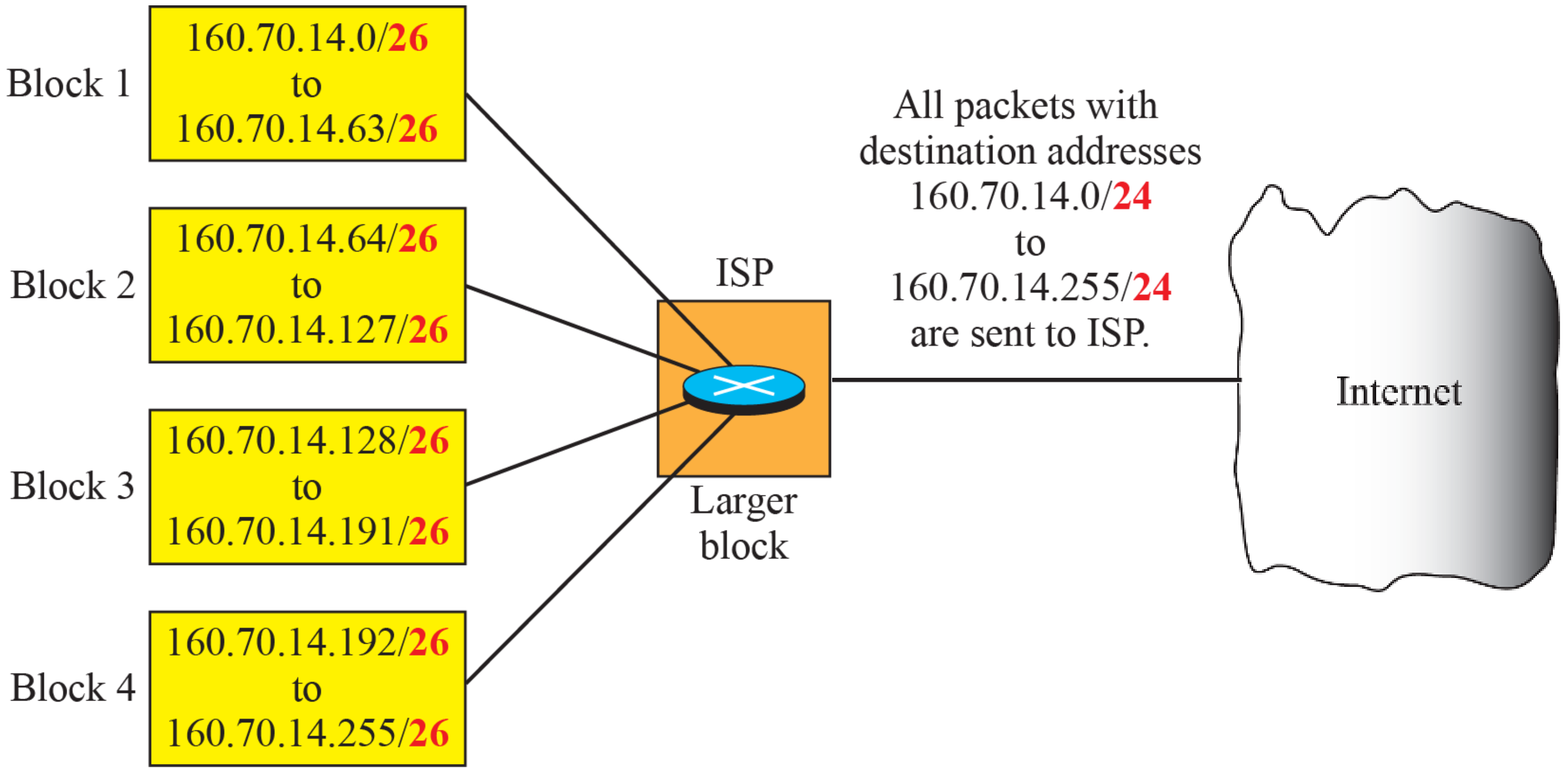
Example 4.6

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



Special Addresses

- In classful addressing some addresses were reserved for special purposes. The classless addressing scheme inherits some of these special addresses from classful addressing
- Special block

All-Zero Address(0.0.0.0/32)

All-One Address(255.255.255.255/32)

Loopback Address(127.0.0.0/8)

Private Address(10.0.0.0/8,172.16.0.0/12, 192.168.0.0./16 and 169.254.0.0./16)

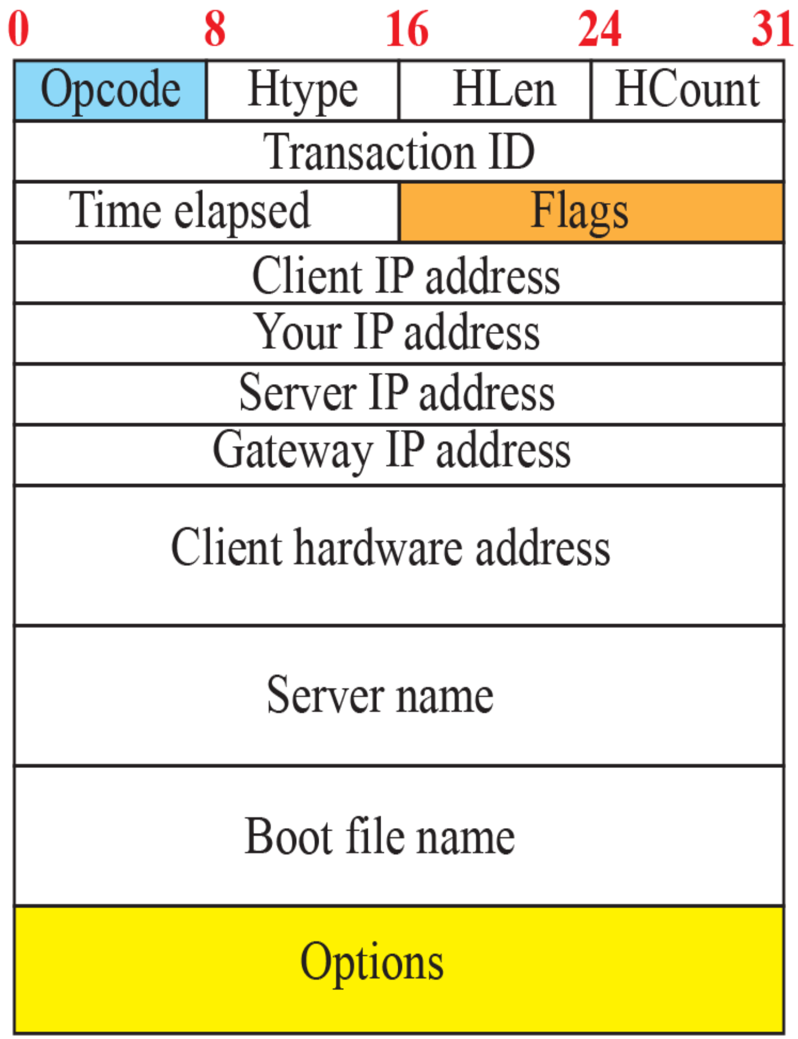
Multicast Address(224.0.0.0/24)

- Special address in each block

Network Address

Direct broadcast address

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

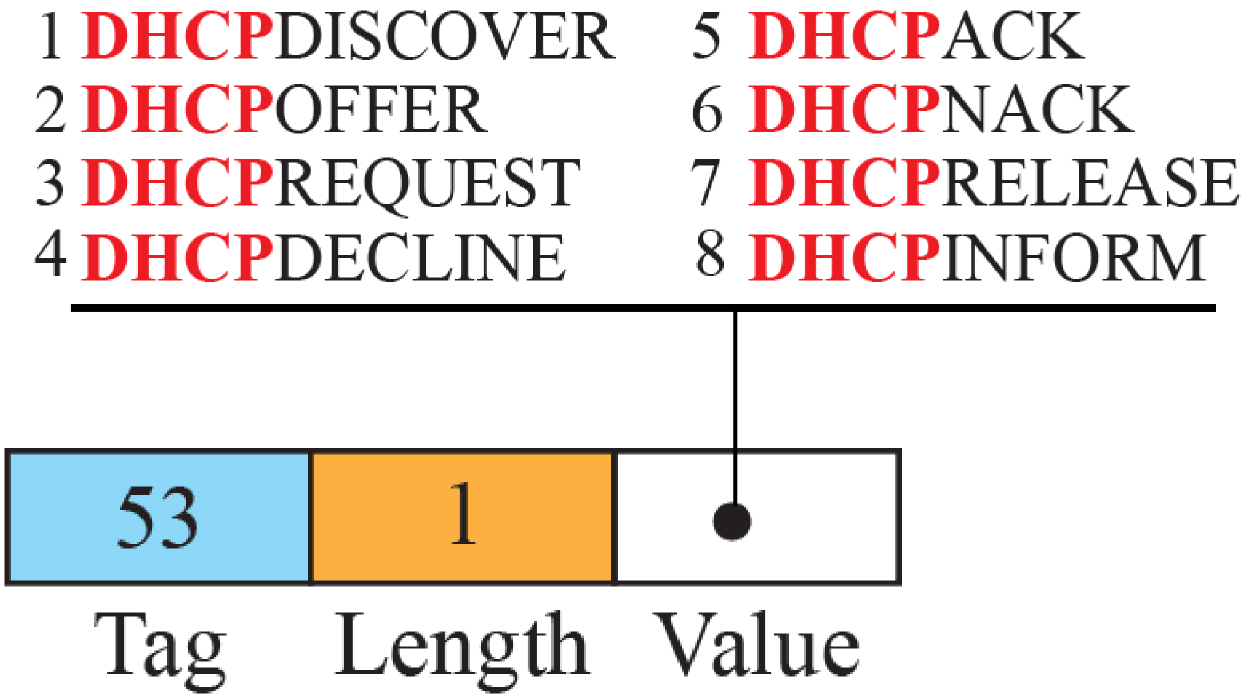
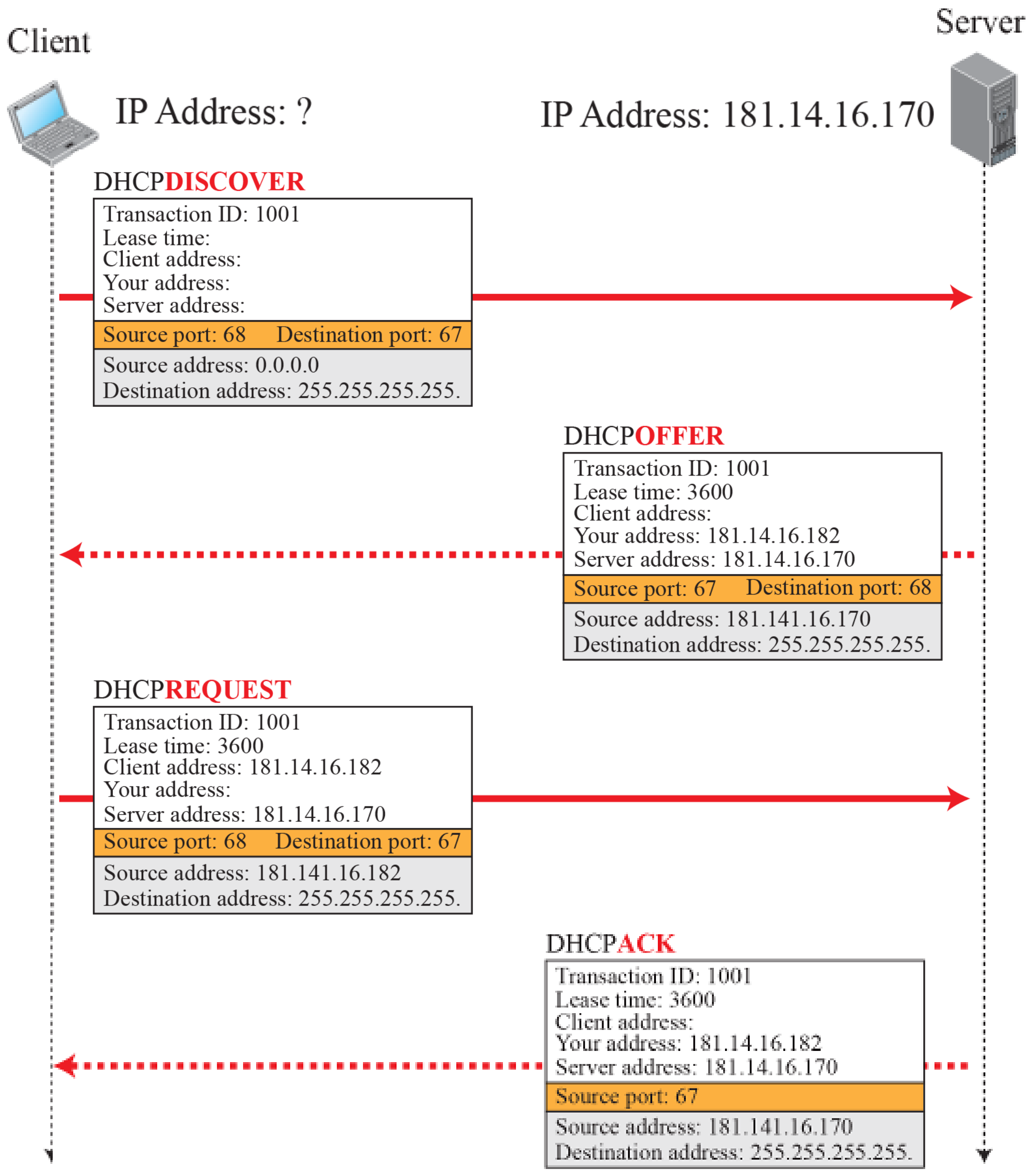


Figure 4.40: Operation of DHCP


Client
Server
IP Address: ?
IP Address: 181.14.16.170
DHCPDISCOVER
Transaction ID: 1001
Lease time:
Client address:
Your address:
Server address:
Source port: 68 Destination port: 67
Source address: 0.0.0.0
Destination address: 255.255.255.255.
DHCPOFFER
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.141.16.170
Destination address: 255.255.255.255.
DHCPREQUEST
Transaction ID: 1001
Lease time: 3600
Client address: 181.14.16.182
Your address:
Server address: 181.14.16.170
Source port: 68 Destination port: 67
Source address: 181.141.16.182
Destination address: 255.255.255.255.
DHCPACK
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67
Source address: 181.141.16.170
Destination address: 255.255.255.255.

Legend

Application
UDP
IP

Note:
Only partial information is given.

Figure 4.41: FSM for the DHCP client

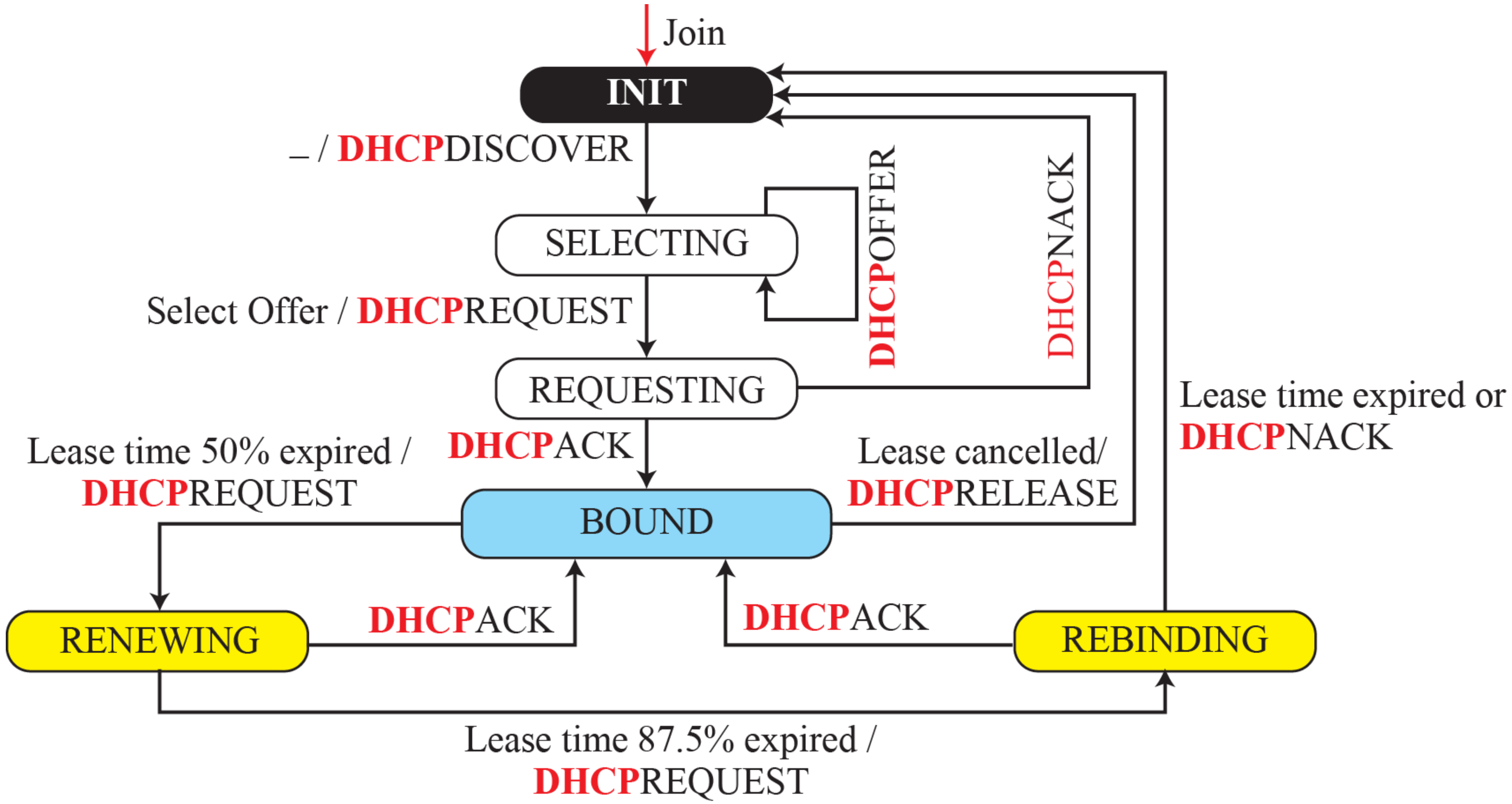

Join
INIT
– / DHCPDISCOVER
SELECTING
DHCPOFFER
DHCPNACK
Select Offer / DHCPREQUEST
REQUESTING
DHCPACK
Lease time 50% expired / DHCPREQUEST
Lease cancelled/ DHCPRELEASE
Lease time expired or DHCPNACK
BOUND
DHCPACK
DHCPACK
RENEWING
REBINDING
Lease time 87.5% expired / DHCPREQUEST

Figure 4.42: NAT

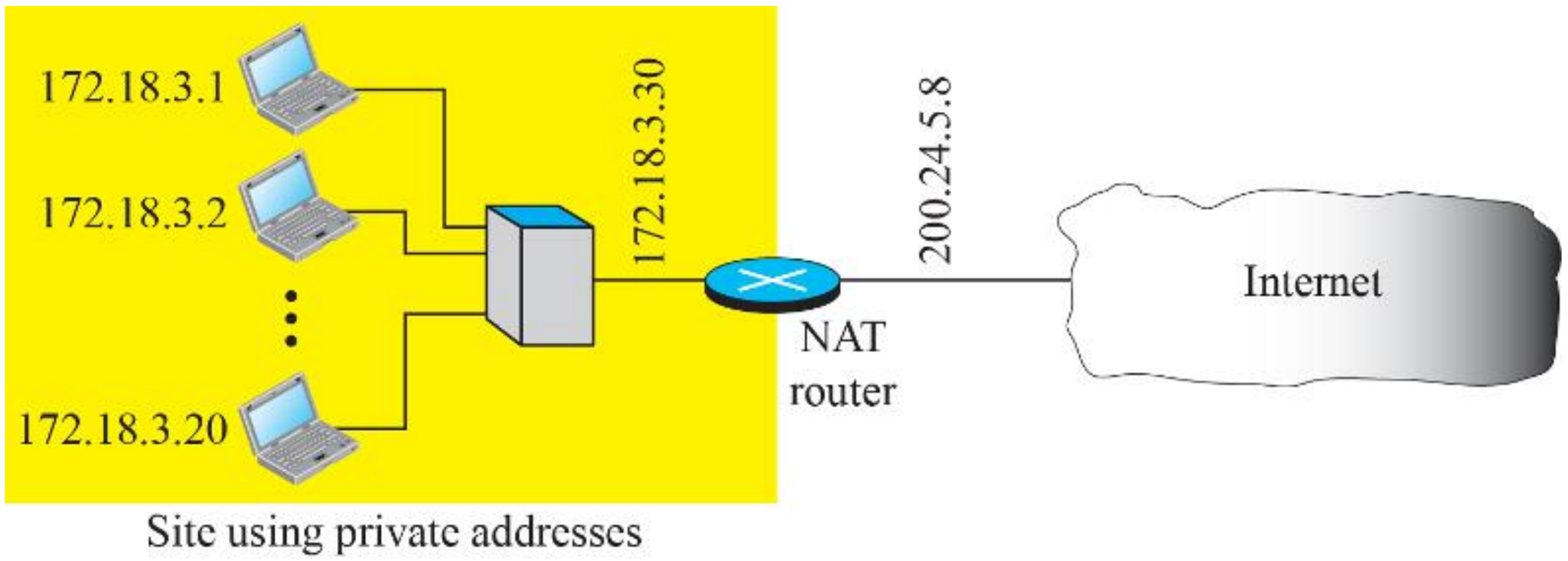


Figure 4.43: Address translation

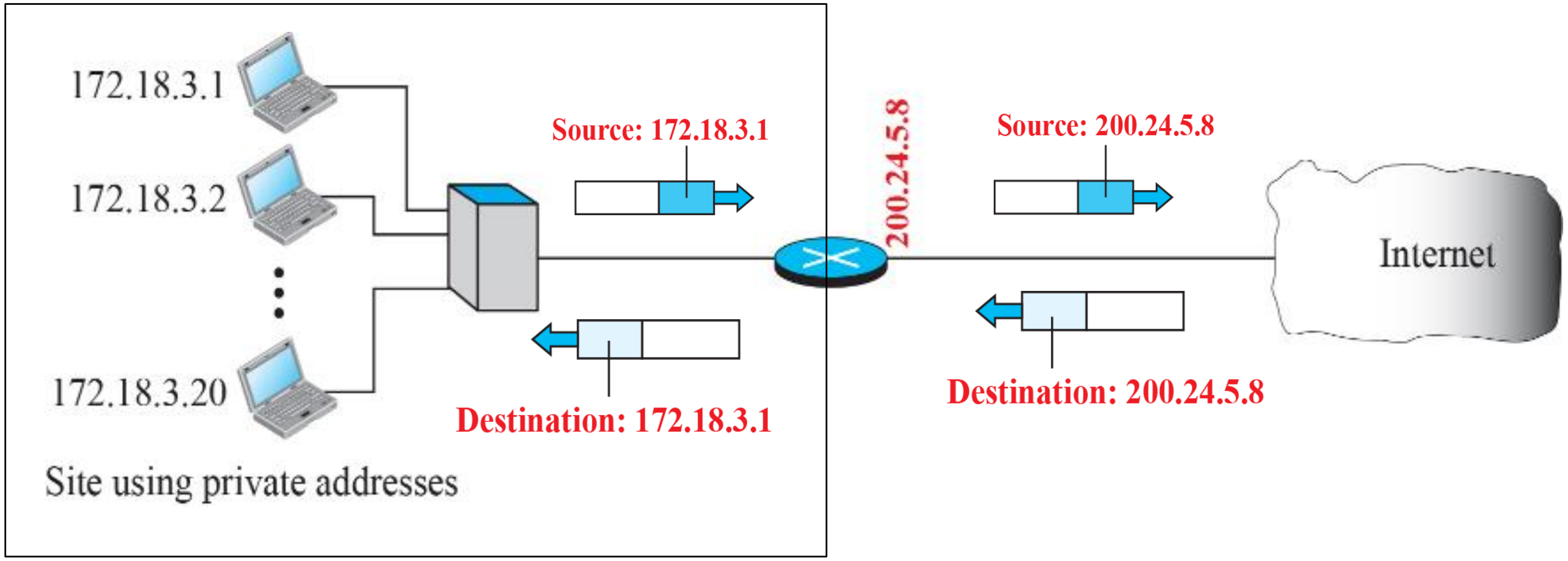


Figure 4.44: Translation

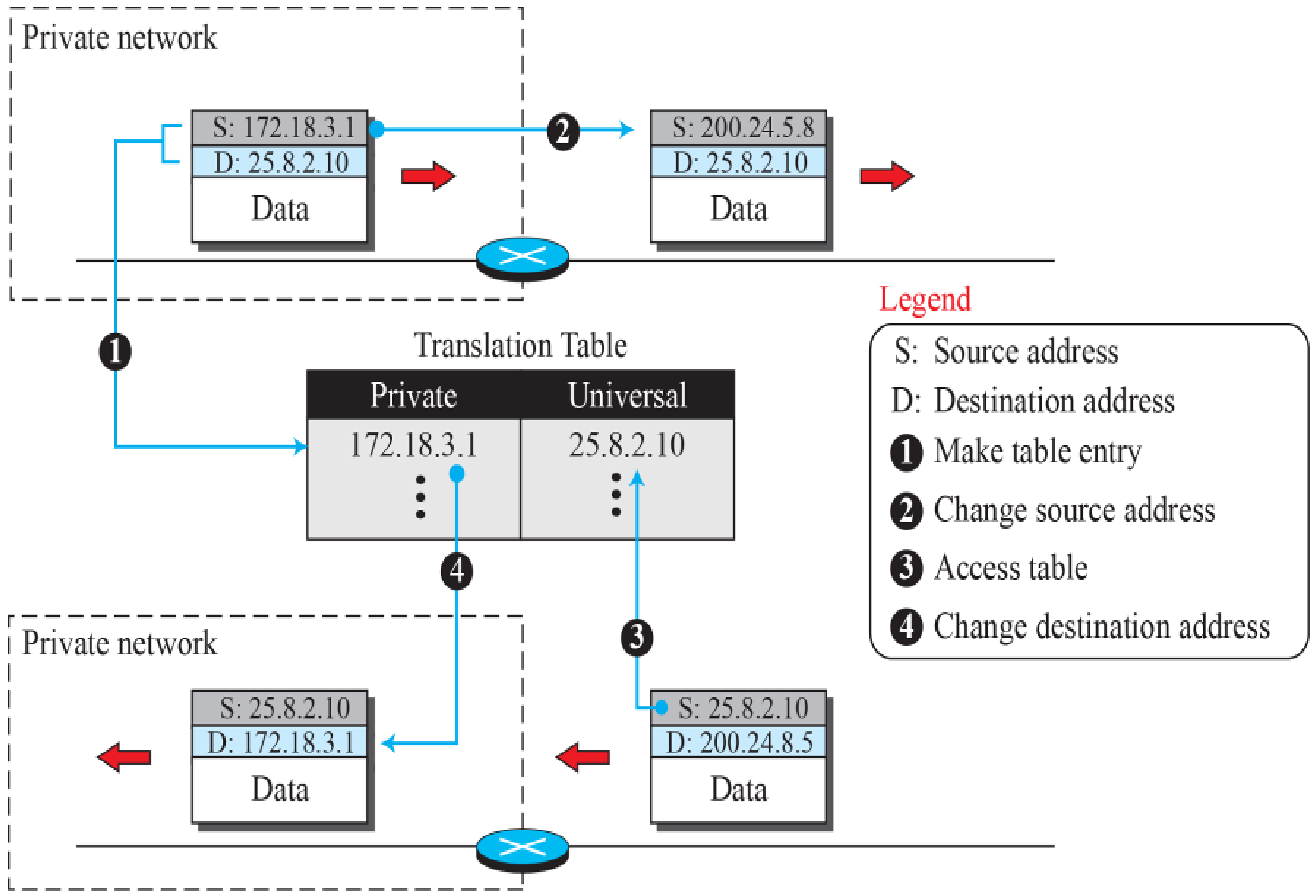


Table 4.1: Five-column translation table

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
⋮	⋮	⋮	⋮	⋮

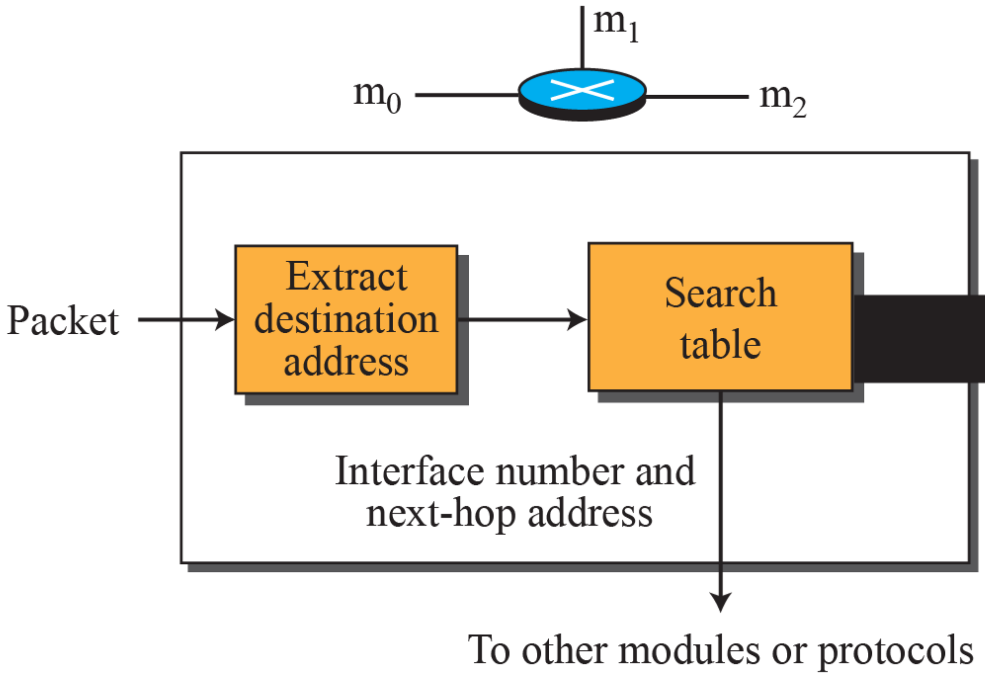
4.2.3 Forwarding of IP Packets

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. Since the Internet today is made of a combination of links (networks), forwarding means to deliver the packet to the next hop (which can be the final destination or the intermediate connecting device).

4.2.3 (continued)

- Forwarding Based On Destination Address
  - Address Aggregation
  - Longest Mask Matching
  - Hierarchical Routing
  - Geographical Routing
  - Forwarding Table Search Algorithms
- Forwarding Based on Label
  - Multi-Protocol Label Switching (MPLS)
  - A New Header
  - Hierarchical Routing

Figure 4.45: Simplified forwarding module in classless address

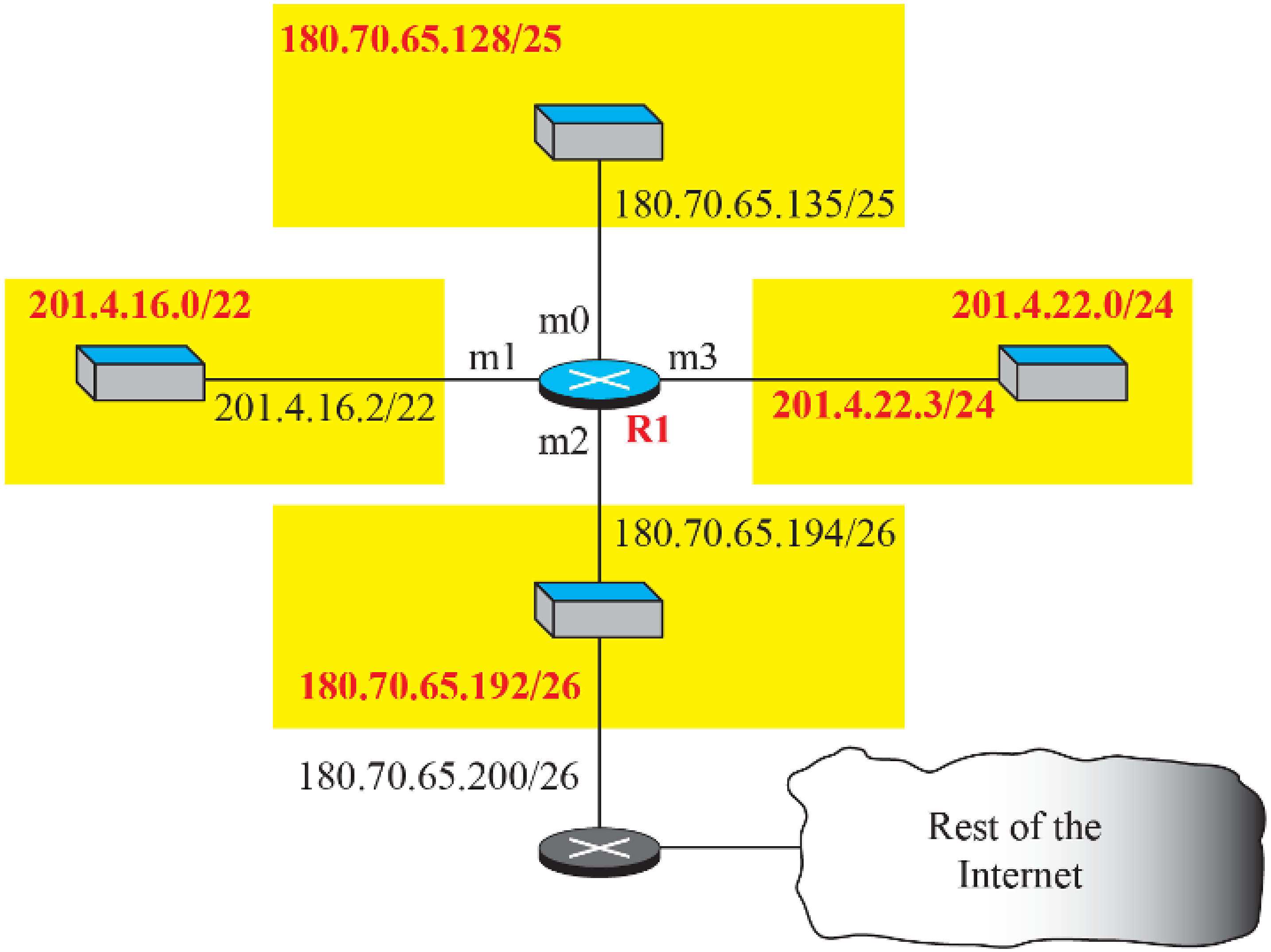


m_1
m_0
m_2
Packet
Extract destination address
Search table
Interface number and next-hop address
To other modules or protocols

Forwarding table

Network address including mask	Next-hop IP address	Interface
$x_0.y_0.z_0.t_0/n_0$	....	m_0
$x_1.y_1.z_1.t_1/n_1$	....	m_1
$x_2.y_2.z_2.t_2/n_1$	....	m_2

Figure 4.46: Configuration for Example 4.7



Example 4.7

Make a forwarding table for router R1 using the configuration in Figure 4.46.

Solution

Table 4.2 shows the corresponding table.

Table 4.2: 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

Example 4.8

Instead of Table 4.2, we can use Table 4.3, in which the network address/mask is given in bits.

Table 4.3: 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.

Example 4.9

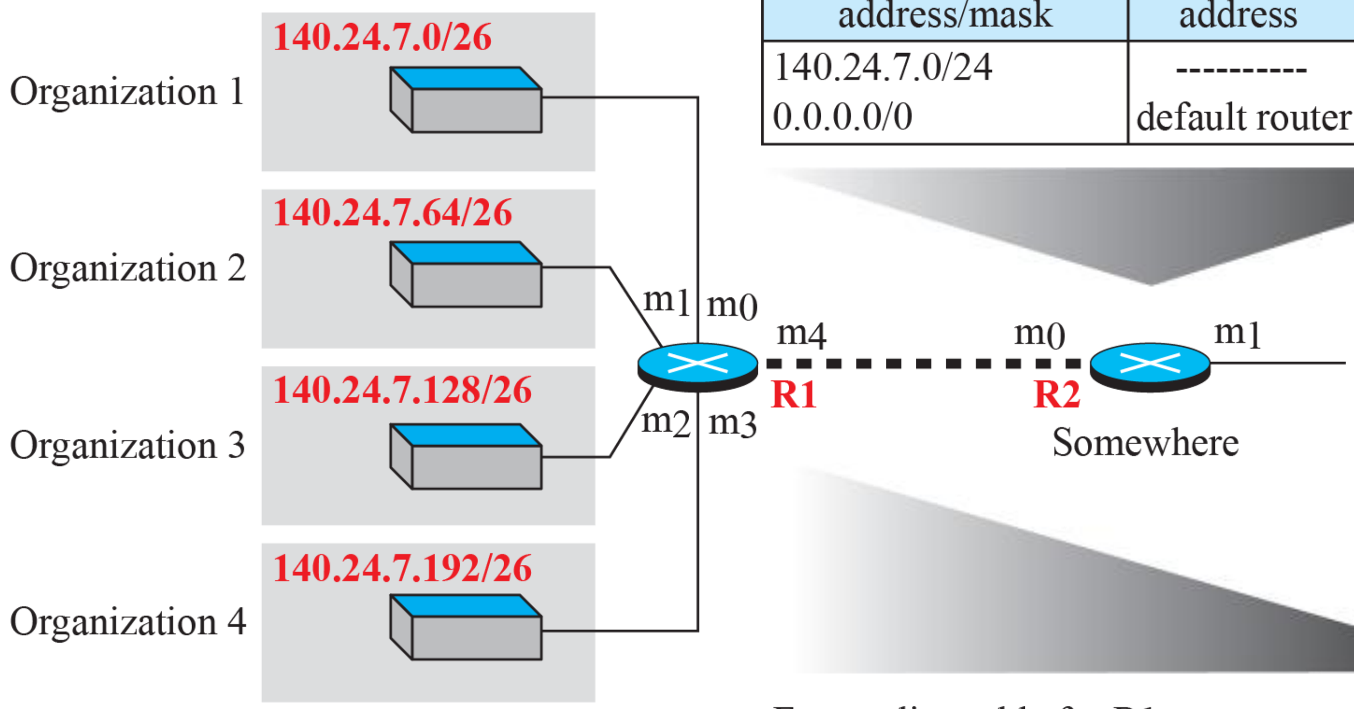
Show the forwarding process if a packet arrives at R1 in Figure 4.46 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

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

Figure 4.47: Address aggregation



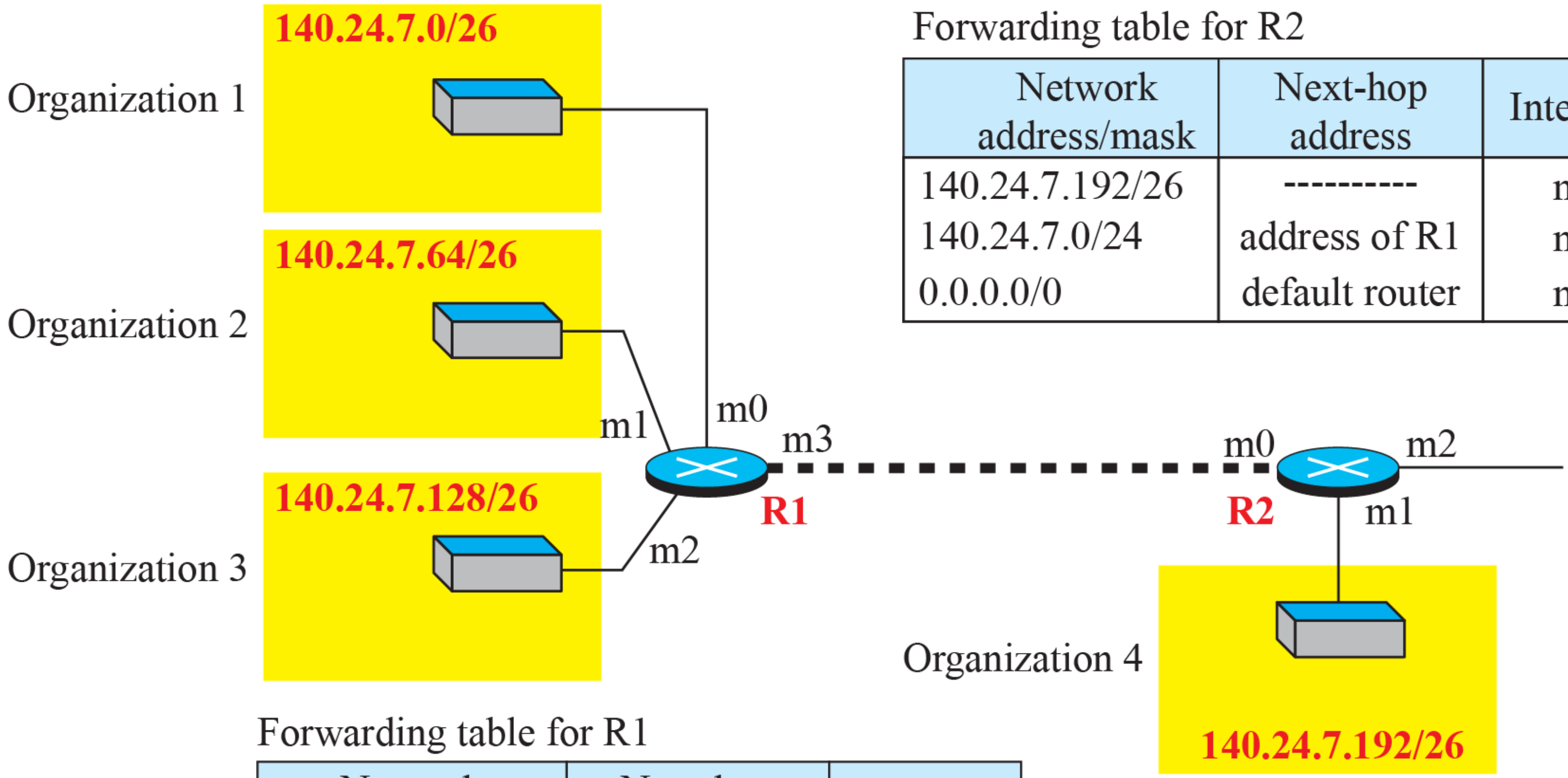
Forwarding table for R2

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 4.48: 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

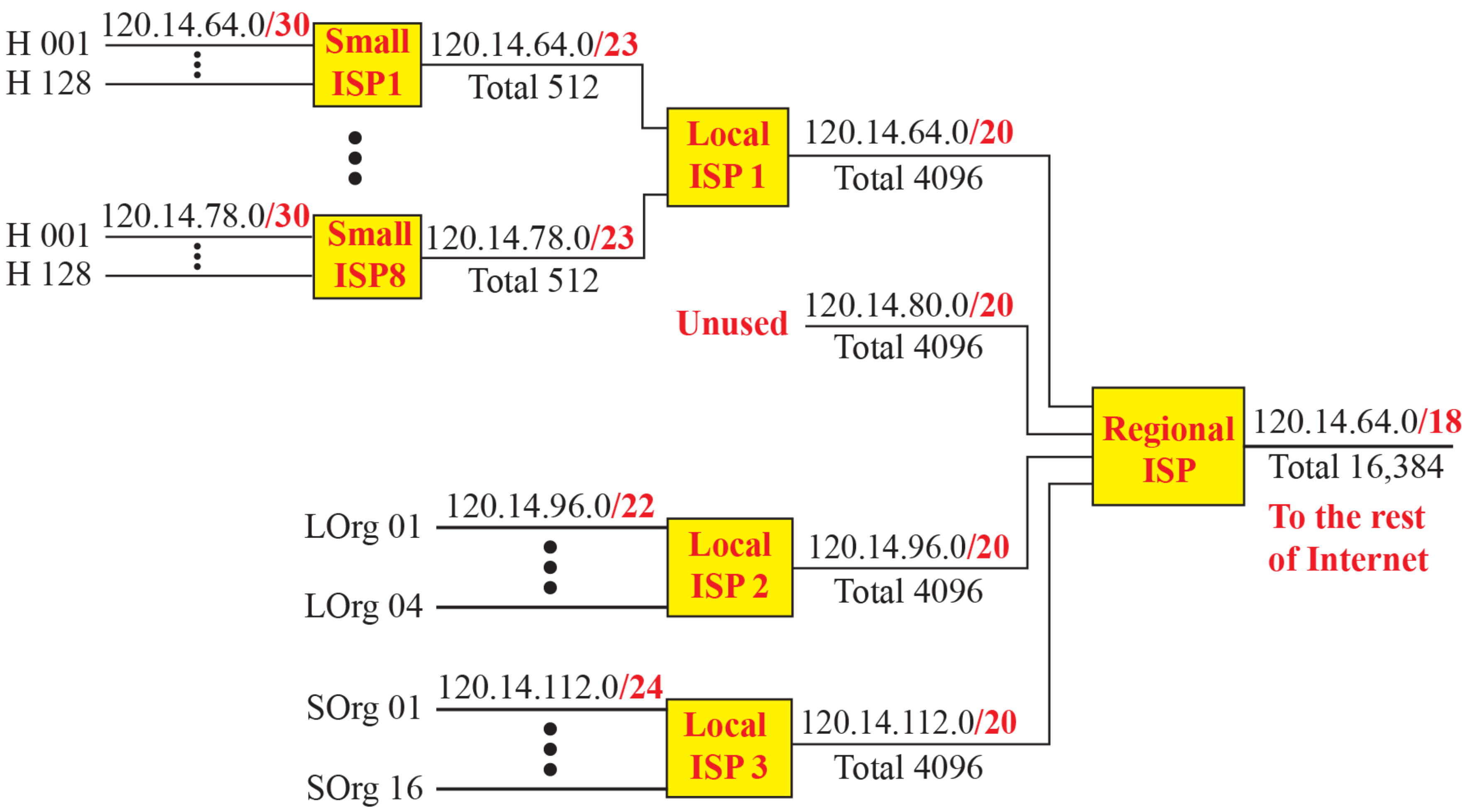
Network address/mask	Next-hop address	Interface
140.24.7.0/26	----------	m0
140.24.7.64/26	----------	m1
140.24.7.128/26	----------	m2
0.0.0.0/0	default router	m3

Example 4.10

As an example of hierarchical routing, let us consider Figure 4.49. 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 4.49: Hierarchical routing with ISPs

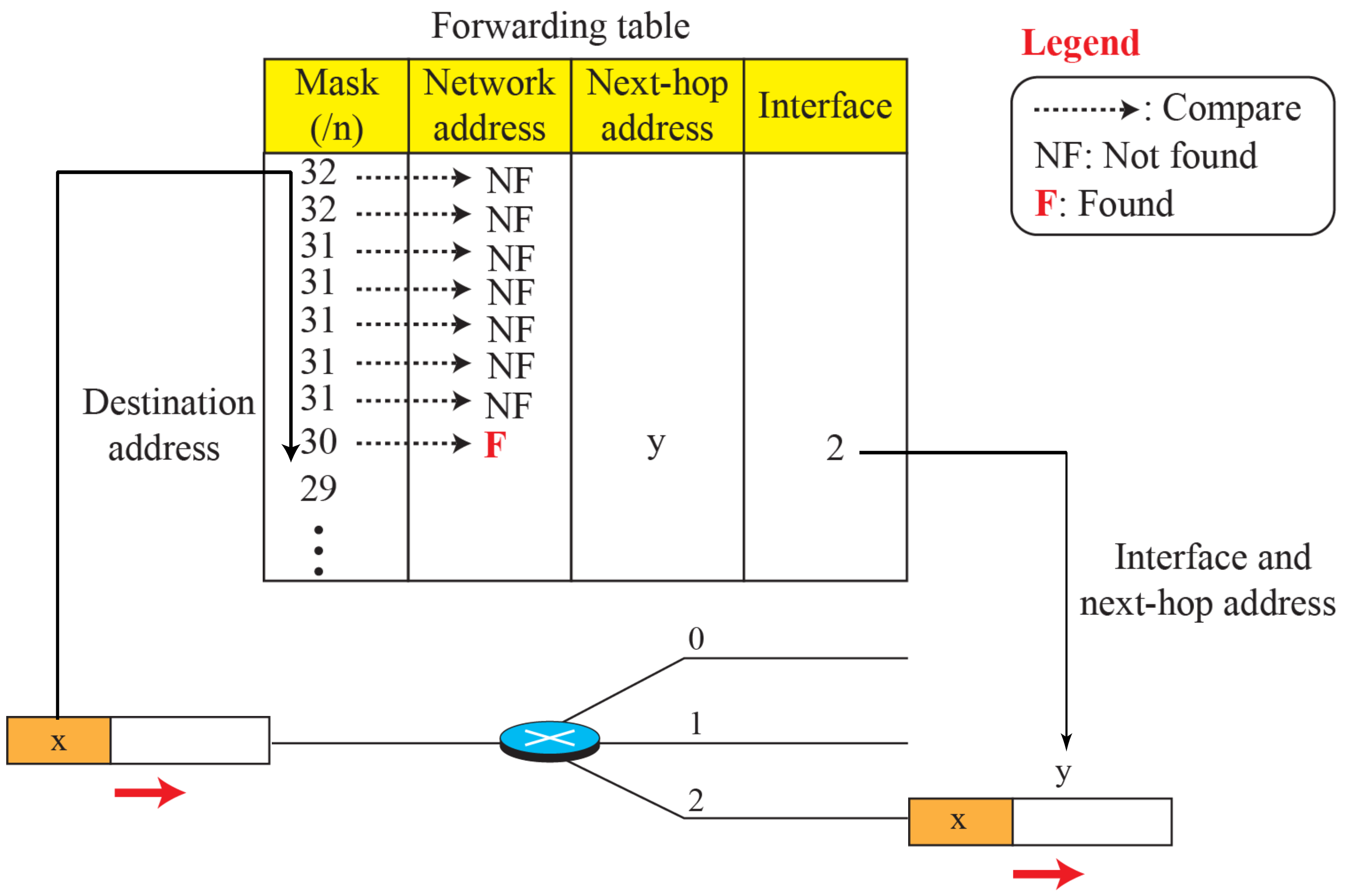


Example 4.11

Figure 4.50 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 4.50: Example 4.11: Forwarding based on destination address



Example 4.12

Figure 4.51 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 4.51: Example 4.12: Forwarding based on label

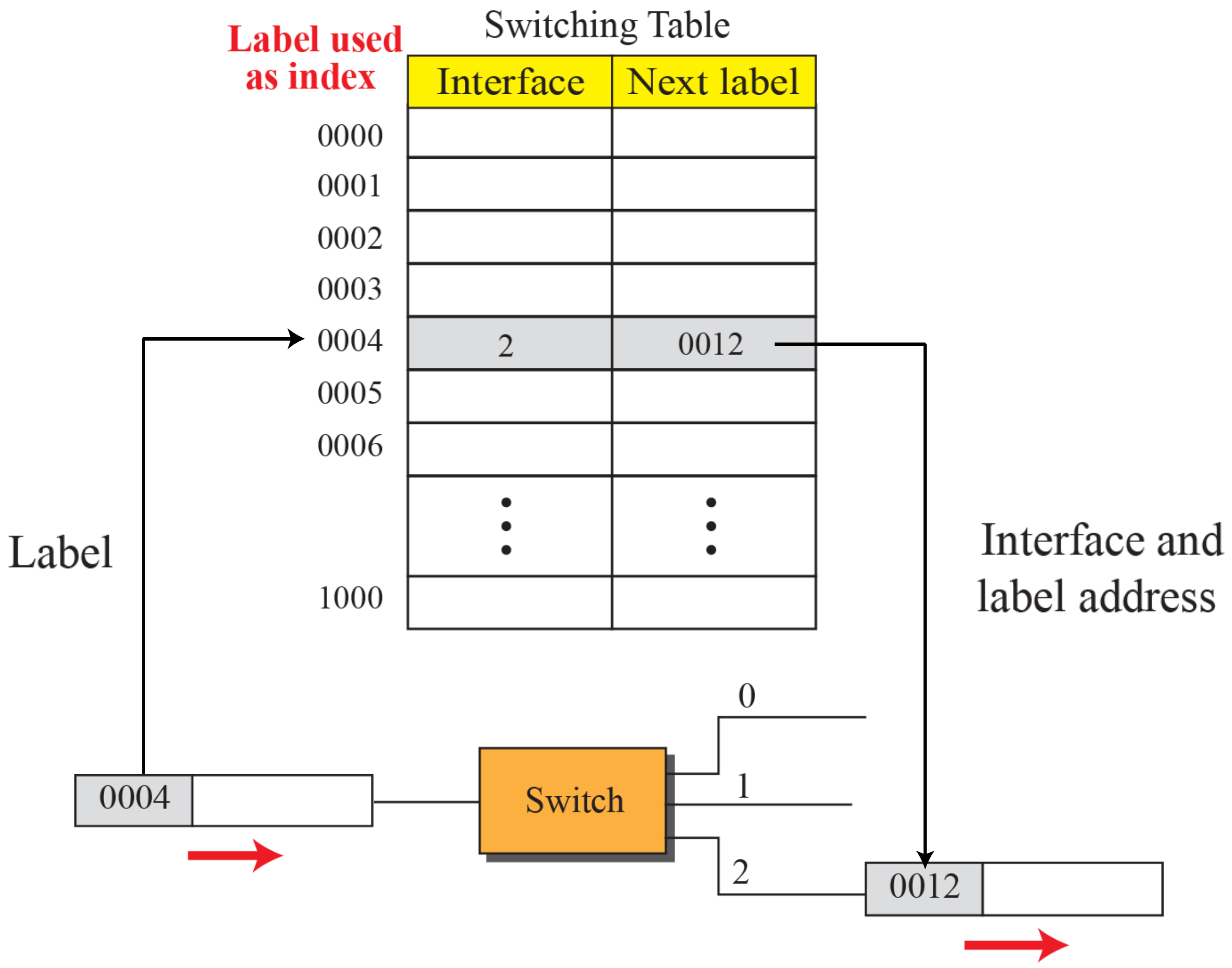


Figure 4.52: MPLS header added to an IP packet

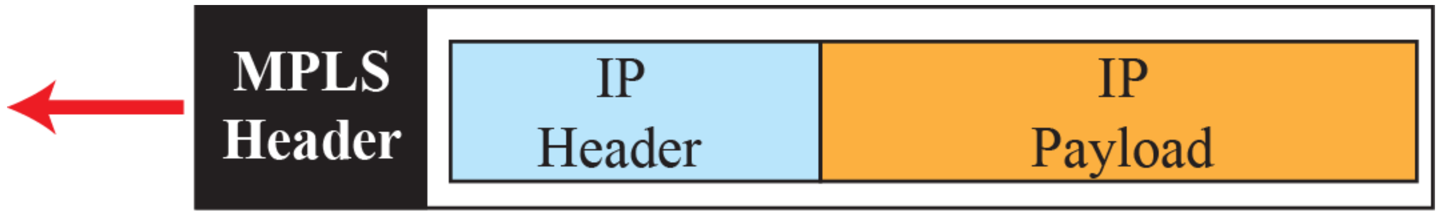
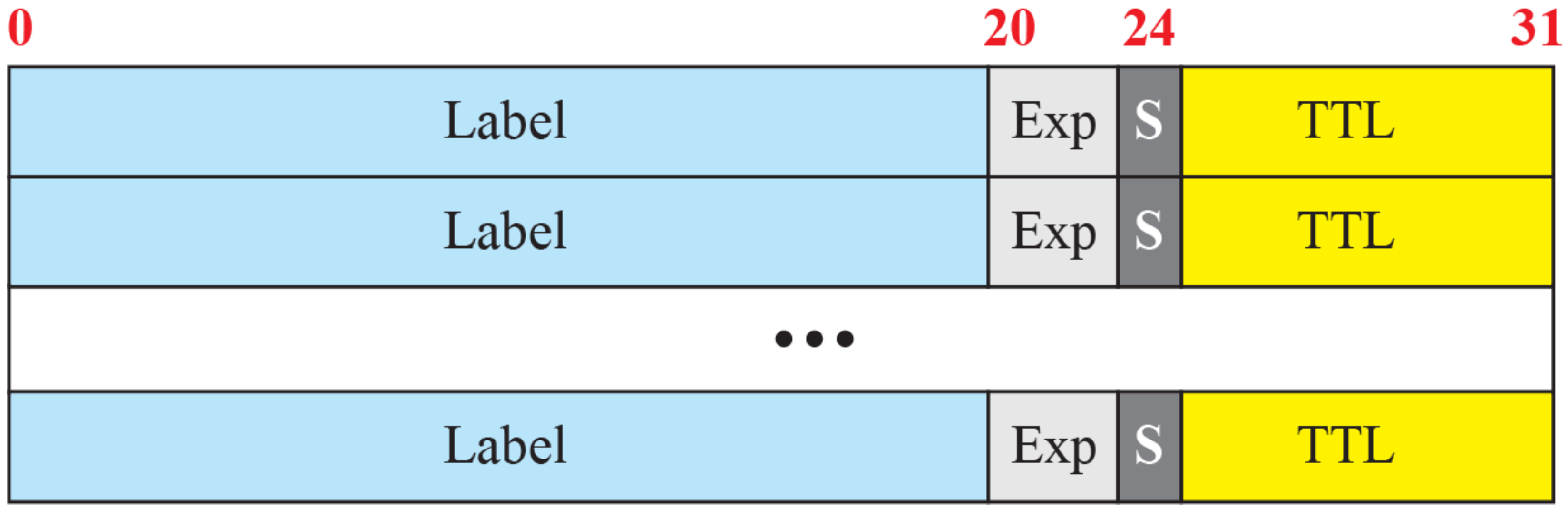


Figure 4.53: MPLS header made of a stack of labels



4.2.4 ICMPv4

The IPv4 has no error-reporting or error-correcting mechanism. What happens if something goes wrong? There are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

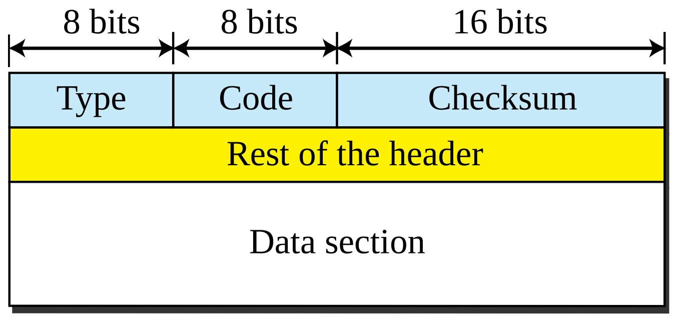
The IP protocol also lacks a mechanism for host and management queries.

The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above two deficiencies.

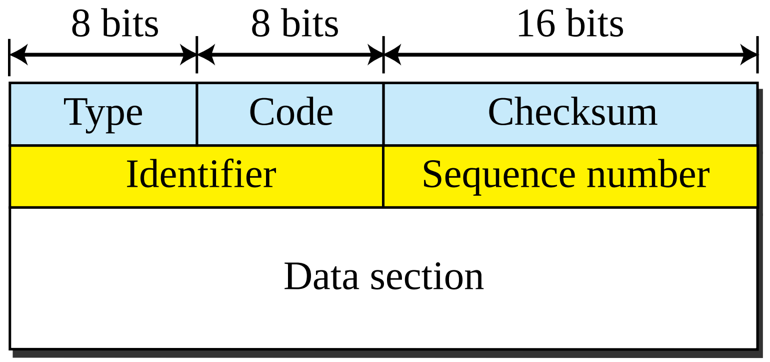
4.2.4 (continued)

- Messages
  - Message Format
  - Error Reporting Messages
  - Query Messages

Figure 4.54: General format of ICMP messages



Error-reporting messages



Query messages

Type and code values

Error-reporting messages

03: Destination unreachable (codes 0 to 15)
04: Source quench (only code 0)
05: Redirection (codes 0 to 3)
11: Time exceeded (codes 0 and 1)
12: Parameter problem (codes 0 and 1)

Query messages

08 and 00: Echo request and reply (only code 0)
13 and 14: Timestamp request and reply (only code 0)

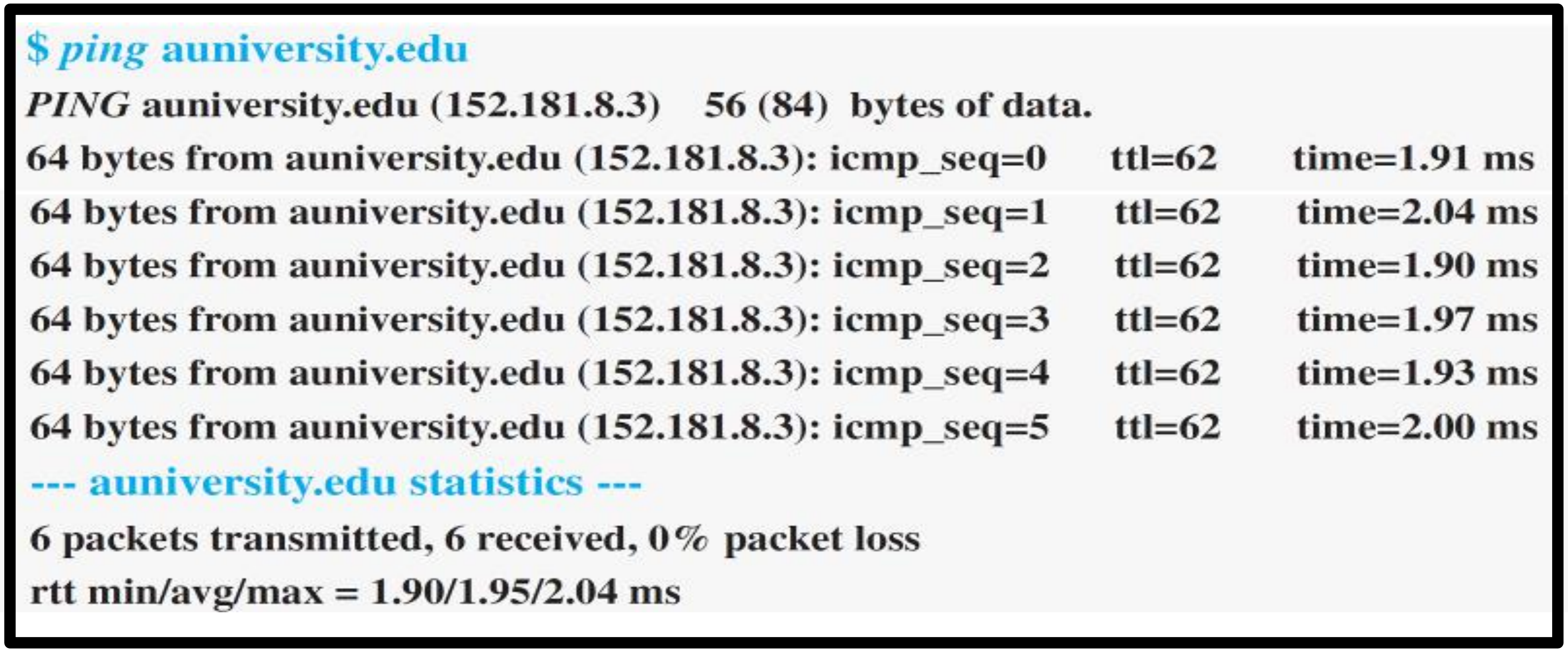
Note: See the book website for more explanation about the code values.

Example 4.13

One of the tools that a host can use to test the liveliness of another host is the ping program. The ping program takes advantage of the ICMP echo request and echo reply messages. A host can send an echo request (type 8, code 0) message to another host, which, if alive, can send back an echo reply (type 0, code 0) message. To some extent, the ping program can also measure the reliability and congestion of the router between the two hosts by sending a set of request-reply messages.

Example 4.13 (continued)

The following shows how we send a ping message to the auniversity.edu site.



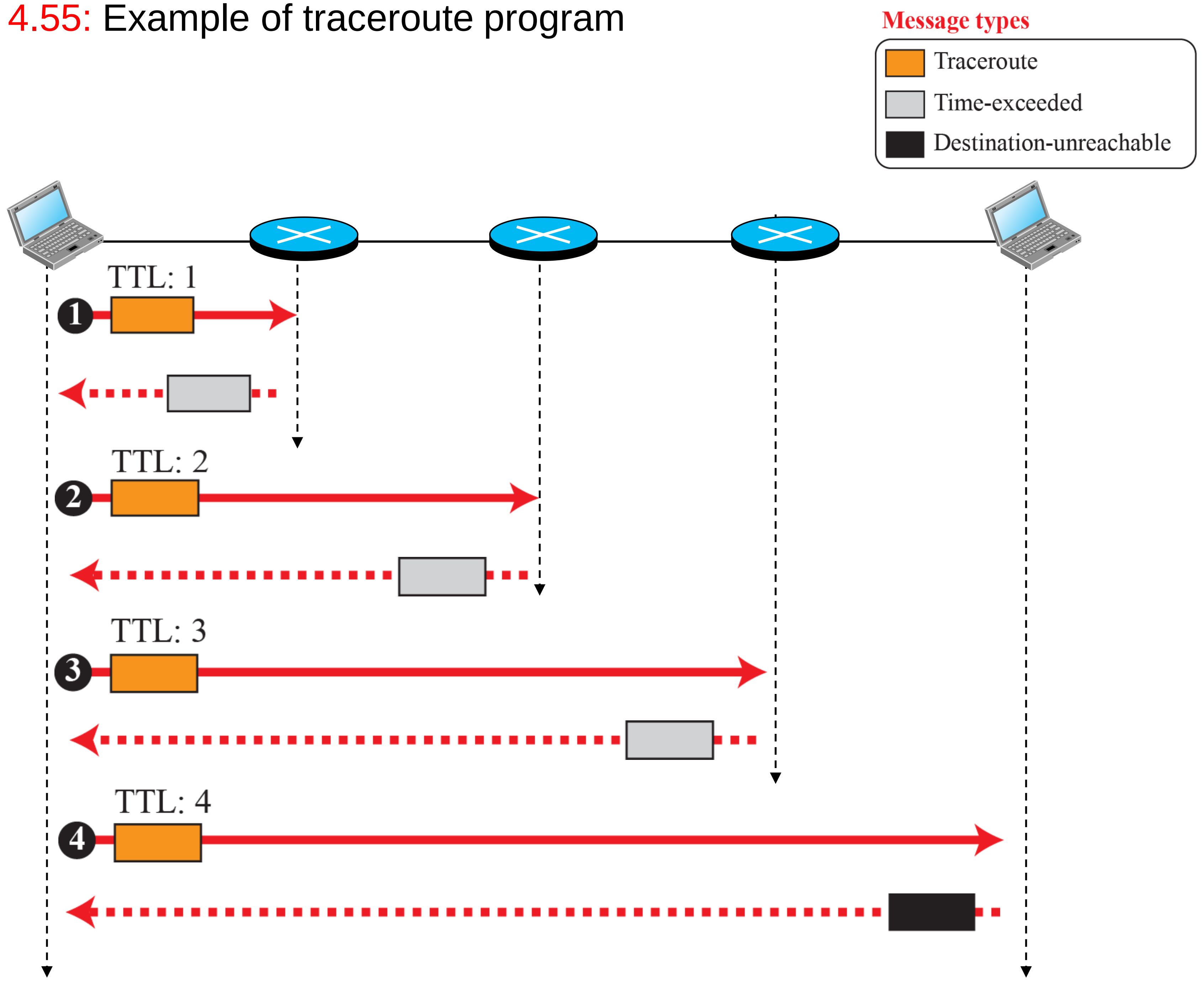
```
$ ping auniversity.edu
PING auniversity.edu (152.181.8.3)   56 (84)  bytes of data.
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=0    ttl=62     time=1.91 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=1    ttl=62     time=2.04 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=2    ttl=62     time=1.90 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=3    ttl=62     time=1.97 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=4    ttl=62     time=1.93 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=5    ttl=62     time=2.00 ms
--- auniversity.edu statistics ---
6 packets transmitted, 6 received, 0% packet loss
rtt min/avg/max = 1.90/1.95/2.04 ms
```

Example 4.14

The traceroute program in UNIX or tracert in Windows can be used to trace the path of a packet from a source to the destination. It can find the IP addresses of all the routers that are visited along the path. The program is usually set to check for the maximum of 30 hops (routers) to be visited. The number of hops in the Internet is normally less than this. The traceroute program is different from the ping program. The ping program gets help from two query messages; the traceroute program gets help from two error-reporting messages: time-exceeded and destination-unreachable

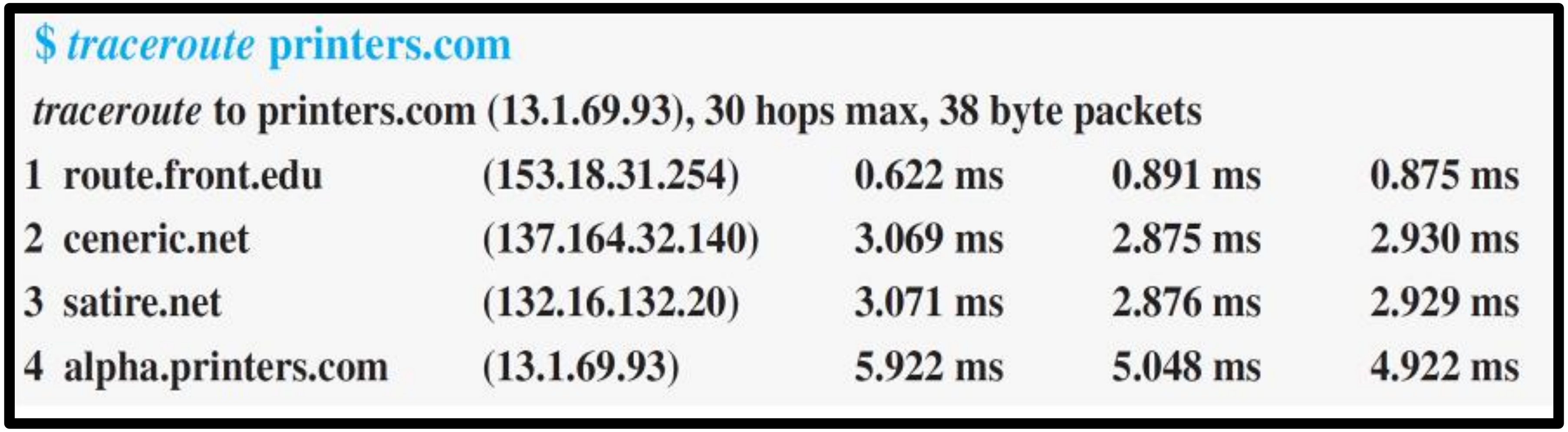
Figure 4.55 shows an example in which n = 3.

Figure 4.55: Example of traceroute program



Example 4.14 (continued)

The traceroute program also sets a timer to find the round-trip time for each router and the destination. Most traceroute programs send three messages to each device, with the same TTL value, to be able to find a better estimate for the round-trip time. The following shows an example of a traceroute program, which uses three probes for each device and gets three RTTs.



```
$ traceroute printers.com
traceroute to printers.com (13.1.69.93), 30 hops max, 38 byte packets
1  route.front.edu       (153.18.31.254)    0.622 ms    0.891 ms    0.875 ms
2  ceneric.net           (137.164.32.140)   3.069 ms    2.875 ms    2.930 ms
3  satire.net            (132.16.132.20)    3.071 ms    2.876 ms    2.929 ms
4  alpha.printers.com    (13.1.69.93)       5.922 ms    5.048 ms    4.922 ms
```