Topics Discussed in the Section

- ✓ **Process-to-Process Communication**
- ✓ **Addressing: Port Numbers**
- ✓ **Encapsulation and Decapsulation**
- ✓ **Multiplexing and Demultiplexing**
- ✓ **Flow Control**
- ✓ **Error Control**
- ✓ **Congestion Control**
- ✓ **Connectionless and Connection-Oriented Services**

Figure 13.1 *Network layer versus transport layer*

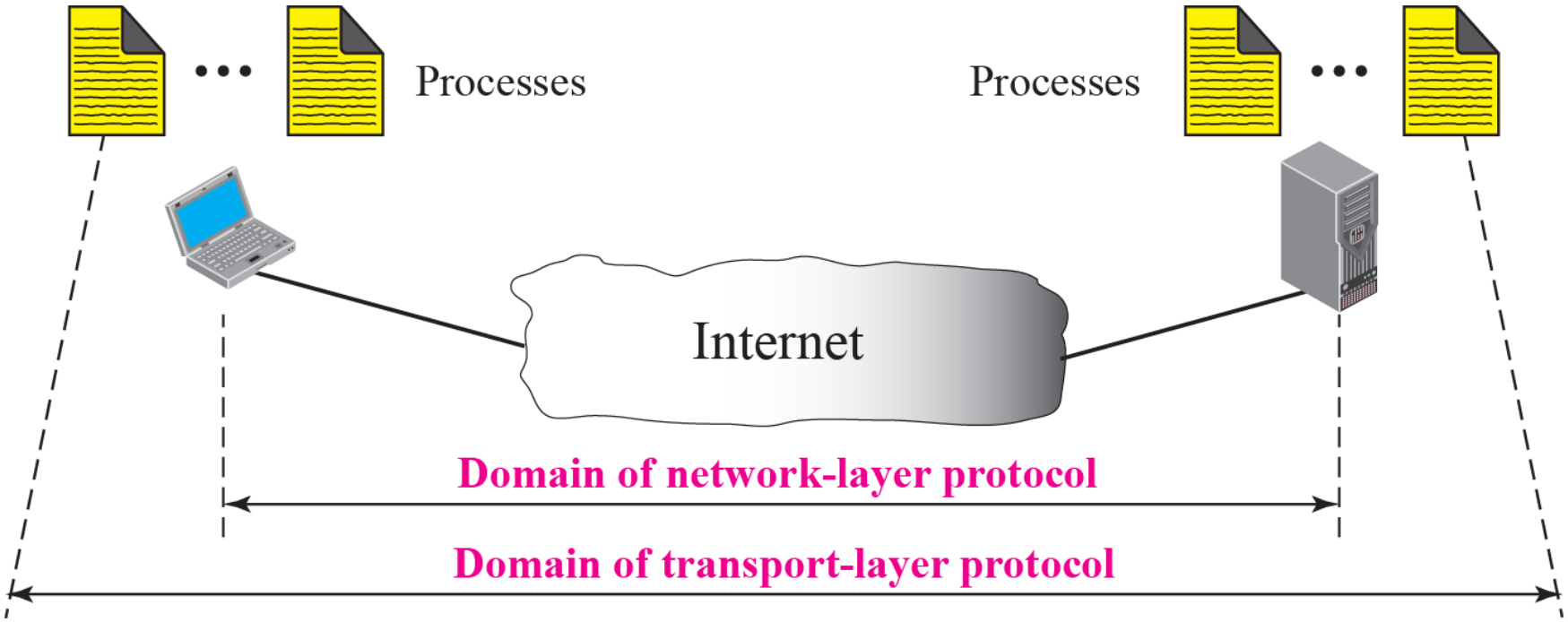


Figure 13.2 *Port numbers*

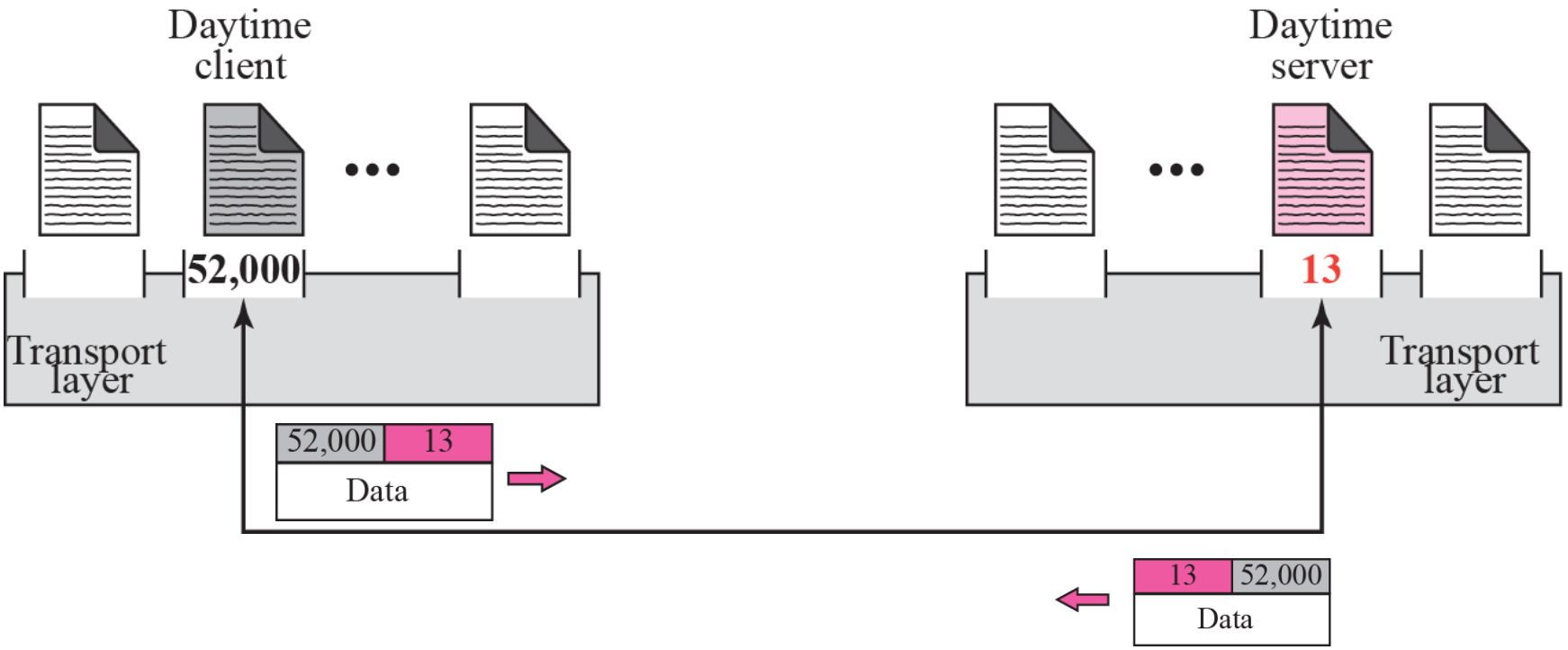


Figure 13.3 *IP addresses versus port numbers*

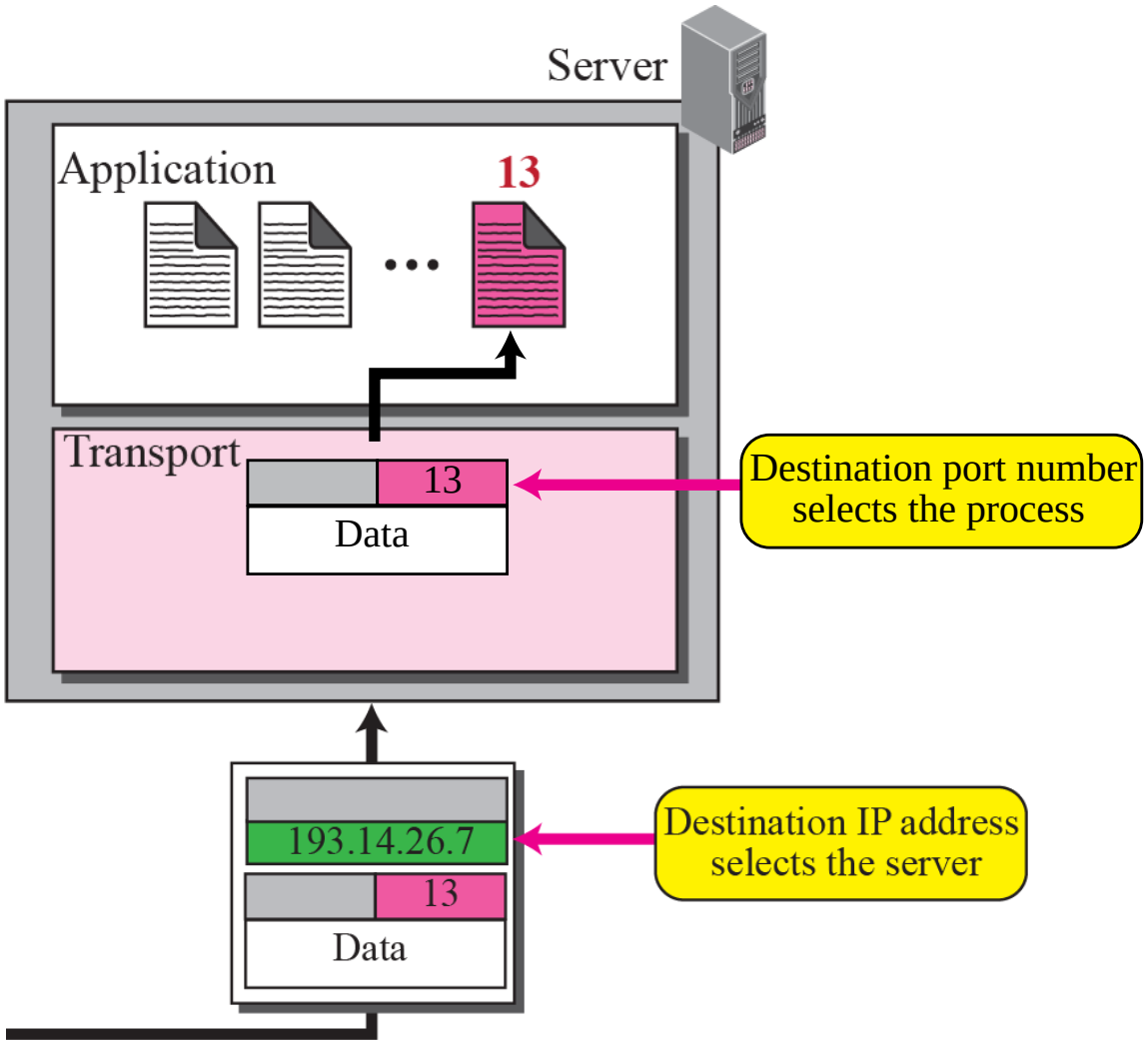
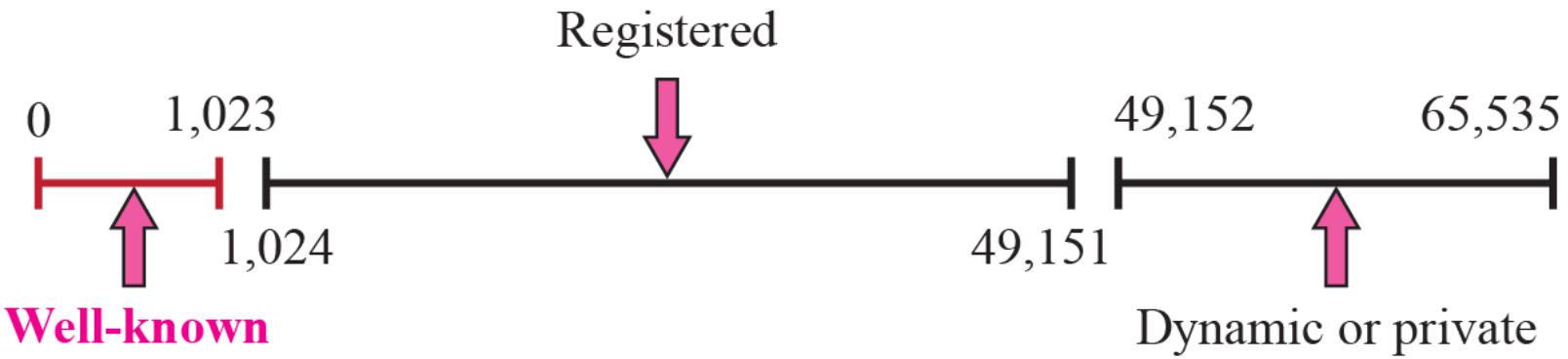


Figure 13.4 *ICANN ranges*



Note

The well-known port numbers are less than 1,024.

Example 13.1

In UNIX, the well-known ports are stored in a file called /etc/services. Each line in this file gives the name of the server and the well-known port number. We can use the grep utility to extract the line corresponding to the desired application. The following shows the port for TFTP. Note that TFTP can use port 69 on either UDP or TCP. SNMP (see Chapter 24) uses two port numbers (161 and 162), each for a different purpose.

```
$grep    tftp   /etc/services
   tftp          69/tcp
   tftp          69/udp
```

```
$grep    snmp /etc/services
   snmp161/tcp#Simple Net Mgmt Proto
   snmp161/udp#Simple Net Mgmt Proto
   snmptrap162/udp#Traps for SNMP
```

Figure 13.5 *Socket address*

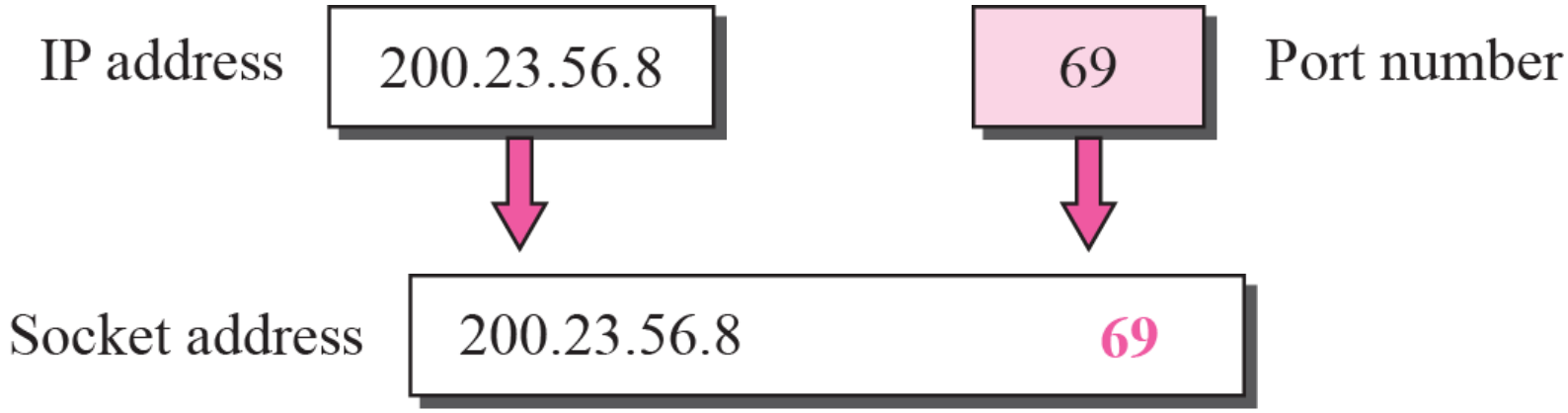


Figure 13.6 *Encapsulation and decapsulation*

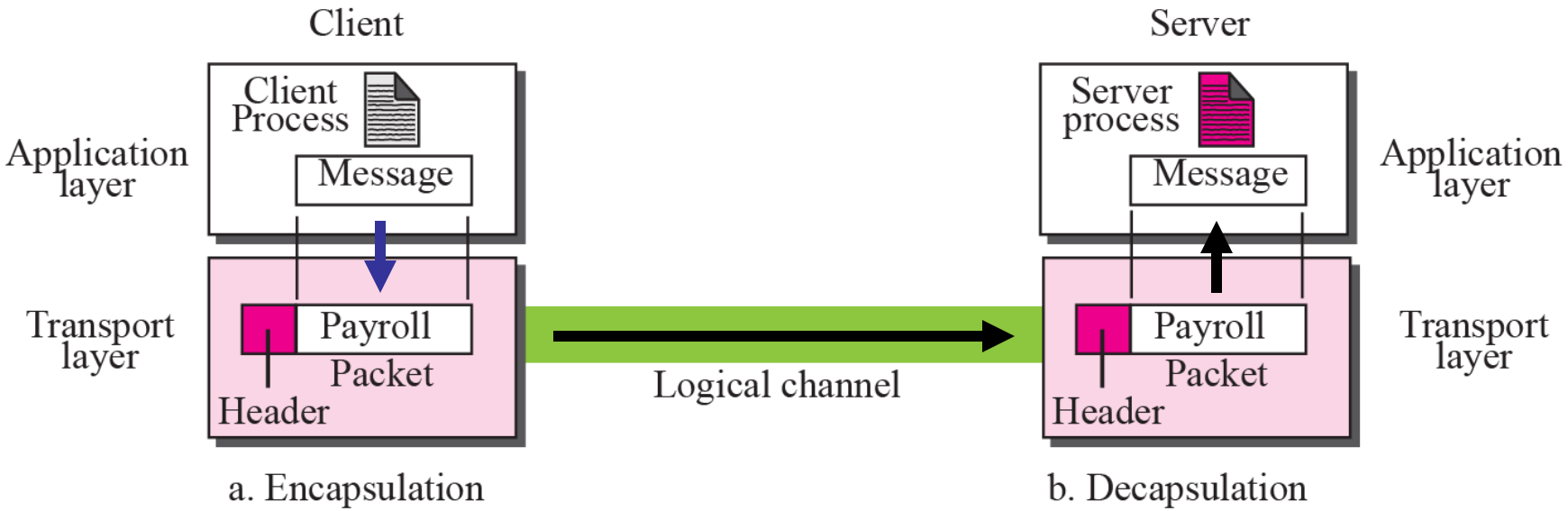


Figure 13.7 *Multiplexing and demultiplexing*

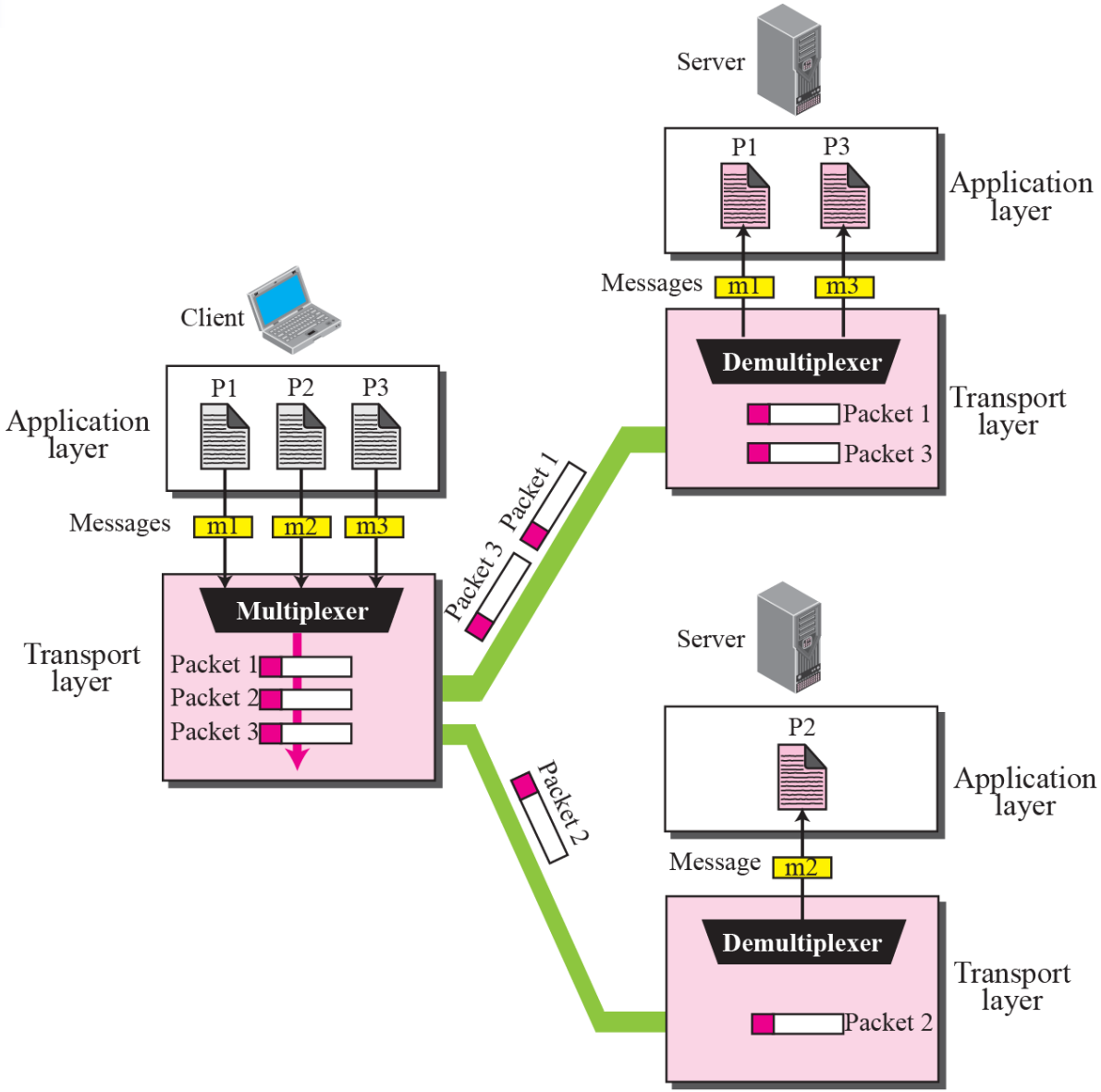


Figure 13.8 *Pushing or pulling*

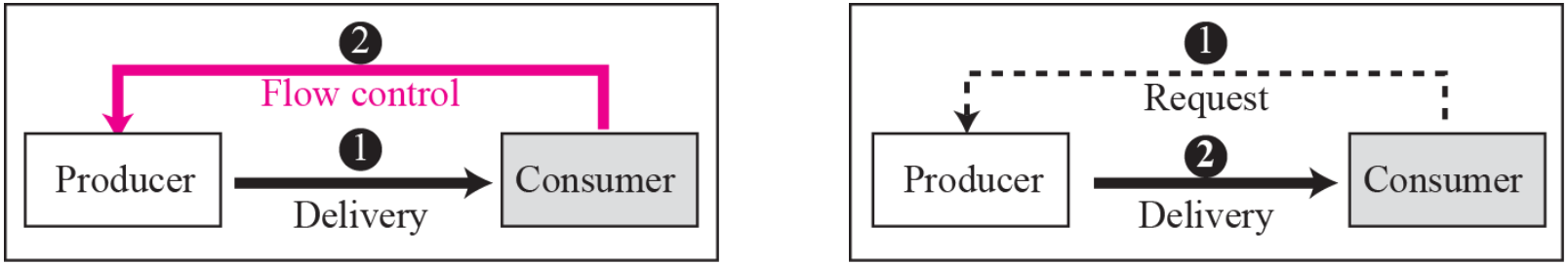
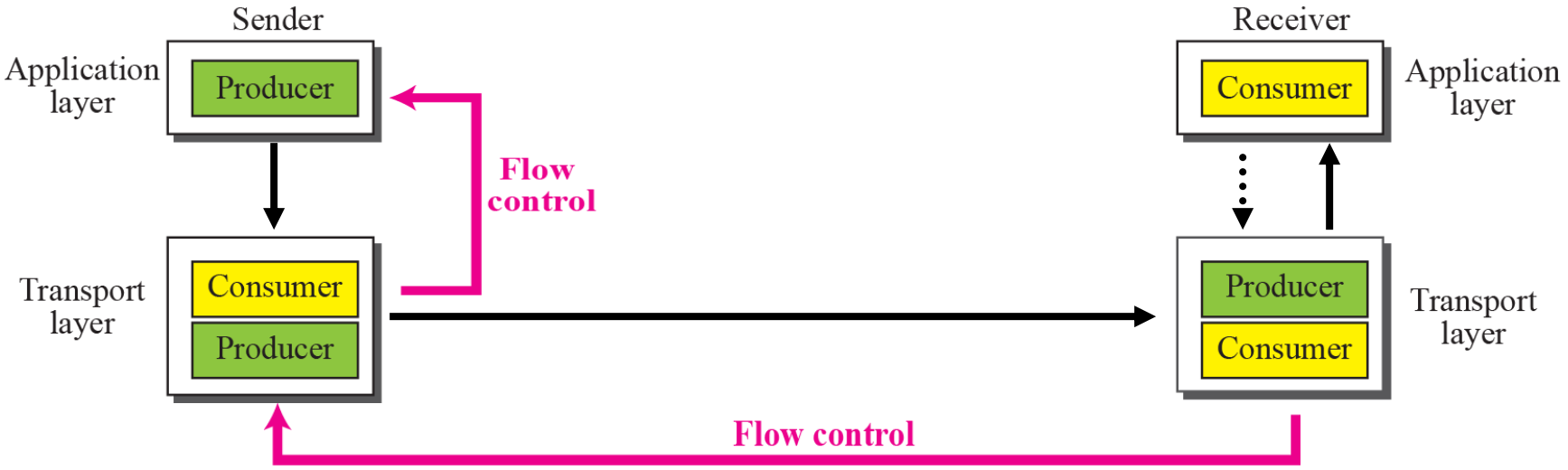


Figure 13.9 *Flow control at the transport layer*



Example 13.2

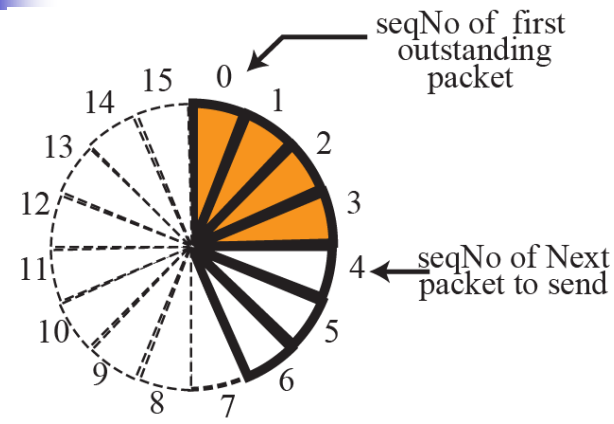
The above discussion requires that the consumers communicate with the producers in two occasions: when the buffer is full and when there are vacancies. If the two parties use a buffer of only one slot, the communication can be easier. Assume that each transport layer uses one single memory location to hold a packet. When this single slot in the sending transport layer is empty, the sending transport layer sends a note to the application layer to send its next chunk; when this single slot in the receiving transport layer is empty, it sends an acknowledgment to the sending transport layer to send its next packet. As we will see later, this type of flow control, using a single-slot buffer at the sender and the receiver, is inefficient.

Figure 13.10 *Error control at the transport layer*

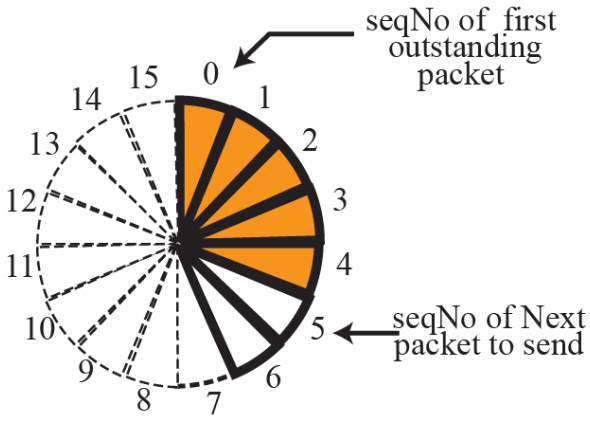
Note

For error control, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.

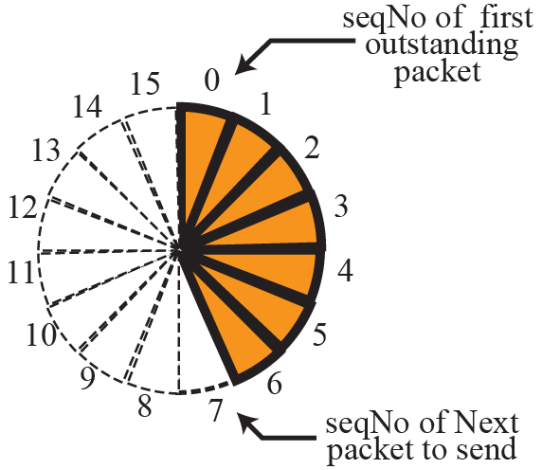
Figure 13.11 *Sliding window in circular format*



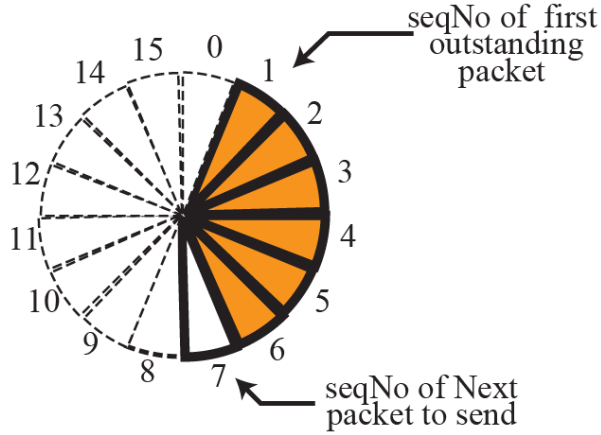
a. Four packets have been sent



b. Five packets have been sent



c. Seven packets have been sent
window is full



d. Packet 0 has been acknowledged,
window slides

Figure 13.12 *Sliding window in linear format*

a. Four packets have been sent

b. Five packets have been sent

c. Seven packets have been sent
window is full

d. Packet 0 have been acknowledged
and window slid

Figure 13.13 *Connectionless service*

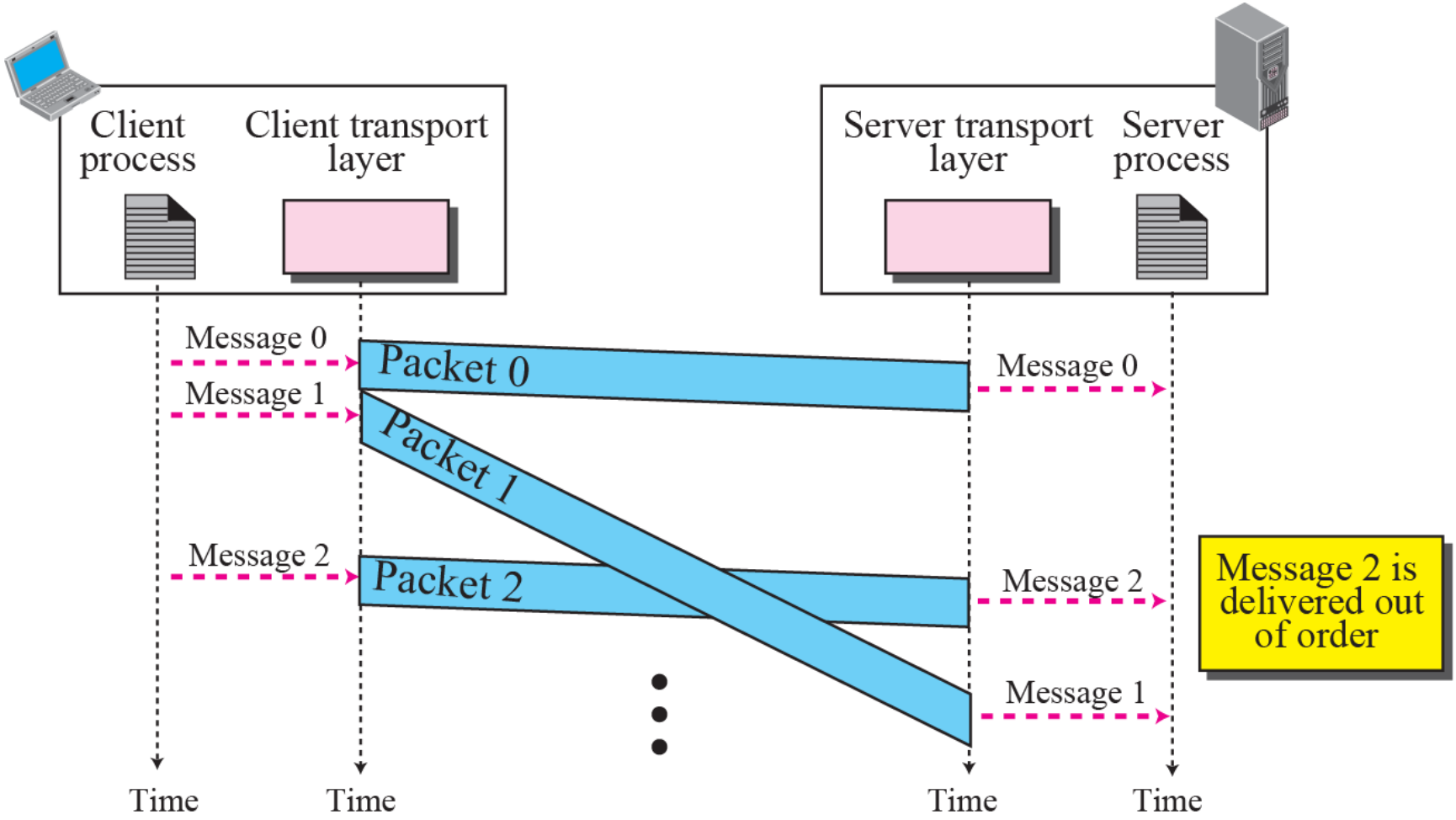


Figure 13.14 *Connection-oriented service*

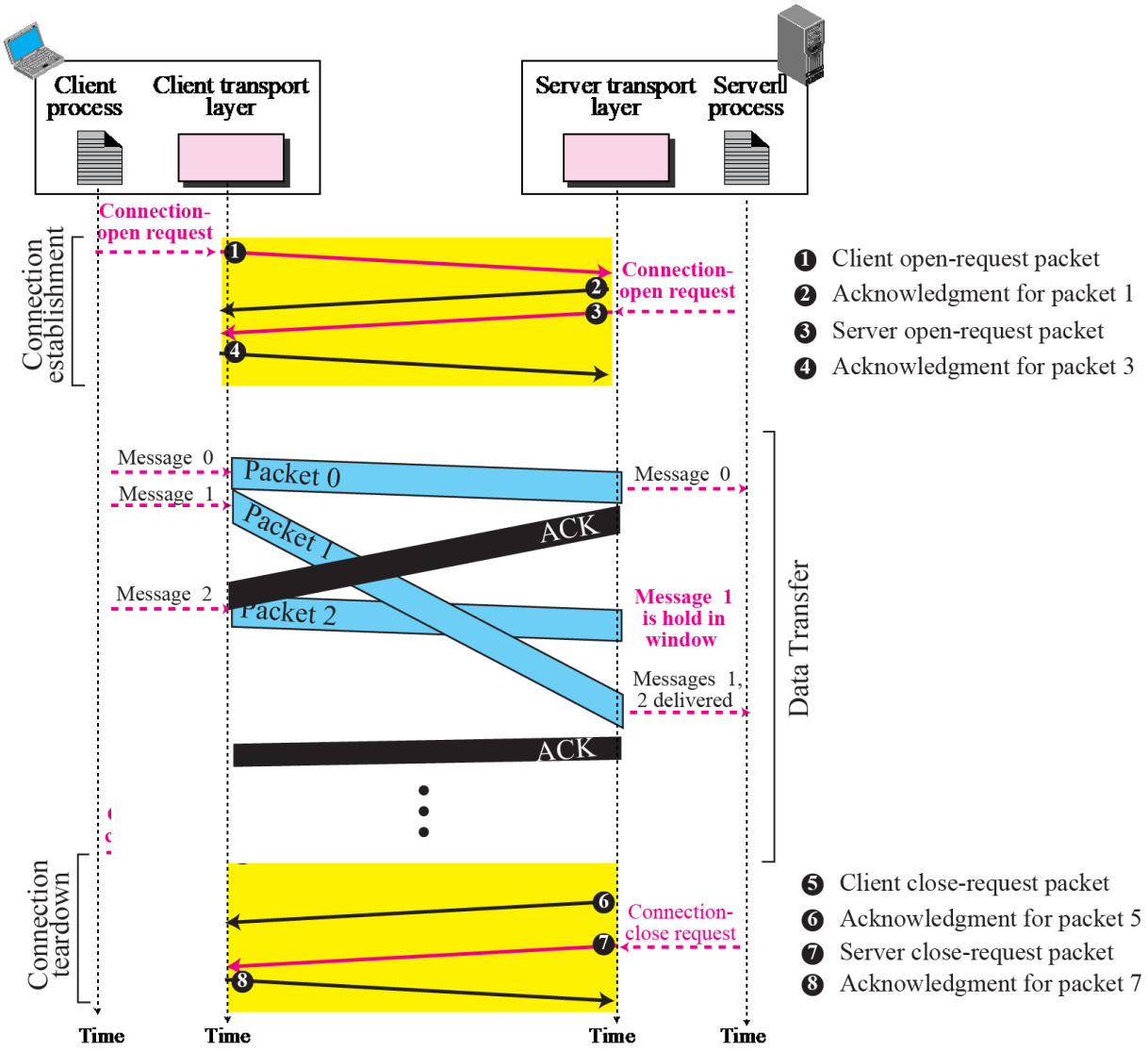


Figure 13.15 *Connectionless and connection-oriented services as FSMs*