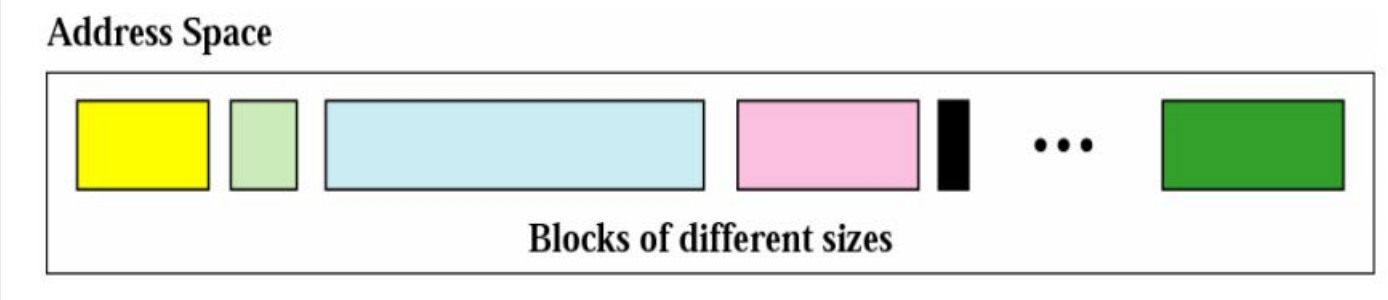
- In classfull addressing, the prefix defines the network and the suffix defines the host.



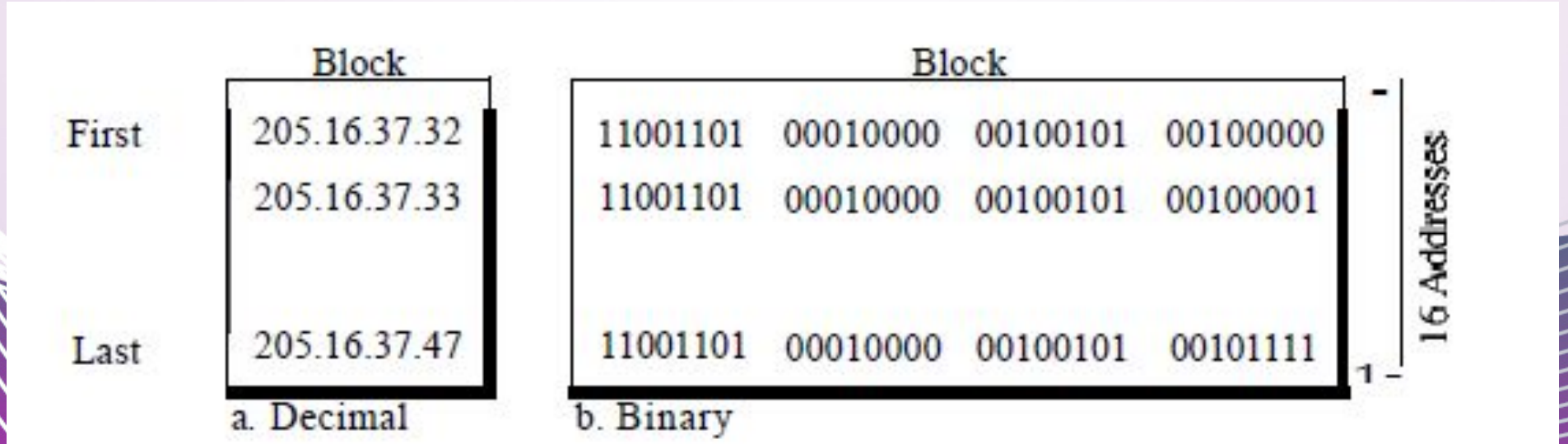
- The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix.
- The prefix is common to all addresses in the network; the suffix changes from one device to another.
- ONLY 2 levels of hierarchical structure (prefix followed by suffix)

Whereas in Classless Addressing

- The whole address space (2^32 addresses) is divided into blocks of different sizes
- Variable-Length Blocks



- Restrictions on classless address blocks:
  1. The addresses in a block must be contiguous, one after another.
  2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
  3. The first address must be evenly divisible by the number of addresses.
- Example:A block of 16 addresses granted to a small organization
  - The addresses are contiguous.
  - The number of addresses is a power of 2 (16 = 24)
  - the first address is divisible by 16. (The first address, when converted to a decimal number, is 3,440,387,360, which when divided by16 results in 215,024,210.)



- Classless Addressing is also called the ***slash notation or CIDR notation*** (Classless Inter-Domain Routing)
- In IPv4 Classless addressing, a block of addresses can be defined as

x.y.z.t /*n*

in which **x.y.z.t** defines one of the addresses and the ***/n*** defines the ***mask***.

- ***Mask*** is a 32-bit number in which the *n* leftmost bits are 1s and the 32 - *n* rightmost bits are 0s

Prefix Lengths

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0	/32	255.255.255.255

Example 1:

What is the network address if one of the addresses is 167.199.170.82/27?

Solution

The prefix length is 27

We must keep the first 27 bits as it is and change the remaining bits (5) to 0s.

The 5 bits affect only the last byte.

The last byte is 01010010.

Changing the last 5 bits to 0s, we get 01000000 or 64.

The network address is 167.199.170.64/27

Example 2:

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−29 rightmost bits to 0, we get

11001101 00010000 00100101 00100000

or

205.16.37.32.

Example 3:

What is the first address in the block if one of the addresses is 140.120.84.24/20?

Solution

140.120.84.24

ANDing with mask 255.255.80.0 (0s 32-21 bits)

140.120.4.0

The first address is 140.120.80.0/20.

Example 4:

Find the last address for the block 205.16.37.39/28

Solution

The binary representation of the given address is

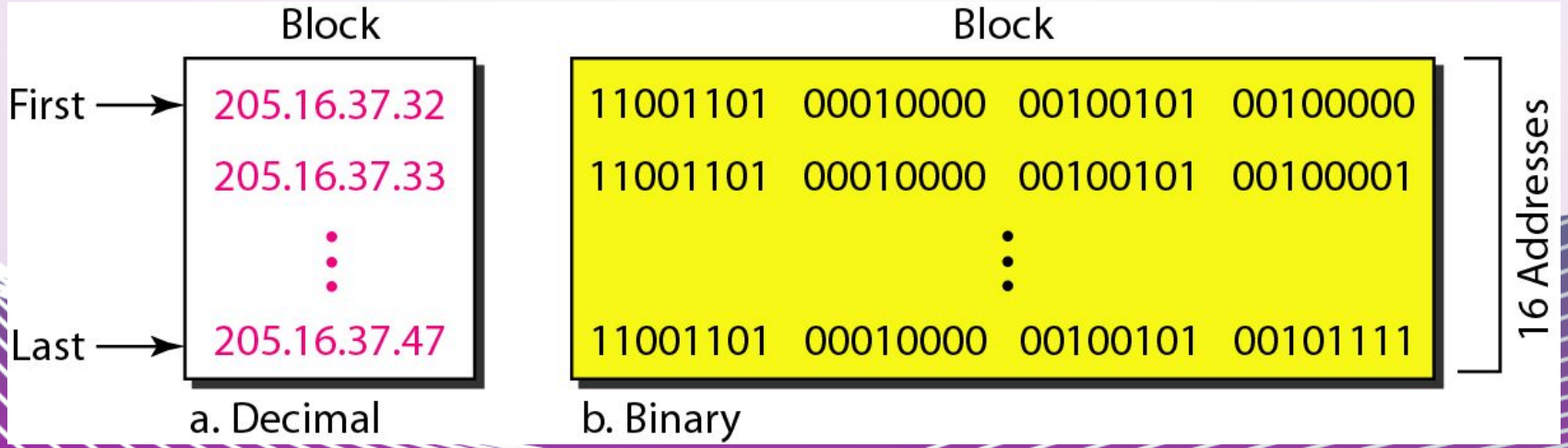
11001101 00010000 00100101 00100111

If we set 32 − 29 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

205.16.37.47



Example 5:

find the last address in the block if one of the addresses is 140.120.84.24/20.

Solution

To the netwok address (Example 3) convert suffix to all 1s.

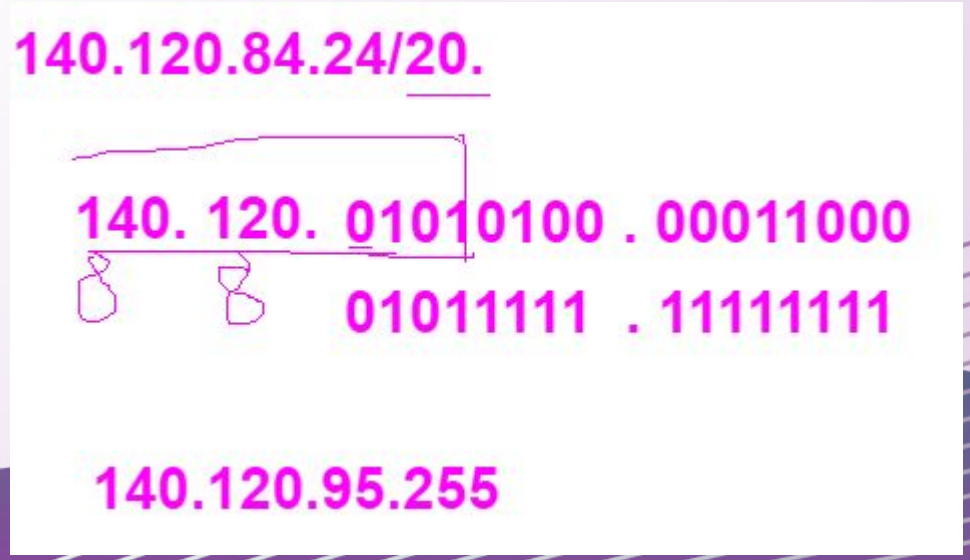
140 . 120 . 80 . 0

140 . 120 .15 . 255

---

140 . 120 . 95 . 255

The last address is 140.120.95.255/20



Other Protocols of Network Layer

DHCP (Dynamic Host Configuration Protocol)

- **is** an application layer program, client/server model helping TCP/IP in network layer
- DHCP uses UDP well know ports 67 on server and 68 on client
- Need a way to simply connect a computer to a new network – No manual configuration
- DHCP provides dynamic configuration – Client can get a temporary address, and move from network to network

- But each computer attached to a TCP/IP network must know the following:
  - Its IP address
  - subnet mask
  - The IP address of a router (default gateway)
  - The IP address of a name server

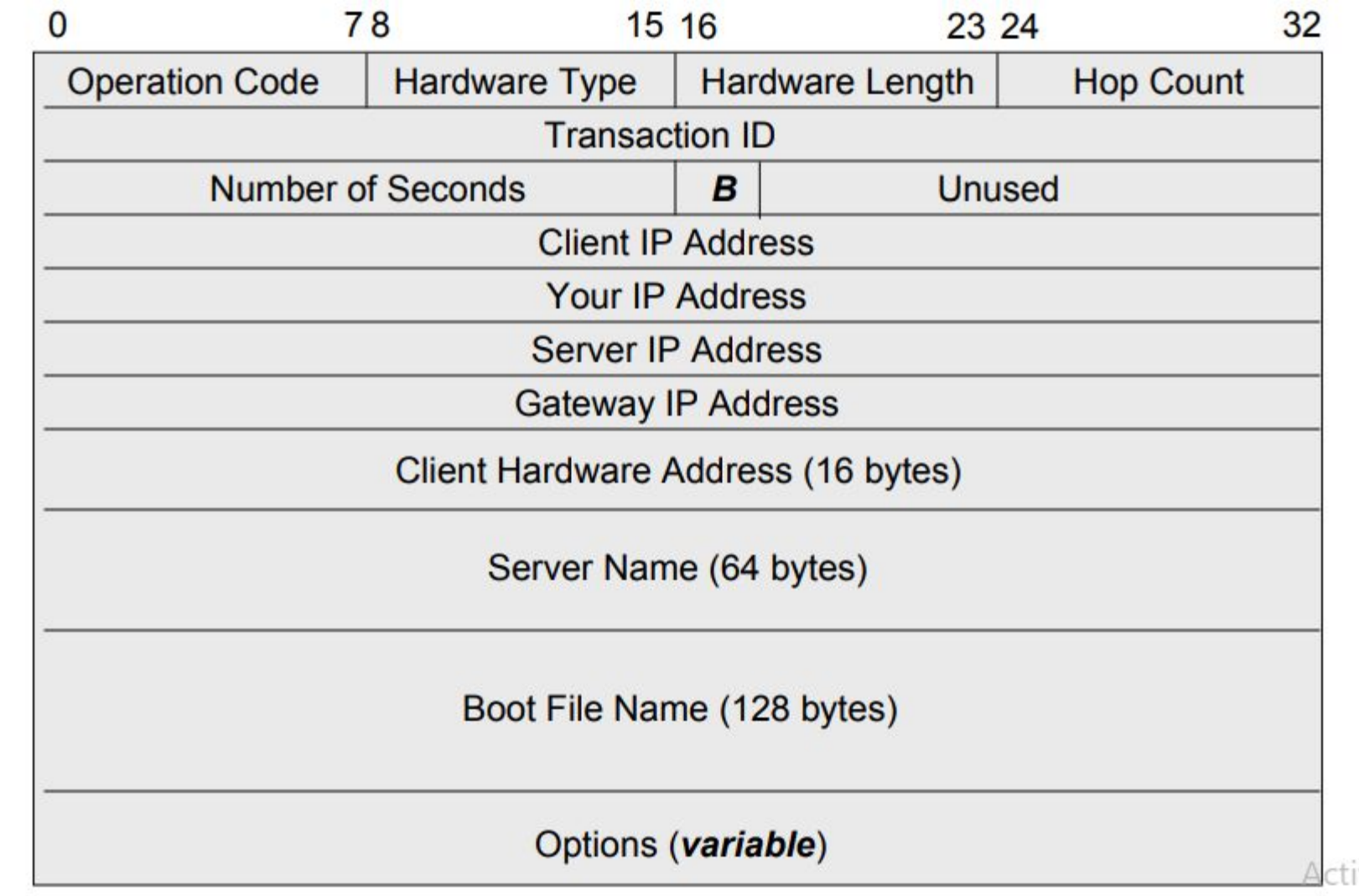
  DHCP can provide these

- DHCP can do static and dynamic address allocation that can be manual or automatic.
  - Static Address Allocation
    - Works like BOOTP
    - DHCP server has a database that statically binds physical addresses to IP addresses.
  - Dynamic Address Allocation
    - dynamically assigns IP addresses to hosts from a pool of available IP addresses.
    - When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.

- On the other
  - On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.
    - DHCP provides temporary IP addresses for a **limited period** of time.
    - The dynamic aspect of DHCP is needed when a **host moves from network to network or is connected and disconnected** from a network frequently

Packet Format

DHCP Message Header



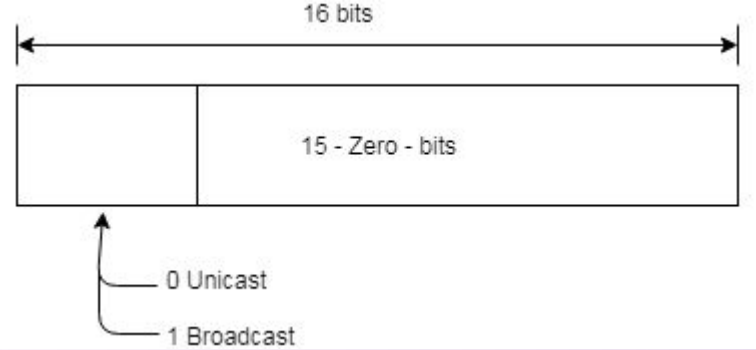
1) **Hop count:** This is an 8 bit field defining the maximum number of hops the packet can travel.

2) **Operation code:** This 8 bit field defines the type of DHCP packet request (1) or reply (2)

3) **Transaction Id:** This is a 4 byte field carrying an integer.The transaction identification is set by the client and is used to match a reply with the request. The server returns the same value in its reply.

4) **Number of seconds:** This is a 16 bit field that indicated the number of seconds elapsed since the time the client started to boot.

5) **Flag:** This is a 16 bit field in which only the leftmost bit is used and the rest of the bits should be set to Os. A left most bit specifies a forced broadcast reply (instead of uncast) from the server. If the reply mere to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client.



6) **Your IP address:** This is a 4 byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.

7) **Server IP address:** This is a 4 byte field containing the server IP address. It is filled by the server in a reply message.

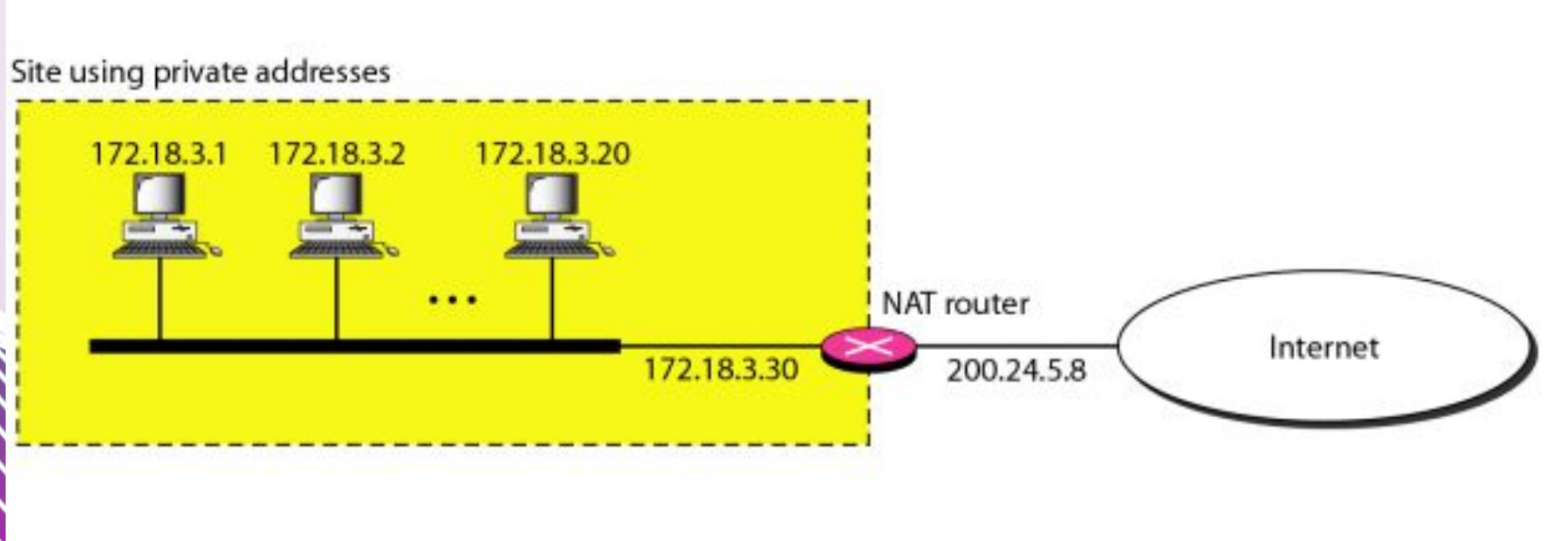
Network Address translation (NAT)

- number of users increasing, there is shortage of addresses
- A quick solution to this problem is called network address translation (NAT).
- NAT enables a user to have a ***large set of addresses internally*** and one address, or a ***smallset of addresses, externally.***
- reserved three sets of addresses as private addresses

Addresses for private networks

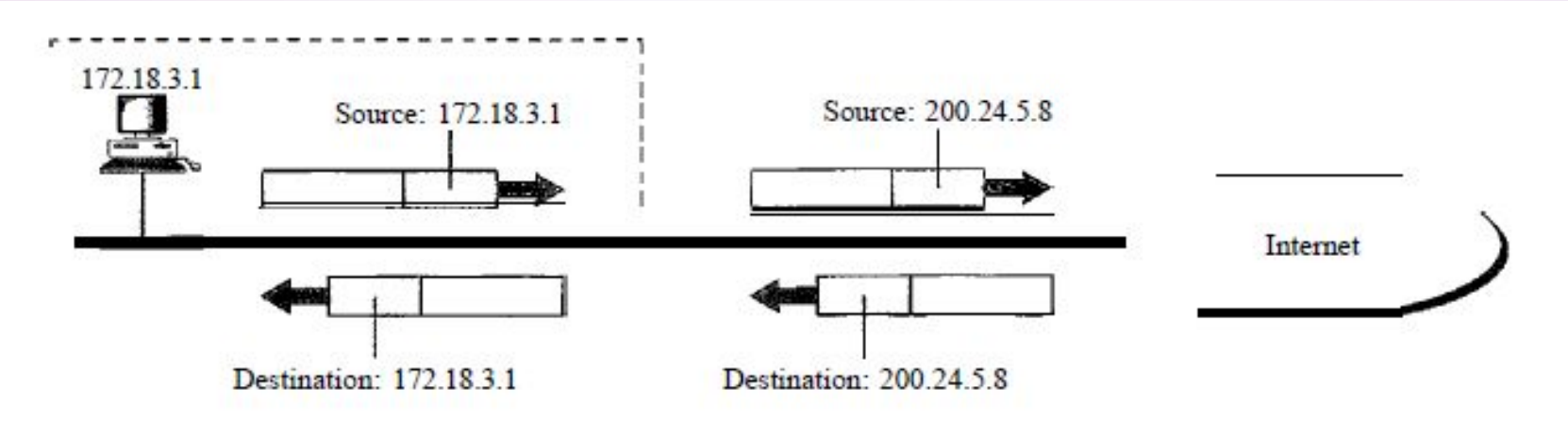
Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

- Everyone knows that these reserved addresses are for private networks.
- They are unique inside the organization, but they are not unique globally.
- No router will forward a packet that has one of these addresses
- The router that connects the network to the global address uses one private address and one global address.

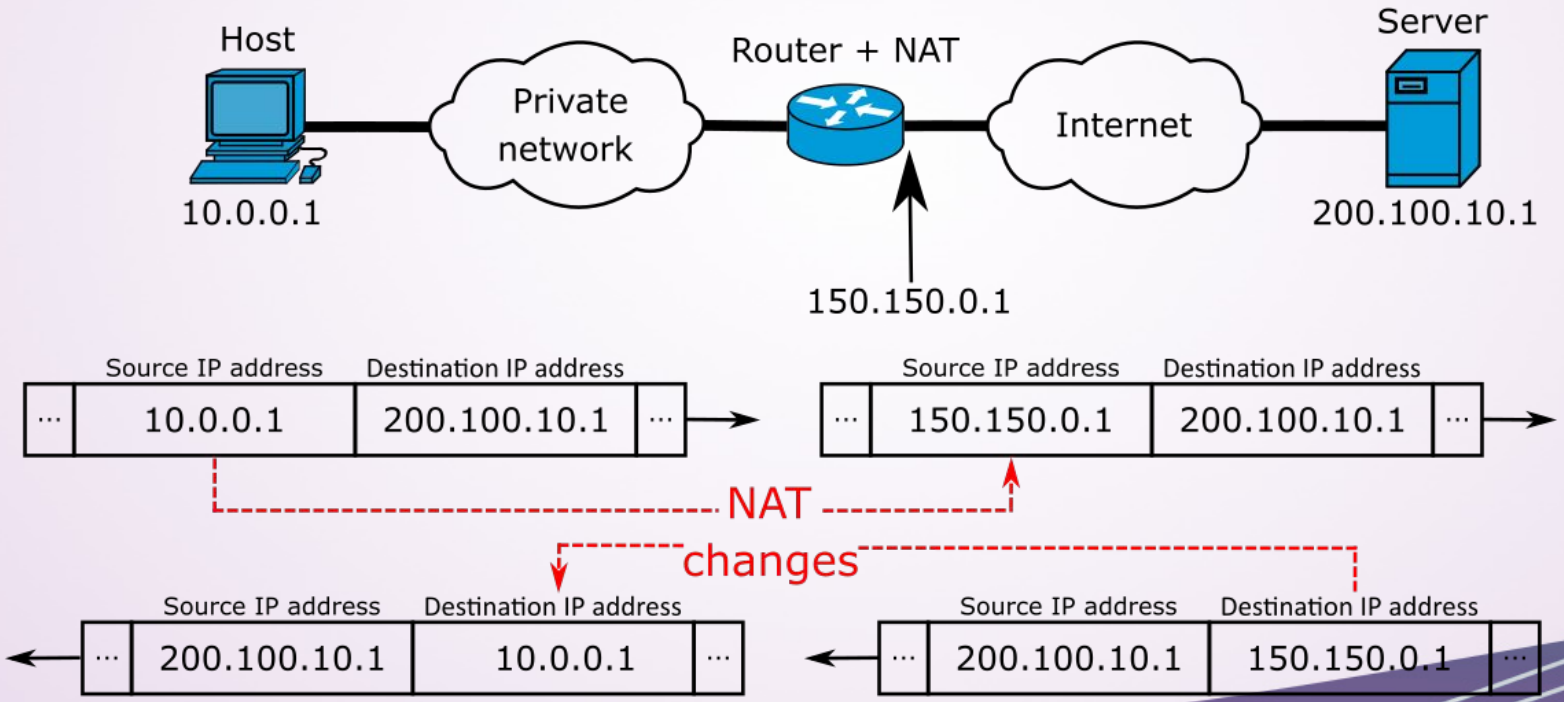


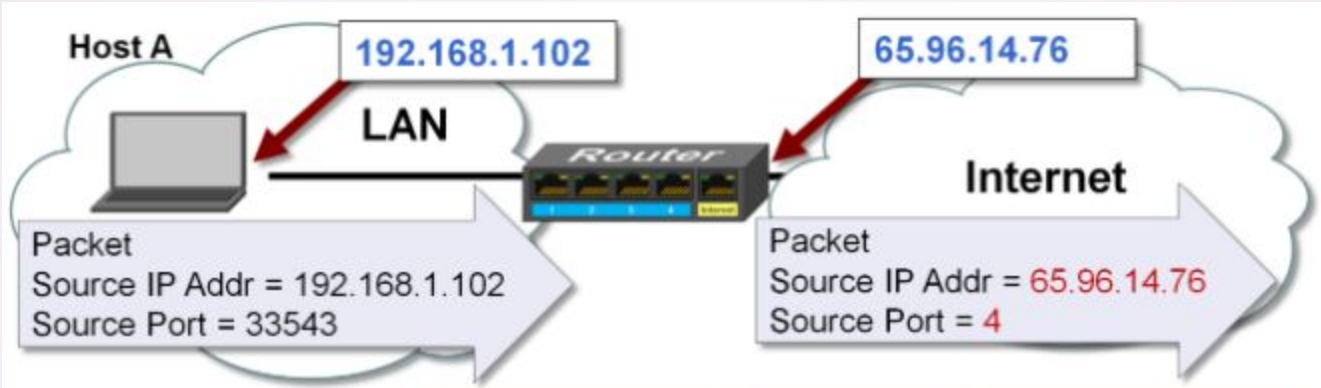
Address Translation

- All 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.



Translation Table





NAT Translation Table

	Local IP Address	Source Port #		Internet IP Address	Source Port #
process X, Host A →	192.168.1.101	54,847	=	65.96.14.76	1
Host B →	192.168.1.103	24,123	=	65.96.14.76	2
process Y, Host A →	192.168.1.101	42,156	=	65.96.14.76	3
Host C →	192.168.1.102	33,543	=	65.96.14.76	4

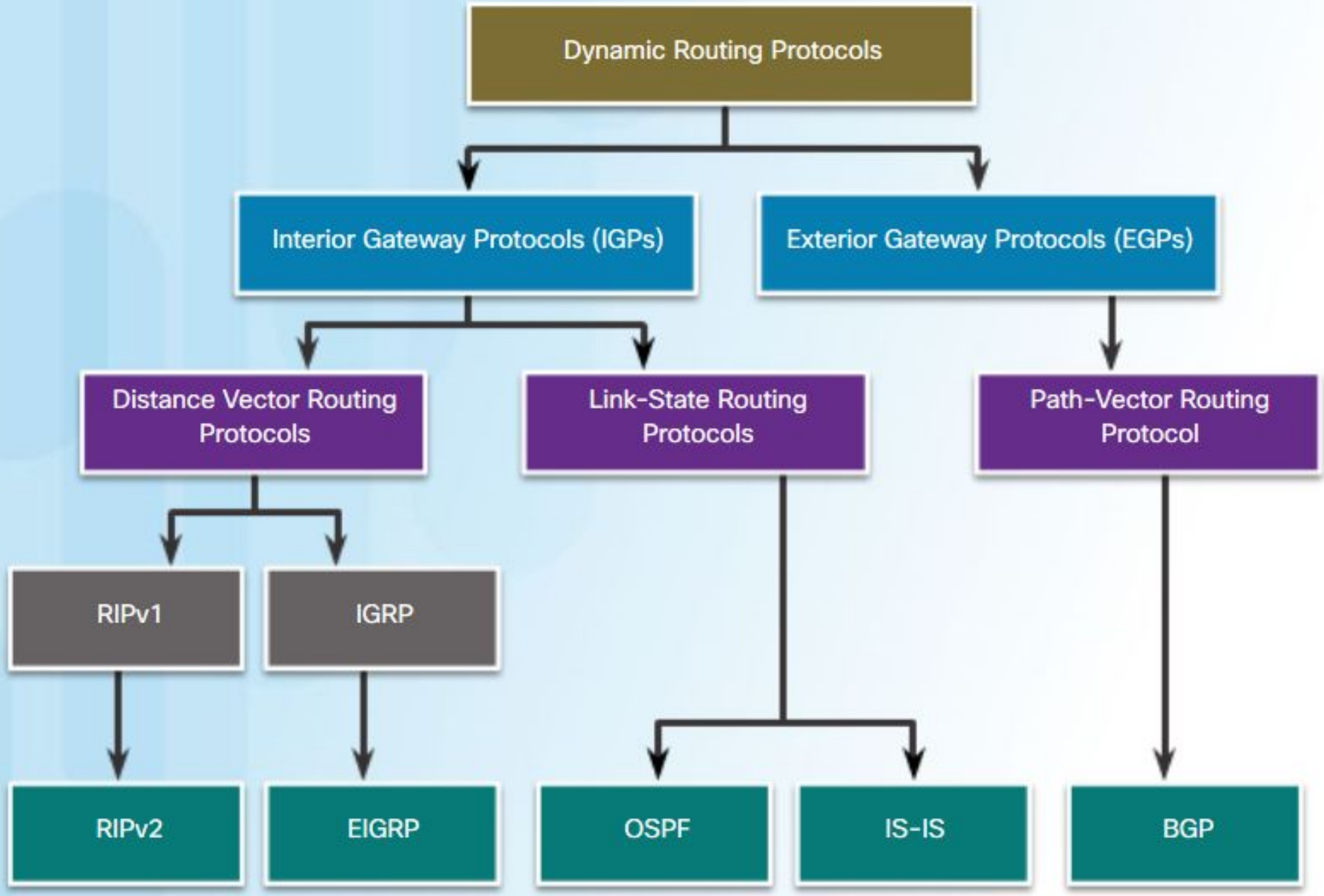
Routing algorithm

- find reachable destinations
- find best paths towards destinations
- best in the sense of some metric - shortest path,number of hops, delay, congestion
- Routing methods can be classified as
  - Static/ Dynamic Routing
  - Intra/ Inter domain Routing

- Static Routing
  - Also referred as Non-Adaptive Routing
  - Manual: follows user defined routing and routing table is not changed until network administrator changes it.
  - simple routing algorithms
  - more security than dynamic routing.
  - implemented in smaller networks.
- Dynamic Routing
  - Also referred as Adaptive Routing
  - routing table changes once changes occur in network nor network topology changes

- During network change, dynamic routing sends a signal to router, recalculates the routes and send the updated routing information.
- less secure
- Automatic
- implemented in large networks

- Intra Domain Routing
  - works only within domains,need to know only about other routers within their domain/ Autonomous System (AS).
  - Also called as Interior-gateway protocols (IGP).
- Inter Domain Routing
  - Routing algorithm works within and between domains, needs to know only about other routers within and between their domain
  - Also called as Exterior-gateway protocols (EGP).


Dynamic Routing Protocols
Interior Gateway Protocols (IGPs)
Exterior Gateway Protocols (EGPs)
Distance Vector Routing Protocols
Link-State Routing Protocols
Path-Vector Routing Protocol
RIPv1
IGRP
RIPv2
EIGRP
OSPF
IS-IS
BGP

Distance Vector Algorithm

- Iterative, distributive and Dynamic Routing technique
- Used in internet as RIP (Routing Information Protocol)
- based on Bellman-Ford equation
- Initially,
  - given nodes know the distance to all it’s neighbours
- Finally,
  - Distance to all other nodes is know, with the next hop

- Each router maintains Routing Table (Distance Vector) which contains
  - Destination
  - Cost to destination
  - Next hop to reach destination
- Each node exchanges with all its neighbours (ONLY) “Routing Table” info
  - Destination and **Estimated** cost to the destination
  - Not hop info is not shared

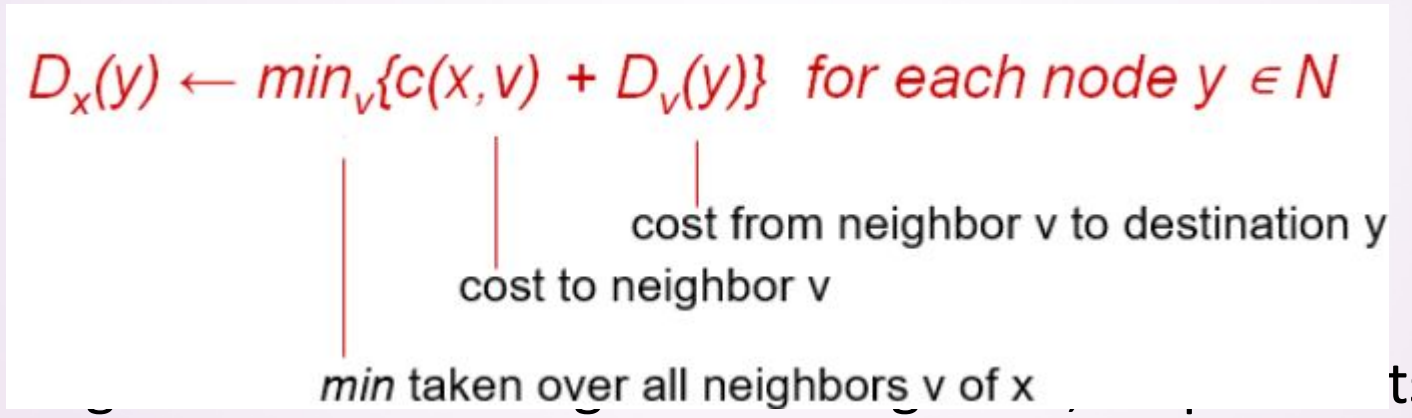
- Each node exchanges with its neighbours (ONLY) DV
  - destination and current distance
  - Not next hop
- The neighbour on receiving the Routing table, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow min_v\{c(x,v) + D_v(y)\} \quad for\ each\ node\ y \in N$$

  - the estimate Dx(y) converge to the actual least cost dx(y)

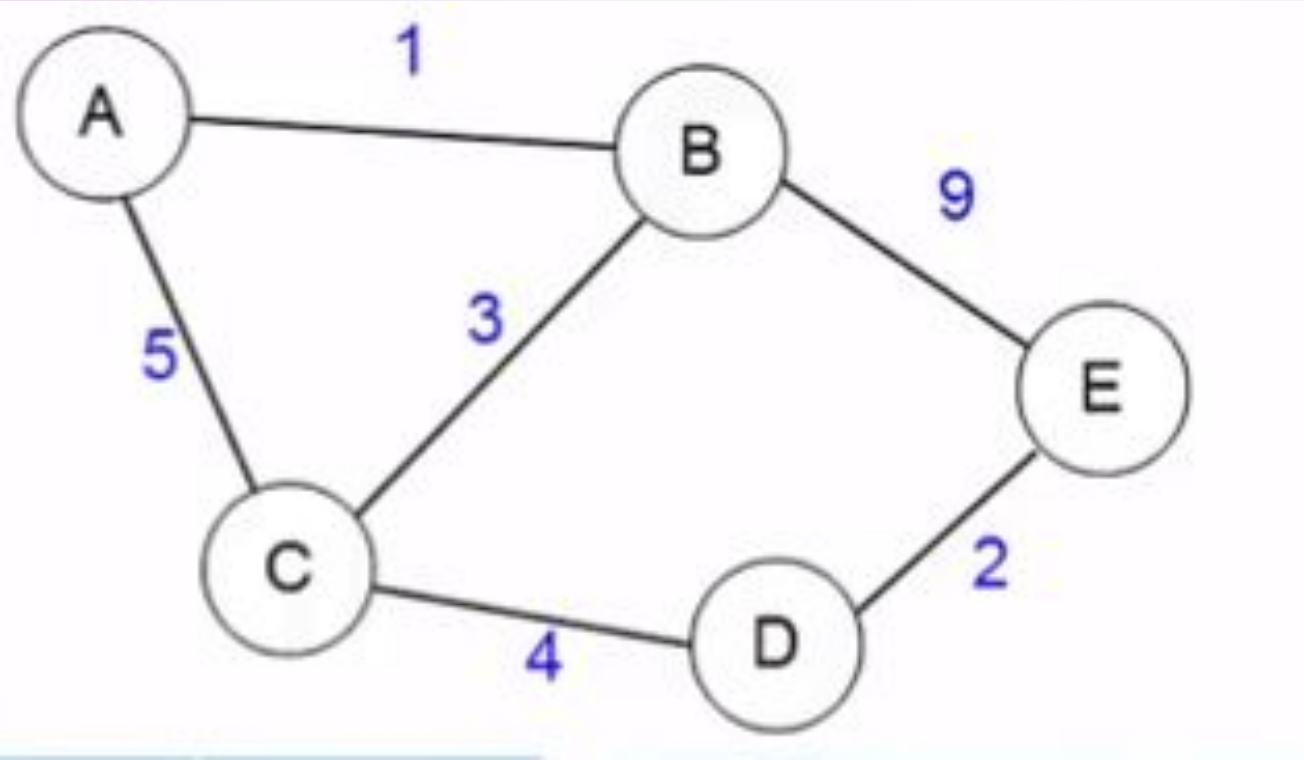
- Bellman Ford Equation
  - $d_x(y)$:= cost of least-cost path from x to y

  Then



  - The ts own DV this B-F equation
  - the estimate Dx(y) converge to the actual least cost dx(y)
  -

Example



Initial Distance Vector of B

Dest	Cost	Next Hop
A	1	A
C	3	C
E	9	E

Updated Distance Vector of B (after updating C's DV into B)

Dest	Cost	Next Hop
A	1	A
C	3	C
D	7	C
E	9	E

Initial Distance Vector of all Nodes

Table A

D	C	H
A	0	A
B	1	B
C	5	C

Table B

D	C	H
A	1	A
C	3	C
E	9	E

Table C

D	C	H
A	5	A
B	3	B
D	4	D

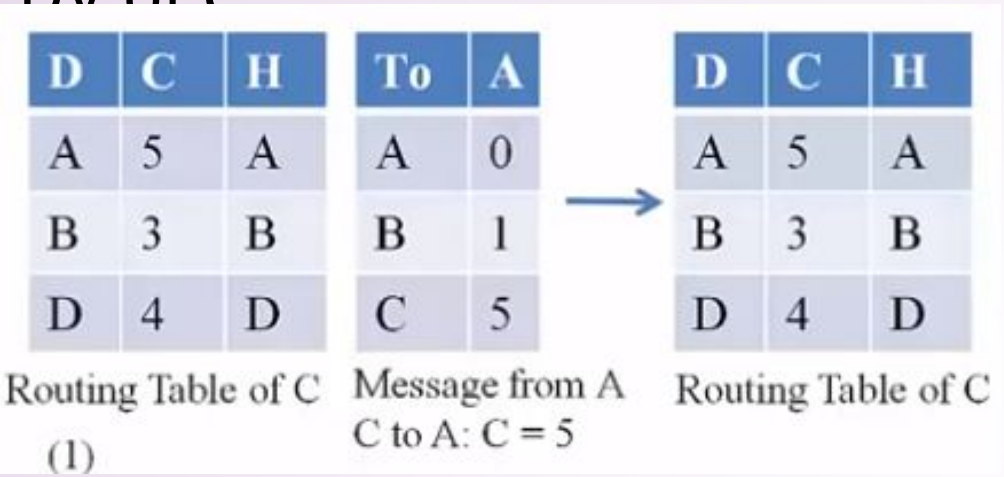
Table D

D	C	H
C	4	C
E	2	E

Table E

D	C	H
B	9	B
D	2	D

DV of C



D	C	H
A	5	A
B	3	B
D	4	D

Routing Table of C

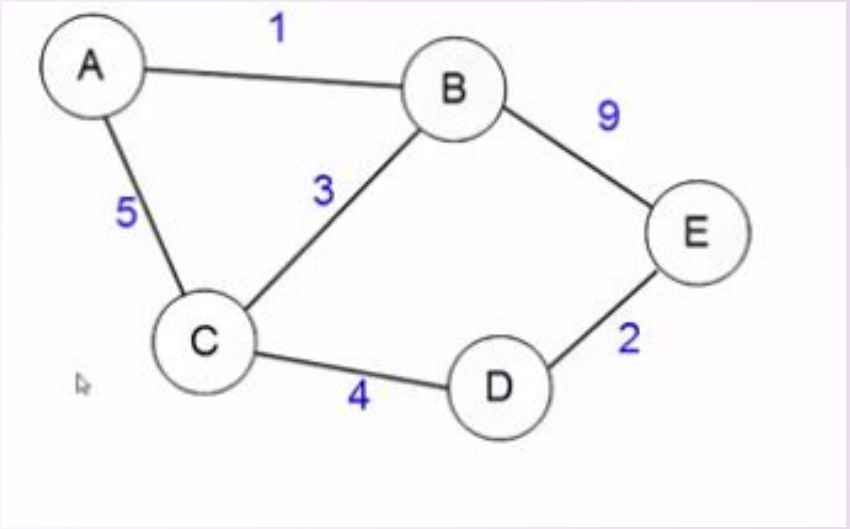
(1)

To	A
A	0
B	1
C	5

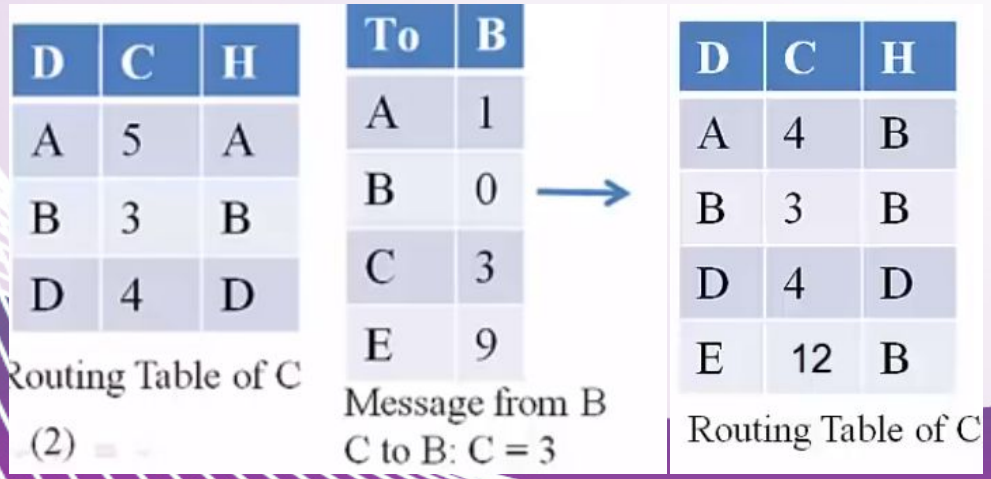
Message from A
C to A: C = 5

D	C	H
A	5	A
B	3	B
D	4	D

Routing Table of C



DV of C after B updation



D	C	H
A	5	A
B	3	B
D	4	D

Routing Table of C

(2)

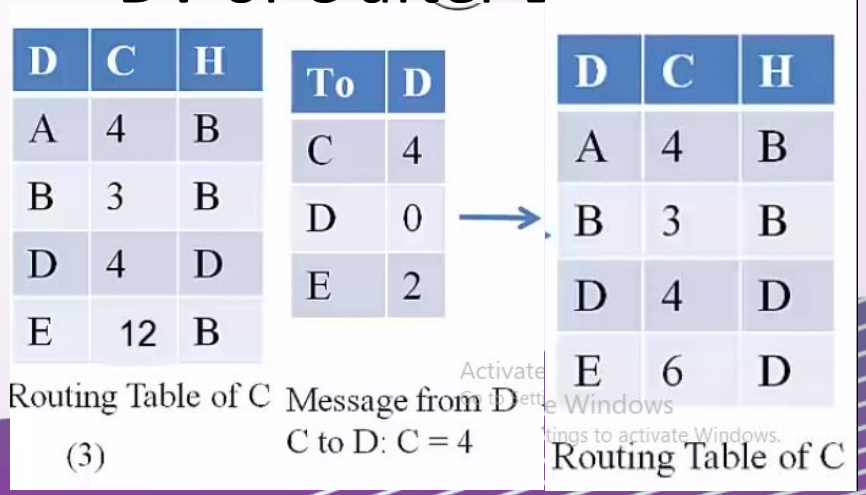
To	B
A	1
B	0
C	3
E	9

Message from B
C to B: C = 3

D	C	H
A	4	B
B	3	B
D	4	D
E	12	B

Routing Table of C

DV of C after D updation



D	C	H
A	4	B
B	3	B
D	4	D
E	12	B

Routing Table of C

(3)

To	D
C	4
D	0
E	2

Message from D
C to D: C = 4

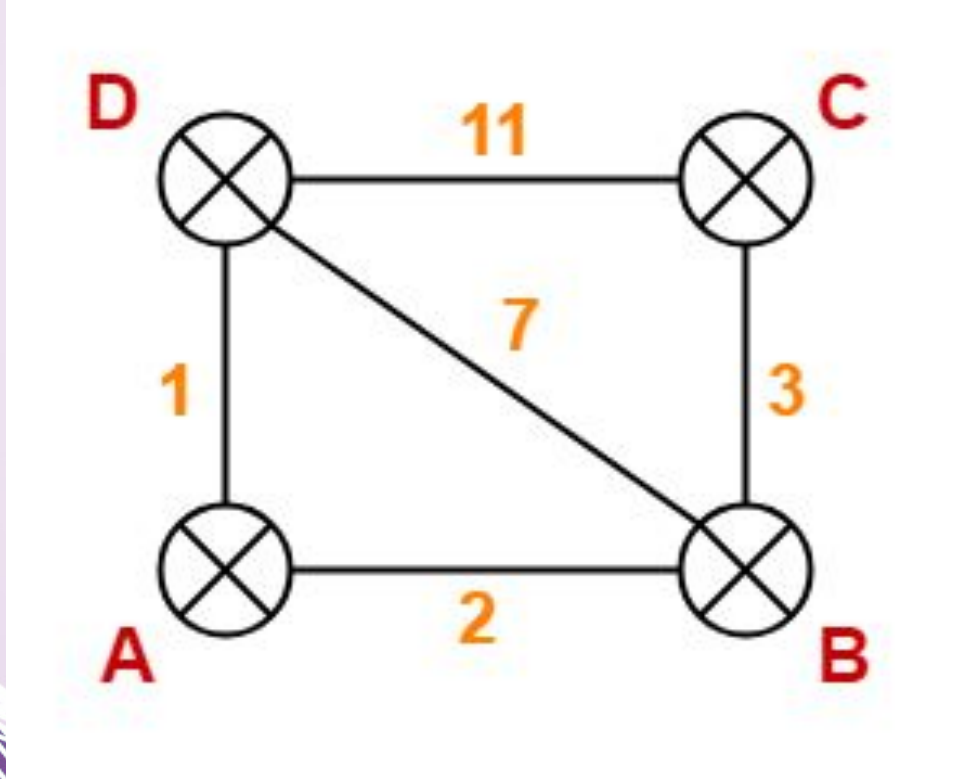
D	C	H
A	4	B
B	3	B
D	4	D
E	6	D

Routing Table of C

- After few rounds, the DV will get updated if there is no change in topology
- the state of nodes is distributes, a node knows abouts it’s neighbours only
- All nodes keep sharing their DV states on regular interval
- Helps in finding broken links

Try it yourself

Find the Routing table at A, B, C, D



Solution

After 1 Round

Table A (received from B, C)

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

After 1 Round

Table B (received from A, C, D)

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

After 1 Round

Table C (received from A, B, C)

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

After 1 Round

Table D (received from A, B, C)

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

After 2nd Round

Table A (received from B, C)

no change

Table B

no change

Table C

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	6	B

Table D

Destination	Distance	Next Hop
A	1	A
B	3	A
C	6	A
D	0	D

Link State Routing

- Global Routing Table
- Also called shortest path first (SPF) forwarding
- Named after Dijkstra's algorithm (1959) which it uses to compute routes
- All routers have tables which contain a representation of the entire network topology (Global Rounting Table)
- Each router creates a *link state packet* (LSP) which contains names (e.g. network addresses) and cost to each of its neighbours
- Every time the topology is updated, te LSP too is updated

Algorithm

Step 1 : Flooding

Step 2: Find the Shortest path Tree (Spanning Tree)

The algorithm requires the following information:

- Link state database: List of all the latest LSPs from each router on the network
- Path: Tree structure storing previously computed best paths

Consider this a sort of cache
Data type for nodes: (ID, path cost, port)

- Tent: Tree structure storing paths currently being tested and compared (tentative)

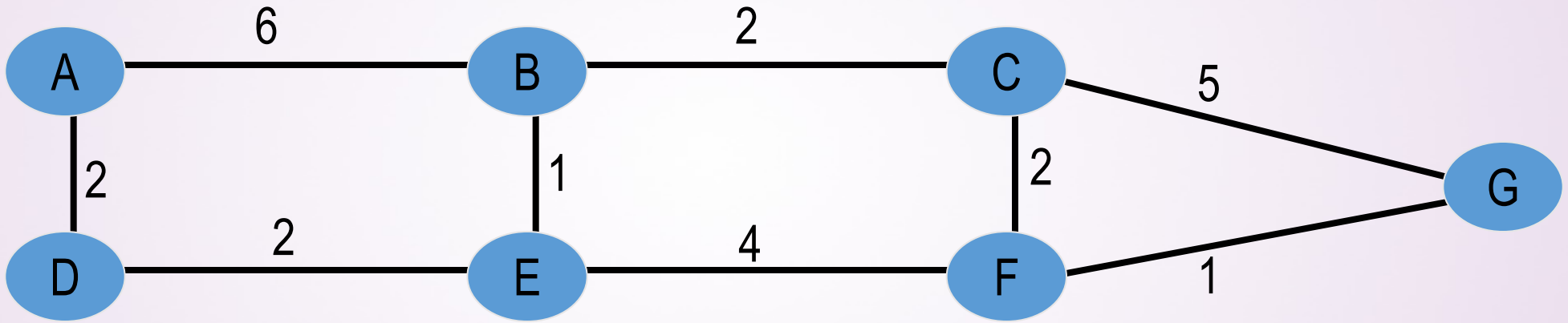
Consider this a sort of rough workspace
Data type for nodes: (ID, path cost, port)

- Forwarding database: Table storing all IDs that can be reached, and the port to which messages should be sent

This is simply a reduced version of the ‘Path’, which contains (destination,port) pairs
This can be used by the router to quickly forward packets for which the best path has already been determined
Data type for table rows: (ID, port)

Dijkstra's LSR Algorithm

- Consider the following network:



Link state database:

A	
B	6
D	2

B	
A	6
C	2
E	1

C	
B	6
F	2
G	5

D	
A	2
E	2

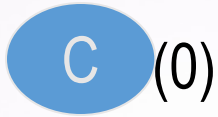
E	
B	1
D	2
F	4

F	
C	2
E	4
G	1

G	
C	5
F	1

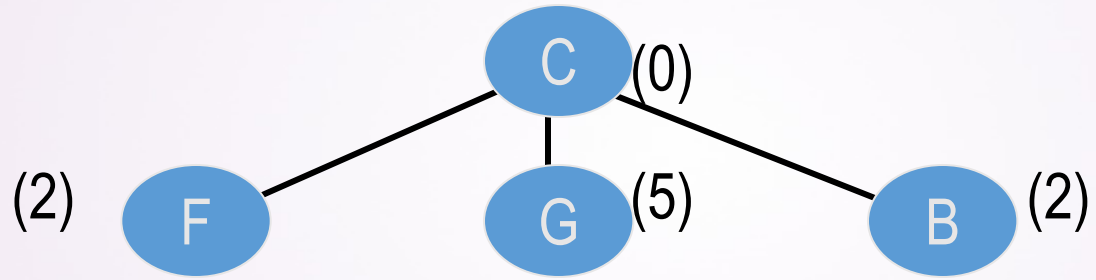
Dijkstra's LSR Algorithm

- Now, if we want to generate a PATH for C:
  - First, we add (C,0,0) to PATH



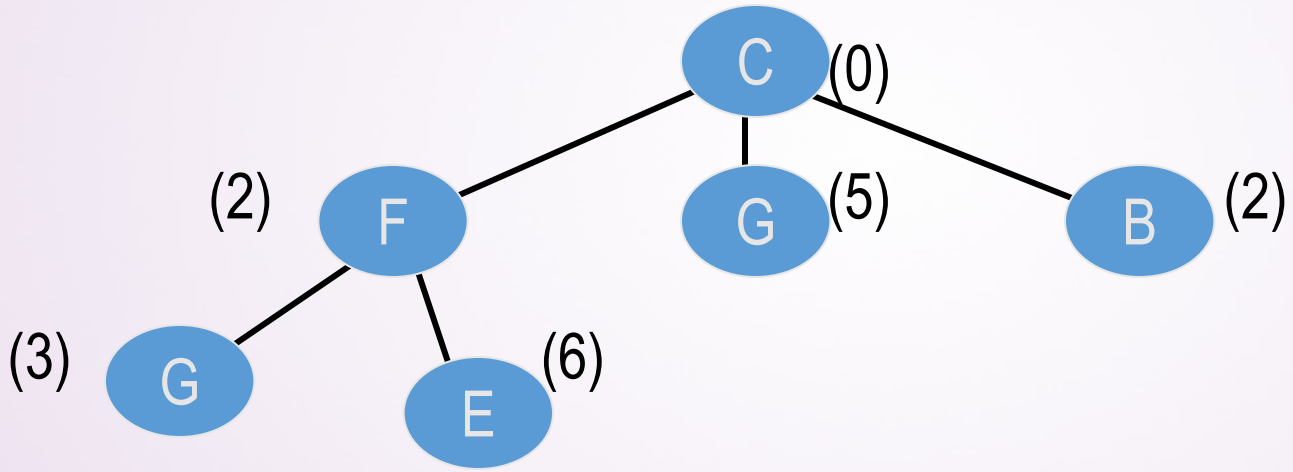
Dijkstra's LSR Algorithm

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



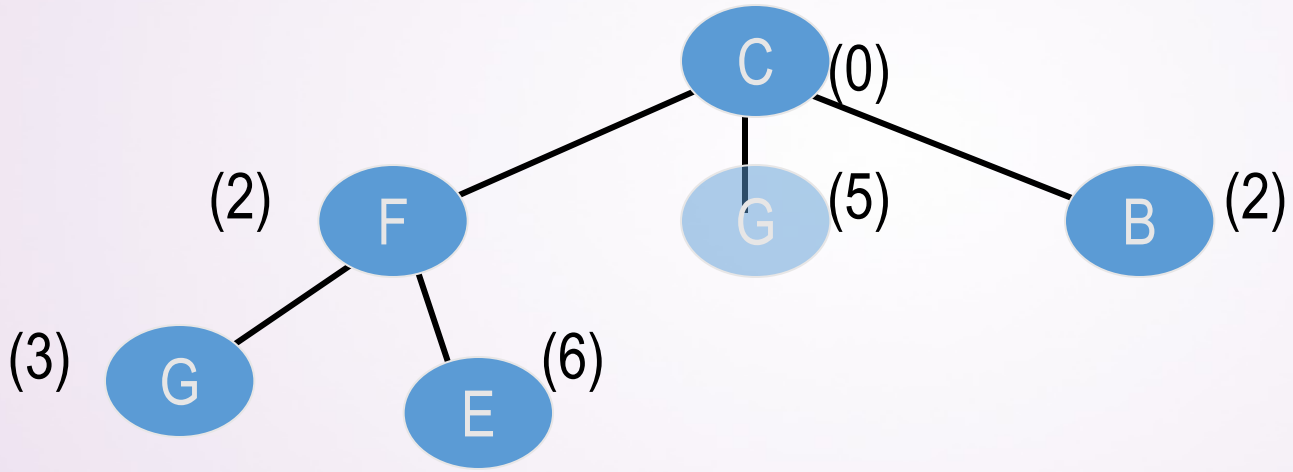
Dijkstra's LSR Algorithm

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



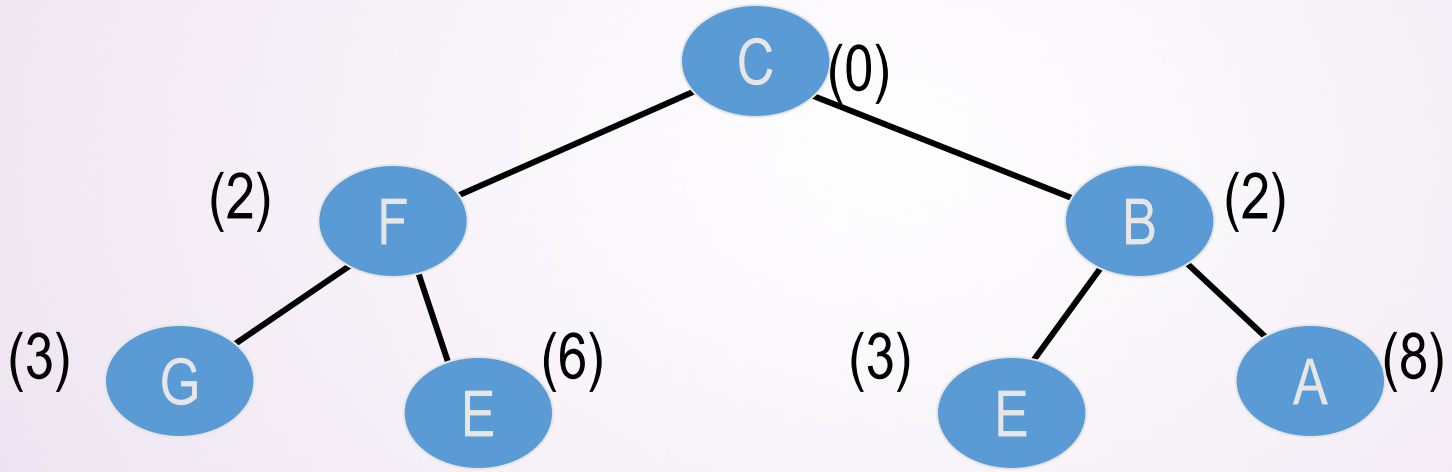
Dijkstra's LSR Algorithm

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



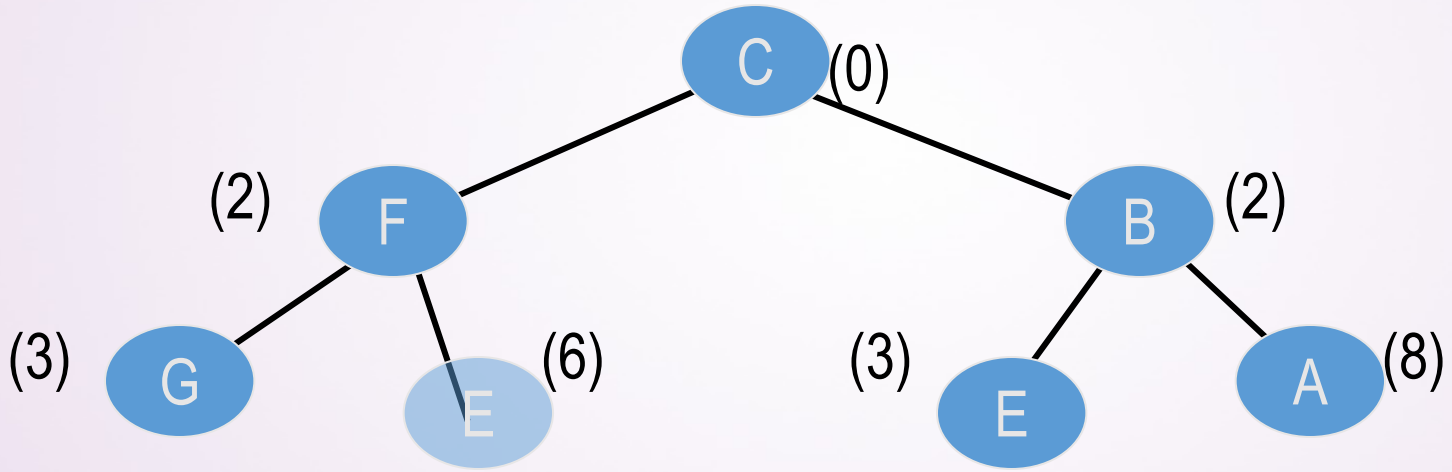
Dijkstra's LSR Algorithm

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



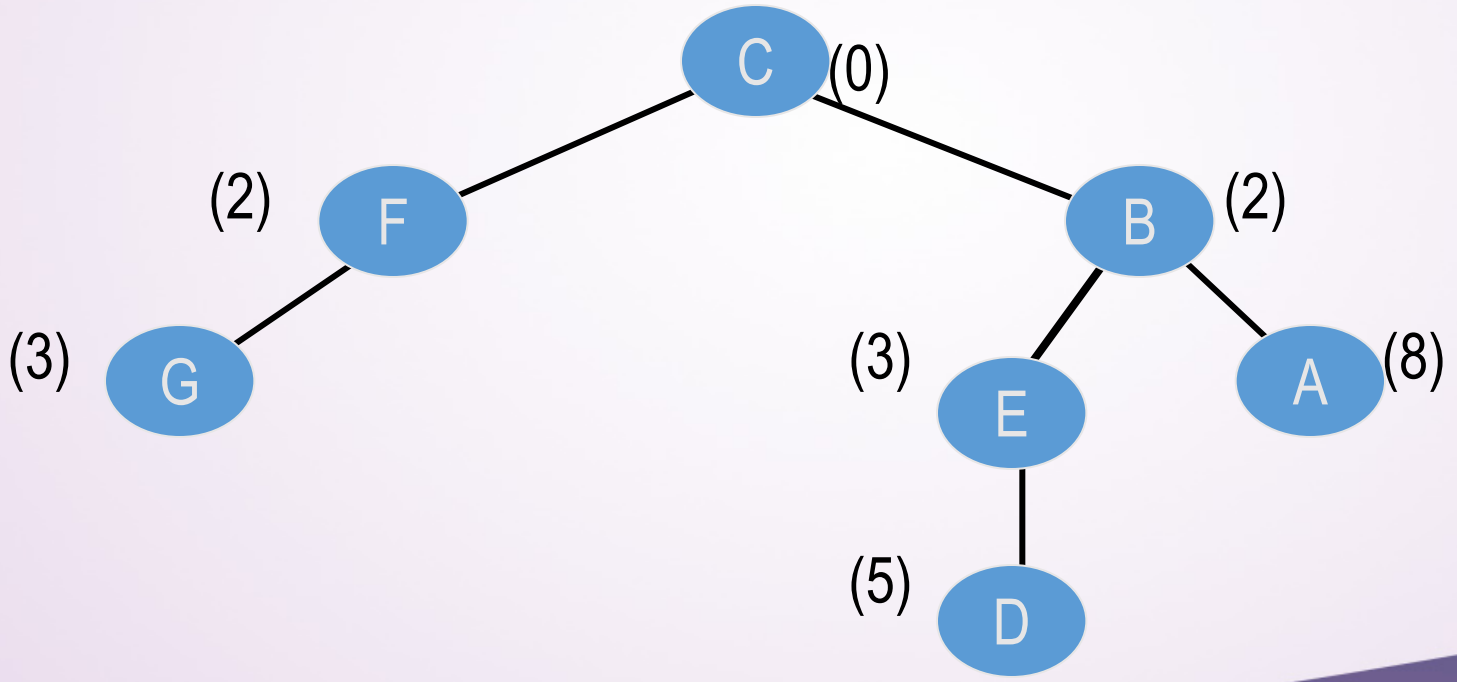
Dijkstra's LSR Algorithm

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



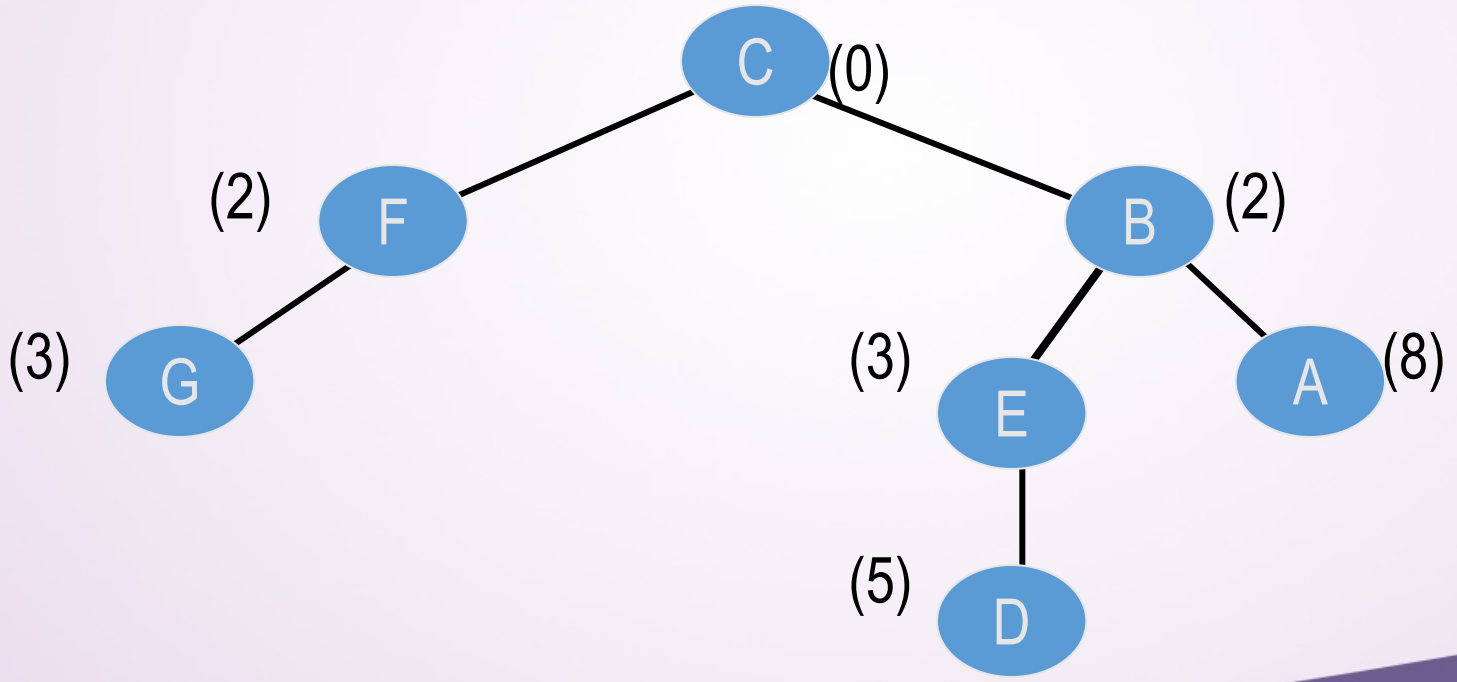
Dijkstra's LSR Algorithm

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



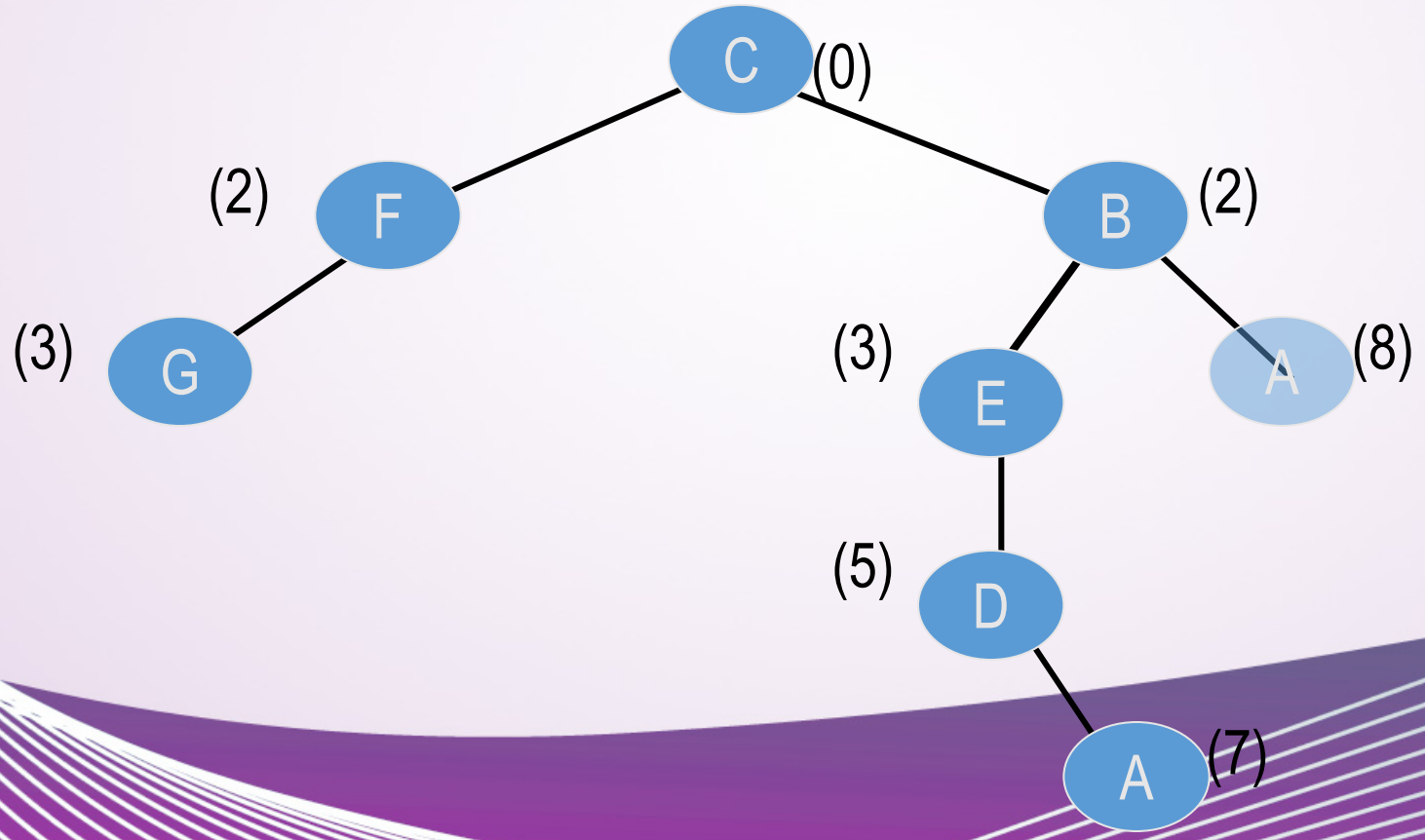
Dijkstra's LSR Algorithm

- Place G in PATH (shown as solid line)
  - All G's LSP elements already exist in TENT



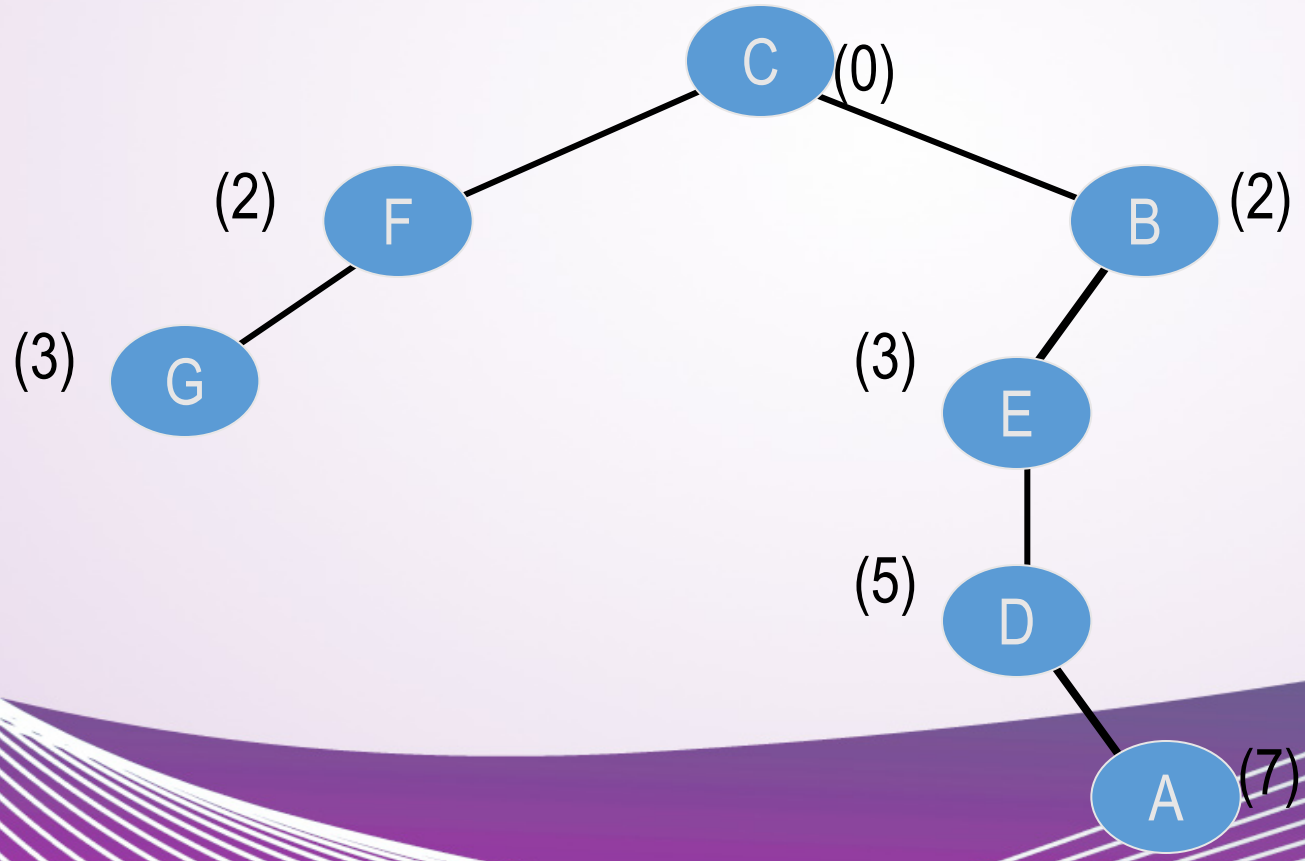
Dijkstra's LSR Algorithm

- Place D in PATH (shown as solid line)
  - Add path to A since it is better than old A



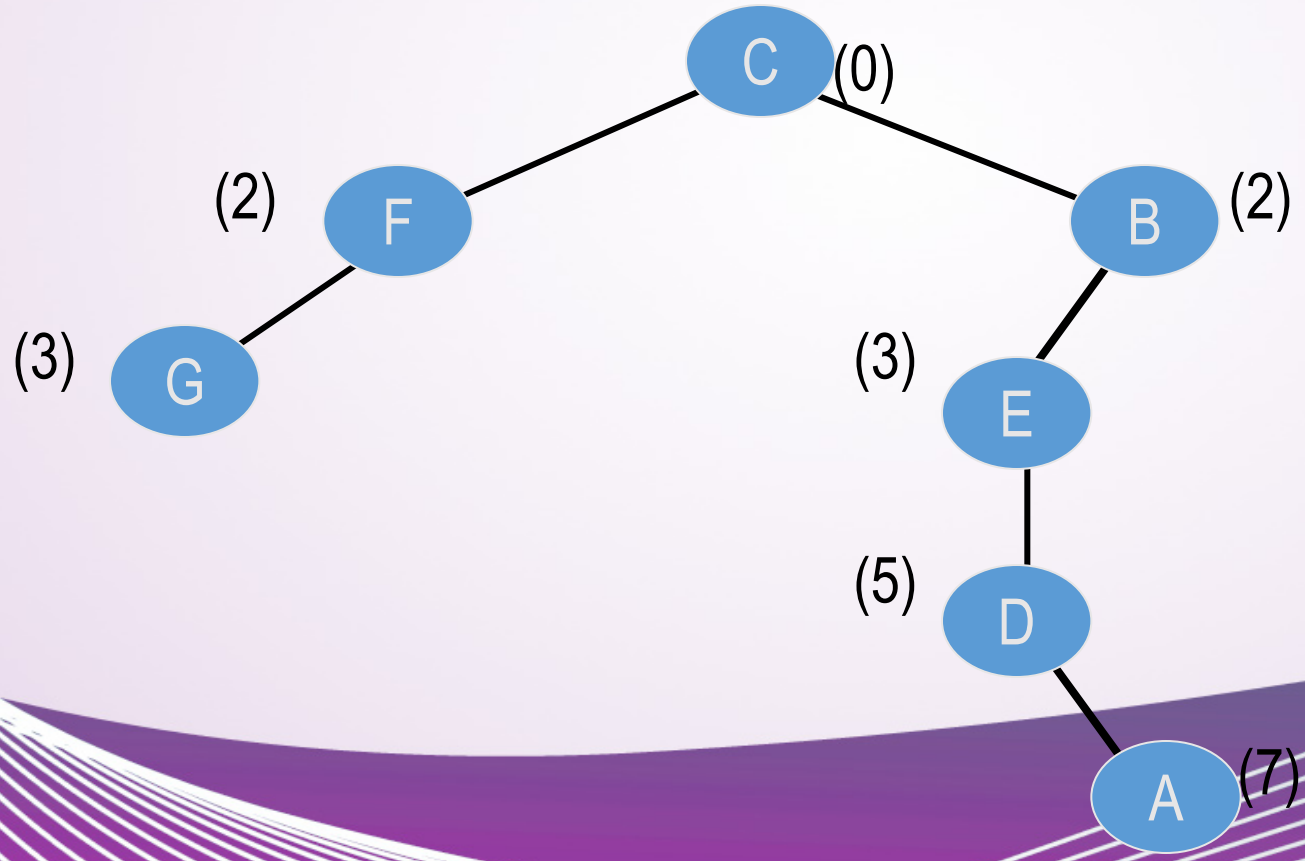
Dijkstra's LSR Algorithm

- Place A in PATH (shown as solid line)
  - All A's LSP elements already exist in PATH



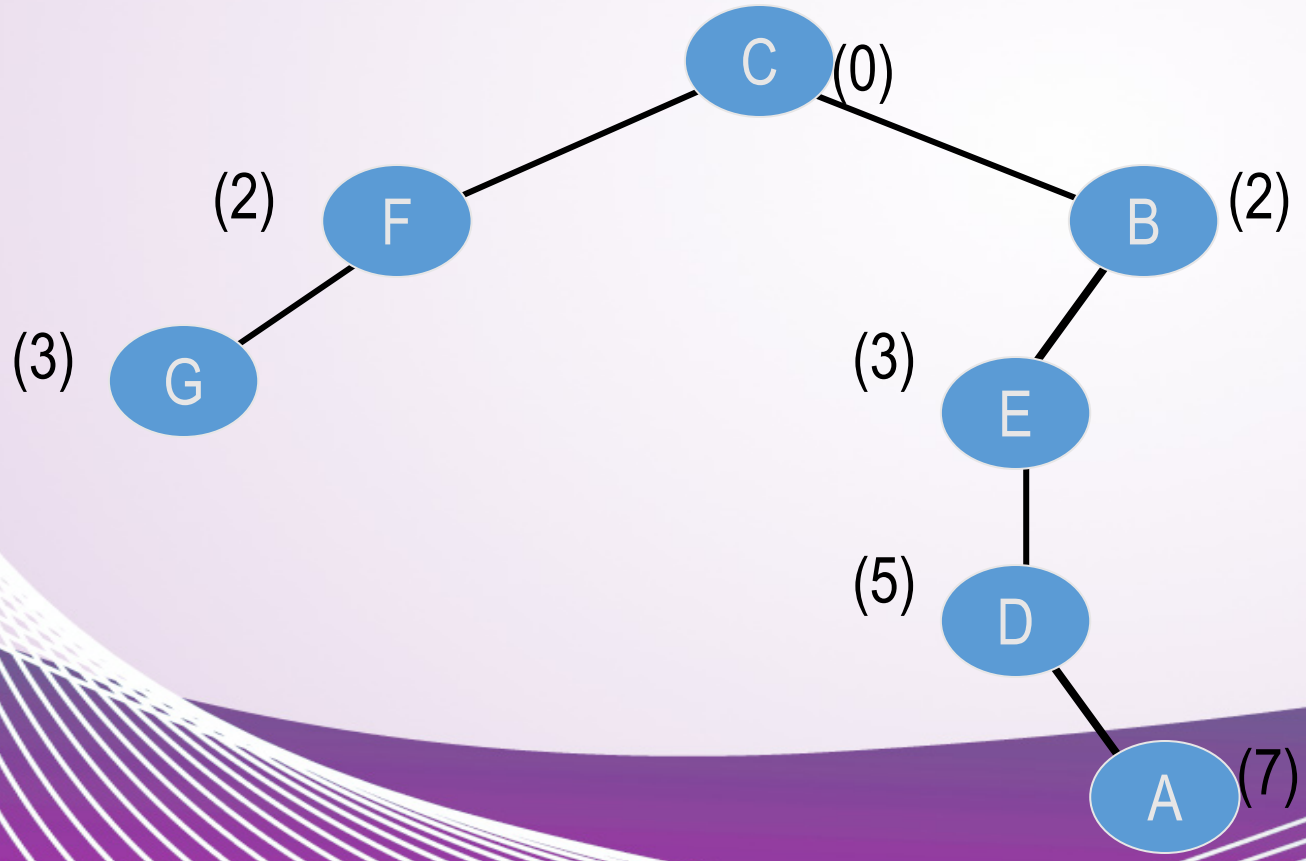
Dijkstra's LSR Algorithm

- We are done since all routes from TENT were placed into PATH



Dijkstra's LSR Algorithm

- We can now create a forwarding database:

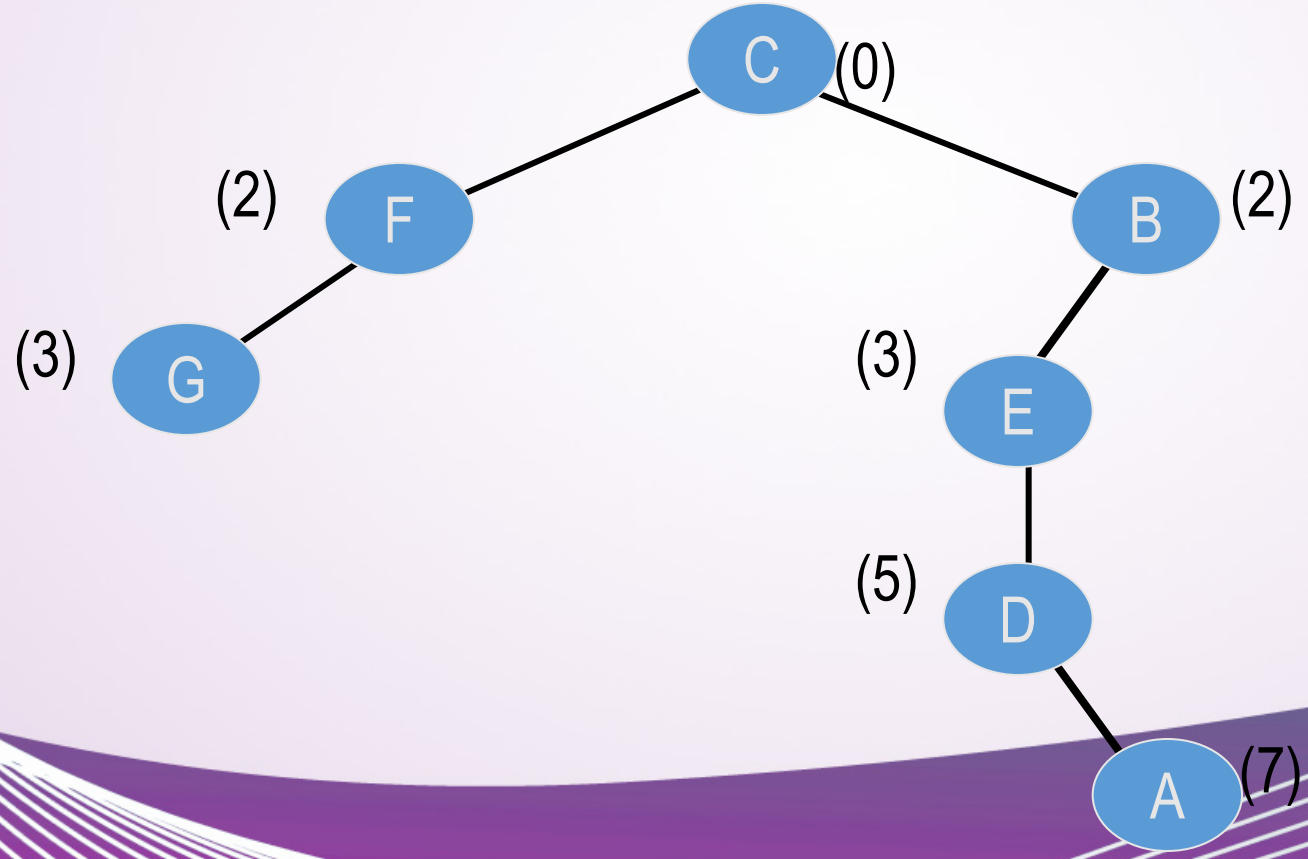


Forwarding Database	
Destination	**Port**
C	C
F	F
G	F
B	B
E	B
D	B
A	B

- LSR forwarding tables must be recalculated whenever a topology change occurs
  - For example, a new router and/or link is added to the network
    - This new link may provide a more efficient route to one or more other nodes
  - For example, a given link's cost is reduced
    - This new link may now provide the lowest total cost route to a destination that was previously forwarded in another direction
  - For example, a given link's cost is increased
    - This new link may no longer provide the lowest total cost route to a given destination, and another route should now be chosen

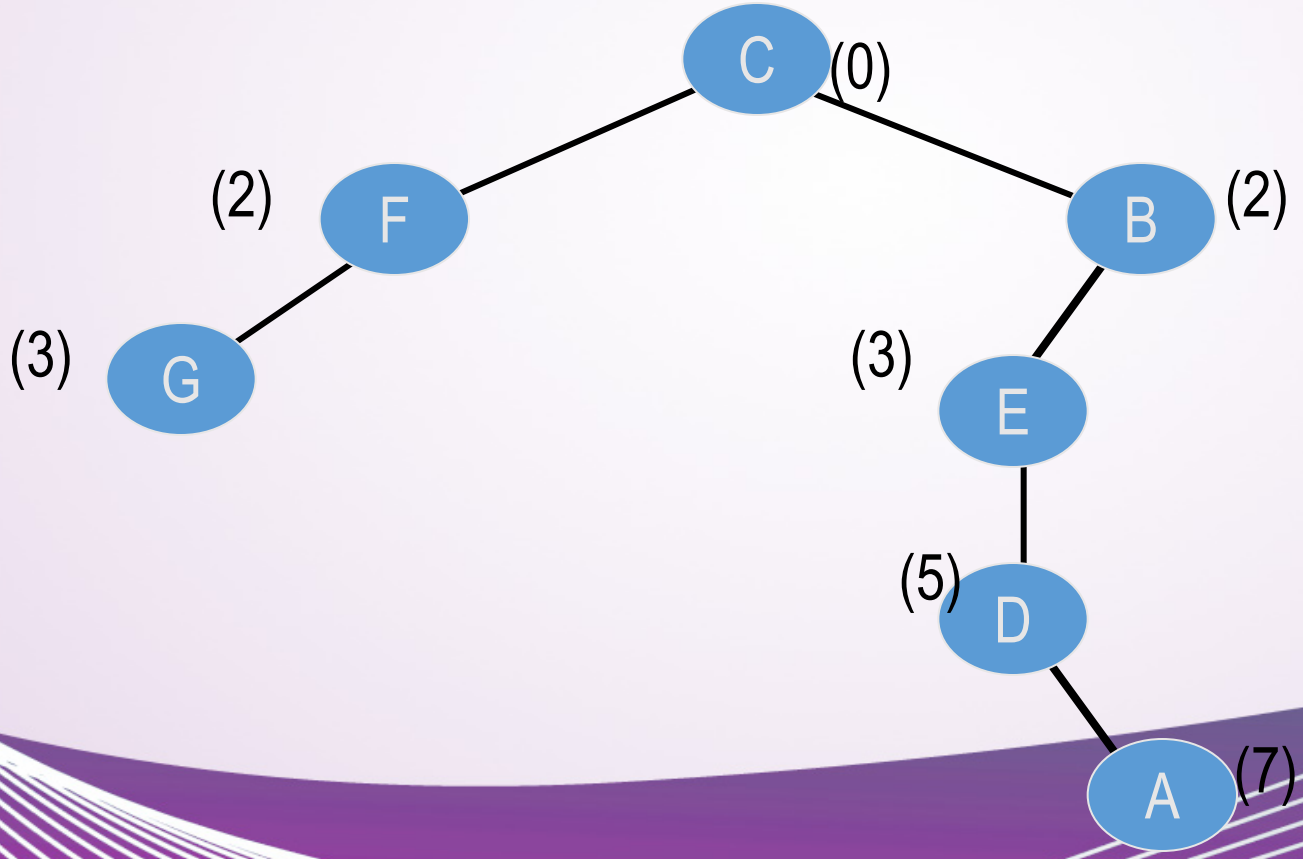
Topology Change Example

- Let’s consider our previously generated PATH structure for the router C



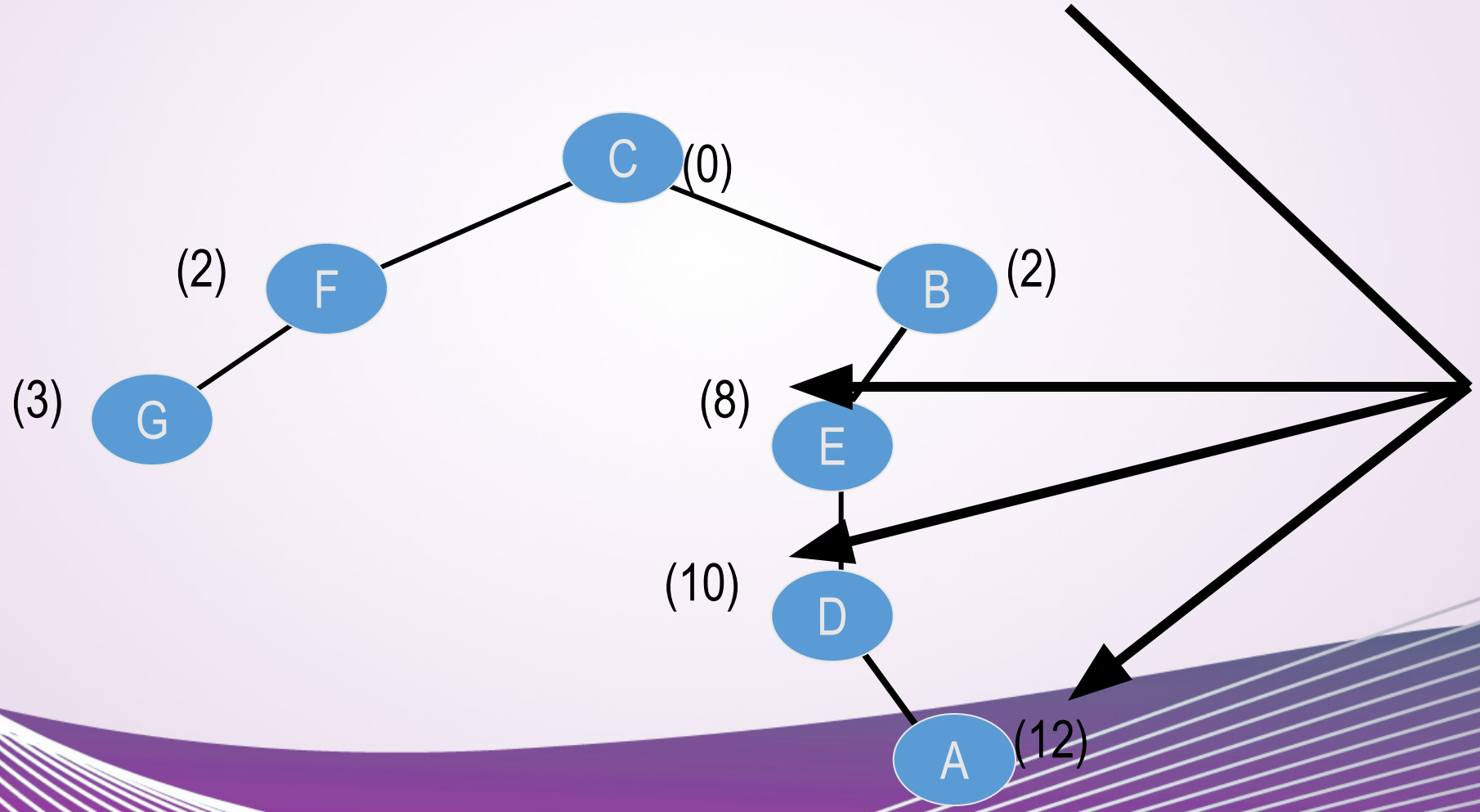
Topology Change Example

- Say we receive an LSP from router B, indicating the link cost from B to E is now 6



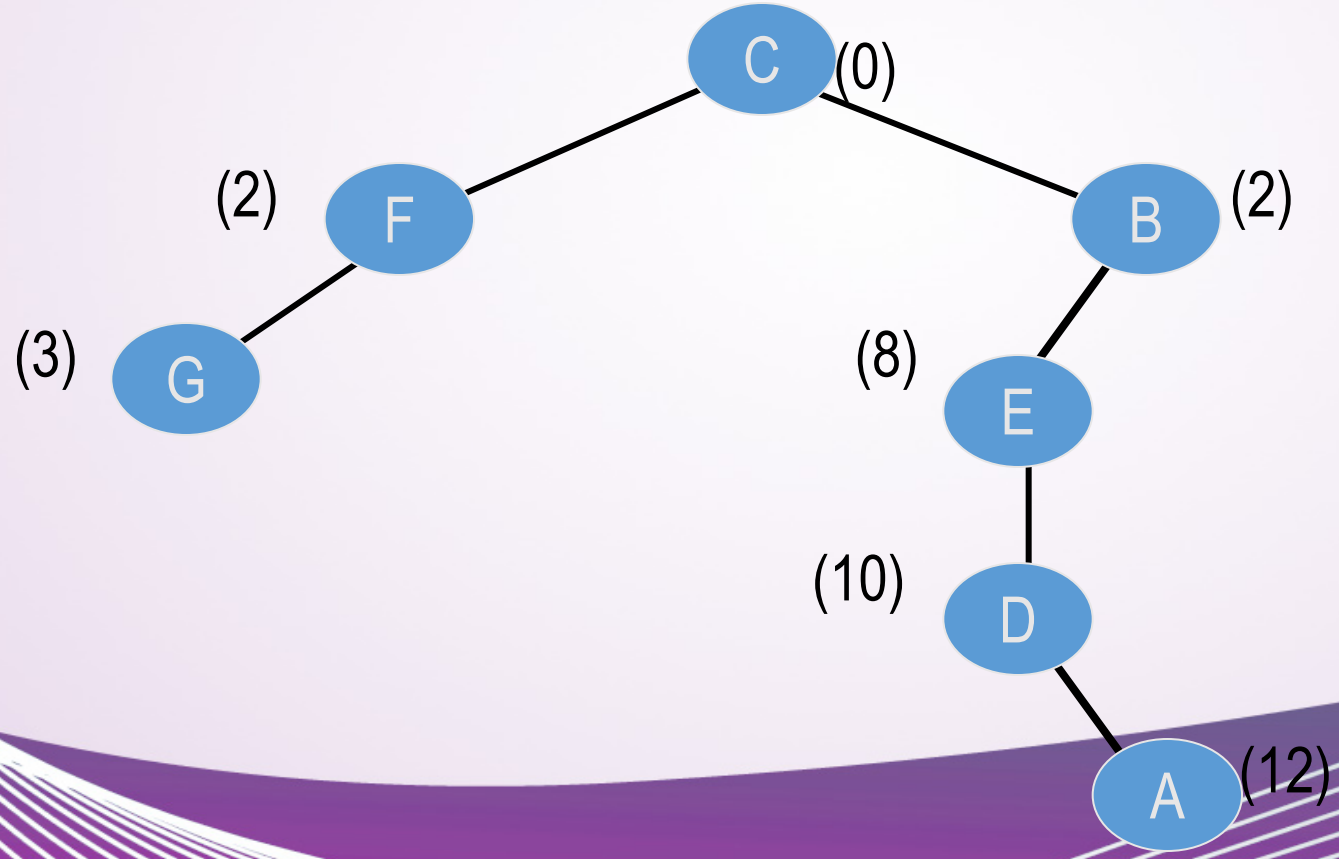
Topology Change Example

- The total route costs are different in PATH:



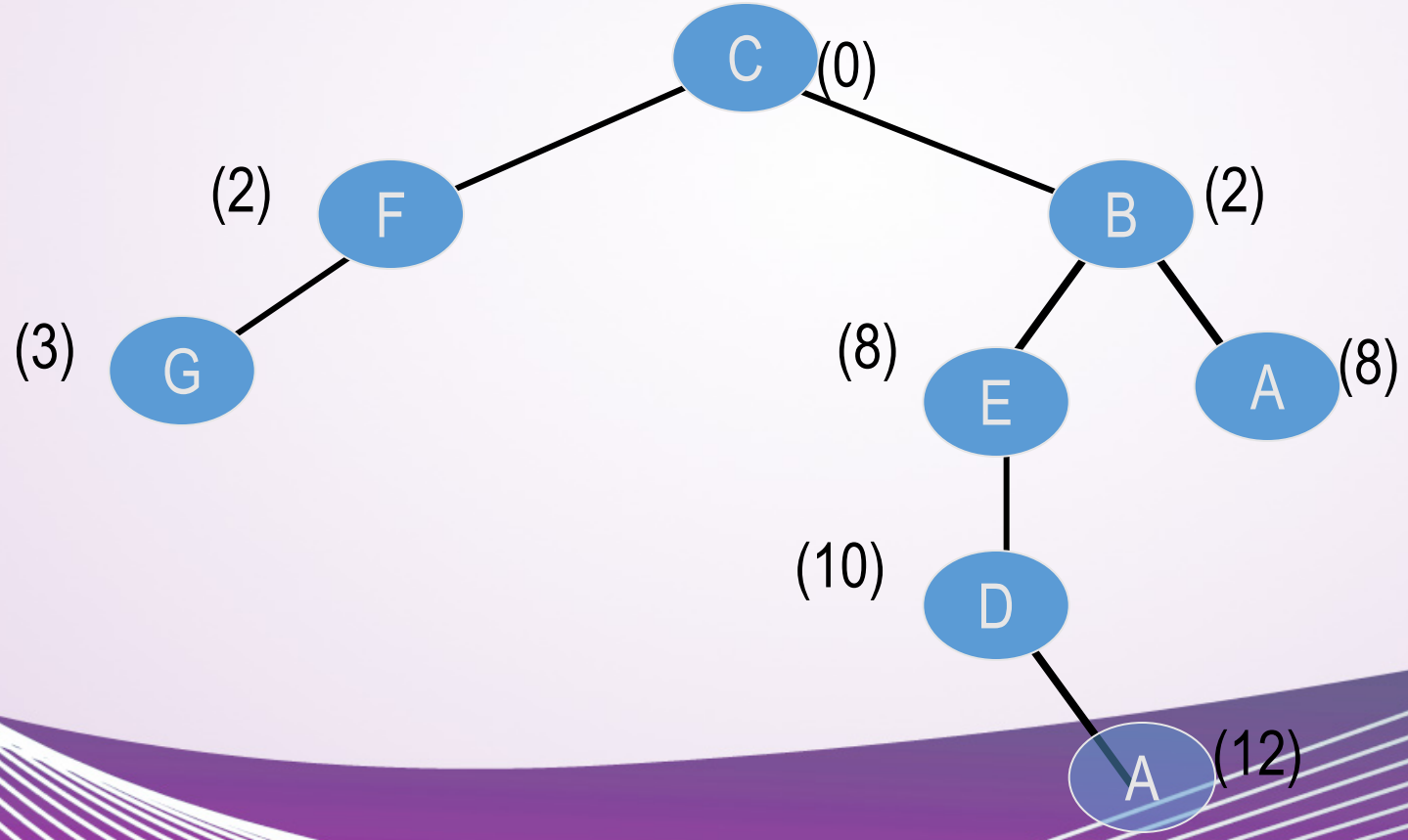
Topology Change Example

- Consider for now, only the cost to A



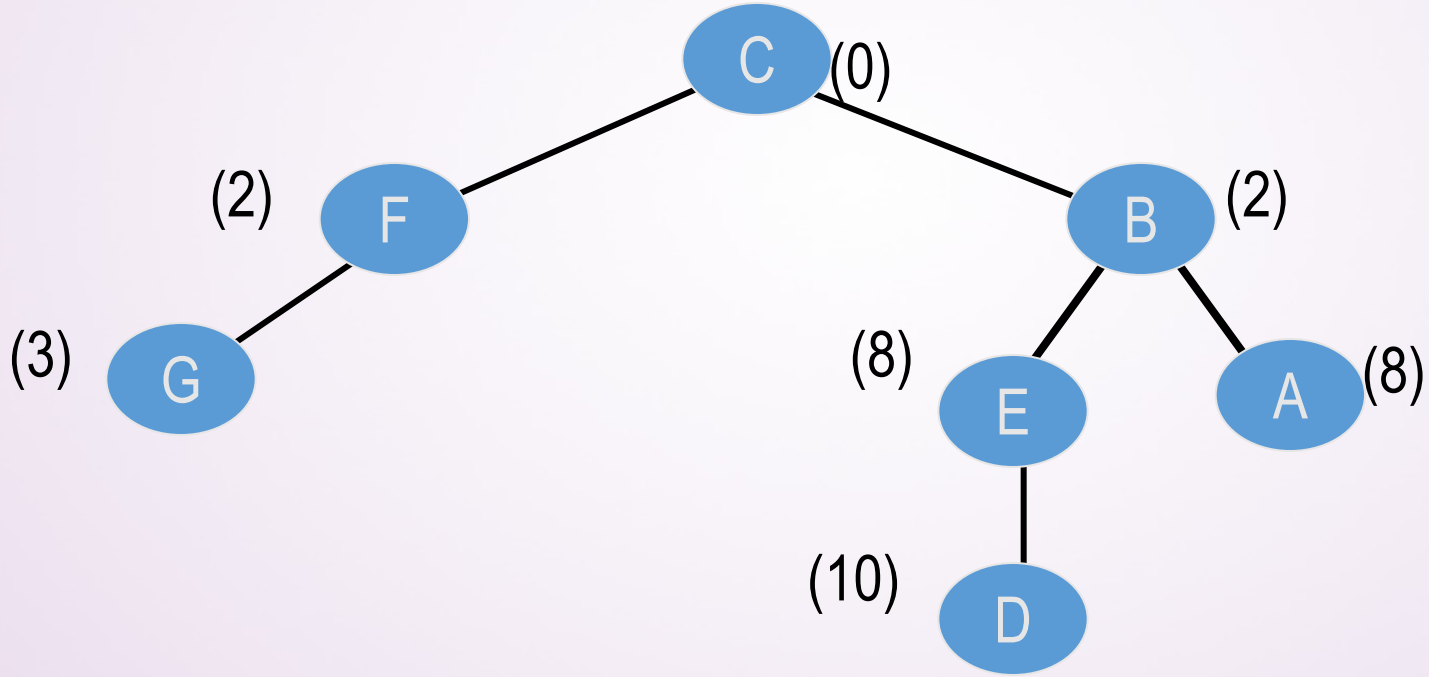
Topology Change Example

- Recall that another path to A existed
- Now, that path is more efficient



Topology Change Example

- The PATH data structure is complete, the forwarding table can now be regenerated



Distance Vector Routing	Link State Routing
--> Bandwidth required is less due to local sharing, small packets and no flooding.	--> Bandwidth required is more due to flooding and sending of large link state packets.
--> Based on local knowledge since it updates table based on information from neighbors.	--> Based on global knowledge i.e. it have knowledge about entire network.
--> Make use of Bellman Ford algo	--> Make use of Dijkastra's algo
--> Traffic is less	--> Traffic is more
--> Converges slowly i.e. good news spread fast and bad news spread slowly.	--> Converges faster.
--> Count to infinity problem.	--> No count to infinity problem.
--> Persistent looping problem i.e. loop will there forever.	--> No persistent loops, only transient loops.
--> Practical implementation is RIP and IGRP.	--> Practical implementation is OSPF and ISIS.

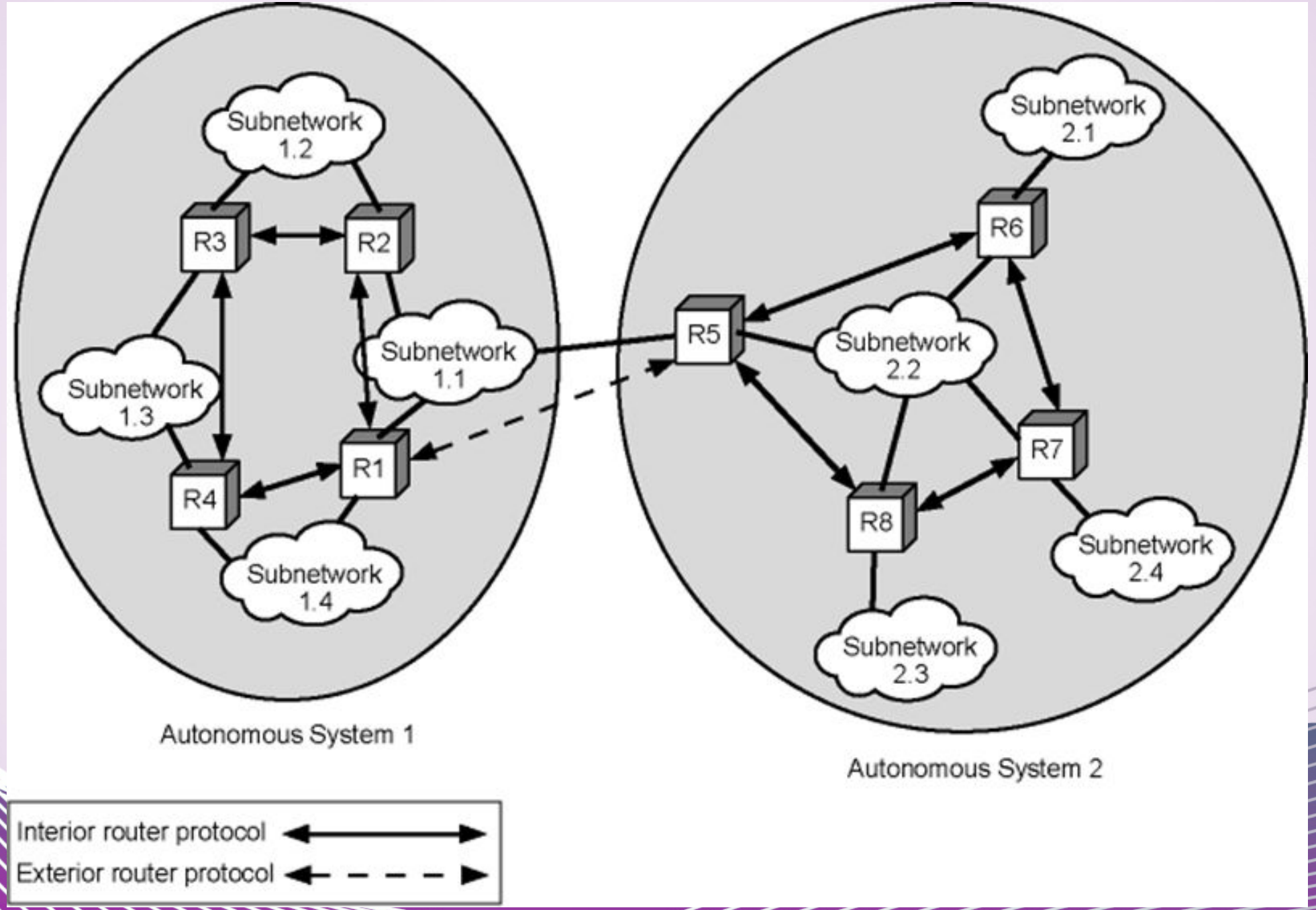
Infinity count problem

Autonomous Systems (AS)

- AS is defined as set of routers and networks managed by single organization (e.g. an ISP – Internet Service Provider)
  - Exchange routing information in itself
  - Common routing protocol
- An AS must be connected in itself
  - There is at least one route between any pair of nodes and networks

Interior Routing Protocol (IRP)
Exterior Routing Protocol (ERP)

- IRP passes routing information between routers within AS
  - Need exchange of info among the routers only in AS
  - Different autonomous systems may have different IRP mechanisms
  - Intra domain
- ERP
  - Autonomous systems need to talk to each other
  - A few routers in each AS must talk with other AS
  - Use Exterior Routing Protocol (ERP)
  - Inter domain


Subnetwork 1.2
R3
R2
Subnetwork 1.1
Subnetwork 1.3
R1
R4
Subnetwork 1.4
Autonomous System 1
R5
Subnetwork 2.1
R6
Subnetwork 2.2
R7
R8
Subnetwork 2.4
Subnetwork 2.3
Autonomous System 2
Interior router protocol
Exterior router protocol

- Distance-vector and Link State are suitable for IRP, not ERP
- Several reasons. Some of them:
  - Both require homogenous metrics that may be the case within an AS, but we cannot assume then same for different AS systems
  - Flooding the link state information across multiple AS systems is not scalable

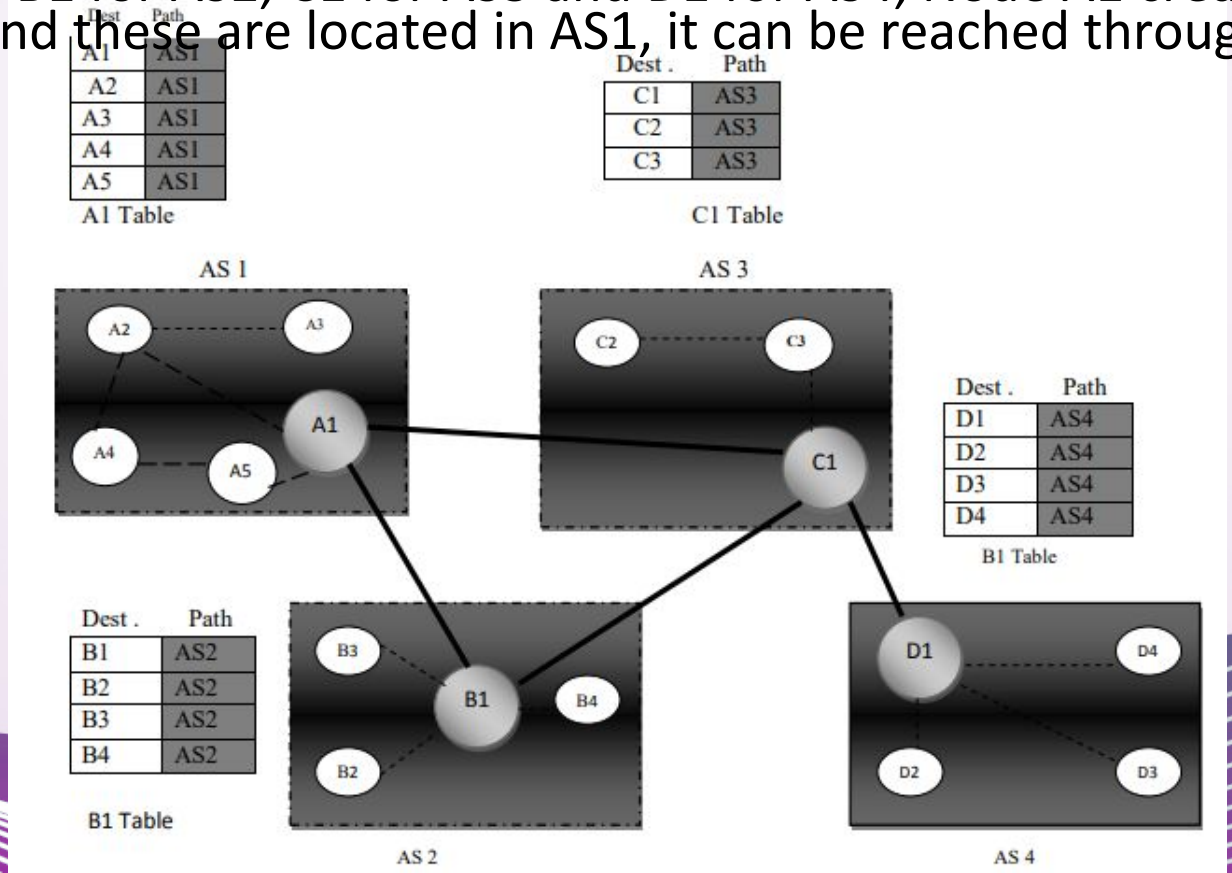
Path Vector Routing

- Suitable approach for Exterior Router Protocols (interdomain)
- Provide information about which networks can be reached by a given router and Autonomous Systems crossed to get there
  - Does not include distance or cost estimate
- Path-vector routing
  - Faster loop detection than distance-vector routing
  - More flexibility than shortest-path routing
- Extension of distance-vector routing
  - Support flexible routing policies
  - Avoid count-to-infinity problem

- It assumes that there is one node in each autonomous system that acts on behalf of the entire autonomous system is called Speaker node .
- The speaker node in an AS creates a routing cable and advertises to the speaker node in the neighbouring ASs
- A speaker node advertises the path, not the metrics of the nodes, in its autonomous system or other autonomous systems
  - Initialization
  - Sharing
  - Updating

Initialization

It is the initial table for each speaker node in a system made four ASs. Here Node A1 is the speaker node for AS1, B1 for AS2, C1 for AS3 and D1 for AS4, Node A1 creates an initial table that shows A1 to A5 and these are located in AS1, it can be reached through it

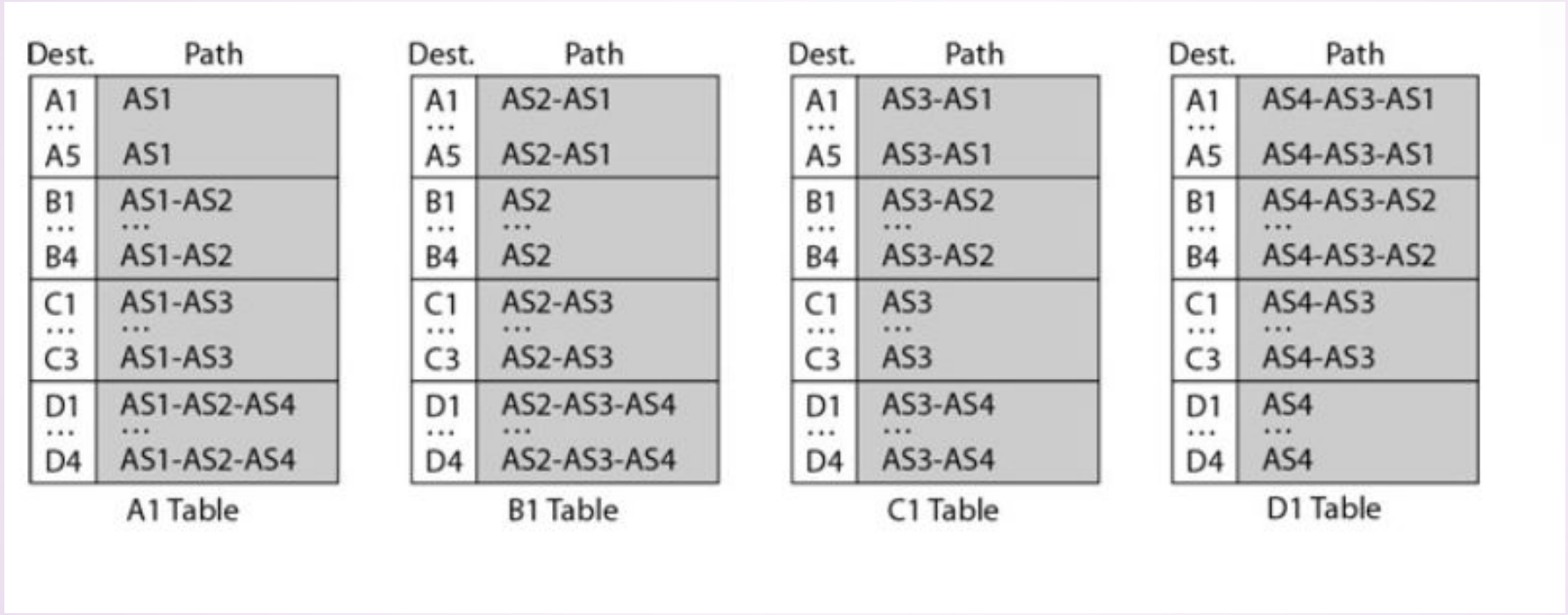


Sharing

A speaker in an autonomous system shares its table with immediate neighbours ,here Node A1 share its table with nodes B1 and C1 , Node C1 share its table with nodes A1,B1 and D1 , Node B1 share its table with nodes A1 and C1 , Node D1 share its table with node C1

Updating

If router A1 receives a packet for nodes A3 , it knows that the path is in AS1,but if it receives a packet for D1,it knows that the packet should go from AS1,to AS2 and then to AS3 ,then the routing table shows that path completely on the other hand if the node D1 in AS4 receives a packet for node A2,it knows it should go through AS4,AS3,and AS1,

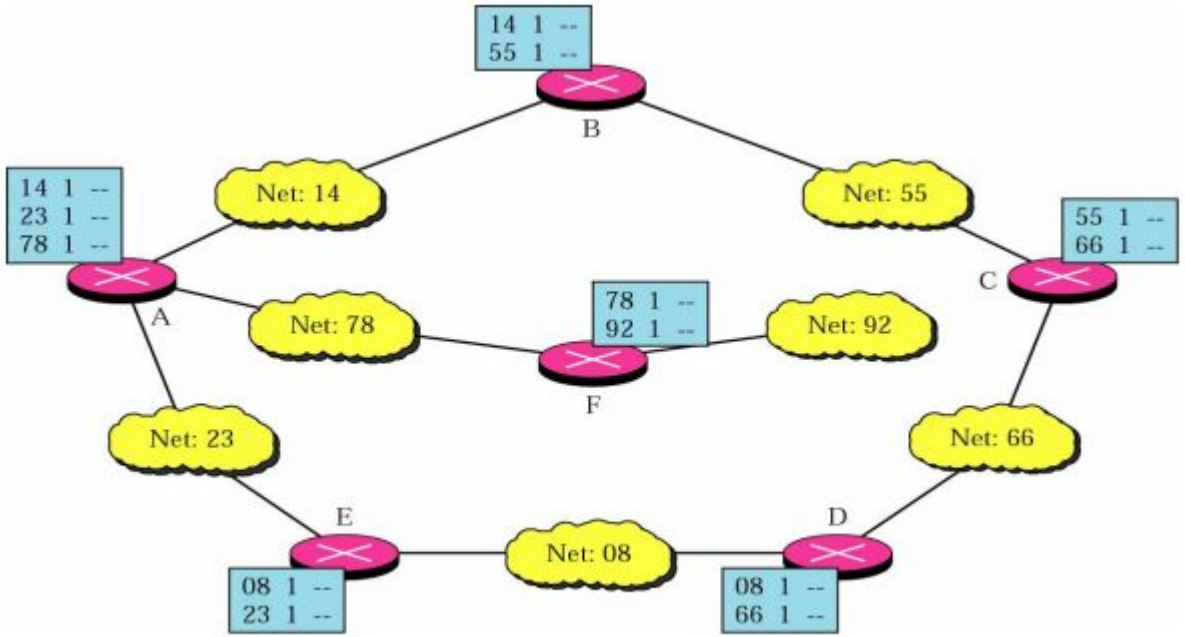

Dest. Path
A1 ... A5 AS1 AS1
B1 ... B4 AS1-AS2 ... AS1-AS2
C1 ... C3 AS1-AS3 ... AS1-AS3
D1 ... D4 AS1-AS2-AS4 ... AS1-AS2-AS4
A1 Table
Dest. Path
A1 ... A5 AS2-AS1 AS2-AS1
B1 ... B4 AS2 ... AS2
C1 ... C3 AS2-AS3 ... AS2-AS3
D1 ... D4 AS2-AS3-AS4 ... AS2-AS3-AS4
B1 Table
Dest. Path
A1 ... A5 AS3-AS1 AS3-AS1
B1 ... B4 AS3-AS2 ... AS3-AS2
C1 ... C3 AS3 ... AS3
D1 ... D4 AS3-AS4 ... AS3-AS4
C1 Table
Dest. Path
A1 ... A5 AS4-AS3-AS1 AS4-AS3-AS1
B1 ... B4 AS4-AS3-AS2 ... AS4-AS3-AS2
C1 ... C3 AS4-AS3 ... AS4-AS3
D1 ... D4 AS4 ... AS4
D1 Table

Path Vector

- Prevents loop
- Optimal path - all AS use diff metrics

Routing Protocols

1. Routing Information Protocol (RIP)
   - An intra- domain (interior) routing protocol used inside an autonomous system.
   - It is a very simple protocol based on distance vector routing.
   - RIP implements distance vector routing

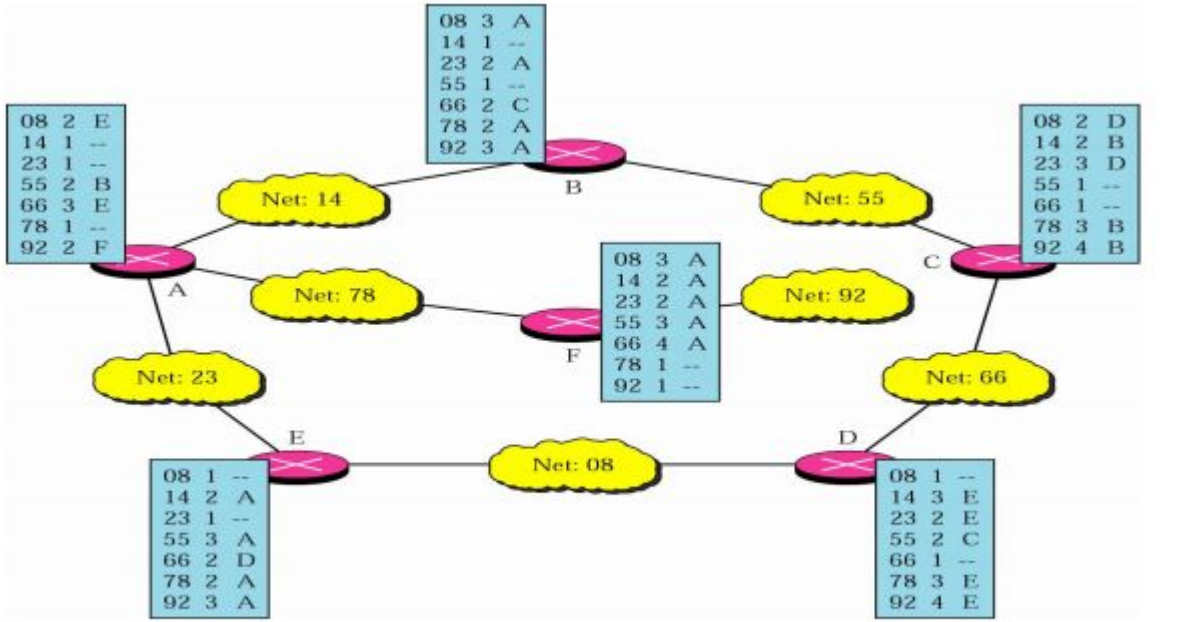


Initial

- is small

Final

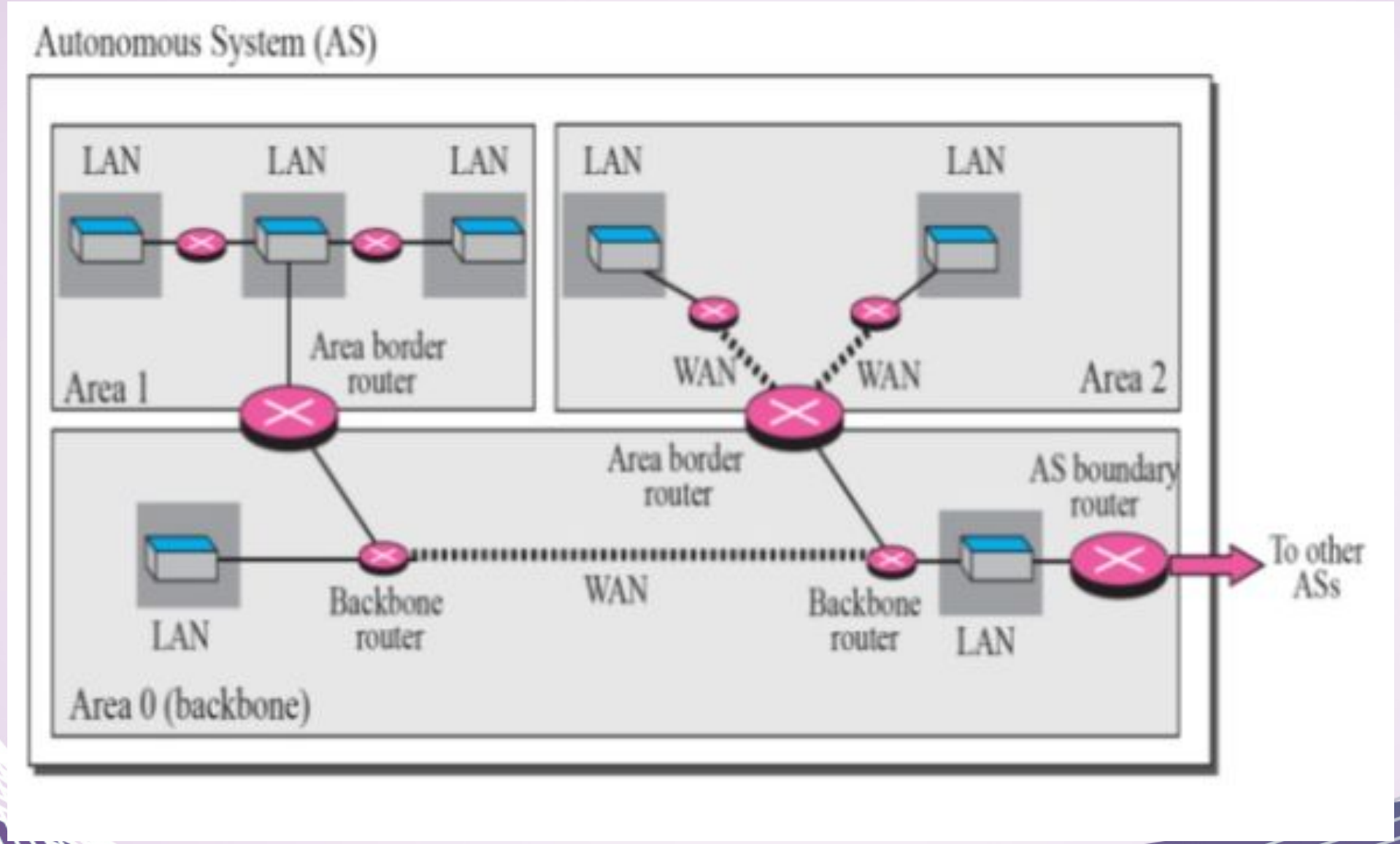
-grows



- The destination, first column defines the network address
- The next node,defines the address of the router to which the packet as to be sent to reach destination
- Metric is hop count
- Infinity is defined as 16, an AS can’t have more than 15 hops

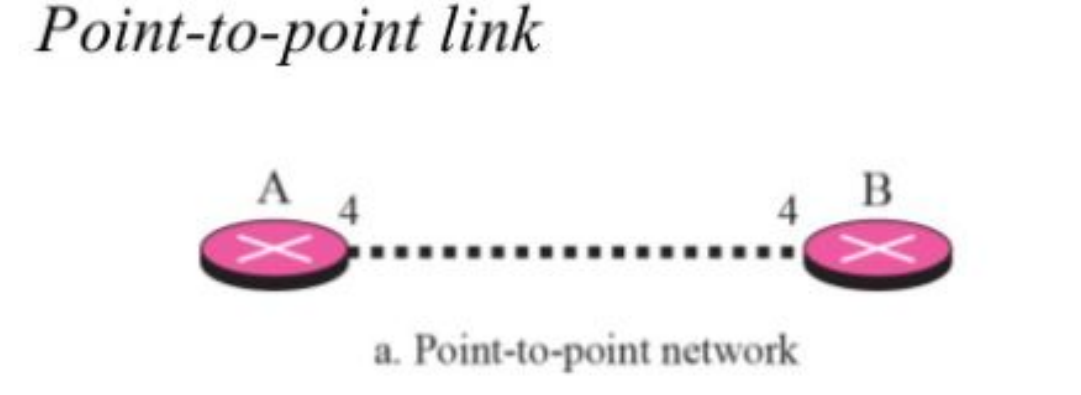
2. Open Shortest Path First (OSPF) protocol

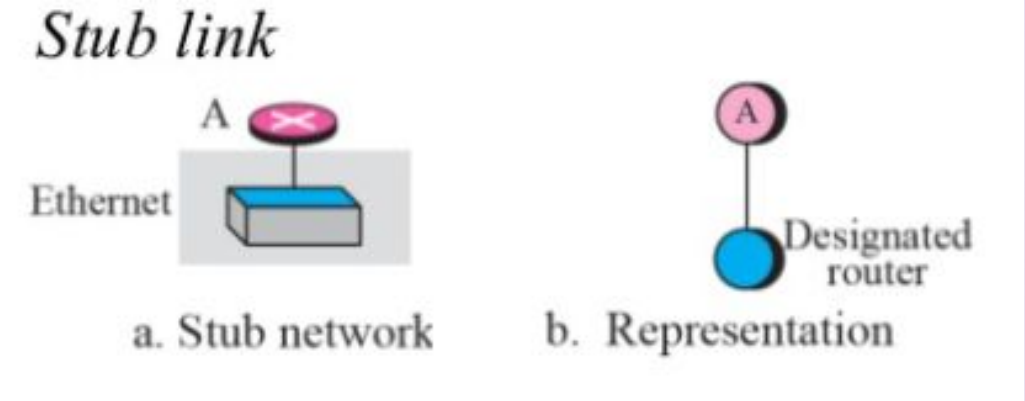
- An intra-domain routing protocol based on **link state routing**.
- Its domain is also an autonomous system.
- Metric used is based on service - min delay, max throughput etc
- Area in OSPF
  - A collection of networks with area ID
  - Area border routers summarize the information about the area and send it to other areas (flooding)
  - **Backbone** area and backbone routers
  - All of the area inside an AS must be connected to the backbone

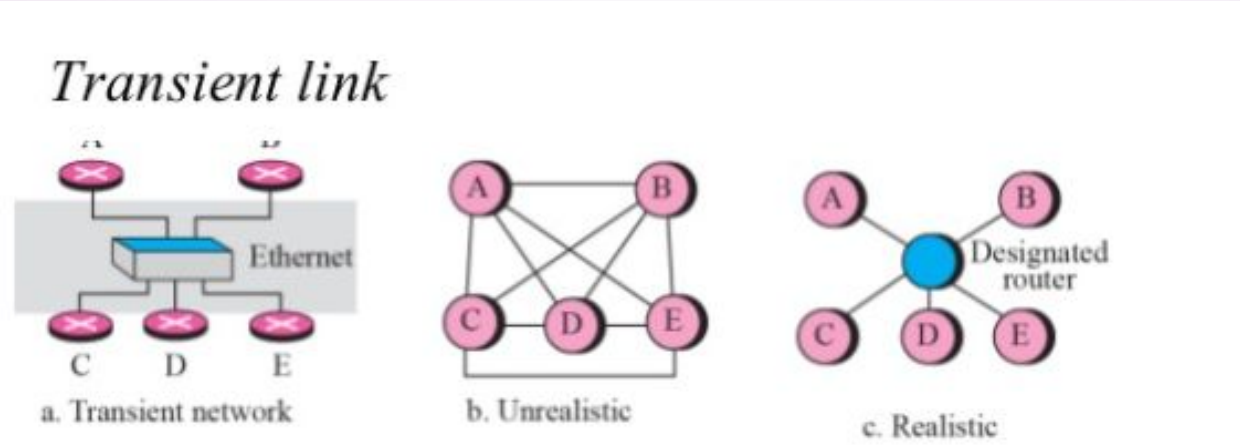

Autonomous System (AS)
LAN
LAN
LAN
LAN
LAN
Area border
router
Area 1
WAN
WAN
Area 2
Area border
router
AS boundary
router
To other
ASs
LAN
Backbone
router
WAN
Backbone
router
LAN
Area 0 (backbone)

m

- Types of links







- Point-to-Point Link

Connect two routers without any other host or router in between

- Transient Link

A network with several routers attached to it

Data can enter through any of the routers and leave through any router

All LANs and some WANs with two or more routers

- Stub Link

A network that is connected to only one router n Data packet enter and leave through this only one router

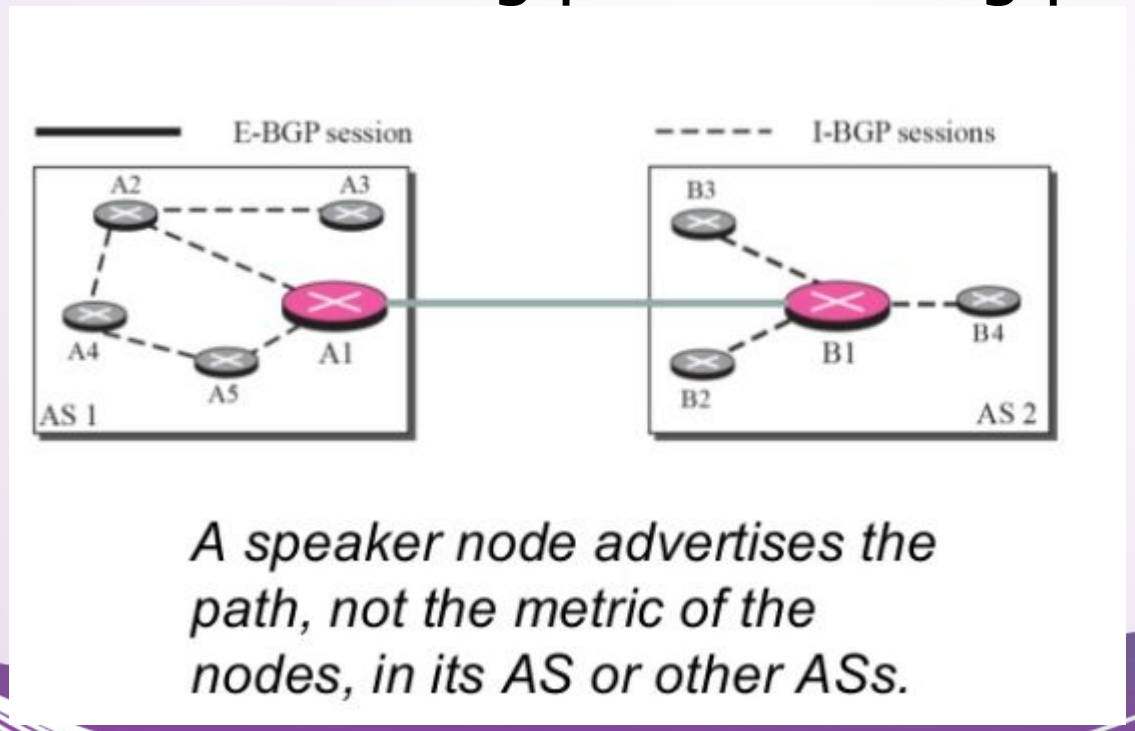
- Virtual Link

When the link between two routers is broken n The administrator may create a virtual path between them using a longer path and may go through several routers

•

BGP Border Gateway Protocol (BGP)

- An interdomain routing protocol using path vector routing



A speaker node advertises the path, not the metric of the nodes, in its AS or other ASs.

Types of AS

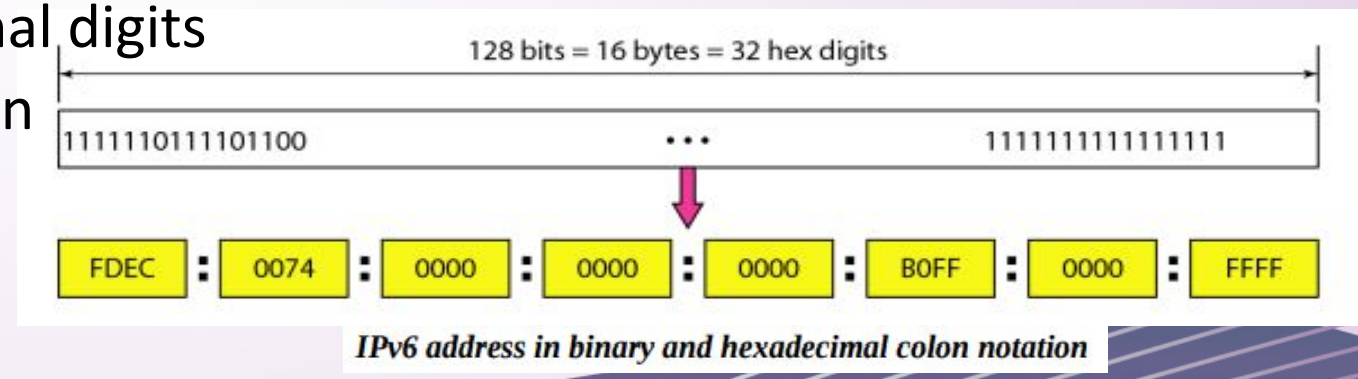
- Stub AS – Only one connection to another AS (only a source or sink for data traffic)
- Multihomed AS – More than one connection to other AS, but it is still only a source or sink for data traffic
- Transit AS – Multihomed AS that also allows transient traffic

IPV6

- The address depletion of IPv4 protocol was one of the major reason for developing IPv6 protocol.
- Despite all short-term solutions, such as classless addressing, DHCP and NAT address depletion is still a long-term problem for the Internet.
- Why IPv6?
  - Address depletion
  - Two level (classfull) addressing is wasteful
  - lack of accommodation for real-time audio and video transmission - MobileIP, IP telephony
  - Security - encryption and authentication of data for some applications

- **Goal of IP v6**
  - IPv6 will also have a goal to make the Internet a more **secure** place for browsers, and with the rapid number of identity theft victims
  - **Efficient and hierarchical addressing** and routing infrastructure- based on the common occurrence of multiple levels of Internet service providers.
  - **Bigger address space** – 128 bits, 16 bytes
  - Efficient and Extensible IP datagram
  - Improved Host and Router Discovery
  - Autoconfiguration (like Plug and play, easy to configure)
  - Multiple addresses per interface
  - Dynamic address configuration
  - Better header format – simple main header
  - New Options – in optional headers
  - Allows future expansion
  - Support for resource allocation, QoS

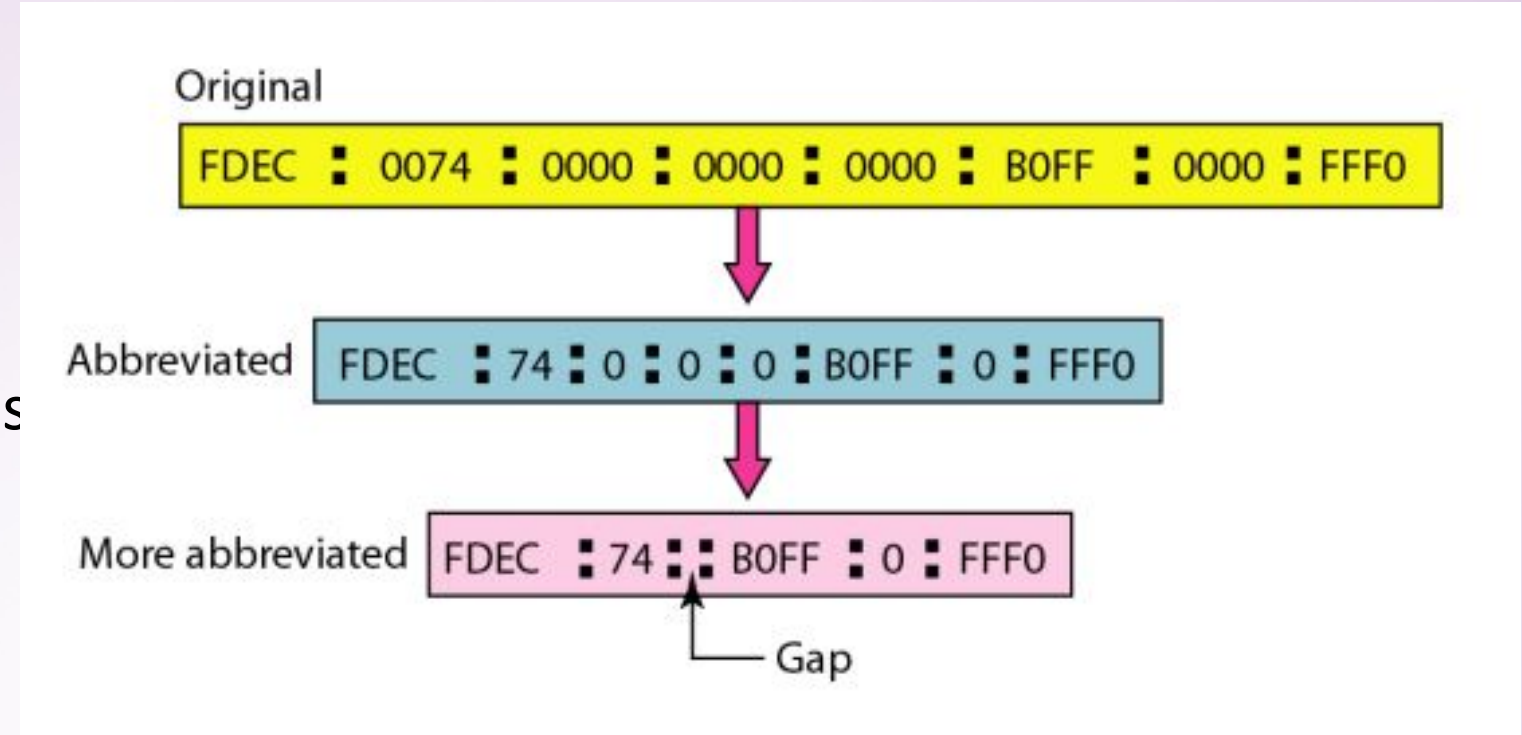
- An IPv6 address is 128 bits long/ 16 bytes (octets)
- Hexadecimal Colon Notation
  - More readable
  - 128 bit is divided into 8 sections ,each bytes in length
  - 2 bytes in hexadecimal notation requires 4 hexadecimal digits
  - Address consists of 32 hexadecimal digits
  - Every 4 digits separated by a colon



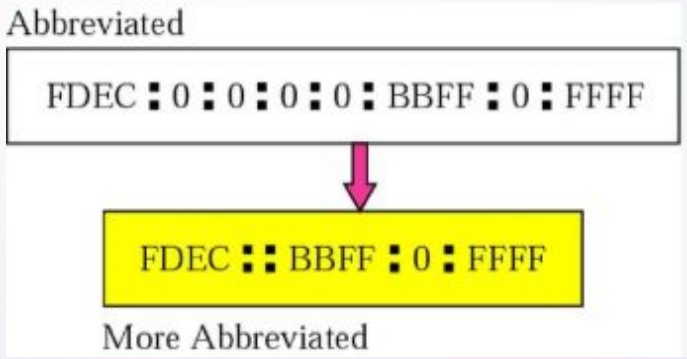
IPv6 address in binary and hexadecimal colon notation

- Pv6 address range from 0000:0000:0000:0000:0000:0000:0000:0000 to ffff: ffff: ffff: ffff: ffff: ffff: ffff: ffff.

- Abbreviated IPv6 addresses



- Abbreviated address with consecutive zeros



- CIDR

FDEC : 0 : 0 : 0 : 0 : BBFF : 0 : FFFF/60

- Note : No more that one leading zeros can be replecd with zeros (see video)

- Expand the address 0:15::1:12:1213 to its original
  - We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

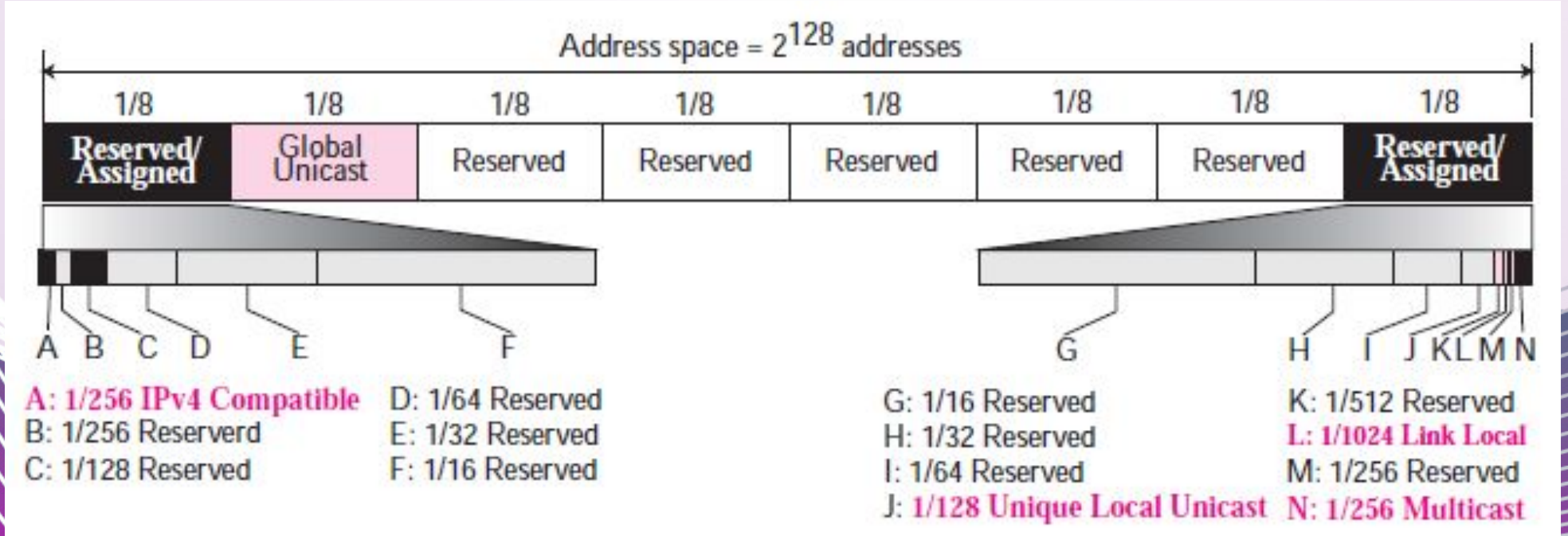
```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
 0:   15:                    :    1:   12:1213
```

  - original address is

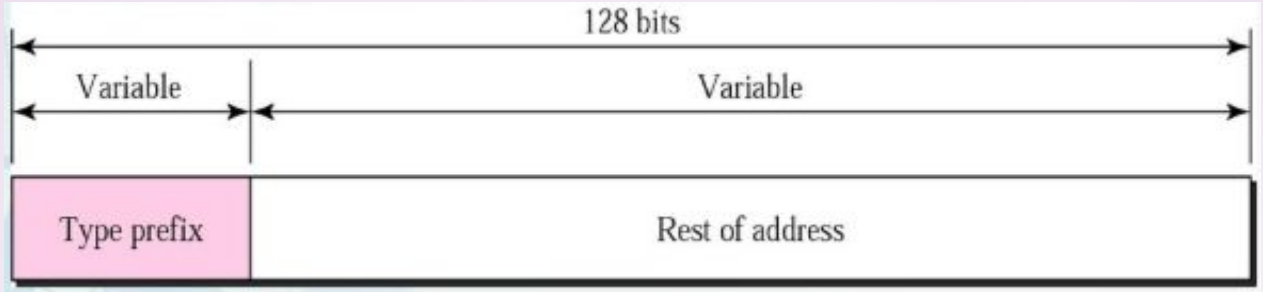
```
0000:0015:0000:0000:0000:0001:0012:1213
```

  - you can assign more than one IPv6 address to an interface!

- **ADDRESS SPACE ALLOCATION**
  - Like the address space of IPv4, IPv6 is divided into several blocks of varying size and each block is allocated for special purpose.
  - Most of the blocks are still unassigned and have been left aside for future use.
  - the whole address space is divided into eight equal ranges.
  - Only 3/8 parts are in use (either Reserved or Assigned or Global Unicast) and rest 5/8 are unused yet or future use



- IPv6 Prefix Length is used to identify how many bits of a Global Unicast IPv6 Address (which is same as Unicast address mentioned previously) are there in network part.
- For example, in 2001:0DB8:0000:000b::/64, the number 64 is used to identify that the first 64 bits are in network part.
- A few leftmost bits, called the type ***prefix*** in each address define its category (see table in next slide)
- Note that the network prefix shown above, 2001:0DB8:0000:000b::/64, includes the 48 bit IPv6 global routing prefix 2001:0DB8:0000::/48 and the next 16 bits "000b" are used for internal subnetting within an organization.

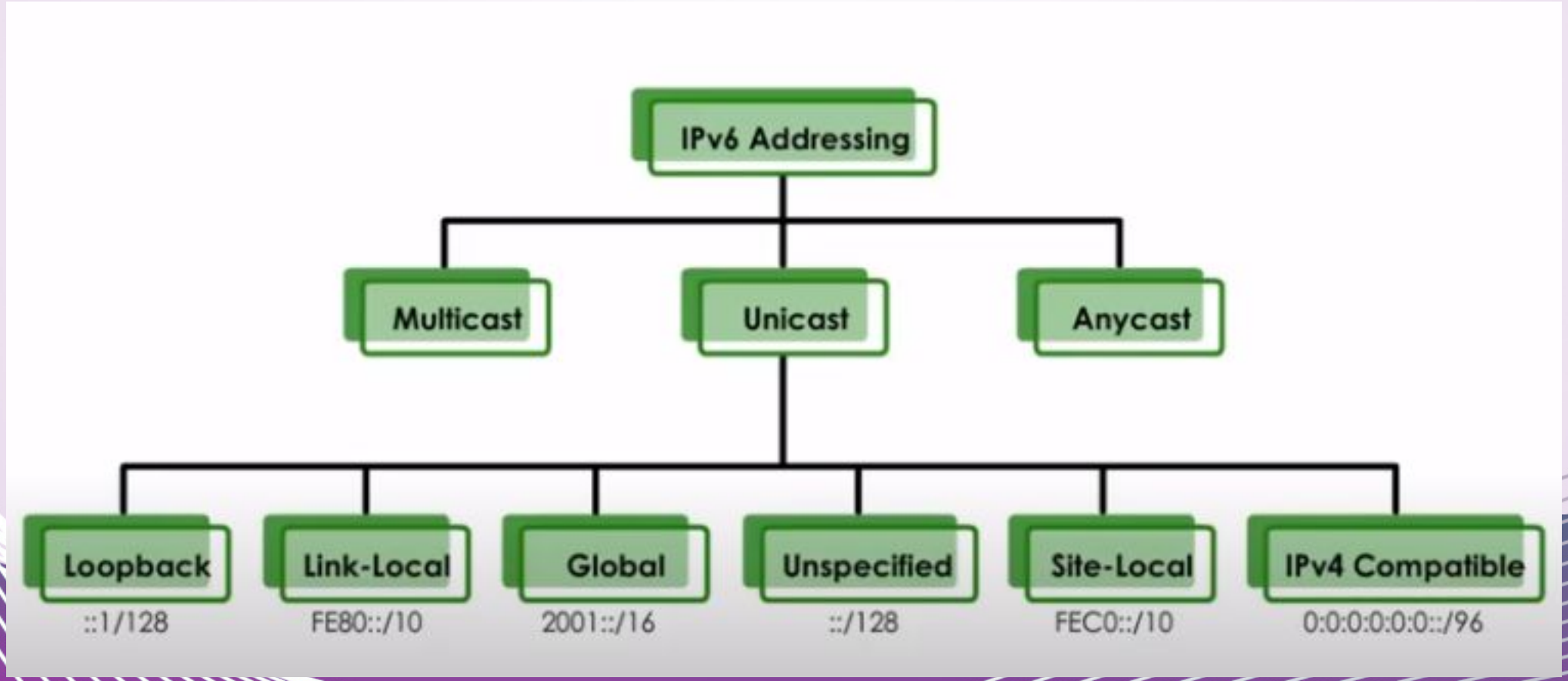


- The leftmost fields of the IPv6 address along with the network bits length represented in **CIDR format** is known as the prefix/ ***network prefix***.
- The prefixes in IPv6 can be considered similar to the **subnet mask** used in **IPv4 addresses**.
- In IPv6, we use a notation similar to **CIDR** mask , using an integer between 1-128 to represent the network bits
- IPv6 address 2001:0DB8:0000:000b:0000:0000:0000:001A/64 OR 2001:0DB8:0000:000b::/64
- represents the network prefix and the possible IPv6 addresses ranges from 2001:0DB8:0000:000b:0000:0000:0000:0001/64 to 2001:0DB8:0000:000b:ffff:ffff:ffff:ffff/64.

Prefixes for IPv6 Addresses

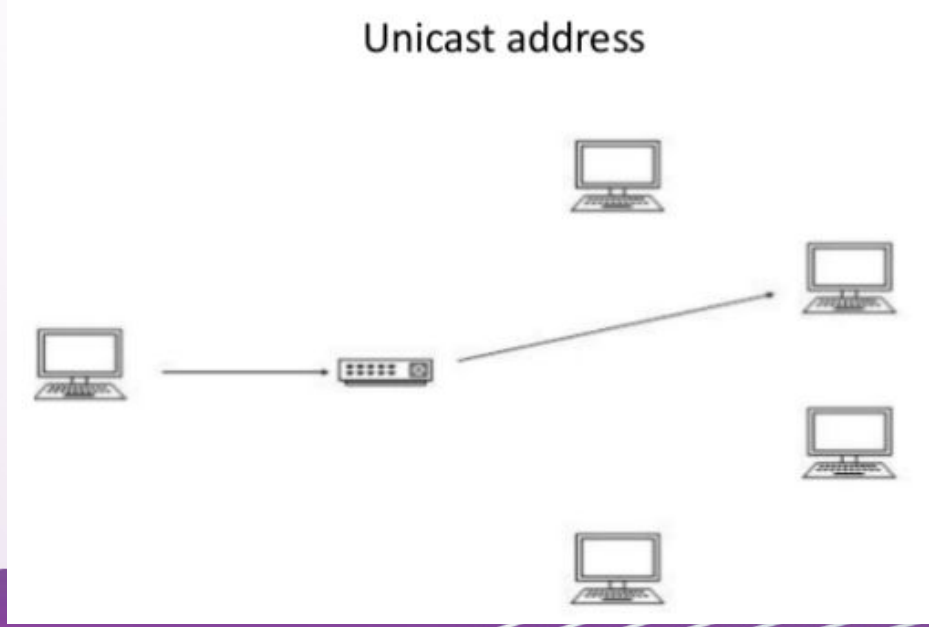
	Block Prefix	*CIDR*	*Block Assignment*	*Fraction*
1	0000 0000	0000::/8	Reserved (IPv4 compatible)	1/256
	0000 0001	0100::/8	Reserved	1/256
	0000 001	0200::/7	Reserved	1/128
	0000 01	0400::/6	Reserved	1/64
	0000 1	0800::/5	Reserved	1/32
	0001	1000::/4	Reserved	1/16
2	001	2000::/3	Global unicast	1/8
3	010	4000::/3	Reserved	1/8
4	011	6000::/3	Reserved	1/8
5	100	8000::/3	Reserved	1/8
6	101	A000::/3	Reserved	1/8
7	110	C000::/3	Reserved	1/8
8	1110	E000::/4	Reserved	1/16
	1111 0	F000::/5	Reserved	1/32
	1111 10	F800::/6	Reserved	1/64
	1111 110	FC00::/7	Unique local unicast	1/128
	1111 1110 0	FE00::/9	Reserved	1/512
	1111 1110 10	FE80::/10	Link local addresses	1/1024
	1111 1110 11	FEC0::/10	Reserved	1/1024
	1111 1111	FF00::/8	Multicast addresses	1/256

- More on types of IPV6 address



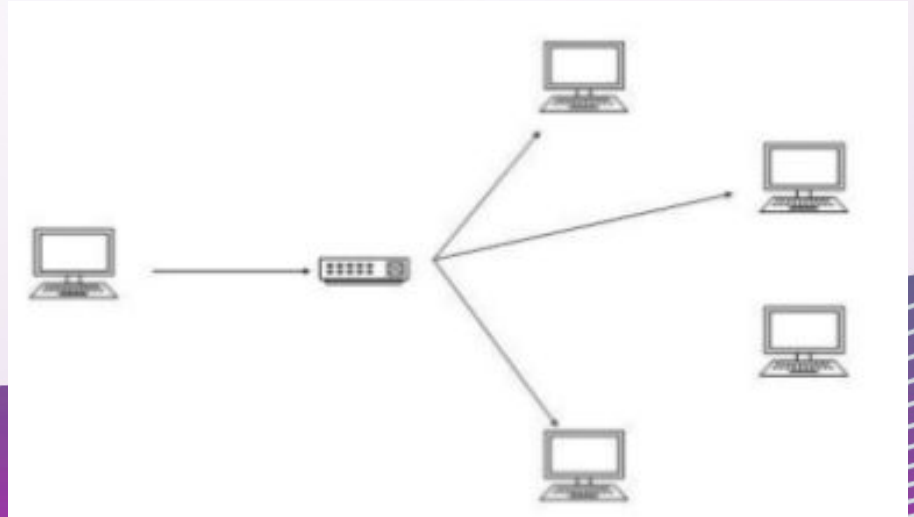
- **Address Structure**
  - There are three types of IPv6 category :

1. ***Unicast***
   - Unicast transmission is the sending of messages to a single network destination identified by a unique address.
   - A unicast address defines a single computer.
   - The packet sent to a unicast address must be delivered to that specific computer.
   - Unicast is one to one



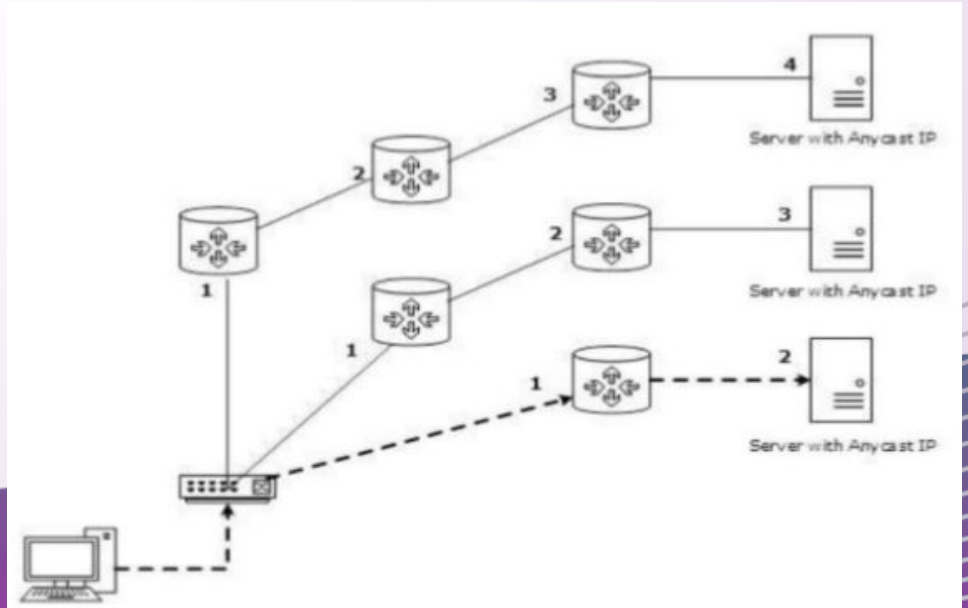
2. *Multicast*

- Ipv6 does not have a broadcasting domain
- Multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source.
- Multicasting is one to many
- Multicast addresses start with FF



3. *Anycast*

- Anycast is a network addressing and routing methodology in which datagrams from a single sender are routed to the topologically nearest node in a group of potential receiver, though it may be sent to a several nodes, all identified by the same destination address.
- Anycast is one to nearest association (router)
- Theses are assigned to routers
- Many routers can have the same anycast address



ICMPv6 (Internet Control Message Protocol version 6)

- Takes care of diagnostic functions, error and data message (same as IPv4).
- The ICMP,ARP, and IGMP protocols in version 4 are combined into one single protocol, ICMPv6.

- ICMPv6 Neighbour Discovery Protocol helps find neighbouring links and routes

DHCPv6

- DHCPv6 is the version of the Dynamic Host Configuration Protocol (DHCP) for Internet Protocol Version 6 (IPv6) networks.
- In addition to stateless address autoconfiguration in IPv6, DHCPv6 provides an alternate solution to assign addresses, name servers and other configuration information in a manner similar to DHCP for IPv4
- Like DHCP IPv4, they are not require to
  - assign IP address, as it is autoconfigured
  - Find DNS, as that is done by ICMPv6

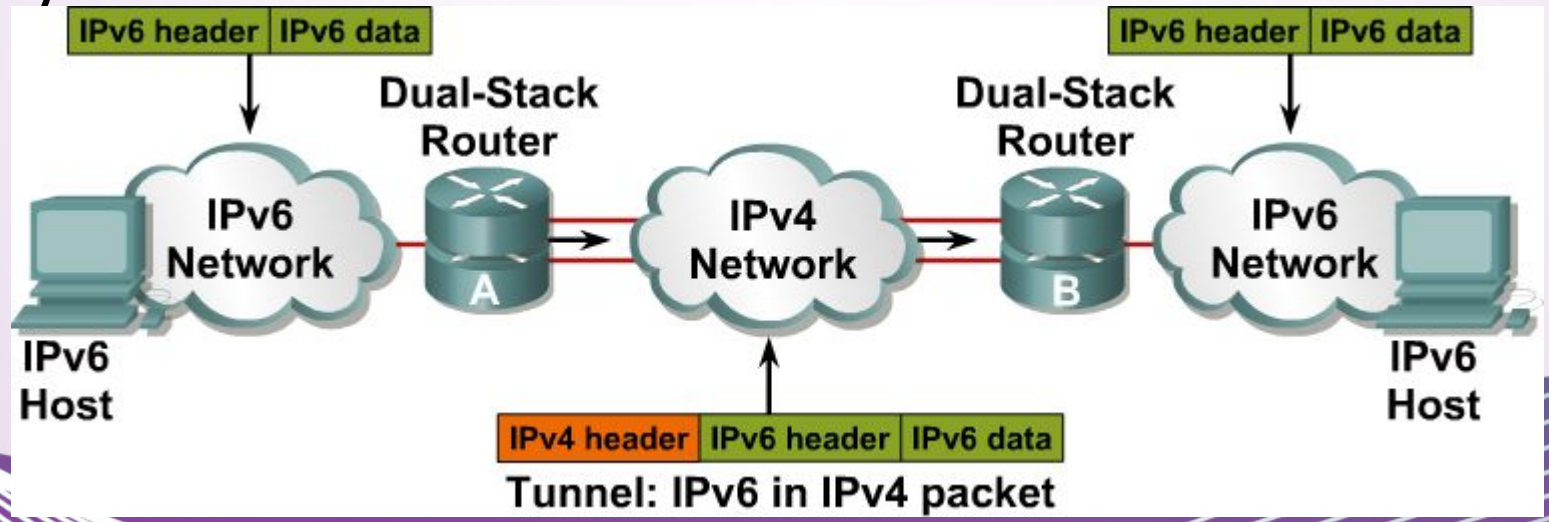
DNS

- No restructuring
-

Transition from IPv4 to IPv6

- Dual Stack
  - This allows all the end hosts and intermediate network devices (like routers, switches, modems etc.) to have both IPv4 and IPv6 addresses and protocol stack.
  - If both the end stations support IPv6, they can communicate using IPv6; otherwise they will communicate using IPv4.
  - This will allow both IPv4 and IPv6 to coexist and slow transition from IPv4 to IPv6 can happen.

- Tunnelling
  - This allows encapsulating IPv6 packets in IPv4 packets for transport over IPv4 only network.
  - This will allow IPv6 only end stations to communicate over IPv4 only networks.



IPv6 and IPv4 compared

IPV4	IPV6
Addresses are 32 bits (4 bytes) in length.	Addresses are 128 bits (16 bytes) in length
Address (A) resource records in DNS to map host names to IPv4 addresses.	Address (AAAA) resource records in DNS to map host names to IPv6 addresses.
Pointer(PTR)resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names.	Pointer(PTR)resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names.
IPSec is optional and should be supported externally	IPSec support is not optional
Header does not identify packet flow or QoS handling by routers.	Header contains Flow Label field, which Identifies packet flow for QoS handling by router.
Both routers and the sending host fragment packets.	Routers do not support packet fragmentation. Sending host fragments packets
Header includes a checksum.	Header does not include a checksum.
Header includes options.	IPV6 Optional data is supported as extension headers.
ARP uses broadcast ARP request to resolve IP to MAC/Hardware address	Multicast Neighbor Solicitation messages resolve IP addresses to MAC addresses.
Internet Group Management Protocol (IGMP) manages membership in local subnet groups.	Multicast Listener Discovery (MLD) messages manage membership in local subnet groups.
Broadcast addresses are used to send traffic to all nodes on a subnet.	IPv6 uses a link-local scope all-nodes multicast address.