Chapter 4

Network Layer

Network Layer

- Network Layer is present between Transport layer and Data Link layer
- The network layer has the following functionalities:
    1. Forwarding
    2. Routing

Forwarding:

- When packet arrives at the router, router must move the packet to the appropriate output link. This is called as forwarding

Routing:

- The network layer must determine the route or path taken by packets as they flow from a sender to a receiver.
- The algorithms that calculate these paths are referred to as routing algorithms.
- With the help of routing algorithms, Forwarding tables are been constructed at every router

Network Layer (contd..)

Forwarding vs Routing

Forwarding	Routing
Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface	Routing refers to the network-wide process that determines the end-to-end paths that packets take from source to destination
Forwarding takes place at very short timescales (typically a few nanoseconds)	Routing takes place on much longer timescales (typically seconds),
Forwarding is typically implemented in hardware	Routing is often implemented in software.

Network Layer (contd..)

The network layer is divided into 2 planes:

1. Data plane
2. Control plane

1. Data plane

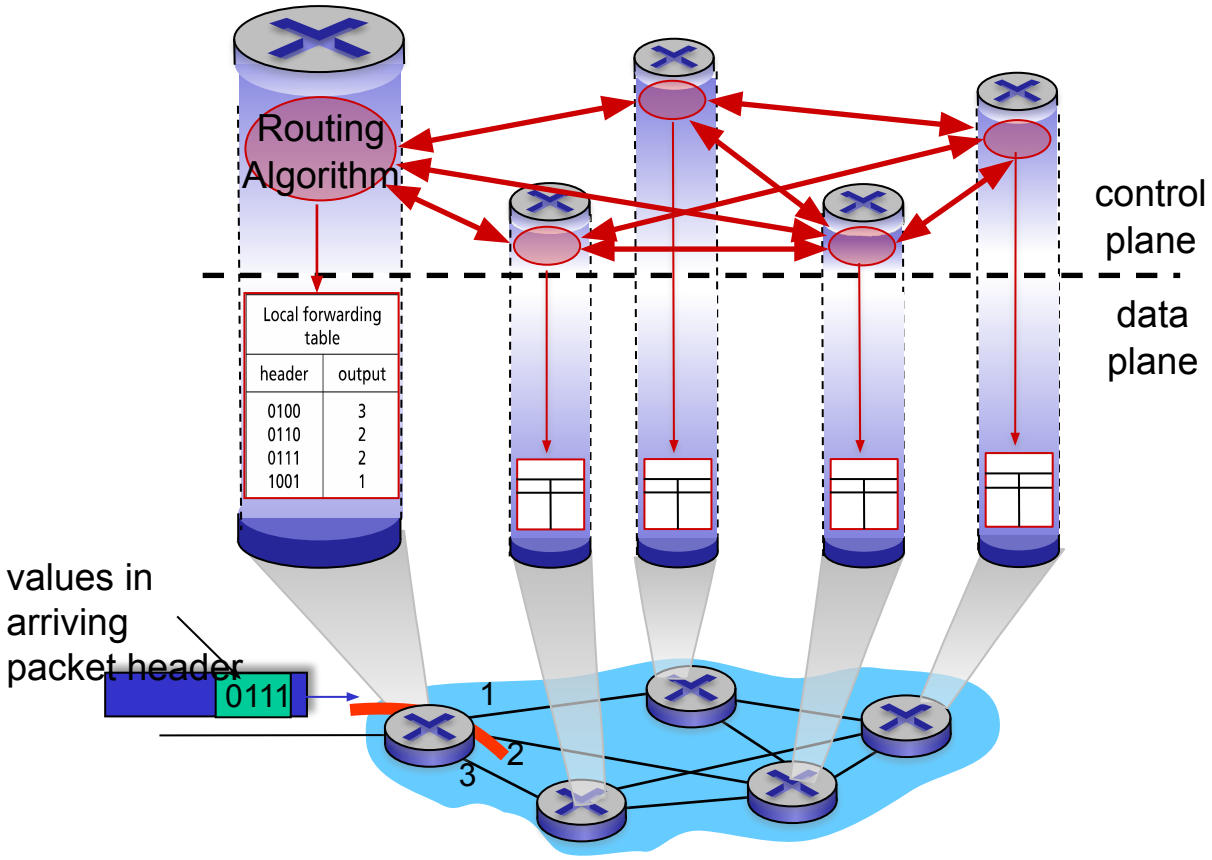
- Data plane is the place where the actual routers are present and forwarding is happening

2. Control plane

- Control plane is the place where the routing algorithms are being executed for constructing the forwarding table

Network Layer (contd..)

Control plane: The Traditional approach



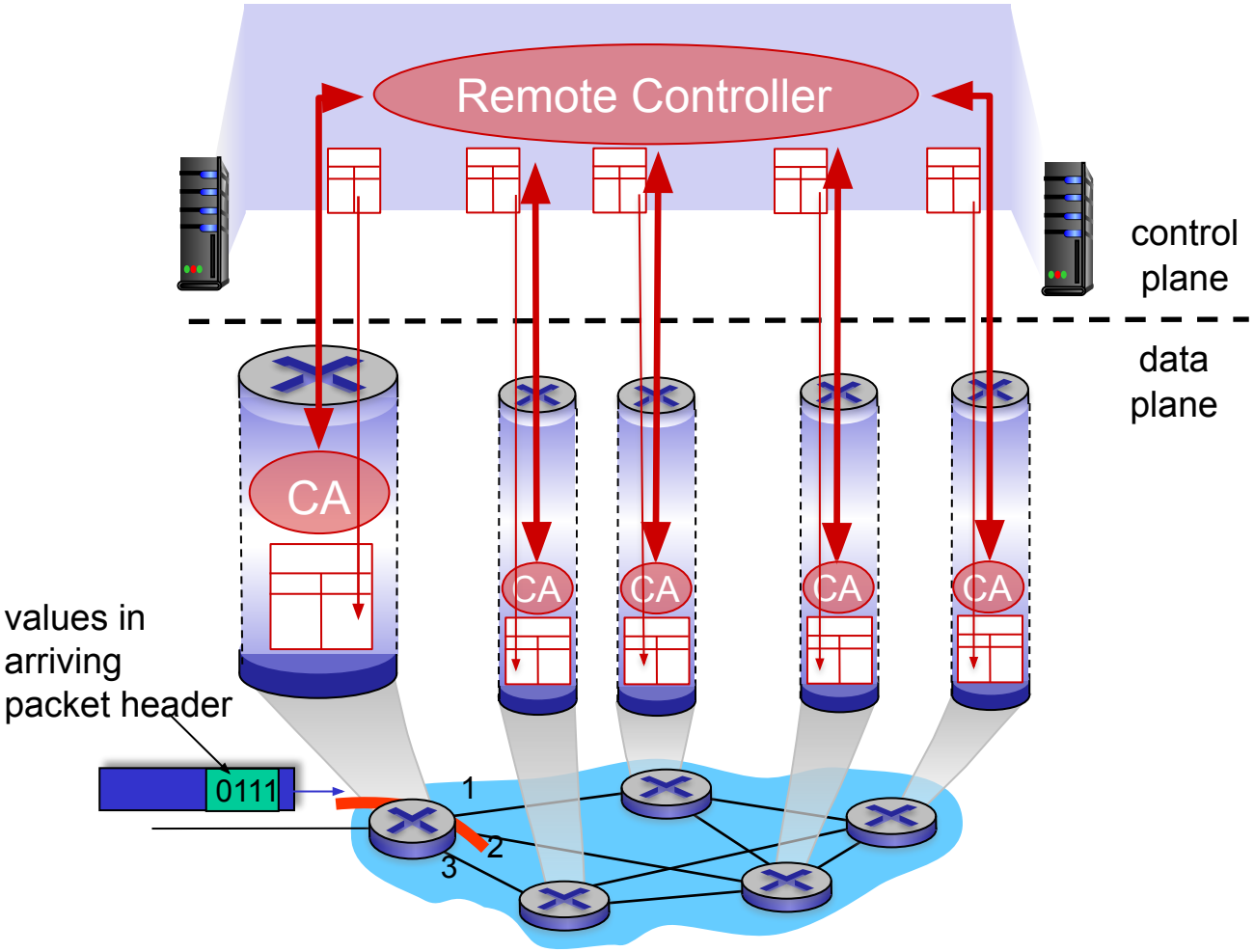
Network Layer (contd..)

Control plane: The Traditional approach(contd..)

- In traditional approach, the routing algorithm running in every router determines the contents of the routers' forwarding tables
- The routing algorithm function in one router communicates with the routing algorithm function in other routers to compute the values for its forwarding table

Network Layer (contd..)

Control plane: SDN approach



Network Layer (contd..)

Control plane: SDN approach (contd..)

- In SDN approach a physically separate, remote controller computes and distributes the forwarding tables to be used by each and every router.
- The router performs forwarding only, while the remote controller computes and distributes forwarding tables

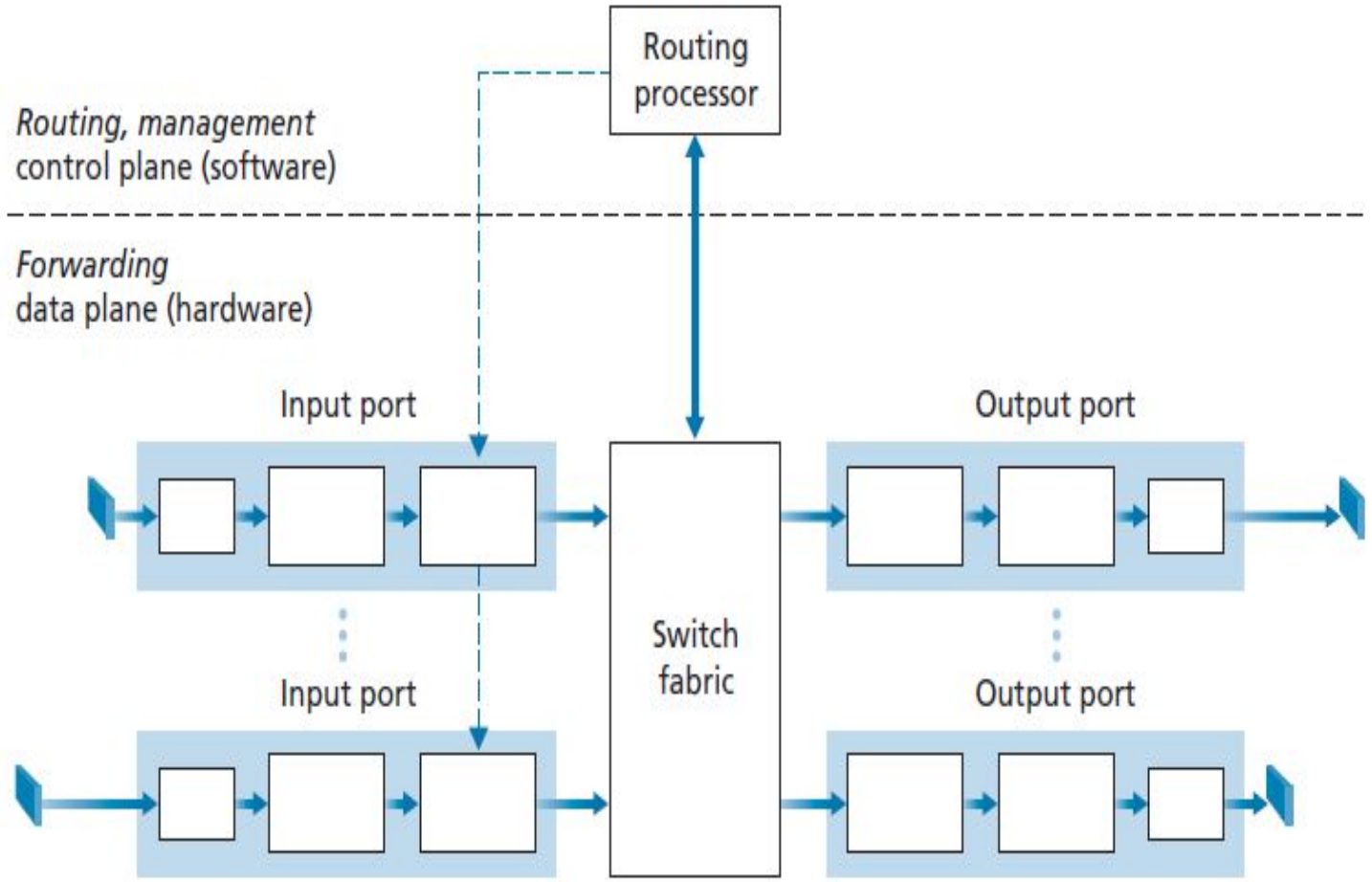
Network Layer Services

Network layer make its best-effort to provide the following services,

1. Guaranteed delivery of data to destination
2. Guaranteed delivery with bounded delay
3. In-order packet delivery
4. Guaranteed minimal bandwidth
5. Security

Inside a Router

- The below diagram shows the architecture of router



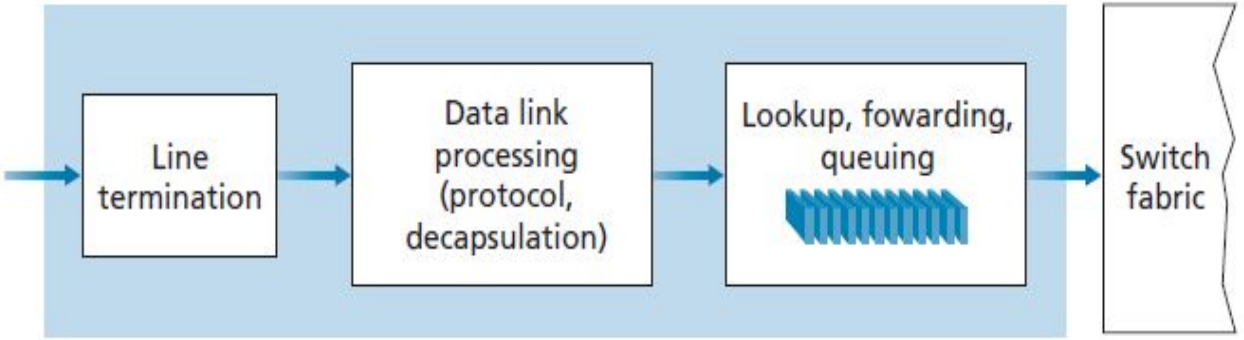
Inside a Router (contd..)

- Inside a router there are 4 components:
    1. Input port
    2. Switching Fabric
    3. Output port
    4. Routing processor

Inside a Router (contd..)

1. Input Port

- The below diagram shows the processing that takes place at input port



- Input port is responsible for several functions:

  1. It is the place where the incoming physical link gets terminated(leftmost box)
  2. It performs link-layer functions (middle box)
  3. It performs lookup function where the forwarding table is consulted to determine the output port for newly arrived data (rightmost box). And also this is the place where queuing occurs

Inside a Router (contd..)

Forwarding Table

- The forwarding table at the router is shown in the below diagram

Destination Address Prefix	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

- Assume the router is having 4 links. Whenever data arrives at router, the router checks the destination IP address in the data. It then consults forwarding table in order to find out through which link interface the data has to be forwarded
- In the destination IP address only prefix bits are used for forwarding the data

Inside a Router (contd..)

Forwarding table

- For example, suppose the packet's destination address is 11001000 00010111 00010110 10100001; because the 21-bit prefix of this address matches the first entry in the table, the router forwards the packet to link interface 0
- If a prefix doesn't match any of the first three entries, then the router forwards the packet to the default interface 3

Inside a Router (contd..)

Forwarding table

Longest Prefix Matching

- For example, suppose the packet's destination address is

```
11001000  00010111  00011000  10101010
```

- If we take the first 21 bits of the address, it matches with the third entry in the table. If we take the first 24 bits of the address, then it matches with the second entry in the table.
- When there are multiple entries, the router uses the longest prefix matching rule, it finds the longest matching entry in the table and forwards the packet to the link interface associated with the longest prefix match
- In the above example, the data is forwarded via the interface link 1

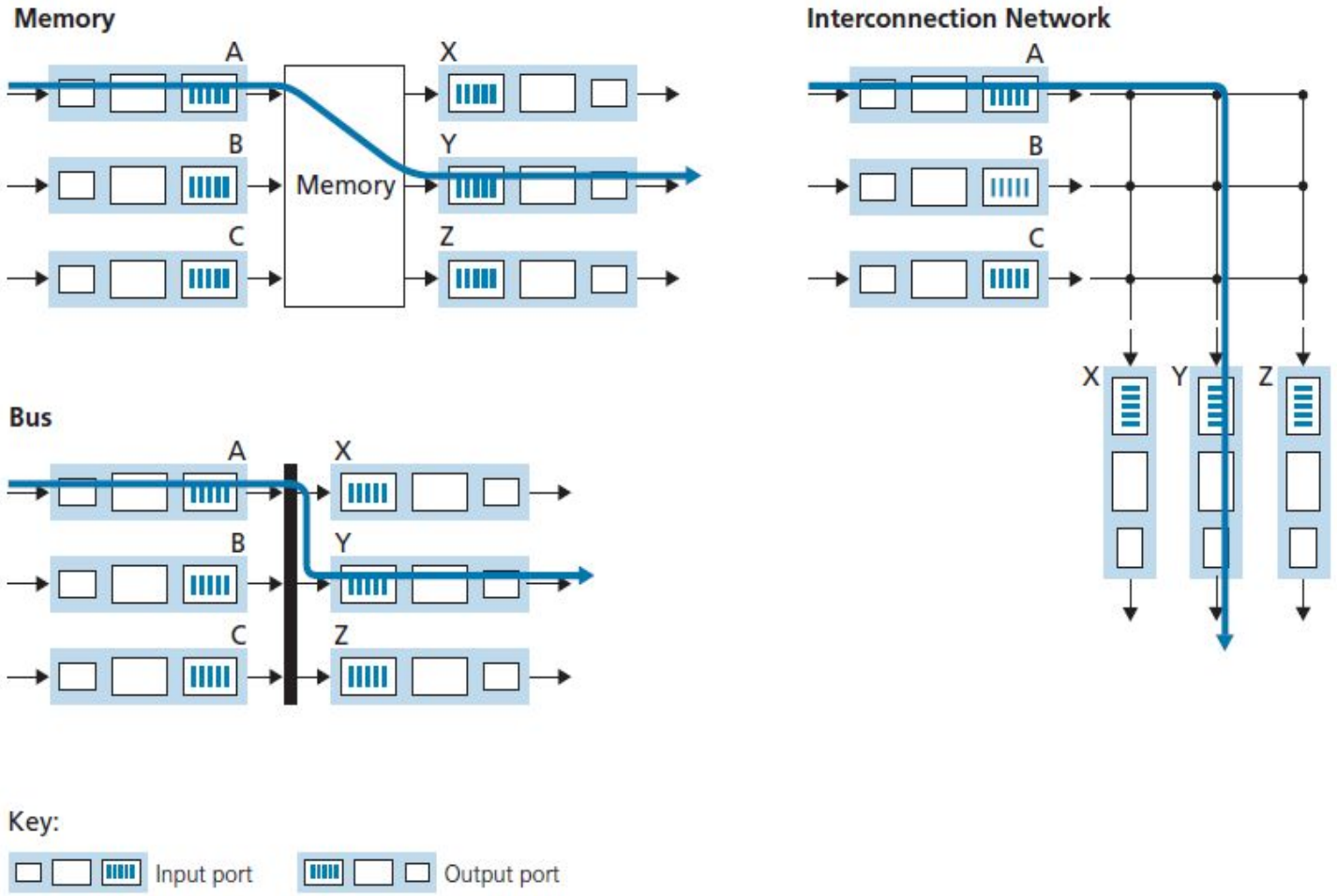
Inside a Router (contd..)

2. Switching Fabric

- The switching fabric is at the very heart of a router
- It is through this fabric that the packets are actually switched (forwarded) from an input port to an output port
- Switching can be accomplished in a number of ways:
    1. Switching via memory
    2. Switching via a bus
    3. Switching via an interconnection network

Inside a Router (contd..)

2. Switching Fabric (contd..)

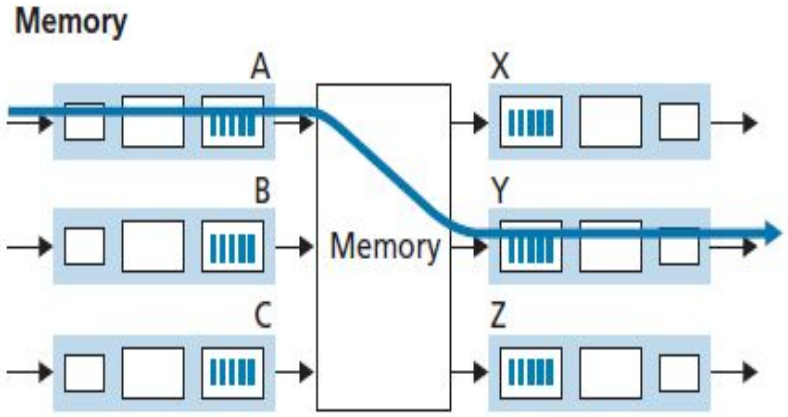


Inside a Router (contd..)

2. Switching Fabric (contd..)

(i). Switching via memory

- In earlier routers, switching between input and output ports are being done under direct control of the CPU (routing processor).
- The packet from the input port are copied into processor memory. The routing processor then extracted the destination address from the header, looked up the appropriate output port in the forwarding table, and copied the packet to the output port's buffers.
- The drawback of this method is that two packets cannot be forwarded at the same time even though they are forwarded through different output ports

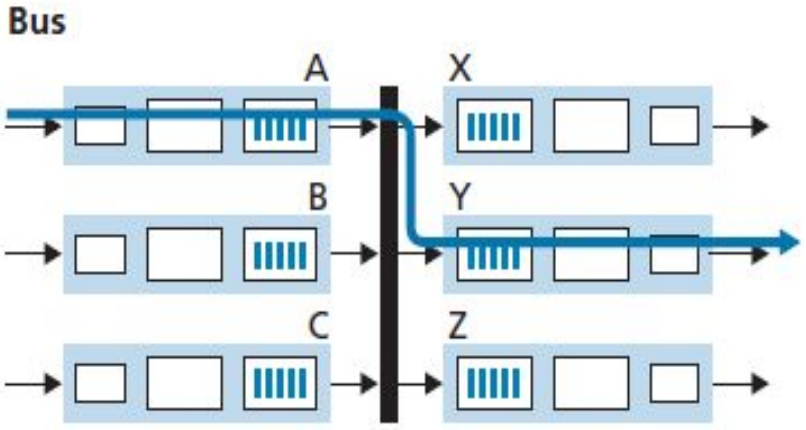


Inside a Router (contd..)

2. Switching Fabric (contd..)

(ii). Switching via bus

- In this approach, an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
- only one packet can cross the bus at a time
- In this method also the disadvantage is that two packets cannot be forwarded at the same time even though they are forwarded through different output ports

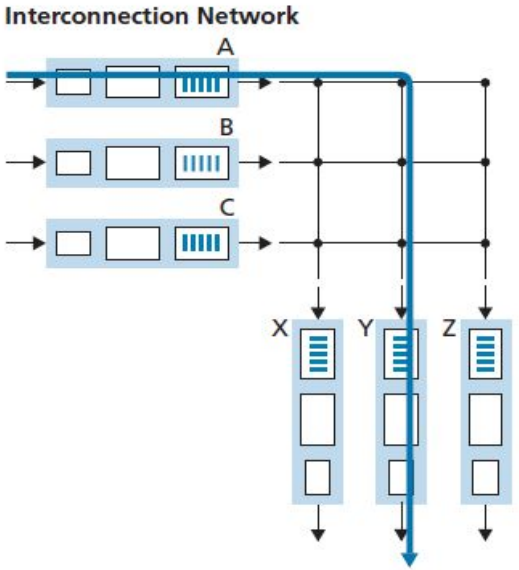


Inside a Router (contd..)

2. Switching Fabric (contd..)

(iii). Switching via an interconnection network

- In this approach, a crossbar switch is used
- Crossbar switch is an interconnection network consisting of 2N buses that all the 'N' input ports is connect to all the 'N' output ports
- Each vertical bus intersects each horizontal bus at a cross point



- Note that a packet from port B can be forwarded to port X at the same time when a packet from port A is forwarded to port Y

Inside a Router (contd..)

2. Switching Fabric (contd..)

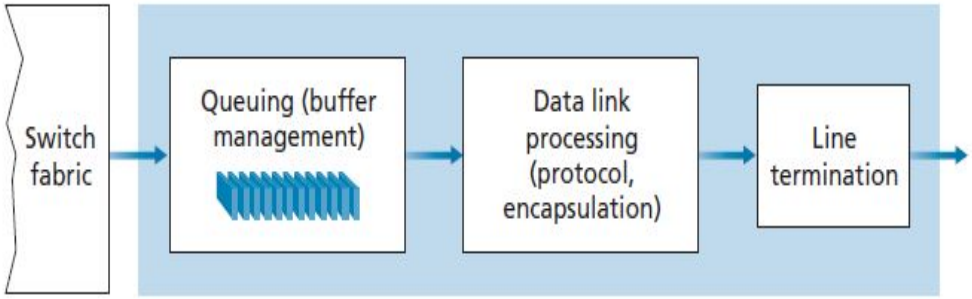
(iii). Switching via an interconnection network

- Advantage of switching via an interconnection network is that it is capable of forwarding multiple packets in parallel

Inside a Router (contd..)

3. Output port

- An output port stores packets received from the switching fabric and transmits these packets on the outgoing link



- Above diagram shows the processing that takes place at output port
- There will be queue to hold the data (leftmost box)
- It performs link-layer functions (middle box)
- There is also physical line termination(rightmost box)

Inside a Router (contd..)

4. Routing Processor

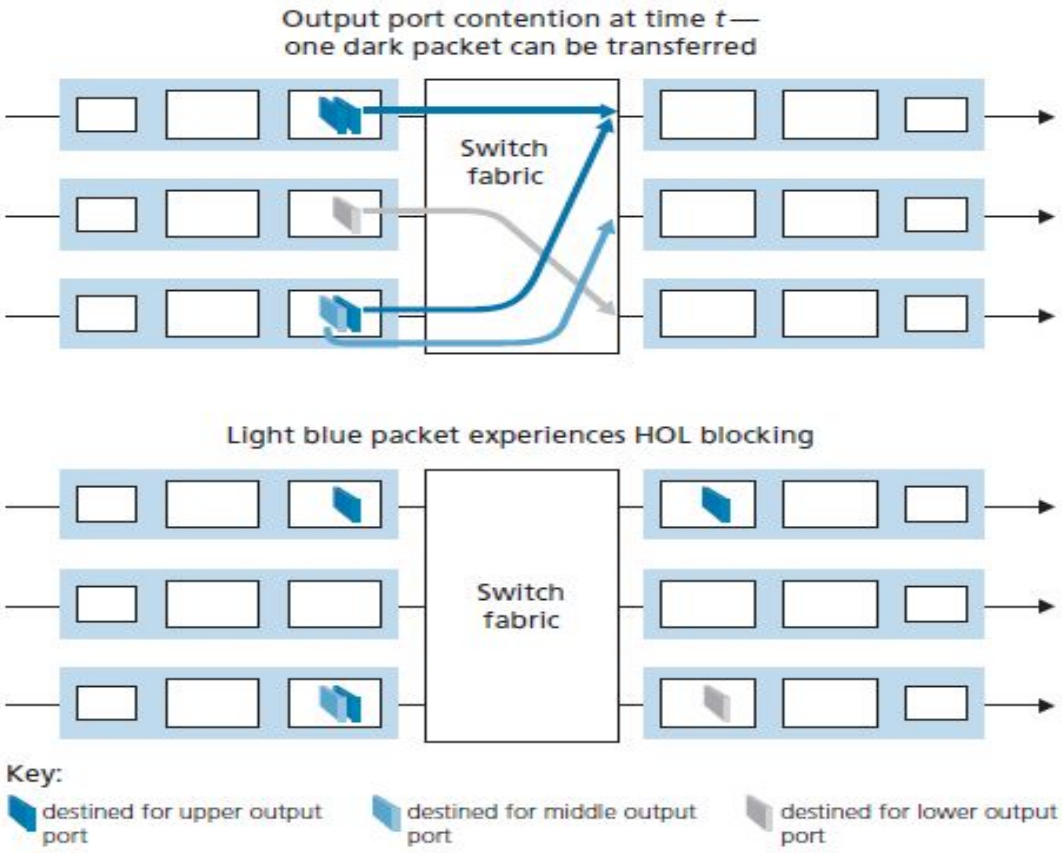
- In traditional routers, routing processor executes the routing protocols
- In SDN routers, the routing processor is responsible for communicating with the remote controller in order to receive forwarding table entries computed by the remote controller, and install these entries in the router's input ports.

Inside a Router (contd..)

Input Queuing

- Queuing can occur at Router Input port

Head-of-the-Line(HOL) Blocking problem



Inside a Router (contd..)

Input Queuing

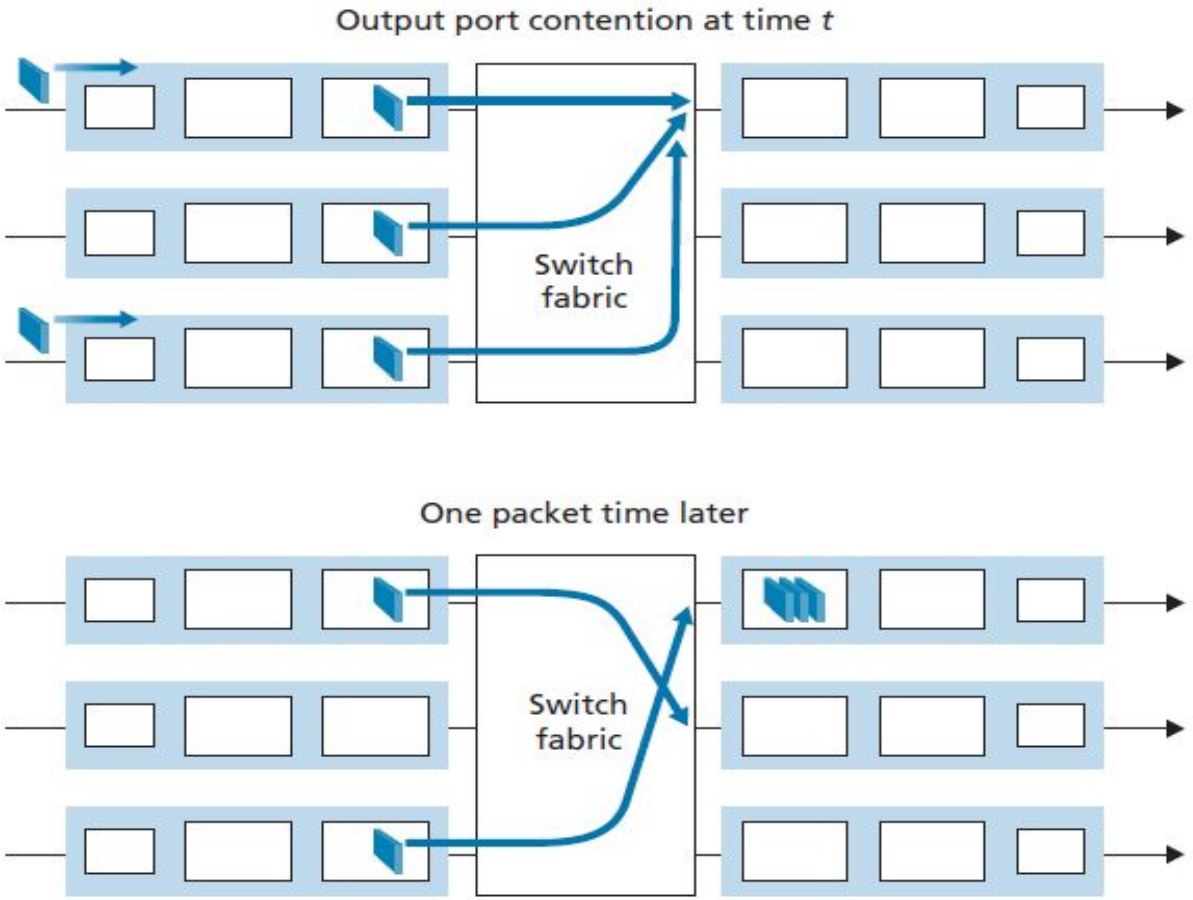
Head-of-the-Line(HOL) Blocking problem

- In the diagram shown in the previous slide, upper-left and lower-left input port have dark shaded packets destined for the same upper-right output port
- Suppose that the switch fabric chooses to transfer the packet from the front of the upper-left queue. In this case, the darkly shaded packet in the lower-left queue must wait. But not only must this darkly shaded packet wait, and also the lightly shaded packet that is queued behind that packet in the lower-left queue, even though there is no contention for the middle-right output port (the destination for the lightly shaded packet).
- This phenomenon is known as head-of-the-line (HOL) blocking
- The packet in the head of the queue is blocking the next packet, even though the next packet is destined for different output port which is free

Inside a Router (contd..)

Output Queuing

- Queuing can occur at Router output port also



Inside a Router (contd..)

Output Queuing

- In the diagram shown in the previous slide, 3 packets from different input port is destined for the same output port at time t
- Since the output port can transmit only a single packet in a unit of time, the 3 arriving packets will have to queue for transmission over the outgoing link

Inside a Router (contd..)

Packet Scheduling

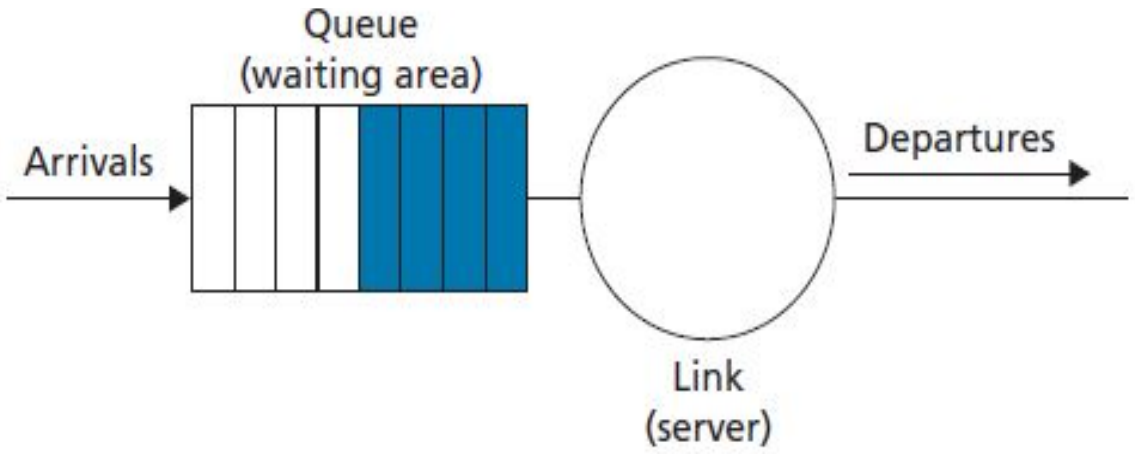
- When packets are queued at the router, the following scheduling mechanisms are used for determining the order in which queued packets are transmitted over an outgoing link

    1. First-in-First-Out (FIFO)
    2. Priority Queuing
    3. Round Robin
    4. Weighted Fair Queuing (WFQ)

Inside a Router (contd..)

1. First-in-First-Out (FIFO)

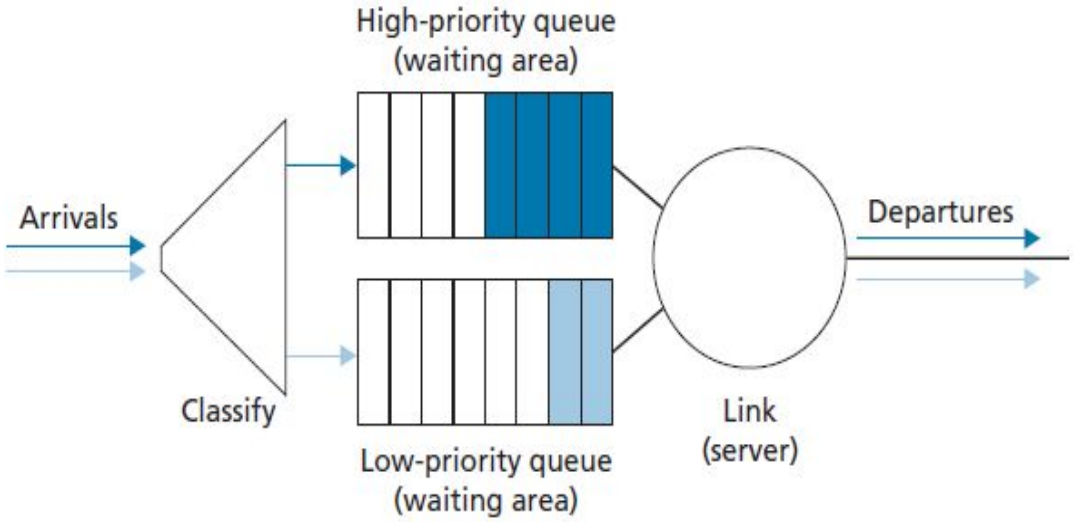
- Packets arriving at the link output queue wait for transmission if the link is currently busy transmitting another packet
- The FIFO scheduling discipline selects packets for link transmission in the same order in which they arrived at the output link queue.



Inside a Router (contd..)

2. Priority Queuing

- In priority queuing, packets arriving at the output link are classified into priority classes
- Each priority class typically has its own queue
- When choosing a packet to transmit, the priority queuing will transmit a packet from the highest priority class that has a nonempty queue (The choice among packets in the same priority class is typically done in a FIFO manner)
- After completing high priority queue control will to low priority queue



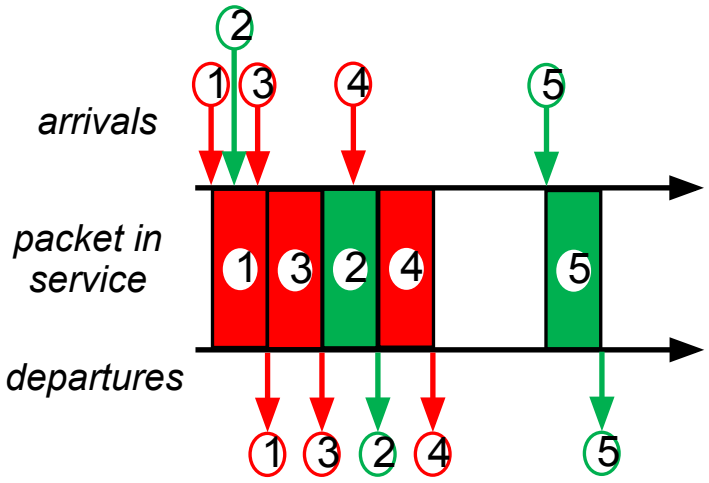
Inside a Router (contd..)

Priority Queuing scheduling Example

- In the given diagram,

  red color packets represent high priority

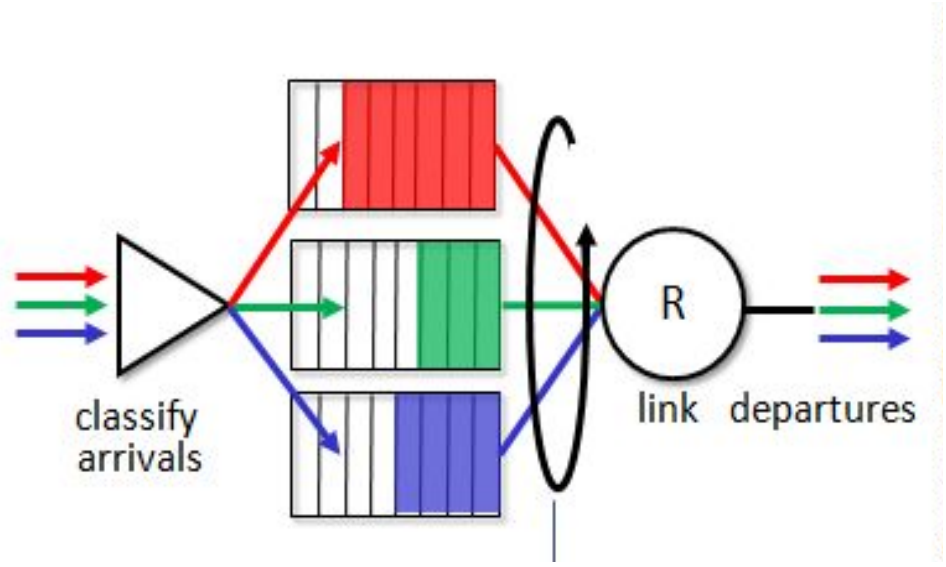
  Green color packets represent low priority



Inside a Router (contd..)

3. Round Robin scheduling

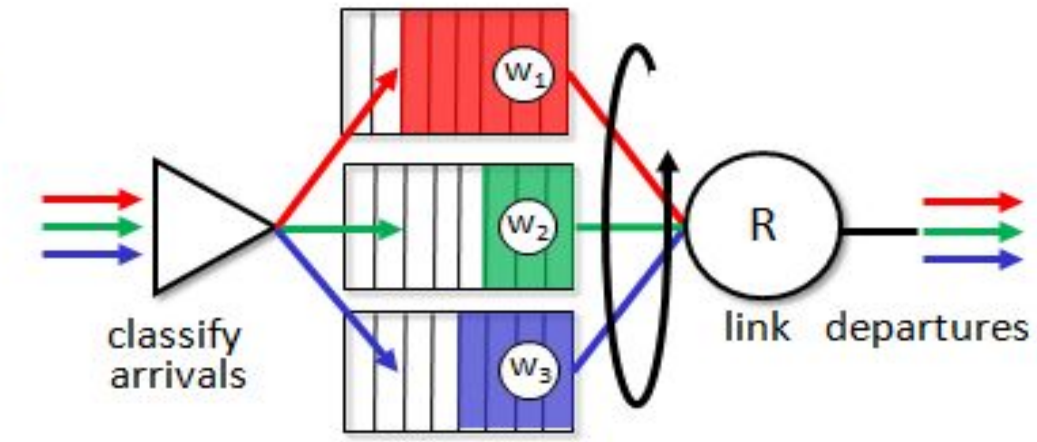
- Under the round robin queuing discipline, packets are sorted into classes as with priority queuing. However, rather than there being a strict service priority among classes, a round robin scheduler alternates service among the classes.
- In the simplest form of round robin scheduling, a class 1 packet is transmitted, followed by a class 2 packet, followed by a class 1 packet, followed by a class 2 packet, and so on



Inside a Router (contd..)

4. Weighted Fair Queuing (WFQ) scheduling

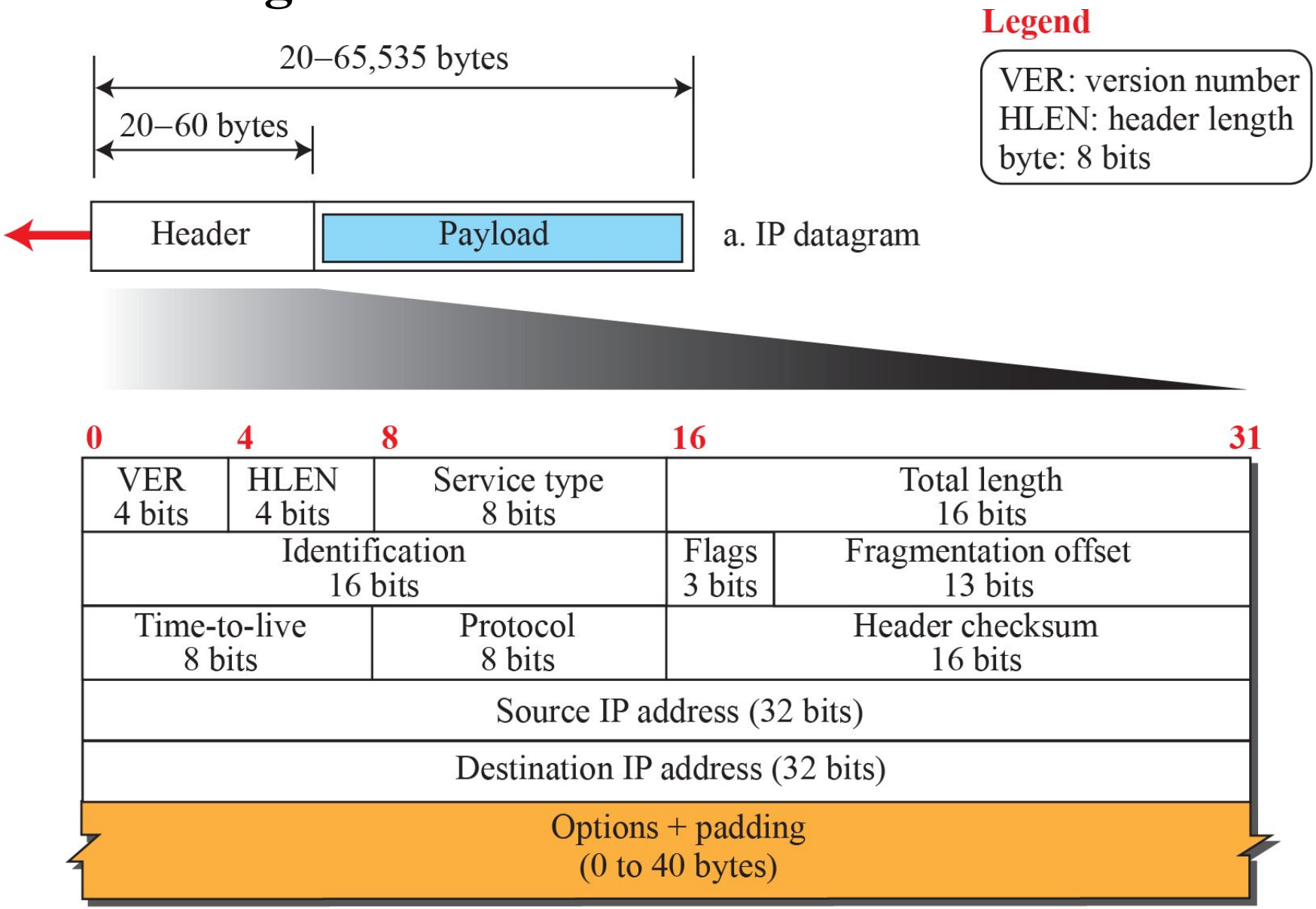
- As in round robin scheduling, a WFQ scheduler will serve classes in a circular manner
- In WFQ scheduling each class, i, is assigned a weight w_i



Internet Protocol (IP)

- Internet Protocol is the most important network layer protocol that is used in internet
- This is the protocol which is used for adding IP address to the data
- Internet Protocol comes in 2 versions:
    1. IPv4
    2. IPv6

IPv4 Datagram Format



IPv4 Datagram Format (contd..)

- The diagram in the previous slide shows the datagram format of IPv4 protocol
- Minimum size of IPv4 header is 20 bytes and the maximum size is 60 bytes

Various fields in the IPv4 are as follows:

1. Version:

- First 4 bits in the IPv4 header represents version of IP. Since it is IPv4 the value in this field should be 4 only(0100)

2. Header Length

- This field contains the length of the header. The value in this field is multiplied by 4 in order to get the actual header length

3. Service Type

- The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other. For example to distinguish a non-real time data (email) from real-time data (video conferencing)

IPv4 Datagram Format (contd..)

4. Total Length

- This field is the total length of the IP datagram (header plus data), measured in bytes
- If we subtract the value in this field with the header length field then we can get the length of the data carried by the IP datagram

The following are the fields used for data Fragmentation and Reassembly,

1. Identification
2. Flag
3. Fragmentation Offset

Why Data is fragmented?

- When the size of the data is larger than the outgoing link capacity at that time data is fragmented
- Data is fragmented at source and intermediate routers. The fragmented data is reassembled only at receiver side

IPv4 Datagram Format (contd..)

5. Identification:

- The data once it is fragmented, all the fragments that belongs to the same data is going to get an identification value
- Identification field has a value that is same for all fragments that belongs to a data
- With the help of this identification field only, receiver is going to identify which are all the fragments that belong to the same data

6. Flags:

- The 3-bit flags field defines three flags. The 1st bit(leftmost bit) is reserved (not used).
- The second bit (D bit) is called the Don't fragment bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary.

IPv4 Datagram Format (contd..)

- The third bit (M bit) is called the More fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

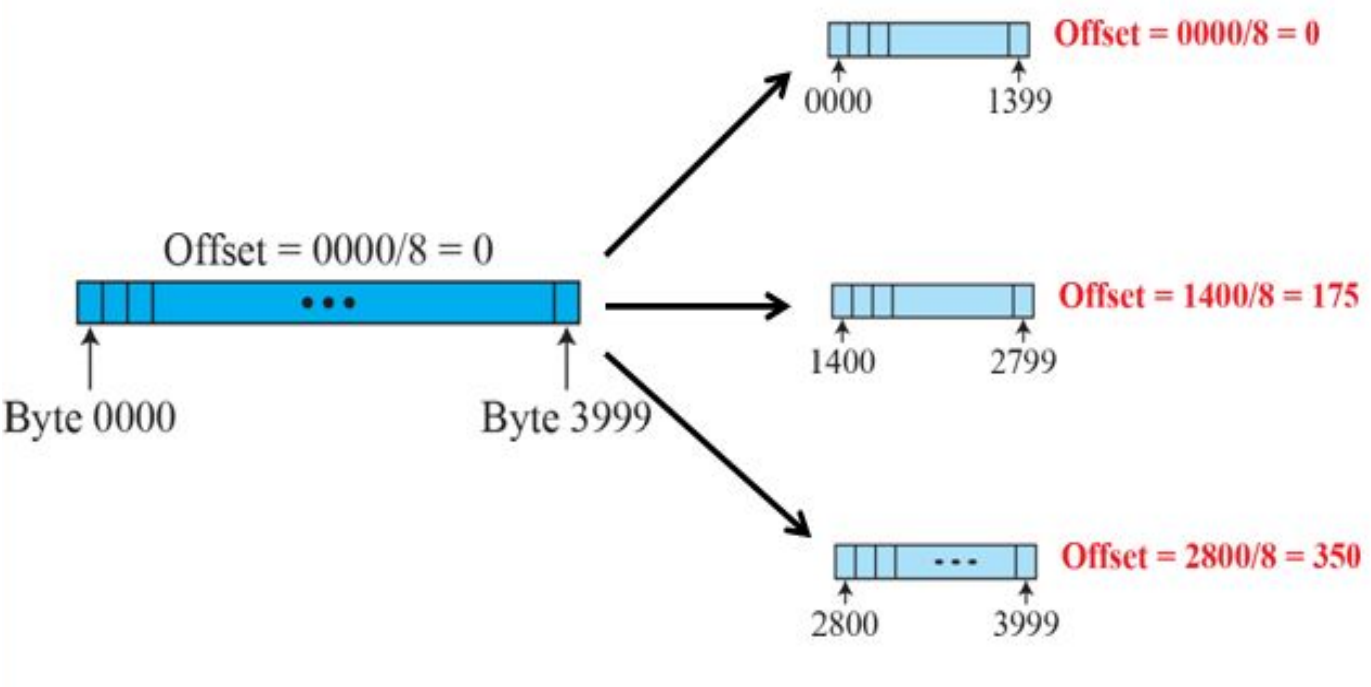
7. Fragmentation Offset:

- Every fragment gets a unique offset number. Receiver uses this offset field for reassembling the data
- Offset field is measured in units of 8 bytes
- Number of bytes in each fragment should always be multiple of 8 except the last fragment
- Offset is calculated for every fragment as first byte number/8

IPv4 Datagram Format (contd..)

Fragment Offset calculation Example

- Assume a data of 4000 bytes is divided into three fragments. The below diagram shows how the offset value is calculated for each fragments



IPv4 Datagram Format (contd..)

8. Time-to-live(TTL):

- The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever in the network.
- This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, a router must drop that datagram.

9. Protocol:

- This field specifies what is the Upper-layer protocol
- This field is typically used only when an IP datagram reaches its final destination
- The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed
- 6 indicates that the data portion is passed to TCP, while a value of 17 indicates that the data is passed to UDP

IPv4 Datagram Format (contd..)

10. Header Checksum:

- The header checksum is used in detecting bit errors in a received IP datagram
- In IP datagram, checksum is calculated for header fields alone and not for data

11. Source IP address:

- This is the field which is going to contain the IPv4 address of the source system

12. Destination IP address:

- This is the field which is going to contain the IPv4 address of the destination system
- Destination system IP address is found by the source system with the help of DNS protocol

13. Options:

- The options fields allow an IP header to be extended. This field is used to specify any extra information by the source

IPv4 Datagram Format Problems

Example 1:

- An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$ The receiver discards the packet. Why?

Solution

- There is an error in this packet. The 4 leftmost bits $(0100)_2$ show the version, which is correct.
- The next 4 bits $(0010)_2$ show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission

Example 2:

- In an IPv4 packet, the value of HLEN is $(1000)_2$. How many bytes of options are being carried by his packet?

Solution

- The HLEN value is 8, which means the total number of bytes in the header is 8×4, or 32 bytes.
- The first 20 bytes are the base header, the next 12 bytes are the options.

IPv4 Datagram Format Problems (contd..)

Example 3:

- In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is $(0028)_{16}$. How many bytes of data are being carried by this packet?

Solution

- The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options).
- The total length is $(0028)_{16}$ or 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

IPv4 Datagram Format Problems (contd..)

Example 4:

- A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

- If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non fragmented packet is considered the last fragment.

Example 5:

- A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution:

- If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

IPv4 Datagram Format Problems (contd..)

Example 6:

- A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

- Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0,it is the first fragment.

Example 7:

- A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

- To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

IPv4 Datagram Format Problems (contd..)

Example 8:

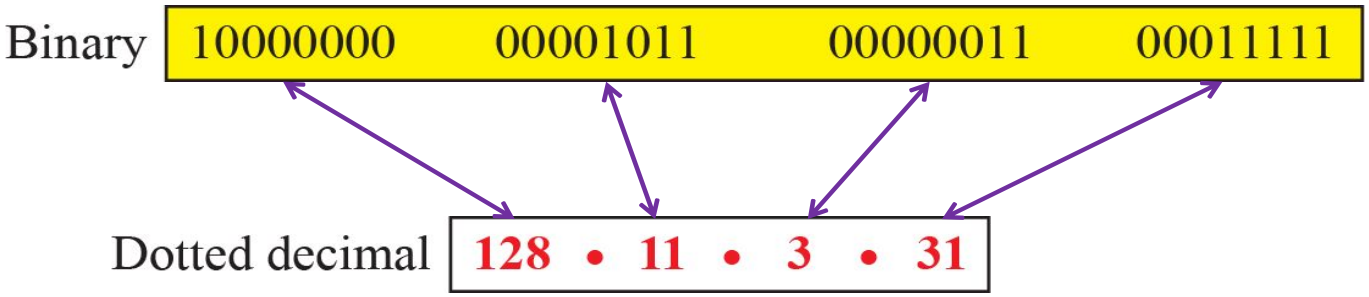
- A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

- The first byte number is 100 × 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

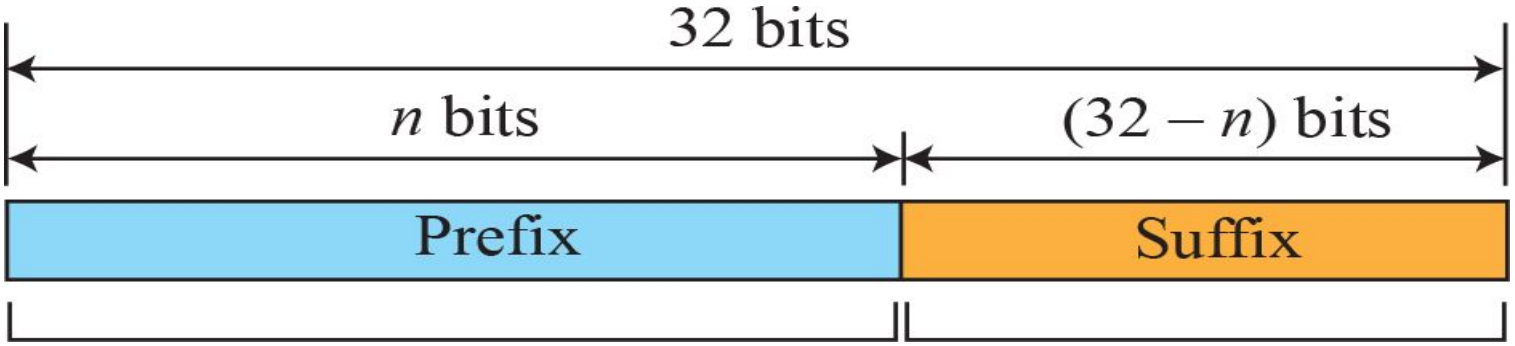
IPv4 Addressing

- IPv4 address is the identifier used in the Network layer to identify each device in the Internet
- An IPv4 address is a 32-bit address that uniquely defines the host or a router in the Internet
- Every IPv4 address is of 32 bits which is written in dotted decimal format. There are 4 sections in IPv4 address and every section is separated by dot(.). Each section is of 1 byte
- Given below is an IPv4 address in dotted decimal and in binary format:



IPv4 Addressing (contd..)

- An IPv4 address is divided into 2 parts: Prefix and Suffix

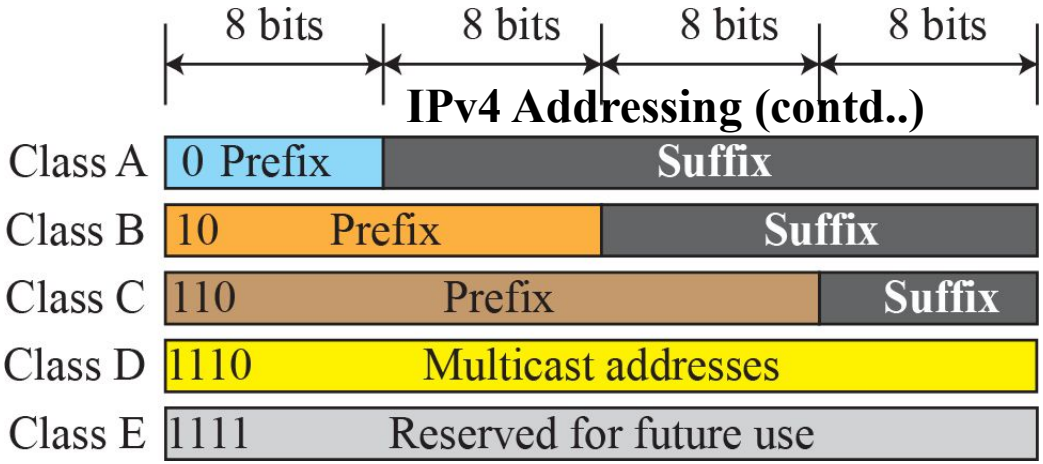


- Prefix: The ‘n’ prefix bits in the IPv4 address is used to define the network
- Suffix: The ‘32-n’ bits in the IPv4 address is called as suffix which is used to identify the machine inside the network

IPv4 Addressing (contd..)

Classful Addressing

- The whole address space was divided into five classes (class A, B, C, D, and E)
- This scheme is referred to as classful addressing



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

- For class A prefix length is 8, suffix is 24 (2^{24} hosts inside every network)
- For class B prefix length is 16, suffix is 16 (2^{16} hosts inside every network)
- For class C prefix length is 24, suffix is 8 (2^{8} hosts inside every network)

IPv4 Addressing (contd..)

Going from classful addressing to classless addressing

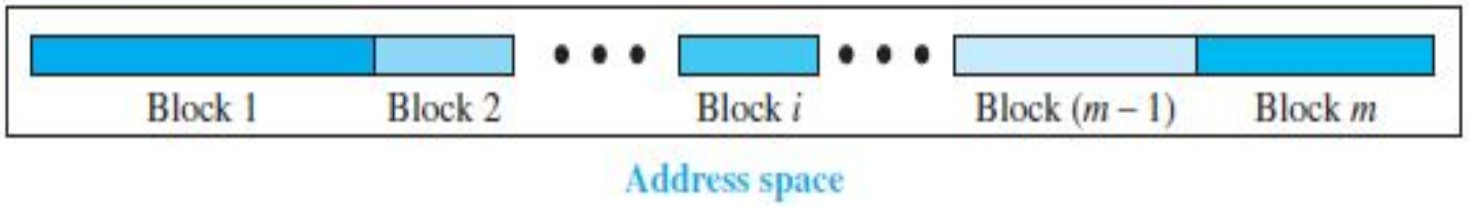
- In case of classful addressing, classes A, B and C is going to support prefix of length 8, 16 and 24 only
- We can't get a prefix value other than this. Suppose an organization wants 2000 address means, class B type address only will be allocated where we can get 65536 address. Since organization needs only 2000 and remaining 65536-2000 addresses gets wasted
- In order to overcome from this address wastage problem we are going from classful addressing to classless addressing
- In classless addressing, we can have different prefix length

IPv4 Addressing (contd..)

Classless Addressing

- In classless addressing, the whole address space is divided into variable length blocks
- The prefix in the address defines the block(network) and suffix defines the node(device)
- A block can have 2^0, 2^1, 2^2…..2^{32} addresses
- Number of addresses in every block must be always of power of 2

Figure 18.19 *Variable-length blocks in classless addressing*



IPv4 Addressing (contd..)

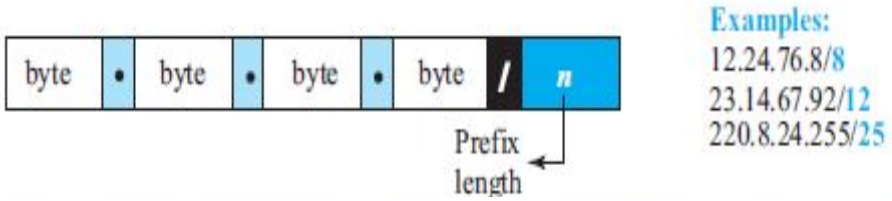
Classless Addressing (contd..)

- Prefix length in classless addressing is variable
- Prefix length ranges from 0 to 32
- Classful addressing is a special case of classless addressing because class A can be called classless addressing with prefix length 8, class B can be called classless addressing with prefix length 16 and class C can be called classless addressing with prefix length 24

Prefix Length: Slash Notation

- In order to find the prefix length (n) in case of classless addressing a notation called slash notation or classless interdomain routing(CIDR) is used.

Figure 18.20 *Slash notation (CIDR)*



IPv4 Addressing (contd..)

Classless Addressing (contd..)

- In classless addressing the prefix length, n can also be represented using Mask
- This mask is sometimes called as Subnet Mask
- Given a subnet mask for a network , we can calculate the prefix length

Example:

What is the prefix value for the network whose subnet mask is 255.255.254.0?

Solution:

Convert given subnet mask to binary values which is,

11111111 11111111 11111110 00000000

- Count the number of 1's in the above binary value which is 23. So prefix value is 23
- Mask will contain all the starting bits as 1 followed by zero bits
- For example following is not a valid mask,

11111111 11111111 00000001 00000000 (255.255.1.0)

IPv4 Addressing (contd..)

Classless Addressing (contd..)

Extracting information from classless addressing:

Given a classless address with prefix length n, following information can be retrieved:

1. Number of addresses in the block(network) is $N = 2^{32-n}$

   (N should be always in power of 2)
2. First address in the block can be found by keeping n leftmost bits and set the (32-n) rightmost bits all to 0s.
3. Last address in the block can be found by keeping n leftmost bits and set the (32-n) rightmost bits all to 1s.

Example:

- A classless address is given as 167.199.170.82/27. Find number of addresses in the block, first address and last address of the block

Solution:

From the given classless IP address prefix length, n=27

Number of addresses in the block, $N = 2^{32-n} = 2^{32-27} = 2^5 = 32$ addresses

IPv4 Addressing (contd..)

Classless Addressing (contd..)

Example (contd..):

- The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

  Address: 167.199.170.82/**27** 10100111 11000111 10101010 01010010

  First address: 167.199.170.64/**27** 10100111 11000111 10101010 01000000

- The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

  Address: 167.199.170.82/**27** 10100111 11000111 10101010 01010010

  Last address: 167.199.170./**27** 10100111 11000111 10101010 01011111

IPv4 Addressing (contd..)

- The below example shows how to calculate prefix length (n) from the number of addresses (N) in a block

Example:

- An ISP has requested a block of 1000 addresses. Since 1000 is not a power of 2, 1024 addresses are granted. Then prefix length is calculated as,

Here number of addresses granted, $N = 1024 = 2^{10}$

Prefix length, n = 22

IPv4 Addressing (contd..)

Subnetting

- ISP which is holding a block of IP addresses needs to assign to the organizations which request it
- Each organization network will be considered as one sub network

IPv4 Addressing (contd..)

Subnetting (contd..)

Example:

- An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of 10 addresses, one subblock of 60 addresses, and one subblock of 120 addresses. Design the subblocks.

Solution:

From the given classless address 14.24.74.0/24,

Prefix length, n = 24

Number of addresses in the block, $N = 2^{32-n} = 2^{32-24} = 2^8 = 256$ addresses

Given the first address in the block is,

First address: 14.24.74.0/24

The last address in the block is,

Address: 14.24.74.0/24 00001110 00011000 01001010 00000000

Last address: 14.24.74.255/24 00001110 00011000 01001010 11111111

IPv4 Addressing (contd..)

Subnetting (contd..)

Example:

- We assign addresses to subblocks, starting with the largest and ending with the smallest one.

1st subblock:

The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as

$n_1 = 25$

The first address in this block is

First address: 14.24.74.0/25 00001110 00011000 01001010 00000000

The last address in this block is

First Address: 14.24.74.0/25 00001110 00011000 01001010 00000000

Last address: 14.24.74.127/25 00001110 00011000 01001010 01111111

IPv4 Addressing (contd..)

Subnetting (contd..)

Example:

- We assign addresses to subblocks, starting with the largest and ending with the smallest one.

2nd subblock:

The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2. We allocate 64 addresses. The subnet mask for this subnet can be found as

$$n_2 = 26$$

The first address in this block is

First address: 14.24.74.128/26 00001110 00011000 01001010 10000000

The last address in this block is

First Address: 14.24.74.128/26 00001110 00011000 01001010 10000000

Last address: 14.24.74.191/26 00001110 00011000 01001010 10111111

IPv4 Addressing (contd..)

Subnetting (contd..)

Example:

- We assign addresses to subblocks, starting with the largest and ending with the smallest one.

3rd subblock:

The number of addresses in the smallest subblock, which requires 10 addresses, is not a power of 2. We allocate 16 addresses. The subnet mask for this subnet can be found as

$$n_3 = 28$$

The first address in this block is

First address: 14.24.74.192/28 00001110 00011000 01001010 11000000

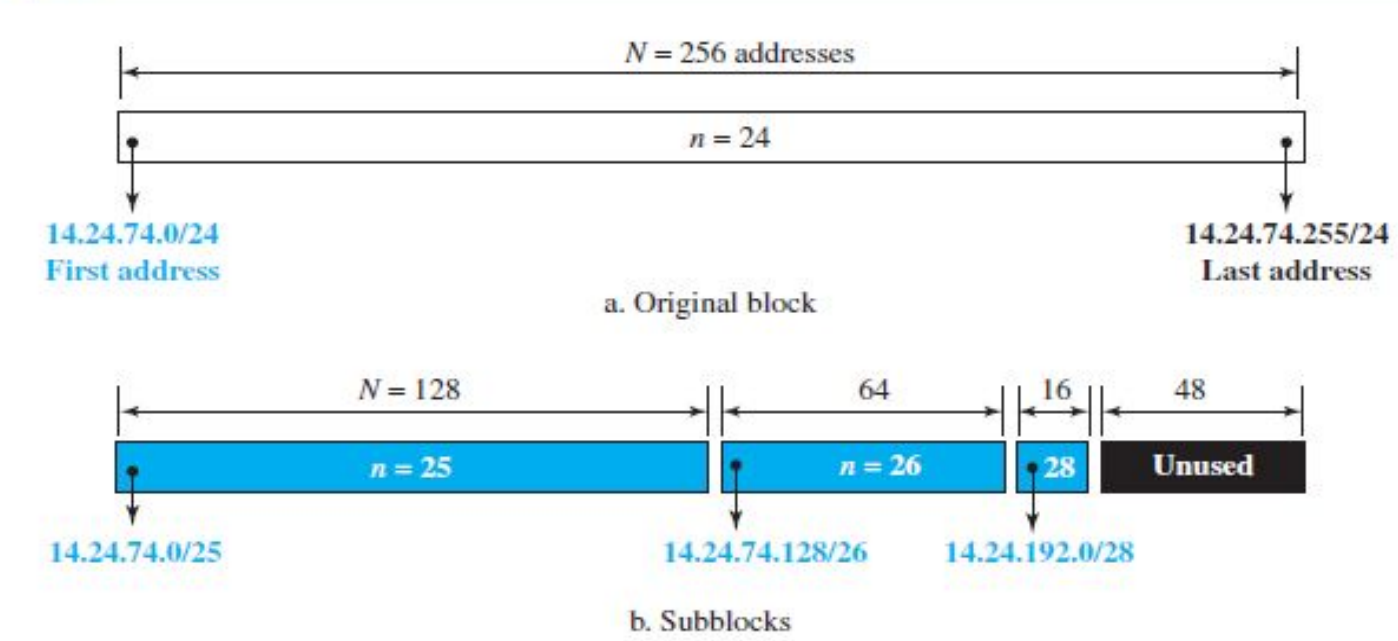
The last address in this block is

First Address: 14.24.74.192/28 00001110 00011000 01001010 11000000

Last address: 14.24.74.207/28 00001110 00011000 01001010 11001111

IPv4 Addressing (contd..)

Diagrammatic representation of example in previous slide,

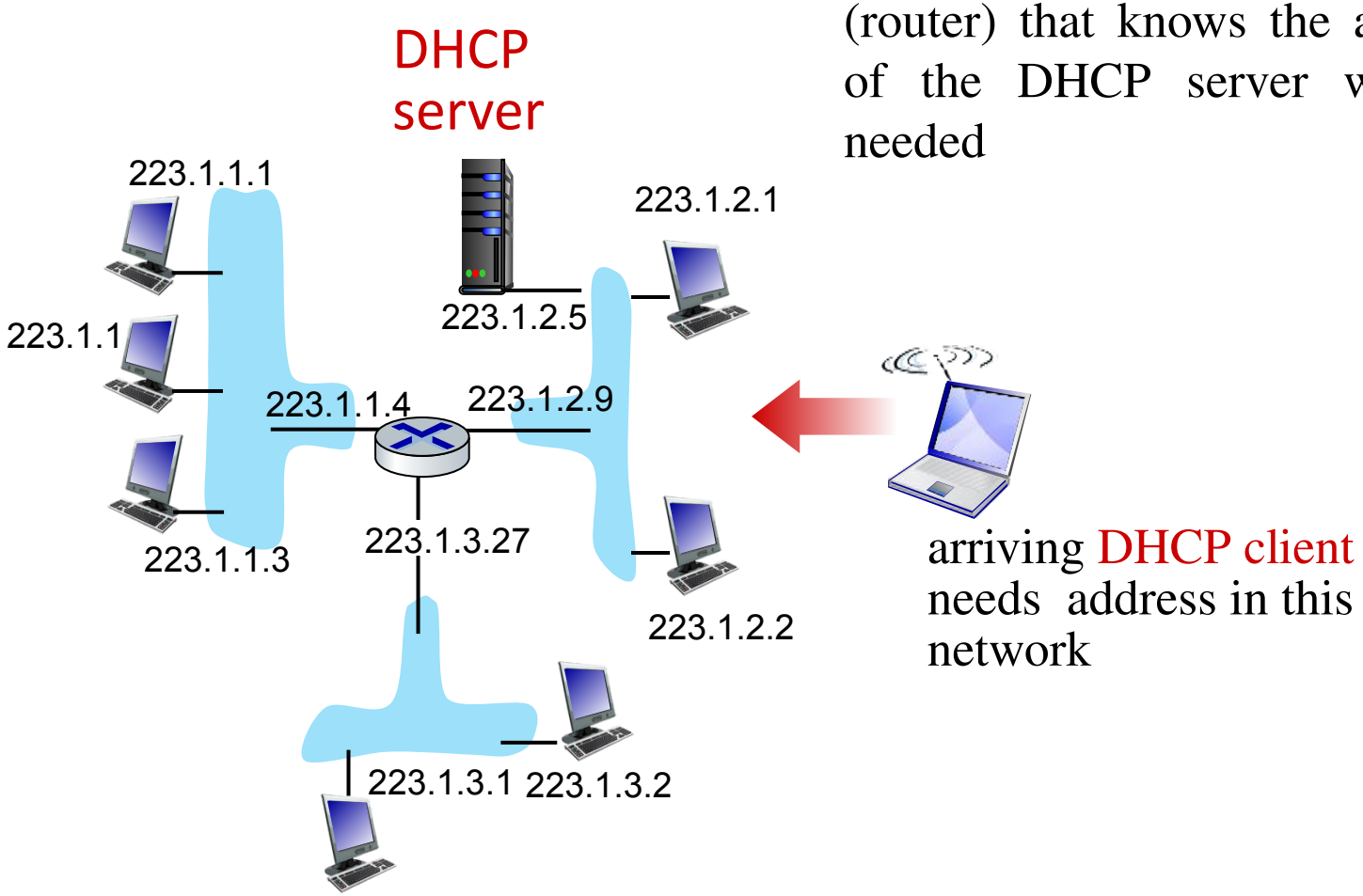


DHCP Protocol

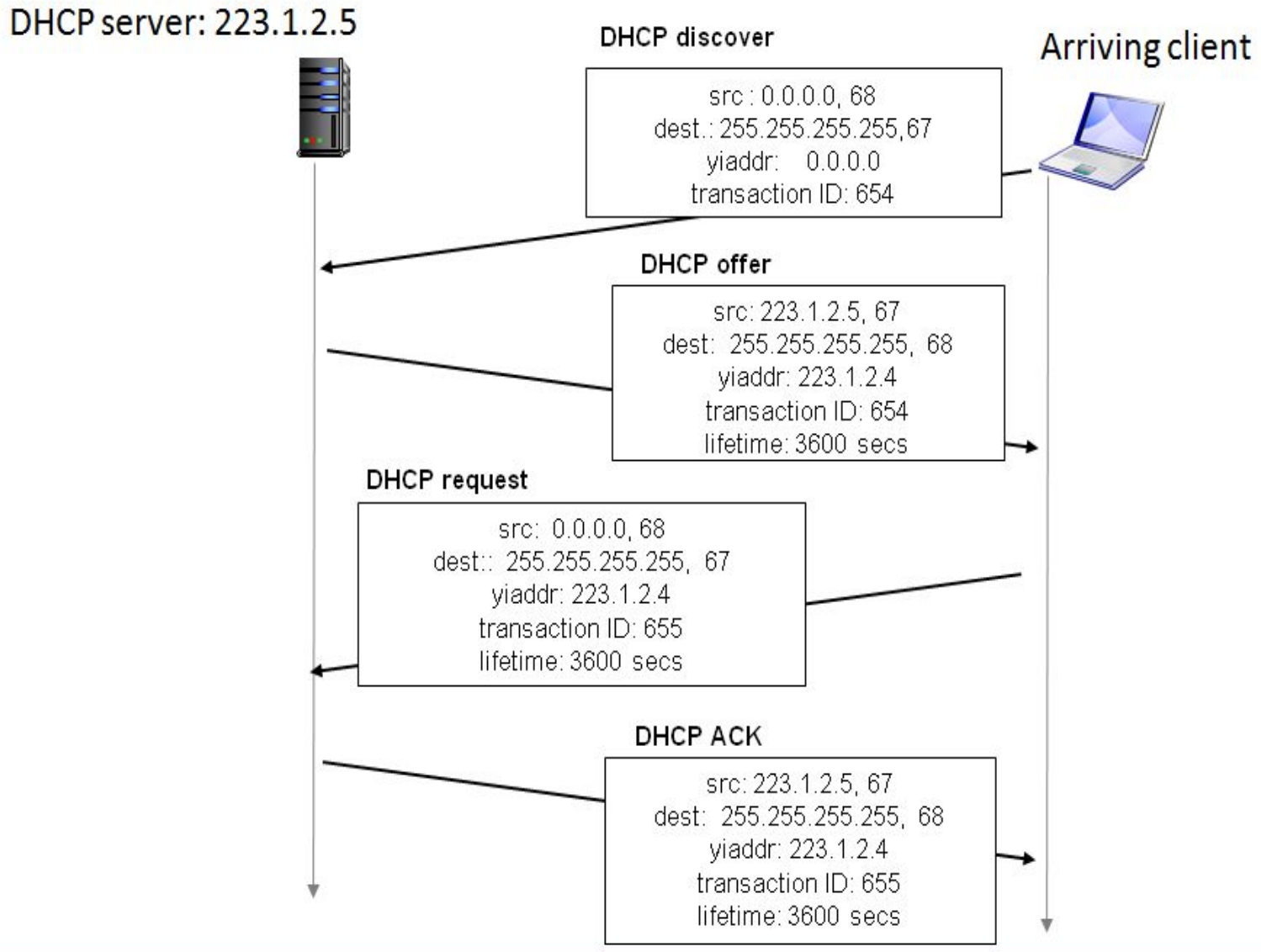
- DHCP – Dynamic Host Configuration Protocol
- DHCP is the protocol used to assign IP address for a newly arrived client system in a network
- In addition to IP address assignment, DHCP also allows a host to learn additional information, such as its subnet mask, the address of its first-hop router (often called the default gateway), and the address of its local DNS server
- DHCP protocol is called as plug-and-play protocol or zeroconf (zero-configuration) protocol

DHCP Protocol (contd..)

Each subnet will have a DHCP server. If no server is present on that subnet, DHCP relay agent (router) that knows the address of the DHCP server will be needed



DHCP Protocol (contd..)



DHCP Protocol (contd..)

- Assignment of IP address to a newly arrived client is done using following 4 steps:
  1. DHCP Discover
  2. DHCP Offer
  3. DHCP Request
  4. DHCP ACK

1. DHCP Discover

- Newly arrived client machine sends DHCP Discover message in order to discover who is the DHCP server. Source and Destination IP address in this message is as follows

  Source IP : 0.0.0.0 (since client does not have any IP address)

  Destination IP : 255.255.255.255 (Broadcast IP since client does not know who is DHCP server)

DHCP Protocol (contd..)

2. DHCP Offer

- DHCP server after receiving Discover message sends DHCP offer message to client. This offer message contains the IP address that is offered by the DHCP server (offered IP address is present in Your IP address(yiaddr) field). Source and Destination IP address in this message is as follows

  Source IP : 223.1.2.5(IP address of DHCP server)

  Destination IP : 255.255.255.255 (Broadcast IP since client is not having any IP address)

3. DHCP Request

- Client machine after been offered an IP address requests that IP address from DHCP server. Source and Destination IP address in this message is as follows

  Source IP : 0.0.0.0 (since client does not have any IP address)

  Destination IP : 255.255.255.255 (Broadcast IP since client wants to inform all about the IP addr it chooses)

DHCP Protocol (contd..)

4. DHCP ACK

- Finally DHCP server acknowledge the IP address requested by the client. Source and Destination IP address in this message is as follows

  Source IP : 223.1.2.5(IP address of DHCP server)

  Destination IP : 255.255.255.255 (Broadcast IP since client is not having any IP address)

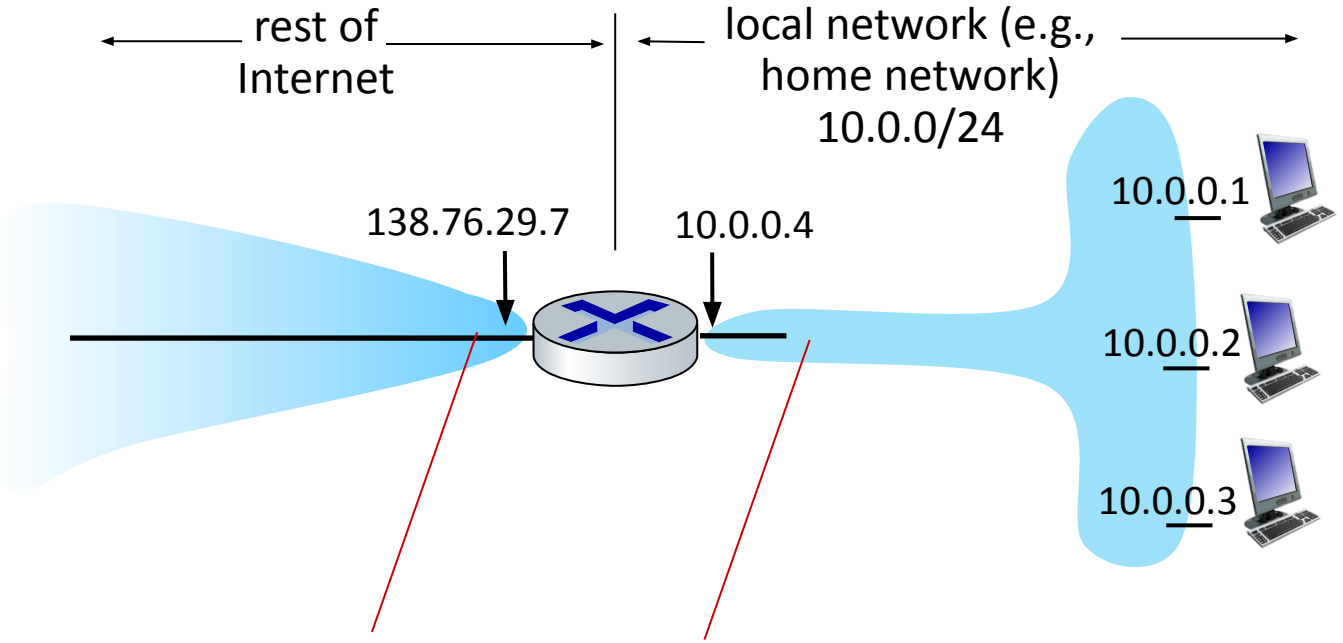
- After DHCP ACK step completed, an IP address will be assigned to the newly arrived client machine

Network Address Translation (NAT)

- Sometimes the IP address that is hold by a machine have meaning only within the network. Such IP addresses are called as Private IP address which is used for communication within the network. For communication outside the network, this IP address cannot be used and needs to be converted
- Network Address Translation is used to convert an private IP address into Global IP address
- This address translation will be done at router and router performing this address translation is called as NAT router
- In the diagram shown in the next slide, router is having 2 interfaces one connected to local network and another connected to the rest of the world
- Whenever any machine in the local network wants to send data to a machine which is outside of its network, it first send data to its router
- NAT router when receives data is going to convert private IP address and add its IP address to the data(IP address of interface that is connected to the outside world)

Network Address Translation (contd..)

NAT: all devices in local network share just one IPv4 address as far as outside world is concerned

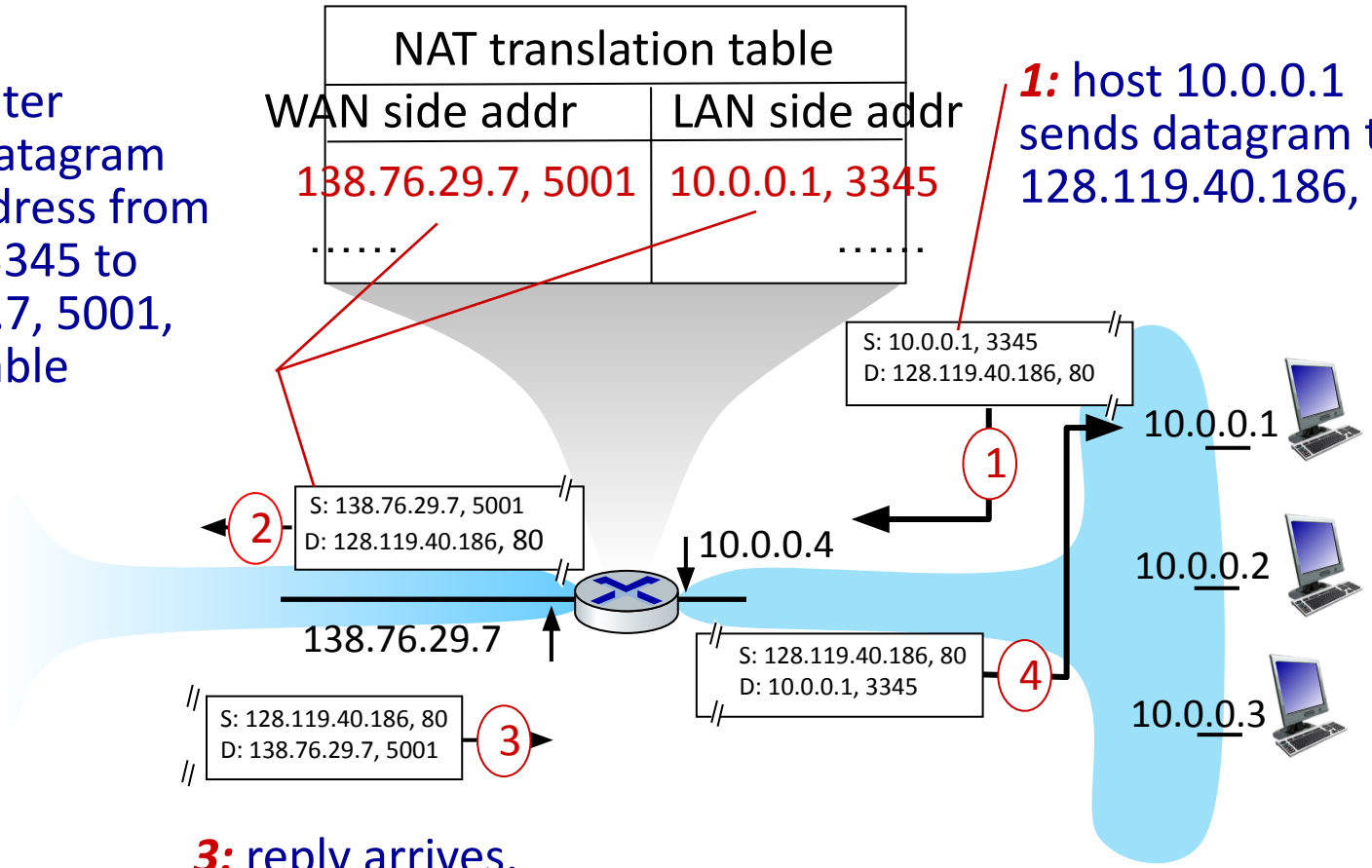


all datagrams leaving local network have same source NAT IP address: 138.76.29.7

Datagrams from local network

Network Address Translation (contd..)

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table



3: reply arrives, destination address: 138.76.29.7, 5001

Network Address Translation (contd..)

- When NAT router performs translation from private IP address to global IP address, an entry will be made to a table called NAT Translation table (shown in previous slide)
- This table contains IP address and also port numbers
- When data is received by NAT router from outside world it has to be given to system present inside the local network. NAT router again consults the table and obtain the address of machine inside local network to whom data to be delivered

IPv6

- IPv6 is a network layer protocol as like that of IPv4

Moving from IPv4 to IPv6

- Number of address in IPv4 is 2^{32} (2^{32} = nearly greater than 4 billion addresses)
- Currently all the 4 billion addresses of this IPv4 has been allocated
- When a new organization requests for IP address, some IP address has to be allocated
- In order to address this, a new IP protocol has been proposed whose version is 6

Advantages of IPv6 over IPv4

- Larger address space. An IPv6 address is 128 bits long whereas IPv4 is only 32-bit long.
- Better header format. Options of IPv6 are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process
- New options. IPv6 has new options to allow for additional functionalities
- Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications

IPv6 Addresses

- An IPv6 address is 128 bits long(16 bytes). It is 4 times greater than IPv4 address

Representation of IPv6 address:

- Representing in binary format requires 128 bits which is difficult for humans to read. So normally all IPv6 address is represented using Colon hexadecimal notation

Colon hexadecimal Notation:

- In this notation, 128 bits are divided into 8 sections each section contains 4 hexadecimal digit of 16 bits length

Binary (128 bits)	1111111011110110	…	1111111100000000
Colon Hexadecimal	FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00		

IPv6 Addresses (contd..)

Abbreviation of IPv6 addresses

- If many of the digits are zeros, we can abbreviate the address.
- The leading zeros of a section can be omitted

  0074 □ 74

  000F □ F
- 3210 cannot be abbreviated as 321. Only leading zeros can be omitted not trailing zeros

Zero Compression

- If consecutive sections in the address are zeros then zero compression can be applied which means remove all zeros and replace with double semicolon

FDEC:0:0:0:0:BBFF:0:FFFF ⟶ FDEC::BBFF:0:FFFF

IPv6 Addresses (contd..)

CIDR Notation

- As like in IPv4, IPv6 also has CIDR notation

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

- The above address shows the first 60 bits in the IPv6 address is used as the prefix

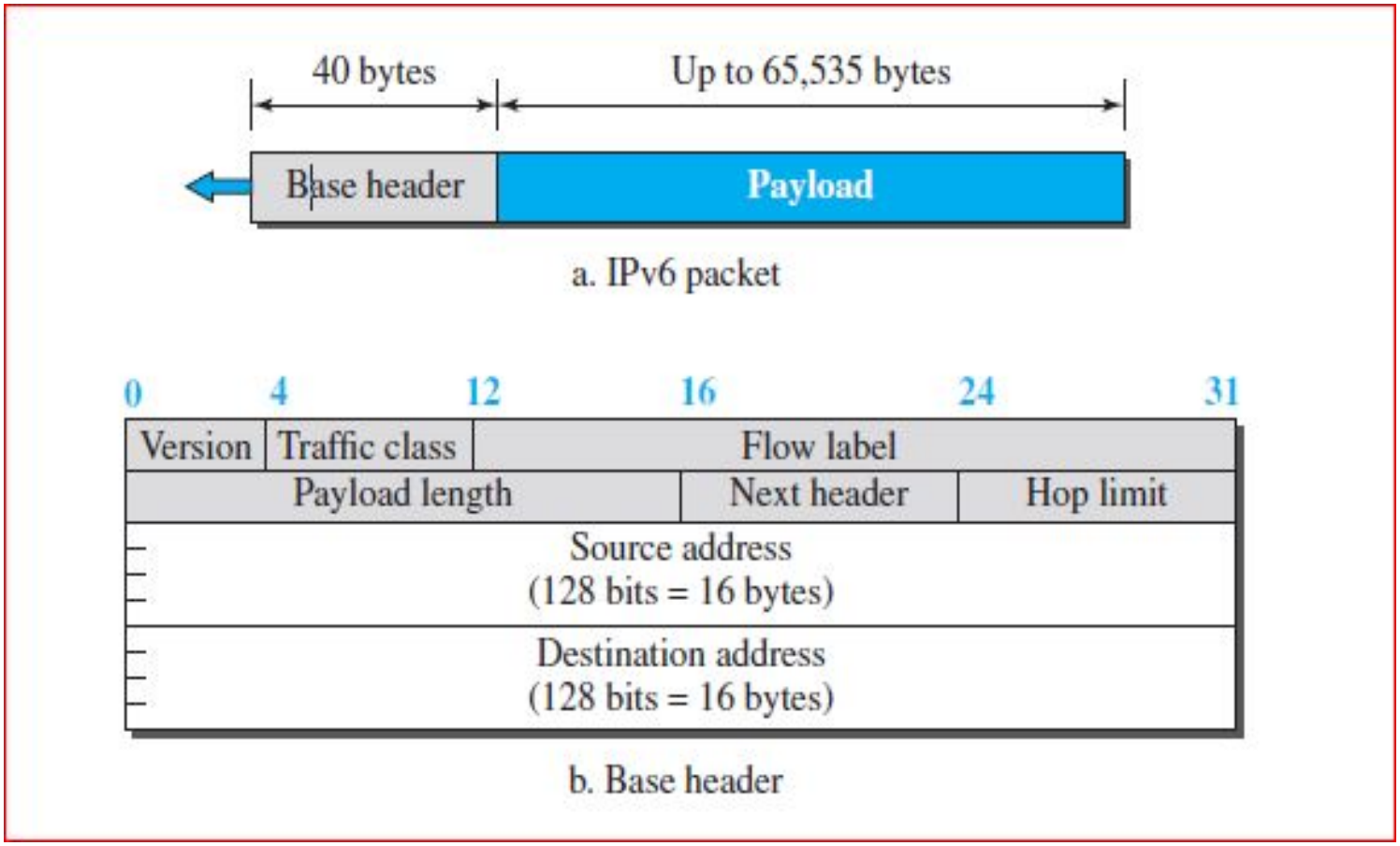
Address Space

- Since IPv6 is of 128 bits, number of addresses is 2^{128}
- The number of addresses in IPv6 is

340, 282, 366, 920, 938, 463, 374, 607, 431, 768, 211, 456

IPv6 Datagram Format

- The datagram format for IPv6 is,



- Base Header occupies 40 bytes and payload can be upto 65,535 bytes

IPv6 Datagram Format (contd..)

The various fields in the base header are,

1.Version field:

- 4-bit version field that defines the version number of IP. For IPv6 this field has value 6

2.Traffic class:

- 8-bit traffic class is used to distinguish different payloads and specify how the payload should be handled. This field to similar to Type of Service field in IPv4

3.Flow Label:

- This 20-bit field is used to provide special handling for a particular flow of data

4.Payload Length:

- This 16-bit field is used to define the length of the payload alone

5.Next Header:

- This 8-bit next header field is used to define the type of the first extension header or the type of data that follows the base header

IPv6 Datagram Format (contd..)

6.Hop Limit:

- The 8-bit hop limit is used for the same purpose as like that of TTL field in IPv4.

7.Source Address:

- 128-bit source address contains the IP address of source system

8.Destination Address:

- 128-bit destination address contains the IP address of destination system

IPv6 Datagram Format (contd..)

Fragmentation and Reassembling

- Fragmentation means breaking the data into various fragments(pieces) if the size of the data is greater than the Maximum Transmission Unit(MTU) of a link
- IPv6 datagrams are fragmented only at the source not at the router. Reassembling takes place at destination
- Source can check the size of the packet and do fragmentation if required
- Router while receiving the data from source, check if it can transmit that data in the next link. If not possible means router is will send ICMPv6 error message to source

IPv4 vs IPv6

IPv4	IPv6
No. of bits in the IPv4 address is 32bits	No. of bits in the IPv6 address is 128 bits
It is written i dotted decimal notation	It is written in colon hexadecimal notation
Number of possible IPv4 address is nearly 4.3 billion (smaller address space than IPv6)	Number of possible IPv6 address is infinite(larger address space)
Address depletion problem arise with IPv4 address space	No address depletion problem with IPv6
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by the sender
In IPv4 checksum field is available	In IPv6 checksum field is not available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20 to 60 bytes	IPv6 has header of 40 bytes fixed
Example: 192.1.1.0	**Example:** FDEC::BBFF:0:FFFF

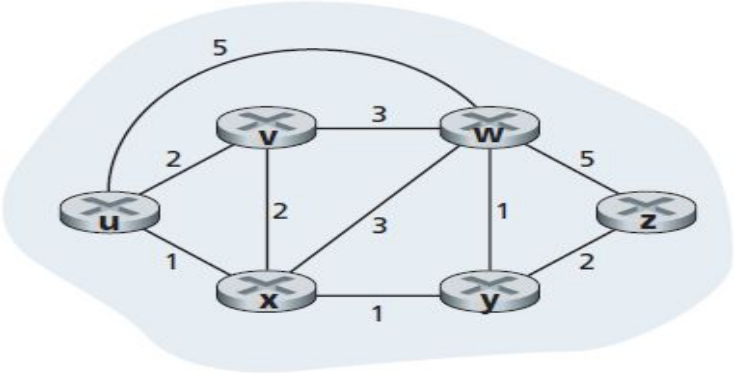
Routing Algorithms and Protocols

Routing Algorithms

- Routing algorithms is used to determine good paths (equivalently, routes), from senders to receivers, through the network of routers. Typically, a "good" path is one that has the least cost.

Least-cost path:

- Given any two nodes there are typically many paths between the two nodes, with each path having a cost. Among the multiple paths which path is having the least cost is called as Least-cost path



Abstract graph model of a computer network

- In the above graph, least-cost path between source node u and destination node w is (u, x, y, w) with a path cost of 3

Routing Algorithms

- 2 types of routing algorithms are there:
    1. Centralized routing algorithm
    2. Decentralized routing algorithm

1. Centralized routing algorithm:

- computes the least-cost path between a source and destination using complete, global knowledge about the network.
- The algorithm takes the connectivity between all nodes and all link costs as inputs
- Example: Link-state Routing algorithm

2. Decentralized routing algorithm:

- In this algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner by the routers.
- No node has complete information about the costs of all network links. Instead, each node begins with only the knowledge of the costs of its own directly attached links. Then, through an iterative process of calculation and exchange of information with its neighboring nodes, a node gradually calculates the least-cost path to a destination or set of destinations.
- Example: Distance-vector Routing algorithm

Link-State (LS) Routing Algorithm

- In a link-state algorithm, the network topology and all link costs are known, (ie) available as input to the LS algorithm. This is accomplished by having each node broadcast link-state packets to all other nodes in the network
- The link-state routing algorithm is also known as Dijkstra's algorithm

Link-State (LS) Routing Algorithm (contd..)

The working of LS algortihm is shown below:

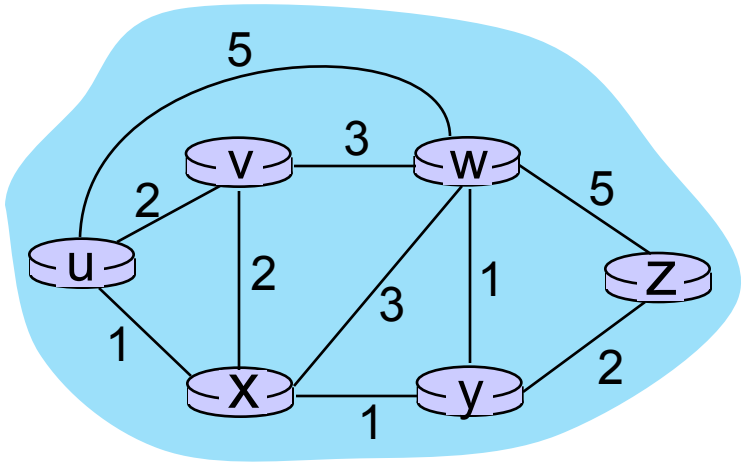
Dijkstra's link-state routing algorithm

```
1  Initialization:
2    N' = {u}                    /* compute least cost path from u to all other nodes */
3    for all nodes v
4      if v adjacent to u        /* u initially knows direct-path-cost only to direct neighbors */
5        then D(v) = c_u,v       /* but may not be minimum cost! */
6      else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N' :
12     D(v) = min ( D(v), D(w) + c_w,v )
13   /* new least-path-cost to v is either old least-cost-path to v or known
14   least-cost-path to w plus direct-cost from w to v */
15 until all nodes in N'
```

Link-State (LS) Routing Algorithm (contd..)

Example 1:

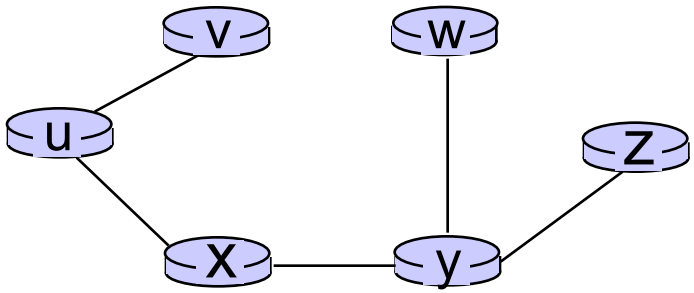
- Consider the following graph. Construct the forwarding table for the node ‘u’ for the given graph



Link-State (LS) Routing Algorithm (contd..)

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Least cost path Tree from node u



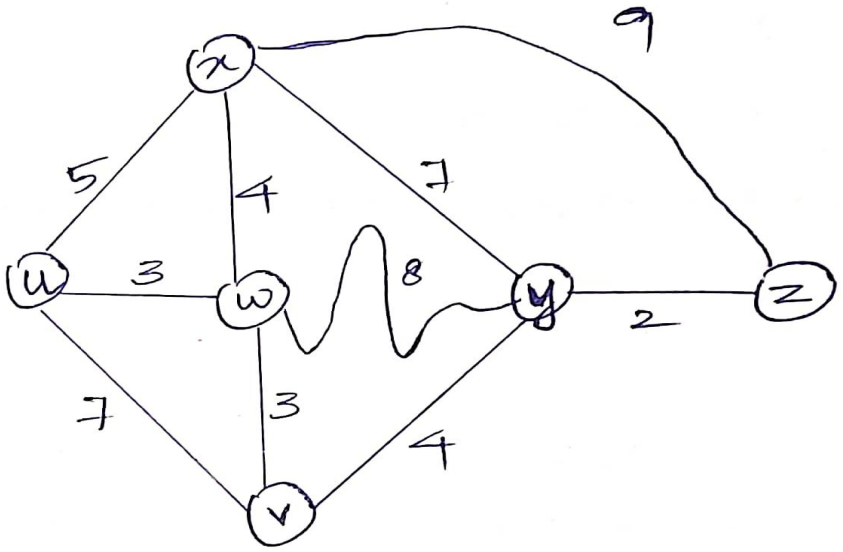
Forwarding table at u

destination	outgoing link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Link-State (LS) Routing Algorithm (contd..)

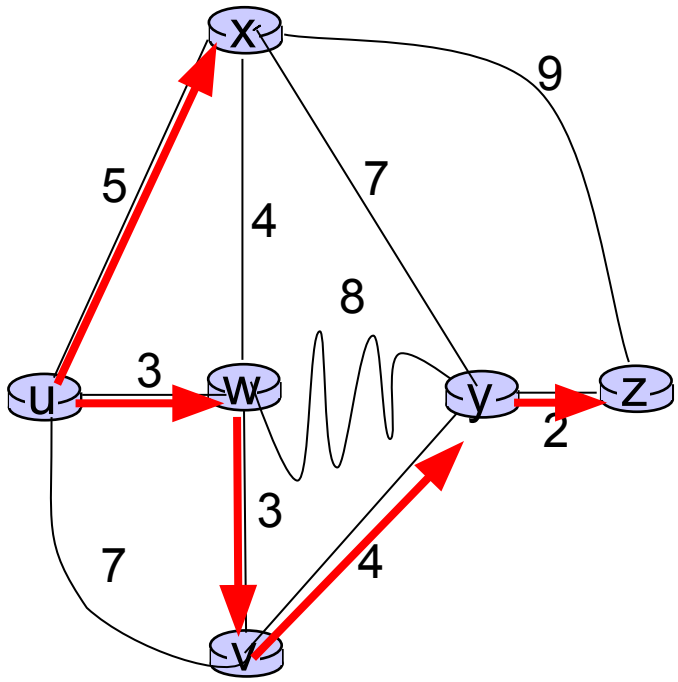
Example 2:

- Consider the following graph. Construct the forwarding table for the node ‘u’ for the given graph using LS routing algorithm



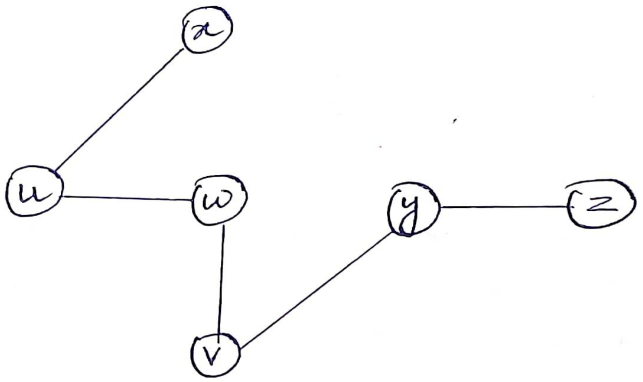
Link-State (LS) Routing Algorithm (contd..)

Step	N′	v D(v), p(v)	w D(w), p(w)	x D(x), p(x)	y D(y), p(y)	z D(z), p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12, y
5	uwxvyz					



Link-State (LS) Routing Algorithm (contd..)

Least cost path tree from u



Forwarding table at u

destination	outgoing link
x	(u, x)
w	(u, w)
v	(u, w)
y	(u, w)
z	(u, w)

Distance-Vector(DV) Routing Algorithm

- LS algorithm is an algorithm using global information
- But the distance vector(DV) algorithm is iterative, asynchronous, and distributed.
- It is distributed in that each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors.
- It is iterative in that this process continues on until no more information is exchanged between neighbors.

Distance-Vector(DV) Routing Algorithm (contd..)

The working of DV algortihm is shown below:

Table 20.1 *Distance-Vector Routing Algorithm for a Node*

```
1   Distance_Vector_Routing ( )
2   {
3       // Initialize (create initial vectors for the node)
4       D[myself] = 0
```

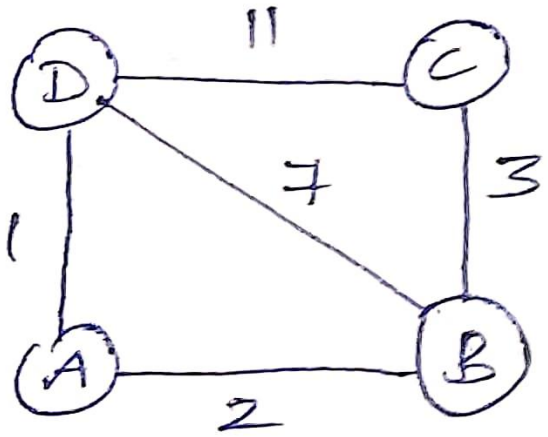
Table 20.1 *Distance-Vector Routing Algorithm for a Node (continued)*

```
5       for (y = 1 to N)
6       {
7           if (y is a neighbor)
8               D[y] = c[myself][y]
9           else
10              D[y] = ∞
11      }
12      send vector {D[1], D[2], ..., D[N]} to all neighbors
13      // Update (improve the vector with the vector received from a neighbor)
14      repeat (forever)
15      {
16          wait (for a vector D_w from a neighbor w or any change in the link)
17          for (y = 1 to N)
18          {
19              D[y] = min [D[y], (c[myself][w] + D_w[y])]      // Bellman-Ford equation
20          }
21          if (any change in the vector)
22              send vector {D[1], D[2], ..., D[N]} to all neighbors
23      }
24  } // End of Distance Vector
```

Distance-Vector(DV) Routing Algorithm (contd..)

Example 1:

- ✔ Consider the following graph and construct the forwarding table at node A using DV algorithm



Distance-Vector(DV) Routing Algorithm (contd..)

Example 1(contd..)

Initial Distance Vector at A

Step 1:

A	0
B	2
C	∞
D	1

Step 2:

At A[ ]

A	0
B	2
C	∞
D	1

Initial DV from B[ ]

A	2
B	0
C	3
D	7

⇒

Revised A1

A	0
B	2
C	5
D	1

Revised A1[ ] = min(A[ ], 2 + B[ ])

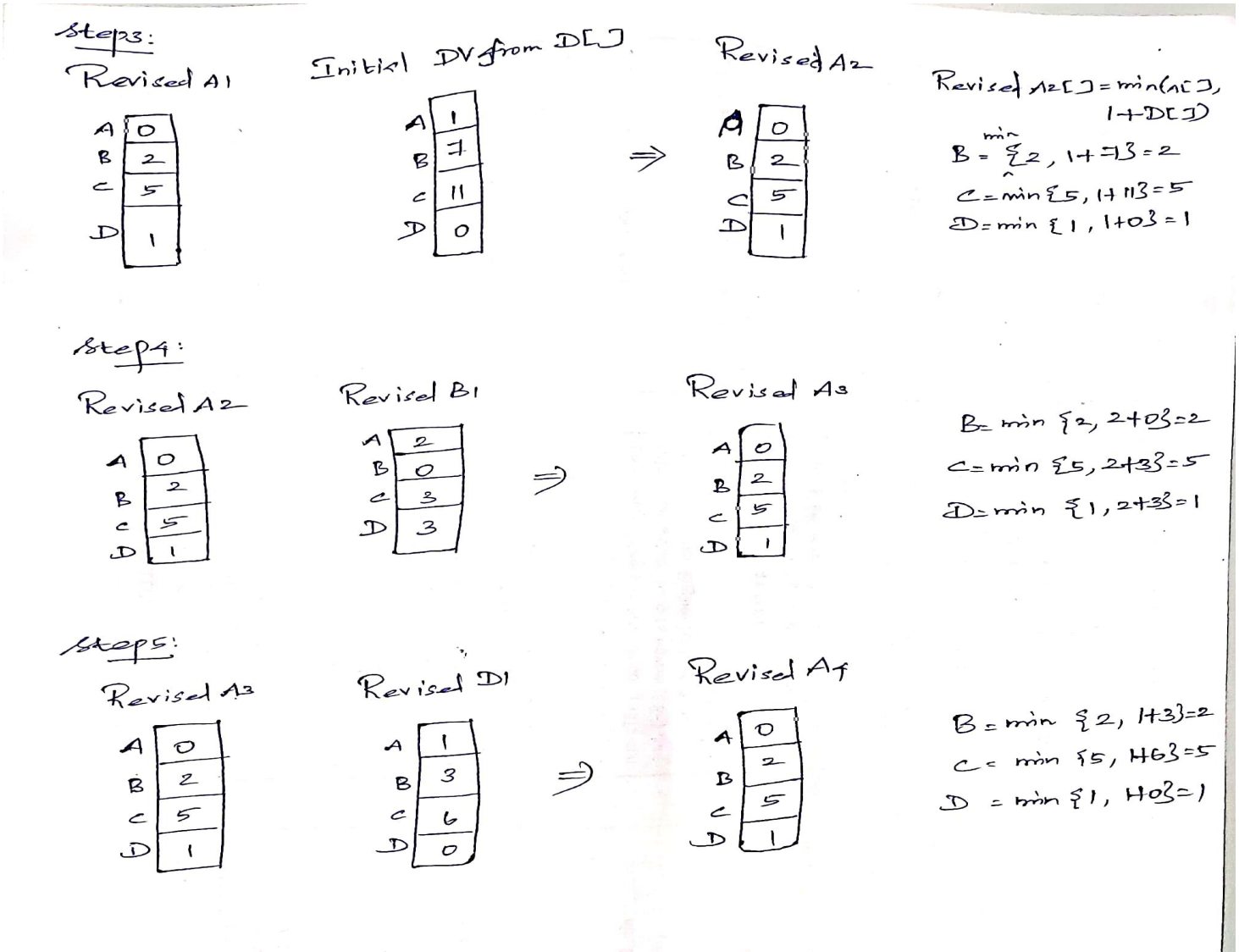
B = min(2, 2+0) = 2

C = min(∞, 2+3) = 5

D = min(1, 2+7) = 1

Distance-Vector(DV) Routing Algorithm (contd..)

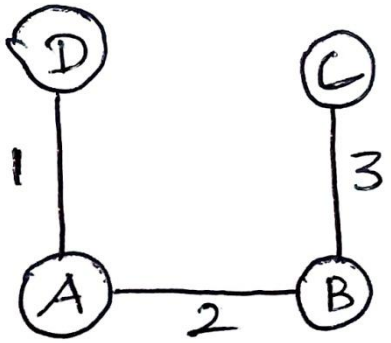
Example 1(contd..)



Distance-Vector(DV) Routing Algorithm (contd..)

Example 1(contd..)

Least Cost path tree from A



Forwarding table at A

destination	outgoing link
B	(A, B)
C	(A, B)
D	(A, D)

Autonomous System

Autonomous System (AS)

- Autonomous System(AS) consisting of a group of routers that are under the same administrative control. Often the routers in an ISP, and the links that interconnect them, constitute a single AS.
- Routers within the same AS all run the same routing algorithm and have information about each other.
- The routing algorithm running within an autonomous system is called an intra-autonomous system routing protocol (Ex:OSPF)
- The routing algorithm running between two autonomous systems is called an inter-autonomous system routing protocol (Ex:BGP)

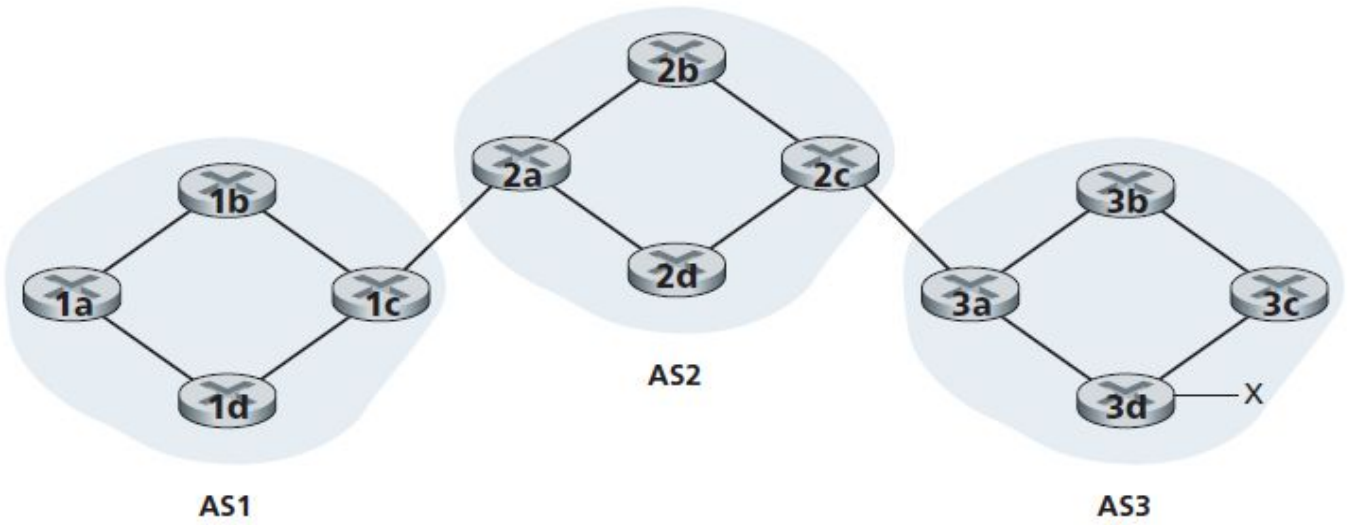
Intra-AS Routing in the Internet: OSPF

- OSPF(Open Shortest Path First) routing protocol is widely used for intra-AS routing in the Internet
- OSPF is a link-state protocol that uses flooding of link-state information and a Dijkstra's least-cost path algorithm
- Each router locally runs Dijkstra's shortest-path algorithm to determine a shortest-path tree to all subnets

Routing Among the ISPs: BGP

- To route a packet across multiple ASs, we need an inter-autonomous system routing protocol. Since an inter-AS routing protocol involves coordination among multiple ASs, communicating ASs must run the same inter-AS routing protocol.
- In the Internet, all ASs run the same inter-AS routing protocol, called the Border Gateway Protocol(BGP)

Routing Among the ISPs: BGP (contd..)



- Above diagram shows that there are 3 Autonomous Systems (AS1, AS2 and AS3)
- Each router in the AS can be either,
    1. Internal Router
    2. Gateway router

Routing Among the ISPs: BGP (contd..)

1. Internal Router

- An internal router connects only to hosts and routers within its own AS. In AS1

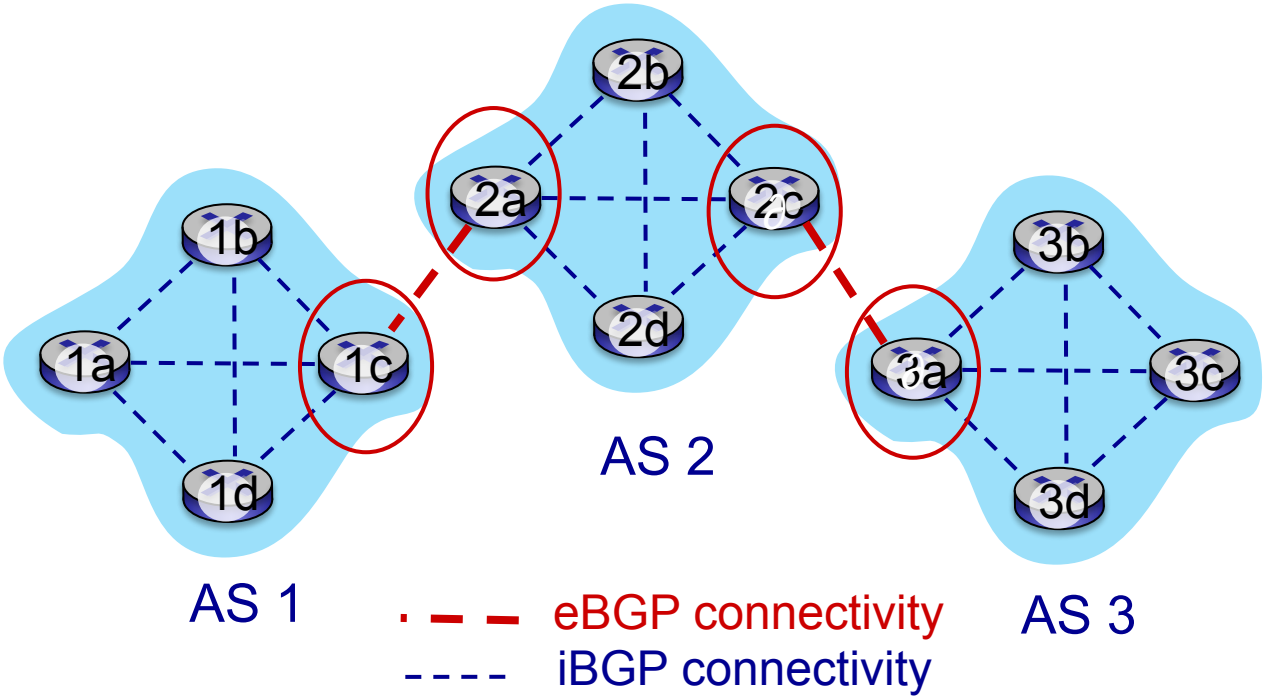
2. Gateway router

- gateway router is a router on the edge of an AS that directly connects to one or more routers in other ASs
- For example in AS1 router 1c is a gateway router; routers 1a, 1b, and 1d are internal routers

There are two types of connection in BGP, (diagram shown in next slide)

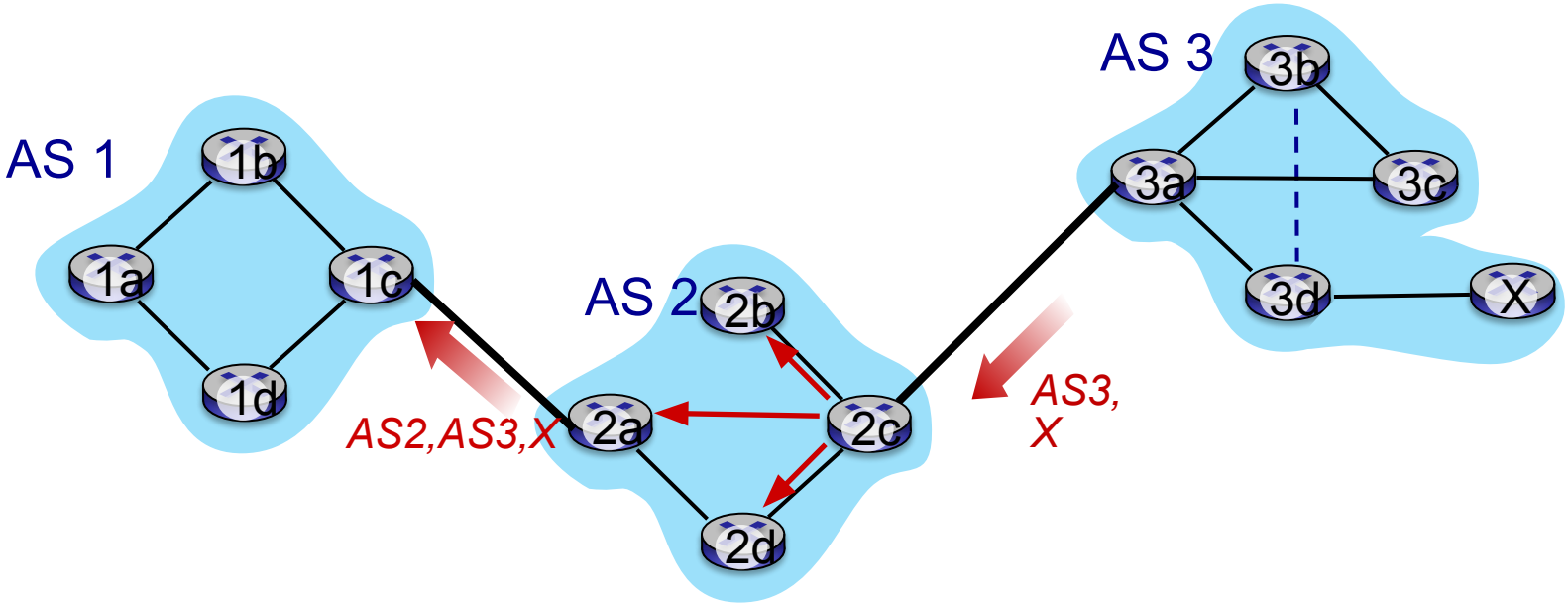
1. eBGP connection(External BGP connection)
   - connection between two AS
2. iBGP connection (internal BGP connection)
   - connection within an AS

Routing Among the ISPs: BGP (contd..)



gateway routers run both eBGP and iBGP protocols

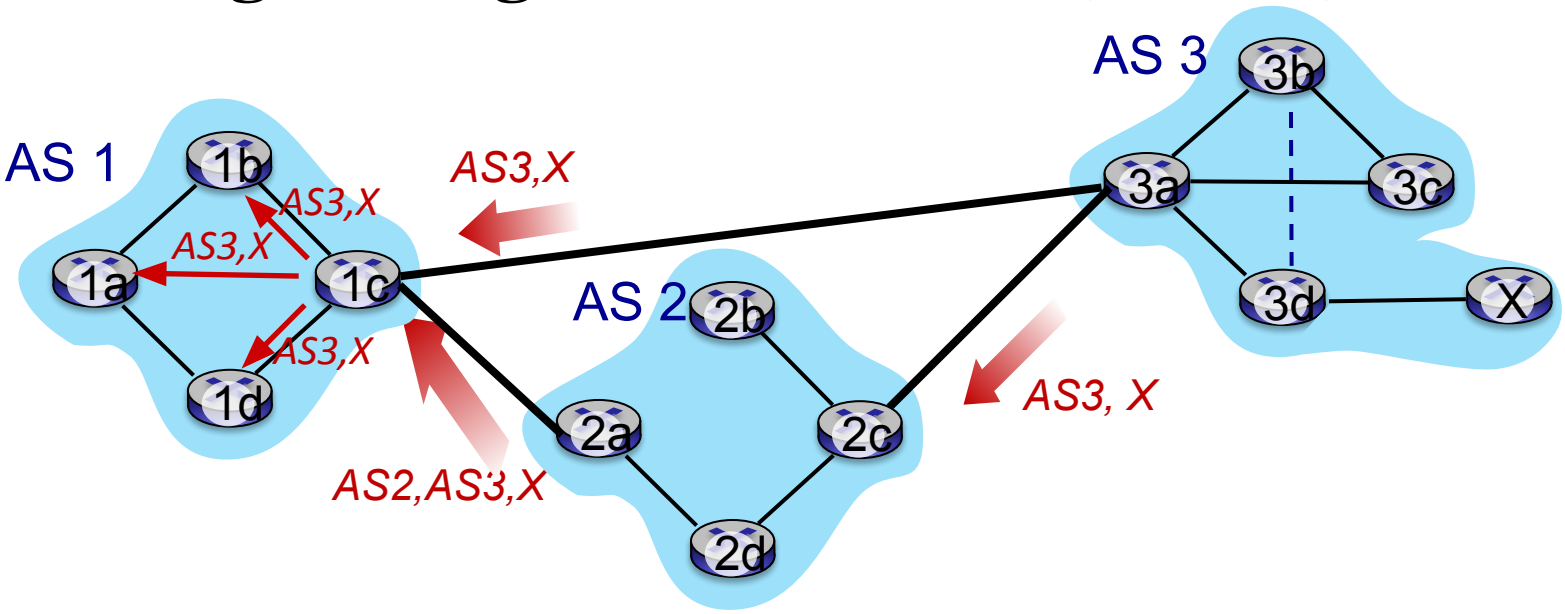
Routing Among the ISPs: BGP (contd..)



Path Advertisement in BGP

- Assume the prefix (prefix length of that autonomous system) of AS3 is X
- The AS3 is going to advertise its prefix to the remaining ASs as shown in the above diagram
- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

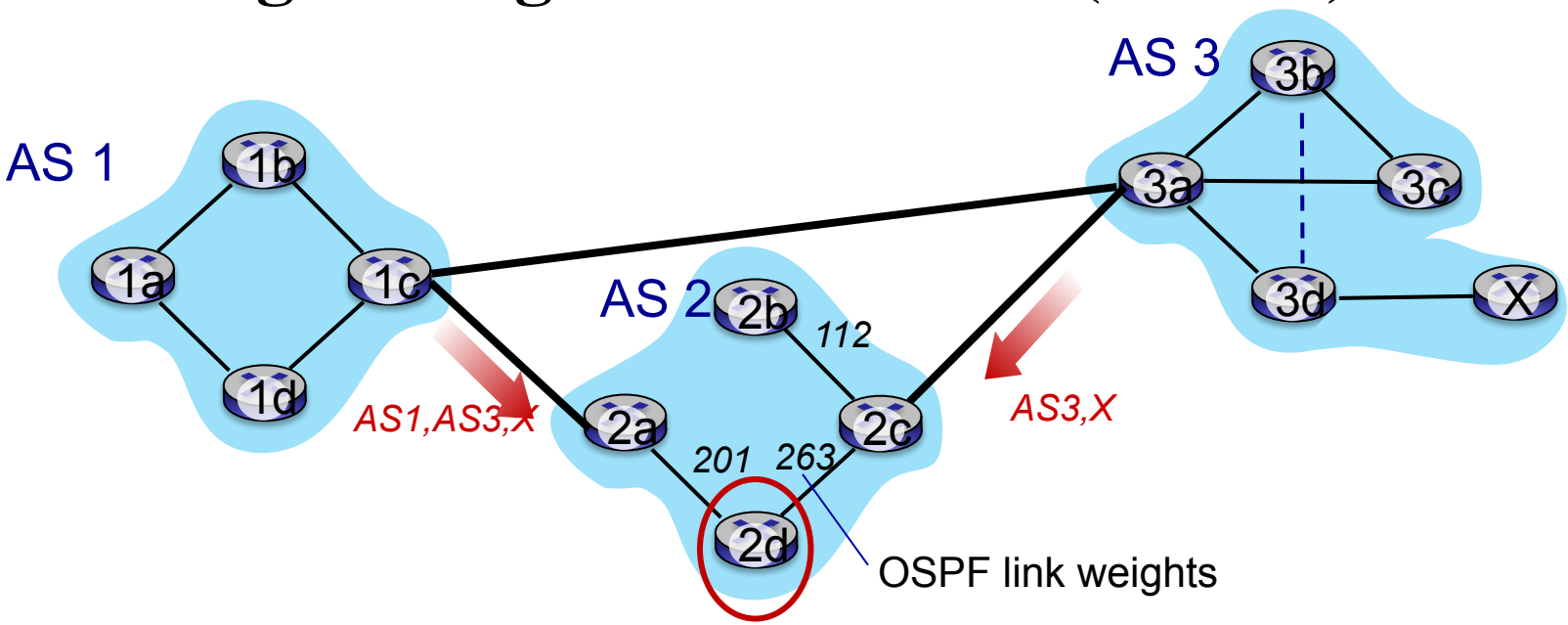
Routing Among the ISPs: BGP (contd..)



Shortest Path

- In the diagram shown above there are two paths between AS1 and AS3. One path is a direct path connecting routers 1c and 3a and another path via AS2
- AS1 gateway router 1c learns path AS2,AS3,X from 2a
- AS1 gateway router 1c learns path AS3,X from 3a
- based on policy, AS1 gateway router 1c chooses path AS3,X (shortest path) and advertises that path within AS1 via iBGP

Routing Among the ISPs: BGP (contd..)



Hot potato routing

- BGP uses Hot Potato routing
- In above diagram, router 2d has two paths to reach AS3. One path via 2c and another path via 2a and 1c
- But 2d always chooses the next router as 2a instead of 2c because 2a only has least cost(201) to reach from 2d. This type routing is called as Hot potato routing

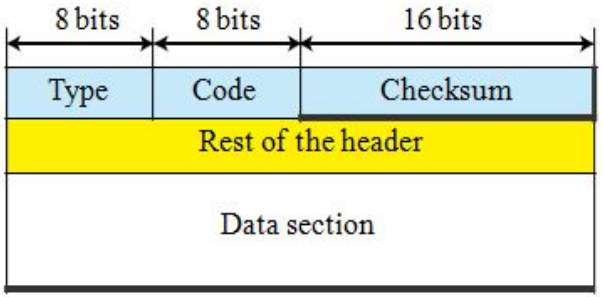
ICMPv4

- The IPv4 has no error-reporting mechanism.
- The IP protocol also lacks a mechanism for host and management queries.
- The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above two deficiencies.
- ICMP is a network layer protocol. It is actually a companion to the IP protocol.
- ICMP messages are encapsulated inside IP datagram. If IP datagram encapsulates ICMP message, protocol field in IP datagram is set to 1

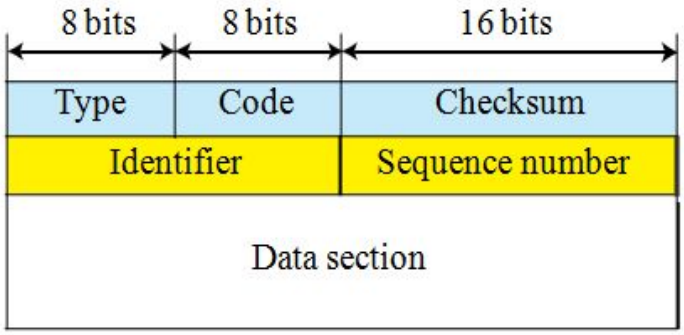
ICMPv4 (contd..)

- ICMP messages divided into 2 types:
  1. Error-reporting message
  2. Query message

- **Format of ICMP messages:**



Error-reporting messages



Query messages

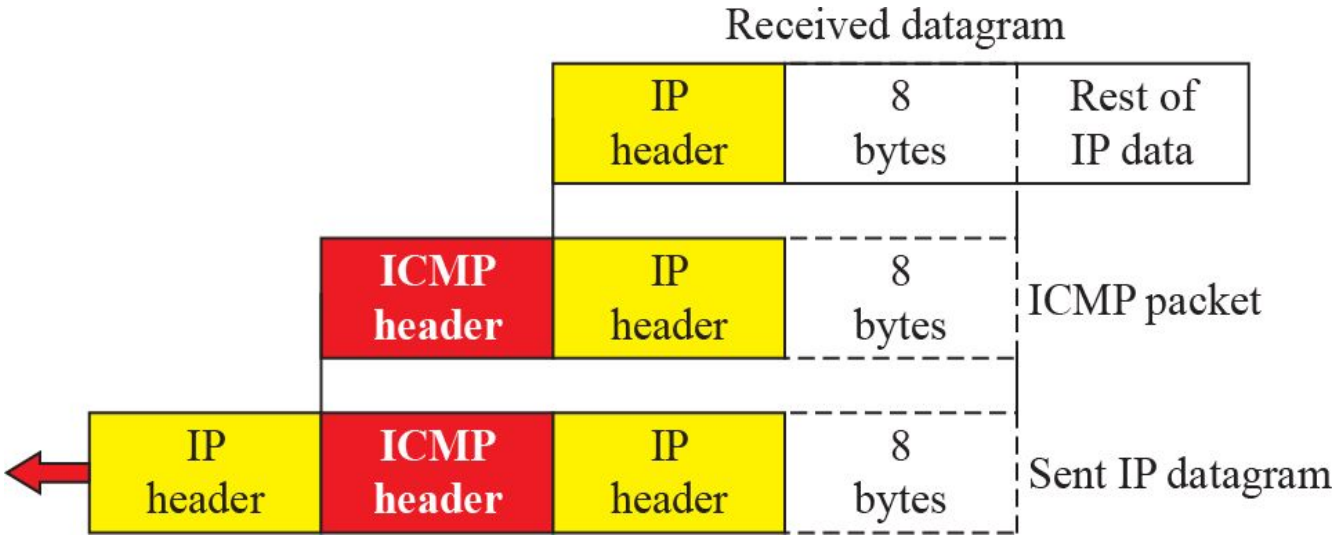
Type and code values

Error-reporting messages	Query messages
03: Destination unreachable (codes 0 to 15)	08 and 00: Echo request and reply (only code 0)
04: Source quench (only code 0)	13 and 14: Timestamp request and reply (only code 0)
05: Redirection (codes 0 to 3)	
11: Time exceeded (codes 0 and 1)	
12: Parameter problem (codes 0 and 1)	

ICMPv4 (contd..)

1. **Error Reporting Message:**

- In order to report the errors that occur during processing of IP datagram, error reporting messages are used (These messages only report errors not correct errors)
- Error messages are send to the source
- Data section in Error reporting message contains IP header of original datagram plus first 8 bytes of data in the datagram.
- Following diagram shows the contents of data section in error reporting messages



ICMPv4 (contd..)

Types of Error Reporting Message:

1. **Destination Unreachable**
   - This error message is used to report when destination is not able to be reached by the IP datagram
   - type:3, code: 0 to 15 is used for this error message

2. **Source Quench**
   - This error message informs source that the network is congested and the datagram has been drooped. The source has to slow down sending more datagrams
   - type:4, code: 0 is used for this error message

3. **Redirection Message**
   - This error message is used when source uses wrong router to send out its message
   - The router redirects the message to the appropriate router but informs the source that it needs to change its default router in future
   - type:5, code: 0 to 3 is used for this error message

ICMPv4 (contd..)

Types of Error Reporting Message:

4. **Time exceeded**

- This error message is used when the TTL field in the datagram has the value 0
- When TTL field has the value 0, router simple drops the packet and sends this error message to the source
- This error message is also used when not all fragments of a datagram arrive within a predefined period of time
- type:11, code: 0 to 1 is used for this error message

5. **Parameter problem**

- This error message is used when there is a problem in the datagram header or some options are missing or cannot be interpreted
- type:12, code: 0 and 1 is used for this error message

ICMPv4 (contd..)

2. **Query Message:**

- Query messages are used to test the liveliness of hosts or routers, find round trip time between two devices, find clocks in two devices are synchronized

Types of Query message:

1. **Echo Request and Echo Reply:**

- This Query message is used by a source or router in order to test the liveliness of another host or router
- Source or router sends echo request to another host or router. If the latter is alive it responds with echo reply.
- Type 08(echo request) and 00(echo reply) with code 0 is used for this message

2. **Timestamp Request and Reply:**

- Used to find the round-trip time between two devices or to check whether the clocks in two devices are synchronized
- Type 13(timestamp request) and 14(timestamp reply) with code 0 is used for this message

CSC465 – Computer Networks

Dr. J. Harrison

These slides were produced almost entirely from material by Behrouz Forouzan for the text "TCP/IP Protocol Suite (2nd Edition)", McGraw Hill Publisher

Chapter 9

Internet Control Message Protocol (ICMP)

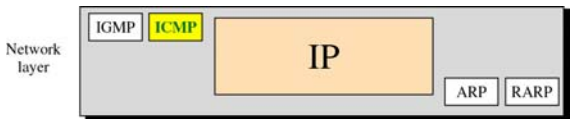
CONTENTS

ICMP

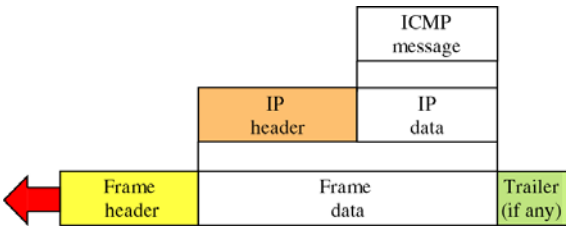
- IP unreliable, connectionless datagram delivery
  - Efficient use of network resources
  - Best effort service to send from source to destination
- No error control
  - What if router must discard datagram because it cannot find route to final destination?
  - What if final destination must discard all fragments because some don't arrive within time limit?
  - Error has occurred and IP Protocol has no built-in mechanism to notify the original host
- No method to obtain node information
  - Is router or host alive?

ICMP

ICMP addresses IP deficiencies



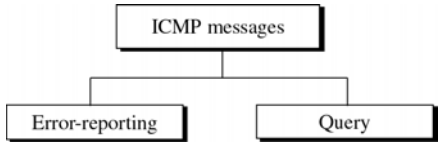
Encapsulation of ICMP packet



Value of protocol field in IP datagram is 1 to indicate data is ICMP

9.1 TYPES OF MESSAGES

ICMP messages

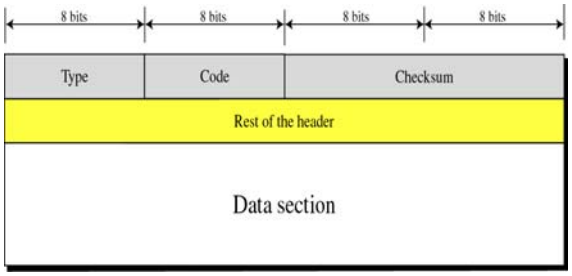


3 Dest Unreachable
4 Source Quench
11 Time Exceeded
12 Parameter Problem
5 Redirection

8 / 0 Echo Request/Reply
13 / 14 Timestamp Req/Reply
17 / 18 Address Mask
10 / 9 Router Solicitation

9.2 MESSAGE FORMAT

General format of ICMP messages



First 4 bytes common to all ICMP messages

Code specifies reason for particular message type

9.3 ERROR REPORTING

ICMP

- ICMP reports errors
- Higher protocols must correct them
- ICMP always reports error messages to the original source
- Source address within IP datagram

Error-reporting messages

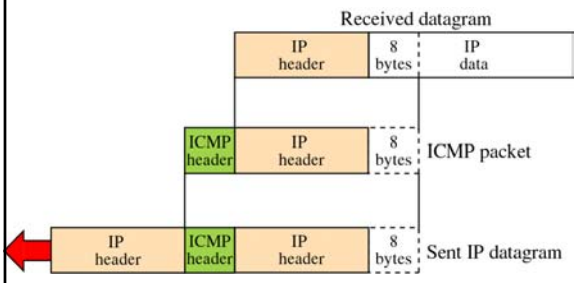
ICMP handles five (5) types of errors

- Destination unreachable
- Source Quench
- Time exceeded
- Parameter Problem
- Redirection

Important points about ICMP error messages:

1. ***No ICMP error message for a datagram carrying an ICMP error message.***
2. ***No ICMP error message for a fragmented datagram that is not the first fragment.***
3. ***No ICMP error message for a datagram having a multicast address.***
4. ***No ICMP error message for a datagram with a special address such as 127.0.0.0 or 0.0.0.0.***

Contents of data field for error messages



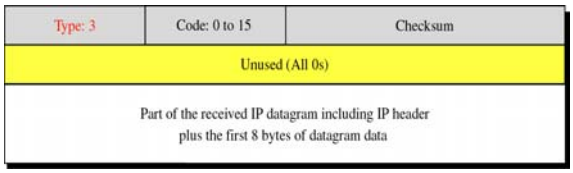
8 bytes UDP (TCP) header

(Source/Dest ports, length, checksum)

Destination-unreachable format

When a **router cannot route a datagram** the datagram is discarded and the router sends a "destination unreachable" message back to source host.

When a **host cannot deliver a datagram**, the datagram is discarded and the destination host sends a "destination unreachable" message back to source host.



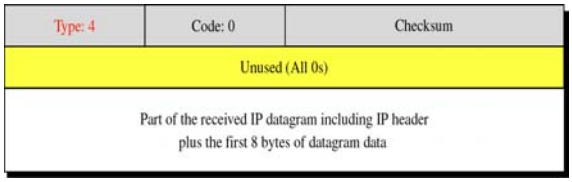
- 0: Net Unreachable
- 1: Host Unreachable
- 2: Protocol Unreachable
- 3: Port Unreachable
- 4: Frag Needed but "Don't Frag" was Set
- 5: Source Route Failed
- 6: Dest. Net Unknown
- 7: Dest. Host Unknown
- 8: Source Host Isolated
- 9: Communication with Dest Net is Admin Prohibited
- 10: Communication with Dest Host is Admin Prohibited
- 11: Dest Net Unreachable for Type of Service
- 12: Dest Host Unreachable for Type of Service
- 13: Communication Administratively Prohibited
- 14: Host Precedence Violation
- 15: Precedence cutoff in effect

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Some destination-unreachable messages can be created only by routers.

Source Quench

- IP offers no inherent support to guide flow control
- The source host does not know if the routers or dest host have been overwhelmed with datagrams
- Lack of flow control can create congestion in routers or destination host
  - Router forwarding buffers may overflow
  - Host processing buffers may overflow
- Source Quench messages in ICMP
- Routers or hosts that discard packets sends SQ
  - Informs source of discarding; warns source of speed

Source-quench format



Note

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.

The source must slow down the sending of datagrams until the congestion is relieved.

One source-quench message should be sent, from router or destination host for each datagram that is discarded due to congestion.

- There is no mechanism for telling source that congestion is relieved and transmission can resume at previous rate.
- Source continues to send at reduced rate.
- If transmission is many-to-one, the destination may drop packets from slower sending host but not those from faster (congestion causing) senders.

Time Exceeded Message

Sent in two cases:

Case 1:

Whenever a router receives a datagram with a time-to-live value of zero, it discards the datagram and sends a time-exceeded message to the original source.

Time Exceeded Message

Sent in two cases:

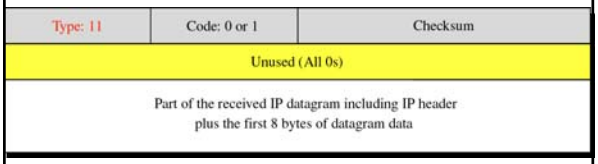
Case 2:

When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

Time Exceeded Message

In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.

Time-exceeded message format



Code 0: Time to live
Code 1: Fragmentation

Parameter-problem message format

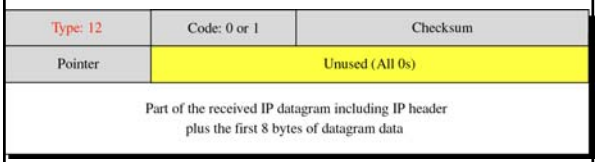
Error or ambiguity in one of the header fields (Code 0)

Required part of an IP option is missing

(Code 1)

A parameter-problem message can be created by a router or the destination host.

Parameter-problem message format



Code 0: Ptr field points to problem byte
Code 1: Ptr field unused

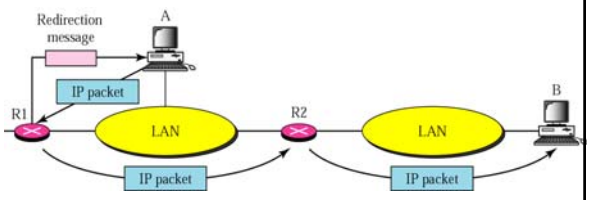
ICMP Redirection

- Router's routing tables updated dynamically using routing protocols
  - Hosts don't participate (for efficiency) since many more hosts than routers
- Hosts usually use static routing
  - Can result in misrouted datagram
  - In this case the recipient router forwards datagram to correct router
  - Sends ICMP "redirection" message to sending host to update its routing table

Note

A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message.

Redirection concept



A redirection message is sent from a router to a host on the same local network.

Redirection message format

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Code 0: Network specific
Code 1: Host specific
Code 2: Network specific (specified service)
Code 3: Host specific (specified service)

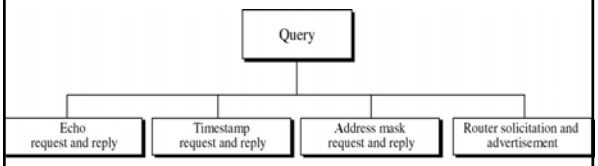
9.4

QUERY

Query messages

In addition to error detection, ICMP can also diagnose some network problems

Uses query/response system



Echo Request / Reply

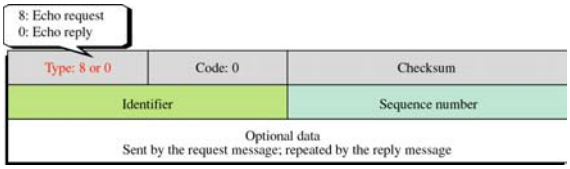
- Used by network managers and users for diagnosing network problems
- Tests if IP stack functioning on destination and routers in between
- Tests for the "reachability" of a host
- Used to implement the PING command
  - **Packet INternet Groper**

An echo-request message can be sent by a host or router.

An echo-reply message is sent by the host or router which receives an echo-request message

Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol..

Echo-request and echo-reply message format



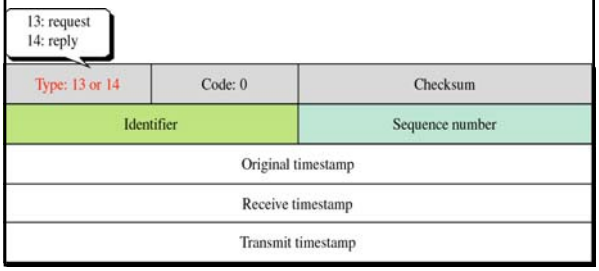
- Optional Data must be returned exactly as sent
- Identifier and Sequence # not formally defined
- Identifier often Process ID of sender
- Sequence # keeps track of particular request/reply

Timestamp Request/Reply

- Used by two machines to determine the roundtrip time for an IP datagram to travel between them
- Also used to synchronize the clocks in two machines
- Format contains three timestamps, each 32-bits
- Represents time (in milliseconds) from midnight in Universal Time (formerly GMT)

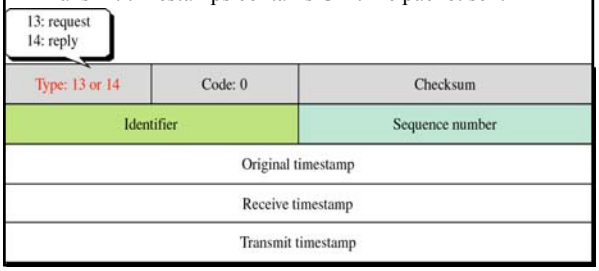
Timestamp-request format

- Original Timestamp receives Univeral Time shown by clock at departure time
- Receive/Transmit timestamps initialized to 0s



Timestamp-reply format

- Original Timestamp receives value copied from request
- Receive timestamp contains UT time dest received packet
- Transmit timestamps contains UT time packet sent



Sending time = value of receive timestamp – value of original timestamp

Receiving time = time the packet returned – value of transmit timestamp

Round-Trip Time = Sending time + Receiving time

The Round-Trip Time computation correct even if their clocks are not synchronized.

Mask-request and mask-reply message format

- Used by Host to obtain its IP address mask
- Host sends request to router if it knows IP of router
- If not, host broadcasts request and then router replies
- Diskless workstations use RARP to first get IP
- Then use ICMP Mask-request to get address mask

Router solicitation message format

Hosts need to know addresses of routers

Request broadcast by host to obtain the operating routers

Routers reply with all routers they are aware of including themselves (Sometimes reply without request)

Type: 10	Code: 0	Checksum
Identifier		Sequence number

Router advertisement message format

Preference level is used to select default router

If pref level is 0 then it is default. If level is 0x80000000 never selected as default

Type: 9	Code: 0	Checksum
Number of addresses	Address entry size	Lifetime
Router address 1		
Address preference 1		
Router address 2		
Address preference 2		
• • •		

9.5 CHECKSUM

Example of checksum calculation

8	0	0
1		9
TEST		

8 and 0	→	00001000 00000000
0	→	00000000 00000000
1	→	00000000 00000001
9	→	00000000 00001001
T & E	→	01010100 01000101
S & T	→	01010011 01010100
Sum	→	10101111 10100011
Checksum	→	01010000 01011100

9.6 ICMP PACKAGE

ICMP package