

# Computer Networks(2015 Pattern)

## Unit IV- Network Layer

**By Prof. A.R.Jain**

**PVG's COE,Nashik**

**Note: Material for this presentations are taken from Internet and books and only being used for student reference**

8/3/2017

# Network Layer Design Issues

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

# Outline

## Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

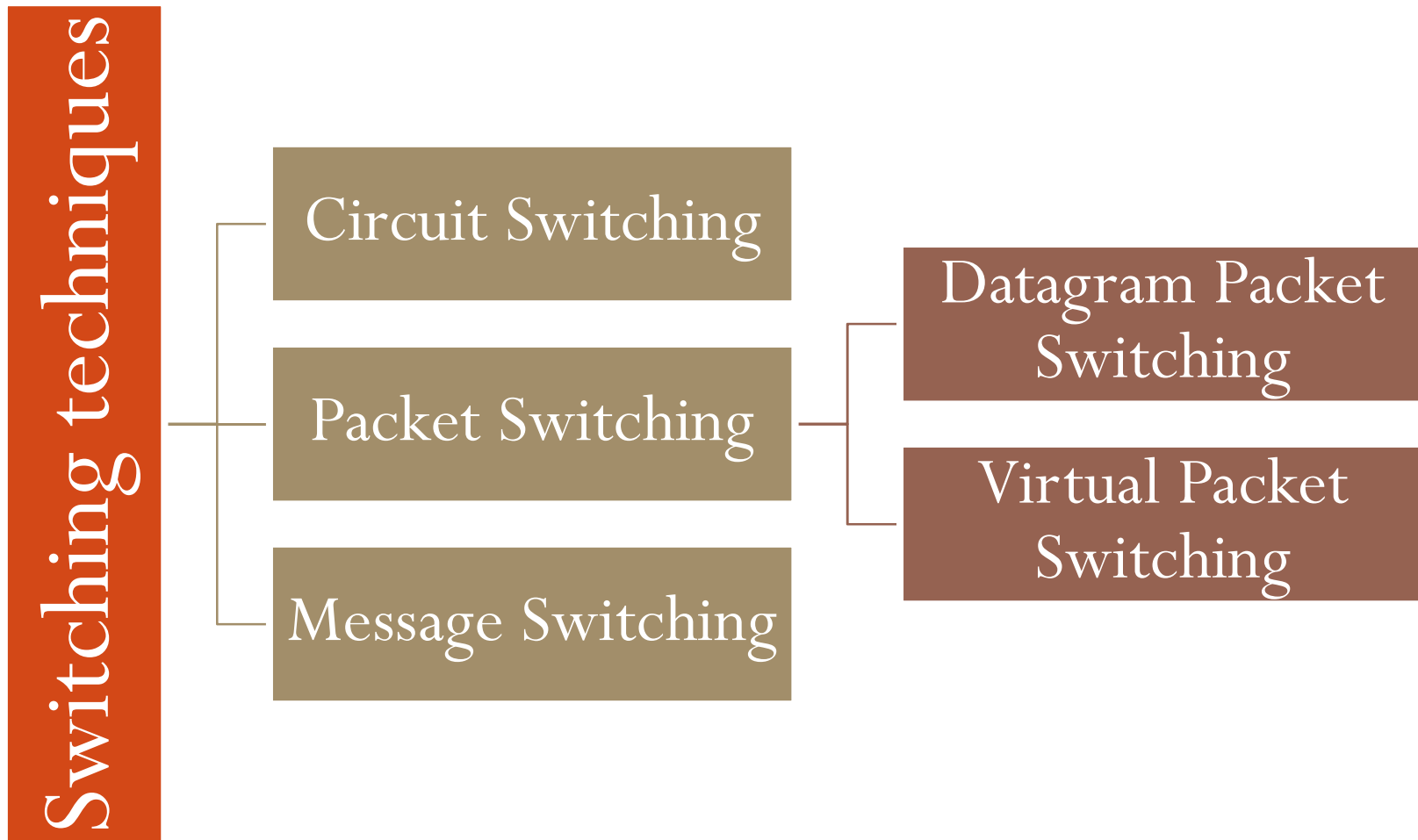
Congestion control and QoS,

MPLS,

Mobile IP,

Routing in MANET : AODV, DSR

# Switching techniques





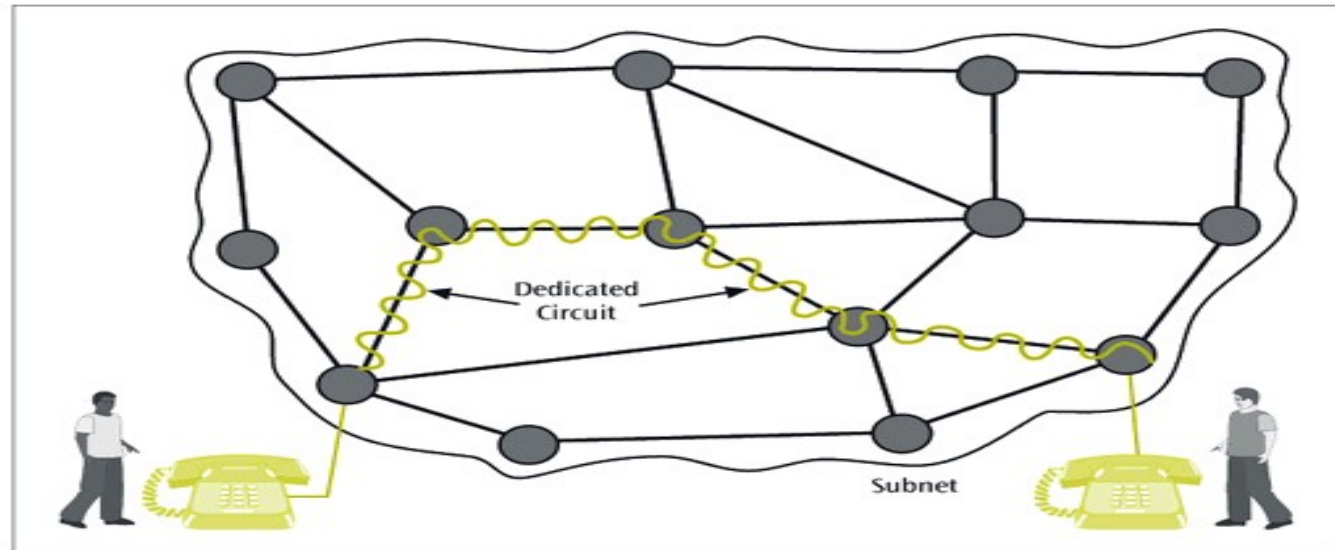
# Circuit Switching

- When two nodes communicate with each other over a dedicated communication path, it is called circuit switching.
- There 'is a need of pre-specified route from which data will travels and no other data is permitted.
- In circuit switching, to transfer the data, circuit must be established so that the data transfer can take place.
- Circuits can be permanent or temporary. Applications which use circuit switching may have to go through **three phases**:
  - Establish a circuit
  - Transfer the data
  - Disconnect the circuit

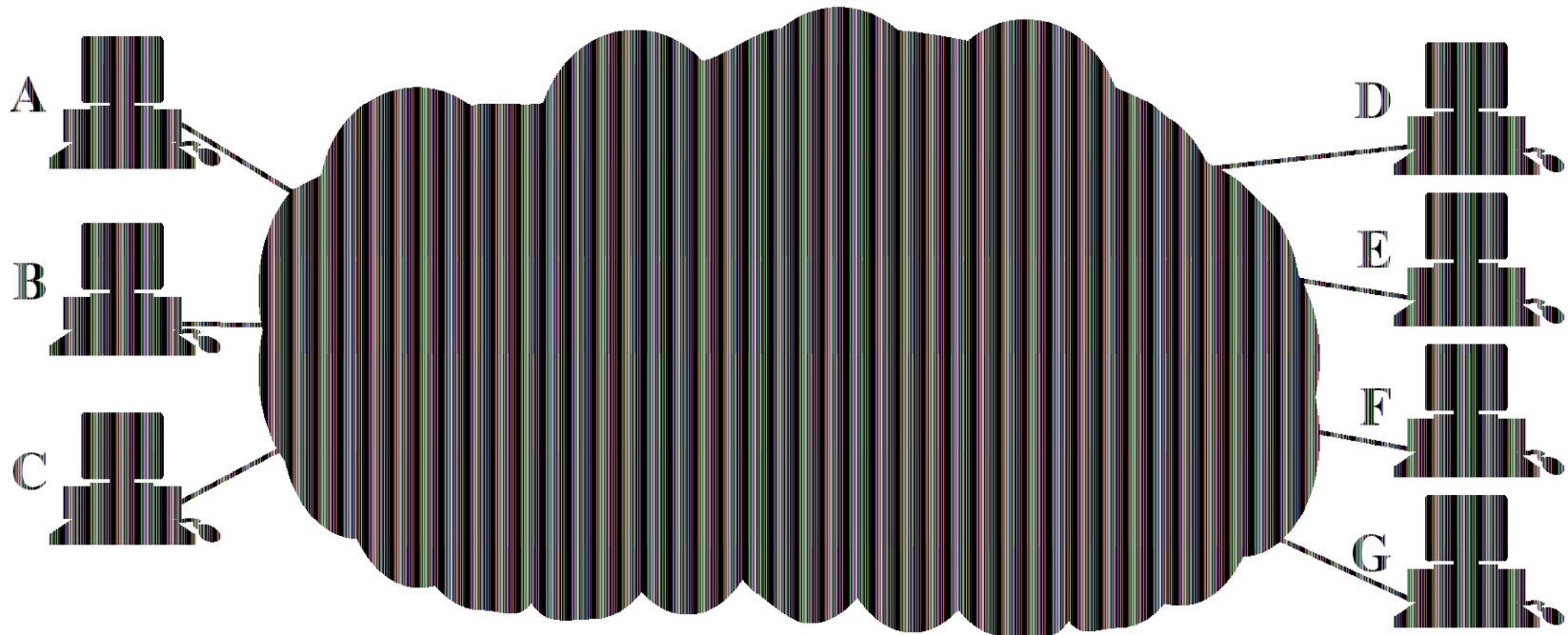
# Circuit Switching

- Circuit switching was designed for voice applications.
- Telephone is the best suitable example of circuit switching. Before a user can make a call, a virtual path between caller and callee is established over the network

**Figure 10-6**  
*Two people carrying on a telephone conversation using a circuit-switched network*



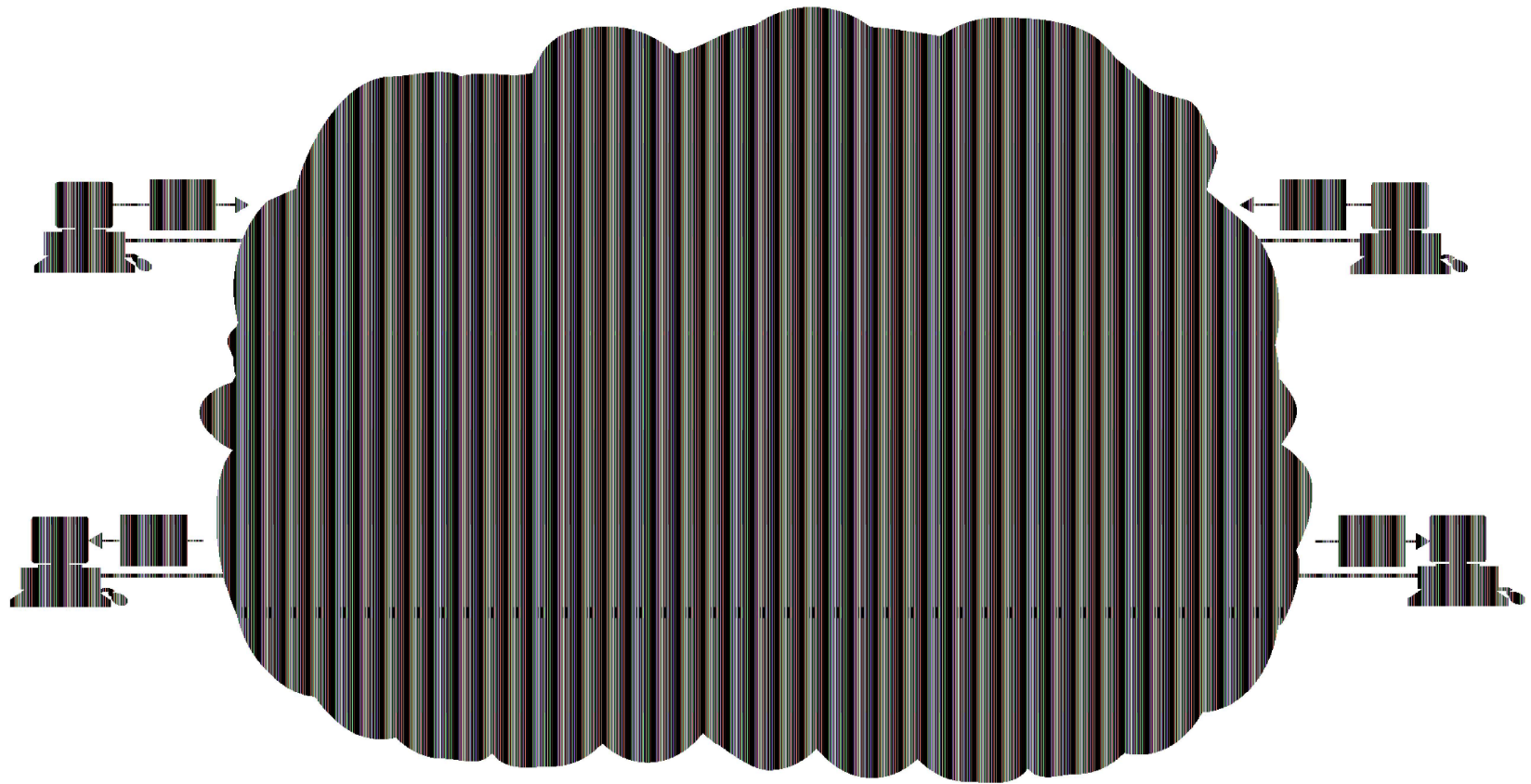
# Circuit Switched Networks



# Message Switching

- This technique was somewhere in middle of circuit switching and packet switching. In message switching, the whole message is treated as a data unit and is switching / transferred in its entirety.
- A switch working on message switching, first receives the whole message and buffers it until there are resources available to transfer it to the next hop.
- If the next hop is not having enough resource to accommodate large size message, the message is stored and switch waits.

# Message Switching



# Message Switching drawbacks

- Every switch in transit path needs enough storage to accommodate entire message.
- Because of store-and-forward technique and waits included until resources are available, message switching is very slow.
- Message switching was not a solution for streaming media and real-time applications.

# Packet Switching

- Shortcomings of message switching gave birth to an idea of packet switching.
- The entire message is broken down into smaller chunks called packets.
- The switching information is added in the header of each packet and transmitted independently.
- It is easier for intermediate networking devices to store small size packets and they do not take much resources either on carrier path or in the internal memory of switches.

# Packet Switching Technique

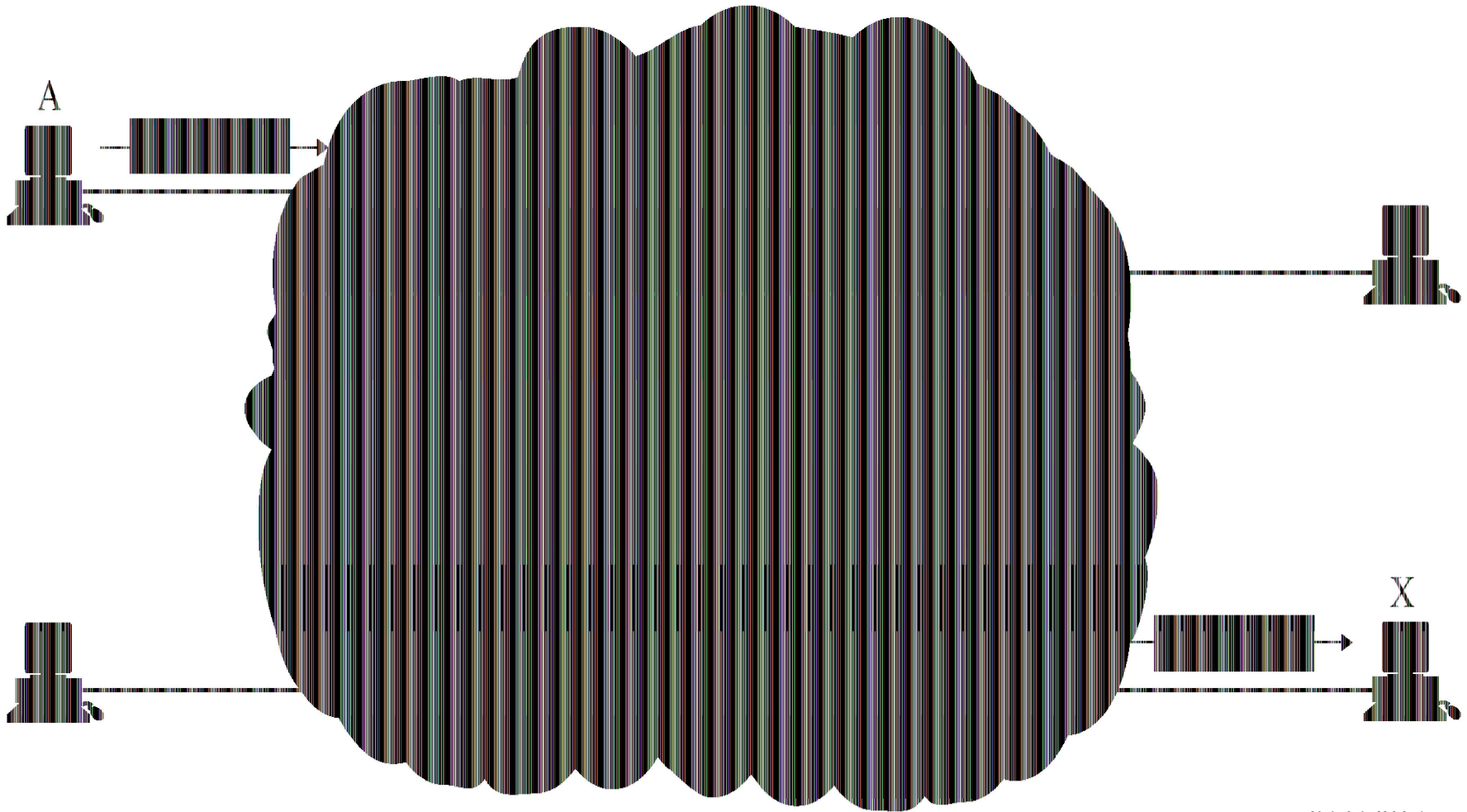
- A station breaks long message into packets
- Packets are sent out to the network sequentially, one at a time
- How will the network handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination?
  - Two approaches
    - **Datagram** approach
    - **Virtual circuit** approach



# Datagram

- Each packet is treated independently, with no reference to packets that have gone before.
  - Each node chooses the next node on a packet's path.
- Packets can take any possible route.
- Packets may arrive at the receiver out of order.
- Packets may go missing.
- It is up to the receiver to re-order packets and recover from missing packets.
- Example: **Internet**

# Datagram Approach



# Virtual Circuit

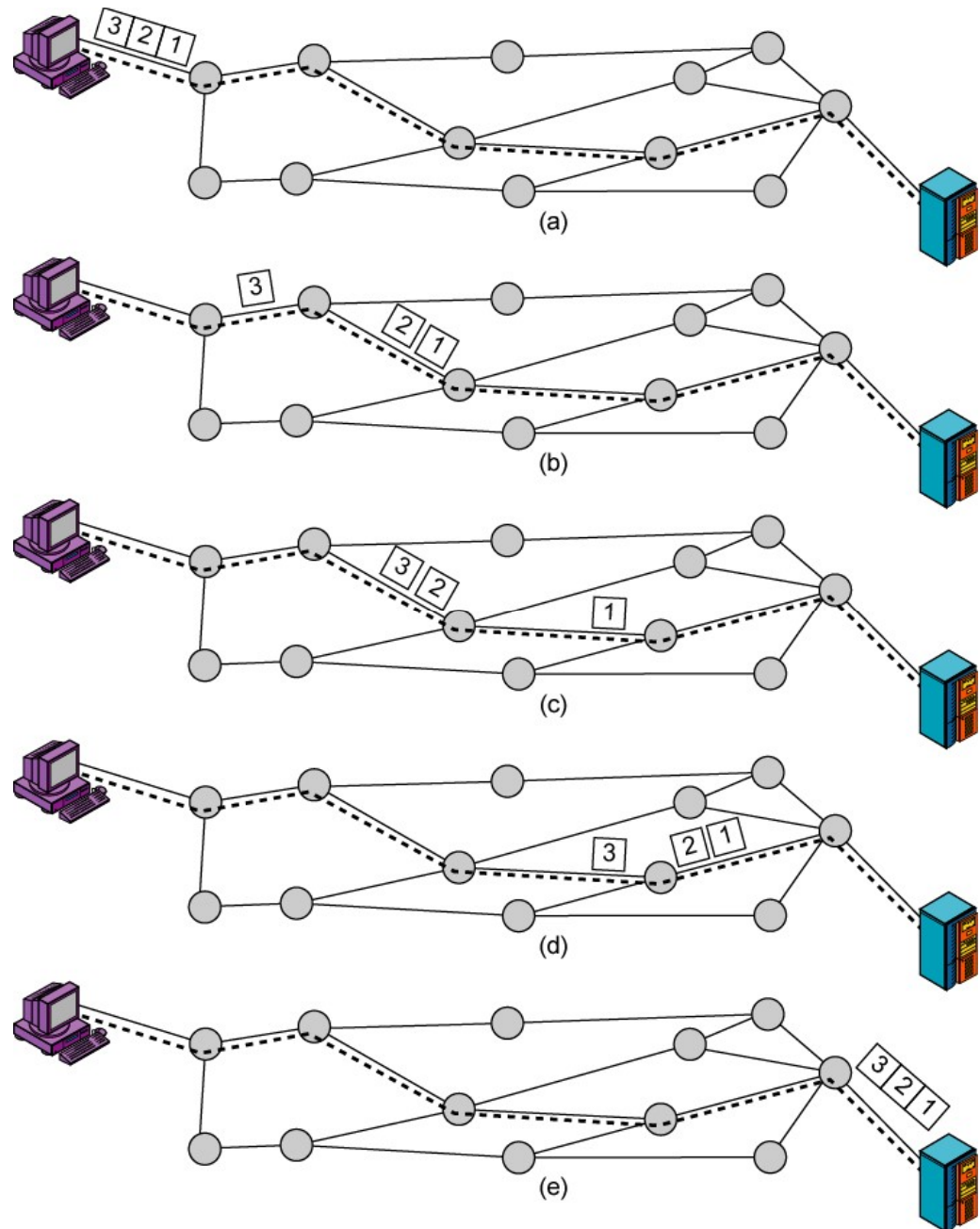
- In virtual circuit, a preplanned route is established before any packets are sent, then all packets follow the same route.
- Each packet contains a **virtual circuit identifier** instead of destination address, and each node on the preestablished route knows where to forward such packets.
  - The node need not make a routing decision for each packet.
- Example: X.25, Frame Relay, ATM

# Virtual Circuit Approach

A route between stations is set up prior to data transfer.

All the data packets then follow the same route.

But there is no dedicated resources reserved for the virtual circuit! Packets need to be stored-and-forwarded.



# Comparison of Virtual-Circuit and Datagram Subnets

<b>Issue</b>	<b>Datagram subnet</b>	<b>Virtual-circuit subnet</b>
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

<b>Circuit Switching</b>	<b>Datagram Packet Switching</b>	<b>Virtual Circuit Packet Switching</b>
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

# Outline

Switching techniques,

**IP Protocol,**

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

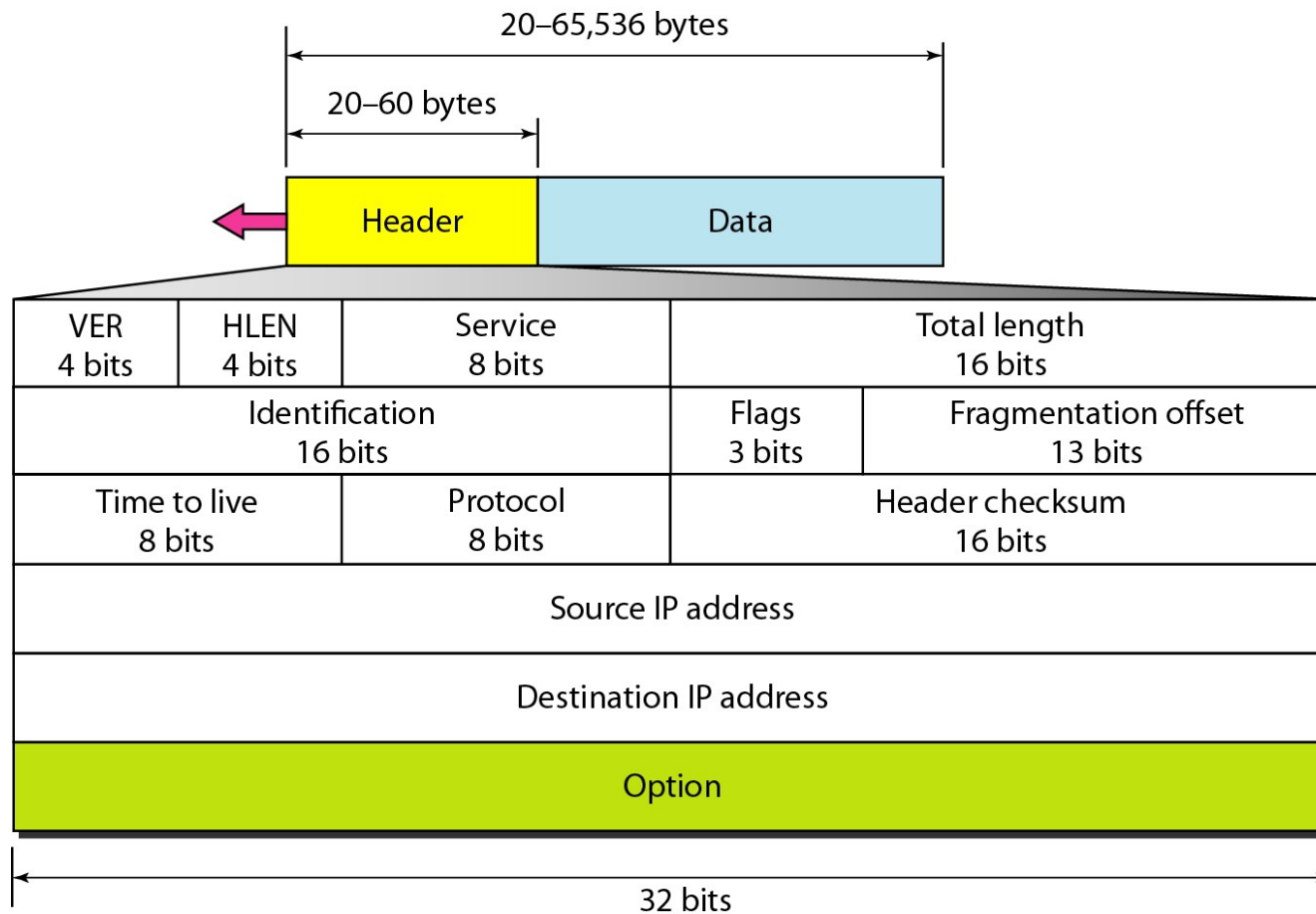
Congestion control and QoS,

MPLS,

Mobile IP,

Routing in MANET : AODV, DSR

# IPv4 datagram format (IPV4 Header)

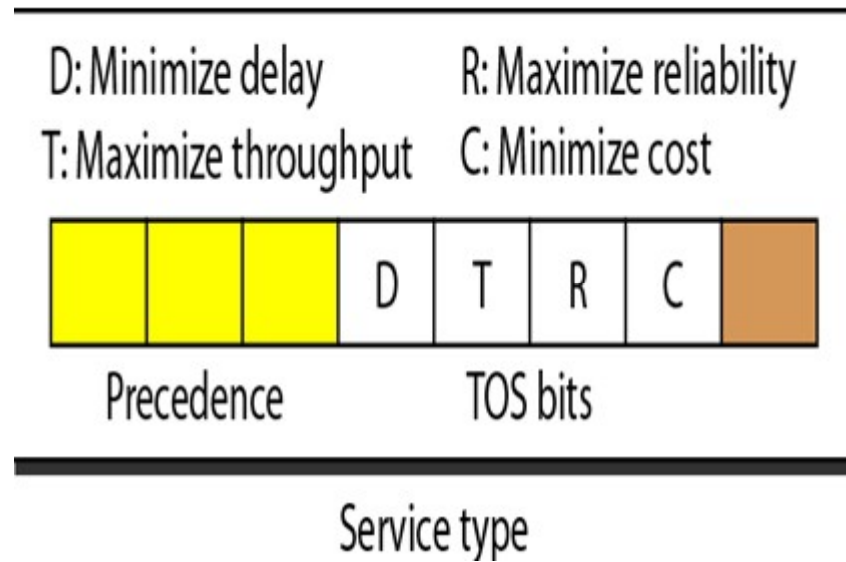




# IPv4 datagram format (IPV4 Header)

- **Version: IP Version**
  - 4 for IPv4
  - 6 for IPv6
- **HLen: Header Length**
  - 32-bit words (typically 5)
- **TOS: Type of Service**
  - Priority information
- **Identifier, flags, fragment offset** → used primarily for fragmentation
- **Time to live**
  - Must be decremented at each router
  - Packets with TTL=0 are thrown away
  - Ensure packets exit the network
- **Protocol**
  - Demultiplexing to higher layer protocols
  - TCP = 6, ICMP = 1, UDP = 17...
- **Header checksum**
  - Ensures some degree of header integrity
  - Relatively weak – only 16 bits
- **Options**
  - E.g. Source routing, record route, etc.
  - Performance issues at routers
    - Poorly supported or not at all
- **Source Address**
  - 32-bit IP address of sender
- **Destination Address**
  - 32-bit IP address of destination

## ***Service type field in IPV4***



## ***Protocol values***

<i>Value</i>	<i>Protocol</i>
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

## Some of the IPv4 options.

<b>Option</b>	<b>Description</b>
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

# Outline

Switching techniques,

IP Protocol,

**IPv4 and IPv6 addressing schemes,**

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

Congestion control and QoS,

MPLS,

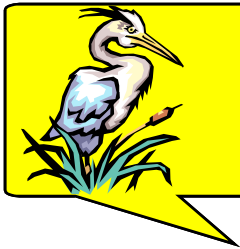
Mobile IP,

Routing in MANET : AODV, DSR

# IPv4 Addressing- Introduction

*An IP address is a **32-bit address** that uniquely and universally defines the connection of a host or a router to the Internet.*

*IP addresses are unique.*

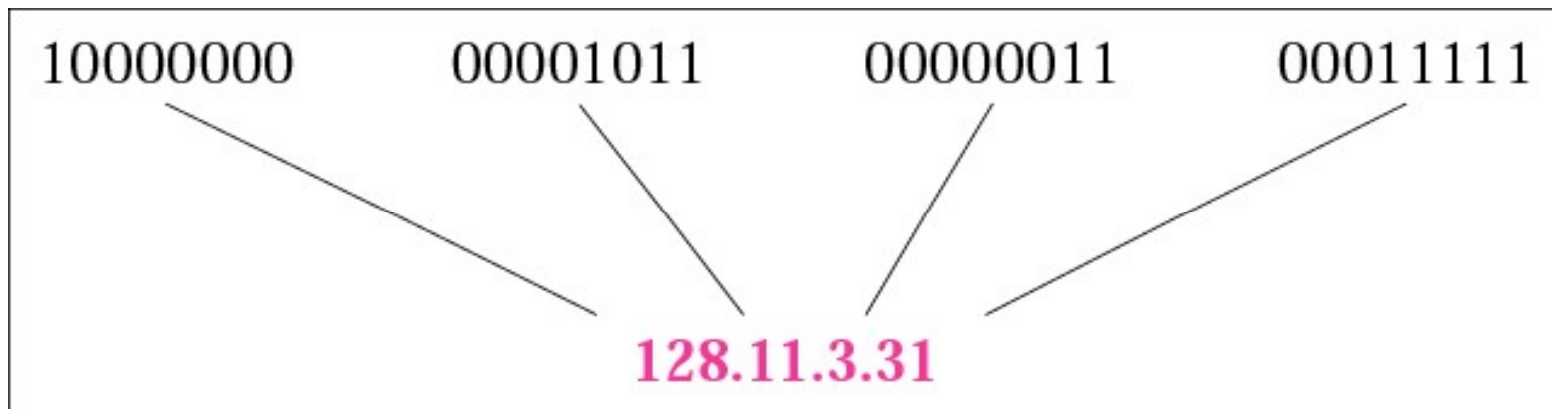


**Note:**

*An IP address is a 32-bit address.*

*The address space of IPv4 is  
 $2^{32}$  or 4,294,967,296.*

## Dotted-decimal and Binary equivalent notation





## *EXAMPLE 1*

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

- a. 10000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111
- c. 11100111 11011011 10001011 01101111
- d. 11111001 10011011 11111011 00001111

### *Solution*

We replace each group of 8 bits with its equivalent decimal number and add dots for separation:

- |                    |                   |
|--------------------|-------------------|
| a. 129.11.11.239   | b. 193.131.27.255 |
| c. 231.219.139.111 | d. 249.155.251.15 |

## *EXAMPLE 2*

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

a. 111.56.45.78

b. 221.34.7.82

c. 241.8.56.12

d. 75.45.34.78

### *Solution*

We replace each decimal number with its binary equivalent:

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

c. 11110001 00001000 00111000 00001100

d. 01001011 00101101 00100010 01001110

# IP Addresses formats and ranges.

← 32 Bits →

Class	Range of host addresses
A	1.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 255.255.255.255

Class	Format	Range of host addresses
A	0   Network   Host	1.0.0.0 to 127.255.255.255
B	10   Network   Host	128.0.0.0 to 191.255.255.255
C	110   Network   Host	192.0.0.0 to 223.255.255.255
D	1110   Multicast address	224.0.0.0 to 239.255.255.255
E	1111   Reserved for future use	240.0.0.0 to 255.255.255.255

# Finding the class in binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	<b>0</b>			
Class B	<b>10</b>			
Class C	<b>110</b>			
Class D	<b>1110</b>			
Class E	<b>1111</b>			

## *EXAMPLE*

Find the class of each address:

- a. 00000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111
- c. 10100111 11011011 10001011 01101111
- d. 11110011 10011011 11111011 00001111

### *Solution*

- a. The first bit is 0. This is a class A address.
- b. The first 2 bits are 1; the third bit is 0. This is a class C address.
- c. The first bit is 0; the second bit is 1. This is a class B address.
- d. The first 4 bits are 1s. This is a class E address..

# Finding the class in decimal notation

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

## *EXAMPLE*

*Find the class of each address:*

*a. 227.12.14.87 b. 193.14.56.22 c. 14.23.120.8*

*d. 252.5.15.111 e. 134.11.78.56*

### *Solution*

*a. The first byte is 227 (between 224 and 239); the class is D.*

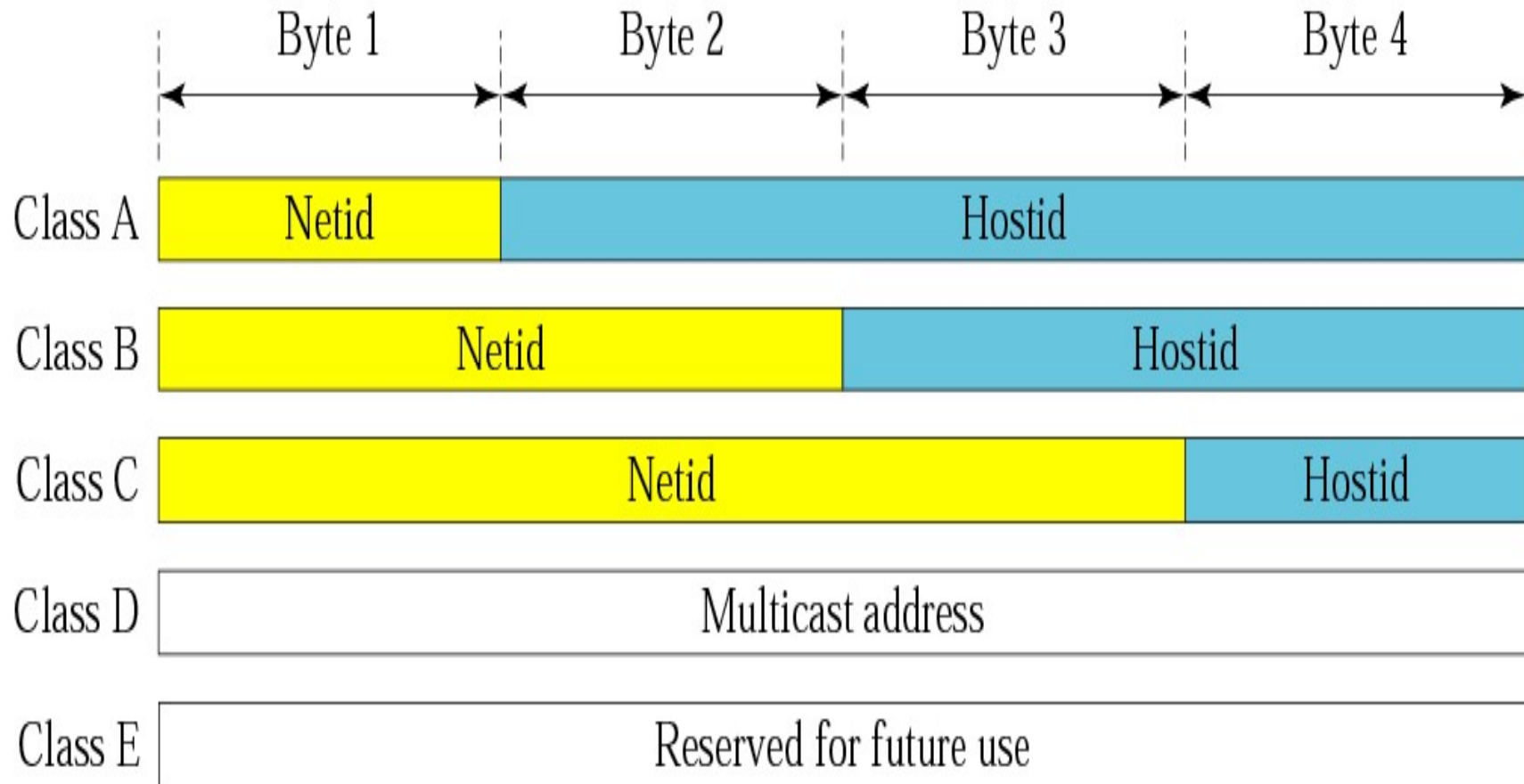
*b. The first byte is 193 (between 192 and 223); the class is C.*

*c. The first byte is 14 (between 0 and 127); the class is A.*

*d. The first byte is 252 (between 240 and 255); the class is E.*

*e. The first byte is 134 (between 128 and 191); the class is B.*

# Netid and hostid





## *EXAMPLE*

*Given the network address 17.0.0.0, find the class, the block, and the range of the addresses.*

### *Solution*

*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.*

## *EXAMPLE*

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

### *Solution*

*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.*

## *EXAMPLE*

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

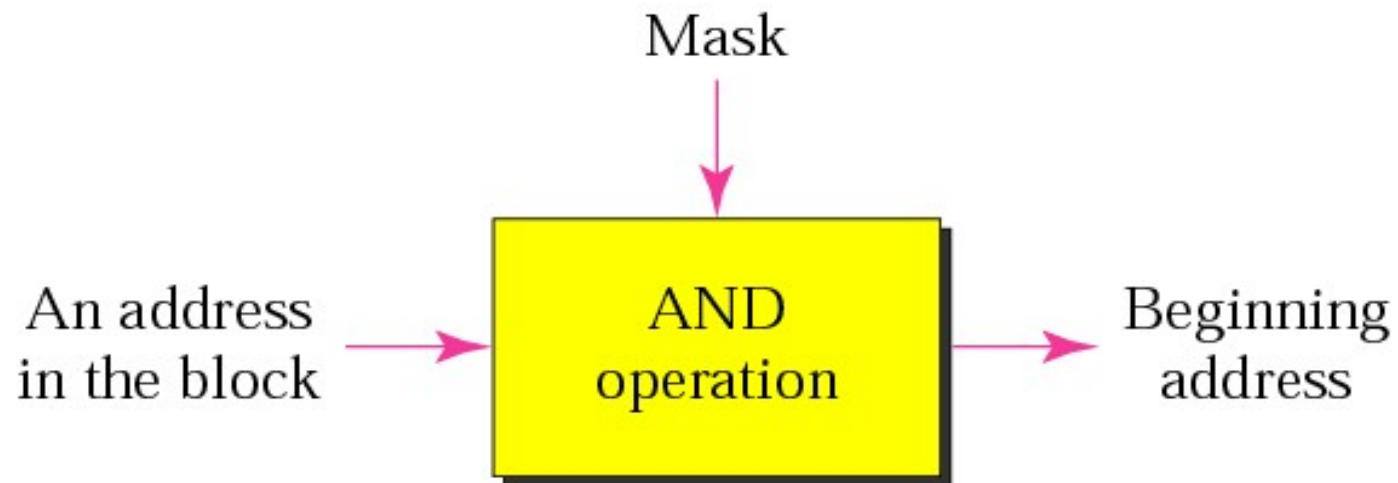
## *Solution*

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

*The block has a netid of 220.34.76.*

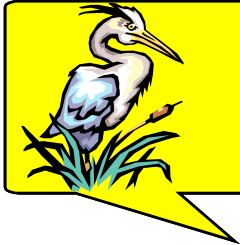
*The addresses range from 220.34.76.0 to 220.34.76.255.*

# Masking concept



# Default masks

<i>Class</i>	<i>Mask in binary</i>	<i>Mask in dotted-decimal</i>
A	<b>11111111</b> 00000000 00000000 00000000	<b>255.0.0.0</b>
B	<b>11111111 11111111</b> 00000000 00000000	<b>255.255.0.0</b>
C	<b>11111111 11111111 11111111</b> 00000000	<b>255.255.255.0</b>



## Note:

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

## ***EXAMPLE***

*Given the address 23.56.7.91, find the beginning address (network address).*

## **Solution**

*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**.*

## *EXAMPLE*

*Given the address 132.6.17.85, find the beginning address (network address).*

### ***Solution***

*The default mask is 255.255.0.0,  
which means that the first 2 bytes are preserved  
and the other 2 bytes are set to 0s.*

*The network address is 132.6.0.0.*





## *EXAMPLE*

*Given the address 201.180.56.5, find the beginning address (network address).*

### *Solution*

*The default mask is 255.255.255.0, which means that the first 3 bytes are preserved and the last byte is set to 0. The network address is 201.180.56.0.*

# Special IP addresses

0 0

This host

0 0      ...      0 0

Host

A host on this network

1 1

Broadcast on the  
local network

Network

1 1 1 1

...

1 1 1 1

Broadcast on a  
distant network

127

(Anything)

Loopback

## IPv6 ADDRESSES

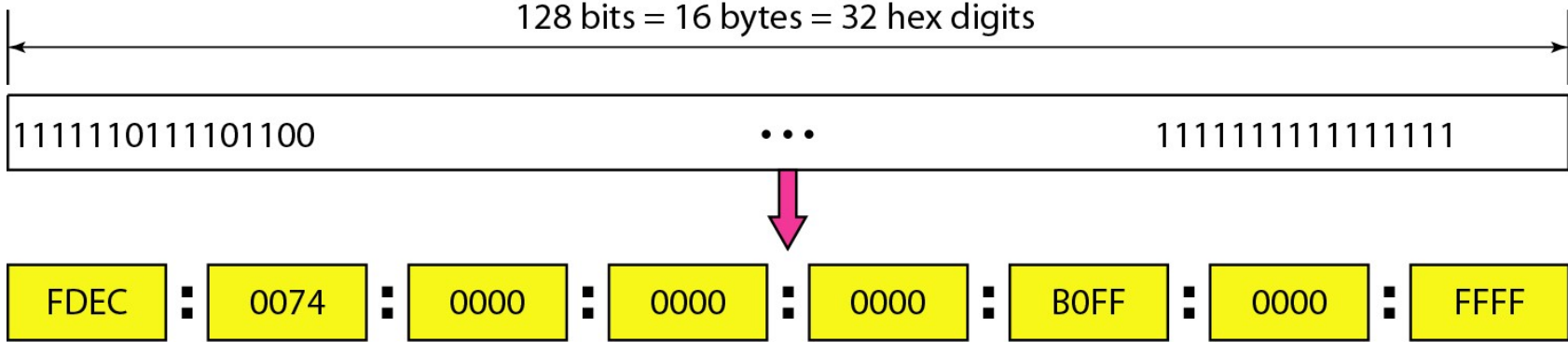
*Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.*



*Note*

An IPv6 address is 128 bits long.

# IPv6 address in binary and hexadecimal colon notation



# Abbreviated IPv6 addresses

Original

FDEC : 0074 : 0000 : 0000 : 0000 : BOFF : 0000 : FFF0



Abbreviated

FDEC : 74 : 0 : 0 : 0 : BOFF : 0 : FFF0



More abbreviated

FDEC : 74 : : BOFF : 0 : FFF0



# IPv6 Colon Hexadecimal Notation

- 128 bit number expressed as dotted decimal

104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255 becomes

68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF

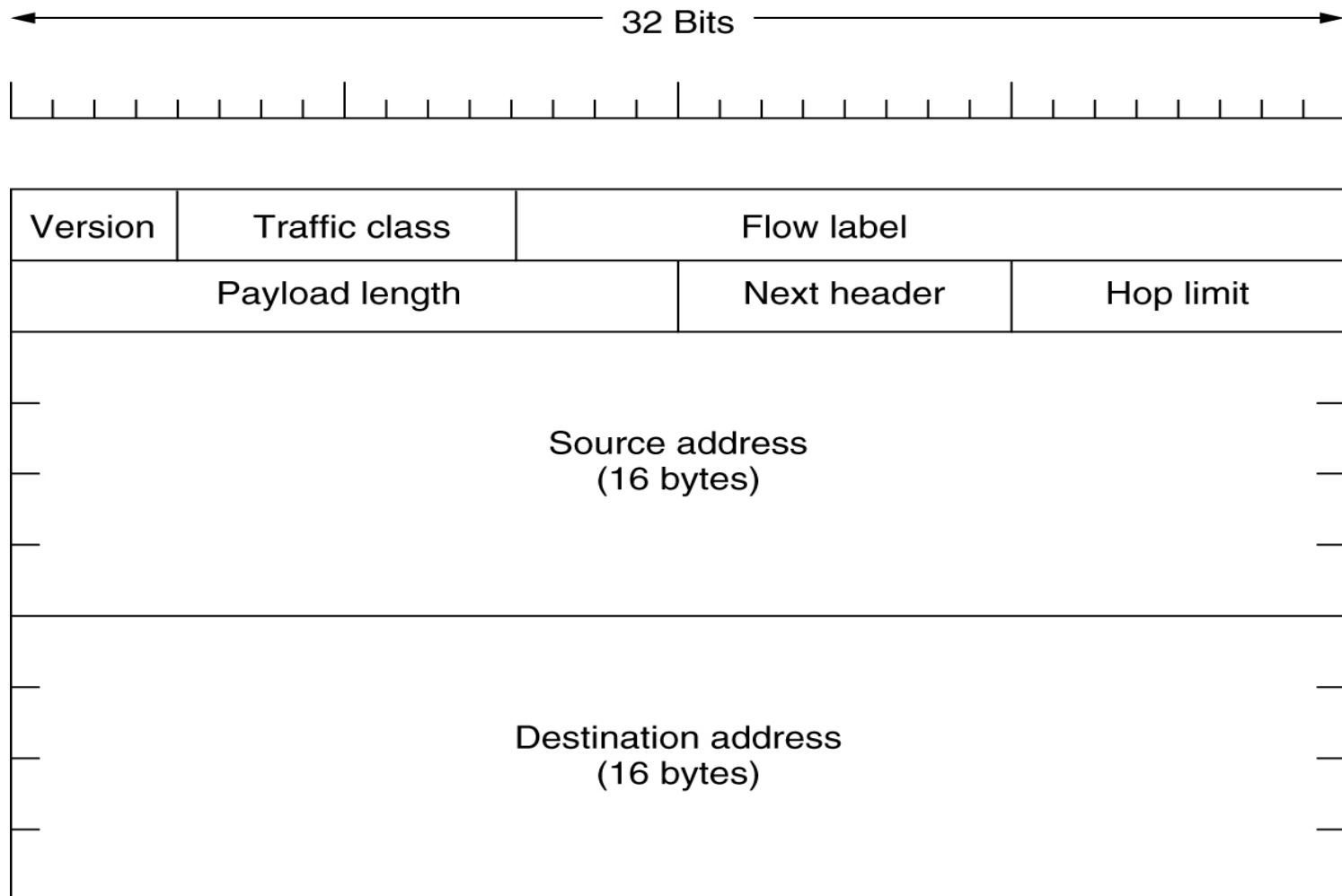
- **Hex notation allows zero compression**
  - A string of repeated zeros is replaced with a pair of colons
  - FF05:0:0:0:0:0:0:B3 becomes FF05::B3
  - Can be applied only once in any address

# Basic IPv6 Address Types

- **Unicast** – Destination address specifies a single computer. Route datagram along shortest path.
- **Anycast** – Destination is a set of computers, possibly at different locations, that all share a single address. Route datagram along shortest path and deliver to exactly one member of the group (i.e. closest member)
- **Multicast** – Destination is a set of computers, possibly at different locations. One copy of the datagram will be delivered to each member of the group using hardware multicast or broadcast if viable.



# The Main IPv6 Header



**The IPv6 fixed header (required).**

8/3/2017

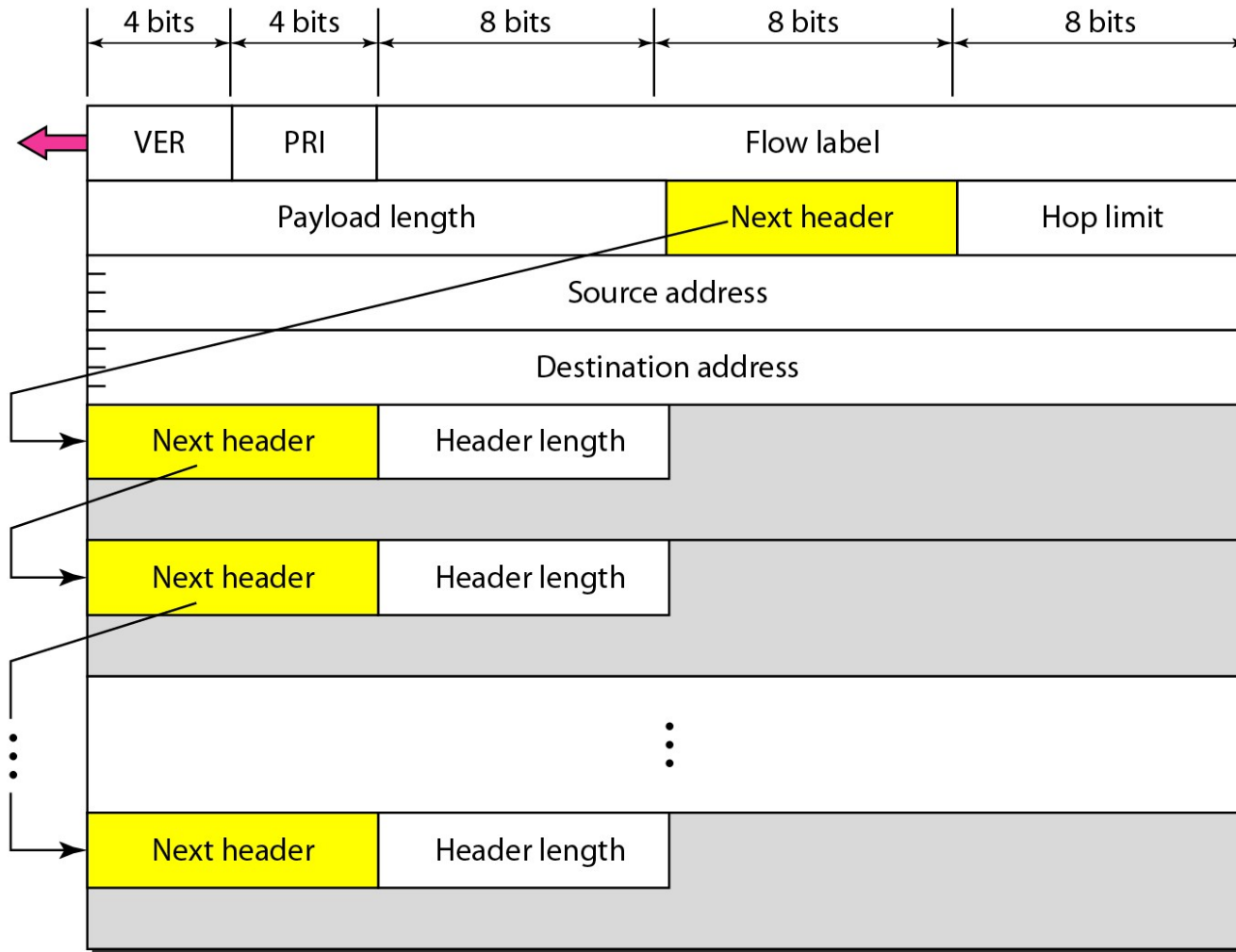
# IPv6 Header Description

- **Version** (4-bits): It represents the version of Internet Protocol
- **Traffic Class** (8-bits): These 8 bits are divided into two parts. The most significant **6 bits are used for Type of Service** & The least significant **2 bits are used for Explicit Congestion Notification** (ECN).
- **Flow Label** (20-bits): This label is used to **maintain the sequential flow of the packets belonging to a communication**. **Payload Length** (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload.

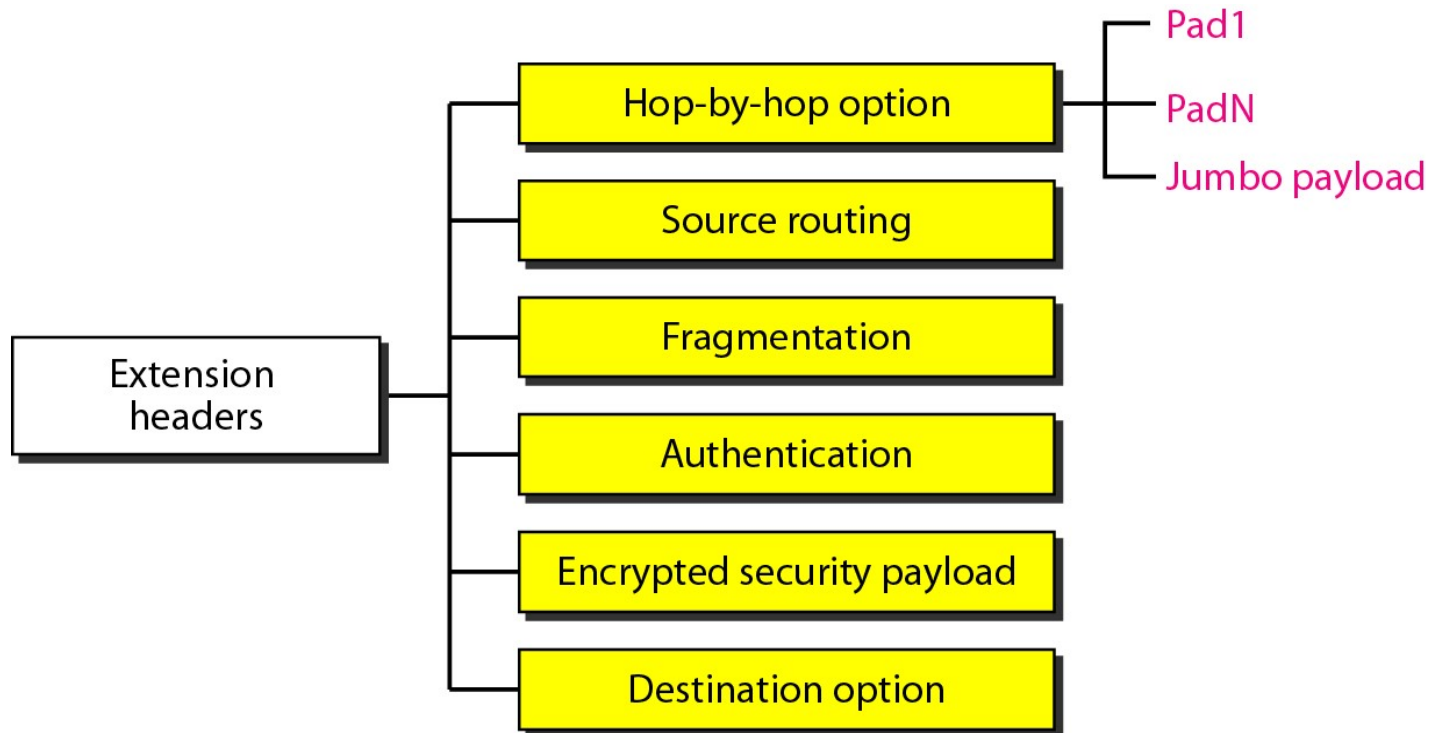
# IPV6 Header Description

- **Next Header** (8-bits): This field is used to indicate either the type of Extension Header.
- **Hop Limit** (8-bits): This field is used to stop packet to loop in the network infinitely. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
- **Source Address** (128-bits): This field indicates the address of originator of the packet.
- **Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

# Format of an IPv6 datagram



# Extension header types



## Advantages of IPv6 over IPv4(Ipv4 v/s Ipv6)

Feature	IPv4	IPv6
Source and destination address	32 bits	128 bits
Address Format	Dotted Decimal	Hexadecimal Notation
No of Address	$2^{32}$	$2^{128}$
IPSec	Optional	required
Payload ID for QoS in the header	No identification	Using <b>Flow label</b> field
Fragmentation	Both router and the sending hosts	<b>Only supported at the sending hosts</b>
Header checksum	included	Not included
Resolve IP address to a link layer address	broadcast ARP request	<b>Multicast Neighbor Solicitation message</b>

# Advantages of IPv6 over IPv4

## (Ipv4 v/s Ipv6) (2)

Feature	IPv4	IPv6
Determine the address of the best default gateway	ICMP Router Discovery(optional)	ICMPv6 Router Solicitation and Router Advertisement (required)
Send traffic to all nodes on a subnet	Broadcast	Link-local scope all-nodes multicast address
Configure address	Manually or DHCP	Autoconfiguration
Manage local subnet group membership	(IGMP)	Multicast Listener Discovery (MLD)

# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

**Subnetting,**

**NAT, CIDR,**

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

Congestion control and QoS,

MPLS,

Mobile IP,

Routing in MANET : AODV, DSR



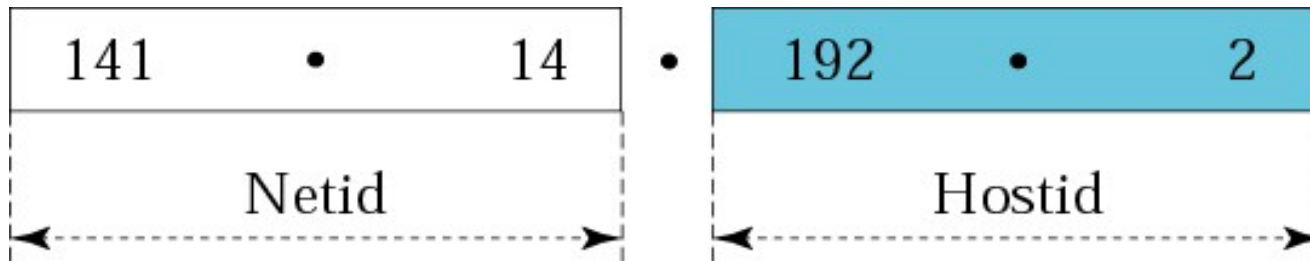
# IPv4 Addressing- Subnetting

The problems associated with classful addressing is that the network addresses available for assignment to organizations are close to depletion.

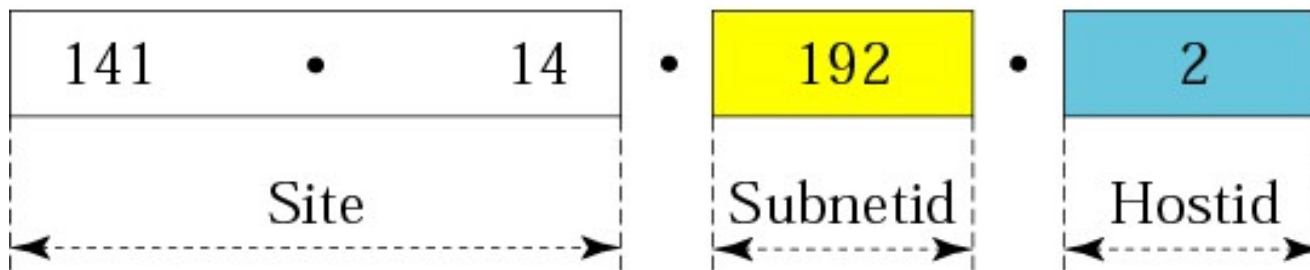
This is coupled with the ever-increasing demand for addresses from organizations that want connection to the Internet.

In this section we briefly discuss two solutions: subnetting and supernetting.

## Addresses in a network with and without subnetting

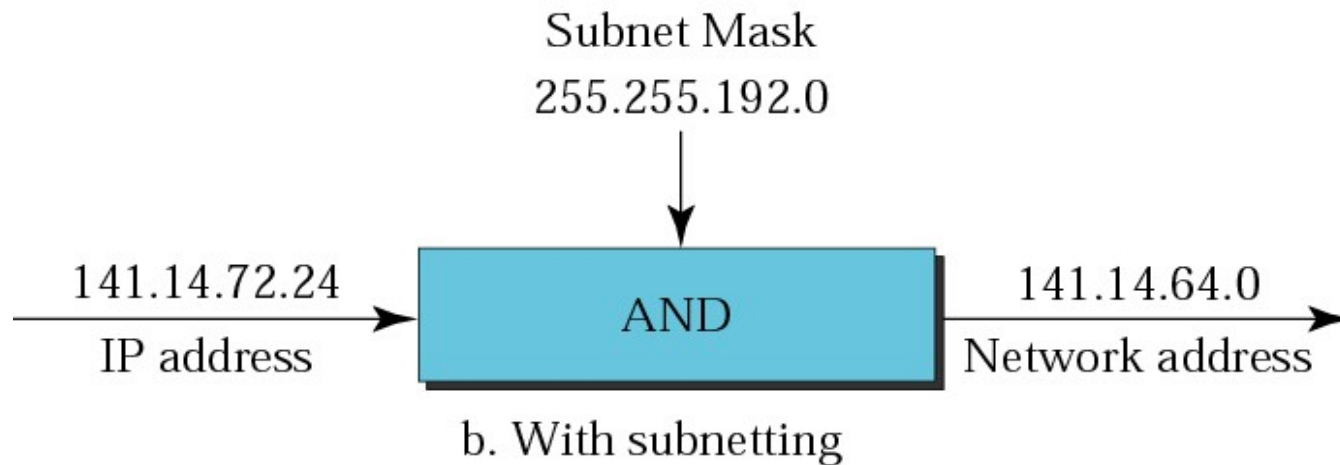
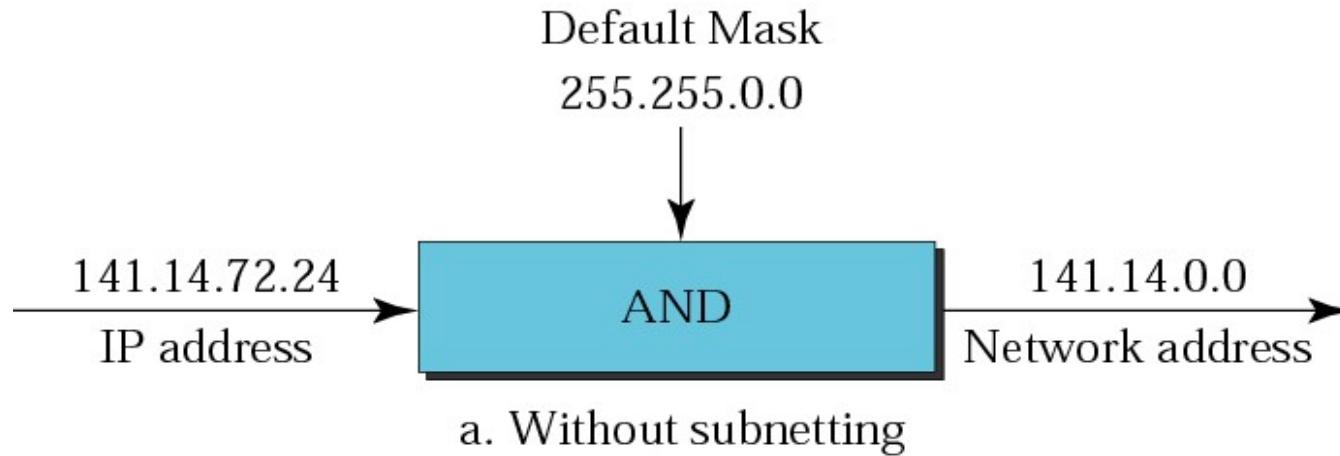


a. Without subnetting

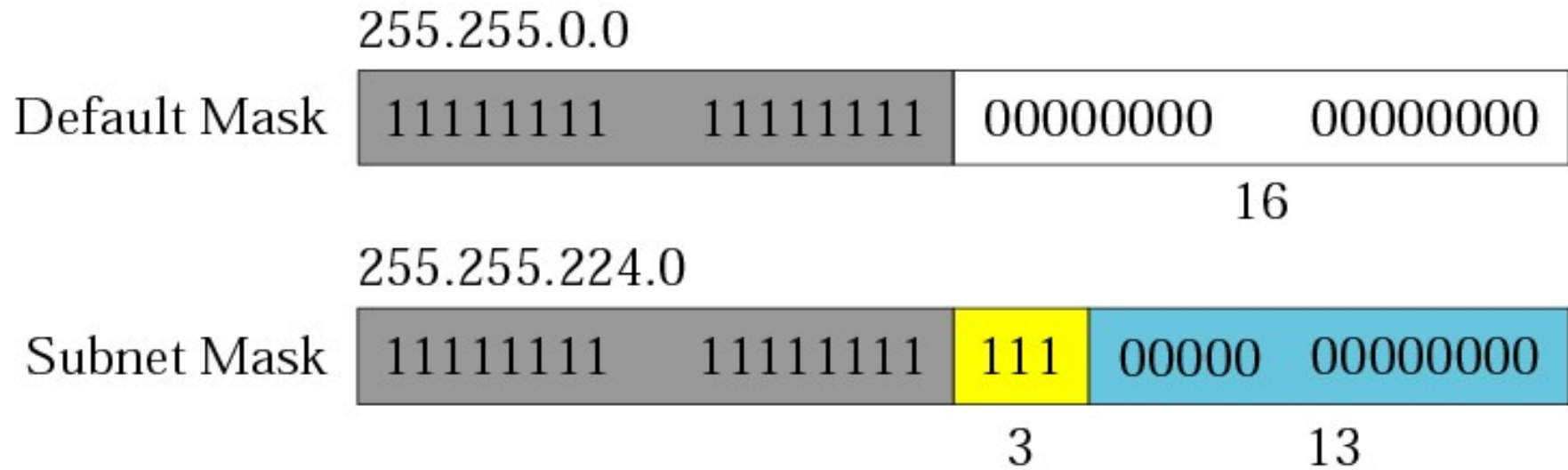


b. With subnetting

## ***Default mask and subnet mask***



## *Comparison of a default mask and a subnet mask*



## For more examples refer

- [https://www.kirkwood.edu/pdf/uploaded/569/ip\\_addressing & subnetting workbook.pdf](https://www.kirkwood.edu/pdf/uploaded/569/ip_addressing_%20and%20subnetting_workbook.pdf)
- <http://www.routeralley.com/guides/ipv4.pdf>

### PPTS from NPTEL

- <http://www.facweb.iitkgp.ernet.in/~isg/INTERNET/SLIDES/Lecture-06.pdf>

## *EXAMPLE*

*What is the subnetwork address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0?*

### *Solution*

*We apply the AND operation on the address and the subnet mask.*

<i>Address</i>	<i>➔ 11001000 00101101 00100010 00111000</i>
<i>Subnet Mask</i>	<i>➔ 11111111 11111111 11110000 00000000</i>
<i>Subnetwork Address</i>	<i>➔ 11001000 00101101 00100000 00000000.</i>

## *STYLE -1* Subnetting when given a required number of networks

**Example 1: A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into 20 separate subnets.**

### *Solution*

**Step 1) Determine the number of subnets and convert to binary**

-In this example, the binary representation of 20 = 00010100.

**Step 2) Reserve required bits in subnet mask and find incremental value**

- The binary value of 20 subnets tells us that we need at least 5 network bits to satisfy this requirement

### *EXAMPLE 1 CONTINUED*

- Our original subnet mask is 255.255.255.0 (Class C subnet) - The full binary representation of the subnet mask is as follows:

255.255.255.0 = 11111111.11111111.11111111.00000000

- We must “convert” 5 of the client bits (0) to network bits (1) in order to satisfy the requirements:

New Mask = 11111111.11111111.11111111.11111000

-If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks – 255.255.255.248 –



*EXAMPLE 1 CONTINUED*

New subnet mask 255.255.255.248

Our increment bit is the last possible network bit, converted back to a binary number:

New Mask = 11111111.11111111.11111111.1111(1)000 –

bit with the parenthesis is your increment bit.

If you convert this bit to a decimal number, it becomes the number “8” that is every subnet is having 8 addresses allotted to it (from 0 to 7, then 8 to 15 etc)

## *EXAMPLE 1 CONTINUED*

### Step 3) **Use increment to find network ranges**

You can now fill in your end ranges, which is the last possible IP address before you start the next range

209.50.1.0 – 209.50.1.7

209.50.1.8 – 209.50.1.15

209.50.1.16 – 209.50.1.23 ...etc

You can then assign these ranges to your networks!

***Remember the first and last address from each range (network / broadcast IP) are unusable***

## *STYLE 2- Subnetting when given a required number of clients*

**Example 1:** A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into as many subnets as possible as long as there are *at least 50 clients per network*.

### *Solution*

Step 1) **Determine the number of clients and convert to binary**

-In this example, the binary representation of 50 = 00110010

-Step 2) **Reserve required bits in subnet mask and find incremental value**

- The binary value of 50 clients tells us that we need at least 6 client bits to satisfy this requirement

## *EXAMPLE 2 CONTINUED*

- Our original subnet mask is 255.255.255.0 (Class C subnet) - The full binary representation of the subnet mask is as follows:

255.255.255.0 = 11111111.11111111.11111111.00000000

-We must ensure 6 of the client bits (0) *remain client bits (save the clients!) in order to satisfy the requirements*. All other bits can become network bits:

-New Mask = 11111111.11111111.11111111.11 000000

-□ **note the 6 client bits that we have saved**

-If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks –

255.255.255.192

*EXAMPLE 2 CONTINUED*

New subnet Mask - 255.255.255.192

Our increment bit is the last possible network bit, converted back to a binary number:

New Mask = 11111111.11111111.11111111.1(1)000000

– bit with the parenthesis is your increment bit.

If you convert this bit to a decimal number, it becomes the number “64” (i.e from 0 to 63, 64 to 127 etc)

*EXAMPLE 2 CONTINUED*

**Step 3) Use increment to find network ranges**

209.50.1.0 – 209.50.1.63

209.50.1.64 – 209.50.1.127

209.50.1.128 – 209.50.1.191

209.50.1.192 – 209.50.1.255

You can then assign these ranges to your networks!

***Remember the first and last address from each range (network / broadcast IP) are unusable***

***STYLE 3*** - Given an IP address & Subnet Mask, find original network range

**Example - You are given the following IP address and subnet mask: 192.168.1.58 255.255.255.240 Identify the original range of addresses (the subnet) that this IP address belongs to**

***Solution***

Break the subnet mask back into binary

255.255.255.240 = 11111111.11111111.11111111.11110000

-As before, the last possible network bit is your increment.

-In this case, the increment is 16

-

-Use this increment to find the network ranges until you pass the given IP address:

192.168.1.0

192.168.1.16

192.168.1.32

192.168.1.48

192.168.1.64 (*passed given IP address 192.168.1.58*)

*EXAMPLE 3 CONTINUED*

- Now, fill in the end ranges to find the answer to the scenario:

192.168.1.0 – 192.168.1.15

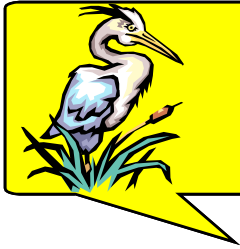
192.168.1.16 – 192.168.1.31

192.168.1.32 – 192.168.1.47

**192.168.1.48 – 192.168.1.63**

***(IP address 192.168.1.58 belongs to this range)***



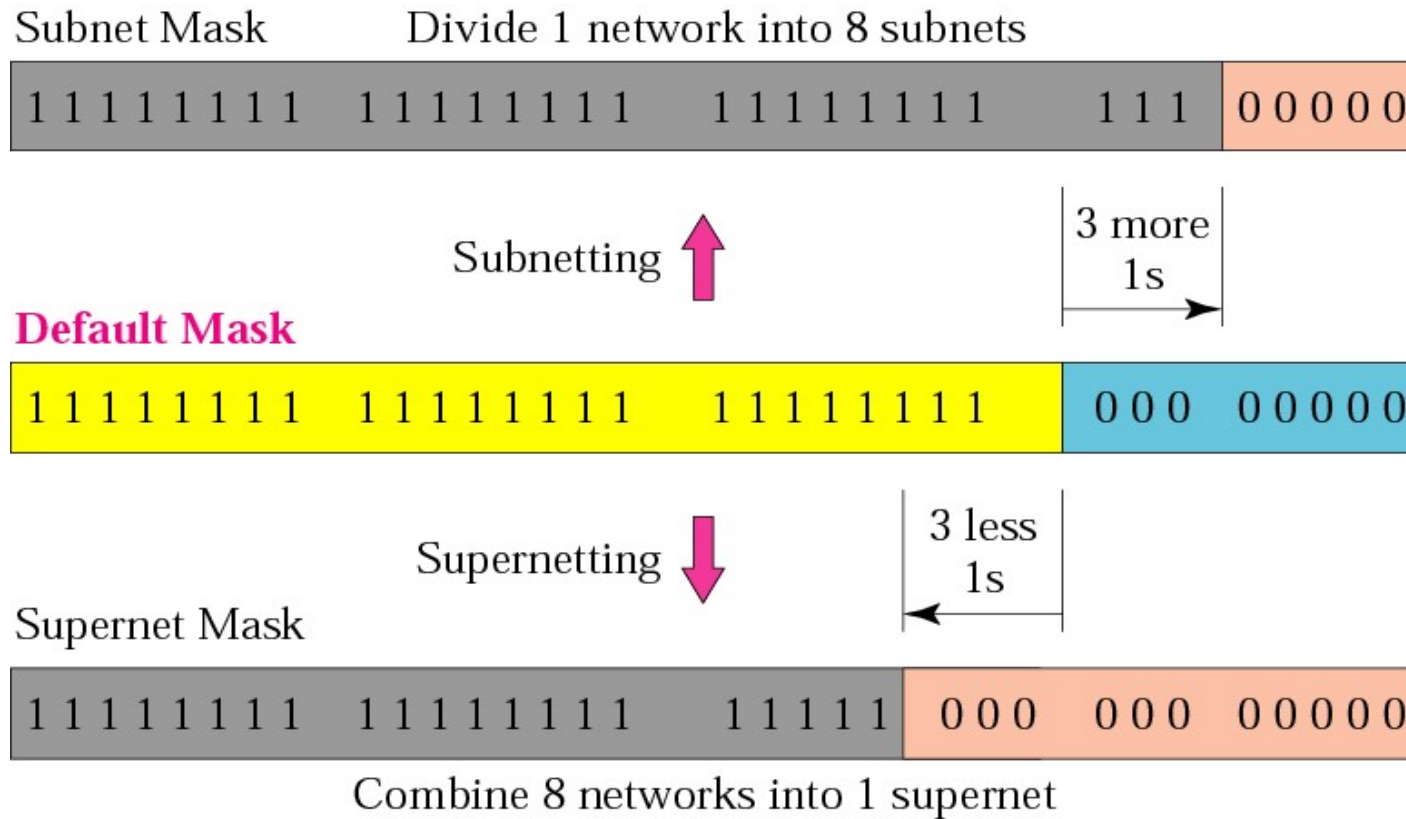


## Note:

*In subnetting, we need the first address of the subnet and the subnet mask to define the range of addresses.*

*In supernetting, we need the first address of the supernet and the supernet mask to define the range of addresses.*

## Comparison of subnet, default, and supernet masks



*IP Addresses:*  
***Classless Addressing***  
***(CIDR- Classless Inter domain***  
***Routing)***

# Classless Addressing

It uses slash notation with IP Address

Example: 142.4.7.3/27

Here /27 means from total 32bit address first 27 bits are for Network and remaining i.e.  $32-27=5$  bits are for host

## Prefix lengths

<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>	<i>/n</i>	<i>Mask</i>
/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

The addresses in color are the default masks for classes A, B, and C.  
Thus, classful addressing is a special case of classless addressing.



*EXAMPLE - FIND FIRST ADDRESS*

*What is the first address in the block if one of the addresses is 167.199.170.82/27?*



## *EXAMPLE - FIND FIRST ADDRESS*

*What is the first address(network address) in the block if one of the addresses is 167.199.170.82/27?*

### *Solution*

*The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The following shows the process:*

*Address in binary: 10100111 11000111 10101010 01010010*  
*Keep the left 27 bits: 10100111 11000111 10101010 01000000*  
*Result in CIDR notation: 167.199.170.64/27*



*Example 19.6*

*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  $32 - 28 = 4$  rightmost bits to 0, we get*

*11001101 00010000 00100101 00100000*

*or*

*205.16.37.32.*





*Note*

The last address in the block can be found  
by setting the rightmost  
 $32 - n$  bits to 1s.



*Example 19.7*

*Find the last address for the block in 205.16.37.39/28*

*Solution*

*The binary representation of the given address is*

*11001101 00010000 00100101 00100111*

*If we set  $32 - 28 = 4$  rightmost bits to 1, we get*

*11001101 00010000 00100101 0010**1111***

*or*

*205.16.37.47*



*Note*

The number of addresses in the block can be found by using the formula

$$2^{32-n}.$$



*Example 19.8*

*Find the number of addresses in 205.16.37.39/28*

*Solution*

*The value of  $n$  is 28, which means that number of addresses is  $2^{32-28}$  or 16.*



### *Example 19.9*

*Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as*

*11111111 11111111 11111111 11110000*

*(twenty-eight 1s and four 0s).*

*Find*

- a. The first address*
- b. The last address*
- c. The number of addresses.*

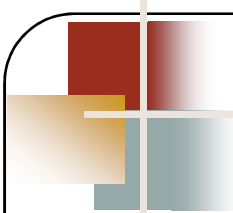


### *Example 19.10*

*An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:*

- a. The first group has 64 customers; each needs 256 addresses.*
- b. The second group has 128 customers; each needs 128 addresses.*
- c. The third group has 128 customers; each needs 64 addresses.*

*Design the subblocks and find out how many addresses are still available after these allocations.*



### *Example 19.10 (continued)*

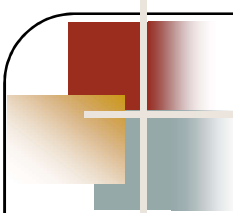
#### *Solution*

.

#### *Group 1*

*For this group, each customer needs 256 addresses. This means that 8 ( $\log_2 256$ ) bits are needed to define each host. The prefix length is then  $32 - 8 = 24$ . The addresses are*

<i>1st Customer:</i>	<i>190.100.0.0/24</i>	<i>190.100.0.255/24</i>
<i>2nd Customer:</i>	<i>190.100.1.0/24</i>	<i>190.100.1.255/24</i>
<i>...</i>		
<i>64th Customer:</i>	<i>190.100.63.0/24</i>	<i>190.100.63.255/24</i>
<i>Total = <math>64 \times 256 = 16,384</math></i>		



## *Example 19.10 (continued)*

### *Group 2*

*For this group, each customer needs 128 addresses. This means that 7 ( $\log_2 128$ ) bits are needed to define each host. The prefix length is then  $32 - 7 = 25$ . The addresses are*

<i>1st Customer:</i>	<i>190.100.64.0/25</i>	<i>190.100.64.127/25</i>
<i>2nd Customer:</i>	<i>190.100.64.128/25</i>	<i>190.100.64.255/25</i>
<i>...</i>		
<i>128th Customer:</i>	<i>190.100.127.128/25</i>	<i>190.100.127.255/25</i>
<i>Total = <math>128 \times 128 = 16,384</math></i>		



### *Example 19.10 (continued)*

#### *Group 3*

*For this group, each customer needs 64 addresses. This means that 6 ( $\log_2 64$ ) bits are needed to each host. The prefix length is then  $32 - 6 = 26$ . The addresses are*

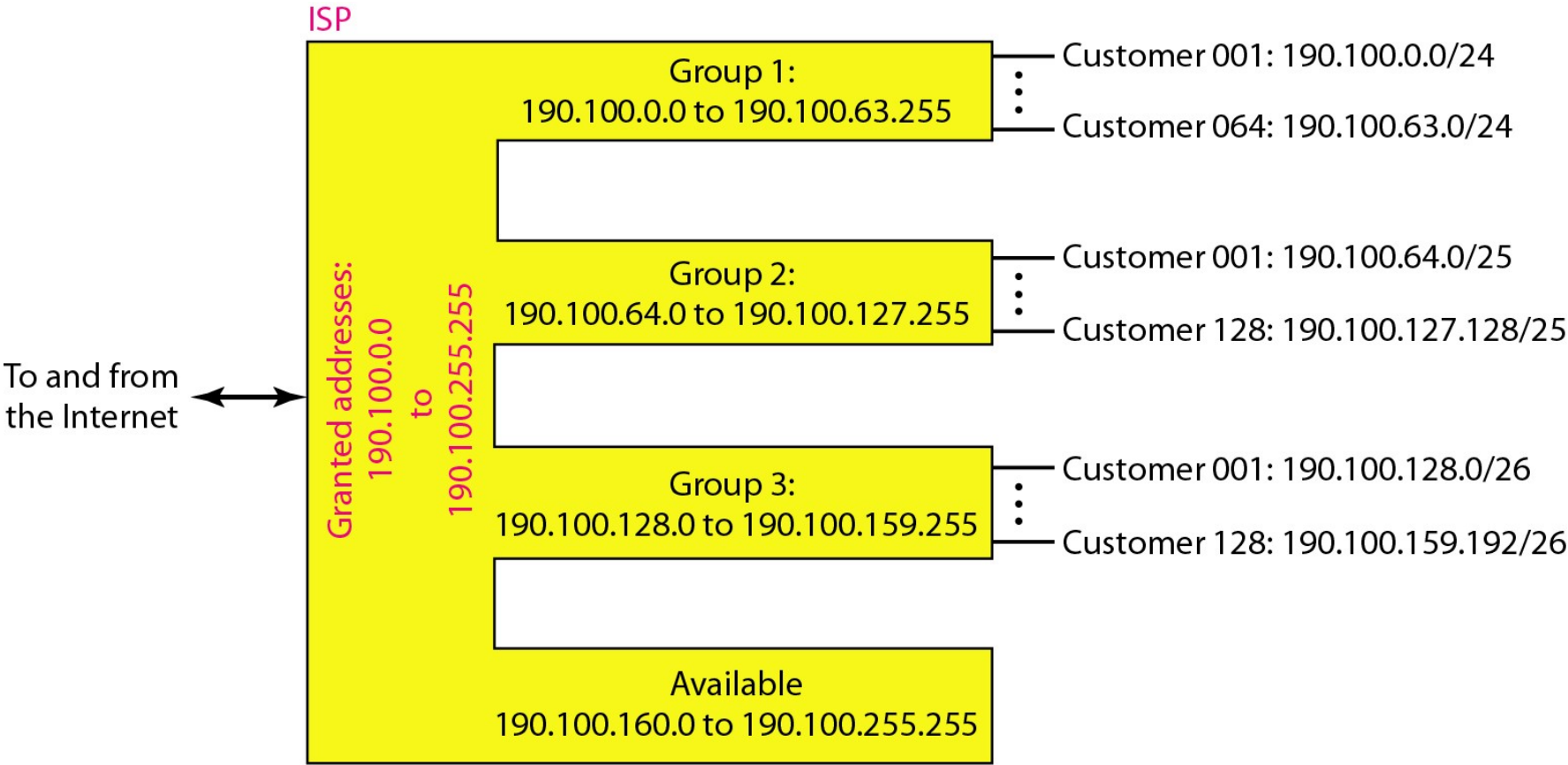
<i>1st Customer:</i>	<i>190.100.128.0/26</i>	<i>190.100.128.63/26</i>
<i>2nd Customer:</i>	<i>190.100.128.64/26</i>	<i>190.100.128.127/26</i>
<i>...</i>		
<i>128th Customer:</i>	<i>190.100.159.192/26</i>	<i>190.100.159.255/26</i>
<i>Total =</i>	<i><math>128 \times 64 = 8192</math></i>	

*Number of granted addresses to the ISP: 65,536*

*Number of allocated addresses by the ISP: 40,960*

*Number of available addresses: 24,576*

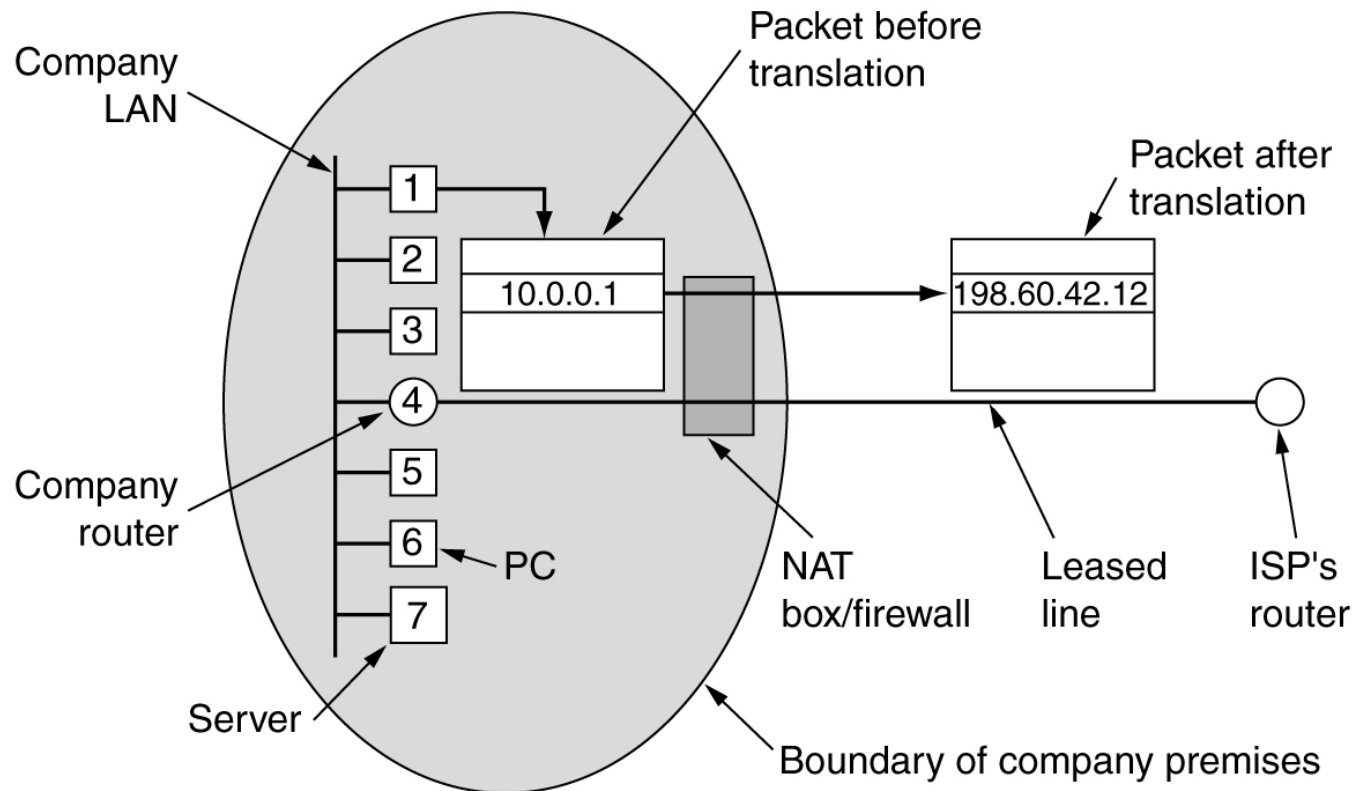
Figure 19.9 *An example of address allocation and distribution by an ISP*



# Addresses for **private networks**

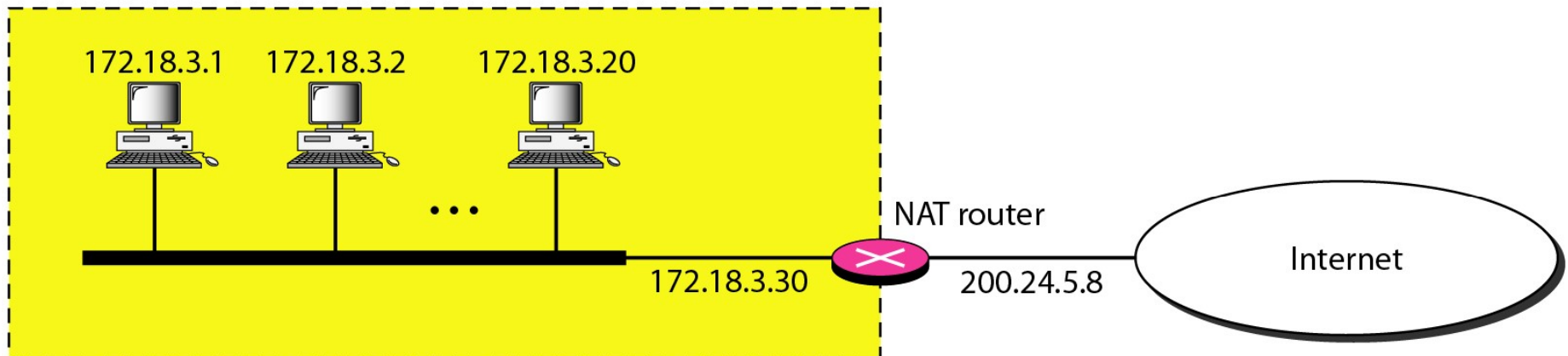
<i>Class</i>	<i>Netids</i>	<i>Blocks</i>
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

# NAT – Network Address Translation

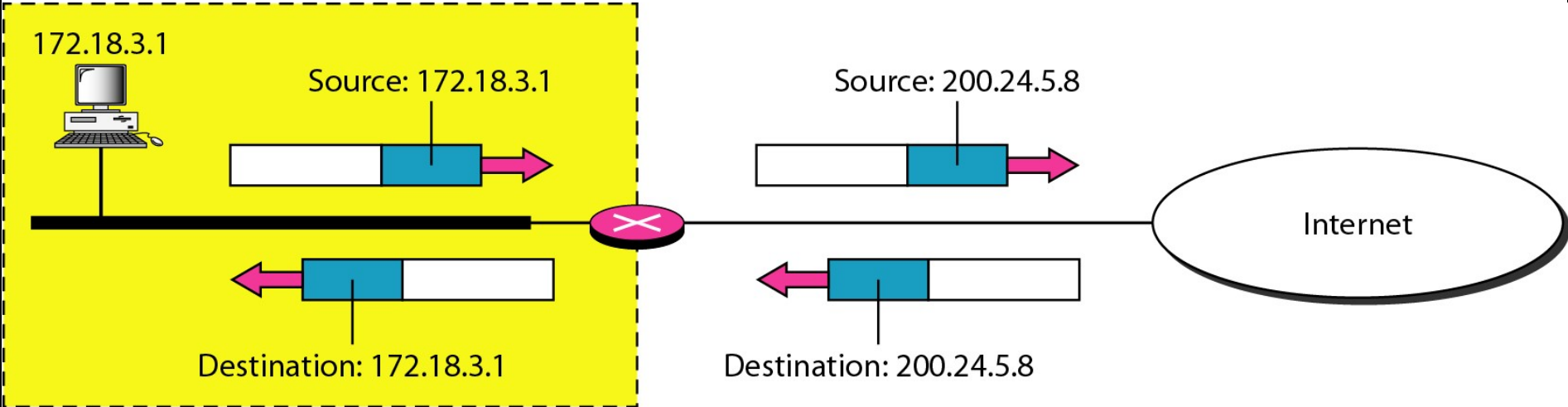


## A NAT implementation

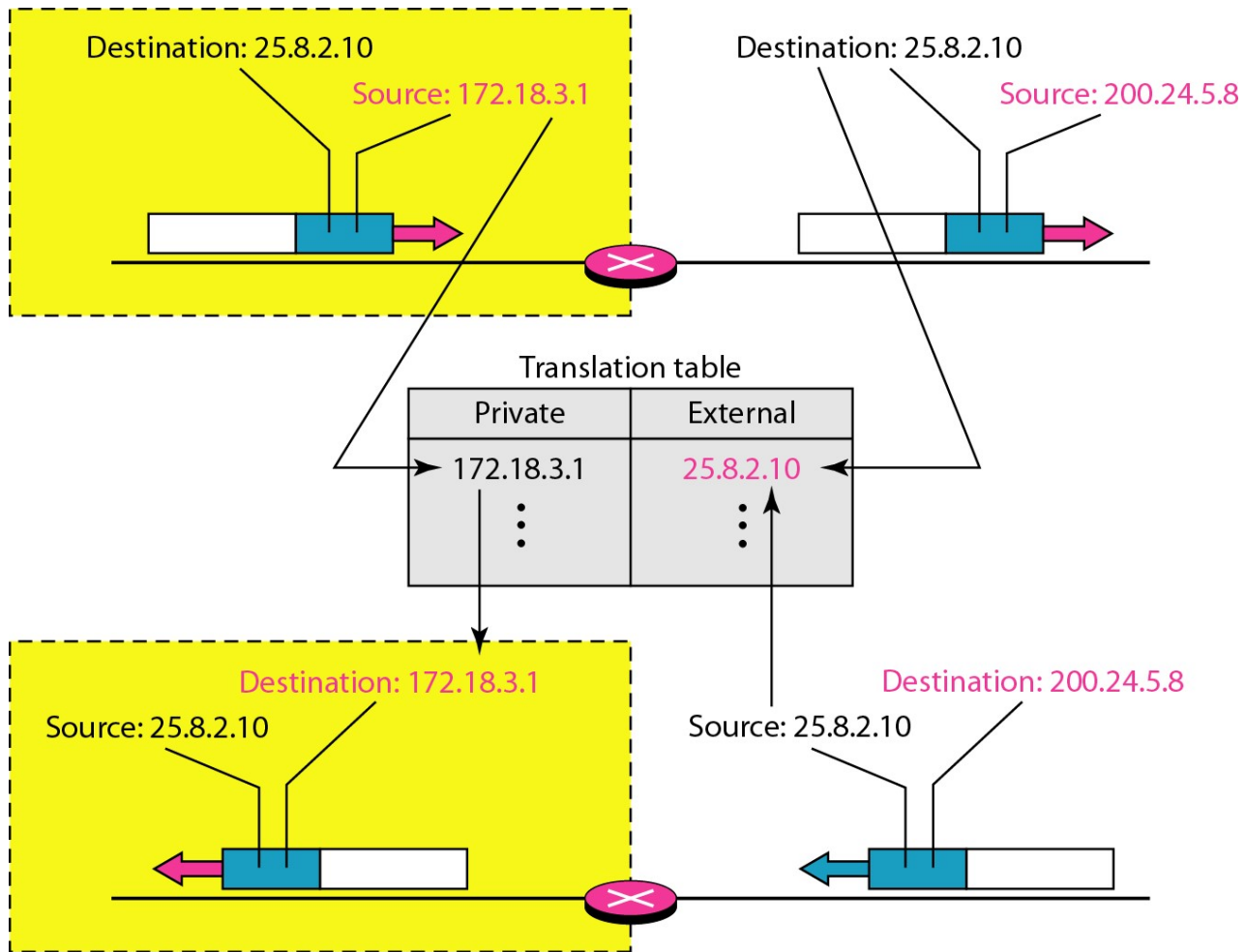
Site using private addresses



# Addresses in a NAT



# NAT address translation



# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

**ICMP,**

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

Congestion control and QoS,

MPLS,

Mobile IP,

Routing in MANET : AODV, DSR



# ICMP V4 -Introduction

❑The IP protocol has no error-reporting or error correcting mechanism.

❑What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a router to the final destination, or

❑Because the time-to-live field has a zero value?

❑These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

❑The solution is ICMP protocol

# ICMP V4 - MESSAGES

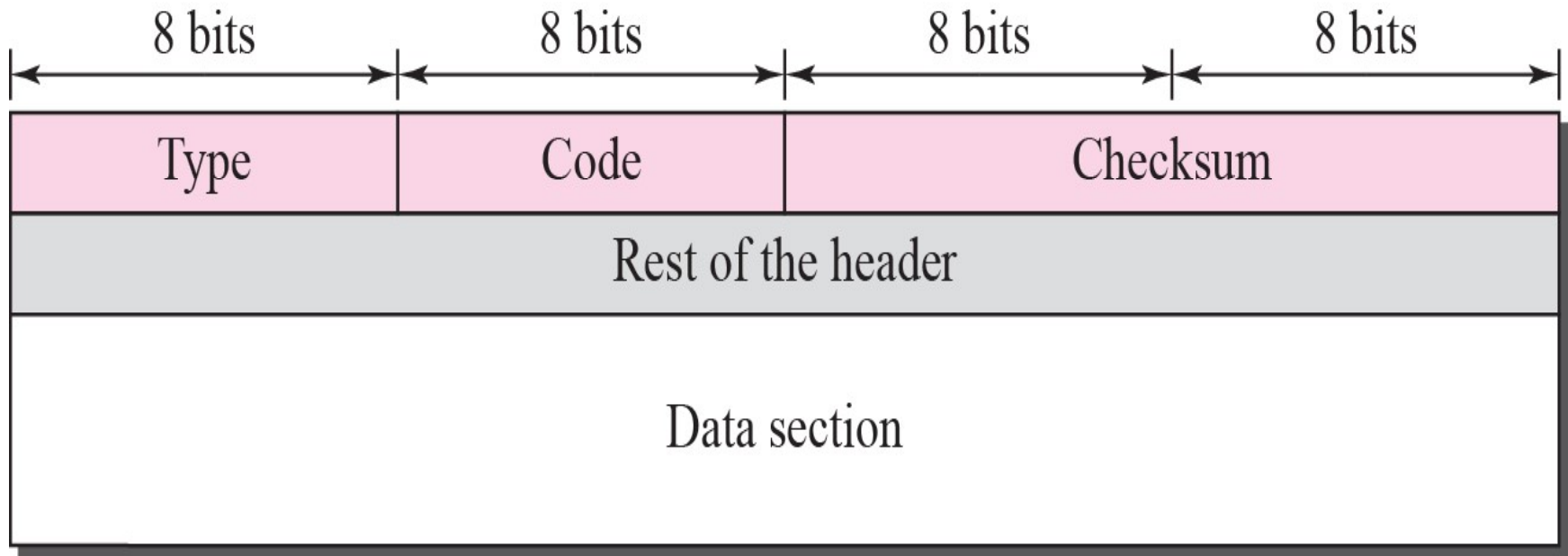
❑ ICMP messages are divided into two broad categories:

1. error-reporting messages
2. query messages.

❑ The **error-reporting messages** report problems that a router or a host (destination) may encounter when it processes an IP packet.

❑ The **query messages**, help a host or a network manager get specific information from a router or another host. Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.

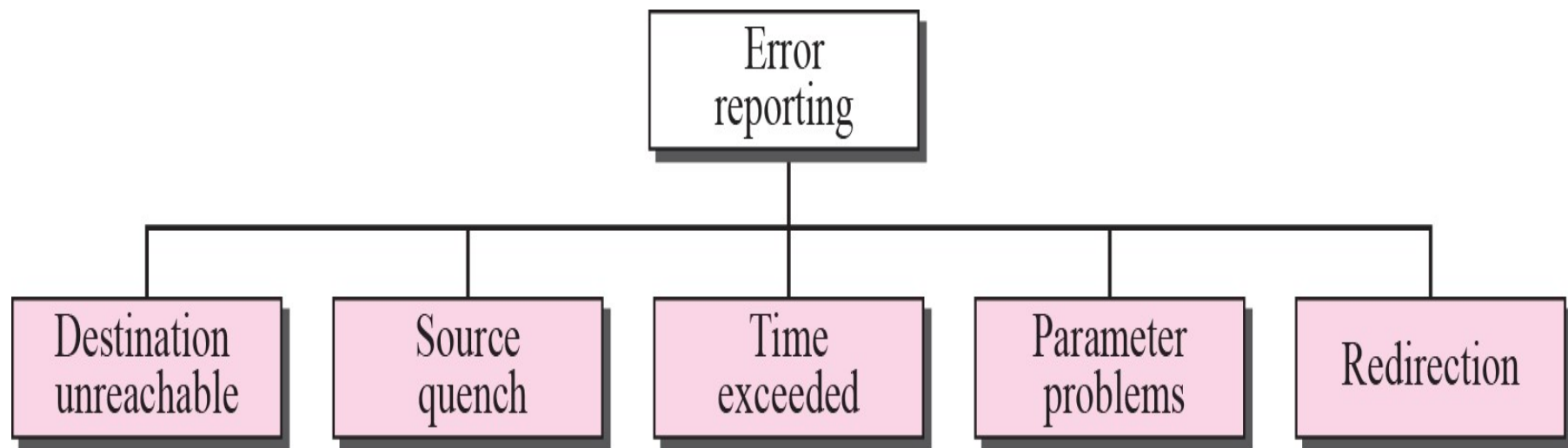
# General format of ICMP messages or ICMP header



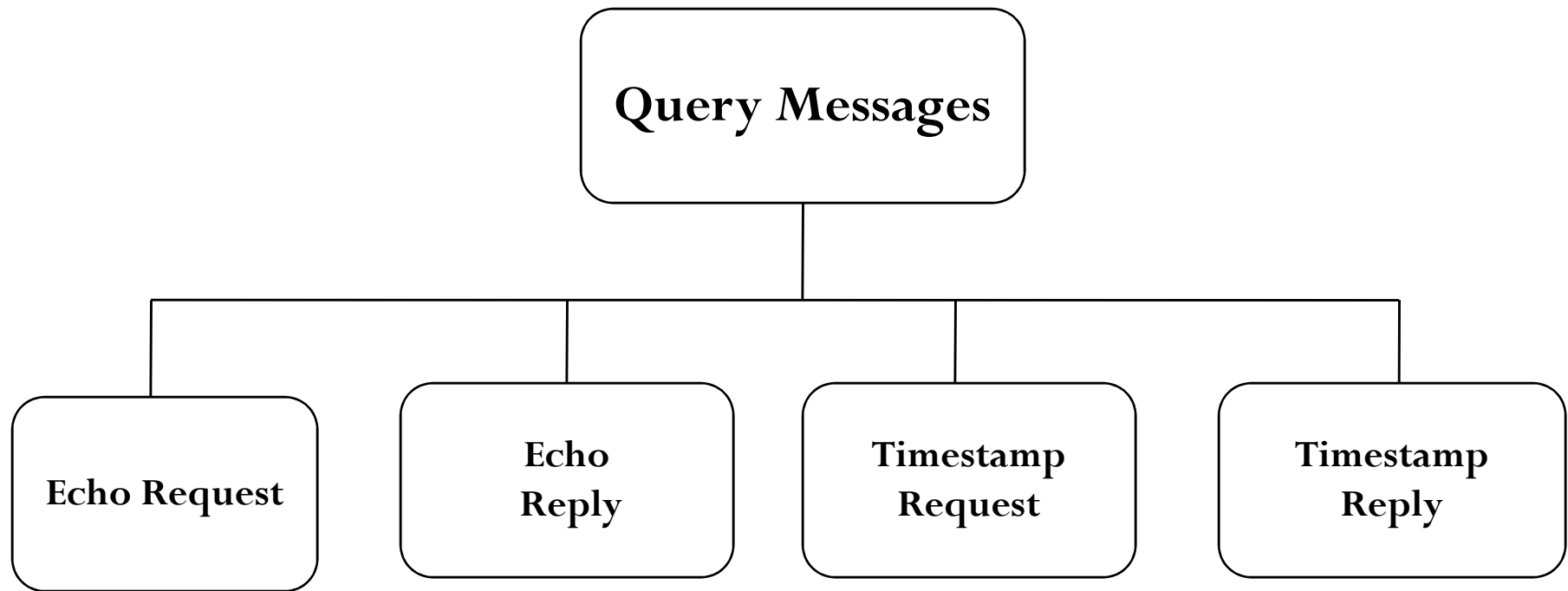
# Basic ICMP Header

- ▶ Headers are 32 bits in length; all contain same three fields
  - type - 8 bit message type code
    - Thirteen message type are defined
  - code - 8 bit;
    - indicating why message is being sent
  - checksum - standard internet checksum
    - for purpose of calculation the checksum field is set to zero

# Error-reporting messages



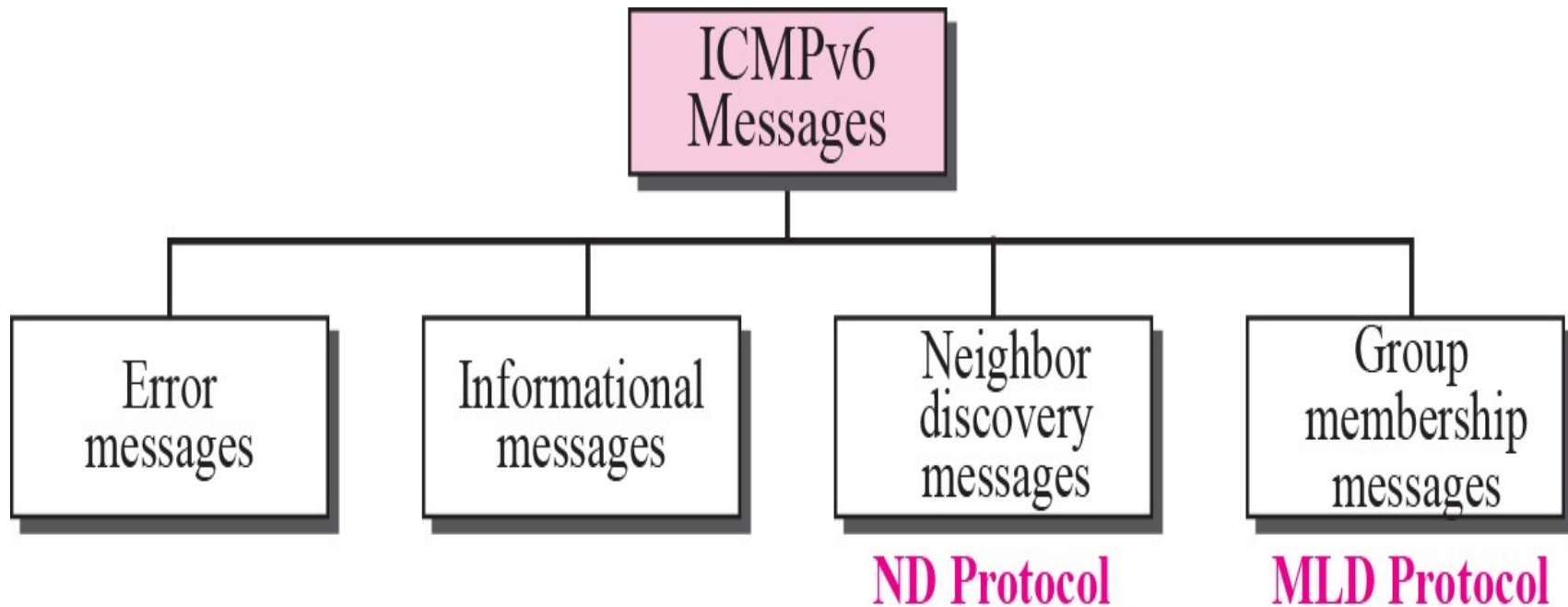
# Query Messages



# ICMP V6- INTRODUCTION

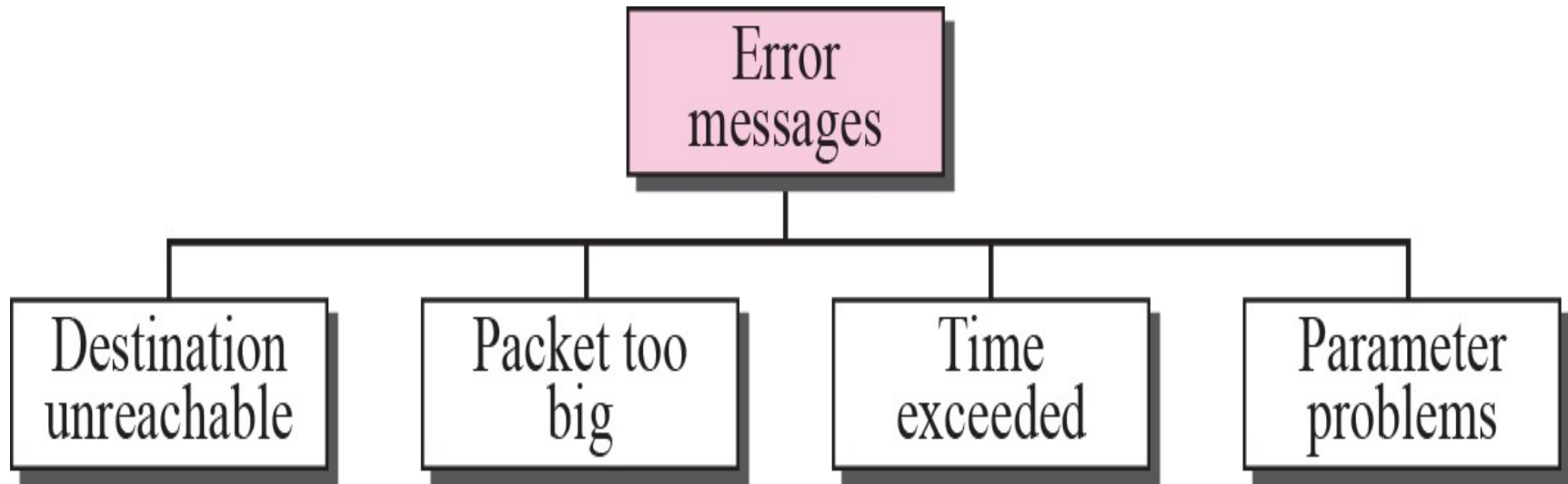
- ❑ Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP.
- ❑ This new version, Internet Control Message Protocol version 6 ( ICMPv6 ), follows the same strategy and purposes of version 4.
- ❑ ICMPv6, however, is more complicated than ICMPv4: some protocols that were independent in version 4 are now part of ICMPv6 and
- ❑ some new messages have been added to make it more useful.

# Taxonomy of ICMPv6 messages





# Error-reporting messages



# Informational Messages

- ❑ Two of the ICMPv6 messages can be categorized as informational messages: **echo request and echo reply messages.**
- ❑ The echo request and echo response messages are designed to check if two devices in the Internet can communicate with each other.
- ❑ A host or router can send an echo request message to another host; the receiving computer or router can reply using the echo response message.

# Neighbor-Discovery Messages

❑ The most important issue is the definition of two new protocols that clearly define the functionality of these group messages:

1. Neighbor-Discovery (ND) protocol

2. Inverse-Neighbor-Discovery (IND) protocol.

❑ These two protocols are used by nodes (hosts or routers) on the same link (network).

# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

**Routing Protocols: Distance Vector, Link State, Path Vector,**

**Routing in Internet: RIP ,OSPF, BGP,**

Congestion control and QoS,

MPLS,

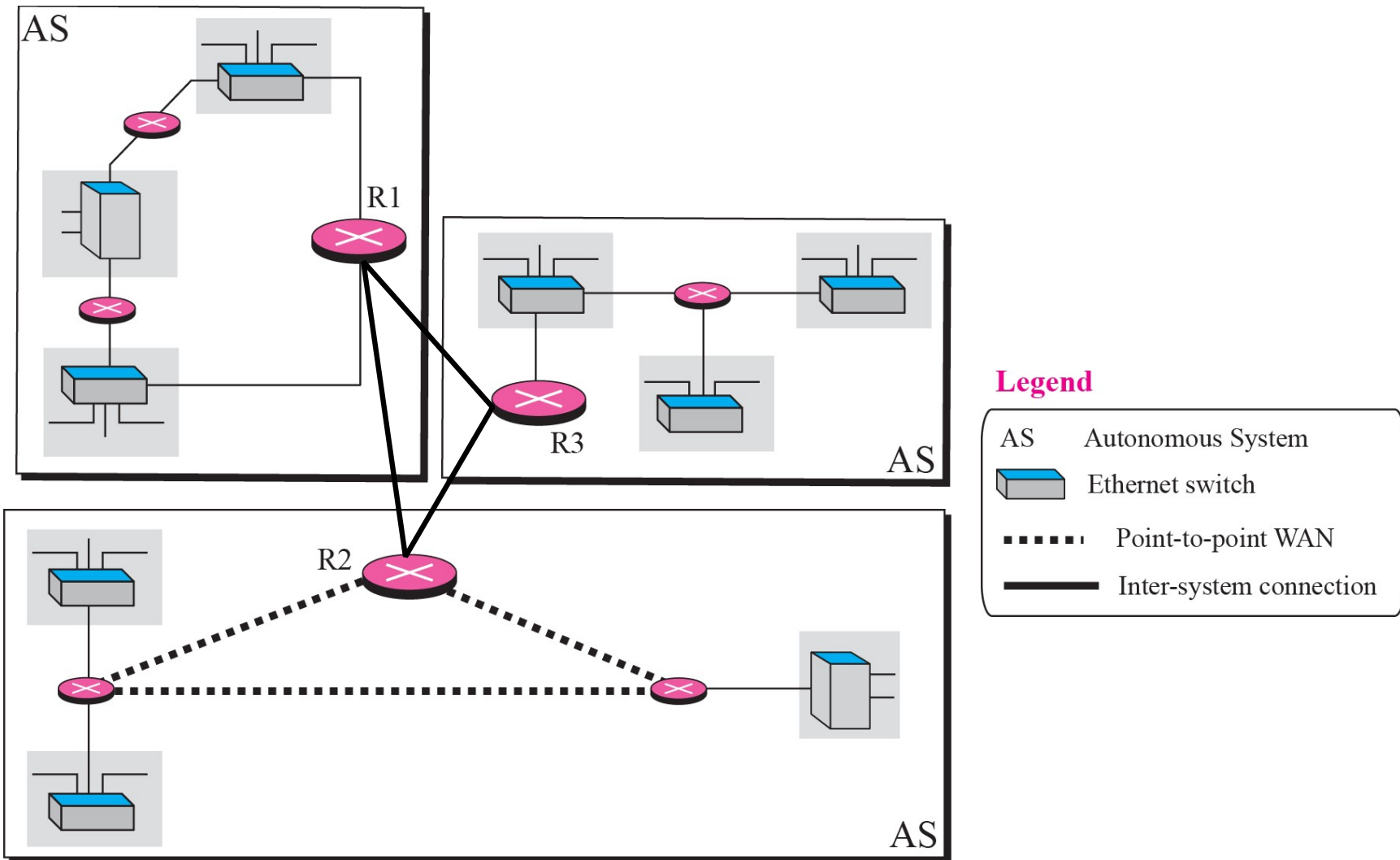
Mobile IP,

Routing in MANET : AODV, DSR

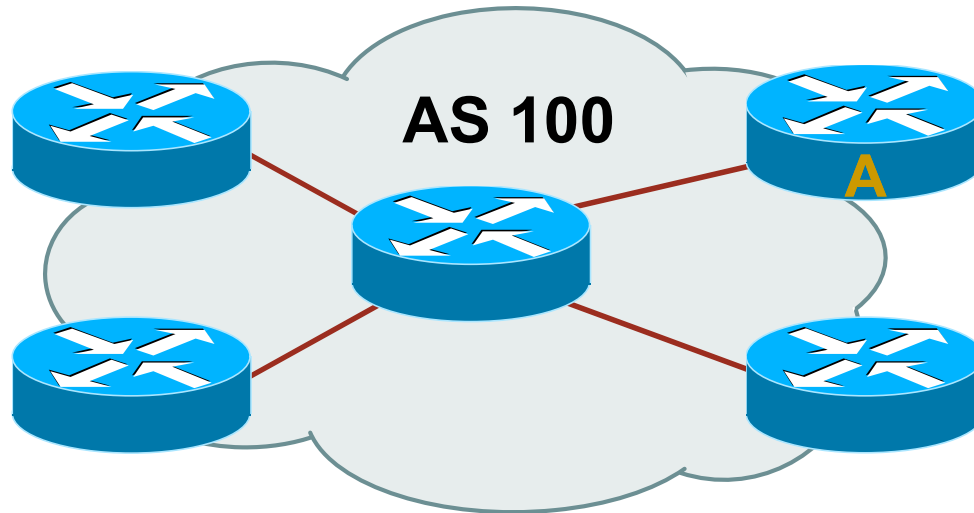
# INTER-AND INTRA-DOMAIN ROUTING

- Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.
- For this reason, an internet is divided into autonomous systems.
- An autonomous system (AS) is a group of networks and routers under the authority of a single administration.
- Routing inside an autonomous system is called intra-domain routing.
- Routing between autonomous systems is called inter-domain routing

Figure *Autonomous systems*

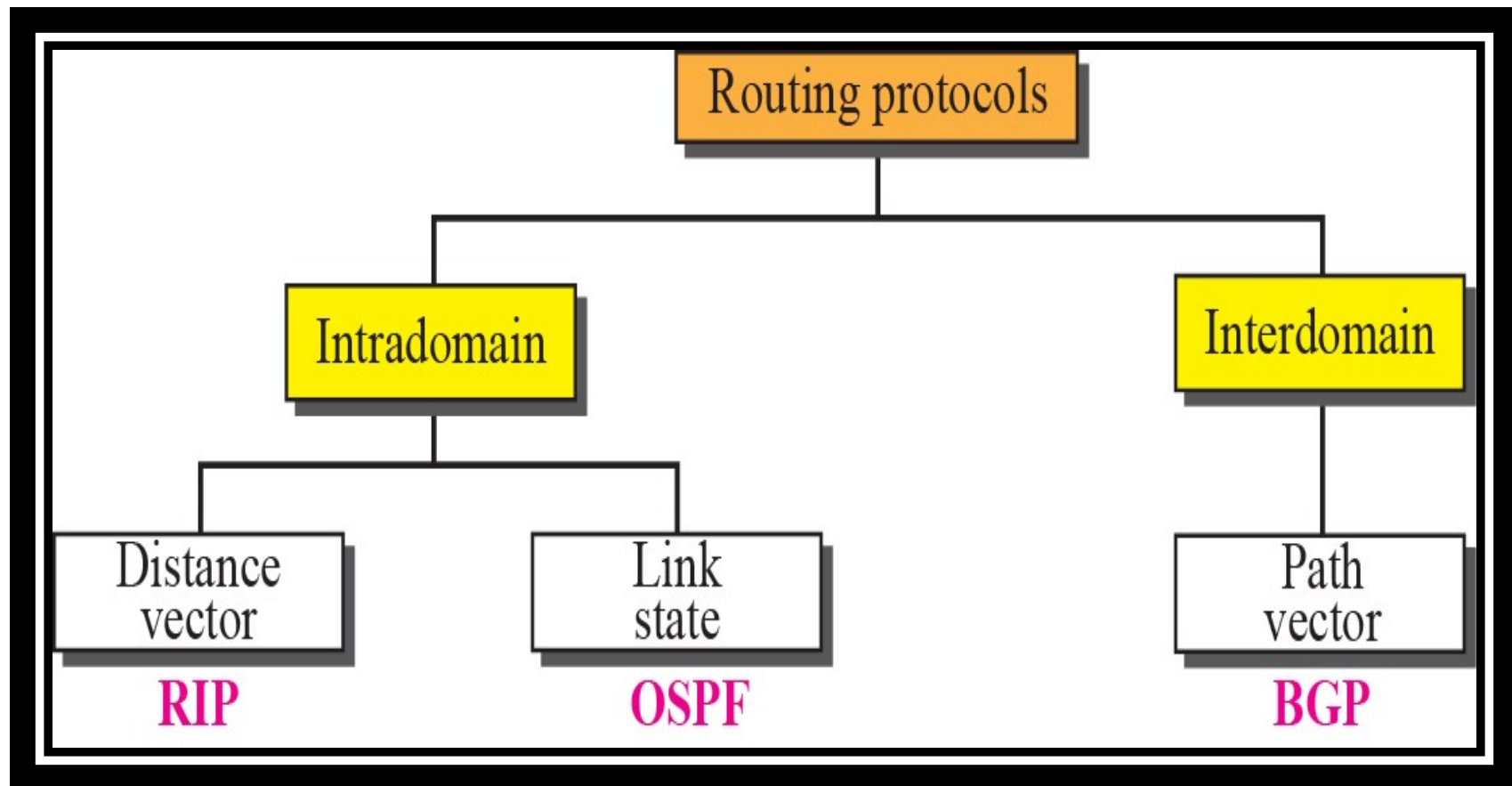


# Autonomous System (AS)



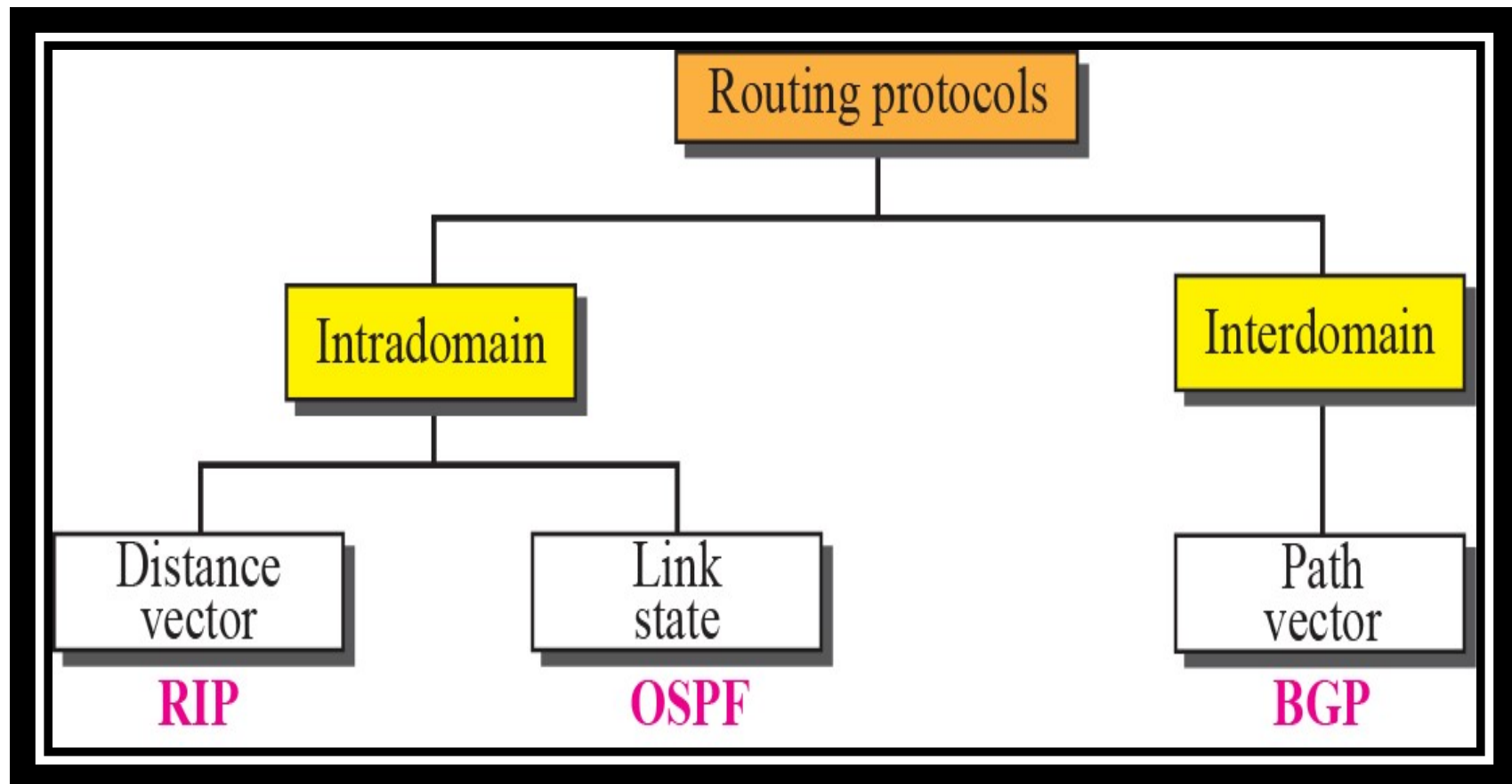
- Collection of networks with **same policy**
- **Single routing protocol**
- Usually under **single administrative control**

# Popular routing protocols





# Popular routing protocols



# DISTANCE VECTOR ROUTING

Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems.

An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

Routing inside an autonomous system is called intra-domain routing.

# Distance Vector Routing Working

- No node has complete information about the costs of all network links
- Gradual calculation of path by exchanging information with neighbors
- Each node constructs a one-dimensional array containing the “distances” or “costs” to all other nodes (as it relates to its knowledge) and distributes it to its immediate neighbors.
- Key thing -- each node knows the cost of links to its neighbors.
- If no link exists between two nodes, the cost of a direct link between the nodes is “infinity”.

# Distance Vector Routing

- The least-cost route between any two nodes is the route with **minimum distance**.
- Each node maintains a vector(table) of **minimum distances** to every node.
- The table at **each node also guides the packets** to the desired node by showing the next hop routing.

Example:

Assume each **node as the cities**.

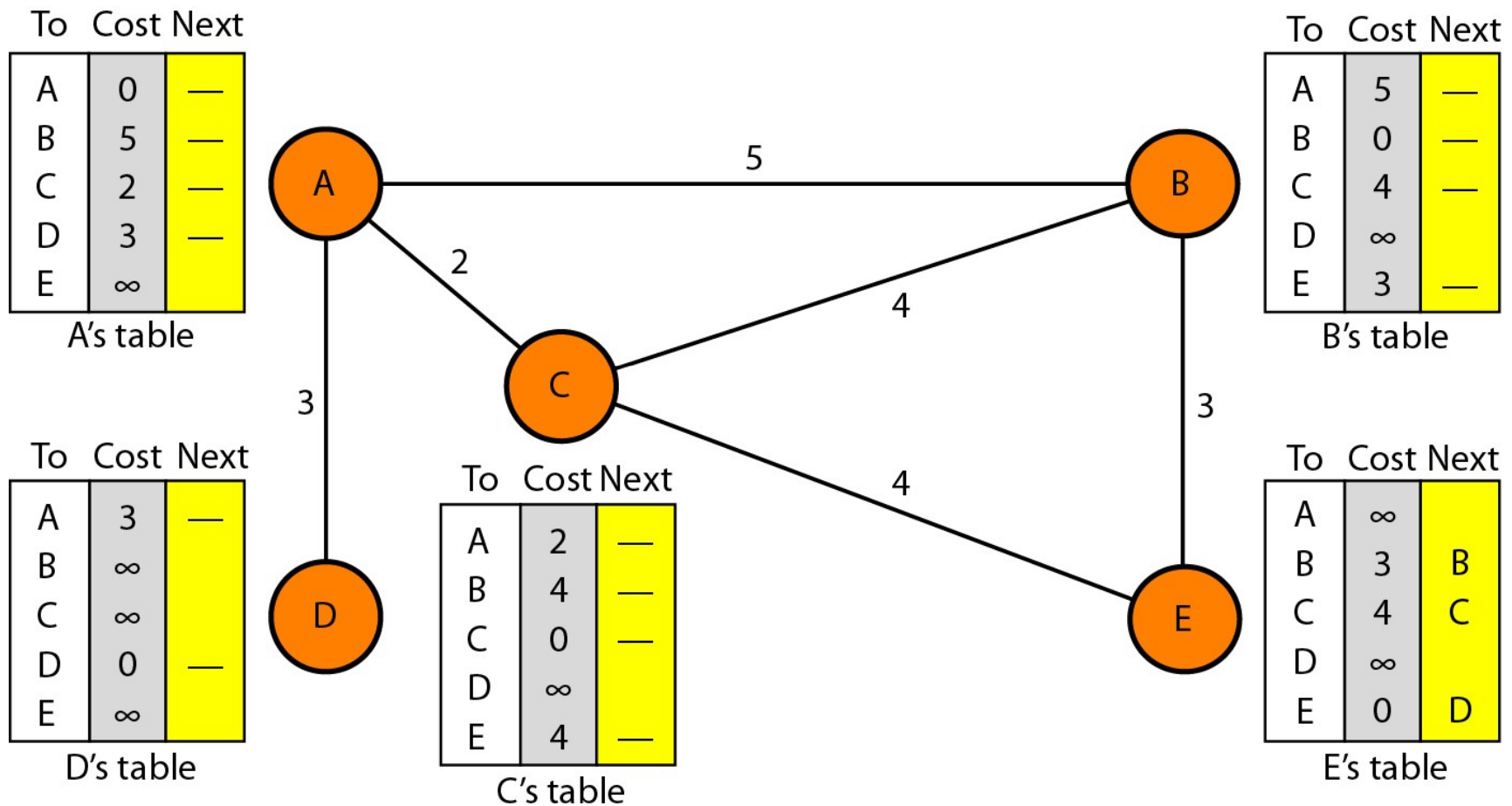
**Lines as the roads** connecting them.

# Distance Vector Routing-Initialization

At the beginning, each node know the cost of itself and its immediate neighbor

The distance of any entry that is not a neighbor is marked as infinite(unreachable).

# Distance Vector Routing-Initialization



# Distance Vector Routing-Sharing

- Idea is to share the information between neighbors.
- The node A does not know the distance about E, but node C does.
- If node C share it routing table with A, node A can also know how to reach node E.
- On the other hand, node C does not know how to reach node D, but node A does.
- If node A share its routing table with C, then node C can also know how to reach node D.
- Node A and C are immediate neighbors, can improve their routing tables if they help each other.

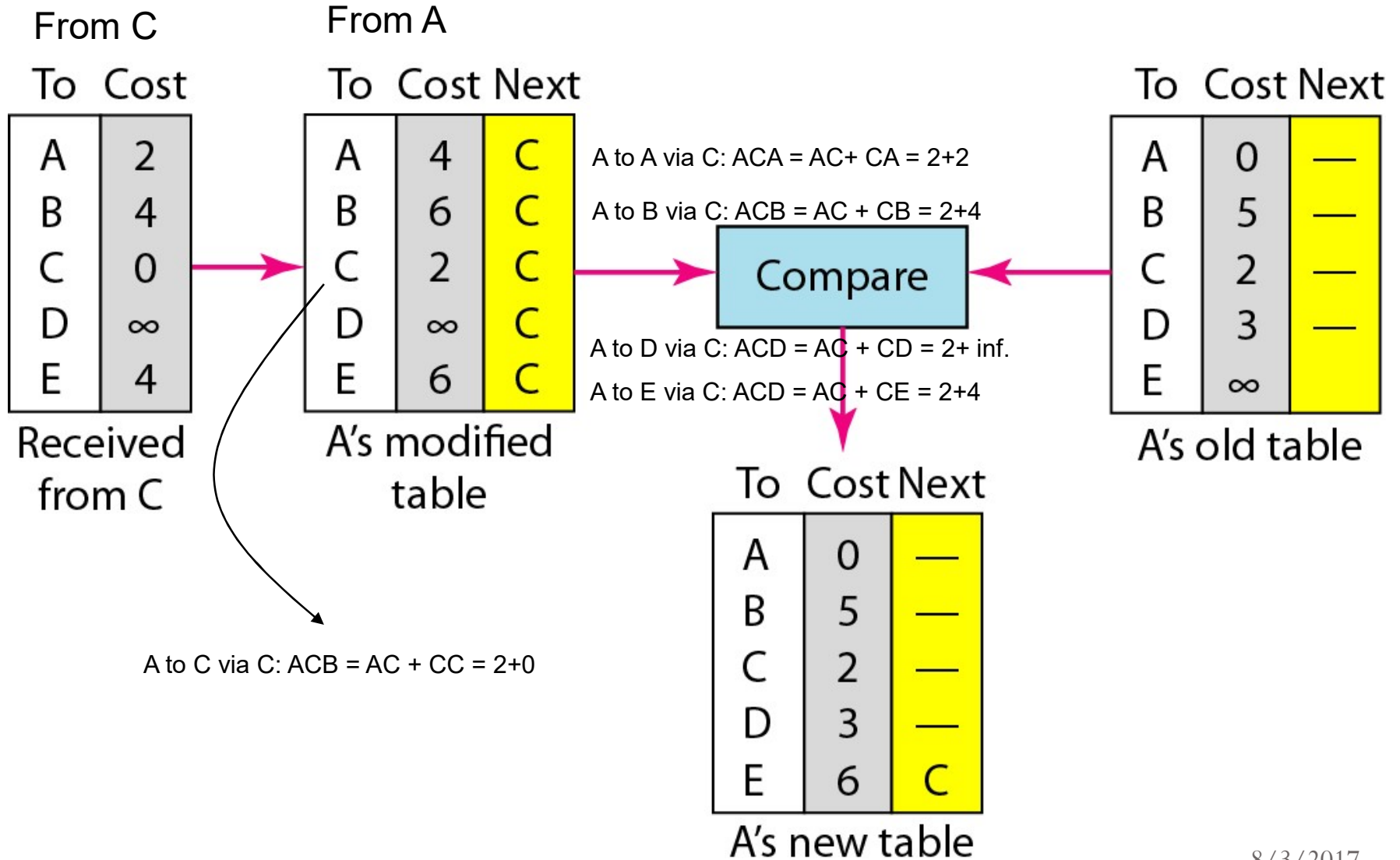
# Distance Vector Routing-Sharing

- How much of the table must be shared with each neighbor?
- The third column of the table(next hop) is not useful for the neighbor.
- When the neighbor receives a table, this column needs to be replaced with the **sender's name**.
- If any of the rows can be used, the next node column filled with sender of the table.
- Therefore, a node can send only the **first two column** of its table to any neighbor.



# Updating in distance vector routing

## example: C to A



# *Final Distance vector routing tables*

To Cost Next

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

A's table

To Cost Next

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

B's table

To Cost Next

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

D's table

To Cost Next

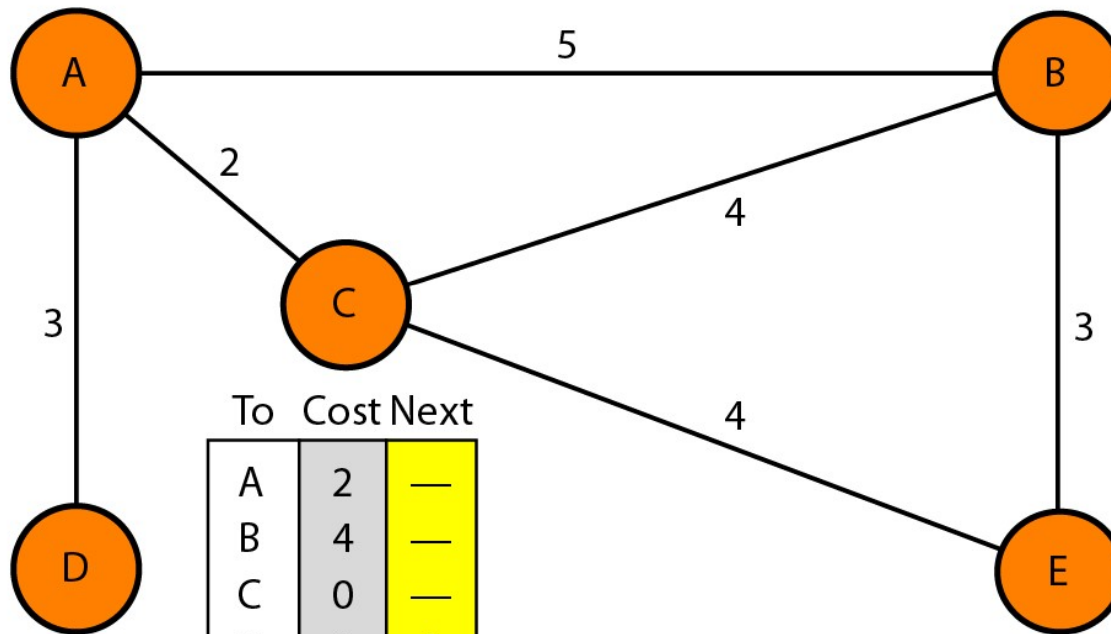
To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C's table

To Cost Next

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

E's table



# When to Share Routing table with neighbors

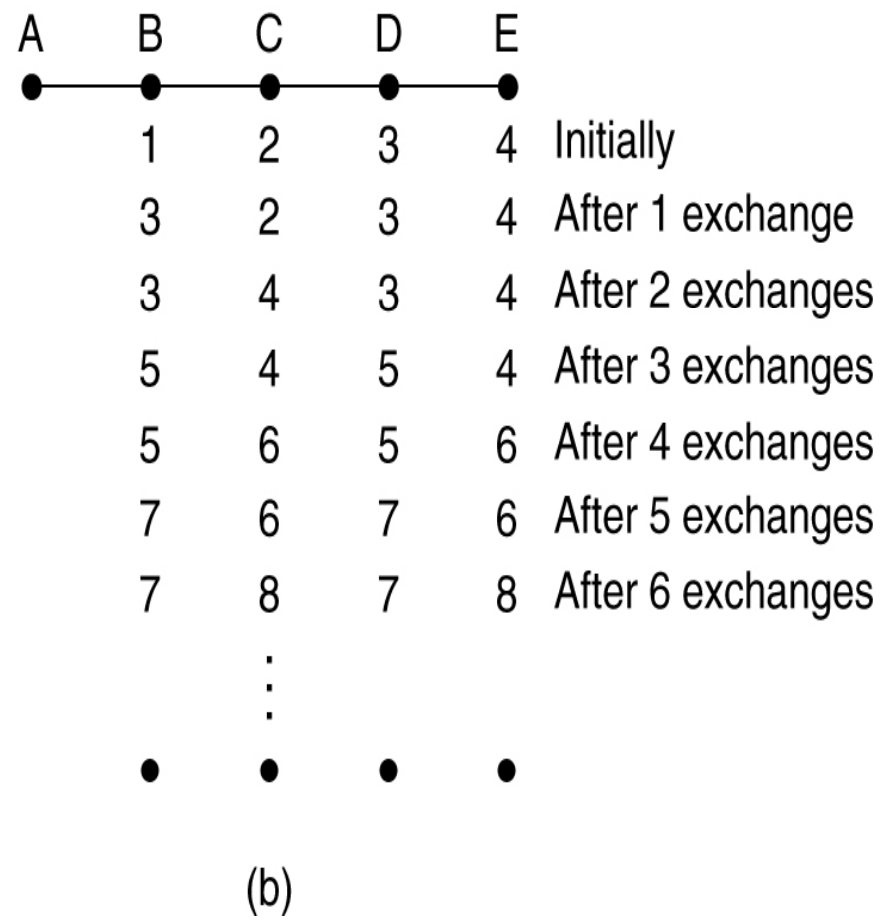
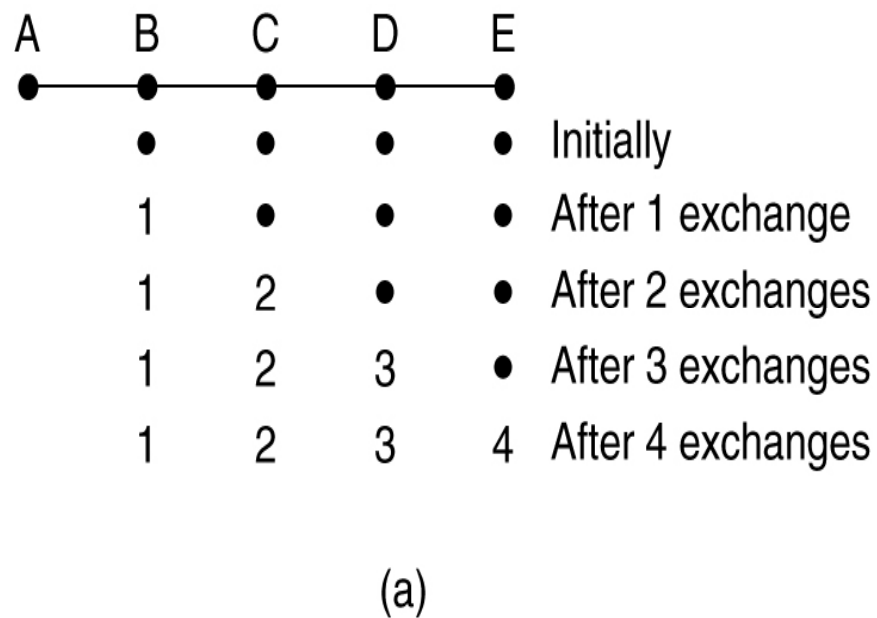
## Periodic Update

- A node sends its routing table, normally 30 seconds, in a periodic update

## Triggered Update

- A node sends its routing table to its neighbors any time when there is a change in its routing table
  - 1. After updating its routing table, or
  - 2. Detects some failure in the neighboring links

# Distance Vector Routing – The count-to-infinity problem.



## Distance Vector Routing – The count-to-infinity problem.

To see the problem clearly, imagine a subnet connected like A–B–C–D–E, and let the metric between the routers be "number of jumps(Hops)".

Now suppose that A is taken offline.

In the vector-update-process B notices that the route to A, which was distance 1, is down – B does not receive the vector update from A.

# Distance Vector Routing –

## The count-to-infinity problem cont....

The problem is, B also gets an update from C, and C is still not aware of the fact that A is down – so it tells B that A is only two jumps from C (C to B to A), which is false.

Since B doesn't know that the path from C to A is through itself (B), it updates its table with the new value "B to A = 2 + 1".

Later on, B forwards the update to C and due to the fact that A is reachable through B (From C point of view), C decides to update its table to "C to A = 3 + 1".

This slowly propagates through the network until it reaches to infinity (hop 16)

# RIP- Routing Information Protocol

The Routing Information Protocol (RIP) is an **intra-domain** (interior) routing protocol used inside an autonomous system.

It is a very simple protocol **based on distance vector routing.**

In the Internet, goal of routers is to learn how to forward packets to various networks.

# Routing Information Protocol (RIP)

RIP treats all network equals; the cost of passing thru a network is the same: one hop count per network.

Each router/node maintains a vector (table) of minimum distances to every node.

The hop-count is the number of networks that a packet encounters to reach its destination. Path costs are based on number of hops.

In distance vector routing, each **router periodically shares its table** with its neighbour.

Each router keeps a routing table that has one entry for each destination network . The entry consists of **Destination Network Address, Hop-Count and Next-Router.**



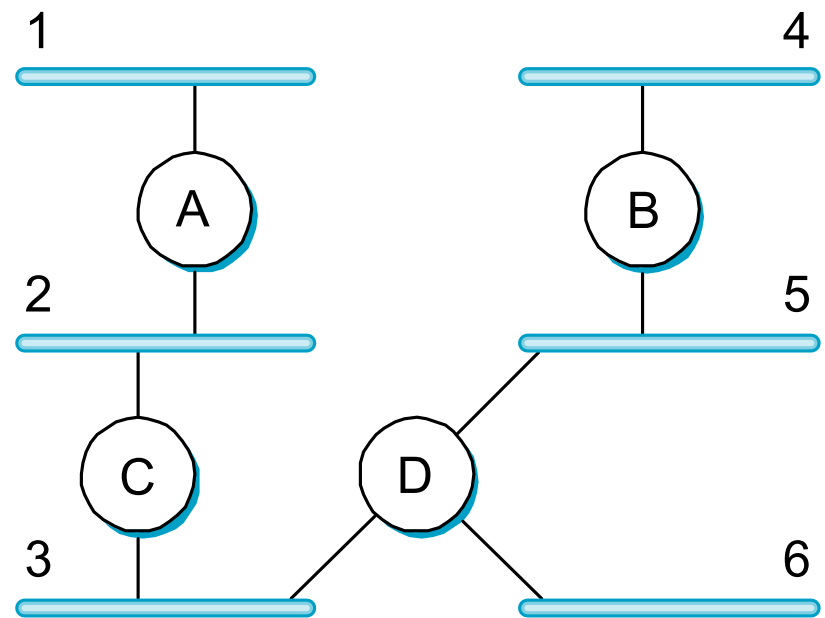
# An Example of RIP

Routers advertise the cost of reaching networks.

In this example, C's update to A would indicate that C can reach Networks 2 and 3 with cost 0,

Networks 5 and 6 with cost 1

and Network 4 with cost 2.



# RIP messages

## Request

- A request message is sent by a router that has just come up

## Response

- A response can be within 30s or when there is a change in the routing table

# RIP Timers

## Periodic timer

- Routing tables are exchanged every 30 seconds using the RIP advertisement.

## Expiration timer

- If a router does not hear from its neighbor once every 180 seconds, the neighbor is deemed unreachable.

# LINK STATE ROUTING

Link-state routers exchange messages to allow each router to learn the entire network topology.

Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation [[Dijkstra1959](#)].

# Link State Routing Algorithm Steps

Discover its **neighbors**, learn their network address.

Measure the **delay or cost** to each of its neighbors.

**Construct a packet** telling all it has just learned.

**Send this packet** to all other routers.

Compute the **shortest path** to every other router.

# Measure the **delay or cost**

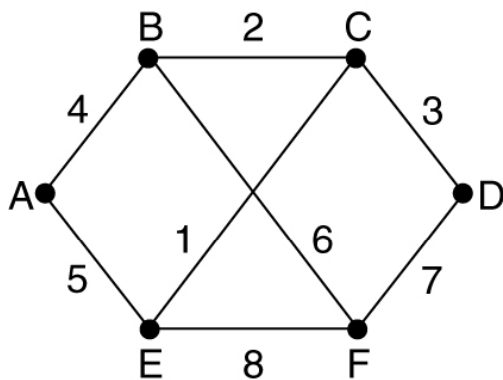
Echo packets are used to measure the line cost

Calculate total time used to echo packet

$t = \text{Arrival time} - \text{Departure time}$

Then  $t/2$  gives cost(time) of line

# Construct Link State packet



(a)

		Link		State		D		E		F	
A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

(a) A subnet. (b) The link state packets for this subnet.

**Send this packet** to all other routers  
and compute the **shortest path**

**Flood the LSP** in subnet and then by  
using Shortest path algorithm  
(**Dijkstra's algorithm**) compute the  
shortest path for each router



# Distance Vector Routing Vs Link State Routing (DVR Vs LSR)

Distance Vector Routing	Link State Routing
used in small networks	used in larger networks
it has a limited number of hops.	it has unlimited number of hops
high convergence time	convergence time is low.
periodically advertise updates	only new changes in a network.
It has loop problem	No loop problem
Updates are broadcasted	Updates are multicasted
advertises only the directly connected routers and full routing tables,	advertise the updates, and flood the advertisement.
Eg. <b>RIP ,IGRP , BGP .</b>	Eg. : <b>OSPF , IS-IS</b>

# OSPF- Open Shortest Path First

The Open Shortest Path First (OSPF) protocol is an intra-domain routing protocol based on link state routing.

Its domain is also an autonomous system.

Support variety of distance metrics

Dynamic algorithm that adapted to changes in the topology automatically and quickly

## OSPF- Open Shortest Path First

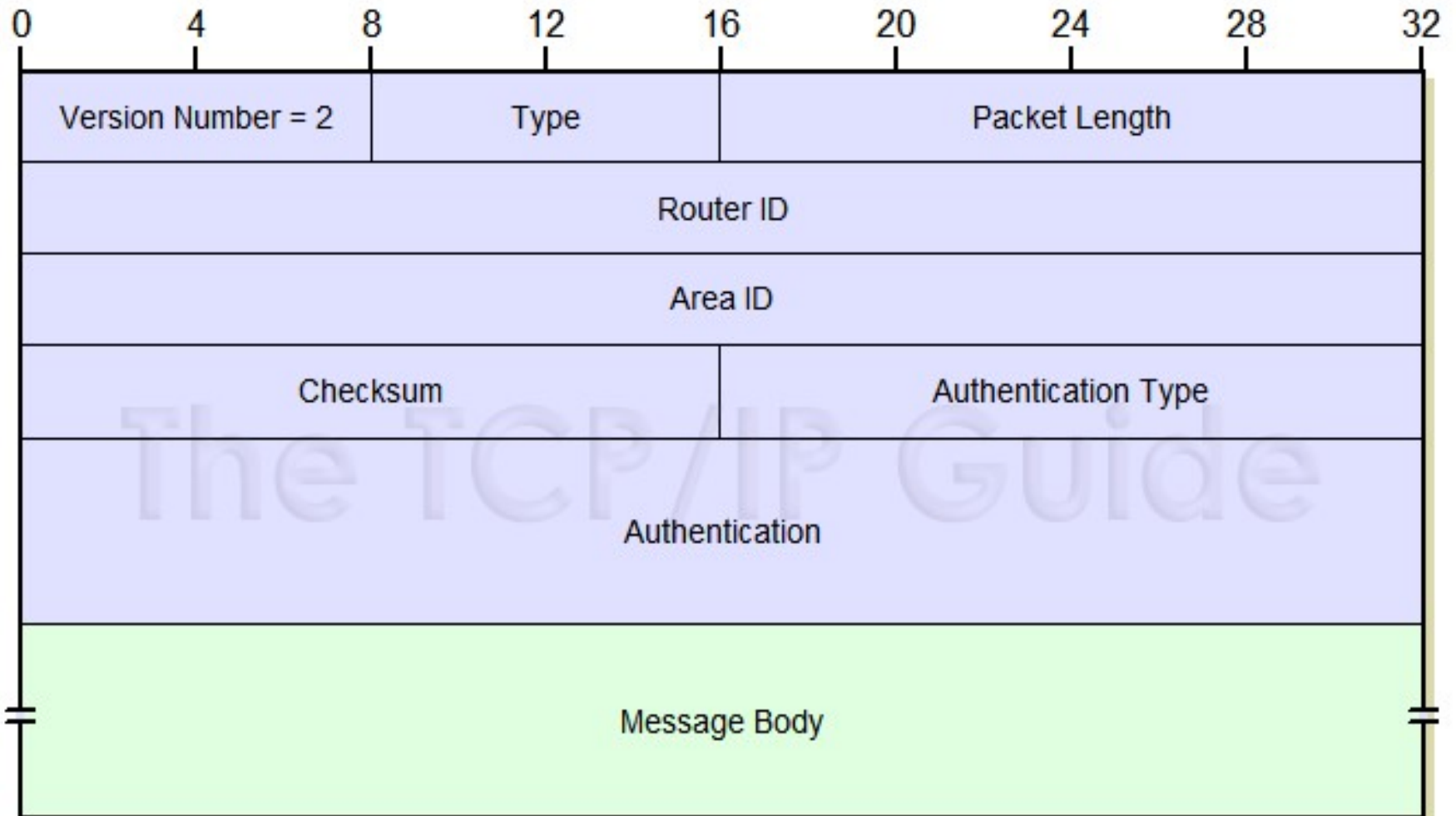
Support routing based on type of service

Do load balancing, splitting the load over multiple lines

Prevent spoofing ie better security provision

Provision for dealing with routers that were connected to the internet via a tunnel

# OSPF Header Format



# OSPF- Open Shortest Path First

➤ OSPF divides AS into **areas**.

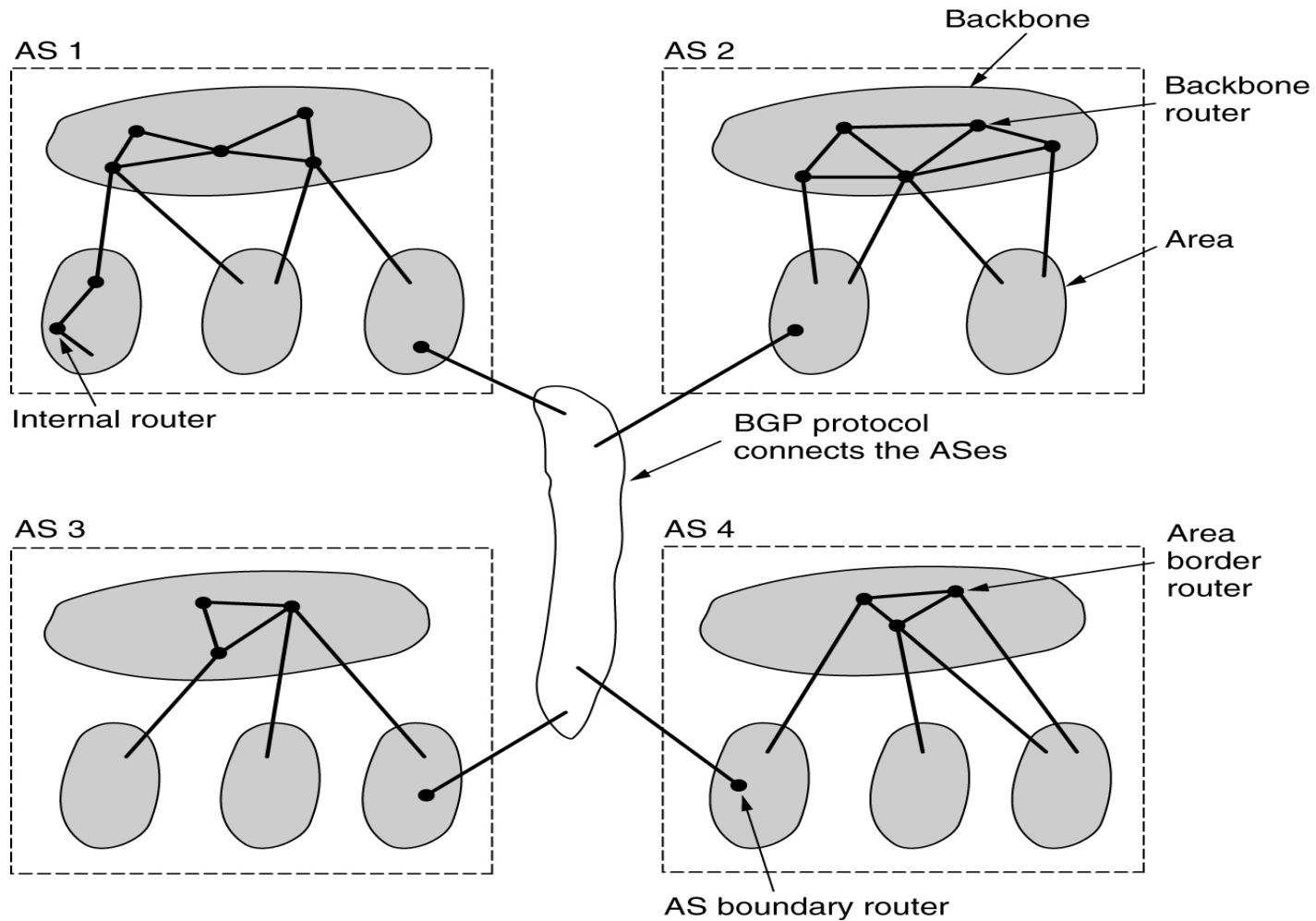
Every AS has a backbone area called **area 0**

All areas are connected to backbone areas

➤ **OSPF has four classes of router**

1. Internal routers -wholly within on area
2. Area border routers -connect two or more areas
3. Backbone routers -On the backbone area
4. AS boundary routers -Talk to other routers in  
other AS

# OSPF- Open Shortest Path First



The relation between ASes, backbones, and areas in OSPF.

# OSPF - **WORKING**

When a router starts, it first initializes the routing protocol

It then uses the OSPF's handshaking **Hello Protocol** to learn about each other.

The routers exchange information describing their knowledge of the routing domain. This information is called database description and is placed in **LSA** messages.

# OSPF - WORKING

Using the above LSA messages the receiving router knows if its LSD is consistent with its peer's databases. If all is consistent the neighbor is now defined as fully adjacent.

A router periodically advertises its state (link state) to detect dead routers in a timely fashion.

From this database each router calculates a shortest path tree with itself the root.

This shortest path tree in turn yields a routing table for the protocol.



# OSPF- Routing protocol packets

**Hello packet:** It is used to discover and maintain neighbor relationships.

**Data Description packet and Link State Request packets:** They are used in forming adjacencies.

**Link State Update and Link State Acknowledgment packets:** Used for reliable update mechanisms.

# OSPF- Databases

**Neighbor Database:** Initial table displaying neighbors learned through Hello packets.

**Link State Database:** Similar in all routers. Formed after each router floods its neighbor database.

**Routing table:** Each router builds this table by using SPF technology. It gives the shortest path to all the routers in the AS.

# BGP- Border Gateway Protocol

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing.

The Border Gateway Protocol makes routing decisions based on paths, network policies or rule-sets configured by a network administrator,

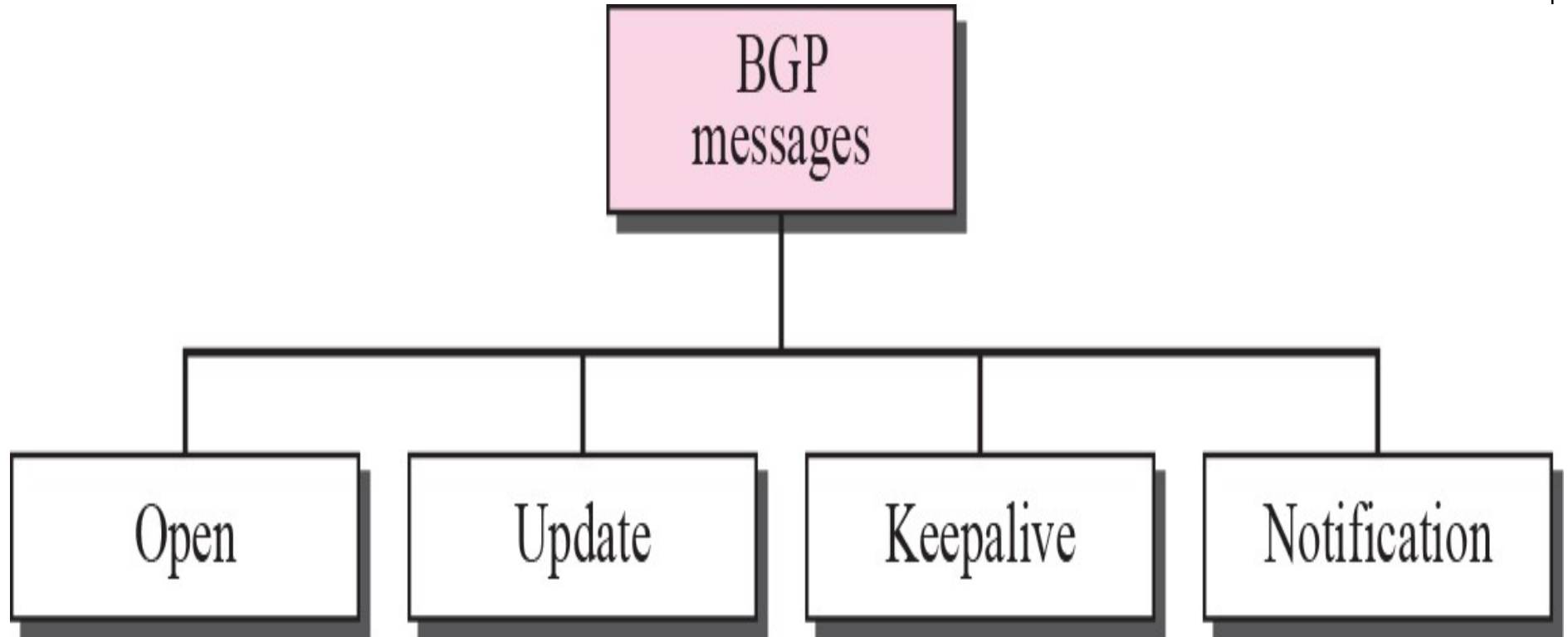
# BGP- Border Gateway Protocol

When BGP runs between two peers in the same autonomous system (AS), it is referred to as ***Internal BGP (iBGP or Interior Border Gateway Protocol)***.

When it runs between different autonomous systems, it is called ***External BGP (EBGP or Exterior Border Gateway Protocol)***.

Routers on the boundary of one AS exchanging information with another AS are called *border or edge routers*. BGP uses the services of TCP on port 179.

# Types of BGP messages



# BGP Messages

## Open

- Announces AS ID
- Determines hold timer – interval between keep\_alive or update messages, zero interval implies no keep\_alive

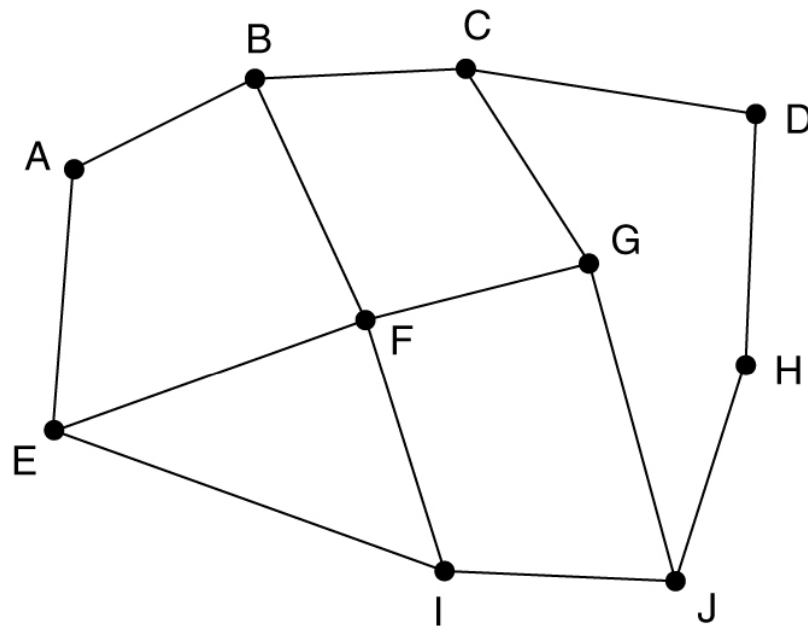
## Keep\_alive

- Sent periodically to peers to ensure connectivity.

## Notification

- Used for error notification
- TCP connection is closed *immediately* after notification

# BGP – Example



(a)

Information F receives  
from its neighbors about D

From B: "I use BCD"  
From G: "I use GCD"  
From I: "I use IFGCD"  
From E: "I use EFGCD"

(b)

(a) A set of BGP routers.      (b) Information sent to F.

# Path Attributes

## ORIGIN

- The source of the routing information (RIP, OSPF, etc)

## AS\_PATH

- The list of ASs through which the destination can be reached

## NEXT-HOP

- The next router to which the data packet should be sent



# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

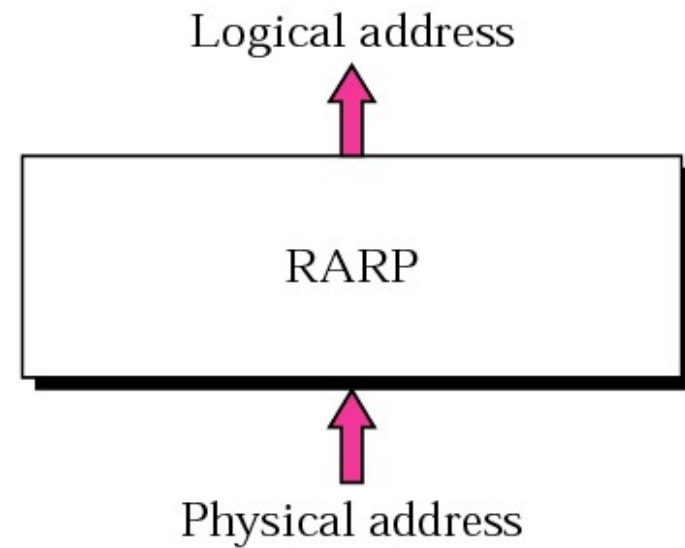
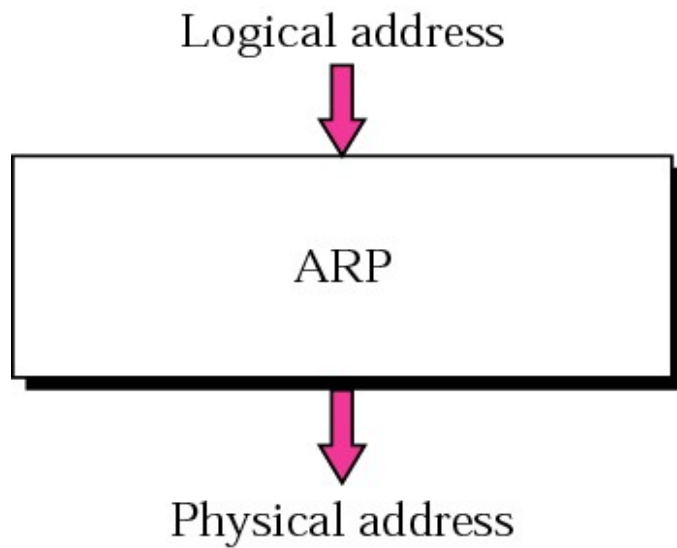
**ARP and RARP**

MPLS,

Mobile IP,

Routing in MANET : AODV, DSR

# ARP and RARP



# ARP (Address Resolution Protocol)

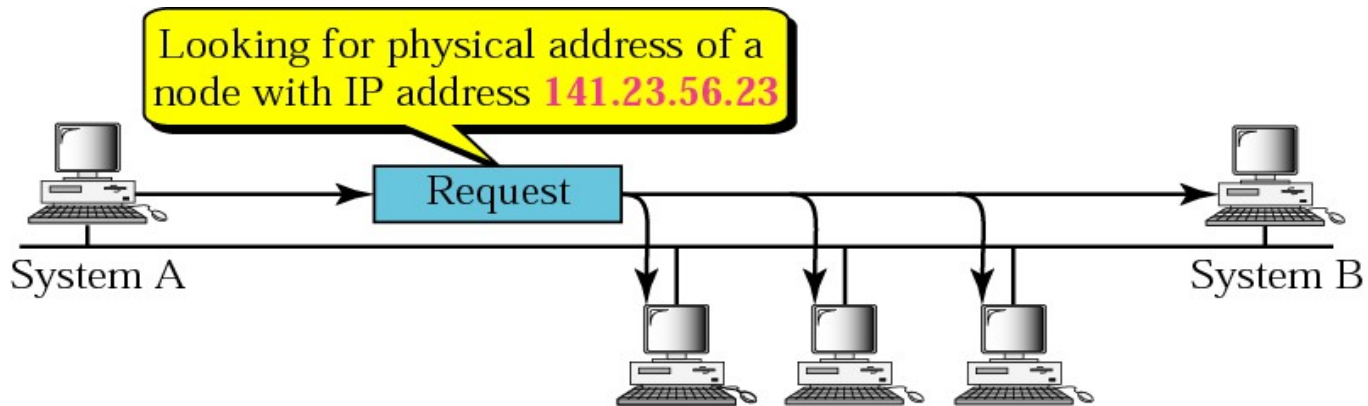
ARP associates an IP address with its physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC.

The delivery of a packet to a host or a router requires two levels of addressing: logical and physical.

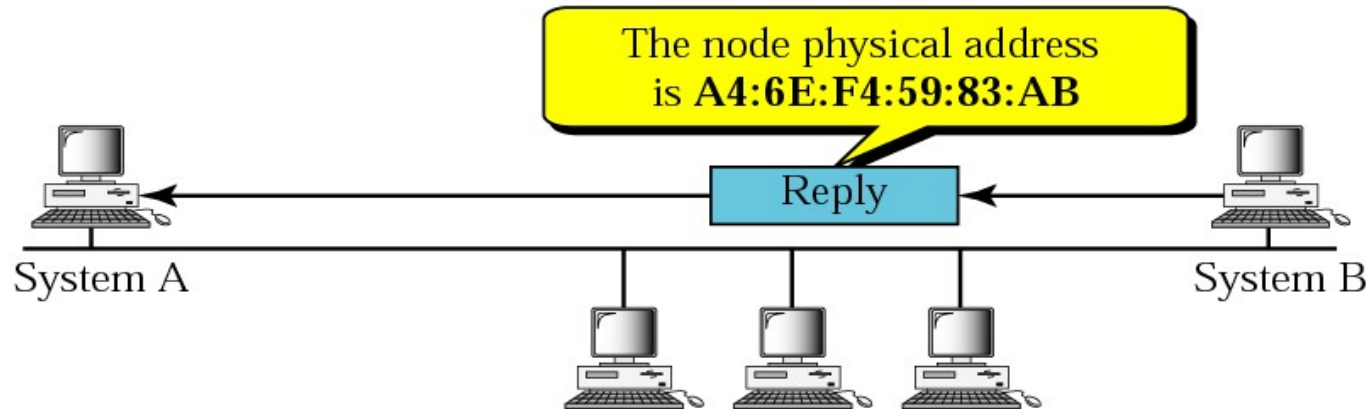
We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done using either static or dynamic mapping.

Logical address to physical address translation can be done statically (not practical) or dynamically (with ARP).

# ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

# ARP packet

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

*Note*

*An ARP request is broadcast;  
an ARP reply is unicast.*

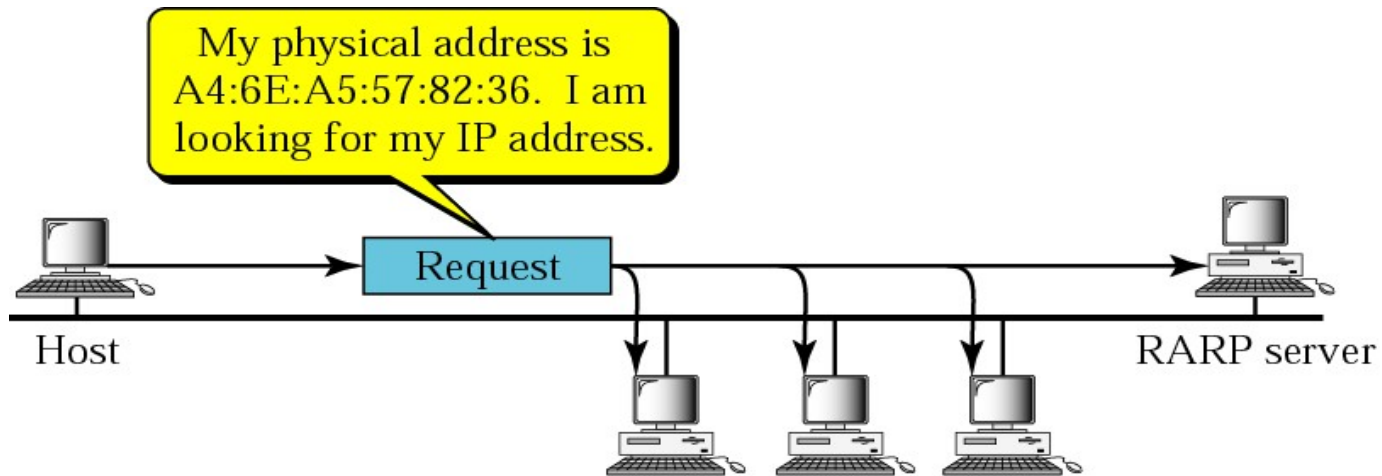
# RARP (Reverse Address resolution Protocol)

RARP finds the logical address for a machine that only knows its physical address. RARP requests are broadcast, RARP replies are unicast.

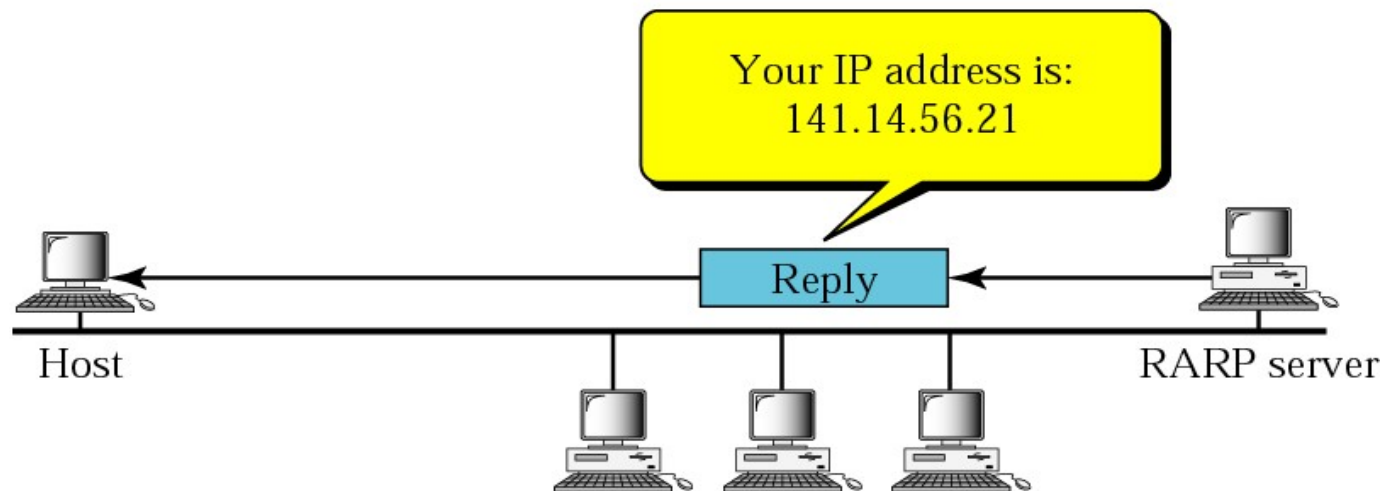
This is often encountered on thin-client workstations. No disk, so when machine is booted, it needs to know its IP address (don't want to burn the IP address into the ROM).

If a thin-client workstation needs to know its IP address, it probably also needs to know its subnet mask, router address, DNS address, etc. So we need something more than RARP. BOOTP, and now DHCP have replaced RARP.

# RARP operation



a. RARP request is broadcast



b. RARP reply is unicast



# RARP packet

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

ARP and RARP

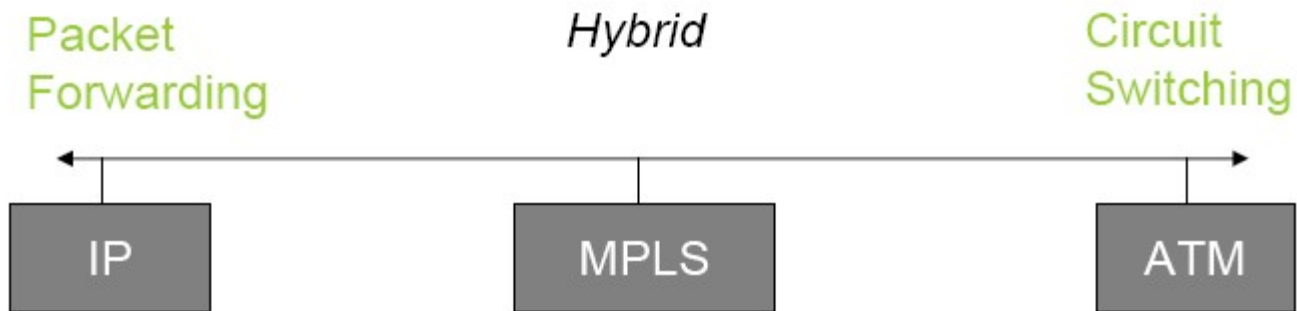
**MPLS,**

Mobile IP,

Routing in MANET : AODV, DSR

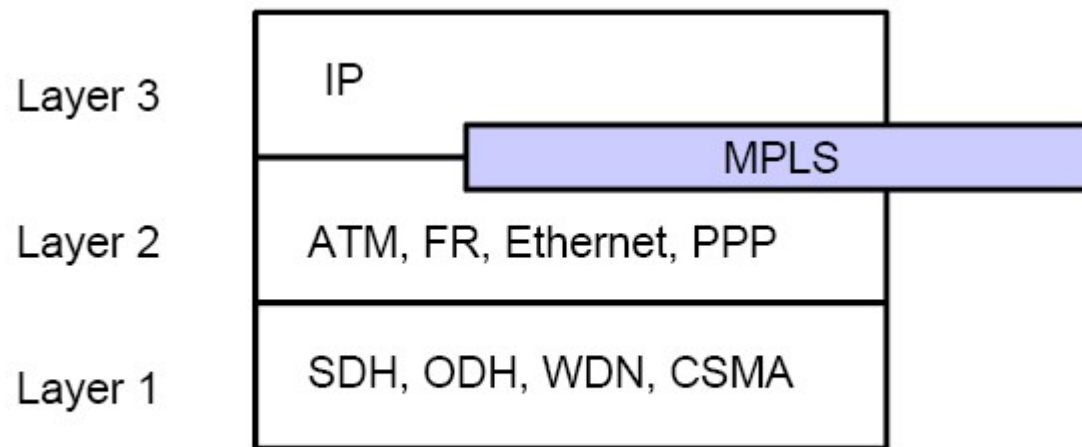
# Motivation

- Idea: Combine the forwarding algorithm used in ATM with IP.



# MPLS Basics

- Multi Protocol Label Switching is arranged between Layer 2 and Layer 3

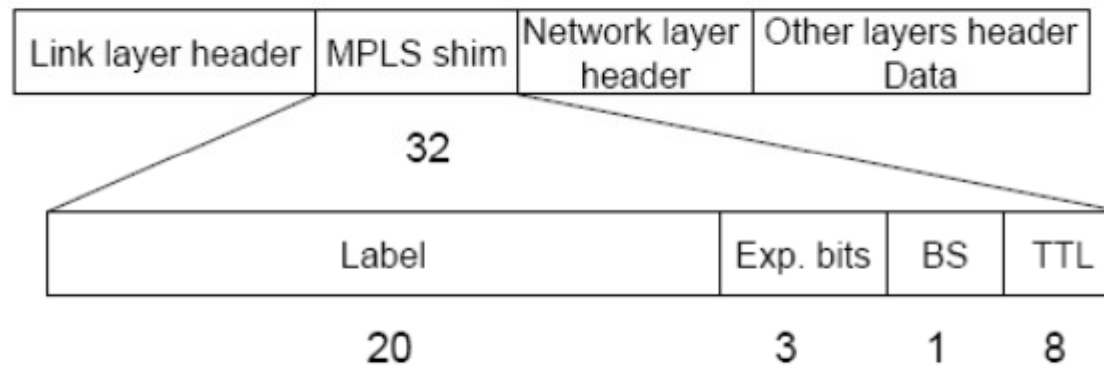


# MPLS Basics (cont.)

- MPLS Characteristics
  - Mechanisms to manage traffic flows of various granularities (*Flow Management*)
  - Is independent of Layer-2 and Layer-3 protocols
  - Maps IP-addresses to fixed length labels
  - Supports ATM, Frame-Relay and Ethernet

# Label

- Generic label format



Exp.bits: Experimental Bits, often used for Class of Service

BS: Bottom of Stack bit, is set if no label follows

TTL: Time To Leave, used in the same way like in IP

# MPLS Routers

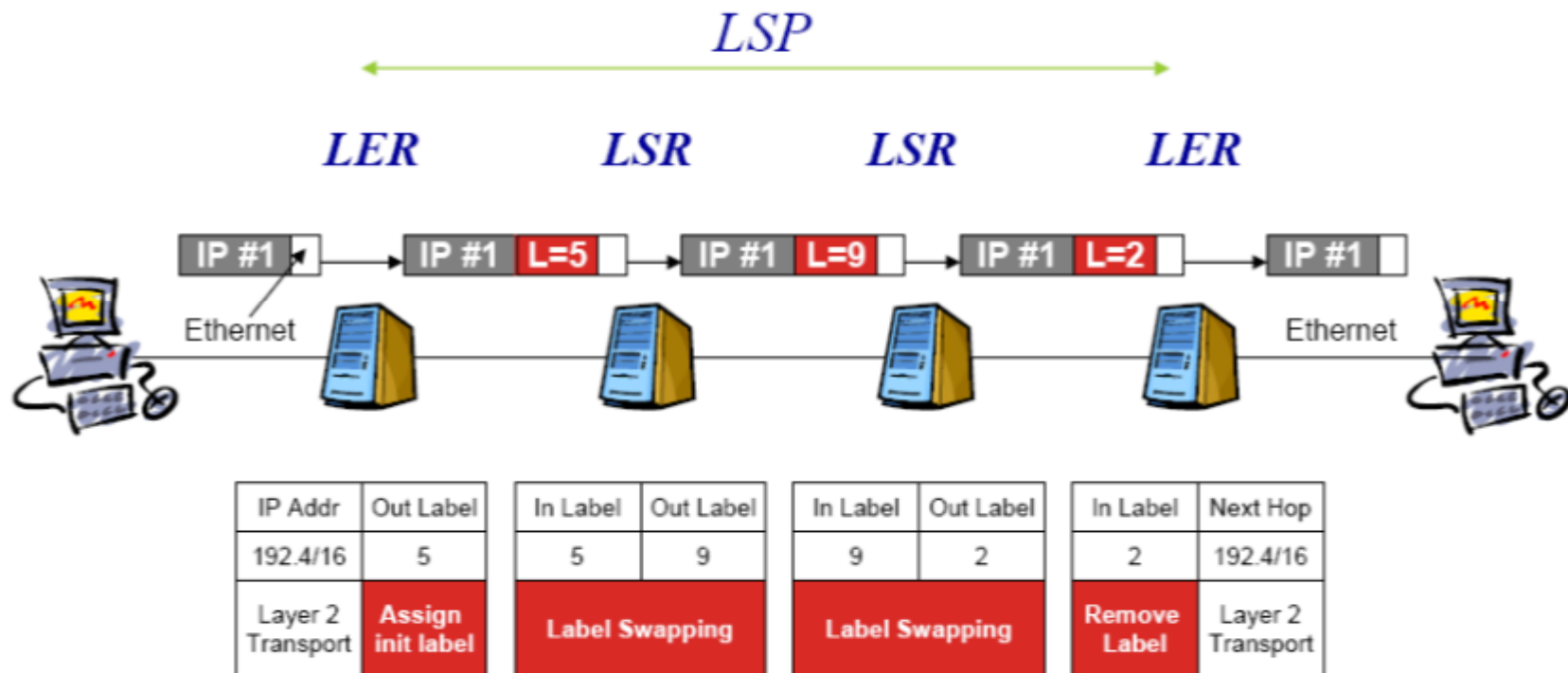
- **Label Edge Router - LER**

- Resides at the edge of an MPLS network and assigns and removes the labels from the packets.
- Support multiple ports connected to dissimilar networks (such as frame relay, ATM, and Ethernet).

- **Label Switching Router – LSR**

- Is a high speed router in the core on an MPLS network.
- ATM switches can be used as LSRs without changing their hardware. Label switching is equivalent to VP/VC switching.

# Positions of LERs & LSRs



“ROUTE AT EDGE, SWITCH IN CORE”



# MPLS Advantages & Disadvantages

- Advantages
  - Improves packet-forwarding performance in the network
  - Supports QoS and CoS for service differentiation
  - Supports network scalability
  - Integrates IP and ATM in the network
  - Builds interoperable networks
- Disad.
  - An additional layer is added
  - The router has to understand MPLS

# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

Congestion control and QoS,

MPLS,

**Mobile IP,**

Routing in MANET : AODV, DSR

# Mobile IP

- **Developed as a means for transparently dealing with problems of mobile users**
- Enables hosts to stay connected to the Internet regardless of their location and without changing IP addresses
- Requires no changes to software of non-mobile hosts/routers
- **Requires addition of some infrastructure**
- Has no geographical limitations
- Requires no modifications to IP addresses
- Supports security
- IETF standardization process is still underway

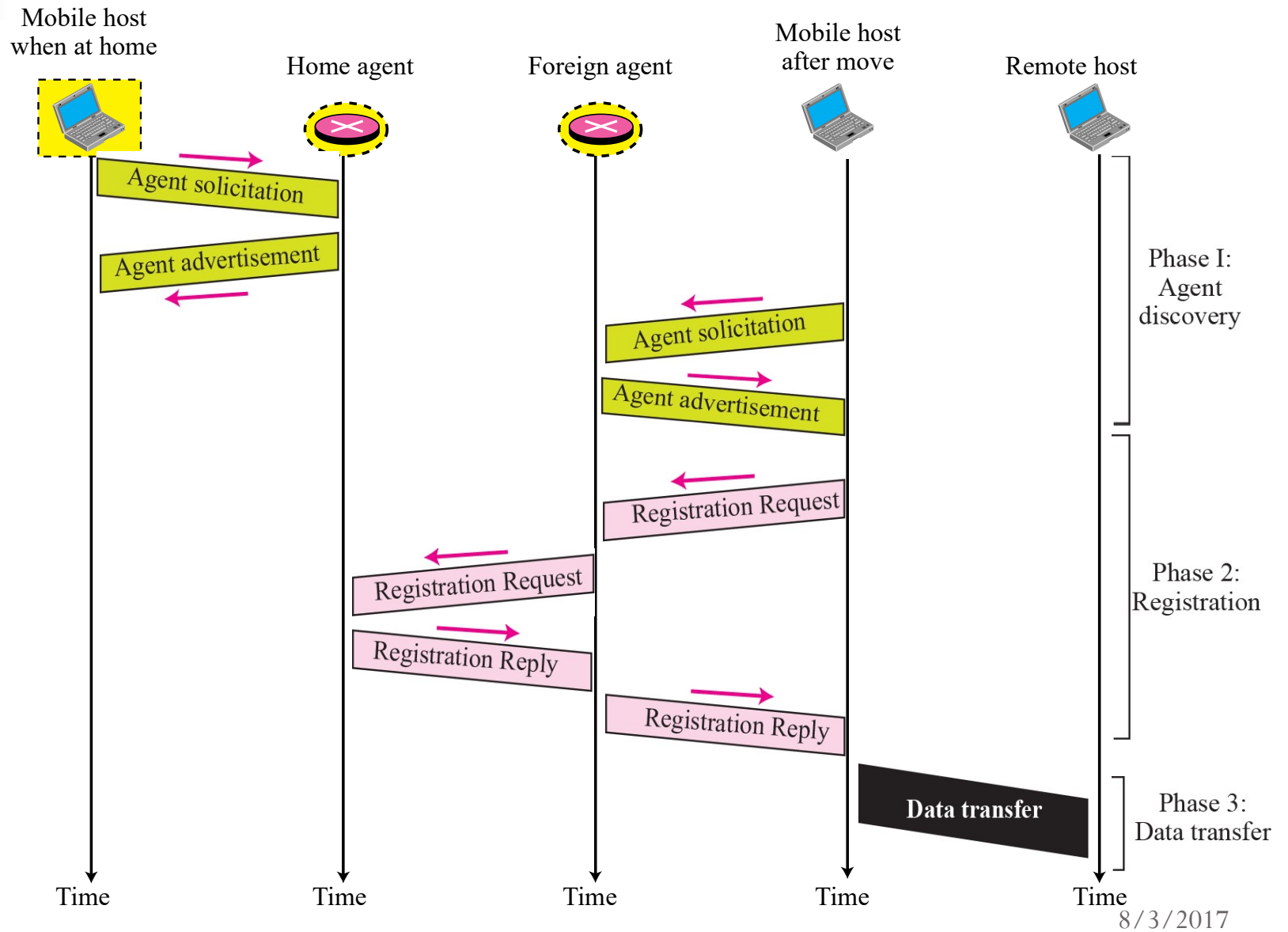
# Mobile IP Entities

- Mobile Node (MN)
  - The entity that moves from network to network
  - Assigned a permanent IP called its *home address* to which other hosts send packets regardless of MN's location
- Home Agent (HA)
  - Router with additional functionality
  - Located on home network of MN
  - Mobility binding of MN's IP with its *Care of Address (COA)*
  - Forwards packets to appropriate network when MN is away – uses encapsulation

# Mobile IP Entities contd.

- Foreign Agent (FA)
  - Another router with enhanced functionality
  - Used to send/receive data between MN and HA
  - Advertises itself periodically
- Care-of-address (COA)
  - Address which identifies MN's current location
  - Sent by FA to HA when MN attaches
  - Usually the IP address of the FA
- Correspondent Node (CN)
  - End host to which MN is corresponding (eg. a web server)

Figure Remote host and mobile host configuration



# Mobile IP Support Services

- Agent Discovery
  - HA's and FA's broadcast their presence on each network to which they are attached
  - MN's listen for advertisement and then initiate registration
- Registration
  - When MN is away, it registers its COA with its HA, via FA
  - Registration control messages sent via UDP to well known port
- Encapsulation/decapsulation – just like standard IP only with COA

# Mobile IP Operation

- A MN listens for agent advertisement and then initiates registration
  - If responding agent is the HA, then mobile IP is not necessary
- After receiving the registration request from a MN, the HA acknowledges and registration is complete
  - Registration happens as often as MN changes networks
- HA intercepts all packets destined for MN
  - This is simple unless sending application is on or near the same network as the MN
  - HA masquerades as MN
  - There is a specific lifetime for service before a MN must re-register
  - There is also a de-registration process with HA if an MN returns home



# Outline

Switching techniques,

IP Protocol,

IPv4 and IPv6 addressing schemes,

Subnetting,

NAT, CIDR,

ICMP,

Routing Protocols: Distance Vector, Link State, Path Vector,

Routing in Internet: RIP ,OSPF, BGP,

Congestion control and QoS,

MPLS,

Mobile IP,

**Routing in MANET : AODV, DSR**

# Routing in MANET

# Unicast Routing Protocols

- Many protocols have been proposed
- Some specifically invented for MANET
- Others adapted from protocols for wired networks
- No single protocol works well in all environments
  - some attempts made to develop adaptive/hybrid protocols
- Standardization efforts in IETF
  - MANET, MobileIP working groups
  - <http://www.ietf.org>

# Routing Protocols

## Proactive protocols

- Traditional distributed shortest-path protocols
- Maintain routes between every host pair at all times
- Based on periodic updates; High routing overhead
- Example: DSDV (destination sequenced distance vector)

## Reactive protocols

- Determine route if and when needed
- Source initiates route discovery
- Example: DSR (dynamic source routing)

## Hybrid protocols

- Adaptive; Combination of proactive and reactive
- Example : ZRP (zone routing protocol)

# Protocol Trade-offs

## Proactive protocols

- Always maintain routes
- Little or no delay for route determination
- Consume bandwidth to keep routes up-to-date
- Maintain routes which may never be used

## Reactive protocols

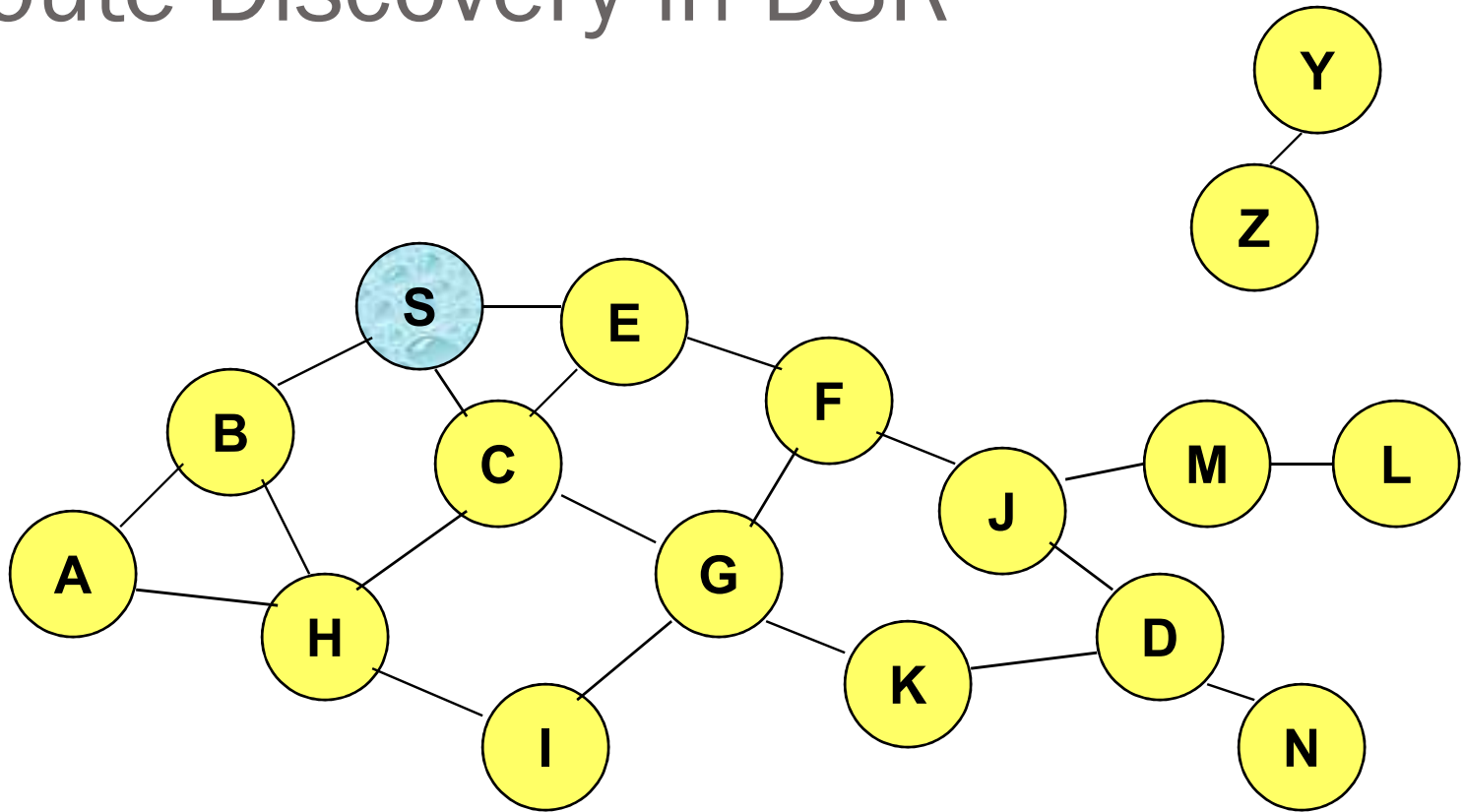
- Lower overhead since routes are determined on demand
- Significant delay in route determination
- Employ flooding (global search)
- Control traffic may be bursty

Which approach achieves a better trade-off depends on the traffic and mobility patterns

# Dynamic Source Routing (**DSR**)

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a **route discovery**
- Source node S floods **Route Request (RREQ)**
- Each node *appends own identifier* when forwarding RREQ

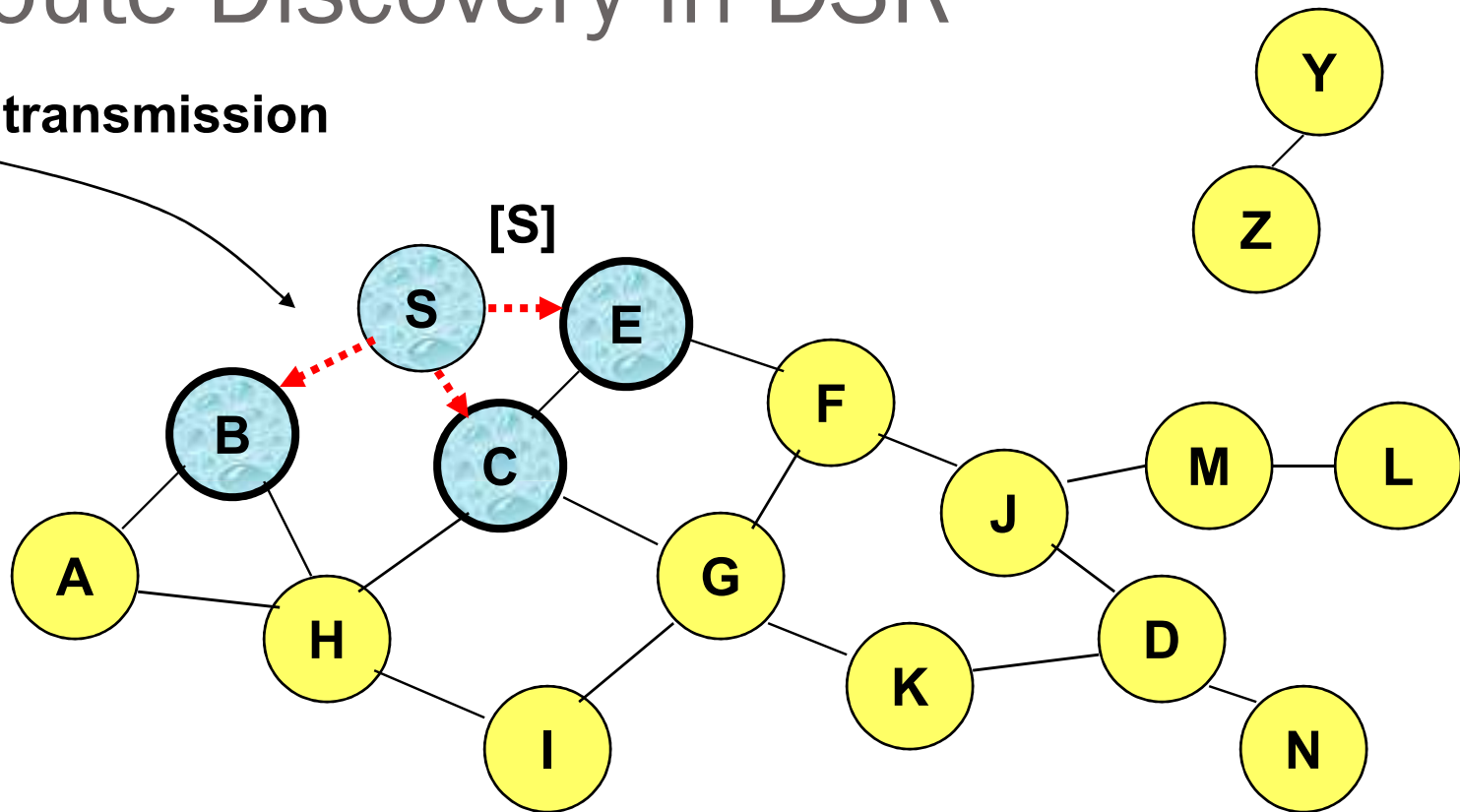
# Route Discovery in DSR



**Represents a node that has received RREQ for D from S**

# Route Discovery in DSR

Broadcast transmission

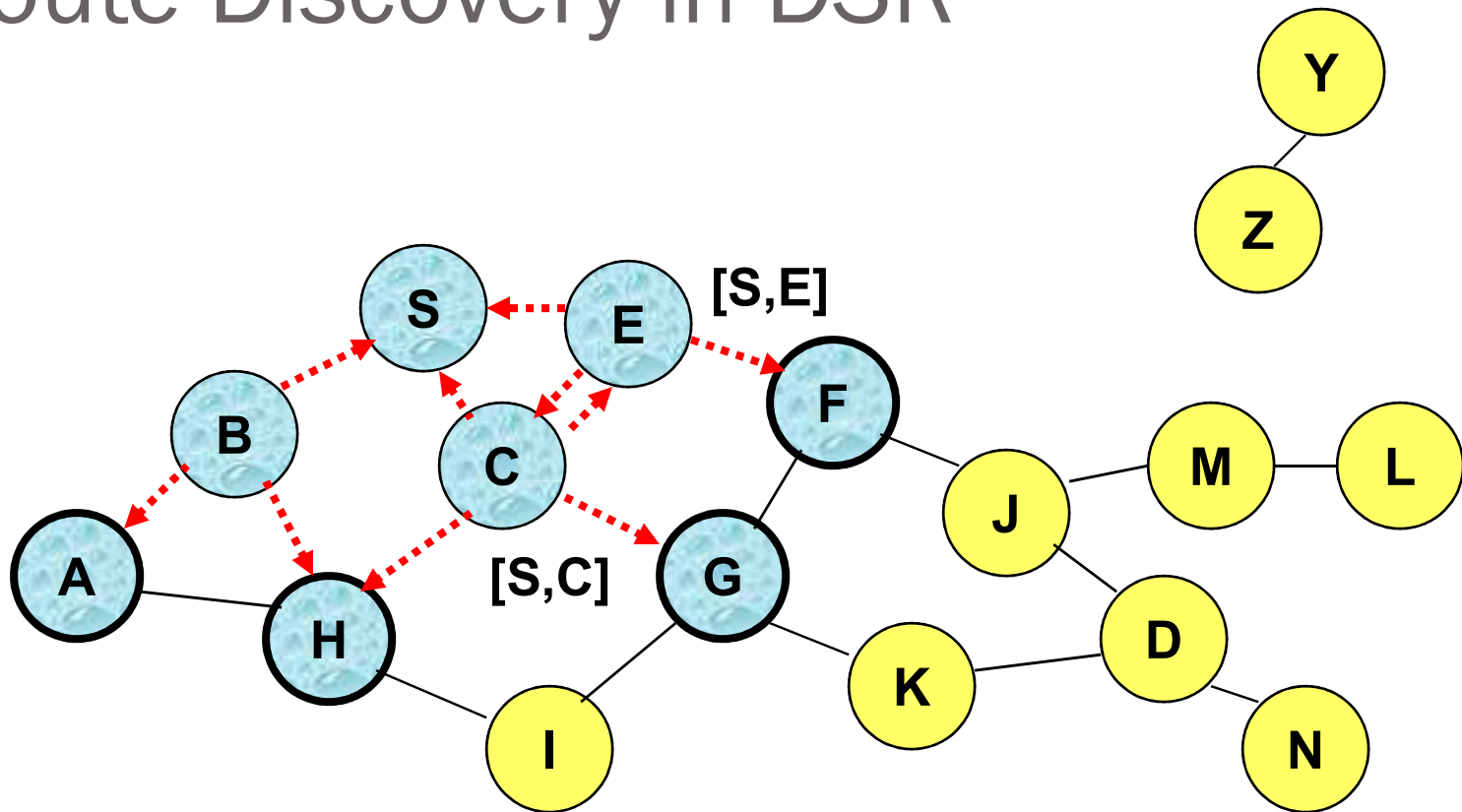


.....> Represents transmission of RREQ

[X,Y] Represents list of identifiers appended to RREQ

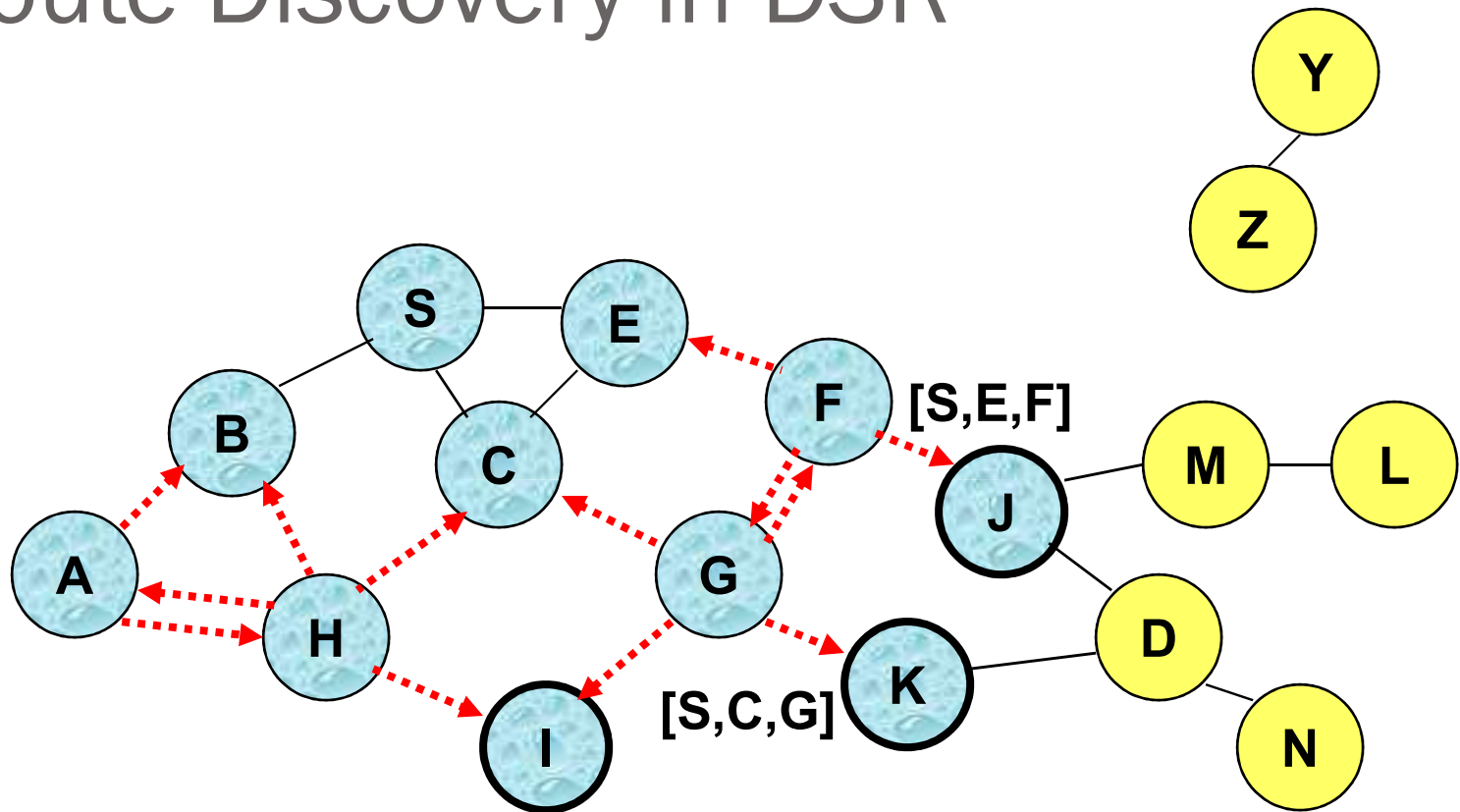


# Route Discovery in DSR



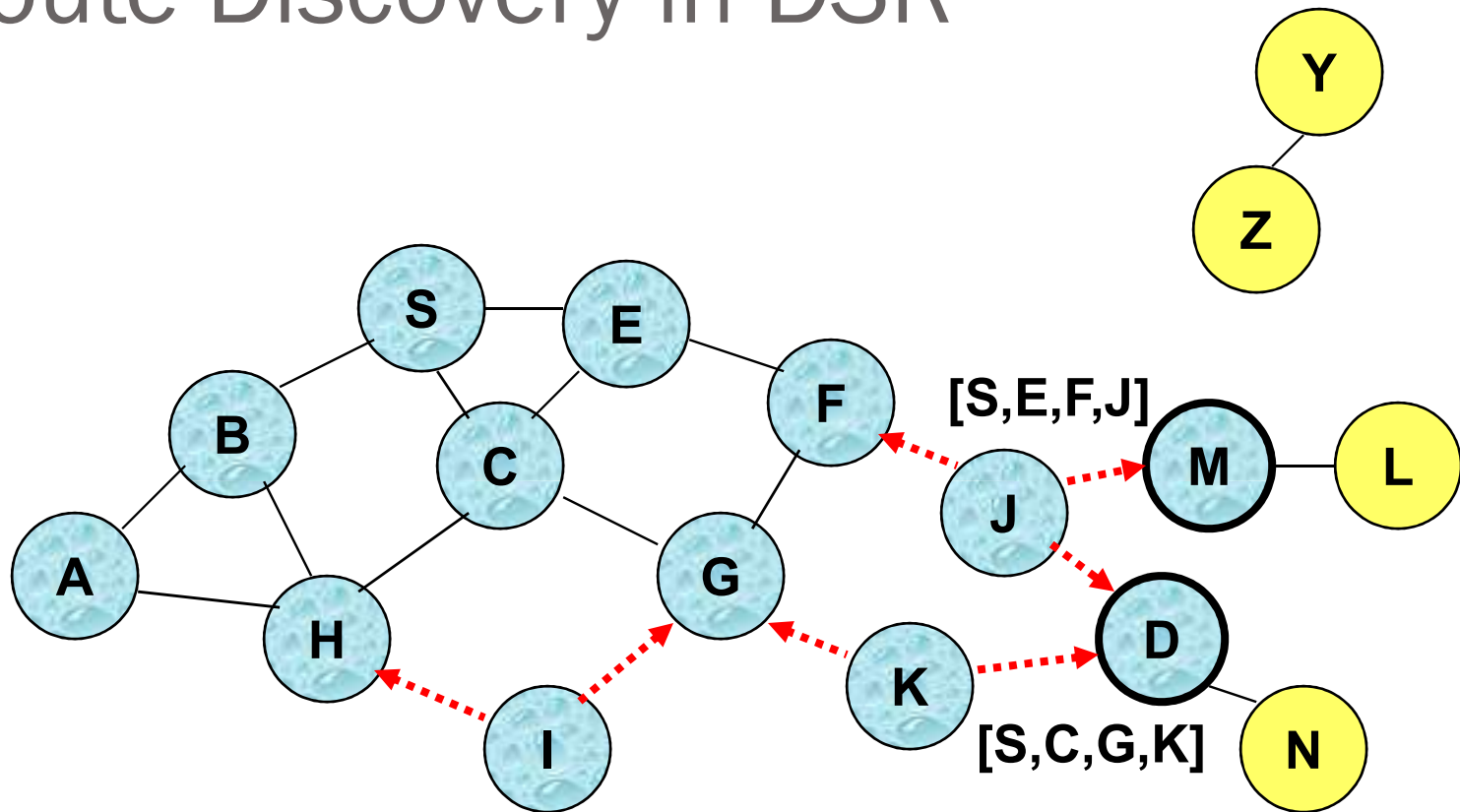
- Node H receives packet RREQ from two neighbors:  
**potential for collision**

# Route Discovery in DSR



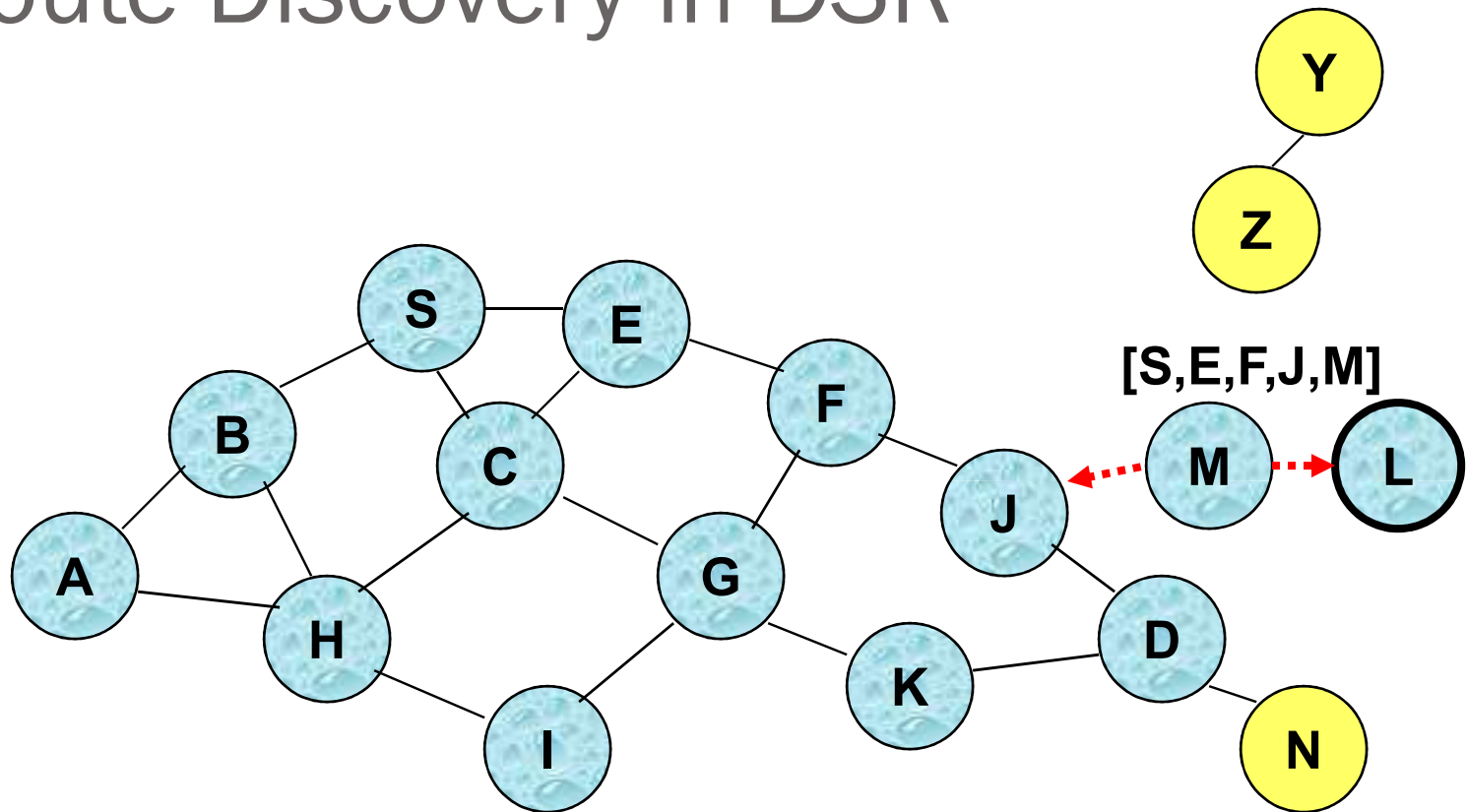
- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

# Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are **hidden** from each other, their **transmissions may collide**

# Route Discovery in DSR

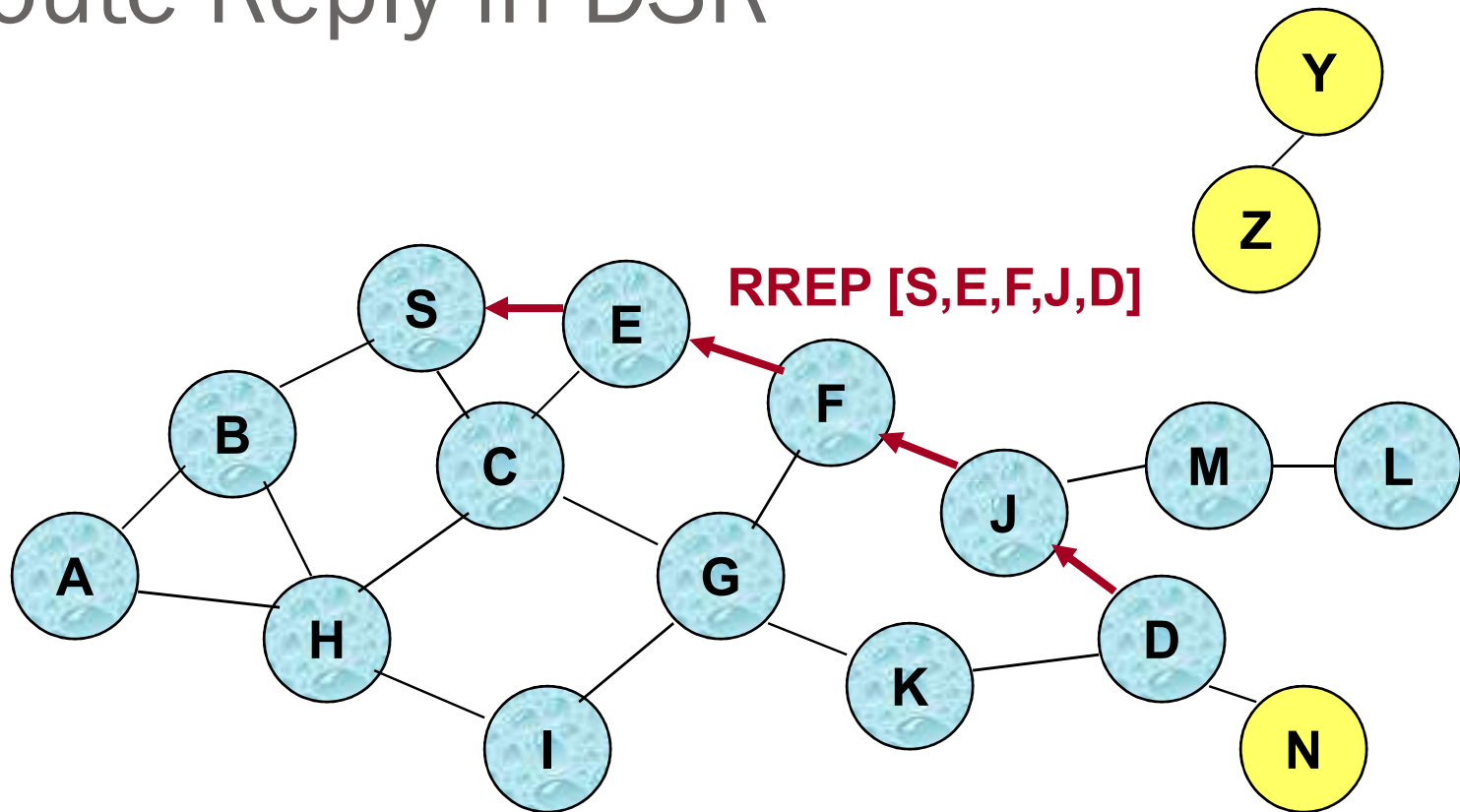


- Node D **does not forward** RREQ, because node D is the **intended target** of the route discovery

# Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a **Route Reply (RREP)**
- RREP is sent on a route obtained by **reversing** the route appended to received RREQ
- RREP **includes the route** from S to D on which RREQ was received by node D

# Route Reply in DSR

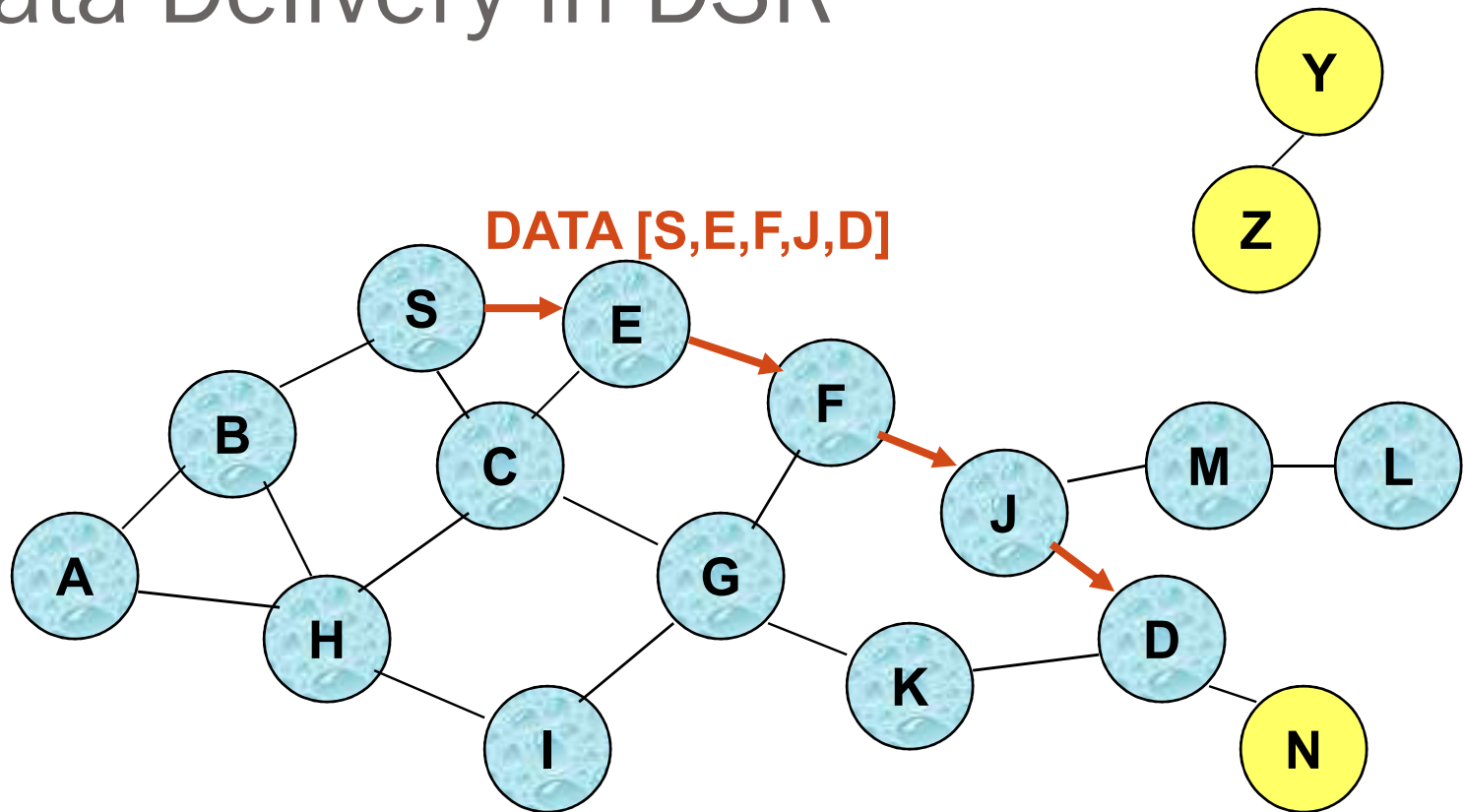


← Represents RREP control message

# Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name **source routing**
- Intermediate nodes use the **source route** included in a packet to determine to whom a packet should be forwarded

# Data Delivery in DSR



**Packet header size grows with route length**



# DSR Optimization: Route Caching

- Each node caches a new route it learns by *any means*
- When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F
- When node K receives Route Request [S,C,G] destined for node, node K learns route [K,G,C,S] to node S
- When node F forwards Route Reply RREP [S,E,F,J,D], node F learns route [F,J,D] to node D
- When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
- A node may also learn a route when it overhears Data
- **Problem:** Stale caches may increase overheads

# Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

# Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Potential collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
- Stale caches will lead to increased overhead

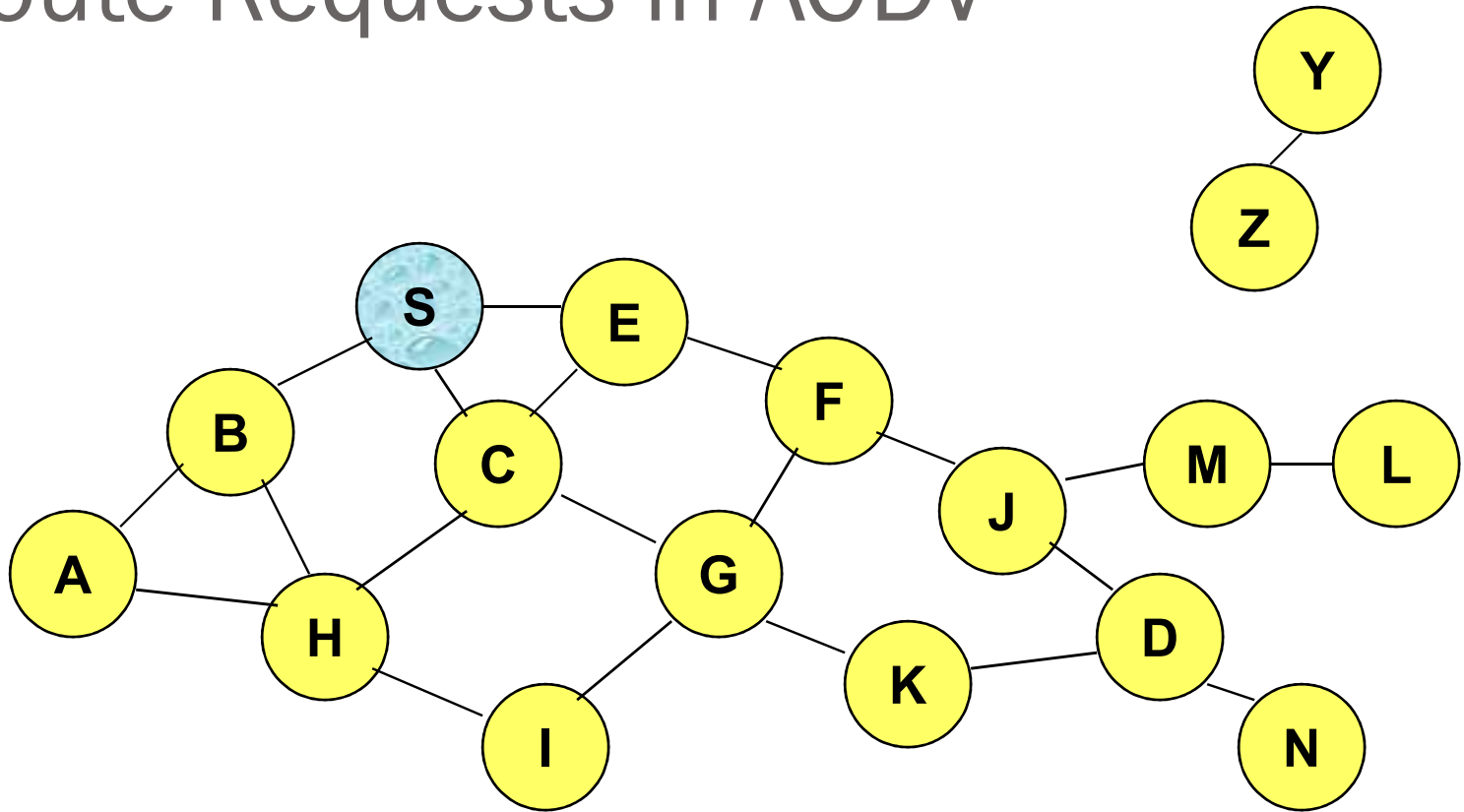
# Ad Hoc On-Demand Distance Vector Routing (**AODV**)

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate

# AODV

- **Route Requests (RREQ)** are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a **Route Reply (RREP)**
- Route Reply travels along the reverse path set-up when Route Request is forwarded

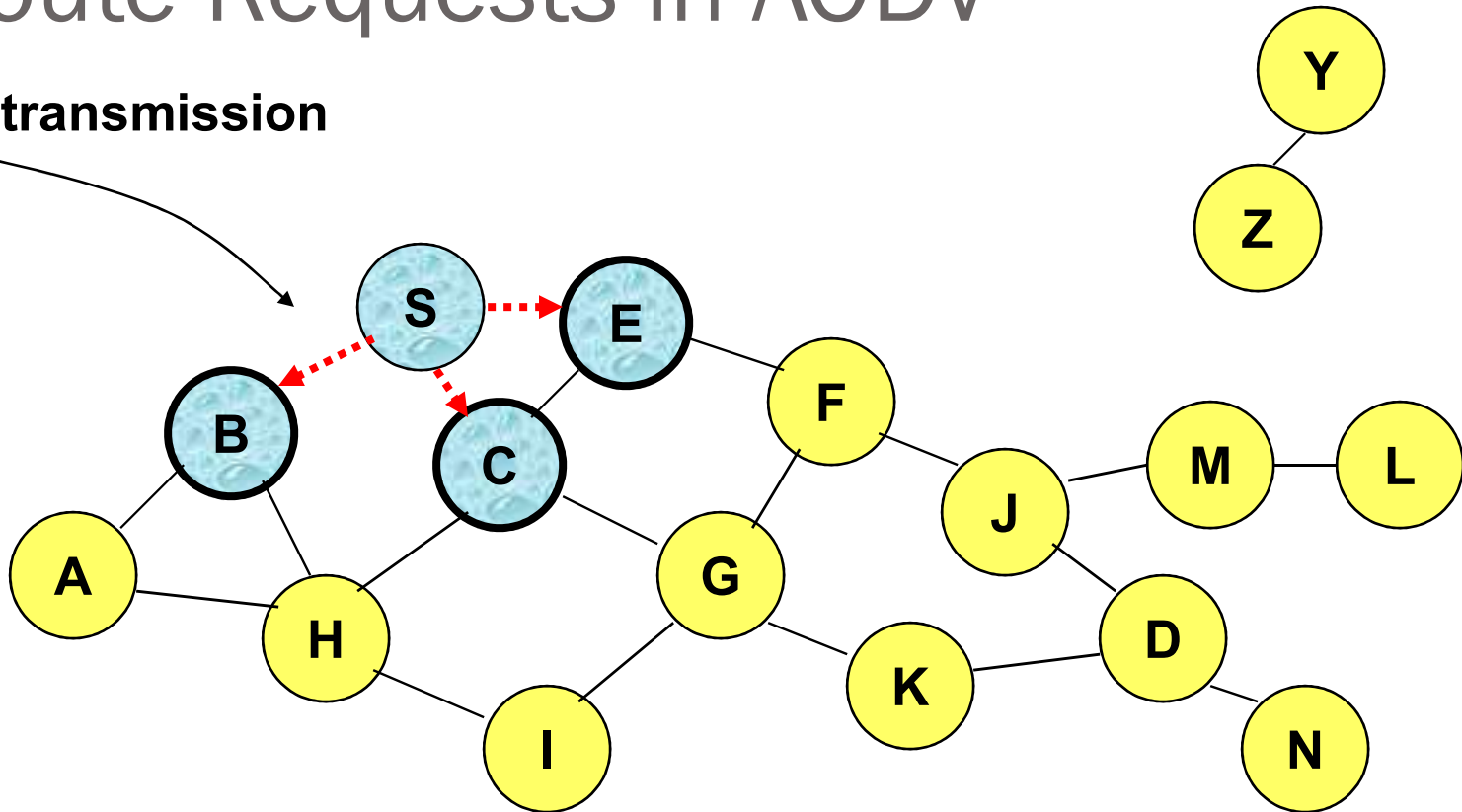
# Route Requests in AODV



**Represents a node that has received RREQ for D from S**

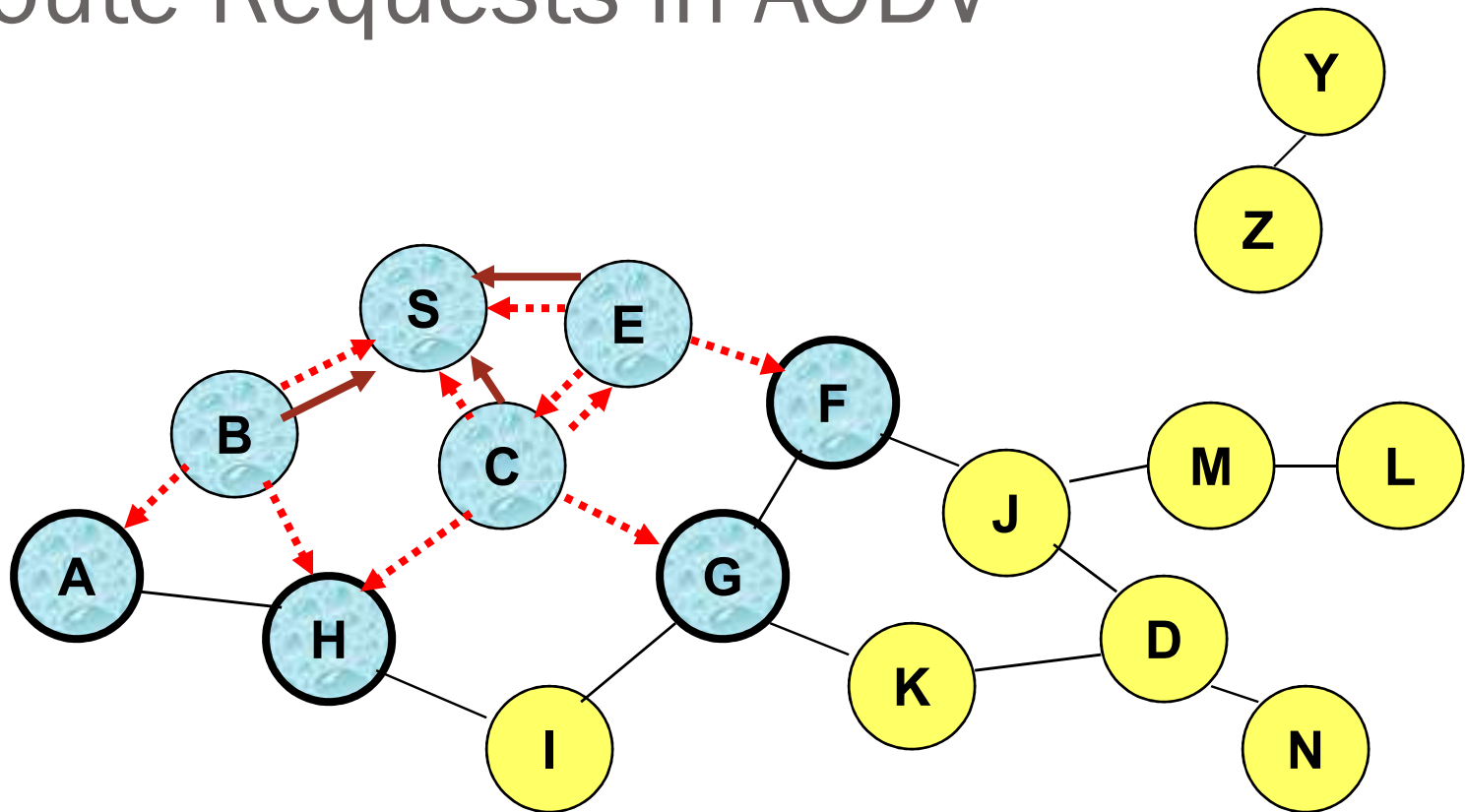
# Route Requests in AODV

Broadcast transmission



.....→ Represents transmission of RREQ

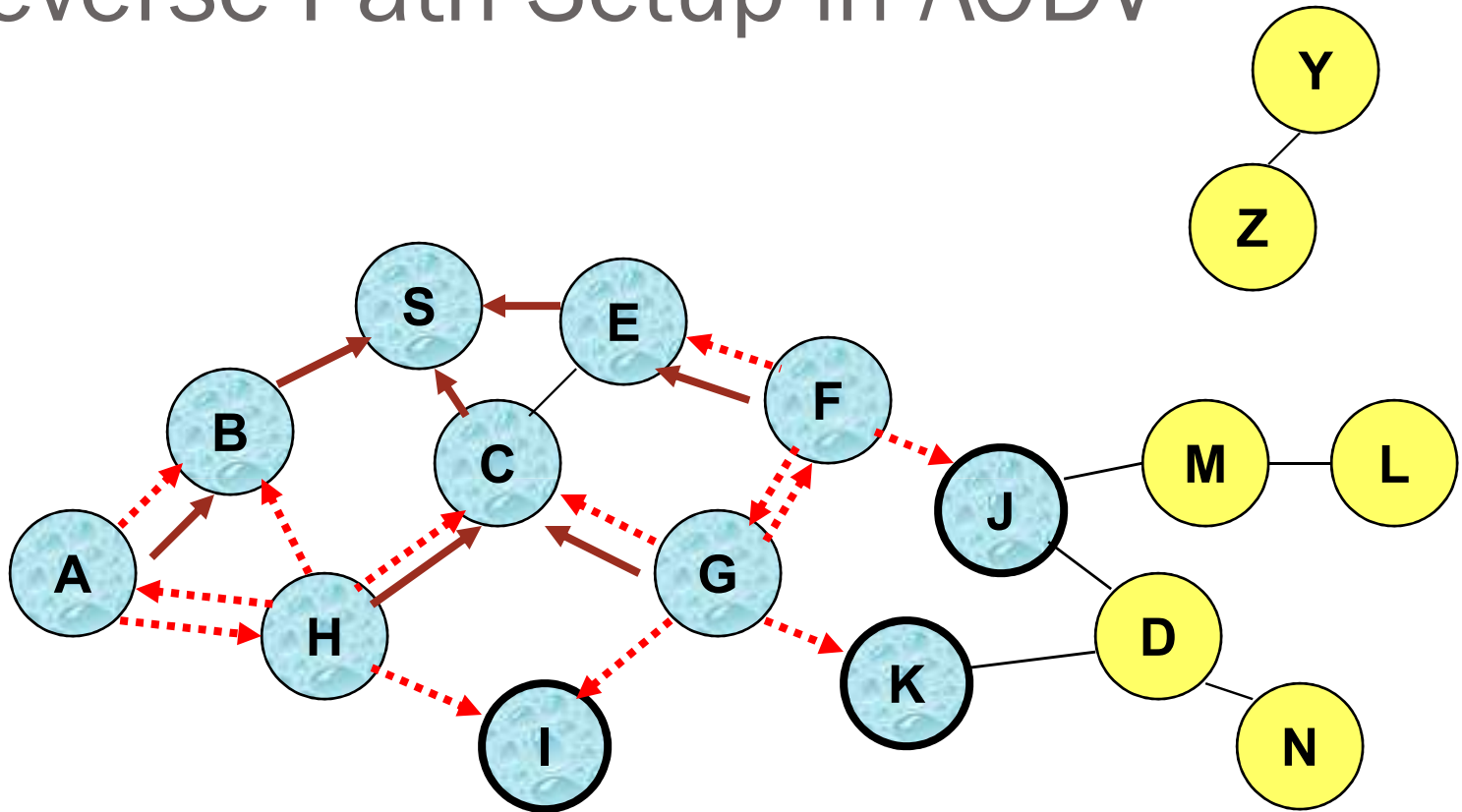
# Route Requests in AODV



← Represents links on Reverse Path

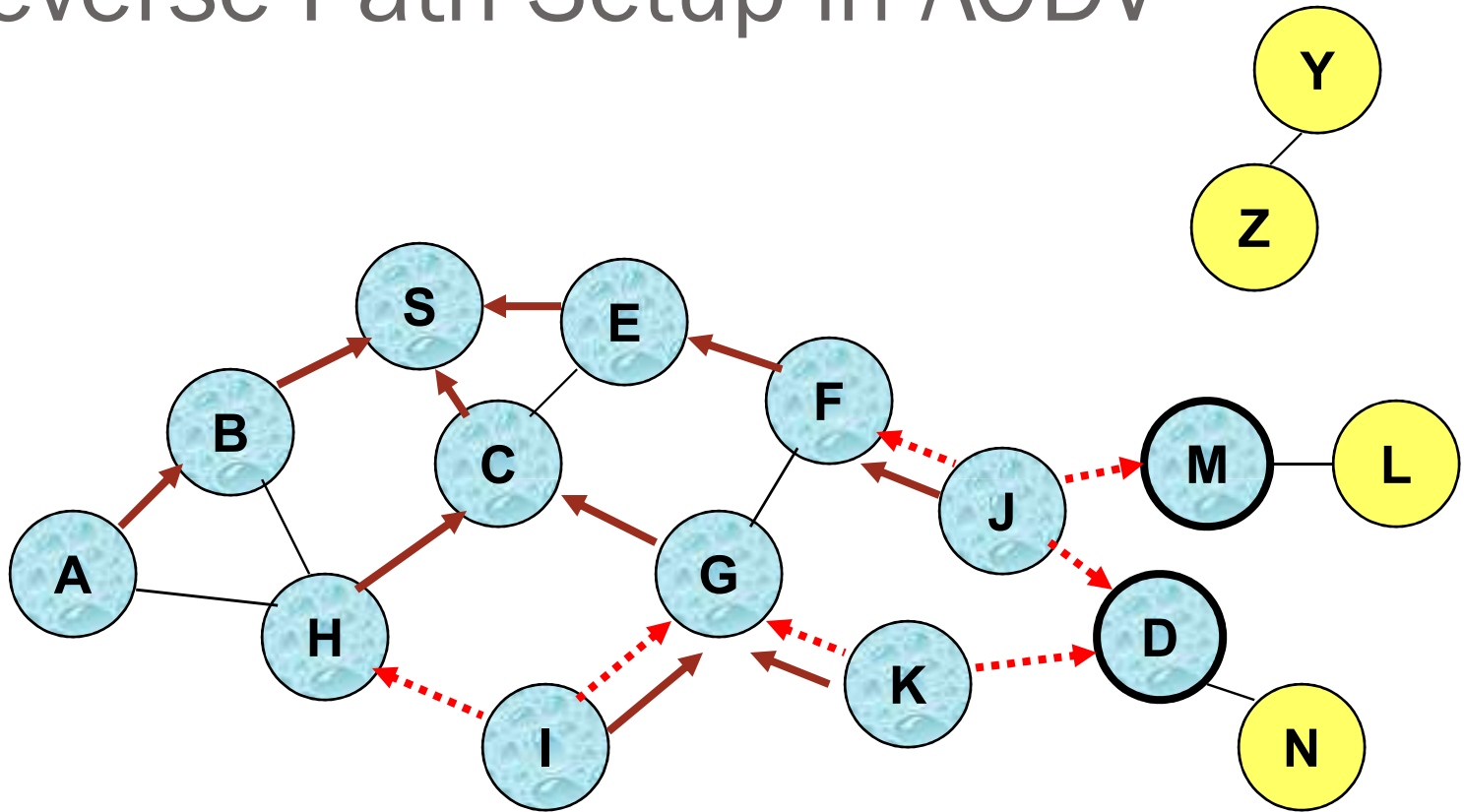


# Reverse Path Setup in AODV

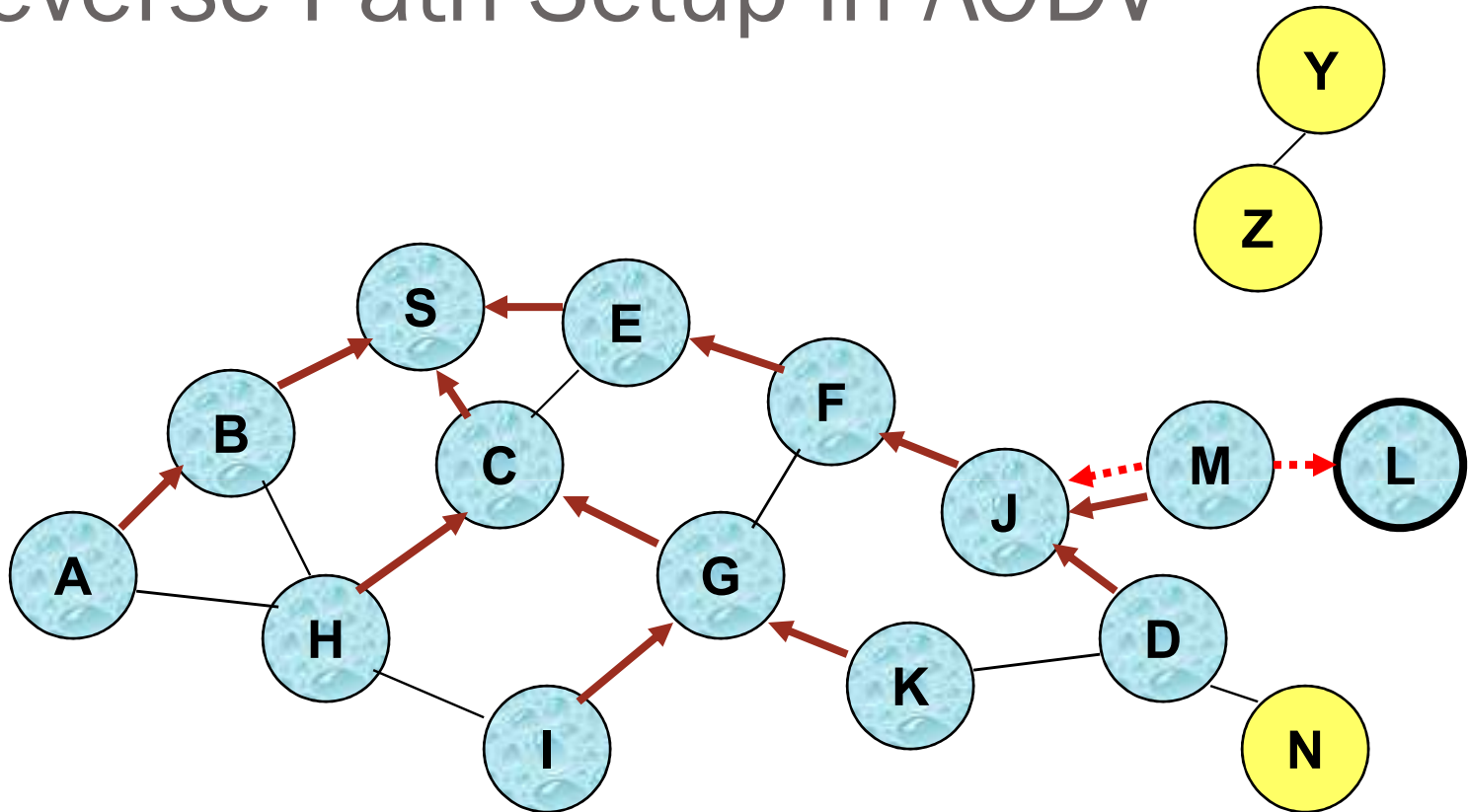


- Node C receives RREQ from G and H, but does not forward it again, because node C has **already forwarded RREQ** once

# Reverse Path Setup in AODV

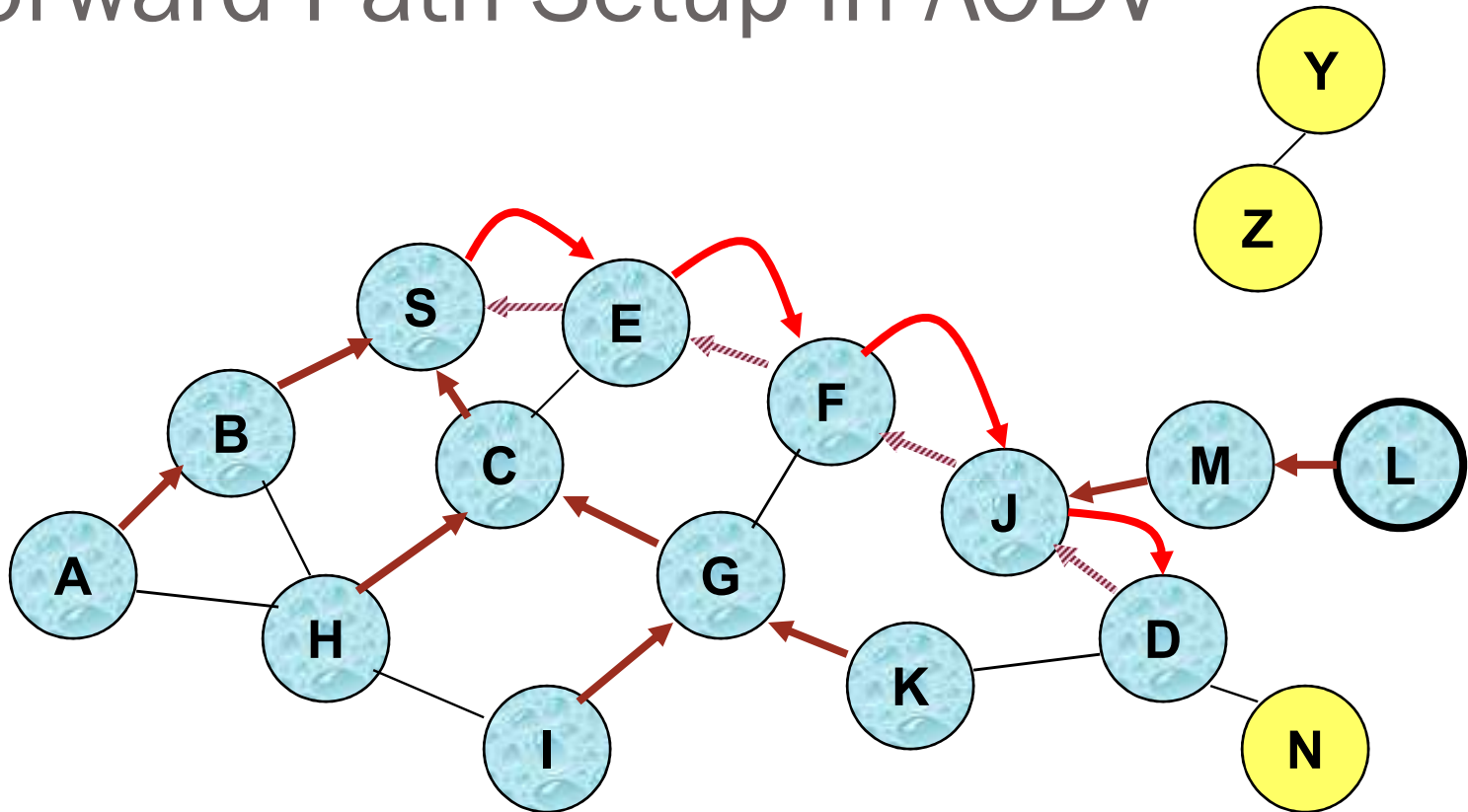


# Reverse Path Setup in AODV



- Node D **does not forward** RREQ, because node D is the **intended target** of the RREQ

# Forward Path Setup in AODV



**Forward links are setup when RREP travels along the reverse path**



**Represents a link on the forward path**

# Route Request and Route Reply

- Route Request (RREQ) includes the last known **sequence number** for the destination
- An intermediate node may also send a Route Reply (RREP) provided that it knows a **more recent path** than the one previously known to sender
- Intermediate nodes that forward the RREP, also record the next hop to destination
- A routing table entry maintaining a **reverse path** is purged after a timeout interval
- A routing table entry maintaining a **forward path** is purged if *not used* for a *active\_route\_timeout* interval

# Link Failure

- A neighbor of node X is considered **active** for a routing table entry if the neighbor sent a packet within *active\_route\_timeout* interval which was forwarded using that entry
- Neighboring nodes periodically exchange **hello** message
- When the next hop link in a routing table entry breaks, all **active** neighbors are informed
- Link failures are propagated by means of **Route Error (RERR)** messages, which also update destination sequence numbers

# Route Error

- When node X is unable to forward packet P (from node S to node D) on link (X,Y), it generates a RERR message
- Node X increments the destination sequence number for D cached at node X
- The **incremented sequence number  $N$**  is included in the RERR
- When node S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as  $N$
- When node D receives the route request with destination sequence number  $N$ , node D will set its sequence number to  $N$ , unless it is already larger than  $N$

# AODV: Summary

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- At most one next-hop per destination maintained at each node
  - DSR may maintain several routes for a single destination
- Sequence numbers are used to avoid old/broken routes
- Sequence numbers prevent formation of routing loops
- Unused routes expire even if topology does not change



# References

- [ece626web.groups.et.byu.net/Lectures/](http://ece626web.groups.et.byu.net/Lectures/)
- <http://www.mhhe.com/engcs/compsci/forouzan/dcn/student/olc/powerpoints13.mhtml>
- [http://highereducation.com/sites/0072460601/student\\_view0/chapter5/powerpoint\\_slides.html](http://highereducation.com/sites/0072460601/student_view0/chapter5/powerpoint_slides.html)
- <http://www.mhhe.com/engcs/compsci/forouzan/dcn/>
- <https://www.slideshare.net/tameemyousaf/switching-techniques>
- [https://www.tutorialspoint.com/data\\_communication\\_computer\\_network/physical\\_layer\\_switching.htm](https://www.tutorialspoint.com/data_communication_computer_network/physical_layer_switching.htm)
- <https://www.slideshare.net/vipinsahu/mpls-multiprotocol-label-switching>
- [www.it.iitb.ac.in/~sri/talks/manet.ppt](http://www.it.iitb.ac.in/~sri/talks/manet.ppt)