Chapter 4

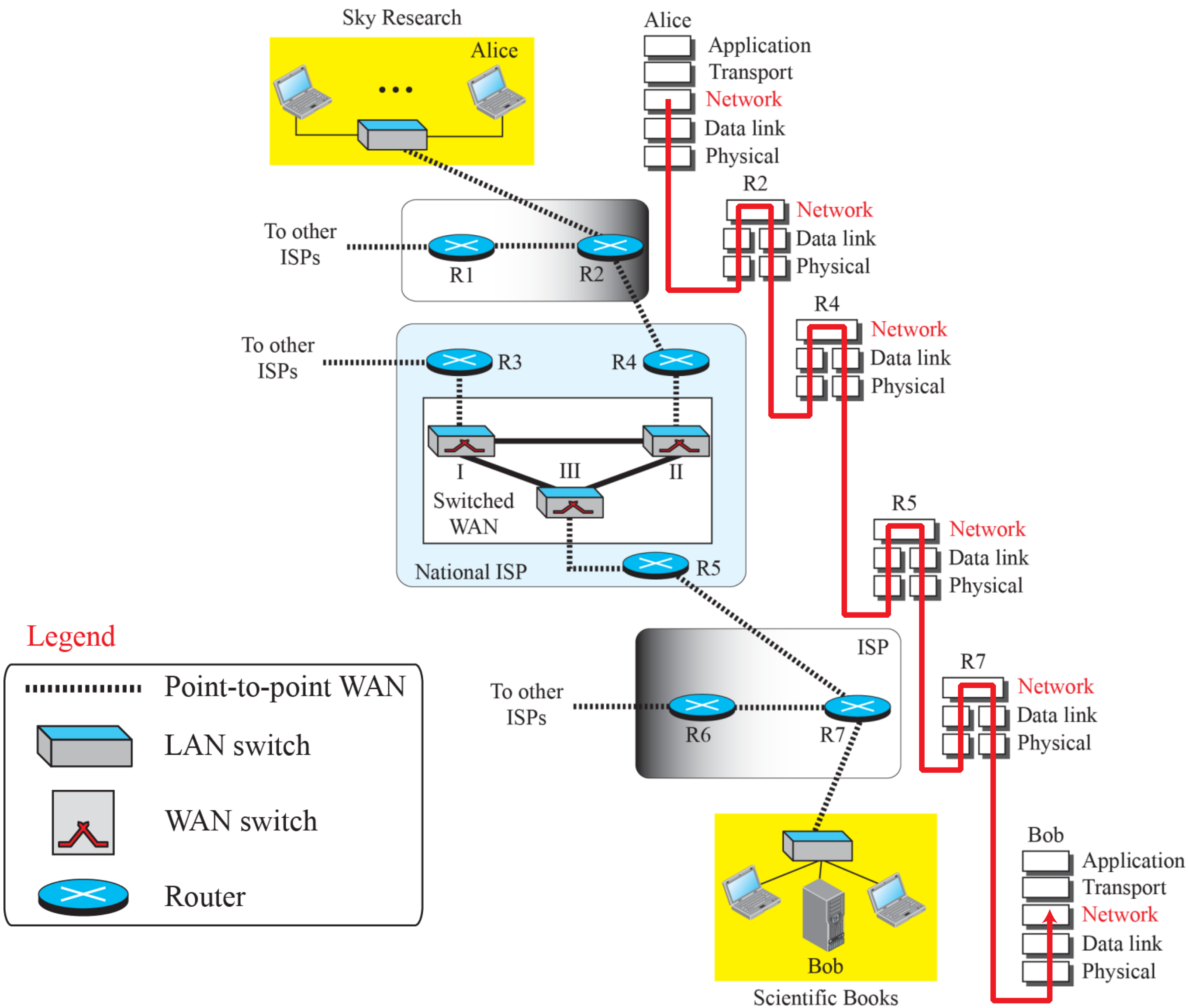
Network Layer

4-1 INTRODUCTION

- ***Figure 4.1 shows the communication between Alice and Bob at the network layer.***

- ***This is the same scenario we used in Chapters 2 and 3 to show the communication at the application and the transport layers, respectively.***

Figure 4.1: Communication at the network layer

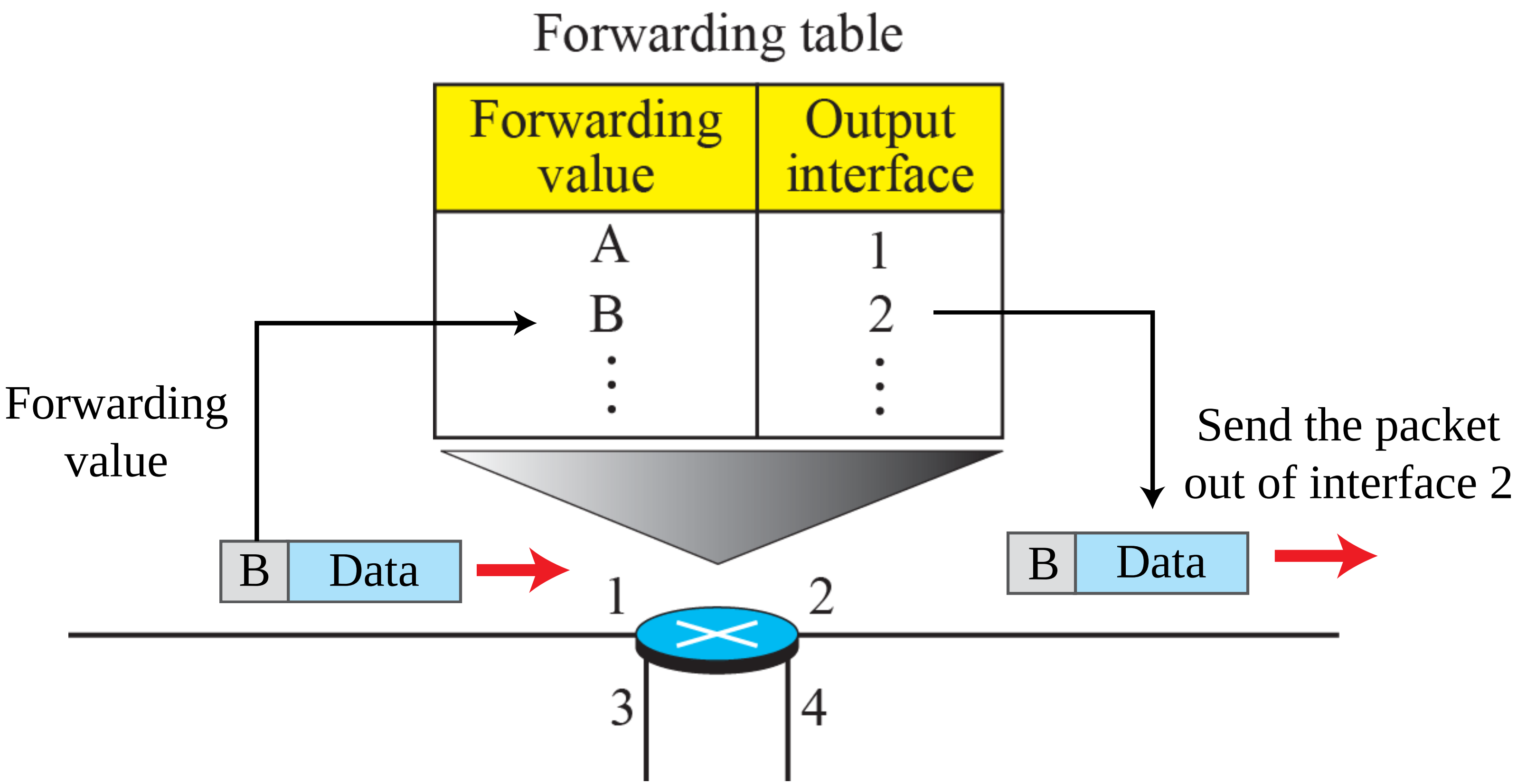


Network-Layer Services

The network-layer services that, in general, are expected from a network-layer protocol.

- ***Packetizing***
- ***Routing***
- ***Forwarding***

Figure 4.2: Forwarding process



Packet Switching

- *A kind of switching occurs at the network layer*

- *A router is a switch that creates a connection between an input port and an output port (or a set of output ports), just as an electrical switch connects the input to the output to let electricity flow.*

- ***Datagram Approach***
- ***Virtual-Circuit Approach***
  - ***Setup Phase***
  - ***Data-Transfer Phase***
  - ***Teardown Phase***

Figure 4.3: A connectionless packet-switched network

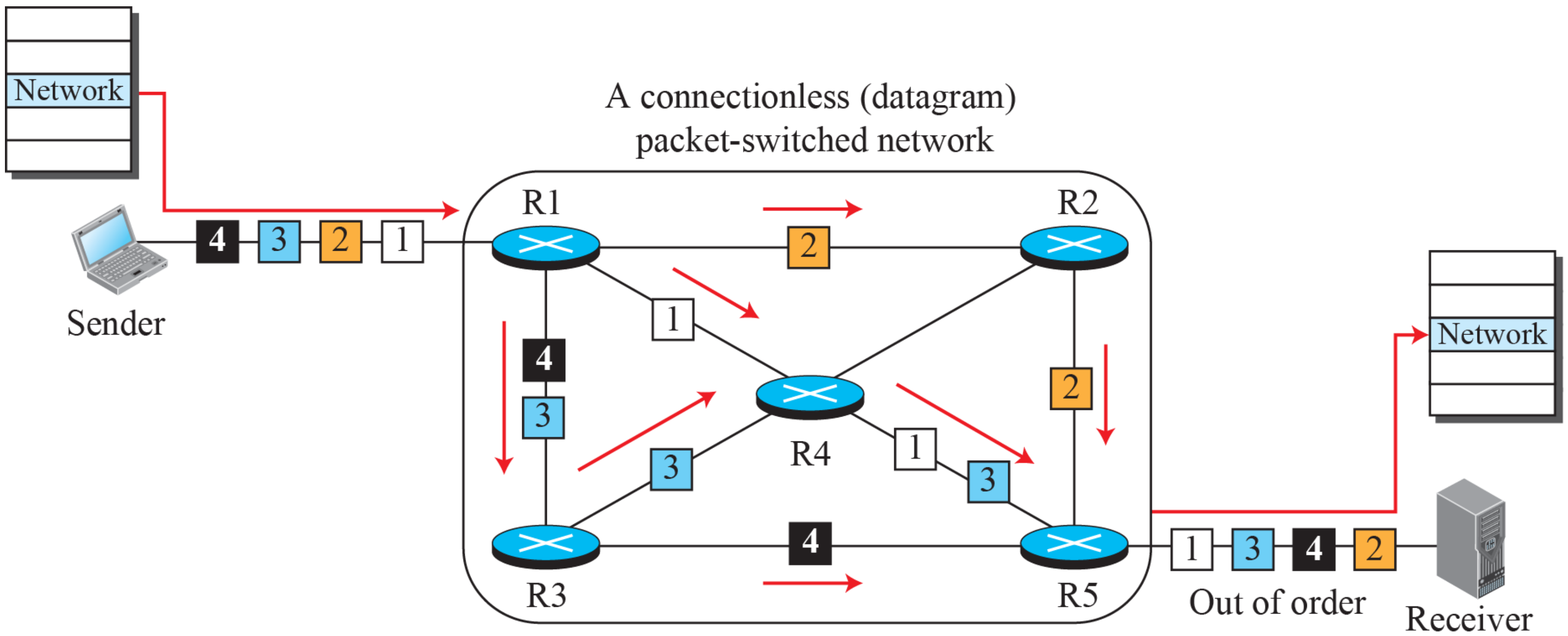


Figure 4.4: Forwarding process in a router when used in a connectionless network

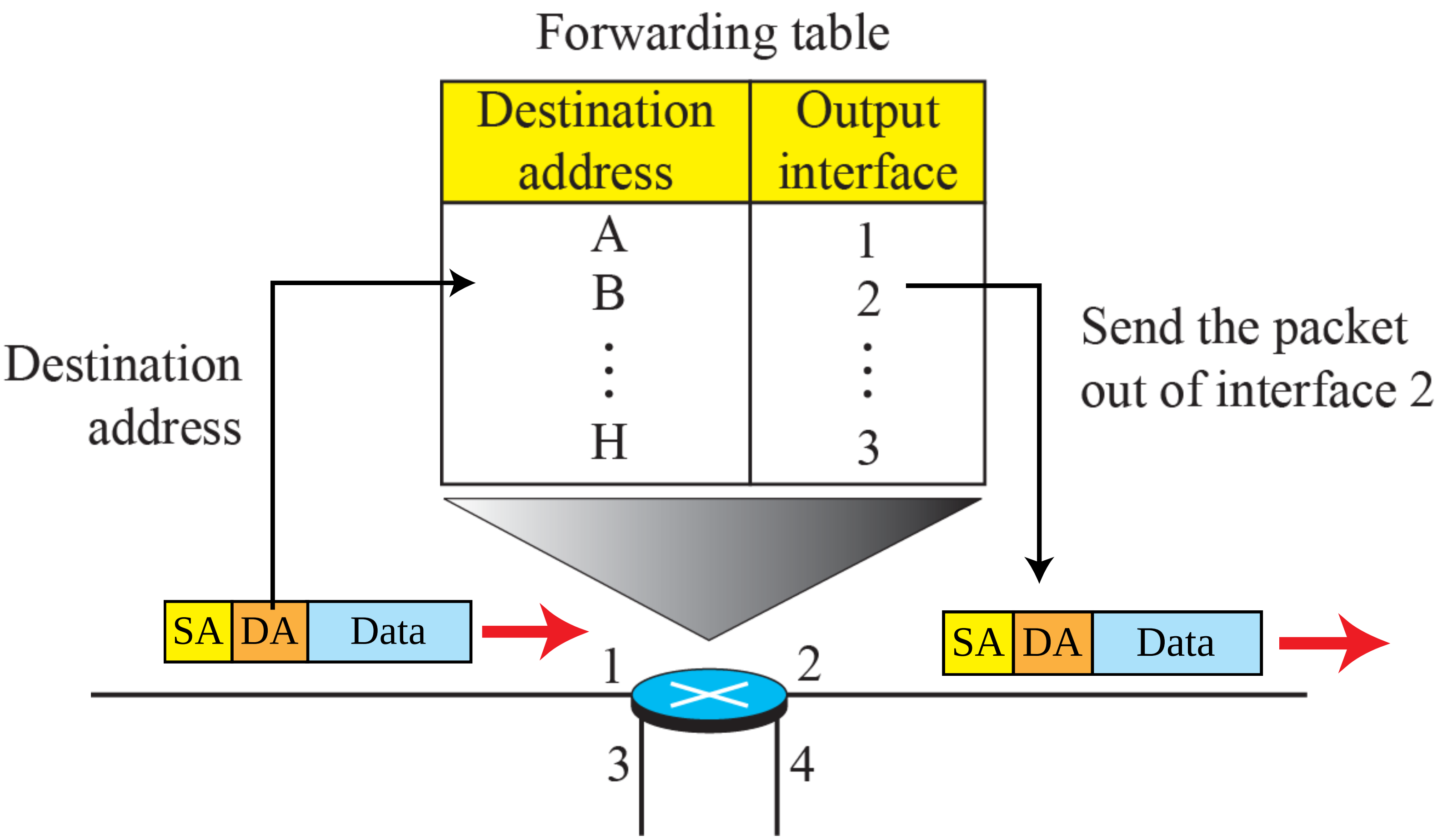


Figure 4.5: A virtual-circuit packet-switched network

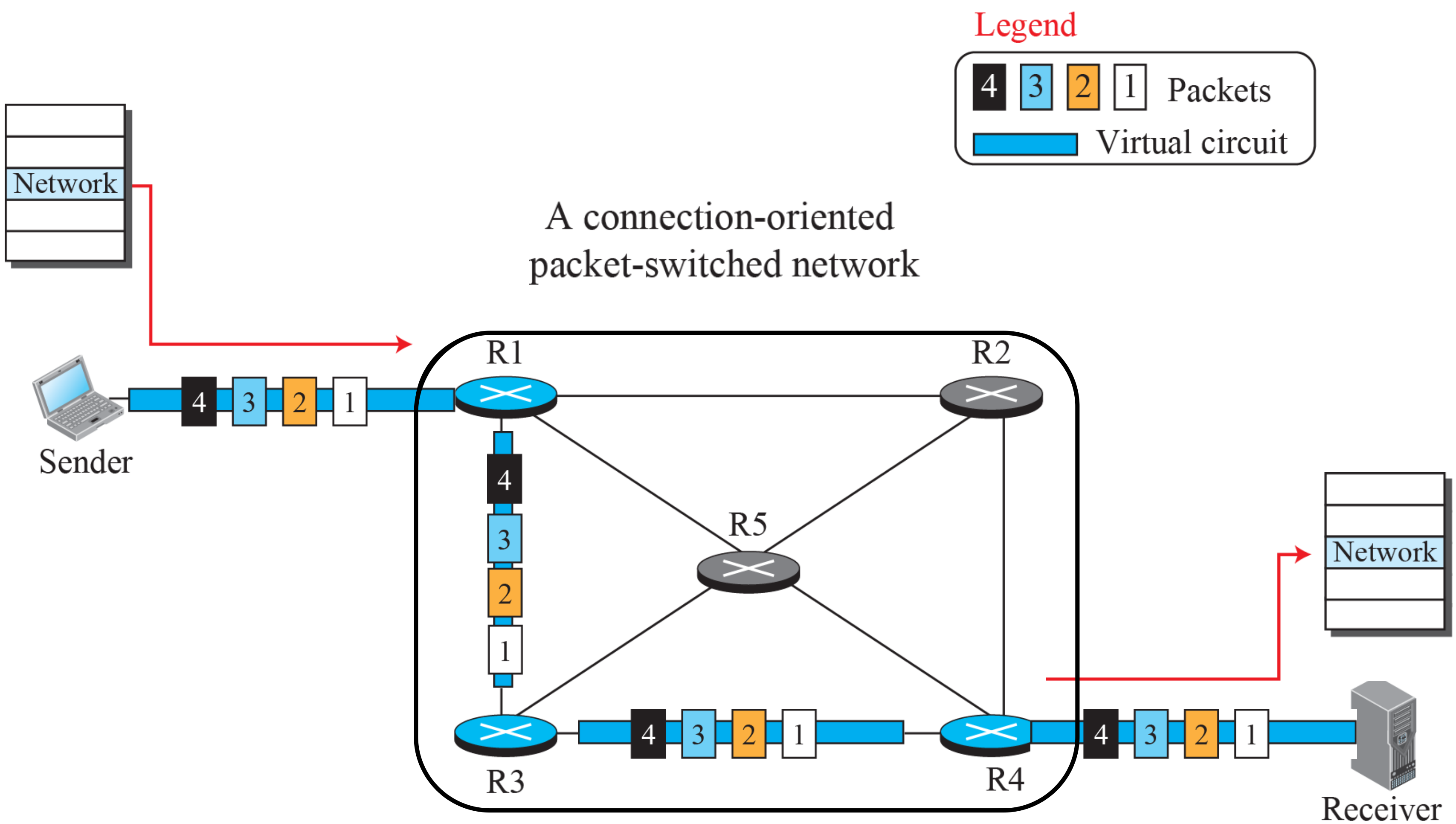


Figure 4.6: Forwarding process in a router when used in a virtual circuit network

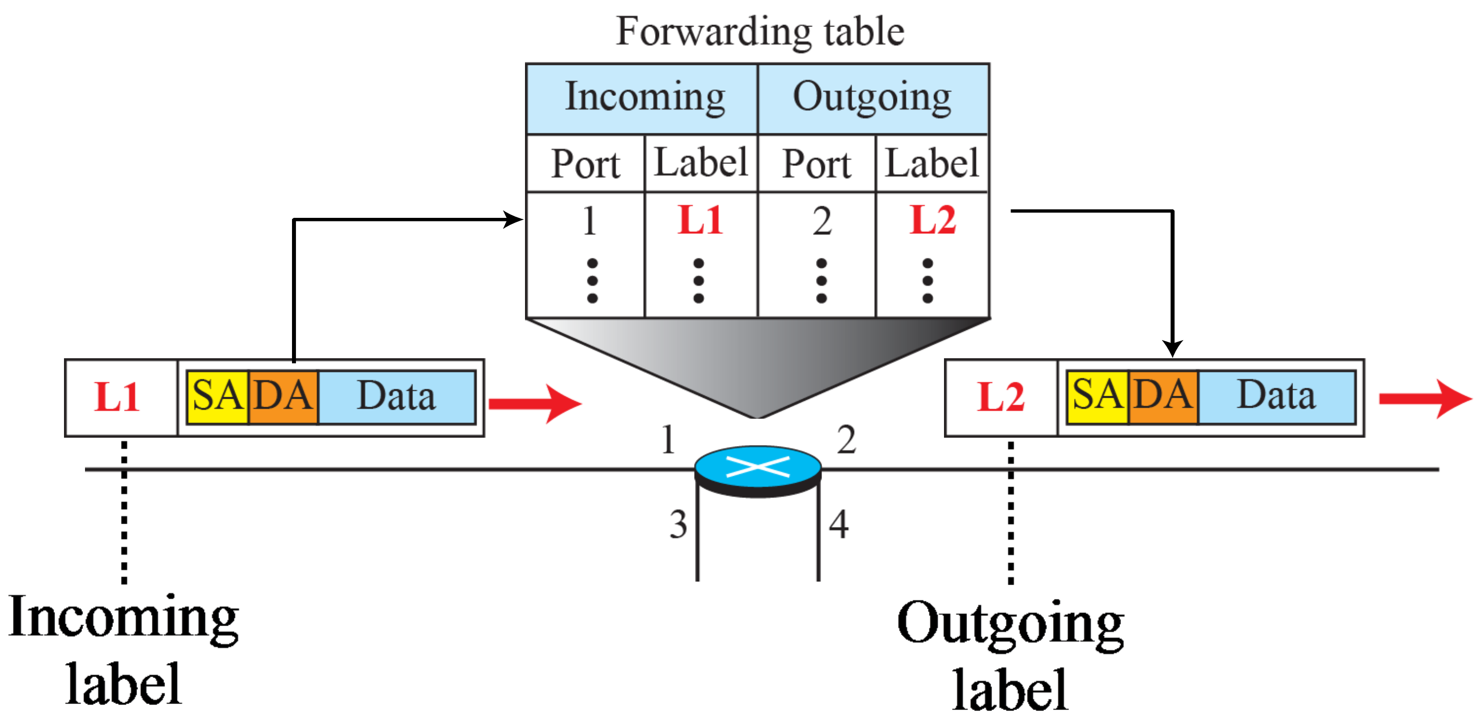


Figure 4.7: *Sending request packet in a virtual-circuit network*

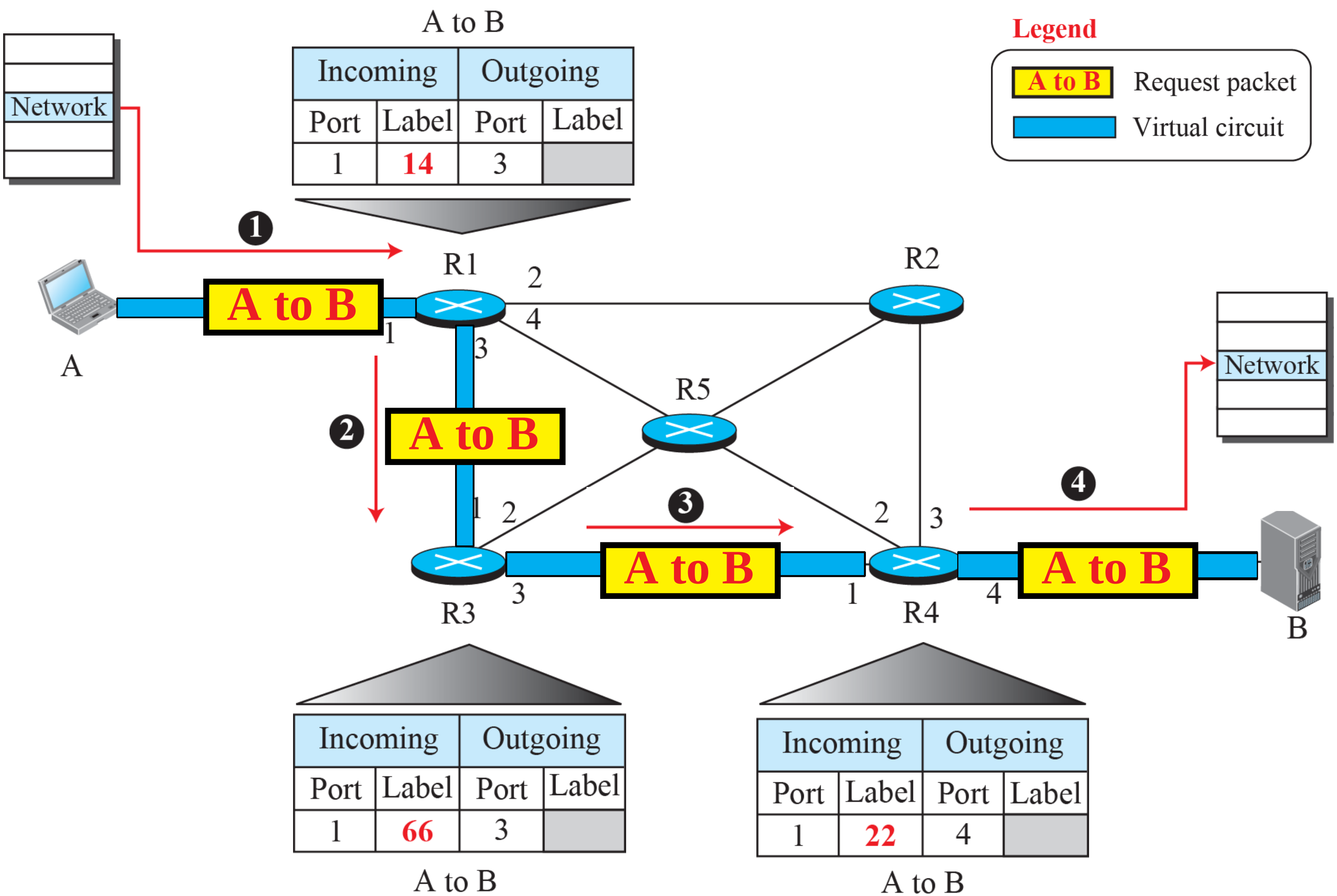


Figure 4.8: Sending acknowledgments in a virtual-circuit network

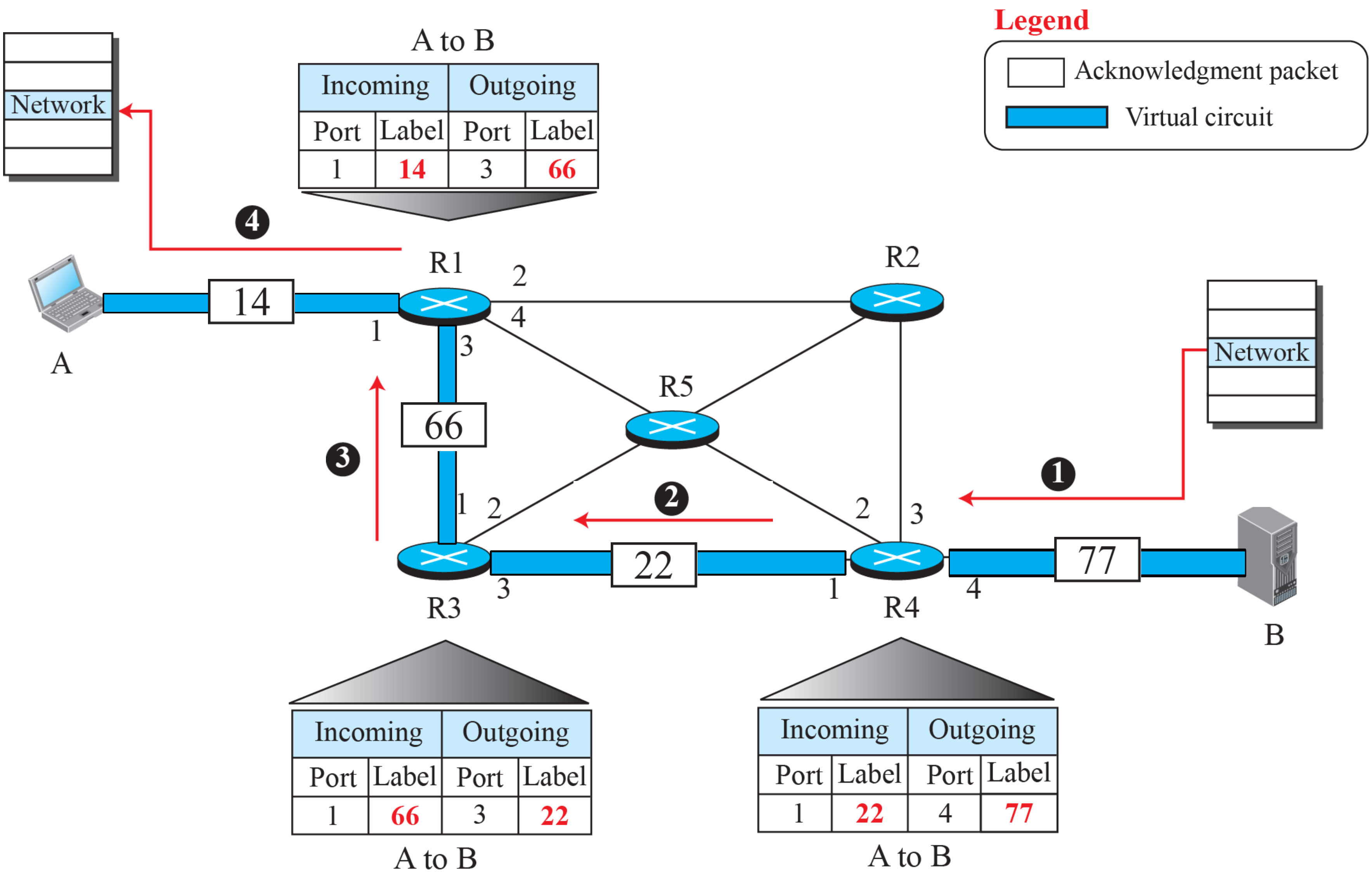
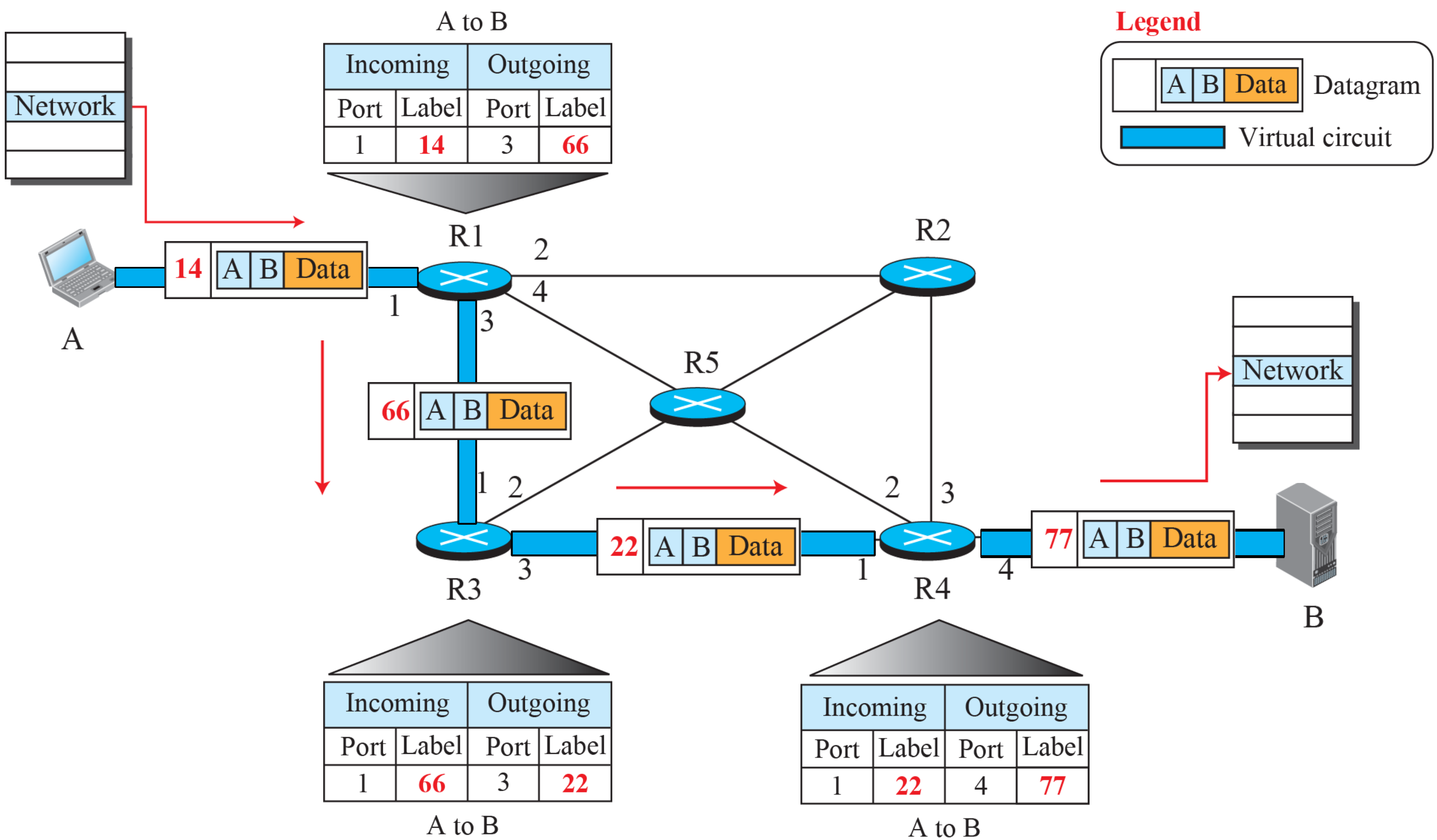


Figure 4.8: Sending acknowledgments in a virtual-circuit network



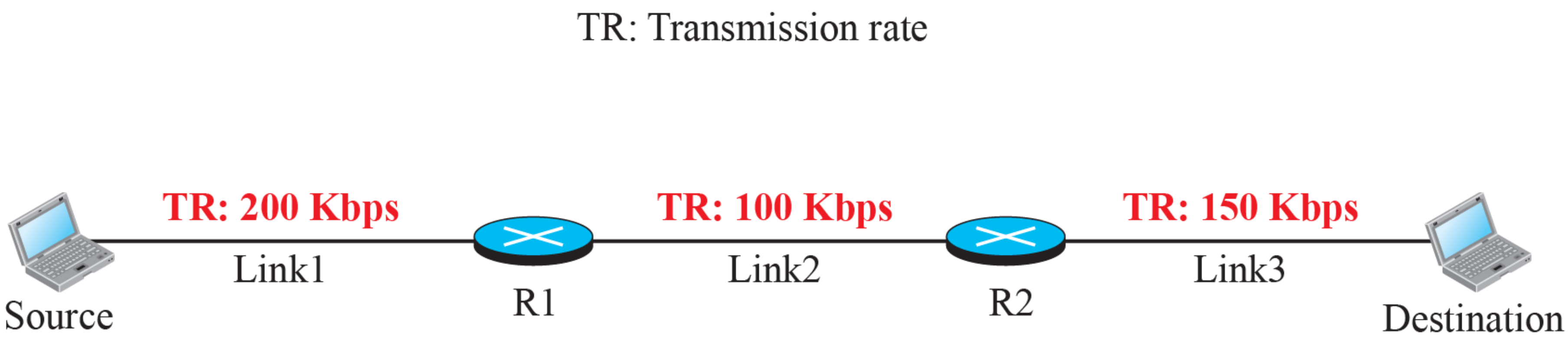
Network-Layer Performance

- ***The upper-layer protocols that use the service of the network layer expect to receive an ideal service, but the network layer is not perfect***

- ***The performance of a network can be measured in terms of delay, throughput, and packet loss.***

- ***Delay***
  - ***Transmission Delay***
  - ***Propagation Delay***
  - ***Processing Delay***
  - ***Queuing Delay***
- ***Throughput***
- ***Packet Loss***

Figure 4.10: ***Throughput in a path with three links in a series***



a. A path through three links

b. Simulation using pipes

Figure 4.11: A path through the Internet backbone

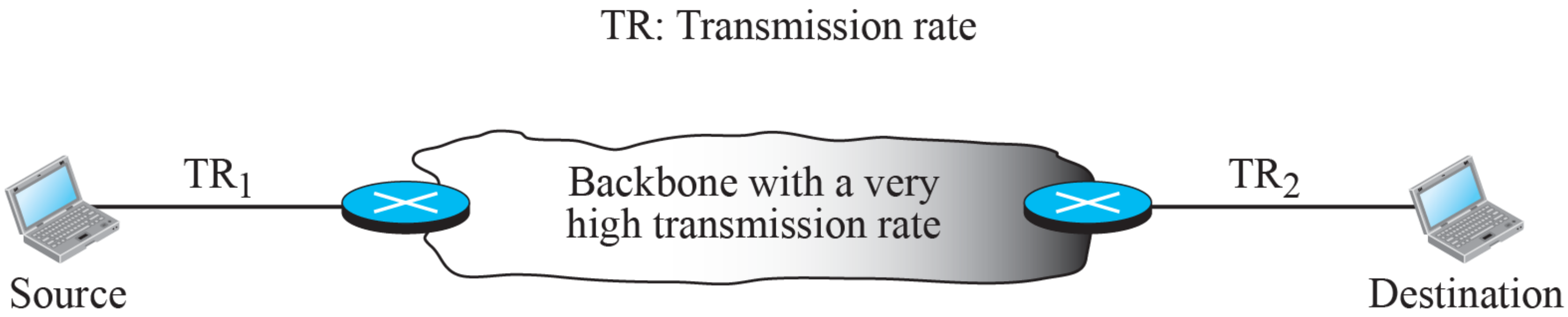
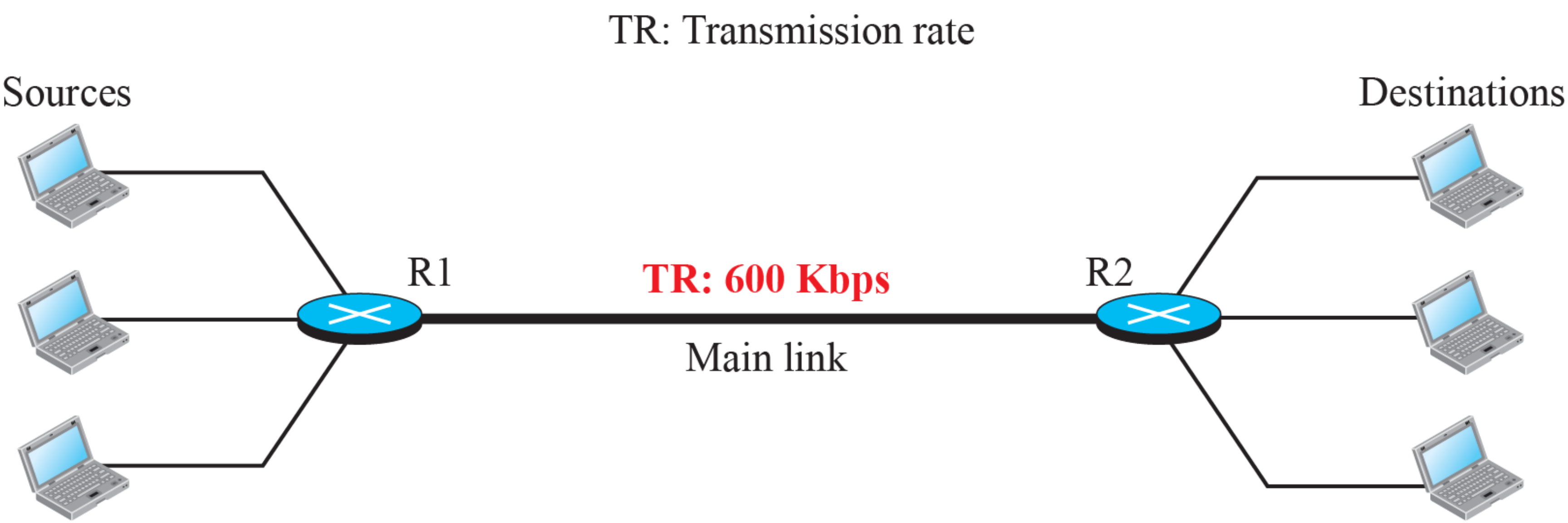


Figure 4.12: ***Effect of throughput in shared links***



Structure of A Router

- ***accepts incoming packets from one of the input ports (interfaces)***

- ***uses a forwarding table to find the output port from which the packet departs***

- ***sends the packet from this output port.***

Components

- Input Ports
- Output Ports
- Routing Processor
- Switching Fabrics
  - Crossbar Switch
  - Banyan Switch
  - Batcher-Banyan Switch

Figure 4.16: ***Router components***

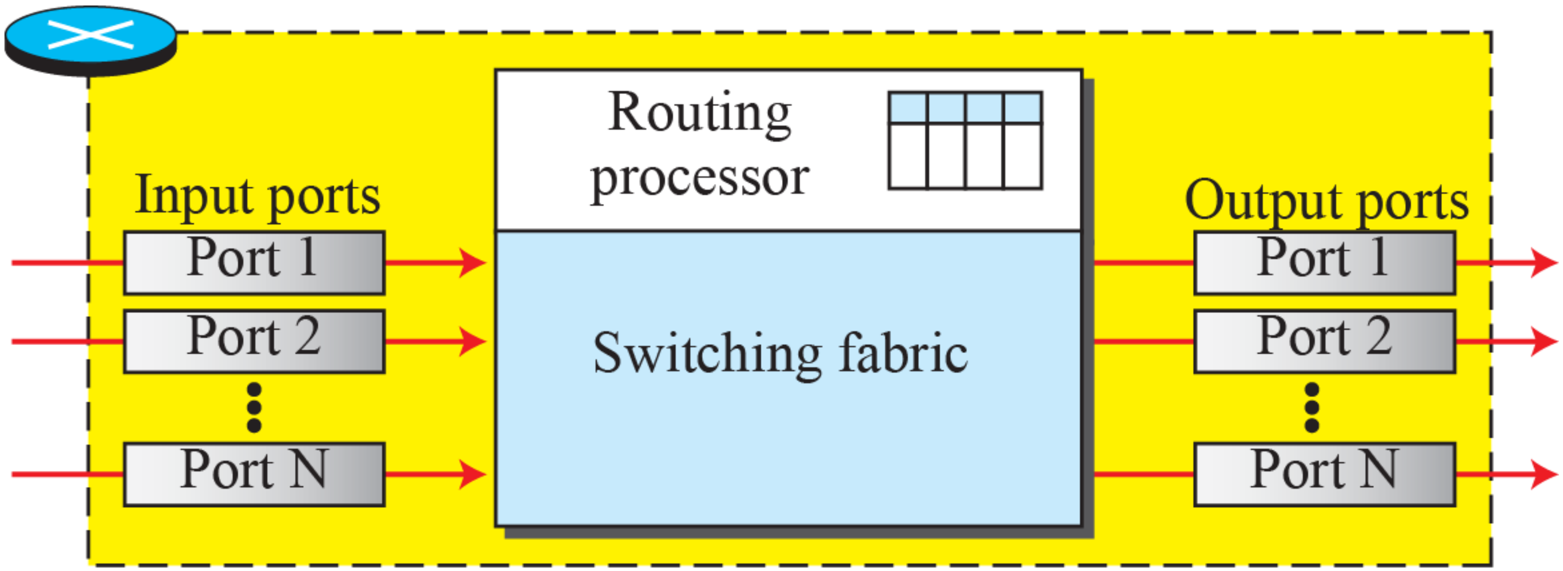


Figure 4.17: ***Input port***

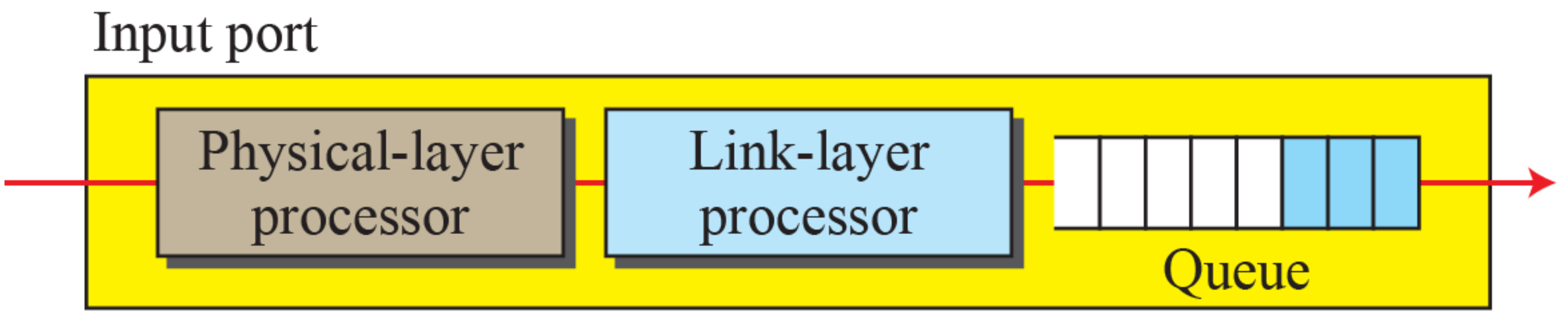


Figure 4.18: Output port

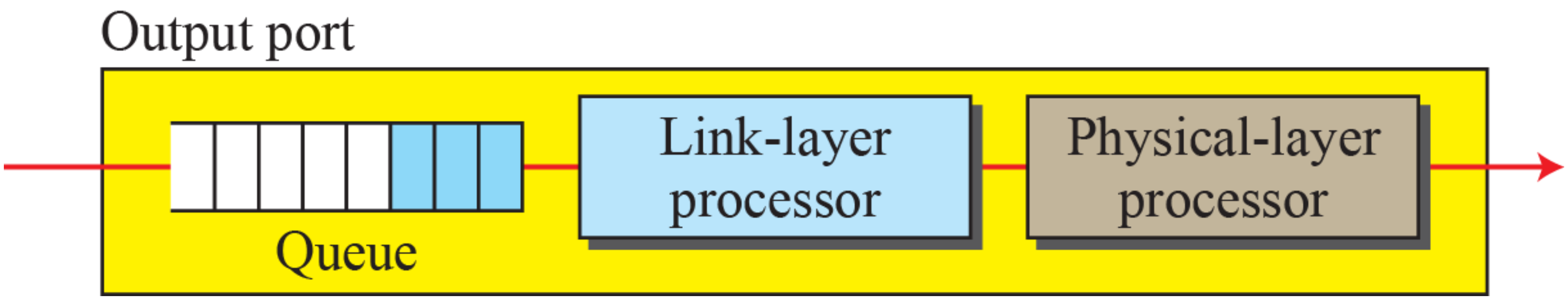


Figure 4.19: ***Crossbar switch***

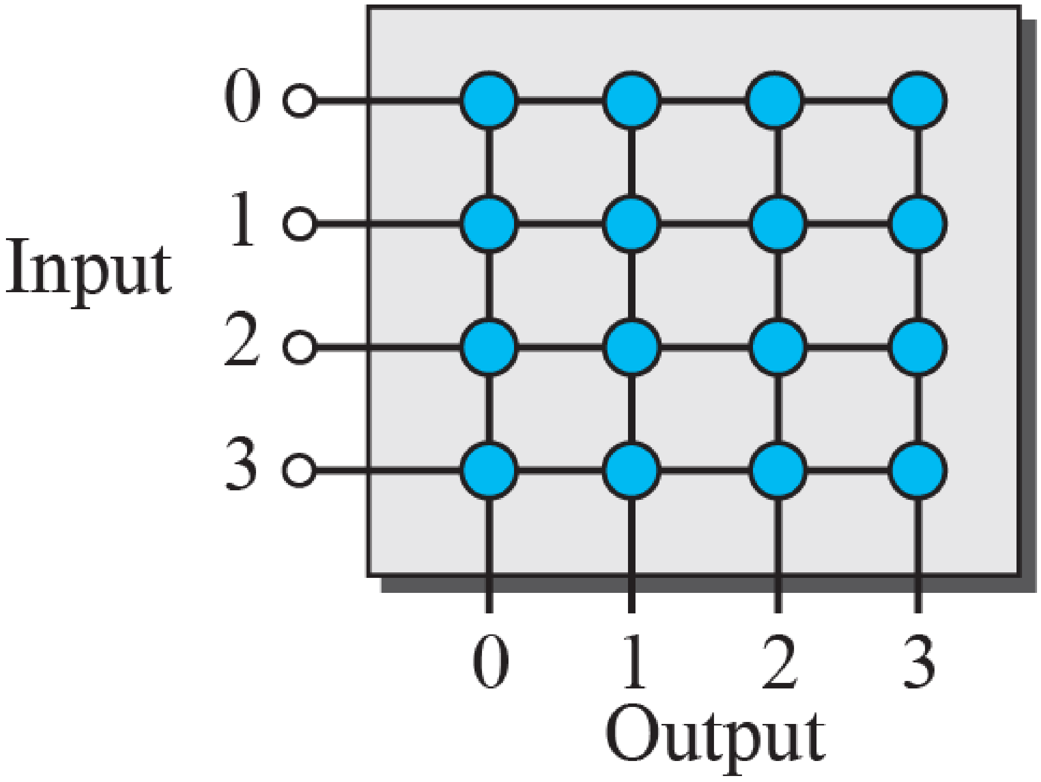


Figure 4.20: **Banyan switch**

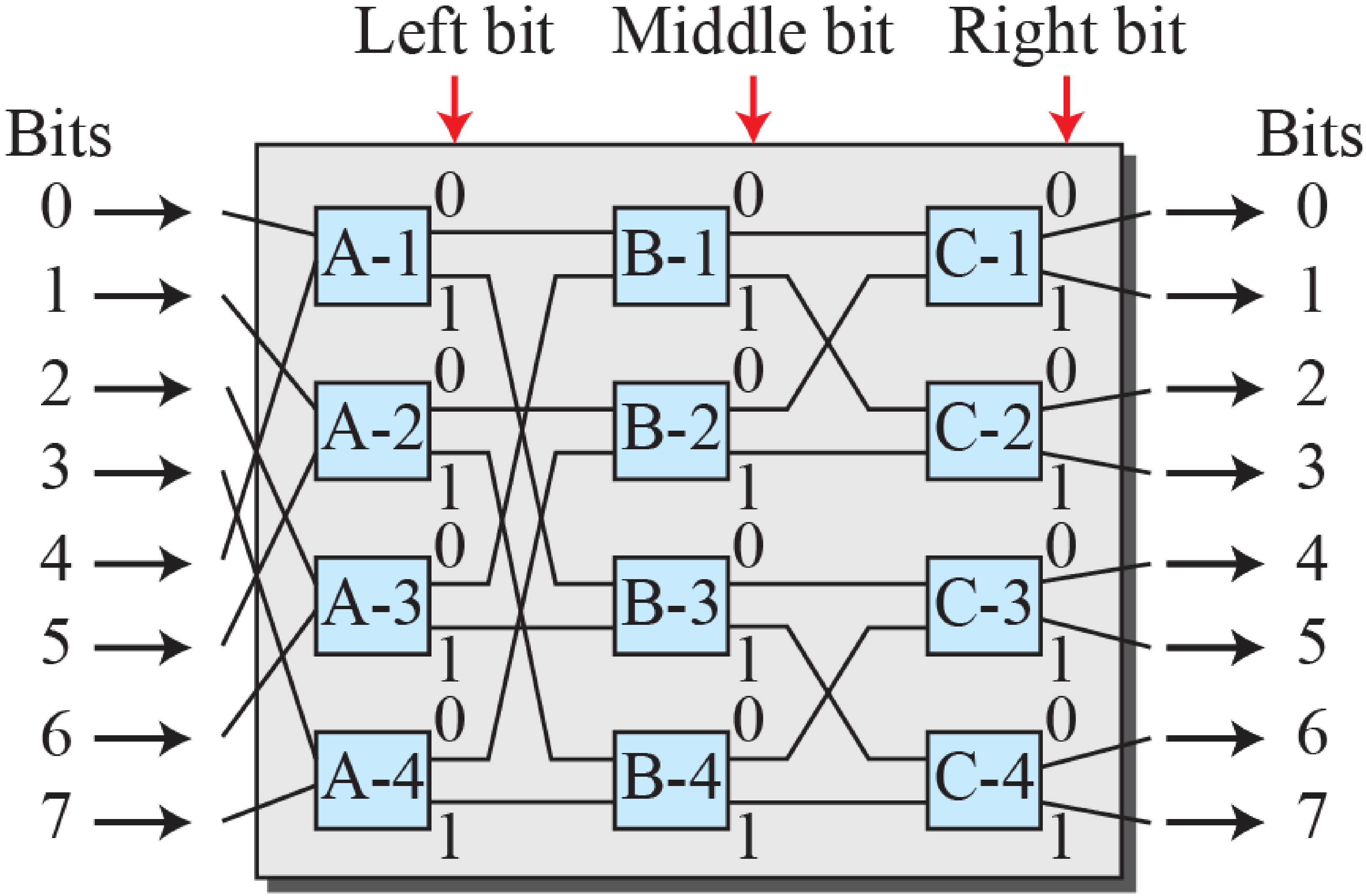
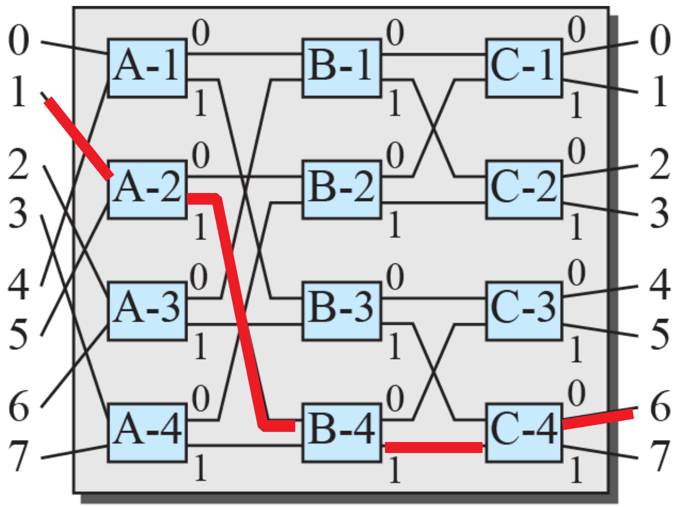
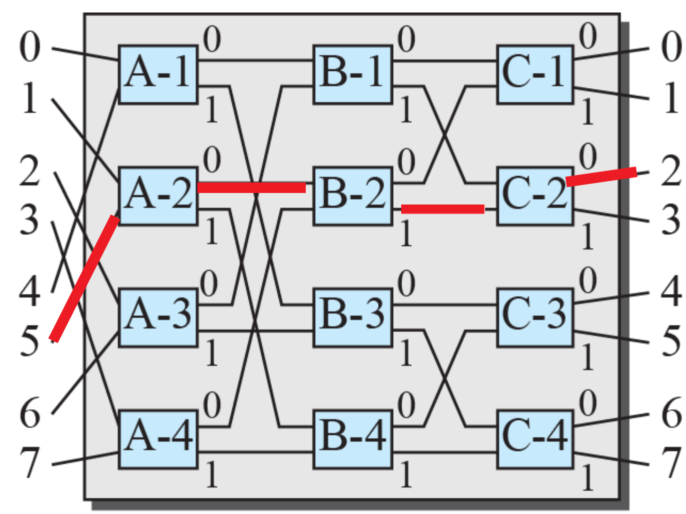


Figure 4.21: Examples of routing in a banyan switch



a. Input 1 sending to output 6 (110)



b. Input 5 sending to output 2 (010)

Figure 4.22: *Batcher-banyan switch*

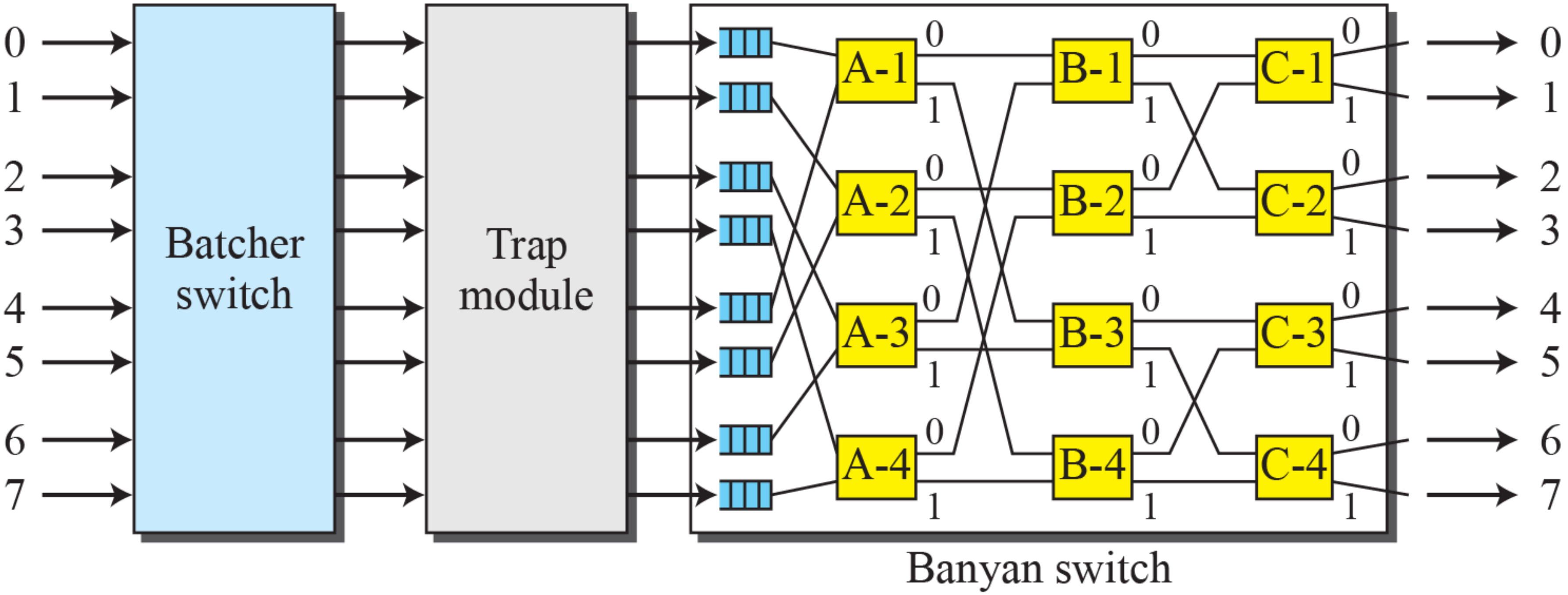


Figure 4.23: Position of IP and other network-layer protocols in TCP/IP protocol suite

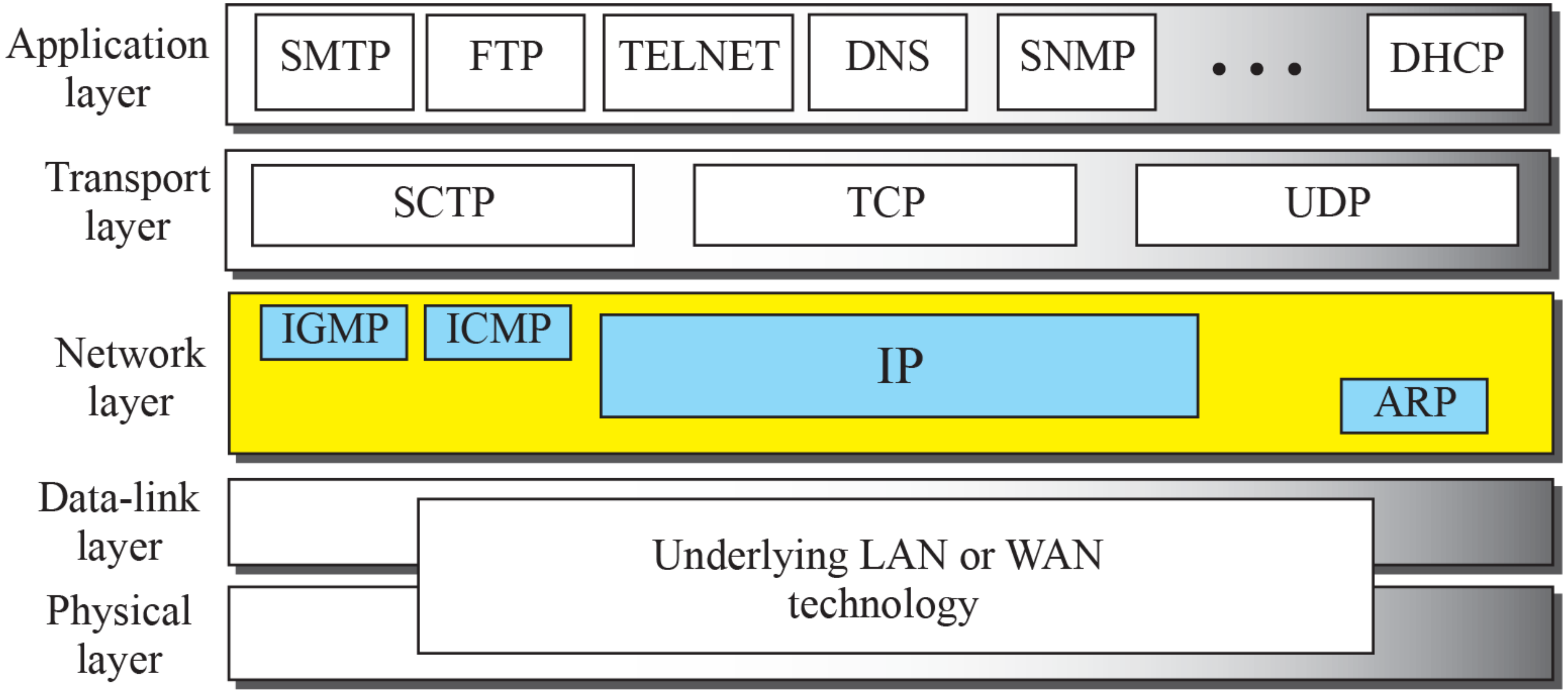


Figure 4.24: *IP datagram*

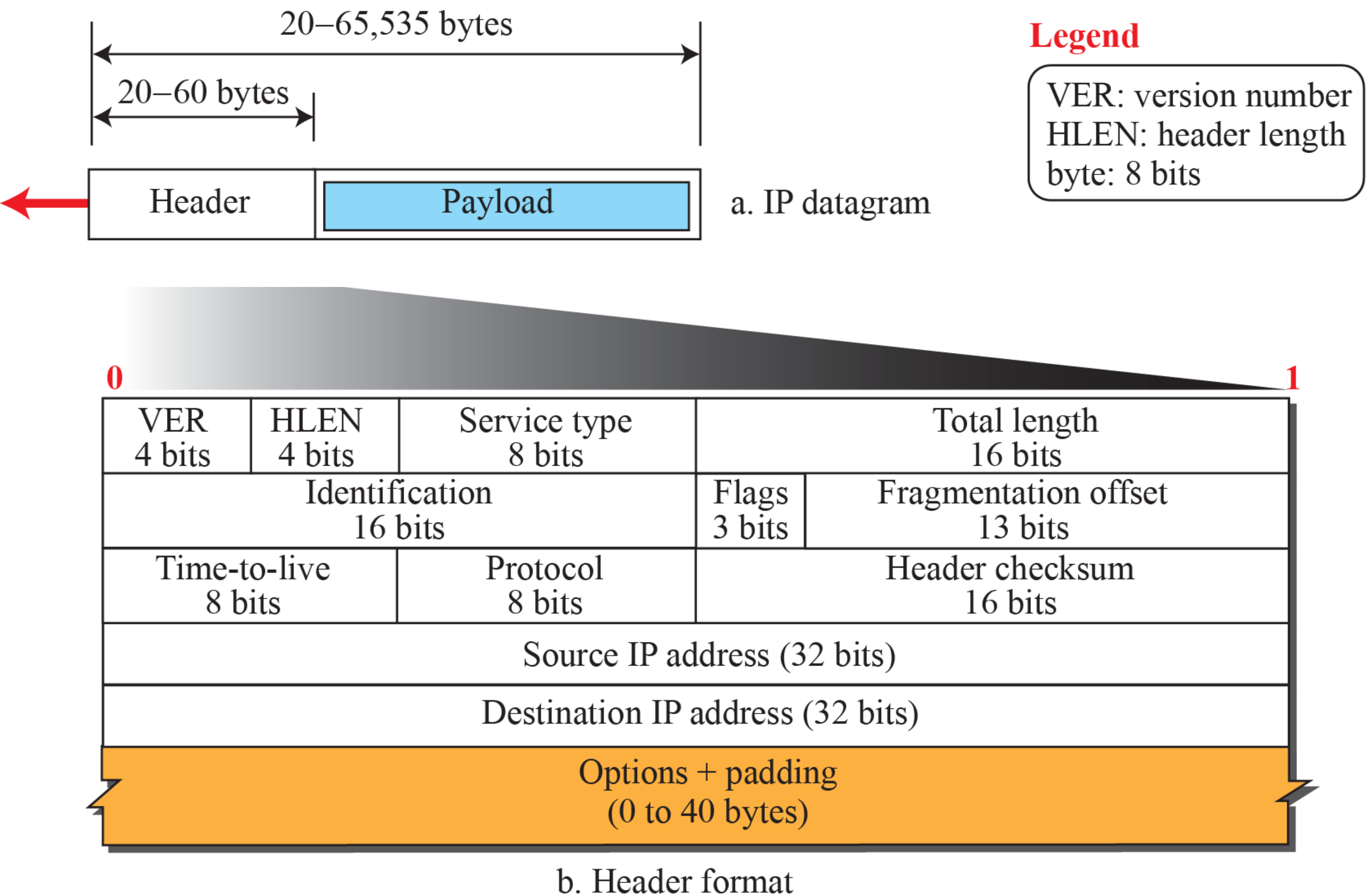


Figure 4.25: Multiplexing and demultiplexing using the value of the protocol field

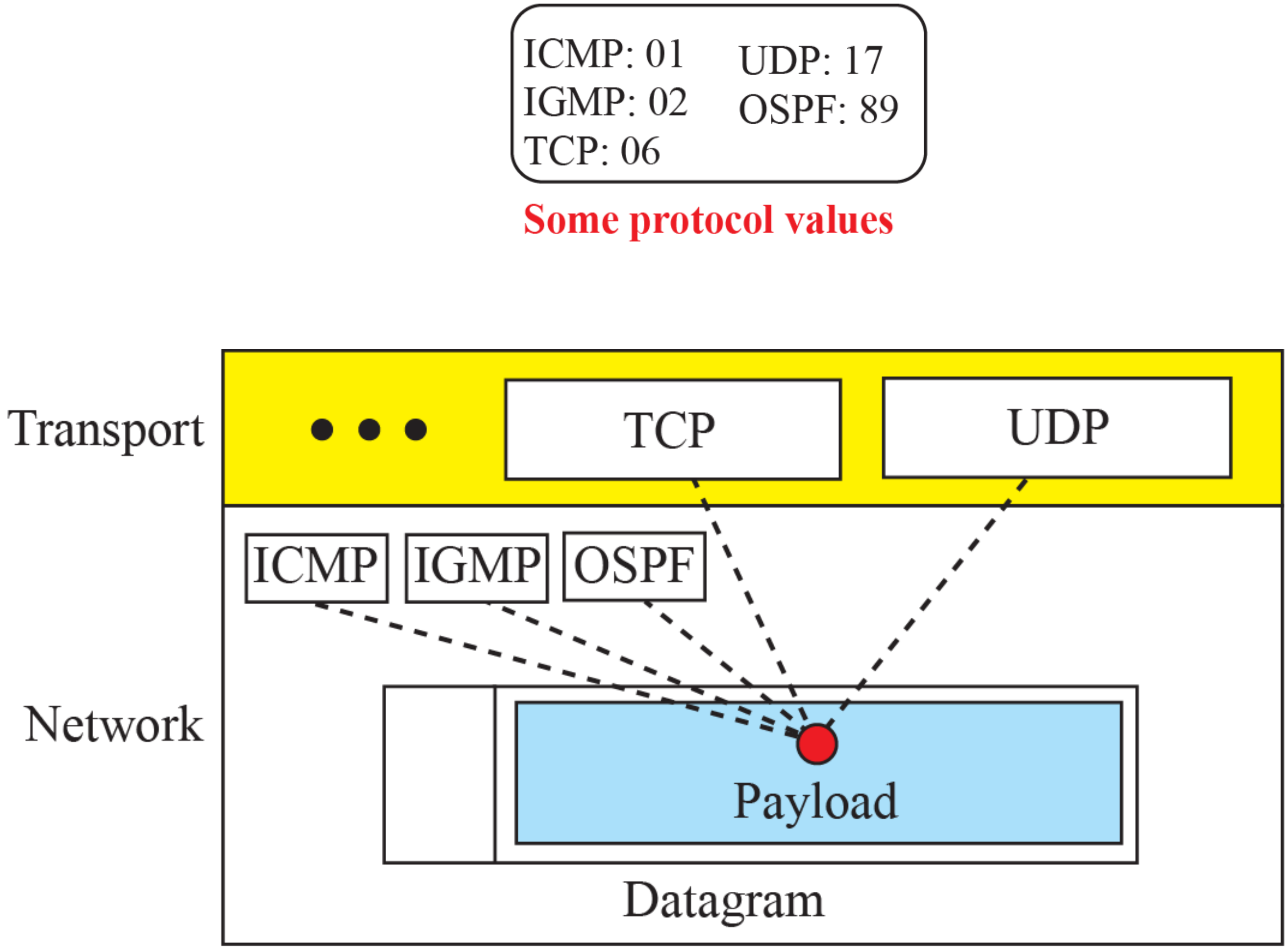


Figure 4.26: Maximum transfer unit (MTU)

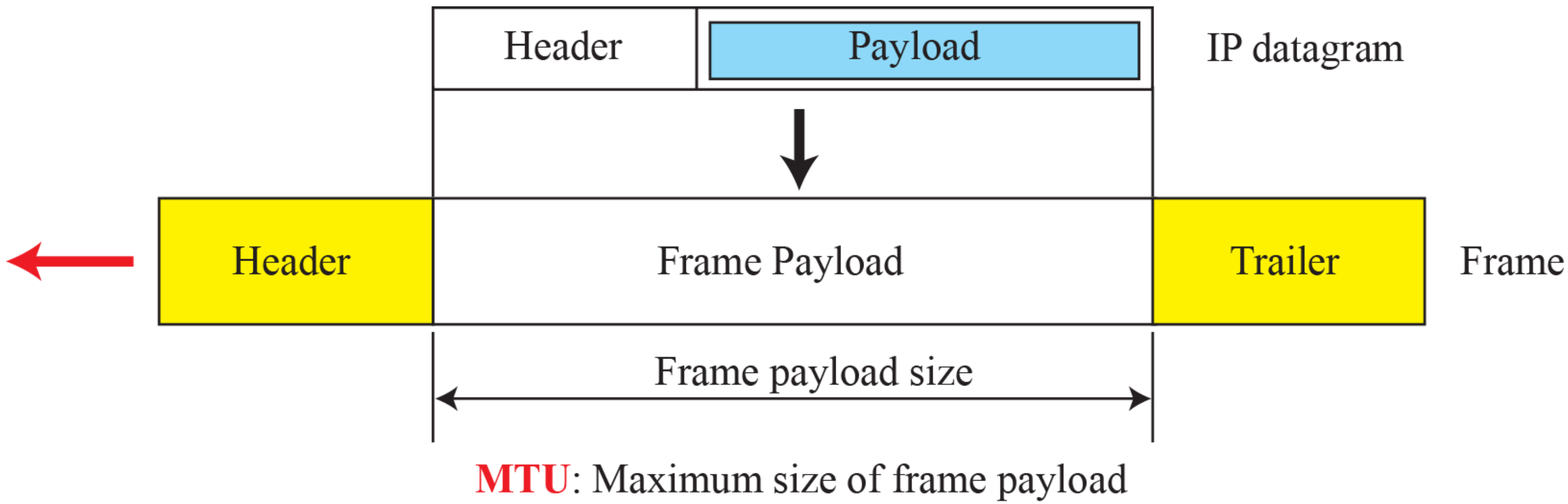


Figure 4.27: Fragmentation example

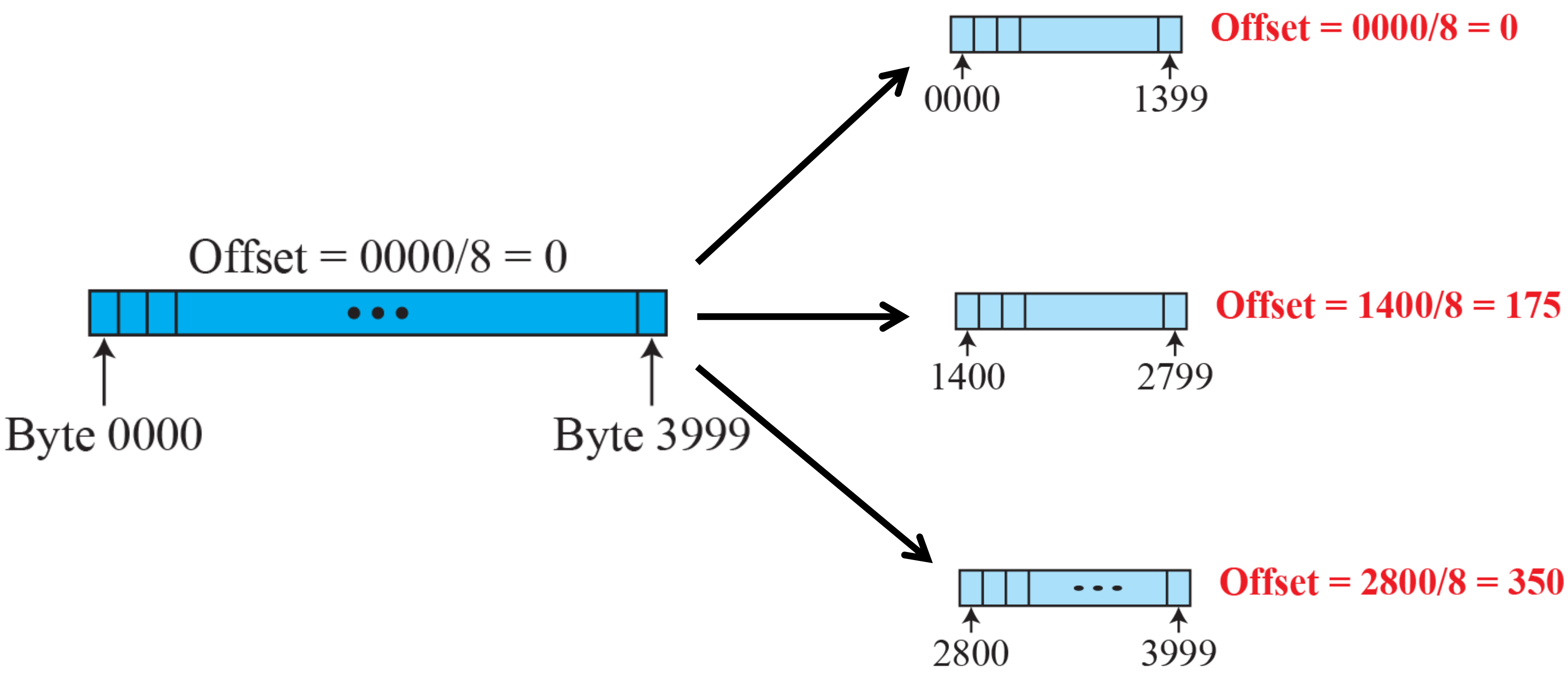


Figure 4.28: Detailed fragmentation example

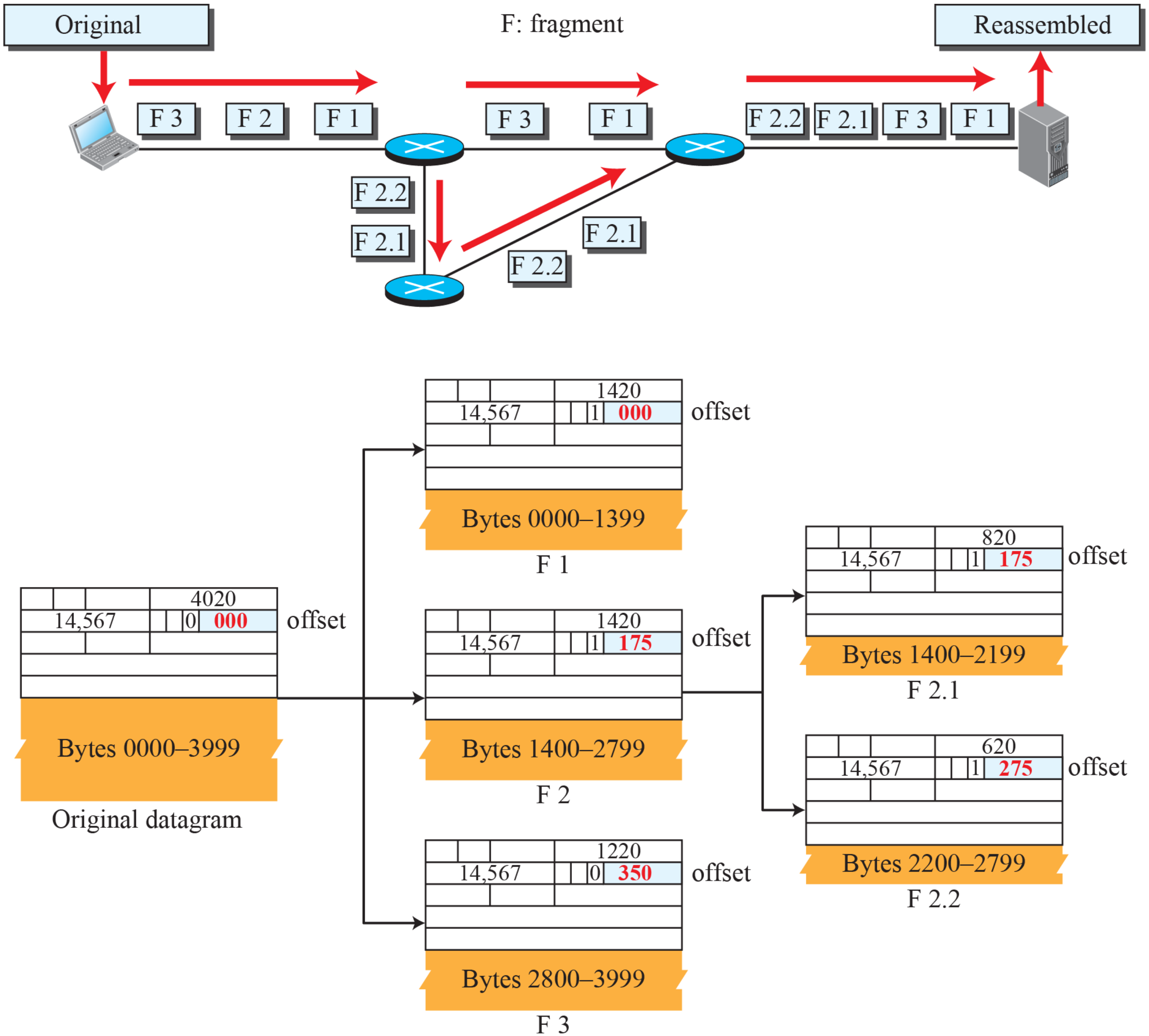


Figure 4.29: Three different notations in IPv4 addressing

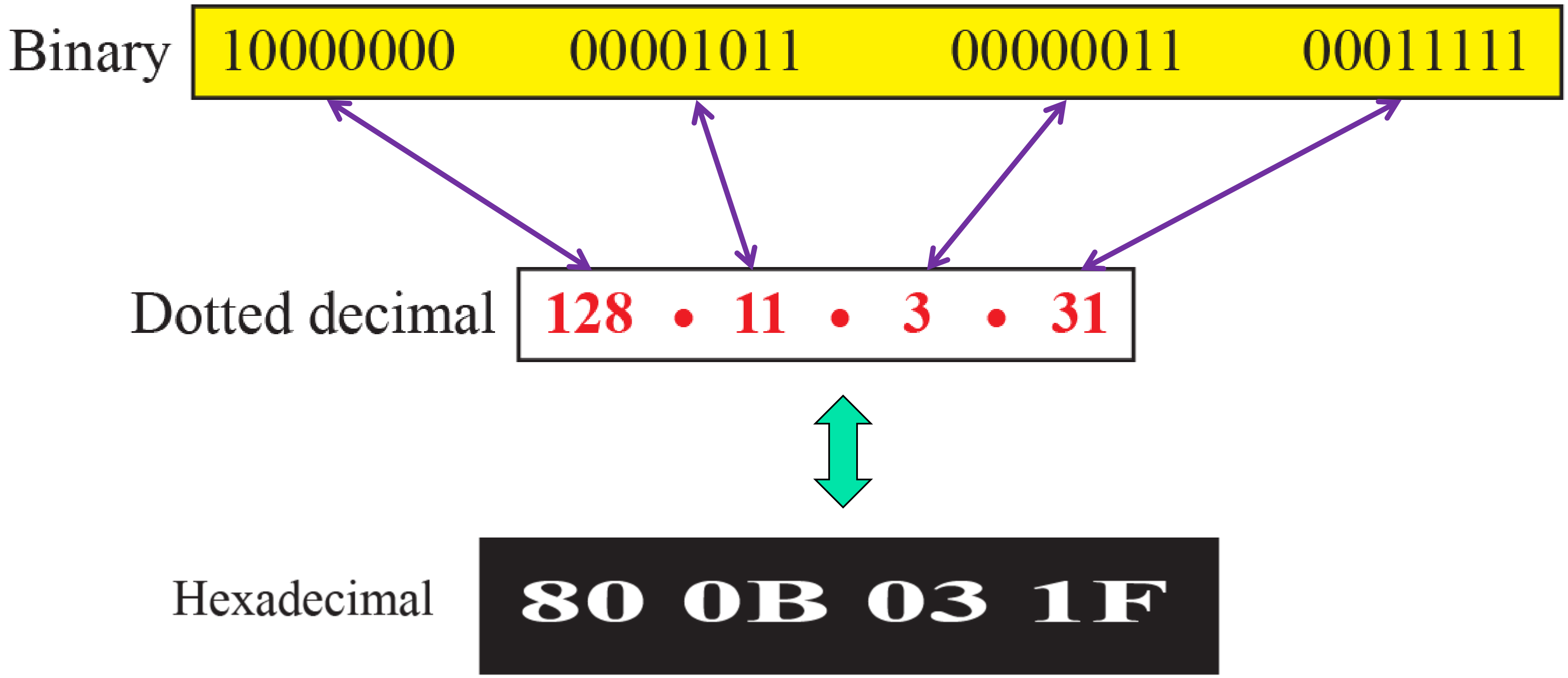


Figure 4.30: ***Hierarchy in addressing***

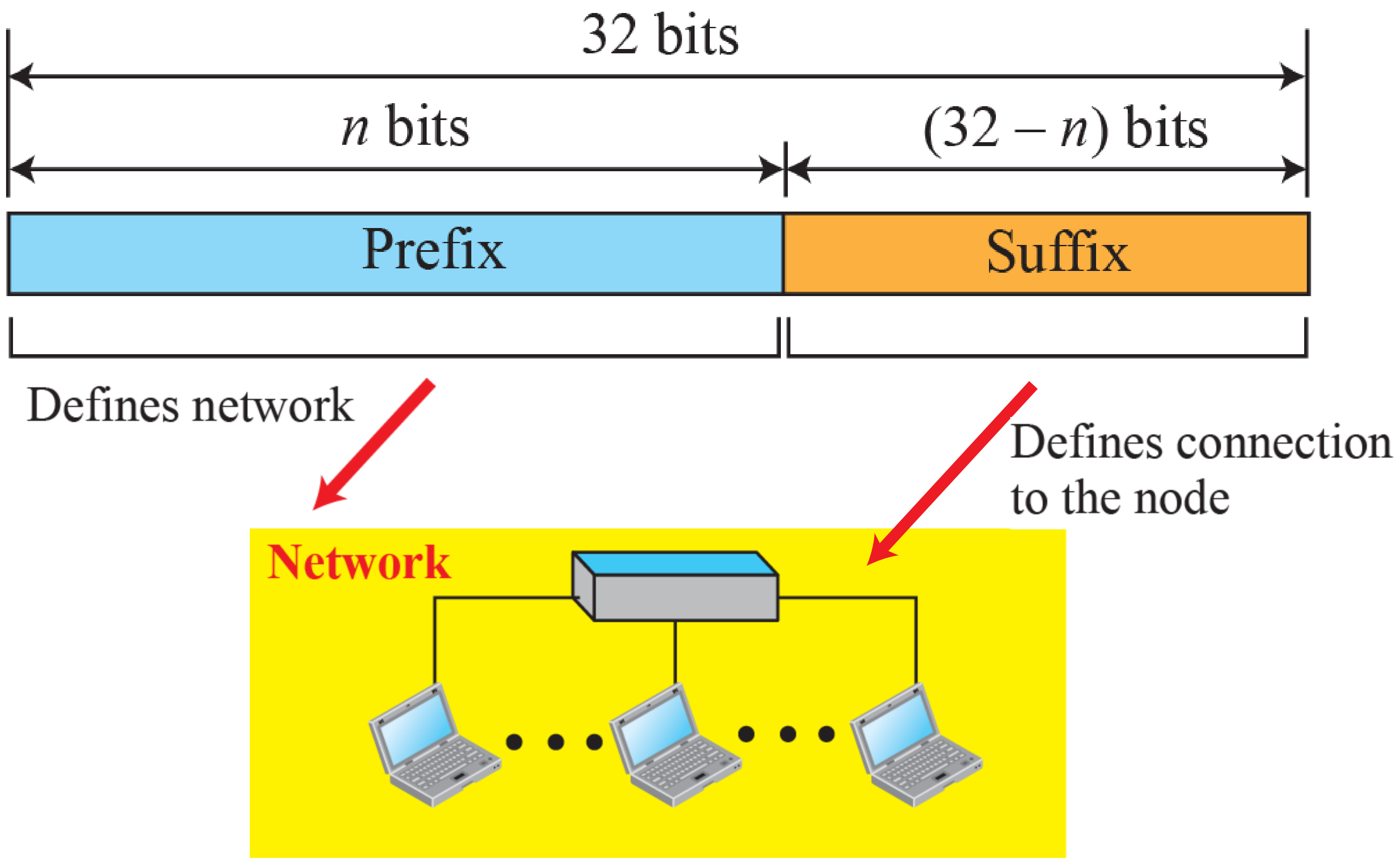
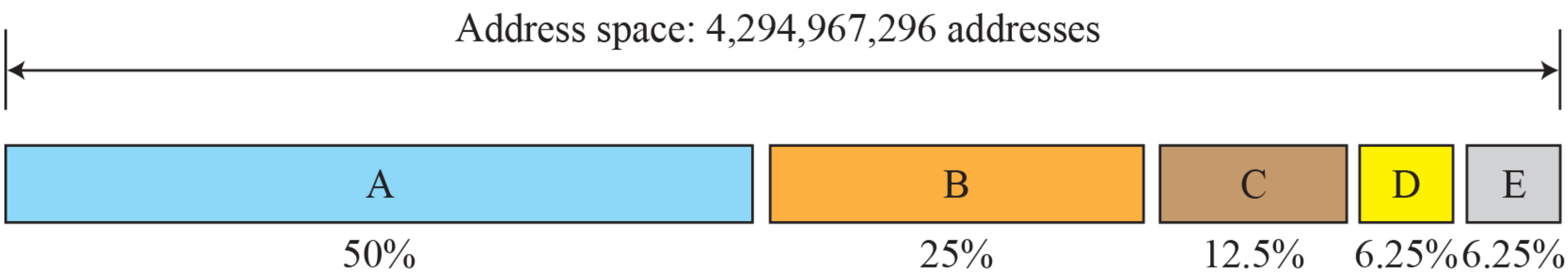
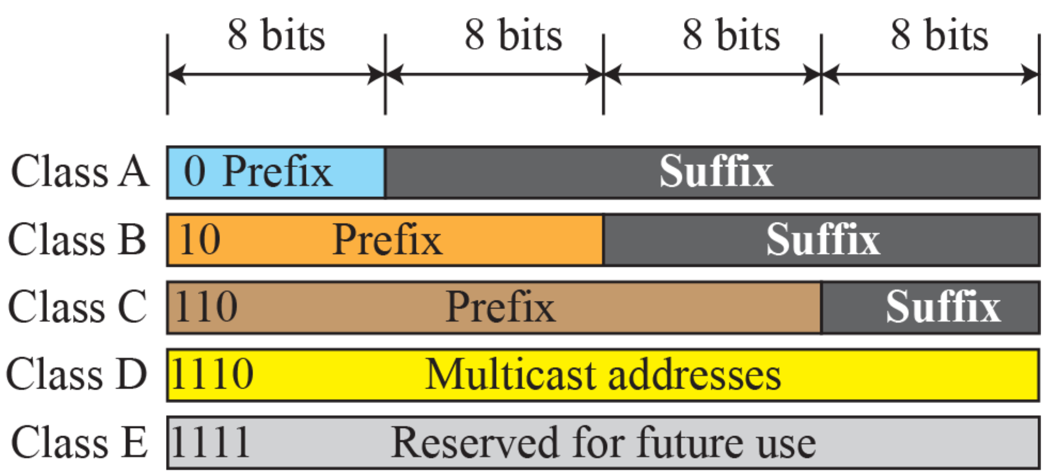


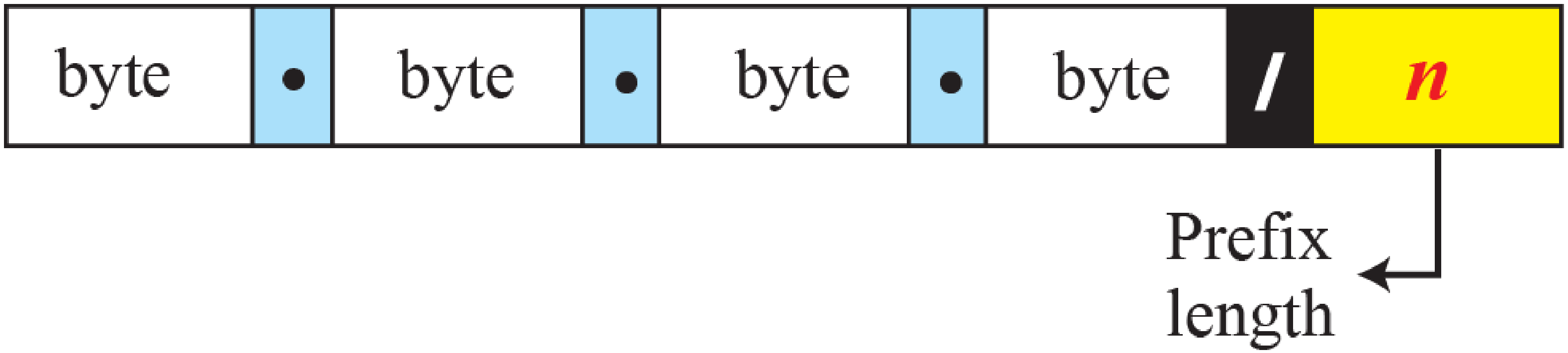
Figure 4.31: Occupation of the address space in classful addressing





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

Figure 4.33: Slash notation (CIDR)



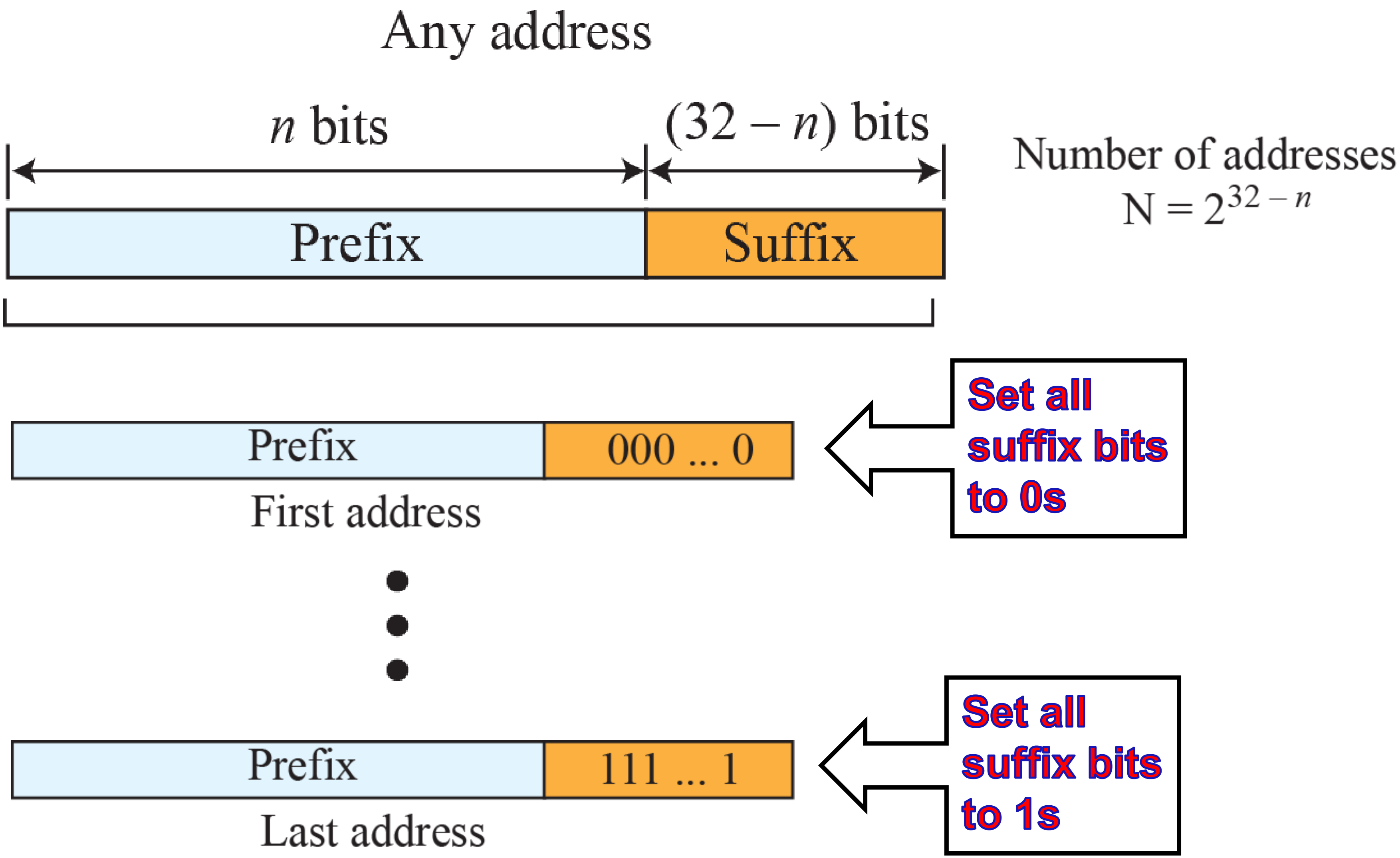
Examples:

12.24.76.8/**8**

23.14.67.92/**12**

220.8.24.255/**25**

Figure 4.34: Information extraction in classless addressing



Example 4.1

A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows. The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses. The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

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

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

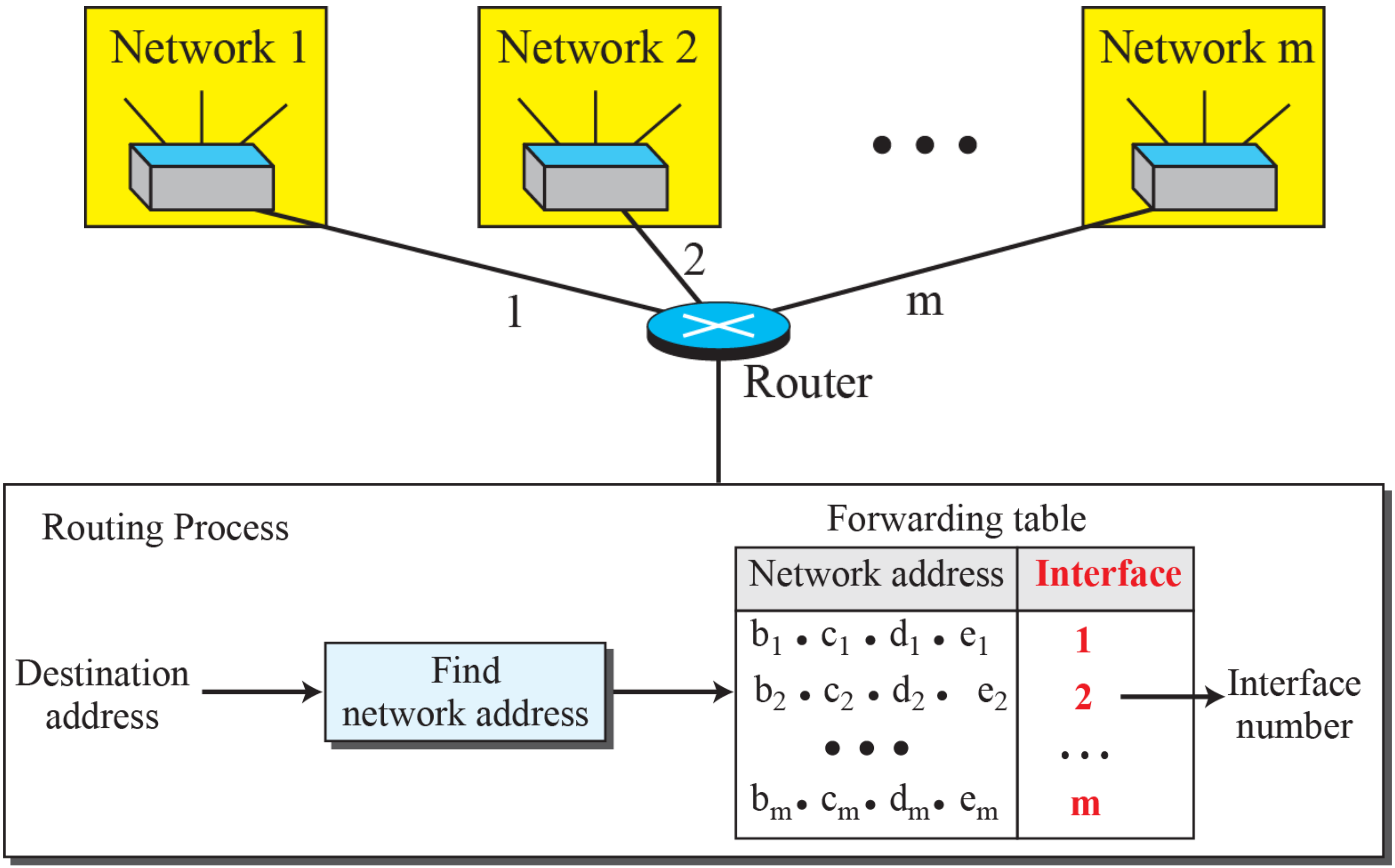
Address: 167.199.170.82/**27**	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/**27**	10100111	11000111	10101010	01011111

Example 4.2

We repeat Example 4.1 using the mask. The mask in dotted-decimal notation is 256.256.256.224 The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.

Number of addresses in the block:	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

Figure 4.35: Network address



Example 4.4

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

Example 4.5

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

Solution

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

Example 4.5 (continued)

a. 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**.

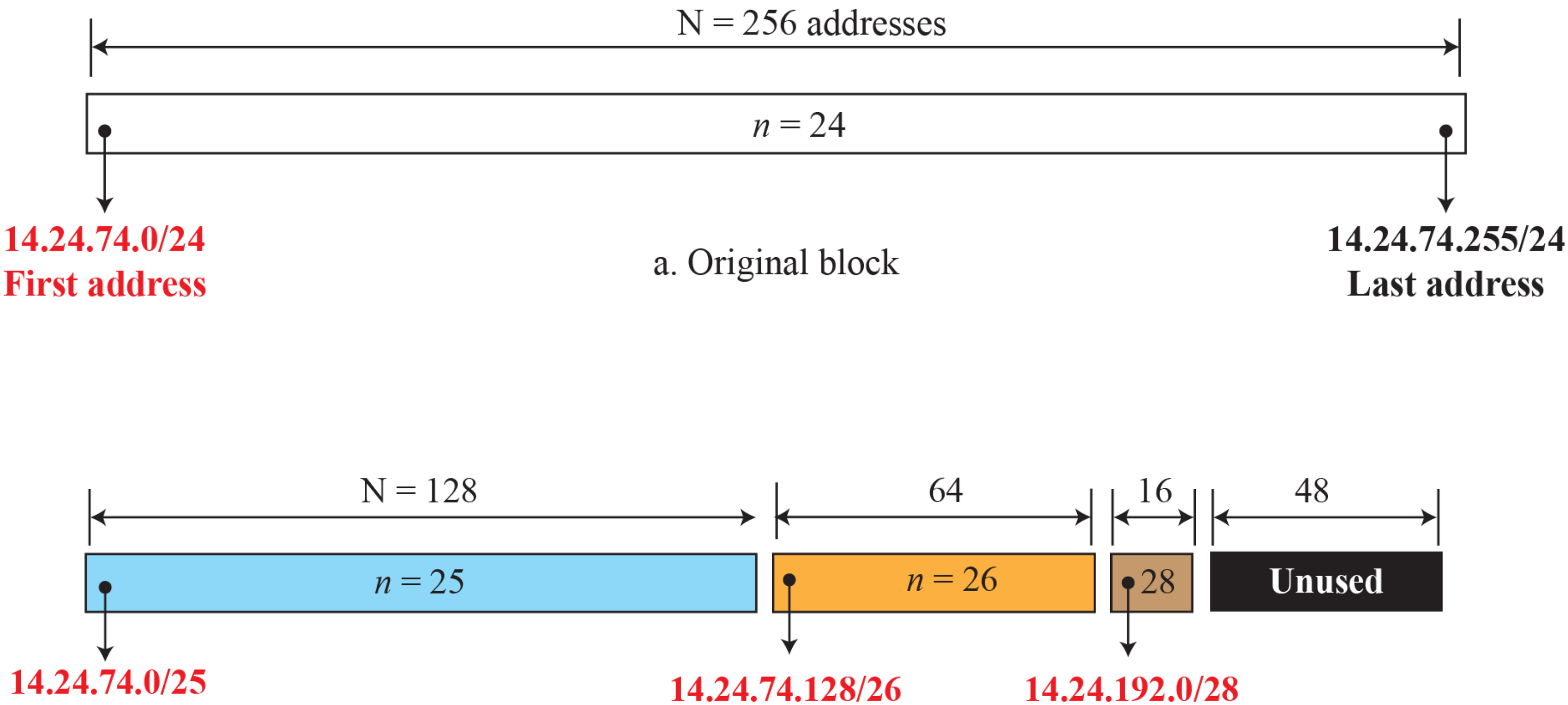
b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128**/26**; the last address is 14.24.74.191**/26**.

Example 4.5 (continued)

c. 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**.

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255. We don't know about the prefix length yet. Figure 4.36 shows the configuration of blocks. We have shown the first address in each block.

Figure 4.36: Solution to Example 4.5



a. Original block

b. Subblocks

Example 4.6

Figure 4.37 shows how four small blocks of addresses are assigned to four organizations by an ISP. The ISP combines these four blocks into one single block and advertises the larger block to the rest of the world. Any packet destined for this larger block should be sent to this ISP. It is the responsibility of the ISP to forward the packet to the appropriate organization. This is similar to routing we can find in a postal network. All packages coming from outside a country are sent first to the capital and then distributed to the corresponding destination.

Figure 4.37: Example of address aggregation

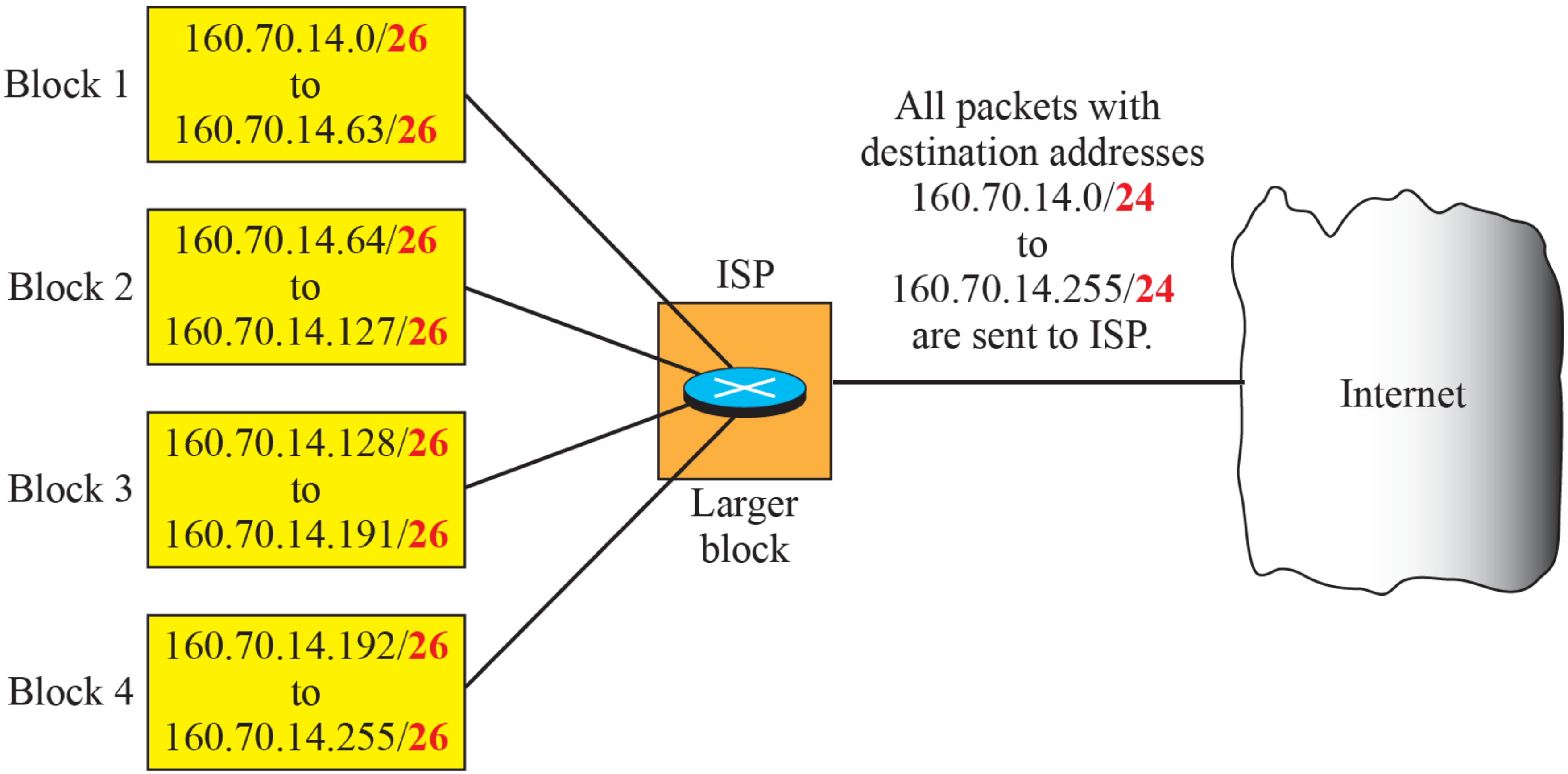
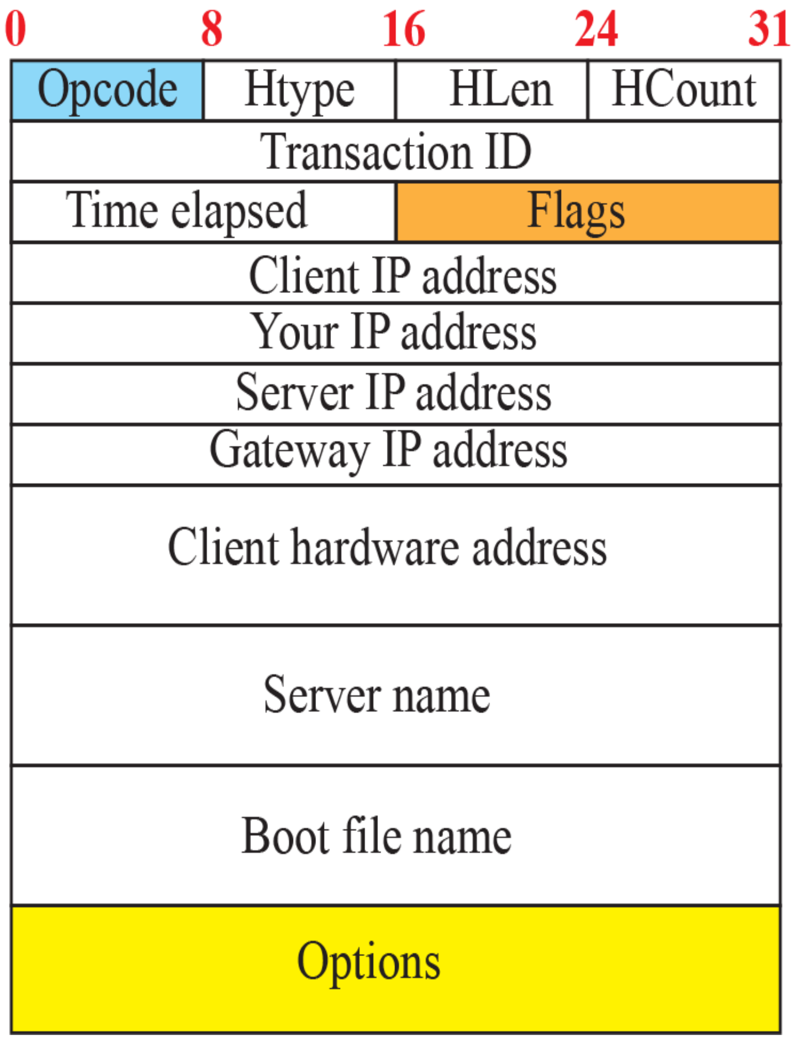


Figure 4.38: *DHCP message format*



Fields:

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

Figure 4.39: Option format

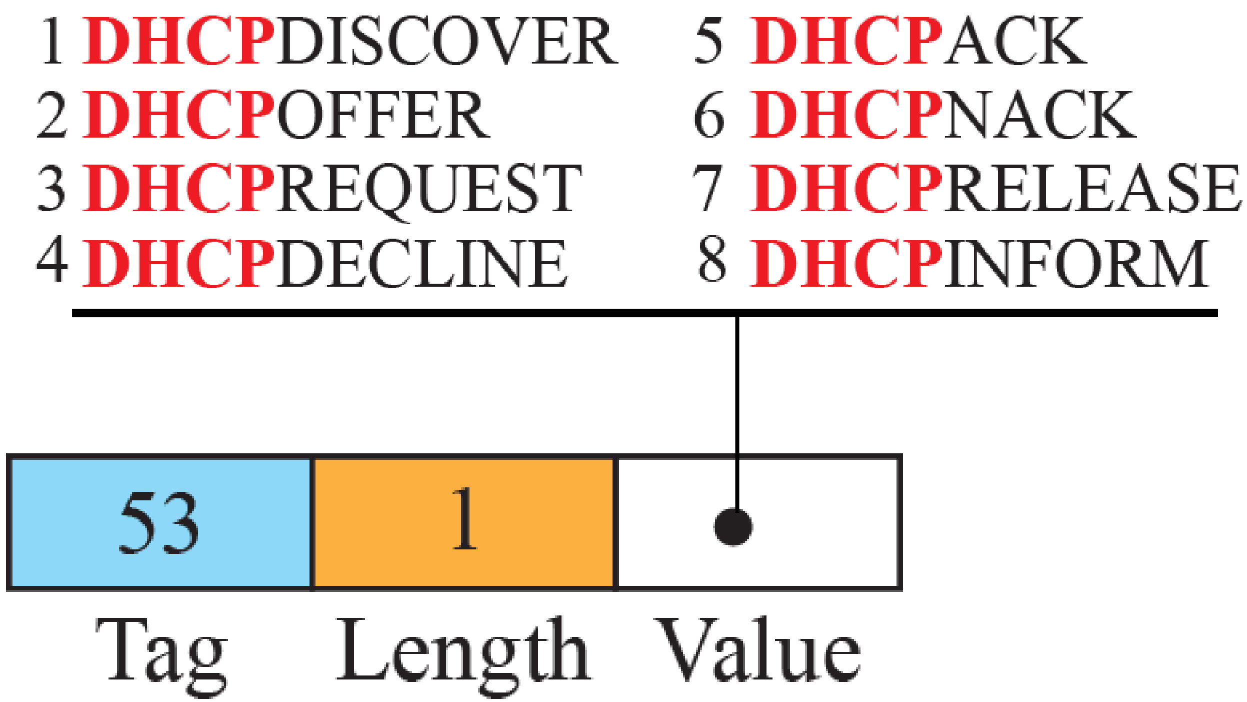


Figure 4.40: *Operation of DHCP*

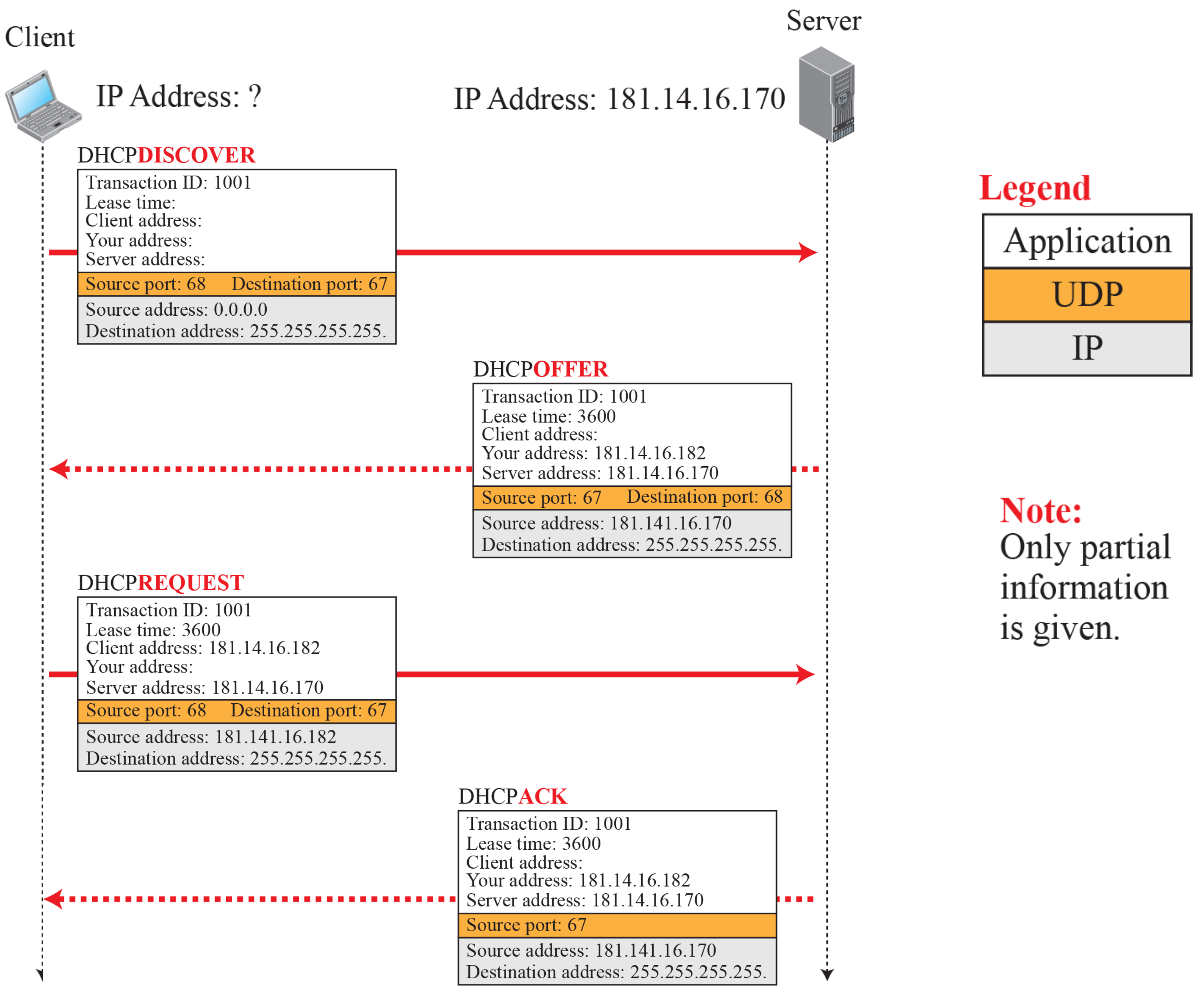
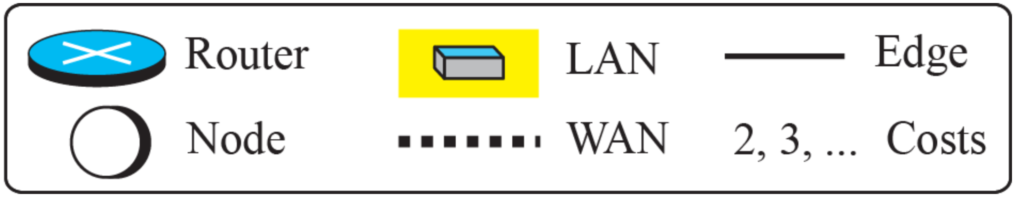
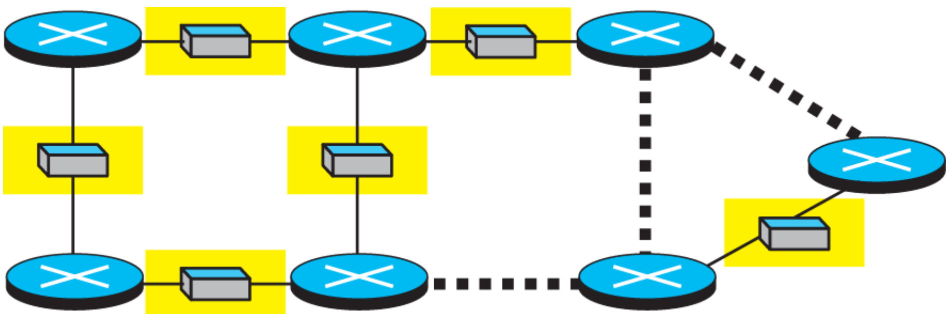


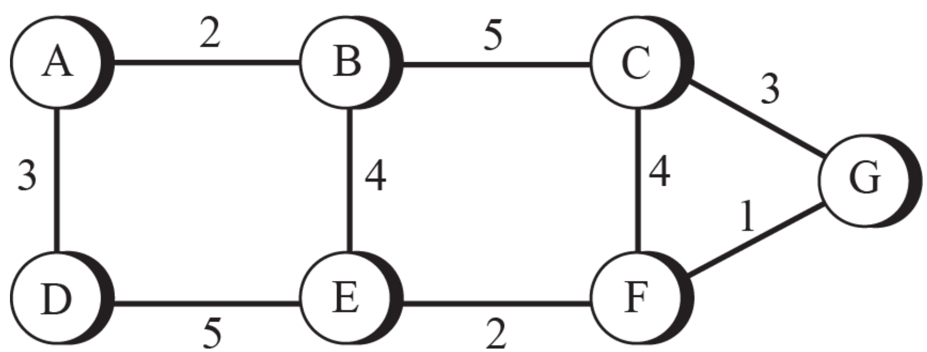
Figure 4.56: *An internet and its graphical representation*

Legend





a. An internet



b. The weighted graph

Figure 4.57: *Least-cost trees for nodes in the internet of Figure 4.56*

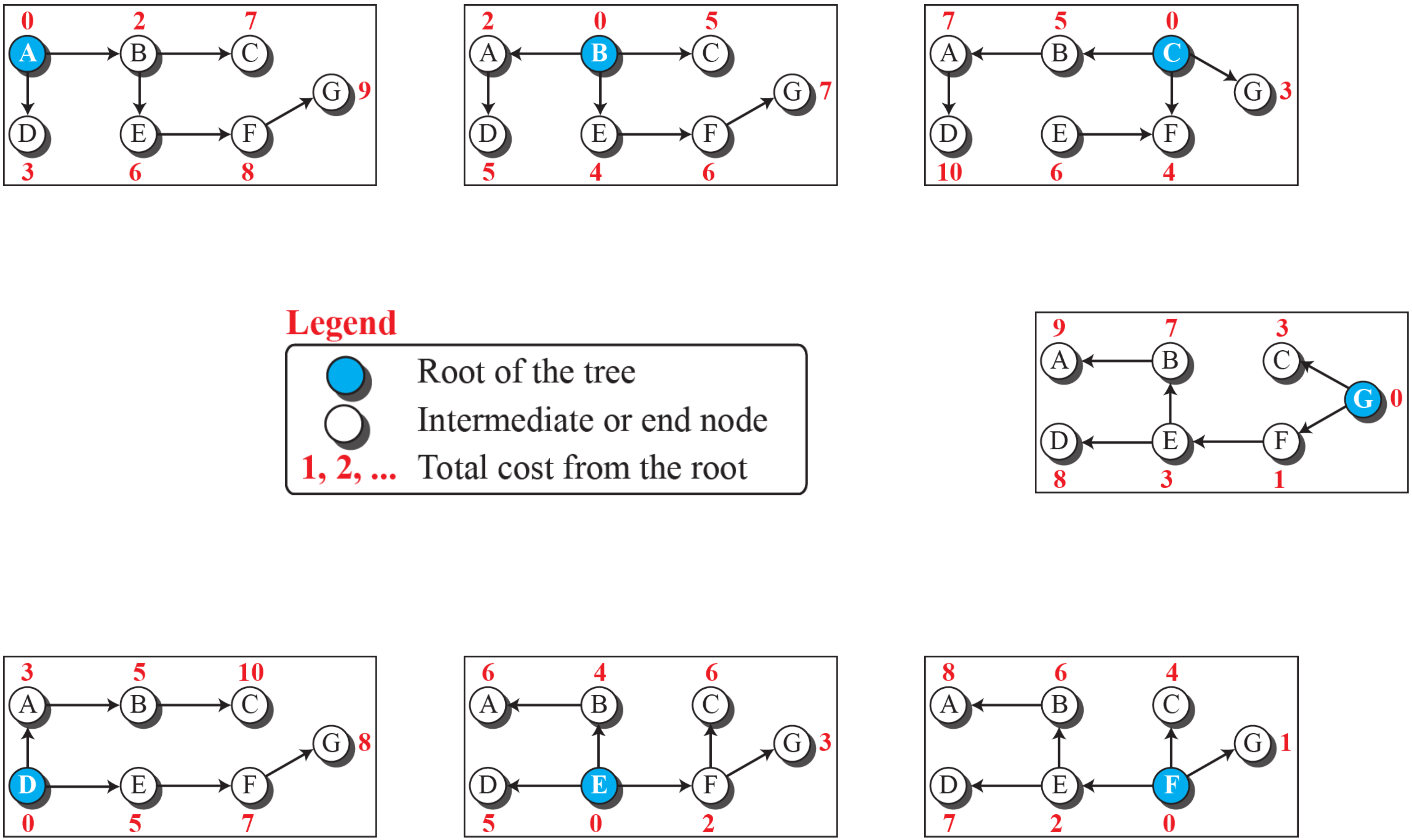
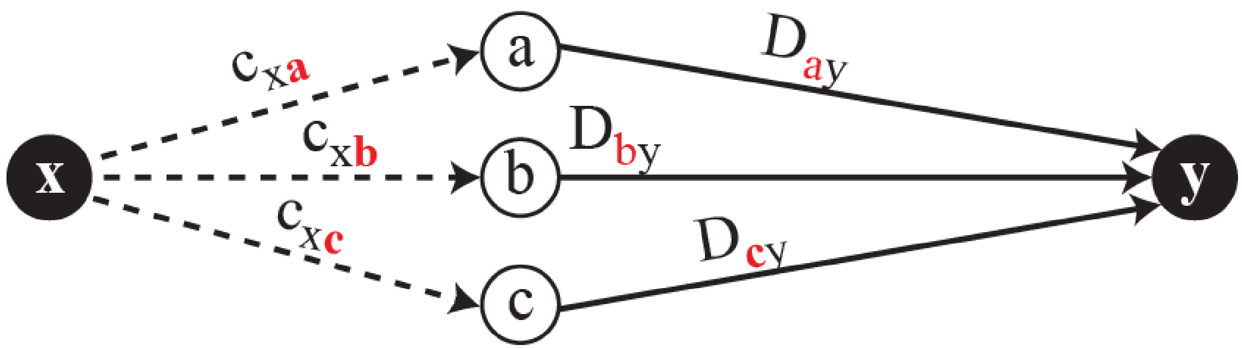
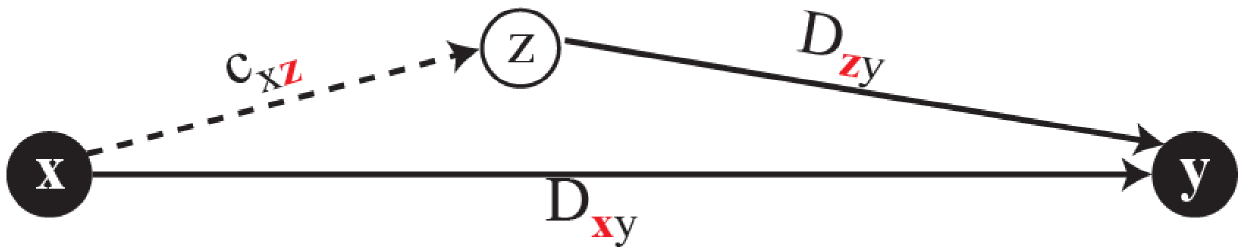


Figure 4.58: *Graphical idea behind Bellman-Ford equation*

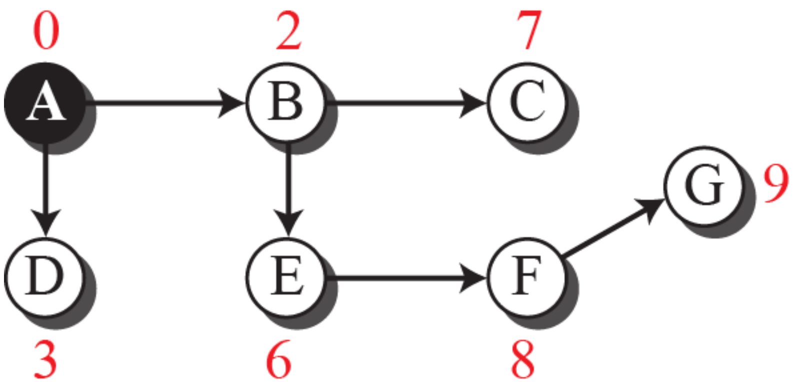


a. General case with three intermediate nodes



b. Updating a path with a new route

Figure 4.59: The distance vector corresponding to a tree



a. Tree for node A

	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

b. Distance vector for node A

Figure 4.60: *The first distance vector for an internet*

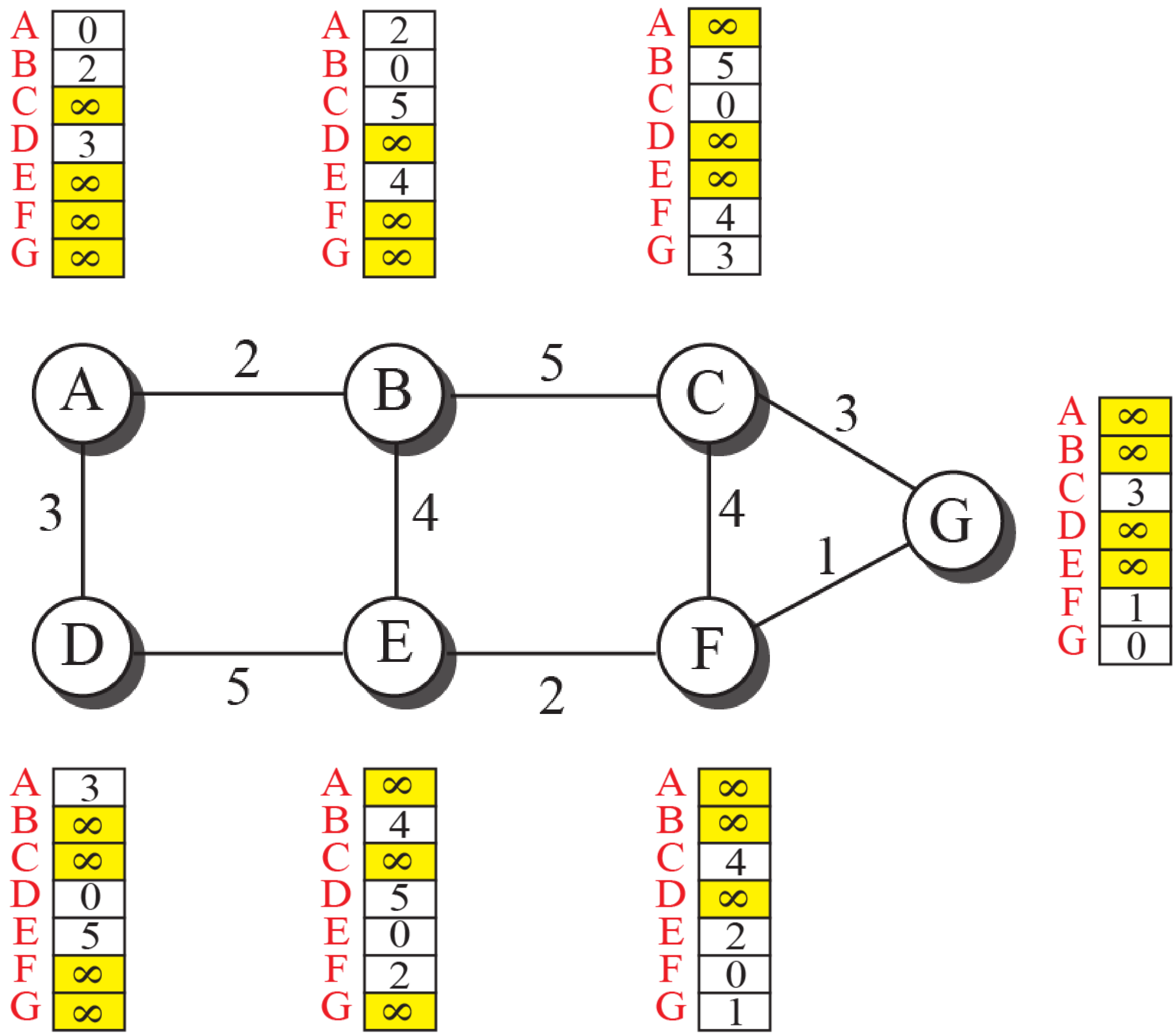


Figure 4.61: Updating distance vectors

	New B	Old B	A
A	2	2	0
B	0	0	2
C	5	5	∞
D	5	∞	3
E	4	4	∞
F	∞	∞	∞
G	∞	∞	∞

$B[\ ] = \min(B[\ ],\ 2 + A[\ ])$

a. First event: B receives a copy of A's vector.

Note:
X[ ]: the whole vector

	New B	Old B	E
A	2	2	∞
B	0	0	4
C	5	5	∞
D	5	5	5
E	4	4	0
F	6	∞	2
G	∞	∞	∞

$B[\ ] = \min(B[\ ],\ 4 + E[\ ])$

b. Second event: B receives a copy of E's vector.

Figure 4.62: Two-node instability

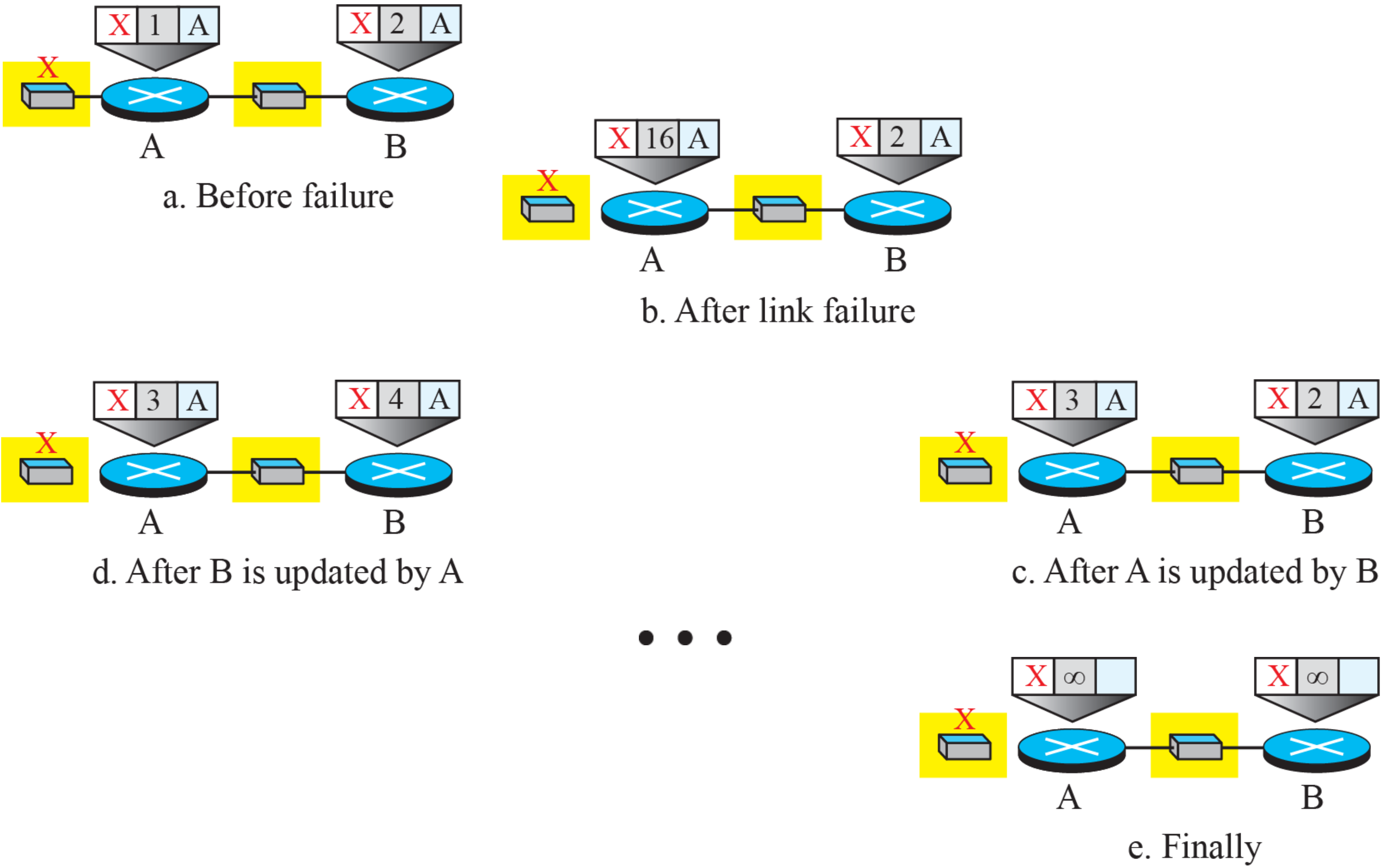
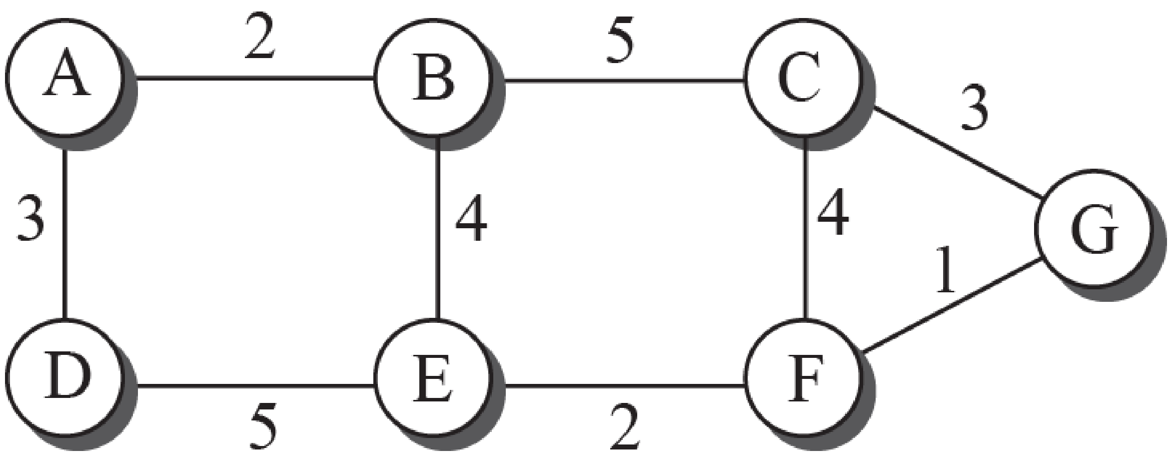


Figure 4.63: ***Example of a link-state database***



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

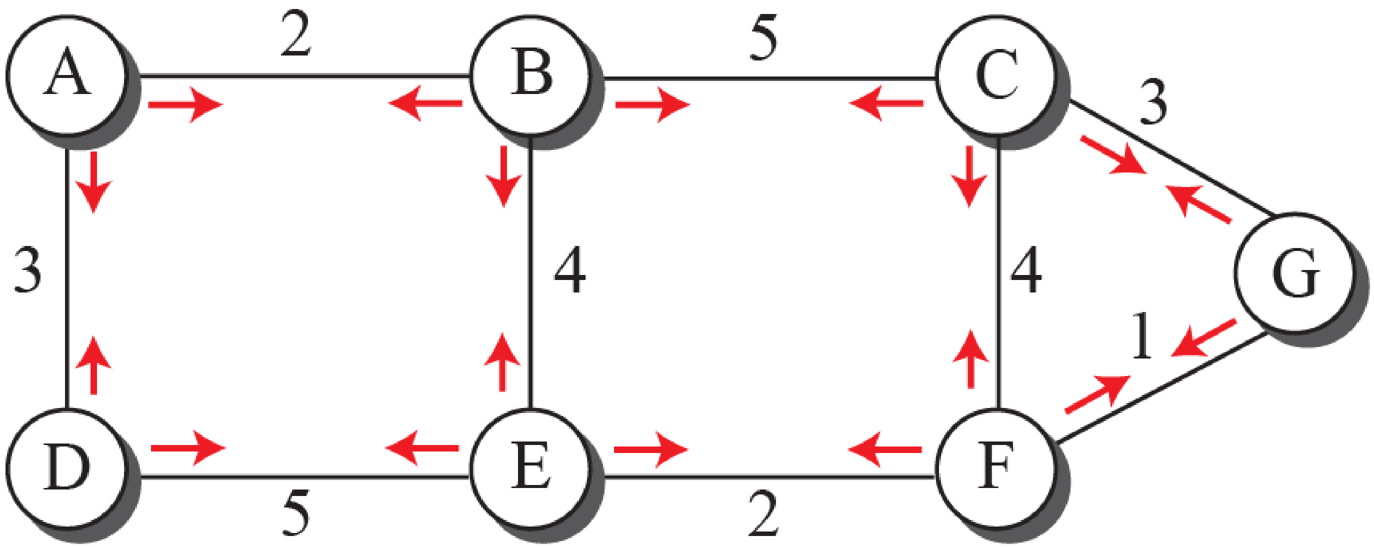
b. Link state database

Figure 4.64: ***LSPs created and sent out by each node to build LSDB***

Node	Cost
B	2
D	3

Node	Cost
A	2
C	5
E	4

Node	Cost
B	5
F	4
G	3



Node	Cost
C	3
F	1

Node	Cost
A	3
E	5

Node	Cost
B	4
D	5
E	2

Node	Cost
C	4
E	2
G	1

Figure 4.66: Spanning trees in path-vector routing

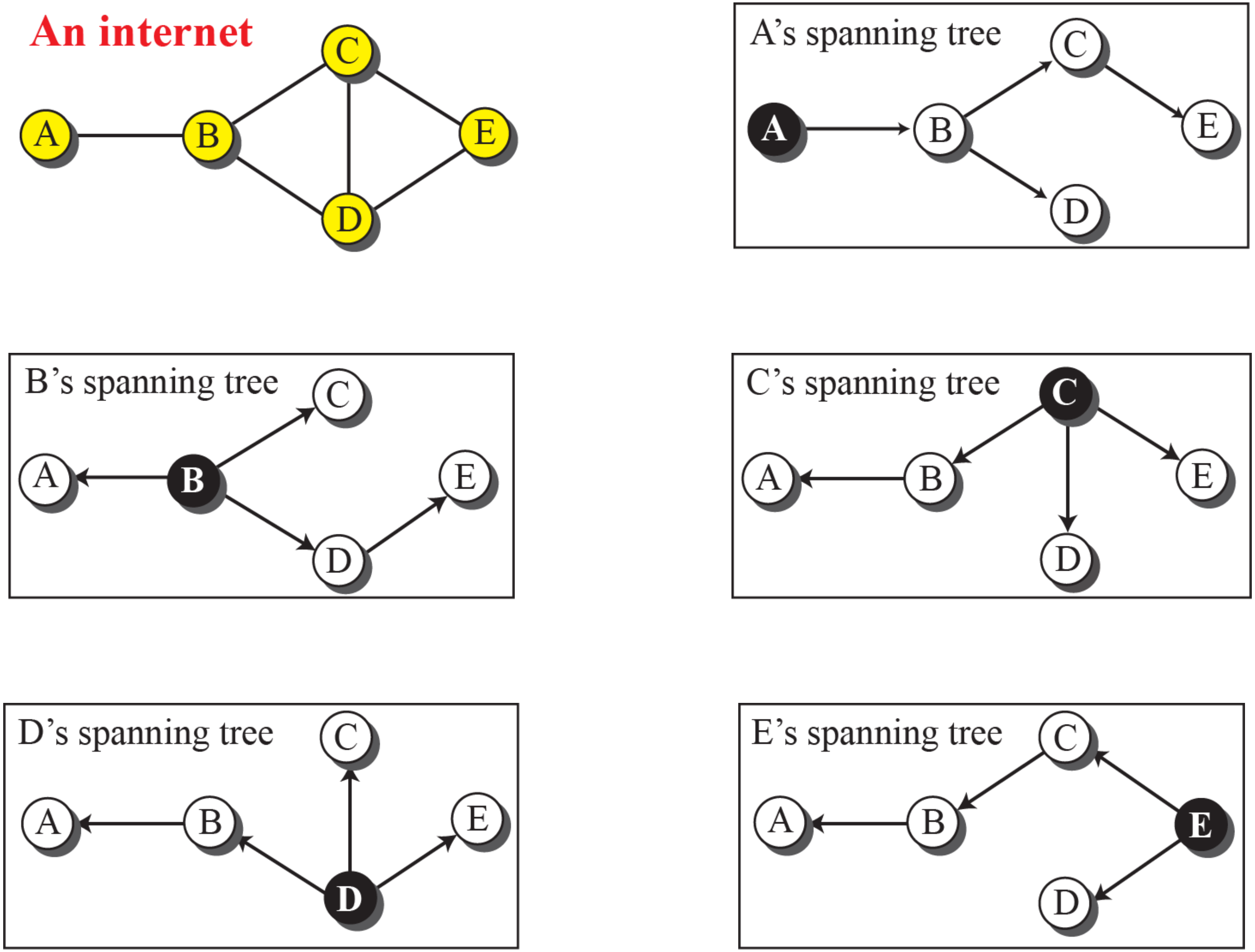


Figure 4.69: ***Internet structure***

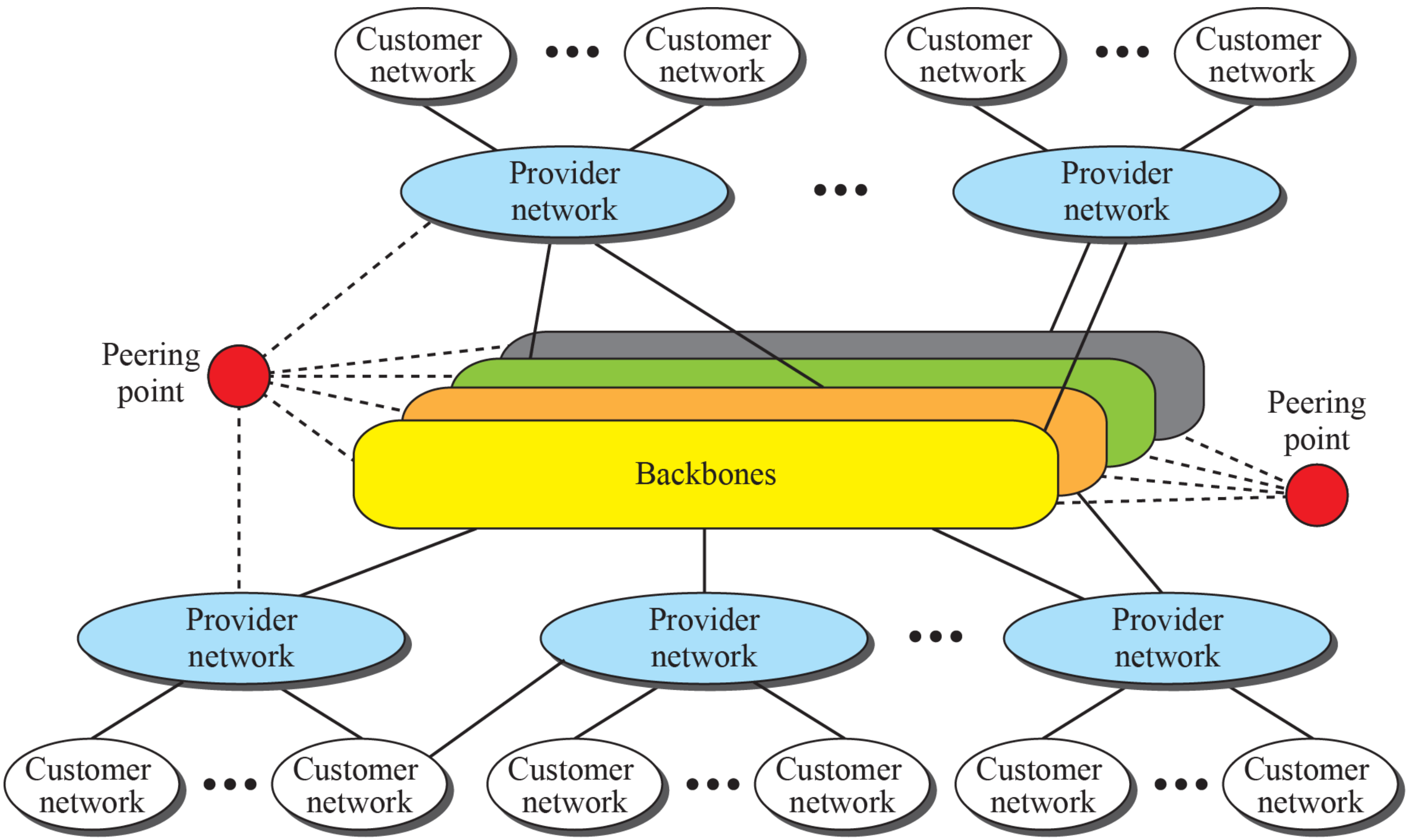


Figure 4.70: Hop counts in RIP

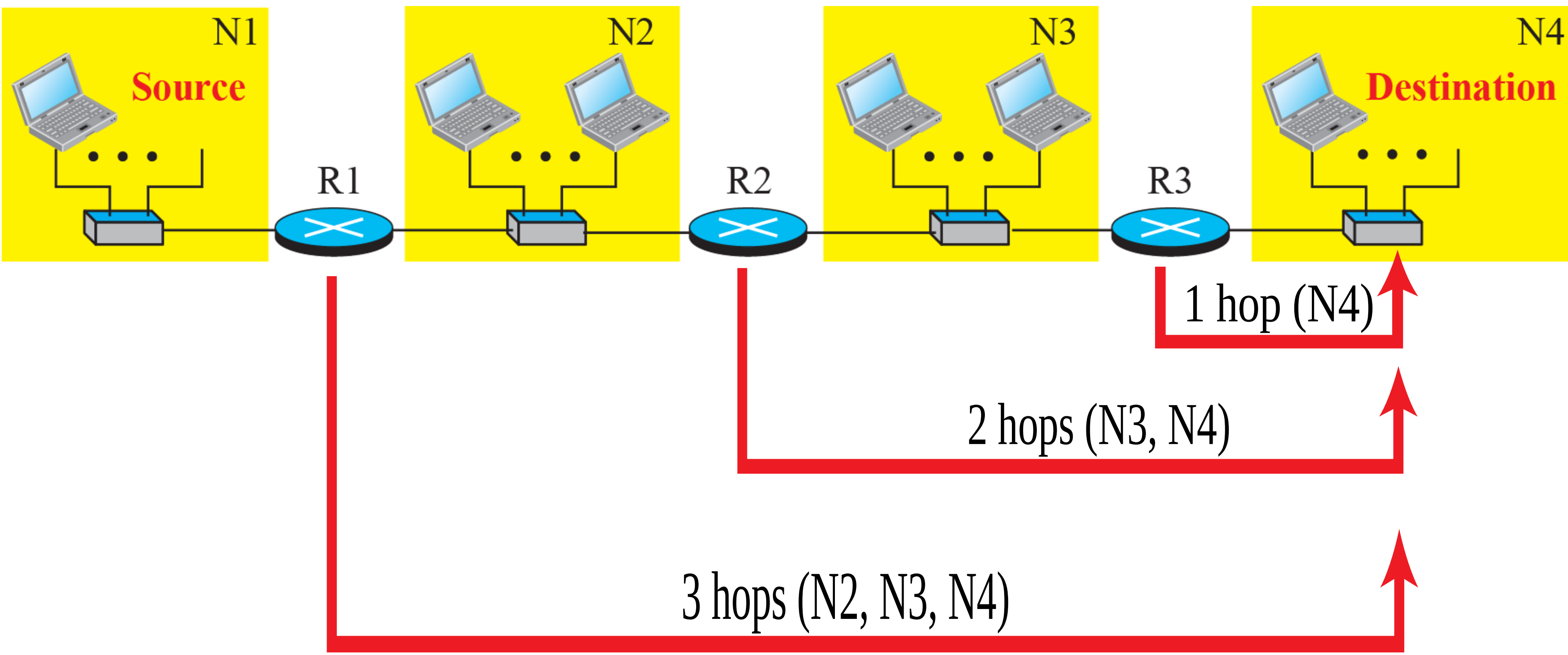


Figure 4.71: Forwarding tables

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

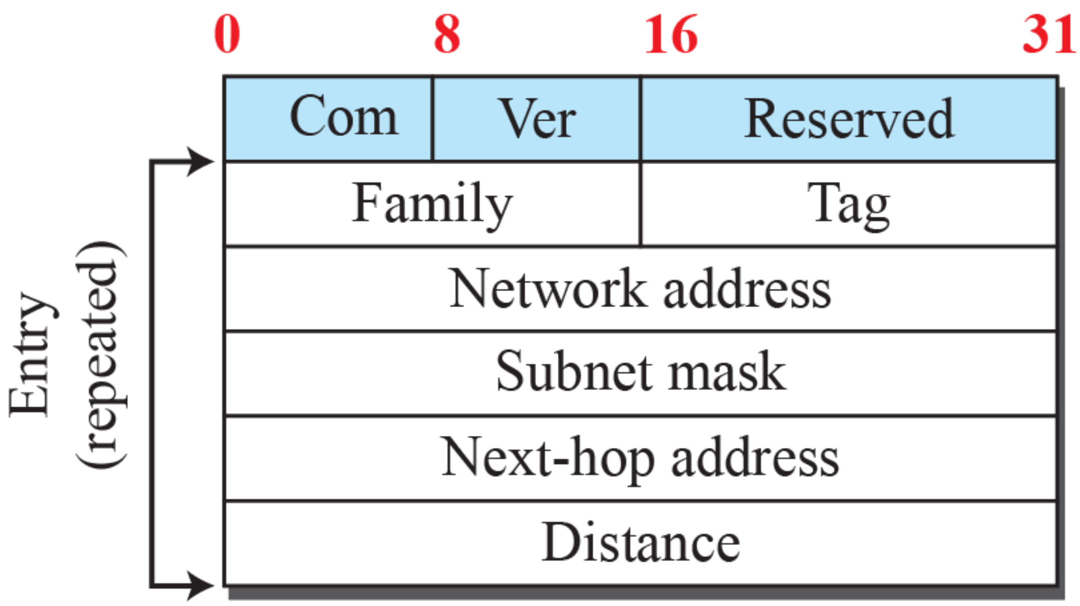
Forwarding table for R2

Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	——	1
N4	——	1

Figure 4.72: *RIP message format*



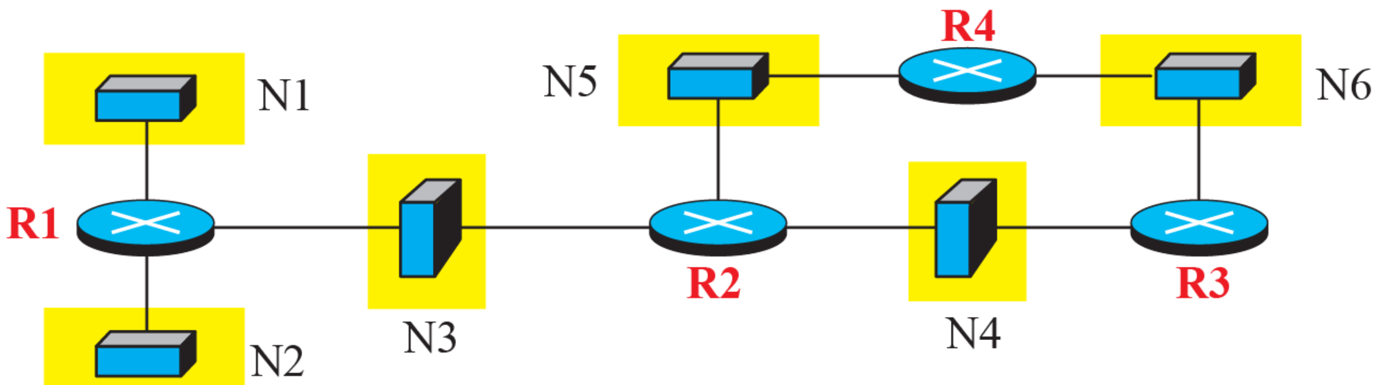
Fields

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

Example 4.15

Figure 4.73 shows a more realistic example of the operation of RIP in an autonomous system. First, the figure shows all forwarding tables after all routers have been booted. Then we show changes in some tables when some update messages have been exchanged. Finally, we show the stabilized forwarding tables when there is no more change.

Figure 4.73: Example of an autonomous system using RIP (Part I)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops

R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1

R2

Des.	N. R.	Cost
N3	———	1
N4	———	1
N5	———	1

R3

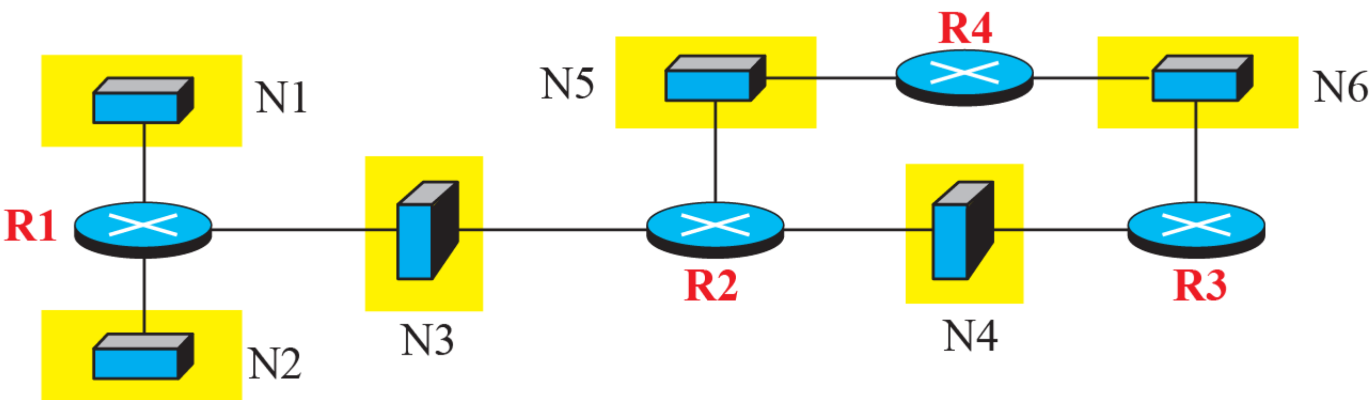
Des.	N. R.	Cost
N4	———	1
N6	———	1

R4

Des.	N. R.	Cost
N5	———	1
N6	———	1

Forwarding tables after all routers booted

Figure 4.73: Example of an autonomous system using RIP (Part II)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops
← : New route (red)
← : Old route (black)

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

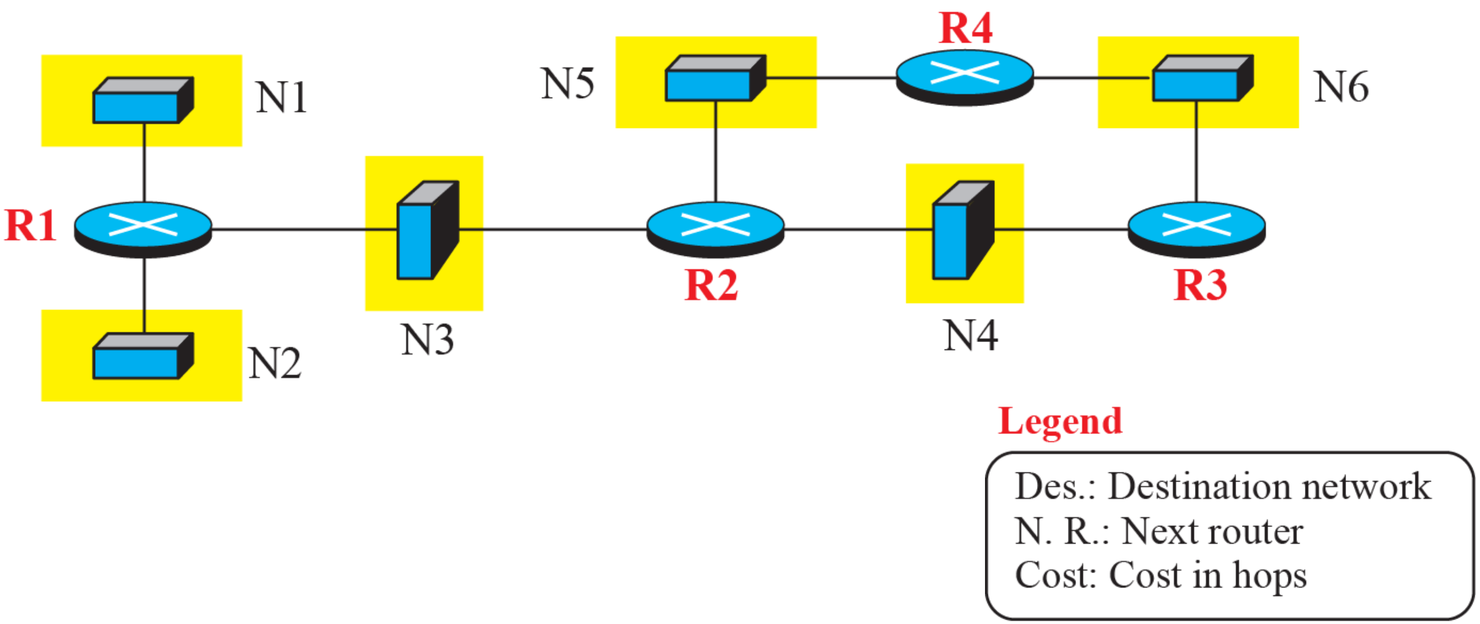
Des.	N. R.	Cost
N5	——	1
N6	——	1

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

Figure 4.73: Example of an autonomous system using RIP (Part III)



Forwarding tables for all routers after they have been stablized

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

Figure 4.74: Metric in OSPF

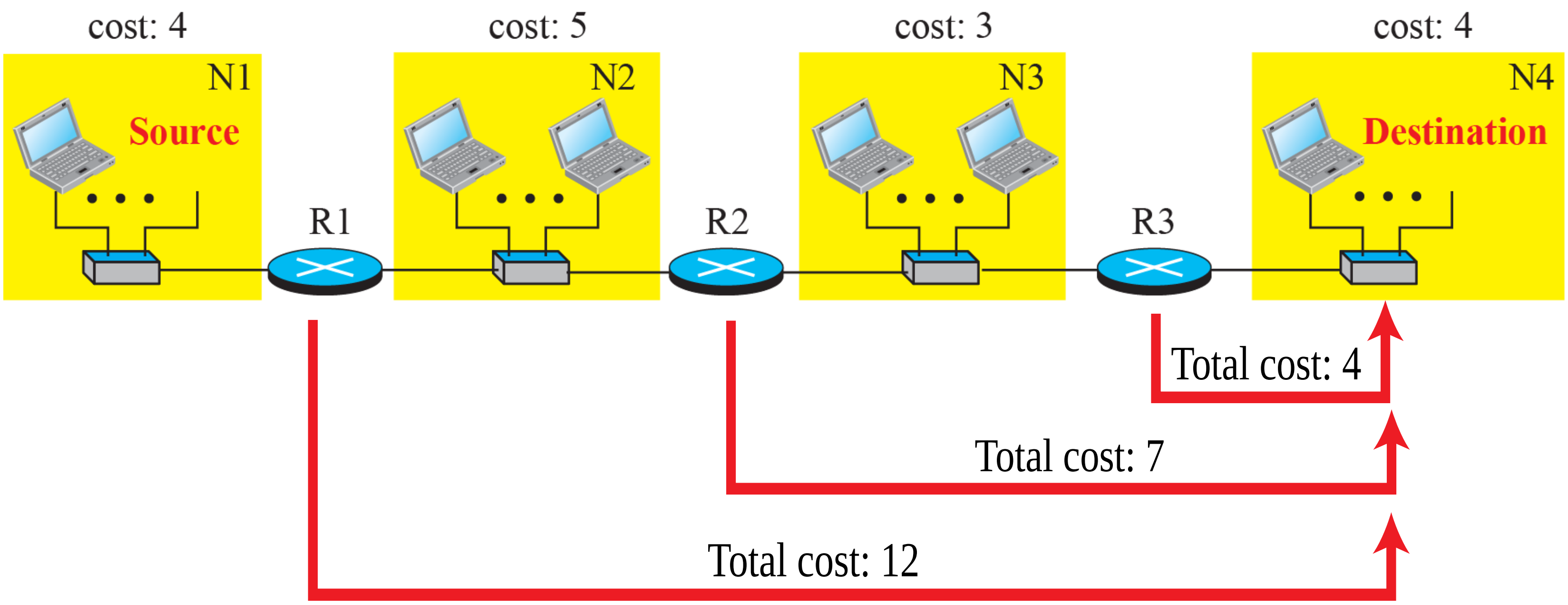
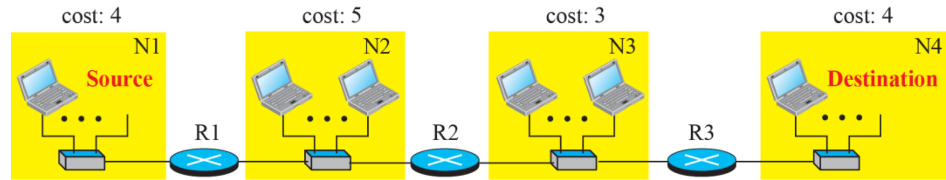


Figure 4.75: Forwarding tables in OSPF

Forwarding table for R1

Destination network	Next router	Cost
N1	——	
N2	——	
N3	R2	
N4	R2	12



The internet from previous figure

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	——	5
N3	——	3
N4	R3	7

Forwarding table for R3

Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	——	3
N4	——	4

Figure 4.76: Areas in an autonomous system

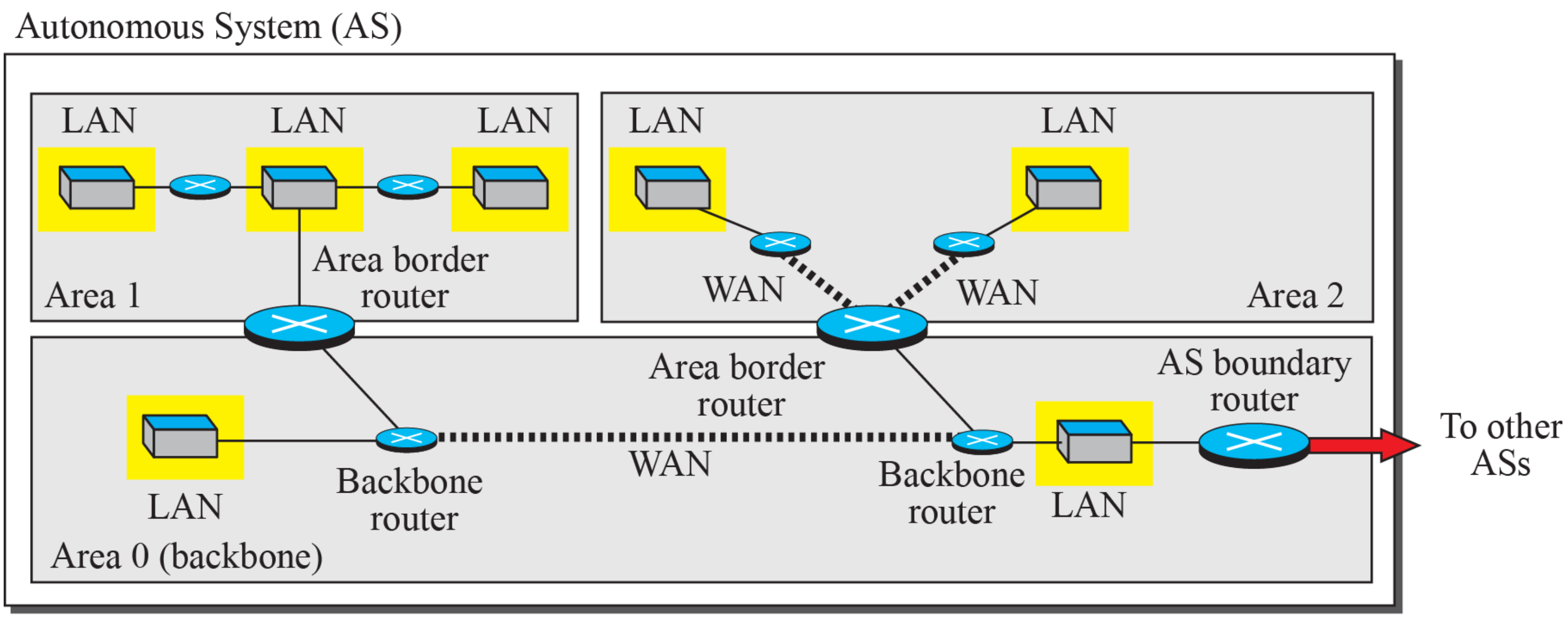
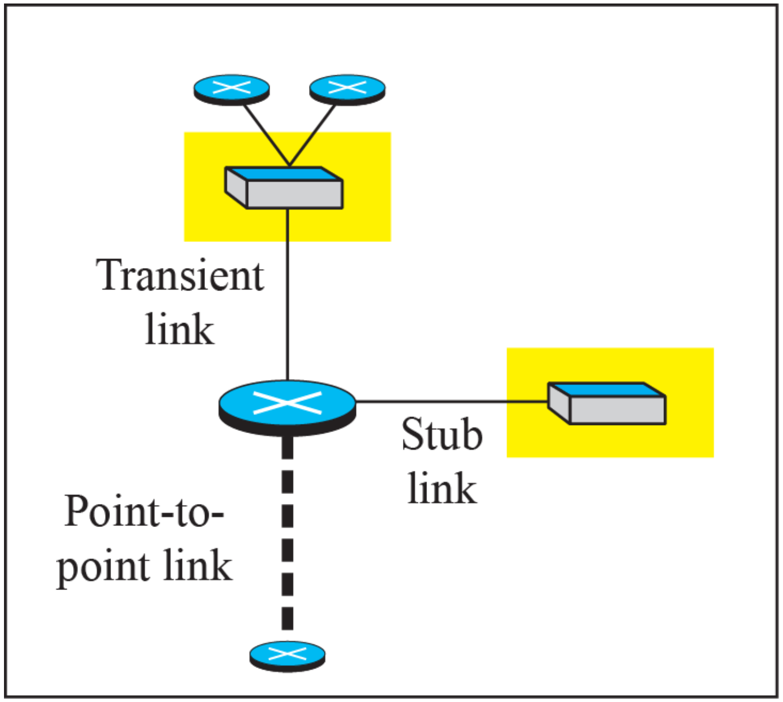
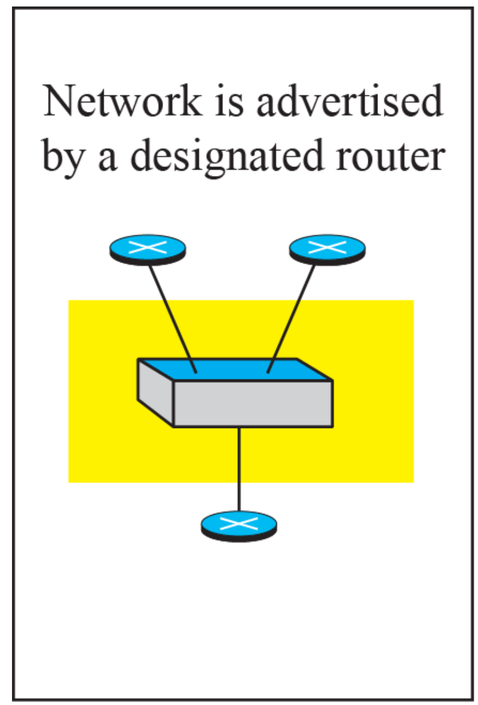


Figure 4.77: Five different LSPs (Part I)

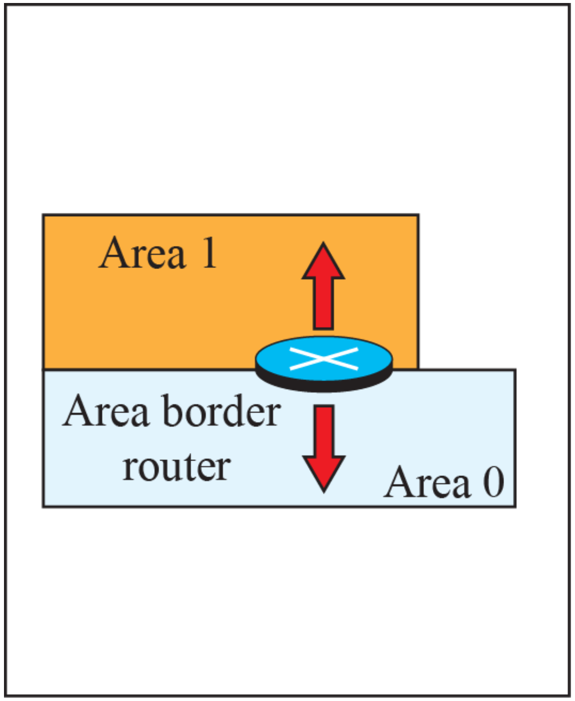


a. Router link

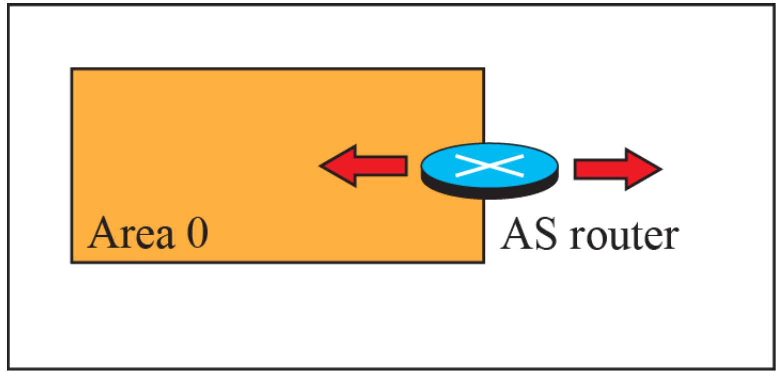


b. Network link

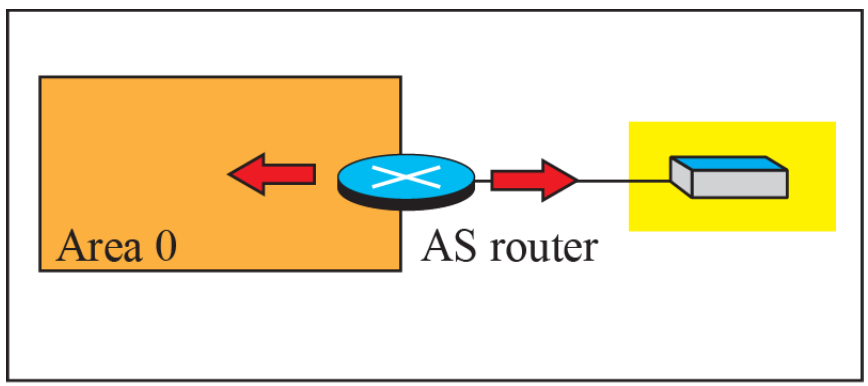
Figure 4.77: Five different LSPs (Part II)



c. Summary link to network

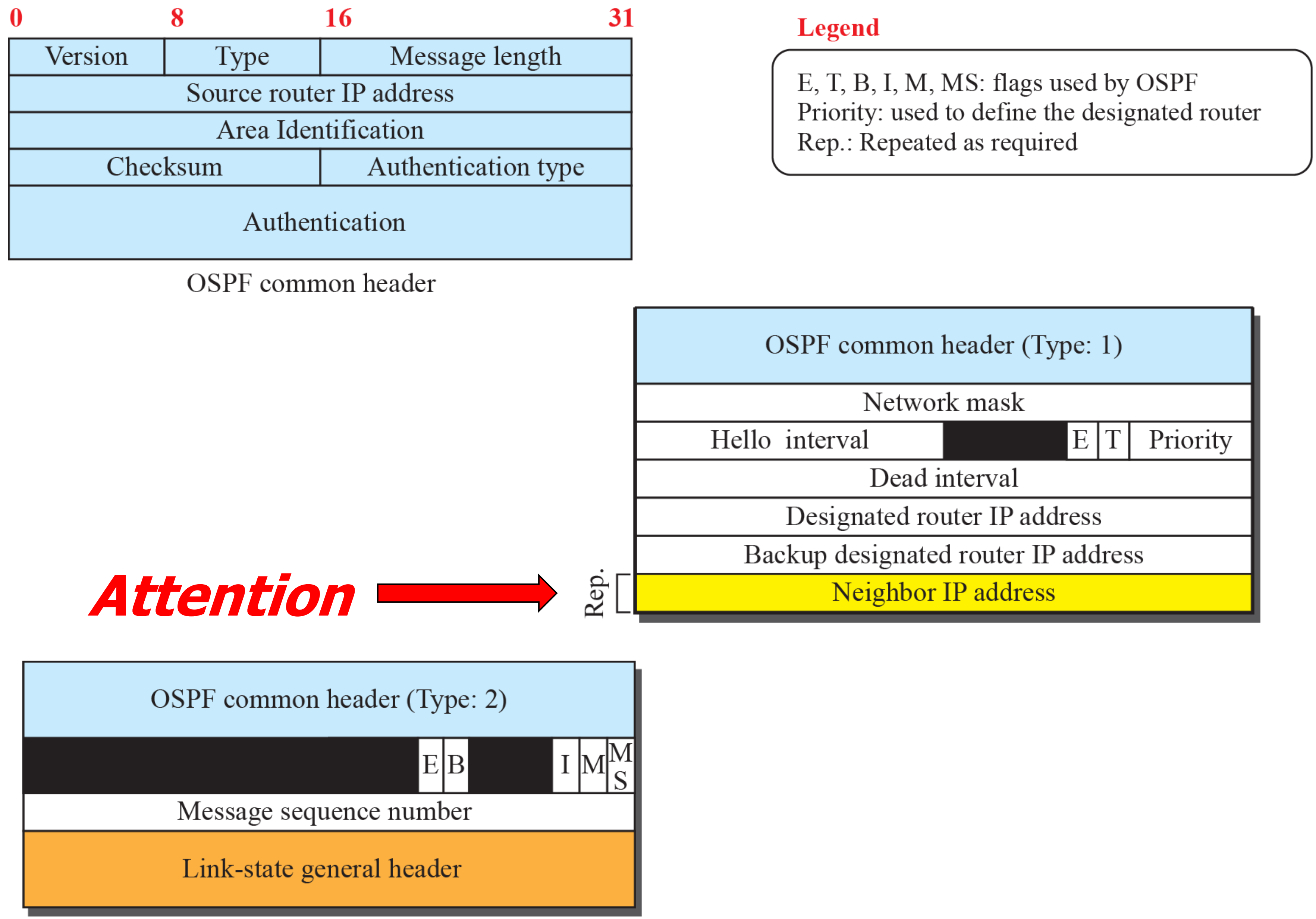


d. Summary link to AS



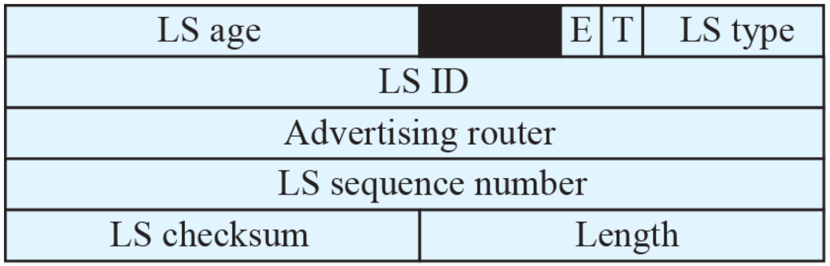
e. External link

Figure 4.78: ***OSPF message formats (Part I)***



Database description

Figure 4.78: OSPF message formats (Part II)

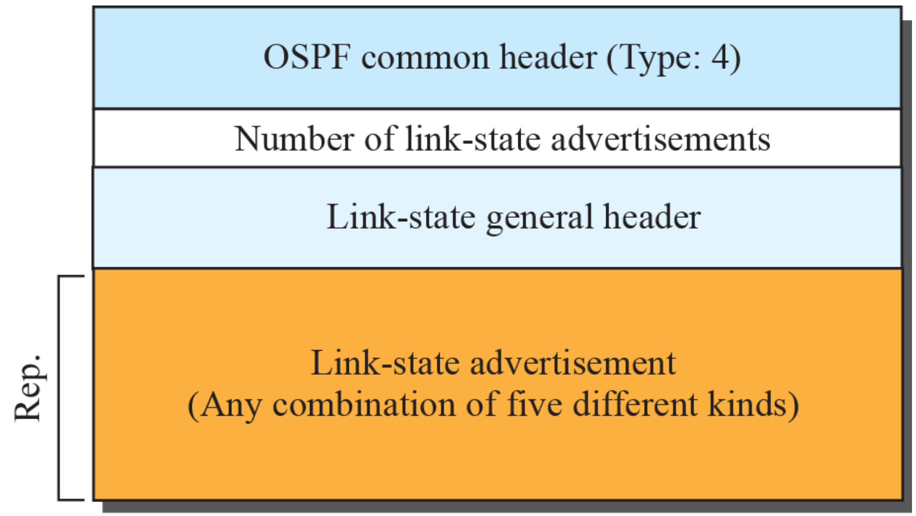


Link-state general header

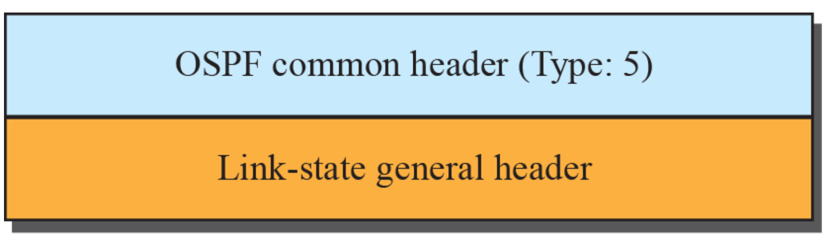
Legend

E, T, B, I, M, MS: flags used by OSPF
Priority: used to define the designated router
Rep.: Repeated as required

Attention →



Link-state update



Link-state acknowledgment

Figure 4.79: A sample internet with four ASs

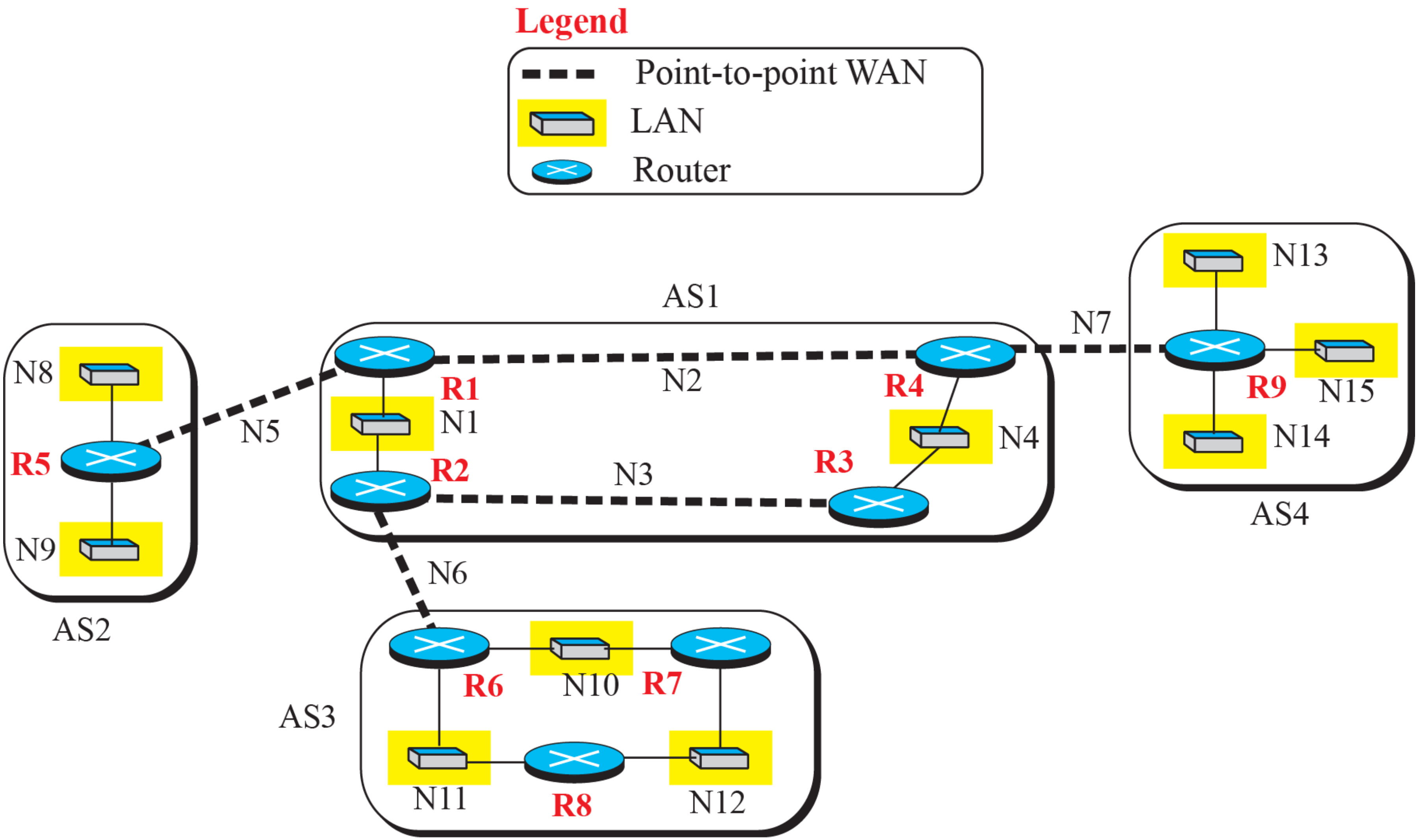
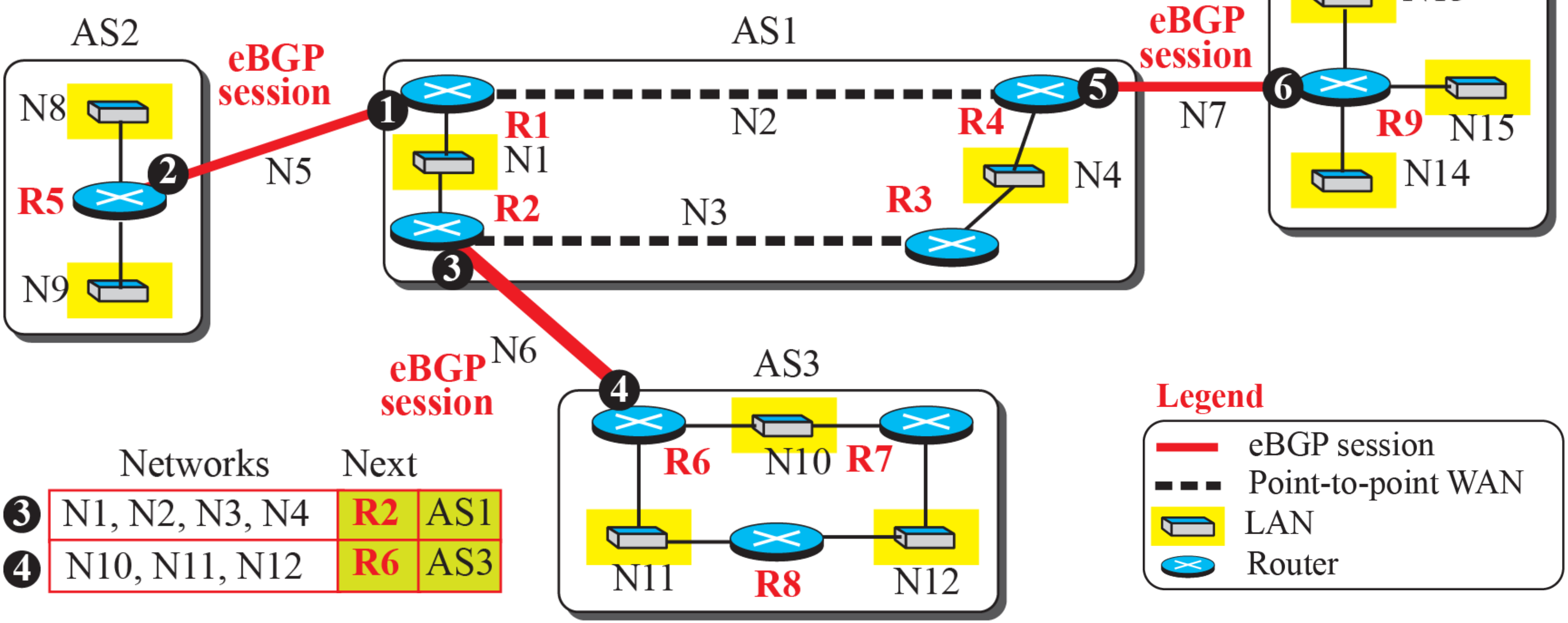


Figure 4.80: ***eBGP operation***

	Networks	Next AS	
❶	N1, N2, N3, N4	R1	AS1
❷	N8, N9	R5	AS2

	Networks	Next AS	
❺	N1, N2, N3, N4	R4	AS1
❻	N13, N14, N15	R9	AS4



	Networks	Next	
❸	N1, N2, N3, N4	R2	AS1
❹	N10, N11, N12	R6	AS3

Figure 4.81: Combination of eBGP and iBGP sessions in our internet

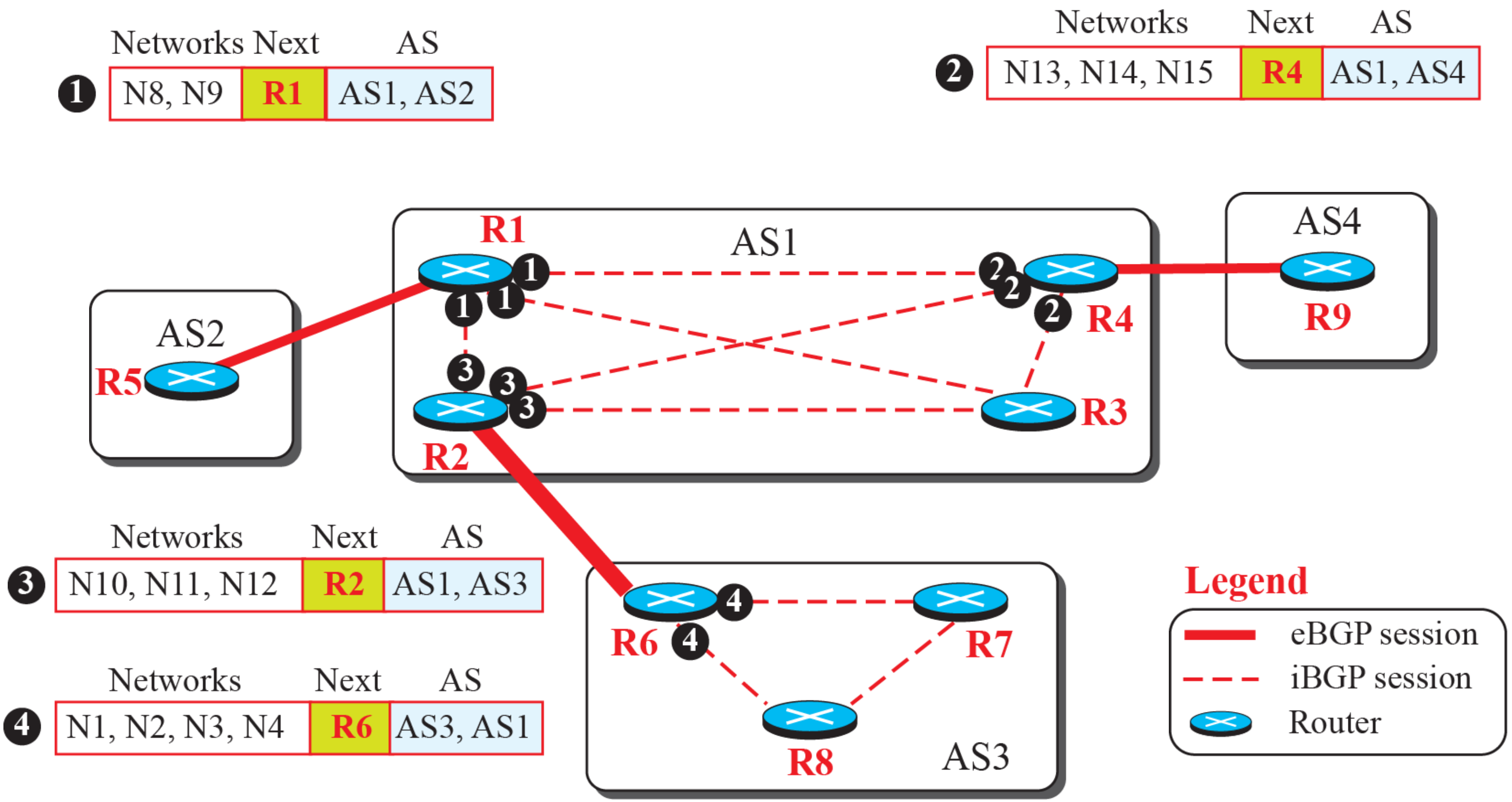
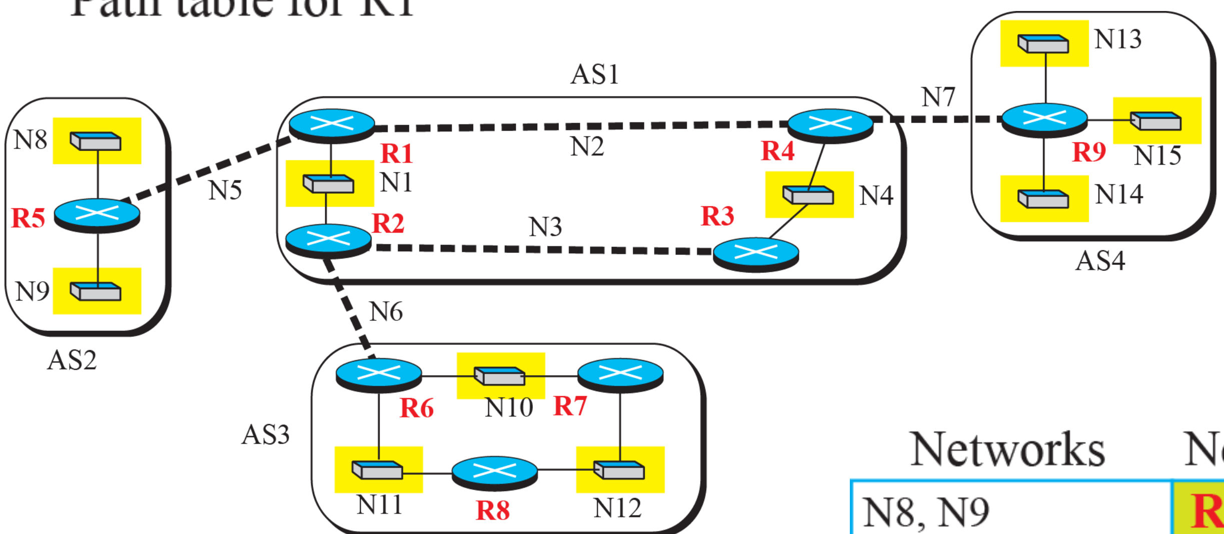


Figure 4.82: *Finalized BGP path tables (Part I)*

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	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	R6	AS1, AS3
N13, N14, N15	R1	AS1, AS4

Path table for R2

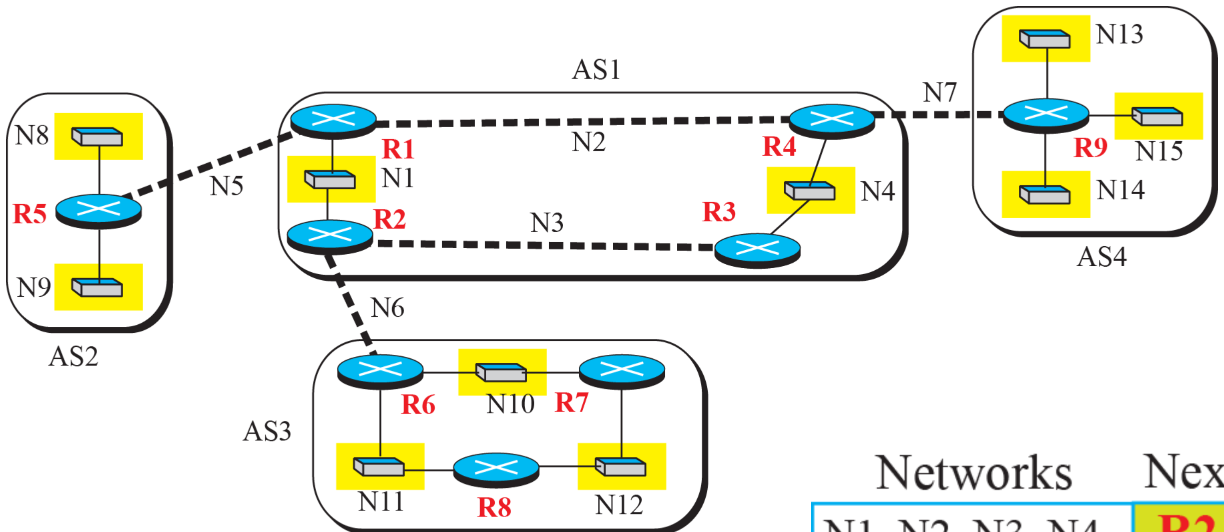
Figure 4.82: *Finalized BGP path tables (Part II)*

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

Path table for R4



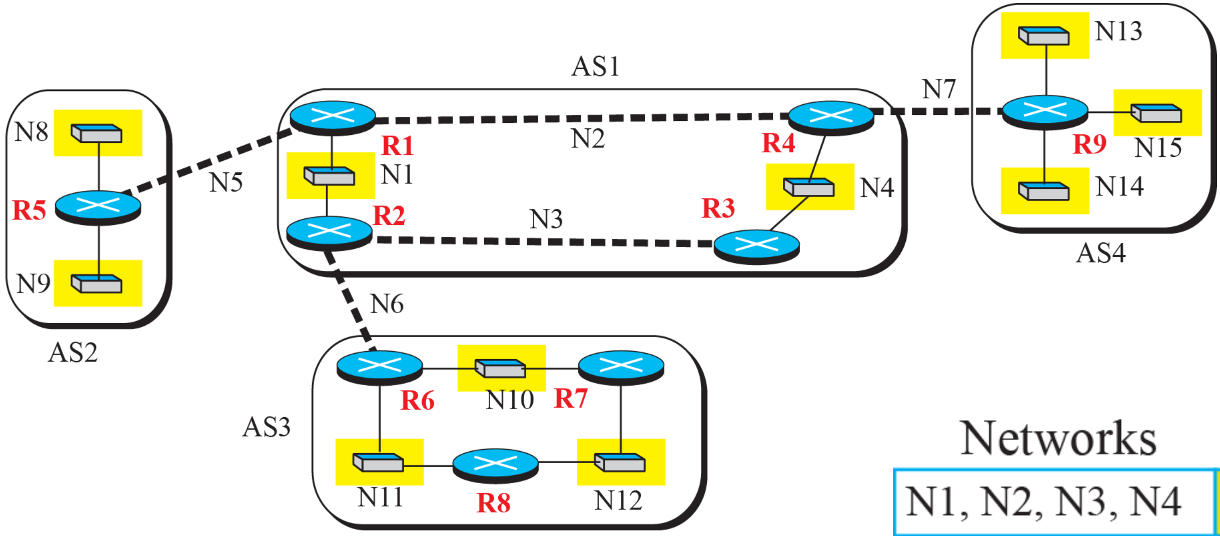
Networks	Next	Path
N1, N2, N3, N4	R2	AS3, AS1
N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R2	AS3, AS1, AS4

Path table for R6

Figure 4.82: Finalized BGP path tables (Part III)

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



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	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R7

Figure 4.83: *Forwarding tables after injection from BGP (Part I)*

Des.	Next	Cost
N1	—	1
N4	R4	2
N8	R5	1
N9	R5	1
N10	R2	2
N11	R2	2
N12	R2	2
N13	R4	2
N14	R4	2
N15	R4	2

Table for R1

Des.	Next	Cost
N1	—	1
N4	R3	2
N8	R1	2
N9	R1	2
N10	R6	1
N11	R6	1
N12	R6	1
N13	R3	3
N14	R3	3
N15	R3	3

Table for R2

Des.	Next	Cost
N1	R2	2
N4	—	1
N8	R2	3
N9	R2	3
N10	R2	2
N11	R2	2
N12	R2	2
N13	R4	2
N14	R4	2
N15	R4	2

Table for R3

Des.	Next	Cost
N1	R1	2
N4	—	1
N8	R1	2
N9	R1	2
N10	R3	3
N11	R3	3
N12	R3	3
N13	R9	1
N14	R9	1
N15	R9	1

Table for R4

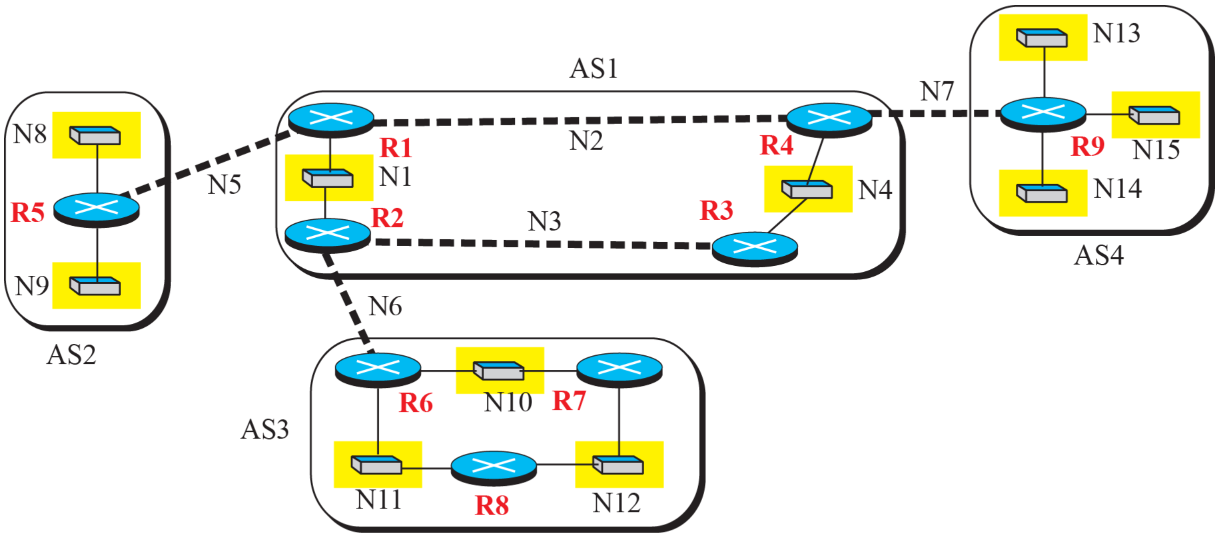


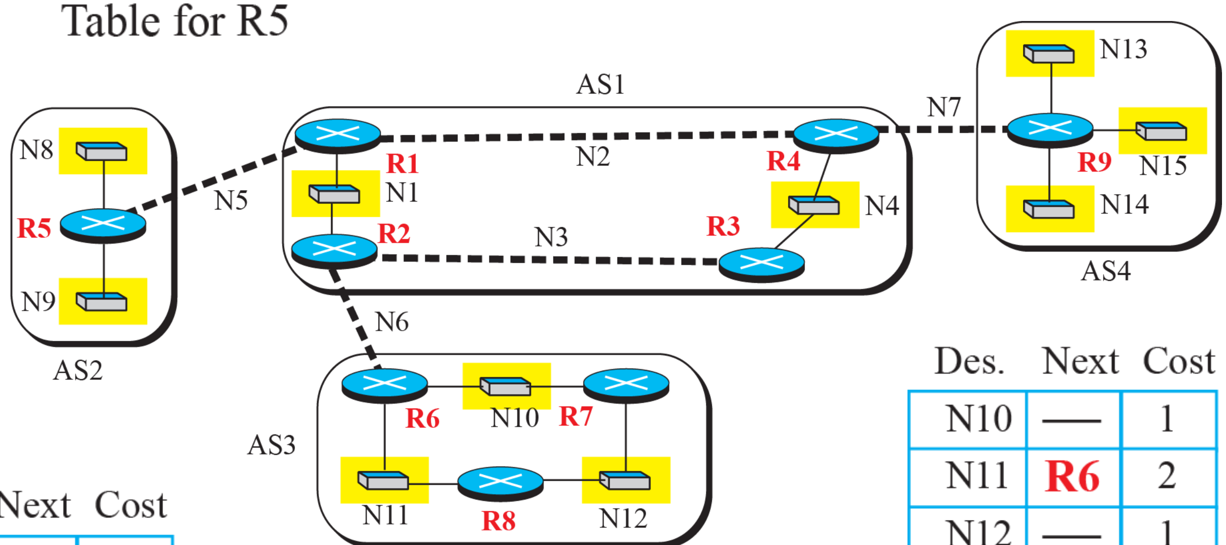
Figure 4.83: *Forwarding tables after injection from BGP (Part II)*

Des.	Next	Cost
N8	—	1
N9	—	1
0	R1	1

Table for R5

Des.	Next	Cost
N13	—	1
N14	—	1
N15	—	1
0	R4	1

Table for R9



Des.	Next	Cost
N10	—	1
N11	R6	2
N12	—	1
0	R6	2

Table for R7

Des.	Next	Cost
N10	—	1
N11	—	1
N12	R7	2
0	R2	1

Table for R6

Des.	Next	Cost
N10	R6	2
N11	—	1
N12	—	1
0	R6	2

Table for R8

Figure 4.84: Format of path attribute

O: Optional bit (set if attribute is optional)
P: Partial bit (set if an optional attribute in lost in transit)
T: Transitive bit (set if attribute is transitive)
E: Extended bit (set if attribute length is two bytes)

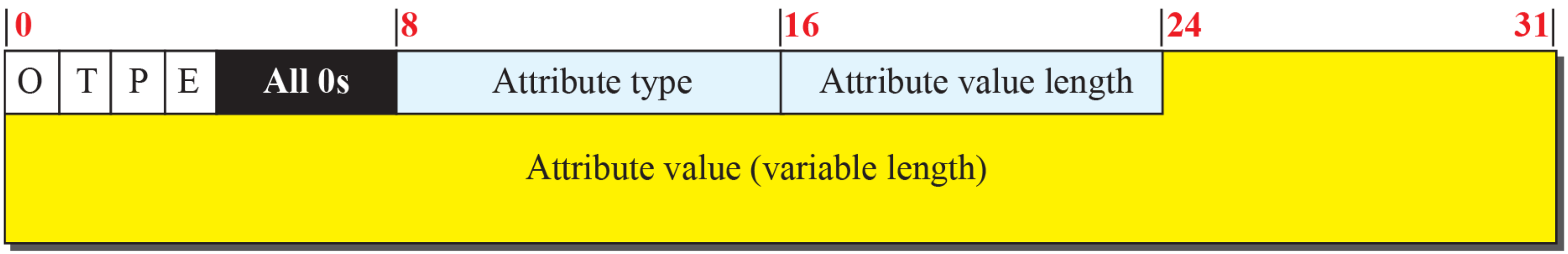


Figure 4.85: *Flow diagram for route selection*

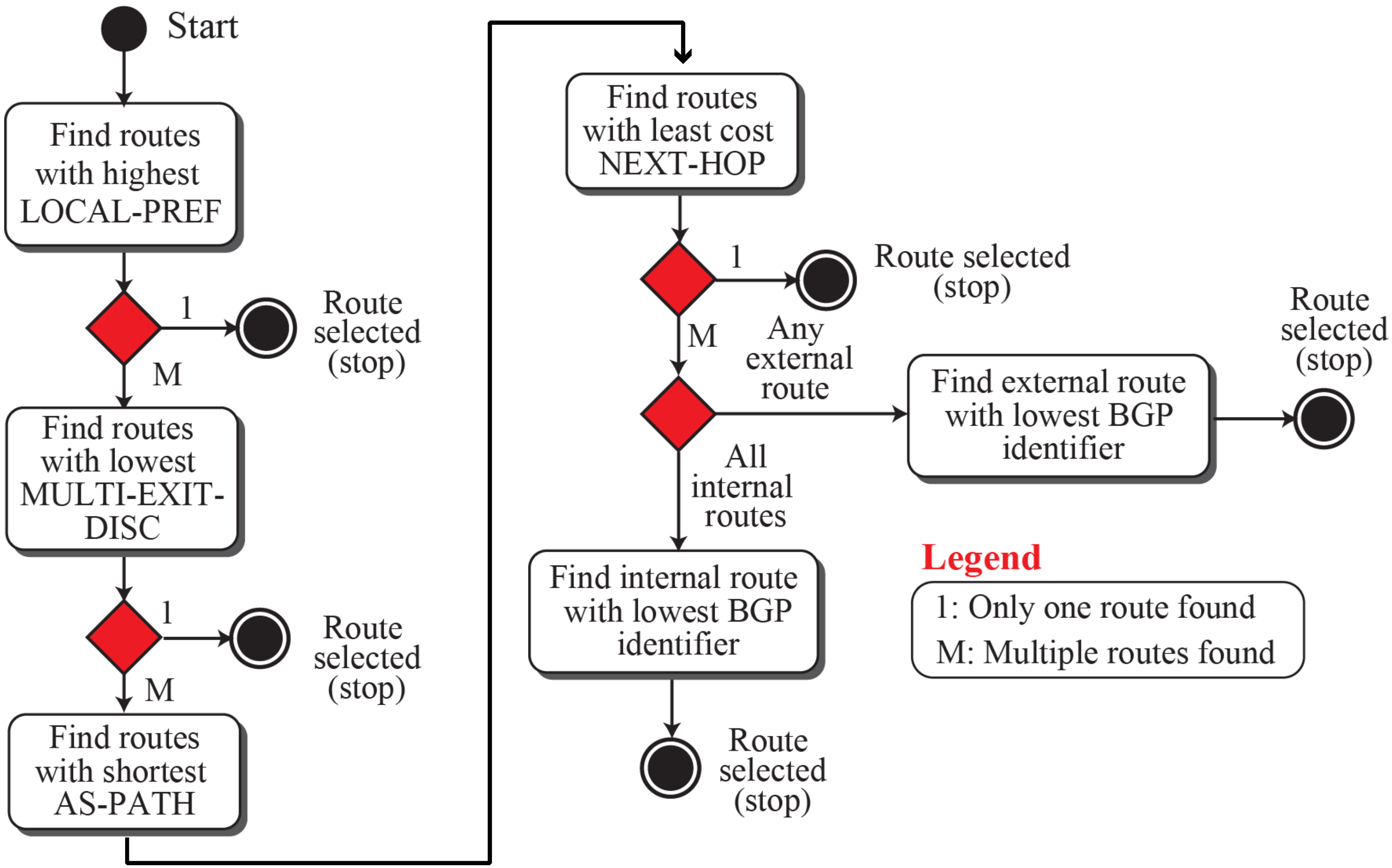


Figure 4.86: BGP messages