Data Communications and Networking Fourth Edition

Forouzan

Process-to-Process Delivery: UDP, TCP, and SCTP

PROCESS-TO-PROCESS DELIVERY

The transport layer is responsible for process-to-process delivery—the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship, as we will see later.

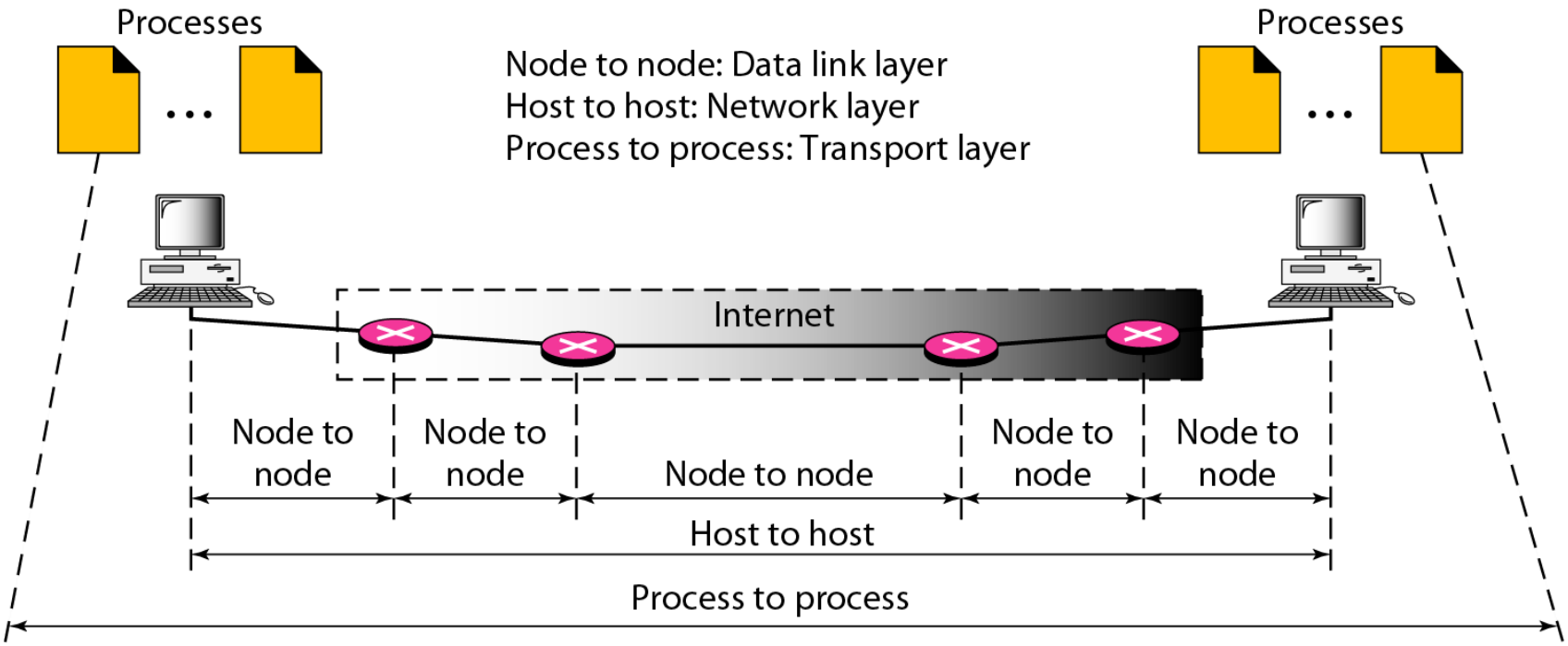
Topics discussed in this section:

Client/Server Paradigm
Multiplexing and Demultiplexing
Connectionless Versus Connection-Oriented Service
Reliable Versus Unreliable
Three Protocols

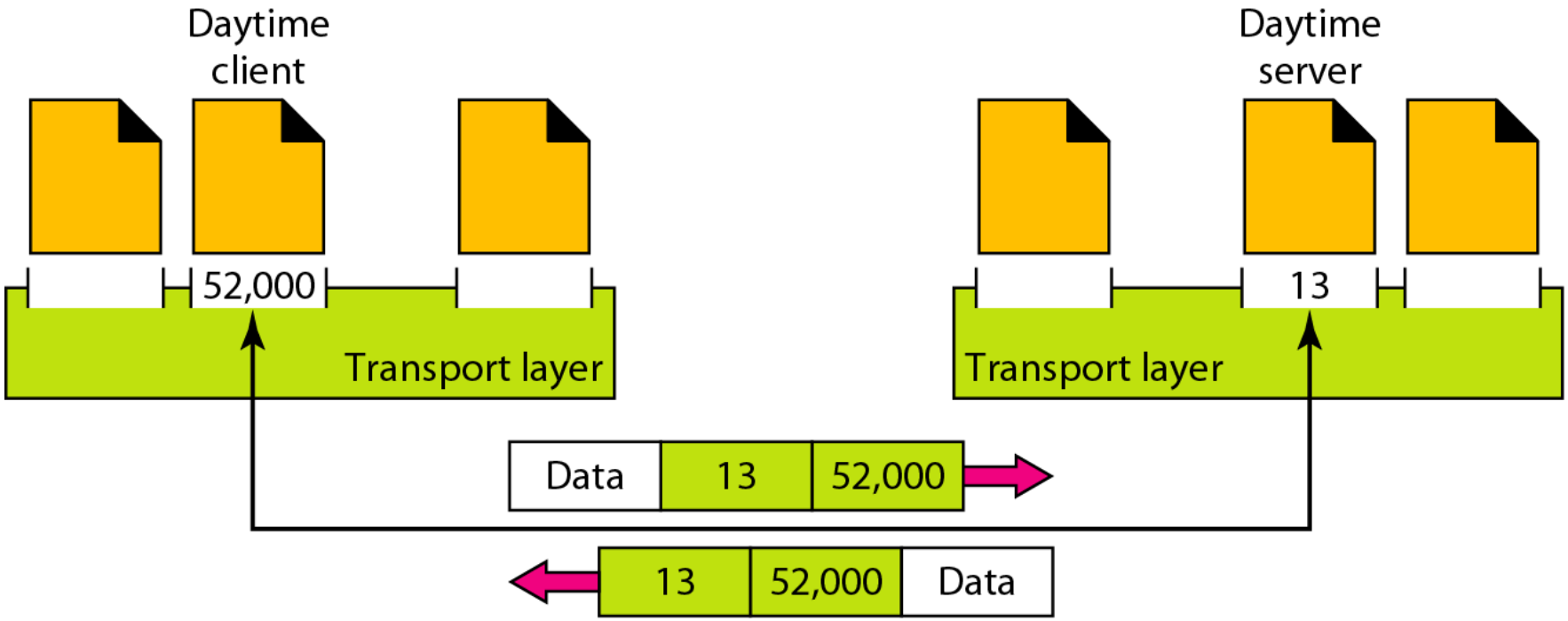
Note

The transport layer is responsible for process-to-process delivery.

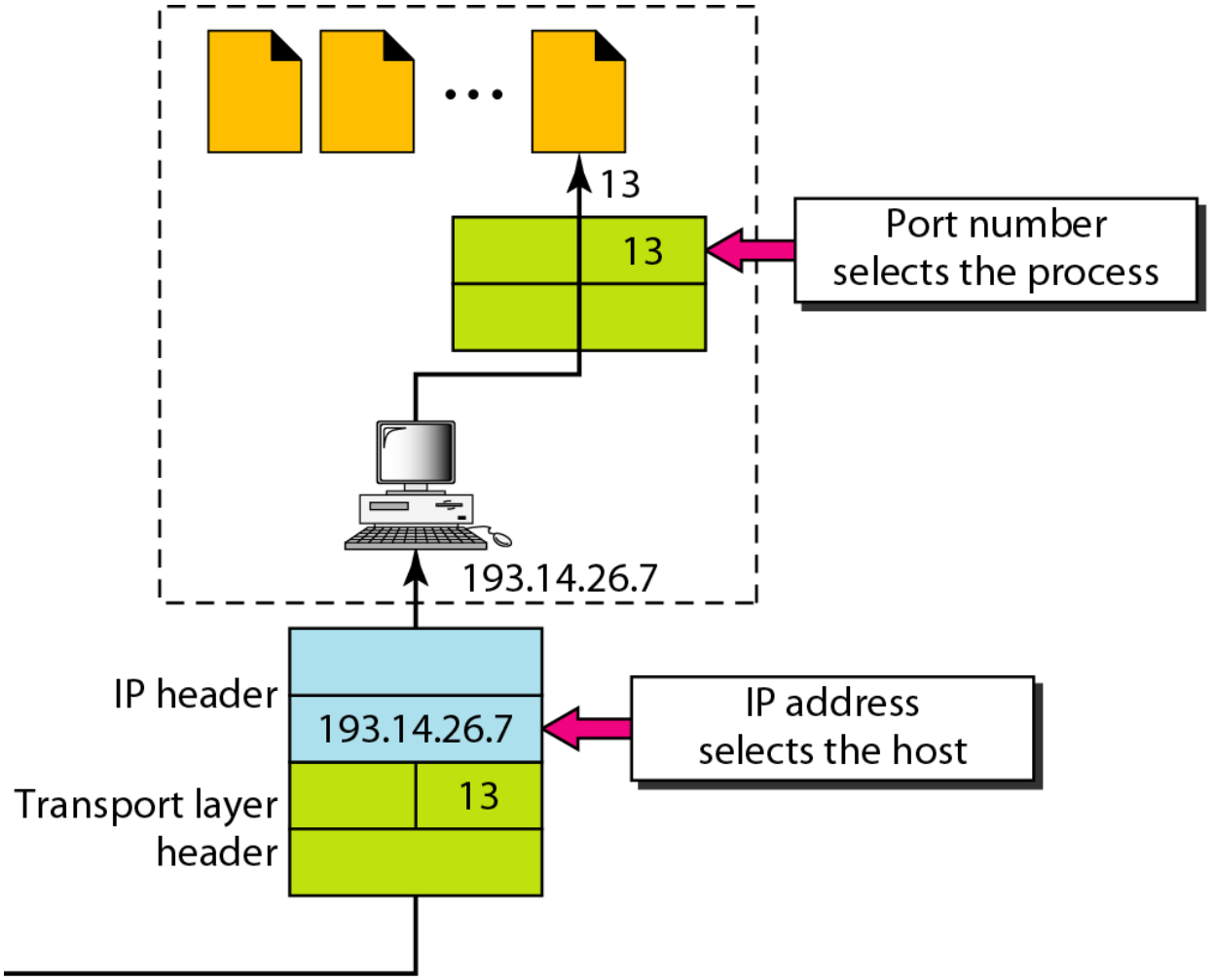
Types of data deliveries



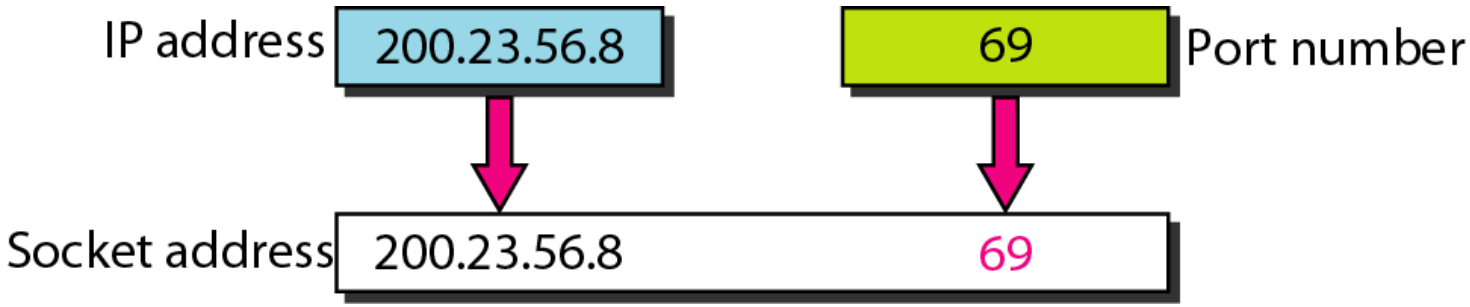
Port numbers



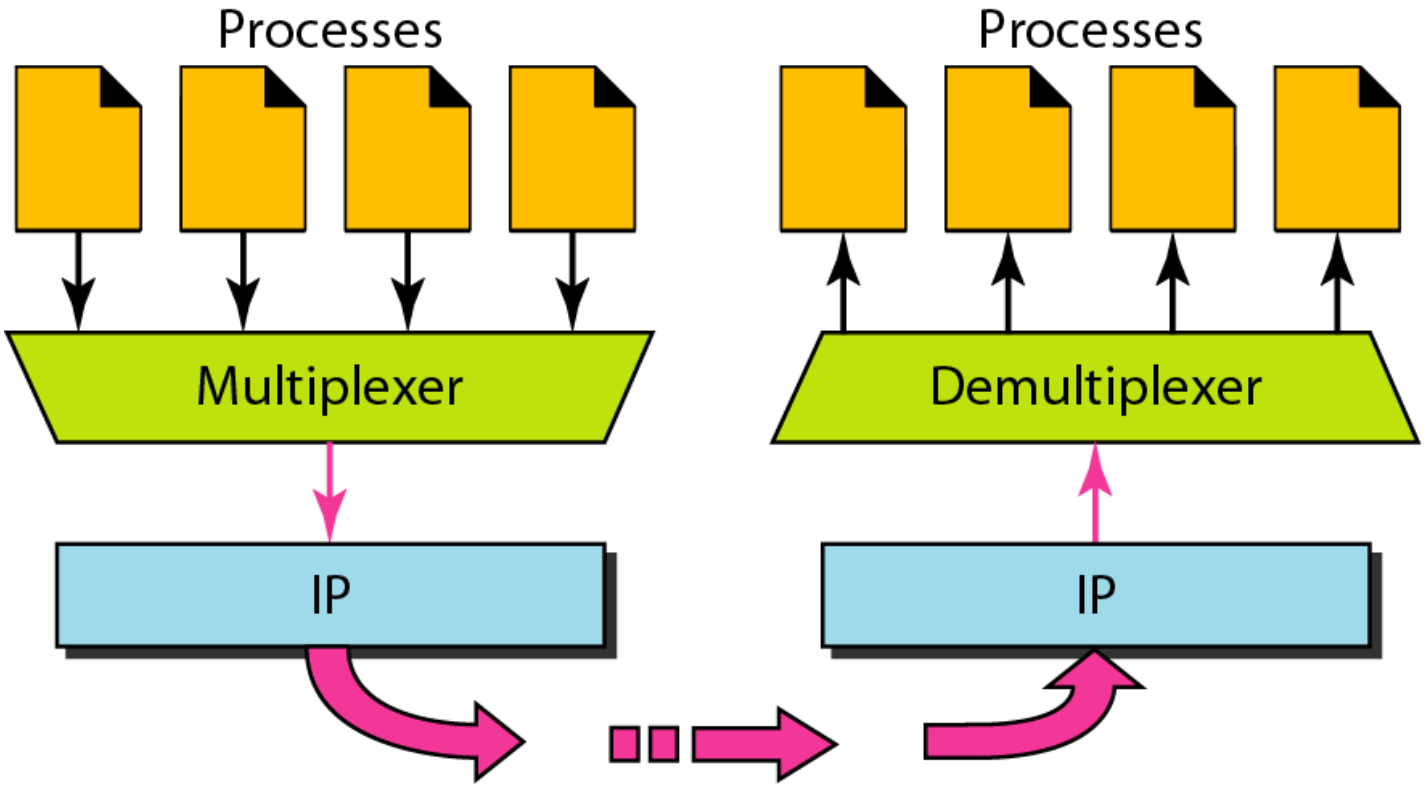
IP addresses versus port numbers



Socket address

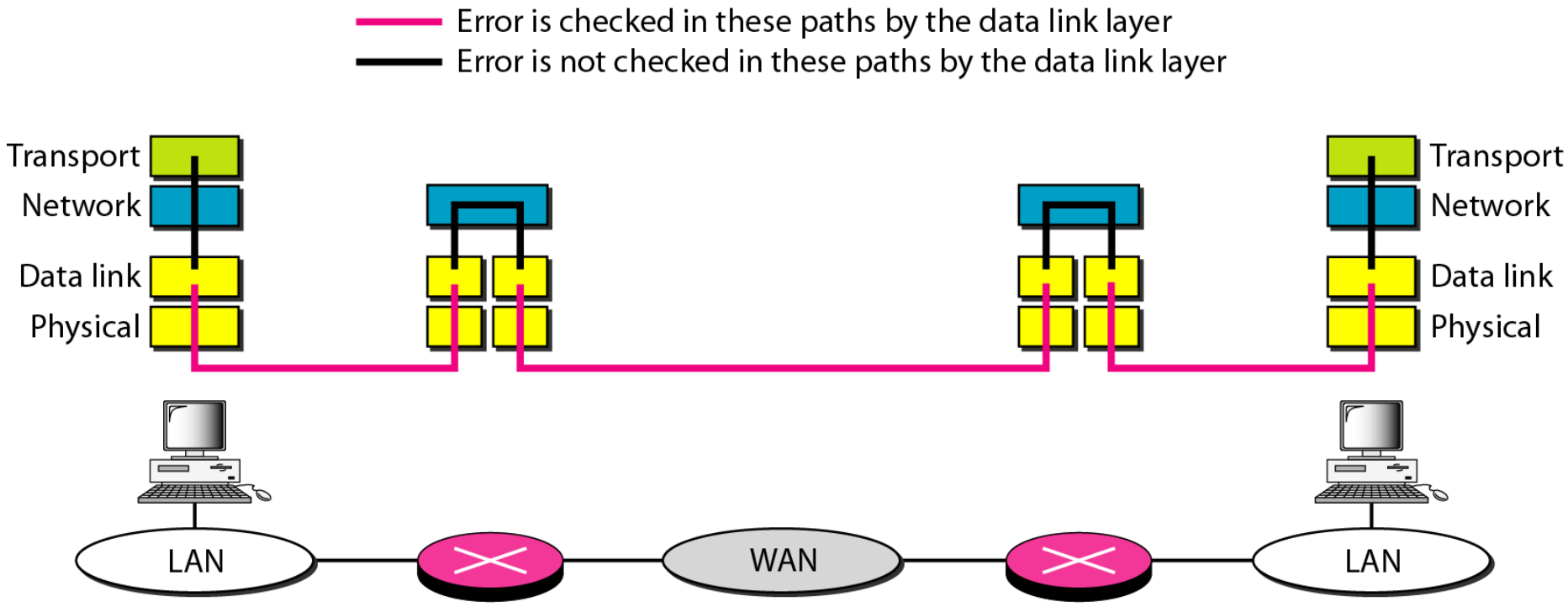


Multiplexing and demultiplexing

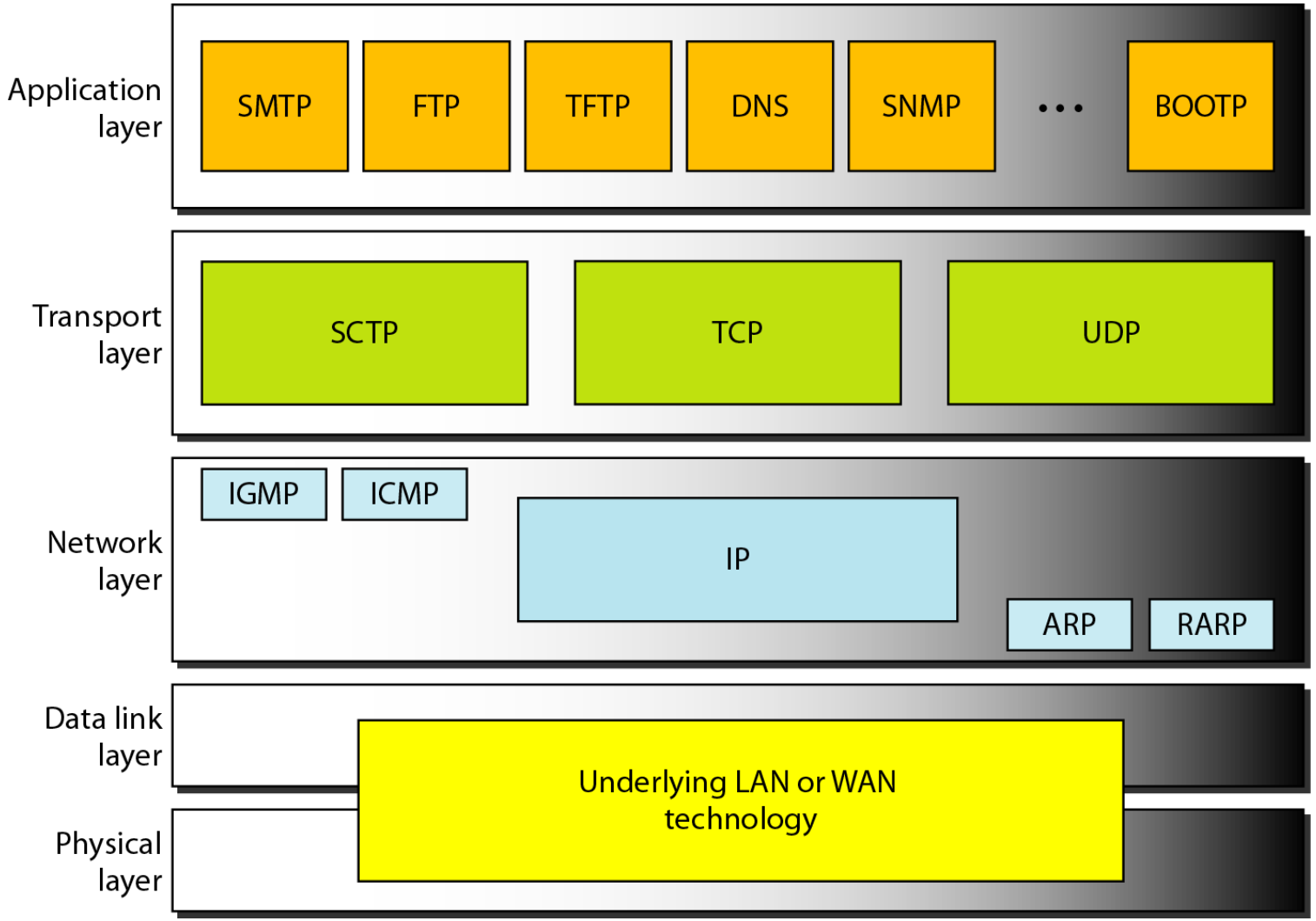


Reliable Vs Unreliable

Error control



Position of UDP, TCP, and SCTP in TCP/IP suite



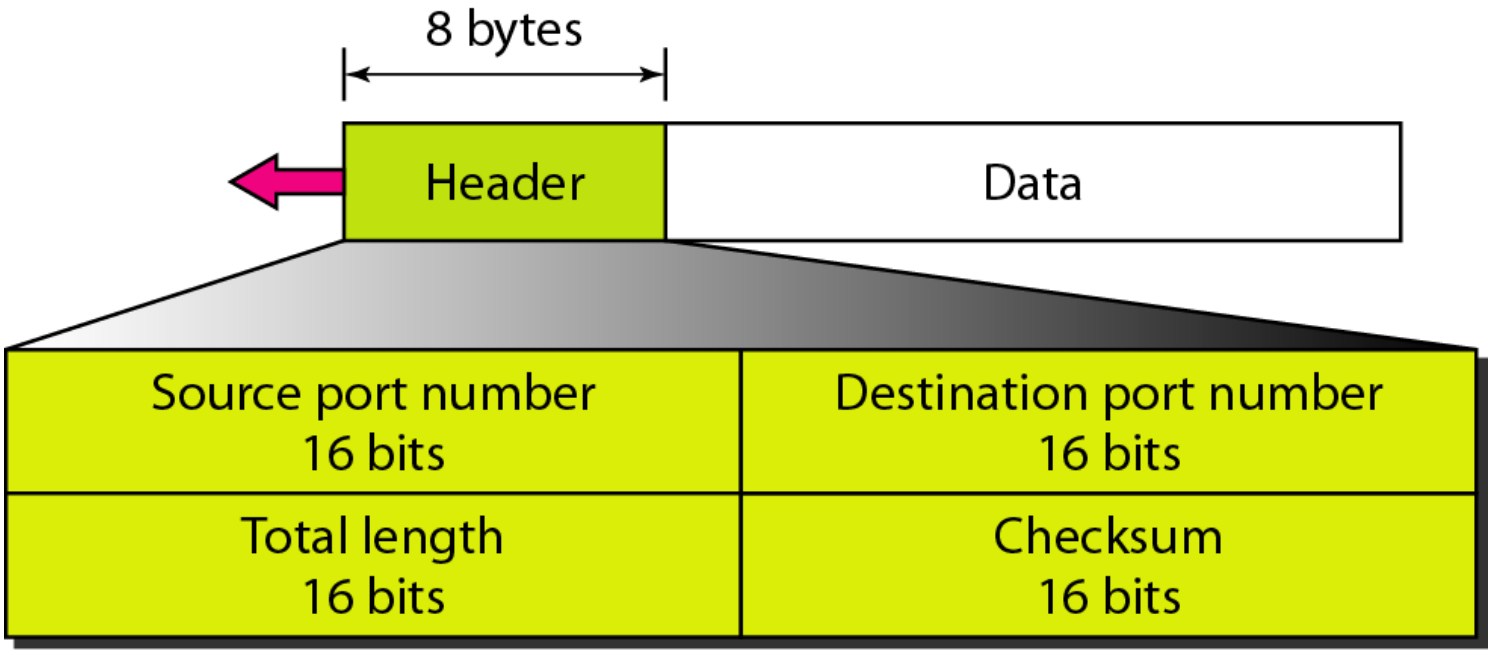
USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

Topics discussed in this section:

Well-Known Ports for UDP
User Datagram
Checksum
UDP Operation
Use of UDP

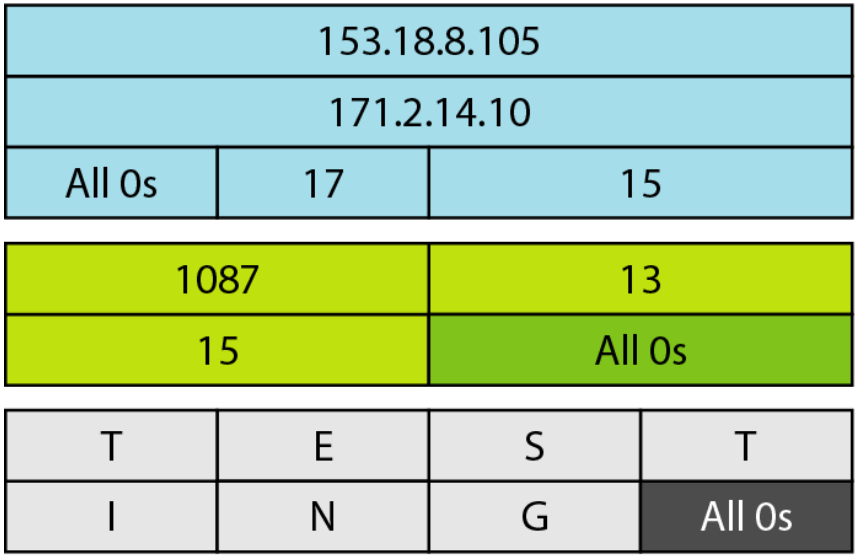
User datagram format



Note

UDP length
= IP length – IP header's length

Checksum calculation of a simple UDP user datagram



10011001 00010010 → 153.18
00001000 01101001 → 8.105
10101011 00000010 → 171.2
00001110 00001010 → 14.10
00000000 00010001 → 0 and 17
00000000 00001111 → 15
00000100 00111111 → 1087
00000000 00001101 → 13
00000000 00001111 → 15
00000000 00000000 → 0 (checksum)
01010100 01000101 → T and E
01010011 01010100 → S and T
01001001 01001110 → I and N
01000111 00000000 → G and 0 (padding)

10010110 11101011 → Sum
01101001 00010100 → Checksum

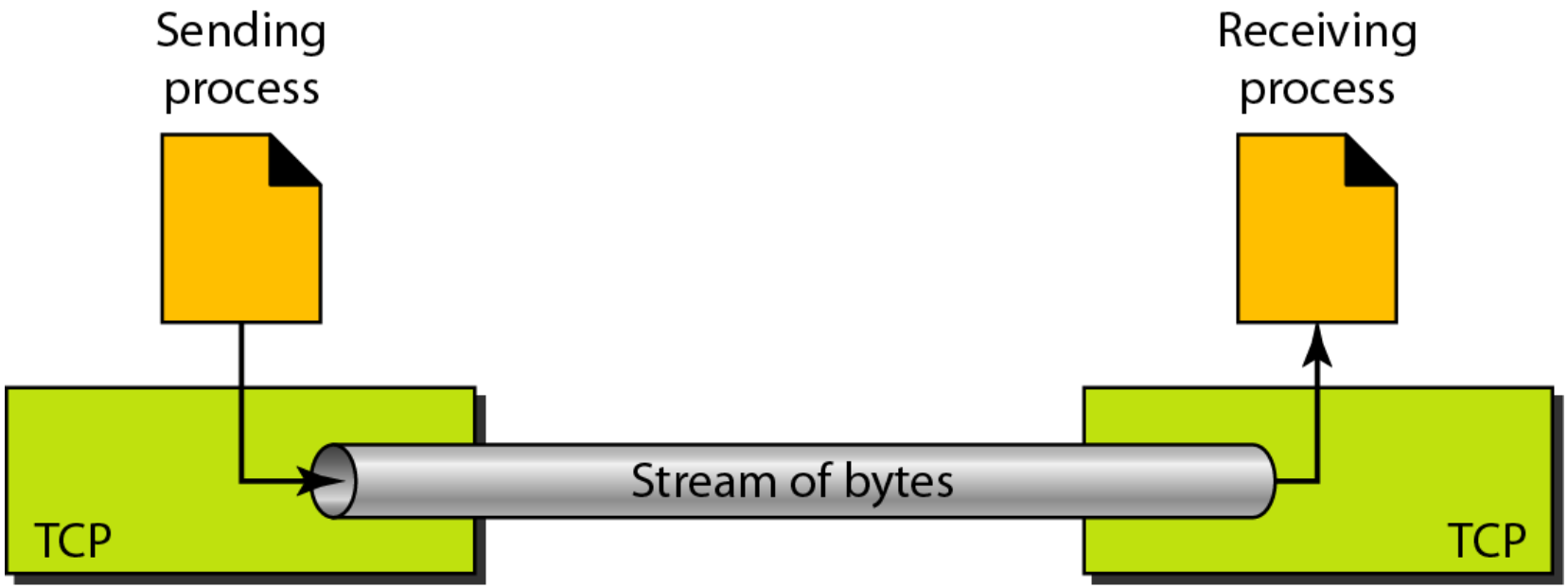
Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.

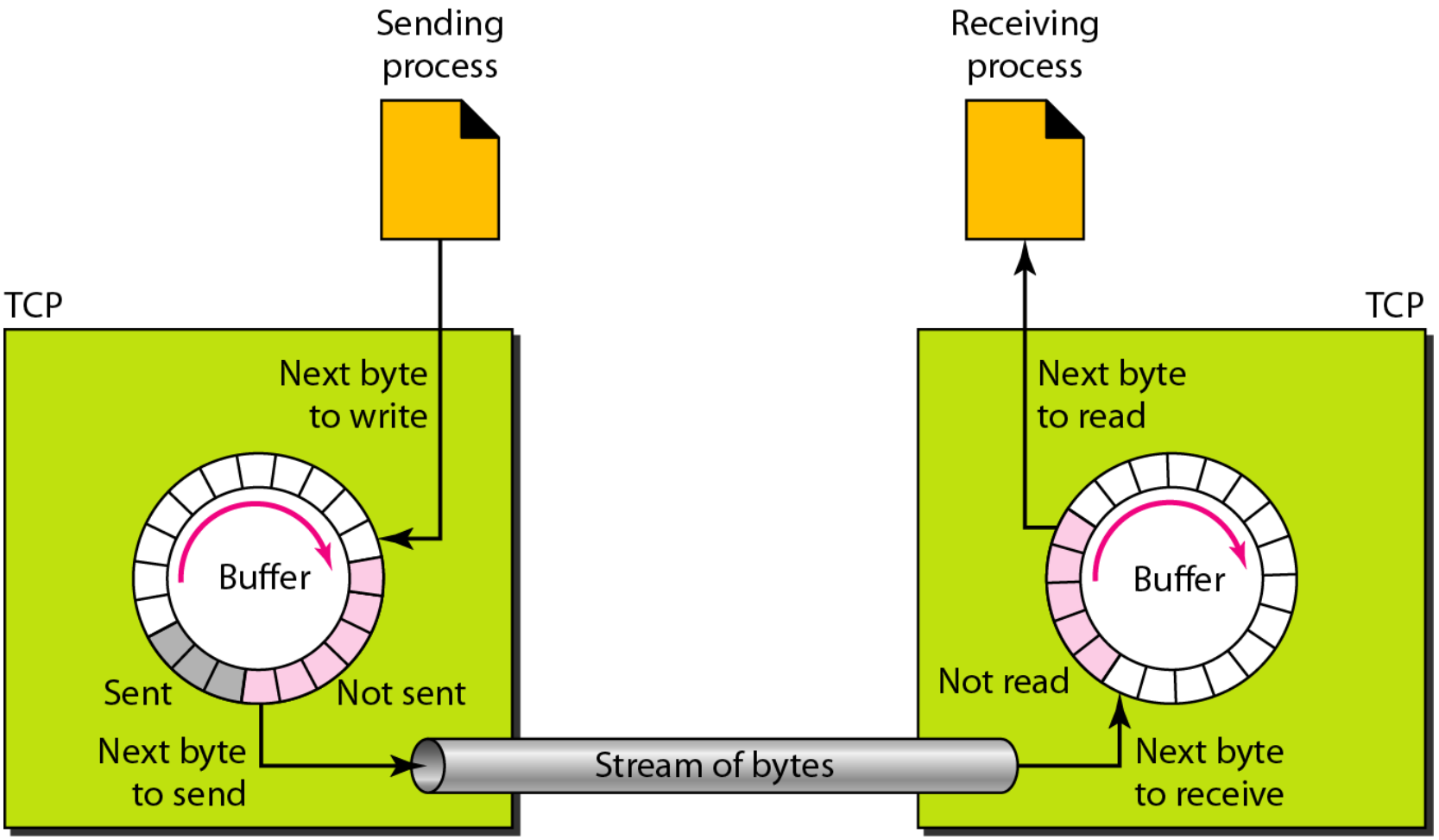
TCP Services

1. *Process to Process Communication*
2. *Stream Delivery Service*
3. *Sending and Receiving Buffers*
4. *Full Duplex Communication*
5. *Connection -Oriented Service*
6. *Reliable Service*

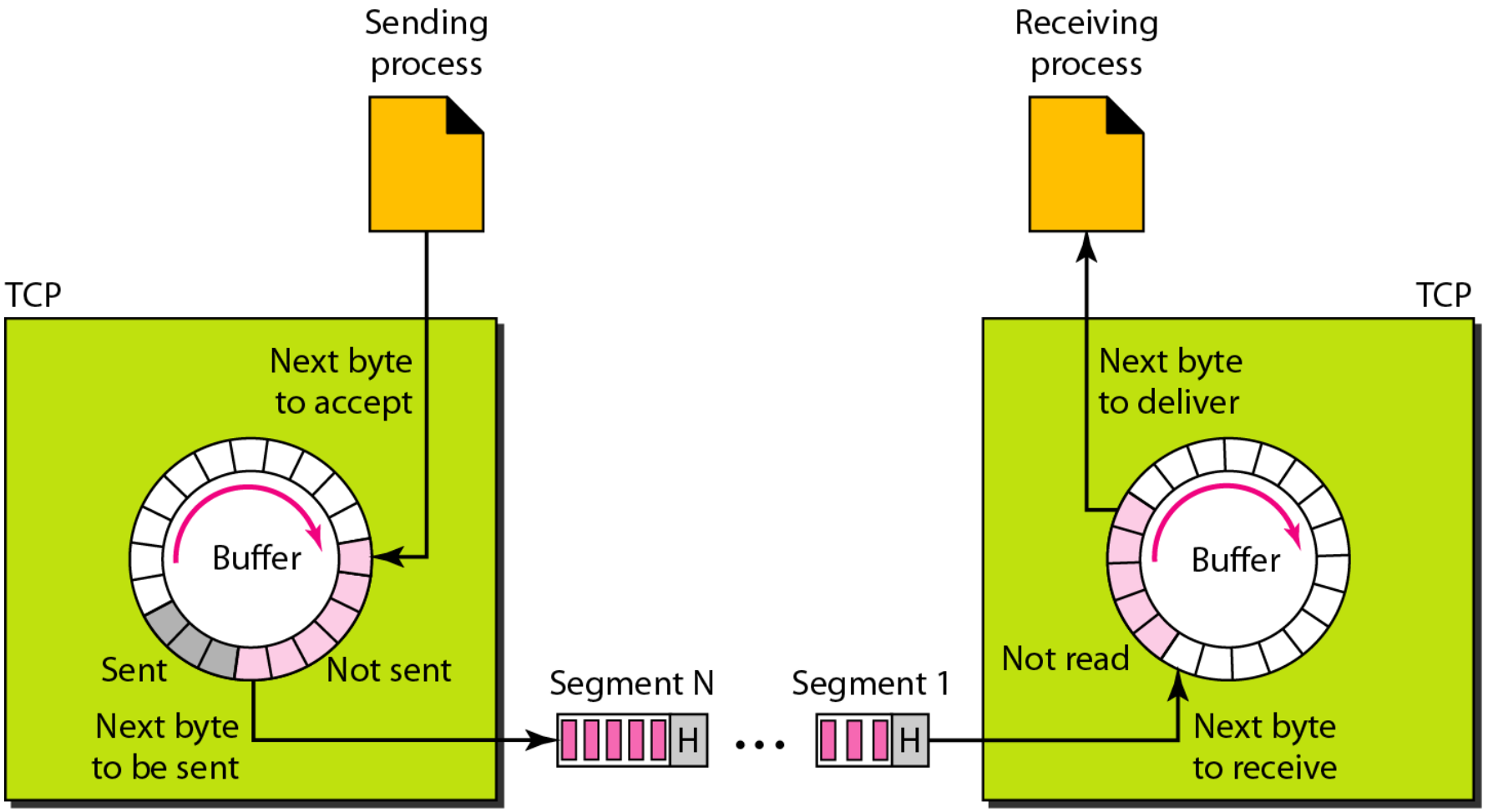
Stream delivery



Sending and receiving buffers



TCP segments



TCP Features

1. Numbering System
2. Flow Control
3. Error Control
4. Congestion Control

TCP Features

1. Numbering System

Byte Number

(TCP generates a random number b/w 0 to 2^{32} *-1)*

Sequence Number

Note

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

Example 23.3

Example

Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1000 bytes?

The following shows the sequence number for each segment:

Segment 1	→	Sequence Number: 10,001 (range: 10,001 to 11,000)
Segment 2	→	Sequence Number: 11,001 (range: 11,001 to 12,000)
Segment 3	→	Sequence Number: 12,001 (range: 12,001 to 13,000)
Segment 4	→	Sequence Number: 13,001 (range: 13,001 to 14,000)
Segment 5	→	Sequence Number: 14,001 (range: 14,001 to 15,000)

Note

The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

Note

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

TCP segment format

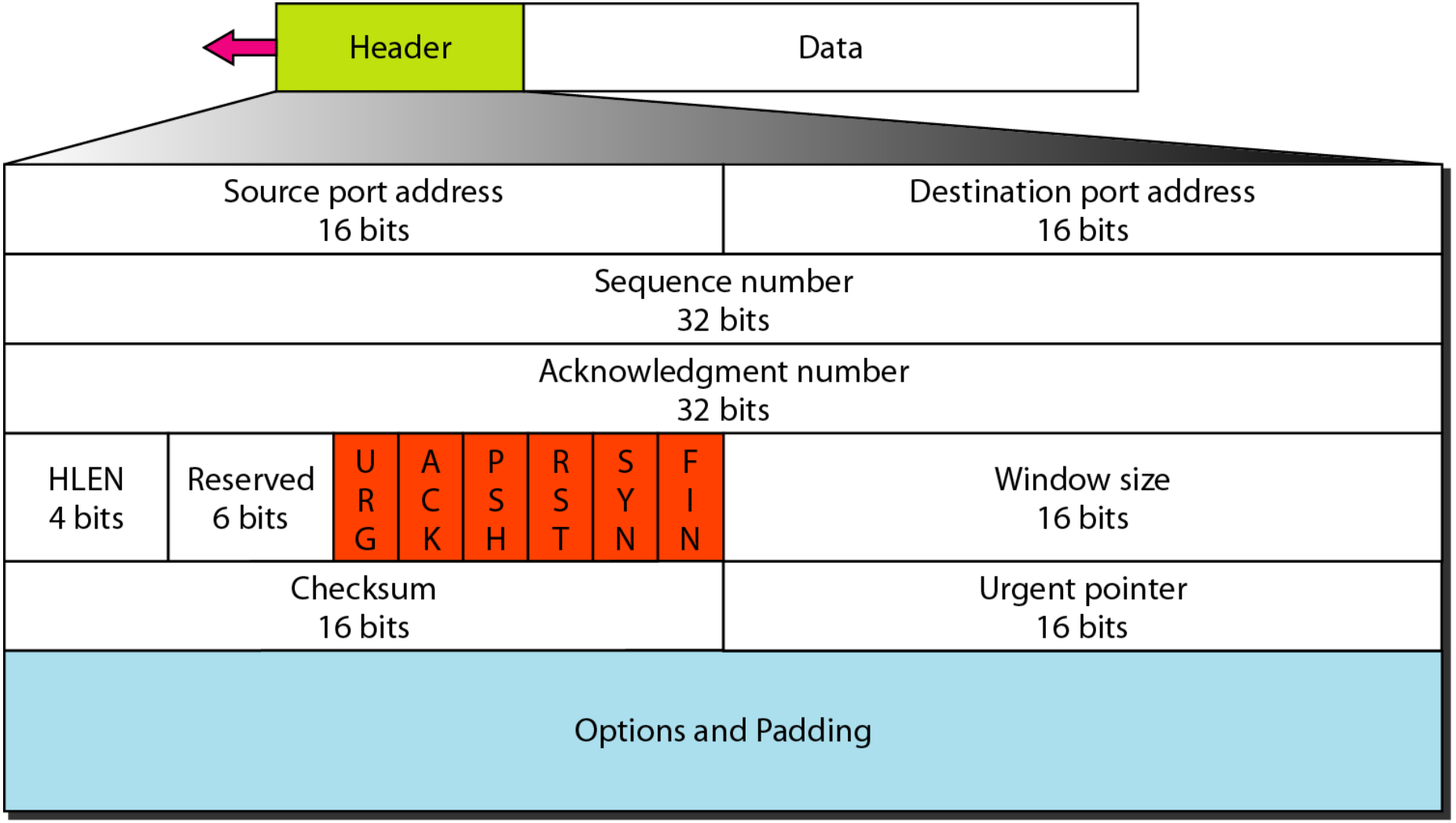


Figure 23.17 *Control field*

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push

RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

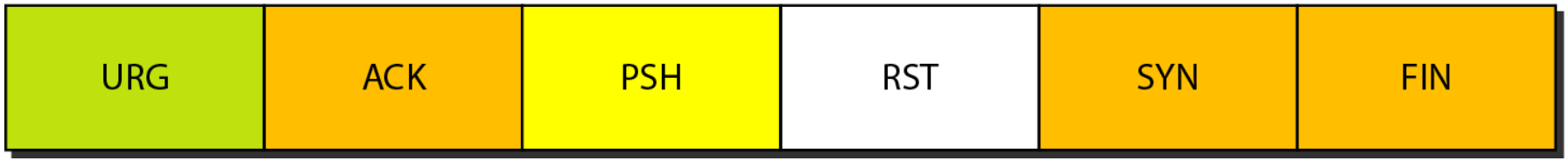
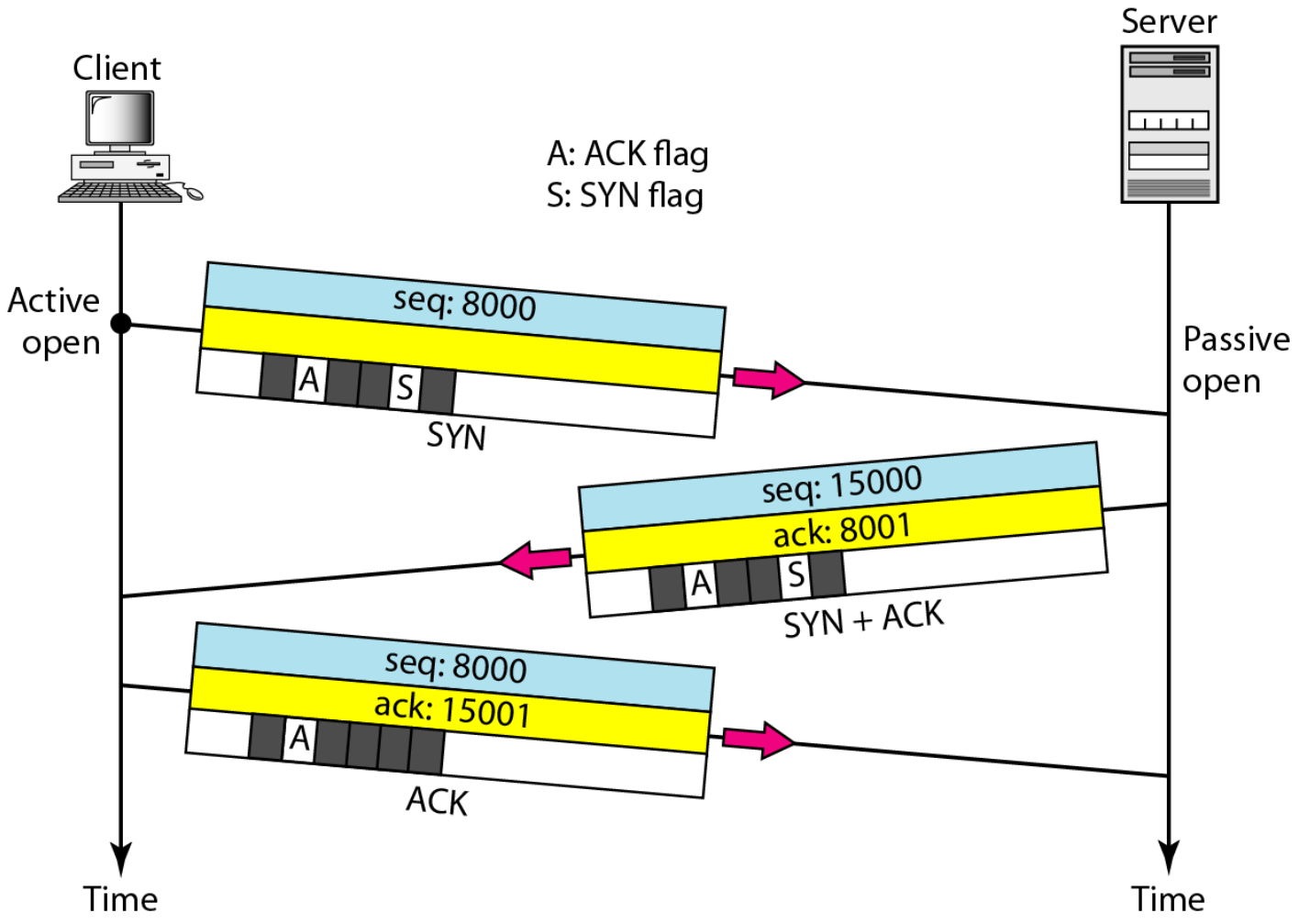


Table 23.3 ***Description of flags in the control field***

Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

Connection establishment using three-way handshaking



Note

A SYN segment cannot carry data, but it consumes one sequence number.

Note

A SYN + ACK segment cannot carry data, but does consume one sequence number.

Note

An ACK segment, if carrying no data, consumes no sequence number.

Figure 23.19 *Data transfer*

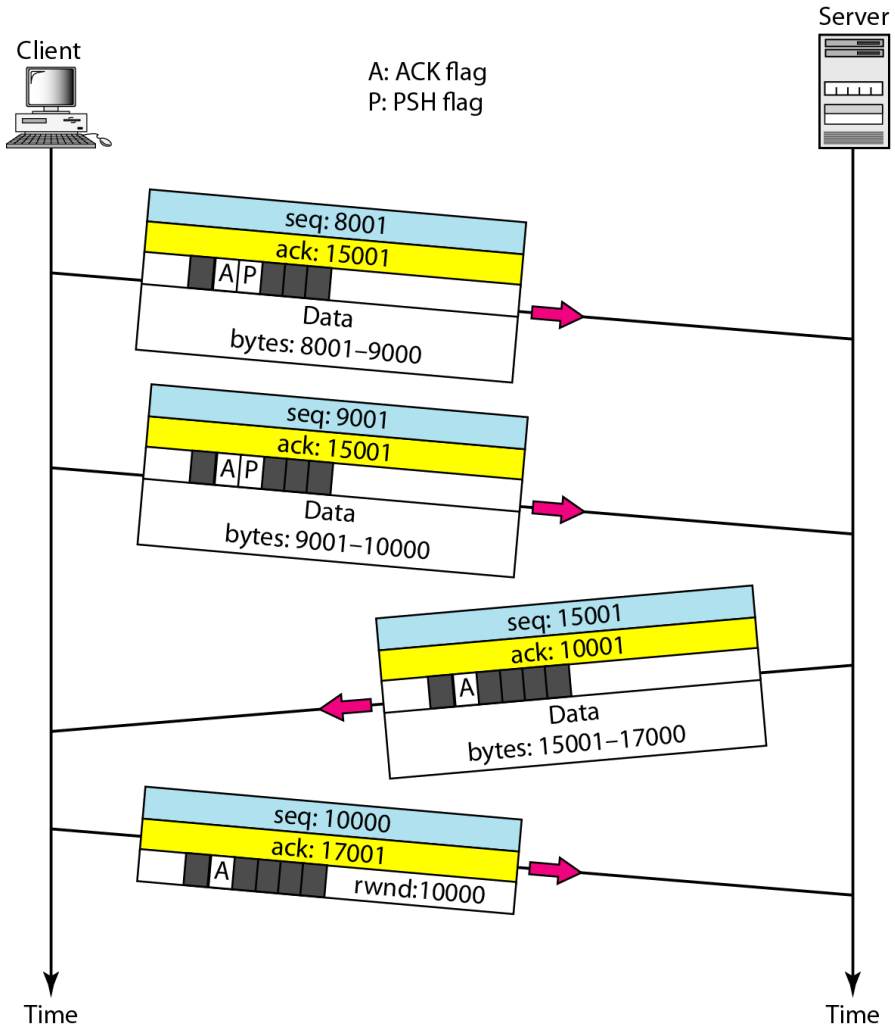
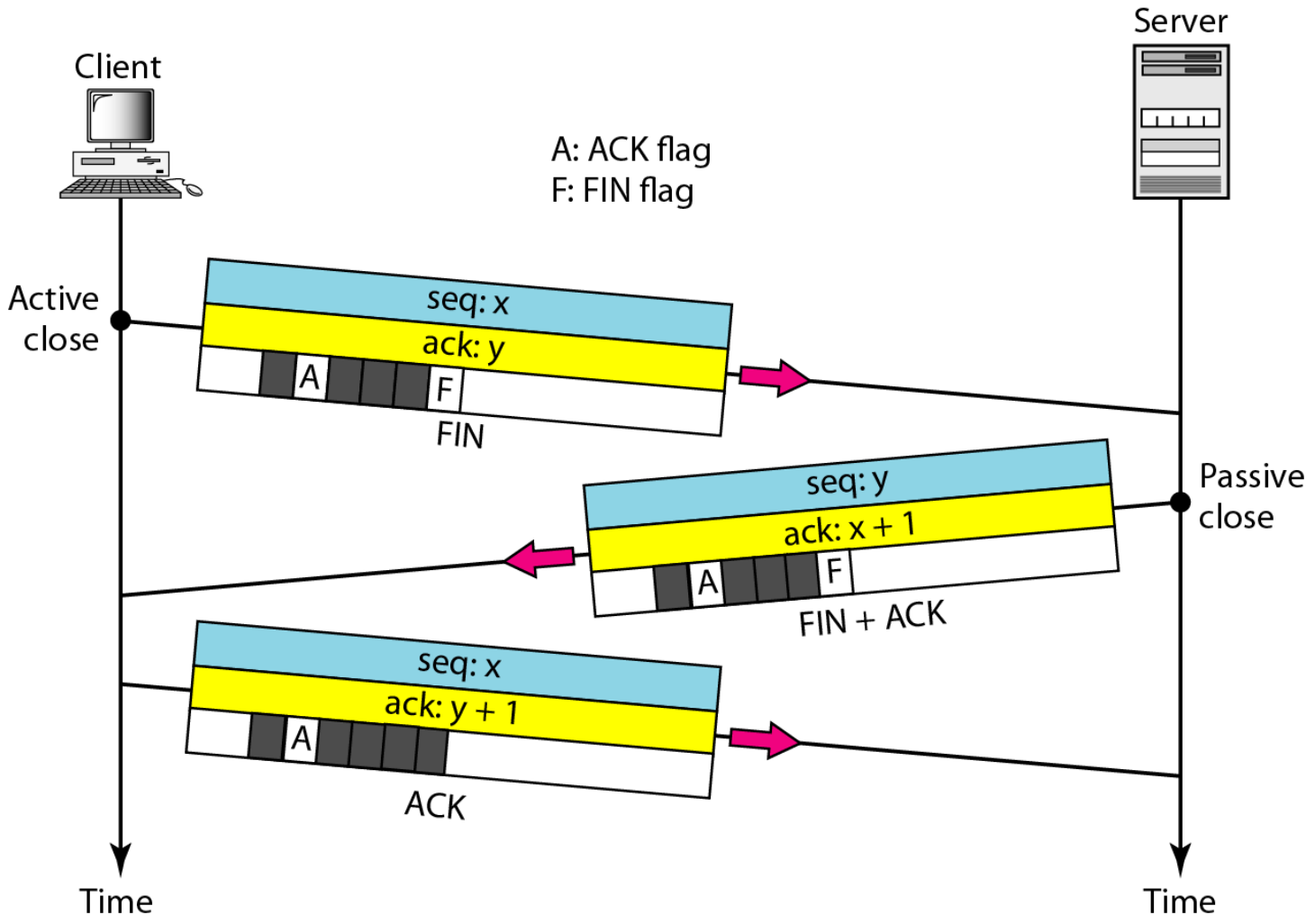


Figure 23.20 *Connection termination using three-way handshaking*



Note

The FIN segment consumes one sequence number if it does not carry data.

Note

The FIN + ACK segment consumes one sequence number if it does not carry data.

Question

Q:1 ***Explain TCP segment and UDP datagram format in detail*****.**

Q:2 Explain TCP sending and receiving buffer with diagram.

SCTP

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol. SCTP, however, is mostly designed for Internet applications that have recently been introduced. These new applications need a more sophisticated service than TCP can provide.

Topics discussed in this section:

SCTP Services and Features
Packet Format
An SCTP Association
Flow Control and Error Control

Note

SCTP is a message-oriented, reliable protocol that combines the best features of UDP and TCP.

SCTP Services

Process-to-Process Communication

Multiple Streams

Multihoming

Full-Duplex Communication

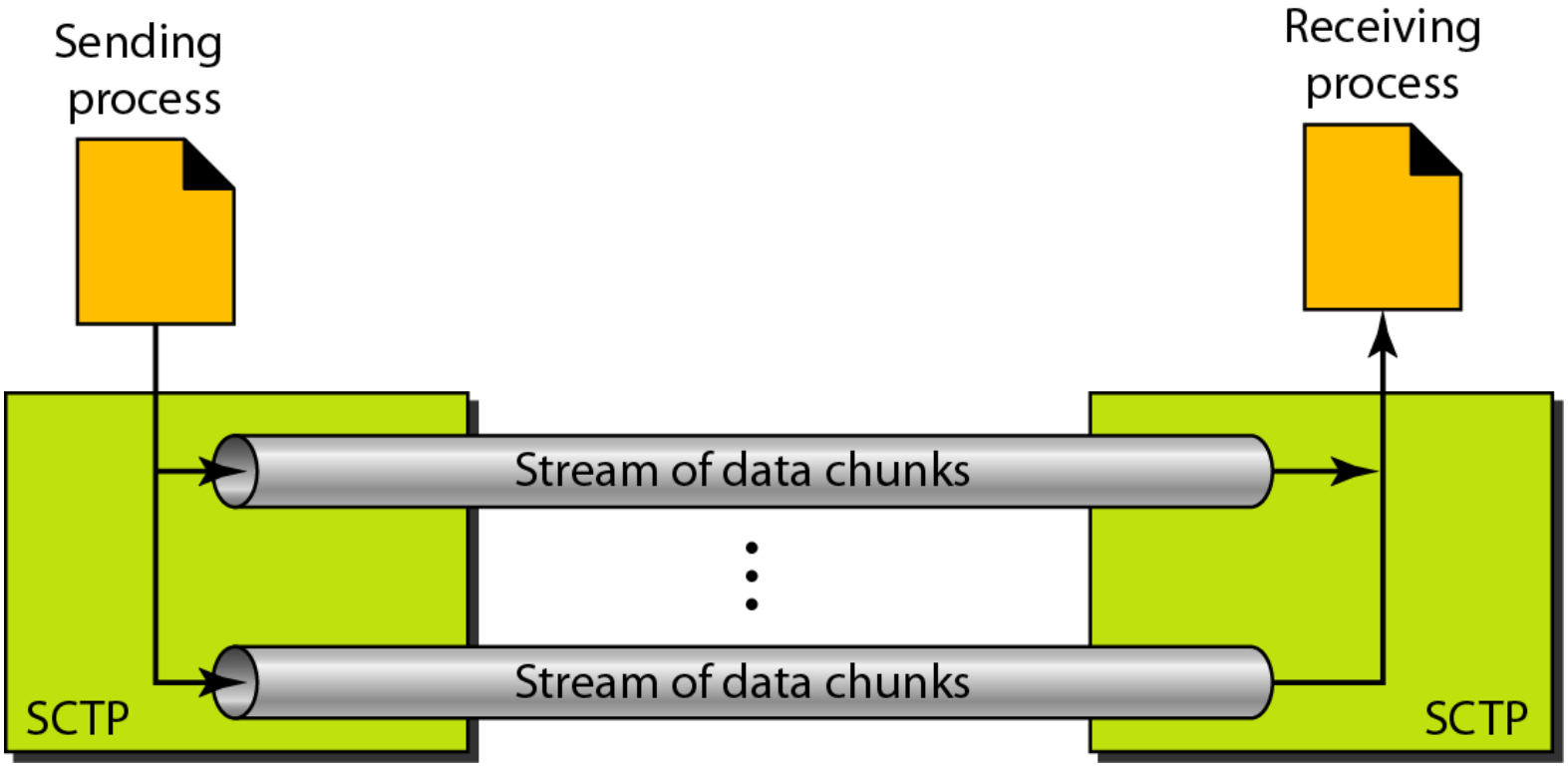
Connection-Oriented Service

Reliable Service

Table 23.4 *Some SCTP applications*

Protocol	*Port Number*	*Description*
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

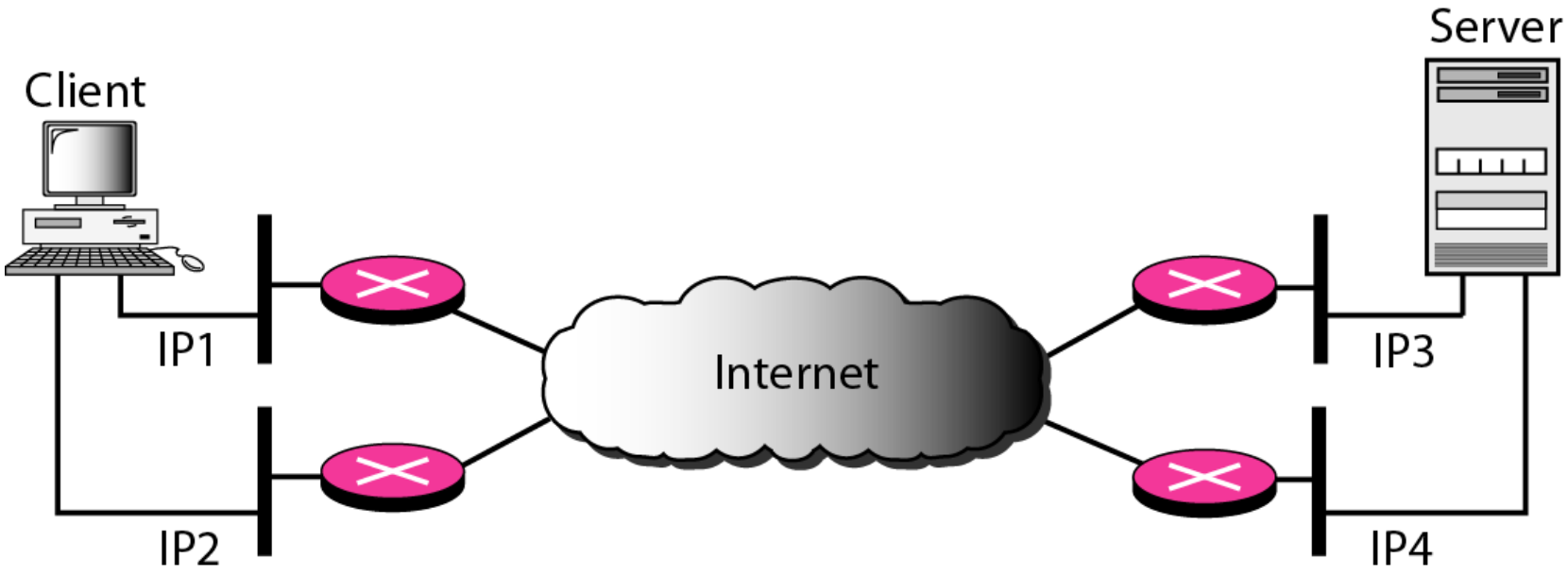
Figure 23.27 *Multiple-stream concept*



Note

An association in SCTP can involve multiple streams.

Figure 23.28 *Multihoming concept*



Note

SCTP association allows multiple IP addresses for each end.

SCTP Features

Transmission Sequence Number

Stream Identifier

Stream Sequence Number

Packets

Acknowledgment Number

Flow Control

Error Control

Congestion Control

SCTP Features

Transmission Sequence Number

The unit of data in TCP is a **byte**. Data transfer in TCP is controlled by numbering bytes by using a **sequence number**.

The unit of data in SCTP is a **DATA chunk.**

SCTP uses a transmission sequence number (TSN) to number the data chunks.

TSNs are 32 bits long and randomly initialized between 0 and $2^{32} - 1$.

Note

In SCTP, a data chunk is numbered using a TSN.

TSN: Transmission Sequence Number

SCTP Features

Stream Identifier

In TCP, there is only one stream in each connection.

In SCTP, there may be several streams in each association.

Each stream in SCTP needs to be identified by using a stream identifier (SI).

Note

To distinguish between different streams, SCTP uses an SI.

SCTP Features

Stream Sequence Number

When a data chunk arrives at the destination SCTP, it is delivered to the appropriate stream and in the proper order.

SCTP defines each data chunk in each stream with a stream sequence number (SSN).

Note

To distinguish between different data chunks belonging to the same stream, SCTP uses SSNs.

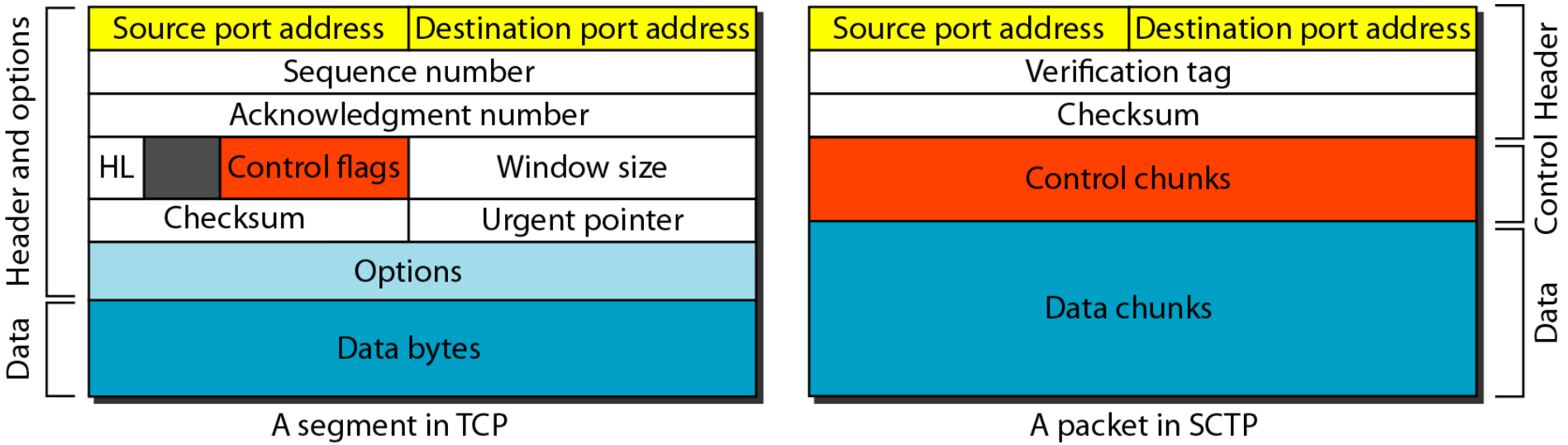
SSN : Stream sequence number

SCTP Features

Packets

TCP has segments; SCTP has packets.

Figure 23.29 *Comparison between a TCP segment and an SCTP packet*



Note

In SCTP, control information and data information are carried in separate chunks.

Example

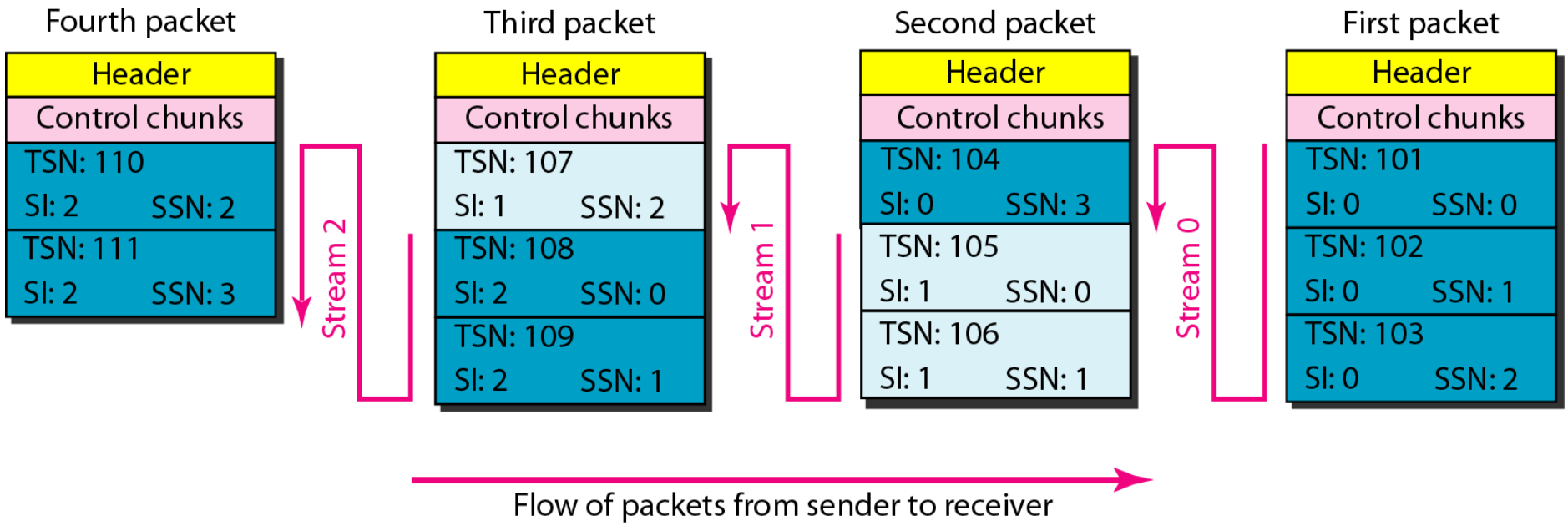
let us suppose that process A needs to send 11 messages to process B in three streams.

The first four messages are in **the first stream, the second three messages are in the second stream, and the last four messages are in the third stream.**

we assume that each message fits into one data chunk. Therefore, we have 11 data chunks in three streams.

We also assume that the network allows only three data chunks per packet, which
means that we need.

Figure 23.30 *Packet, data chunks, and streams*



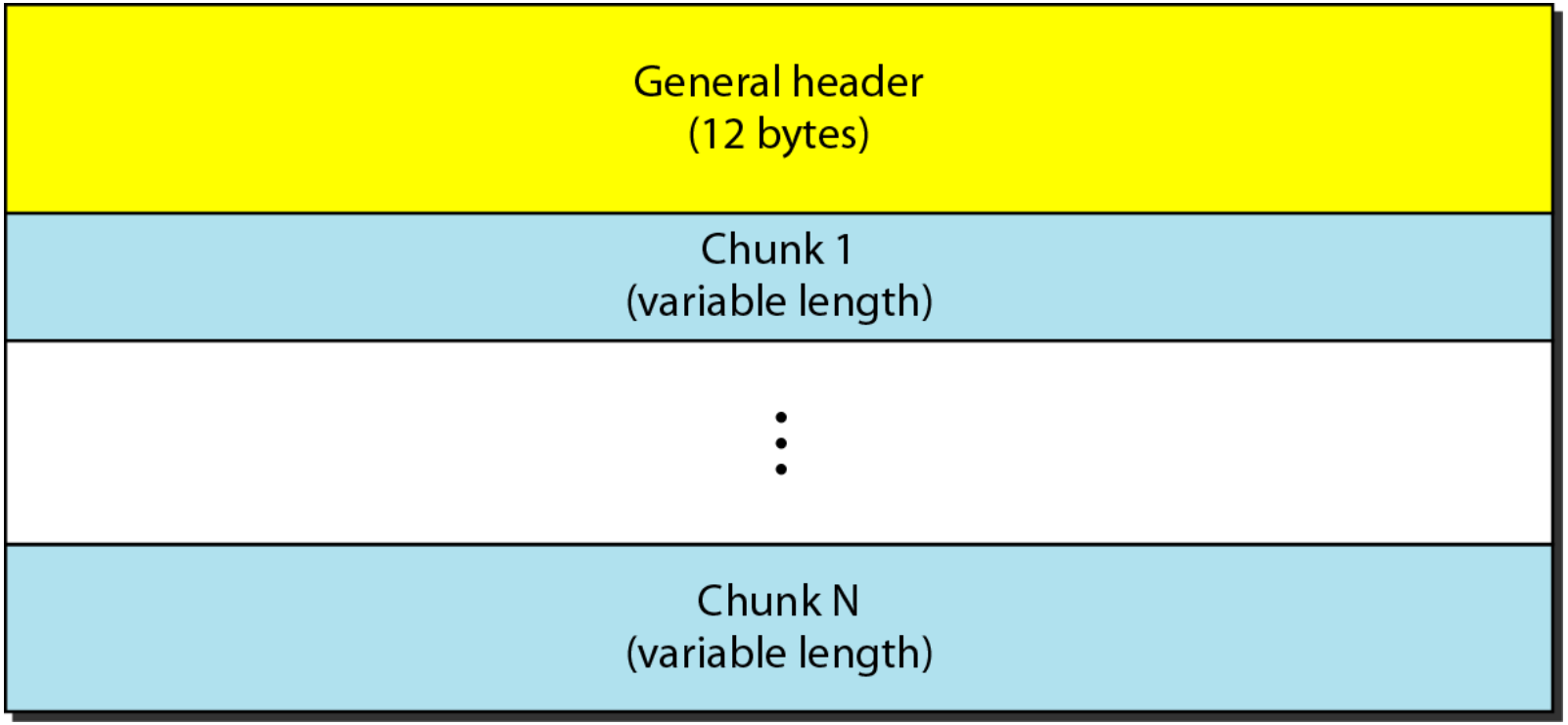
Note

Data chunks are identified by three items: TSN, SI, and SSN. TSN is a cumulative number identifying the association; SI defines the stream; SSN defines the chunk in a stream.

Note

In SCTP, acknowledgment numbers are used to acknowledge only data chunks; control chunks are acknowledged by other control chunks if necessary.

Figure 23.31 *SCTP packet format*



Note

In an SCTP packet, control chunks come before data chunks.

Figure 23.32 *General header*

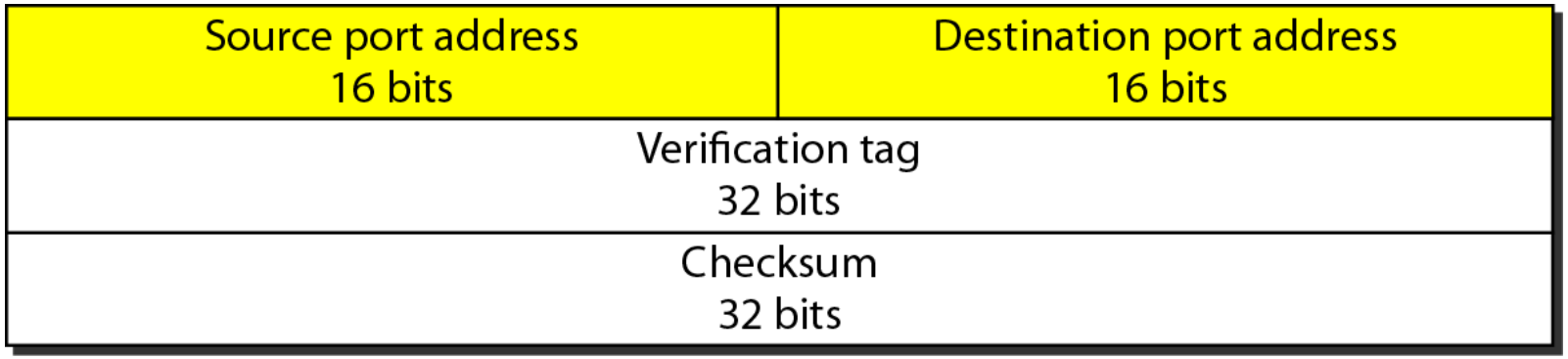


Table 23.5 *Chunks*

Type	*Chunk*	*Description*
0	**DATA**	User data
1	**INIT**	Sets up an association
2	**INIT ACK**	Acknowledges INIT chunk
3	**SACK**	Selective acknowledgment
4	**HEARTBEAT**	Probes the peer for liveliness
5	**HEARTBEAT ACK**	Acknowledges HEARTBEAT chunk
6	**ABORT**	Aborts an association
7	**SHUTDOWN**	Terminates an association
8	**SHUTDOWN ACK**	Acknowledges SHUTDOWN chunk
9	**ERROR**	Reports errors without shutting down
10	**COOKIE ECHO**	Third packet in association establishment
11	**COOKIE ACK**	Acknowledges COOKIE ECHO chunk
14	**SHUTDOWN COMPLETE**	Third packet in association termination
192	**FORWARD TSN**	For adjusting cumulative TSN

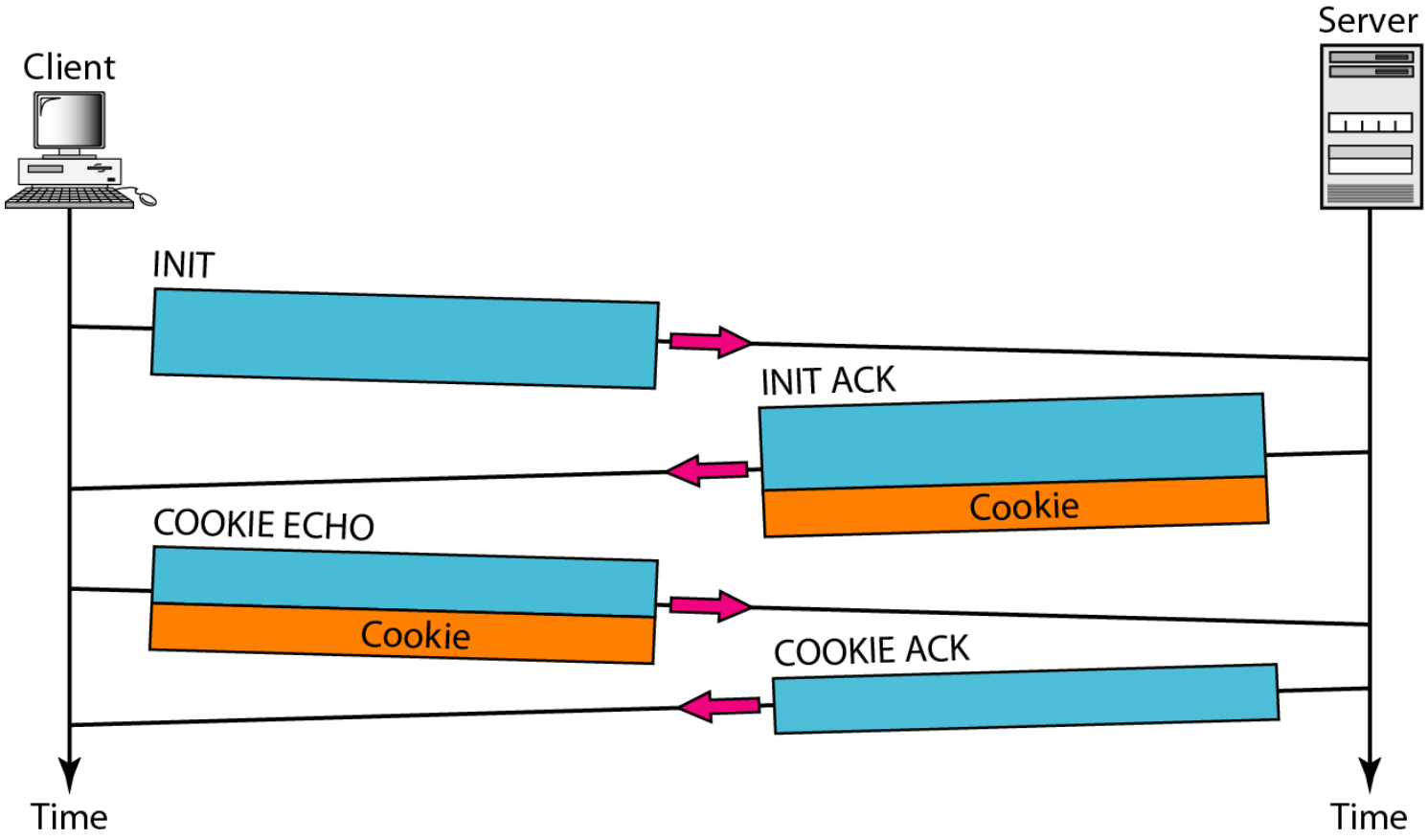
Note

A connection in SCTP is called an association.

Note

No other chunk is allowed in a packet carrying an INIT or INIT ACK chunk. A COOKIE ECHO or a COOKIE ACK chunk can carry data chunks.

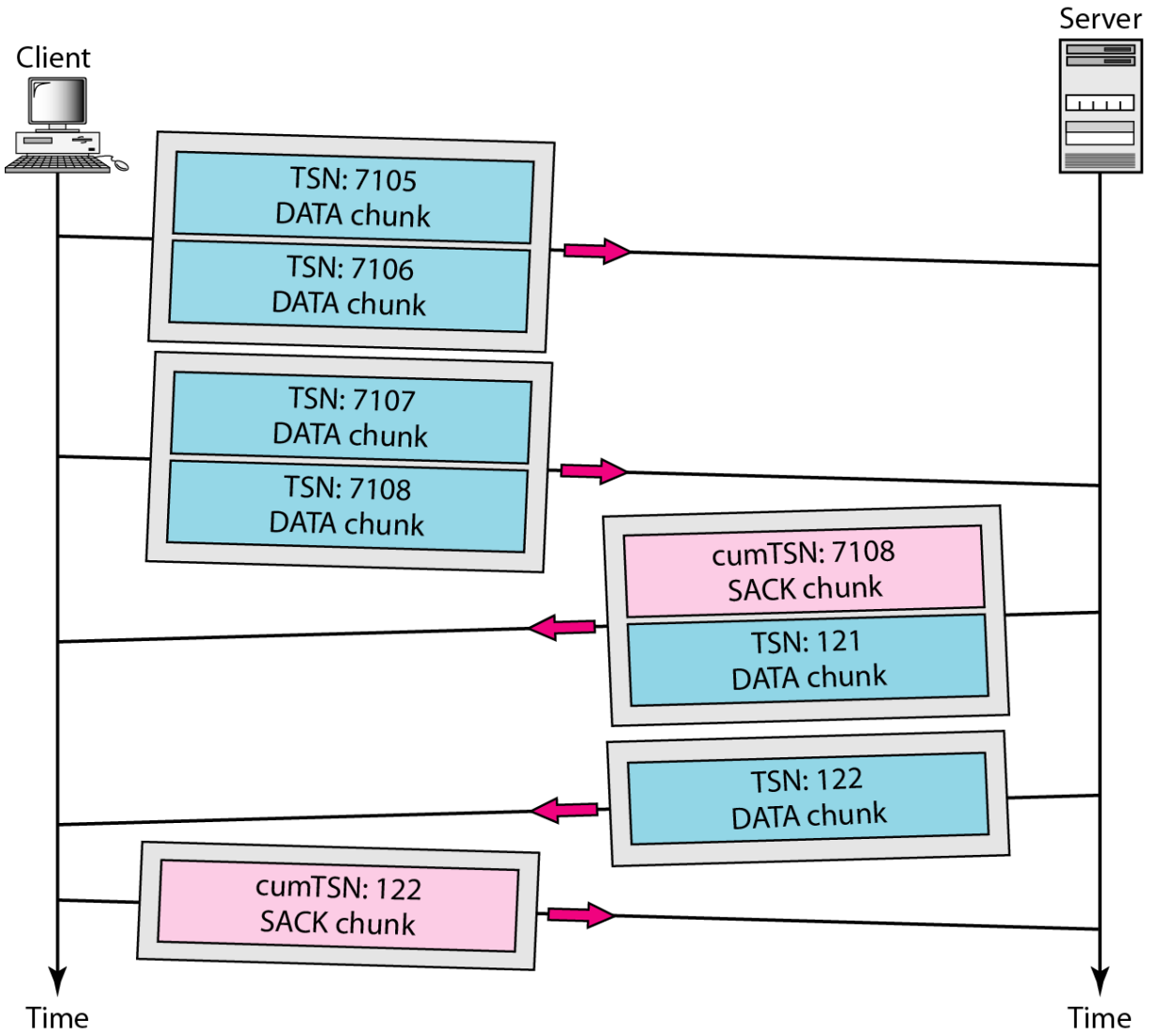
Figure 23.33 *Four-way handshaking*



Note

In SCTP, only DATA chunks consume TSNs; DATA chunks are the only chunks that are acknowledged.

Figure 23.34 *Simple data transfer*



Note

The acknowledgment in SCTP defines the cumulative TSN, the TSN of the last data chunk received in order.

Figure 23.35 *Association termination*

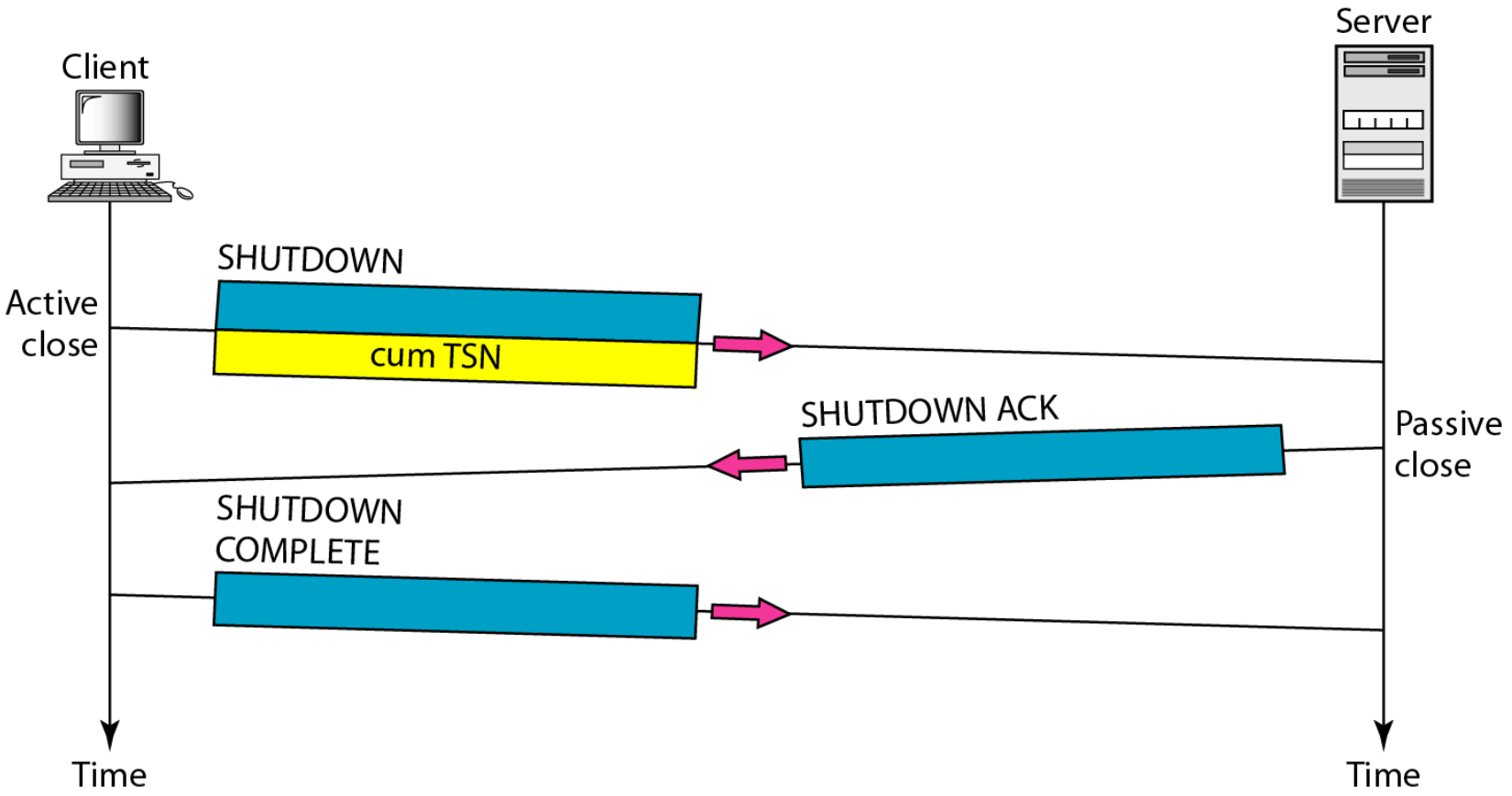


Figure 23.36 *Flow control, receiver site*

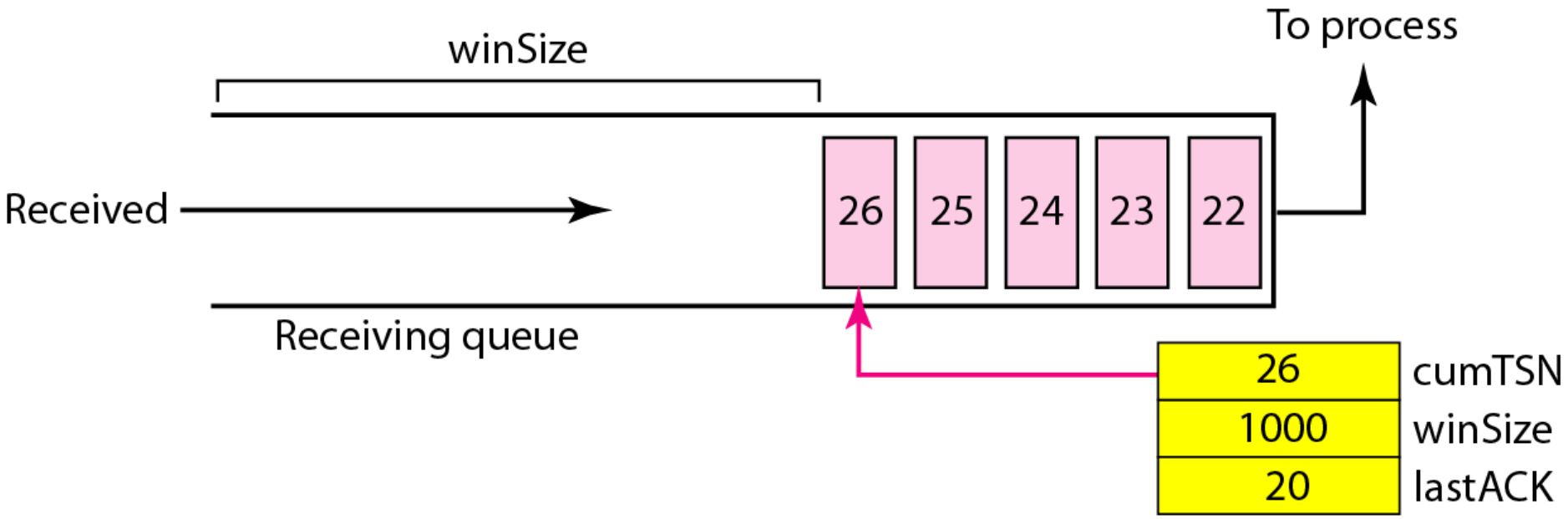


Figure 23.37 *Flow control, sender site*

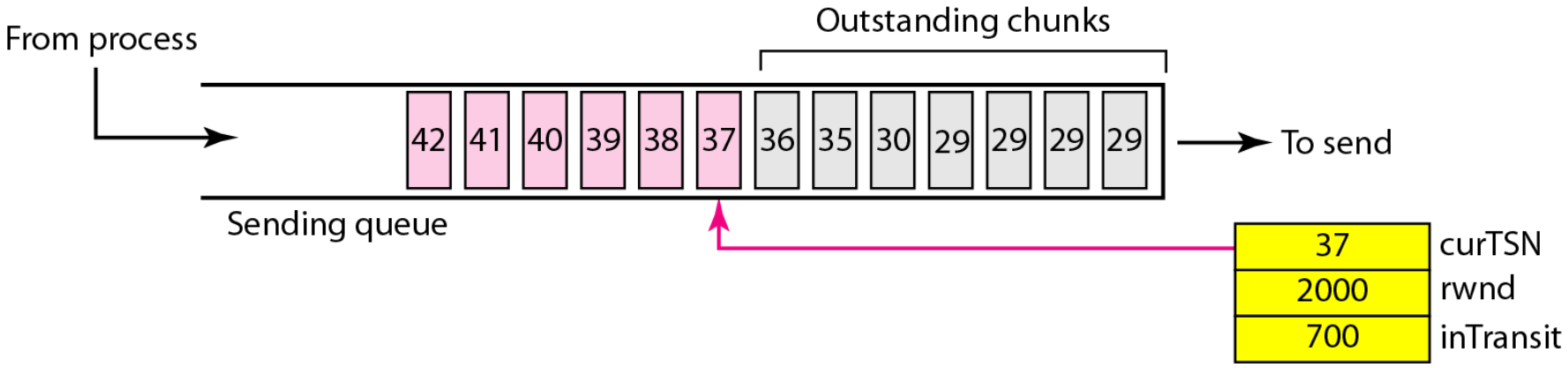


Figure 23.38 *Flow control scenario*

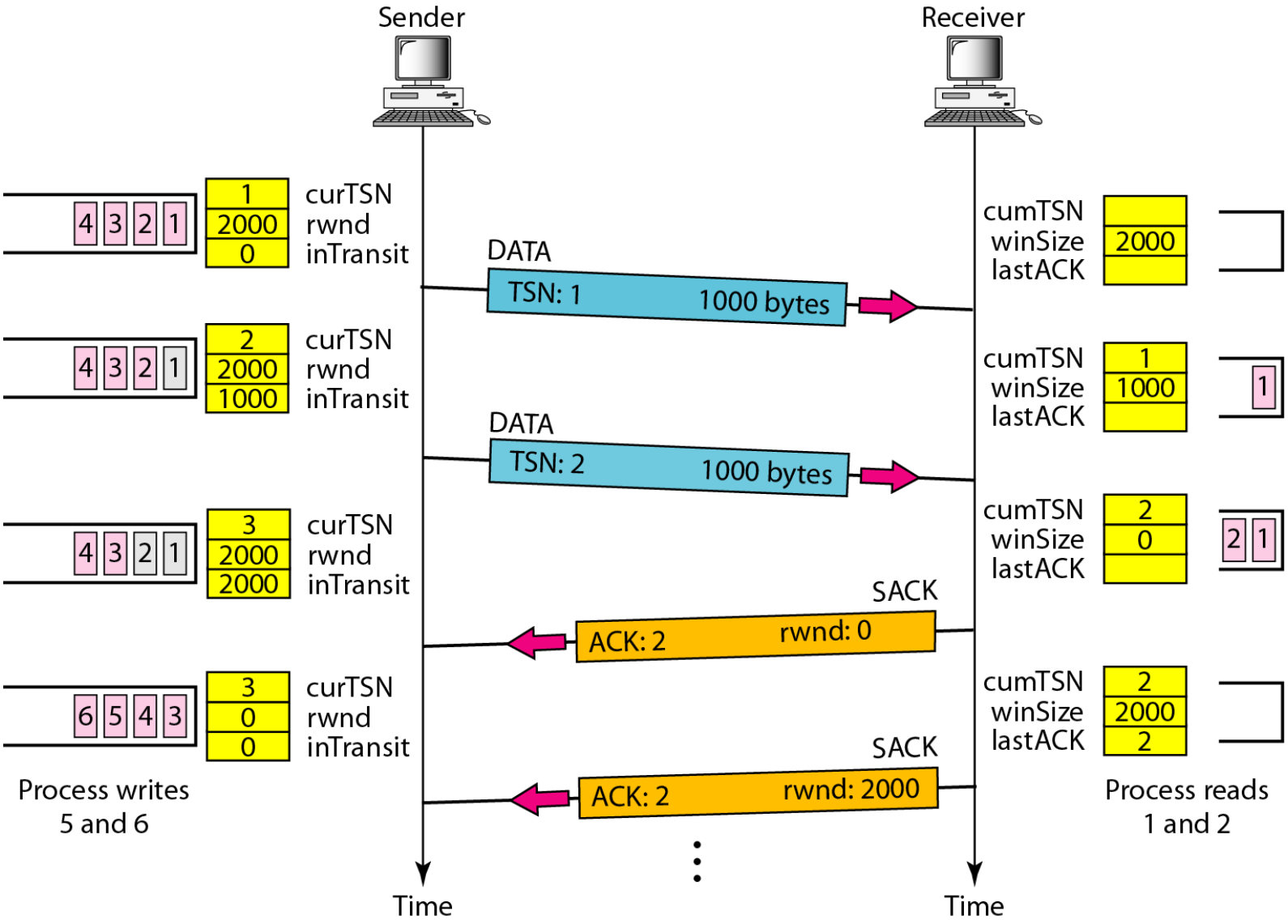


Figure 23.39 *Error control, receiver site*

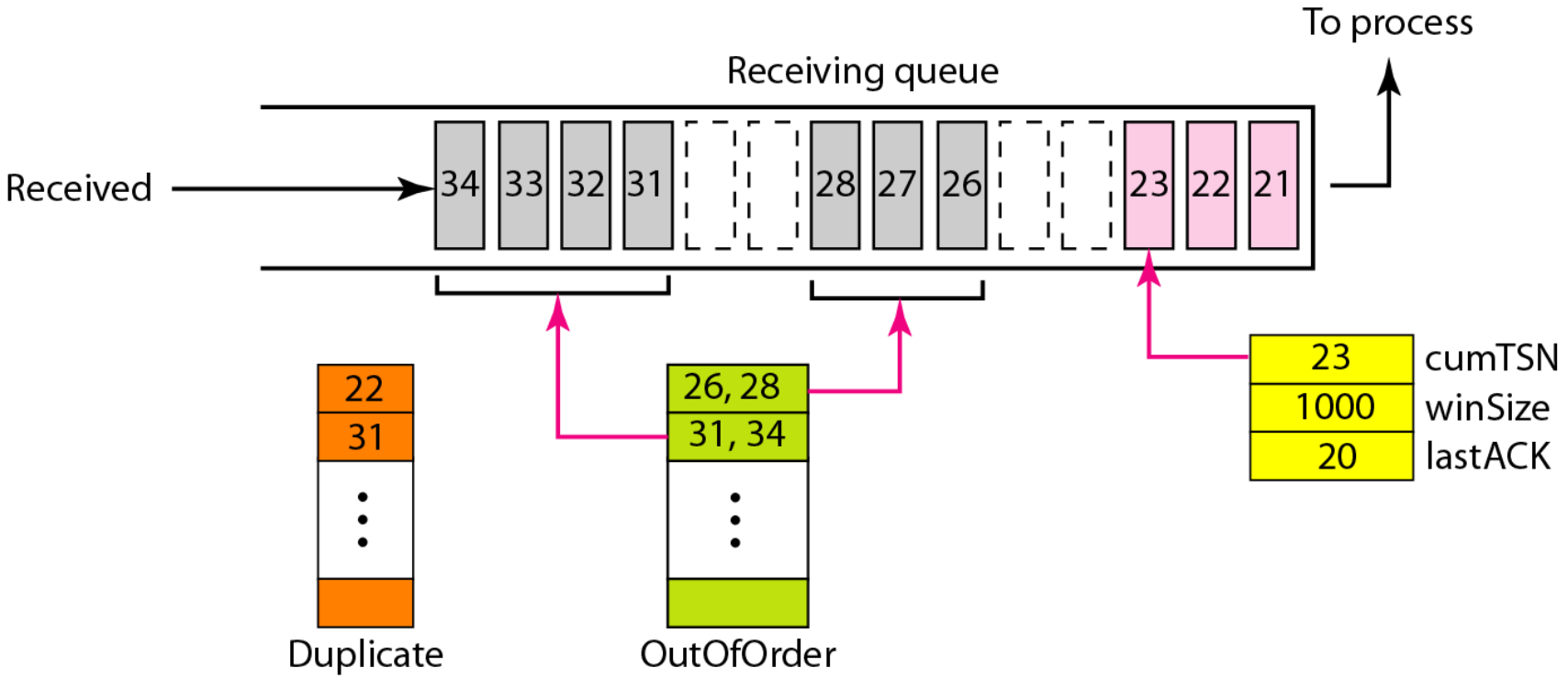


Figure 23.40 *Error control, sender site*

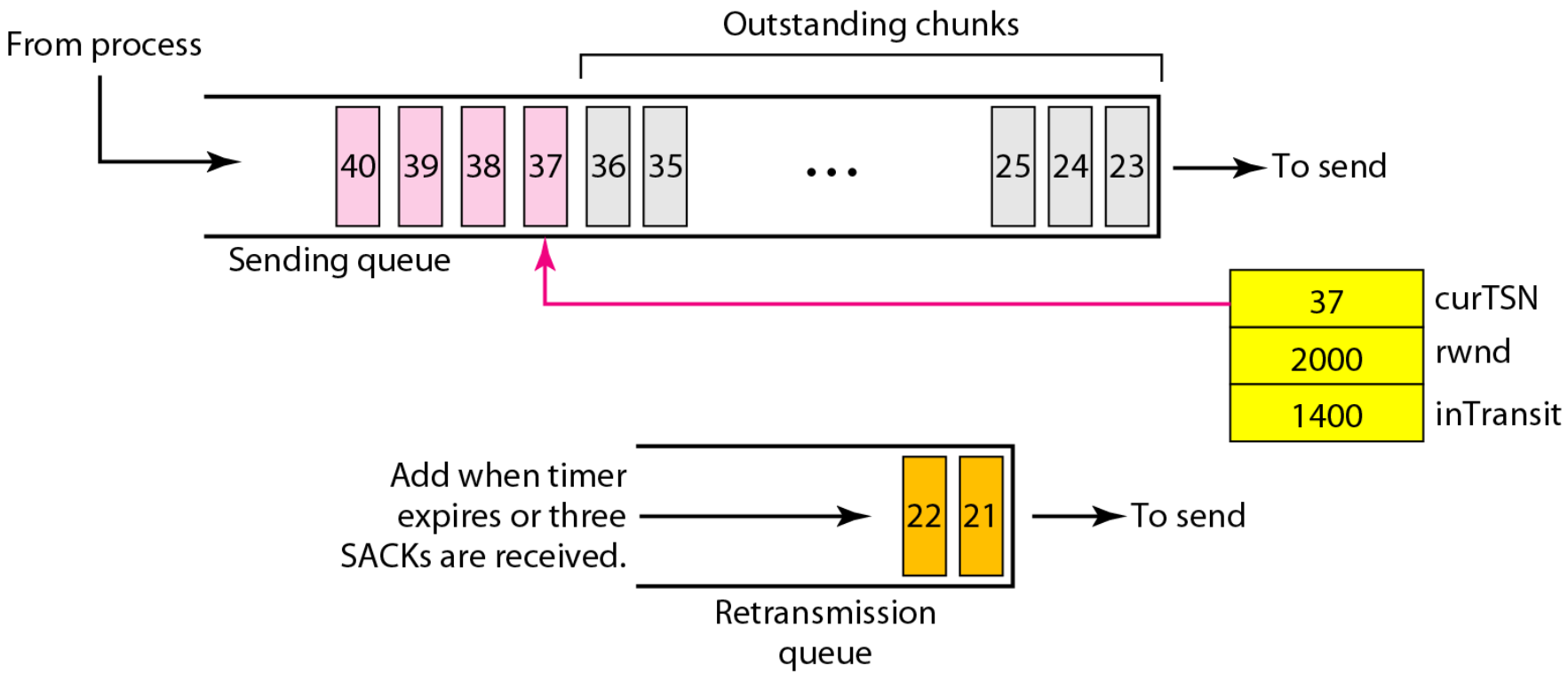


Figure 23.40 *Error control, sender site*