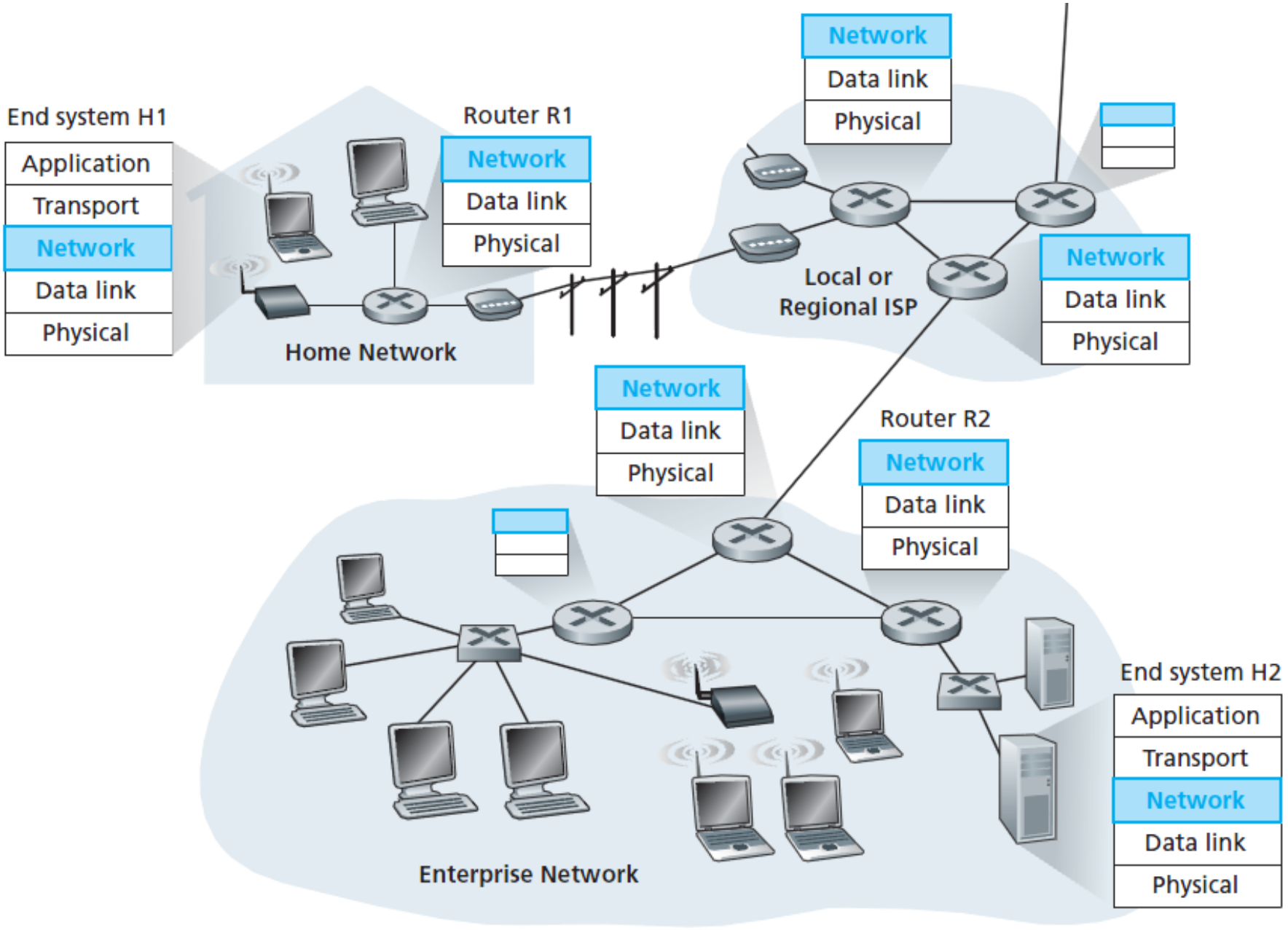
20MCA104
ADVANCED COMPUTER NETWORKS

Module 3

NETWORK LAYER

Network Layer

- One of the most complex layers in the protocol stack.
- Host-to-host communication service.
  - Move segments from sending host to receiving host.
  - On sending side, encapsulates segments into datagrams.
  - On receiving side, delivers segments to transport layer.
- Network layer protocols present in *every* host and router.
- Router examines header fields in all IP datagrams passing through it.


End system H1
Application
Transport
Network
Data link
Physical
Router R1
Network
Data link
Physical
Home Network
Network
Data link
Physical
Local or
Regional ISP
Network
Data link
Physical
Network
Data link
Physical
Router R2
Network
Data link
Physical
Enterprise Network
End system H2
Application
Transport
Network
Data link
Physical

Forwarding and Routing

- The role of the network layer is—to move packets from a sending host to a receiving host.
- To do so, two important network-layer functions can be identified:

1. *Forwarding.*
   - When a packet arrives at a router's input link, the router must move the packet to the appropriate output link.
2. *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.**

Forwarding Table

- Every router has a **forwarding table**.
- A router forwards a packet by examining the value of a field in the arriving packet's header, and then using this header value to index into the router's forwarding table.
- The value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded.
- Depending on the network-layer protocol, the header value could be the destination address of the packet or an indication of the connection to which the packet belongs.

Routing Algorithm

- The **routing algorithm** determines the values that are inserted into the routers' forwarding tables.
- The routing algorithm may be
  - Centralized or
  - Decentralized.
- A router receives routing protocol messages, which are used to configure its forwarding table.

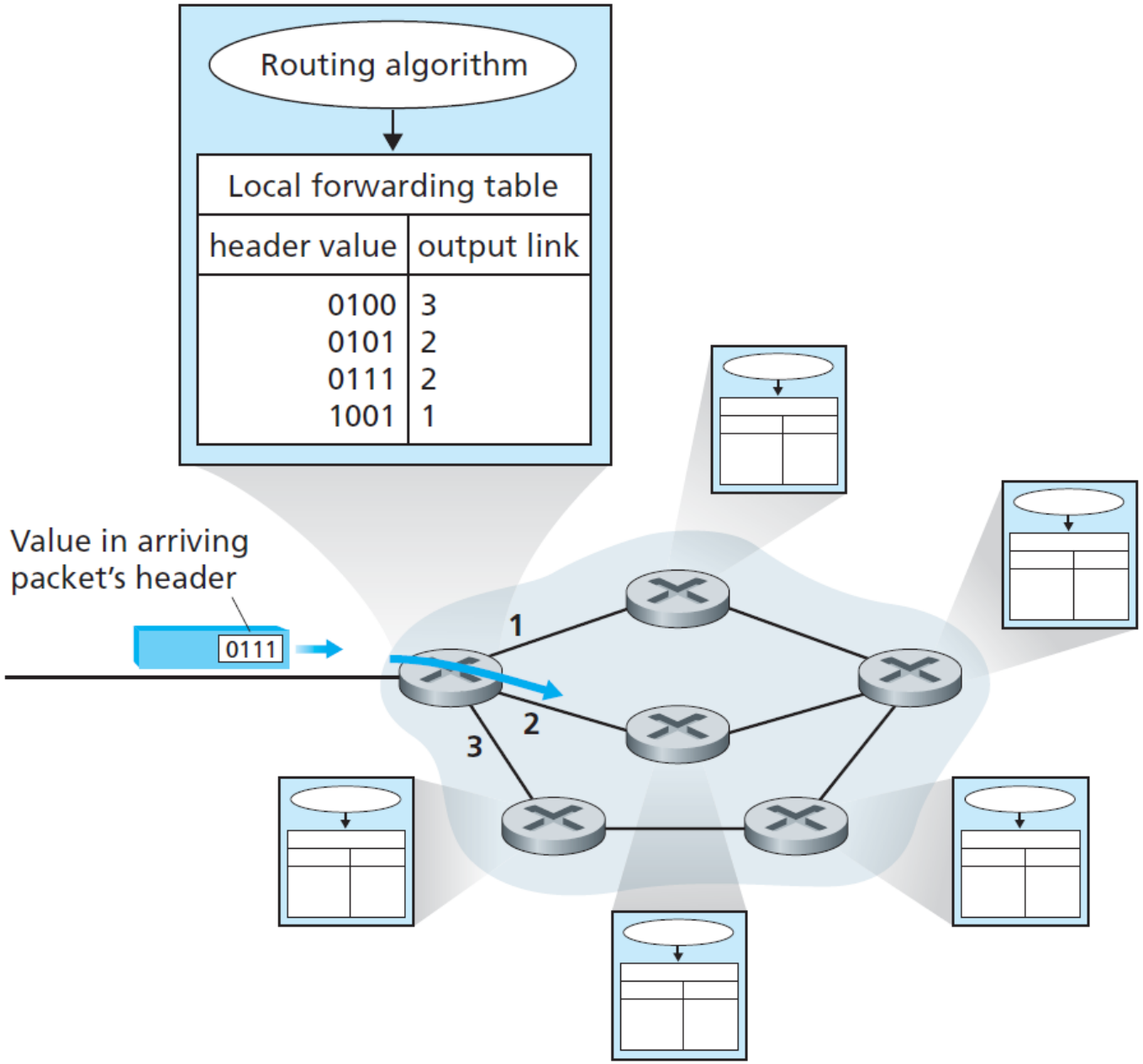


Figure 4.2 ♦ Routing algorithms determine values in forwarding tables

Connection Setup

- 3rd important function in some network architectures:
  - ATM, frame relay, Multiprotocol Label Switching (MPLS)
- Before datagrams flow, the routers along the chosen path from source to destination to handshake with each other in order to set up state.

Network Service Models

- Defines the characteristics of end-to-end transport of packets between sending and receiving end systems.

- Specific services that could be provided by the network layer include:
  - *Guaranteed delivery.*
  - *Guaranteed delivery with bounded delay.*
  - *In-order packet delivery.*
  - *Guaranteed minimal bandwidth.*
  - *Guaranteed maximum jitter.*
  - *Security services.*
    - *Confidentiality,*
    - *Data integrity and*
    - *Source authentication services.*

Network Architectures

- *Internet*
- *Constant bit rate (CBR) ATM network service.*
- *Available bit rate (ABR) ATM network service.*

Network Architecture	Service Model	Bandwidth Guarantee	No-Loss Guarantee	Ordering	Timing	Congestion Indication
Internet	Best Effort	None	None	Any order possible	Not maintained	None
ATM	CBR	Guaranteed constant rate	Yes	In order	Maintained	Congestion will not occur
ATM	ABR	Guaranteed minimum	None	In order	Not maintained	Congestion indication provided

Virtual Circuit and Datagram Networks

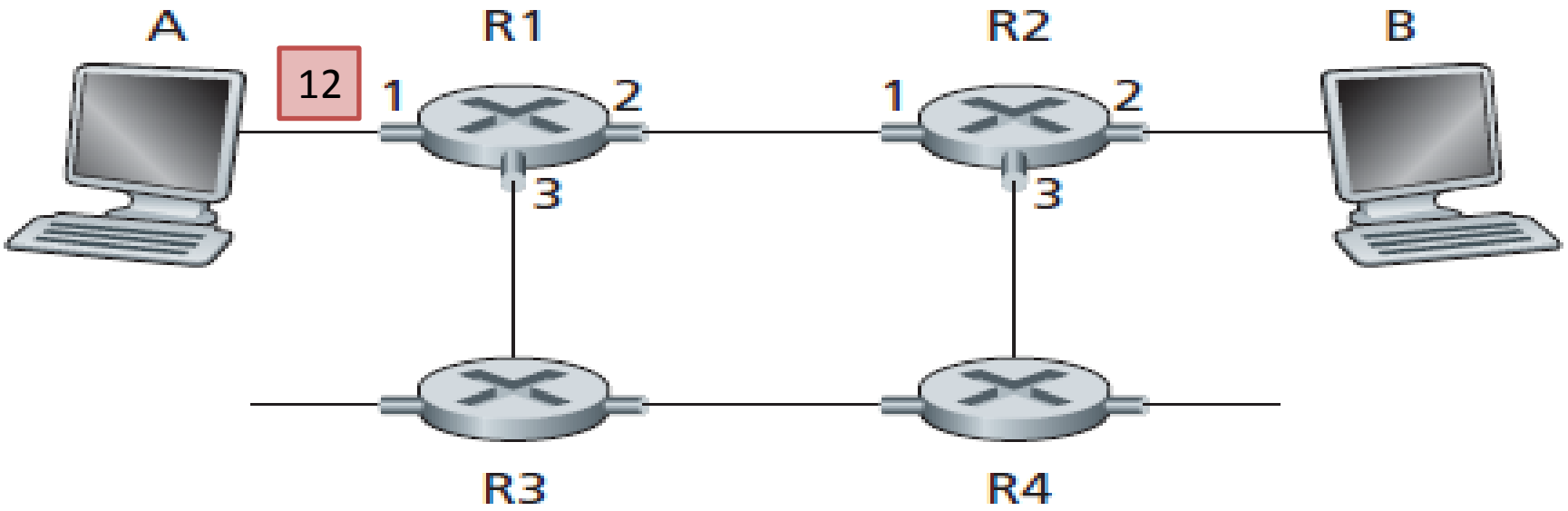
- Computer networks that provide only a connection service at the network layer are called **virtual-circuit (VC) networks**.
- Computer networks that provide only a connectionless service at the network layer are called **datagram networks**.

Virtual-Circuit Networks

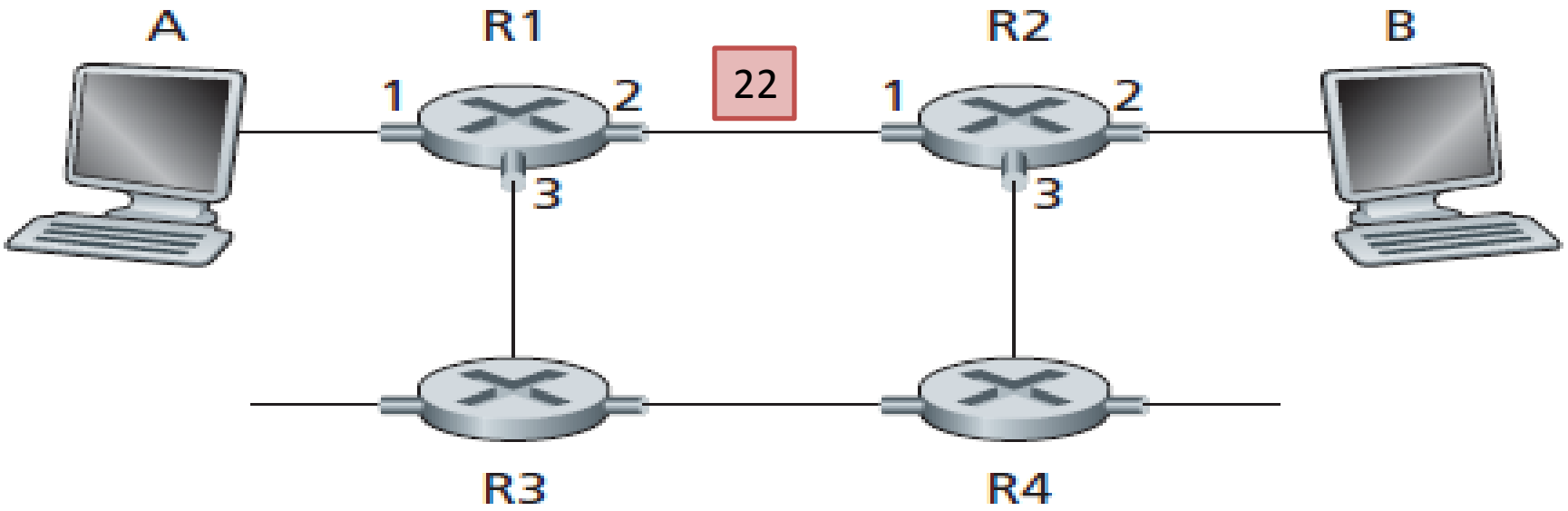
- The network architectures like—ATM and frame relay—are virtual-circuit networks.
- They use connections at the network layer.
- These network-layer connections are called **virtual circuits (VCs).**

VC implementation

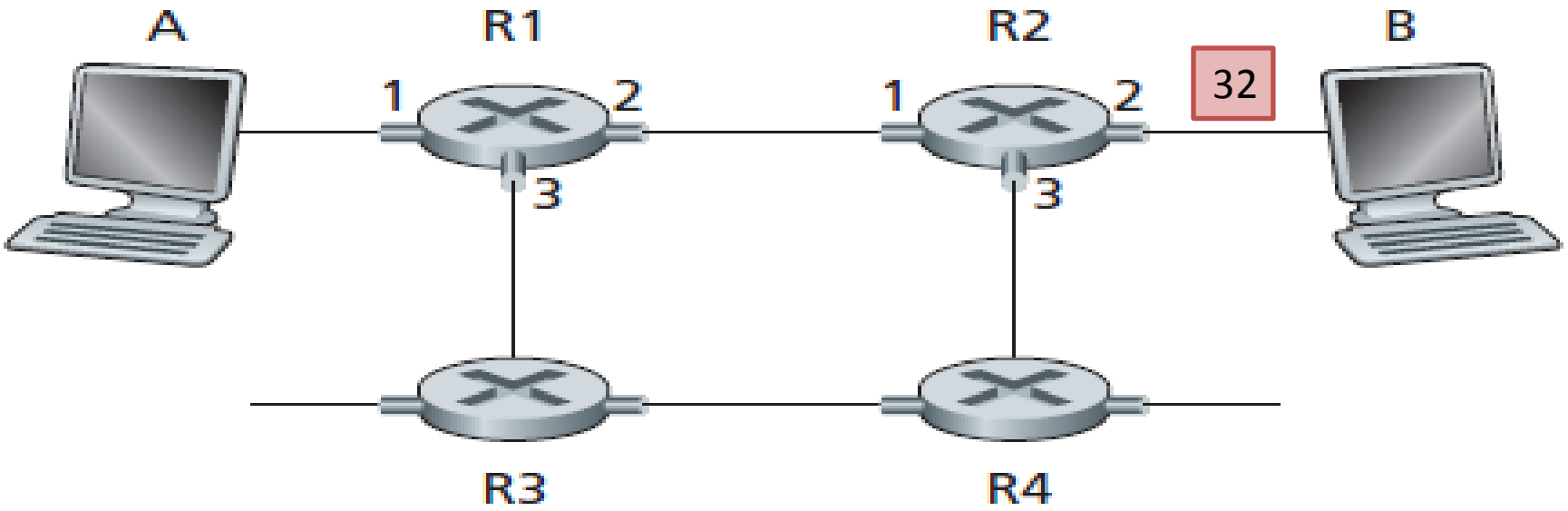
- A VC consists of
  1) A path (that is, a series of links and routers) between the source and destination hosts,
  2) VC numbers, one number for each link along the path, and
  3) Entries in the forwarding table in each router along the path.
- A packet belonging to a virtual circuit will carry a VC number in its header.
- Because a virtual circuit may have a different VC number on each link, each intervening router must replace the VC number of each traversing packet with a new VC number.
- The new VC number is obtained from the forwarding table.



- Host A requests that the network establish a VC between itself and Host B.
- Path A-R1-R2-B
- VC numbers 12, 22, and 32 to the three links.
- When a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12;
- when it leaves R1, the value is 22; and
- when it leaves R2, the value is 32.



- Host A requests that the network establish a VC between itself and Host B.
- Path A-R1-R2-B
- VC numbers 12, 22, and 32 to the three links.
- When a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12;
- when it leaves R1, the value is 22; and
- when it leaves R2, the value is 32.



- Host A requests that the network establish a VC between itself and Host B.
- Path A-R1-R2-B
- VC numbers 12, 22, and 32 to the three links.
- When a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12;
- when it leaves R1, the value is 22; and
- when it leaves R2, the value is 32.

How does the router determine the replacement VC number for a packet?

- For a VC network, each router's forwarding table includes VC number translation.
- Whenever a new VC is established across a router, an entry is added to the forwarding table.
- Similarly, whenever a VC terminates, the appropriate entries in each table along its path are removed.

Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Why a packet doesn't just keep the same VC number on each of the links along its route?

1. Replacing the number from link to link reduces the length of the VC field in the packet header.
2. Simplified VC setup.
   - With multiple VC numbers, each link in the path can choose a VC number independently of the VC numbers chosen at other links along the path.
   - Can avoid exchange and process a number of messages to agree on a common VC number to be used for a connection.

Connection State Information

- In a VC network, the network's routers must maintain **connection state information** for the ongoing connections.
- Each time a new connection is established across a router, a new connection entry must be added to the router's forwarding table; and each time a connection is released, an entry must be removed from the table.
- Even if there is no VC-number translation, it is still necessary to maintain connection state information that associates VC numbers with output interface numbers.

Phases in a Virtual Circuit

- *VC Setup.*
- *Data Transfer.*
- *VC Teardown.*

VC Setup

- During the setup phase, the sending transport layer contacts the network layer, specifies the receiver's address, and waits for the network to set up the VC.
- The network layer determines the path between sender and receiver, that is, the series of links and routers through which all packets of the VC will travel.
- The network layer also determines the VC number for each link along the path.
- Finally, the network layer adds an entry in the forwarding table in each router along the path.
- During VC setup, the network layer may also reserve resources like
  - bandwidth of the links and
  - buffers in the routers.

Data Transfer

- Once the VC has been established, packets can begin to flow along the VC.

VC Teardown

- This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC.
- The network layer will then inform the end system on the other side of the network of the call termination.
- Update the forwarding tables in each of the packet routers on the path to indicate that the VC no longer exists.

Signaling Messages and Protocols

- **Signaling Messages**
  - The messages that the end systems send into the network to initiate or terminate a VC, and
  - The messages passed between the routers to set up the VC (that is, to modify connection state in router tables).
- The protocols used to exchange these messages are often referred to as **signaling protocols.**

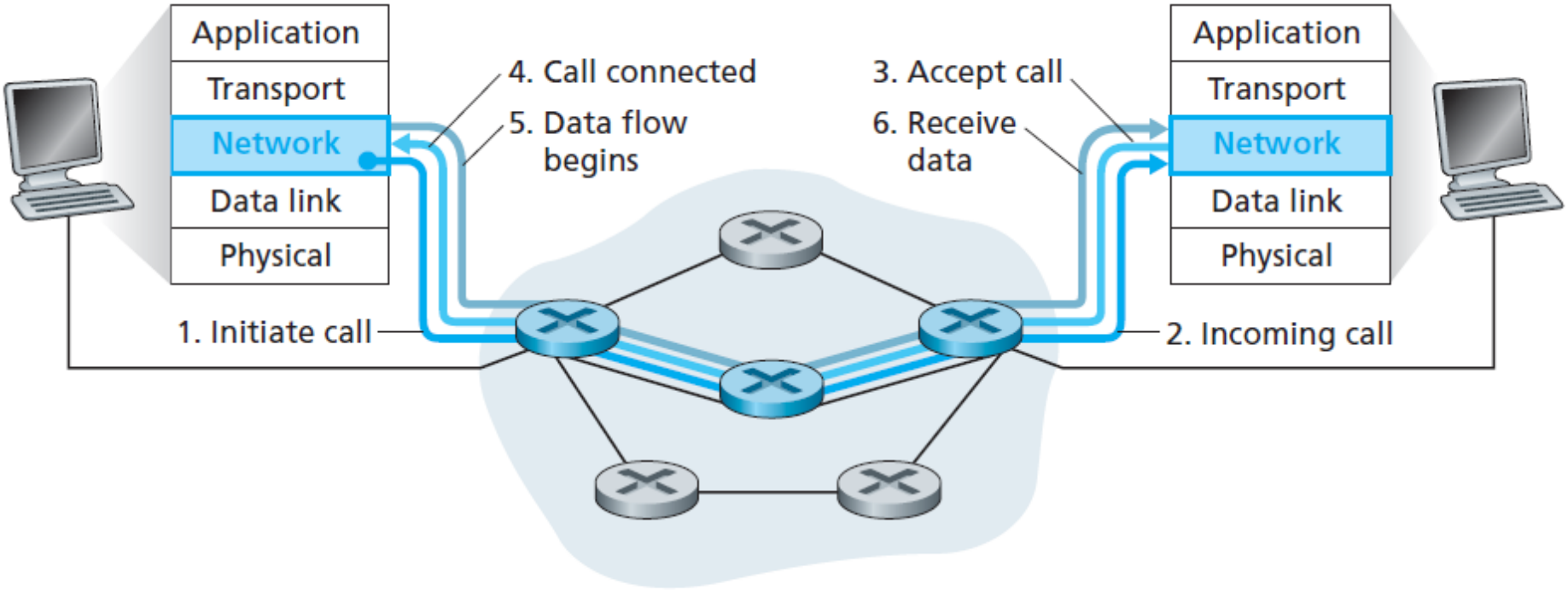


Figure 4.4 ♦ Virtual-circuit setup

Distinction between VC setup at the network layer and connection setup at the transport layer

Connection setup at the transport layer

- Only the two end systems are involved.
- End systems alone determine the parameters of their transport-layer connection.
  - initial sequence number, flow-control window size, etc.
- Routers within the network are completely unaware of it.

VC setup at the network layer

- Routers along the path between the two end systems are involved.
- Each router is fully aware of all the VCs passing through it.

Datagram Networks

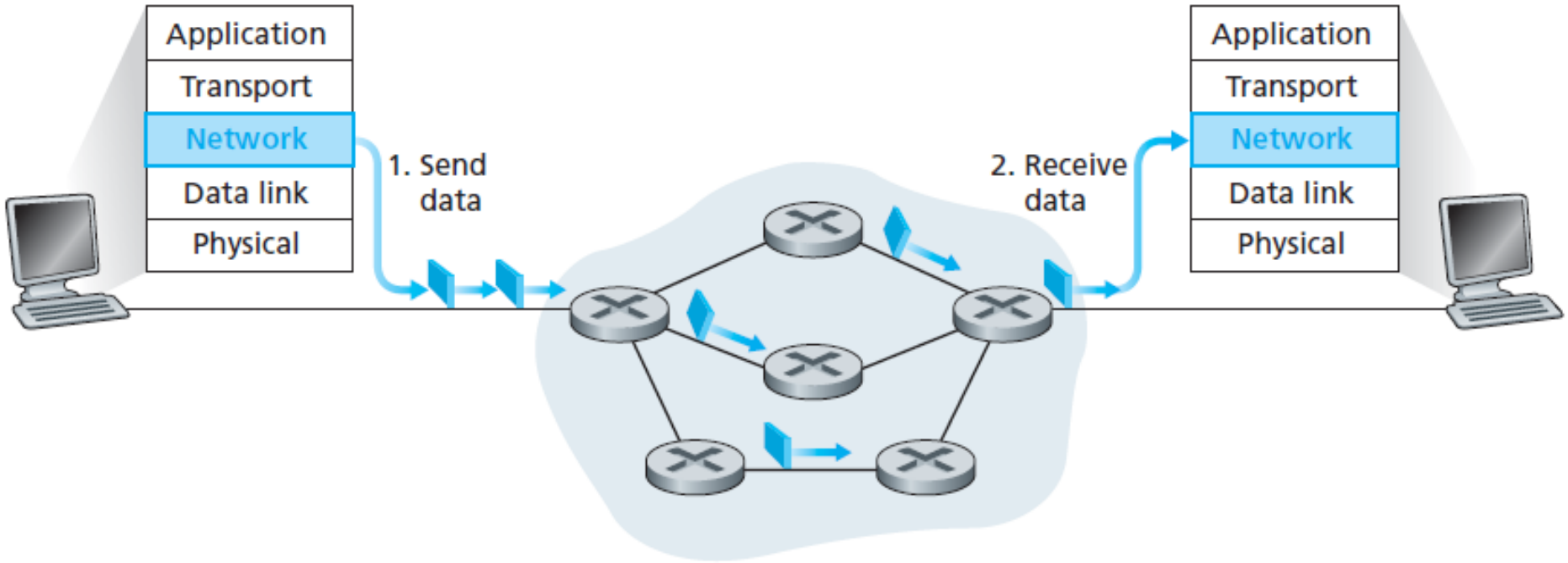
- In a **datagram network,** each time an end system wants to send a packet,
  - it stamps the packet with the address of the destination end system and
  - then pops the packet into the network.
- As a packet is transmitted from source to destination, it passes through a series of routers.
- Each of these routers uses the packet's destination address to forward the packet.

Datagram Networks…

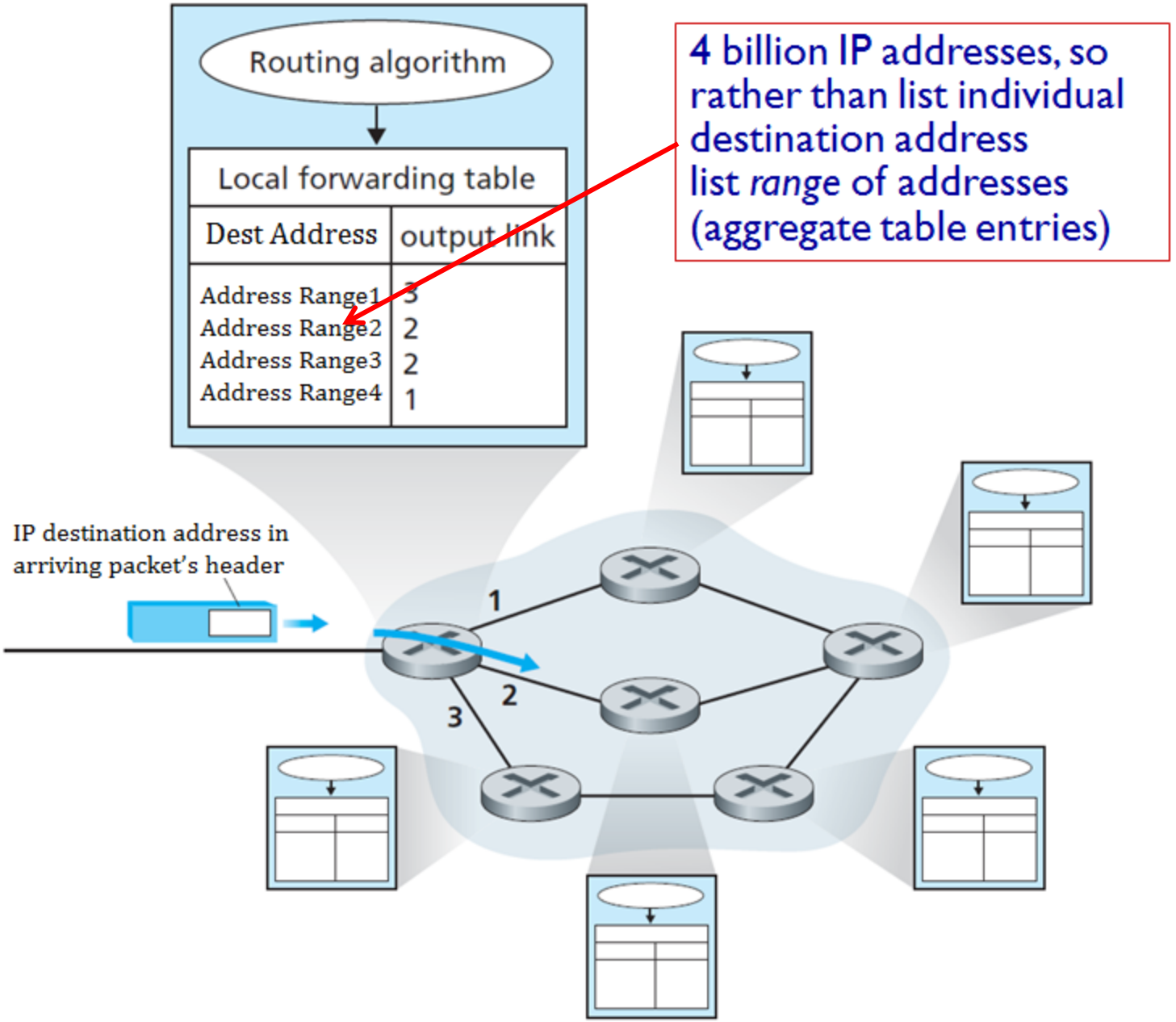
- Each router has a **forwarding table** that maps destination addresses to link interfaces.
- When a packet arrives at the router, the router uses the packet's destination address to look up the appropriate output link interface in the forwarding table.
- The router then intentionally forwards the packet to that output link interface.

Datagram Networks…

- No call setup at network layer.
- Routers: no state about end-to-end connections.
  - No network-level concept of "connection".
- Packets forwarded using destination host address.



Datagram Forwarding Table



Datagram Forwarding Table…

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest Prefix Matching

Longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

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

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

Datagram or VC network: why?

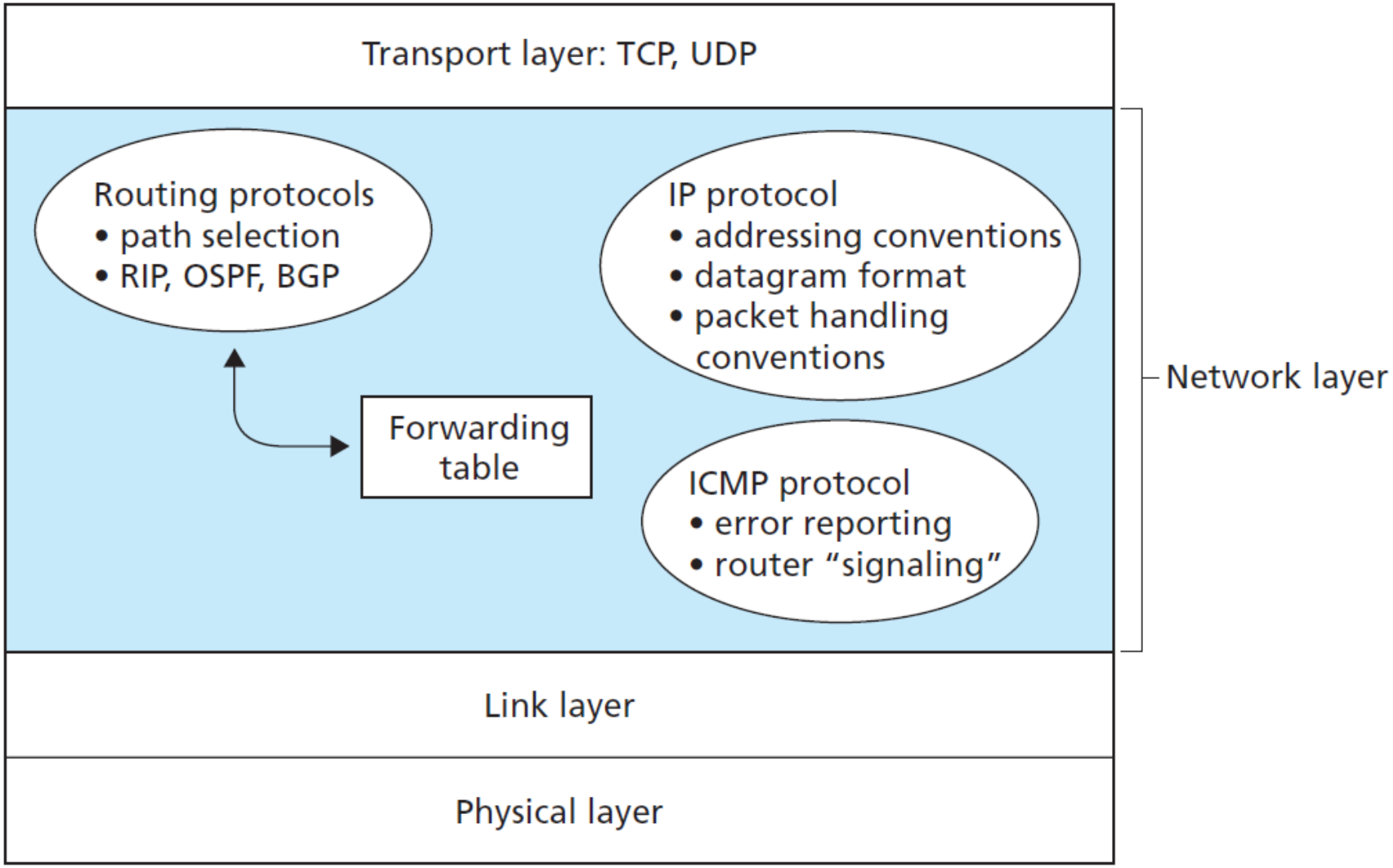
Internet (datagram)

- Data exchange among computers
  - "Elastic" service, no strict timing req.
- Many link types
  - Different characteristics
  - Uniform service difficult
- "Smart" end systems (computers)
  - Can adapt, perform control, error recovery
  - ***Simple inside network, complexity at "edge"***
- Update a forwarding table every one-to-five minutes.
- A series of packets sent from one end system to another may follow different paths through the network and may arrive out of order.

ATM (VC)

- Evolved from telephony
- Human conversation:
  - Strict timing, reliability requirements
  - Need for guaranteed service
- "Dumb" end systems
  - Telephones
  - ***Complexity inside network***
- Whenever a new connection is set up through the router or whenever an existing connection through the router is torn down.
- This could easily happen at a microsecond timescale

A look inside the Internet's network layer!



The Internet Protocol (IP)

- Internet addressing and forwarding are important components of the Internet Protocol (IP).
- There are two versions of IP.
  - IPv4
  - IPv6

IPv4

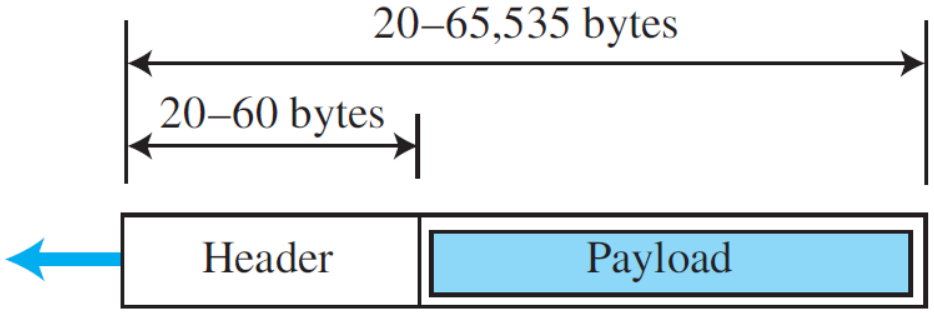
- IPv4 is an unreliable and connectionless datagram protocol—a best-effort delivery service.
- The term best-effort means that IPv4 packets can be
  - corrupted, lost, arrive out of order, or delayed, and may create congestion for the network.
- If reliability is important, IPv4 must be paired with a reliable transport-layer protocol such as TCP.

IPv4…

- IPv4 is also a connectionless protocol for a packet-switched network that uses the datagram approach.
  - Each datagram is handled independently.
  - Each datagram can follow a different route to the destination.
  - Datagrams sent by the same source to the same destination could arrive out of order.
  - Also, some could be lost or corrupted during transmission.
- Again, IPv4 relies on a higher-level protocol to take care of all these problems.

IPv4 Datagram Format

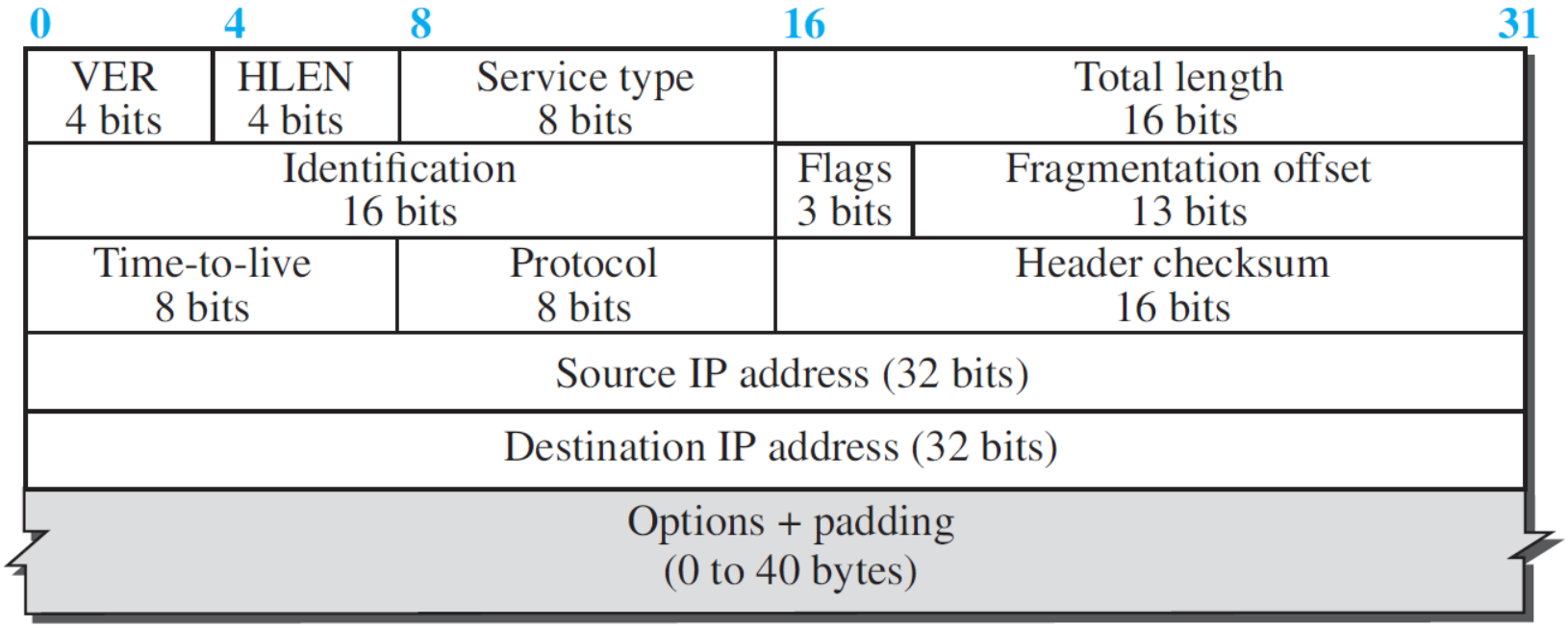
- A network-layer packet is referred to as a *datagram*.



a. IP datagram

Legend

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



b. Header format

The key fields in the IPv4 datagram

- ***Version Number.***
  - The 4-bit version number (VER) field defines the version of the IPv4 protocol, which has the value of 4.
- ***Header Length.***
  - The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words.
  - The IPv4 datagram has a variable-length header.
  - When a device receives a datagram, it needs to know where the header stops and the data starts.
  - The total length of the header is calculated as 4-byte words.
  - The total length is divided by 4 and the value is inserted in the field.
  - The receiver needs to multiply the value of this field by 4 to find the total length.

The key fields in the IPv4 datagram…

- ***Service Type.***
  - In the original design of the IP header, this field was referred to as type of service (TOS), which defined how the datagram should be handled.
  - In the late 1990s, IETF redefined the field to provide *differentiated services* (DiffServ).
- ***Datagram length.***
  - This is the total length of the IP datagram (header plus data), measured in bytes.
  - Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes.
  - However, datagrams are rarely larger than 1,500 bytes.

The key fields in the IPv4 datagram…

- ***Identification, Flags, and Fragmentation Offset.***
  - Fields are related to the fragmentation of the IP datagram when the size of the datagram is larger than the underlying network can carry.
- ***Time-to-live.***
  - The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network.
  - This field is decremented by one each time the datagram is processed by a router.
  - If the TTL field reaches 0, the datagram must be dropped.

The key fields in the IPv4 datagram…

- ***Protocol.***
  - This field is 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.
  - Any protocol that uses the service of IP has a unique 8-bit number which is inserted in the protocol field.
  - When the datagram arrives at the destination, the value of this field helps to define to which protocol the payload should be delivered.

The key fields in the IPv4 datagram…

- ***Header checksum.***
  - The header checksum aids a router in detecting bit errors in a received IP datagram.
  - The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic.
  - The checksum needs to be recalculated at each router.
- ***Source and destination IP addresses.***
  - These 32-bit source and destination address fields define the IP address of the source and destination respectively.
  - The value of these fields must remain unchanged during the time the IP datagram travels from source to destination.

The key fields in the IPv4 datagram…

- ***Options.***
  - A datagram header can have up to 40 bytes of options.
  - Options can be used for network testing and debugging.
- ***Data (payload).***
  - Payload is the packet coming from other protocols that use the service of IP.

IP datagram header size

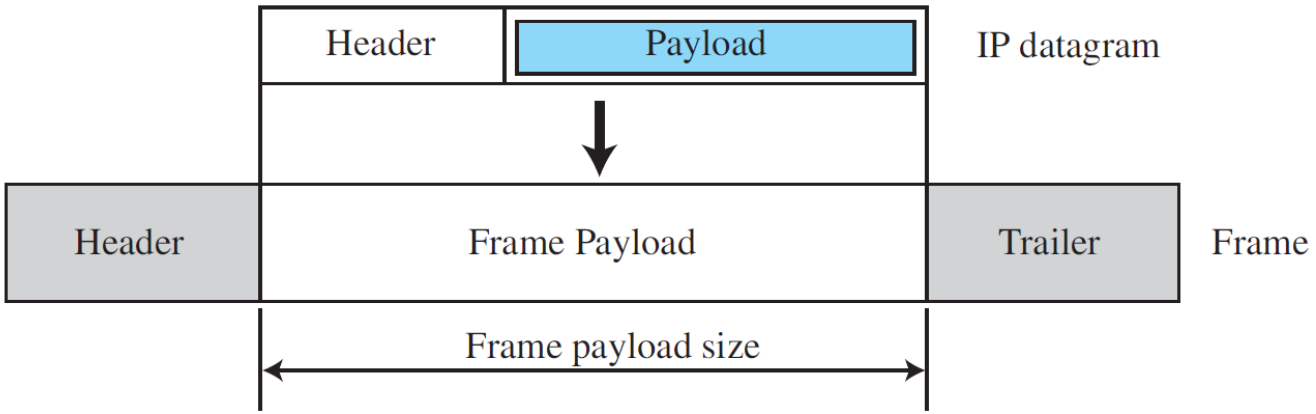
- An IP datagram has a total of 20 bytes of header (assuming no options).
- If the datagram carries a TCP segment, then each datagram carries a total of 40 bytes of header (20 bytes of IP header plus 20 bytes of TCP header) along with the application-layer message.

IP Datagram Fragmentation

- A datagram can travel through different networks.
- Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
  - For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Maximum Transfer Unit (MTU)

- Not all link-layer protocols can carry network-layer packets of the same size.
- Some protocols can carry big datagrams, whereas other protocols can carry only little packets.
  - For example, Ethernet frames can carry up to 1,500 bytes of data, whereas frames for some wide-area links can carry no more than 576 bytes.
- The maximum amount of data that a link-layer frame can carry is called the maximum transmission unit (MTU).



MTU: Maximum size of frame payload

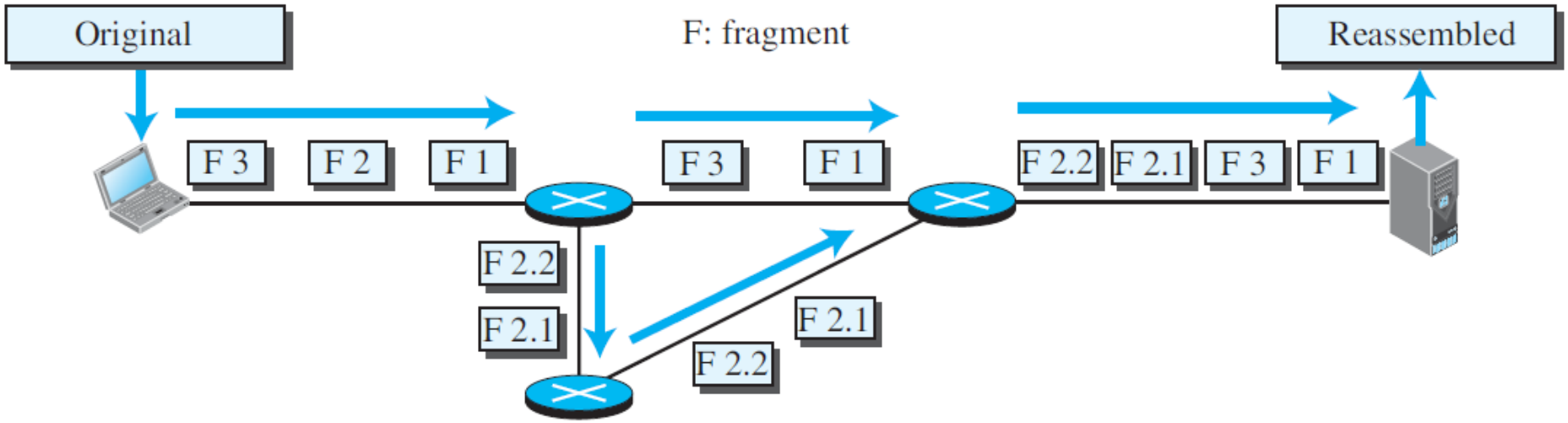
IP Datagram Fragmentation…

- The maximum length of the IP datagram equal to 65,535 bytes.
- The value of the MTU differs from one physical network protocol to another.
- Fragmentation.
  - The process of dividing data in the IP datagram into two or more smaller IP datagrams.
  - Encapsulates each of these smaller IP datagrams in a separate link-layer frame.
  - Each of these smaller datagrams is referred to as a **fragment.**

IP Datagram Fragmentation…

- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some have been changed.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.
- A datagram may be fragmented several times before it reaches the final destination.
- A datagram can be fragmented by the source host or any router in the path.
- The reassembly of the datagram, is done only by the destination host.

IP Datagram Fragmentation…



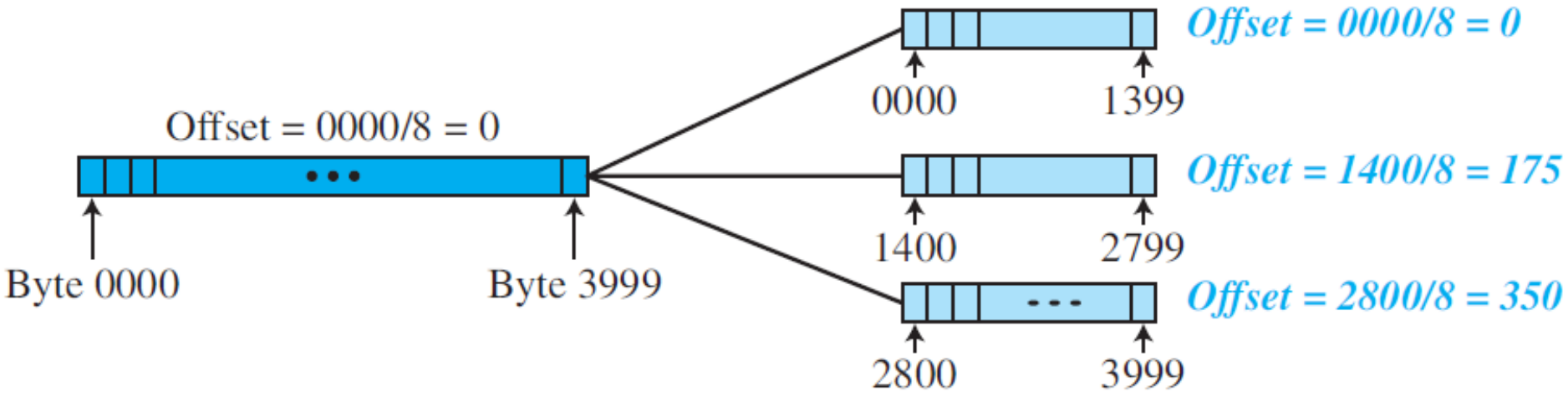
Fields Related to Fragmentation

- *Identification,*
- *Flags*, and
- *Fragmentation offset.*

- ***Identification***
  - The 16-bit identification field identifies a datagram originating from the source host.
  - The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.
  - When a datagram is fragmented, the value in the identification field is copied into all fragments.
  - All fragments have the same identification number, which is also the same as the original datagram.
  - The identification number helps the destination in reassembling the datagram.

- ***Flags***
  - The 3-bit flags field defines three flags.
  - The leftmost bit is reserved (not used).
  - The second bit (D bit) is called the *do not 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.
  - 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.

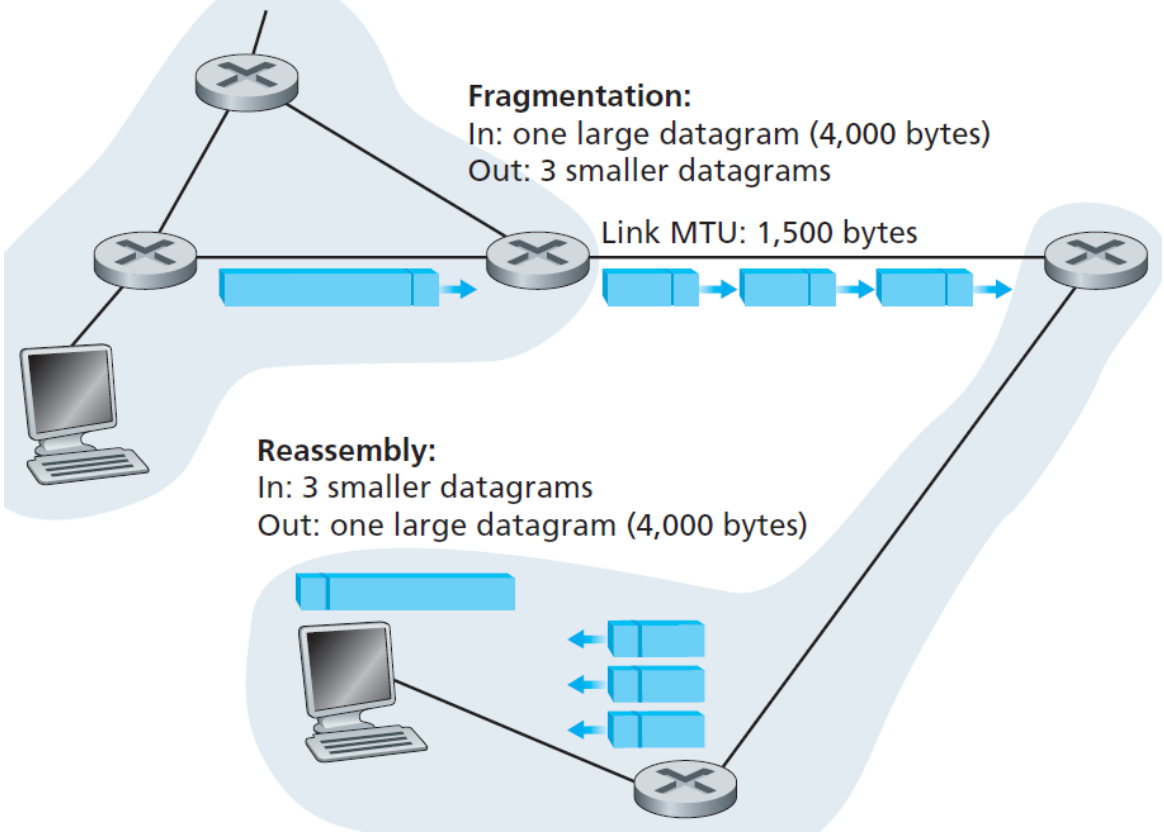
- ***Fragmentation offset***
  - The 13-bit fragmentation offset field shows the relative position of this fragment with respect to the whole datagram.
  - It is the offset of the data in the original datagram measured in units of 8 bytes.
    - Example - a datagram with a data size of 4000 bytes fragmented into three fragments.



Reassembly of fragments

- When the destination receives a series of datagrams from the same sending host, it can examine the identification numbers of the datagrams to determine which of the datagrams are actually fragments of the same larger datagram.
- The last fragment has a flag bit set to 0, whereas all the other fragments have this flag bit set to 1.
- The destination host determines whether a fragment is missing, by using the offset field.

Fragmentation & Reassembly



Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

Merits and demerits of IP Fragmentation

- Merits
  - Gluing together the many disparate link-layer technologies.
- Demerits
  - Complicates routers and end systems.
  - Can be used to create harmful **Denial-of-service** (DoS) attacks.

IPv4 Addressing

- 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.

IPv4 Addressing…

- The boundary between the host and the physical link is called an **interface.**
- A router thus has multiple interfaces, one for each of its links.
- Because every host and router is capable of sending and receiving IP datagrams, IP requires each host and router interface to have its own IP address.

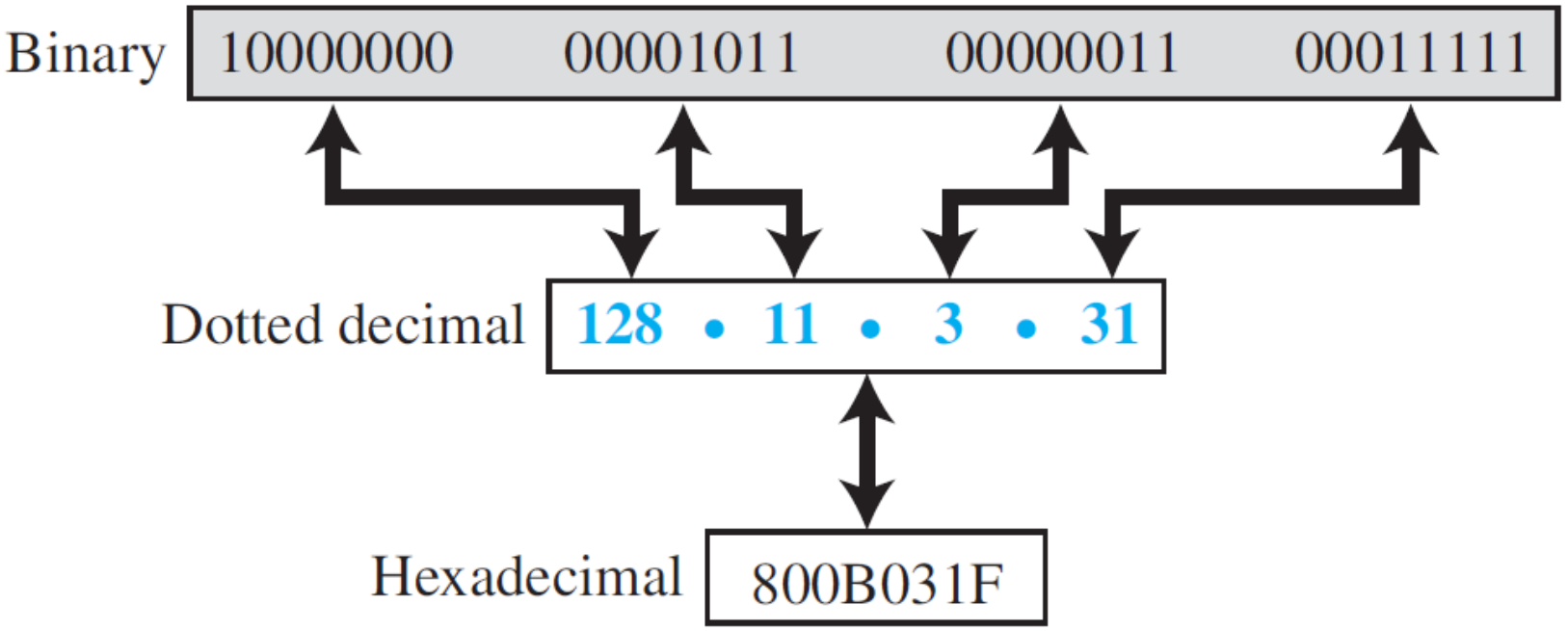
Address Space

- Each IP address is 32 bits long (4 bytes).
- There are a total of 2^{32} possible IP addresses.
- An **address space** is the total number of addresses used by the protocol.
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).

Notation

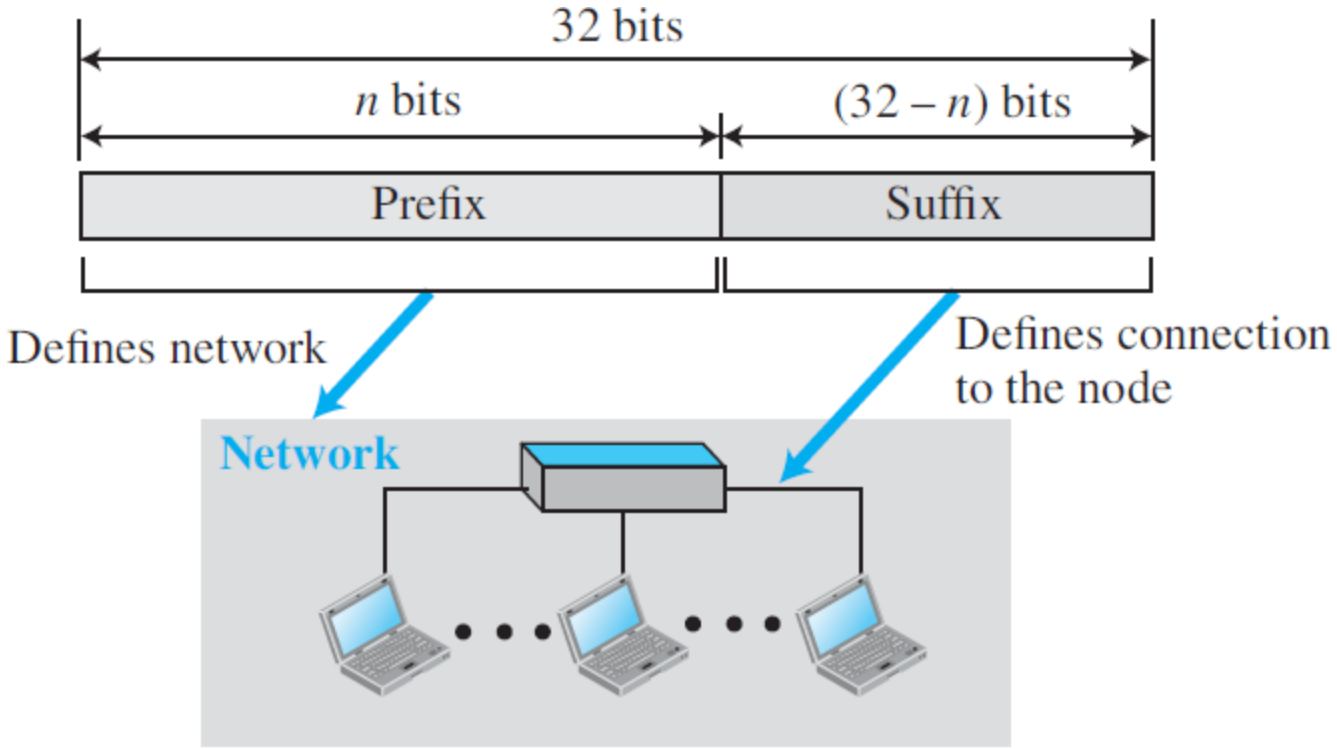
- There are three common notations to show an IPv4 address:
  - Binary notation (base 2)
    - Address is displayed as 32 bits.
    - To make it readable, one or more spaces are inserted between each octet (8 bits).
  - Dotted-decimal notation (base 256)
    - Written in decimal form with a decimal point (dot) separating the bytes.
    - each number in the dotted-decimal notation is between 0 and 255.
  - Hexadecimal notation (base 16)
    - Each hexadecimal digit is equivalent to four bits.
    - A 32-bit address has 8 hexadecimal digits.
    - Often used in network programming.

Three different notations in IPv4 addressing



Hierarchy in Addressing

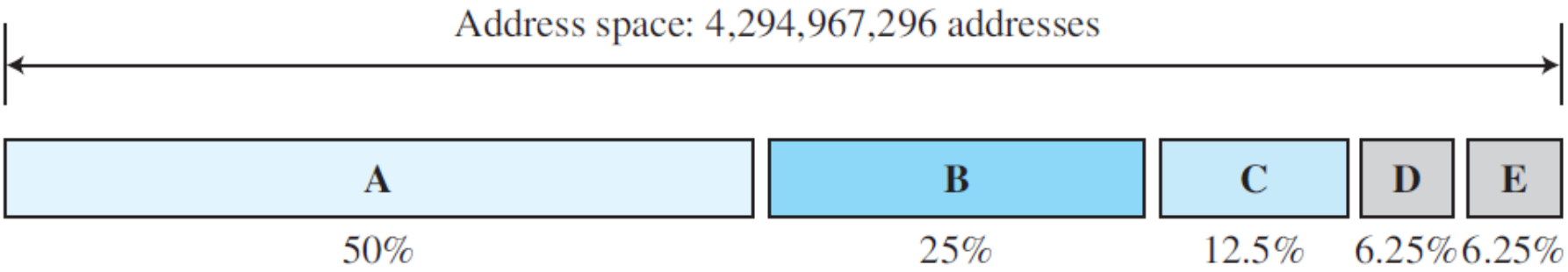
- A 32-bit IPv4 address is also hierarchical, but divided only into two parts.
  - ***Prefix***, defines the network.
  - ***Suffix***, defines the node.

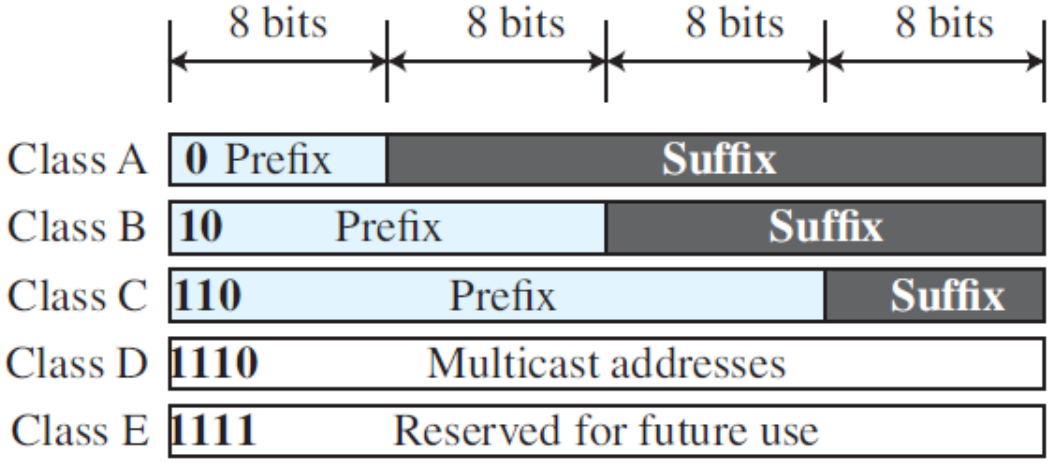


Classful Addressing

- To accommodate both small and large networks, three fixed-length prefixes were designed.
- n = 8, n = 16, and n = 24.
- The whole address space was divided into five classes
- Class A, B, C, D, and E.
- This scheme is referred to as ***classful addressing***.
- Adv:
  - Given an address, we can easily find the class of the address and, the prefix length immediately.

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

Address Depletion

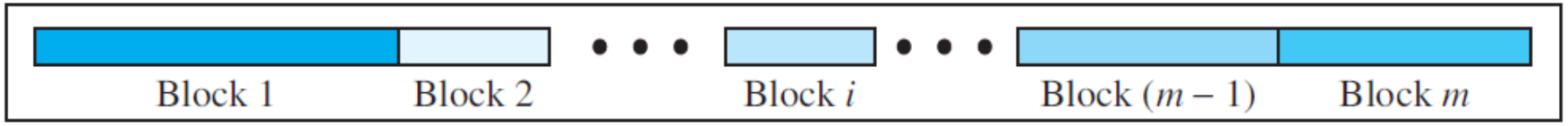
- Since the IP addresses were not distributed properly, the Internet was faced with the problem :
  - *The addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.*
  - Class A can be assigned to only 128 organizations, most of the addresses in this class were wasted (unused).
  - Class B addresses were designed for midsize organization, but many of the addresses in this class also remained unused.
  - In class C, 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 E addresses were almost never used, wasting the whole class.

Subnetting and Supernetting

- To alleviate address depletion, two strategies were proposed:
- Subnetting
  - Class A or class B block is divided into several subnets.
  - Each subnet has a larger prefix length than the original network.
  - Most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.
- Supernetting
  - Used to combine several class C blocks into a larger block.
  - Organizations that need more than the 256 addresses available in a class C block can use this facility.
  - But, routing of packets more difficult.

Classless Addressing

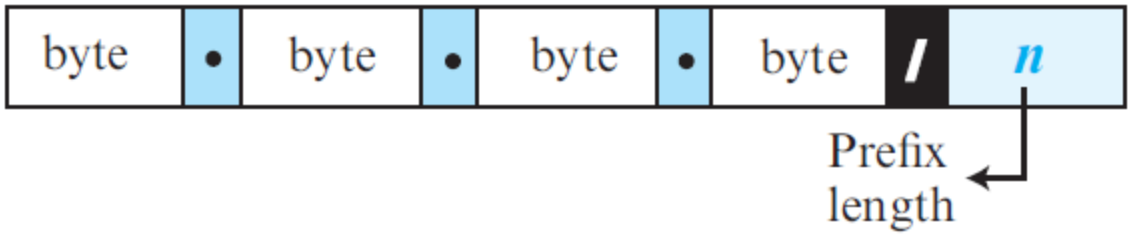
- Another solution to address depletion problem.
- In classless addressing, the whole address space is divided into variable length blocks.
- The prefix in an address defines the block (network); the suffix defines the node (device).
- The number of addresses in a block is a power of 2.
- An organization can be granted one block of addresses.
- The prefix length in classless addressing is variable.
- A small prefix means a larger network; a large prefix means a smaller network.



Address space

Classless Inter Domain Routing (CIDR)

- In classless addressing, 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*** (pronounced ***cider***) strategy.

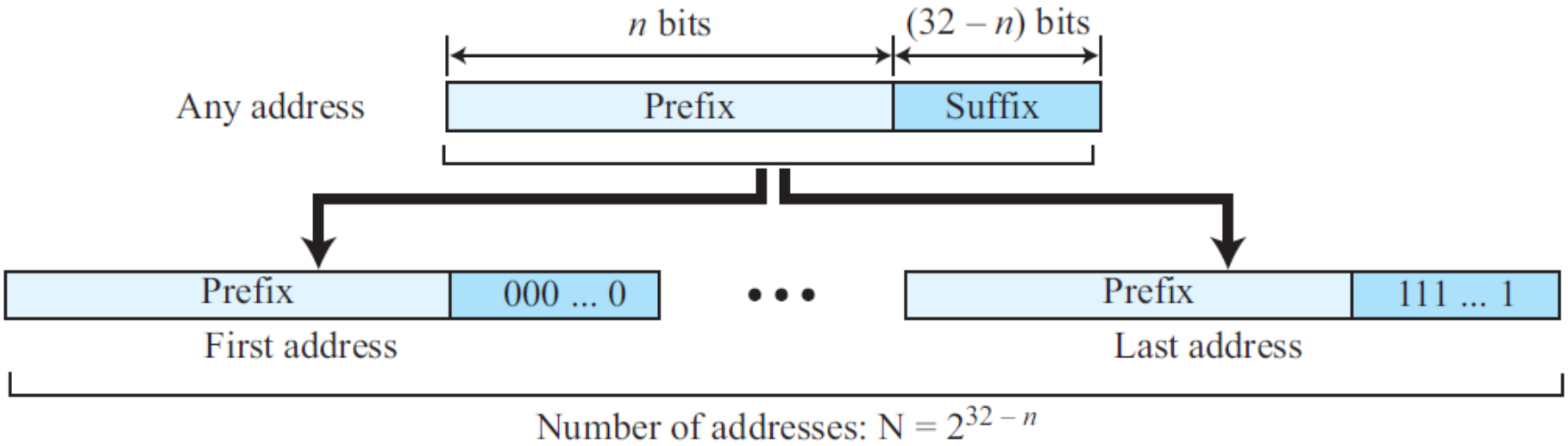


Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

Extracting Information from an Address

- Given any address in the block, we can extract three pieces of information:
  - The number of addresses,
  - The first address in the block, and
  - The last address.
- Procedure : Given the value of prefix length, n.
  1) The number of addresses in the block is found as $N = 2^{32-n}$.
  2) To find the first address, keep the n leftmost bits and set the (32 – n) rightmost bits all to 0s.
  3) To find the last address, we keep the n leftmost bits and set the (32 – n) rightmost bits all to 1s.

Information extraction in classless addressing



Example

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

Example…

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

$$2^{32-n} = 2^5 = 32 \text{ addresses.}$$

- The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.
- 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
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

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 $(2^{32-n} - 1)$.
- A computer program can extract the information in a block, using the mask and the three bit-wise operations NOT, AND, and OR.
  - The number of addresses in the block

    N = NOT (Mask) + 1.
  - The first address in the block

    = (Any address in the block) AND (Mask).
  - The last address in the block

    = (Any address in the block) OR [(NOT (Mask)].

Example

- A classless address is given as 167.199.170.82/27.

  10100111 11000111 10101010 01011111

- To get the mask, the n leftmost bits are set to 1s and the rest of the bits (32−n) are set to 0s.

  11111111 11111111 11111111 11100000

- Mask in dotted-decimal notation –

  255.255.255.224

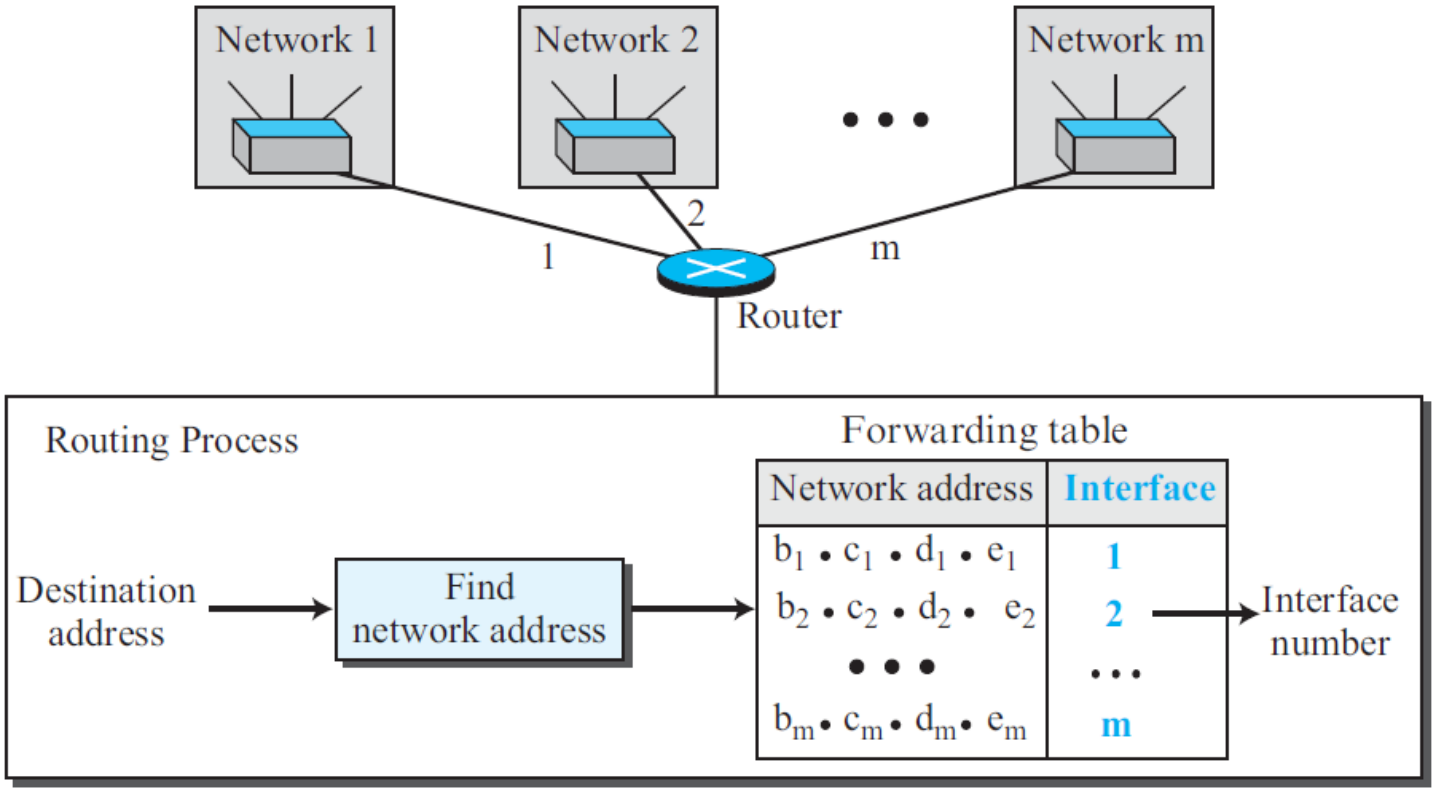
No of addresses: N = NOT (mask) + 1= 0.0.0.31 + 1 = 32 addresses

First address: First = (address) AND (mask) = 167.199.170. 82

Last address: Last = (address) OR (NOT mask) = 167.199.170. 255

Network Address

- The network address is actually the identifier of the network;
- Each network is identified by its network address.



Block Allocation

- A global authority called the Internet Corporation for Assigned Names and Numbers (ICANN) has the responsibility of block allocation.
- It assigns a large block of addresses to an ISP.
  - ICANN does not allocate addresses to individual Internet users.
- Two restrictions:
  1. The number of requested addresses, N, needs to be a power of 2.
  2. The first address needs to be divisible by the number of addresses in the block.
  - The decimal value of the first address is :

$$\text{first address} = (\text{prefix in decimal}) \times 2^{32-n} = (\text{prefix in decimal}) \times \text{N}.$$

Example

- 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.

00010010 00001110 00001100 00000000

- It can be seen that the first address in decimal is 302,910,464, which is divisible by 1024.

Subnetting

- An organization (or an ISP) that is granted a range of addresses may divide the range into several sub-ranges and assign each sub-range to a subnetwork (or subnet).
- A subnetwork can be divided into several sub-subnetworks.

Designing Subnets

- Assume
  - N : total number of addresses granted to the organization
  - n : prefix length
  - N_{sub} : number of addresses to each subnetwork
  - n_{sub} : prefix length for each subnetwork
- Steps
  1) The number of addresses in each subnetwork should be a power of 2.
  2) The prefix length for each subnetwork should be found using the following formula:

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

  3) 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

- 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.
- Designing subblock of 120 addresses
  - 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

$$n_1 = 32 - \log_2 128 = 25.$$

  - The first address in this block is 14.24.74.0/25;
  - The last address is 14.24.74.127/25.

Solution…

- Designing subblock of 60 addresses
  - 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.

Solution…

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

$$n_3 = 32 - \log_2 16 = 28.$$

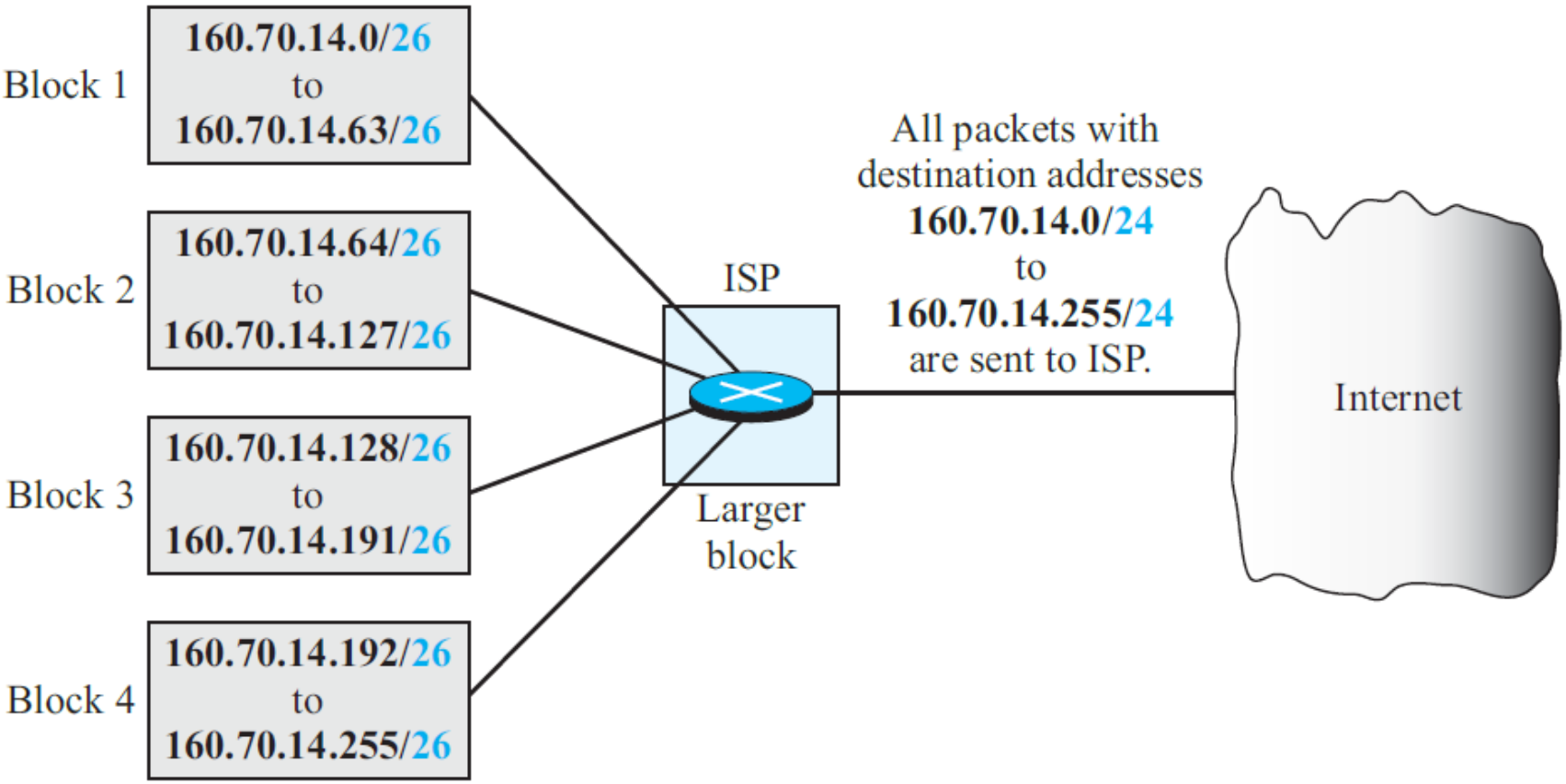
  - The first address in this block is 14.24.74.192/28;
  - The last address is 14.24.74.207/28.

- If we add all addresses in the previous subblocks, the result is 208 addresses(128+64+16), which means 48 addresses are left in reserve.

Address Aggregation

- ***Address Aggregation*** is one of the advantages of the CIDR strategy.
  - Also 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.

Address Aggregation…



Special Addresses

- Five special addresses are used for special purposes:
  - This-host address,
  - Limited-broadcast address,
  - Loopback address,
  - Private addresses, and
  - Multicast addresses.

- *This-host Address*
  - The only address in the block 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.*
  - The only address in the block 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.

- *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.
  - For example,
    - we can write a client and a server program in which one of the addresses in the block is used as the server address.
    - We can test the programs using the same host to see if they work before running them on different computers.

- *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.

Routing Algorithms

- When a packet arrives to a router, the router indexes a forwarding table and determines the link interface to which the packet is to be directed.
- Routing algorithms, operating in network routers, exchange and compute the information that is used to configure these forwarding tables.
- The network layer must determine the path that packets take from senders to receivers.
- The job of routing is to determine good paths from senders to receivers, through the network of routers.

Routing Algorithms…

- A host is attached directly to one router, the **default router** for the host (also called the **first-hop router** for the host).
- Whenever a host sends a packet, the packet is transferred to its default router.
- The default router of the source host as the **source router** and the default router of the destination host as the **destination router.**
- The purpose of a routing algorithm:
  - Given a set of routers, with links connecting the routers, a routing algorithm finds a "good" path from source router to destination router.
  - A **good path** is one that has the **least cost.**

Classification of Routing Algorithms

1. Global(centralized) or Decentralized
2. Static or Dynamic
3. Load-sensitive or Load-insensitive.

Classification of Routing Algorithms…

- **Global/centralized algorithm**
  - Computes the least-cost path between a source and destination using complete, global knowledge about the network.
  - Has complete information about connectivity and link costs.
  - Algorithms with global state information are often referred to as **link-state (LS) algorithms**, since the algorithm must be aware of the cost of each link in the network.

Classification of Routing Algorithms…

- **Decentralized algorithm**
  - The calculation of the least-cost path is carried out in an iterative, distributed manner.
  - 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.
  - Calculate and exchange the information with its neighboring nodes, a node gradually calculates the least-cost path to a destination or set of destinations.
  - The decentralized routing algorithm is called a **distance-vector (DV) algorithm**, because each node maintains a vector of estimates of the costs (distances) to all other nodes in the network.

Classification of Routing Algorithms…

- **Static routing algorithms**
  - Routes change very slowly over time, often as a result of human intervention (for example, a human, manually editing a router's forwarding table).
- **Dynamic routing algorithms**
  - Change the routing paths as the network traffic loads or topology change.
  - A dynamic algorithm can be run either periodically or in direct response to topology or link cost changes.
  - While dynamic algorithms are more responsive to network changes, they are also more susceptible to problems such as routing loops and oscillation in routes.

Classification of Routing Algorithms…

- **Load-sensitive Algorithm**
  - Link costs vary dynamically to reflect the current level of congestion in the underlying link.
  - If a high cost is associated with a link that is currently congested, a routing algorithm will tend to choose routes around such a congested link.
  - E.g. ARPAnet routing algorithms
- **Load-insensitive Algorithm**
  - A link's cost does not explicitly reflect its current (or recent past) level of congestion.
  - E.g. RIP, OSPF, BGP

The Link-State (LS) Routing Algorithm

- The network topology and all link costs are known.
- Each node broadcast ***link-state packets*** to *all other nodes* in the network.
- Each link-state packet containing the identities and costs of its attached links.
- All nodes have an identical and complete view of the network.
- Each node can then run the LS algorithm and compute the same set of least-cost paths as every other node.
- **Dijkstra's algorithm** computes the least-cost path from one node to all other nodes in the network.

The Link-State (LS) Routing Algorithm…

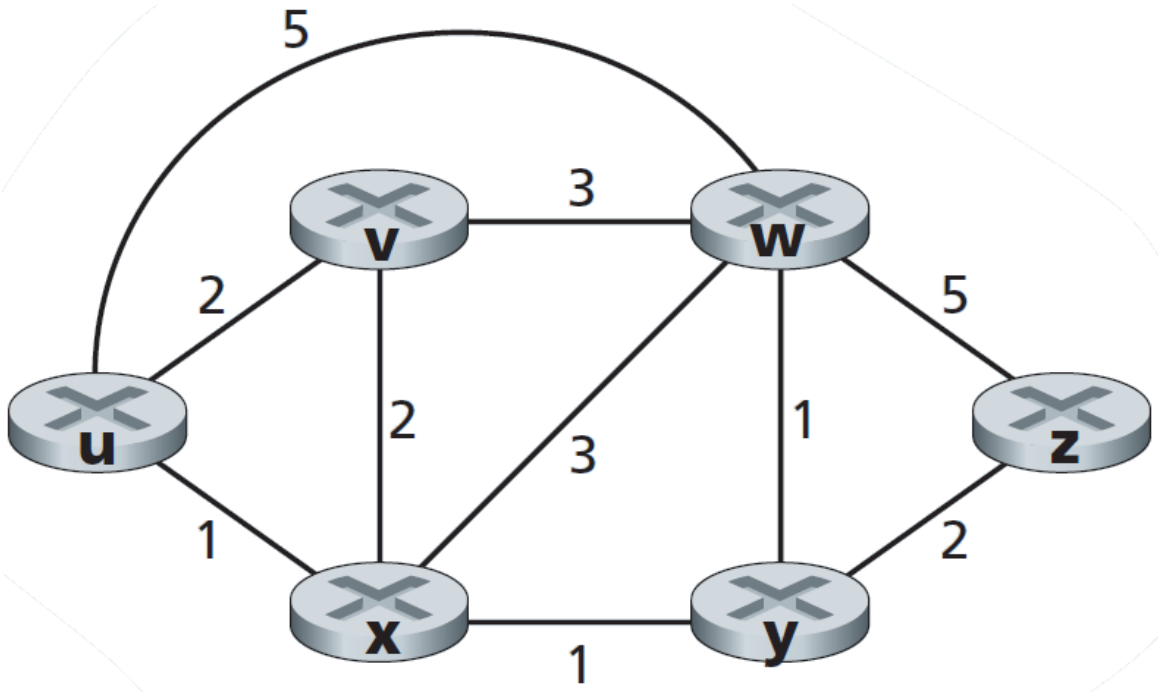
- Dijkstra's algorithm is iterative and has the property:
  - "After the *kth iteration* of the algorithm, the *least-cost paths* are known to *k* destination nodes, and among the least-cost paths to all destination nodes, these *k* paths will have the *k smallest costs.*"
- Notations :
  - ***D(v):*** cost of the least-cost path from the source node to destination *v* as of this iteration of the algorithm.
  - ***p(v):*** previous node (neighbor of v) along the current least-cost path from the source to *v*.
  - ***N':*** subset of nodes; *v* is in N' if the least-cost path from the source to *v* is definitively known.

The Link-State (LS) Routing Algorithm…

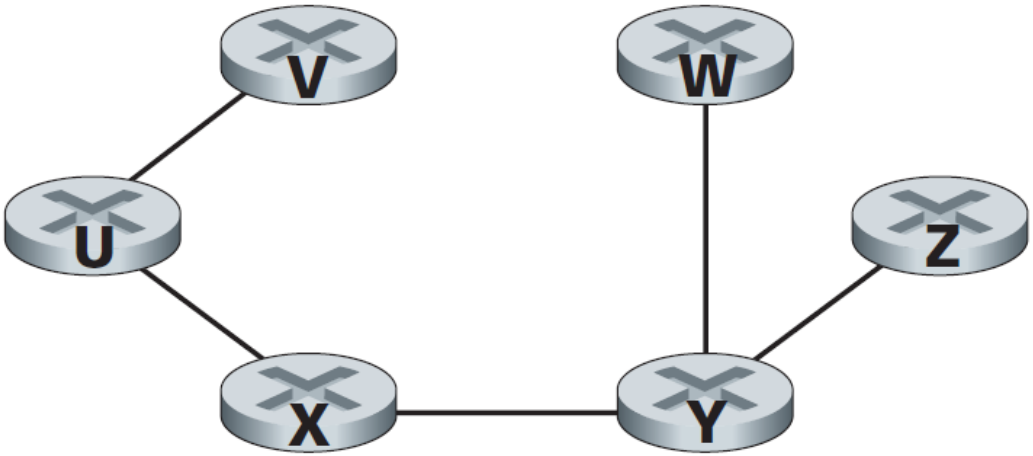
- The global routing algorithm consists of an initialization step followed by a loop.
- The number of times the loop is executed is equal to the number of nodes in the network.
- Upon termination, the algorithm will have calculated the shortest paths from the source node *u* to every other node in the network.

Link-State (LS) Algorithm for Source Node *u*

```
1   Initialization:
2     N' = {u}
3     for all nodes v
4       if v is a neighbor of u
5         then D(v) = c(u,v)
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 each neighbor v of w and not in N':
12          D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     least path cost to w plus cost from w to v */
15  until N'= N
```



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					



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Fig : Least cost path and forwarding table for nodule u

The Distance-Vector (DV) Routing Algorithm

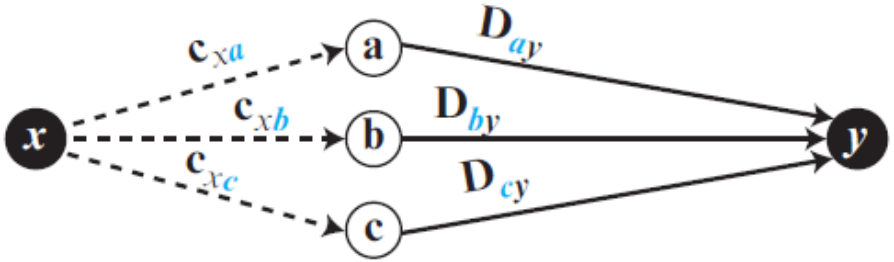
- The **distance vector** (**DV) algorithm** is iterative, asynchronous, and distributed.
- ***Distributed:*** each node receives some information from one or more of its *directly attached neighbors*, performs a calculation, and then distributes the results of its calculation back to its neighbors.
- ***Iterative:*** this process continues on until no more information is exchanged between neighbors.
- ***Asynchronous:*** it does not require all of the nodes to operate in lockstep with each other.

The Distance-Vector (DV) Routing Algorithm…

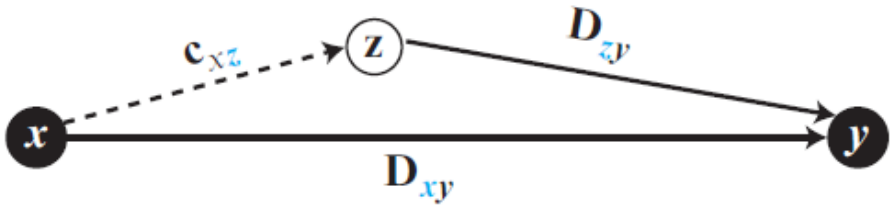
- Let $d_x(y)$ be the cost of the least-cost path from node *x* to node *y.*
- Then the least costs are related by the Bellman-Ford equation:

$$d_x(y) = min_v\{c(x,v) + d_v(y)\}$$

- where the min_v is taken over all of x's neighbors.
- After traveling from *x to v,* if we then take the least-cost path from *v to y, the path cost will be* $c(x,v) + d_v(y)$.
- Since we must begin by traveling to some neighbor v, the least cost from x to y is the minimum of $c(x,v) + d_v(y)$ taken over all neighbors v.



a. General case with three intermediate nodes



b. Updating a path with a new route

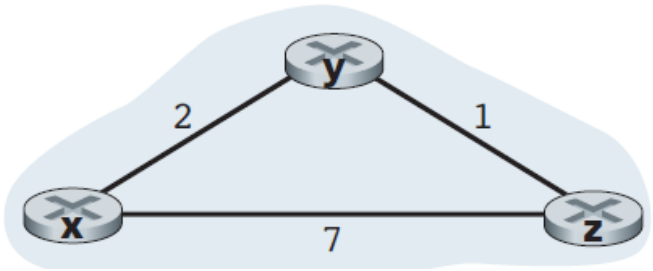
The Distance-Vector (DV) Routing Algorithm…

- With the DV algorithm, each node *x* maintains the following routing information:
  - For each neighbor *v, the cost c(x,v)* from x to directly attached neighbor, *v*
  - Node *x's distance vector,* that is, $\boldsymbol{D_x = [D_x(y): y \text{ in } N]}$, containing x's estimate of its cost to all destinations, *y, in N*
  - The distance vectors of each of its neighbors, that is, $\boldsymbol{D_v = [D_v(y): y \text{ in } N]}$ for each neighbor *v of x*

Distance-Vector (DV) Algorithm

At each node, x:

```
1  Initialization:
2     for all destinations y in N:
3        D_x(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        D_w(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector D_x = [D_x(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       D_x(y) = min_v{c(x,v) + D_v(y)}
15
16    if D_x(y) changed for any destination y
17       send distance vector D_x = [D_x(y): y in N] to all neighbors
18
19 forever
```



Node x table

from	cost to x	cost to y	cost to z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Node y table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Node z table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	3	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Time

Count to Infinity / Instability problem

- A problem with distance-vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) will propagate slowly.
- For a routing protocol to work properly, if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance-vector routing, this takes some time.
- The problem is referred to as count to infinity.
- It sometimes takes several updates before the cost for a broken link is recorded as infinity by all routers.

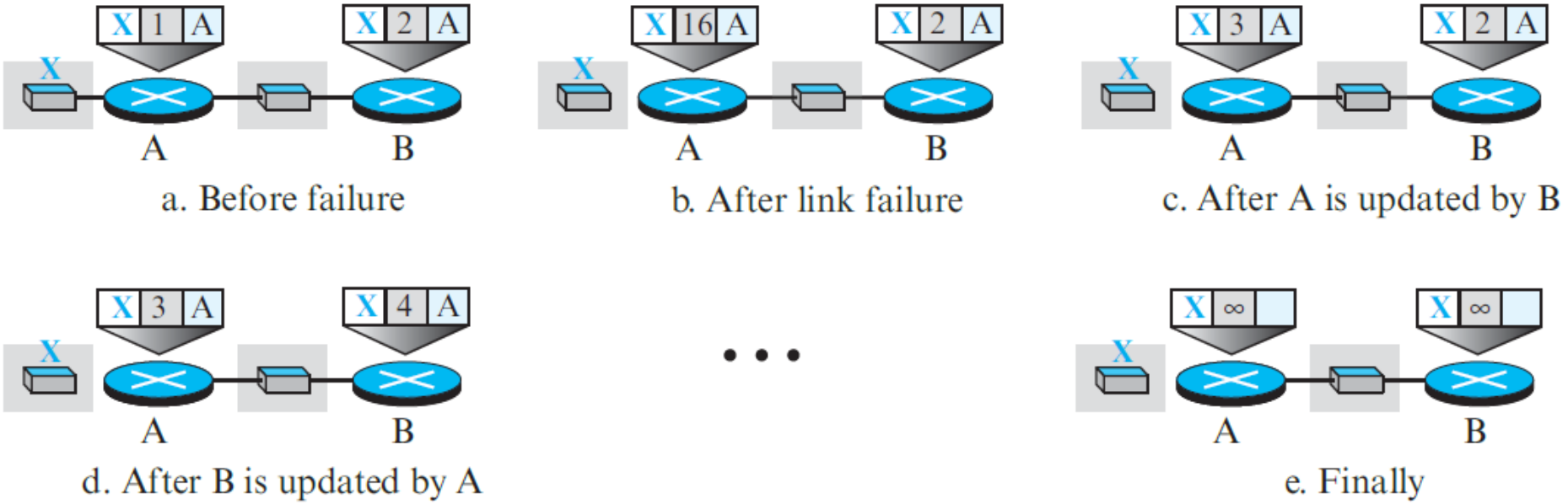


Fig : Two-node instability

Solutions

- Split Horizon:
  - Instead of flooding the table through each interface, each node sends only part of its table through each interface.
  - If node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows).
- Poisoned Reverse:
  - In this strategy B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: “Do not use this value; what I know about this route comes from you.”

Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network "flat"

… *not* true in practice

Scale: with 600 million destinations:

- Can't store all dest's in routing tables!
- Routing table exchange would swamp links!

Administrative autonomy

- Internet = network of networks
- Each network admin may want to control routing in its own network

Hierarchical Routing…

Autonomous Systems (AS)

- Aggregate routers into regions.
- A group of routers that are typically under the same administrative control.

Intra-AS routing protocol

- Intra-Domain Routing
- Routers in same AS run same routing protocol.
- Routers in different AS can run different intra-AS routing protocol.

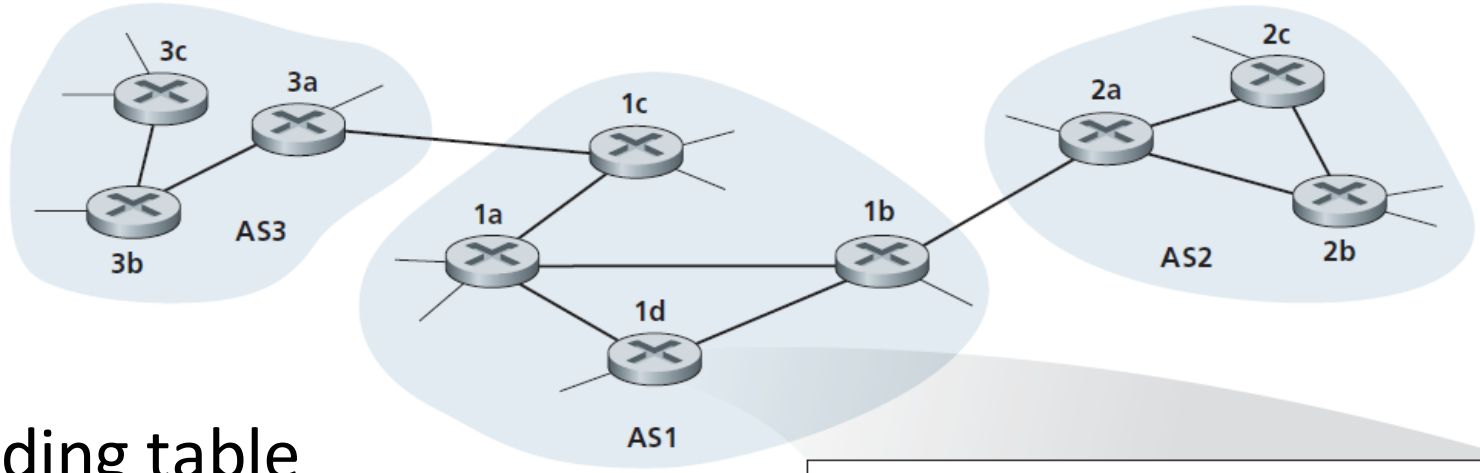
Gateway Router:

- At "edge" of its own AS.
- Has link to router in another AS.

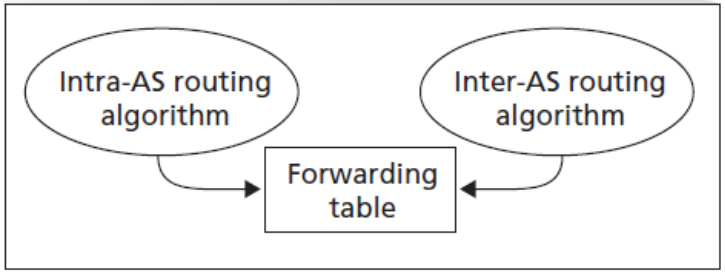
Inter-AS routing protocol.

- Inter-Domain Routing
- Involves communication between two ASs,
- The two communicating ASs must run the same inter-AS routing protocol.

Interconnected Autonomous Systems



- Forwarding table configured by both intra- and inter-AS routing algorithm.
  - Intra-AS sets entries for internal destinations
  - Inter-AS & intra-AS sets entries for external destinations.



Inter-AS Tasks

- 2 tasks :
  - Obtaining reachability information from neighboring ASs and
  - Propagating the reachability information to all routers internal to the AS.

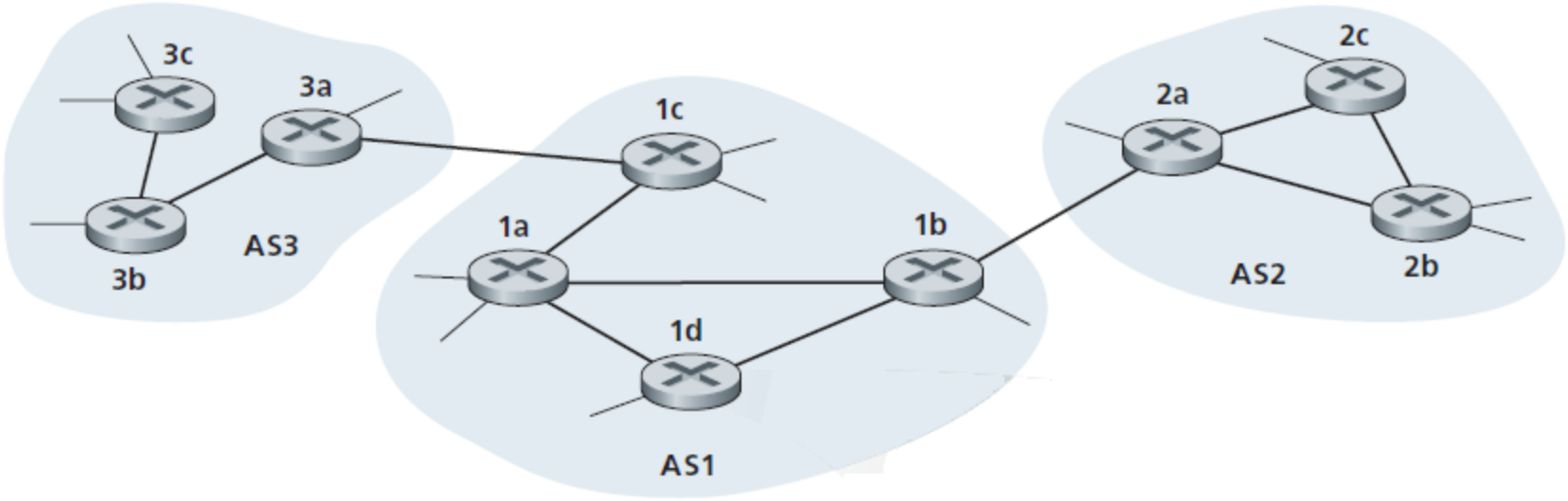
Inter-AS Tasks…

- Suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

AS1 must:

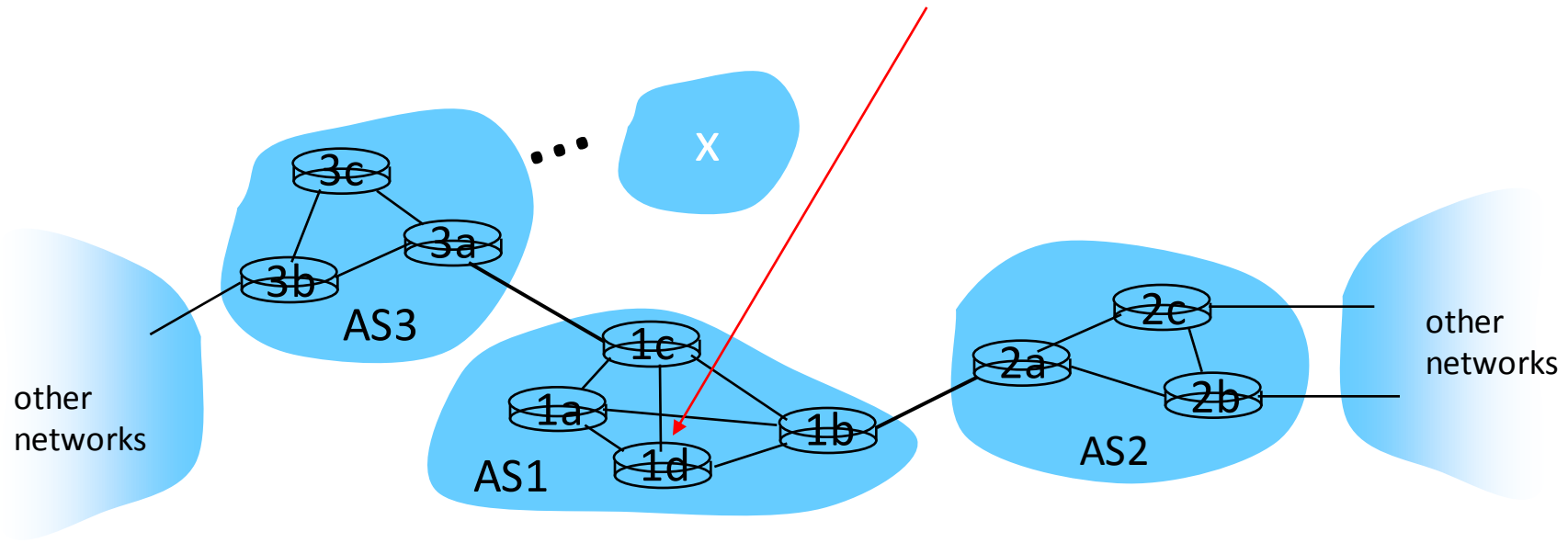
1. Learn which destinations are reachable through AS2, which through AS3.
2. Propagate this reachability info to all routers in AS1.

job of inter-AS routing!



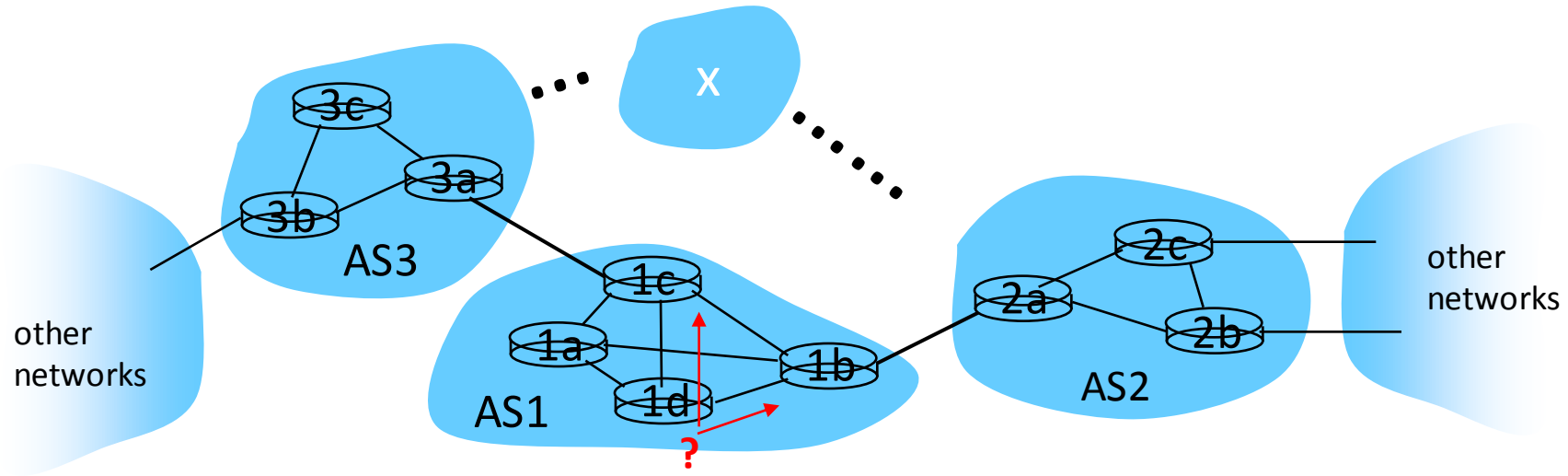
Example: setting forwarding table in router 1d

- Suppose AS1 learns (via inter-AS protocol) that subnet *x* reachable via AS3 (gateway 1c), but not via AS2.
  - Inter-AS protocol propagates reachability info to all internal routers.
- Router 1d determines from intra-AS routing info that its interface *I* is on the least cost path to 1c.
  - installs forwarding table entry *(x,I)*



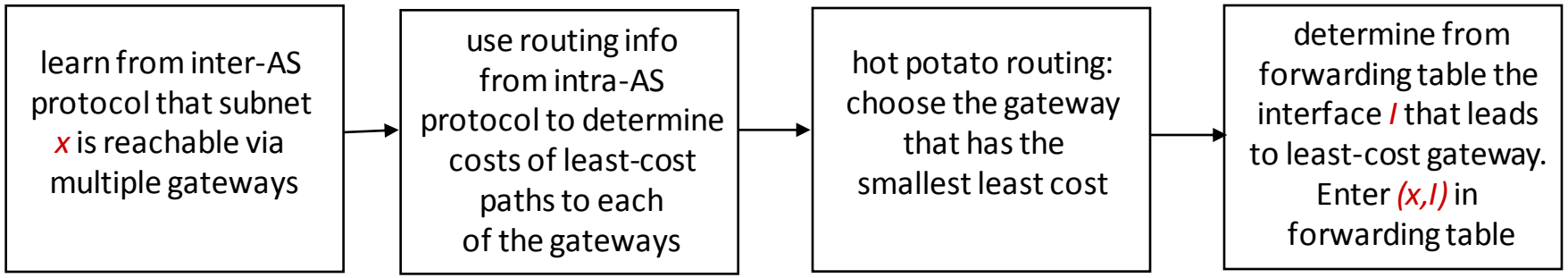
Example: choosing among multiple ASes

- Now suppose AS1 learns from inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.
- To configure forwarding table, router 1d must determine which gateway it should forward packets towards for dest x.



Example: choosing among multiple Ases…

- *Hot potato routing:* send packet towards closest of two routers.



Routing on the Internet

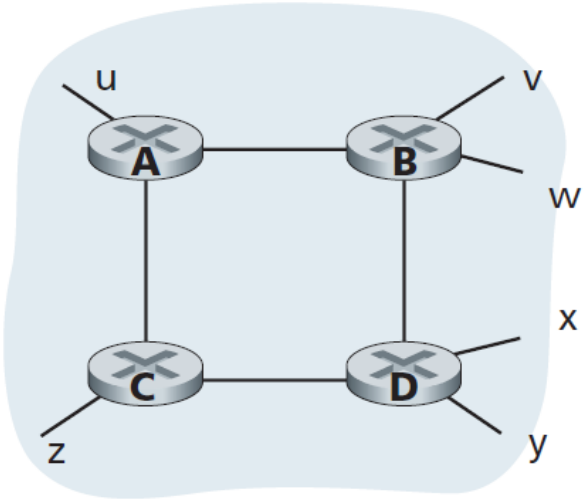
- **Intra-AS Routing protocols**
  - also known as *interior gateway protocols (IGP)*
  - 2 types:
    - **Routing Information Protocol (RIP)**
    - **Open Shortest Path First (OSPF)**
- **Inter-AS Routing protocol**
  - also known as *exterior gateway protocol (EGP)*
  - The only inter-domain routing protocol is
    - **Border Gateway Protocol (BGP)**

Routing Information Protocol (RIP)

- Based on the distance-vector routing algorithm.
- Started as part of the Xerox Network System (XNS).
- Included in BSD-UNIX distribution in 1982.
- RIP uses the term ***hop***, which is the number of subnets traversed along the shortest path from source router to destination subnet, including the destination subnet.
- In RIP, the maximum cost of a path can be 15, which means 16 is considered as infinity (no connection).

Routing Information Protocol (RIP)…

- In RIP, routing updates are exchanged between neighbors every 30 seconds using a **RIP response message.**
- The response message contains
  - a list of up to 25 destination subnets within the AS,
  - the sender's distance to each of those subnets.
- Response messages are also known as **RIP advertisements.**

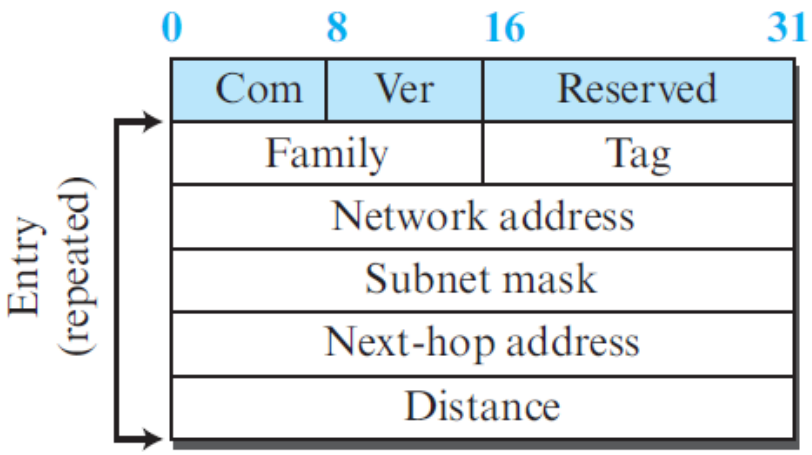


Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP Messages

- RIP has two types of messages:
  - request and response.
- A request message is sent by a router that has just come up or by a router that has some time-out entries.
  - A request message can ask about specific entries or all entries.
- A response (or update) message can be either solicited or unsolicited.
  - A solicited response message is sent only in answer to a request message.
    - It contains information about the destination specified in the corresponding request message.
  - An unsolicited response message, is sent periodically, every 30 seconds or when there is a change in the forwarding table.

RIP Message Format

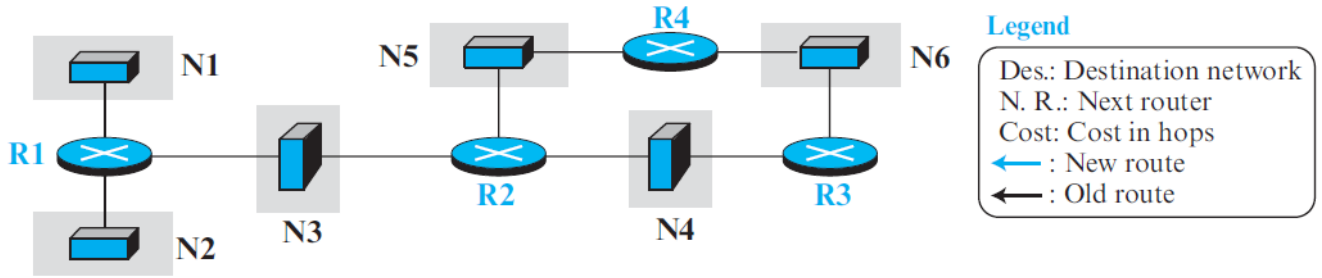


Fields

Com: Command, request (1), response (2)
Ver: Version, current version is 2
Family: Family of protocol, for TCP/IP value is 2
Tag: Information about autonomous system
Network address: Destination address
Subnet mask: Prefix length
Next-hop address: Address length
Distance: Number of hops to the destination

RIP Algorithm

- A router needs to send the whole contents of its forwarding table in a response message.
- The receiver adds one hop to each cost and changes the next router field to the address of the sending router.
- The receive route should be selected as the new one -
  1) If the received route does not exist in the old forwarding table,.
  2) If the cost of the received route is lower than the cost of the old one.
  3) If the cost of the received route is higher than the cost of the old one, but the value of the next router is the same in both routes.
- The new forwarding table needs to be sorted according to the destination route



R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2

Des.	N. R.	Cost
N3	—	1
N4	—	1
N5	—	1

R3

Des.	N. R.	Cost
N4	—	1
N6	—	1

R4

Des.	N. R.	Cost
N5	—	1
N6	—	1

Forwarding tables after all routers booted

New R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1
N4	R2	2
N5	R2	2

Old R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2 Seen by R1

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R3

Des.	N. R.	Cost
N3	R2	2
N4	—	1
N5	R2	2
N6	—	1

Old R3

Des.	N. R.	Cost
N4	—	1
N6	—	1

R2 Seen by R3

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	—	1
N6	—	1

Old R4

Des.	N. R.	Cost
N5	—	1
N6	—	1

R2 Seen by R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

Changes in the forwarding tables of R1, R3, and R4 after they receive a copy of R2's table

Final R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1
N4	R2	2
N5	R2	2
N6	R2	3

Final R2

Des.	N. R.	Cost
N1	R1	2
N2	R1	2
N3	—	1
N4	—	1
N5	—	1
N6	R3	2

Final R3

Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	—	1
N5	R2	2
N6	—	1

Final R4

Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	R2	2
N5	—	1
N6	—	1

Forwarding tables for all routers after they have been stablized

Timers in RIP

- *Periodic timer*
  - Controls the advertising of regular update messages.
  - Each router has one periodic timer that is set to a number between 25 and 35 seconds.
  - The timer counts down; when zero is reached, the update message is sent and timer is set again.

Timers in RIP…

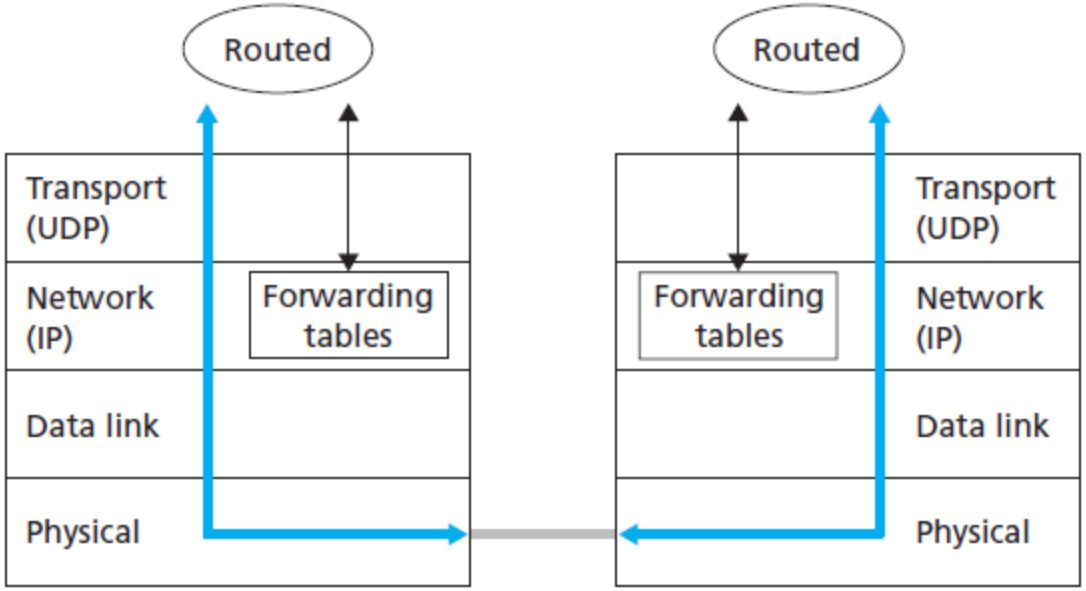
- *Expiration timer*
  - governs the validity of a route.
  - When a router receives update information for a route, the expiration timer is set to 180 seconds for that particular route.
  - Every time a new update for the route is received, the timer is reset.
  - If there is a problem on an internet and no update is received within the allotted 180 seconds, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable.

Timers in RIP…

- *Garbage collection timer*
  - Used to purge a route from the forwarding table.
  - When the information about a route becomes invalid, the router does not immediately purge that route from its table.
  - Instead, it continues to advertise the route with a metric value of 16.
  - At the same time, a garbage collection timer is set to 120 seconds for that route.
  - When the count reaches zero, the route is purged from the table.
  - This timer allows neighbors to become aware of the invalidity of a route prior to purging.

RIP Implementation

- RIP is implemented as a process that uses the service of UDP on the well-known port number 520.
- In BSD, RIP is a daemon process (a process running at the background), named *routed* (abbreviation for *route daemon* and pronounced *route-dee*).
- RIP runs at the application layer, but creates forwarding tables for IP at the network later.



Performance of RIP

- Update Messages. The update messages in RIP have a very simple format and are sent only to neighbors; they are local.
- Convergence of Forwarding Tables. RIP uses the distance-vector algorithm, which can converge slowly if the domain is large, but, since RIP allows only 15 hops in a domain (16 is infinity), there is normally no problem in convergence. The problems that slow down convergence are count-to-infinity and loops created in the domain; use of poison-reverse and split-horizon strategies may alleviate the situation.
- Robustness. The calculation of the forwarding table depends on information received from immediate neighbors, which in turn receive their information from their own neighbors. If there is a failure or corruption in one router, the problem will be propagated to all routers and the forwarding in each router will be affected.

Open Shortest Path First (OSPF)

- “Open” : publicly available
- OSPF is a link-state protocol that uses flooding of link-state information and a Dijkstra least-cost path algorithm.
- With OSPF, a router constructs a complete topological map (that is, a graph) of the entire autonomous system.
- The router then locally runs Dijkstra’s shortest-path algorithm to determine a shortest-path tree to all *subnets,* with itself as the root node.

Open Shortest Path First (OSPF)…

- A router broadcasts link state information whenever there is a change in a link's state.
- It also broadcasts a link's state periodically (every 30 minutes), even if the link's state has not changed.
- The OSPF protocol also checks that links are operational (via a HELLO message that is sent to an attached neighbor).
- Allows an OSPF router to obtain a neighboring router's database of network-wide link state.

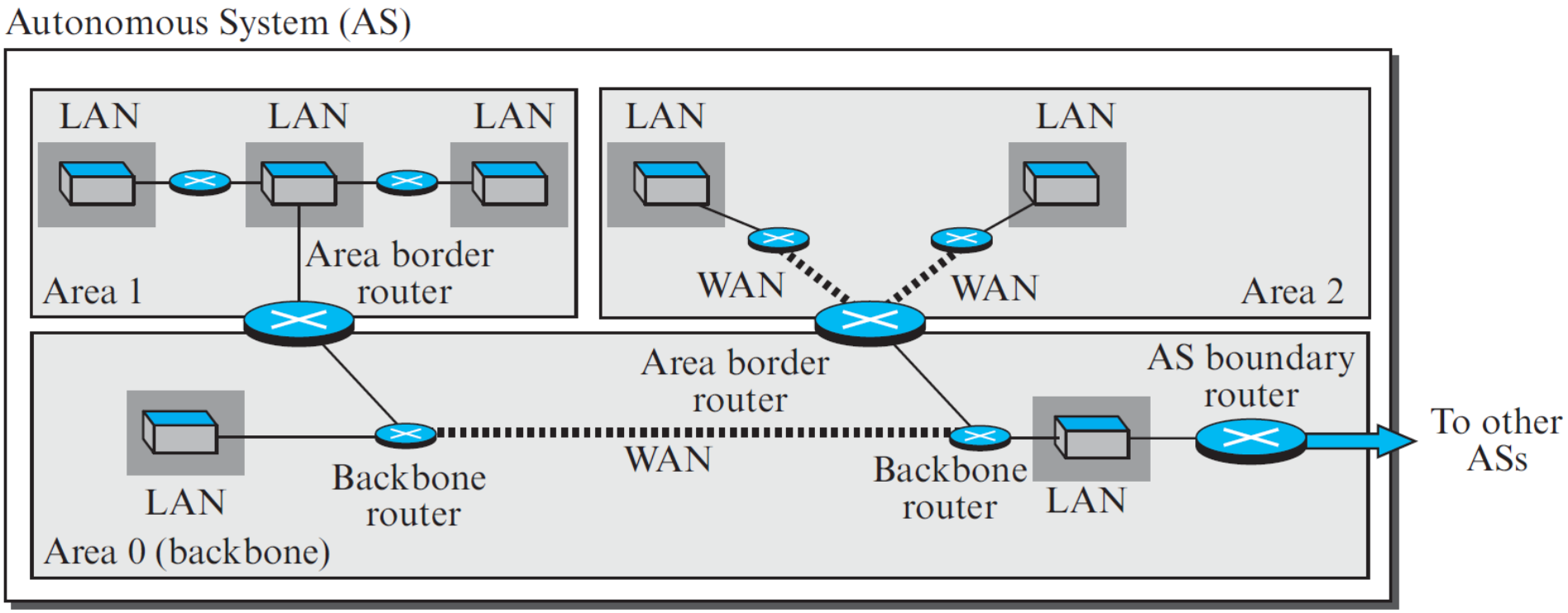
OSPF Advanced Features

- *Security.*
- *Multiple same-cost paths.*
- *Integrated support for unicast and multicast routing.*
- *Support for hierarchy within a single routing domain.*

Areas

- AS is divided into small sections called areas.
- Each area acts as a small independent domain for flooding LSPs.
- Each area runs its own OSPF link-state routing algorithm, with each router in an area broadcasting its link state to all other routers in that area.
- Within each area, one or more **area border routers** are responsible for routing packets outside the area.
- Exactly one OSPF area in the AS is configured to be the **backbone area.**
- The primary role of the backbone area is to route traffic between the other areas in the AS.
- The backbone always contains all area border routers in the AS and may contain nonborder routers as well.

Areas…


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

OSPF Implementation

- OSPF is implemented as a program in the network layer that uses the service of the IP for propagation.
- An IP datagram that carries a message from OSPF sets the value of the protocol field to 89.

OSPF Messages – 5 Types

- Hello message (type 1) is used by a router to introduce itself to the neighbors and announces all neighbors that it already knows.
- Database description message (type 2) is normally sent in response to the hello message to allow a newly joined router to acquire the full LSDB.
- Link-state request message (type 3) is sent by a router that needs information about a specific LS.
- Link-state update message (type 4) is the main OSPF message used for building the LSDB.
  - This message has five different versions (router link, network link, summary link to network, summary link to AS border router, and external link).
- Link-state acknowledgment message (type 5) is used to create reliability in OSPF; each router that receives a link-state update message needs to acknowledge it.

Performance of OSPF

- Update Messages. The link-state messages in OSPF have a complex format. They also are flooded to the whole area. If the area is large, these messages may create heavy traffic and use a lot of bandwidth.
- Convergence of Forwarding Tables. When the flooding of LSPs is completed, each router can create its own shortest-path tree and forwarding table; convergence is fairly quick. However, each router needs to run the Dijkstra's algorithm, which may take some time.
- Robustness. The OSPF protocol is more robust than RIP because, after receiving the completed LSDB, each router is independent and does not depend on other routers in the area. Corruption or failure in one router does not affect other routers as seriously as in RIP.

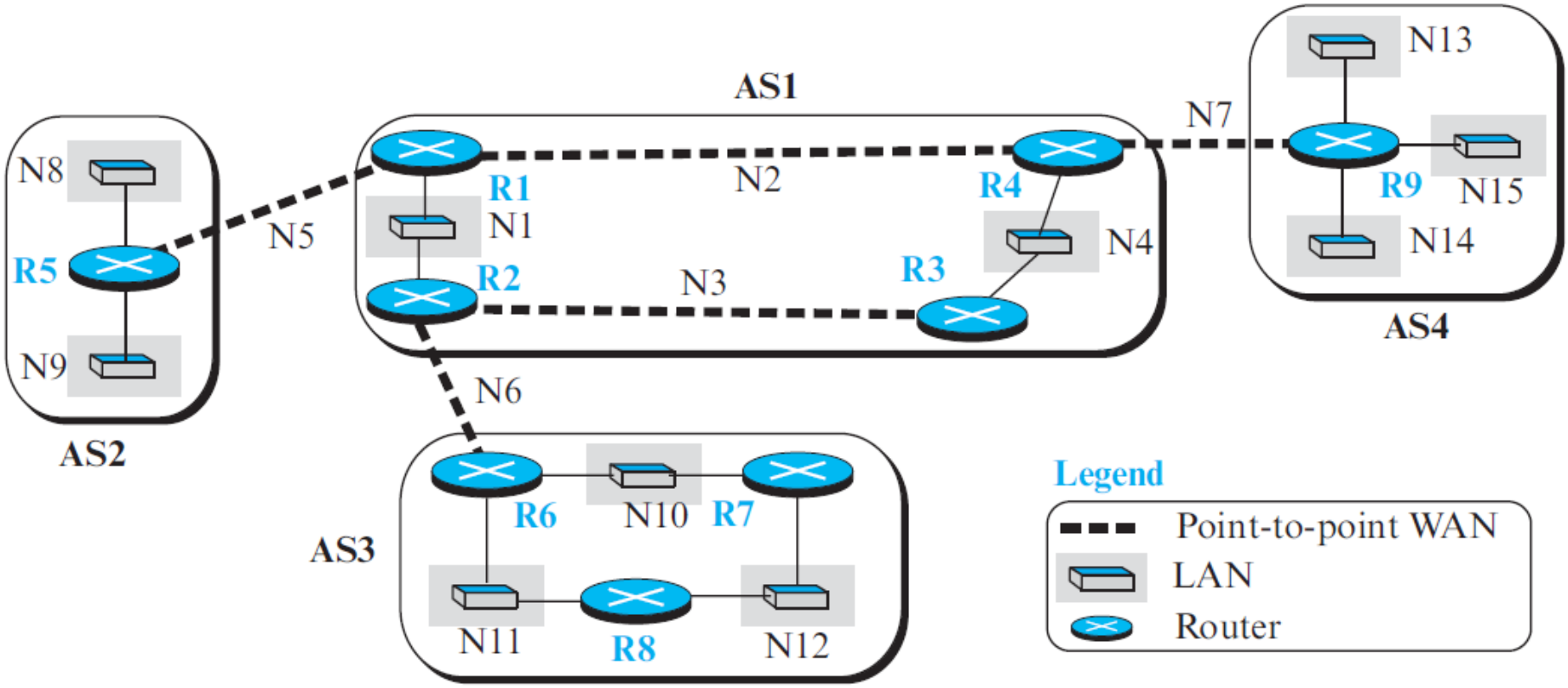
Inter-AS Routing: BGP

- The **Border Gateway Protocol** **version 4,** is the standard inter-AS routing protocol in today's Internet.
- As an inter-AS routing protocol, **BGP** provides each AS a means to
  1) Obtain subnet reachability information from neighboring ASs.
  2) Propagate the reachability information to all routers internal to the AS.
  3) Determine "good" routes to subnets based on the reachability information and on AS policy.
- BGP allows each subnet to advertise its existence to the rest of the Internet.

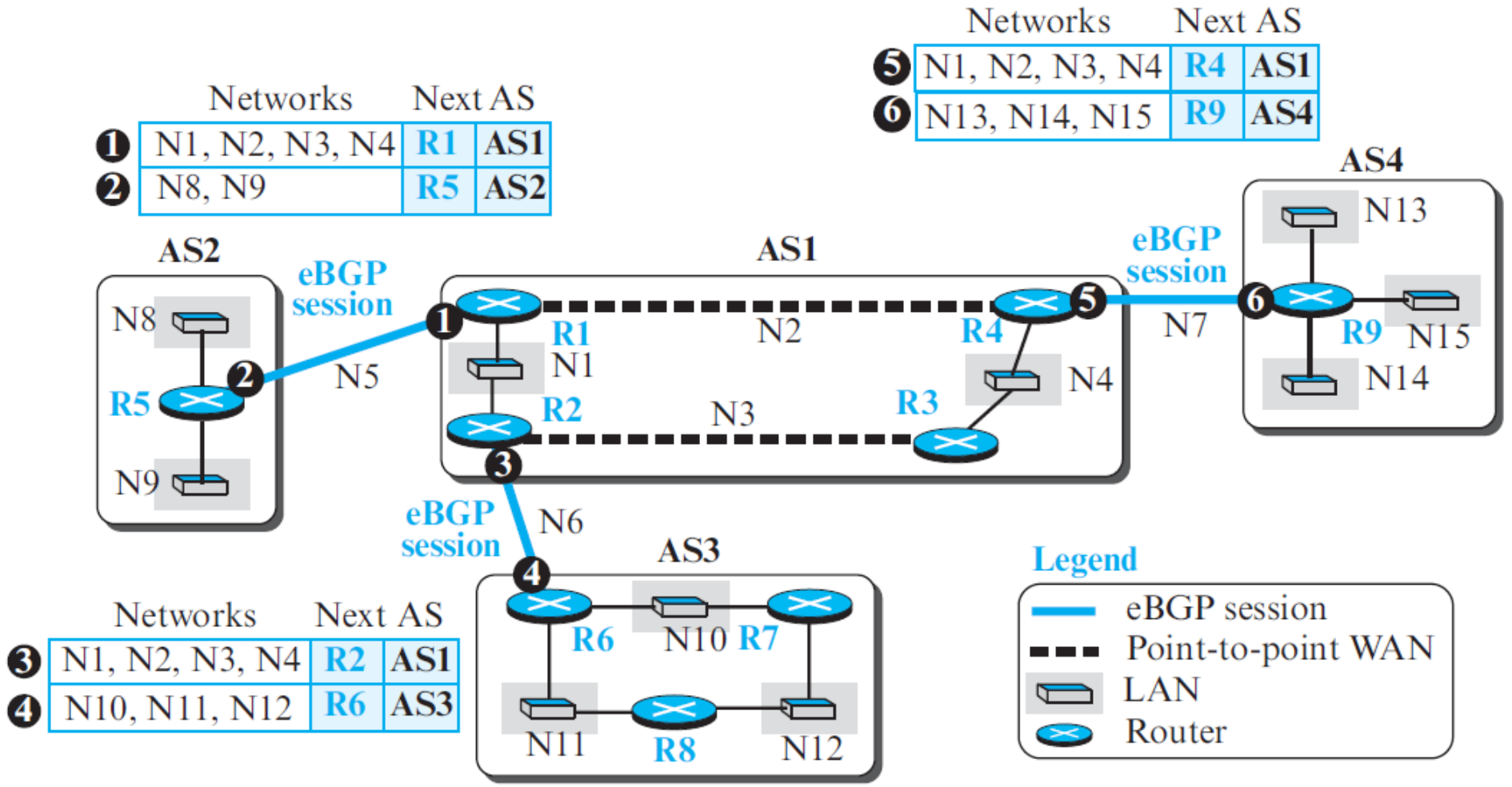
BGP Basics

- In BGP, pairs of routers exchange routing information over semipermanent TCP connections using **port 179.**
- For each TCP connection, the two routers at the end of the connection are called **BGP peers** or **BGP speakers**.
- The TCP connection along with all the BGP messages sent over the connection is called a **BGP session.**
- A BGP session that spans two ASs is called an **external BGP (eBGP) session.**
- BGP session between routers in the same AS is called an **internal BGP (iBGP) session.**

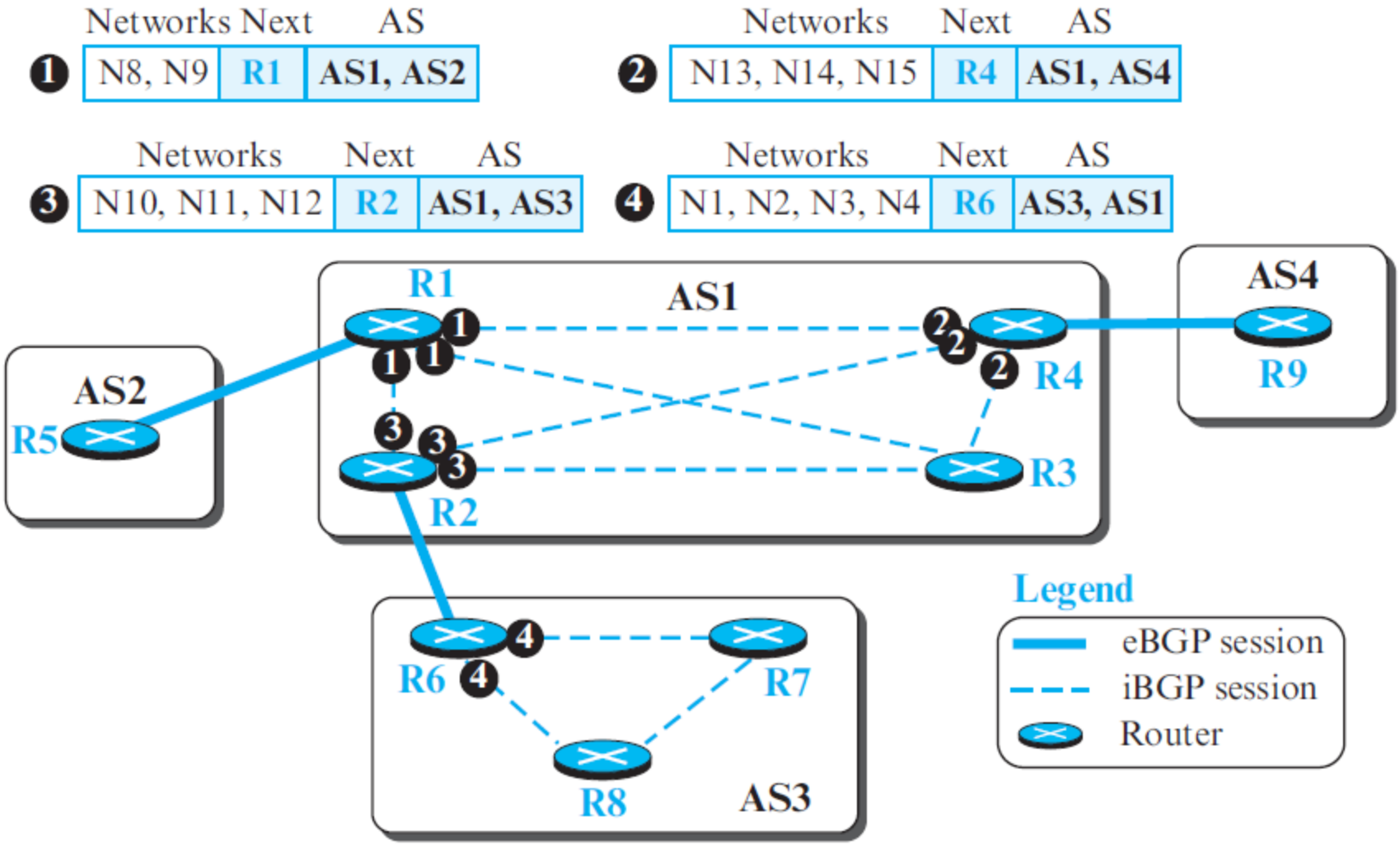
A sample internet with four ASs



eBGP Operation



Combination of eBGP and iBGP sessions



Finalized BGP path tables

Networks	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R1

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R6	AS1, AS3
N13, N14, N15	R1	AS1, AS4

Path table for R2

Networks	Next	Path
N8, N9	R2	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R3

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R1	AS1, AS3
N13, N14, N15	R9	AS1, AS4

Path table for R4

Networks	Next	Path
N1, N2, N3, N4	R1	AS2, AS1
N10, N11, N12	R1	AS2, AS1, AS3
N13, N14, N15	R1	AS2, AS1, AS4

Path table for R5

Networks	Next	Path
N1, N2, N3, N4	R2	AS3, AS1
N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R2	AS3, AS1, AS4

Path table for R6

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R7

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R8

Networks	Next	Path
N1, N2, N3, N4	R4	AS4, AS1
N8, N9	R4	AS4, AS1, AS2
N10, N11, N12	R4	AS4, AS1, AS3

Path table for R9

Path Attributes and BGP Routes

- In BGP, an autonomous system is identified by its globally unique **autonomous system number (ASN)** assigned by ICANN.
- When a router advertises a prefix across a BGP session, it includes a number of **BGP attributes.**
- In BGP, a prefix along with its attributes is called a **route.**
- 2 important attributes are :
  - **AS-PATH** - contains the ASs through which the advertisement for the prefix has passed.
  - **NEXT-HOP** -is the router interface that begins the AS-PATH.
- BGP also includes attributes that
  - Allow routers to assign preference metrics to the routes,
  - Indicates how prefix was inserted into BGP at the origin AS.

- **Import Policy**
  - When a gateway router receives a route advertisement, it uses its **import policy** to decide whether to accept or filter the route and whether to set certain attributes such as the router preference metrics.
  - The import policy may filter a route because the AS may not want to send traffic over one of the ASs in the route's AS-PATH.
  - The gateway router may also filter a route because it already knows of a preferable route to the same prefix.

BGP Route Selection

- If there are two or more routes to the same prefix, then BGP sequentially invokes the following elimination rules:
  1) Routes are assigned **a local preference value** as one of their attributes. The routes with the highest local preference values are selected.
  2) Then, the route with the shortest AS-PATH is selected.
  3) After, the route with the closest NEXT-HOP router is selected.
  4) If more than one route still remains, the router uses BGP identifiers to select the route.

BGP Messages

- Open Message – a router running BGP opens a TCP connection with a neighbor and sends an open message.
- Update Message – used by a router to withdraw destinations that have been advertised previously, to announce a route to a new destination, or both.
- Keepalive Message – The BGP peers, that are running, exchange keepalive messages regularly to tell each other that they are alive.
- Notification – sent by a router whenever an error condition is detected or a router wants to close the session.

Performance of BGP

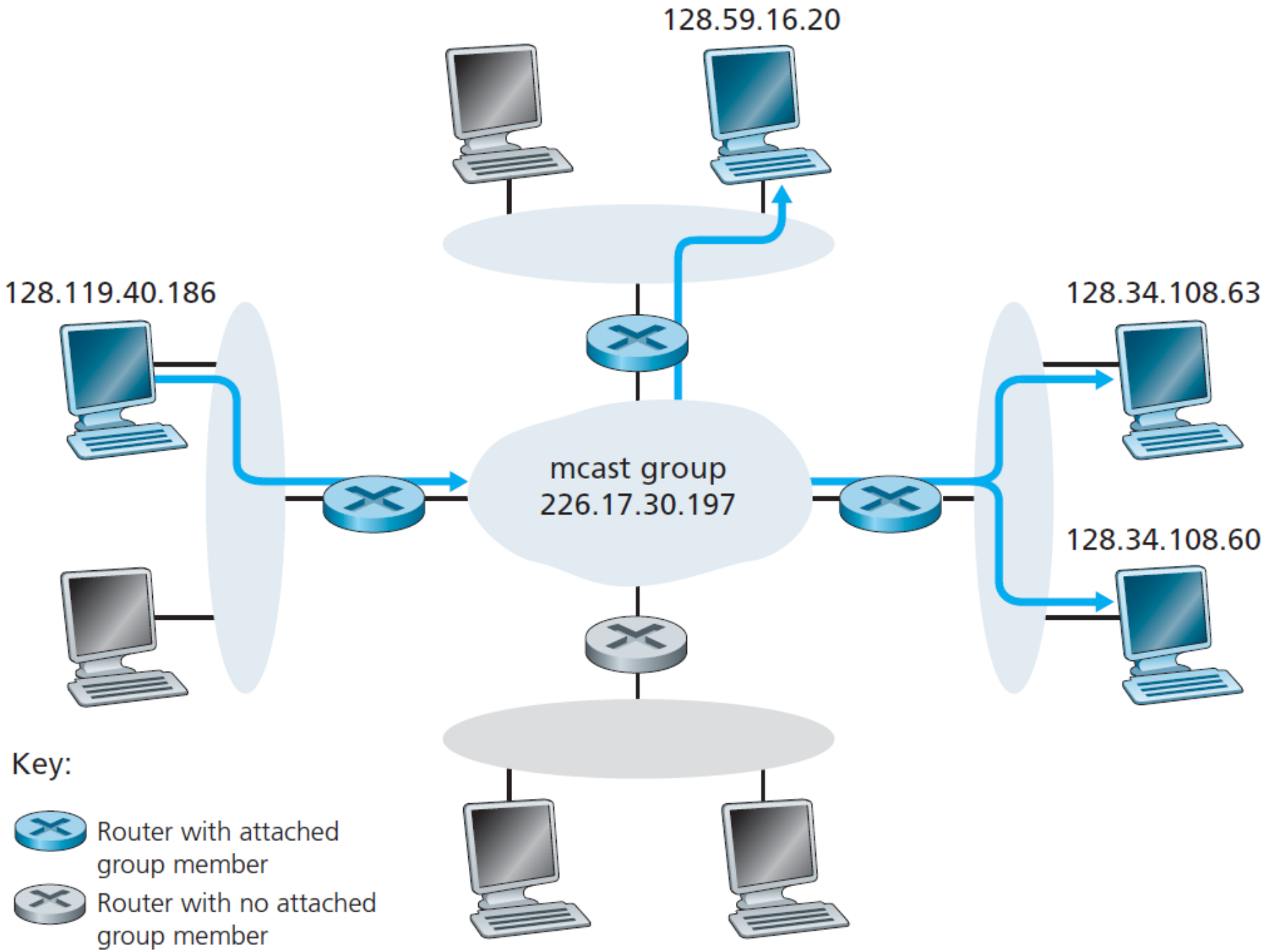
- BGP performance can be compared with RIP.
- BGP speakers exchange a lot of messages to create forwarding tables, but BGP is free from loops and count-to-infinity.
- The same weakness for RIP about propagation of failure and corruptness also exists in BGP.

Multicast Routing

- **Multicast routing** enables a single source node to send a copy of a packet to a subset of the other network nodes.
- Applications that use multicast, are
  - Bulk data transfer (e.g. The transfer of a software upgrade),
  - Streaming continuous media (e.g. The transfer of the audio, video, and text of a live lecture),
  - Shared data applications (e.g. A whiteboard or teleconferencing application),
  - Data feeds(e.g. Stock quotes),
  - Web cache updating, and
  - Interactive gaming (e.g. Distributed interactive virtual environments or multiplayer games).

Multicast Routing…

- Two problems—
  - How to identify the receivers of a multicast packet ?
  - How to address a packet sent to these receivers?
- A multicast packet is addressed using **address indirection.**
  - A single identifier is used for the group of receivers, and
  - A copy of the packet that is addressed to the group using this single identifier is delivered to all of the multicast receivers associated with that group.
- In the Internet, the single identifier that represents a group of receivers is a **class D multicast IP address**.
- The group of receivers associated with a class D address is referred to as a **multicast group.**
- The block 224.0.0.0/4 is used for multicast in classless addressing.


128.59.16.20
128.119.40.186
128.34.108.63
mcast group
226.17.30.197
128.34.108.60
Key:
Router with attached group member
Router with no attached group member

Multicast Routing…

- Network-layer multicast in the Internet consists of two complementary components:
  - Internet Group Management Protocol (IGMP)
  - Multicast Routing Protocols.

Internet Group Management Protocol (IGMP)

- The IGMP protocol version 3 operates between a host and its directly attached router.
- IGMP provides the means for a host to inform its attached router that an application running on the host wants to join a specific multicast group.
- IGMP messages are carried within an IP datagram, with an IP protocol number of 2.

IGMP Messages

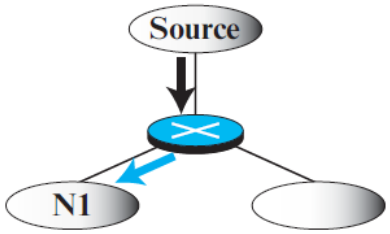
- Membership_query message
  - sent by a router to all hosts on an attached interface
  - to determine the set of all multicast groups that have been joined by the hosts on that interface.
  - 3 types : General, group-specific, source-and-group-specific query messages
- Membership_report message
  - Hosts respond to a membership_ query message with an IGMP membership_report message.
  - Can also be generated by a host when an application first joins a multicast group without waiting for a membership_query message from the router.
- Leave_group message.
  - This message is optional.

How does a router detect when a host leaves the multicast group?

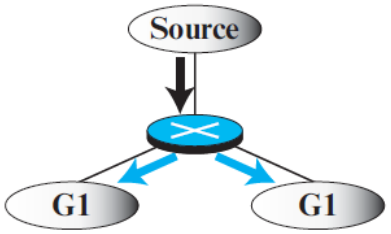
- The router *infers* that a host is no longer in the multicast group if it no longer responds to a membership_query message with the given group address.
- This is an example of **soft state** in an Internet protocol.
  - In a soft state protocol, the state is removed via a timeout event if it is not explicitly refreshed.

Multicast Forwarding

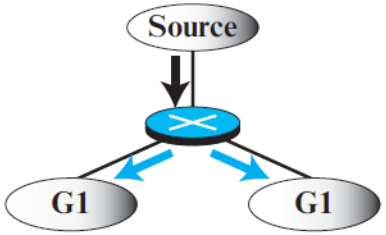
- In multicast communication, the destination of the packet defines one group, but that group may have more than one member in the internet.
- To reach all of the destinations, the router may have to send the packet out of more than one interface.
- Forwarding decisions in multicast communication depend on both the destination and the source address of the packet.



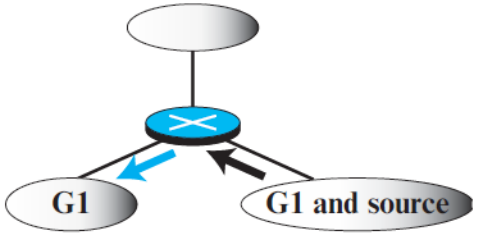
a. Destination in unicasting is one



b. Destination in mulicasting is more than one



a. Packet sent out of two interfaces



b. Packet sent out of one interface

Two Approaches to Multicasting

- Source-Based Tree Approach
- Group-Shared Tree Approach

Two Approaches to Multicasting…

- Source-Based Tree Approach
  - Each router needs to create a separate tree for each source-group combination.
  - If there are m groups and n sources in the internet, a router needs to create (m × n) routing trees
  - In each tree, the source is the root, the members of the group are the leaves, and the router itself is somewhere on the tree.

Two Approaches to Multicasting…

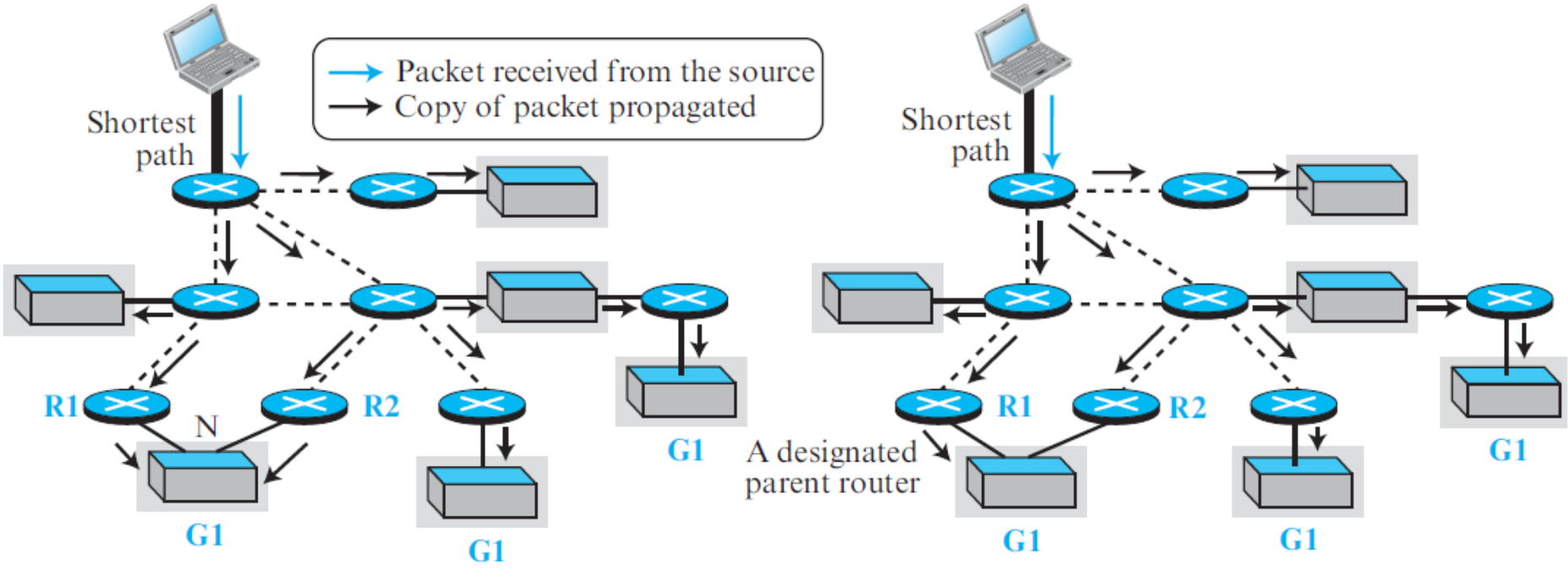
- Group-Shared Tree Approach
  - designate a router to act as the phony source for each group.
  - The designated router, which is called the core router or the rendezvous point router, acts as the representative for the group.
  - Any source that has a packet to send to a member of that group sends it to the core center (unicast communication) and the core center is responsible for multicasting.
  - The core center creates one single routing tree with itself as the root and any routers with active members in the group as the leaves.
  - There are m core routers (one for each group) and each core router has a routing tree, for the total of m trees.

Multicast Routing Algorithms

- Intra-domain
  - Distance-Vector Multicast Routing Protocol (DVMRP)
  - Multicast Open Shortest Path First (MOSPF)
  - Protocol Independent Multicast (PIM)
- Inter-domain
  - Multicast Border Gateway Protocol (MBGP)

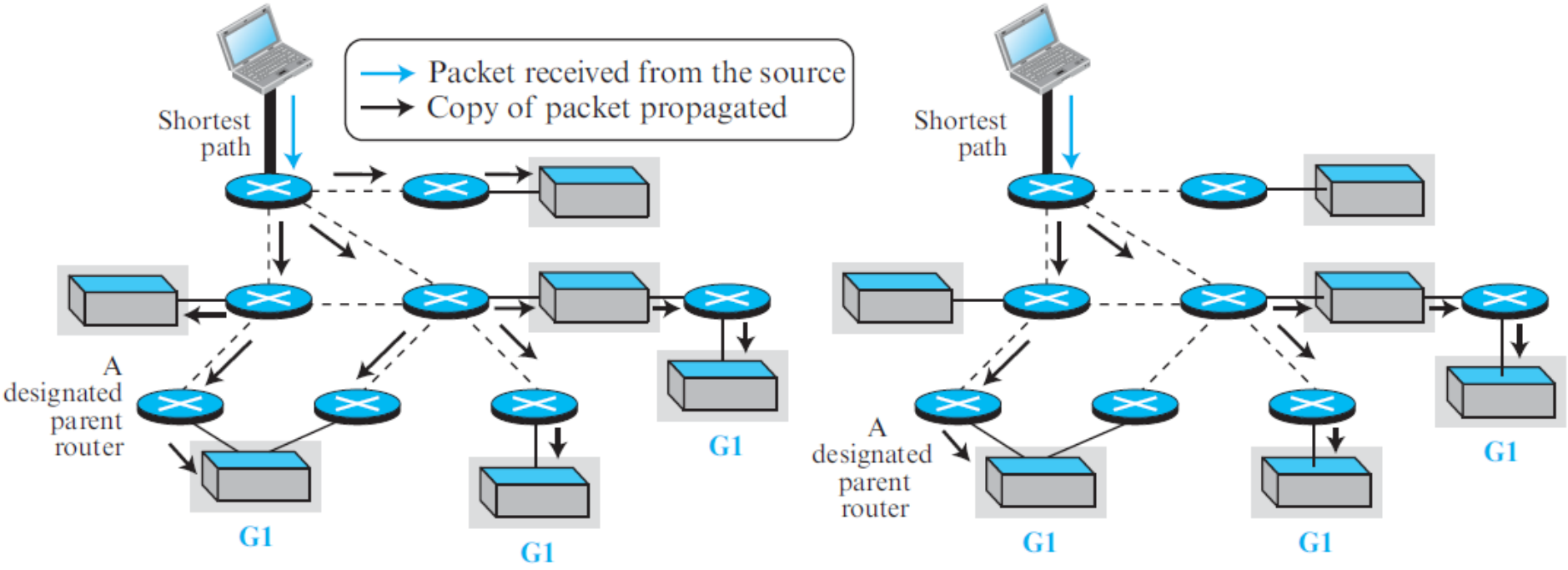
Distance-Vector Multicast Routing Protocol (DVMRP)

- Extension of the Routing Information Protocol (RIP) which is used in unicast routing.
- It uses the source based tree approach to multicasting.
- Each router in this protocol that receives a multicast packet to be forwarded implicitly creates a source-based multicast tree in three steps:
  - Reverse path forwarding (RPF)
  - Reverse path broadcasting (RPB)
  - Reverse path multicasting (RPM)



a. Using RPF, N receives two copies.

b. Using RPB, N receives only one copy.



a. Using RPB, all networks receive a copy.

b. Using RPM, only members receive a copy.

Multicast Open Shortest Path First (MOSPF)

- Extension of the Open Shortest Path First (OSPF) protocol, which is used in unicast routing.
- Uses the source based tree approach to multicasting.
- Each router needs to have another LSDB database, to show which interface has an active member in a particular group.
- Steps
  1) The router uses the Dijkstra algorithm to create a shortest-path tree, the root of the tree is the source of the packet defined in the source address of the packet. the source address of the packet.
  2) The router finds itself in the shortest-path tree created. the router creates a shortest-path subtree with itself as the root of the subtree.
  3) The router prunes the broadcast subtree to a multicast tree.
  4) The router can now forward the received packet out of only those interfaces that correspond to the branches of the multicast tree.

IPv6

- The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP protocol in the early 1990s.
- The new version, which is called **Internet Protocol version 6 (IPv6)** or **IP new generation (IPng)** was a proposal to augment the address space of IPv4 and at the same time redesign the format of the IP packet and revise some auxiliary protocols such as ICMP.

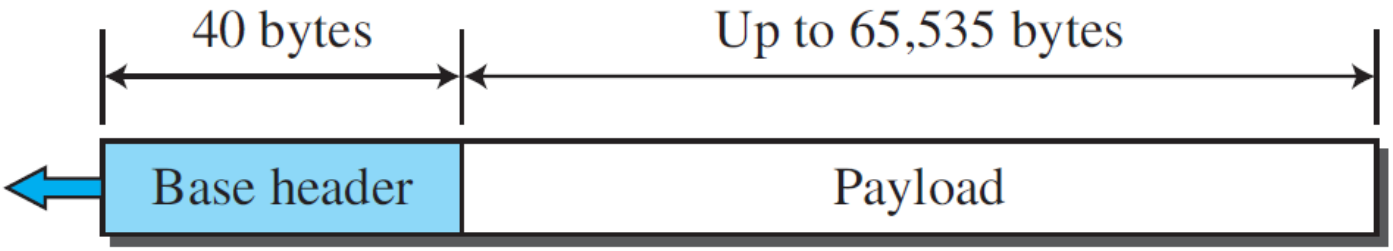
Main changes in the IPv6 protocol

- **Larger address space**.
  - An IPv6 address is 128 bits long.
  - Compared with the 32-bit address of IPv4, this is a huge (2^{96} times) increase in the address space.
  - IPv6 has introduced a new type of address, called an **anycast address**, which allows a datagram to be delivered to any one of a group of hosts.
- **Better header format**.
  - 40-byte fixed-length header.
  - Options are separated from the base header and inserted, when needed, between the base header and the data.
  - This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options**.
  - To allow for additional functionalities.

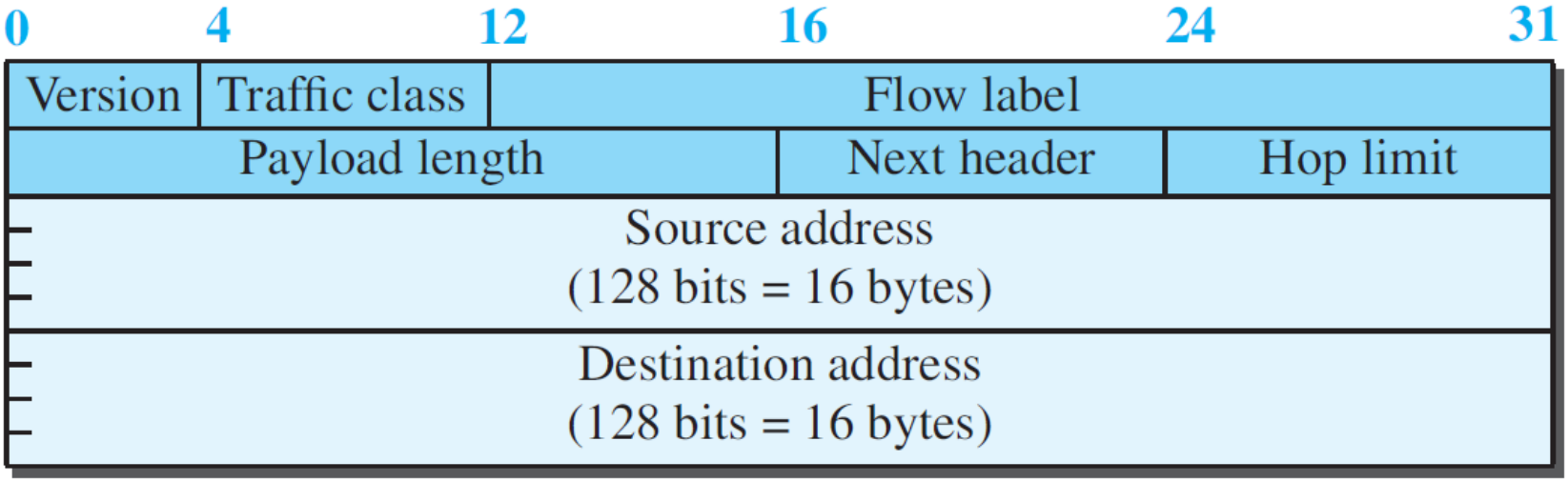
Main changes in the IPv6 protocol…

- **Allowance for extension**.
  - Allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation.**
  - The type-of-service field has been removed.
  - Two new fields, traffic class and flow label, have been added to enable the source to request special handling of the packet.
  - Flow label support traffic such as real-time audio and video which is treated as a flow.
  - Traffic class is used to give priority to certain datagrams within a flow, or it can be used to give priority to datagrams from certain applications (for example, ICMP).
- **Support for more security.**
  - The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

IPv6 Datagram Format



a. IPv6 packet

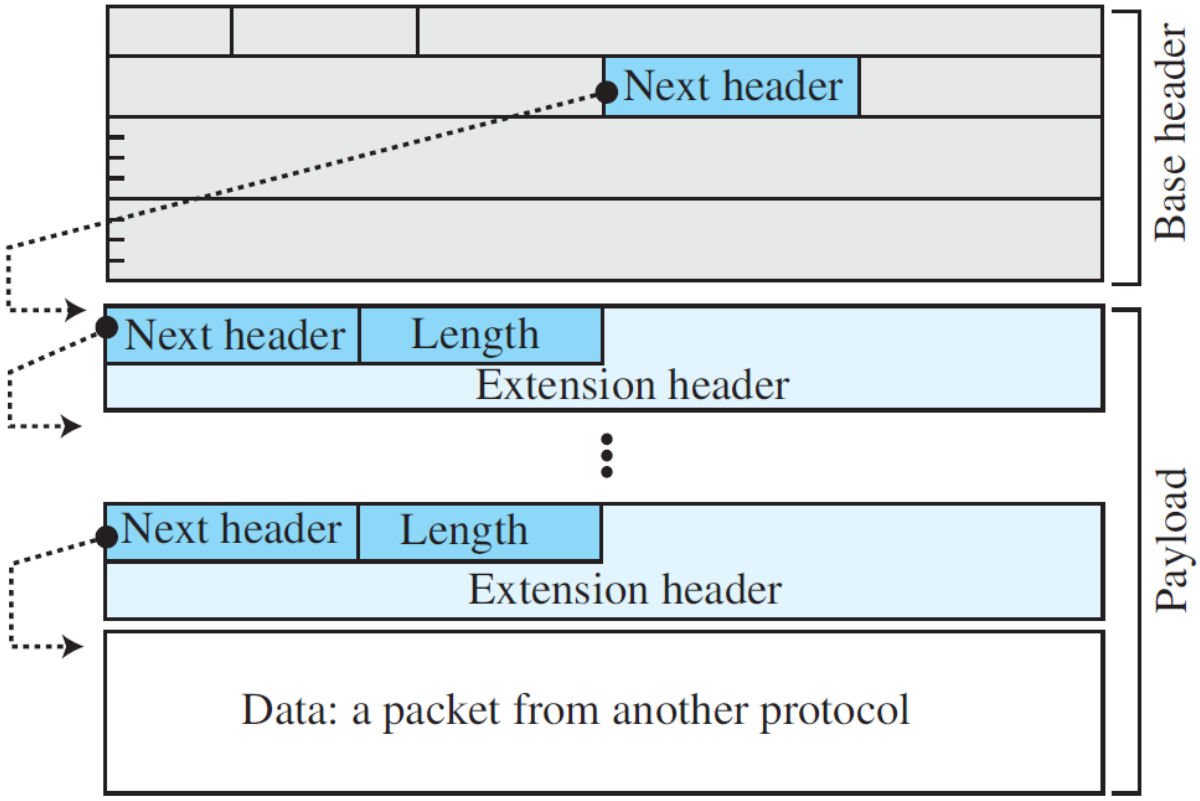


b. Base header

- ***Version.*** The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.
- ***Traffic class.*** The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements. It replaces the type-of-service field in IPv4.
- ***Flow label.*** The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data.
- ***Payload length.*** The 2-byte payload length field defines the length of the IP datagram excluding the header. In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined.

- ***Next header.*** The next header is an 8-bit field defining the type of first extension header (if present) or the type of the data that follows the base header in the datagram.
  - This field is similar to the protocol field in IPv4.
- ***Hop limit.*** The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- ***Source and destination address.*** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.

- ***Payload.*** A combination of zero or more extension headers (options) followed by the data from other protocols.



Some next-header codes

00: Hop-by-hop option
02: ICMPv6
06: TCP
17: UDP
43: Source-routing option
44: Fragmentation option
50: Encrypted security payload
51:Authentication header
59: Null (no next header)
60: Destination option

Other changes from IPv4

- Fields no longer present in the IPv6 datagram:
  - *Fragmentation/Reassembly.*
    - IPv6 does not allow for fragmentation and reassembly at intermediate routers; performed only by the source and destination.
    - When a router receives the packet, whose size is larger than allowed by the MTU , drops it.
    - Then sends a packet-too-big ICMPv6 error message to the source.
  - *Header checksum*
    - Redundant and costly.
  - *Options.*
    - The removal of this field results in a fixed-length, 40- byte IP header.

IPv6 Addressing

- An IPv6 address is 128 bits or 16 bytes long, four times the address length in IPv4.
- 2 Notations : binary and colon hexadecimal.
  - Binary 1111111011110110 … 1111111100000000
  - Colon Hexadecimal

    FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00
- Abbreviations in Colon Hexadecimal notation.
  - The leading zeros of a section can be omitted.
    - 0074 can be written as 74, 000F as F.
  - zero compression :remove all the zeros altogether and replace them with a double semicolon.
    - FDEC:0:0:0:0:BBFF:0:FFFF → FDEC::BBFF:0:FFFF
    - Allowed only once per address. If there is more than one run of zero sections, only one of them can be compressed.
- Slash or CIDR notation.
  - FDEC::BBFF:0:FFFF/60

Address Space

- The address space of IPv6 contains 2^{128} addresses.
- This address space is 2^{96} times the IPv4 address—definitely no address depletion.

Three Address Types

- In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast.
  - A unicast address defines a single interface (computer or router).
    - The packet sent to a unicast address will be routed to the intended recipient.
  - An anycast address defines a group of computers that all share a single address.
    - A packet with an anycast address is **delivered to only one member** of the group, the most reachable one.
  - A multicast address also defines a group of computers.
    - **Each member of the group receives a copy.**
    - IPv6 has designated a block for multicasting from which the same address is assigned to the members of the group.
- IPv6 does not define broadcasting.

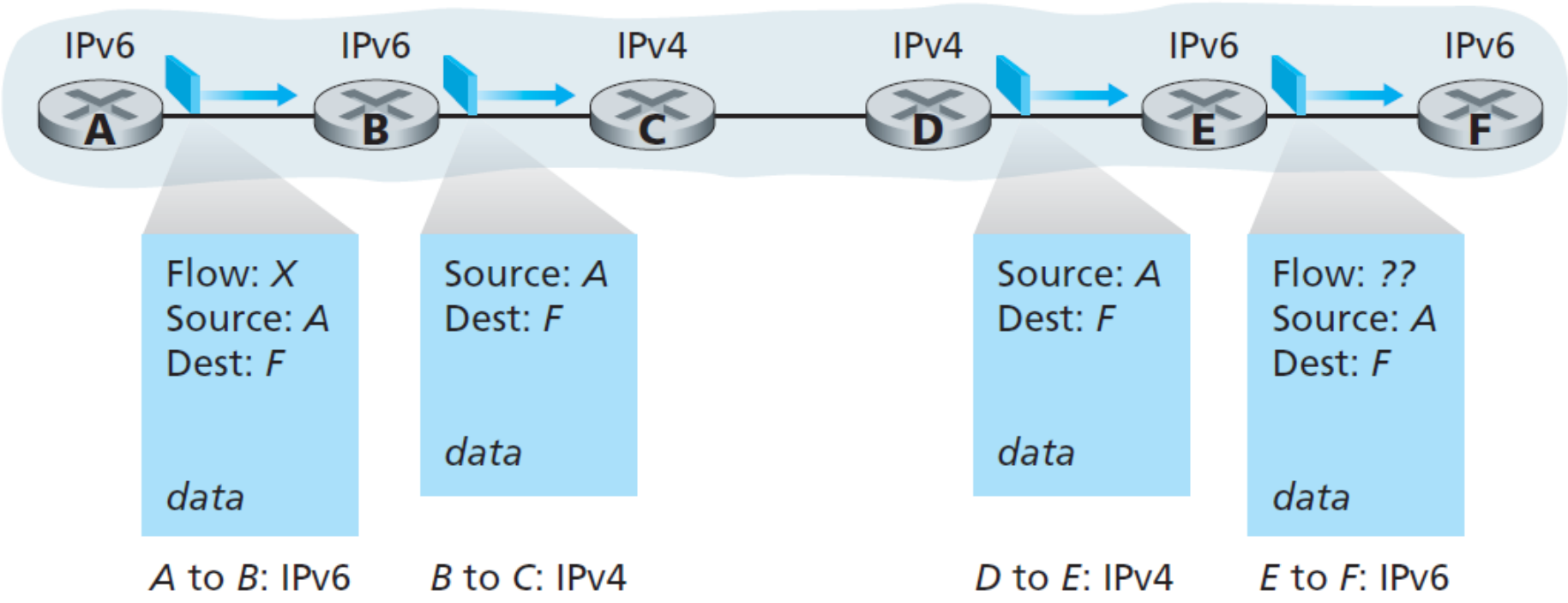
Transition from IPv4 to IPv6

- **Dual-stack** approach.
- **Tunneling**

Dual-stack approach

- **Dual-stack** approach, where IPv6 nodes also have a complete IPv4 implementation.
- Such a node, referred to as an IPv6/IPv4 node.
  - has the ability to send and receive both IPv4 and IPv6 datagrams.
  - When interoperating with an IPv4 node, an IPv6/IPv4 node can use IPv4 datagrams;
  - when interoperating with an IPv6 node, it can speak IPv6.
  - IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses.
- In the dual-stack approach, if either the sender or the receiver is only IPv4- capable, an IPv4 datagram must be used.

A dual-stack approach

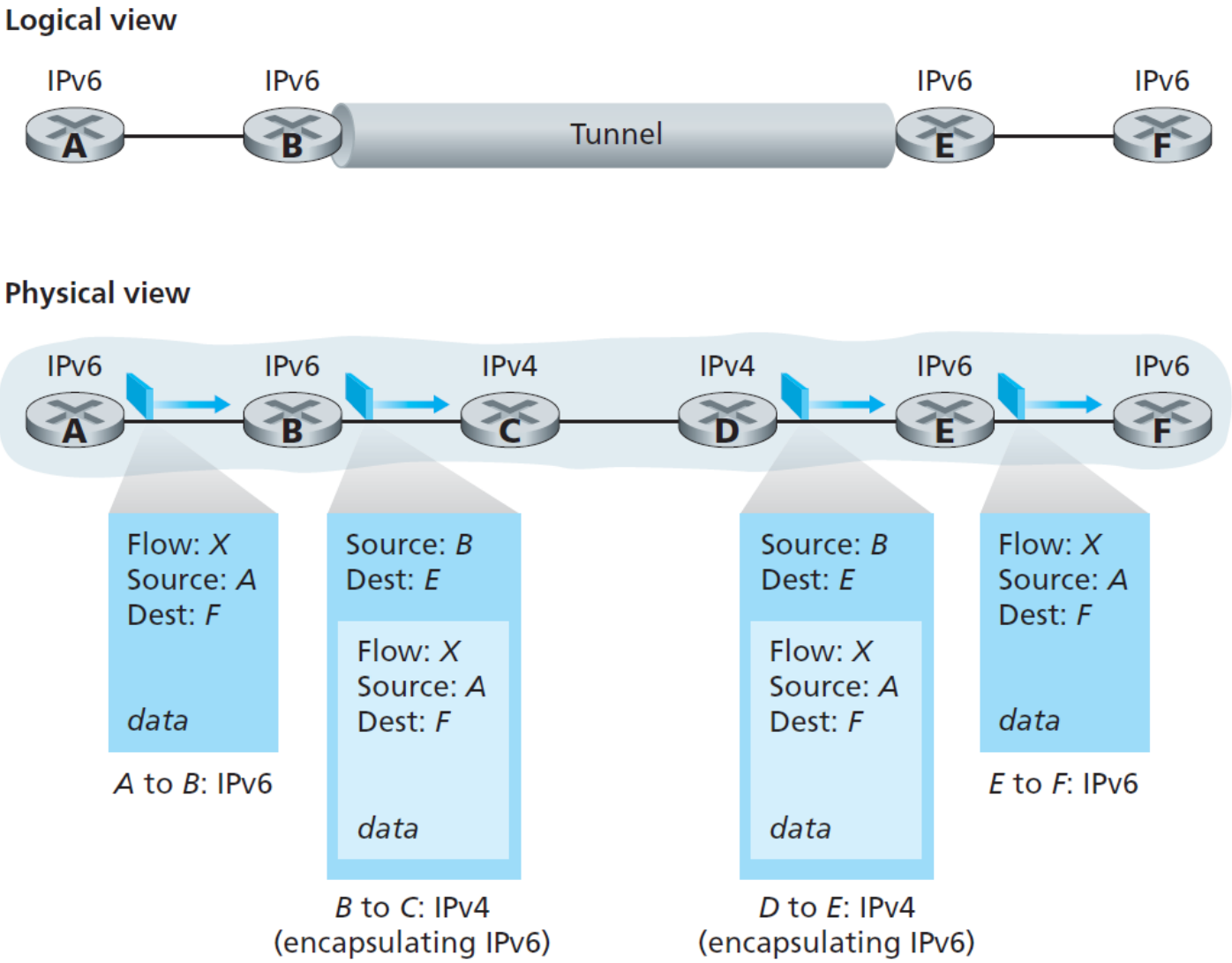


- In performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram that have no counterpart in IPv4.
- The information in these fields will be lost.

Tunneling

- Suppose two IPv6 nodes want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers.
- We refer to the intervening set of IPv4 routers between two IPv6 routers as a **tunnel**.
- With tunneling, the IPv6 node on the sending side of the tunnel takes the *entire* IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram.
- This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel and sent to the first node in the tunnel.
- The intervening IPv4 routers in the tunnel route this IPv4 datagram among themselves, unaware that the IPv4 datagram itself contains a complete IPv6 datagram.
- The IPv6 node on the receiving side of the tunnel receives the IPv4 datagram (it is the destination of the IPv4 datagram!),
  - determines that the IPv4 datagram contains an IPv6 datagram,
  - extracts the IPv6 datagram, and
  - then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor.

Tunneling…


Logical view
IPv6
IPv6
IPv6
IPv6
A
B
Tunnel
E
F
Physical view
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
A
B
C
D
E
F
Flow: X
Source: A
Dest: F
data
A to B: IPv6
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
B to C: IPv4
(encapsulating IPv6)
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
D to E: IPv4
(encapsulating IPv6)
Flow: X
Source: A
Dest: F
data
E to F: IPv6