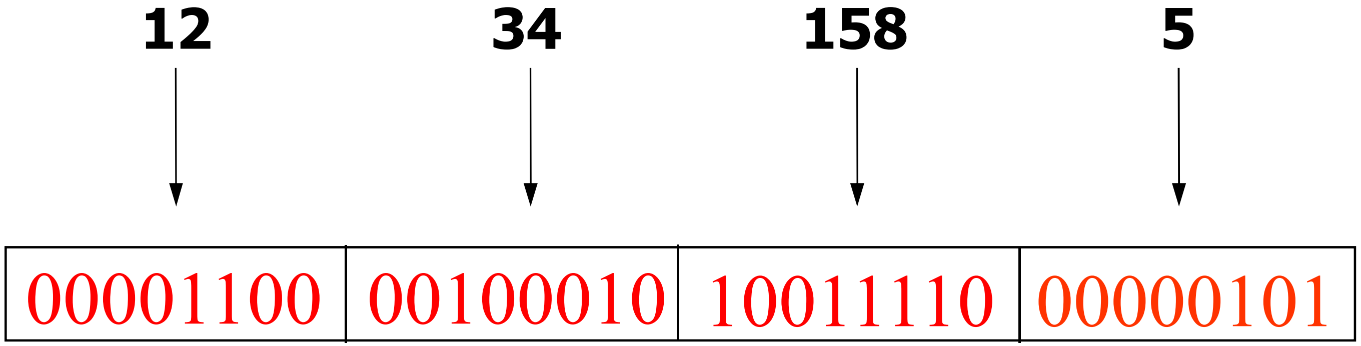
Computer Network and Distributed Systems

Network Layer - Addressing

IP Address (IPv4)

- A unique 32-bit number
- Each connection to a network (interface or network card) on each host connected to the Internet has a globally unique IP address
- Represented in dotted-quad notation W.X.Y.Z

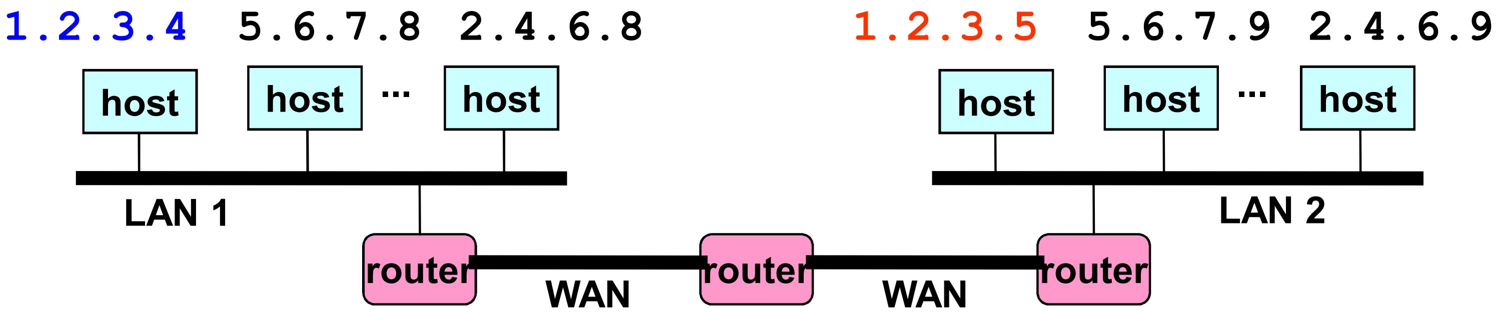


12 34 158 5

00001100 00100010 10011110 00000101

Scalability Challenge

- ❑ Suppose hosts had arbitrary IP addresses
  - ➢ Then every router would need a lot of information to know how to direct packets towards *every* host
- ❑ Scalability challenge => introduce hierarchy

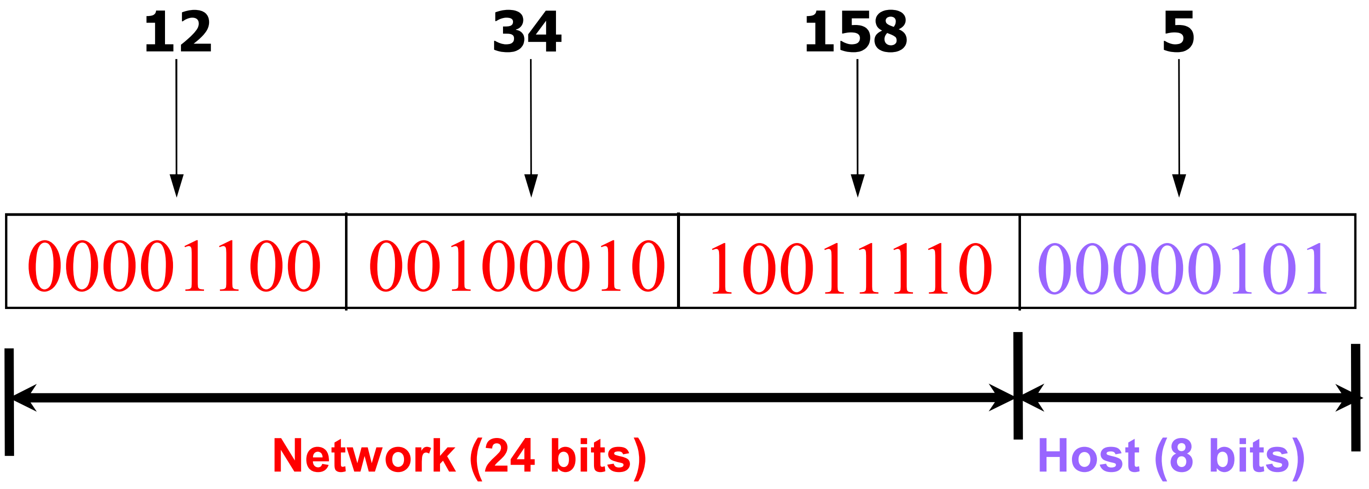


1.2.3.4	←
1.2.3.5	→
⋮	

forwarding table

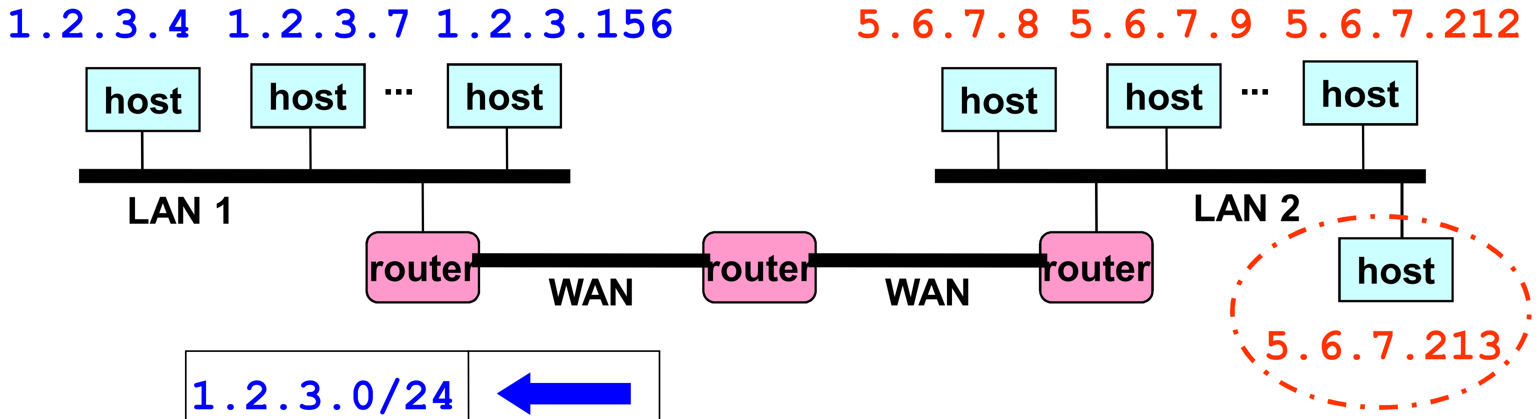
Hierarchical Addressing: IP Prefixes

- IP address divided into two logical parts:
  - network number
  - host number (used to identify hosts within a network)
- All hosts in a network have same network number



Easy to Add New Hosts

- ❑ No need to update the routers
  - ➢ E.g., adding a new host 5.6.7.213 on the right
  - ➢ Doesn’t require adding a new forwarding-table entry



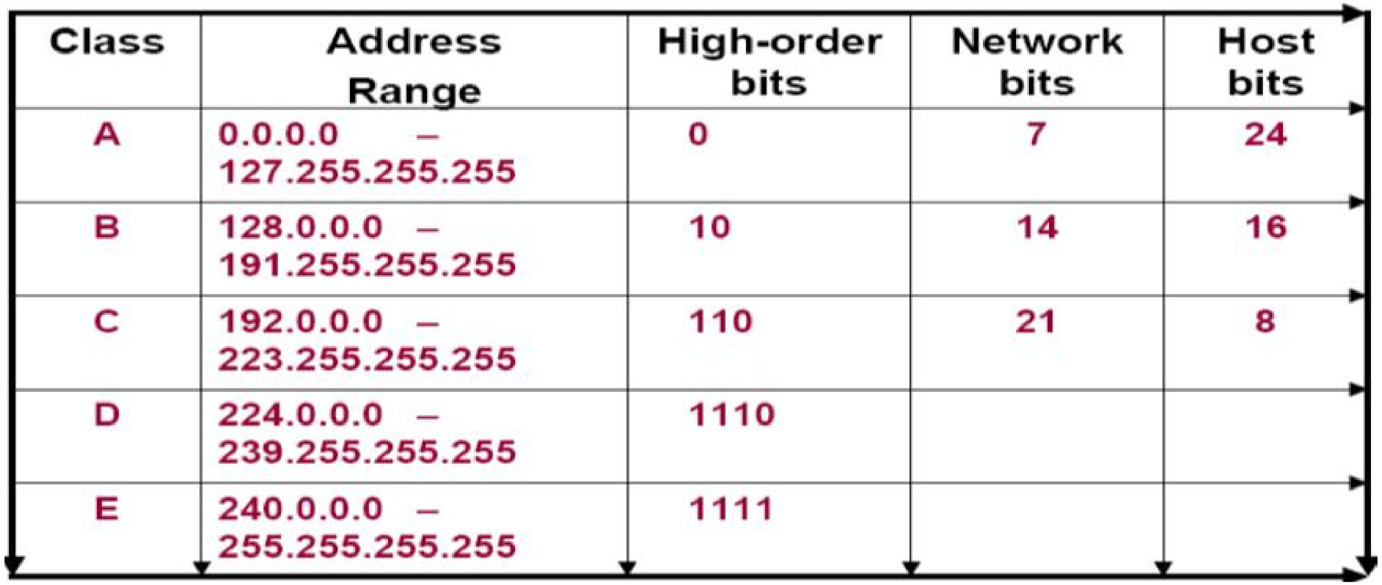
1.2.3.0/24	⬅
5.6.7.0/24	➡

forwarding table

Classful addressing

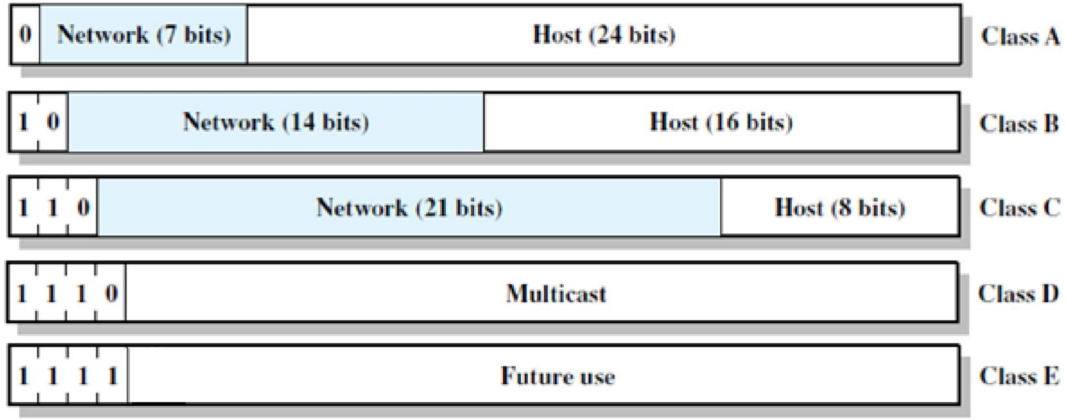
IP address classes

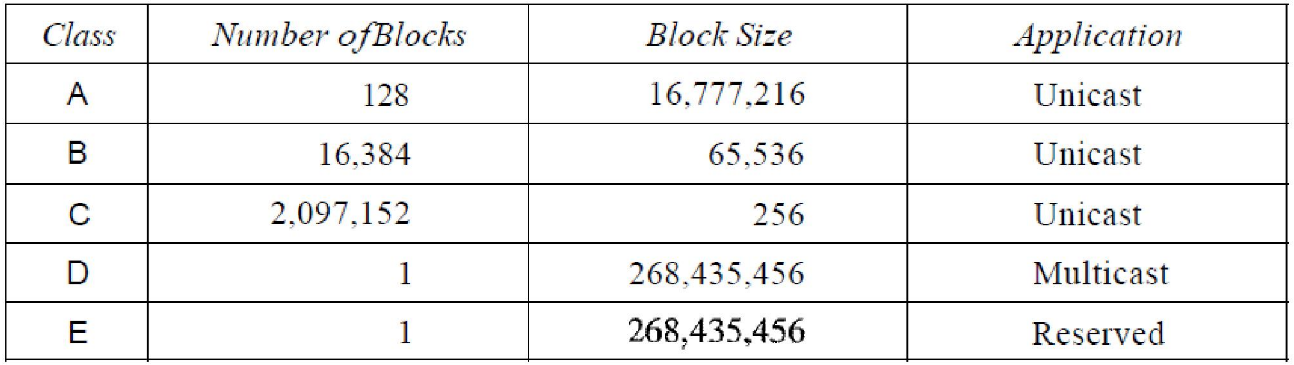
- ❑ Partitioning of IP address into network & host part defines IP address classes
- ❑ Five defined classes for IP addresses:
  - ➢ Class A, Class B, Class C used for unicast
  - ➢ Class D for multicast, Class E reserved



Class	Address Range	High-order bits	Network bits	Host bits
A	0.0.0.0 – 127.255.255.255	0	7	24
B	128.0.0.0 – 191.255.255.255	10	14	16
C	192.0.0.0 – 223.255.255.255	110	21	8
D	224.0.0.0 – 239.255.255.255	1110		
E	240.0.0.0 – 255.255.255.255	1111		

IP address classes





Class	*Number ofBlocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

- ❑ Number of Class-A networks are 126 as network addresses with a first octet of 0 (binary 00000000) and 127 (binary 01111111) are reserved
- ❑ Actual number of hosts per network = 2^(Number of Host bits) – 2. Why ?

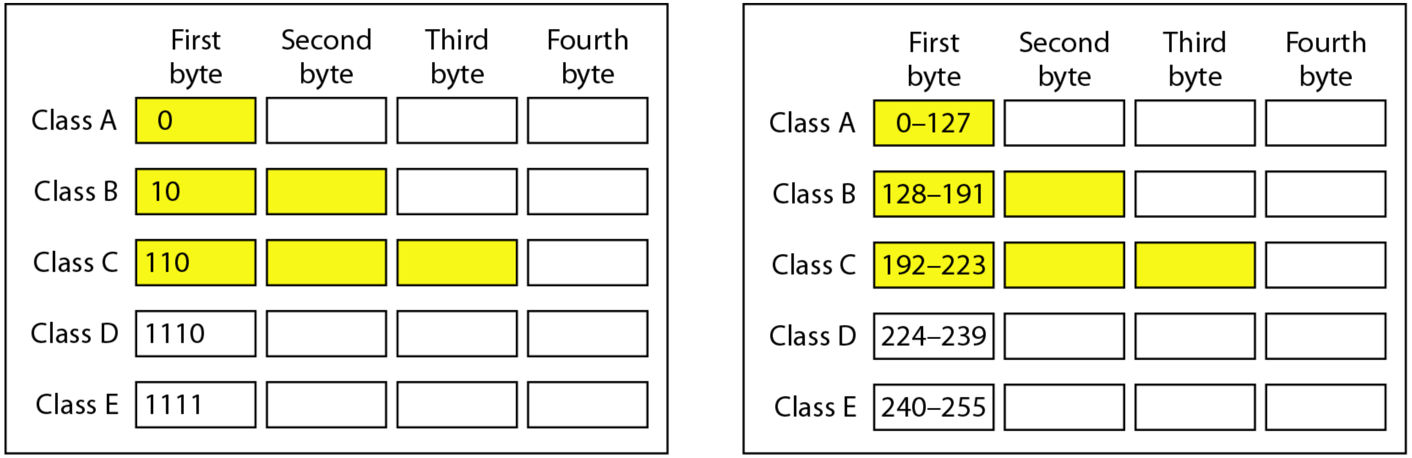
Network and Directed Broadcast addresses

- An IP address with all bits of hostid portion equal to zero (and a network portion) used to refer to the network itself (14.0.0.0)

- Any IP address with all bits of hostid portion equal to 1 (and a network portion) reserved for directed broadcast within the network (14.255.255.255)

- No host in a network should be given a hostid of all 0s or all 1s

Finding the classes in binary and dotted-decimal notation



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

a. Binary notation

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

b. Dotted-decimal notation

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

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 byte is 14; the class is A.

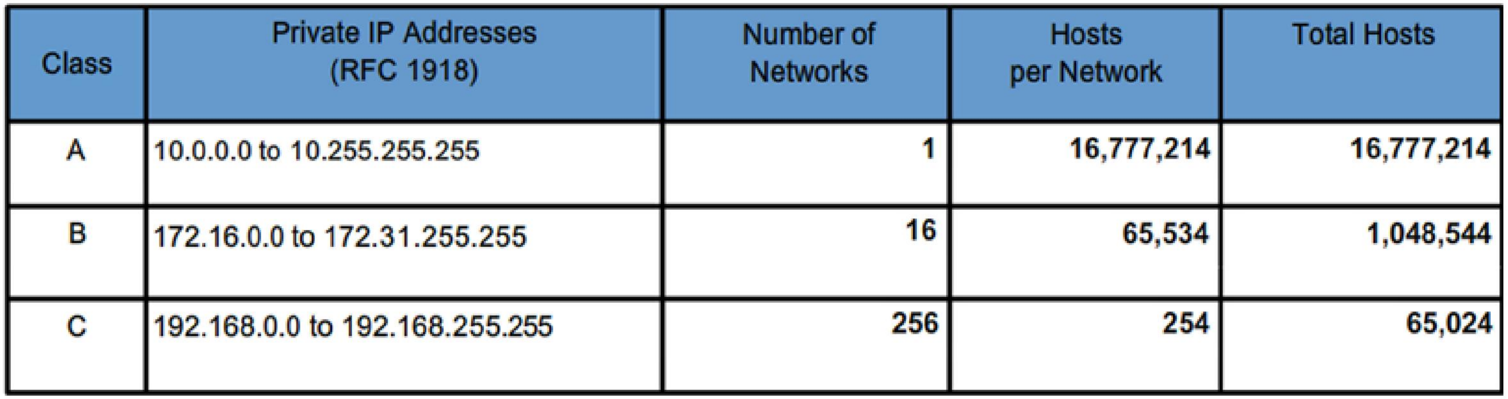
d. The first byte is 252; the class is E.

Some special IP addresses

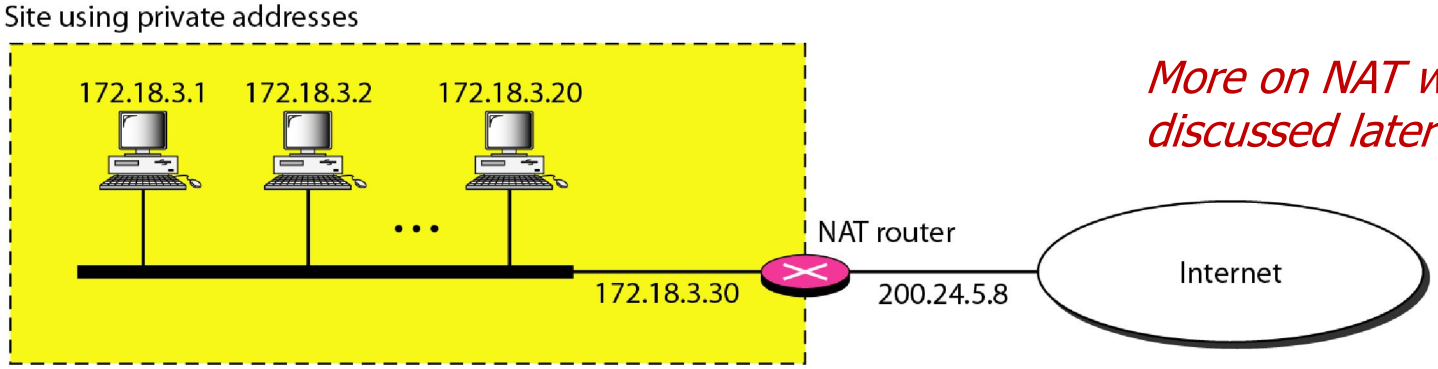
- ❑ Not all possible 32-bit addresses have been assigned to classes
- ❑ Loopback address
  - ➢ The network prefix 127.0.0.0 (a value from the class A range) reserved for loopback
  - ➢ Used to test TCP / IP and for inter-process communication **within a host**
- ❑ All 32 bits zero (0.0.0.0)→ this host on this network
- ❑ All 32 bits one (255.255.255.255)→ limited broadcast on this (local) network

Private IP Address

- To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses



Class	Private IP Addresses (RFC 1918)	Number of Networks	Hosts per Network	Total Hosts
A	10.0.0.0 to 10.255.255.255	1	16,777,214	16,777,214
B	172.16.0.0 to 172.31.255.255	16	65,534	1,048,544
C	192.168.0.0 to 192.168.255.255	256	254	65,024



More on NAT will be discussed later

Internet Addressing Authority

- To ensure that the address assigned to a network is globally unique
- A central organization that sets policy and assigns values for addresses, constants used in protocols
  - Originally Internet Assigned Number Authority (IANA)
  - Later, the Internet Corporation for Assigned Names and Numbers (ICANN)
- To connect its network to the Internet, a firm contacts an Internet Service Provider
  - ISPs assigned blocks of IP addresses by the central authorities

Problems with IP address classes

- Main problem with classful addressing:
  - Fast depletion of available network addresses
  - Wastage of IP addresses (organizations may not need a full class)
  - Class B addresses getting most rapidly depleted

- Cause: exponential growth in number of networks over the years
- Also, routing tables become too large
- Classful addressing, which is almost obsolete, is replaced with classless addressing.

Mechanisms to conserve IP address prefixes

IP Subnets

- Subnet – a subset of a class A, B or C network
- Host portion of IP address partitioned into **subnet part** and **host part**

- Uses a 32-bit subnet mask to specify the division
  - In binary, the mask is a series of contiguous 1's (identifies network portion) followed by a series of contiguous 0's (identifies host portion)
    - ✓ Eg 11111111 11111111 11111111 11000000
    - ✓ Equivalent dotted decimal 255.255.255.192
- Allows more efficient address space allocation (close to what is necessary)
  - less wastage of IP addresses

Space allocation - an example

- 4 organizations, each apply for 50 IP addresses

- Instead of assigning one class C address to each, only one class C address 200.10.12.0 assigned
  - Network divided into four subnets
  - Each organization needs 6-bit host numbers
  - 2 msb's of the host number used to distinguish among the four subnets
  - Subnet mask used: 255.255.255.192

Space allocation - an example

- 4 organizations, each apply for 50 IP addresses
- Instead of assigning one class C address to each, only one class C address (lets assume 200.10.12.0) assigned
- Each organization needs 6-bit host numbers
- 2 msb's of the host number (the 8 lsb's in a class C address) used to distinguish among subnets

Last byte	Host num range	Network addr	Subnet mask
00 000000	Org 1: 0-63	200.10.12.0	255.255.255.192
01 000000 (64)	Org 2: 64-127	200.10.12.64	255.255.255.192
10 000000 (128)	Org 3:128-191	200.10.12.128	255.255.255.192
11 000000 (192)	Org 4:192-255	200.10.12.192	255.255.255.192

Space allocation - an example (2)

- 3 organizations, in which Org-1 applied 115 address, Org-2 and Org-3 applied 55 address
- Instead of assigning one class C address to each, only one class C address (lets assume 200.10.12.0) assigned

Last byte	Host num range	Network addr	Subnet mask
0 0000000	Org 1: 0-127	200.10.12.0	255.255.255.128
10 000000 (128)	Org 2:128-191	200.10.12.128	255.255.255.192
11 000000 (192)	Org 3:192-255	200.10.12.192	255.255.255.192

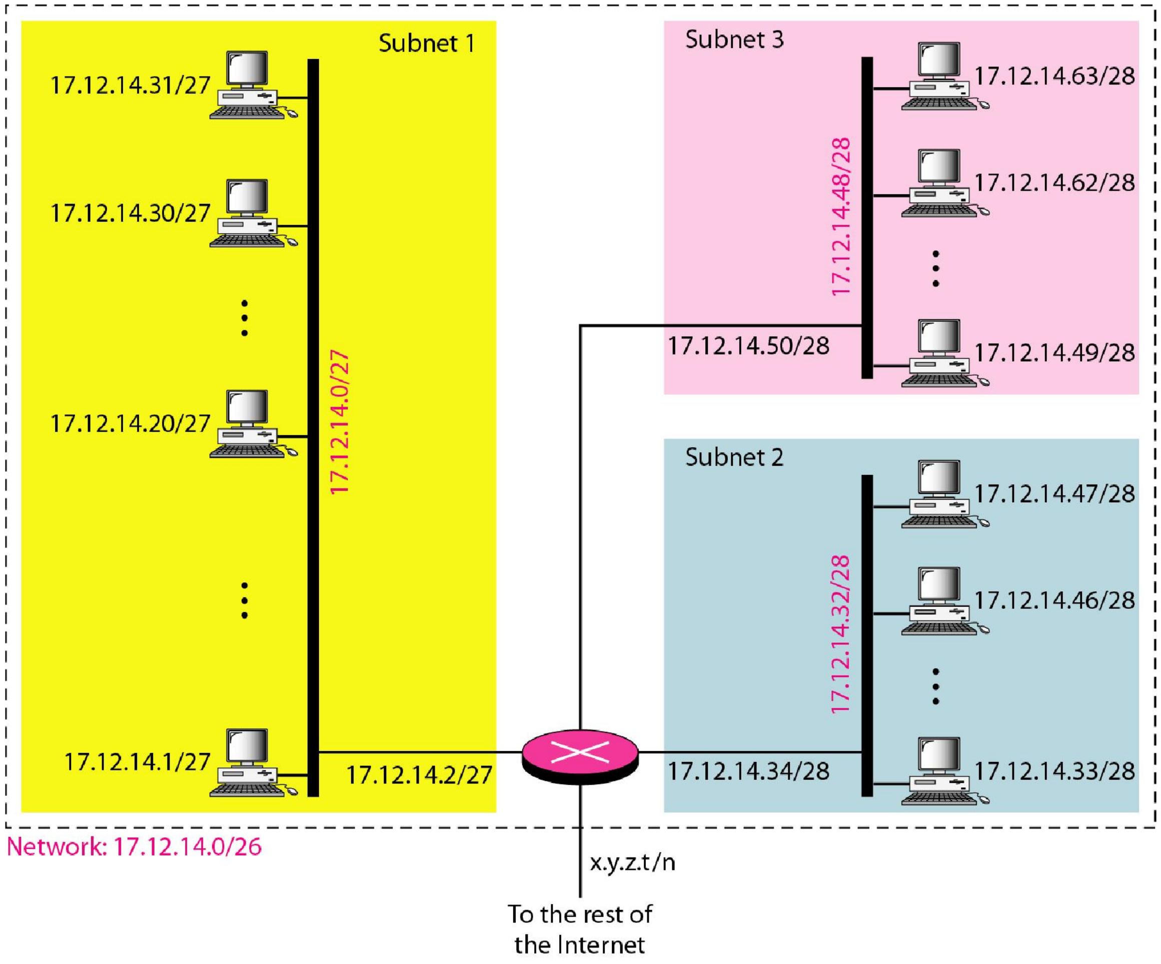
More about Subnet mask

- A subnet mask has 1s for all bits that correspond to the network portion of an IP address within that network
  - The standard does NOT restrict subnet masks to select contiguous bits of the address, however it is recommended that sites use contiguous bits 1
- Bitwise AND of the IP address of a host and subnet mask of the network should give the network number of the network in which the host is

Use of subnet mask

- To check if two IP addresses belong to the same subnet: the bit-wise AND of the two addresses with the netmask must be the same
- Network mask 255.255.255.240 is applied to a class C network 195.16.100.0
  - Mask = 11111111 11111111 11111111 11110000
  - Address of 1st host on this subnet = 195.16.100.1
  - Address of last host on this network = 195.16.100.14
  - Addresses 195.16.100.3 and 195.16.100.12 are in the same subnet
  - Addresses 195.16.100.3 and 195.16.100.19 are in different subnets

Configuration and addresses in a subnetted network



Natural Masks & Special IP addresses

- Class A, B, C addresses each have natural masks
  - Class A : natural mask is 255.0.0.0
  - Class B : natural mask is 255.255.0.0
  - Class C : natural mask is 255.255.255.0

- Classful addresses are self-identifying, but use of subnets make this property invalid
  - Routing algorithms become more complex

Routing in the presence of subnets

- A conventional routing table contains entries like
  - <network address, next hop address>
  - First 3 bits of an IP address informs address class

- In presence of subnets
  - Not possible to know which bits corresponds to the network portion, from the address alone
  - Routing table entries of the form

    <network address, subnet mask, next hop address>

Forwarding example with subnet masks

Consider the following routing table:

SubnetNumber	SubnetMask	NextHop
128.96.170.0	255.255.254.0	Interface 0
128.96.168.0	255.255.254.0	Interface 1
128.96.166.0	255.255.254.0	R2
128.96.164.0	255.255.252.0	R3
Default		R4

Packets with the following destination IP addresses will be sent to which interfaces ?

a. 128.96.171.92

b. 128.96.167.151

c. 128.96.163.151

d. 252.5.15.111

a. Interface 0

b. R2

c. R4

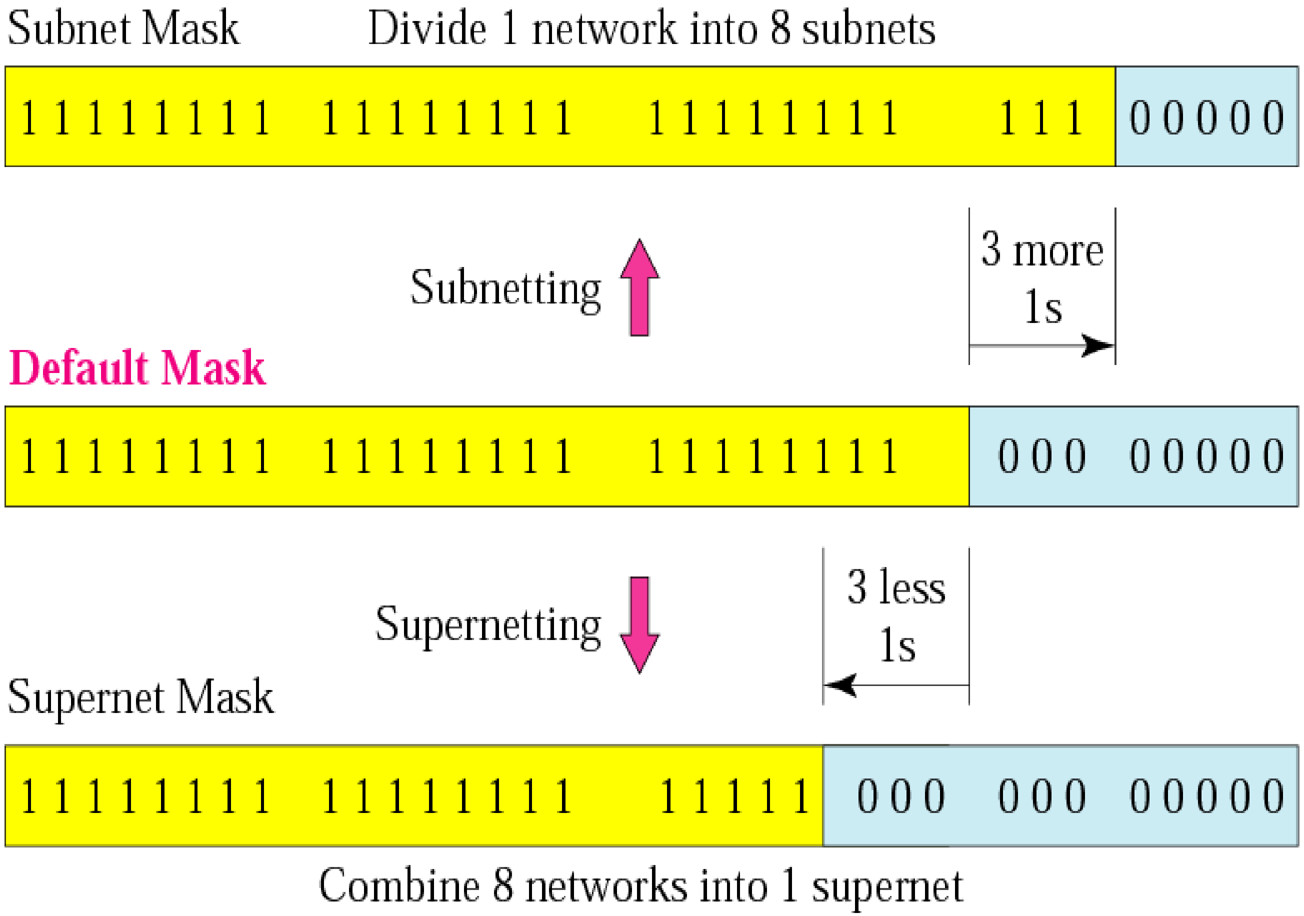
d. R4

Supernetting

- Motivation
  - By 1993, apparent that subnetting (used since 1980s) was not enough to prevent exhaustion of addresses
  - Class B addresses were getting exhausted very quickly

- Supernetting: complementary to subnetting
  - Subnetting aims to use a single IP network prefix for multiple physical networks at a given organization
  - Supernetting allows the addresses assigned to a single organization to span multiple classed prefixes
- Example: If an organization wants a class B address, assign it a block of 256 class C addresses instead of a single class B address

Subnet mask vs supernet mask



Subnet Mask

Divide 1 network into 8 subnets

11111111 11111111 11111111 111 00000

Subnetting

3 more 1s

Default Mask

11111111 11111111 11111111 000 00000

Supernetting

3 less 1s

Supernet Mask

11111111 11111111 11111 000 000 00000

Combine 8 networks into 1 supernet

Effect of subnets and supernets on routing

- ❑ Subnetting and supernetting help to conserve address space, but increases the amount of information that routers need to store and exchange

- ❑ Classless Inter-Domain Routing (CIDR) allows more efficient use of address space as well as solves this problem

Classless Inter-Domain Routing

CIDR

Number of addresses in a block

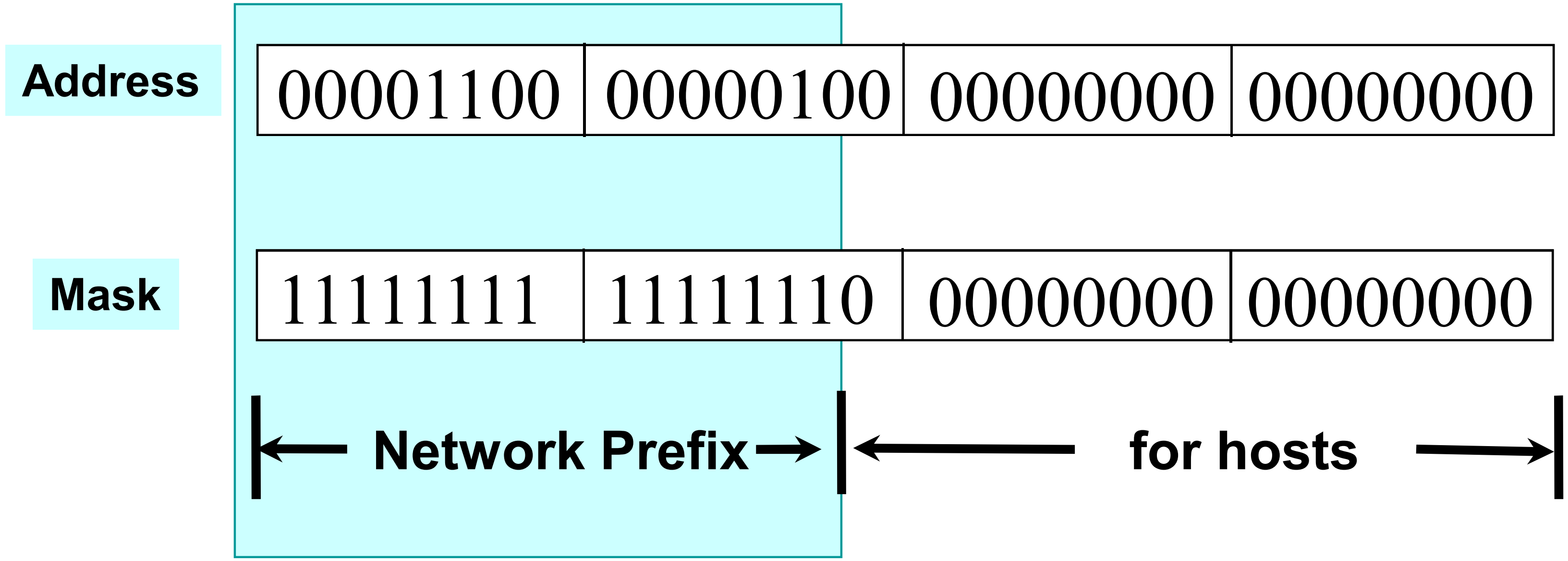
- Using CIDR, there is only one condition on the number of addresses in a block:
  - it must be a power of 2 (2, 4, 8, . . .)

- Number of IP addresses allocated must be as close to the requirement as possible

Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

IP Address : 12.4.0.0   IP Mask: 255.254.0.0



Address	00001100	00000100	00000000	00000000
Mask	11111111	11111110	00000000	00000000

Network Prefix

for hosts

Written as 12.4.0.0/15

Combine contiguous networks – an example

- Contiguous networks combined into a larger address block for the purpose of reducing routing table size
- Suppose Company A needs IP for 1000 machines
- Assign 4 *contiguous* Class C address blocks 192.60.128.0, 192.60.129.0, 192.60.130.0, 192.60.131.0 (last 8 bits 0 indicates a network)
- A single supernet defined: 192.60.128.0 / 22
  - Address : 192.60.128.0
  - Netmask: 255.255.252.0 (last 10 bits 0)

Combine contiguous networks – example (contd.)

- ❑ Routing table at all higher level routers:
  - ➢ Need only 1 entry for the four class C networks: 192.60.128.0/22
- ❑ Routing table at RA distinguishes among nets:
  - ➢ 192.60.128.0/24 –> send to router of first net
  - ➢ 192.60.129.0/24 –> send to router of second net
  - ➢ 192.60.130.0/24 –> send to router of third net
  - ➢ 192.60.131.0/24 –> send to router of fourth net
- ❑ Possible due to contiguous allocation of IP addr

Allocation of IP addresses (contd.)

- Allocation of contiguous blocks of IP addresses to geographically close networks preferred
  - Enables maximal use of super-nets to reduce number of entries at higher level routing tables
  - If class C address 192.60.128.0 assigned to a network in India, and 192.60.129.0 assigned to a network in Brazil, no chance of super-netting, routers at all levels need two different entries
  - If contiguous address blocks allocated to networks in India, all routers (at higher levels) upto some router in India can have just 1 entry

But…problems with CIDR too

- CIDR allows efficient use of the limited address space but makes packet forwarding much harder

- Forwarding table may have many matches
  - E.g., routing table entries for 201.10.0.0/21 and 201.10.6.0/23
  - The destination IP address 201.10.6.17 would match *both* entries
- Routers always do longest prefix match. If multiple entries match, longest match taken
  - 201.10.6.0/23 used even though both match

CIDR blocks reserved for private networks

- ❑ A set of network prefixes reserved for use in private networks
- ❑ These reserved prefixes will never be assigned to networks in the global Internet
- ❑ Known as private addresses or non-routable addresses

- ❑ Example: 10/8, 192.168/16

Computer Network and Distributed Systems

Network Layer Protocols

IPv4, ARP, *ICMP*

Basic Data Transfer Scheme

At source node *(assume destination IP is known)*

```
Bitwise AND own IP and destination IP with netmask
if network numbers same       # destination in same network
        if destination's MAC address known
                send to it directly
        else
                broadcast ARP-Request using broadcast MAC address to get the MAC address
        end if
else                    # destination node in some other network
        send to router, using router's MAC address
                # router's MAC address must be known
end if
```

Basic Data Transfer Scheme (contd.)

At any other machine

```
# at MAC layer
if dest MAC addr = own MAC addr or broadcast MAC addr
        give IP layer PDU to IP layer
else
        discard frame
# at IP layer
if dest IP addr = own IP addr or broadcast IP addr
        give higher layer PDU to higher layer
else if this machine is a router
        route the IP PDU (form a new MAC frame)
else
        discard the packet
```

Address Resolution Protocol (ARP)

Motivation

- ❑A node N may have to send a frame to another node M
  - ➢ If N has to send data outside the network, N must first send the frame to router R in this network
  - ➢ N needs to know the MAC address of M (or R) for this

- ❑Usually, IP layer of node N already knows IP address of the node to which it has to send data
  - ➢ Directly told by users (e.g., ssh to 10.2.1.97)
  - ➢ IP address of router specified while configuring nw connection

- ❑But N needs to know the MAC address of the next hop
- ❑ARP used to know the MAC address, given IP address

Address Resolution Protocol (ARP)

- Provides IP to Hardware (MAC) address mapping

- ARP packet (like IP packet) encapsulated in data link layer frame, e.g., Ethernet frame

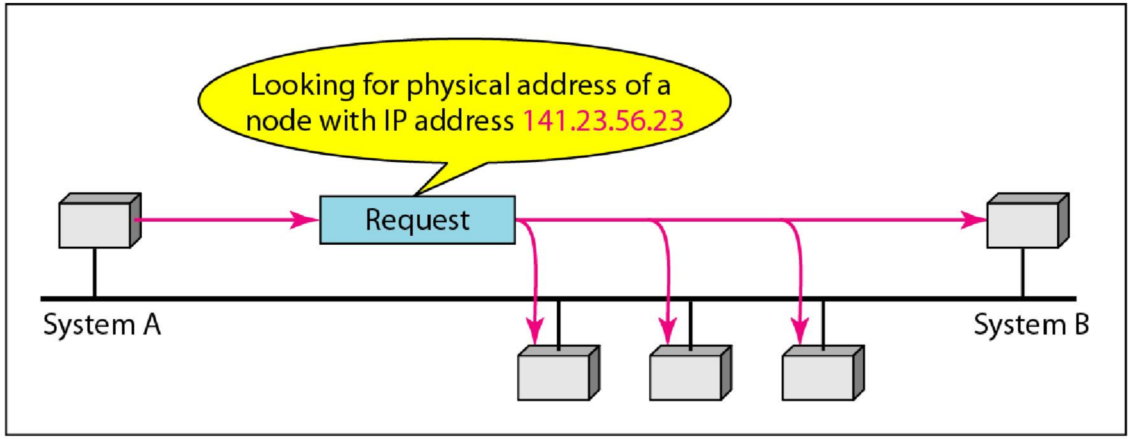
- Type field in MAC frame specifies ARP packet
  - When an MAC frame is received, receiver’s DLL layer sees the type field and
  - decides to which Network layer module (IP, ARP, …) the Network layer PDU will be handed

Use of ARP

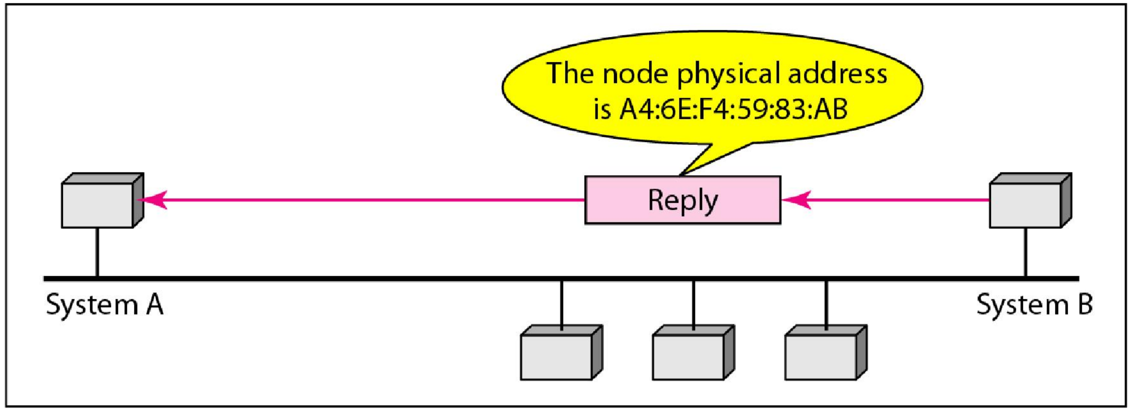
- ❑ Let node N need to know the MAC addr of node R
  - ➢ N already knows IP address of R
- ❑ N's ARP module creates and sends a MAC frame
  - ➢ TYPE field in Ethernet header set to ARP (0806_{16})
  - ➢ Destination MAC address: broadcast MAC address
  - ➢ Frame contains the IP address of R (whose MAC address required), both IP and MAC address of N

- ❑This MAC frame reaches ARP module of all nodes, only R replies
- ❑R sends a MAC frame to N (R already knows N's MAC address), which contains R's MAC address

ARP operation


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B

a. ARP request is broadcast



b. ARP reply is unicast

ARP Packet Format

<table>
<tr><td colspan="2">Hardware Type</td><td>Protocol Type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td>Operation
Request 1, Reply 2</td></tr>
<tr><td colspan="3">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="3">Sender protocol address
(For example, 4 bytes for IP)</td></tr>
<tr><td colspan="3">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="3">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

ARP Packet Format (contd.)

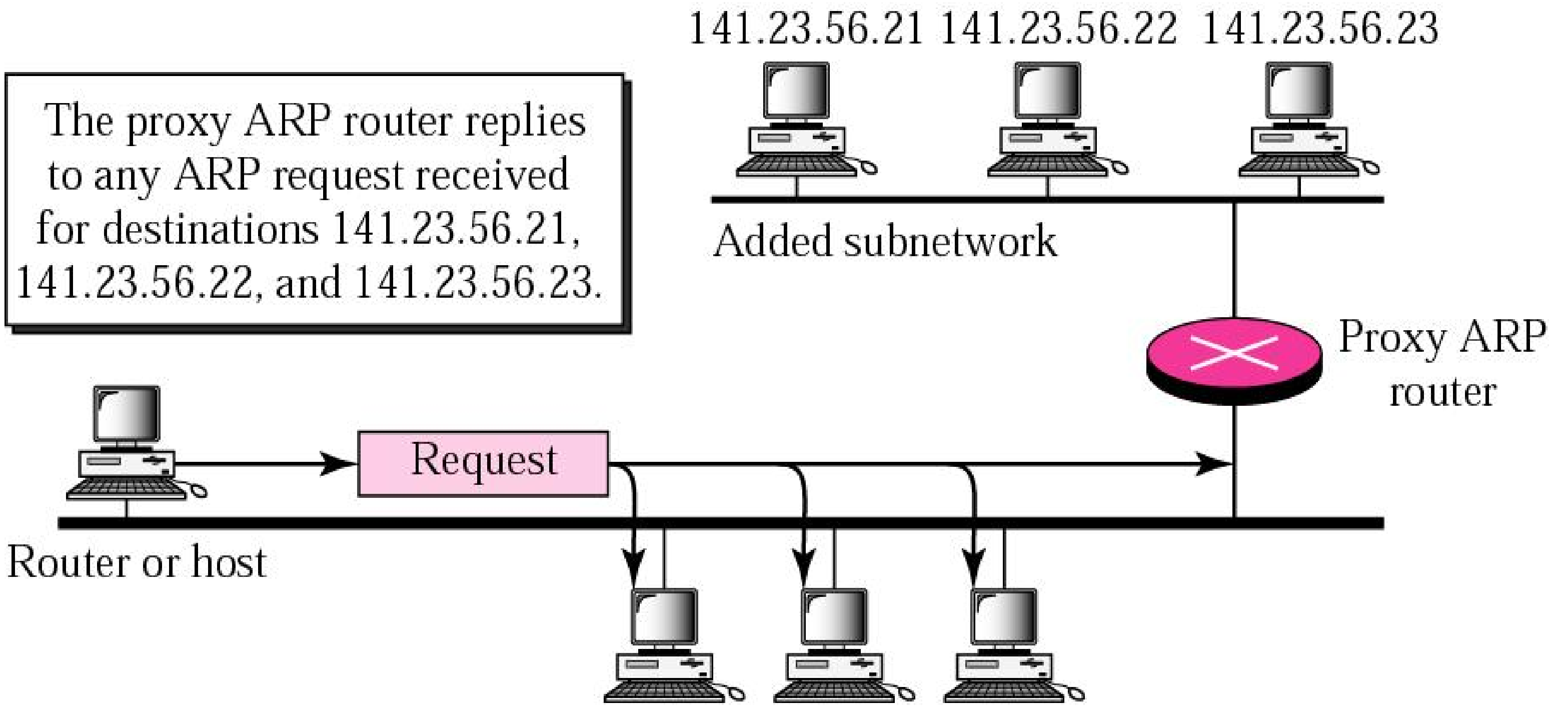
- Hardware type: a node can have multiple h/w interfaces e.g. Ethernet interface, FDDI interface, etc
  - Enquiry for which hardware interface type (value of 1 implies Ethernet)
- Protocol type: which higher level protocol is being used, IP or some other (contains value 0800_{16} for IP)
- Target h/w address
  - Field left blank in the ARP request, filled in the ARP reply by target node
- Target node fills in missing address, swaps the target and sender address pairs, and changes operation to a reply

ARP Cache

- ❑ MAC addresses once known are cached, process not repeated always
  - ➢ When sending a packet, IP looks in its cache for a binding; if found, no broadcast required
  - ➢ Cache entries have a timeout period (typically 20 min)

- ❑ ARP can find mappings of machines only within the same (sub) network as the source node
  - ➢ Proxy ARP routers can be used for multiple connected LANs

Proxy ARP



The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.

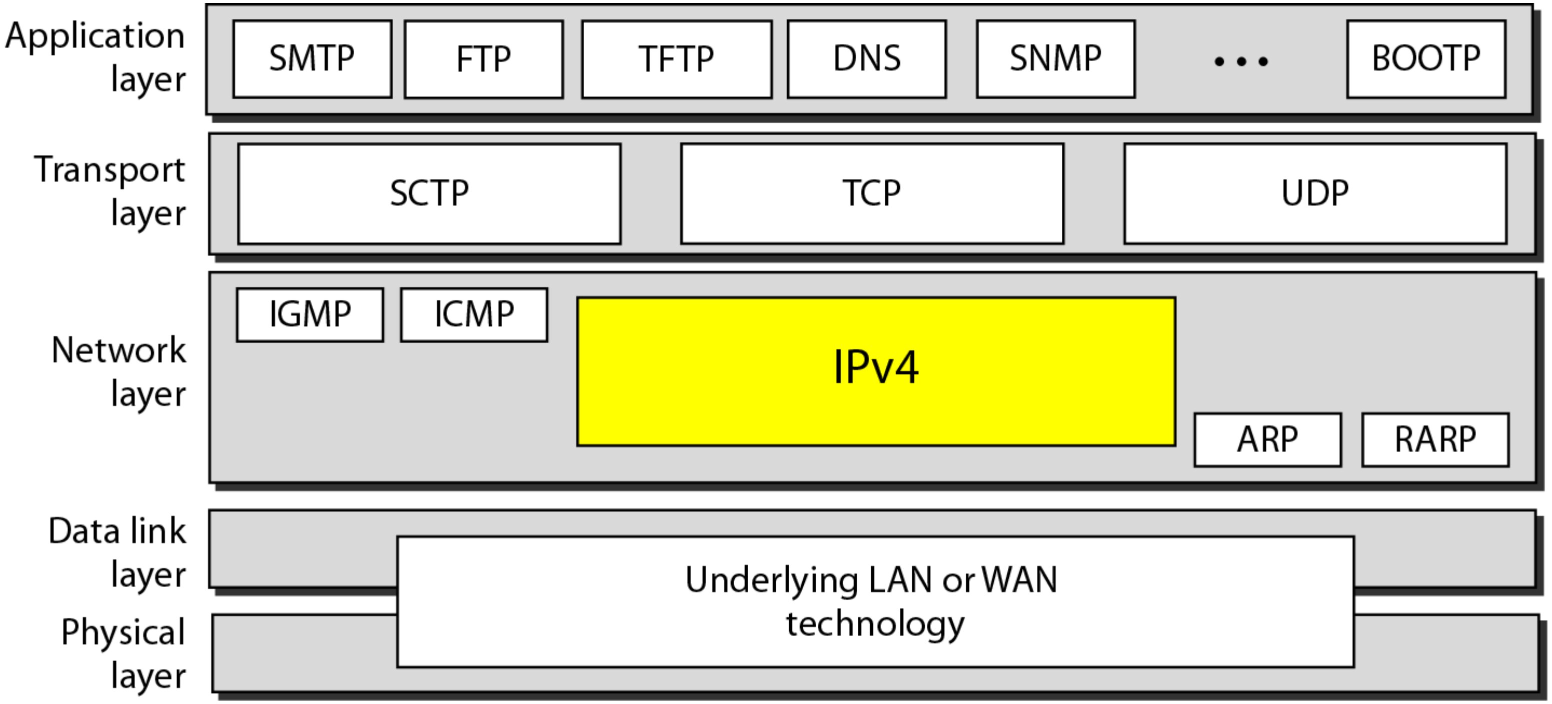
The proxy router is willing to accept packets to be sent to any host on the other subnet.

RARP

- ❑ Reverse of ARP: gets IP address, given hardware address
  - ➢ Can be used by diskless machines / small embedded systems which need to transfer files from some remote server, to obtain their initial boot image
  - ➢ Already has a hardware address, but needs an IP address for file transfer
- ❑ RARP uses same packet format as ARP
- ❑ Requesting node broadcasts RARP request
- ❑ RARP servers (one or more in a network) reply, giving the IP address

IPv4 – Internet Protocol

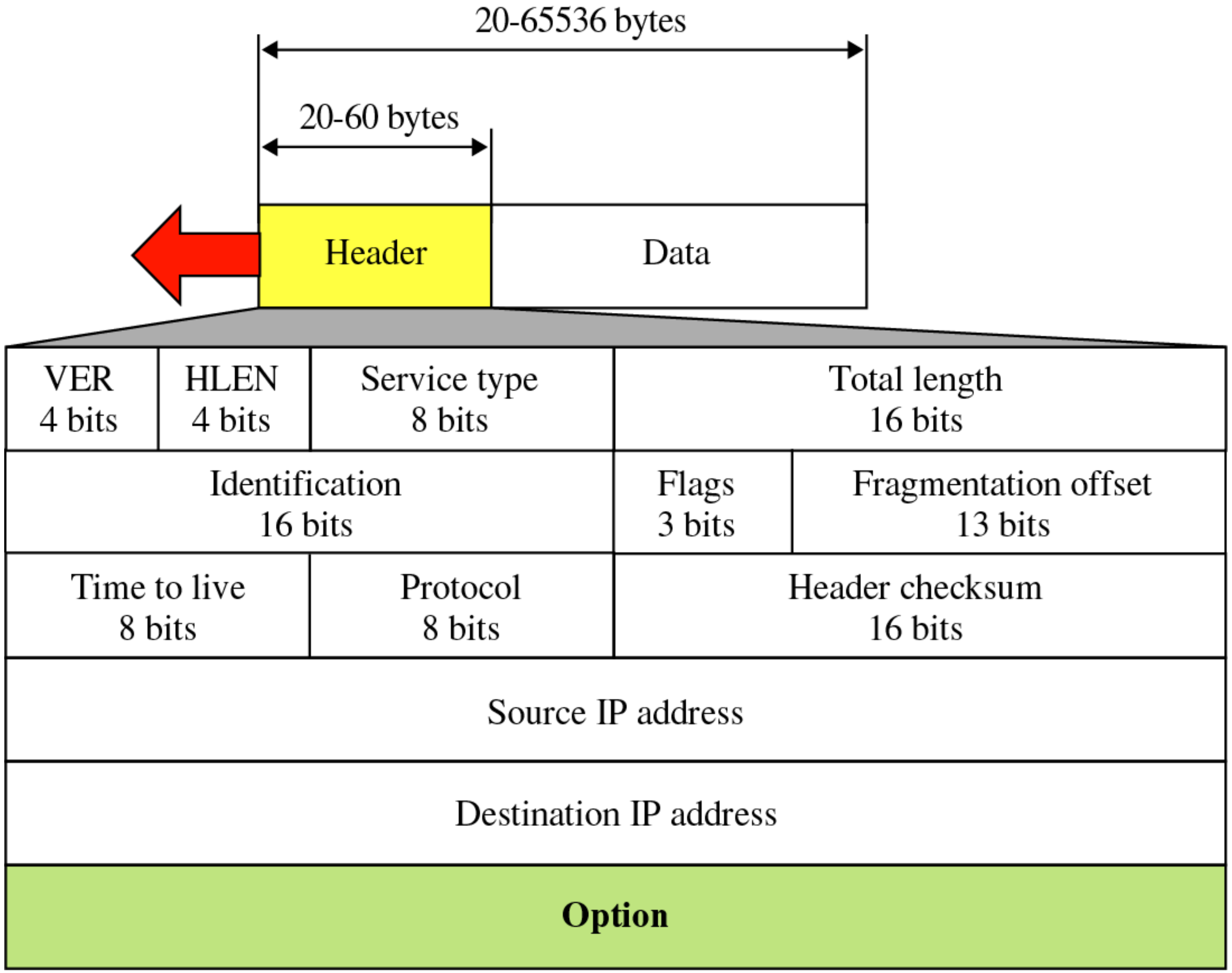
Layering of Protocols



IPv4

- Most widely used network layer protocol in TCP/IP suite, IP defined originally in RFC 791
  - Connectionless: no explicit connection setup / termination phase before / after data transfer
  - Message broken up into packets, each packet switched independently between routers
  - IP header attached to each packet
  - Flexible, robust to failures, no unnecessary overhead
  - Unreliable, best-effort service: Packets can be lost, duplicated, come out-of-sequence
- Main issues handled at network layer: routing and fragmentation / reassembly

IP datagram



20-65536 bytes

20-60 bytes

Header

Data

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

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

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

Source IP address

Destination IP address

Option

Fields in IP Header

- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Typically “4” (for IPv4), sometimes “6” (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically “5” (IPv4 header is at least 20 bytes)
- Total length (16 bits)
  - Number of bytes in the entire packet (including header and data)
  - Maximum size is 63,535 bytes (2^{16} -1)

Fields in IP Header (contd.)

- ❑ Time-To-Live (8 bits)
  - ➢ Used to identify packets stuck in forwarding loops   and eventually discard them from the network (prevents a data packet from circulating indefinitely)
  - ➢ Used to control the max number of hops(router) a datagram can be visited. Source hosts put a number approx. 2 times than the number of router, each router decrements by 1. Once TTL is 0, it is discarded.

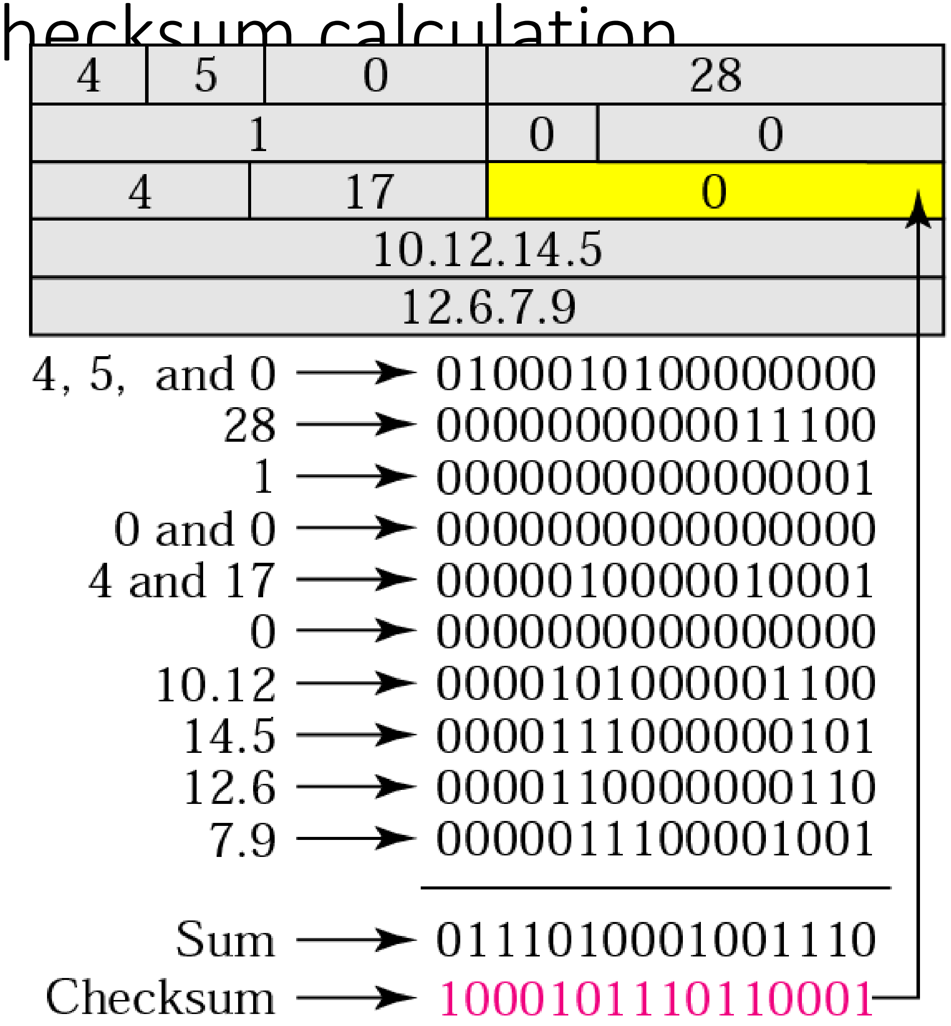
- ❑ Protocol (8 bits)
  - ➢ Identifies the higher-level protocol for which this IP packet is meant
    - ✓ E.g. “6” for the TCP, “17” for UDP

Checksum on the IP Header

- Checksum (16 bits)
  - Break IP Header into 16-bit units (checksum field = 0)
  - Sum these units, using 1's complement arithmetic
  - Take 1's complement of the sum
- Checksum verified and re-computed at each router and at the final destination
  - If mismatch, discard corrupted packets
  - Sending host will retransmit the packet, if needed

- IP checksum computed only on IP header, NOT on the data in the packet

Example of checksum calculation



4	5	0	28
1		0	0
4	17	0	
10.12.14.5			
12.6.7.9			

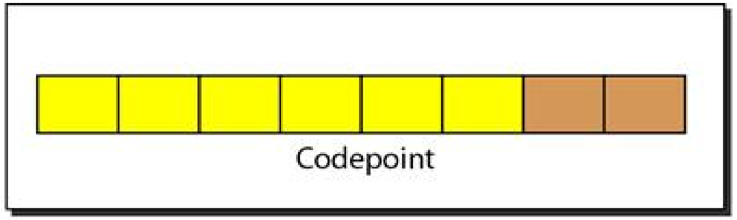
4, 5, and 0	→	0100010100000000
28	→	0000000000011100
1	→	0000000000000001
0 and 0	→	0000000000000000
4 and 17	→	0000010000010001
0	→	0000000000000000
10.12	→	0000101000001100
14.5	→	0000111000000101
12.6	→	0000110000000110
7.9	→	0000011100001001
Sum	→	0111010001001110
Checksum	→	1000101110110001

Type of service field (8 bits)

- 3-bit precedence field: datagram precedence with values 0 (normal data) – 7 (network control)
  - Routers may give more precedence to control information than to normal data
- Three 1-bit fields specifying desired service qualities (as desired by higher layer protocols)
  - D bit: request to minimize delay
  - T bit: request to maximize throughput
  - R bit: request to maximize reliability
  - Only one bit can be set, none set implies ‘normal service’
- Last 2 bits unused

Differentiated Services

- In late 1990, IETF redefined the field to provide Differentiated service (DiffServ) use Quality of Service (QoS)
- DiffService Capable Node uses DHCP 6 bits as an index to a table defining packet handling mechanism for the current packet being processed



Differentiated services

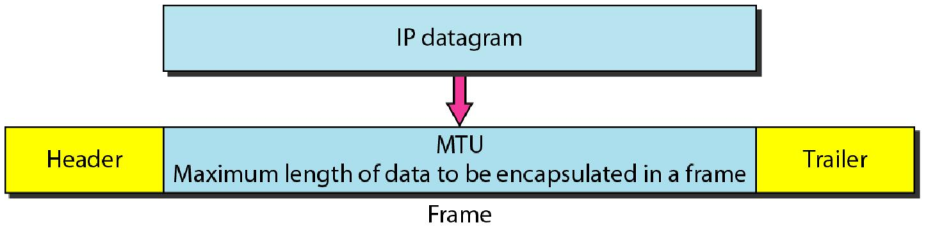
Data in an IP datagram

- ❑ Carries user data from higher transport layer
- ❑ Length: in units of bytes (octet)
- ❑ Maximum total length of datagram (header plus data): 65,535 bytes (2^16 – 1)
  - ➢ Total length is a 16-bit field

- ❑ However, such a large datagram is not usually allowed at lower layers
  - •E.g. Ethernet allows MAC frames of up to 1518 bytes

Fragmentation of IP datagram

- ❑ Maximum transfer unit (MTU)
  - ➢ Any network technology has a MTU (e.g. for Ethernet, higher layer PDU can be at most 1500 bytes)



Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Fragmentation of IP datagram

- When a router has to transmit a datagram too large for the MTU of the outgoing link, datagram is fragmented

- A single IP datagram can arrive at the destination as multiple fragments
  - Fragments re-assembled at the destination node
  - Intermediate routers do NOT re-assemble fragments

Two terms: Packet vs Datagram

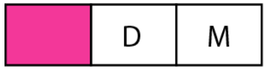
- An IP datagram is the unit of end-to-end transmission at the IP layer (before fragmentation & after reassembly)

- A packet is the unit of data passed between the IP layer and the link layer

- A packet can be a complete IP datagram or a fragment

Fields used for fragmentation

- Identification
  - Identifies each datagram uniquely originated from a host – managed by a counter at IP layer
  - Destination node uses the <source IP, identification> to identify which arriving fragment belongs to which datagram
- Flag: 3-bit field
  - DF: do not fragment. If source sets to 1
    - routers send this datagram un-fragmented if possible, otherwise
    - discard and may send an ICMP message which indicates the condition "*Packet too Big*"
  - MF: are there more fragments (of this datagram) after this one?
  - Third bit is reserved
- Fragment offset

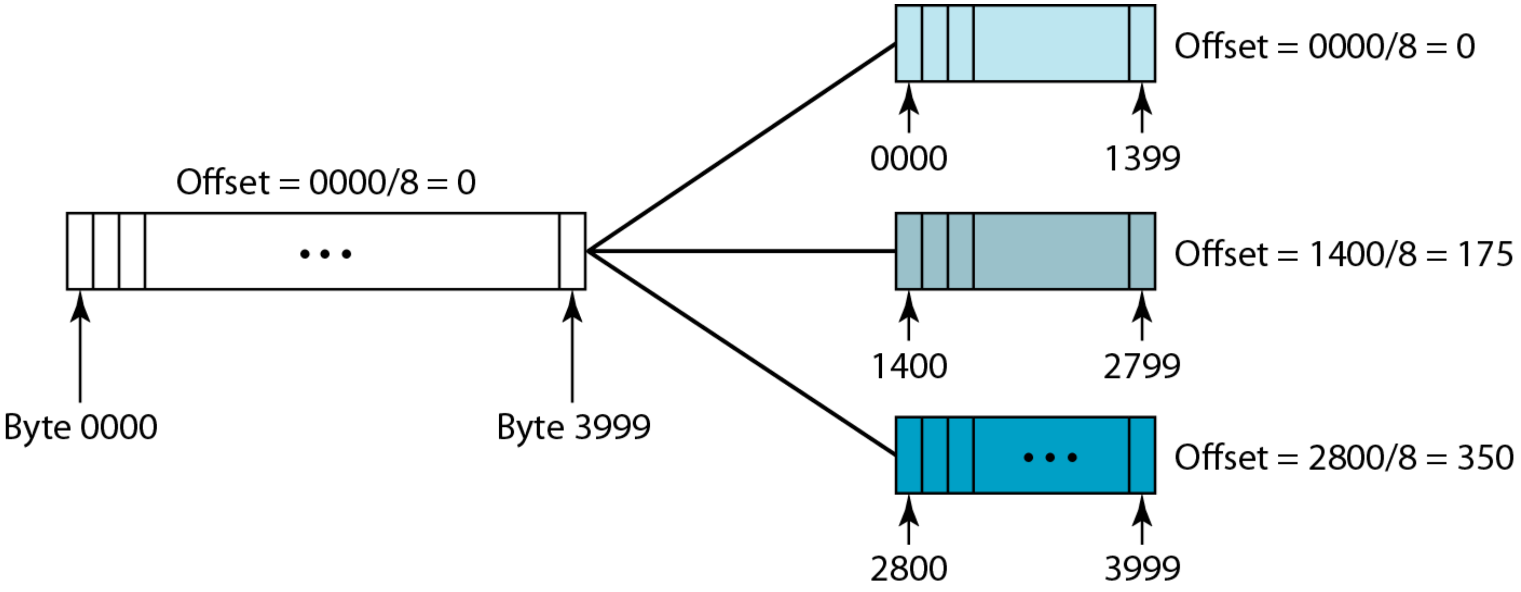


D: Do not fragment
M: More fragments

  - Offset of the data contained in this fragment, in the data contained in the actual IP datagram sent by source
  - Given in units of 8-byte blocks

Fragmentation – an example

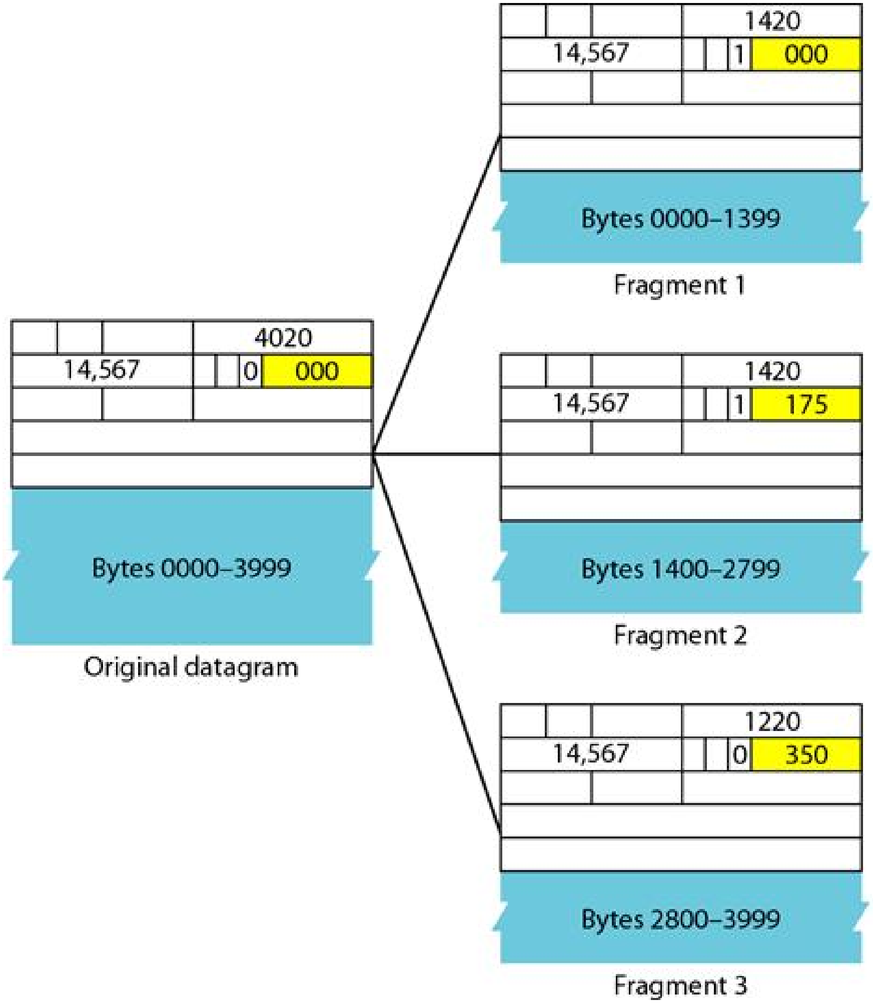
- ❑ 4020 byte datagram including 20 byte IP header
- ❑ Length of data in datagram: 4000 bytes
- ❑ At some intermediate router next hop's MTU is 1420 byte



Fragmentation – an example (contd…)

❑ 4020 byte datagram including 20 byte IP header. Length of data in datagram: 4000 bytes

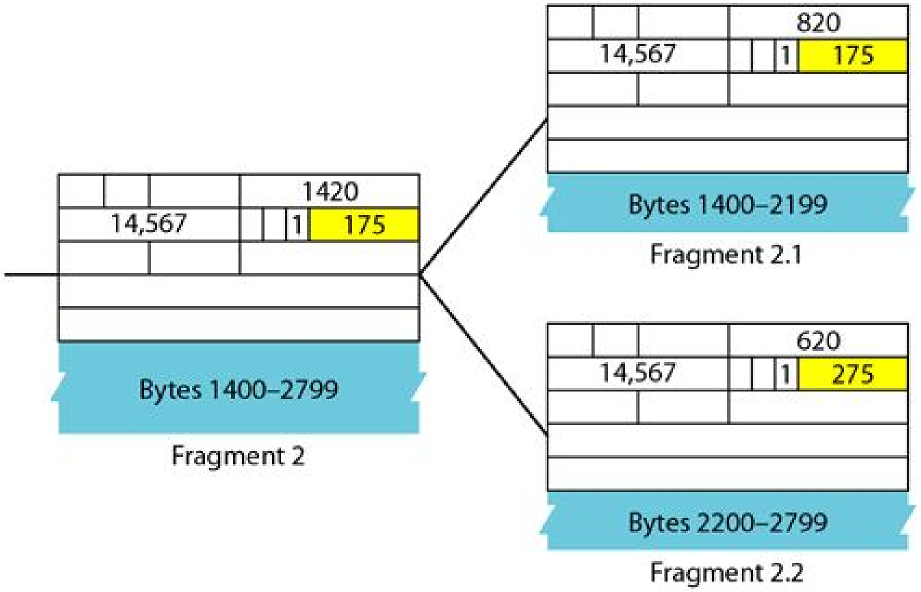
❑ At some intermediate router next hop's MTU is 1420 byte.



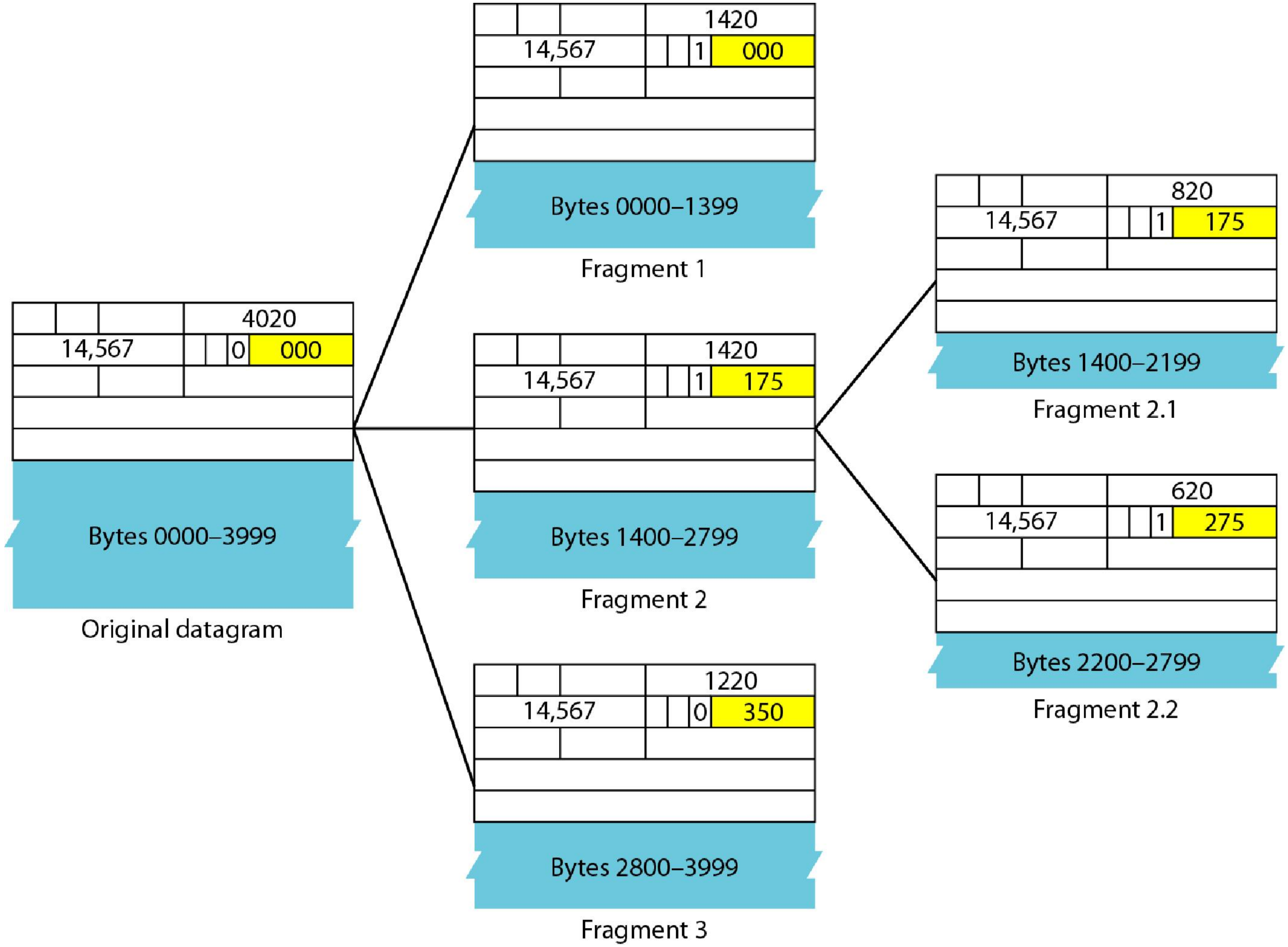
Fragmentation – an example (contd…)

❑ What happens if some more fragmentation needed at another intermediate router ?

✓ For Example if another intermediate router (where Fragment-2 reached and to be forwarded) next hop’s MTU of is 820



Fragmentation – an example (contd…)



Dealing with failure

❑ Receiver starts reassembly timer when first fragment of a datagram is obtained

- ➢ If timeout before all fragments arrive, discard all fragments of this datagram
- ➢ Until the IP layer of the receiver has received all fragments of a datagram, it cannot hand the entire datagram to the higher layer

❑ IP does not guarantee delivery

- ➢ Responsibility of higher layer to re-transmit packet
- ➢ Routers attempt to inform source if packet discarded

Options field in IP header

❑ Options included primarily for network testing or debugging

✓ Examples

- Source routing
- Record route

How a router handles an IP datagram

- ❑ When a router gets an IP datagram
  - ➢ Extract data part, by stripping off IP header
  - ➢ Find outgoing interface using dest. IP and routing table
  - ➢ If data part > MTU of outgoing link to next hop
    - ✓ Fragment data part, put each fragment into a separate IP datagram
    - ✓ Put an IP header within each IP datagram
    - ✓ Copy fields: version, Type of Service, identification, protocol, source address, destination address, some options
    - ✓ Compute length and header checksum individually for each fragment
    - ✓ Put suitable flags and frame offset in each fragment
    - ✓ Put 1 less than TTL of original datagram as TTL in each fragment

Internet Control Message Protocol

ICMP

ICMP

- ❑ Every Network layer implementation must implement ICMP, along with IP and ARP
  - ➢ A required support protocol at the IP layer
- ❑ Used for reporting errors back to the source of an IP packet or for monitoring / measurement / feedback
  - ➢ When a node detects an error, an ICMP packet sent back to the source
  - ➢ Only error *reporting*, no error correction; correction is left to the source node

- RFC 792 and RFC 1122

ICMP (contd.)

- ICMP packet
  - Contains ICMP header and may contain other information depending on type of message
  - Carried in data portion of an IP packet
  - The IP packet contains a IP header and is routed normally back to the source

- Examples of use of ICMP
  - Echo reply (to see if a host is up)
  - Subnet mask request and reply (among routers)
  - Router informs source about packet drop (may be due to unreachable destination, TTL exceeded, congestion)

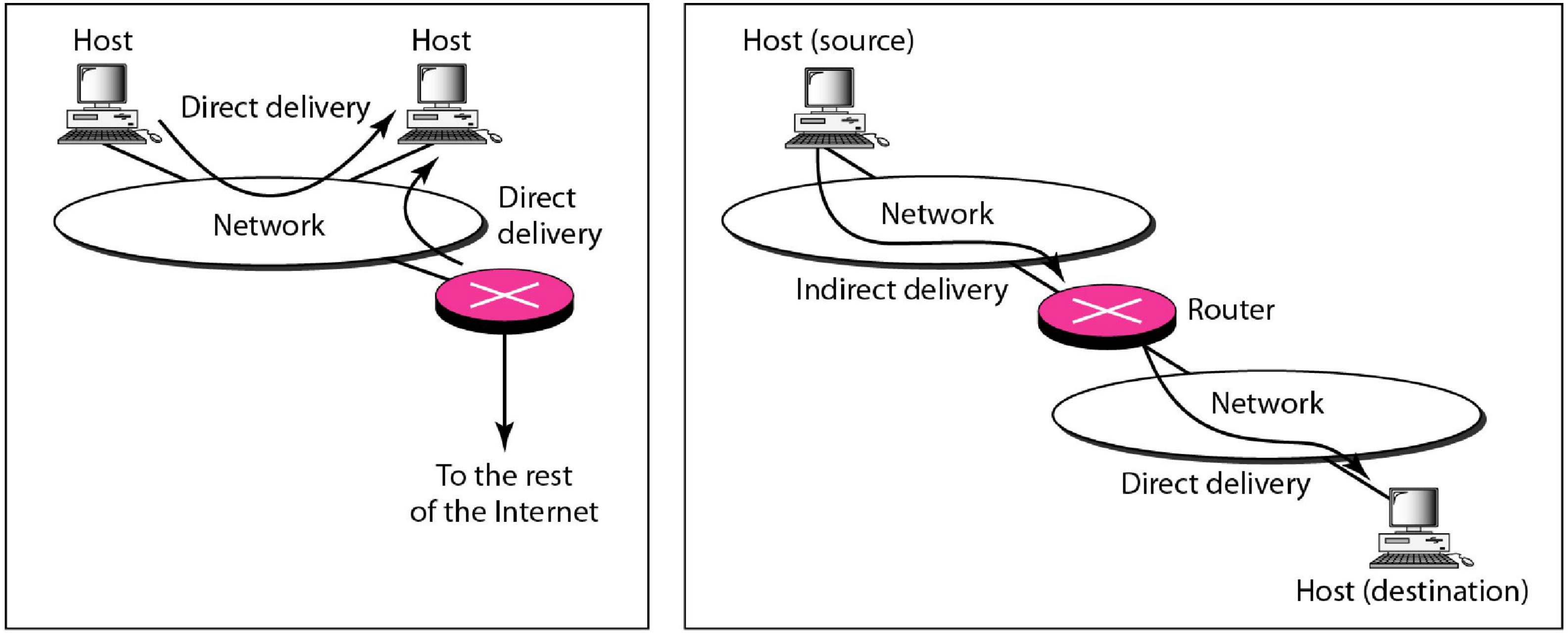
Computer Network and Distributed Systems

Routing

Routing

- The process of sending a message from a source to a destination through intermediate nodes
- A route – a path from the source to the destination
- Routing protocol / algorithm
  - Finds one or more routes between a source and a destination
  - Usually stores routes found in a routing table for future use, each node has a routing table
  - Routing table has to be updated if this node comes to know about new routes or changes in existing routes

Direct and indirect delivery



a. Direct delivery

b. Indirect and direct delivery

Goals of routing protocols

- ❑ Correctness
- ❑ Optimal
- ❑ Efficiency/Low overhead
- ❑ Robust (able to handle failure of node or link)
- ❑ Rapid convergence when network conditions change

- ❑ How to compare among routes?
  - ➢ Network usually modeled as a weighted graph
  - ➢ Cost of path in the weighted graph
  - ➢ Minimum number of hops (if all links have same cost)

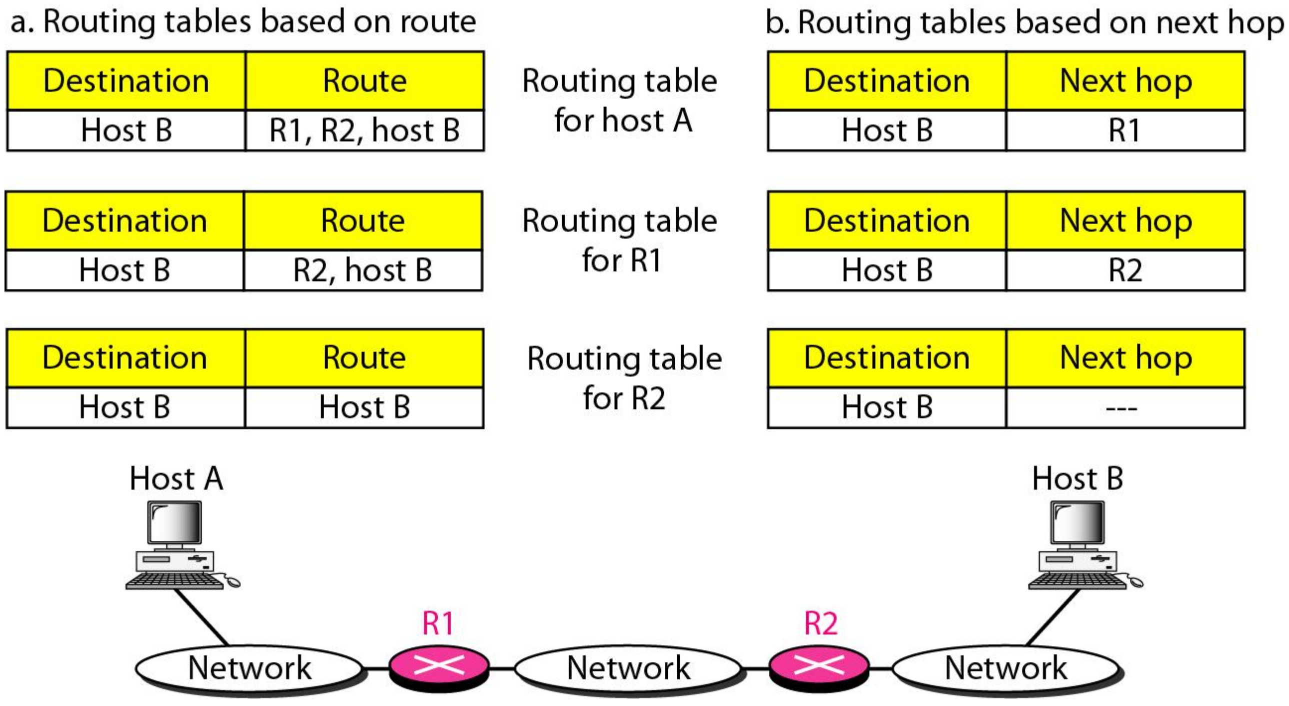
When / where are routing decisions taken?

- ❑ When
  - ➢ During circuit setup (circuit switching / virtual circuit switching), or
  - ➢ During switching of each datagram (packet switching)

- ❑ Where
  - ➢ Distributed: made by each node when it receives a packet to forward, each node collects info from other nodes to learn best route
  - ➢ Centralized: one node decides routes for every node
  - ➢ Source routing: source node decides route and puts complete path to destination in packet

Route method versus next-hop method

Forwarding : Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

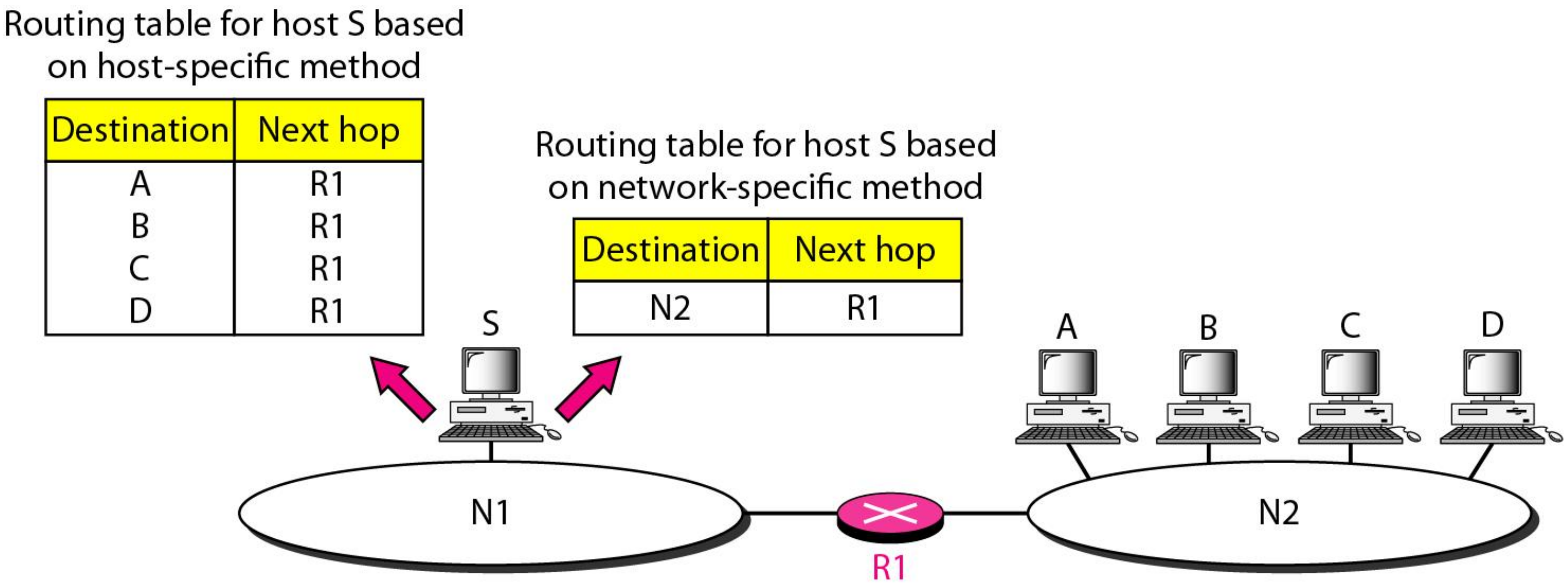


a. Routing tables based on route

b. Routing tables based on next hop

Destination	Route		Destination	Next hop
Host B	R1, R2, host B	Routing table for host A	Host B	R1
Destination	Route		Destination	Next hop
Host B	R2, host B	Routing table for R1	Host B	R2
Destination	Route		Destination	Next hop
Host B	Host B	Routing table for R2	Host B	---

Host-specific versus network-specific method



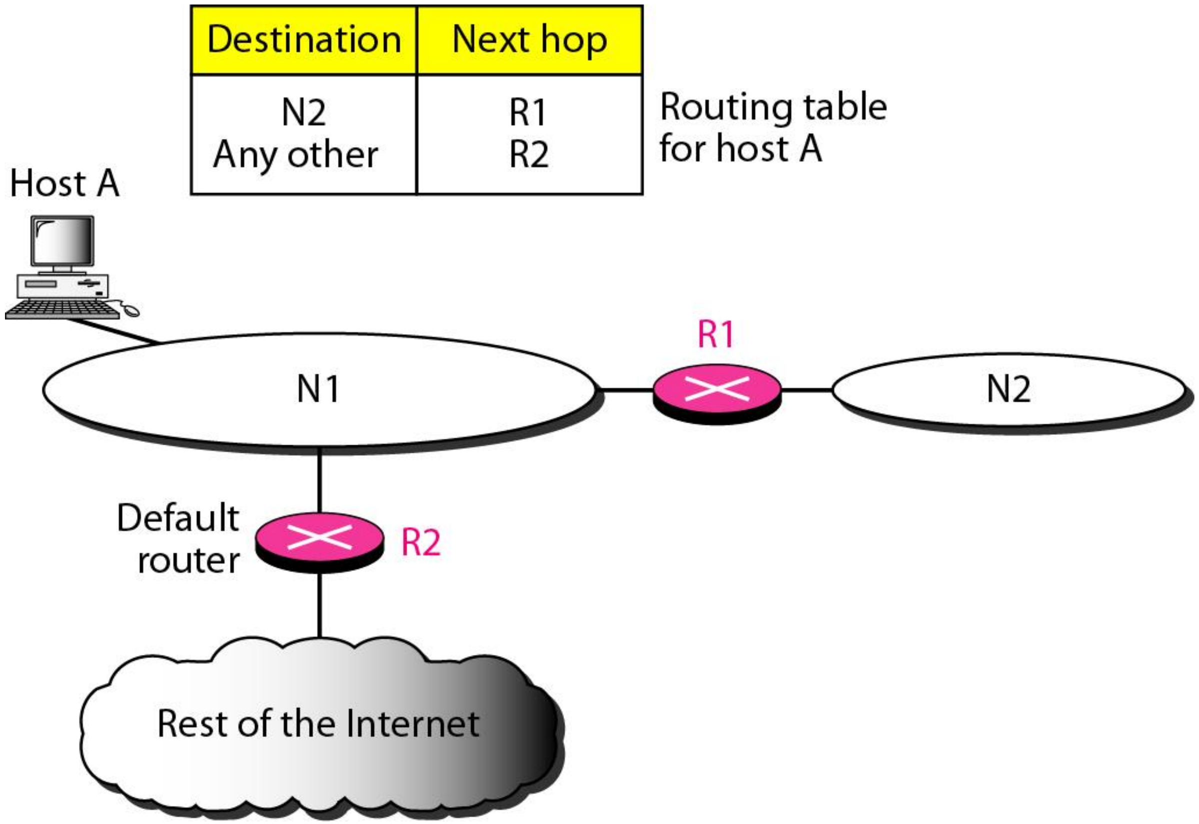
Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

Destination	Next hop
N2	R1

Default method



Destination	Next hop
N2	R1
Any other	R2

Routing table for host A

Routing basics

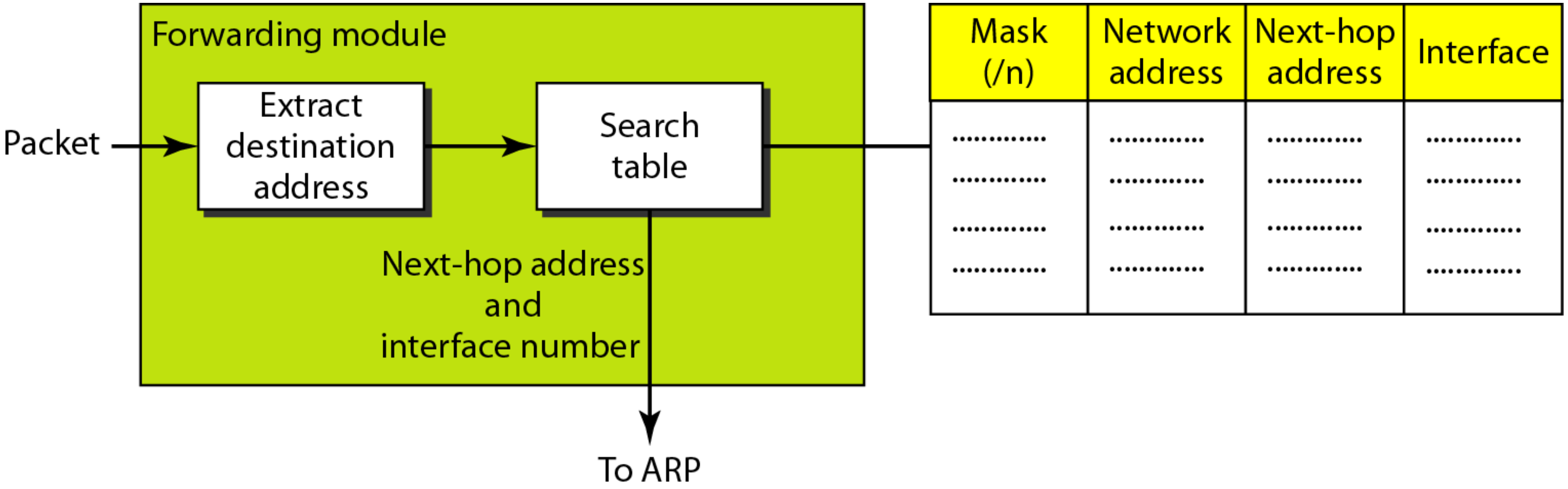
- Routing table usually contains
  - Network addresses (in CIDR notation, or in explicit network and mask form)
  - Host-specific routes (32 bit prefixes) i.e. IP address of individual destination hosts
  - Default entry (both network and subnet mask 0.0.0.0)
  - And a next hop address for each of them

- Usually a routing cache used
  - Contains recent routing decisions
  - Destination address first looked up in cache
  - If not found, longest-prefix match done in routing table

Routing procedure (in CIDR)

```
Extract destination IP address D from the datagram
for each entry in routing table do
    N ← D bitwise AND subnet mask
    verify if N matches the network address of the entry
    # maybe partial match i.e. only some prefix bits match
end for
if no match found
    declare routing error
else if one or more matches found
    select match with longest prefix
    route packet to the next-hop specified in this entry
end if
```

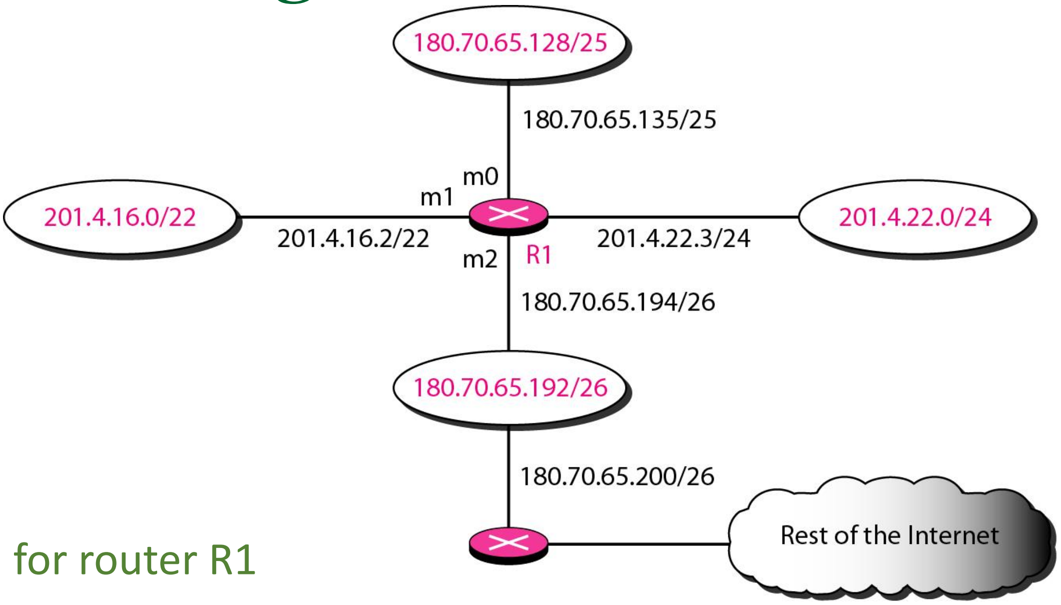
Simplified forwarding module in classless address


Forwarding module
Packet
Extract destination address
Search table
Next-hop address and interface number
To ARP

Mask (/n)	Network address	Next-hop address	Interface
....	....	....	....
....	....	....	....
....	....	....	....
....	....	....	....

In classless addressing, we need at least four columns in a routing table.

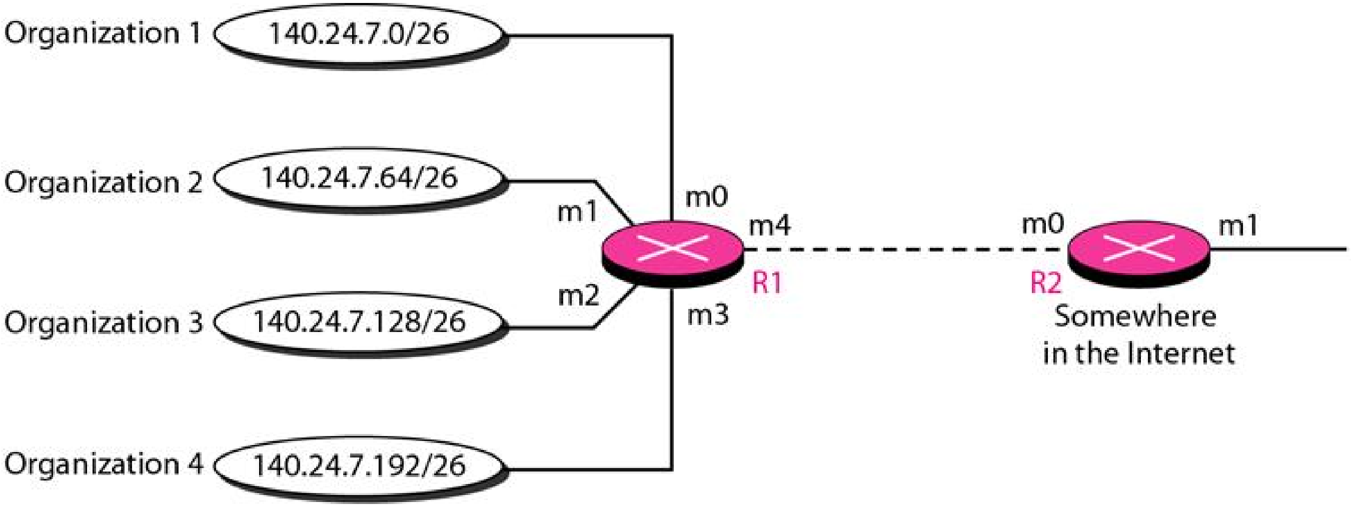
A typical Network Configuration



Routing table for router R1

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Address aggregation



Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Addr of R2	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	Default	m1

Routing table for R2

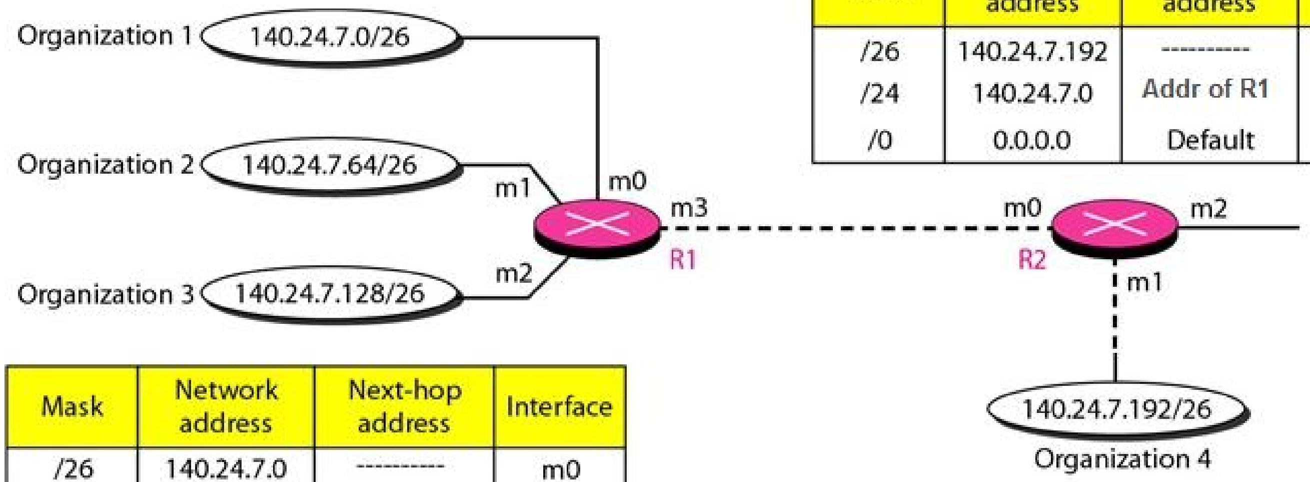
Here four organizations geographically situated close to each other, that's why this kind of configuration is possible. What happens if they are not ?

Longest mask matching

Suppose a packet arrives at router R2 for org-4 with destination address 14.24.7.200. How it will be routed?

Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m1
/24	140.24.7.0	Addr of R1	m0
/0	0.0.0.0	Default	m2



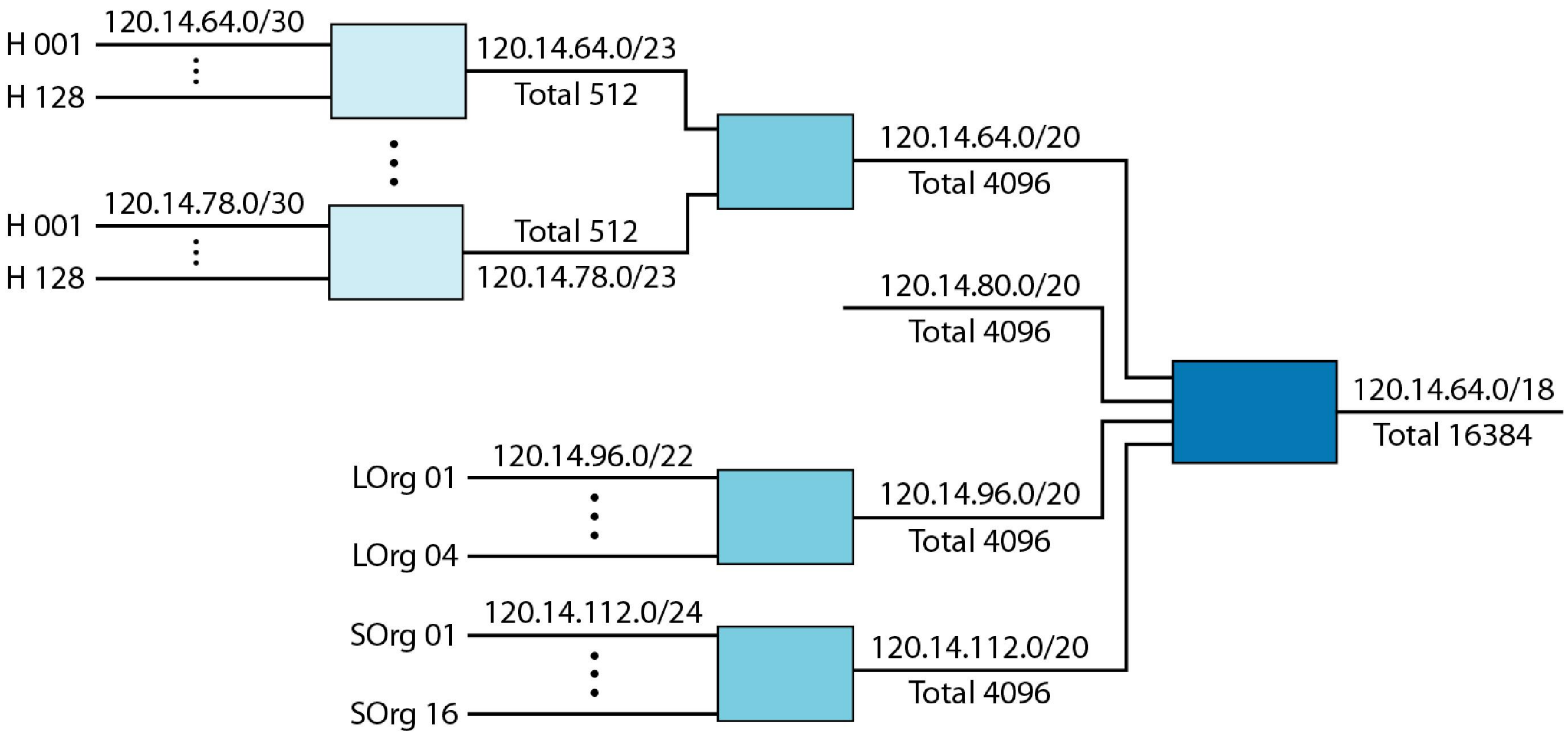
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Addr of R2	m3

Routing table for R1

What happens if the forwarding table is not sorted with the longest prefix first ?

Hierarchical routing with ISPs

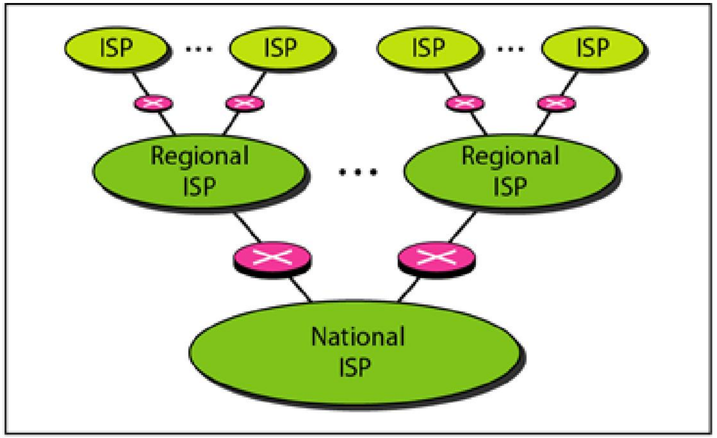
- To solve the problem of prolonged routing tables, a sense of hierarchy in the routing tables can be created.



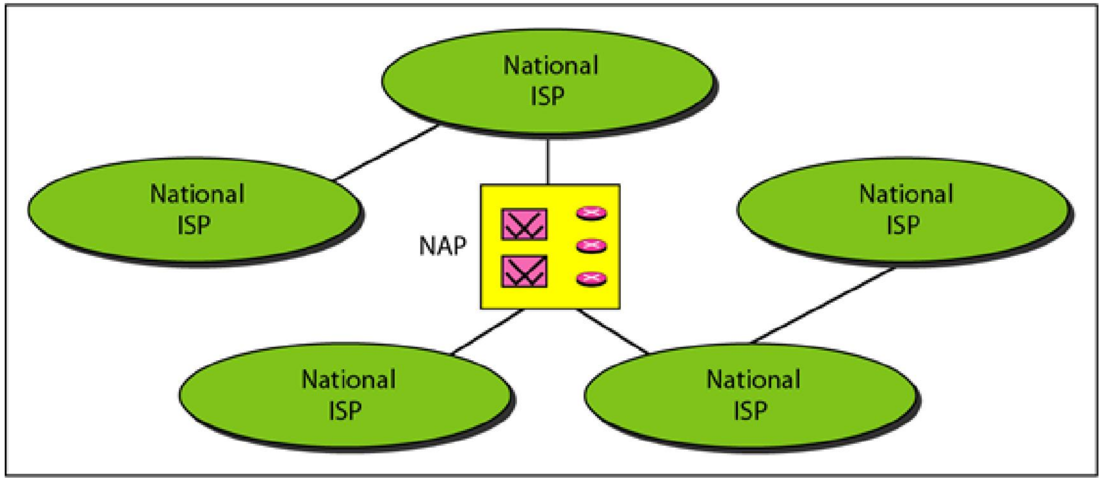
Geographical Routing

- To decrease the size of forwarding table even further, the hierarchical routing is extended to geographical routing
- Entire address space is divided into few large blocks – assign a block to America, a to Europe, a block to Asia and so on
  - The router of ISPs outside Europe will have only one entry for packets to Europe in their forwarding table
  - The routers of ISPs outside North America will have only one entry for packets to North America in their routing tables.
  - And so on.

Hierarchical organization of the Internet



a. Structure of a national ISP

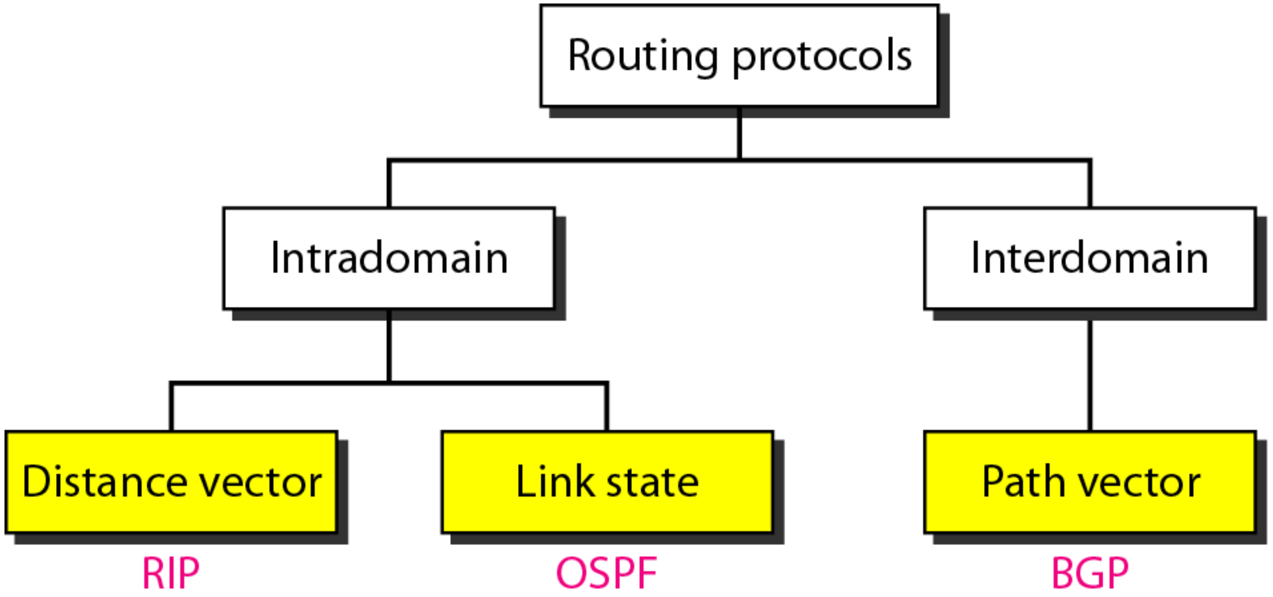


b. Interconnection of national ISPs

The Internet Structure

- Internet has changed from tree-like structure, with a single backbone, to a multi-backbone structure run by different private corporation
- The Internet is so large that one routing protocol cannot handle the task of updating the routing tables of all routers
- Internet is divided into autonomous systems (AS), that is a group of networks and routers under the authority of a single administration (ISP)
  - Each AS can run a routing protocol that meets its need
  - Called *intra-AS routing Protocol* or *intradomain routing protocol* or *interior gateway protocol (IGP)*
  - Two common such protocols are RIP and OSPF - AS free to use any
- However the global Internet runs a global protocol to glue all ASs together
  - Called *inter-AS routing protocol* or *interdomain routing protocol* or *exterior gateway protocol (EGP)*
  - Only such protocol - BGP

The Internet-Protocol (IP) Routing protocols



- ❖ RIP (Routing Information Protocol)
- ❖ OSPF (Open Shortest Path First)
- ❖ BGP (Border Gateway Protocol)

Types of Routing

- ❑ Fields of a routing table
  - ➢ Destination node, next hop, metric, other fields
  - ➢ Metric: estimate of the cost of this route to the destination (through the next hop)

- ❑ Types of routing protocols
  - ➢ Fixed or static
  - ➢ Random
  - ➢ Flooding
  - ➢ Dynamic or adaptive

Fixed / Static routing

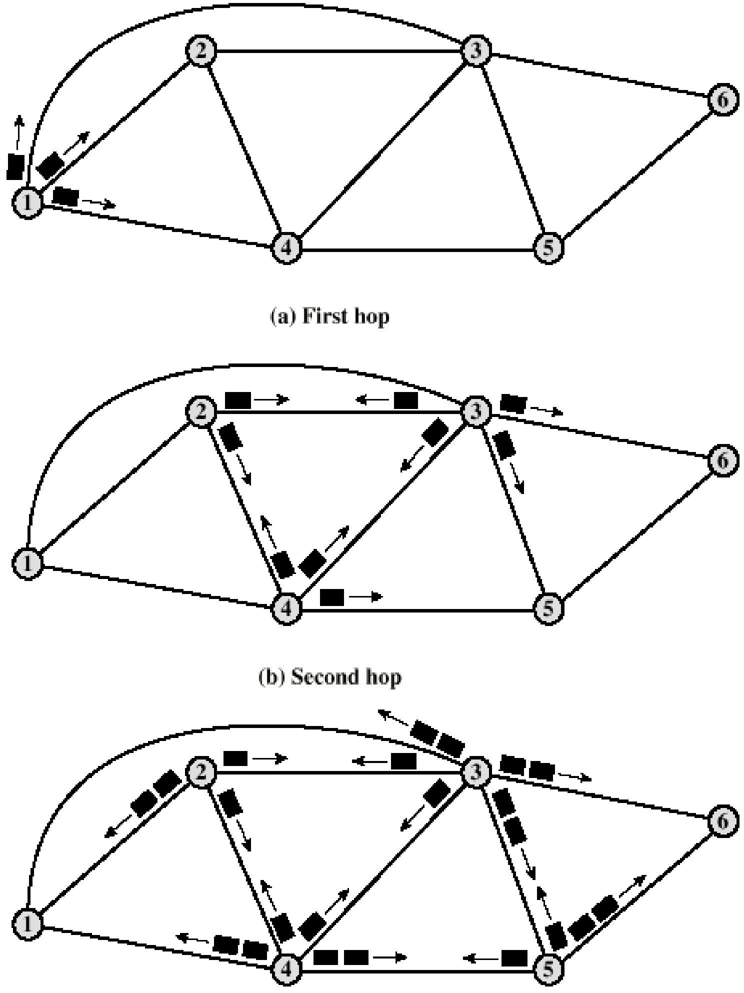
- ❑ Single permanent route for each source to destination pair
- ❑ Routing tables created & updated manually
  - ➢ Routing table is fixed unless manually changed again
  - ➢ No dynamic update when network conditions change
- ❑ Fine for very small networks

Flooding

- Packets sent by a node to every neighbor, except to the one from which the packet was received

- Eventually a number of copies will arrive at destination

- Each packet is uniquely numbered so duplicates can be discarded



(a) First hop

(b) Second hop

(c) Third hop

Flooding

- ❑ Advantages
  - ➢ All possible routes are tried, so very robust
  - ➢ Can work around failed links/nodes
  - ➢ All nodes are visited, so useful to distribute information
  - ➢ No network information needs to be stored at any node

- ❑ Disadvantages
  - ➢ No routes remembered (no routing table)
  - ➢ Too many copies of a packet may be sent

Refinements in flooding

- ❑ Nodes can remember packets already forwarded to keep network load in bounds

- ❑ To handle loops in the network, use hop count

- ❑ At least one packet will have taken minimum cost route (e.g. minimum hop count) to reach destination
  - ➢ Can be used to set up virtual circuit

- ❑ Flooding can be used intermittently, say, when a route to an unknown destination is to be found

Random Routing

- Node selects any one outgoing path for retransmission of incoming packet
  - Selection of outgoing path can be random or round robin or based on probability calculation
- If packet does not reach destination within a time interval, select another outgoing link and transmit
- Advantage
  - No network info needed to be stored at any node, no routes remembered
- Disadvantage
  - Large delay possible, used in networks where delay is not a concern
  - No guarantee that the outgoing path selected will lead to the destination

Adaptive routing

- ❑ Routing decisions change dynamically as conditions on the network change
- ❑ Storage of info about network at nodes
  - ➢ Routes saved in routing tables
  - ➢ Routers communicate to update routing tables dynamically when network conditions change

- ❑ Advantages
  - ➢ No manual intervention necessary, aids congestion control, fault tolerance
- ❑ Disadvantages
  - ➢ Complex, routing messages are overhead

Issues for adaptive routing protocols

- How much and what network info to collect
- From whom to collect information (from neighbors or from all nodes, etc)
- When to collect information (once at the beginning, or periodically, etc)

- Direct tradeoff between
  - Overhead and
  - Optimality, robustness, speed of convergence

Families of adaptive routing protocols

- Based on with which other nodes does a node exchange routing information
  - Distance Vector routing protocols
  - Link State routing protocols

Distance Vector Routing Protocol

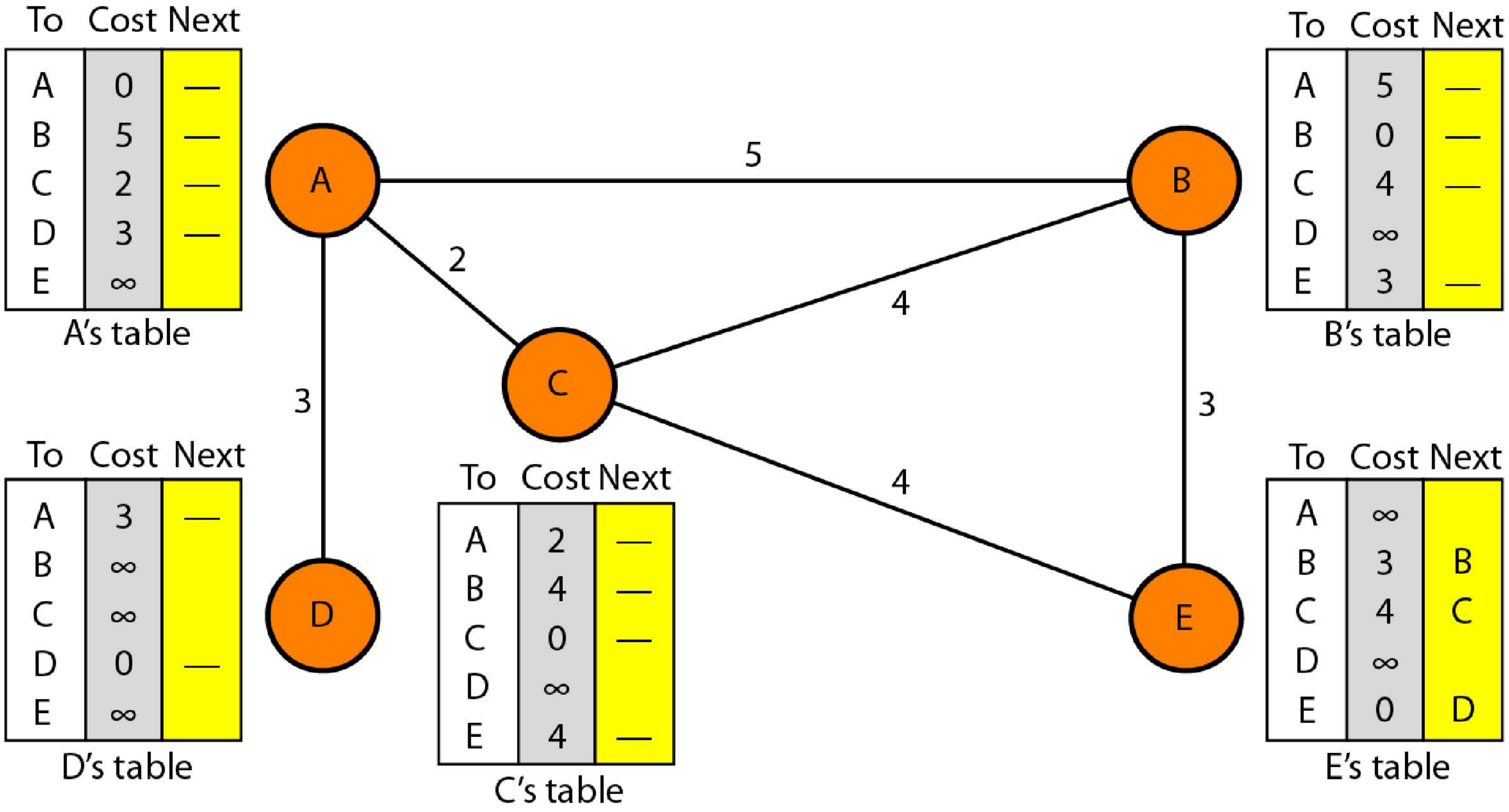
Routing Information Protocol (RIP)

Distance Vector Routing

- Each node N keeps track of its least cost path to all other nodes (all nodes that N knows till now)
  - For each node d, N stores the next hop to reach node d (along the least cost path) and the cost of that path
  - This information stored at N, for all nodes other than N, comprises the distance vector at node N

- Distance vector periodically sent to all nbrs, whether or not vector changed since last send
- If no information received from a neighbor within a time, that link is assumed to be down

Initialization of tables in distance vector routing



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

A's table

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

B's table

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

D's table

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

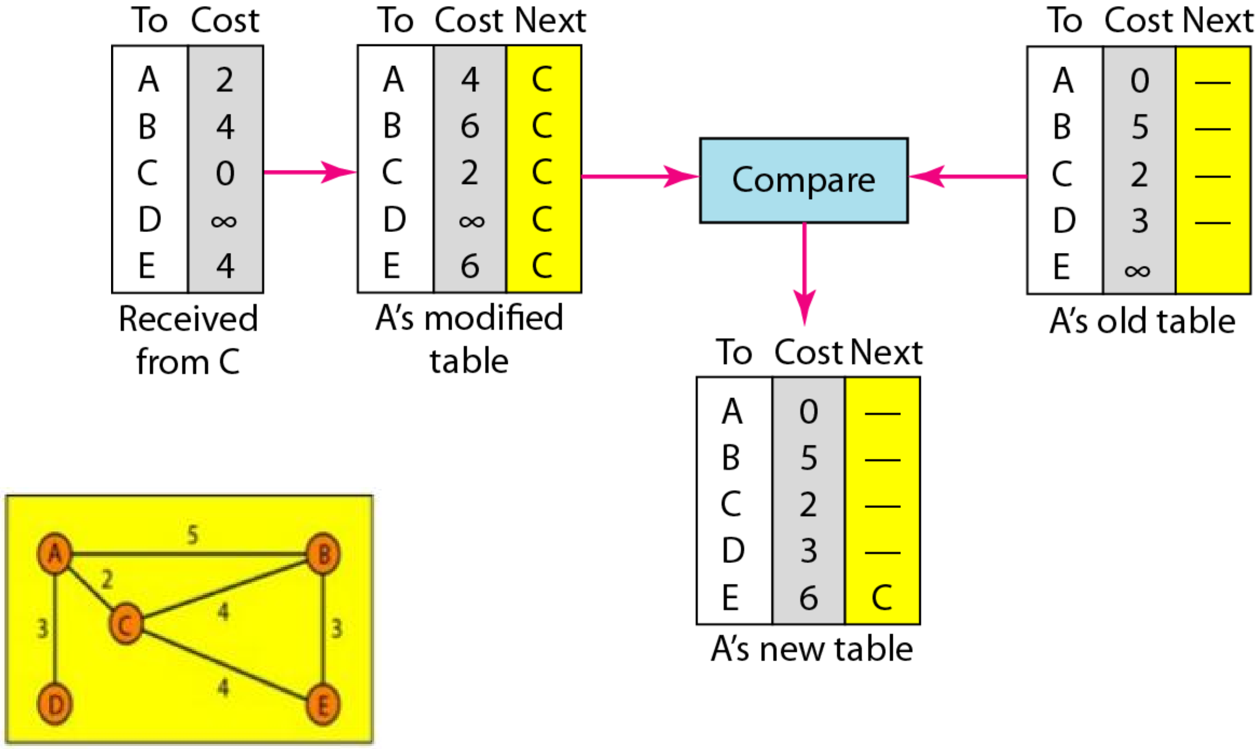
C's table

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

E's table

Think nodes as city and links are path connecting them.

Updating in distance vector routing (only a case)



Received from C

To	Cost
A	2
B	4
C	0
D	∞
E	4

A's modified table

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

A's old table

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

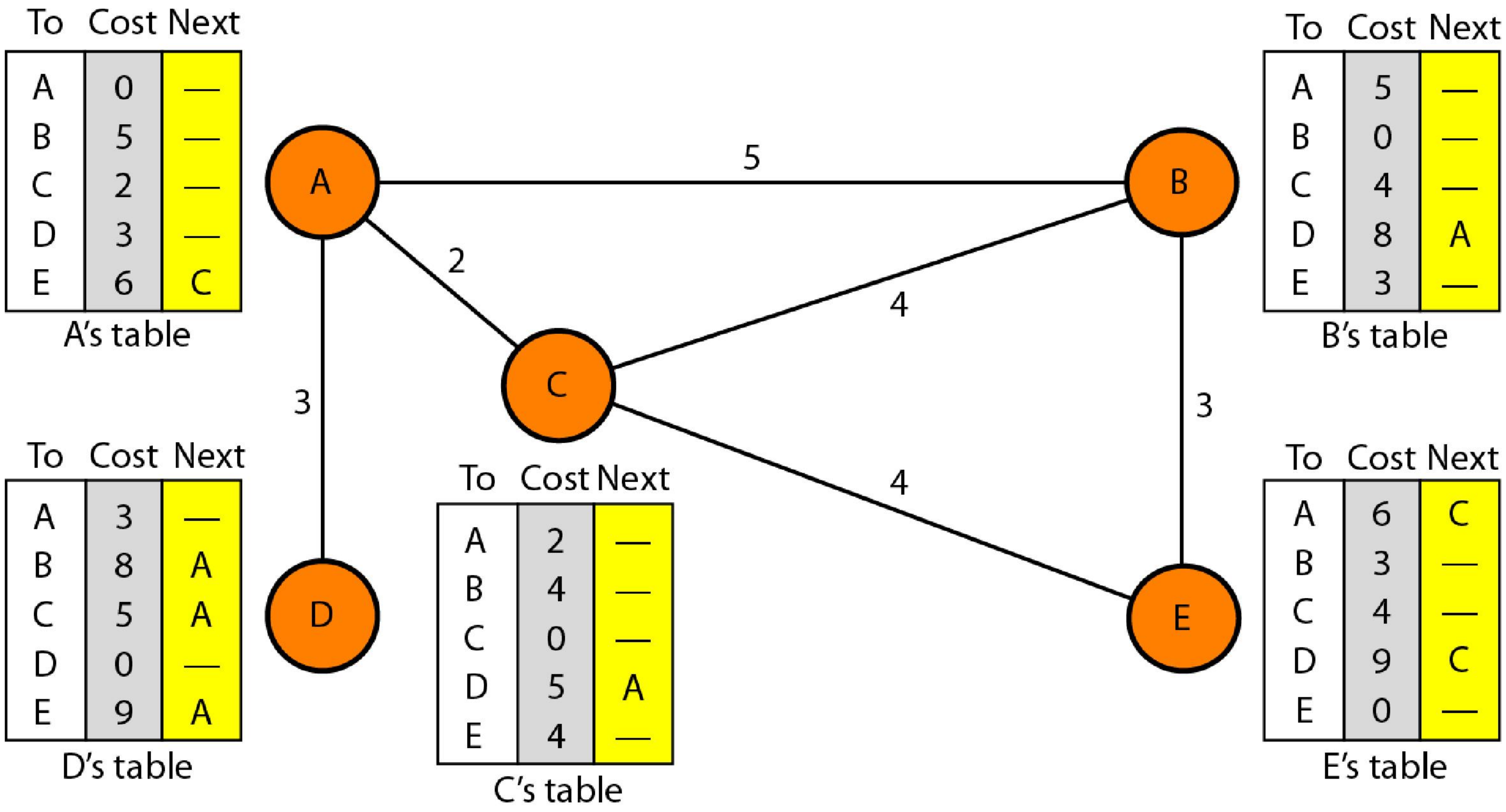
Compare

A's new table

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

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

Distance vector routing tables after convergence



A's table

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

B's table

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

D's table

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

C's table

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

E's table

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

The least cost route between any two nodes is the route with min distance

Detection of a node / link crash

- ❑ Each node sends its distance vector to all its neighbors periodically

- ❑ If nothing received from node N for a certain time, N's neighbor assumes N has crashed
  - ➢ All routes with next hop N are set to have metric INF
  - ➢ Routes with metric INF deleted after 'some' time defined by actual protocol

- ❑ Value of INF defined by the actual protocol being used

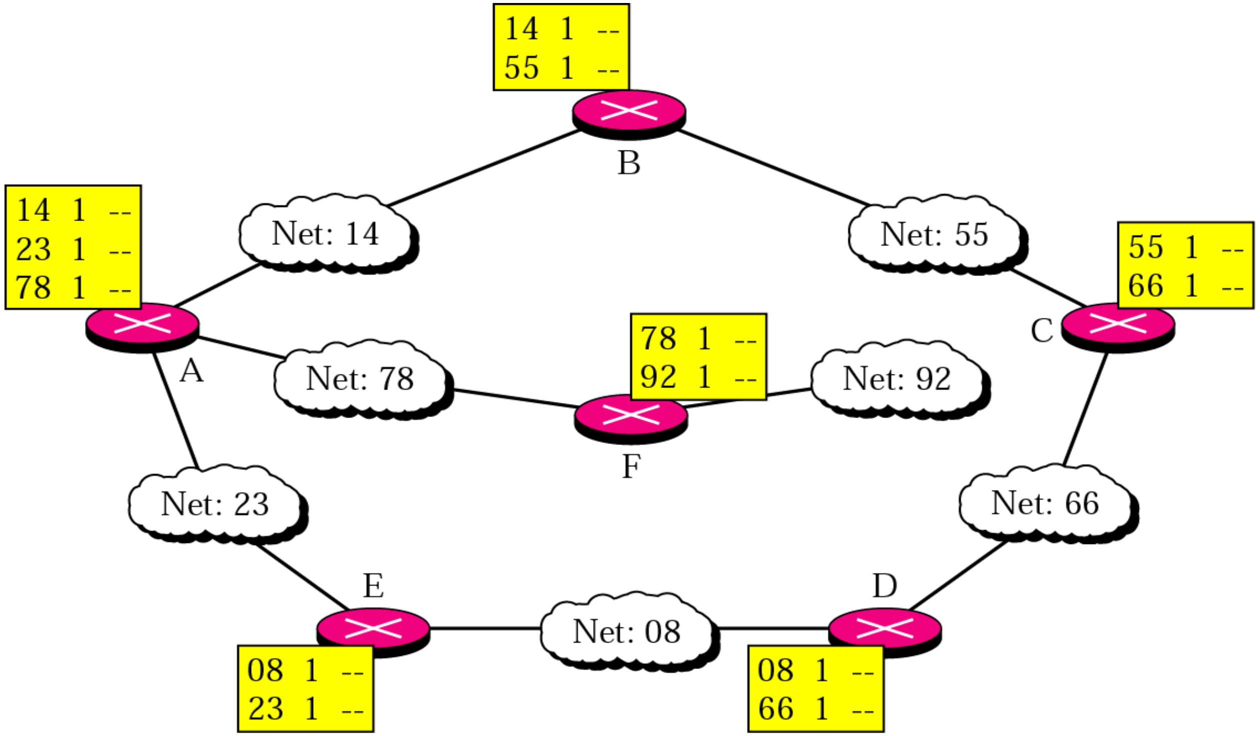
A real-world algorithm based on DVR

- RIP: Routing Information Protocol
  - INF is 16
  - Cost of each link is constant = 1 (as if all links have same cost)
  - Broadcast interval: 30 seconds
  - Route expiry time: 180 seconds
    - ✓ If no message received from a neighbor n for 180 sec, change metric to INF for all routes through n
  - Time to delete a route: 120 sec after metric is set to INF
  - Uses split horizon, hold down, triggered update, split horizon with poissoned reverse (*to be discussed later*)

On receipt of distance vector of node B, at node A in RIP

```
for every entry <d, x, k> in B’s distance vector
    if <d, _, _> is not in A’s routing table
        add <d, B, k+1> to A’s routing table
    else if <d, y, k’> exists in A’s routing table
        if k+1 < k’
            replace <d, y, k’> with <d, B, k+1> in A’s table
        else if y = B      # trust your next hop
            replace <d,B,k’> with <d,B,k+1> in A’s table
        end if
    end if
end for
```

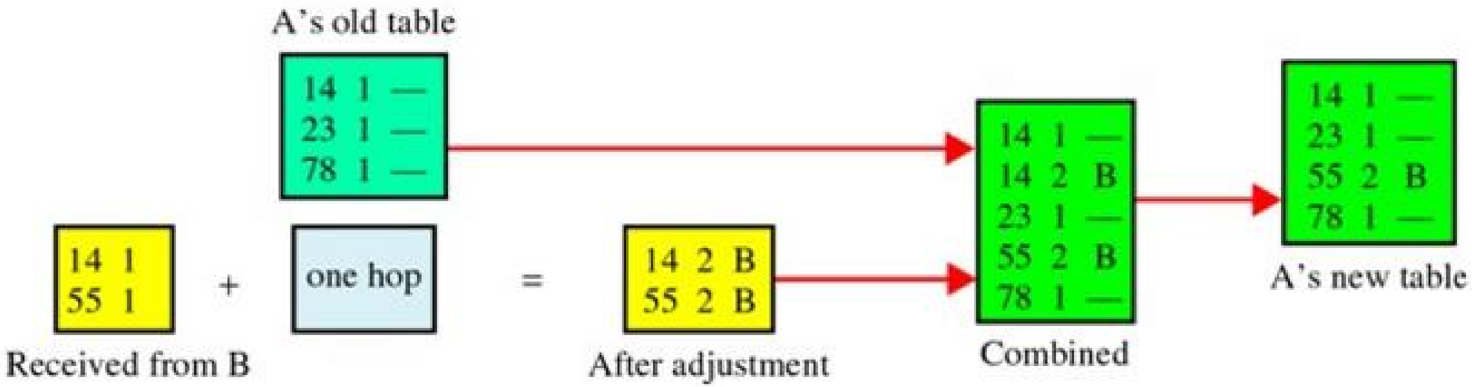
Initial routing tables in an internetwork



- ❑ Each Router initializes a routing table for itself using config files or static routes
- ❑ Table consists only directly attached networks with the hop count set as 1

Note : The next-hop field, which identifies the next router is empty

Updating routing tables of Router A



A's old table

14	1	—
23	1	—
78	1	—

Received from B

14	1
55	1

After adjustment

14	2	B
55	2	B

Combined

14	1	—
14	2	B
23	1	—
55	2	B
78	1	—

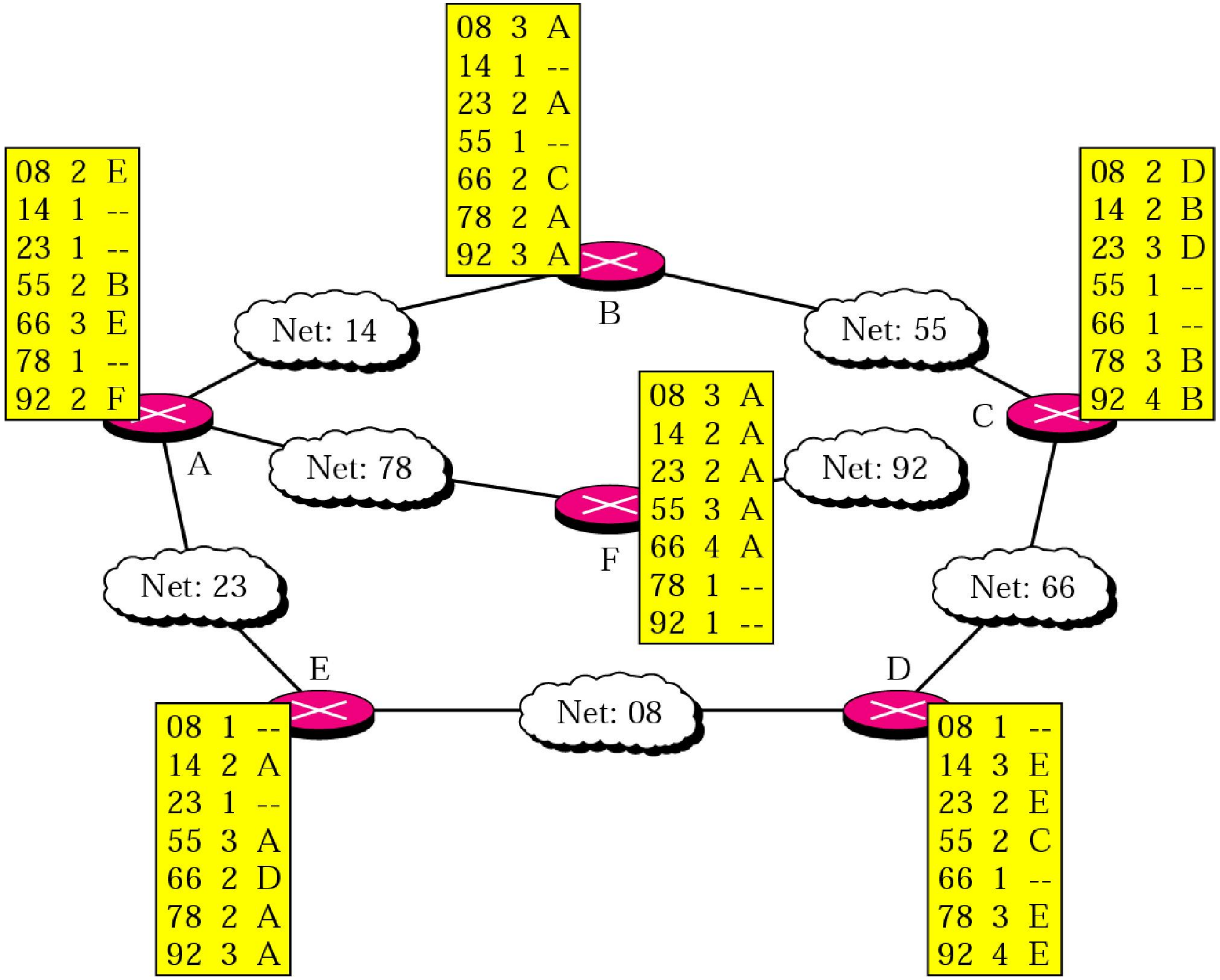
A's new table

14	1	—
23	1	—
55	2	B
78	1	—

❑ Updating routing table of node A upon receipt RIP message from Node B

Note : No need to send next hop while sending distance vectors

Final routing tables in the internetwork



08	3	A
14	1	--
23	2	A
55	1	--
66	2	C
78	2	A
92	3	A

08	2	E
14	1	--
23	1	--
55	2	B
66	3	E
78	1	--
92	2	F

08	2	D
14	2	B
23	3	D
55	1	--
66	1	--
78	3	B
92	4	B

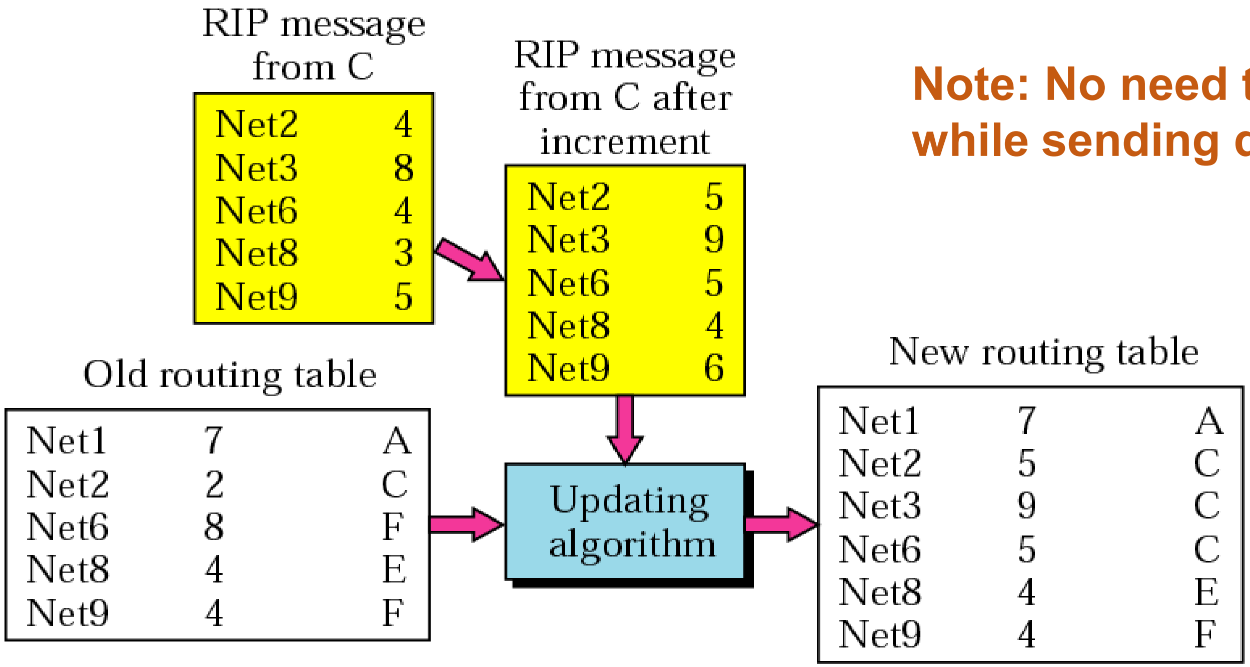
08	3	A
14	2	A
23	2	A
55	3	A
66	4	A
78	1	--
92	1	--

08	1	--
14	2	A
23	1	--
55	3	A
66	2	D
78	2	A
92	3	A

08	1	--
14	3	E
23	2	E
55	2	C
66	1	--
78	3	E
92	4	E

Each routing table is updated upon receipt of RIP messages (few iterations) using the RIP updating algorithm

Example of updating a routing table (detail)



RIP message from C

Net2	4
Net3	8
Net6	4
Net8	3
Net9	5

RIP message from C after increment

Net2	5
Net3	9
Net6	5
Net8	4
Net9	6

Old routing table

Net1	7	A
Net2	2	C
Net6	8	F
Net8	4	E
Net9	4	F

New routing table

Net1	7	A
Net2	5	C
Net3	9	C
Net6	5	C
Net8	4	E
Net9	4	F

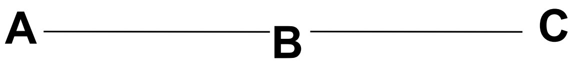
Note: No need to send next hop while sending distance vectors

Net1: No news, do not change
Net2: Same next hop, replace
Net3: A new router, add
Net6: Different next hop, new hop count smaller, replace
Net8: Different next hop, new hop count the same, do not change
Net9: Different next hop, new hop count larger, do not change

Problems with DVR

- ❑ Slow convergence
  - ➢ If a part of the network becomes inaccessible, it may take a long time for all other nodes to know this
- ❑ Too much overhead – updates sent periodically even if no change in routing table
- ❑ Routing loops may take a long time to be detected (count to infinity problem)

Routing loop or Count-to-Infinity problem



A

B

C

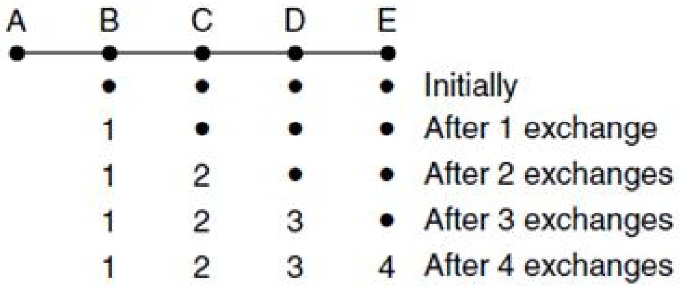
	A, A, 1	A,B,2
B detects A has crashed	A, A, inf	A,B,2

- If B’s routing table reaches C first, no problems
  - B sends <A, inf> to C, so C updates to <A, B, inf>

- But what if C’s routing table reaches B first?

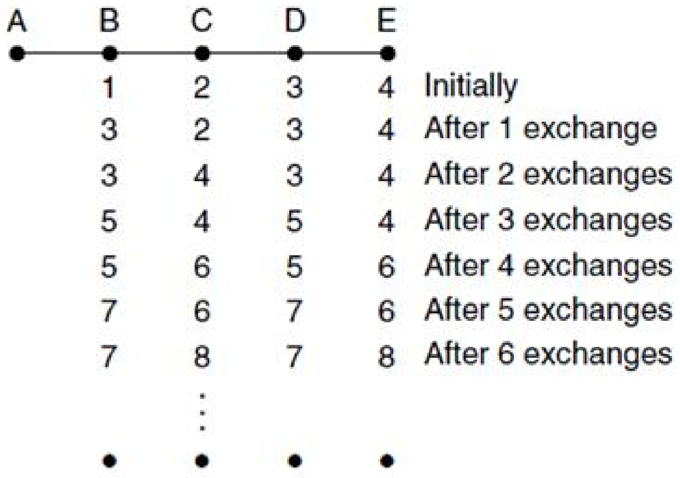
Routing loop or Count-to-Infinity problem

- ❑ Although it converges to the correct answer, it may do so slowly. In particular,
  - ➢ It reacts rapidly to good news, but
  - ➢ leisurely to bad news.



A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a) Response to good news – Node A is down initially and all the other routers know this



A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b) Response to bad news - Suddenly, either A goes down or the link between A and B is cut

Two-node instability

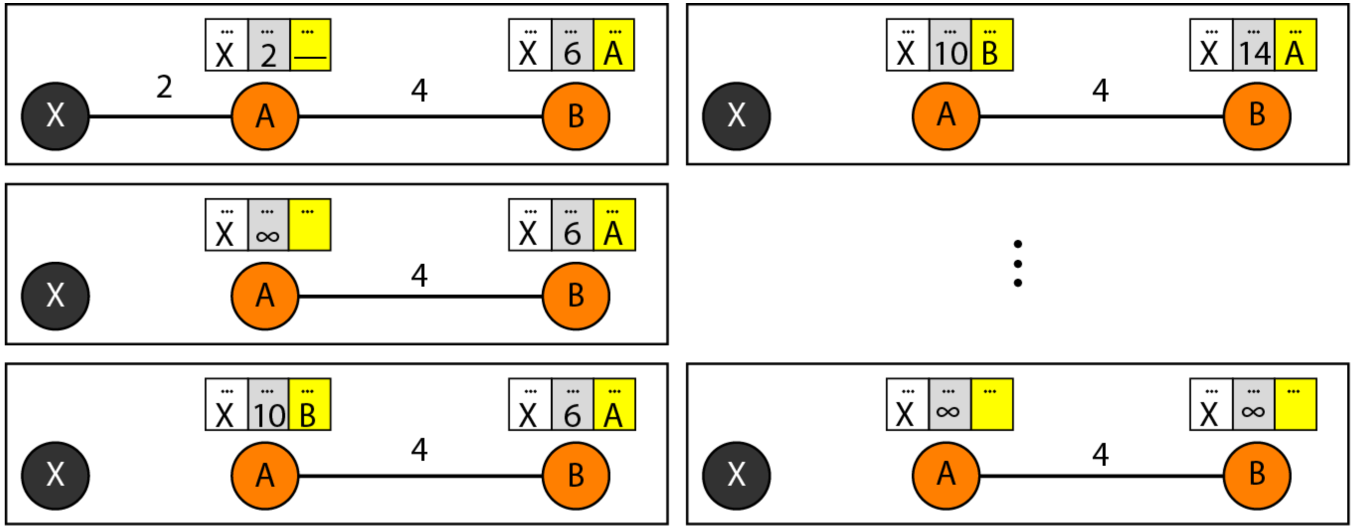
Before failure

After failure

After A receives update from B

After B receives update from A

Finally



To avoid count-to-infinity

- ❑ Split horizon technique
  - – Do not send a route to neighbor X, if the next hop in this route is X itself
  - – If network has a loop of length > 2, problem may occur even if split horizon technique used (*three node instability*)
- ❑ Triggered updates
  - – If a metric changes to INF, send it to all neighbors immediately
- ❑ Hold down
  - – if a metric has changed to INF, do not change it to a lower value for some time
- ❑ Split horizon with poissoned reverse
  - – Send all entries to neighbor X, but advertise the metric as INF if next hop in a route is X
- ❑ All these techniques decrease the probability of formation of routing loops, but they can still occur

Link State Routing Protocol

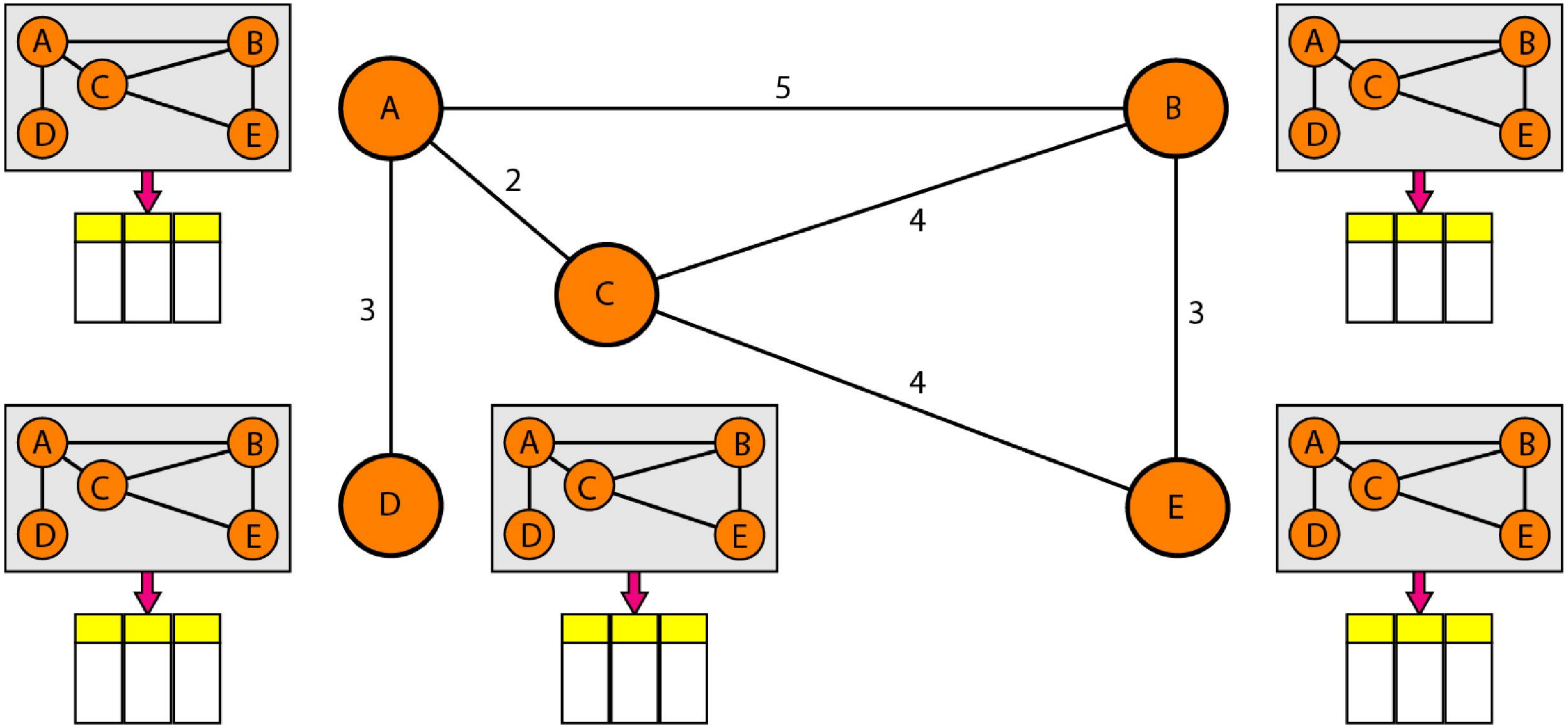
Open Shortest Path First (OSPF)

Link State Routing protocols

- Each node sends only information about its neighbors (who and cost) to **all nodes** in the network

- This information sent to all nodes by Flooding whenever a change is detected (e.g. a broken link, a crashed neighbor) among neighbors
  - Standard flooding optimization techniques used
  - Neighborhood information also sent periodically, along with if change is detected
  - This period can be relatively much larger compared to broadcast interval of DVR

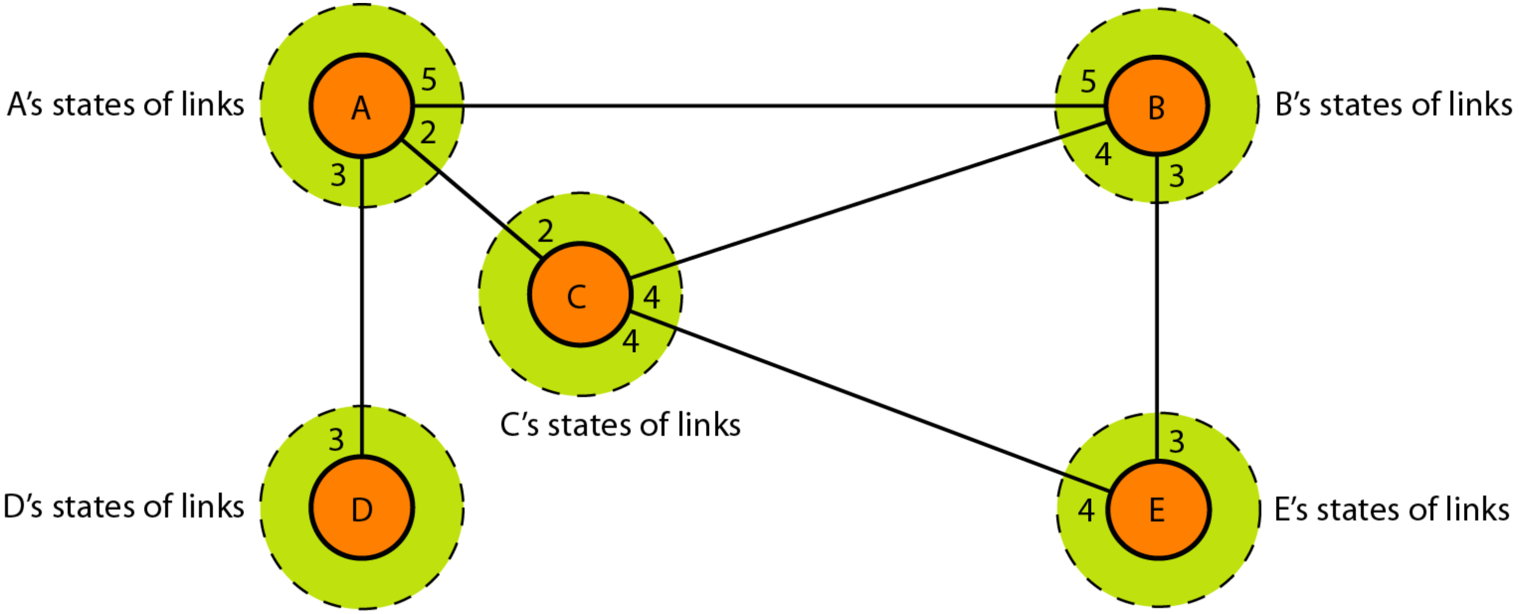
Concept of link state routing



❑ Each node needs to have a complete map of the network, which means it needs to know the state of the each link

❑ Collections of states for all links is called *Link-state Database* (LSDB)

Link state knowledge



Steps in Link State Routing

1. Discover your neighbors
2. Measure the delay to your neighbor
3. Build the Link State Packet (LSP)
4. Distribute the LSP to all nodes
5. Calculate the shortest path to all other nodes

Steps of LS Routing (contd.)

1. Discover your neighbors
   - As a router comes up, sends a ‘hello’ packet (contains own IP address) on all outgoing links
   - Gets the reply (from all neighbors) with “who it is” information e.g. IP address of neighbor

2. Measure the delay to your neighbor
   - Send out an ‘echo’ packet
   - neighbor sends it back immediately
   - (Round trip time / 2) gives the delay to the neighbor
   - Repeated several times to get the average delay

Steps of LS Routing: Build the LSP

3. The LSP sent by node N contains
   - ➢ The identification of the generating router N
   - ➢ Sequence number
   - ➢ Age / time to live (ttl)
   - ➢ List of identifications and distances to the neighbors

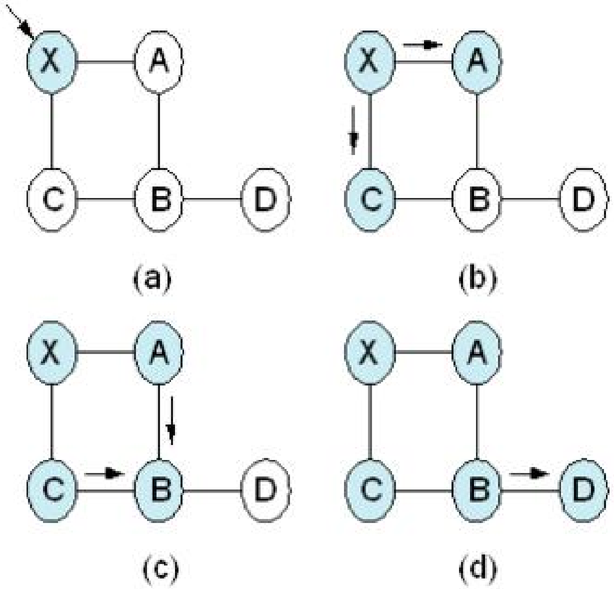
❑ Sequence number in LSP

- ➢ Ensures that no older version of the LSP (sent previously by N) is used by other nodes
- ➢ A node sends LSPs with continually increasing sequence numbers

Steps of LS Routing (contd.)

4. Distribute the LSP by flooding

- ❑ Flooding optimizations used
  - ➢ LSP not sent to the neighbor from whom it is obtained
  - ➢ A router M receives a LSP from router N



- ✓ Holds LSP for some time period T, to see if some other LSP from the same source N is coming
- ✓ After this time T, decides whether or not to flood the LSP
- ✓ If multiple copies of LSP (with same sequence no. and from same source) got within time T, flood one copy only
- ✓ If a LSP with higher sequence number received from same source within time T, discard older LSP, flood new LSP

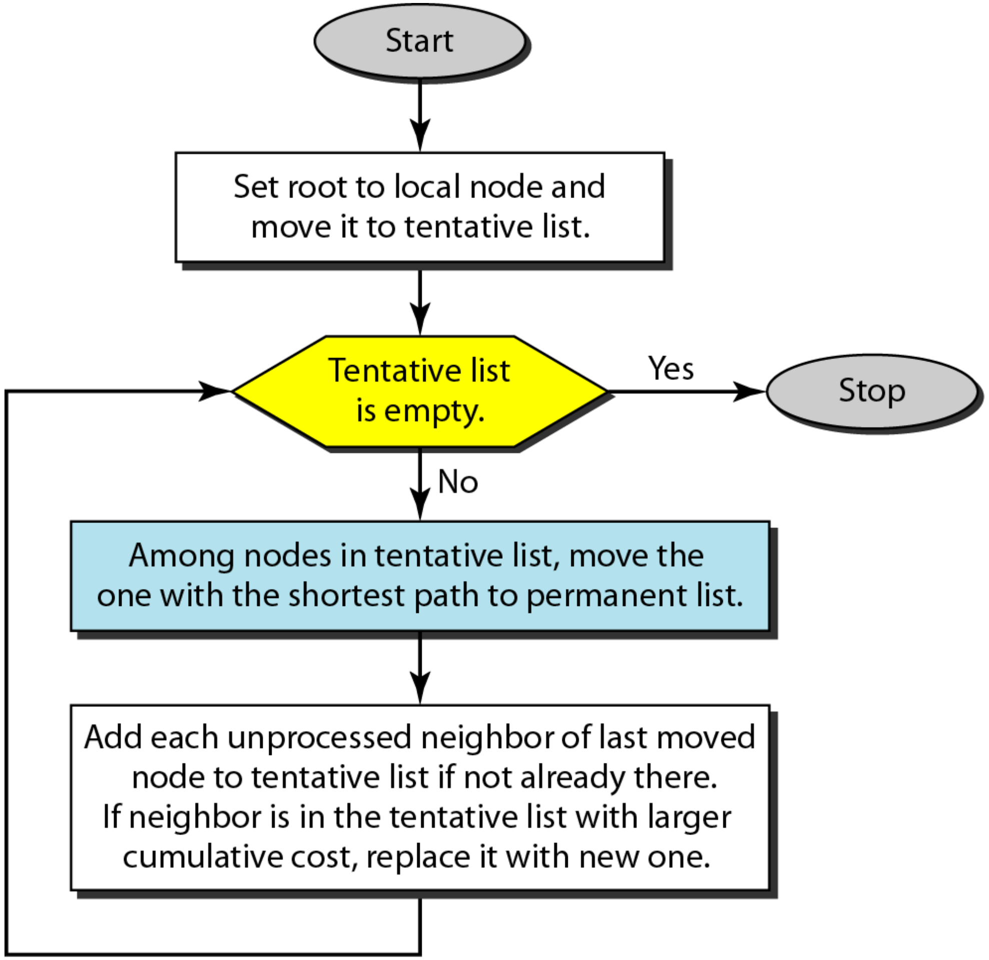
Steps of LS routing: Compute shortest path

5. Once a node has received LSPs from all (most) other nodes
    - it knows the entire network (or a large part of it)
    - Uses Dijkstra’s algorithm locally to compute shortest path to all nodes
    - Routing table (next hop and cost of path to reach each node) obtained from output of Dijkstra’s algo

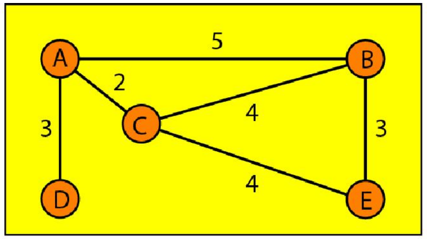
Dijkstra algorithm

- ❑ Calculates the shortest path between two points on a network, using a graph made up of nodes and edges.

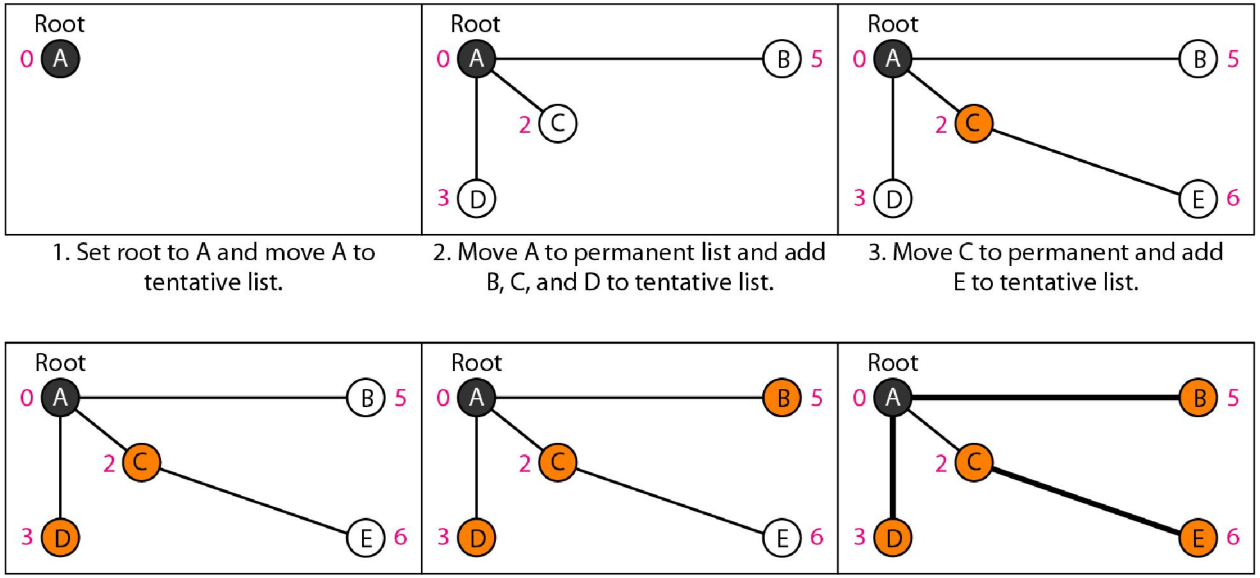
- ❑ Algorithm divides the nodes into two sets: Tentative and permanent. It chooses nodes, makes them tentative, examines them, and if they pass the criteria, makes them permanent.


Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there.
If neighbor is in the tentative list with larger cumulative cost, replace it with new one.

Example of formation of shortest path tree



Topology



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

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

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

4. Move D to permanent list.

5. Move B to permanent list.

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

Routing table for node A

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

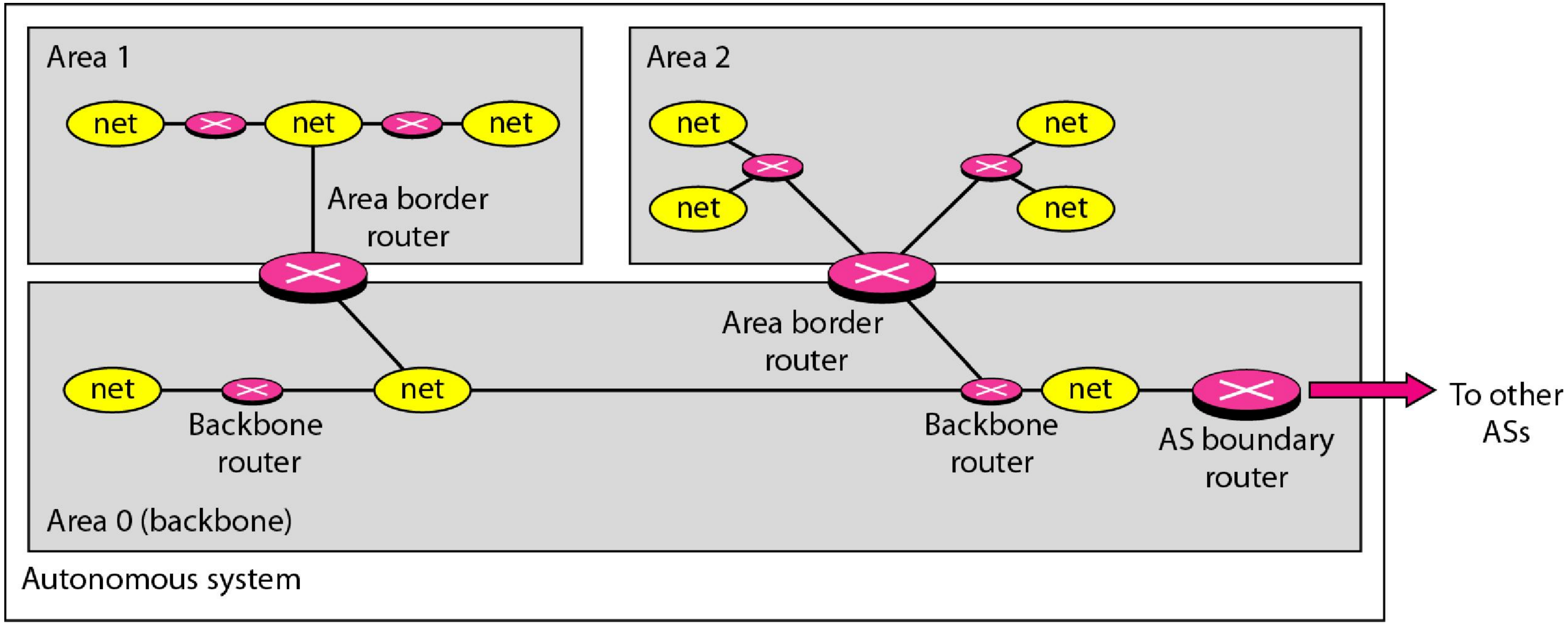
A practical algorithm based on LS

❑ OSPF (Open Shortest Path First)

- Period of flooding: 30 minutes
- Metric used can be flexibly defined to indicate link cost, etc
- OSPF keeps multiple (if more than one exist) least-cost routes to reach same destination
  - ✓ This allows some load balancing

Areas in an autonomous system

- ❑ OSPF may not create problem in small AS, but may create problem in larger network – all router flood the whole AS with their LSP
- ❑ Therefore, OSPF divides an autonomous system into areas.
- ❑ Special routers called autonomous system boundary routers, are responsible for dissipating information about other autonomous system into the current system



Path Vector Routing Protocol

Border Gateway Protocol (BGP)

For Further Study

References

- *Data Communications & Networking, 5th Edition, Behrouz A. Forouzan*
- *Computer Networks, Andrew S. Tanenbaum and David J. Wetherall*
- *Data and Computer Communication, William Stallings*