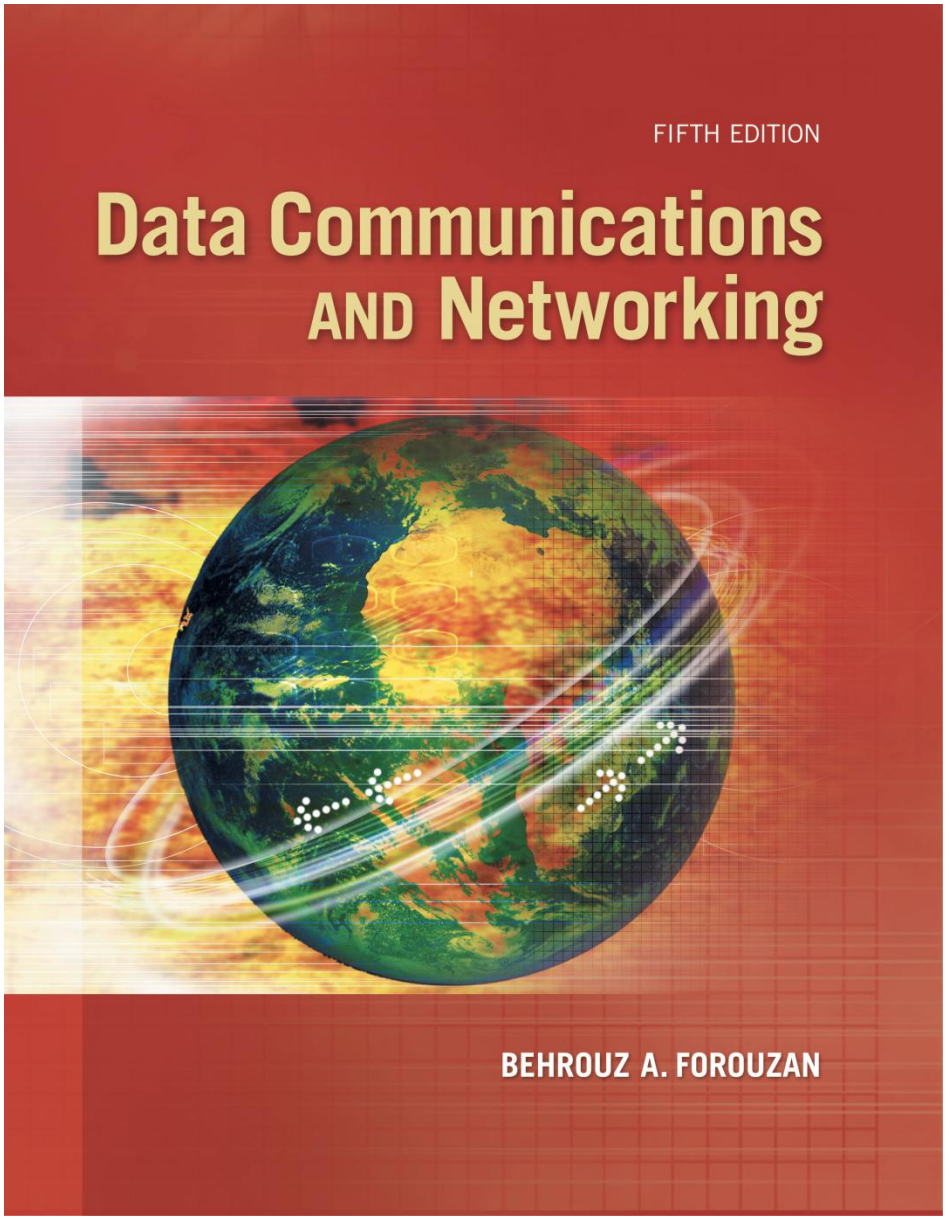
Chapter 18

Introduction to Network Layer



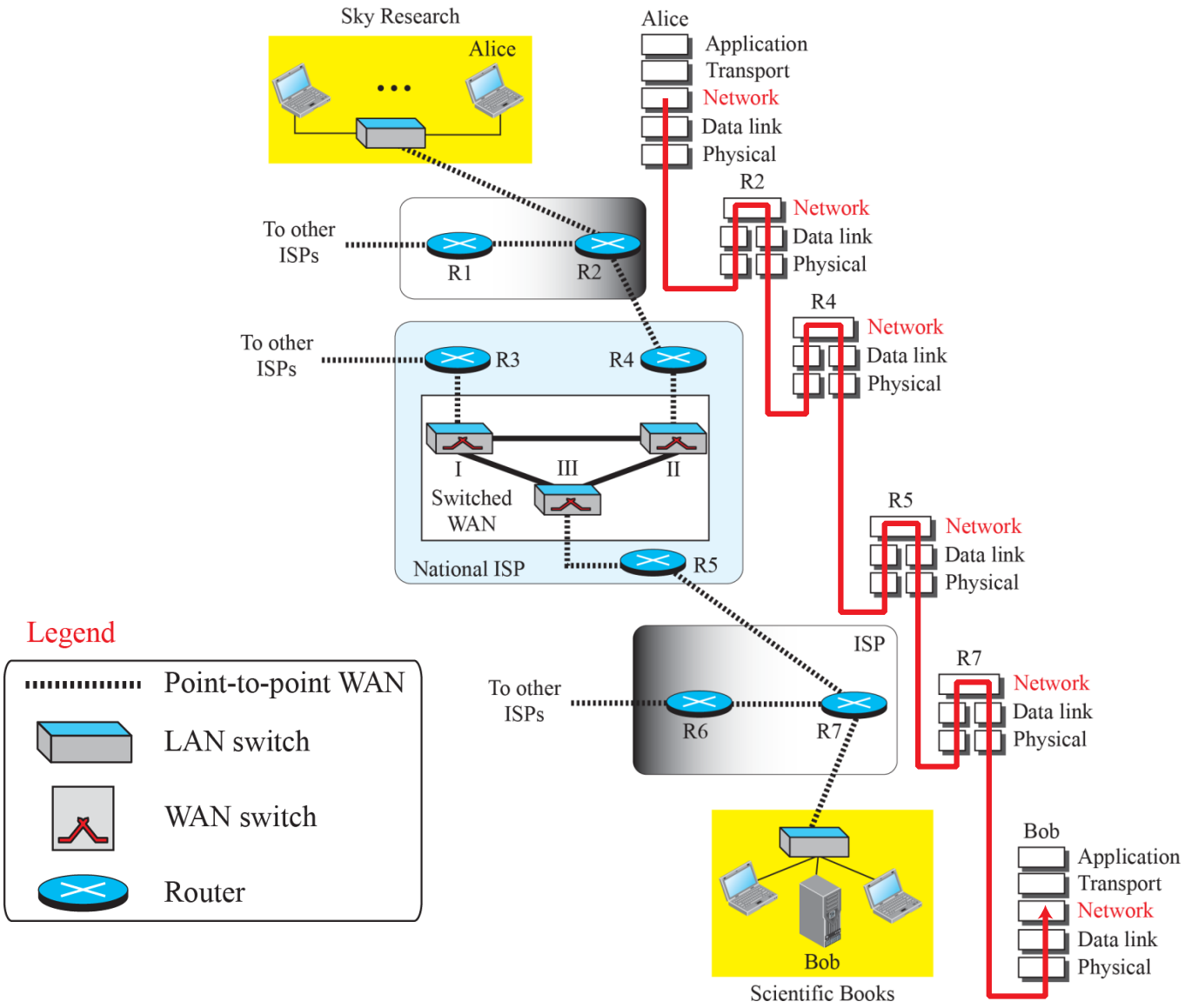
Chapter 18: Outline

- **18.1 NETWORK-LAYER SERVICES**
- **18.2 PACKET SWITCHING**
- **18.3 NETWORK-LAYER PERFORMANCE**
- **18.4 IPv4 ADDRESSES**
- **18.5 FORWARDING OF IP PACKETS**

18-1 NETWORK-LAYER SERVICES

- Before discussing the network layer in the Internet today, let's briefly discuss the network-layer services that, in general, are expected from a network-layer protocol. Figure 18.1 shows the communication between Alice and Bob at the network layer. This is the same scenario we used in Chapters 3 and 9 to show the communication at the physical and the data-link layers, respectively.

Figure 18.1: Communication at the network layer



18.2 Routing and Forwarding

Other duties of the network layer, which are as important as the first, are routing and forwarding, which are directly related to each other.

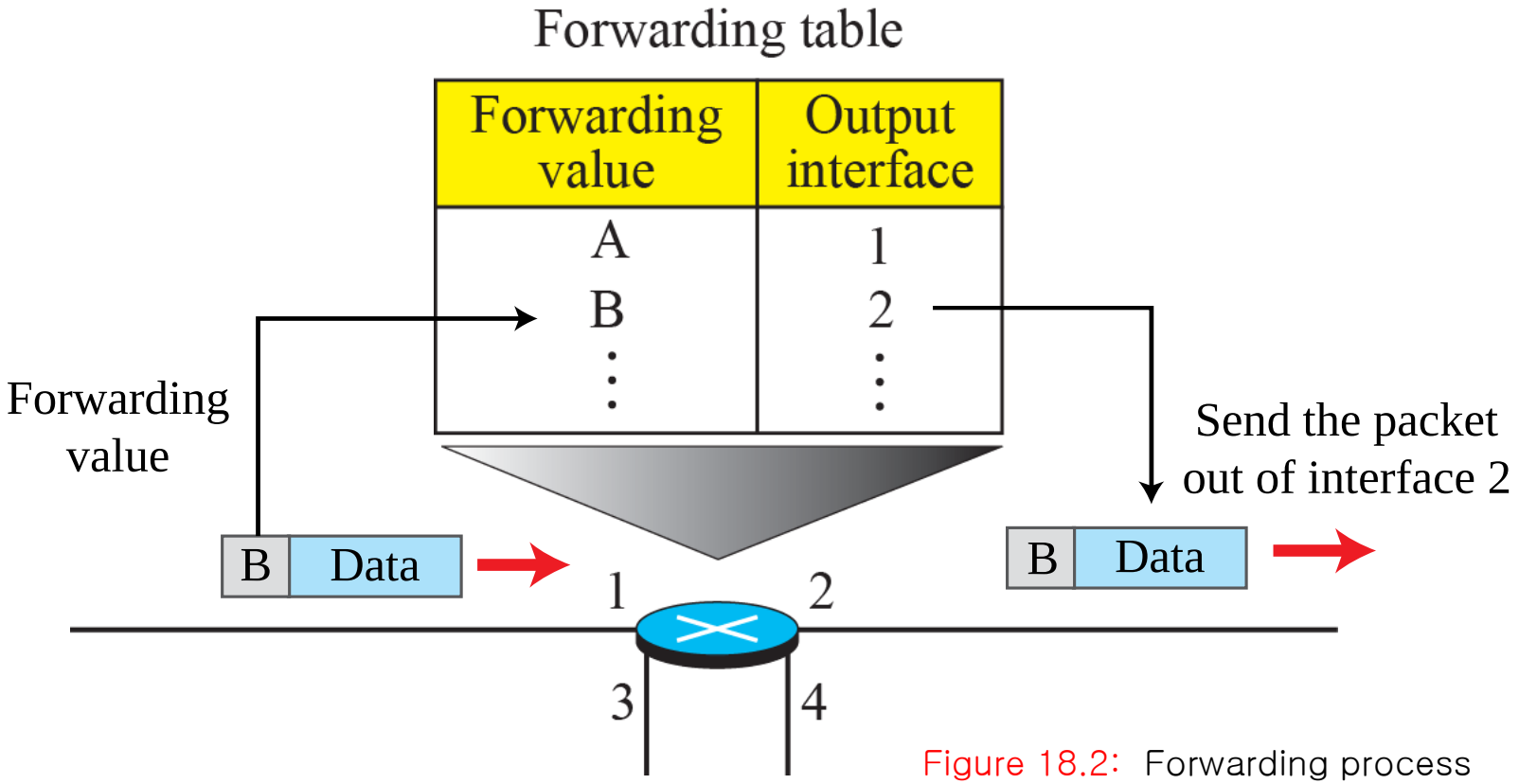


Figure 18.2: Forwarding process

18-2 PACKET SWITCHING

- From the discussion of routing and forwarding in the previous section, we infer that a kind of switching occurs at the network layer. A router, in fact, 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.

18.2.1 Datagram Approach

When the Internet started, to make it simple, the network layer was designed to provide a connectionless service in which the network-layer protocol treats each packet independently, with each packet having no relationship to any other packet. The idea was that the network layer is only responsible for delivery of packets from the source to the destination. In this approach, the packets in a message may or may not travel the same path to their destination. Figure 18.3 shows the idea..

Figure 18.3: A connectionless packet-switched network

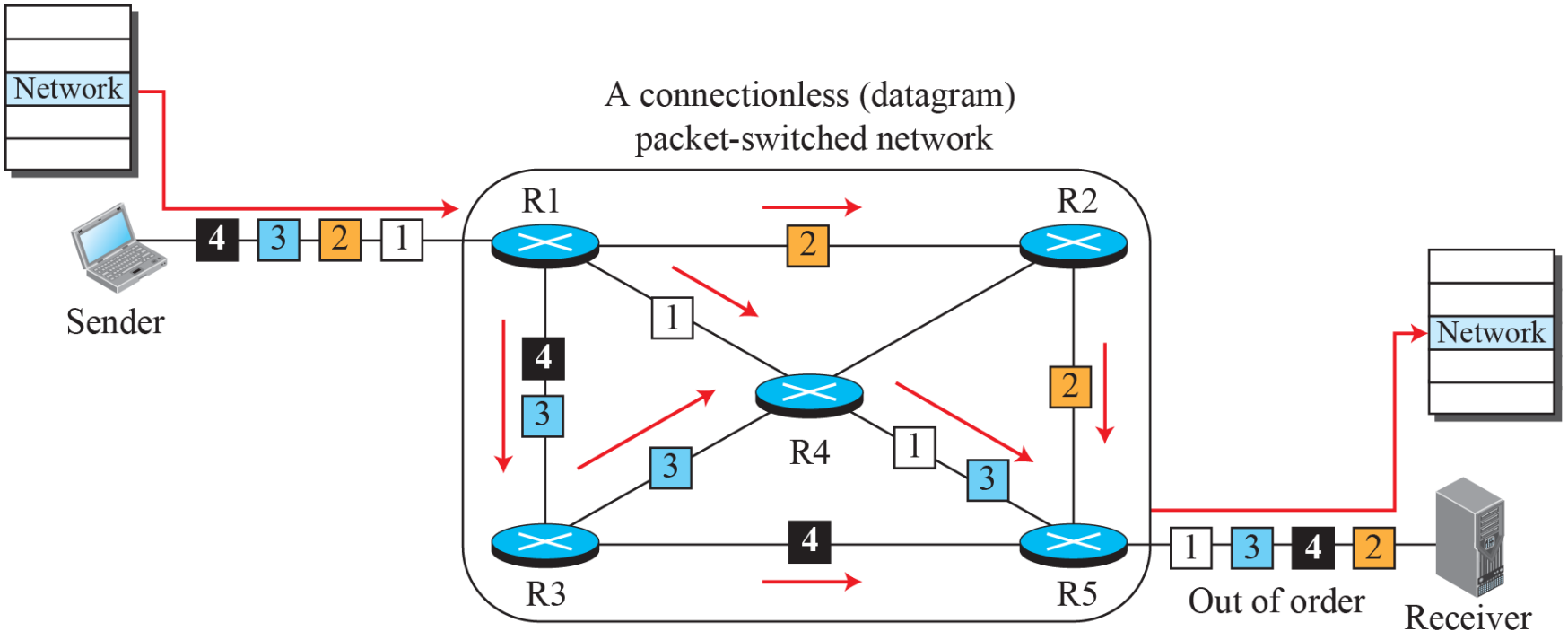
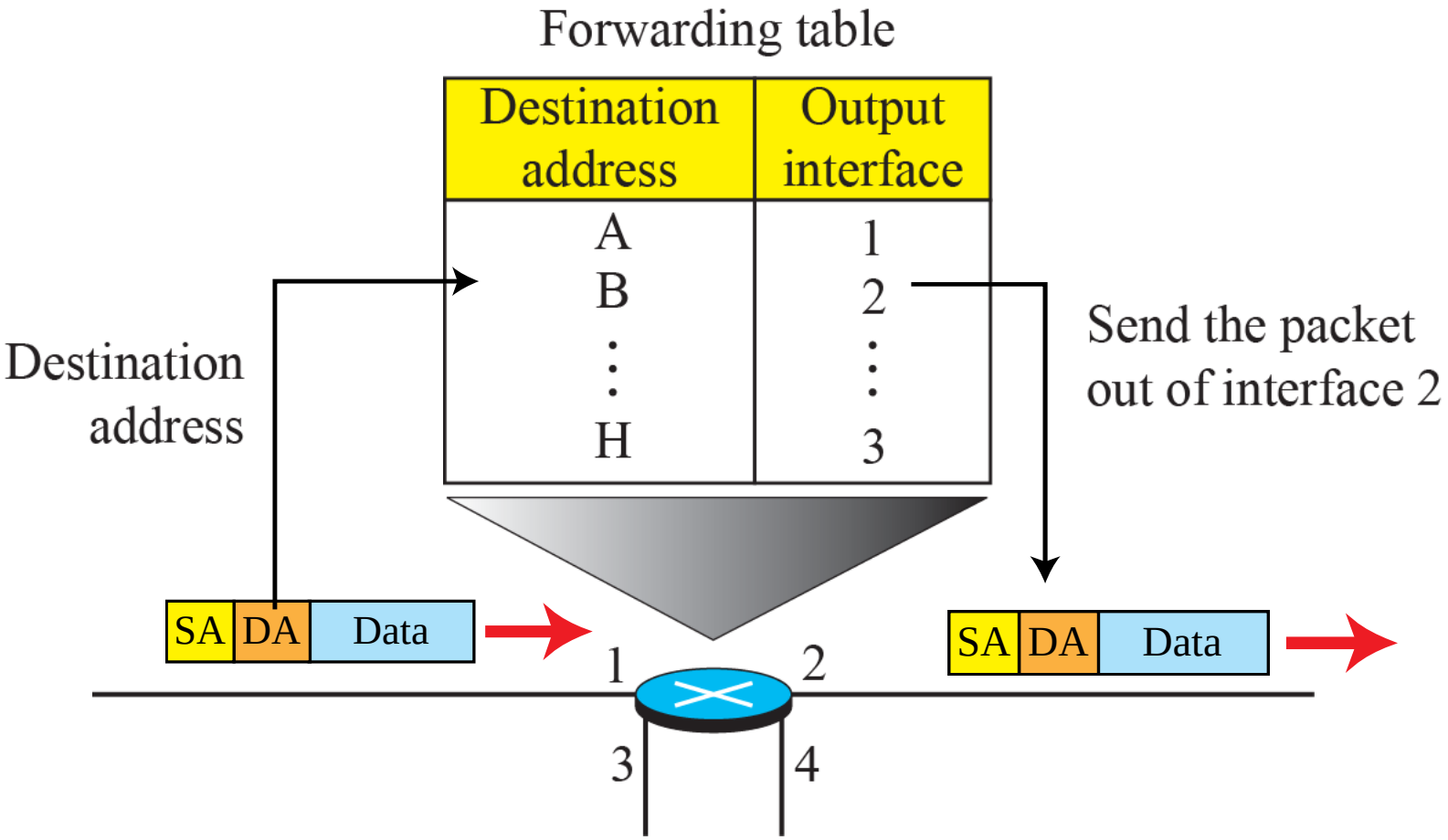


Figure 18.4: Forwarding process in a router when used in a connectionless network



18.2.2 Virtual-Circuit Approach

In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path. In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.

Figure 18.5: A virtual-circuit packet-switched network

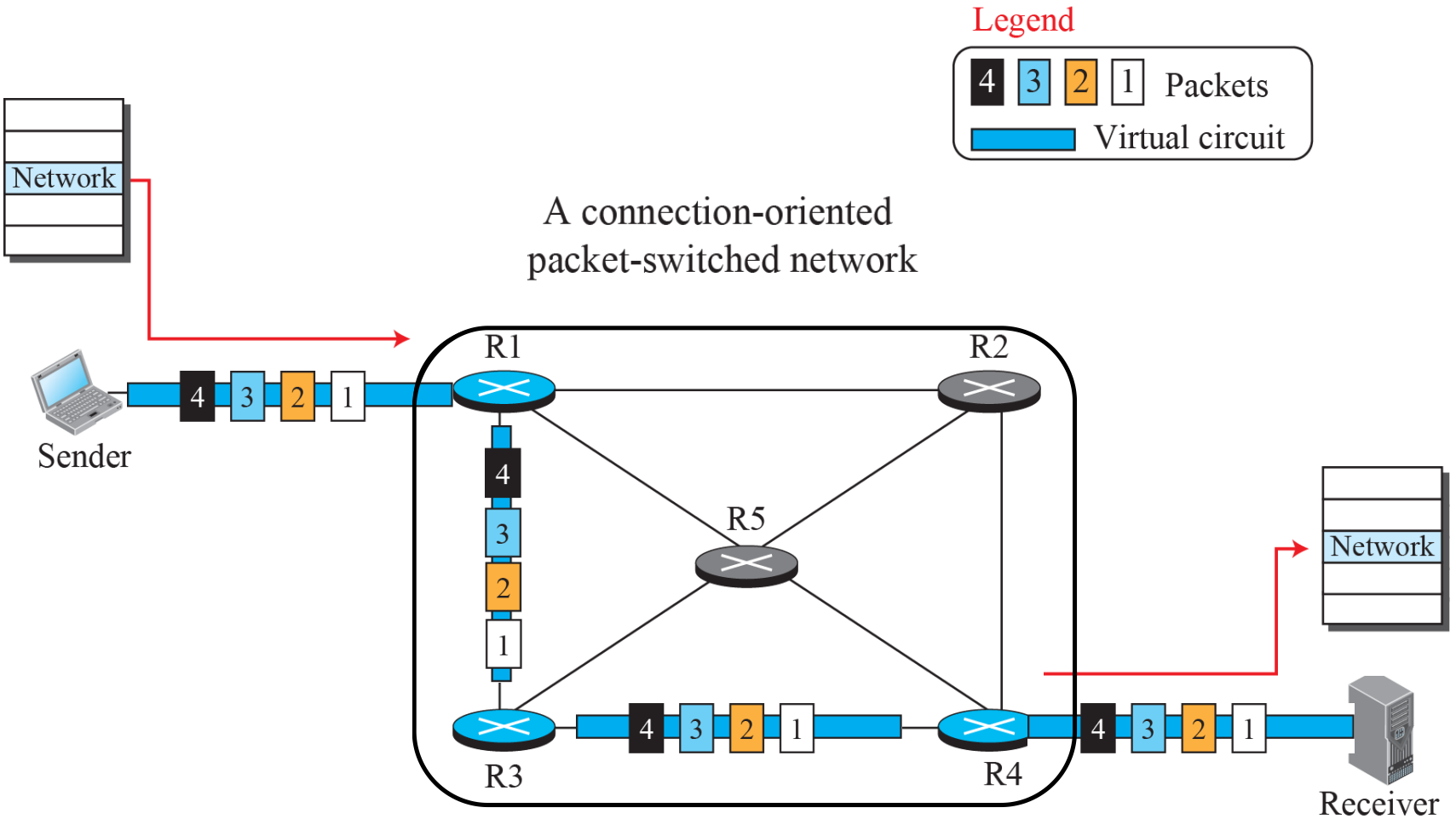


Figure 18.6: Forwarding process in a router when used in a virtual circuit network

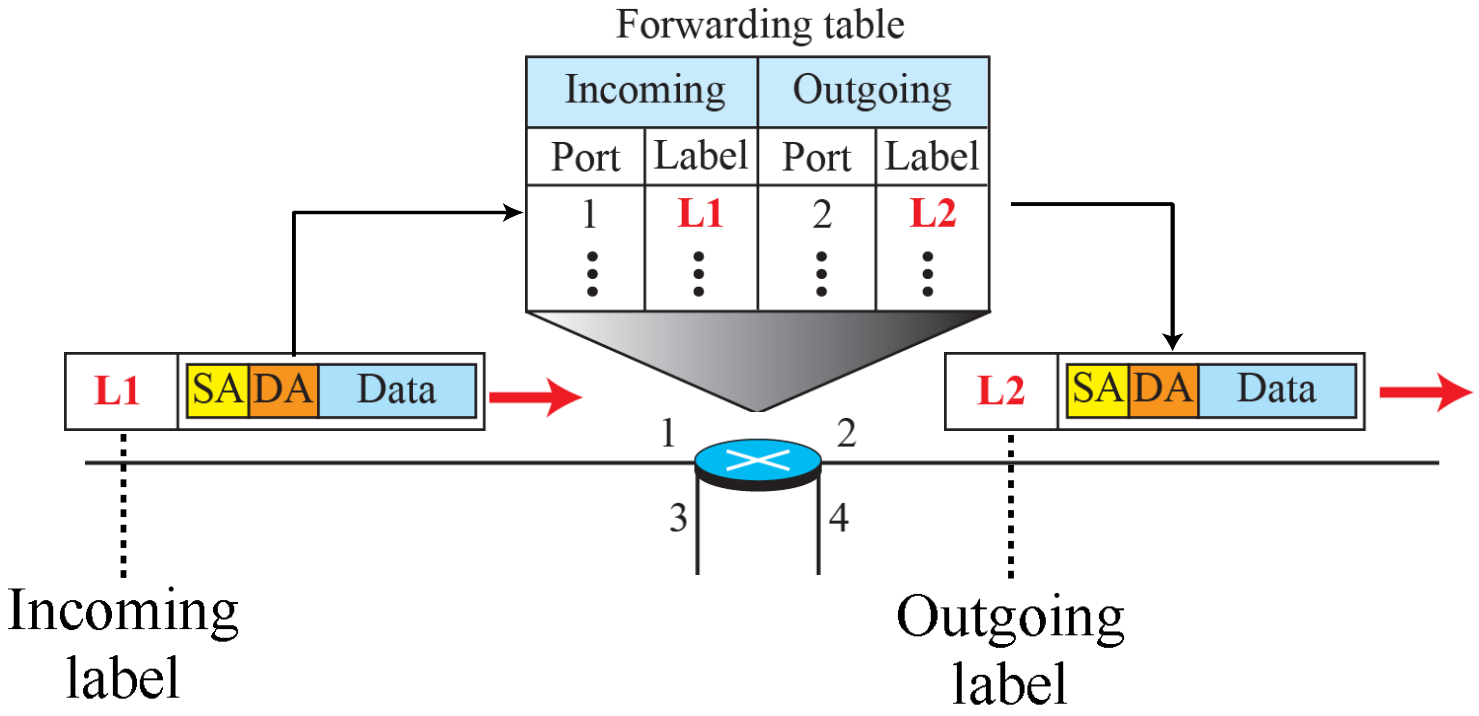


Figure 18.7: Sending request packet in a virtual-circuit network

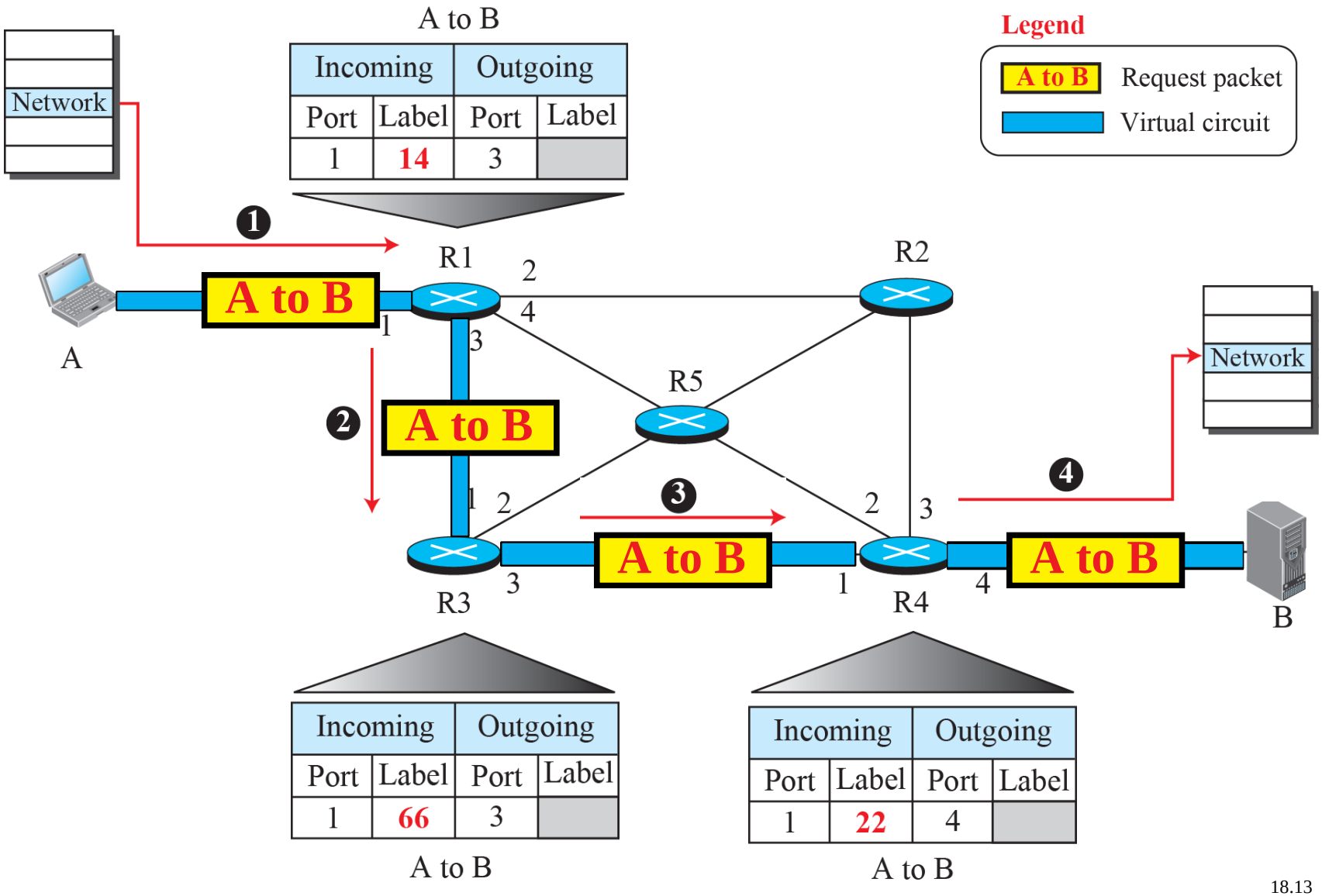


Figure 18.8: Sending acknowledgments in a virtual-circuit network

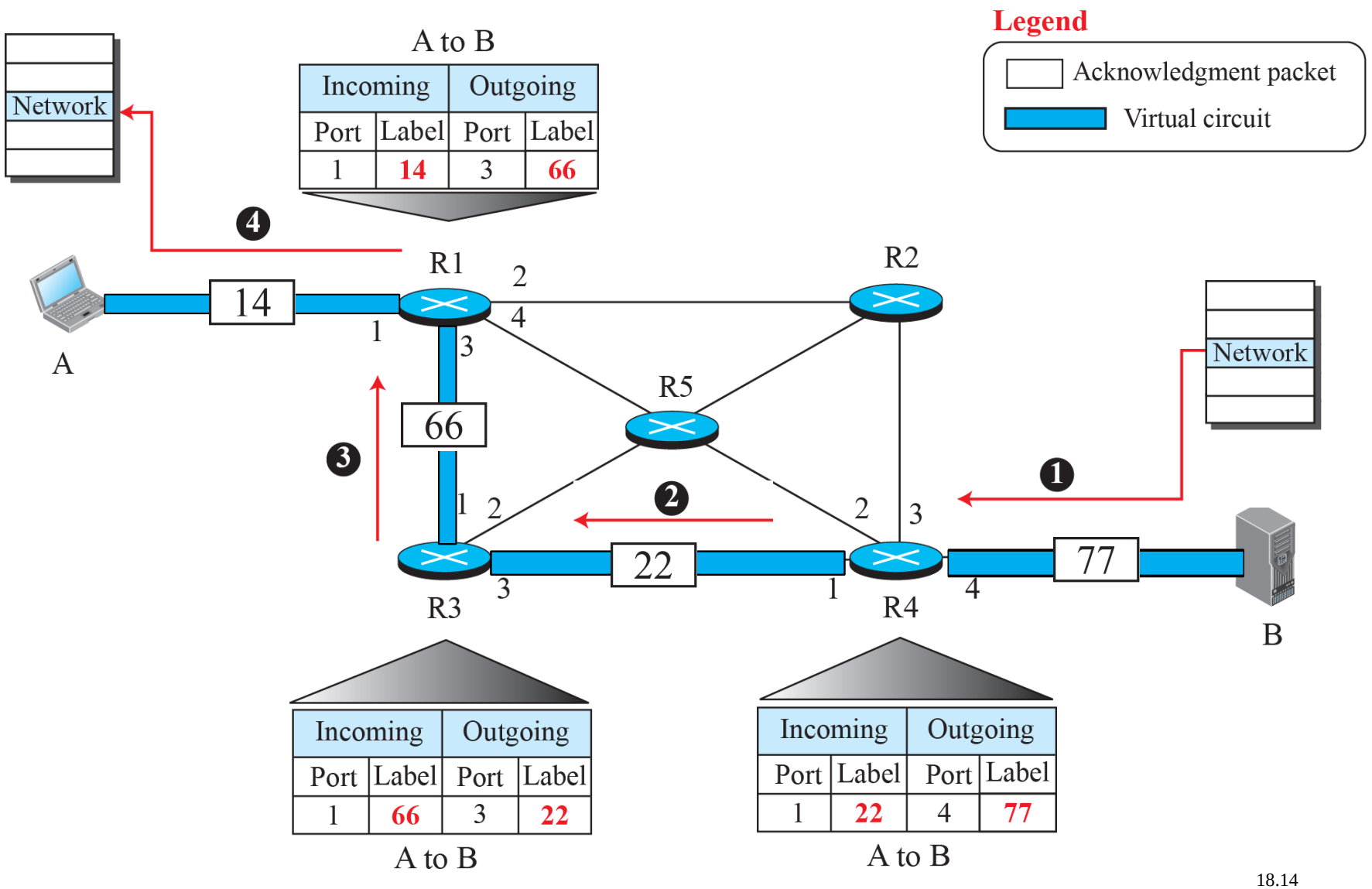
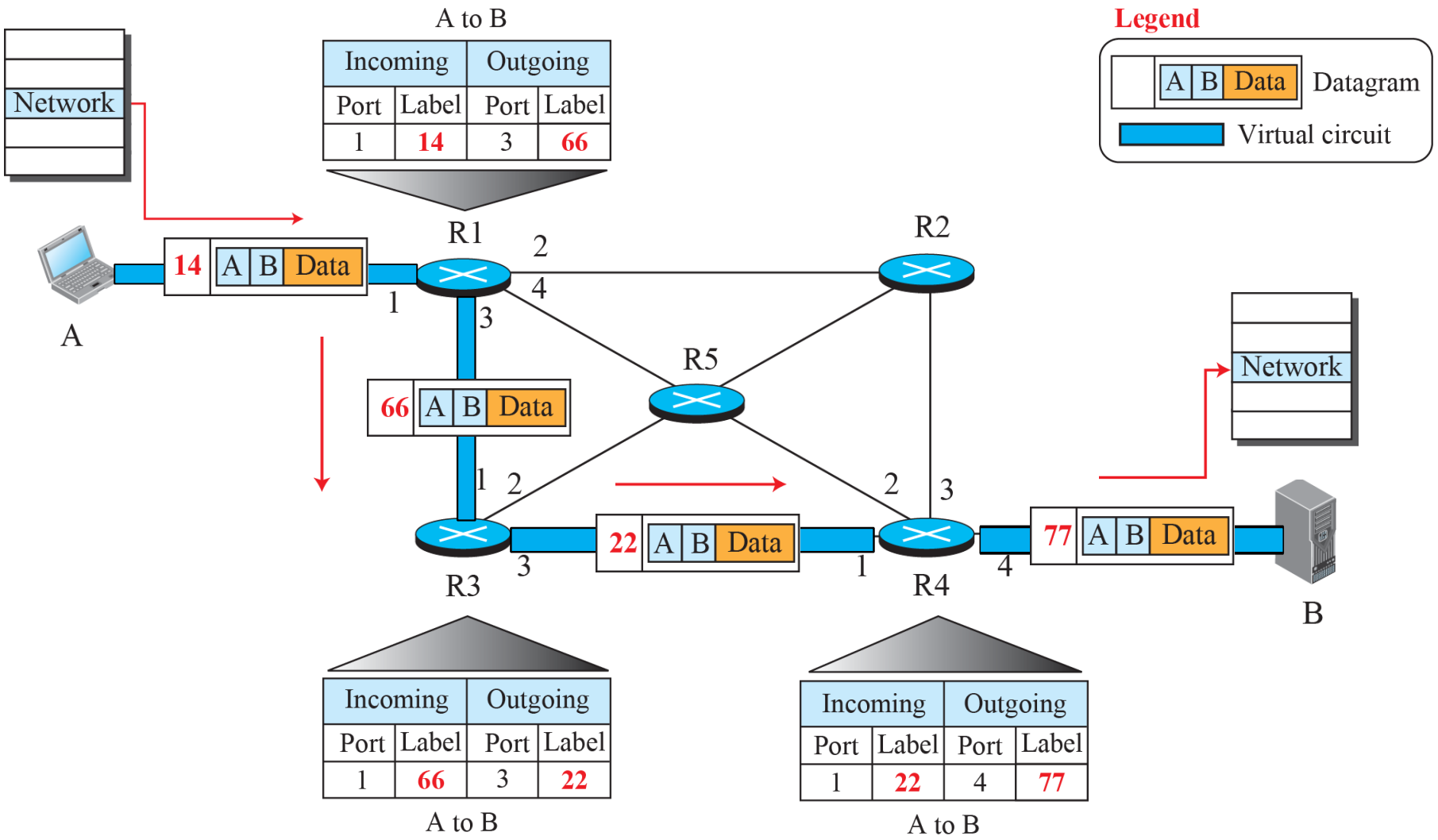
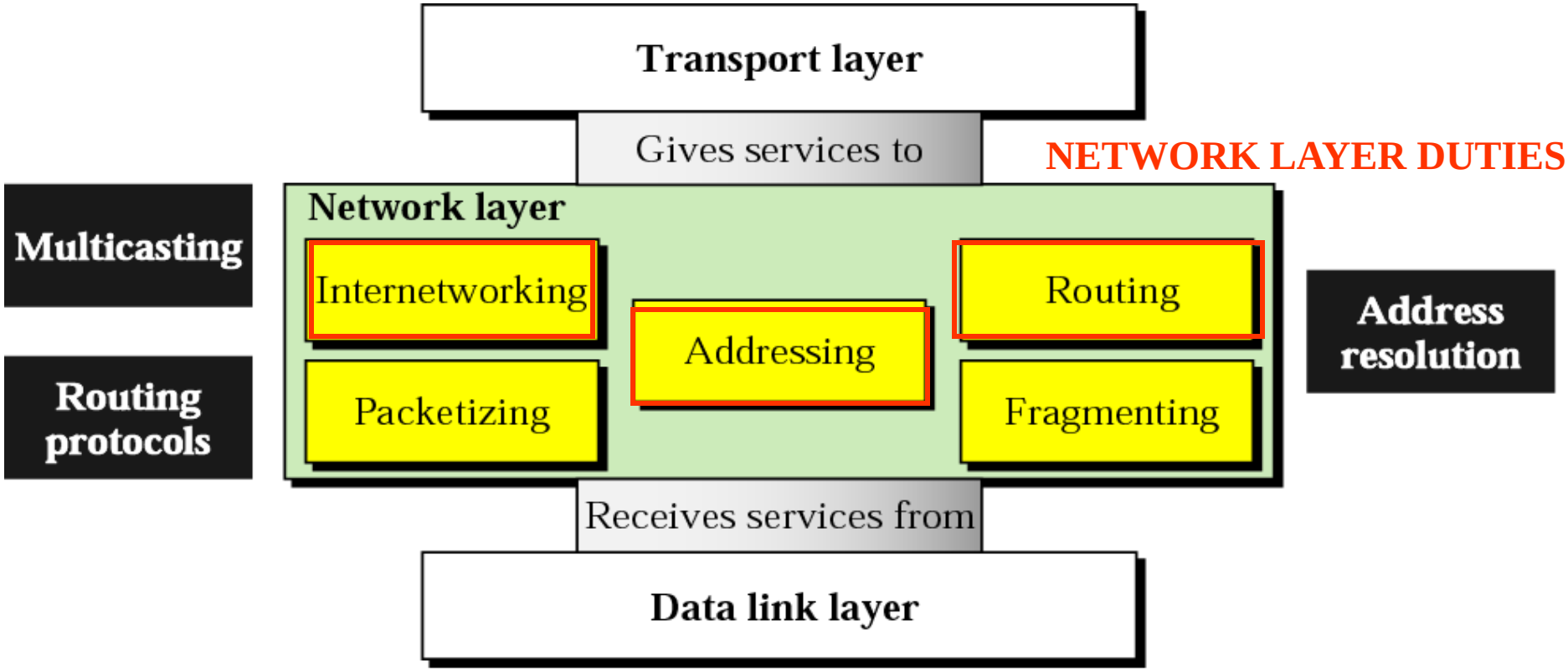


Figure 18.9: Flow of one packet in an established virtual circuit



POSITION of Network Layer Protocol



Review: Inter-Networks

- **Inter-Networks Solution => *IP Protocol* - [Ch. 20]**
  - **internet-layer gateways & global addresses**
  - **simple, application-independent, lowest denominator network service: best-effort datagrams**
  - **stateless gateways could easily route around failures**
  - **with application-specific knowledge out of gateways:**
    - NSPs no longer had monopoly on new services
    - Internet: a platform for rapid, competitive innovation

- Inter-Networks: Interconnects *broadcast domains*.
- IP Protocol is the network layer protocol which is responsible for host-to-host delivery and for routing the packets through the routers or switches (inter-network).

Internet & IP Datagrams

- **Internet**: a collection of networks connected by internetworkig devices such as routers or gateways.

  De factor. ="network of Network" as a connectionless TCP/IP Network

- **IP Network provides connectionless, unreliable delivery of *IP datagrams*. -> *IP Forwarding***
  - ***Connectionless*: each datagram is independent of all others.**
  - ***Unreliable:* there is no guarantee that datagrams are delivered correctly or at all.**

18-4 IPv4 ADDRESSES

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address. An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. The IP address is the address of the connection, not the host or the router.

IP Addresses

- **IP addresses are not the same as the underlying data-link (MAC) addresses.**

Why ?

IP Addressing : introduction

- **IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).**
  - **Uniquely identify on the Internet**
  - ***Indirect connectivity***
- **The address must include information about what *network* the receiving host is on** to allow global communication between all devices**. This makes routing feasible.**

❖ **MAC Address –** ***direct connectivity***

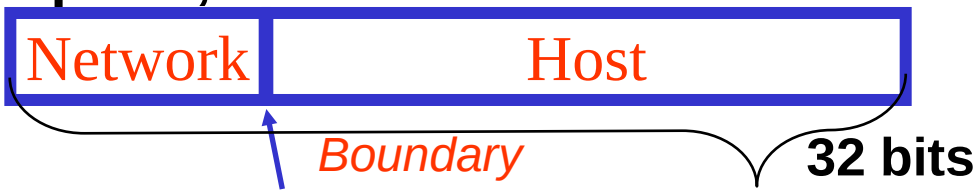
- **it is only valid within the same broadcast network**
- **It does not identify globally unique but Link Local Address within the same network**

Internet Addresses

- **The Internet uses three kinds of addresses:**
  - Application layer addresses **(Process ID) are assigned by network managers and placed in configuration files. Some servers have more than one application layer address.**
  - **Network layer addresses (IP addresses) are also assigned by network managers, or by programs such as DHCP, and placed in configuration files. Every network on the Internet is assigned a range of possible IP addresses for use on its network.**
  - Data link layer addresses **are hardware addresses placed on network interface cards by their manufacturers**
- **For a message to travel from sender to receiver, these addresses must be translated from one type to another. This process is called** address resolution

IPv4 Addresses (4bytes)

- **IP addresses (133.18.14.121) are *logical* addresses (not physical)**
- **Every host must have a unique IP address.**
- **32 bits.**
  - **Binary notation:** *10100110 0001100 00000011 0001111*
  - **Dotted-decimal notation:** *166.12.3.31*
- **Splitting address (called *hierarchical addressing*) into multiple parts; a network ID and a host ID.**



- **IP addresses are assigned by a central authority (*Internet Corporation for Assigned Names and Numbers -- ICANN)***

Network and Host IDs

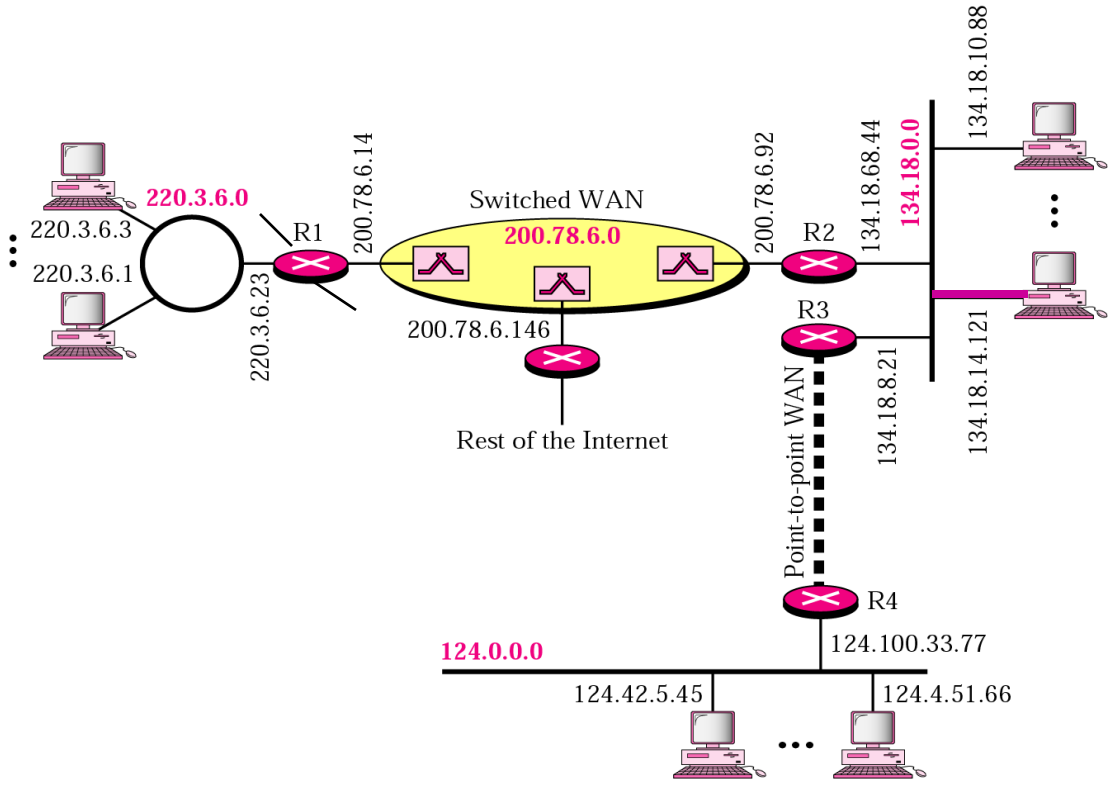
- **A Network ID is assigned to an organization by a global authority.**
  - **Hosts that share the same IP *network* address (the network ID).**

- **Host IDs are assigned locally by a system administrator.**
  - **A single network interface is assigned a single IP address called the *host* address.**
  - **A host may have multiple interfaces, and therefore multiple *host* addresses.**

- **Both the Network ID and the Host ID are used for routing.**

IP Addressing: Example

- ***How to find if destination is in the network ?***
- **133.18.14.121 : An identifier for host, router *interface***
- ***Interface:* connection between host, router and physical link**
  - **router's typically have multiple interfaces**
  - **host may have multiple interfaces**
  - **IP addresses associated with interface, not host, router**



133.18.14.121 = 10000101 00010010 00001110 01111001

133 18 14 121

Example 1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11111001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Figure 18.16: Three different notations in IPv4 addressing

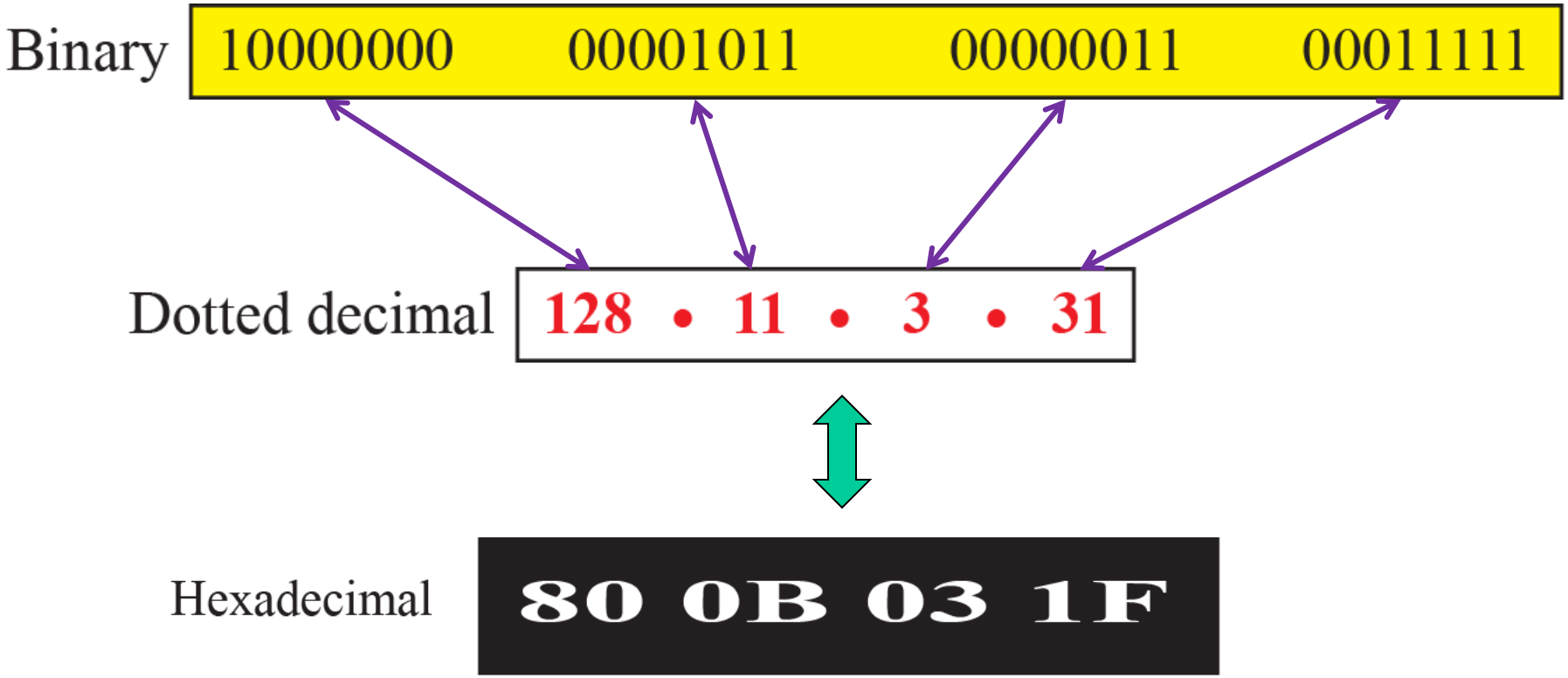


Figure 18.17: Hierarchy in addressing

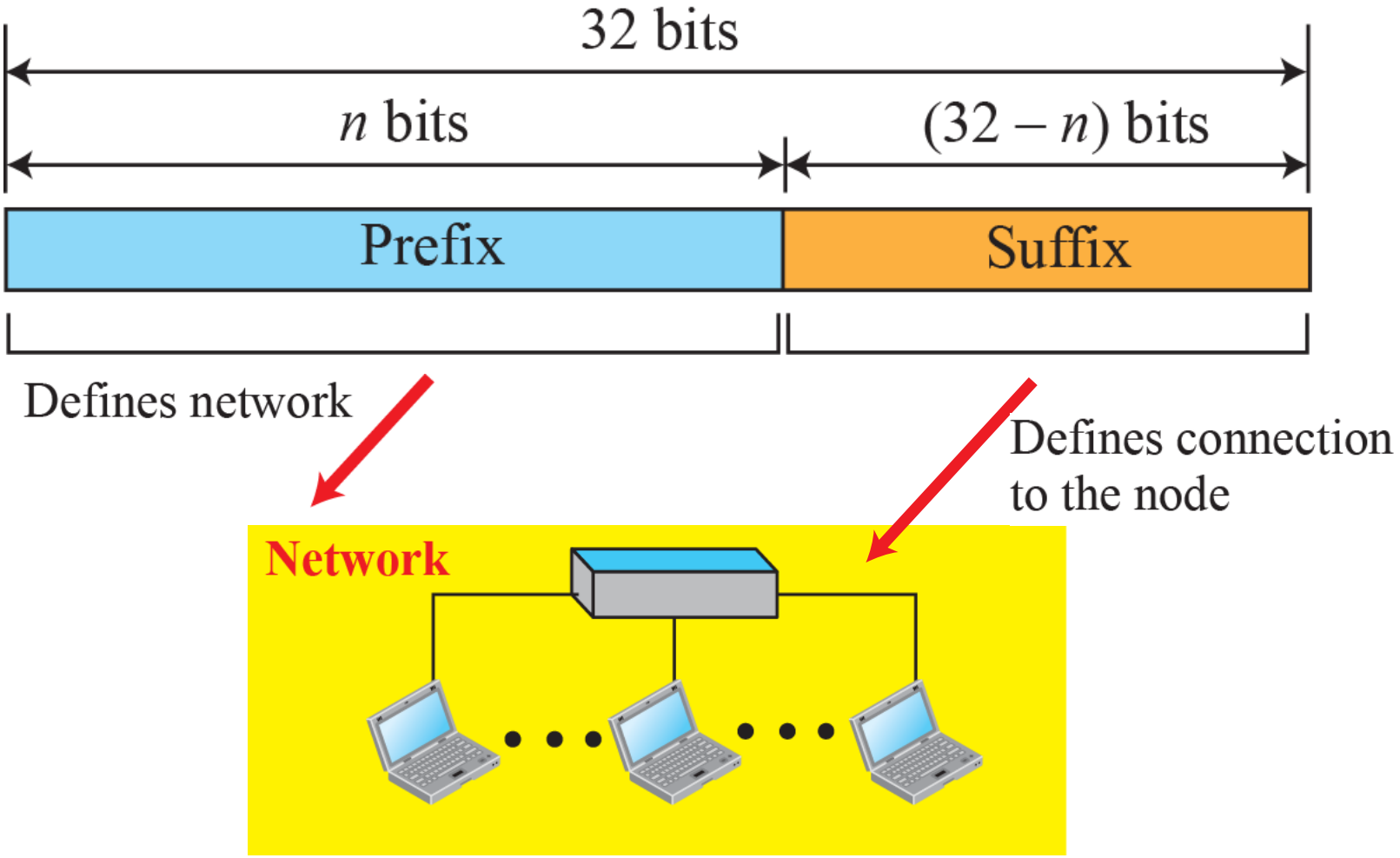
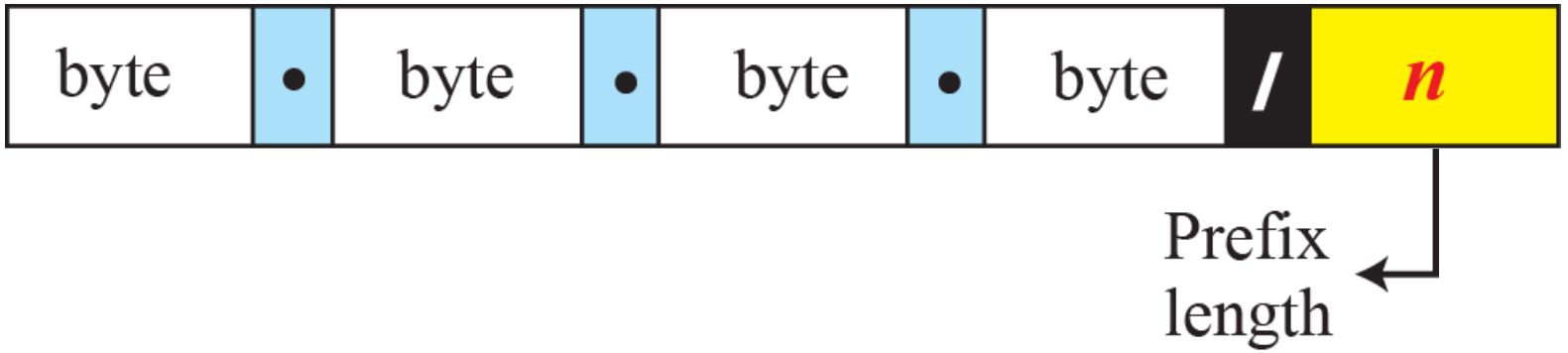


Figure 18.20: Slash notation (CIDR)



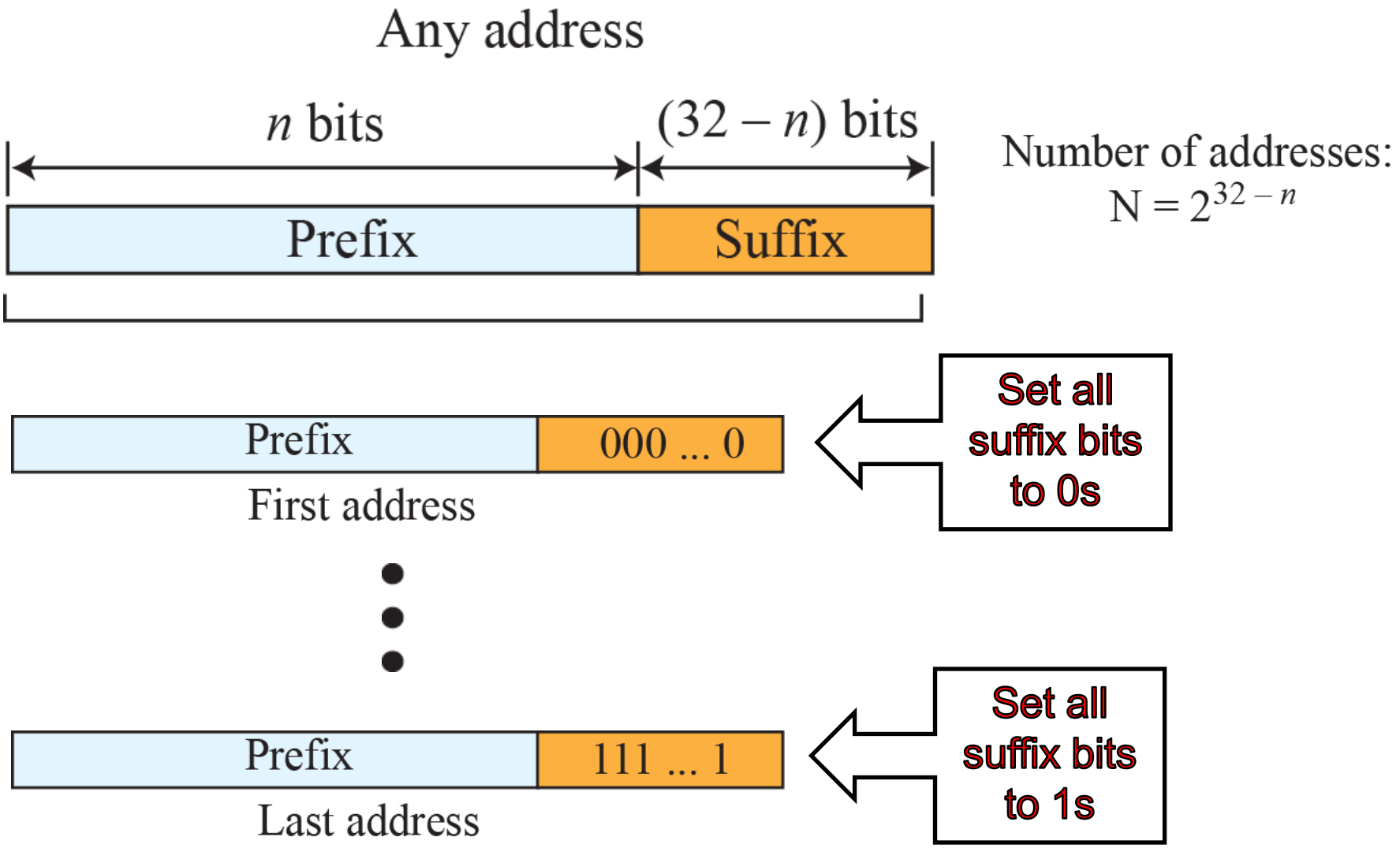
Examples:

12.24.76.8/**8**

23.14.67.92/**12**

220.8.24.255/**25**

Figure 18.21: Information extraction in classless addressing



SPECIAL ADDRESSES

IP Broadcast and Network Addresses

Netid	Hostid	Type of Address	Purpose
all- 0s	all – 0s	This computer	Used during bootstrap
network	all-0s	Network	Identifies a network
Network	all-1s	Directed broadcast	Broadcast on specified net
All – 1s	all –1s	Limited broadcast	Broadcast on local net
127	Any	Loop back	testing

Types of network addresses

Address Type	Example Software	Example Address
Application Layer	Web Browser E-mail	www.kelley.indiana.edu Web@hye.hanyang.ac.kr
Network Layer	**IP**	**129.79.127.4**
Data Link Layer	Ethernet	00-0C-00-F5-03-5A

Addressing Summary

- Unique IP address per interface
- Classful (A,B,C) => address allocation not efficient
- Hierarchical => smaller routing tables
- Provision for broadcast, multicast, loopback addresses
- Subnet masks allow “subnets” within a “network” => improved address allocation efficiency
- Supernet (CIDR) allows variable sized network ID allocation

In IPv4 addressing, a block of addresses can be defined as
x.y.z.t /*n*
in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned

Names and Numbers

- **allocates addresses**
- **manages DNS**
- **assigns domain names, resolves disputes**

IP addresses: how to get one?

Network (network portion):

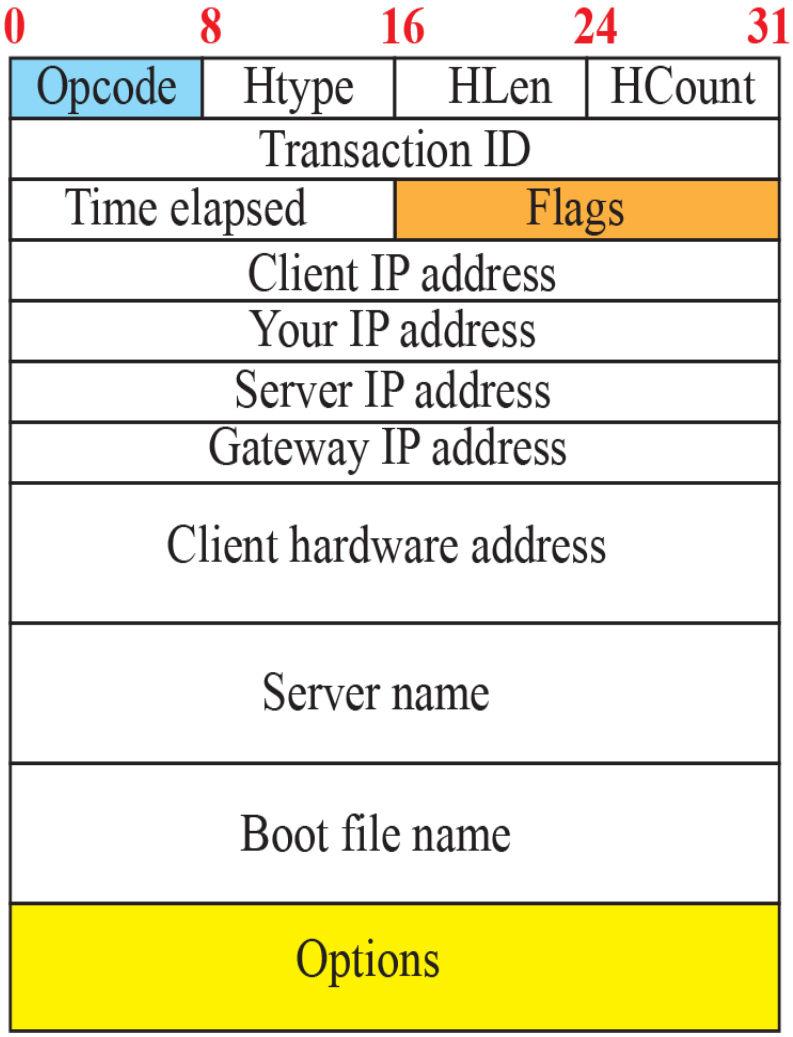
- **get allocated portion of ISP's address space:**

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
…	…..	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

18.4.4 DHCP

After a block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers. However, address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP). DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.

Figure 18.25: 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 18.26: Option format

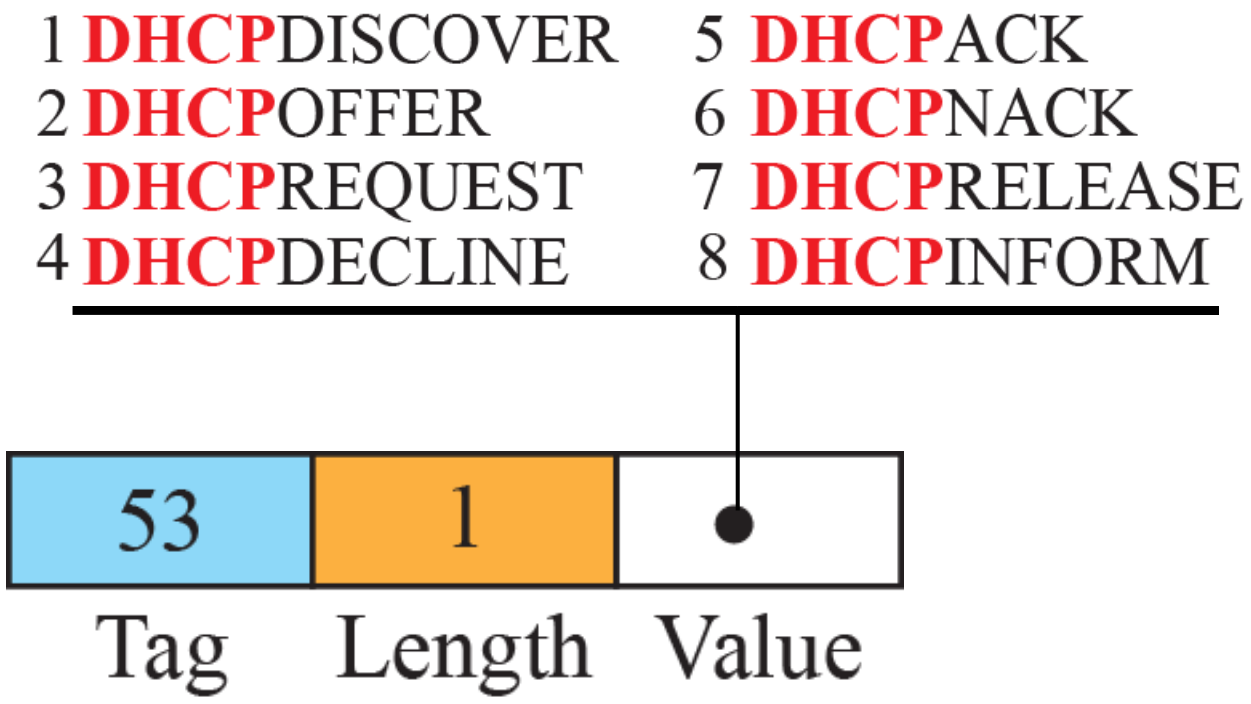


Figure 18.27: Operation of DHCP

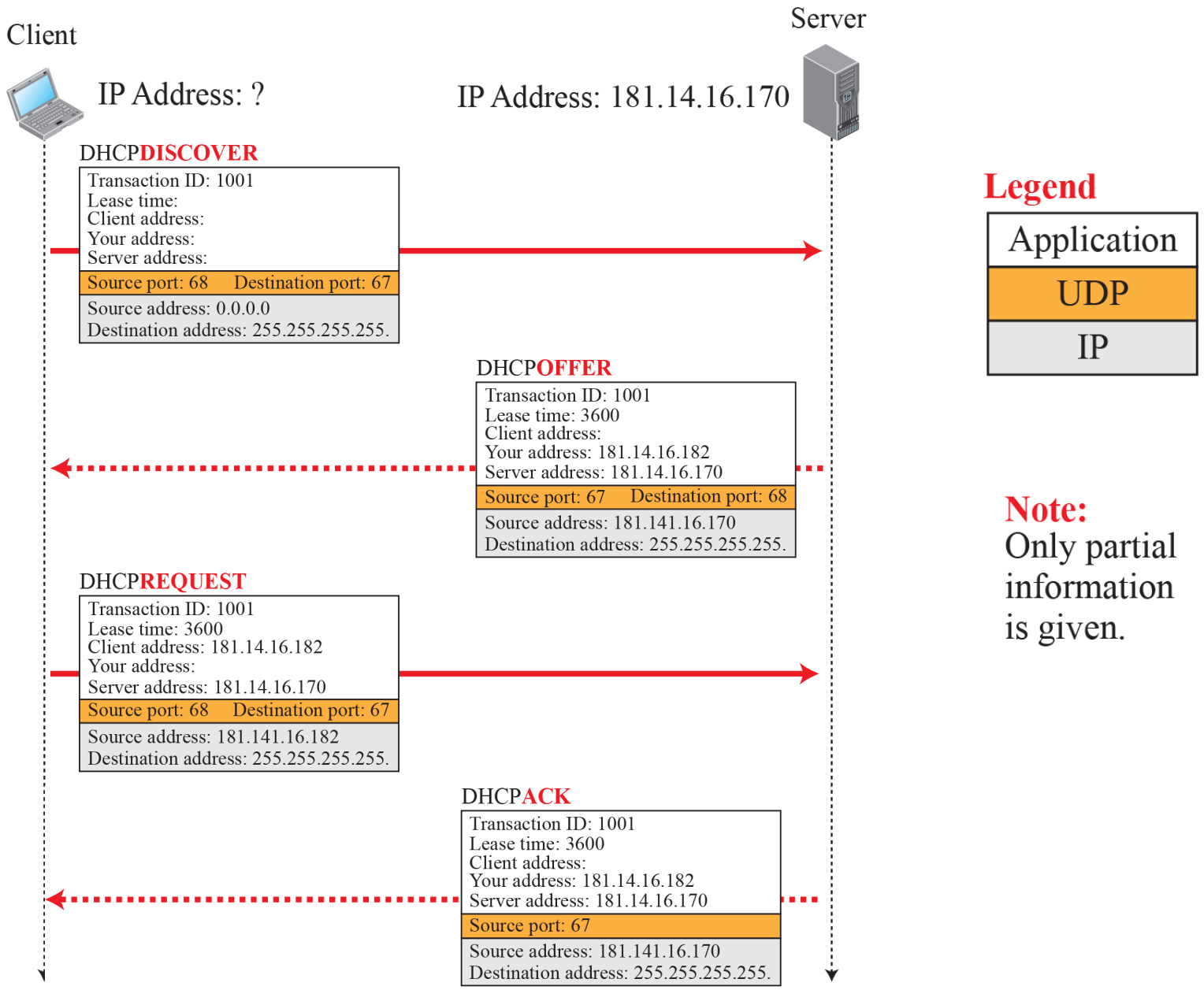
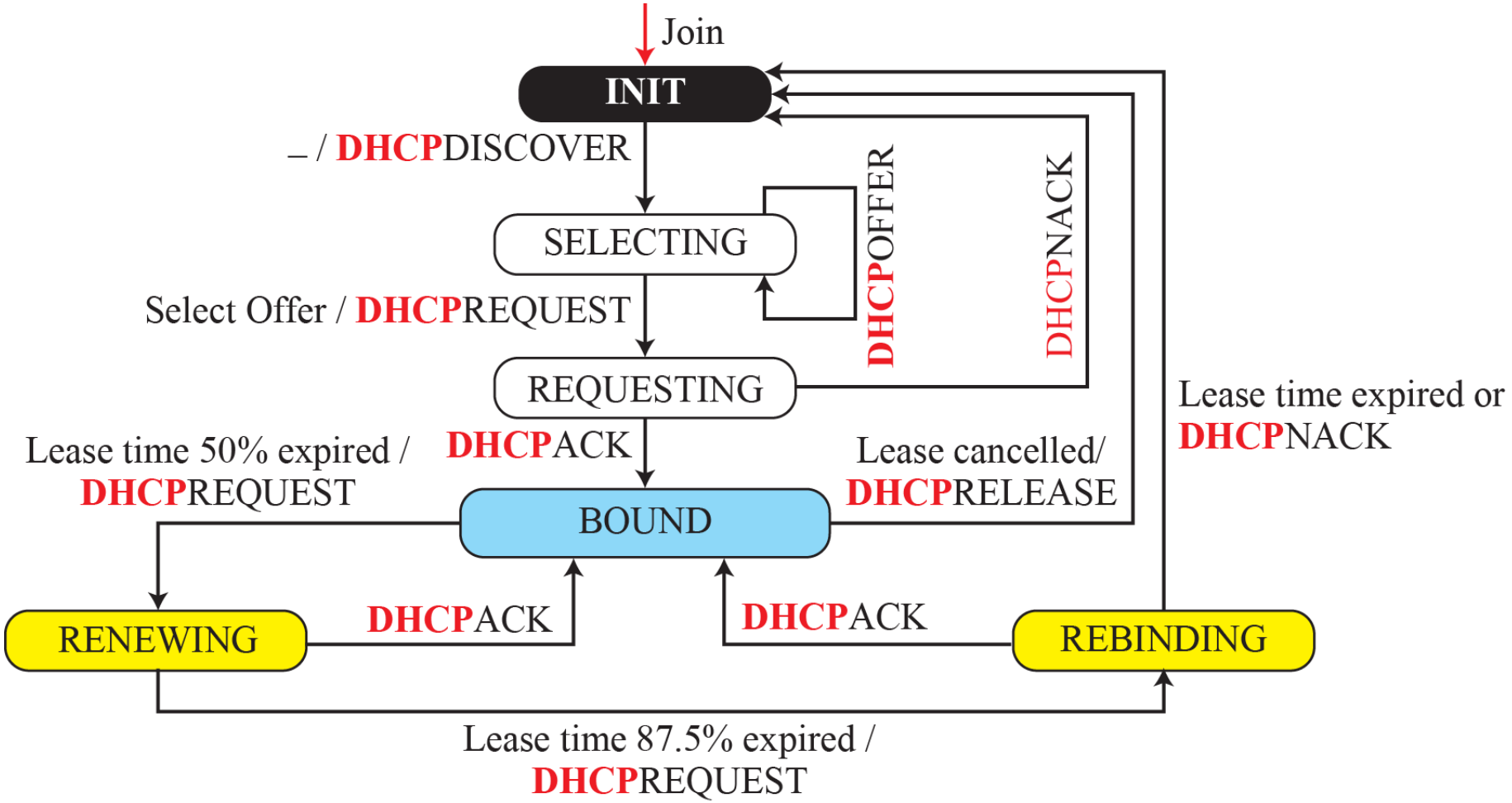
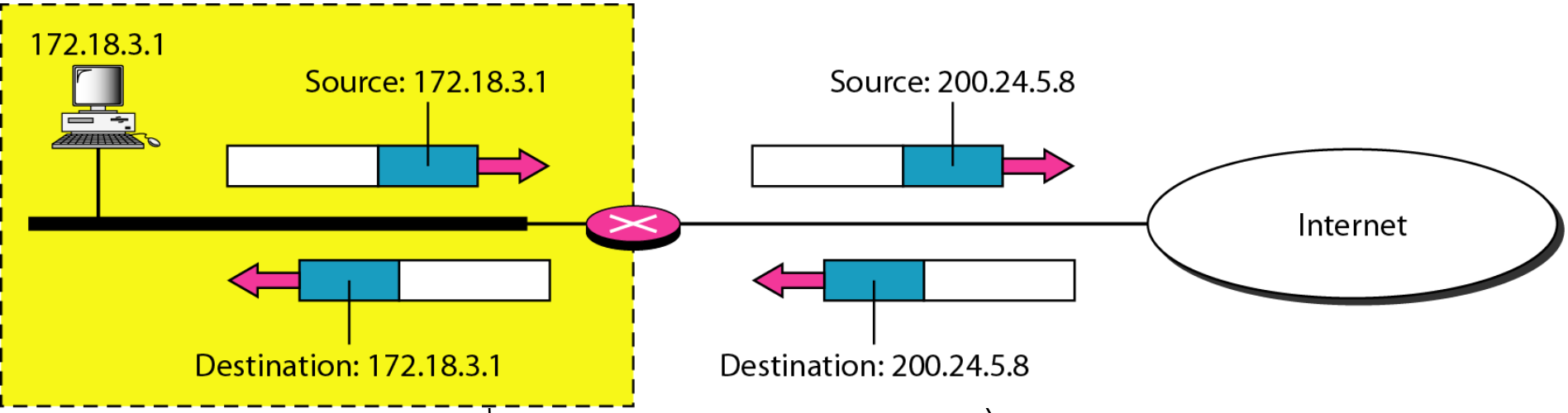


Figure 18.28: FSM for the DHCP client



18.4.5 NAT NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 200.24.5.8, different source port numbers

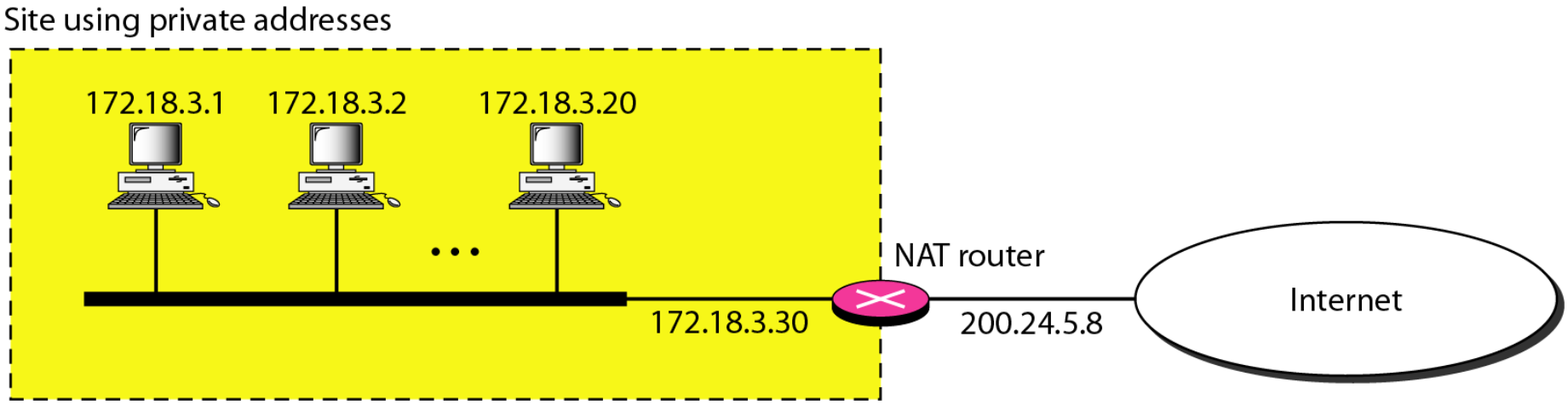
One IP Address

Datagrams with source or destination in this network have 172.18.3.0/24 address for source, destination (as usual)

Largest set of IP Addresses

NAT: Network Address Translation

- **Motivation:** local network uses just one IP address as far as outside word is concerned:
  - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).



NAT: Network Address Translation

Implementation: NAT router must:

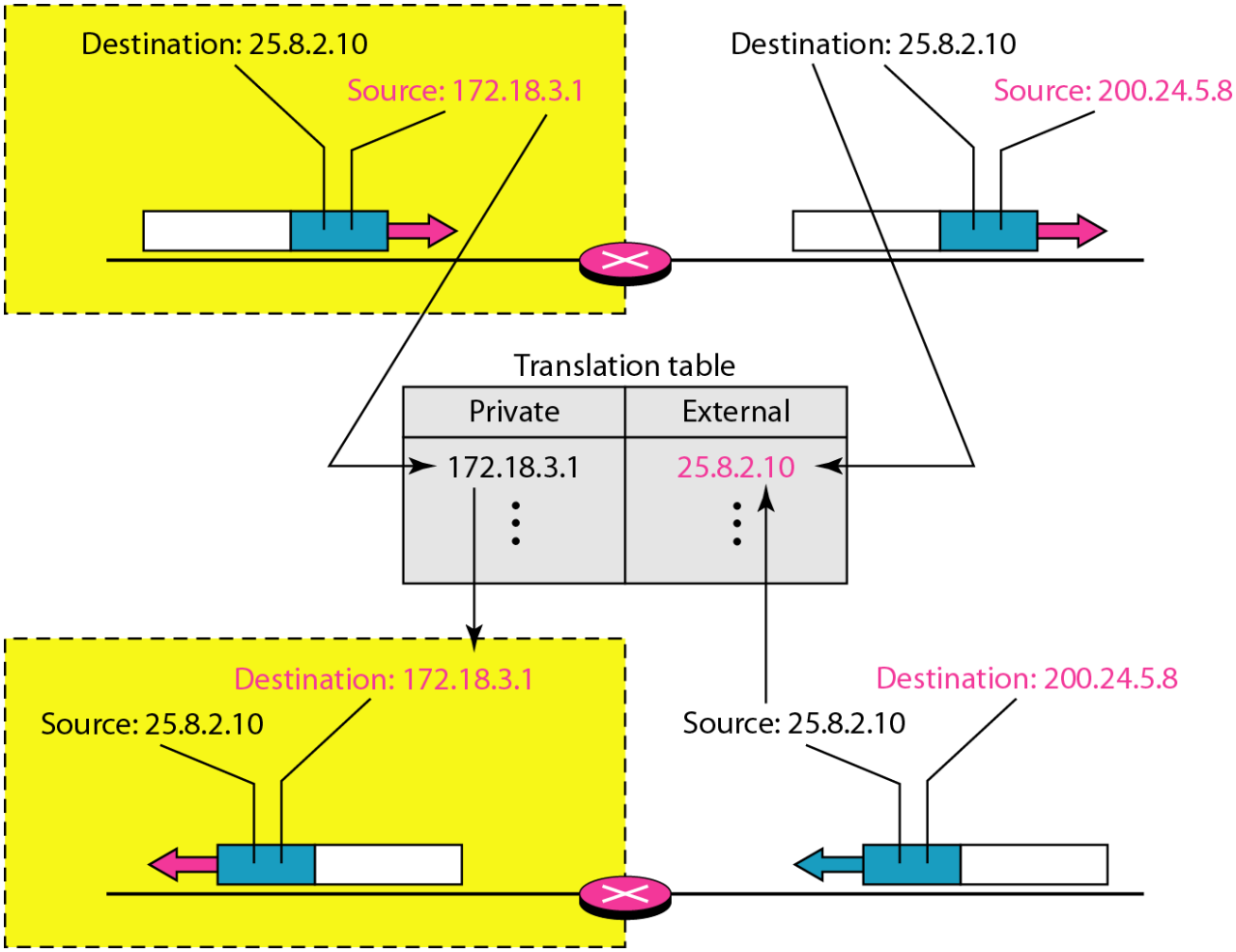
- ***outgoing datagrams: replace*** **(source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)**

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.

- ***remember (in NAT translation table)*** **every (source IP address, port #) to (NAT IP address, new port #) translation pair**

- ***incoming datagrams: replace*** **(NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table**

NAT: Network Address Translation



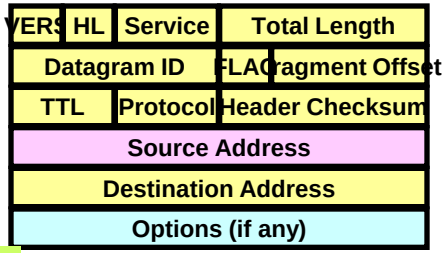
NAT: Network Address Translation

- **16-bit port-number field:**
  - **60,000 simultaneous connections with a single LAN-side address!**
- **NAT is controversial:**
  - **routers should only process up to layer 3**
  - **violates end-to-end argument**
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - **address shortage should instead be solved by IPv6**

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

Table 19.4 *Five-column translation table*

19-2 IPv6 ADDRESSES: Motivation of IPv6



- **Initial motivation:** 32-bit address space completely allocated by 2008.

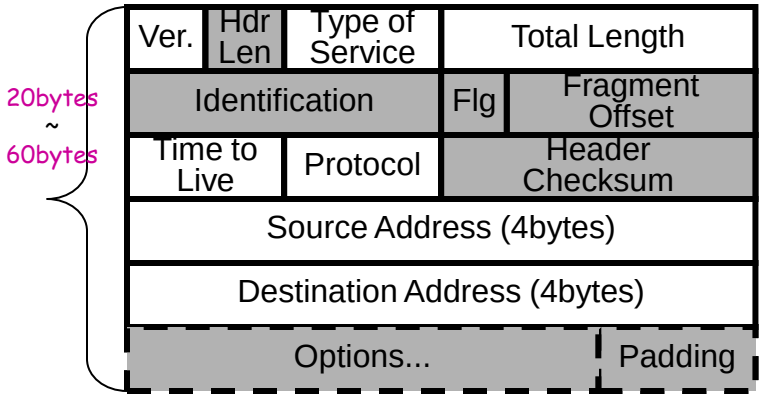
more IP addresses! 16 bytes

- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
  - new "anycast" address: route to "best" of several replicated servers
  - Mobility for mobile user

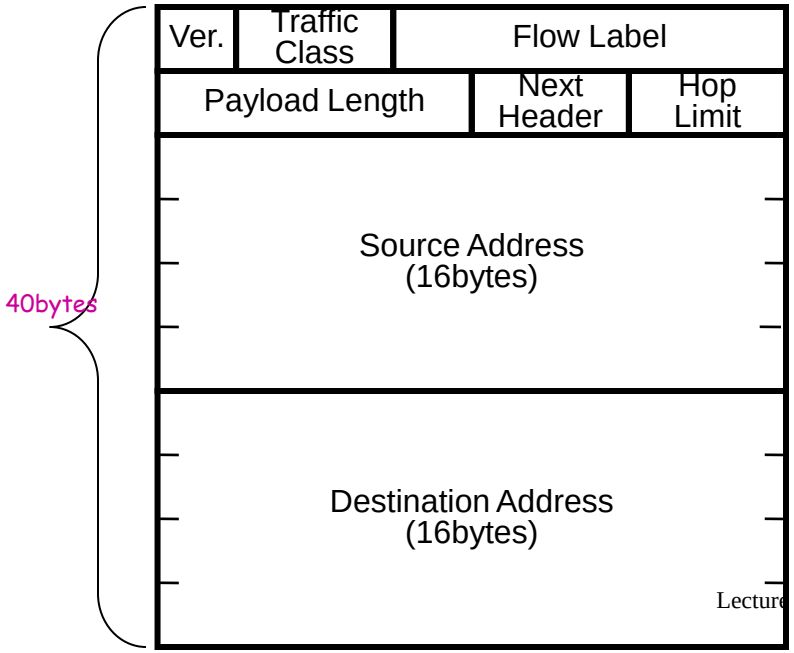
- **IPv6 datagram format:**
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6

- **Next generation IP protocol**
  1. Large address space: 4bytes -> 16 bytes (2^{96})
  2. Better header format: variable size -> fixed size
  3. Support for high-speed processing: no fragmentation
  4. Support for resource allocation: type of service -> flow label
  5. Support for security: encryption and authentication



shaded fields are absent from IPv6

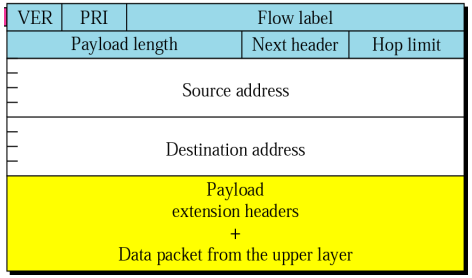


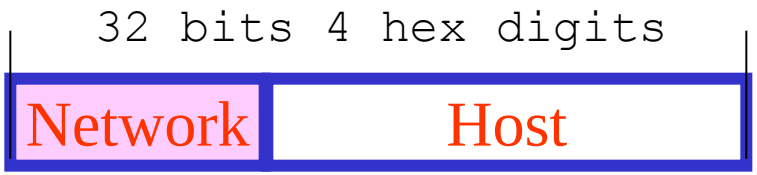
How Was IPv6 Address Size Chosen?

- **some wanted fixed-length, 64-bit addresses**
  - **easily good for 10^{12} sites, 10^{15} nodes, at .0001 allocation efficiency (3 orders of mag. more than IPng requirement)**
  - **minimizes growth of per-packet header overhead**
  - **efficient for software processing**
- **some wanted variable-length, up to 160 bits**
  - **compatible with OSI NSAP addressing plans**
  - **big enough for auto-configuration using IEEE 802 addresses**
  - **could start with addresses shorter than 64 bits & grow later**
- **settled on fixed-length, 128-bit addresses**

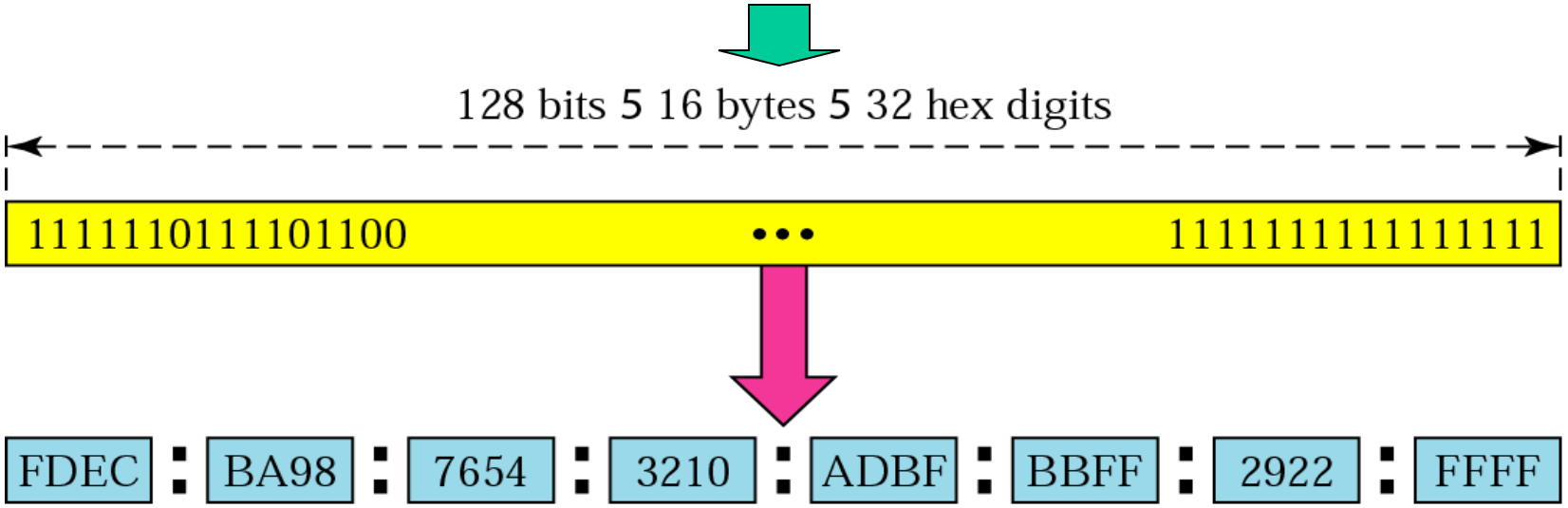
  (340,282,366,920,938,463,463,374,607,431,768,211,456 **in all!)**

Large Address Space





223.1.1.1



IPv6 Address Representation

- Dot-Decimal: **127.23.45.88**
- Colon-Hex:

"preferred" form: 1080:0:FF:0:8:800:200C:417A

compressed form: FF01:0:0:0:0:0:0:43

becomes FF01::43

IPv4-embedded: 0:0:0:0:0:FFFF:13.1.68.3

or ::FFFF:13.1.68.3

in URLs: http://[3FFE::1:800:200C:417A]:8000

(square-bracket convention also used anywhere else there's a conflict with address syntax)

Basic Address Types

unicast:

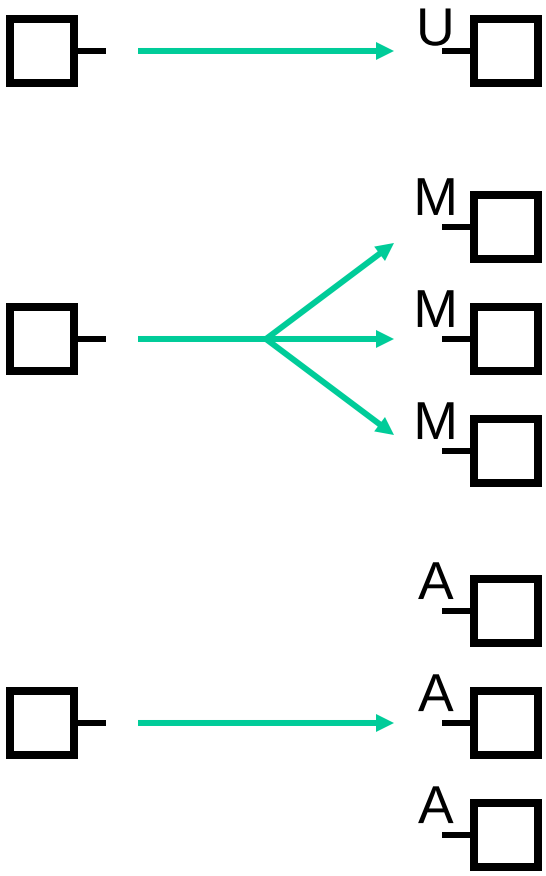
for one-to-one communication

multicast:

for one-to-many communication

anycast:

for one-to-nearest communication



IPv6 Address Types

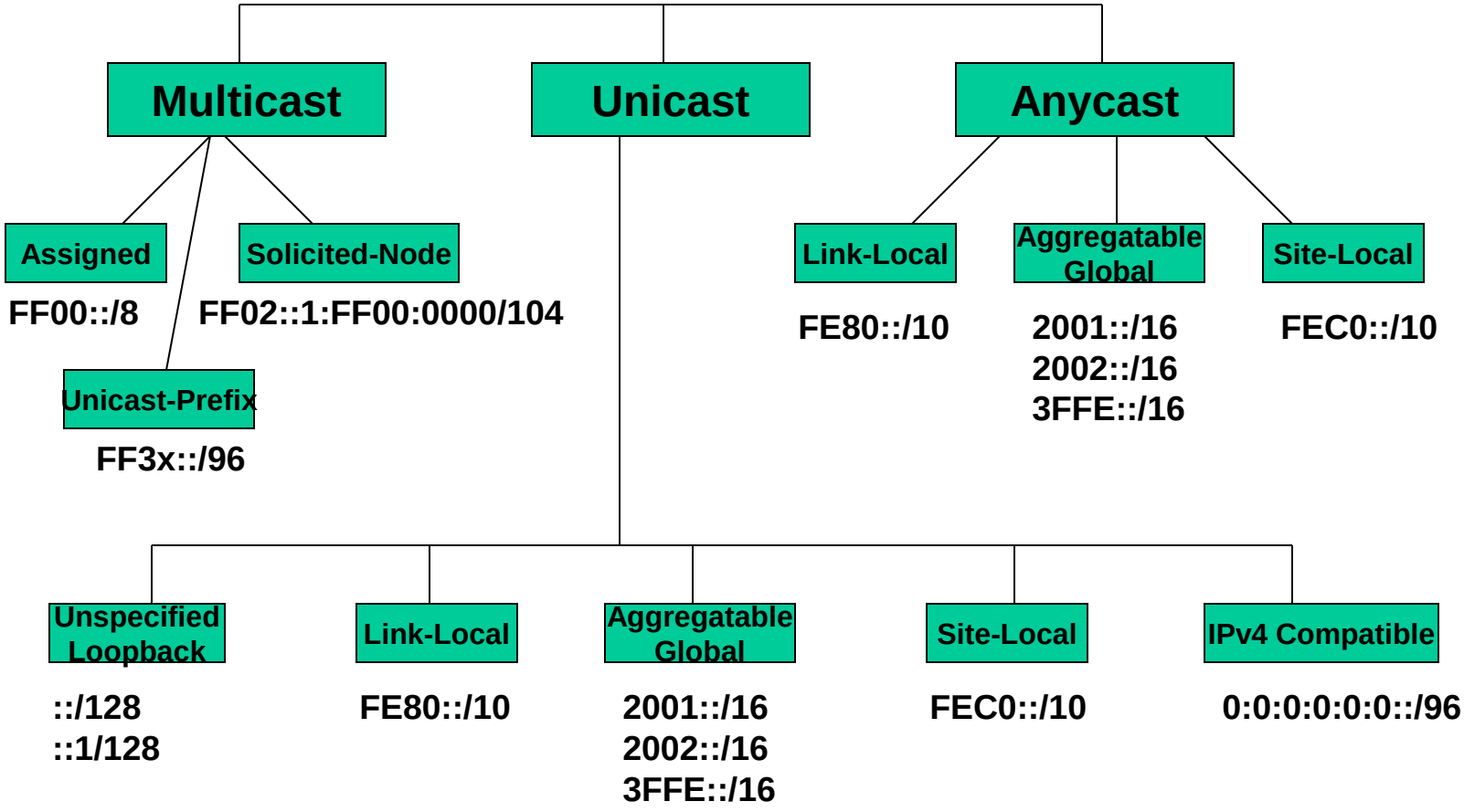


Figure 19.16 *Prefixes for provider-based unicast address*

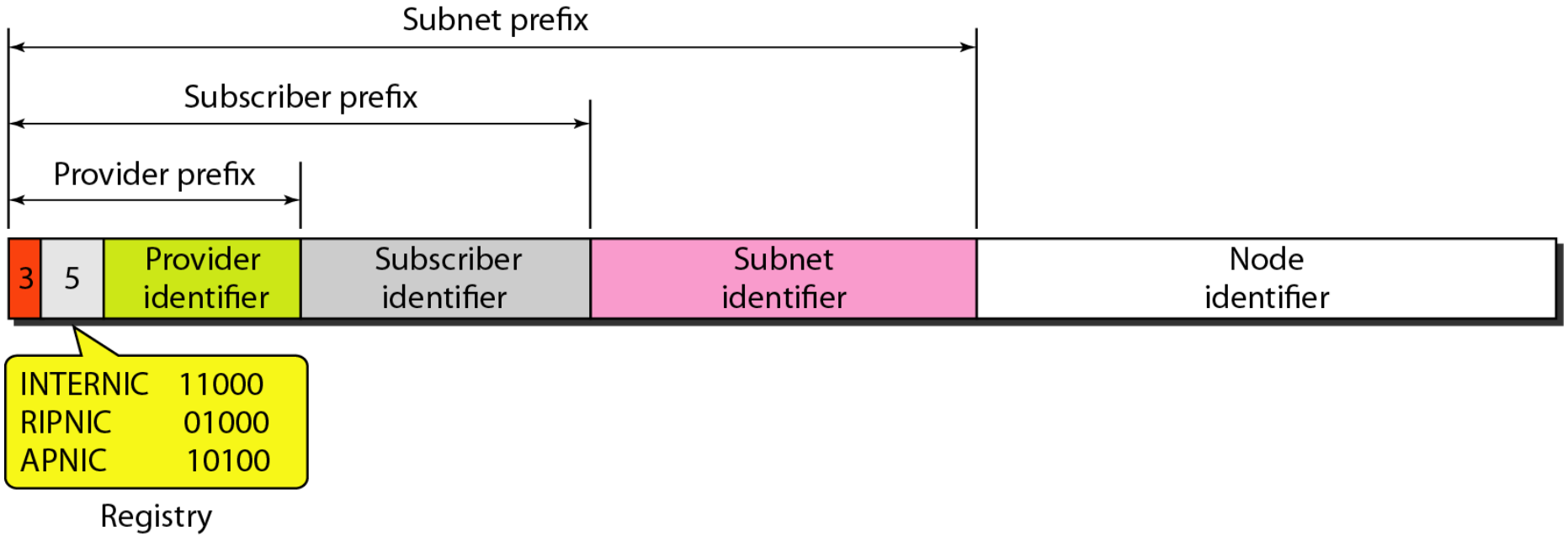


Figure 19.17 *Multicast address in IPv6*

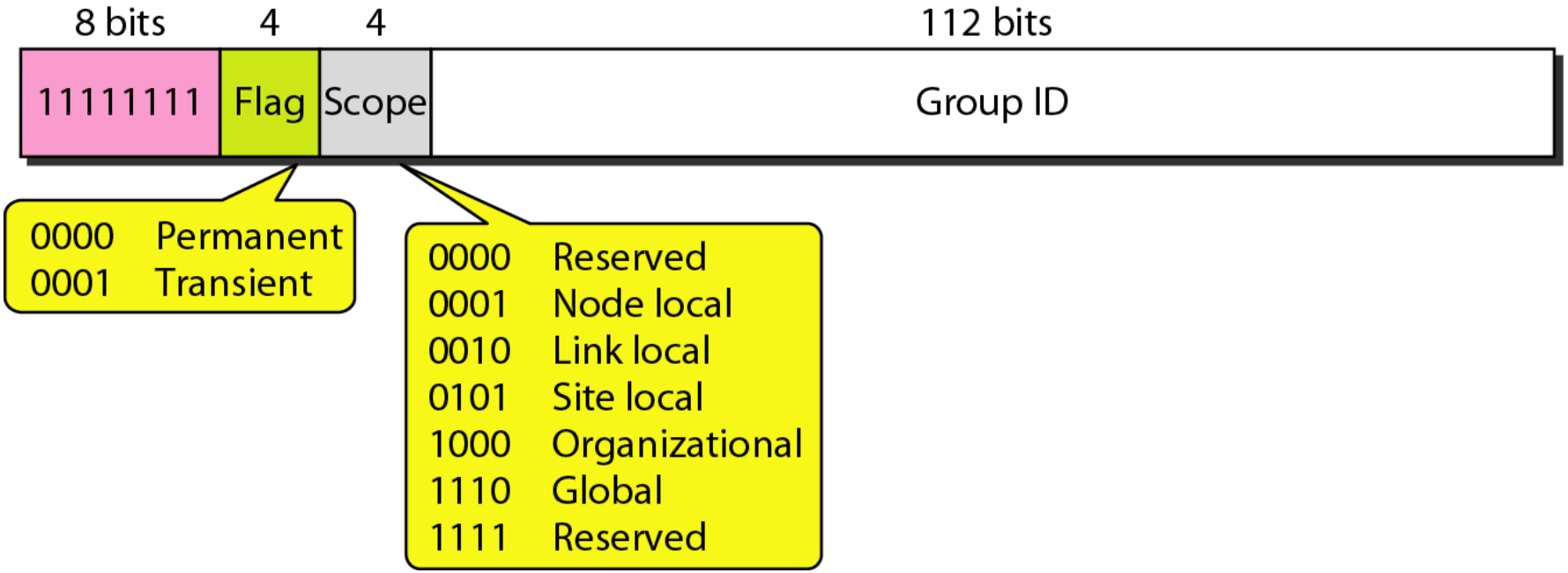


Figure 19.18 *Reserved addresses in IPv6*

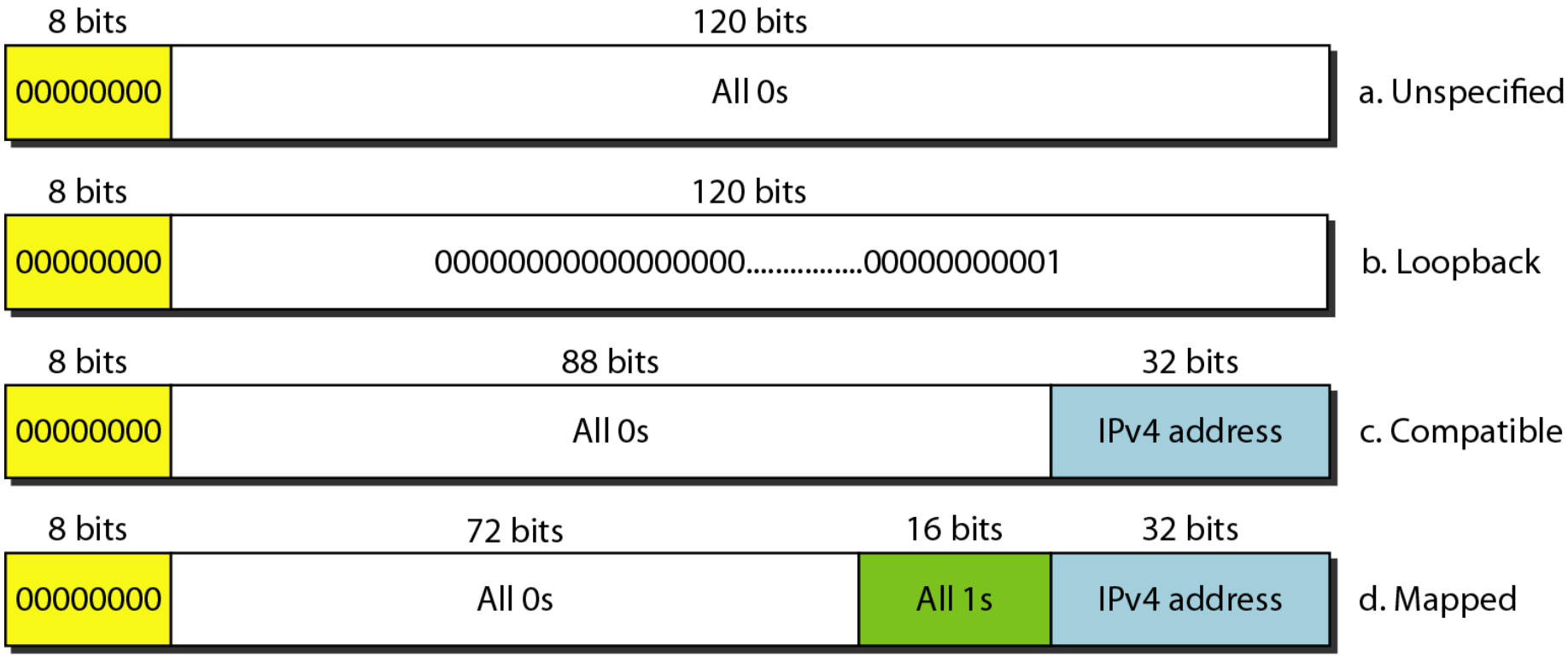
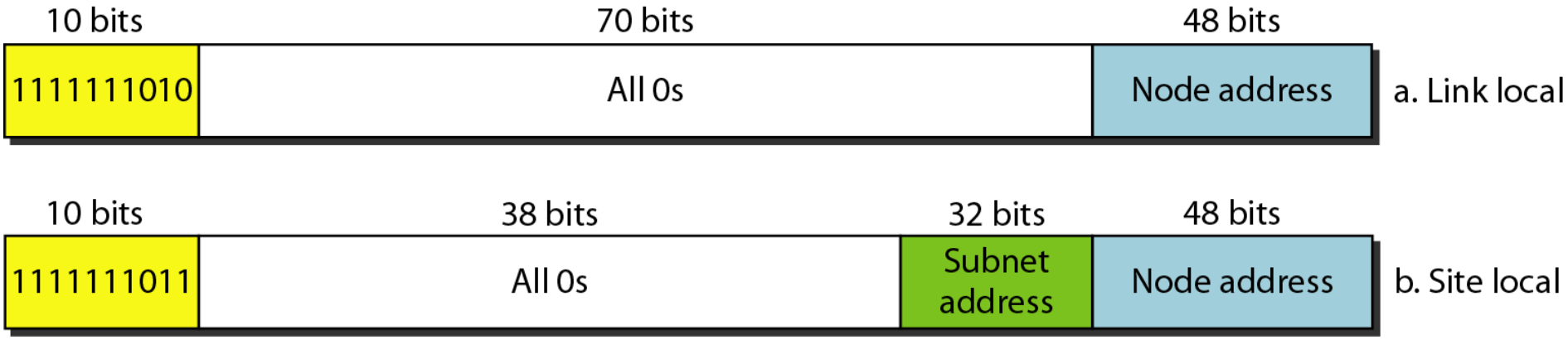


Figure 19.19 *Local addresses in IPv6*



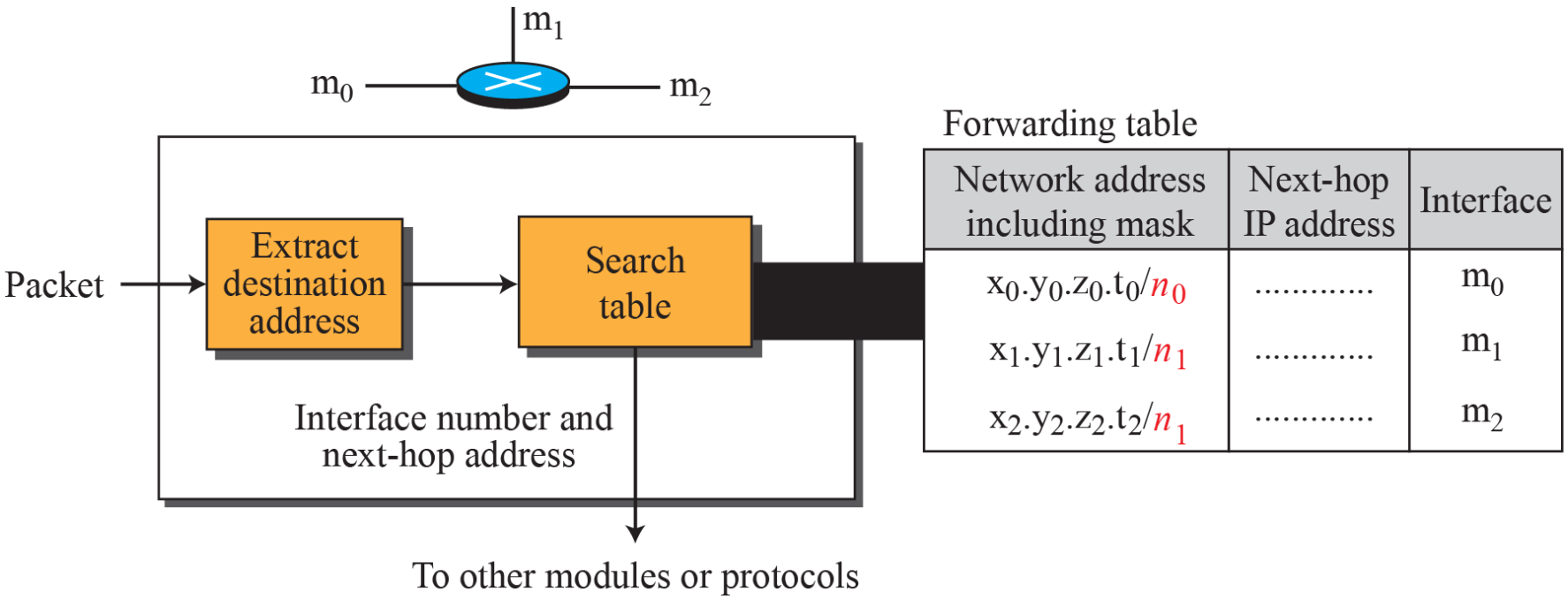
18-5 FORWARDING OF IP PACKETS

We discussed the concept of forwarding at the network layer earlier in this chapter. In this section, we extend the concept to include the role of IP addresses in forwarding. As we discussed before, forwarding means to place the packet in its route to its destination.

18.5.1 Destination Address Forwarding

We first discuss forwarding based on the destination address. This is a traditional approach, which is prevalent today. In this case, forwarding requires a host or a router to have a forwarding table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the next hop to deliver the packet to.

Figure 18.32: Simplified forwarding module in classless address



Make a forwarding table for router R1 using the configuration in Figure 18.33.

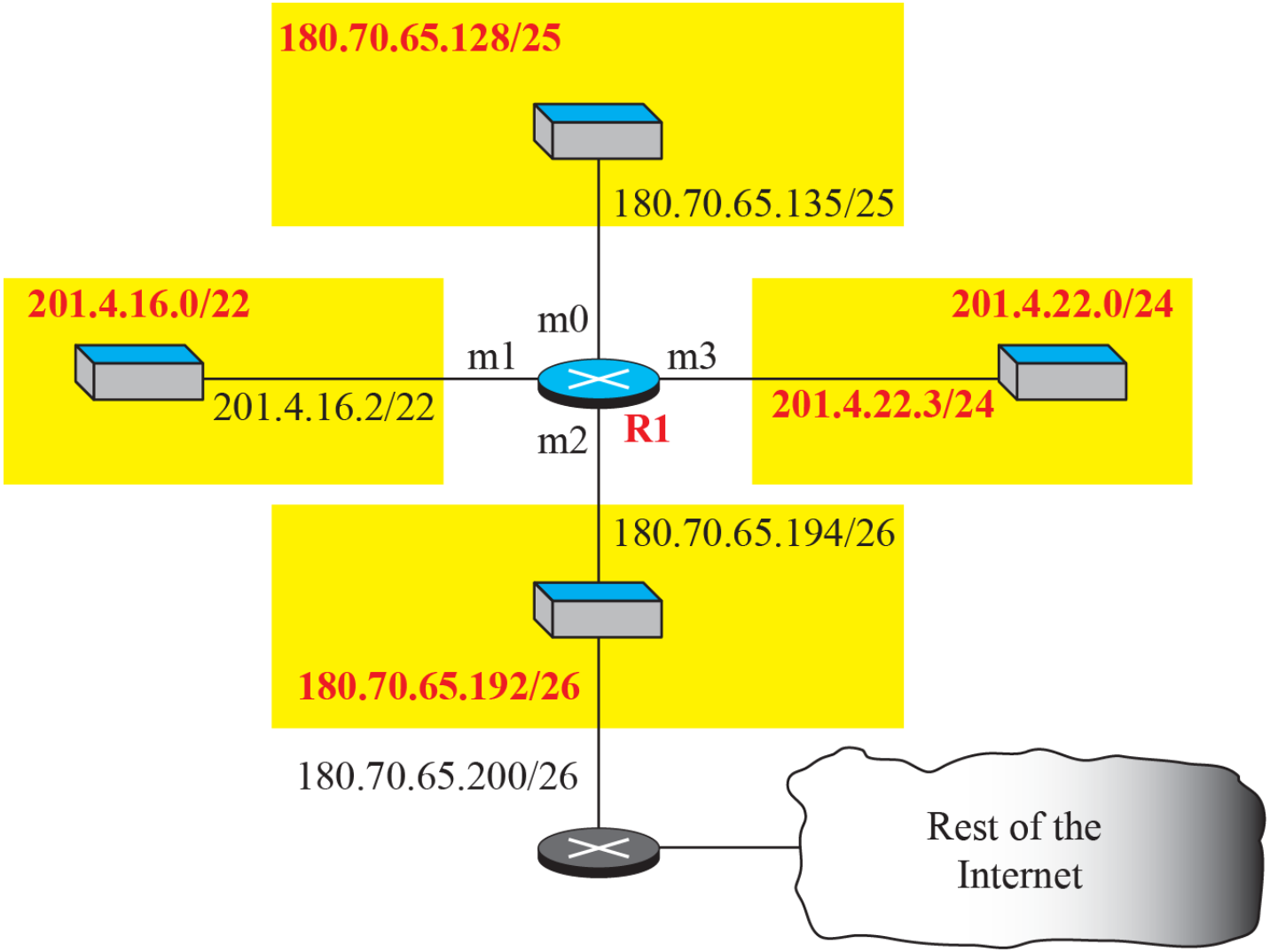
Solution

Table 18.2 shows the corresponding table.

Table 18.2: Forwarding table for router R1 in Figure 4.46

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

Figure 18.33: Configuration for Example 4.7



Instead of Table 18.2, we can use Table 18.3, in which the network address/mask is given in bits.

Table 18.3: Forwarding table for router R1 using prefix bits

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

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

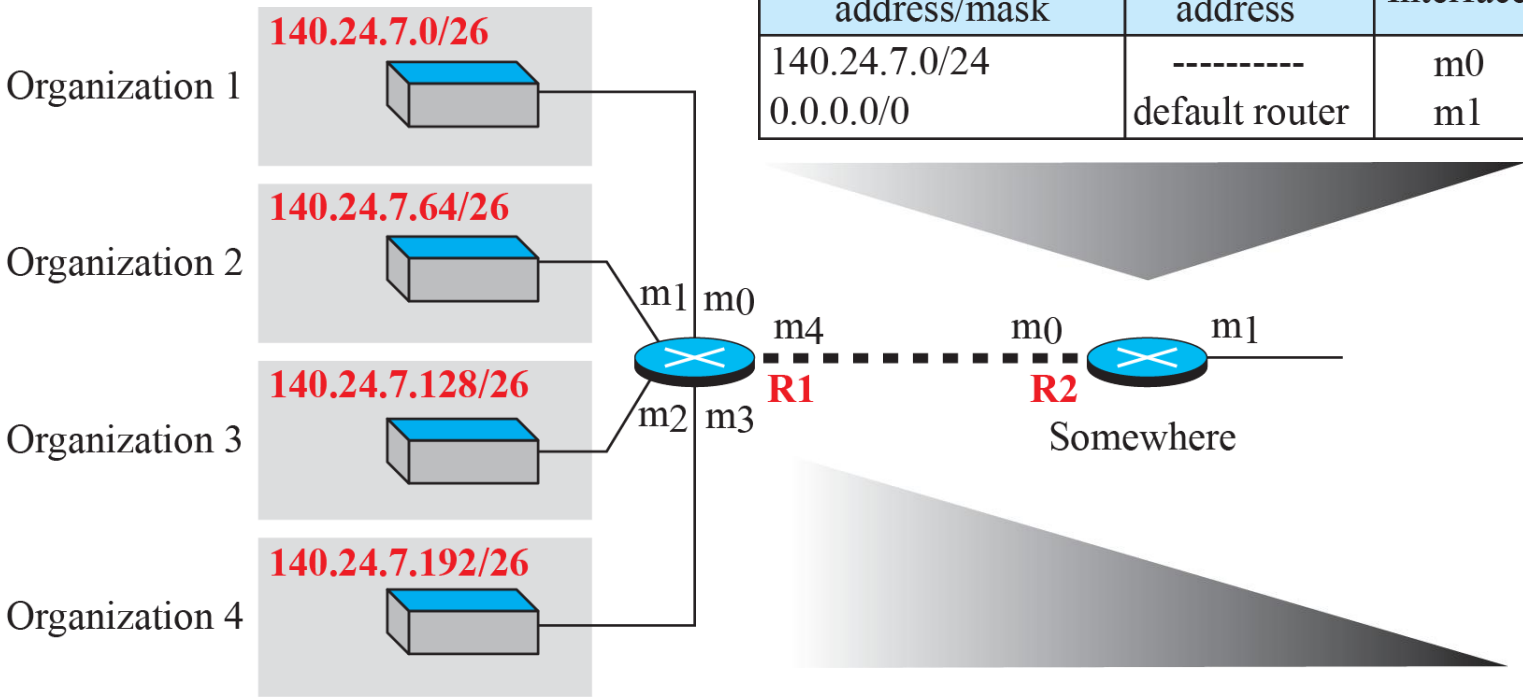
Show the forwarding process if a packet arrives at R1 in Figure 18.33 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

18. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.
2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are extracted for forwarding the packet (see Chapter 5).

Figure 18.34: Address aggregation



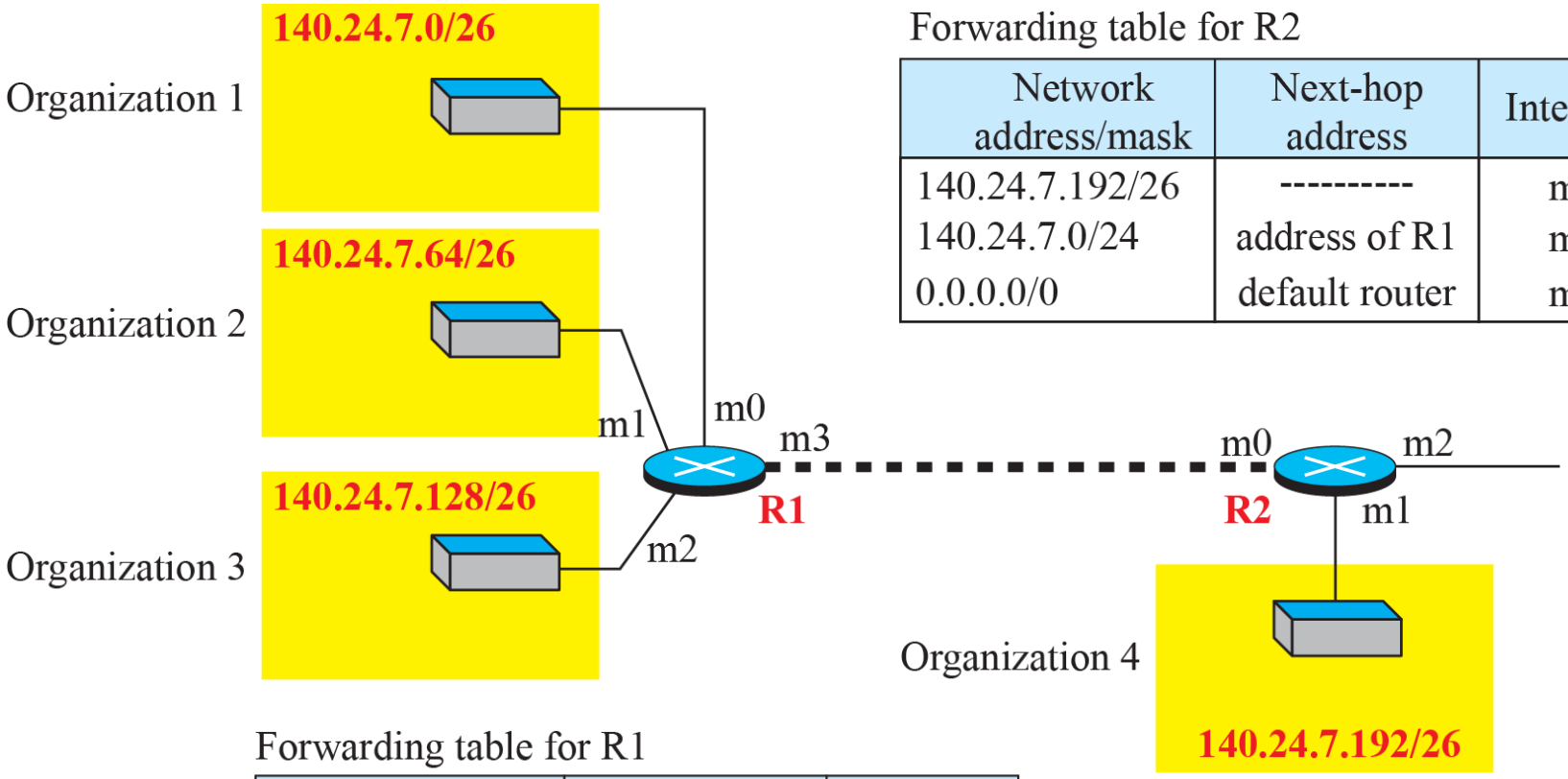
Forwarding table for R2

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

Forwarding table for R1

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

Figure 18.35: Longest mask matching



Forwarding table for R2

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

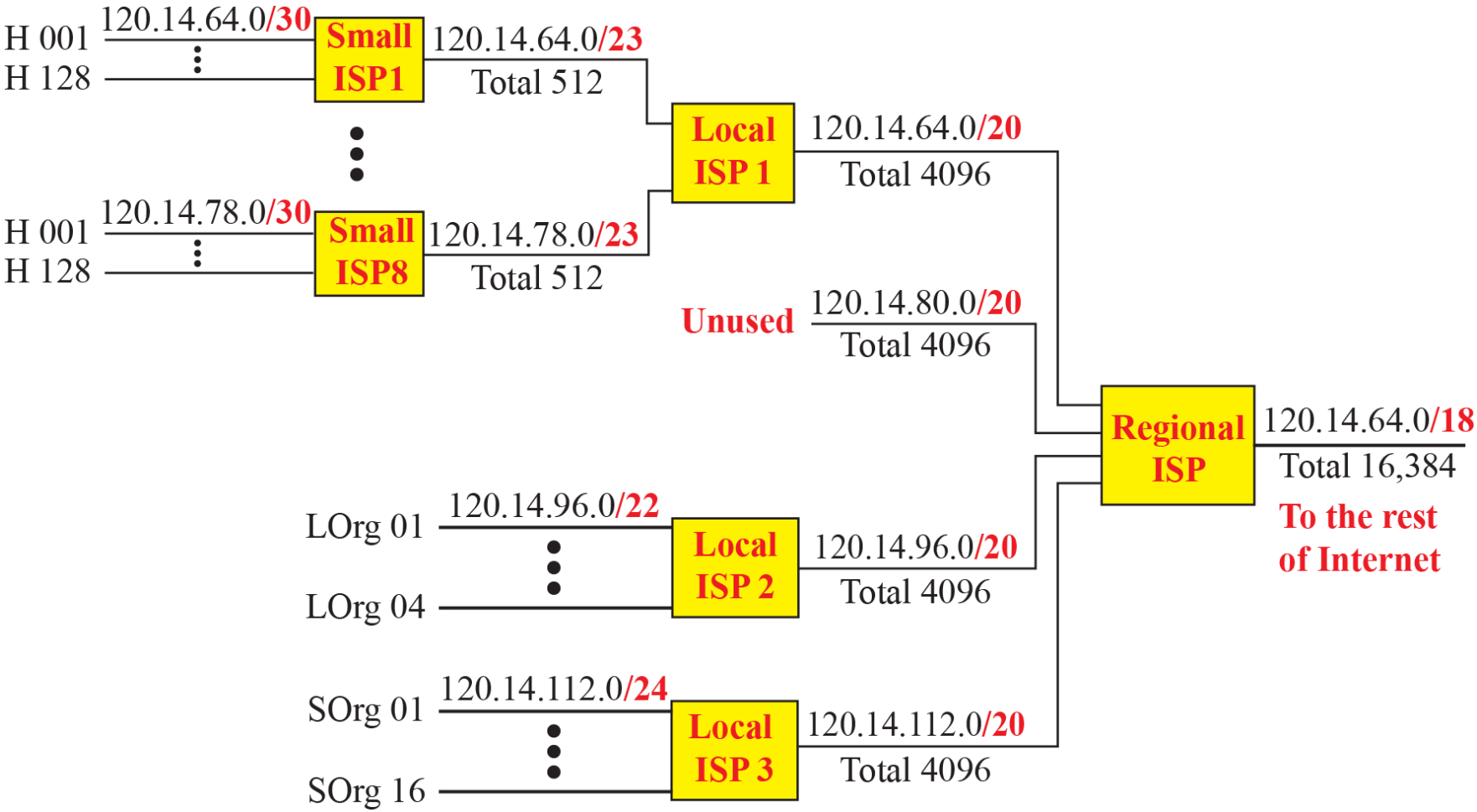
Forwarding table for R1

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

As an example of hierarchical routing, let us consider Figure 18.36. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into 4 subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs, the second subblock is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.

The figure also shows how local and small ISPs have assigned addresses.

Figure 18.35: Hierarchical routing with ISPs



18.5.2 Forwarding Based on Label

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

Figure 18.37 shows a simple example of searching in a forwarding table using the longest mask algorithm. Although there are some more efficient algorithms today, the principle is the same.

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

Figure 18.37: Example 18.11: Forwarding based on destination address

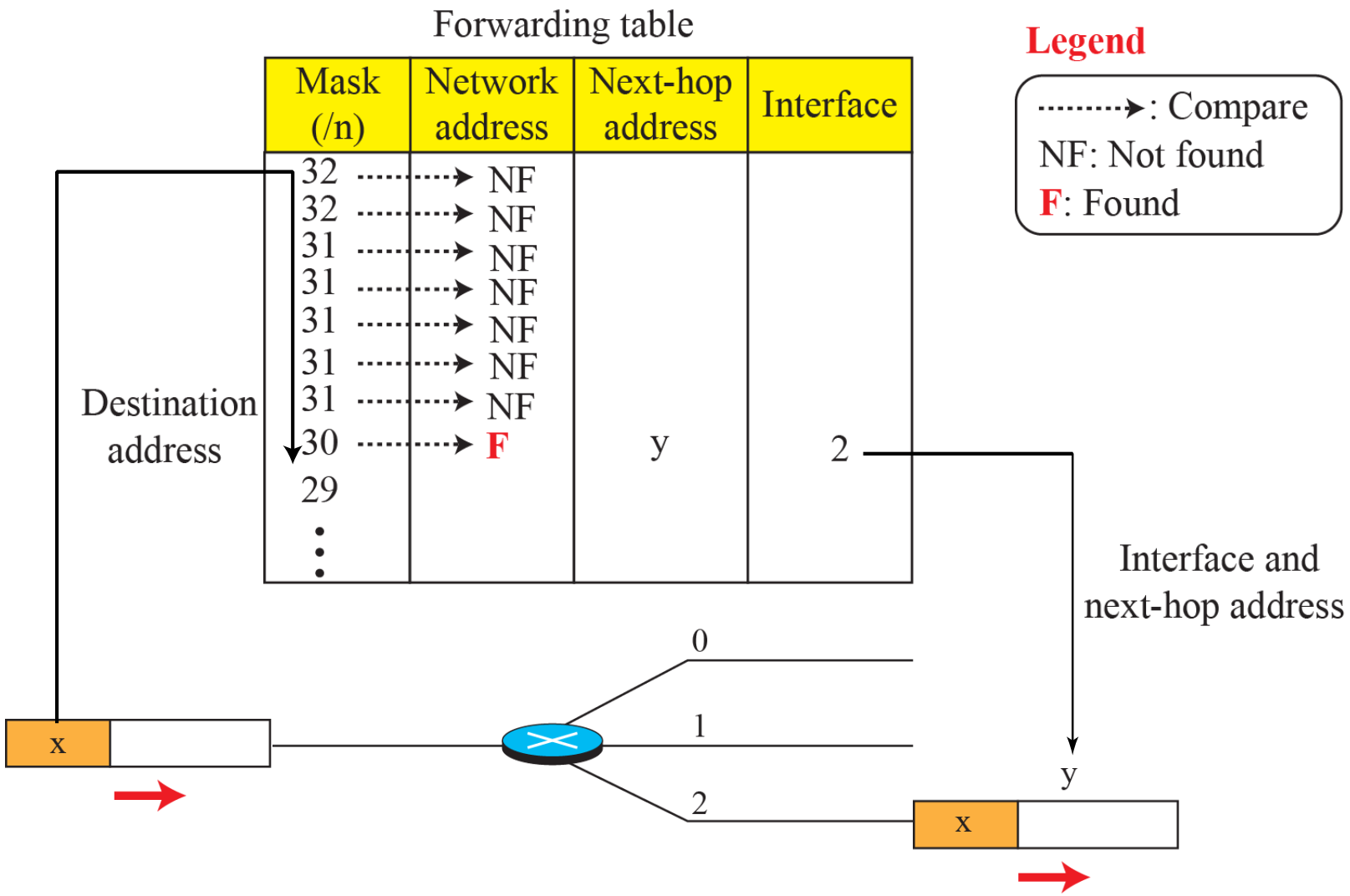


Figure 18.38 shows a simple example of using a label to access a switching table. Since the labels are used as the index to the table, finding the information in the table is immediate.

Figure 18.38: Example 18.12: Forwarding based on label

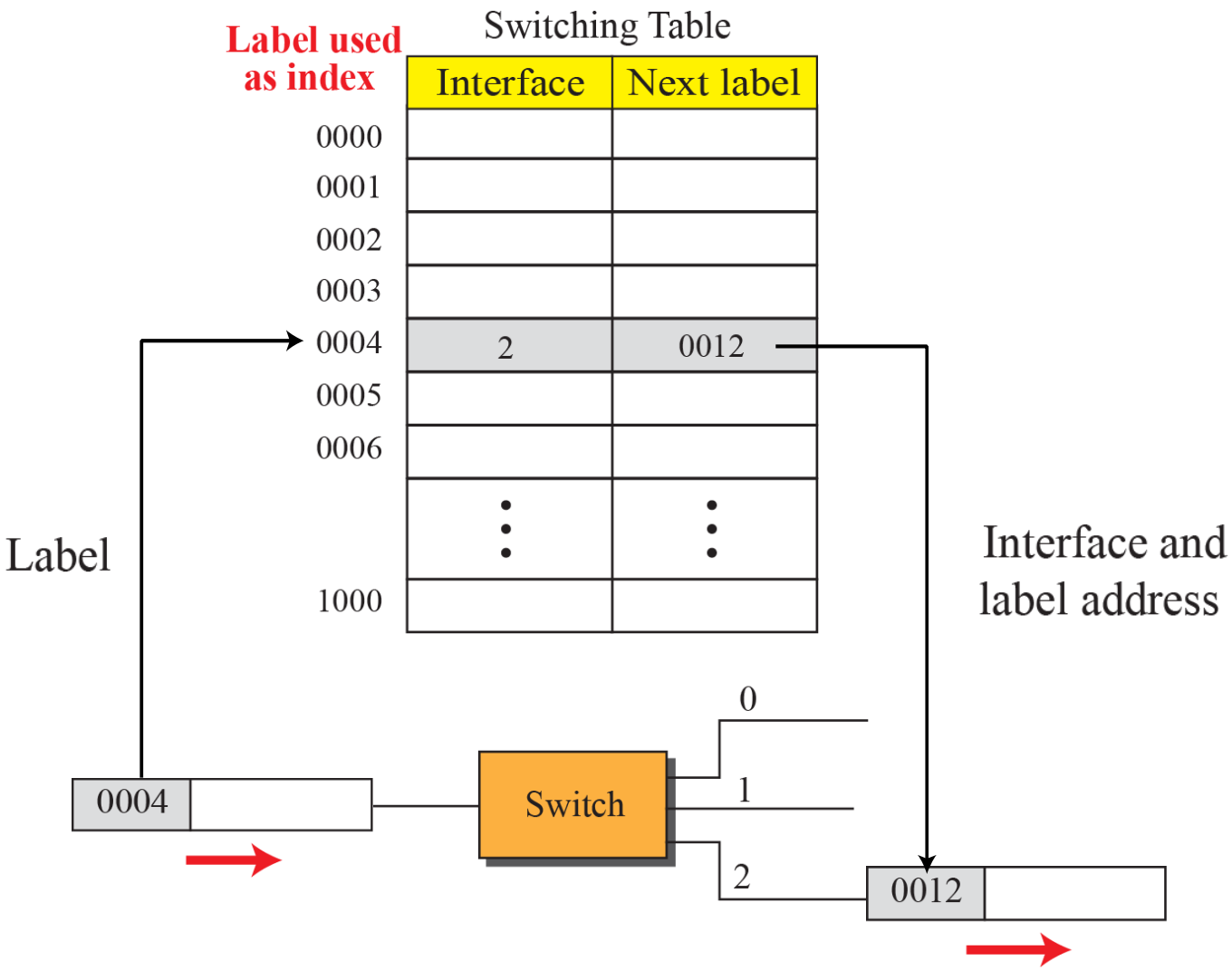


Figure 18.39: MPLS header added to an IP packet

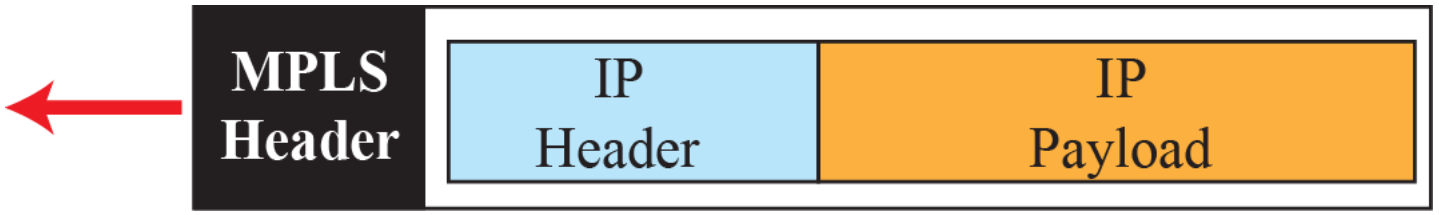
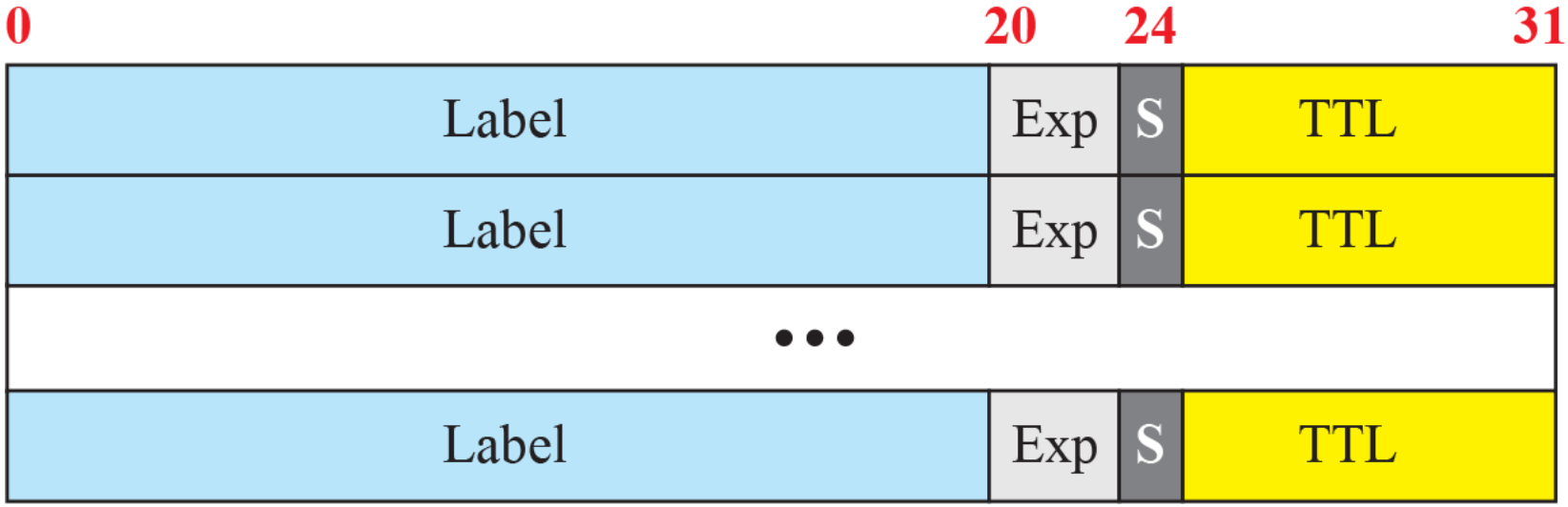


Figure 18.40: MPLS header made of a stack of labels



18.5.3 Routers as Packet Switches

As we may have guessed by now, the packet switches that are used in the network layer are called routers. Routers can be configured to act as either a datagram switch or a virtual-circuit switch. We have discussed the structure of a packet-switch in Chapter 8. The discussion in that chapter can be applied to any router used in the Internet.