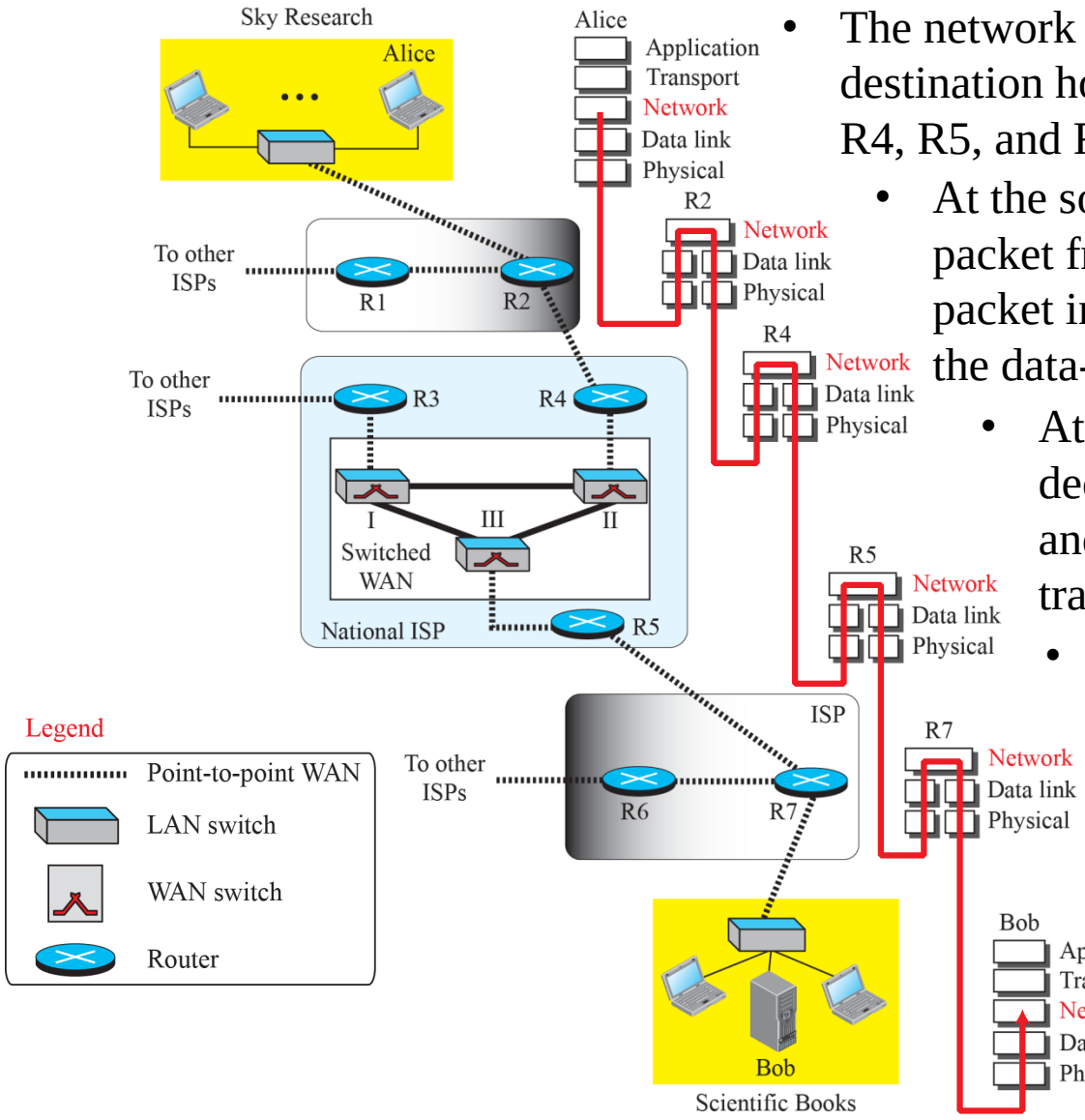
Chapter 18

Introduction to Network Layer

Network Layer Services



- The network layer is involved at the source host, destination host, and all routers in the path (R2, R4, R5, and R7).
  - At the source host, the network layer accepts a packet from a transport layer, encapsulates the packet in a datagram, and delivers the packet to the data-link layer.
  - At the destination host, the datagram is decapsulated, and the packet is extracted and delivered to the corresponding transport layer.
  - Although the source and destination hosts are involved in all five layers of the TCP/IP suite, the routers use three layers if they are routing packets only
  - A router in the path is normally shown with two data-link layers and two physical layers, because it receives a packet from one network and delivers it to another network.

Packetizing

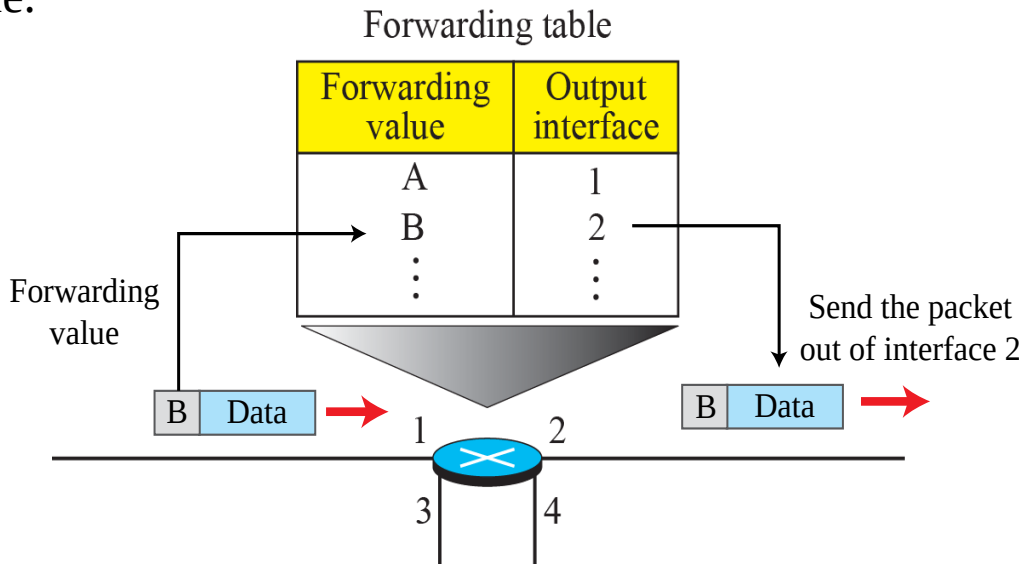
- The first duty of the network layer is **packetizing:** encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- Duty of the network layer is to carry payload from source to destination w/o changing it or using it.
- The source host receives the payload from an upper-layer protocol, adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol and delivers the packet to the data-link layer.
- The source is not allowed to change the content of the payload unless it is too large for delivery and needs to be fragmented.
- The destination host receives the network-layer packet from its data-link layer, decapsulates the packet, and delivers the payload to the corresponding upper-layer protocol.
- If the packet is fragmented at source or at routers along the path, network layer is responsible for waiting until all fragments arrive, reassembling them, and delivering them to the upper-layer
- The routers in the path are not allowed to decapsulate the packets they received unless the packets need to be fragmented.
- The routers are not allowed to change source and destination addresses either.
- They just inspect the addresses for the purpose of forwarding the packet to the next network on the path.
- However, if a packet is fragmented, the header needs to be copied to all fragments and some changes are needed.

Routing

- The network layer is responsible for routing the packet from its source to the destination.
- A physical network is a combination of networks (LANs and WANs) and routers that connect them.
- This means that there is more than one route from the source to the destination.
- The network layer is responsible for finding the best one among these possible routes.
- The network layer needs to have some specific strategies for defining the best route.
- In the Internet today, this is done by running some routing protocols to help the routers coordinate their knowledge about the neighborhood and to come up with consistent tables to be used when a packet arrives.
- The routing protocols, should be run before any communication occurs.

Forwarding

- If routing is applying strategies and running some routing protocols to create the decision-making tables for each router, *forwarding* can be defined as the action applied by each router when a packet arrives at one of its interfaces.
- The decision-making table a router normally uses for applying this action is sometimes called the ***forwarding table*** and sometimes the ***routing table***.
- When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (in unicast routing) or to some attached networks (in multicast routing).
- To make this decision, the router uses a piece of information in the packet header, which can be the destination address or a label, to find the corresponding output interface number in the forwarding table.



Other Services

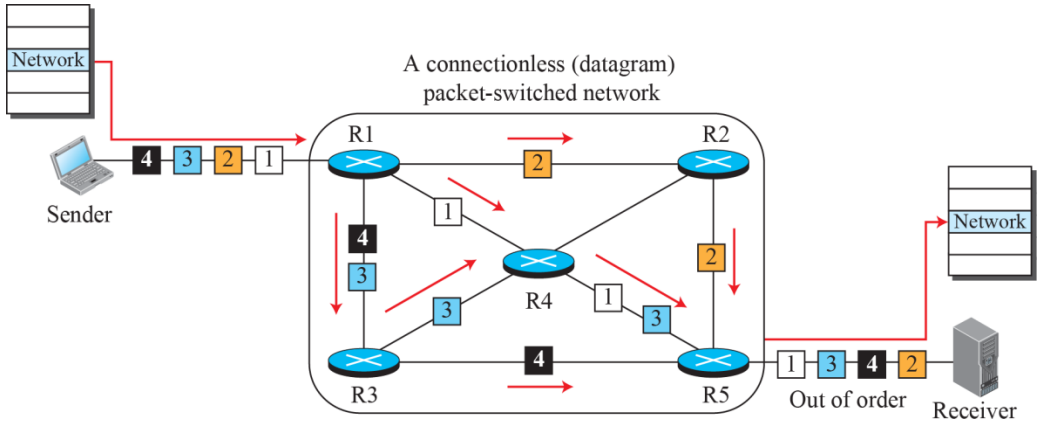
- Error Control
- Flow Control
- Congestion Control
- Quality of Service
- Security

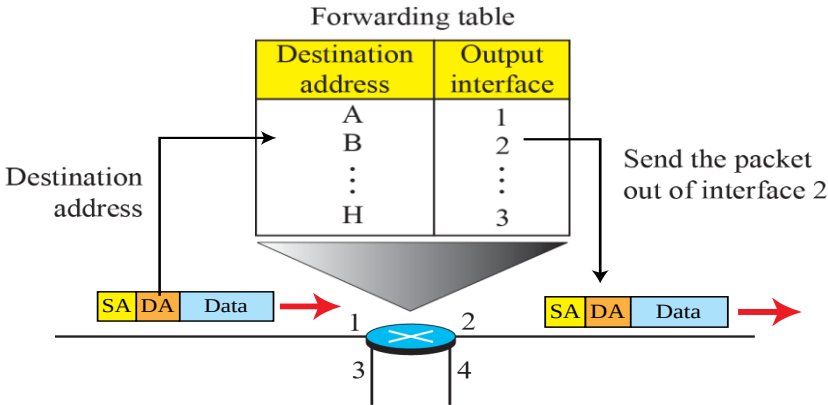
Packet Switching

- Although in data communication switching techniques are divided into two broad categories, circuit switching and packet switching, only packet switching is used at the network layer because the unit of data at this layer is a packet.
- Circuit switching is mostly used at the physical layer
- At the network layer, a message from the upper layer is divided into manageable packets and each packet is sent through the network.
- The source of the message sends the packets one by one; the destination of the message receives the packets one by one.
- The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer.
- The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.
- Today, a packet-switched network can use two different approaches to route the packets:
    - ***Datagram Approach***
    - ***Virtual Circuit Approach***

Datagram Approach: Connectionless Service

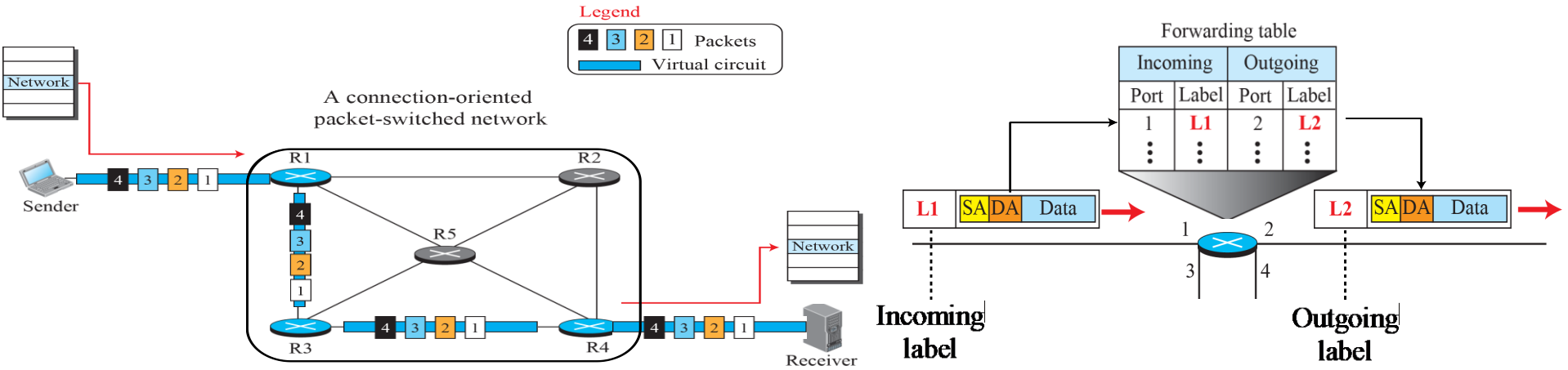
- The network layer was designed to provide a ***connectionless*** service in which the network-layer protocol treats each packet ***independently***, with each packet having no relationship to any other packet
- The idea was that the network layer is only responsible for delivery of packets from the source to the destination.
- In this approach, the packets in a message may or may not travel the same path to their destination.
- When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message.
- Each packet is routed based on the information contained in its header: source and destination addresses.
- The router in this case routes the packet based only on the destination address.



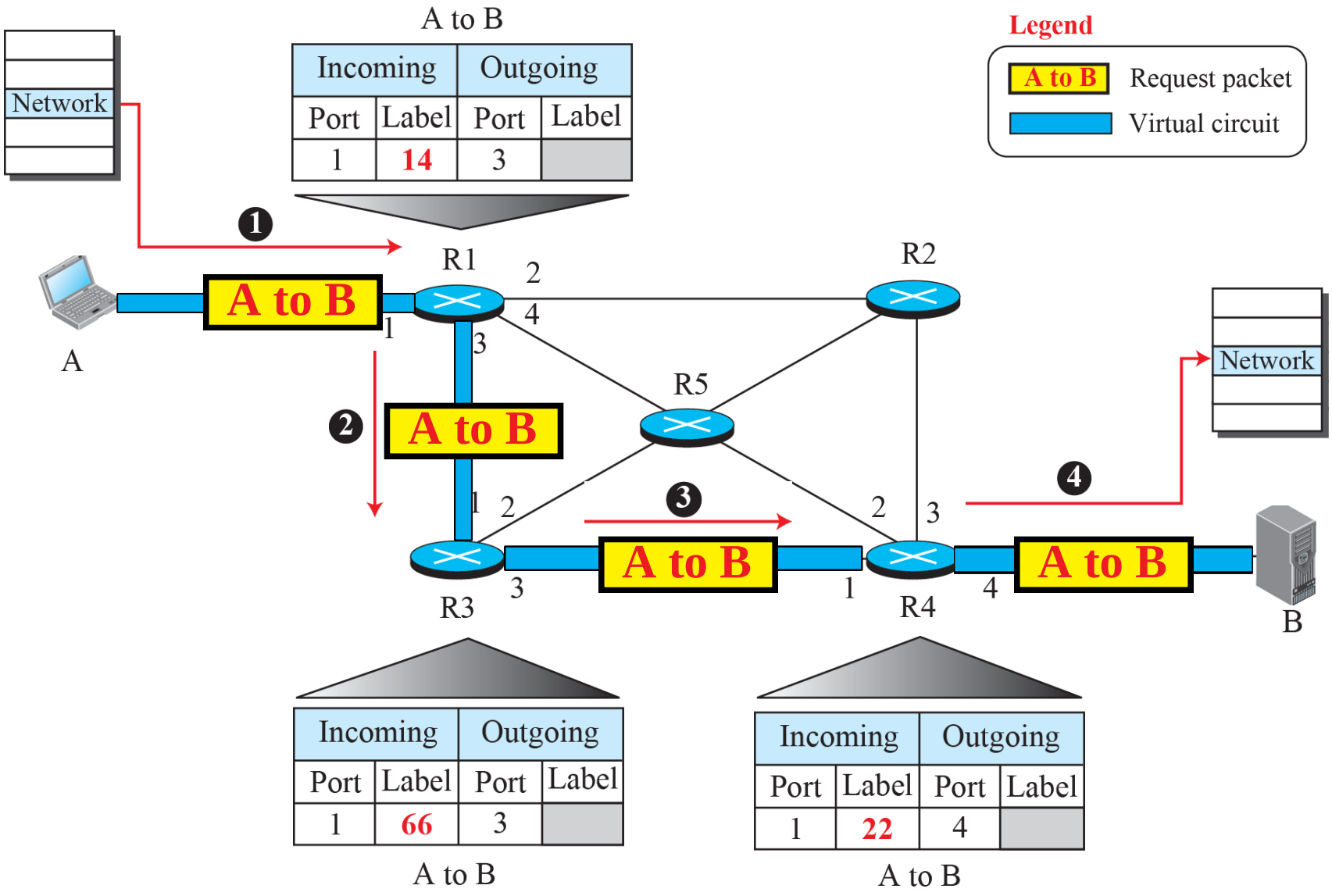


Virtual-Circuit Approach: Connection-Oriented Service

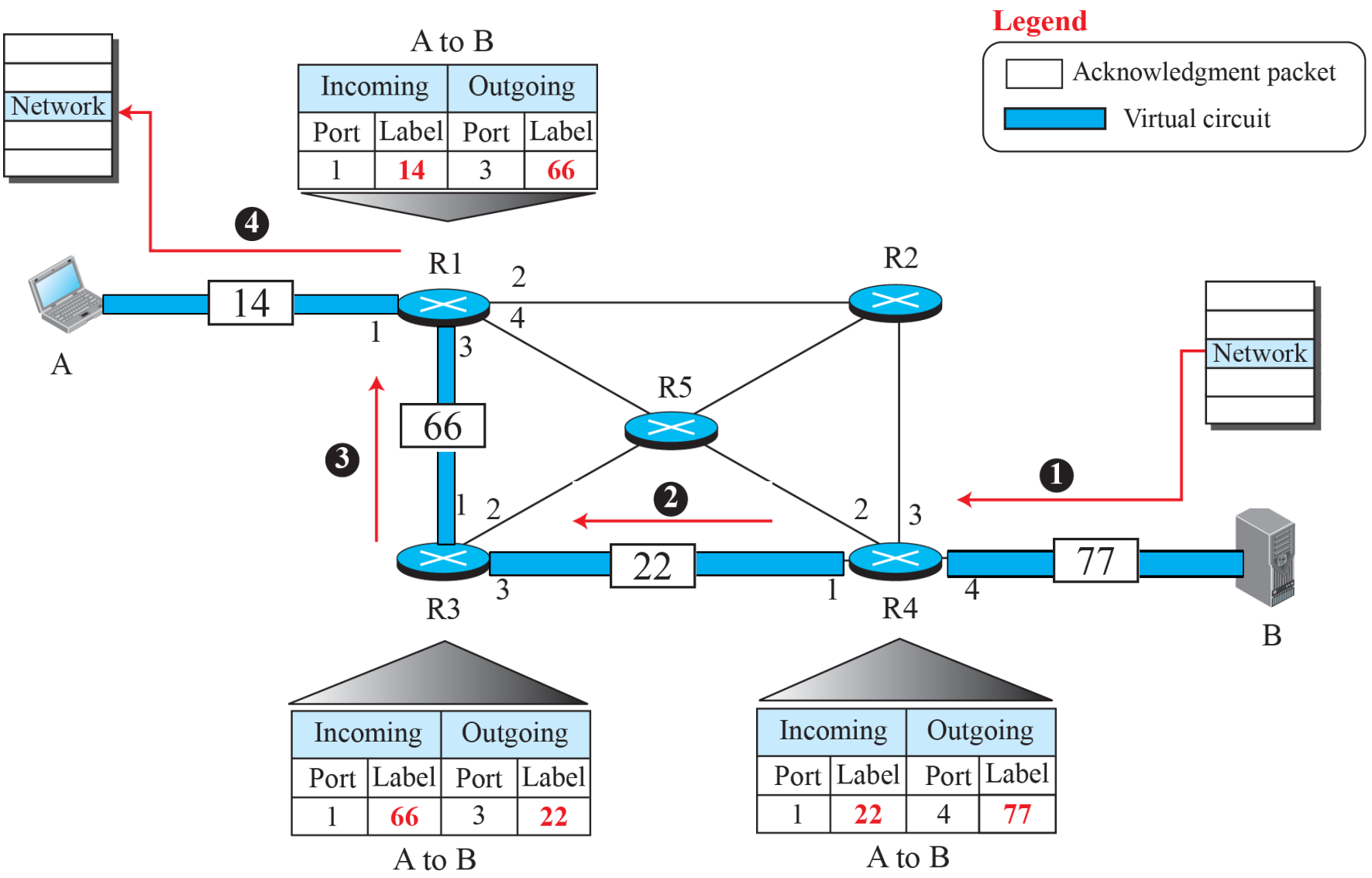
- In a connection-oriented service (also called *virtual-circuit approach*), there is a relationship between all packets belonging to a message.
- Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams.
- After connection setup, the datagrams can all follow the same path.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.
- Each packet is forwarded based on the label in the packet.
- To follow the idea of connection-oriented design to be used in the Internet, we assume that the packet has a label when it reaches the router.
- To create a connection-oriented service, a three-phase process is used: **setup**, **data transfer**, and **teardown**



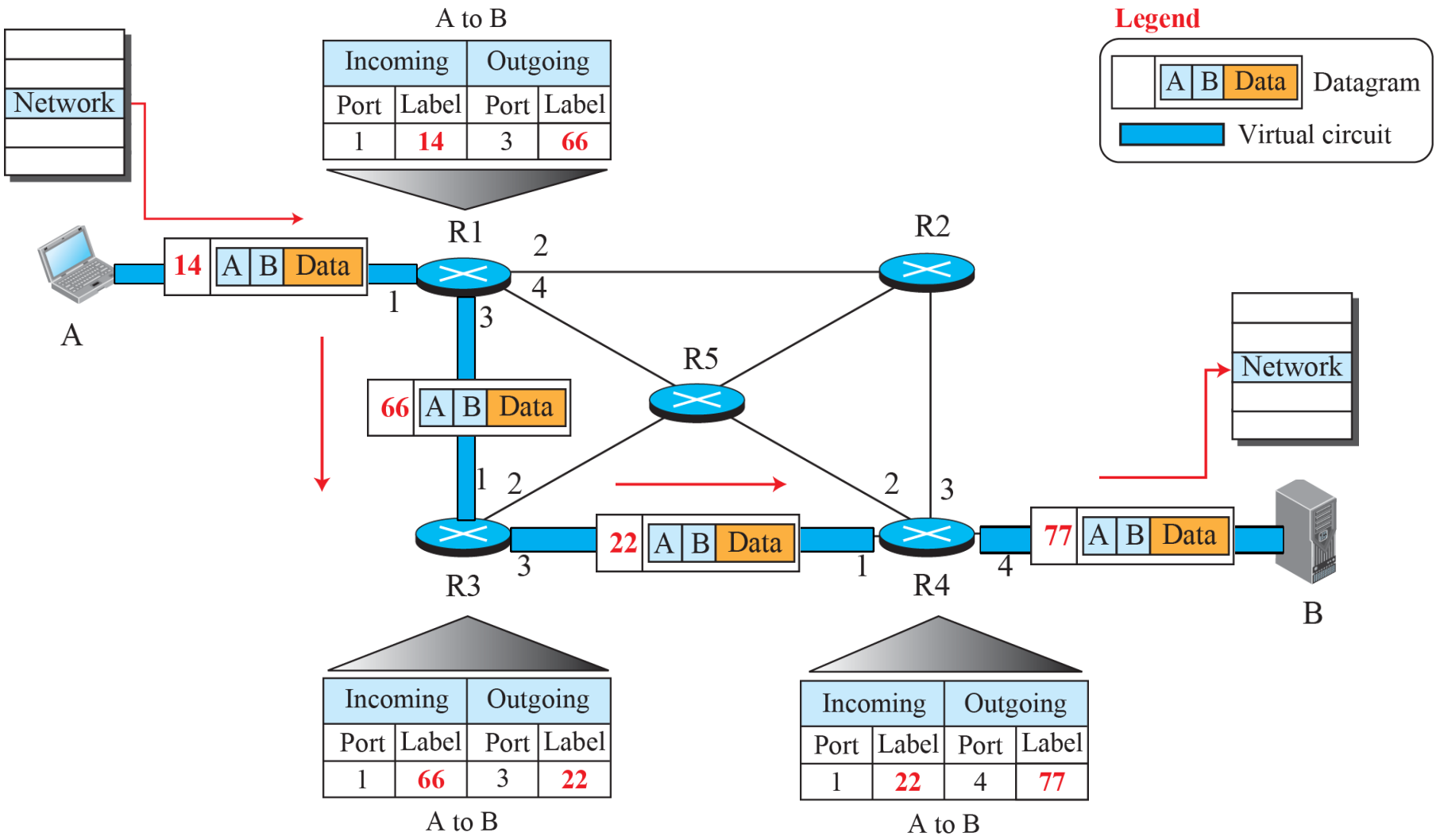
Setup Phase: Sending **request packet** in a virtual-circuit network



Setup Phase: Sending **acknowledgments** in a virtual-circuit network



Data-Transfer Phase: Flow of one packet in an established virtual circuit



Teardown Phase:

- In the teardown phase, source A, after sending all packets to B, sends a special packet called a teardown packet.
- Destination B responds with a confirmation packet.
- All routers delete the corresponding entries from their tables.

Network-Layer Performance

- The performance of a network can be measured in terms of :
  - Delay
  - Throughput
  - Packet loss
- Congestion control is an issue that can improve the performance.

Delay

- All of us expect instantaneous response from a network, but a packet, from its source to its destination, encounters delays.
- The delays in a network can be divided into four types:
    - Transmission delay
    - Propagation delay
    - Processing delay
    - Queuing delay

Transmission Delay

- A source host or a router cannot send a packet instantaneously.
- A sender needs to put the bits in a packet on the line one by one.
- If the first bit of the packet is put on the line at time t1 and the last bit is put on the line at time t2, transmission delay of the packet is (t2 − t1)
- Definitely, the transmission delay is longer for a longer packet and shorter if the sender can transmit faster. In other words, the transmission delay is:

$$\textbf{Delay}_{\textbf{tr}} = \textbf{(Packet length) / (Transmission rate).}$$

- For example, in a Fast Ethernet LAN with the transmission rate of 100 million bits per second (100 Mbps) and a packet of 10,000 bits, it takes (10,000)/(100,000,000) = 100 microseconds for all bits of the packet to be put on the line.

Propagation Delay

- Propagation delay is the time it takes for a bit to travel from point A to point B in the transmission media.
- The propagation delay for a packet-switched network depends on the propagation delay of each network (LAN or WAN).
- The propagation delay depends on the propagation speed of the media, which is 3 X 10^8 meters/second in a vacuum and normally much less in a wired medium; it also depends on the distance of the link.
- In other words, propagation delay is:

$$\textbf{Delay}_{\textbf{pg}} = \textbf{(Distance) / (Propagation speed).}$$

- For example, if the distance of a cable link in a point-to-point WAN is 2000 meters and the propagation speed of the bits in the cable is 2 X10^8 meters/second, then the propagation delay is 10 microseconds.

Processing Delay

- The processing delay is the time required for a router or a destination host to receive packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port (in the case of a router) or deliver the packet to the upper-layer protocol (in the case of the destination host).
- The processing delay may be different for each packet, but normally is calculated as an average:

Delay_{pr} = Time required to process a packet in a router or a destination host

Queuing Delay

- Queuing delay can normally happen in a router.
- A router has an input queue connected to each of its input ports to store packets waiting to be processed; the router also has an output queue connected to each of its output ports to store packets waiting to be transmitted.
- The queuing delay for a packet in a router is measured as the time a packet waits in the input queue and output queue of a router.
- We can compare the situation with a busy airport. Some planes may need to wait to get the landing band (input delay); some planes may need to wait to get the departure band (output delay).

Delay_{qu} = The time a packet waits in input and output queues in a router

Total Delay

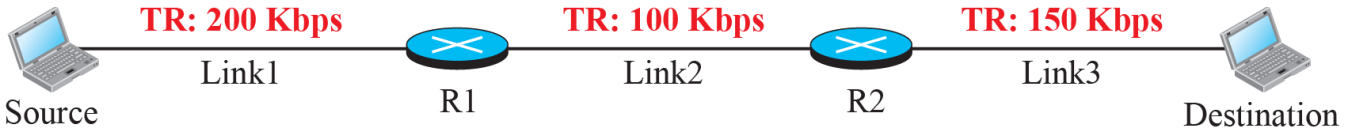
- Assuming equal delays for the sender, routers, and receiver, the total delay (source-to-destination delay) a packet encounters can be calculated if we know the number of routers, n, in the whole path.

$$\textbf{Total delay} = (n + 1)\,(\textbf{Delay}_{tr} + \textbf{Delay}_{pg} + \textbf{Delay}_{pr}) + (n)\,(\textbf{Delay}_{qu})$$

- Note that if we have n routers, we have (n + 1) links.
- Therefore, we have (n + 1) transmission delays related to n routers and the source, (n + 1) propagation delays related to (n + 1) links, (n + 1) processing delays related to n routers and the destination, and only n queuing delays related to n routers.

Throughput

- Throughput at any point in a network is defined as the number of bits passing through the point in a second, which is actually the transmission rate of data at that point.
- In a path from source to destination, a packet may pass through several links (networks), each with a different transmission rate.

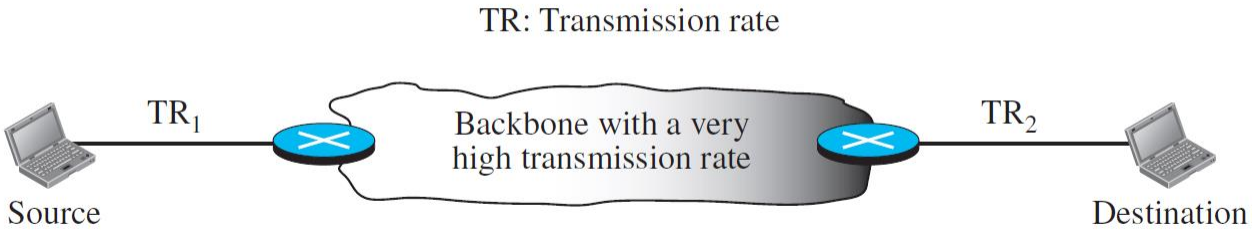


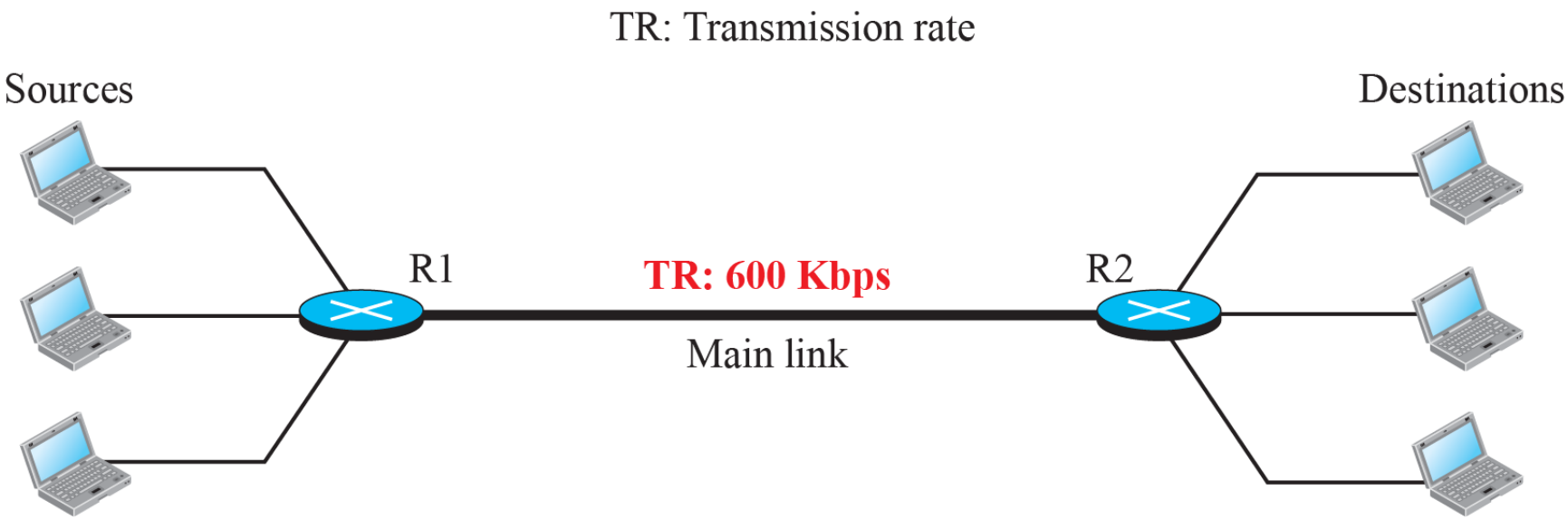
a. A path through three links

b. Simulation using pipes

$$\text{Throughput} = \text{minimum}\{TR_1, TR_2, \ldots TR_n\}.$$

Throughput


TR: Transmission rate
TR1
Backbone with a very
high transmission rate
TR2
Source
Destination


TR: Transmission rate
Sources
Destinations
R1
TR: 600 Kbps
R2
Main link

Packet Loss

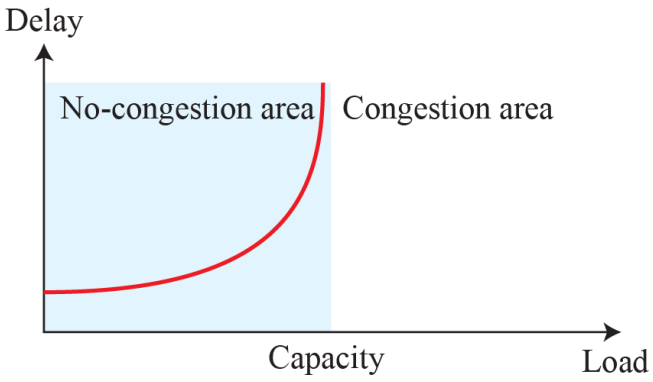
- Another issue that severely affects the performance of communication is the number of packets lost during transmission.
- When a router receives a packet while processing another packet, the received packet needs to be stored in the input buffer waiting for its turn.
- A router has an input buffer with a limited size. A time may come when the buffer is full and the next packet needs to be dropped.
- The effect of packet loss on the Internet network layer is that the packet needs to be resent, which in turn may create overflow and cause more packet loss.
- Packet loss and re-transmission leads to congestion.

Congestion

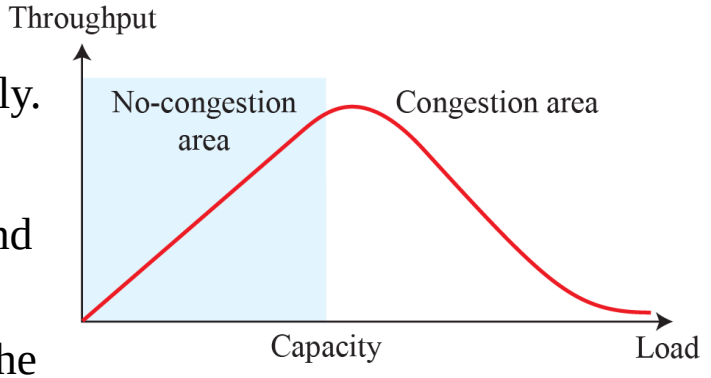
- Congestion is usually handled at the transport layer.
- Although congestion at the network layer is not explicitly addressed in the Internet model, the study of congestion at this layer may help us to better understand the cause of congestion at the transport layer and find possible remedies to be used at the network layer.
- Congestion at the network layer is related to two issues:
    - Throughput
    - Delay

Delay and Throughput as Functions of Load

- When the load is much less than the capacity of the network, the *delay* is at a minimum.
- This minimum delay is composed of propagation delay and processing delay, both of which are negligible.
- However, when the load reaches the network capacity, the delay increases sharply because we now need to add the queuing delay to the total delay.
- Note that the delay becomes infinite when the load is greater than the capacity.
- When the load is below the capacity of the network, the throughput increases proportionally with the *load*.
- We expect the throughput to remain constant after the load reaches the capacity, but instead the throughput declines sharply.
- The reason is the discarding of packets by the routers.
- When the load exceeds the capacity, the queues become full and the routers have to discard some packets.
- Discarding packets does not reduce the number of packets in the network because the sources retransmit the packets, using time-out mechanisms, when the packets do not reach the destinations.



a. Delay as a function of load



b. Throughput as a function of load

Congestion Control

- Congestion control refers to techniques and mechanisms that can either prevent congestion before it happens or remove congestion after it has happened.

- In general, we can divide congestion control mechanisms into two broad categories:

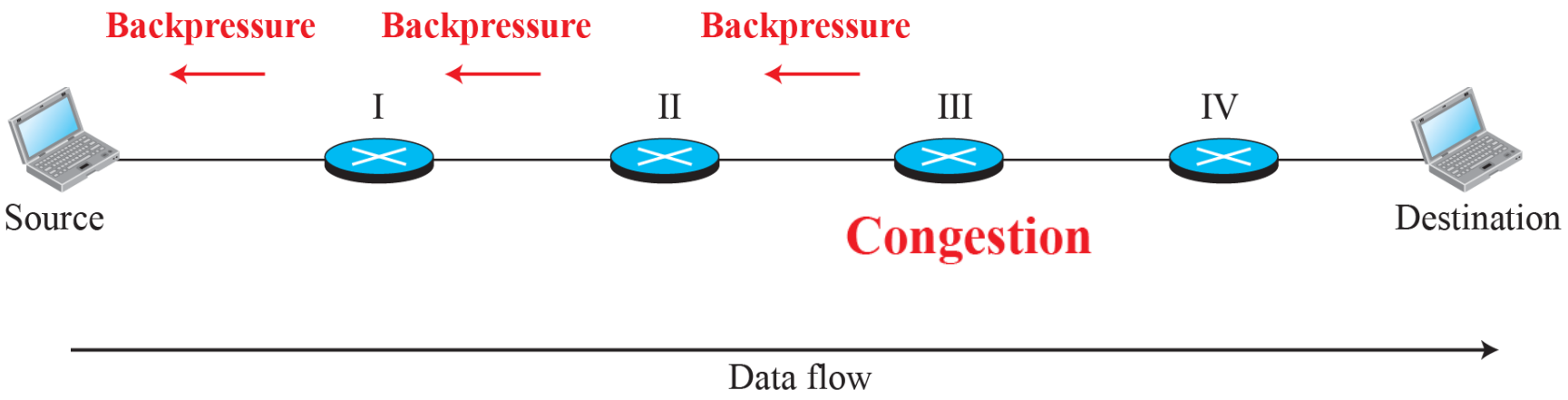
    - **Open-loop Congestion Control** (Prevention)
    - **Closed-loop Congestion Control** (Removal)

Open-Loop Congestion Control

- In open-loop congestion control, policies are applied to prevent congestion before it happens.
- In these mechanisms, congestion control is handled by either the source or the destination.
- List of policies that can prevent congestion:
- **Retransmission Policy**: The retransmission policy and the retransmission timers must be designed to optimize efficiency and prevent congestion.
- **Window Policy:** Selective Repeat window is better than the Go-Back-N window for congestion control.
- **Acknowledgment Policy:** A receiver may decide to acknowledge only N packets at a time. As acknowledgments are also part of the load in a network, sending fewer acknowledgments impose less load on the network
- **Discarding Policy:** If the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented
- **Admission Policy:** A router can deny establishing a virtual-circuit connection if there is congestion in the network or if there is a possibility of future congestion

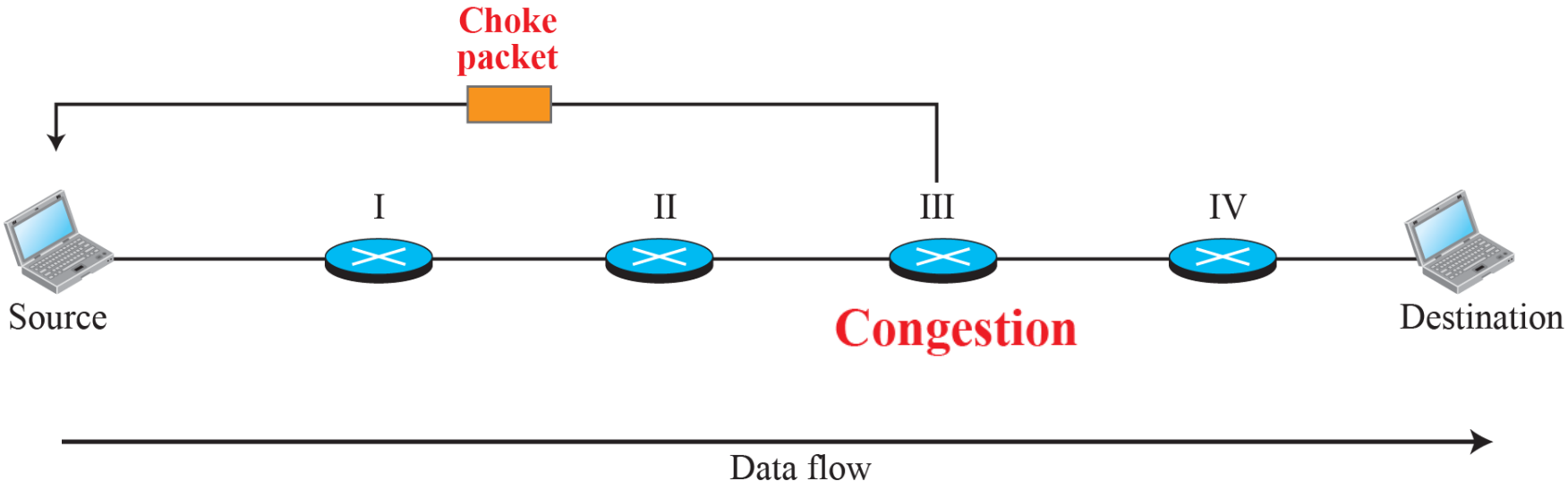
Closed-Loop Congestion Control

- Closed-loop congestion control mechanisms try to alleviate congestion after it happens.
- Several mechanisms have been used by different protocols.
- List of policies that can alleviate congestion:
- **Backpressure**: Backpressure is a node- to- node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is coming.



Closed-Loop Congestion Control (cont.)

- **Choke Packet:** A choke packet is a packet sent by a node to the source to inform it of congestion. In the choke-packet method, the warning is from the router, which has encountered congestion, directly to the source station. The intermediate nodes through which the packet has traveled are not warned



- **Implicit Signaling:** In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms.
- **Explicit Signaling:** The node that experiences congestion can explicitly send a signal to the source or destination. The explicit-signaling method, however, is different from the choke-packet method.

IPv4 Addresses

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address
- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet
- The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.
- If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

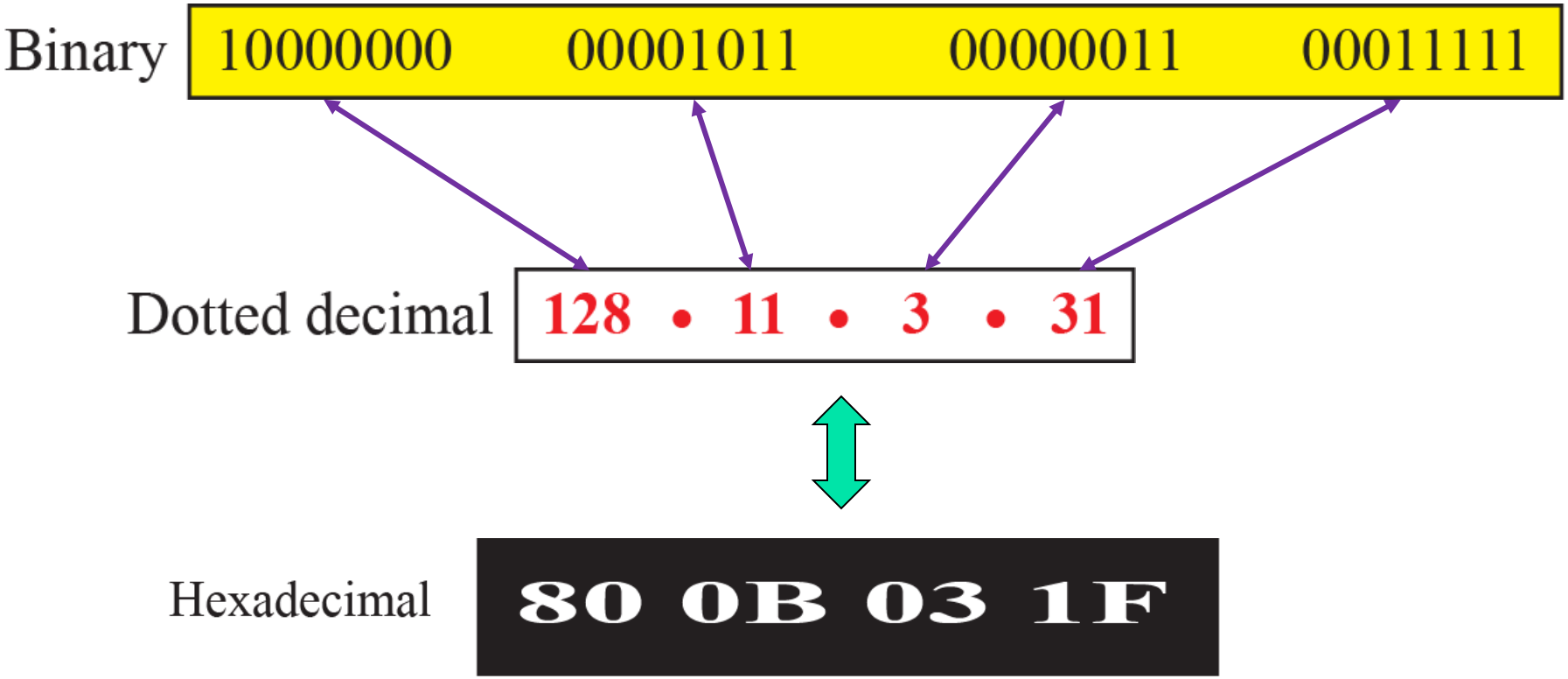
- A protocol like IPv4 that defines addresses has an address space.
- An address space is the total number of addresses used by the protocol.
- If a protocol uses b bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).
- If there were no restrictions, more than 4 billion devices could be connected to the Internet.

Notation

- There are three common notations to show an IPv4 address:

- **Binary Notation (base 2)**
  - In *binary notation*, an IPv4 address is displayed as 32 bits.
  - To make the address more readable, one or more spaces are usually inserted between each octet (8 bits / 1 byte).
- **Dotted-decimal Notation (base 256)**
  - To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes.
  - This format is referred to as *dotted-decimal notation*.
  - Note that because each byte (octet) is only 8 bits, each number in the dotted-decimal notation is between 0 and 255.
- **Hexadecimal Notation (base 16)**
  - We sometimes see an IPv4 address in hexadecimal notation.
  - Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.
  - This notation is often used in network programming.

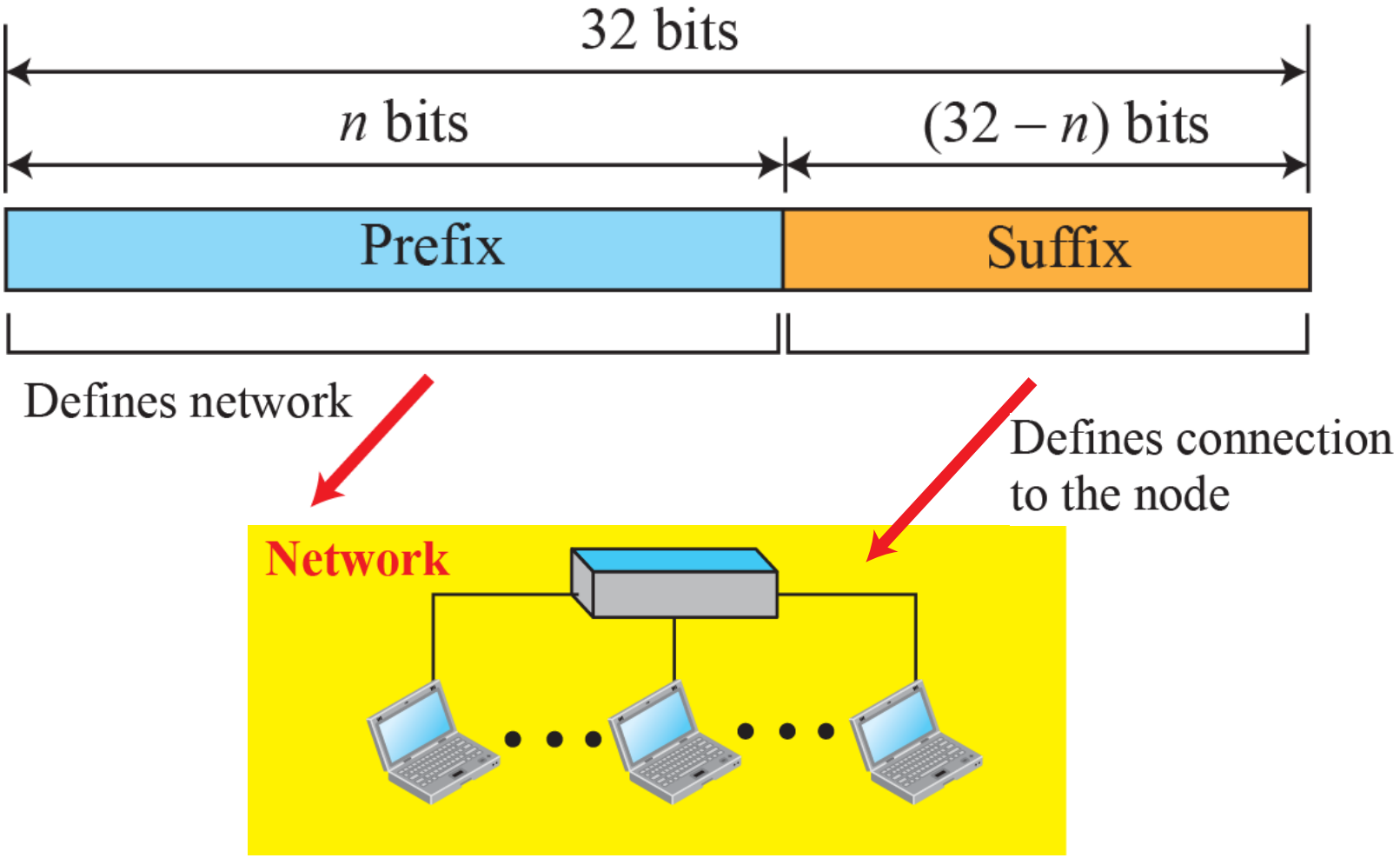
Three different notations in IPv4 addressing



Hierarchy in Addressing

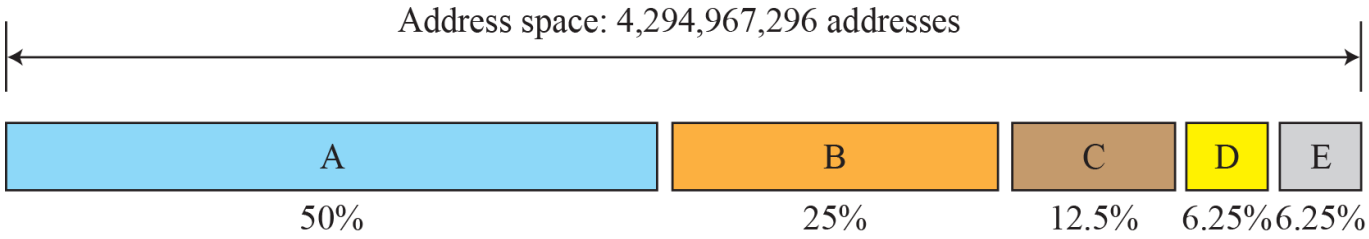
- In any communication network that involves delivery, such as a telephone network or a postal network, the addressing system is hierarchical.
- In a postal network, the postal address (mailing address) includes the country, state, city, street, house number, and the name of the mail recipient.
- Similarly, a telephone number is divided into the country code, area code, local exchange, and the connection.
- A 32-bit IPv4 address is also hierarchical, but divided only into two parts. The first part of the address, called the prefix (net id), defines the network; the second part of the address, called the suffix (host id), defines the node (connection of a device to the Internet).
- The prefix length is n bits and the suffix length is (32 − n) bits.
- A prefix can be fixed length or variable length.
- The network identifier in the IPv4 was first designed as a fixed-length prefix.
- This scheme, which is now obsolete, is referred to as classful addressing.
- The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.

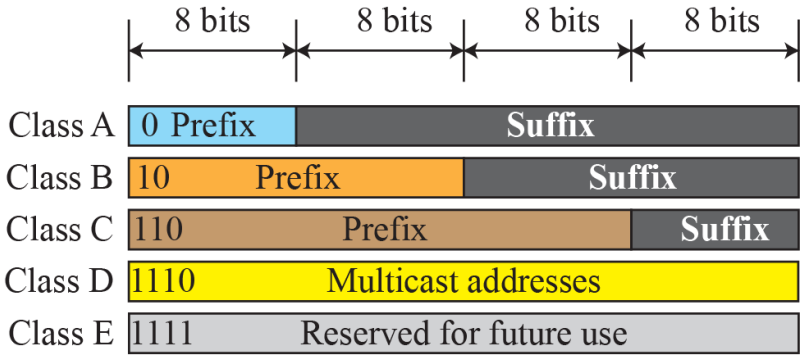
Hierarchy in addressing



Classful Addressing

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





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

Summary of Classful Addressing

Summary of Classful addressing :

CLASS	LEADING BITS	NET ID BITS	HOST ID BITS	NO. OF NETWORKS	ADDRESSES PER NETWORK	START ADDRESS	END ADDRESS
CLASS A	0	8	24	2^7 (128)	2^{24} (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	2^{14} (16,384)	2^{16} (65,536)	128.0.0.0	191.255.255.255
CLASS C	110	24	8	2^{21} (2,097,152)	2^8 (256)	192.0.0.0	223.255.255.255
CLASS D	1110	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	224.0.0.0	239.255.255.255
CLASS E	1111	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	240.0.0.0	255.255.255.255

Address Depletion

- The reason that classful addressing has become obsolete is address depletion.
- Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.
- To understand the problem, let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network (seen by the rest of the world) with 16,777,216 nodes (computers in this single network).
- Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

Subnetting and Supernetting

- To alleviate address depletion, two strategies were proposed and, to some extent, implemented: subnetting and supernetting.
- In subnetting, a class A or class B block is divided into several subnets.
- Each subnet has a larger prefix length than the original network.
    - For example, if a network in class A is divided into four subnets, each subnet has a prefix of $n_{sub} = 10$. At the same time, if all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.
- This idea did not work because most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.
- While subnetting was devised to divide a large block into smaller ones, supernetting was devised to combine several class C blocks into a larger block to be attractive to organizations that need more than the 256 addresses available in a class C block.
- This idea did not work either because it makes the routing of packets more difficult.

Advantage of Classful Addressing

- Although classful addressing had several problems and became obsolete, it had one advantage:
    - Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately.
    - In other words, the prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

Solutions for Address Depletion Problem

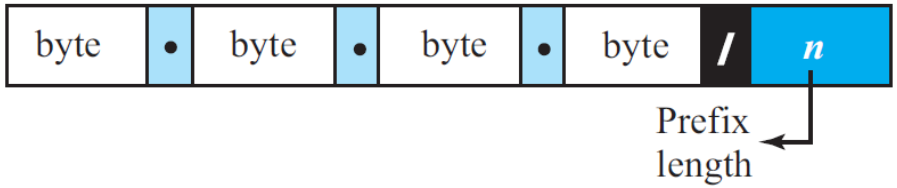
- With the growth of the Internet, classful addressing policy led to the unnecessary rapid depletion of IPv4 addresses.
- The solutions to address depletion:
    - **IPv6 = 128 bit address space. (long term solution)**
    - **CIDR = Classless Inter Domain Routing (immediate solution)**
    - **Private addresses and NAT (immediate solution)**

Classless Addressing

- In Classless addressing the class privilege was removed from the distribution to compensate for the address depletion
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses and so on.
- In classless addressing, the whole address space is divided into variable length blocks.
- The prefix in an address defines the block (network); the suffix defines the node (device).
- Theoretically, we can have a block of 2^0, 2^1, 2^2, . . . , 2^{32} addresses.
- One of the restrictions is that the number of addresses in a block needs to be a power of 2.
- An organization can be granted one block of addresses
- Unlike classful addressing, the prefix length in classless addressing is variable.
- We can have a prefix length that ranges from 0 to 32.
- The size of the network is inversely proportional to the length of the prefix.
- A small prefix means a larger network; a large prefix means a smaller network.

Prefix Length: Slash Notation

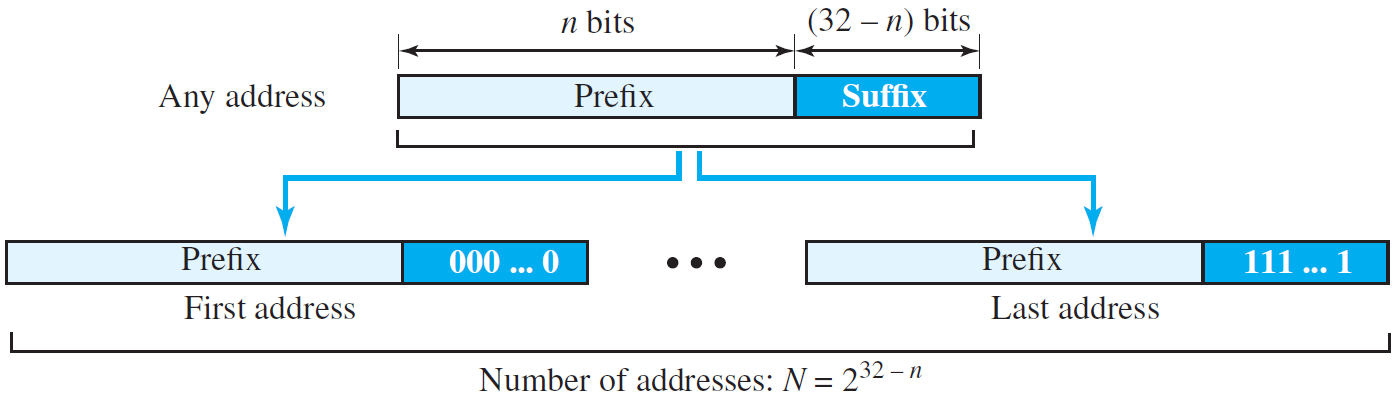
- The first question that we need to answer in classless addressing is how to find the prefix length if an address is given.
- Since the prefix length is not inherent in the address, we need to separately give the length of the prefix.
- In this case, the prefix length, n, is added to the address, separated by a slash.
- The notation is informally referred to as slash notation and formally as classless inter-domain routing or CIDR strategy.
- An address in classless addressing does not, per se, define the block or network to which the address belongs; we need to give the prefix length also



Examples:
12.24.76.8/**8**
23.14.67.92/**12**
220.8.24.255/**25**

Extracting Information from an Address

- Given any address in the block, we normally like to know three pieces of information about the block to which the address belongs: the number of addresses, the first address in the block, and the last address.
- Since the value of prefix length, n, is given, we can easily find these three pieces of information
- The number of addresses in the block is found as $N = 2^{32-n}$.
- To find the first address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 0s.
- To find the last address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 1s.



Example 18.1

A classless address is given as 167.199.170.82/27.
Find: Number of addresses in the network, first and last addresses.

The number of addresses in the network is $2^{32-n} = 2^{32-27} = 2^5 = 32$ addresses.

Address: 167.199.170.82/**27**	10100111	11000111	10101010	01010010
First address: 167.199.170.64/**27**	10100111	11000111	10101010	01000000
Last address: 167.199.170.95/**27**	10100111	11000111	10101010	01011111

Address Mask

- Another way to find the first and last addresses in the block is to use the address mask.
- The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 − n) are set to 0s.
- A computer can easily find the address mask because it is the complement of $(2^{32-n} - 1)$
- The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR.
- 1. The number of addresses in the block N = NOT (mask) + 1.
- 2. The first address in the block = (Any address in the block) AND (mask).
- 3. The last address in the block = (Any address in the block) OR [(NOT (mask)].

Example 18.2

A classless address is given as 167.199.170.82/27.
Find: Number of addresses in the network, first and last addresses.
IP address: 10100111 11000111 10101010 01010010
Musk : 11111111 11111111 11111111 11100000

Number of Address in the block: NOT (11111111111111111111111111100000) + 1
= 00000000000000000000000000011111 + 1 = 31 + 1 = 32

First Address:
10100111 11000111 10101010 01010010 (IP ADDRESS)
11111111 11111111 11111111 11100000 (MASK)

10100111 11000111 10101010 01000000 = 167.199.170. 64 (AND)

Last Address:
10100111 11000111 10101010 01010010 (IP ADDRESS)
00000000 00000000 00000000 00011111 (NOT MASK)

10100111 11000111 10101010 01011111 = 167.199.170. 95 (OR)

Example 18.3

In classless addressing, an address cannot per se define the block the address belongs to. For example, the address 230.8.24.56 can belong to many blocks.

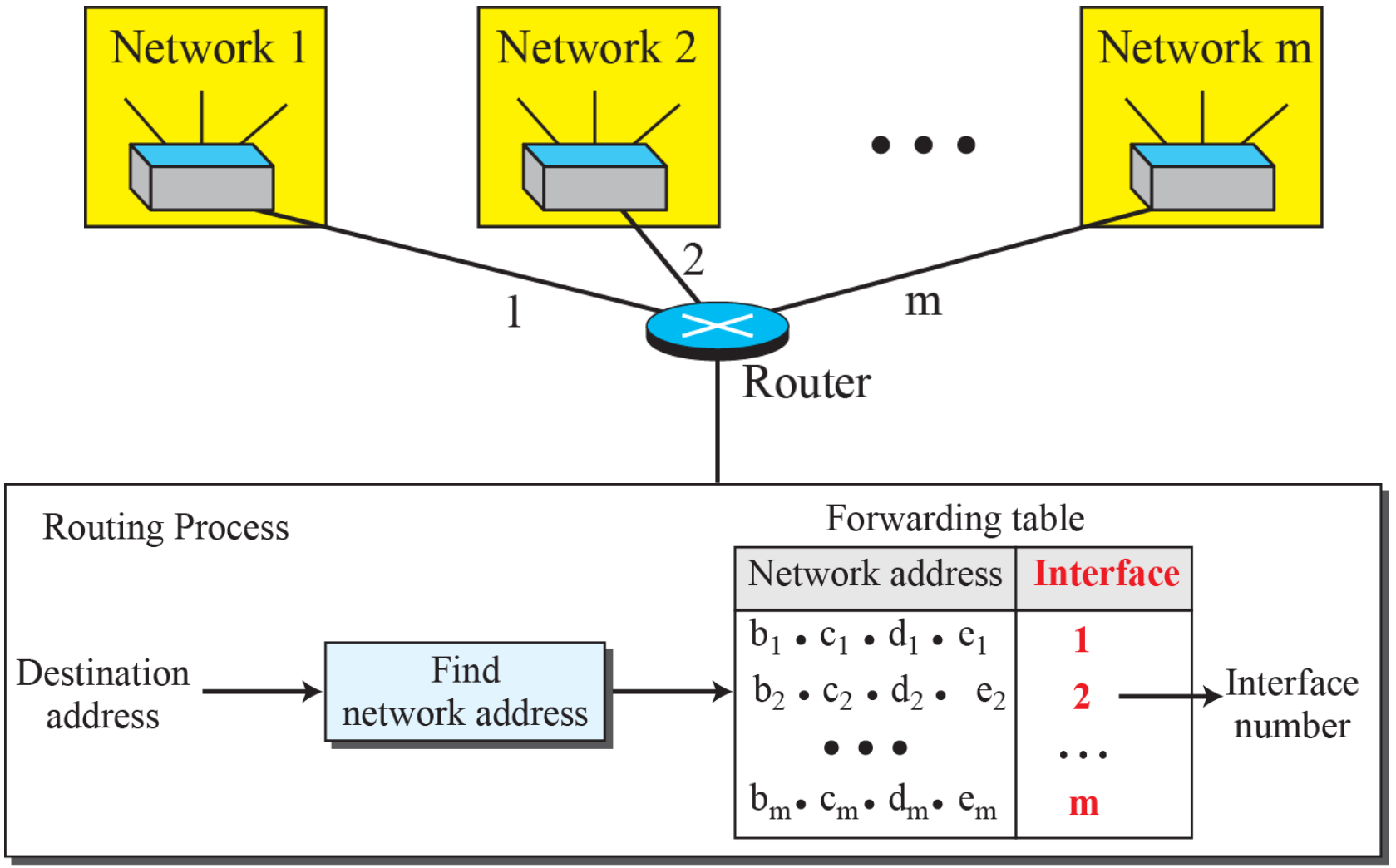
Some of them are shown below with the value of the prefix associated with that block:

Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

Network Address

- Given any address and the mask we can find all information about the block.
- The first address, which is also called the **network address**, is particularly important because it is used in routing a packet to its destination network.
- For the moment, let us assume that an internet is made of *m* networks and a router with *m* interfaces.
- When a packet arrives at the router from any source host, the router needs to know to which network the packet should be sent: from which interface the packet should be sent out.
- After the network address has been found, the router consults its forwarding table to find the corresponding interface from which the packet should be sent out.
- The network address is actually the identifier of the network; each network is identified by its network address.

Network address



Block Allocation

- The next issue in classless addressing is block allocation.
- The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN).
- However, ICANN does not normally allocate addresses to individual Internet users.
- It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case).
- For the proper operation of the CIDR, two restrictions need to be applied to the allocated block.:
    - **The number of addresses in a block must be a power of 2**
      For example: 2, 4, 8, 16,….,256,…, 1024,….. so on
    - **The first address must be evenly divisible by the number of addresses.**
      For example: if a block contains 4 addresses, the first address must be divisible by 4

Example 18.4

An ISP has requested a block of 1000 addresses.

How many actual addresses are granted?
What is the prefix length?

An ISP has requested a block of 1000 addresses. Since 1000 is not a power of 2, 1024 addresses are granted. The prefix length is calculated as $n = 32 - \log_2 1024 = 22$. An available block, 18.14.12.0/22, is granted to the ISP. It can be seen that the first address in decimal is 302,910,464, which is divisible by 1024.

Subnetting

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

Designing Subnets

- The subnetworks in a network should be carefully designed to enable the routing of packets.
- We assume the total number of addresses granted to the organization is **N**, the prefix length is **n**, the assigned number of addresses to each subnetwork is $\mathbf{N_{sub}}$, and the prefix length for each subnetwork is $\mathbf{n_{sub}}$.
- The following steps need to be carefully followed to guarantee the proper operation of the subnetworks:
  - **The number of addresses in each subnetwork should be a power of 2.**
  - **The prefix length for each subnetwork should be found using the following formula:**

$$n_{sub} = 32 - \log_2 N_{sub}$$

  - **The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger subnetworks.**

Example 18.5

An organization is granted a block of addresses with the beginning address 14.24.74.0/24.

How many nodes are granted?

256

What is the last IP address?

14.24.74.255

Example 18.5

An organization is granted a block of addresses with the beginning address 14.24.74.0/24.

The organization needs to have 3 sub-blocks of addresses to use in its three subnets:

one sub-block of 10 addresses,
one sub-block of 60 addresses,
and one sub-block of 120 addresses.

Design the sub-blocks.

Example 18.5

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

There are $2^{32-24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24. To satisfy the third requirement, we assign addresses to subblocks, starting with the largest and ending with the smallest one.

Example 18.5 (cont.)

a. ***The number of addresses in the largest sub-block, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as*** $n_1 = 32 - \log_2 128 = 25$. ***The first address in this block is 14.24.74.0/25; the last address is 14.24.74.127/25.***

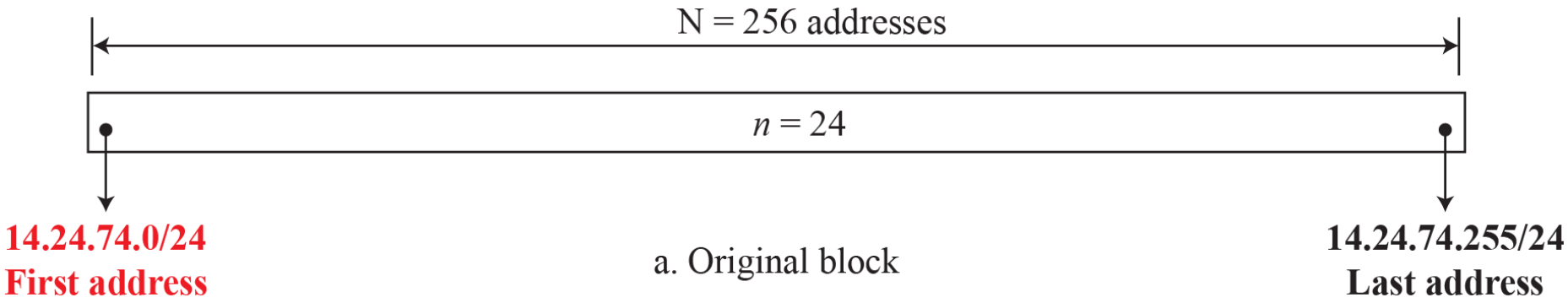
b. ***The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as*** $n_2 = 32 - \log_2 64 = 26$. ***The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.***

Example 18.5 (cont.)

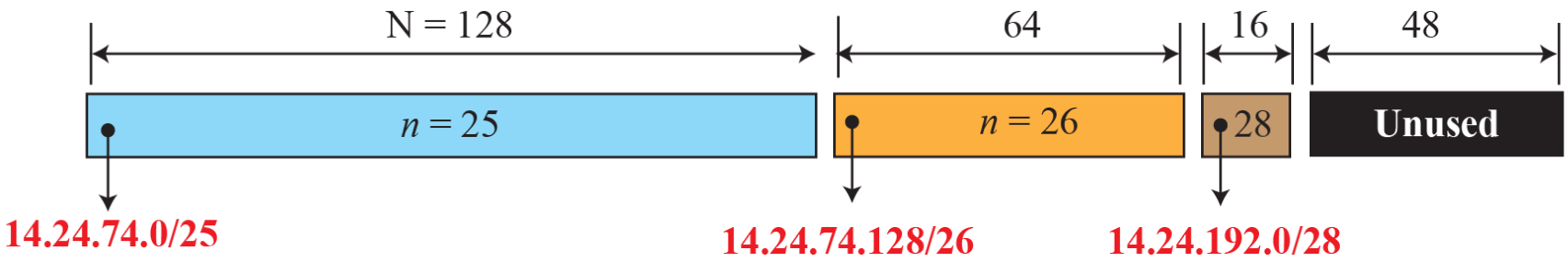
c. ***The number of addresses in the smallest sub-block, which requires 10 addresses, is not a power of 2. We allocate 16 addresses. The subnet mask for this subnet can be found as*** $n_1 = 32 - \log_2 16 = 28$. ***The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.***

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255. We don't know about the prefix length yet.

Solution to Example 18.5



a. Original block

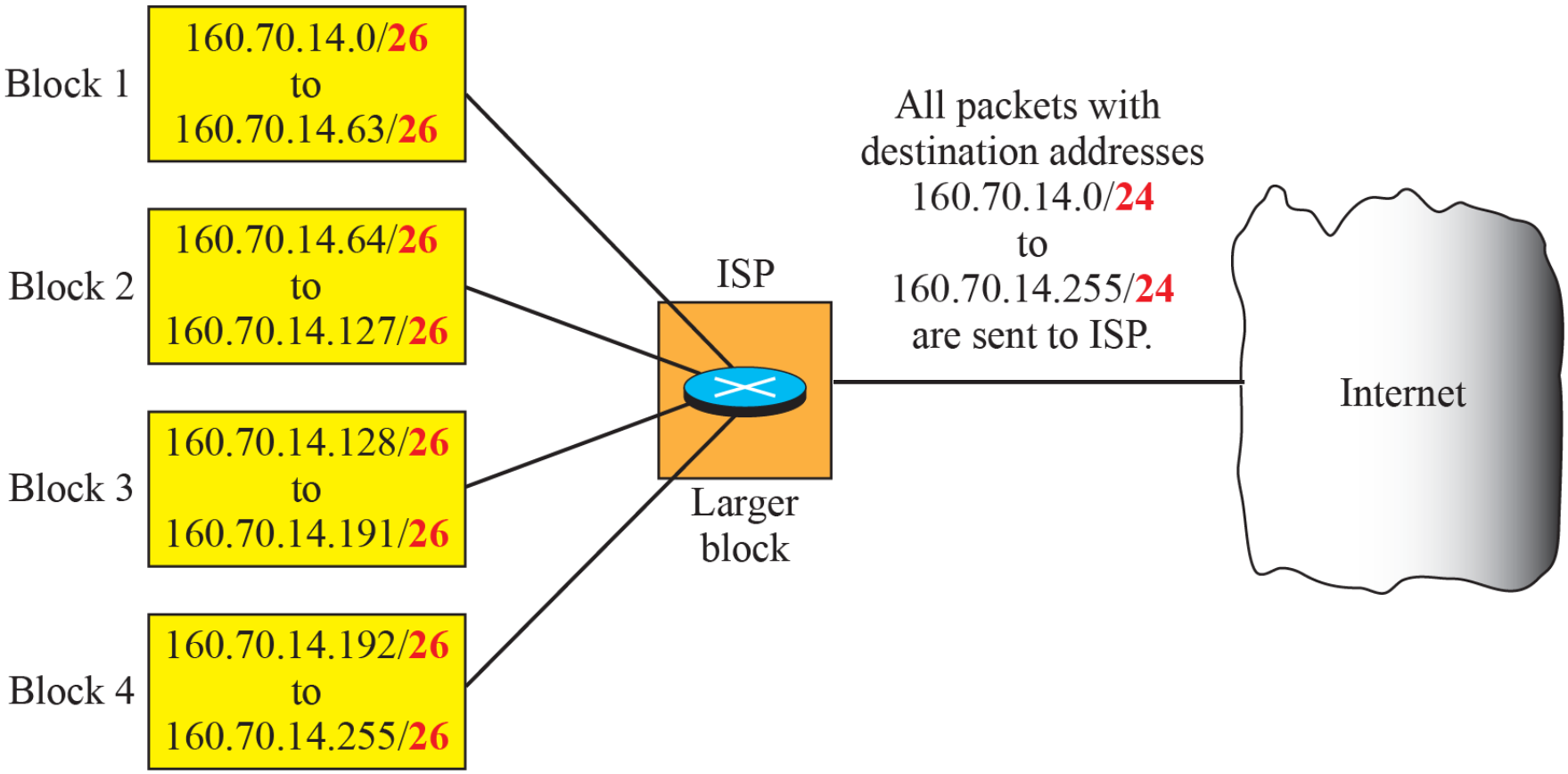


b. Subblocks

Address Aggregation

- One of the advantages of the CIDR strategy is address aggregation (sometimes called address summarization or route summarization).
- When blocks of addresses are combined to create a larger block, routing can be done based on the prefix of the larger block.
- ICANN assigns a large block of addresses to an ISP.
- Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers.

Example of address aggregation



Special Addresses

- **Five** special addresses that are used for special purposes:
- **This-host address:**
    - **0.0.0.0/32** is called the this-host address. It is used whenever a host needs to send an IP datagram but it does not know its own address to use as the source address
- **Limited-broadcast address:**
    - **255.255.255.255/32** is called the limited-broadcast address. It is used whenever a router or a host needs to send a datagram to all devices in a network. The routers in the network, however, block the packet having this address as the destination; the packet cannot travel outside the network.

Special Addresses (cont.)

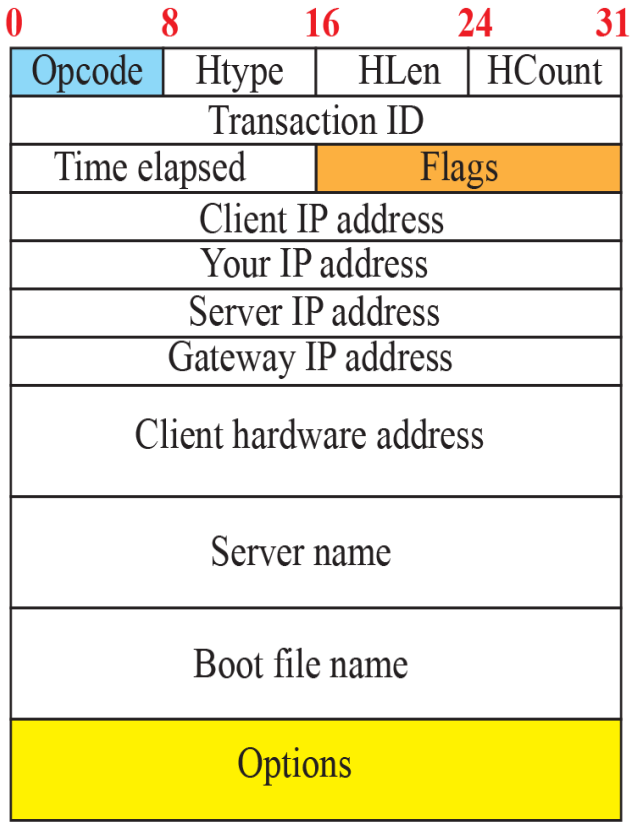
- **Loopback address:**
    - The block **127.0.0.0/8** is called the loopback address. A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host. Any address in the block is used to test a piece of software in the machine
- **Private addresses:**
    - Four blocks are assigned as private addresses: **10.0.0.0/8**, **172.16.0.0/12**, **192.168.0.0/16**, and **169.254.0.0/16**.
- **Multicast addresses:**
    - The block **224.0.0.0/4** is reserved for multicast addresses.

Dynamic Host Configuration Protocol (DHCP)

- Address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP).
- DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.
- DHCP has found such widespread use in the Internet that it is often called a plugand- play protocol.
- A network manager can configure DHCP to assign permanent IP addresses to the host and routers.
- DHCP can also be configured to provide temporary, on demand, IP addresses to hosts.
- The second capability can provide a temporary IP address to a traveller to connect her laptop to the Internet while she is staying in the hotel.
- It also allows an ISP with 1000 granted addresses to provide services to 4000 households, assuming not more than one-forth of customers use the Internet at the same time.
- In addition to its IP address, a computer also needs to know the network prefix (or address mask).
- Most computers also need two other pieces of information, such as the address of a default router to be able to communicate with other networks and the address of a name server to be able to use names instead of addresses

DHCP Message Format

- DHCP is a client-server protocol in which the client sends a request message and the server returns a response message.



Fields:

Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Lengh of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte option field has a dual purpose: It can carry either additional information or some specific vendor information.

DHCP Option Field

- The 64-byte option field has a dual purpose.
- It can carry either additional information or some specific vendor information.
- The server uses a number, called a **magic cookie**, in the format of an IP address with the value of 99.130.83.99.
- When the client finishes reading the message, it looks for this magic cookie.
- If present, the next 60 bytes are options.
- An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field.
- There are several tag fields that are mostly used by vendors.
- If the tag field is 53, the value field defines one of the 8 message types shown below:

1 **DHCP**DISCOVER
2 **DHCP**OFFER
3 **DHCP**REQUEST
4 **DHCP**DECLINE
5 **DHCP**ACK
6 **DHCP**NACK
7 **DHCP**RELEASE
8 **DHCP**INFORM

Operation of DHCP: DHCPDISCOVER

- The joining host creates a DHCP**DISCOVER** message in which only the transaction-ID field is set to a random number.
- No other field can be set because the host has no knowledge with which to do so.
- This message is encapsulated in a UDP user datagram with the source port set to 68 and the destination port set to 67.
- The user datagram is encapsulated in an IP datagram with the source address set to 0.0.0.0 (“this host”) and the destination address set to 255.255.255.255 (broadcast address).
- The reason is that the joining host knows neither its own address nor the server address.

Operation of DHCP: DHCPOFFER

- The DHCP server or servers (if more than one) responds with a DHCP**OFFER** message in which the your address field defines the offered IP address for the joining host and the server address field includes the IP address of the server.
- The message also includes the lease time for which the host can keep the IP address.
- This message is encapsulated in a user datagram with the same port numbers, but in the reverse order.
- The user datagram in turn is encapsulated in a datagram with the server address as the source IP address, but the destination address is a broadcast address, in which the server allows other DHCP servers to receive the offer and give a better offer if they can.

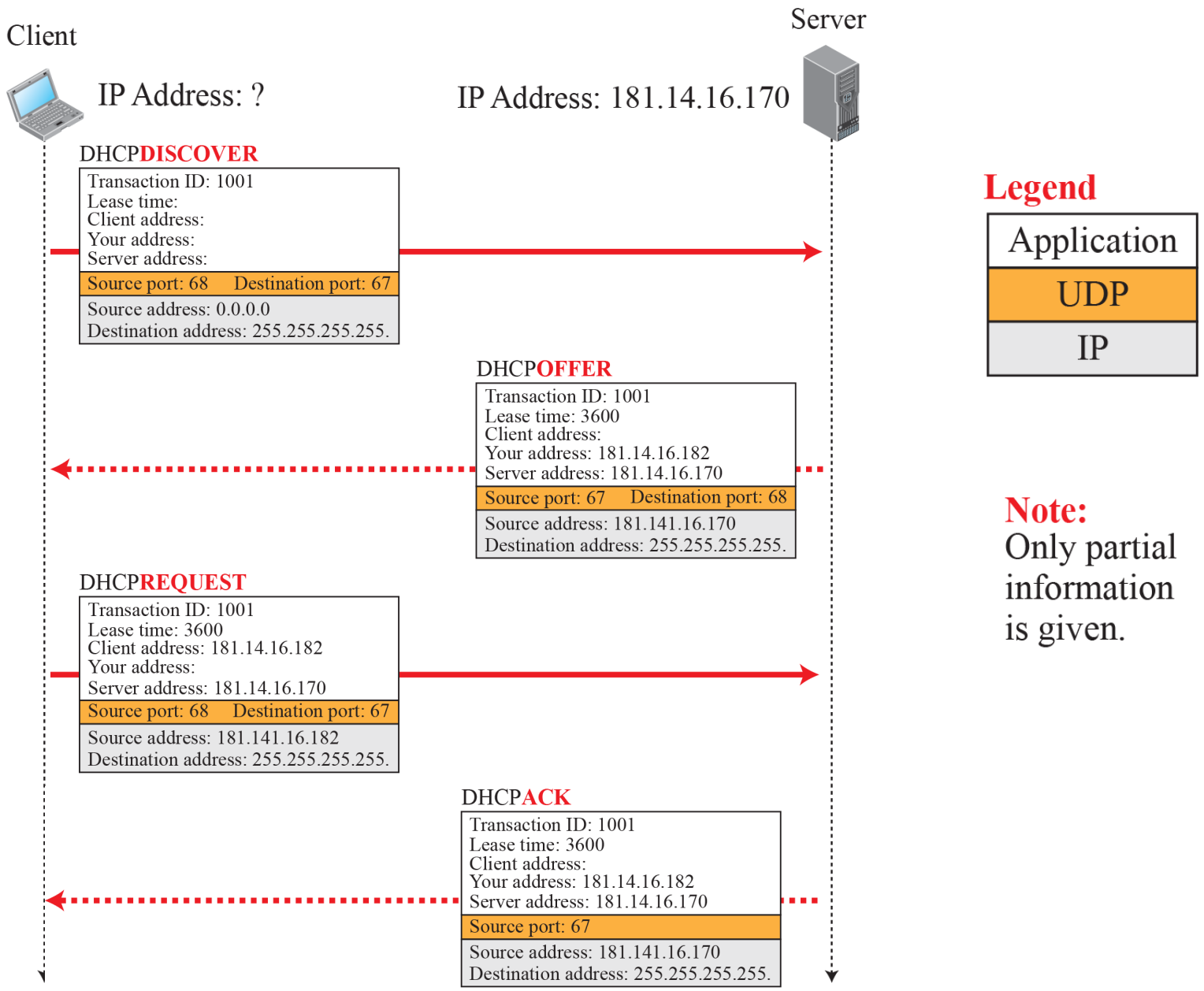
Operation of DHCP: DHCPREQUEST

- The joining host receives one or more offers and selects the best of them.
- The joining host then sends a DHCP**REQUEST** message to the server that has given the best offer.
- The fields with known value are set.
- The message is encapsulated in a user datagram with port numbers as the first message.
- The user datagram is encapsulated in an IP datagram with the source address set to the new client address, but the destination address still is set to the broadcast address to let the other servers know that their offer was not accepted.

Operation of DHCP: DHCPACK

- Finally, the selected server responds with a DHCP**ACK** message to the client if the offered IP address is valid.
- If the server cannot keep its offer (for example, if the address is offered to another host in between), the server sends a DHCP**NACK** message and the client needs to repeat the process.
- This message is also broadcast to let other servers know that the request is accepted or rejected.

Operation of DHCP



Two Well-Known Ports

- DHCP uses two well-known ports (68 and 67) instead of one well-known and one ephemeral.
- The reason for choosing the well-known port 68 instead of an ephemeral port for the client is that the response from the server to the client is broadcast.
- An IP datagram with the limited broadcast message is delivered to every host on the network.
- Now assume that a DHCP client and a DAYTIME client, for example, are both waiting to receive a response from their corresponding server and both have accidentally used the same temporary port number (56017, for example). Both hosts receive the response message from the DHCP server and deliver the message to their clients.
- The DHCP client processes the message; the DAYTIME client is totally confused with a strange message received.
- Using a well-known port number prevents this problem from happening.
- The response message from the DHCP server is not delivered to the DAYTIME client, which is running on the port number 56017, not 68.
- The temporary port numbers are selected from a different range than the well-known port numbers.
- If two DHCP clients are running at the same time: the messages can be distinguished by the value of the transaction ID, which separates each response from the other.

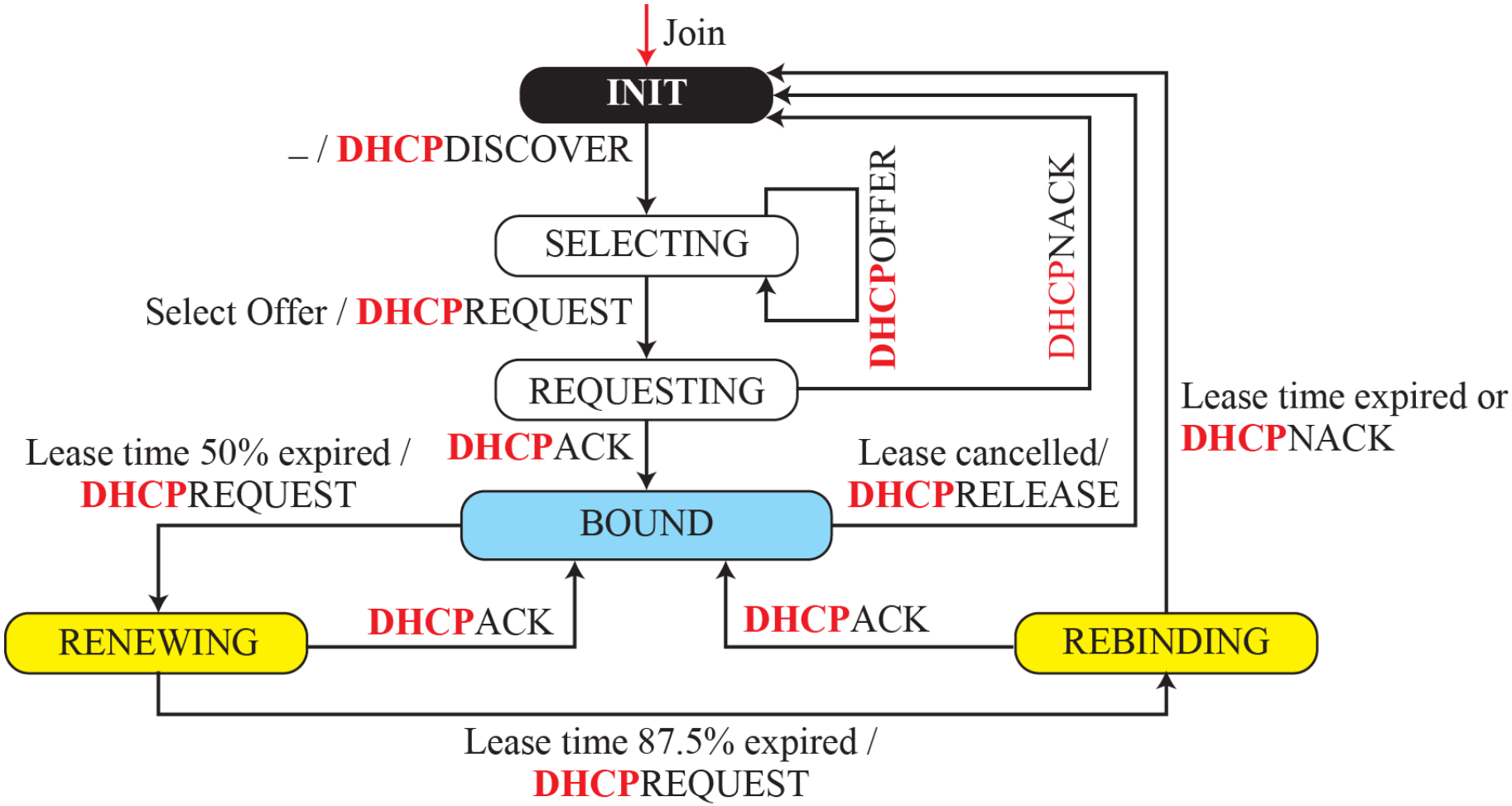
Using FTP

- The server does not send all of the information that a client may need for joining the network.
- In the DHCP**ACK** message, the server defines the pathname of a file in which the client can find complete information such as the address of the DNS server.
- The client can then use a file transfer protocol to obtain the rest of the needed information.

Error Control

- DHCP uses the service of UDP, which is not reliable.
- To provide error control, DHCP uses two strategies:
    - First, DHCP requires that UDP use the checksum.
    - Second, the DHCP client uses timers and a retransmission policy if it does not receive the DHCP reply to a request.
- However, to prevent a traffic jam when several hosts need to retransmit a request (for example, after a power failure), DHCP forces the client to use a random number to set its timers.

Transition States



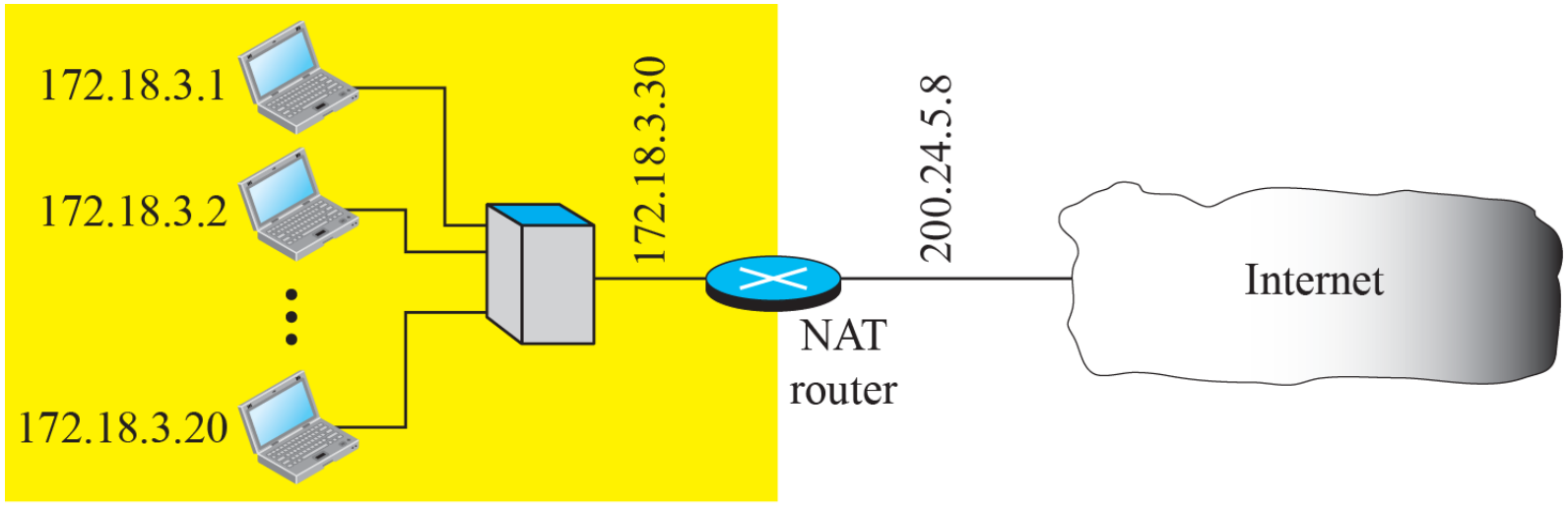
Network Address Resolution (NAT)

- In most situations, only a portion of computers in a small network need access to the Internet.
- For example, assume that in a small business with 20 computers the maximum number of computers that access the Internet simultaneously is only 4.
- Most of the computers are either doing some task that does not need Internet access or communicating with each other.
- This small business can use the TCP/IP protocol for both internal and universal communication.
- The business can use 20 (or 25) addresses from the private block addresses (discussed before) for internal communication; five addresses for universal communication can be assigned by the ISP
- A technology that can provide the mapping between the private and universal addresses, and at the same time support virtual private networks, is Network Address Translation (NAT).
- The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.
- The site must have only one connection to the global Internet through a NAT-capable router that runs NAT software

Private IPv4 Addresses

Class A range	10.0.0.0 to 10.255.255.255
Class B range	172.16.0.0 to 172.31.255.255
Class C range	192.168.0.0 to 192.168.255.255
Other:	127.0.0.0 to 127.255.255.255

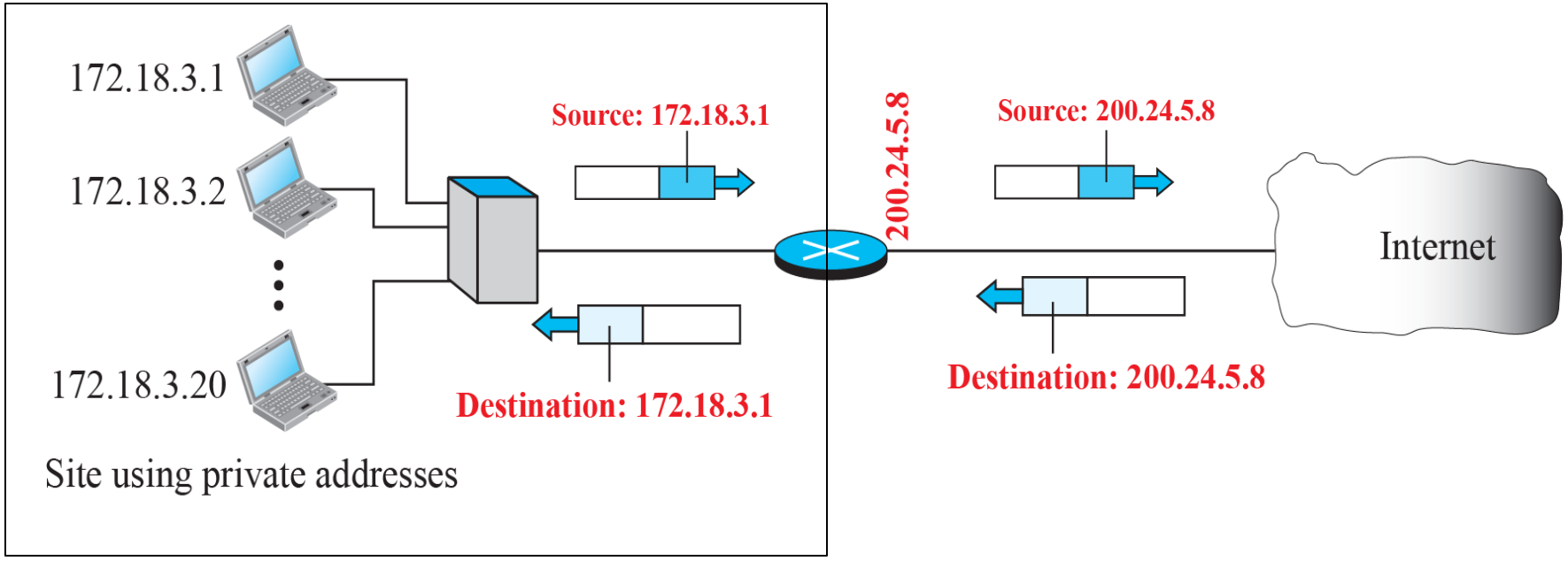
NAT



Site using private addresses

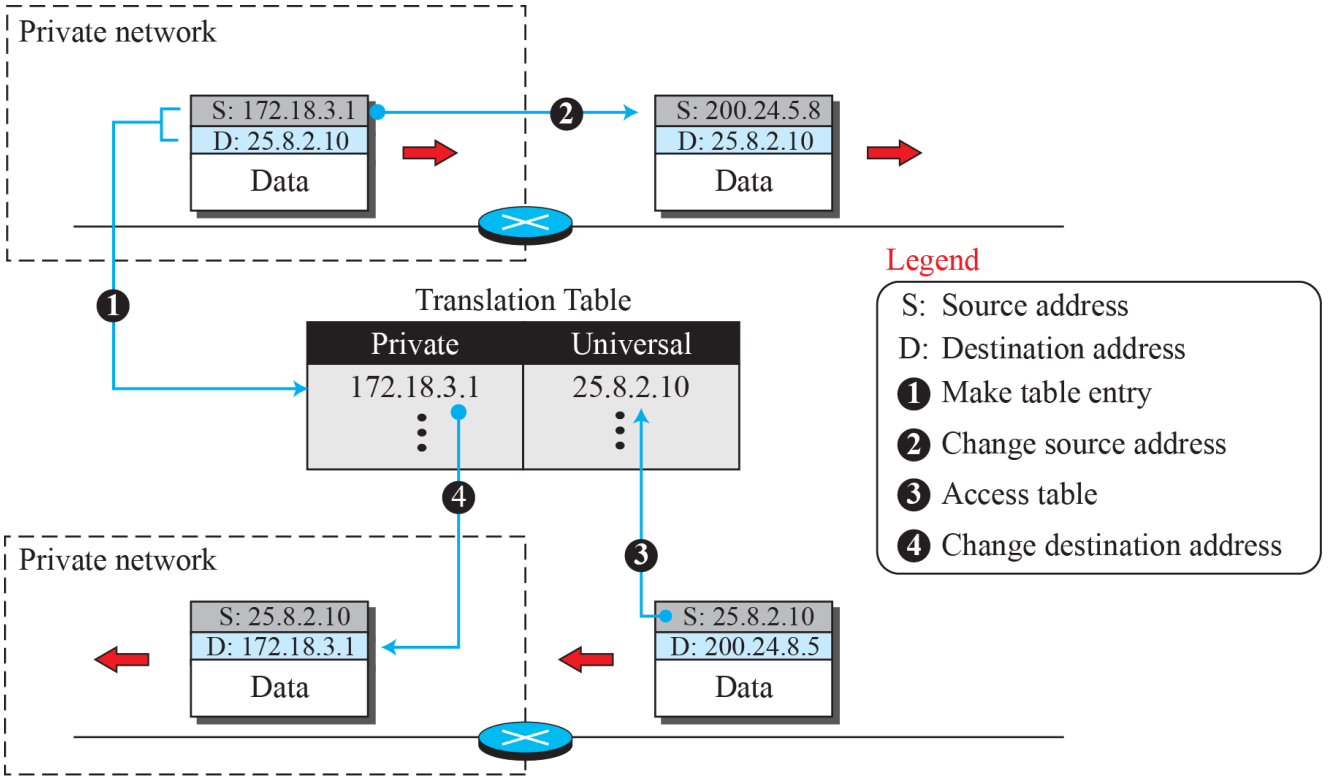
Address translation

- All of the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.



Translation Table: Using One IP Address

- In its simplest form, a translation table has only two columns: the private address and the external address (destination address of the packet).
- When the router translates the source address of the outgoing packet, it also makes note of the destination address— where the packet is going.
- When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.
- In this strategy, communication must always be initiated by the private network.



Translation Table: Using a Pool of IP Addresses

- The use of only one global address by the NAT router allows only one private-network host to access a given external host.
- To remove this restriction, the NAT router can use a pool of global addresses. For example, instead of using only one global address (200.24.5.8), the NAT router can use four addresses (200.24.5.8, 200.24.5.9, 200.24.5.10, and 200.24.5.11).
- In this case, four private-network hosts can communicate with the same external host at the same time because each pair of addresses defines a separate connection.
- However, there are still some drawbacks:
    - No more than four connections can be made to the same destination.
    - No private-network host can access two external server programs (e.g., HTTP and TELNET) at the same time.
    - Two private-network hosts cannot access the same external server program (e.g., HTTP or TELNET) at the same time.

Translation Table: Using Both IP Addresses and Port Addresses

- To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table.
- For example, suppose two hosts inside a private network with addresses 172.18.3.1 and 172.18.3.2 need to access the HTTP server on external host 25.8.3.2. If the translation table has five columns, instead of two, that include the source and destination port addresses and the transport-layer protocol, the ambiguity is eliminated.
- Note that when the response from HTTP comes back, the combination of source address (25.8.3.2) and destination port address (1401) defines the private network host to which the response should be directed.
- Note also that for this translation to work, the ephemeral port addresses (1400 and 1401) must be unique.

Private address	*Private port*	*External address*	*External port*	*Transport protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
⋮	⋮	⋮	⋮	⋮

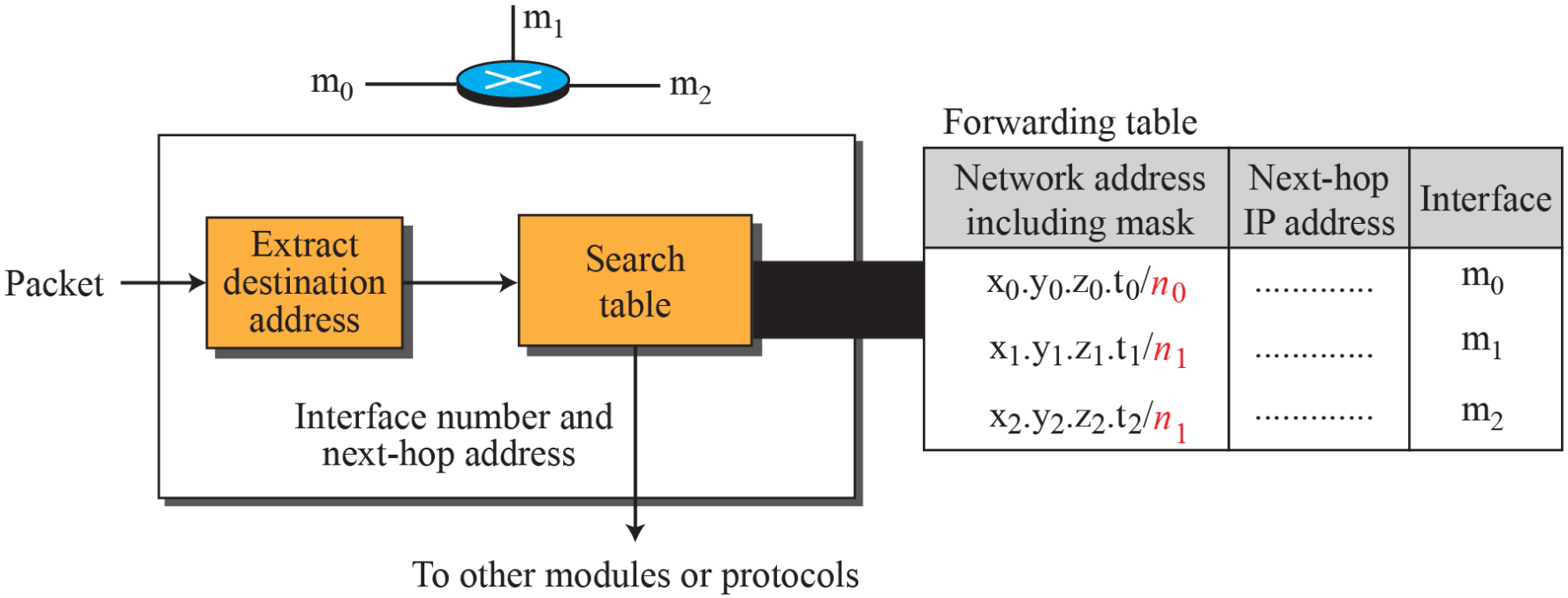
Forwarding of IP Packets

- Forwarding means to place the packet in its route to its destination
- Since the Internet today is made of a combination of links (networks), forwarding means to deliver the packet to the next hop (which can be the final destination or the intermediate connecting device).
- Although the IP protocol was originally designed as a connectionless protocol, today the tendency is to change it to a connection-oriented protocol.
- When IP is used as a connectionless protocol, forwarding is based on the destination address of the IP datagram
- When the IP is used as a connection-oriented protocol, forwarding is based on the label attached to an IP datagram.

Forwarding Based on Destination Address

- This is a traditional approach, which is prevalent today
- In this case, forwarding requires a host or a router to have a forwarding 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 next hop to deliver the packet to
- In classless addressing, the whole address space is one entity; there are no classes
- Forwarding requires one row of information for each block involved.
- The table needs to be searched based on the network address (first address in the block)
- The destination address in the packet gives no clue about the network address
- To solve the problem, we need to include the mask (/*n*) in the table
- In other words, a classless forwarding table needs to include four pieces of information: the mask, the network address, the interface number, and the IP address of the next router
- However, we often see in the literature that the first two pieces are combined
- For example, if *n* is 26 and the network address is 180.70.65.192, then one can combine the two as one piece of information: 180.70.65.192/26.

Simplified forwarding module in classless address

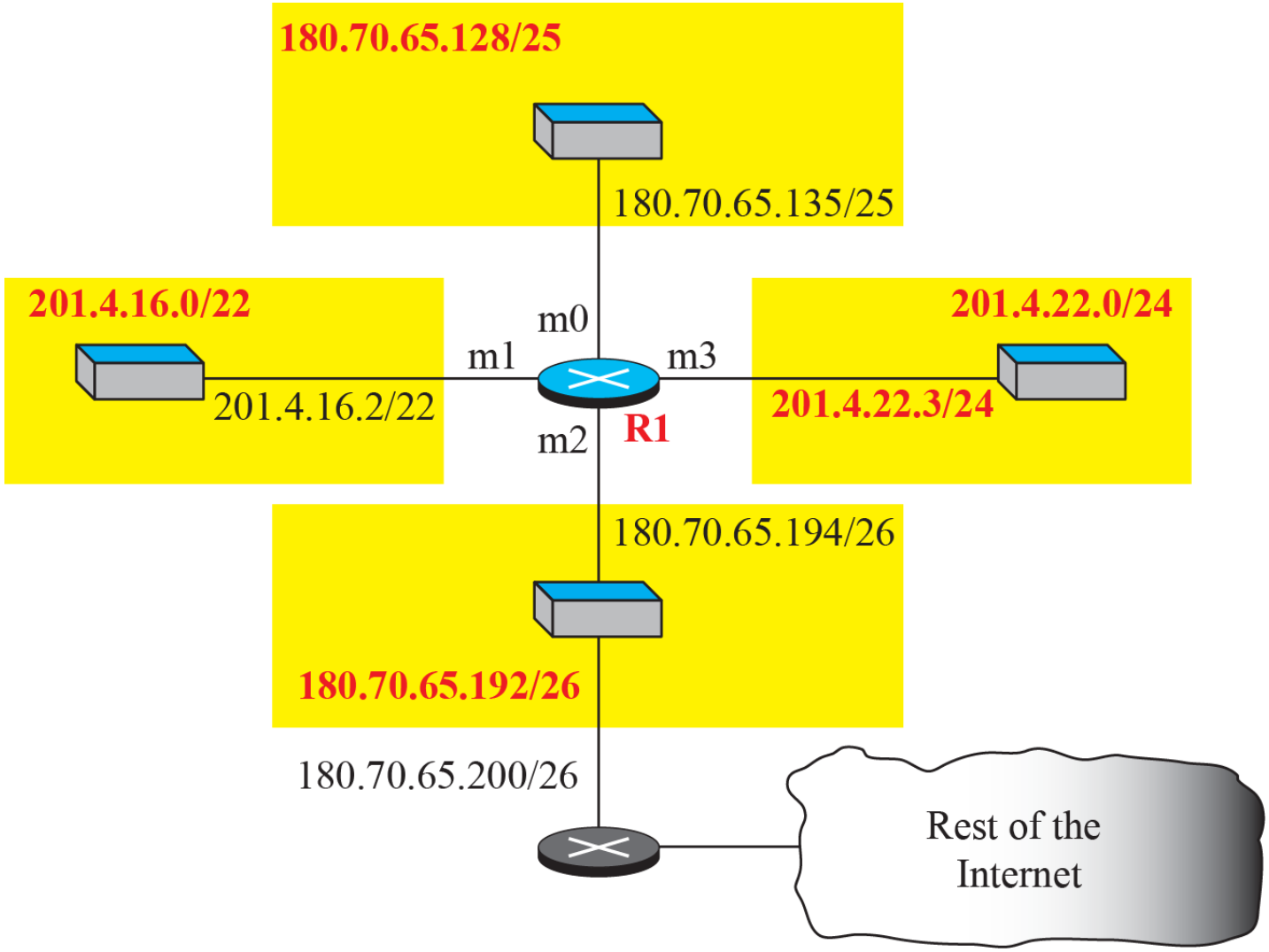


Forwarding Based on Destination Address (cont.)

- The job of the forwarding module is to search the table, row by row.
- In each row, the n leftmost bits of the destination address (prefix) are kept and the rest of the bits (suffix) are set to 0s.
- If the resulting address (which we call the network address), matches with the address in the first column, the information in the next two columns is extracted; otherwise the search continues.
- Normally, the last row has a default value in the first column which indicates all destination addresses that did not match the previous rows.

Example 18.7

Make a forwarding table for router R1 using the configuration in Figure below



Example 18.7

Solution

Table below shows the corresponding table.

Forwarding table for router R1

Network address/mask	*Next hop*	*Interface*
180.70.65.192/**26**	—	m2
180.70.65.128/**25**	—	m0
201.4.22.0/**24**	—	m3
201.4.16.0/**22**	—	m1
Default	180.70.65.200	m2

Example 18.8

Instead of previous Table, we can use Table below, in which the network address/mask is given in bits.

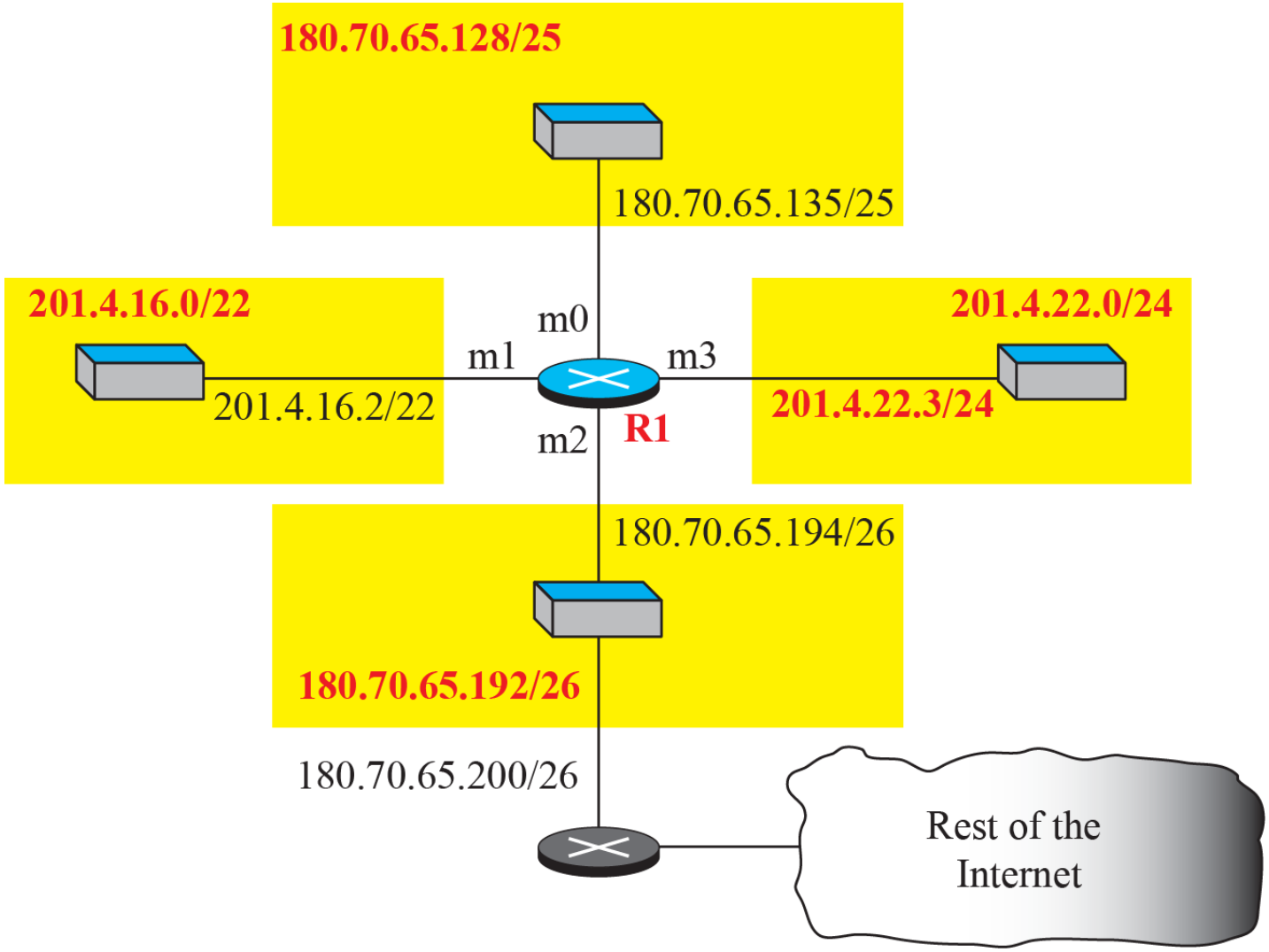
Forwarding table for router R1 using prefix bits

Leftmost bits in the destination address	*Next hop*	*Interface*
10110100 01000110 01000001 11	—	m2
10110100 01000110 01000001 1	—	m0
11001001 00000100 00011100	—	m3
11001001 00000100 000100	—	m1
Default	180.70.65.200	m2

When a packet arrives whose leftmost 26 bits in the destination address match the bits in the first row, the packet is sent out from interface m2. And so on.

Example 18.9

Show the forwarding process if a packet arrives at R1 in Figure below with the destination address 180.70.65.140.



Example 18.9

Solution

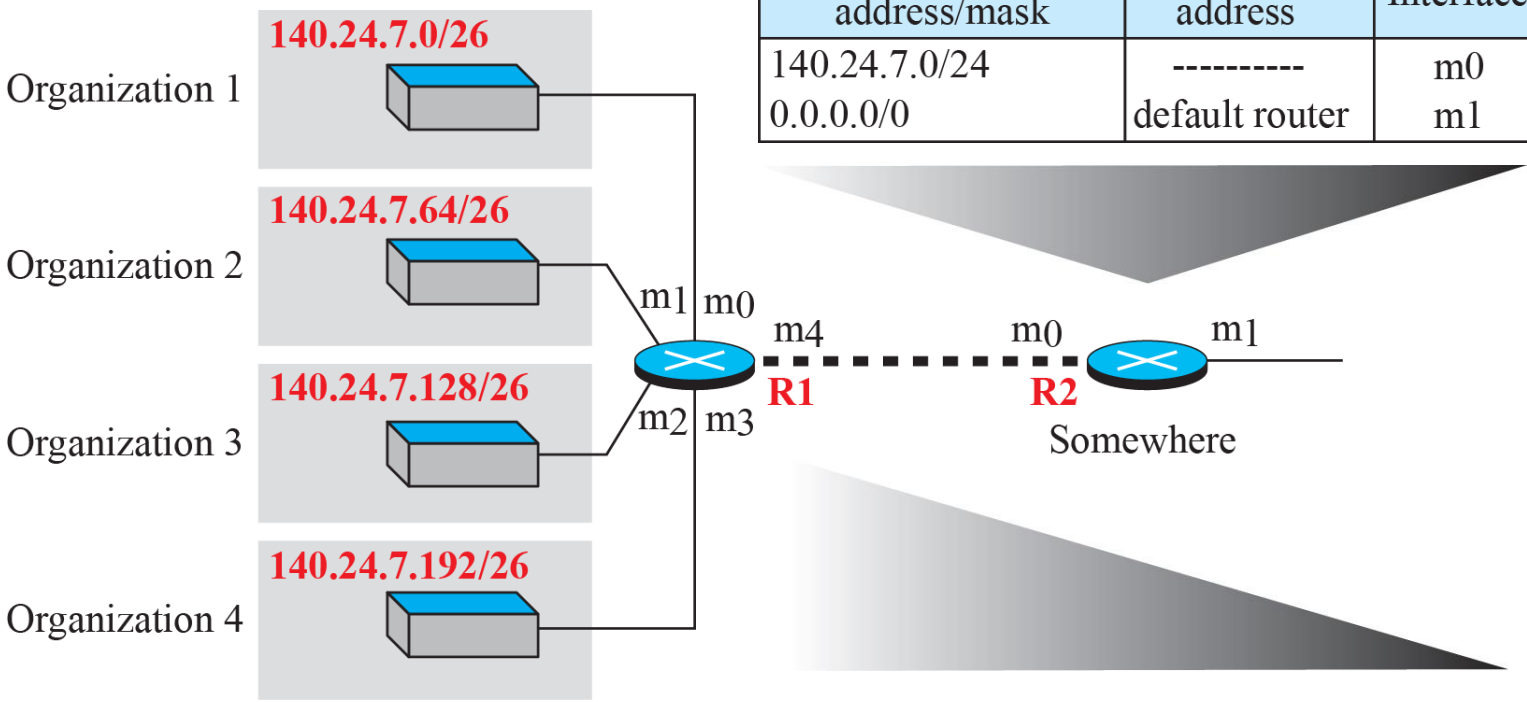
1. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.

2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are extracted for forwarding the packet

Address Aggregation

- When we use classful addressing, there is only one entry in the forwarding table for each site outside the organization.
- The entry defines the site even if that site is subnetted.
- When a packet arrives at the router, the router checks the corresponding entry and forwards the packet accordingly.
- When we use classless addressing, it is likely that the number of forwarding table entries will increase.
- This is because the intent of classless addressing is to divide up the whole address space into manageable blocks.
- The increased size of the table results in an increase in the amount of time needed to search the table.
- To alleviate the problem, the idea of address aggregation was designed.

Address aggregation



Forwarding table for R2

Network address/mask	Next-hop address	Interface
140.24.7.0/24	----------	m0
0.0.0.0/0	default router	m1

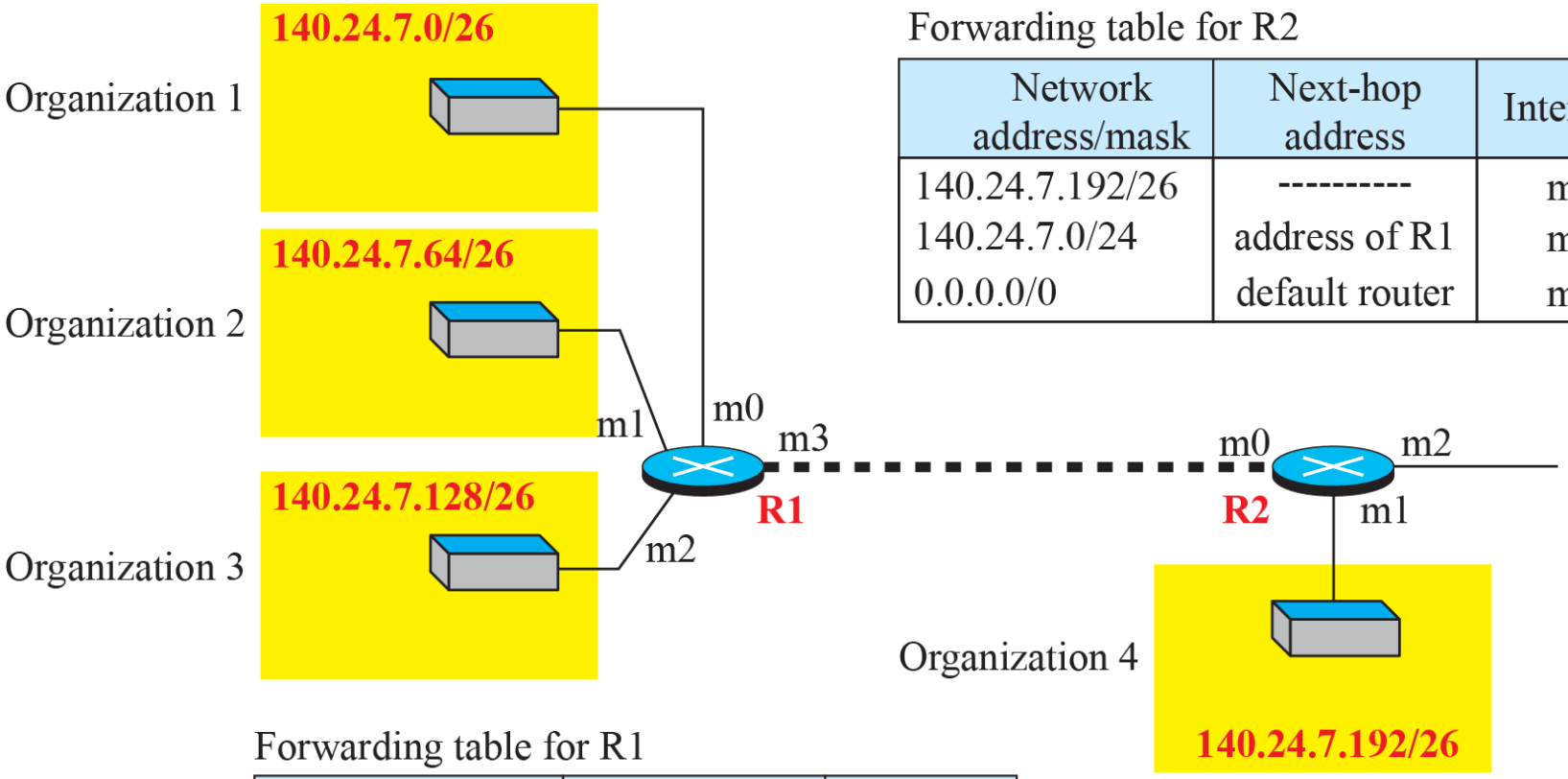
Forwarding table for R1

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

Longest Mask Matching

- What happens if one of the organizations in the previous figure is not geographically close to the other three
- For example, if organization 4 cannot be connected to router R1 for some reason,
    - can we still use the idea of address aggregation and still assign block 140.24.7.192/26 to organization 4?
- The answer is yes, because routing in classless addressing uses another principle, longest mask matching.
- This principle states that the forwarding table is sorted from the longest mask to the shortest mask.
- In other words, if there are three masks, /27, /26, and /24, the mask /27 must be the first entry and /24 must be the last.
- Let us see if this principle solves the situation in which organization 4 is separated from the other three organizations.

Longest mask matching



Forwarding table for R2

Network address/mask	Next-hop address	Interface
140.24.7.192/26	----------	m1
140.24.7.0/24	address of R1	m0
0.0.0.0/0	default router	m2

Forwarding table for R1

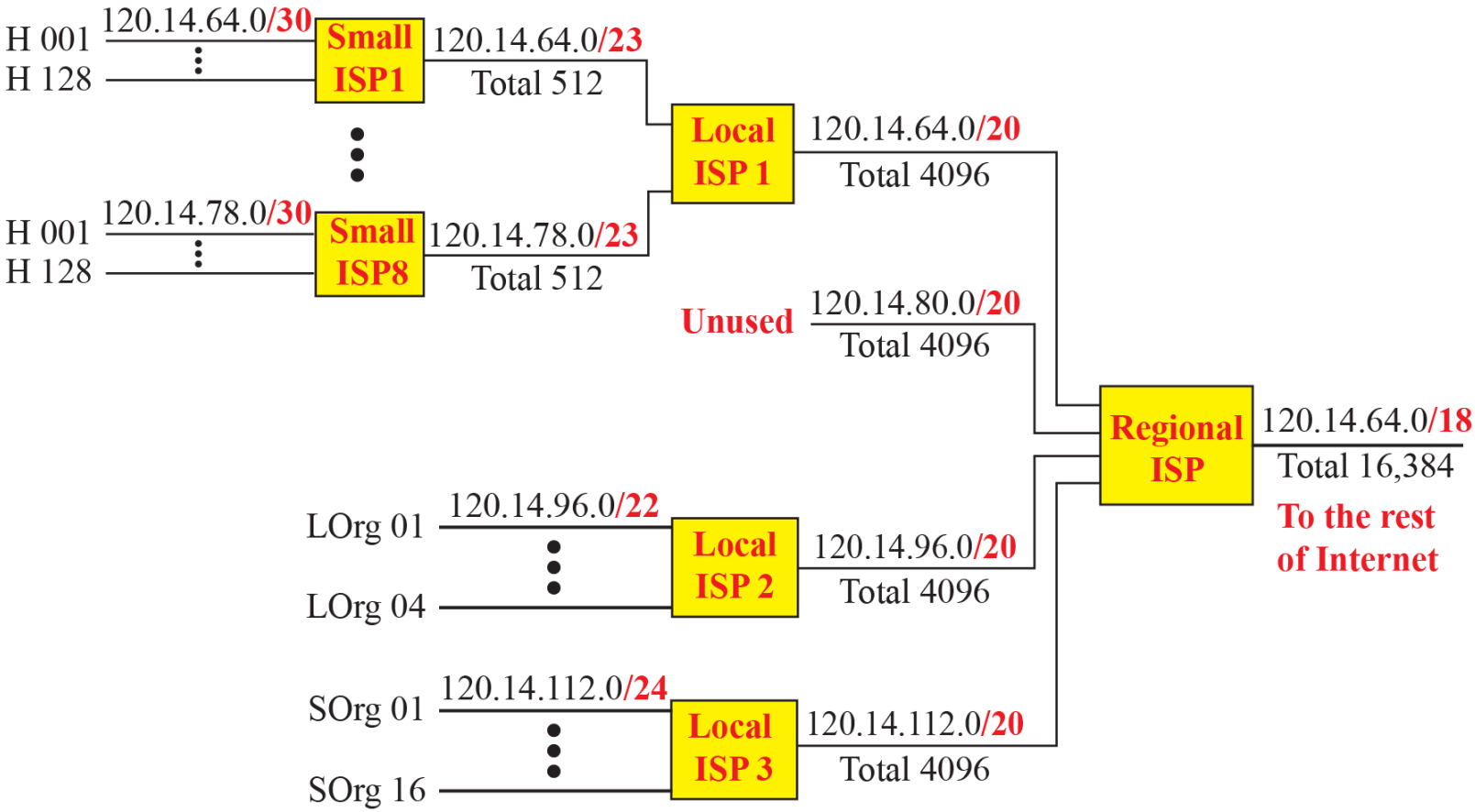
Network address/mask	Next-hop address	Interface
140.24.7.0/26	----------	m0
140.24.7.64/26	----------	m1
140.24.7.128/26	----------	m2
0.0.0.0/0	default router	m3

Hierarchical Routing

- We said that the Internet is divided into backbone and national ISPs.
- National ISPs are divided into regional ISPs, and regional ISPs are divided into local ISPs.
- If the forwarding table has a sense of hierarchy like the Internet architecture, the forwarding table can decrease in size
- A local ISP can be assigned a single, but large, block of addresses with a certain prefix length.
- The local ISP can divide this block into smaller blocks of different sizes, and assign these to individual users and organizations, both large and small.
- If the block assigned to the local ISP starts with a.b.c.d/n, the ISP can create blocks starting with e.f.g.h/m, where m may vary for each customer and is greater than n.
- The rest of the Internet does not have to be aware of this division.
- All customers of the local ISP are defined as a.b.c.d/n to the rest of the Internet.
- Every packet destined for one of the addresses in this large block is routed to the local ISP.
- There is only one entry in every router in the world for all of these customers.
- They all belong to the same group.
- Of course, inside the local ISP, the router must recognize the subblocks and route the packet to the destined customer.

Example 18.10

As an example of hierarchical routing, let us consider Figure below. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into 4 subblocks, each with 4096 addresses. Three of these sub-blocks are assigned to three local ISPs, the second sub-block is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.



Geographical Routing

- To decrease the size of the forwarding table even further, we need to extend hierarchical routing to include geographical routing.
- We must divide the entire address space into a few large blocks.
- We assign a block to America, a block to Europe, a block to Asia, a block to Africa, and so on.
- The routers of ISPs outside of Europe will have only one entry for packets to Europe in their forwarding tables.
- The routers of ISPs outside of America will have only one entry for packets to America in their forwarding tables, and so on.

Forwarding Table Search Algorithms

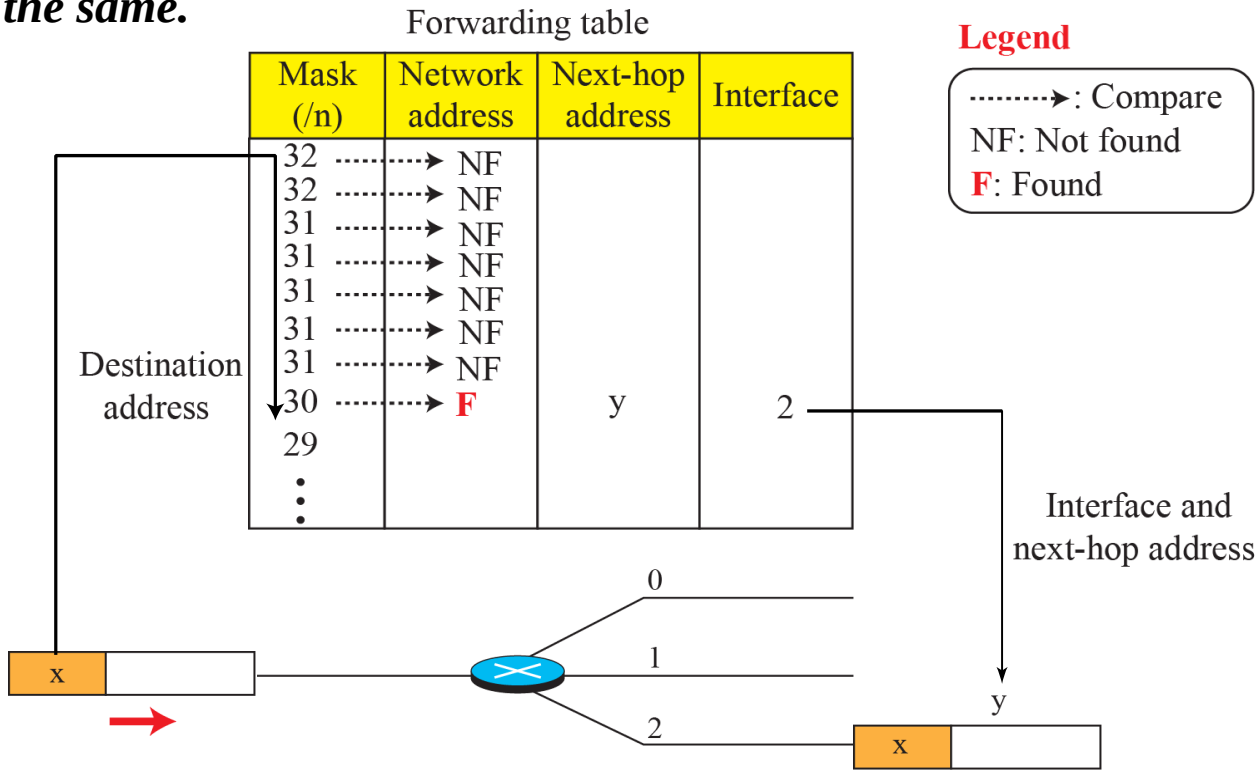
- In classless addressing, there is no network information in the destination address.
- The simplest, but not the most efficient, search method is called the longest prefix match (as we discussed before).
- The forwarding table can be divided into buckets, one for each prefix.
- The router first tries the longest prefix.
- If the destination address is found in this bucket, the search is complete.
- If the address is not found, the next prefix is searched, and so on.
- It is obvious that this type of search takes a long time.
- One solution is to change the data structure used for searching.
- Instead of a list, other data structures (such as a tree or a binary tree) can be used.

Forwarding Based on Label

- In the 1980s, an effort started to somehow change IP to behave like a connection oriented protocol in which the routing is replaced by switching
- As we discussed earlier in the chapter, in a connectionless network (datagram approach), a router forwards a packet based on the destination address in the header of the packet.
- On the other hand, in a connection-oriented network (virtual-circuit approach), a switch forwards a packet based on the label attached to the packet.
- Routing is normally based on searching the contents of a table; switching can be done by accessing a table using an index.
- In other words, routing involves searching; switching involves accessing

Example 18.11

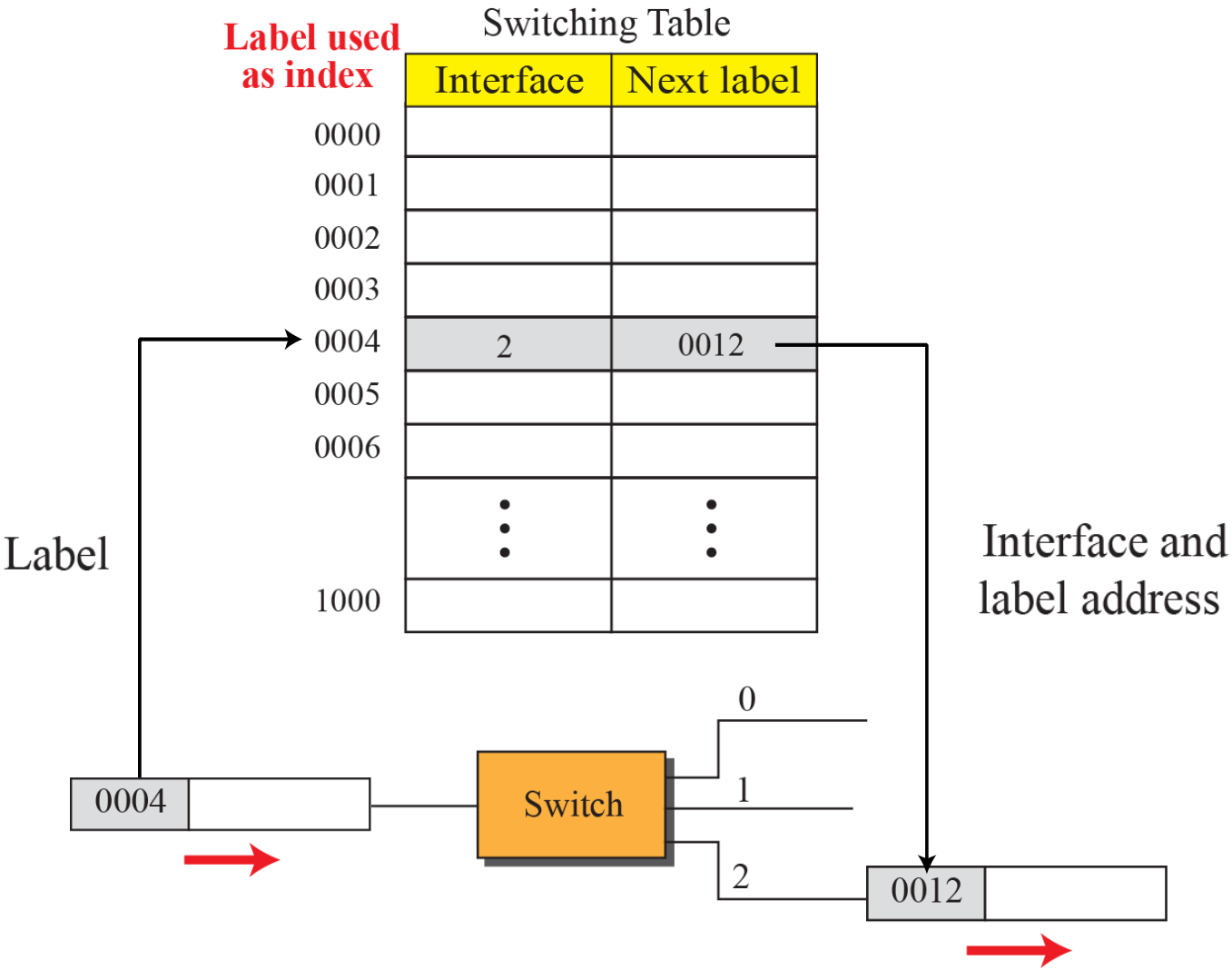
- ***Figure below shows a simple example of searching in a forwarding table using the longest mask algorithm. Although there are some more efficient algorithms today, the principle is the same.***



- ***When the forwarding algorithm gets the destination address of the packet, it needs to delve into the mask column.***
- ***For each entry, it needs to apply the mask to find the destination network address.***
- ***It then needs to check the network addresses in the table until it finds the match.***
- ***The router then extracts the next-hop address and the interface number to be delivered to the data-link layer.***

Example 18.12

- ***Figure below shows a simple example of using a label to access a switching table. Since the labels are used as the index to the table, finding the information in the table is immediate.***

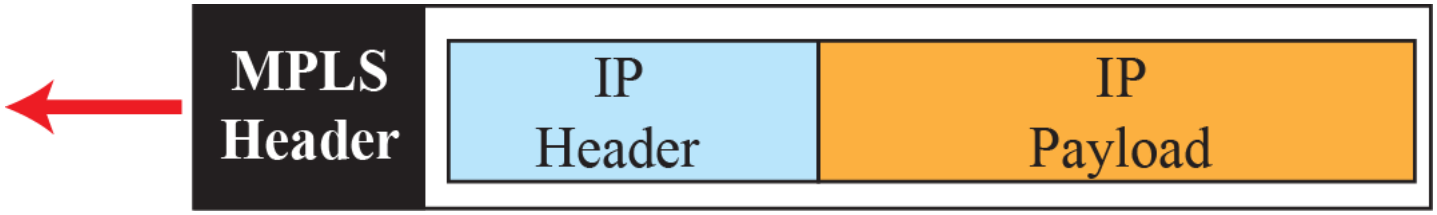


Multi-Protocol Label Switching (MPLS)

- During the 1980s, several vendors created routers that implement switching technology.
- Later IETF approved a standard that is called Multi-Protocol Label Switching.
- In this standard, some conventional routers in the Internet can be replaced by MPLS routers, which can behave like a router and a switch.
- When behaving like a router, MPLS can forward the packet based on the destination address; when behaving like a switch, it can forward a packet based on the label.

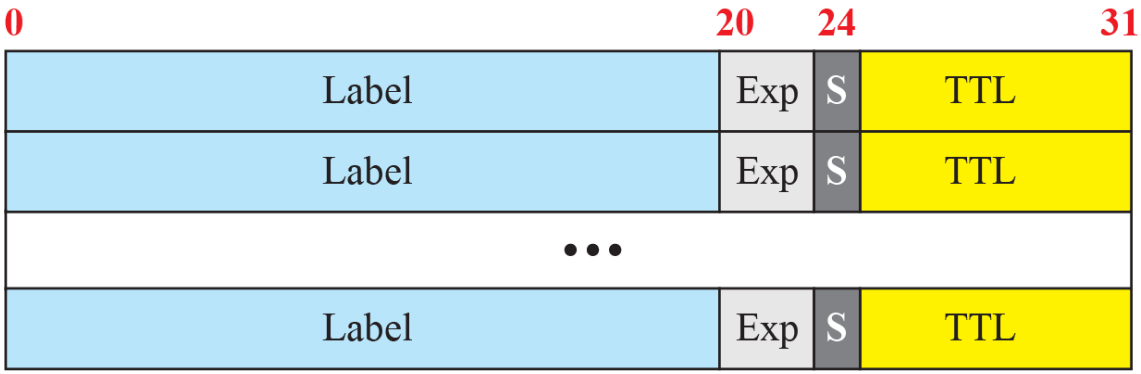
A New Header

- To simulate connection-oriented switching using a protocol like IP, the first thing that is needed is to add a field to the packet that carries the label
- The IPv4 packet format does not allow this extension (although this field is provided in the IPv6 packet format).
- The solution is to encapsulate the IPv4 packet in an MPLS packet (as though MPLS were a layer between the data-link layer and the network layer).
- The whole IP packet is encapsulated as the payload in an MPLS packet and an MPLS header is added



MPLS Header Stack

- The MPLS header is actually a stack of subheaders that is used for multilevel hierarchical switching
- MPLS header in which each subheader is 32 bits (4 bytes) long.



- **Label:** This 20-bit field defines the label that is used to index the forwarding table in the router.
- **Exp:** This 3-bit field is reserved for experimental purposes.
- **S:** The one-bit stack field defines the situation of the subheader in the stack. When the bit is 1, it means that the header is the last one in the stack.
- **TTL:** This 8-bit field is similar to the TTL field in the IP datagram. Each visited router decrements the value of this field. When it reaches zero, the packet is discarded to prevent looping.

Hierarchical Switching

- A stack of labels in MPLS allows hierarchical switching.
- This is similar to conventional hierarchical routing.
- For example, a packet with two labels can use the top label to forward the packet through switches outside an organization; the bottom label can be used to route the packet inside the organization to reach the destination subnet.

Routers as Packet Switches

- As we may have guessed by now, the packet switches that are used in the network layer are called routers. Routers can be configured to act as either a datagram switch or a virtual-circuit switch.