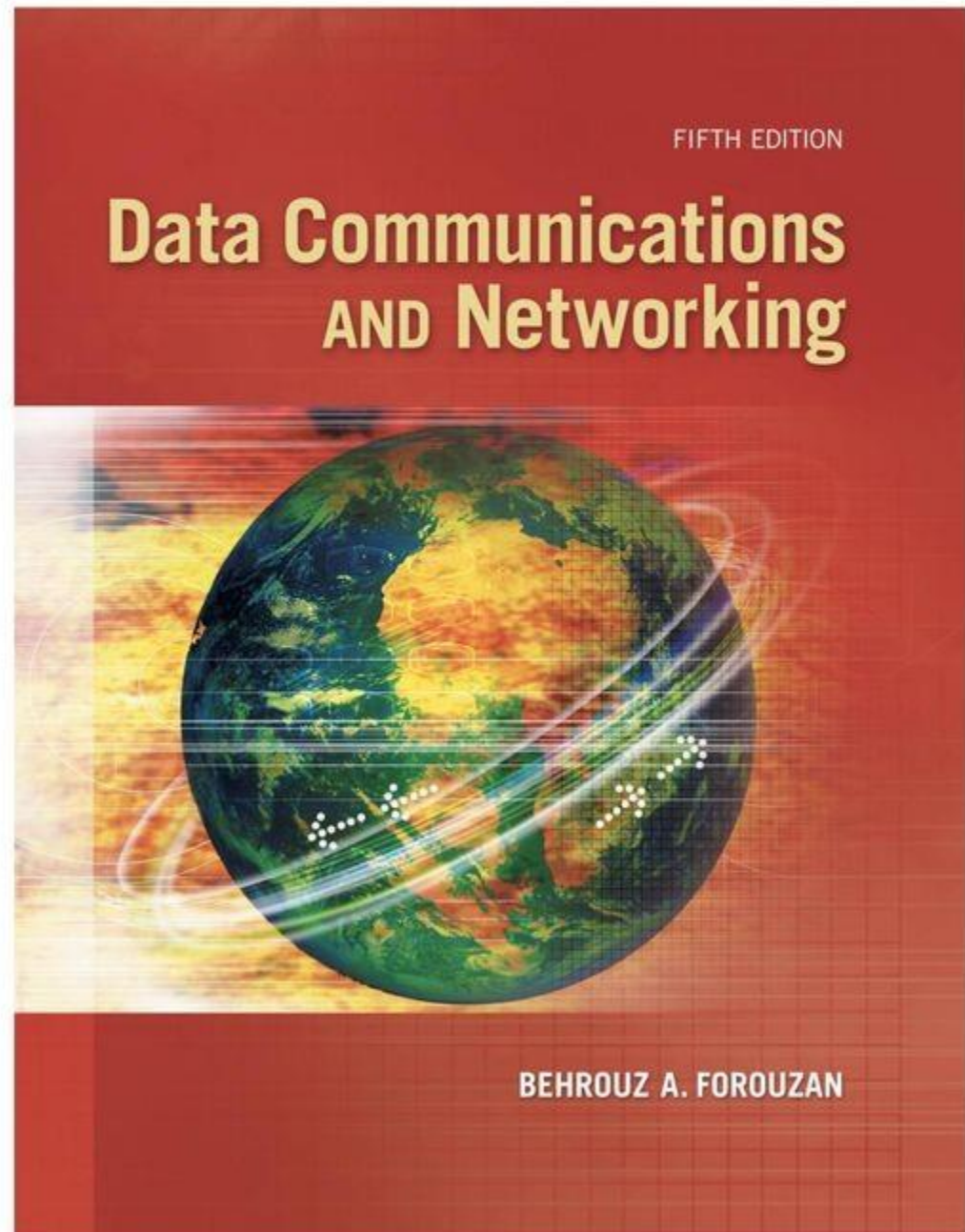


# ***Chapter 18***

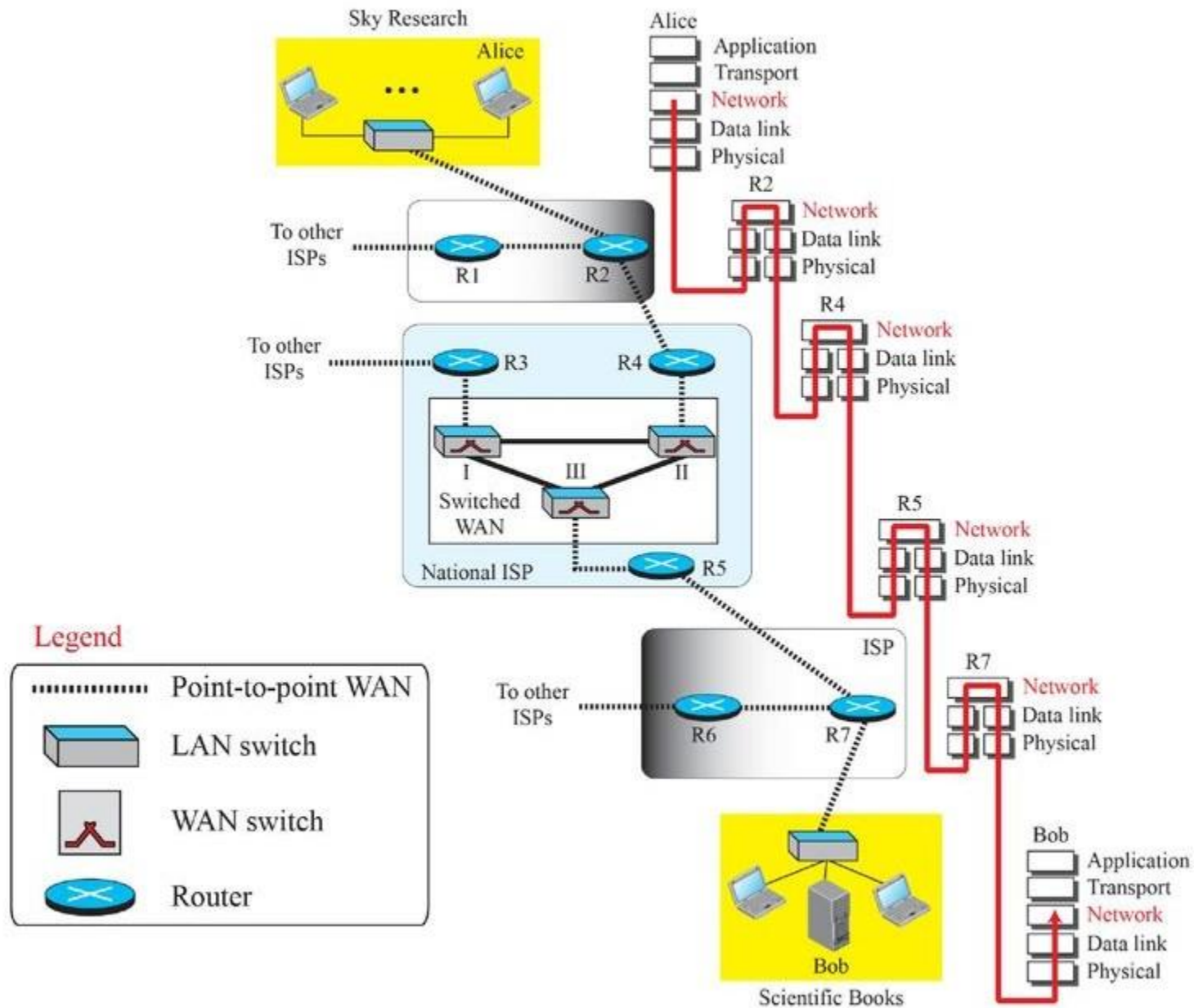
## ***Introduction to Network Layer***



## 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**







## 18.18.1 Packetizing

---

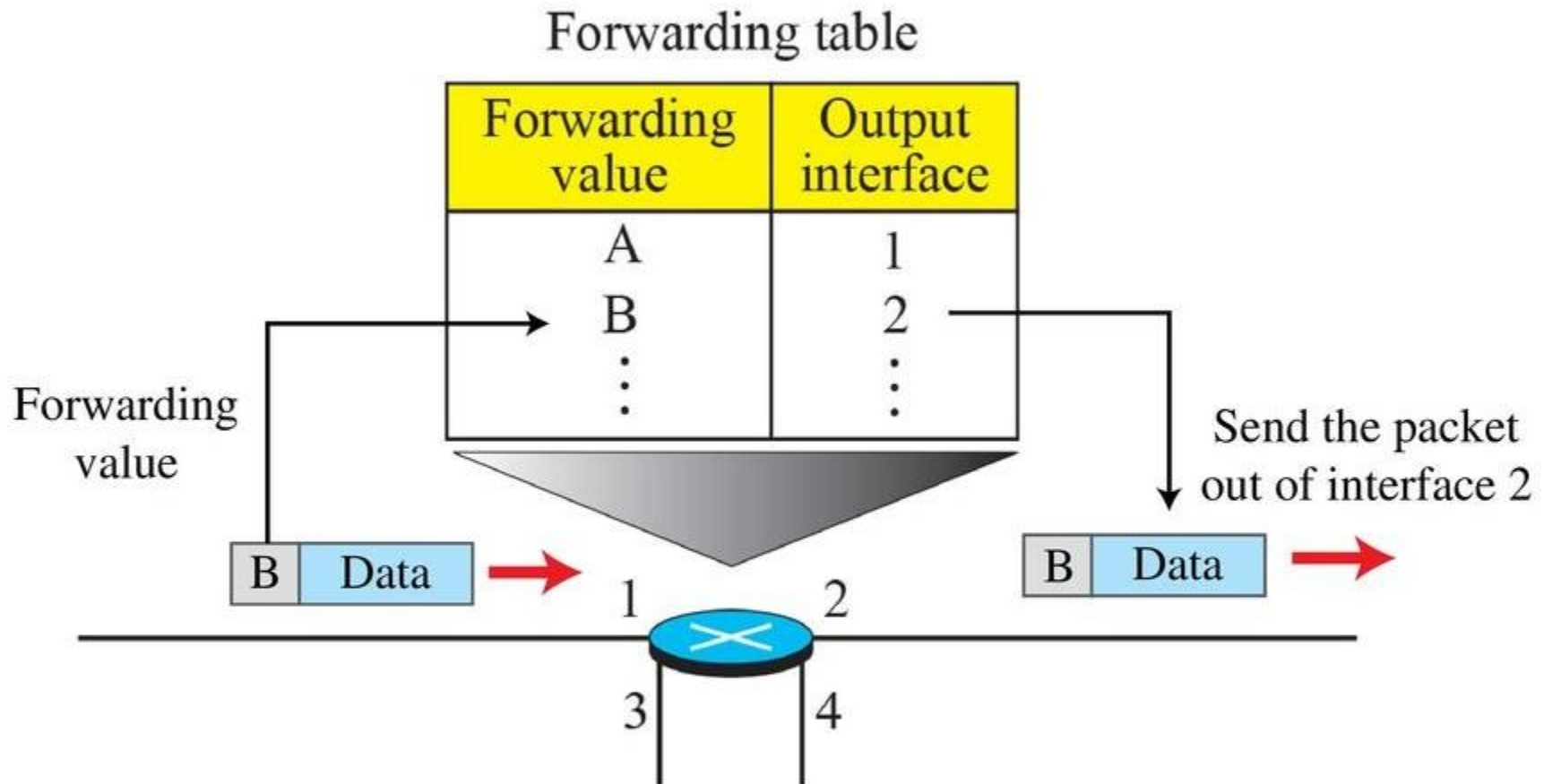
*The first duty of the network layer is definitely packetizing: encapsulating the payload in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination. In other words, one duty of the network layer is to carry a payload from the source to the destination without changing it or using it. The network layer is doing the service of a carrier such as the postal office, which is responsible for delivery of packages from a sender to a receiver without changing or using the contents.*



## ***18.18.2 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**





## ***18.18.3 Other Services***

---

*Let us briefly discuss other services expected from the network layer.*



## 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.*

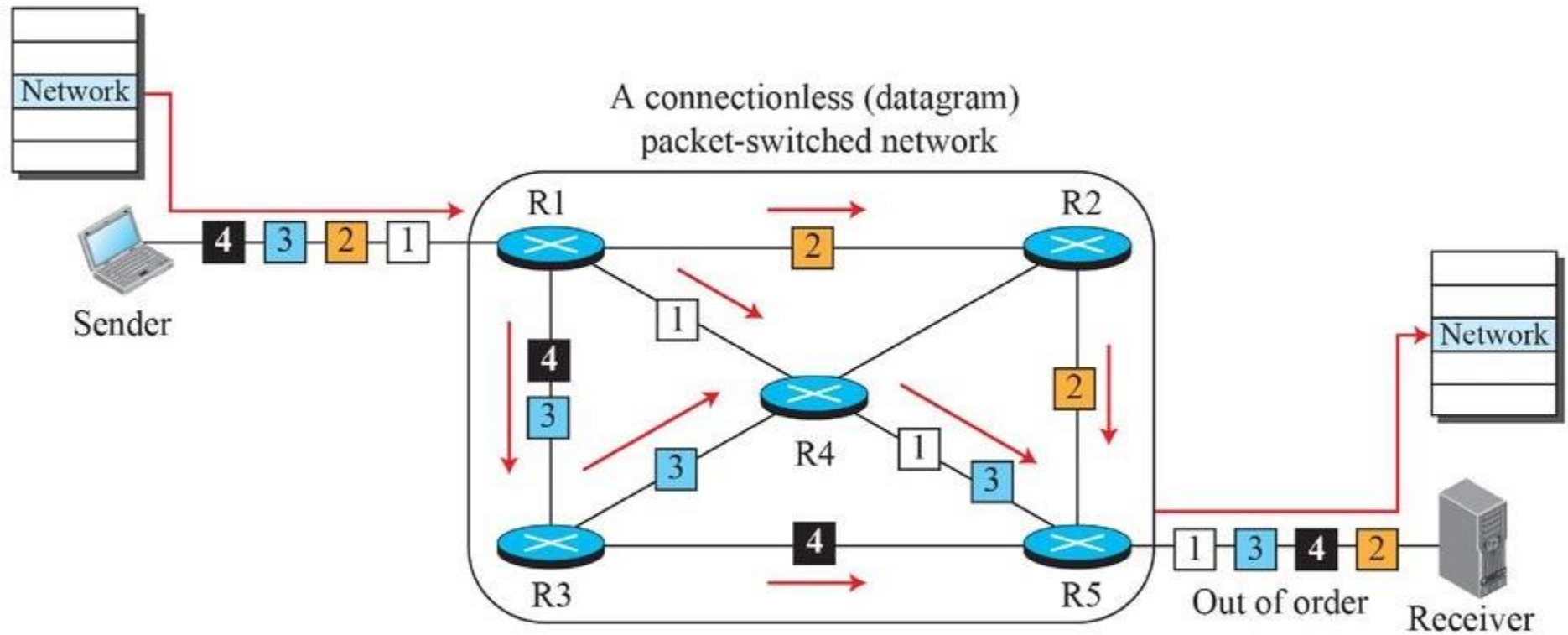




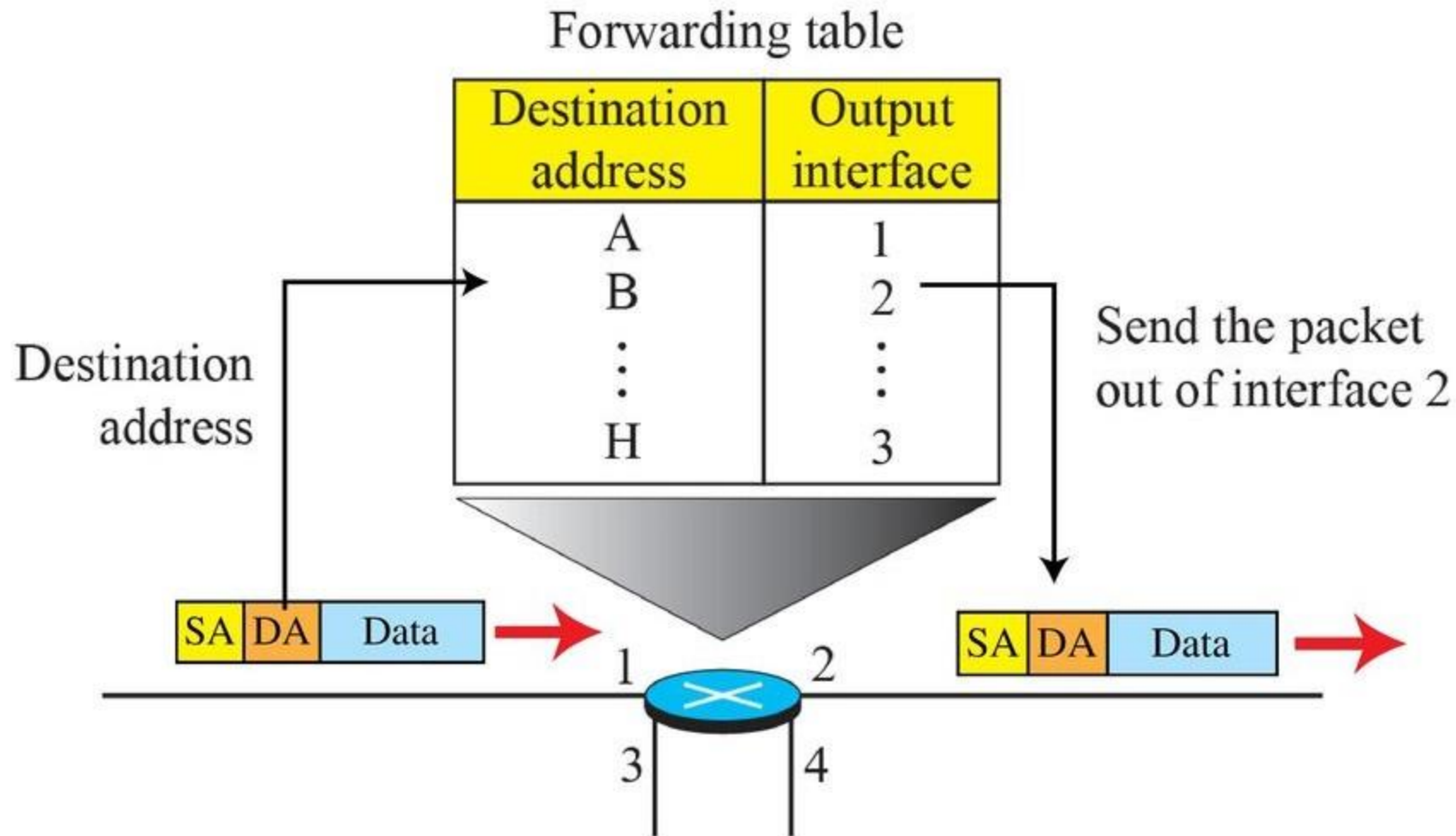
## ***18.2.1 Datagram Approach***

*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



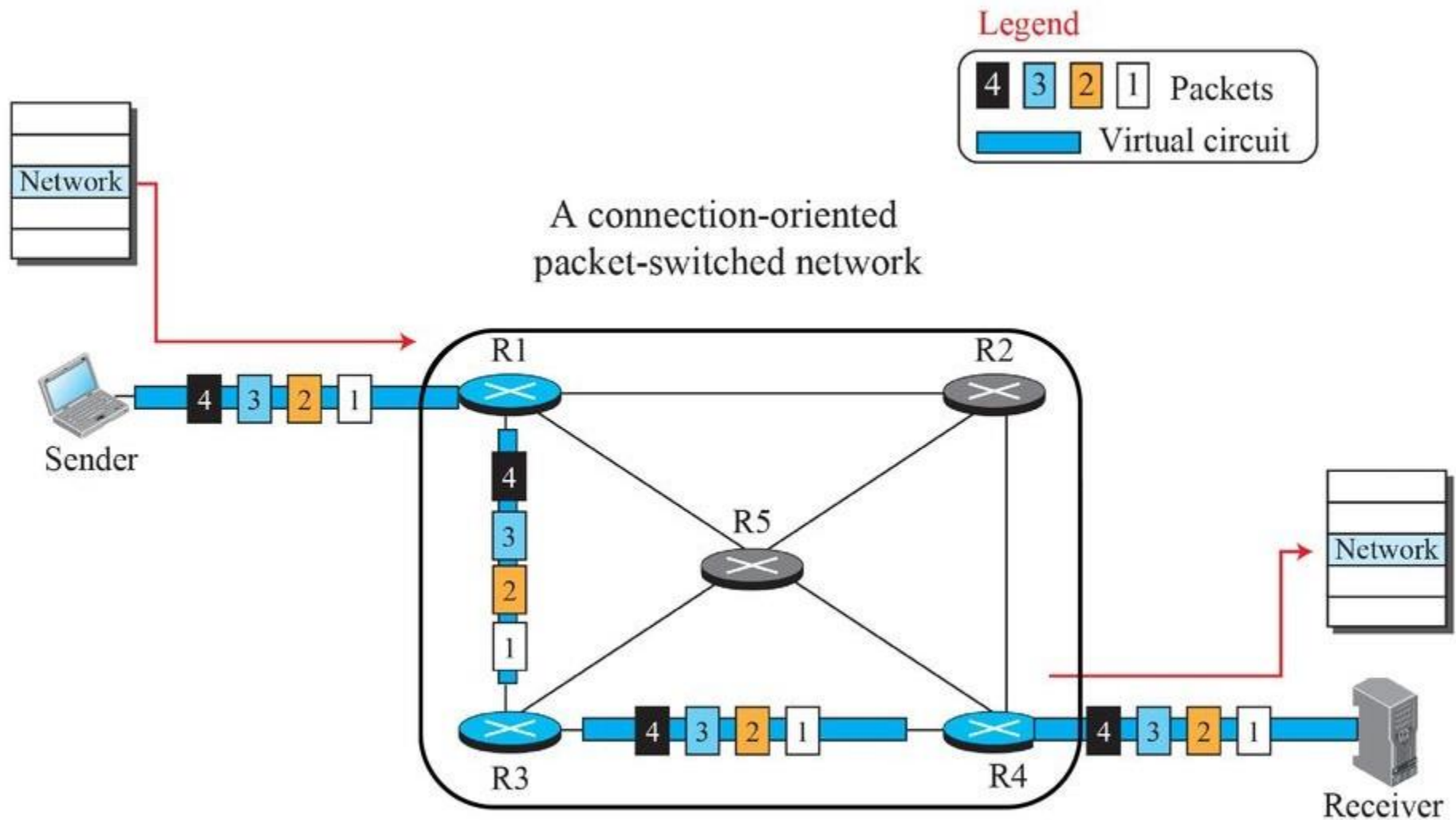




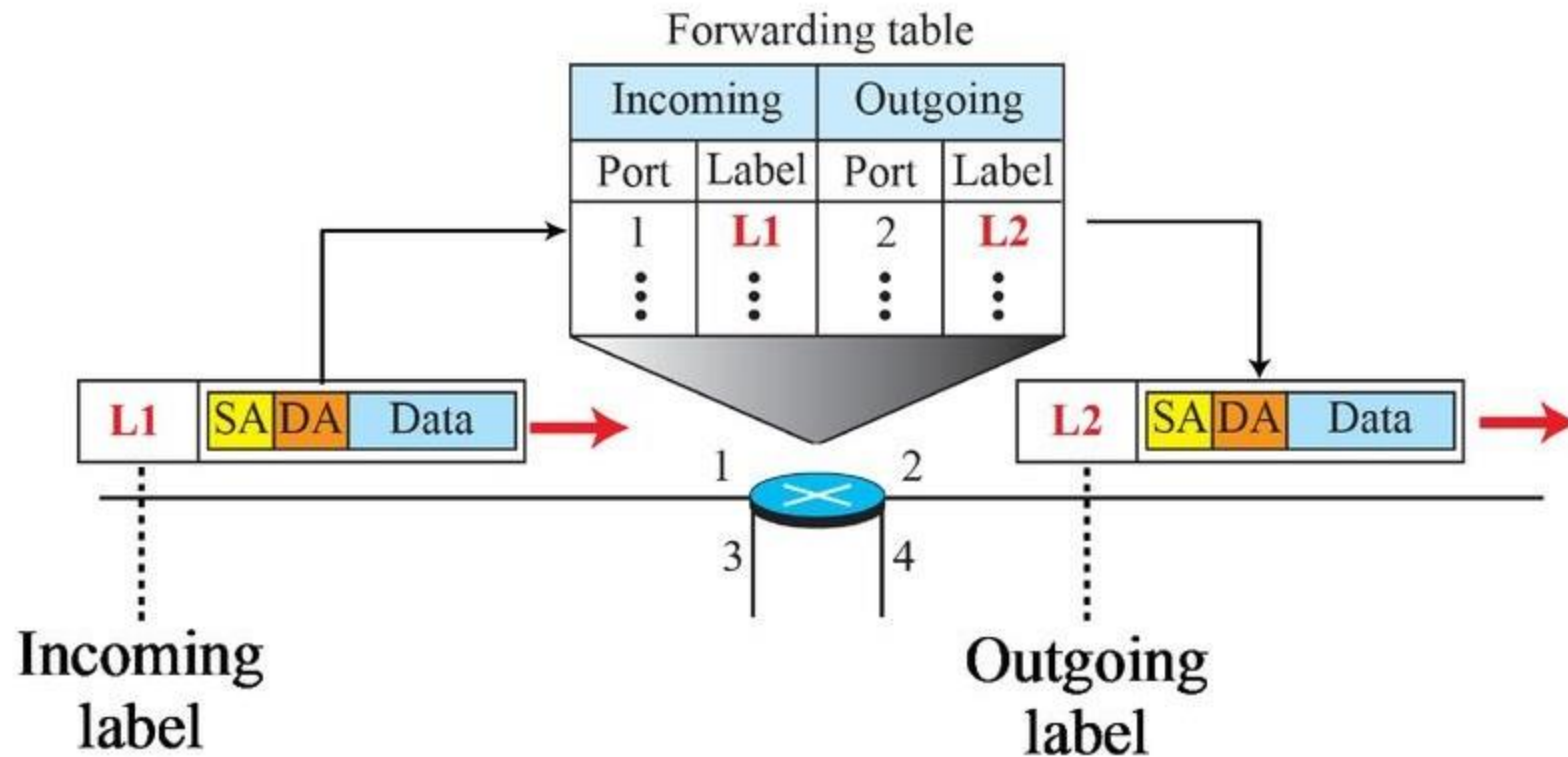
## ***18.2.2 Virtual-Circuit Approach***

*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*

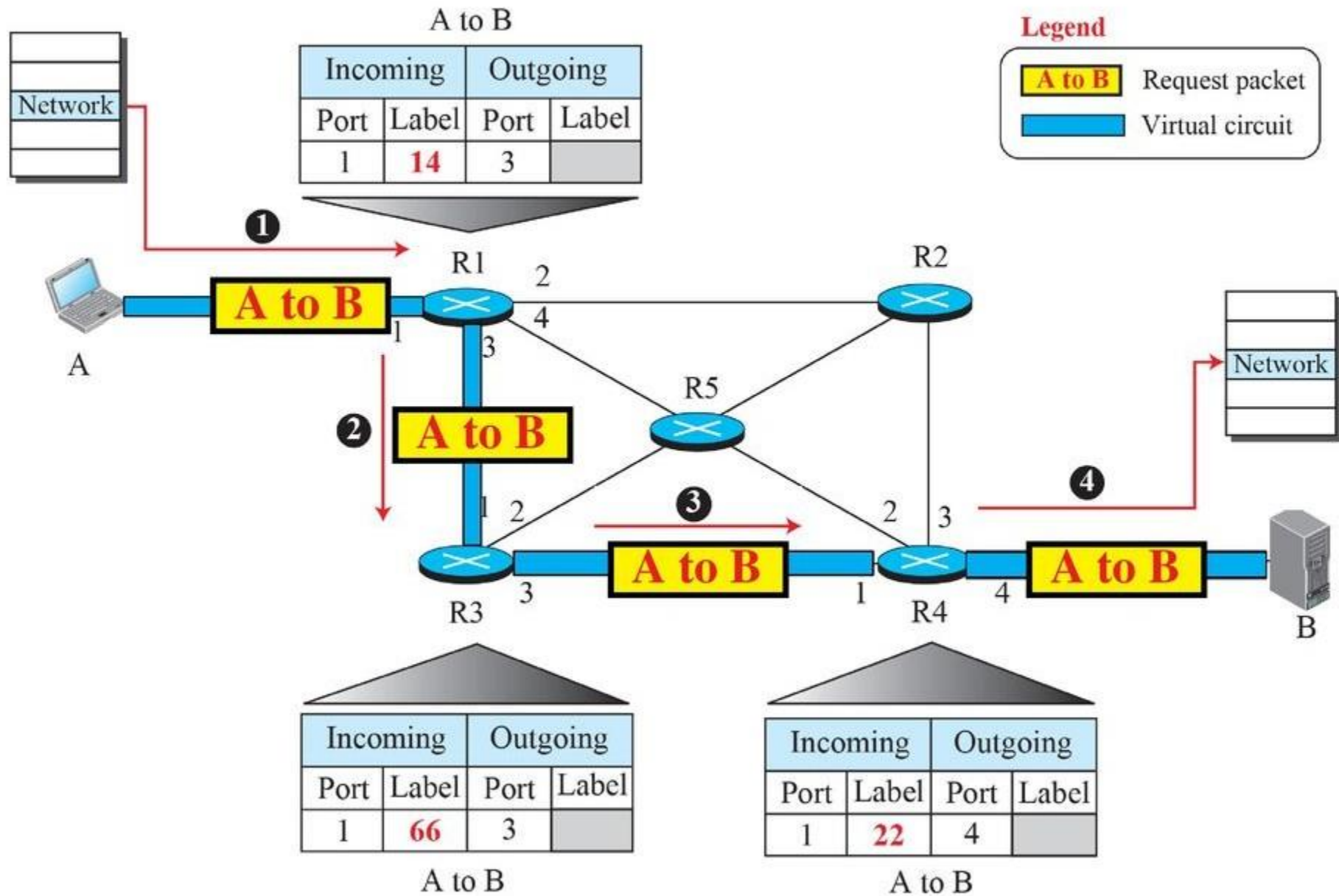


**Figure 18.6:** Forwarding process in a router when used in a virtual circuit network

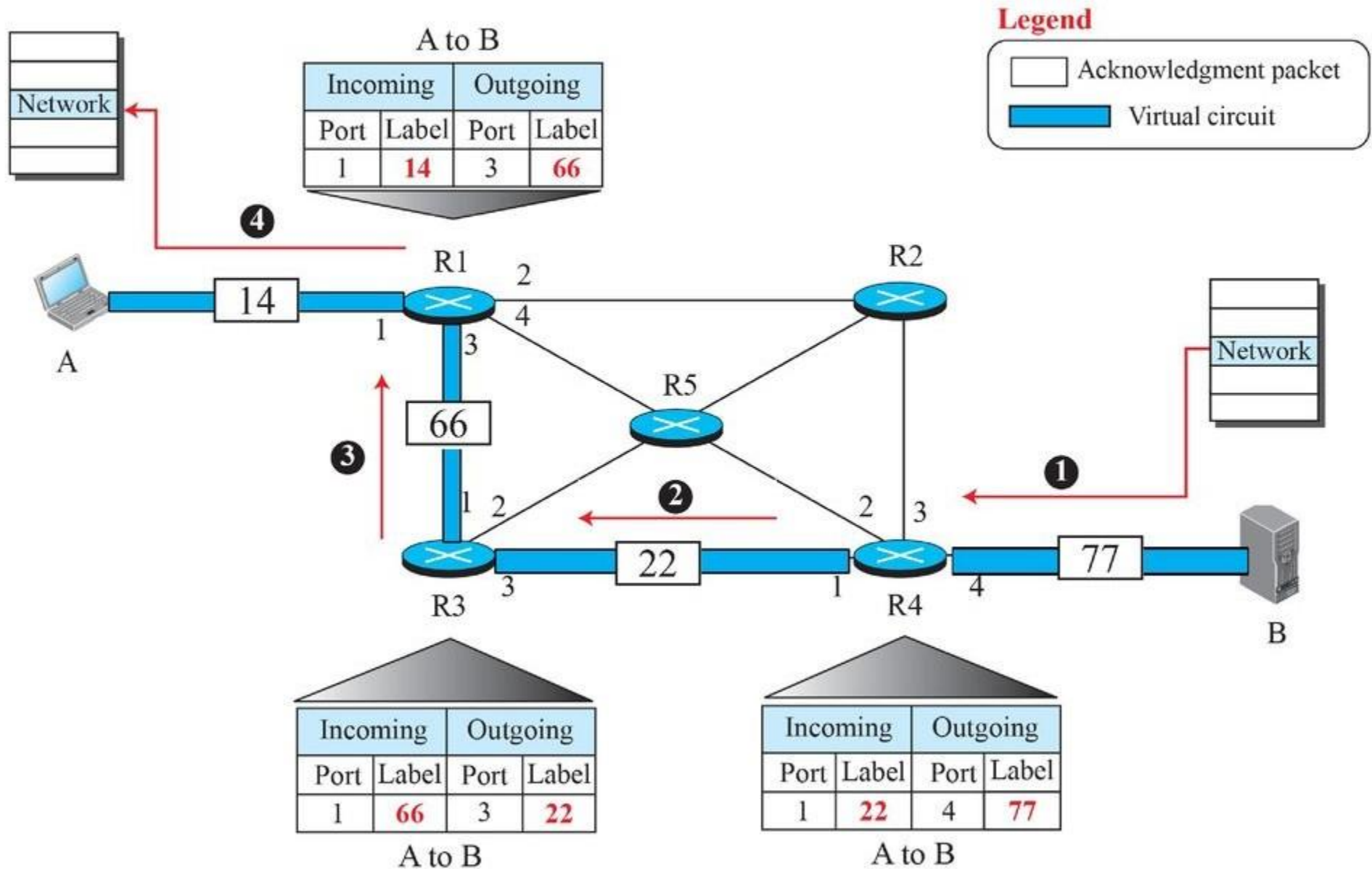




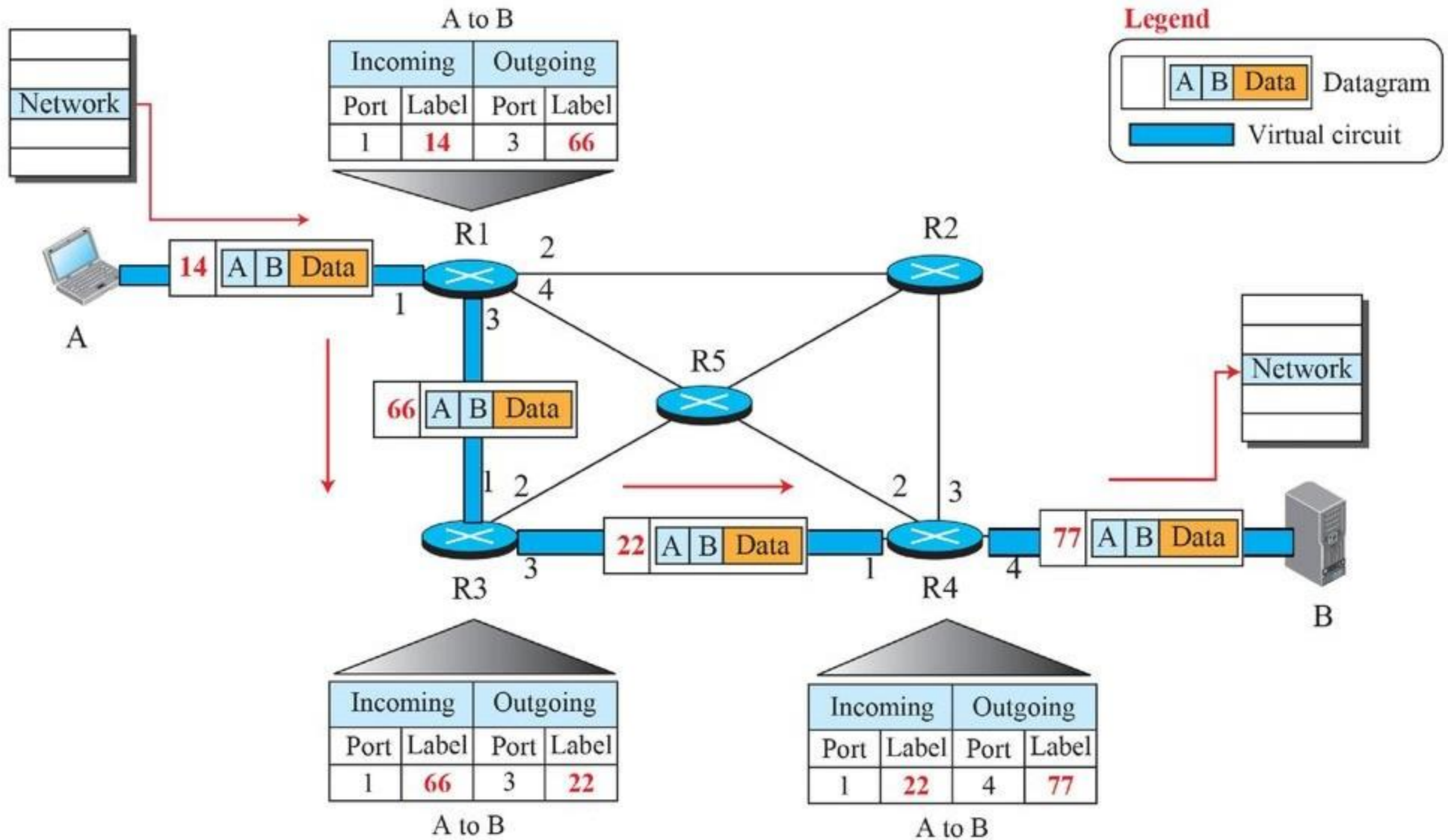
**Figure 18.7: Sending request packet in a virtual-circuit network**



**Figure 18.8: Sending acknowledgments in a virtual-circuit network**



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





## 18-3 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. Congestion control is an issue that can improve the performance.*



## 18.3.1 Delay

---

*All of us expect instantaneous response from a network, but a packet, from its source to its destination, encounters delays. The delays in a network can be divided into four types: transmission delay, propagation delay, processing delay, and queuing delay. Let us first discuss each of these delay types and then show how to calculate a packet delay from the source to the destination..*



## 18.3.2 Throughput

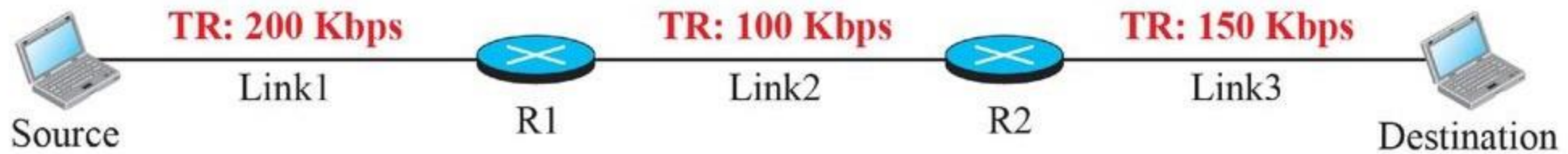
---

*Throughput at any point in a network is defined as the number of bits passing through the point in a second, which is actually the transmission rate of data at that point. In a path from source to destination, a packet may pass through several links (networks), each with a different transmission rate. How, then, can we determine the throughput of the whole path? To see the situation, assume that we have three links, each with a different transmission rate, as shown in Figure 18.10.*



**Figure 18.10:** *Throughput in a path with three links in a series*

TR: Transmission rate

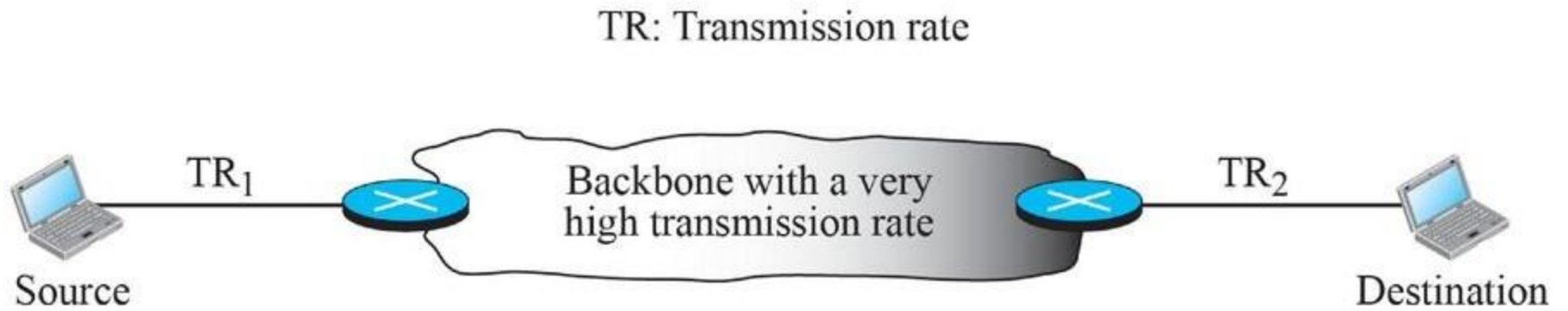


a. A path through three links

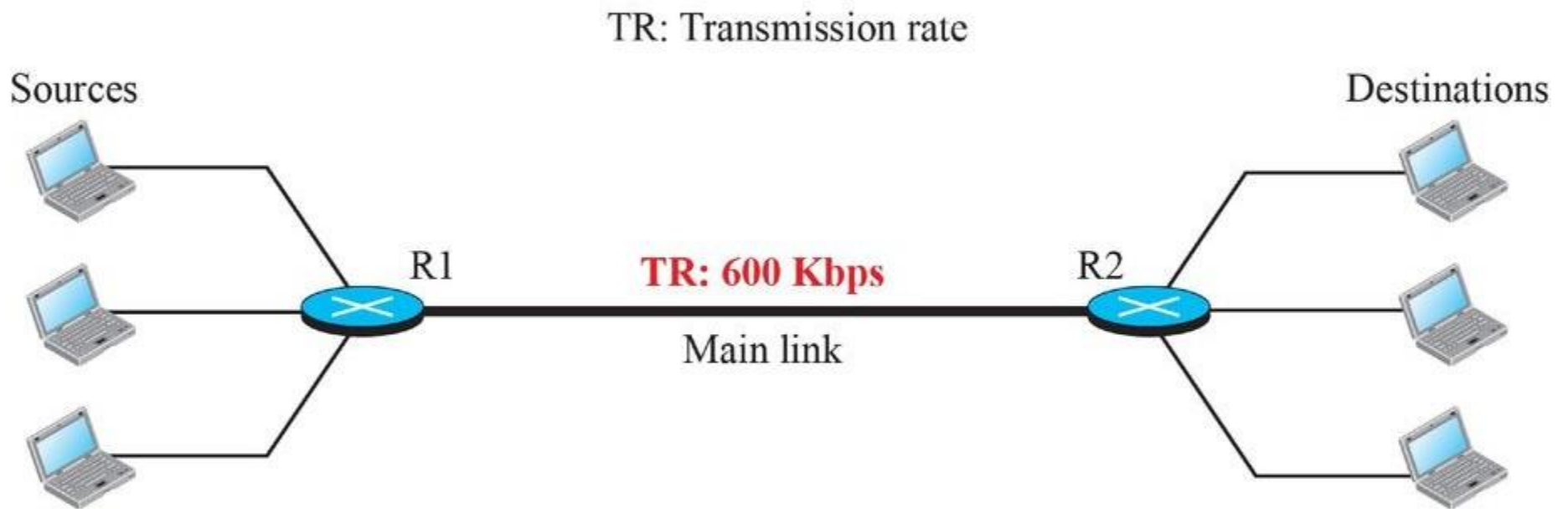


b. Simulation using pipes

**Figure 18.11:** *A path through the Internet backbone*



**Figure 18.12:** *Effect of throughput in shared links*





### ***18.3.3 Packet Loss***

---

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



## 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.***

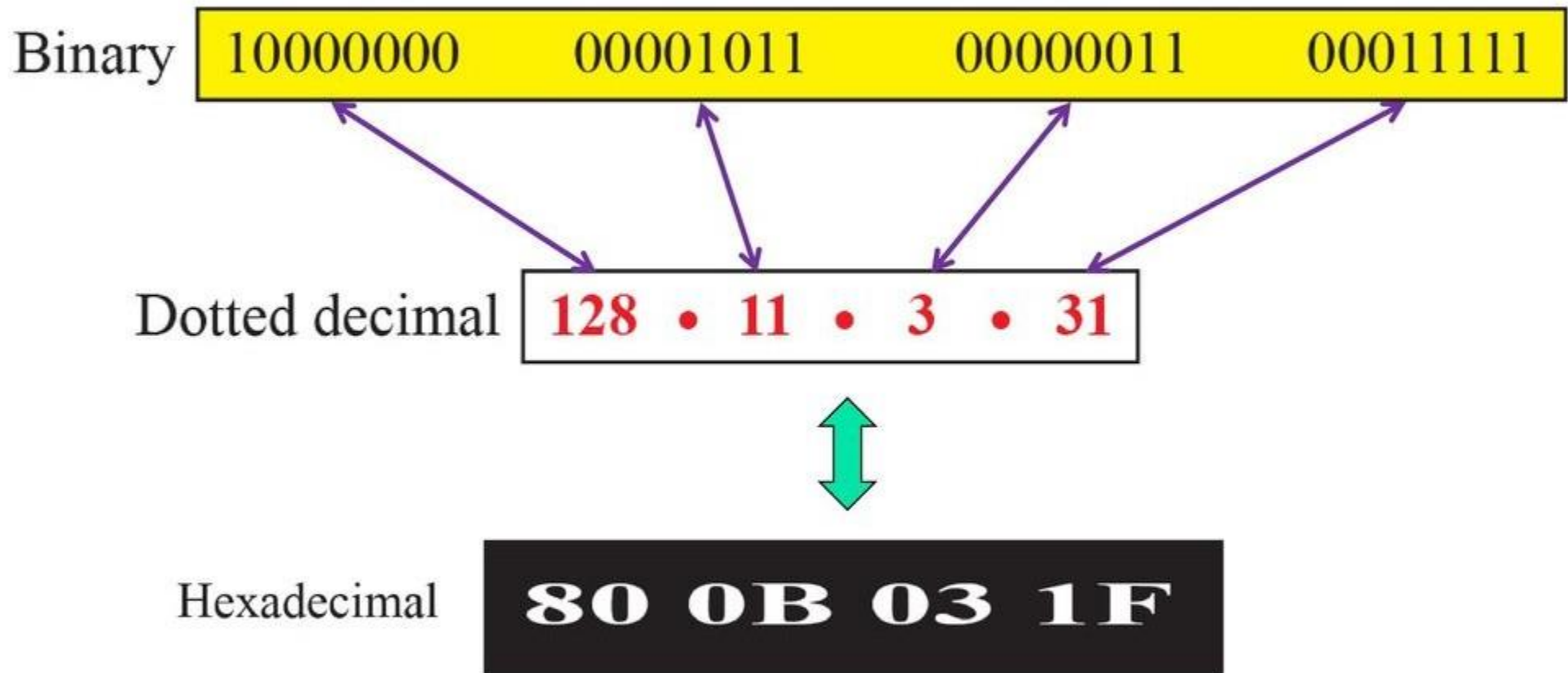


## ***18.4.1 Address Space***

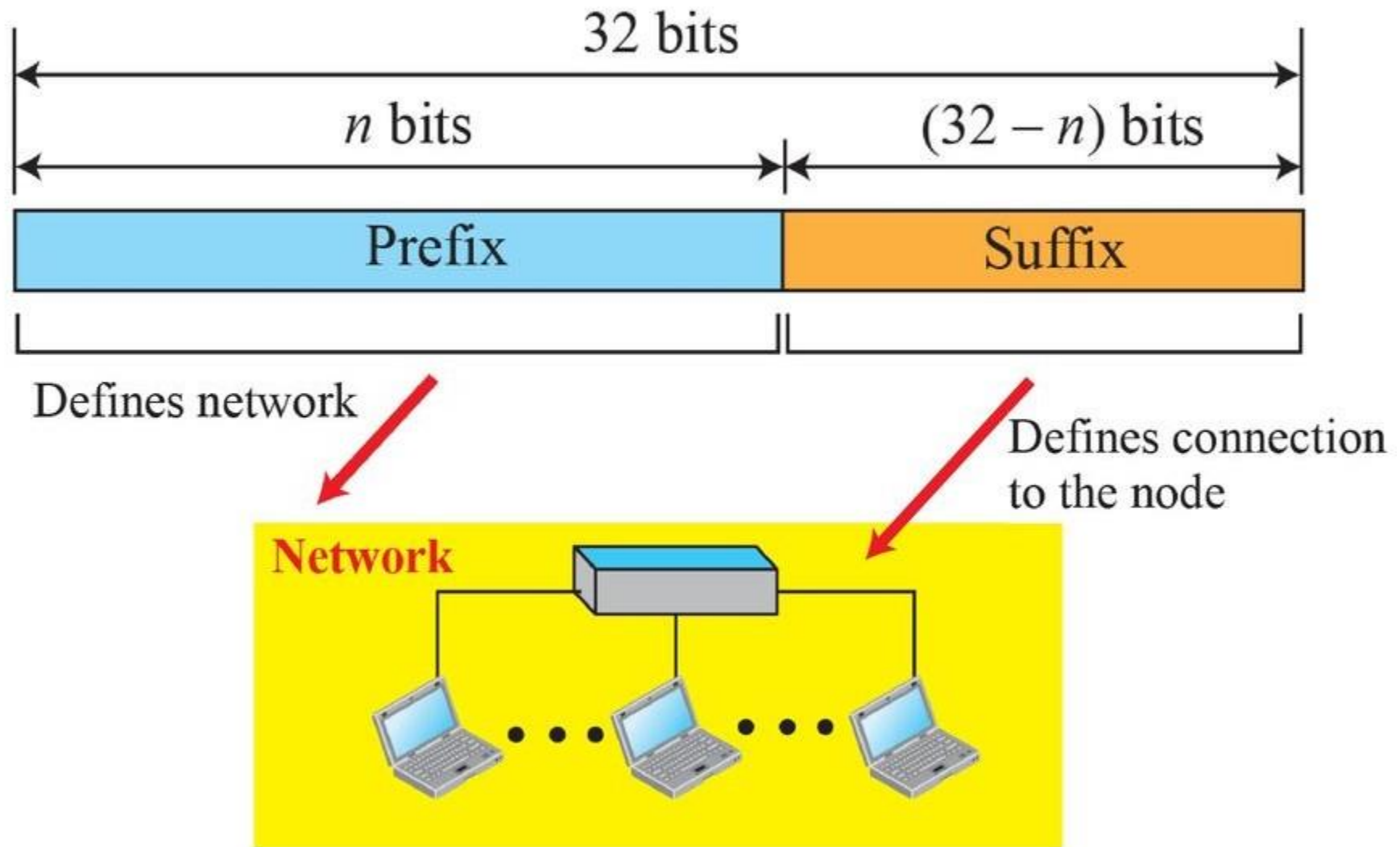
---

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

**Figure 18.16:** *Three different notations in IPv4 addressing*



**Figure 18.17: Hierarchy in addressing**



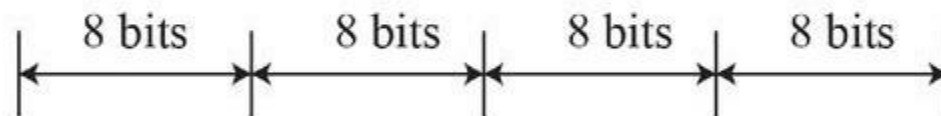
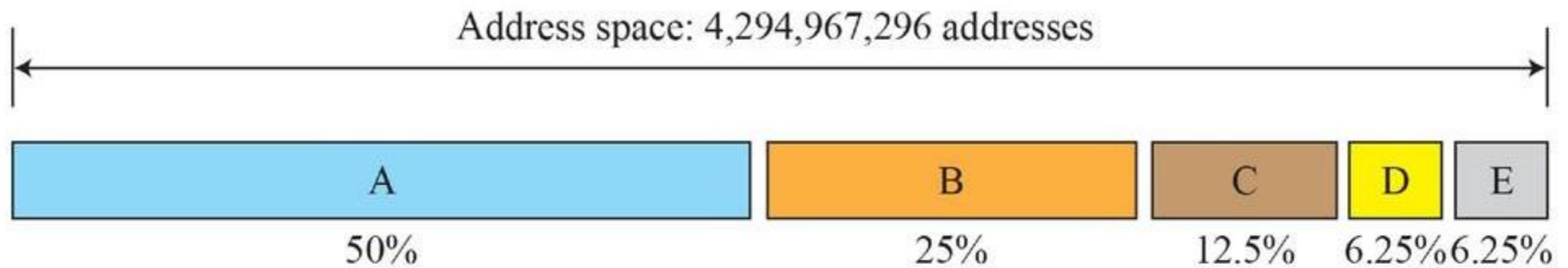




## 18.4.2 Classful Addressing

*When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one ( $n = 8$ ,  $n = 16$ , and  $n = 24$ ). The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure 18.18. This scheme is referred to as classful addressing. Although classful addressing belongs to the past, it helps us to understand classless addressing, discussed later.*

**Figure 18.18: Occupation of the address space in classful addressing**



Class A	0 Prefix	Suffix
Class B	10 Prefix	Suffix
Class C	110 Prefix	Suffix
Class D	1110 Multicast addresses	
Class E	1111 Reserved for future use	

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

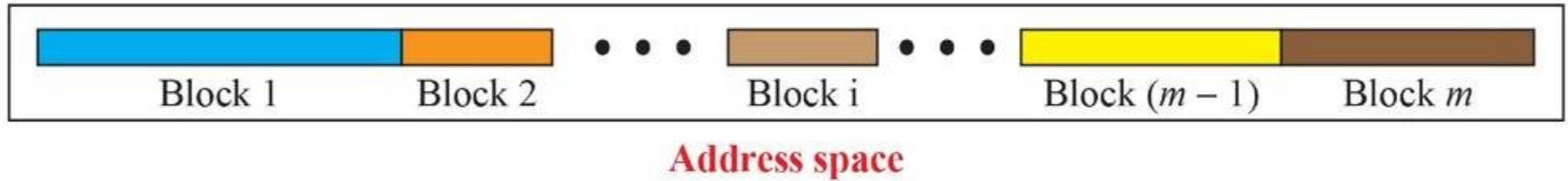


## 18.4.3 Classless Addressing

*With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing.*

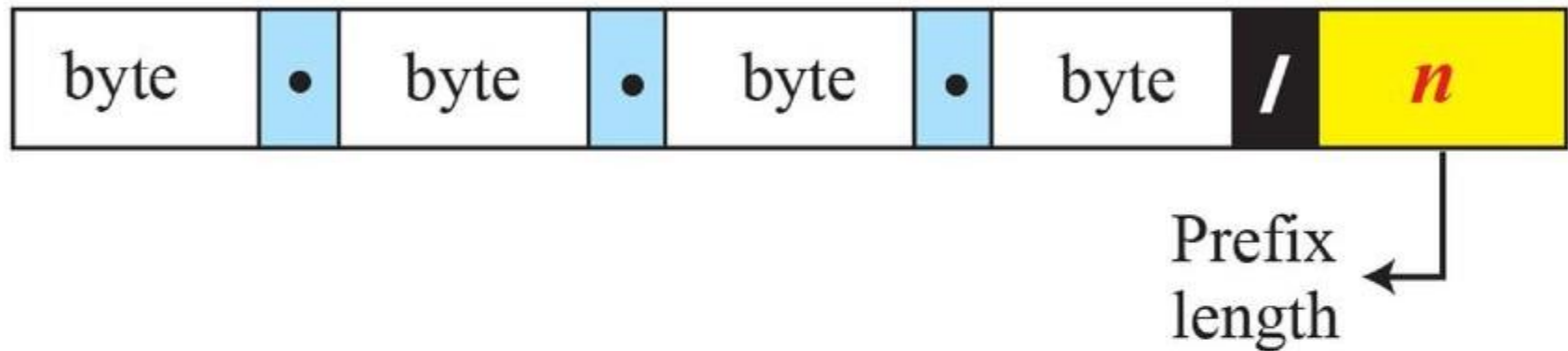


**Figure 18.19:** *Variable-length blocks in classless addressing*





**Figure 18.20: Slash notation (CIDR)**



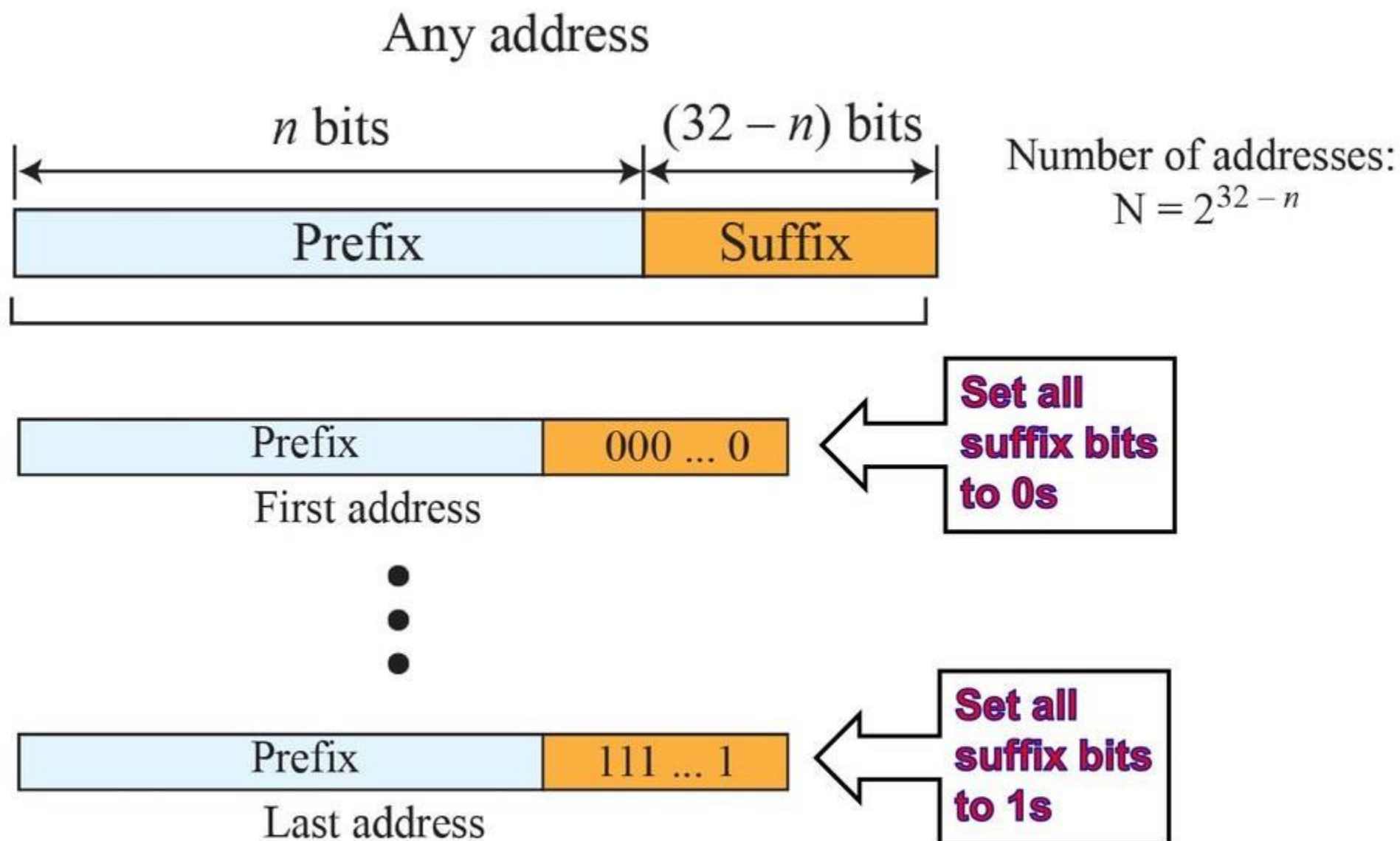
**Examples:**

12.24.76.8/**8**

23.14.67.92/**12**

220.8.24.255/**25**

**Figure 18.21:** Information extraction in classless addressing



## Example 18.1

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 18.2

We repeat Example 18.1 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:	$N = \text{NOT}(\text{mask}) + 1 = 0.0.0.31 + 1 = 32 \text{ addresses}$
First address:	$\text{First} = (\text{address}) \text{ AND } (\text{mask}) = 167.199.170.82$
Last address:	$\text{Last} = (\text{address}) \text{ OR } (\text{NOT mask}) = 167.199.170.255$

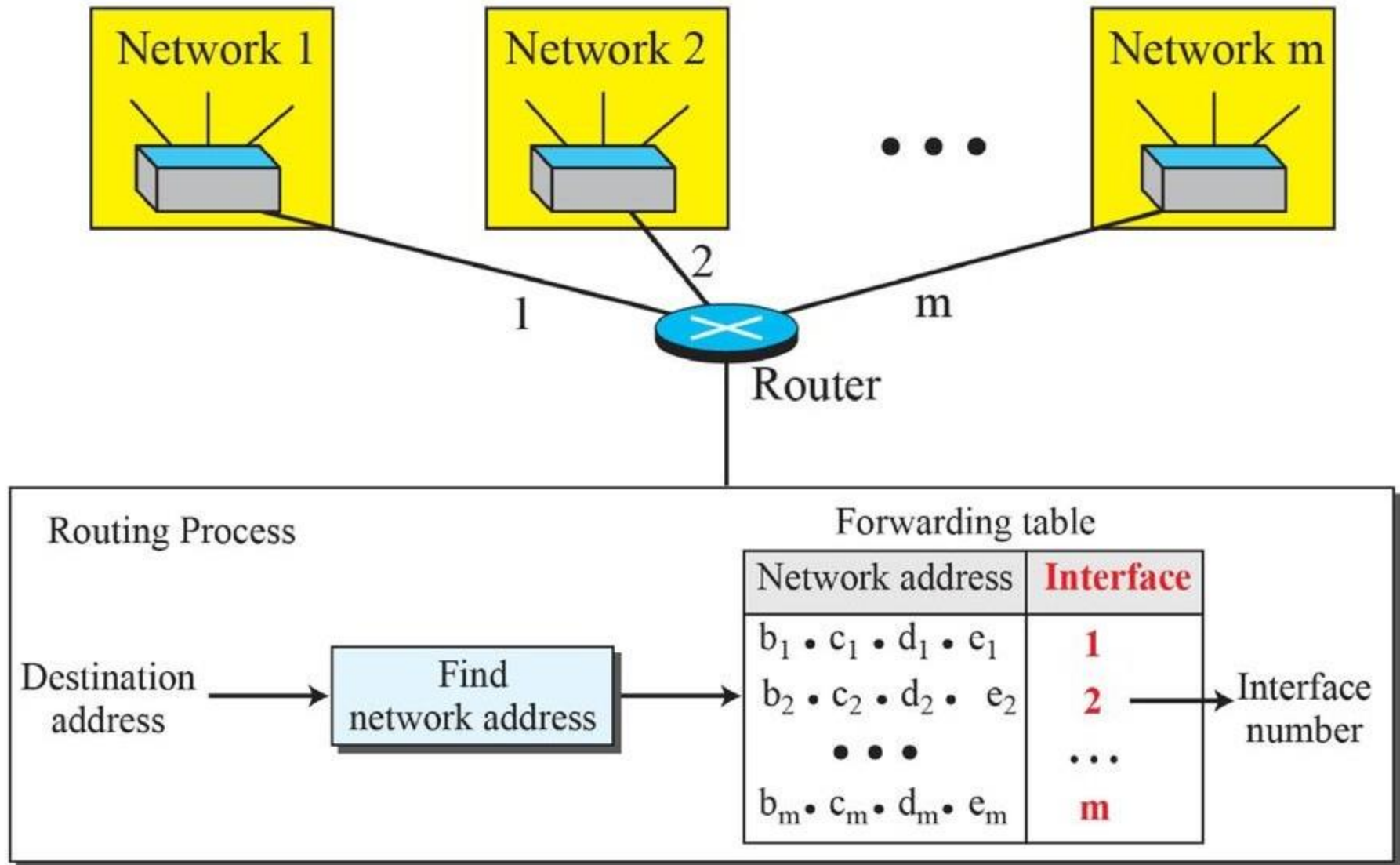


## Example 18.3

In classless addressing, an address cannot per se define the block the address belongs to. For example, the address 230.8.24.56 can belong to many blocks. Some of them are shown below with the value of the prefix associated with that block.

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

**Figure 18.22: Network address**



## Example 18.4

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 18.5

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

### Solution

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



## Example 18.5 (continued)

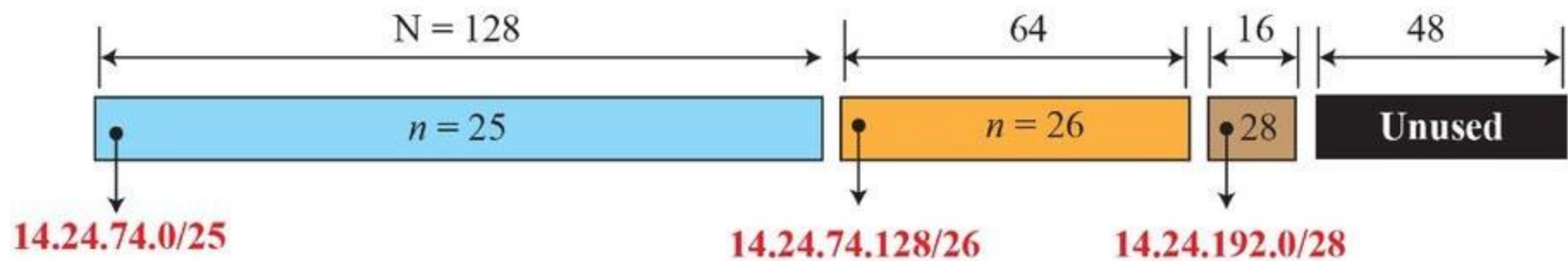
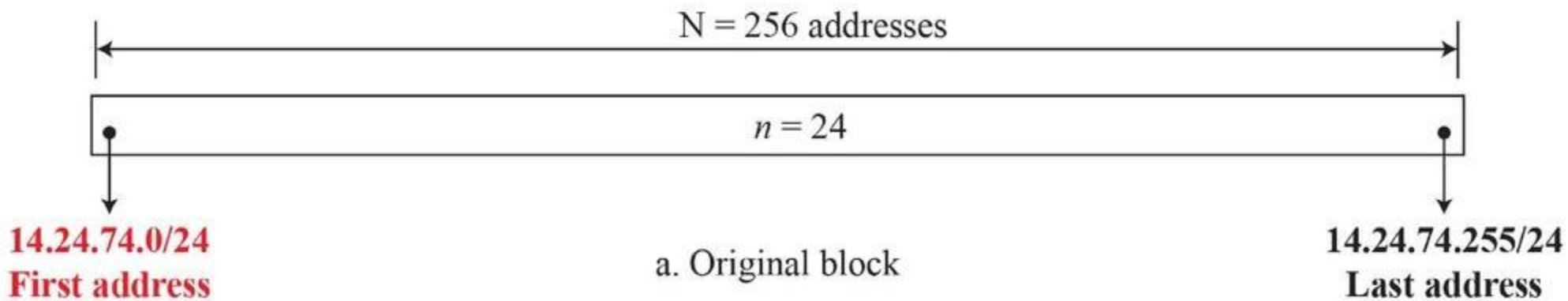
- a.** The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as  $n_1 = 32 - \log_2 128 = 25$ . The first address in this block is 14.24.74.0/**25**; the last address is 14.24.74.127/**25**.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

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

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255. We don't know about the prefix length yet. Figure 18.23 shows the configuration of blocks. We have shown the first address in each block.

**Figure 18.23: Solution to Example 4.5**





## ***Example 18.6***

Figure 18.24 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 18.24: Example of address aggregation**

