Unit IV: Transport Layer

Process to Process Delivery, Services, Socket Programming.

Elements of Transport Layer Protocols: Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, Congestion Control.

Transport Layer Protocols: TCP and UDP, SCTP, RTP, Congestion control and Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless networks.

Why the transport layer ?

The network layer exists on end hosts and routers in the network.

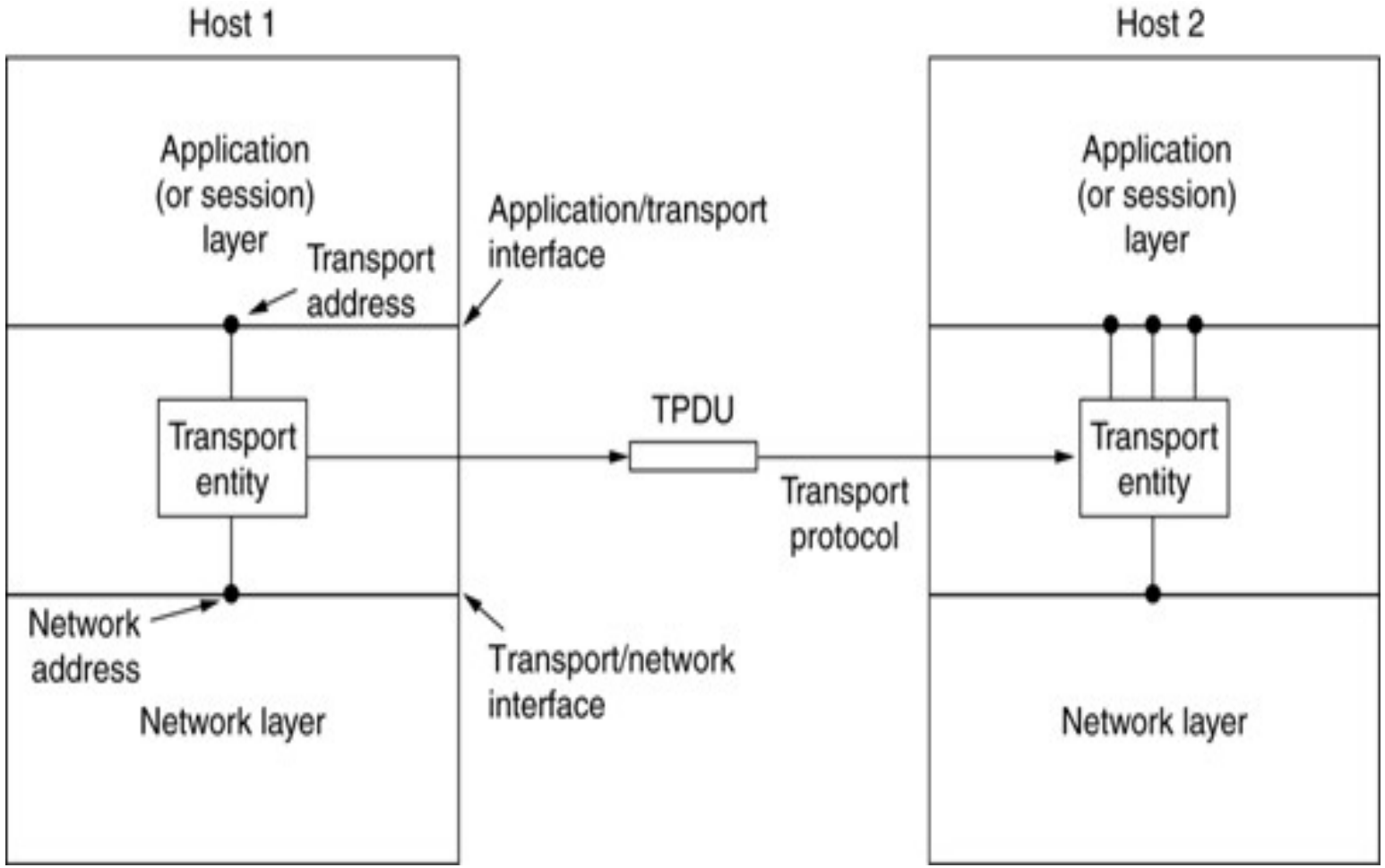
The end-user cannot control what is in the network.

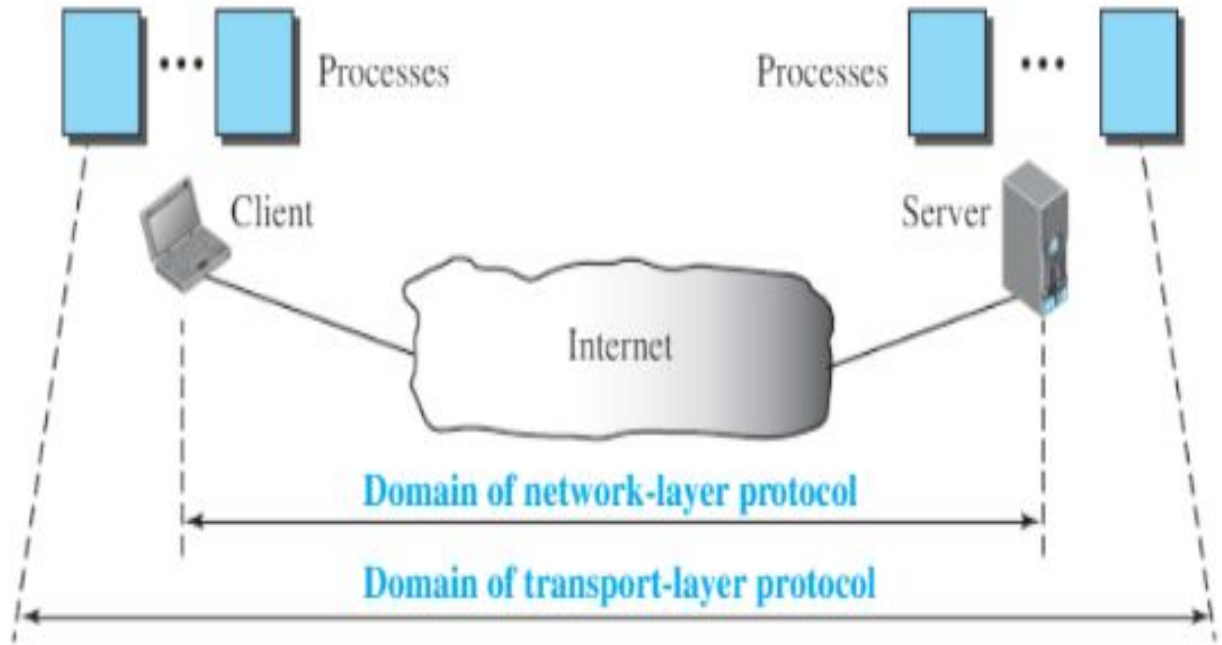
So the end-user establishes another layer, only at end hosts, to provide a transport service

While the network layer deals with only a few transport entities, the transport layer allows several concurrent applications to use the transport service.

It provides a common interface to application writers.

In essence, an application writer can write code once using the transport layer primitive and use it on different networks (but with the same transport layer).


Host 1
Host 2
Application
(or session)
layer
Transport
address
Application/transport
interface
Application
(or session)
layer
TPDU
Transport
entity
Transport
entity
Transport
protocol
Network
address
Transport/network
interface
Network layer
Network layer


Processes
Processes
Client
Server
Internet
Domain of network-layer protocol
Domain of transport-layer protocol

Transport Service Primitives

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

Elements of Transport Protocols

Addressing,

Connection Establishment,

Connection Release,

Flow Control and Buffering,

Multiplexing,

Crash Recovery

Addressing

To achieve process-to-process communication, the most common is through the client-server paradigm.

For communication, we must define the local host, local process, remote host, and remote process.

The local host and the remote host are defined using IP addresses.

To define the processes, we need second identifiers, called port numbers.

In the TCP/IP protocol suite, the port numbers are integers between 0 and

65,535 (16 bits)

ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic (or private)

Well-known ports- The ports ranging from 0 to 1023 are assigned and controlled

by ICANN. These are the well-known ports.

Registered ports-The ports ranging from 1024 to 49,151 are not assigned or con-

trolled by ICANN. They can only be registered with ICANN to prevent duplication.

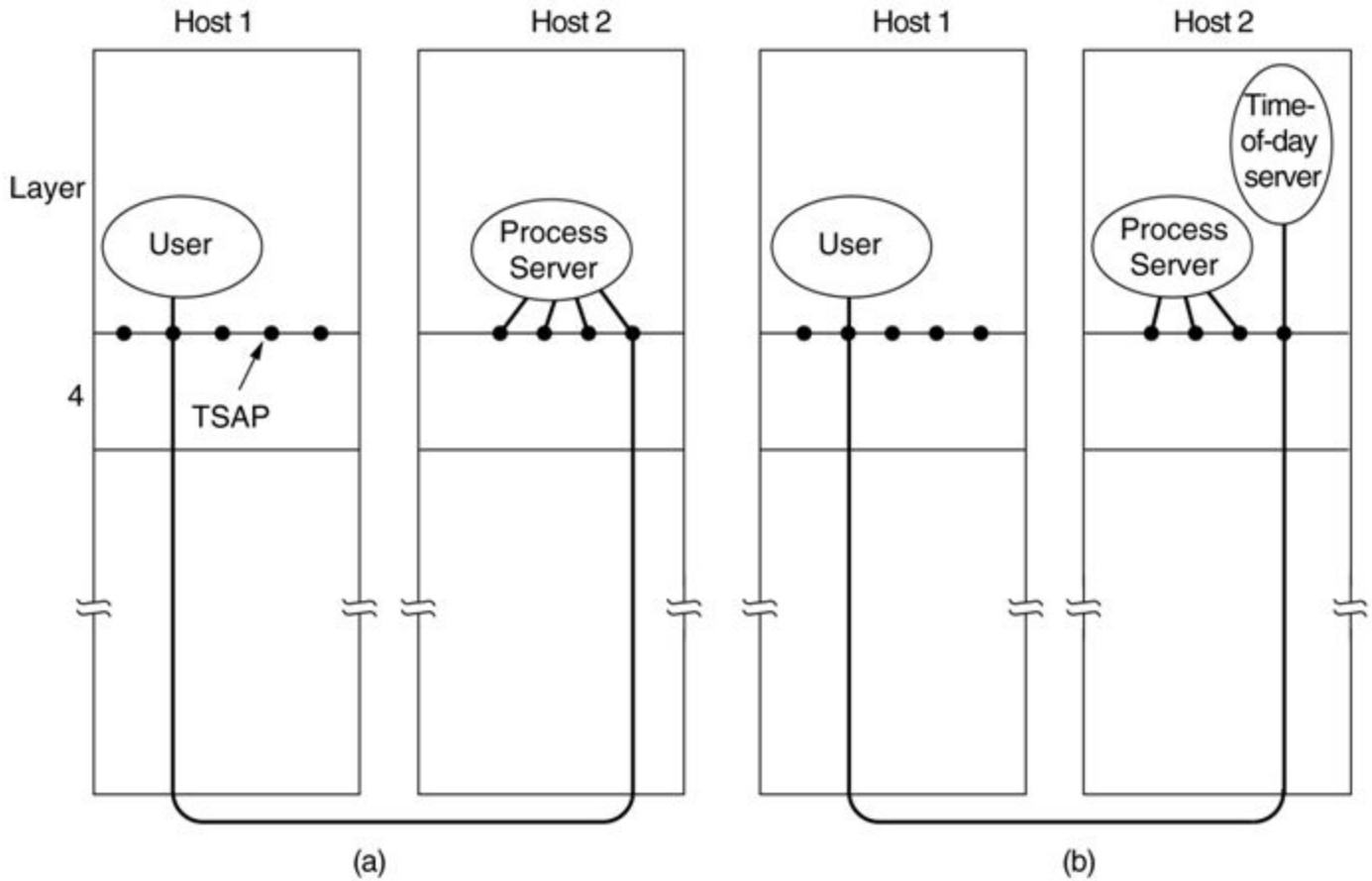
Dynamic ports-The ports ranging from 49,152 to 65,535 are neither controlled nor

registered. They can be used as temporary or private port numbers

Connection Establishment

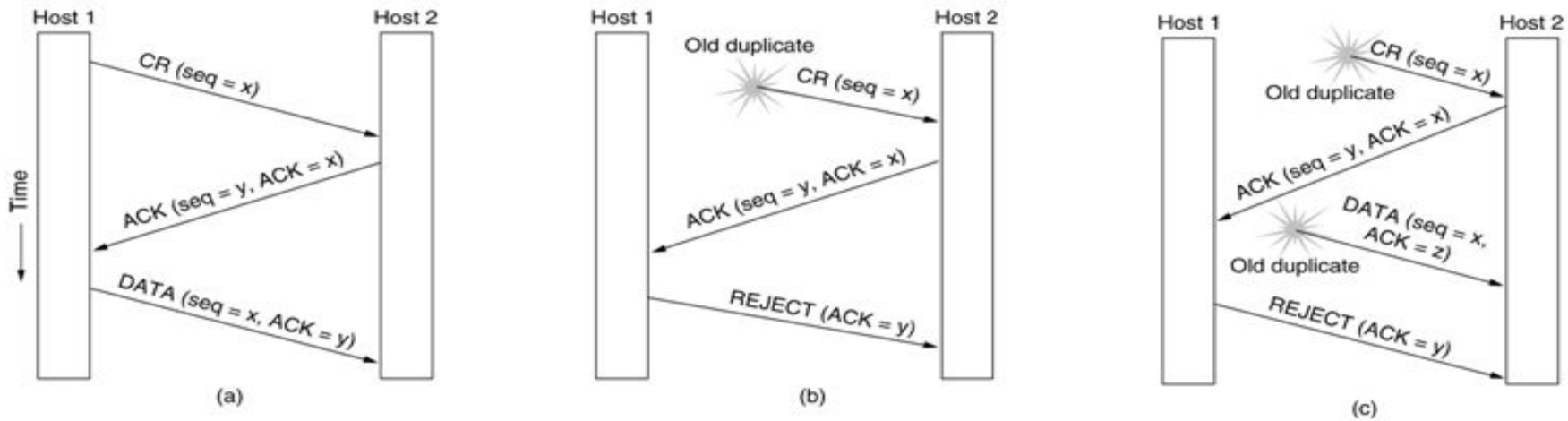
How a user process in host 1 establishes a connection with a time-of-day server in host 2.

Connection Establishment (1)



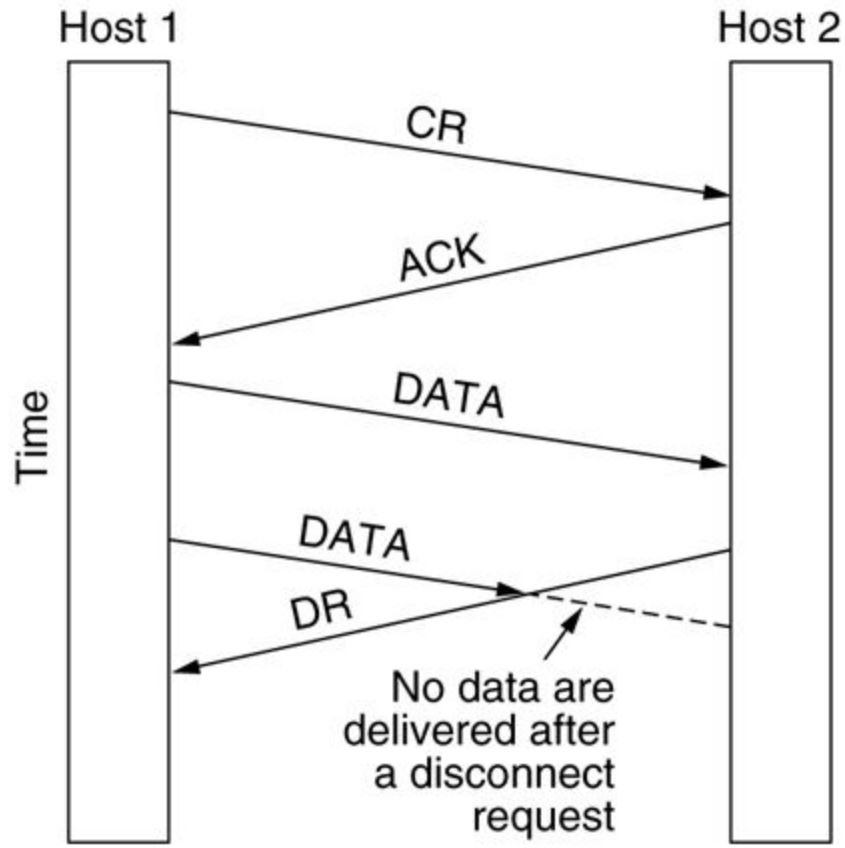
How a user process in host 1 establishes a connection with a time-of-day server in host 2.

Connection Establishment (2)



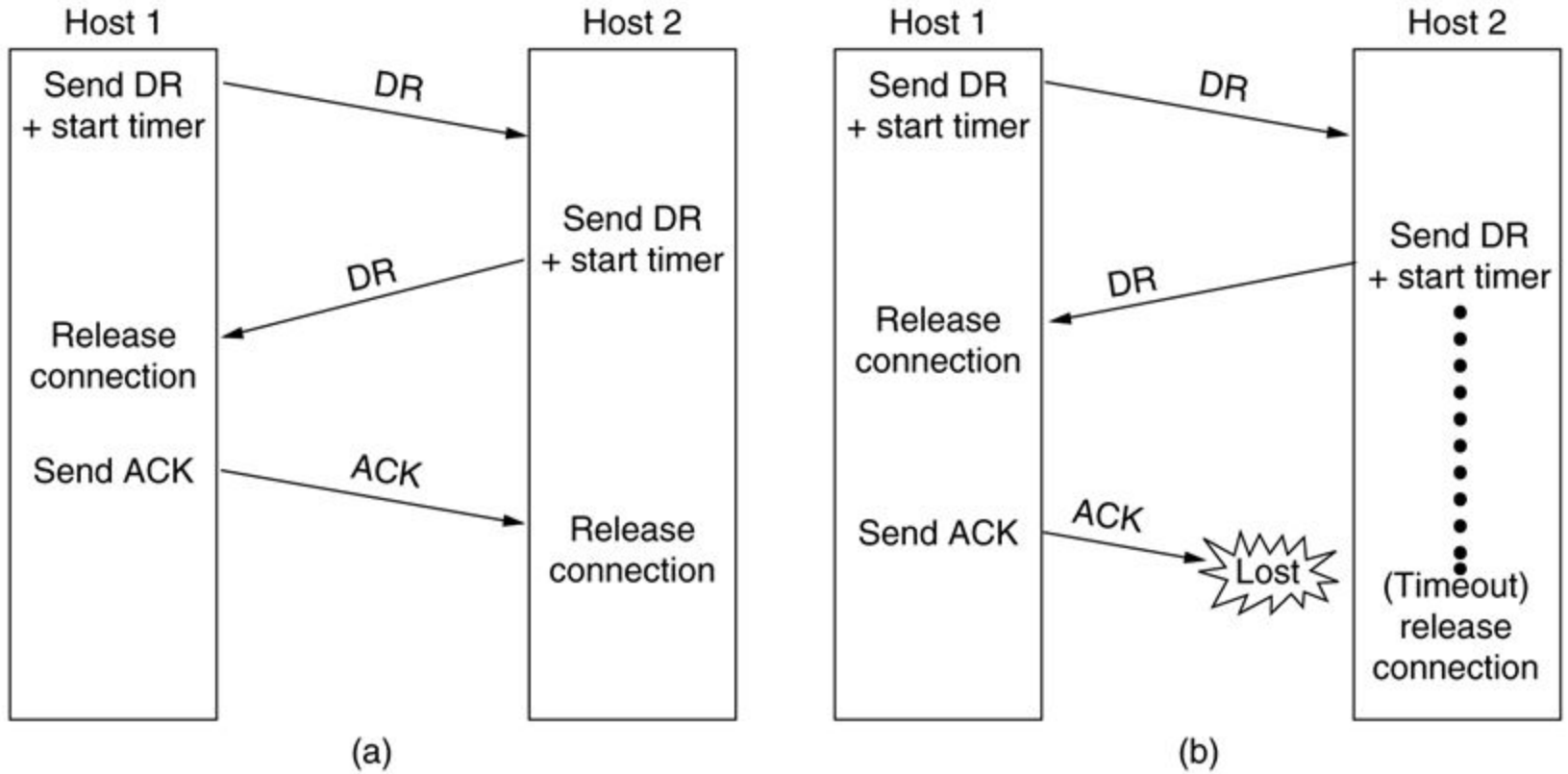
Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST.
(a) Normal operation,
(b) Old CONNECTION REQUEST appearing out of nowhere.
(c) Duplicate CONNECTION REQUEST and duplicate ACK.

Connection Release



Abrupt disconnection with loss of data.

Connection Release (3)



Four protocol scenarios for releasing a connection. (a) Normal case of a three-way handshake. (b) final ACK lost.

Encapsulation and Decapsulation

To send a message from one process to another, the transport-layer protocol encapsu-

lates and decapsulates messages.

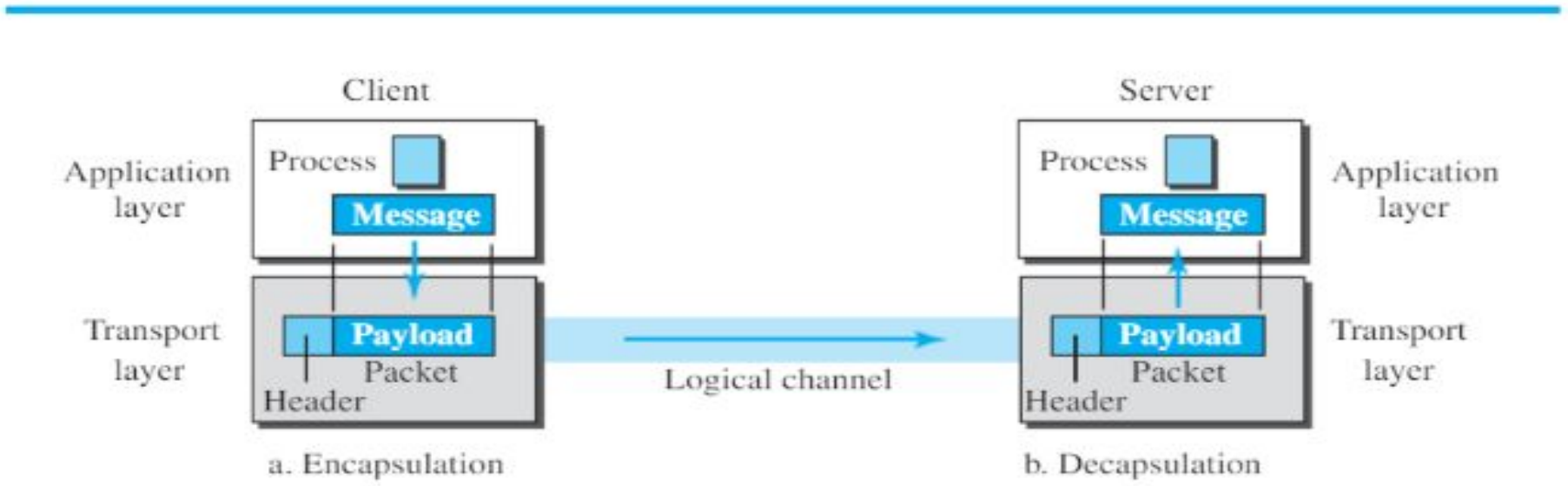
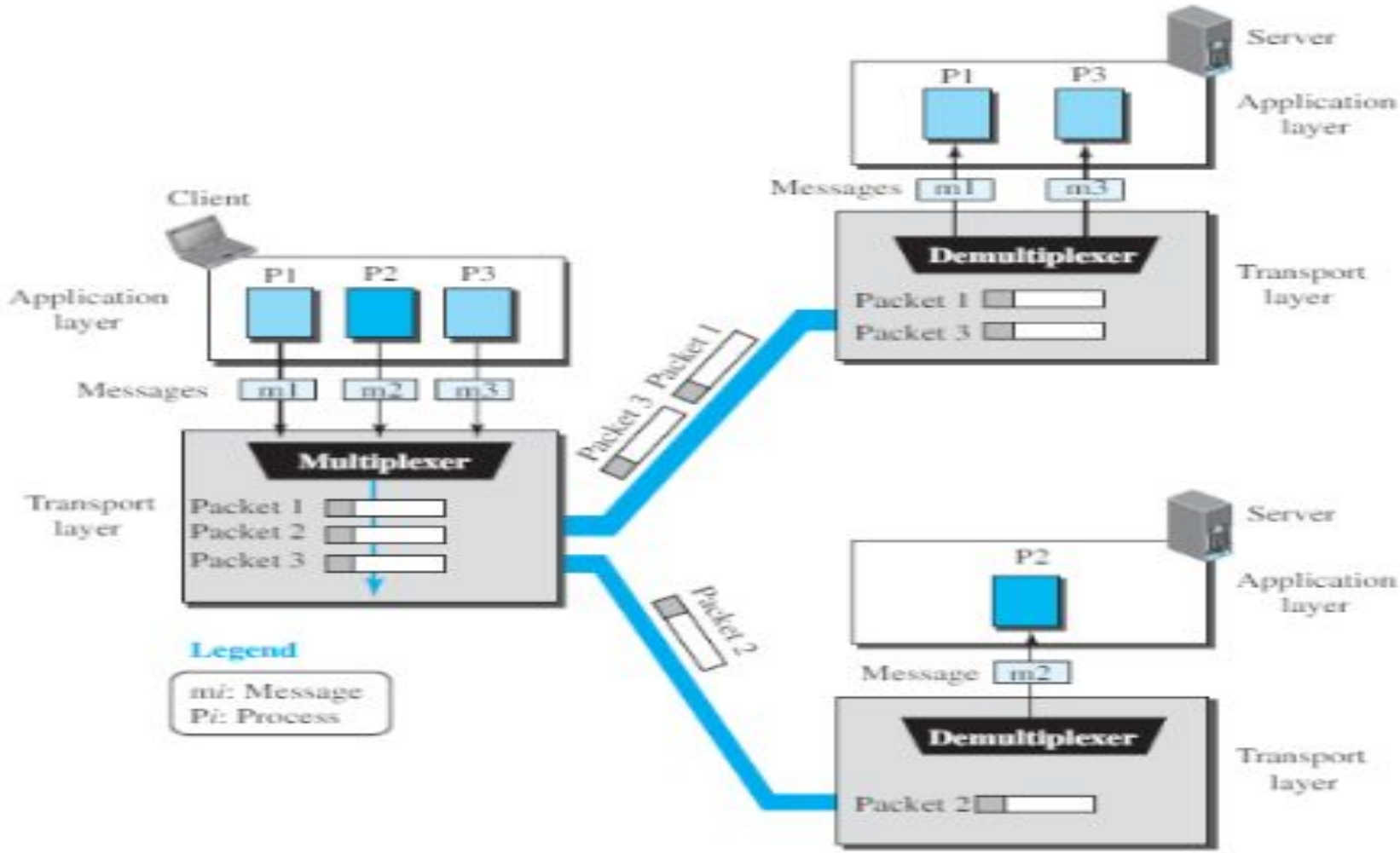


Figure 23.8 *Multiplexing and demultiplexing*



Flow Control

Figure 23.9 *Pushing or pulling*

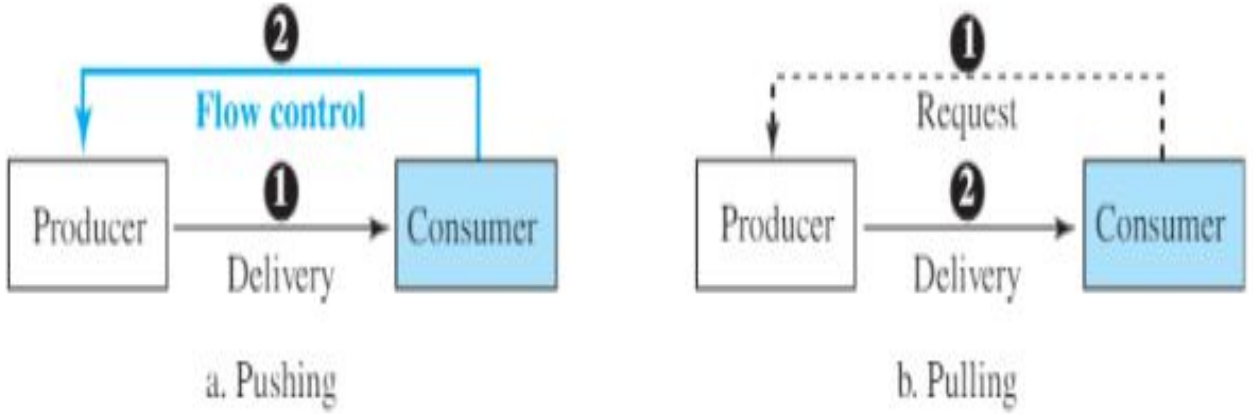
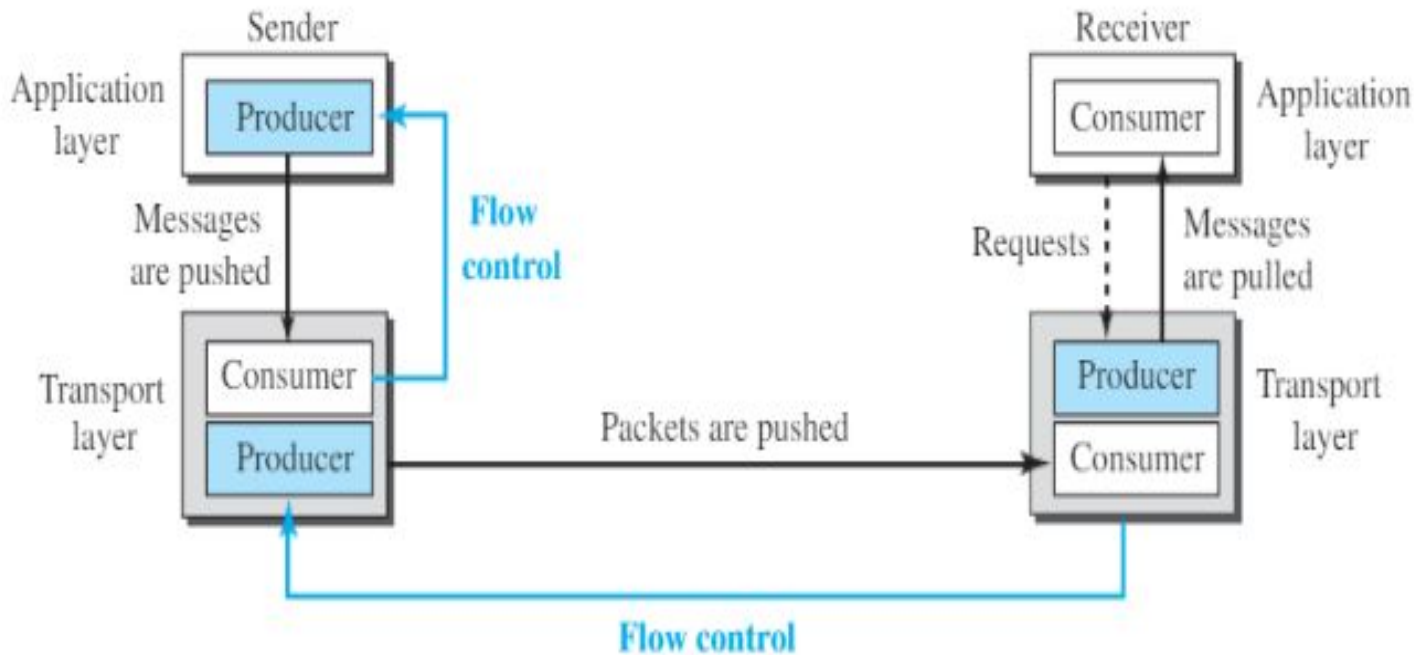


Figure 23.10 *Flow control at the transport layer*



Error control

Transport-Layer Services Error control In the Internet, since the underlying network layer (IP) is unreliable, we need to make the transport layer reliable if the application requires reliability.

Reliability can be achieved to add error control services to the transport layer.

Error control at the transport layer is responsible for

1. Detecting and discarding corrupted packets.
2. Keeping track of lost and discarded packets and resending them.
3. Recognizing duplicate packets and discarding them.
4. Buffering out-of-order packets until the missing packets arrive.

Transport-Layer Services Sequence numbers, Error control requires that the sending transport layer knows which packet is to be resent and the receiving transport layer knows which packet is a duplicate, or which packet has arrived out of order. This can be done if the packets are numbered.

We can add a field to the transport-layer packet to hold the sequence number of the packet

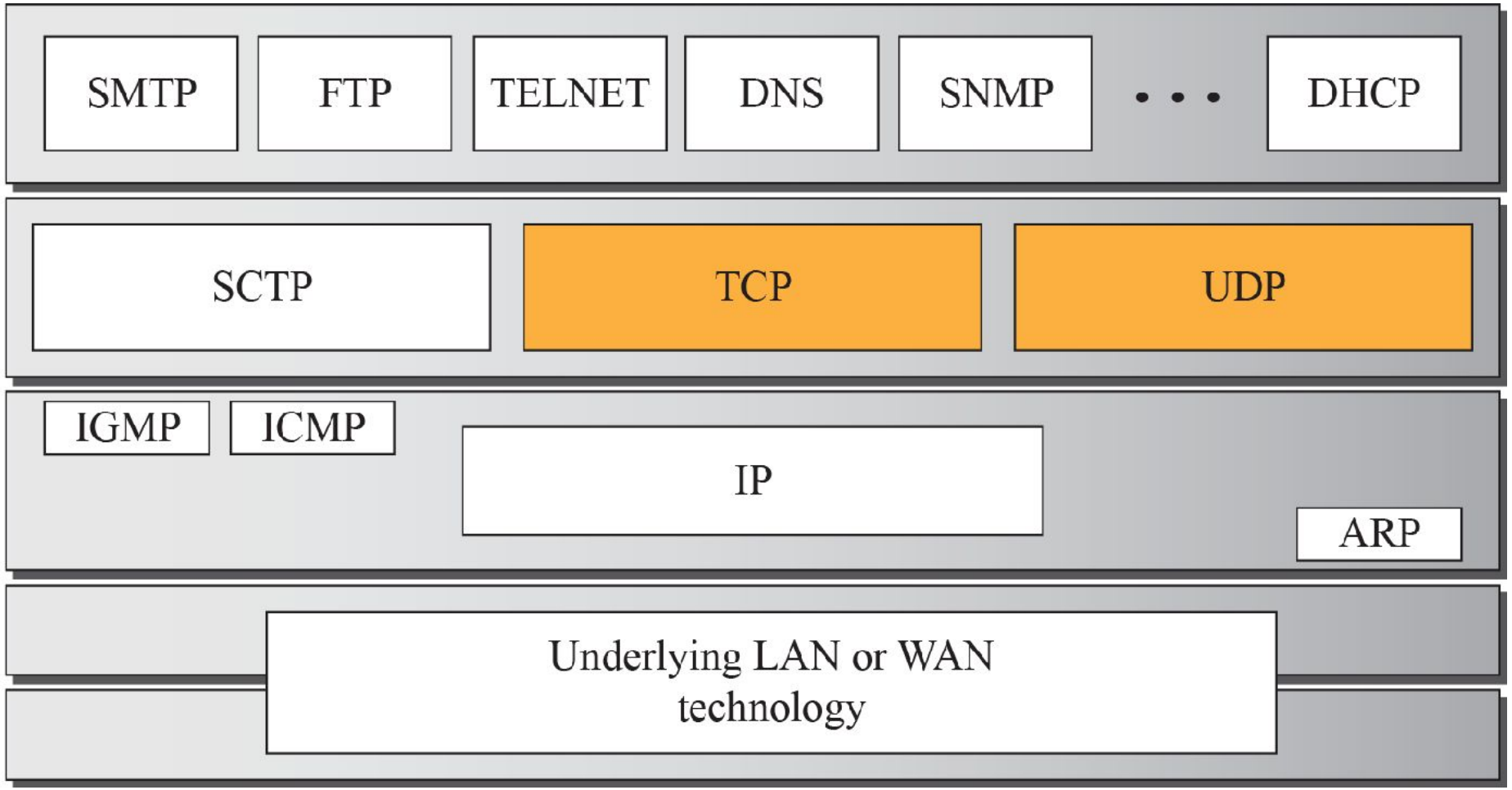
Congestion Control

Transport-Layer Services Congestion Control an important issue in a packet-switched network, such as the Internet, is congestion.

Congestion in a network may occur if the load on the network —the number of packets sent to the network—is greater than the capacity of the network—the number of packets a network can handle.

Congestion control refers to the mechanisms and techniques that control the congestion and keep the load below the capacity.

Position of transport-layer protocols in the TCP/IP protocol suite



TRANSPORT PROTOCOLS

The Internet has two main protocols in the transport layer, a connectionless protocol and a connection oriented protocol

The connectionless protocol is UDP

The connection-oriented protocol is TCP

Services

Each protocol provides a different type of service and should be used appropriately.

Port Numbers

Port numbers provide end-to-end addresses at the transport layer

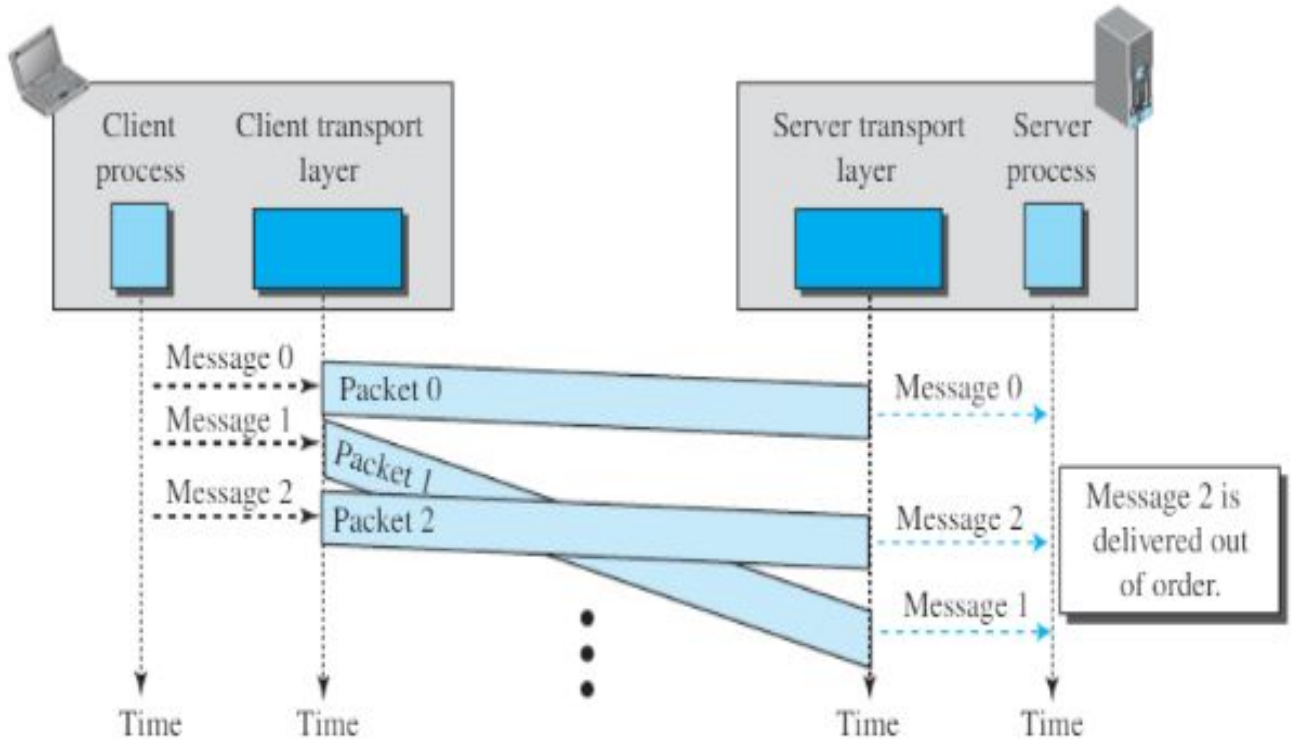
-allow multiplexing and demultiplexing,just as Ip addresses do at the network layer

Connectionless

In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one.

The transport layer treats each chunk as a single unit without any relation between the chunks.

When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it.

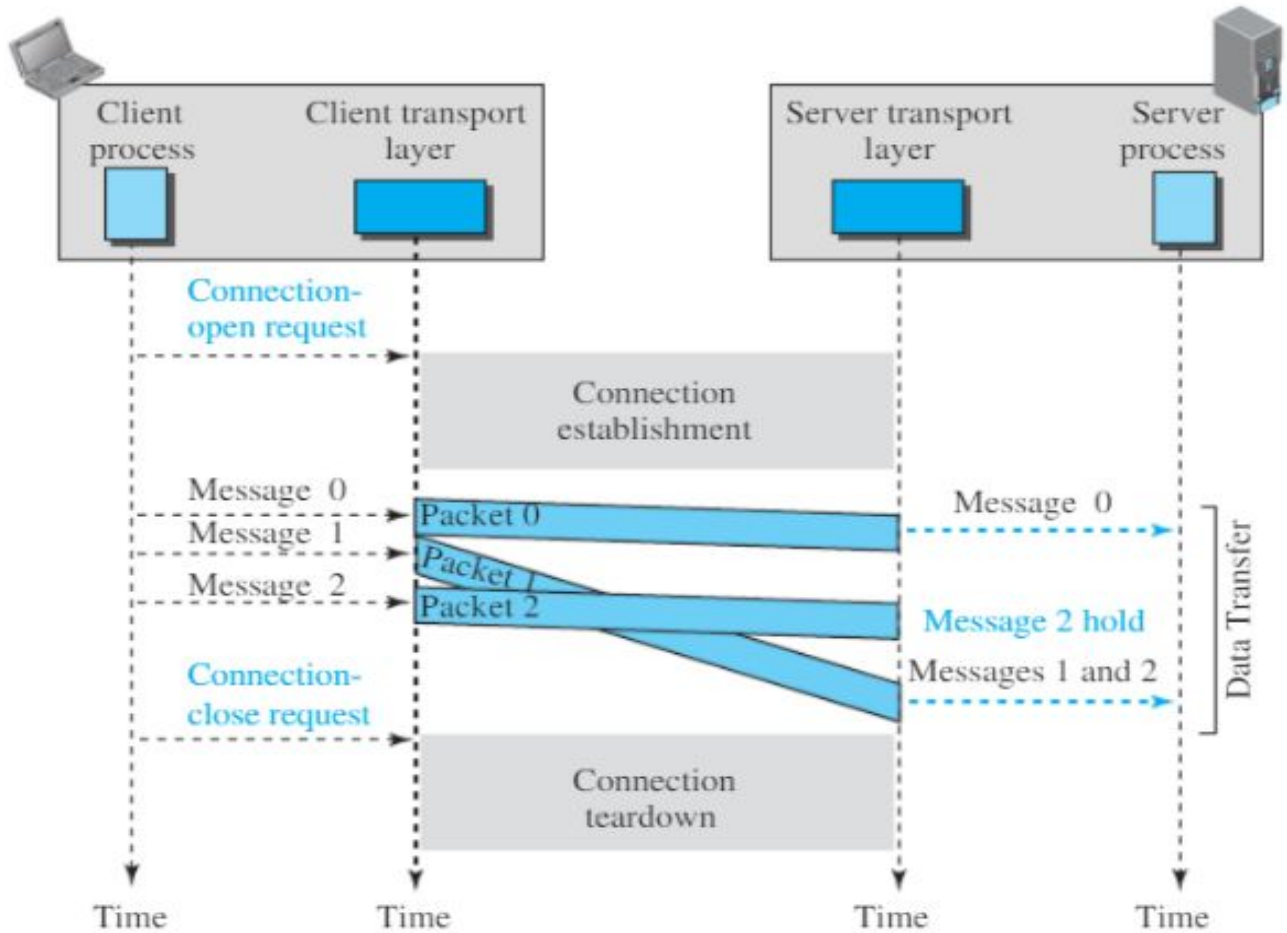

Client process
Client transport layer
Server transport layer
Server process
Message 0
Message 1
Message 2
Packet 0
Packet 1
Packet 2
Message 0
Message 2
Message 1
Message 2 is delivered out of order.
Time
Time
Time
Time

connection-oriented

In a connection-oriented service, the client and the server first need to establish a logical connection between themselves.

The data exchange can only happen after the connection establishment.

After data exchange, the connection needs to be torn down.


Client process
Client transport layer
Server transport layer
Server process
Connection-open request
Connection establishment
Message 0
Message 1
Message 2
Packet 0
Packet 1
Packet 2
Message 0
Message 2 hold
Messages 1 and 2
Connection-close request
Connection teardown
Data Transfer
Time
Time
Time
Time

UDP

The User Datagram Protocol(UDP)is

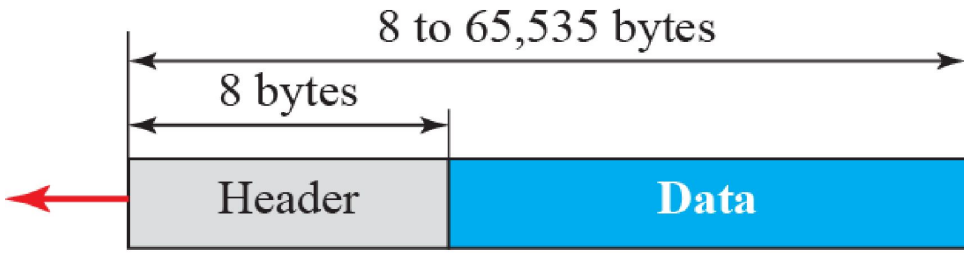
A connectionless, unreliable transport protocol

UDP is a very simple protocol using a minimum of overhead.

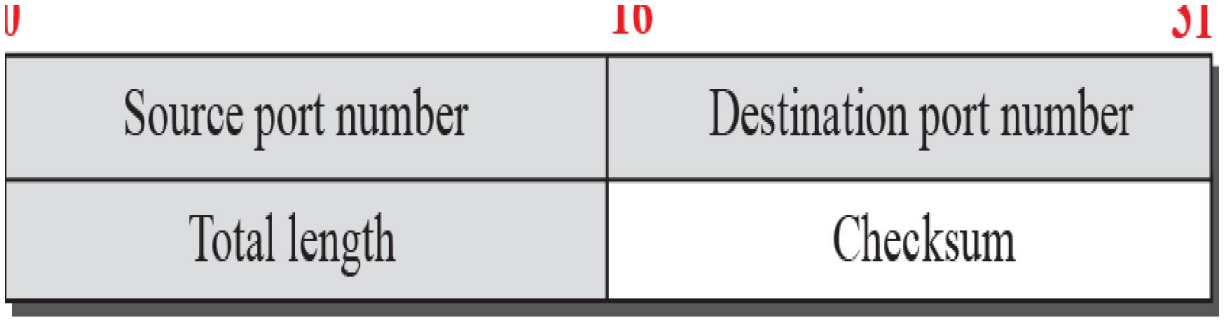
User Datagram

UDP packets, called user datagrams.

A fixed-size header of 8 bytes made off our fields, each of 2bytes(16bits)



a. UDP user datagram



b. Header format

UDP Services

1. Process to process communication
2. Connectionless services
3. Flow Control
4. Error Control
5. Encapsulation and Decapsulation
6. Multiplexing and Demultiplexing

Port	*Protocol*	*UDP*	*TCP*	*Description*
7	Echo	√		Echoes back a received datagram
9	Discard	√		Discards any datagram that is received
11	Users	√	√	Active users
13	Daytime	√	√	Returns the date and the time
17	Quote	√	√	Returns a quote of the day
19	Chargen	√	√	Returns a string of characters
20, 21	FTP		√	File Transfer Protocol
23	TELNET		√	Terminal Network
25	SMTP		√	Simple Mail Transfer Protocol
53	DNS	√	√	Domain Name Service
67	DHCP	√	√	Dynamic Host Configuration Protocol
69	TFTP	√		Trivial File Transfer Protocol
80	HTTP		√	Hypertext Transfer Protocol
111	RPC	√	√	Remote Procedure Call
123	NTP	√	√	Network Time Protocol
161, 162	SNMP		√	Simple Network Management Protocol

Figure 14.5 *Encapsulation and decapsulation*

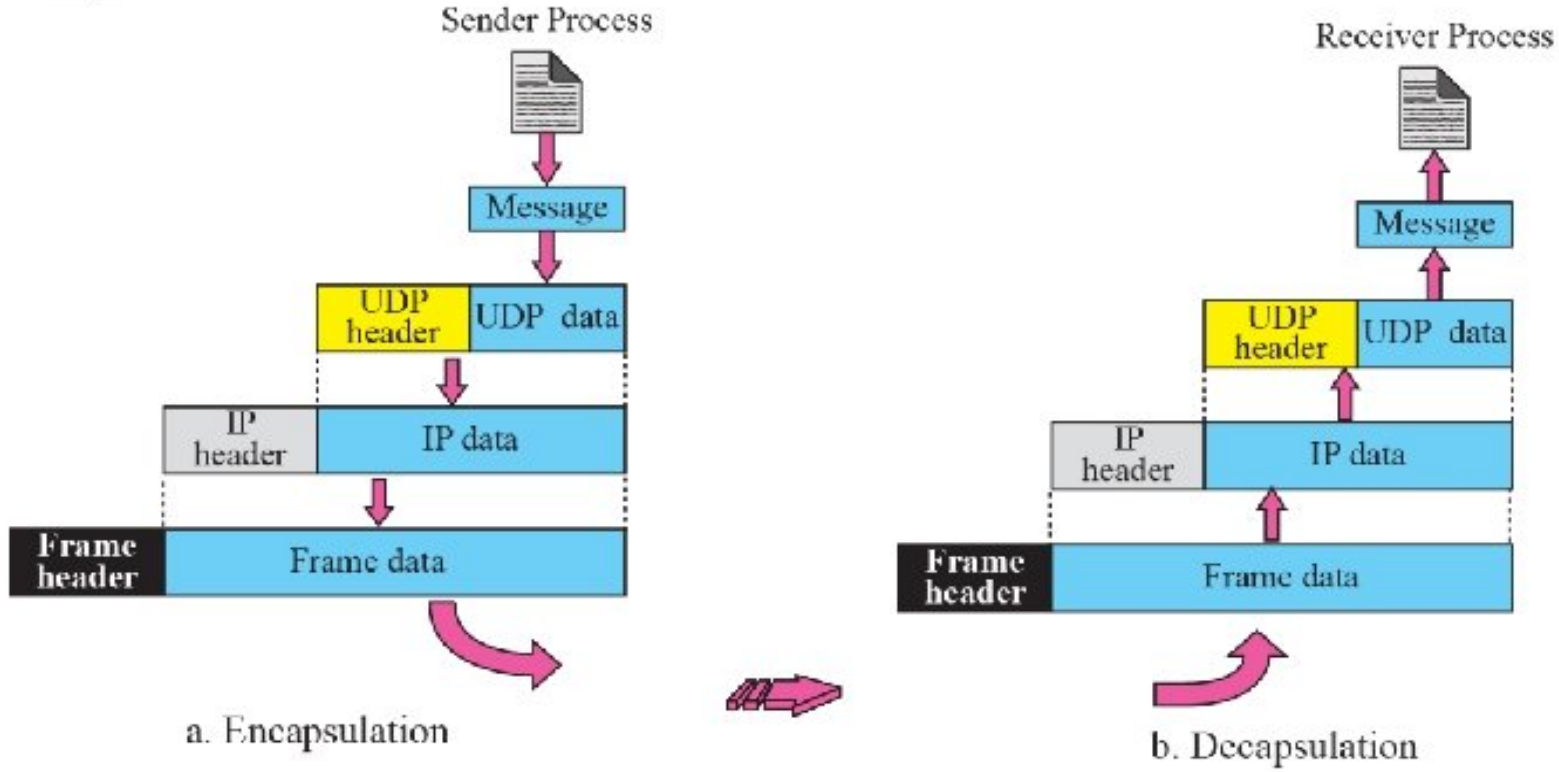
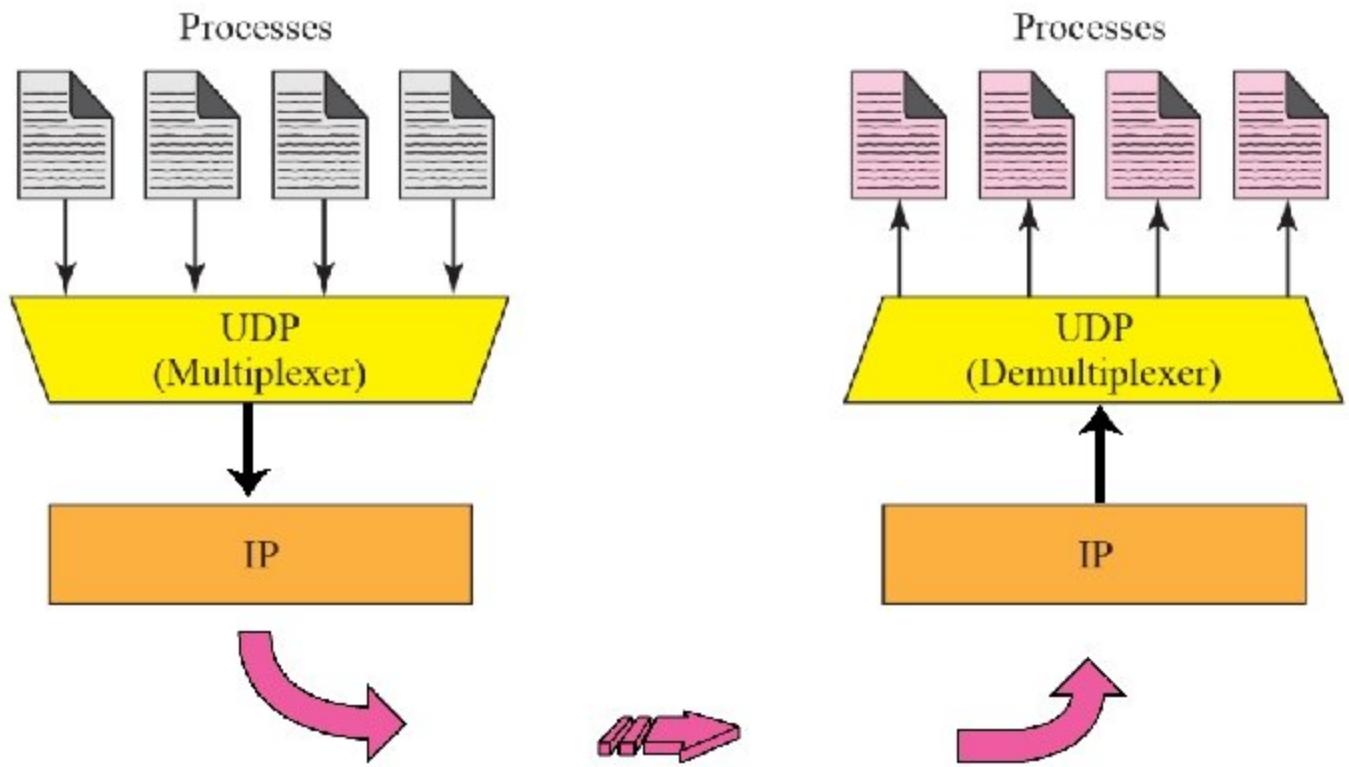


Figure 14.7 *Multiplexing and demultiplexing*



UDP Application

client-server application such as DNS uses the services of UDP because a client needs to send a short request to a server and to receive a quick response from it. The request and response can each fit in one user datagram. Since only one message is exchanged in each direction, the connectionless feature is not an issue; the client or server does not worry that messages are delivered out of order.

A client-server application such as SMTP , which is used in electronic mail, cannot use the services of UDP because a user can send a long e-mail message, which may include multimedia (images, audio, or video). If the application uses UDP and the message does not fit in one single user datagram, the message must be split by the application into different user datagrams. Here the connectionless service may create problems. The user datagrams may arrive and be delivered to the receiver application out of order. The receiver application may not be able to reorder the pieces. This means the connectionless service has a disadvantage for an application program that sends long messages.

TCP

is a connection-oriented

-reliable protocol

-explicitly defines connection establishment, data transfer, and connection teardown phases

-uses a combination of GBN and SR protocols to provide reliability

TCP Services

Process to process communication

Stream delivery

Sending and receiving buffers

TCP segments

Full Duplex

Multiplexing and demultiplexing

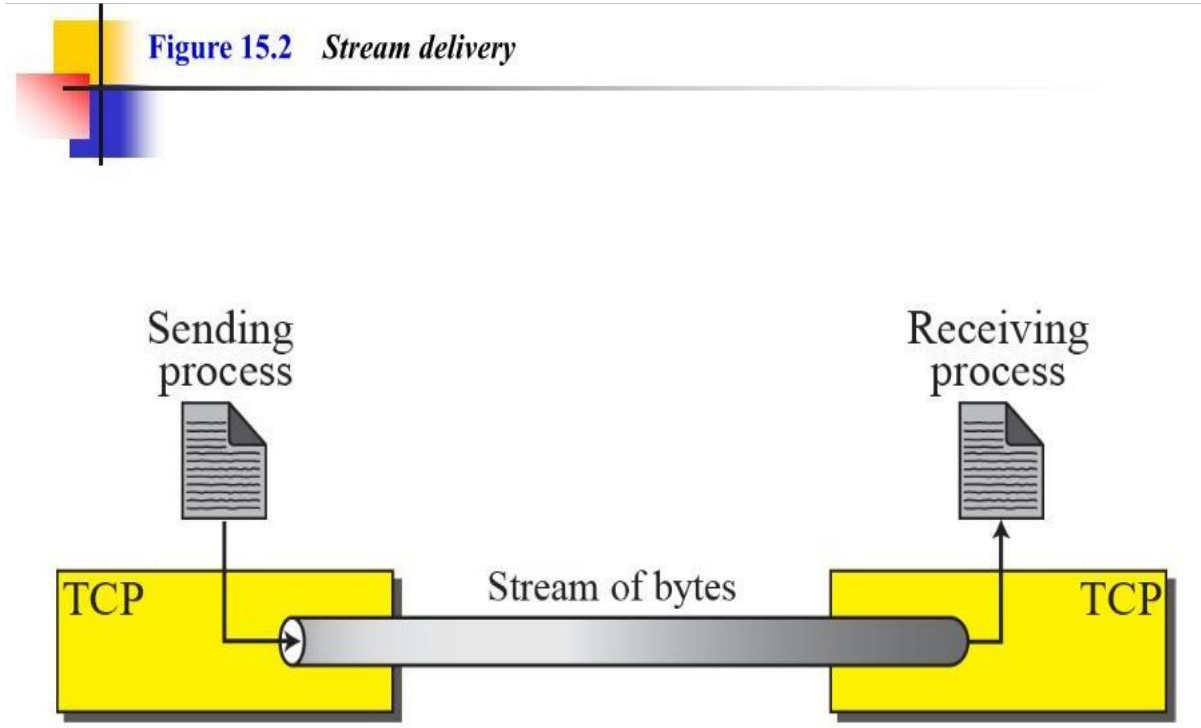
Connection oriented service

Reliable Service

Stream delivery

UDP treats each chunk independently

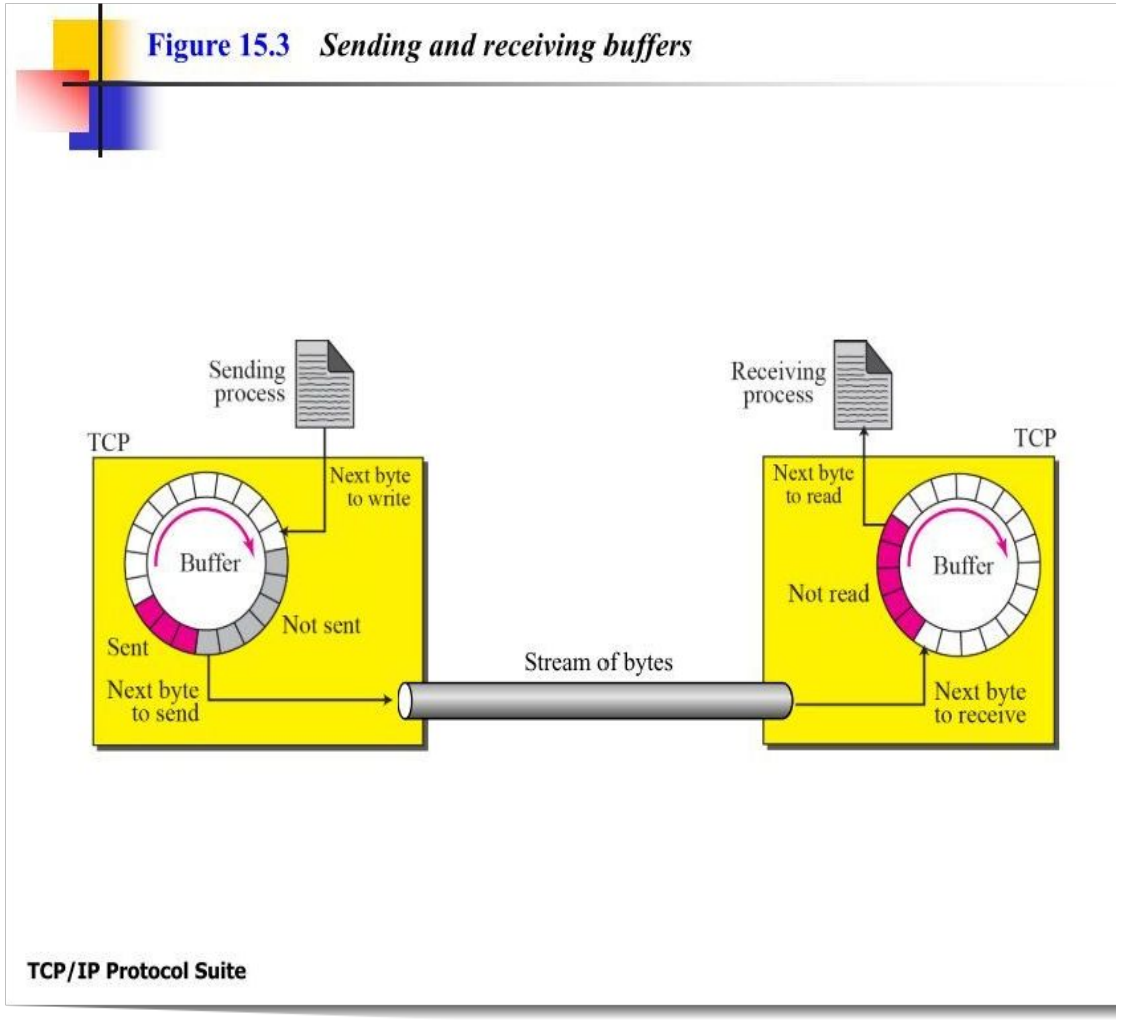
- No any connection between the chuncks
- In contrast, TCP allow the data be delivered/received as a stream of bytes



Sending and Receiving buffers

the sending and receiving speed may not be the same

- TCP needs buffers for storage
- Two buffers in TCP
- Sending buffer and receiving buffer, one for each connection
- Also used in flow- and error-control mechanisms



Segments

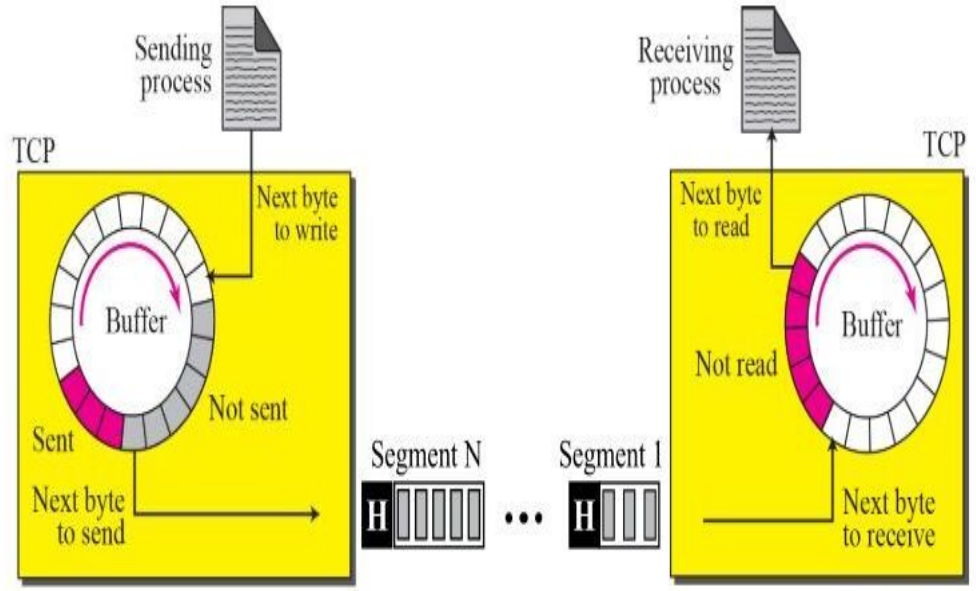
TCP groups a number of bytes together into a packet called a segment

A TCP packet is called a segment

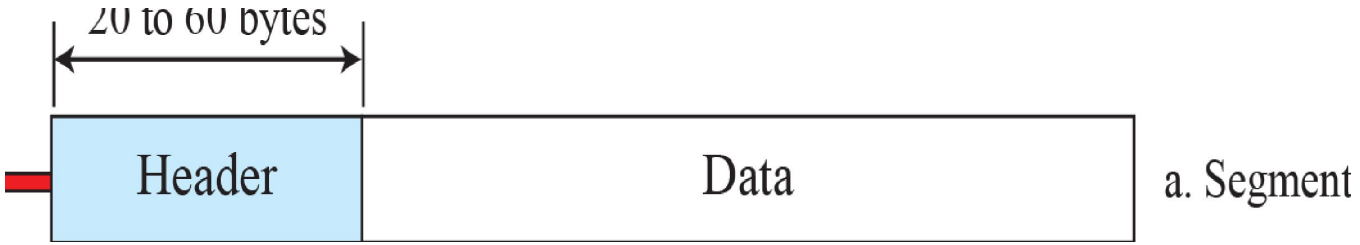
TCP adds a header to each segment

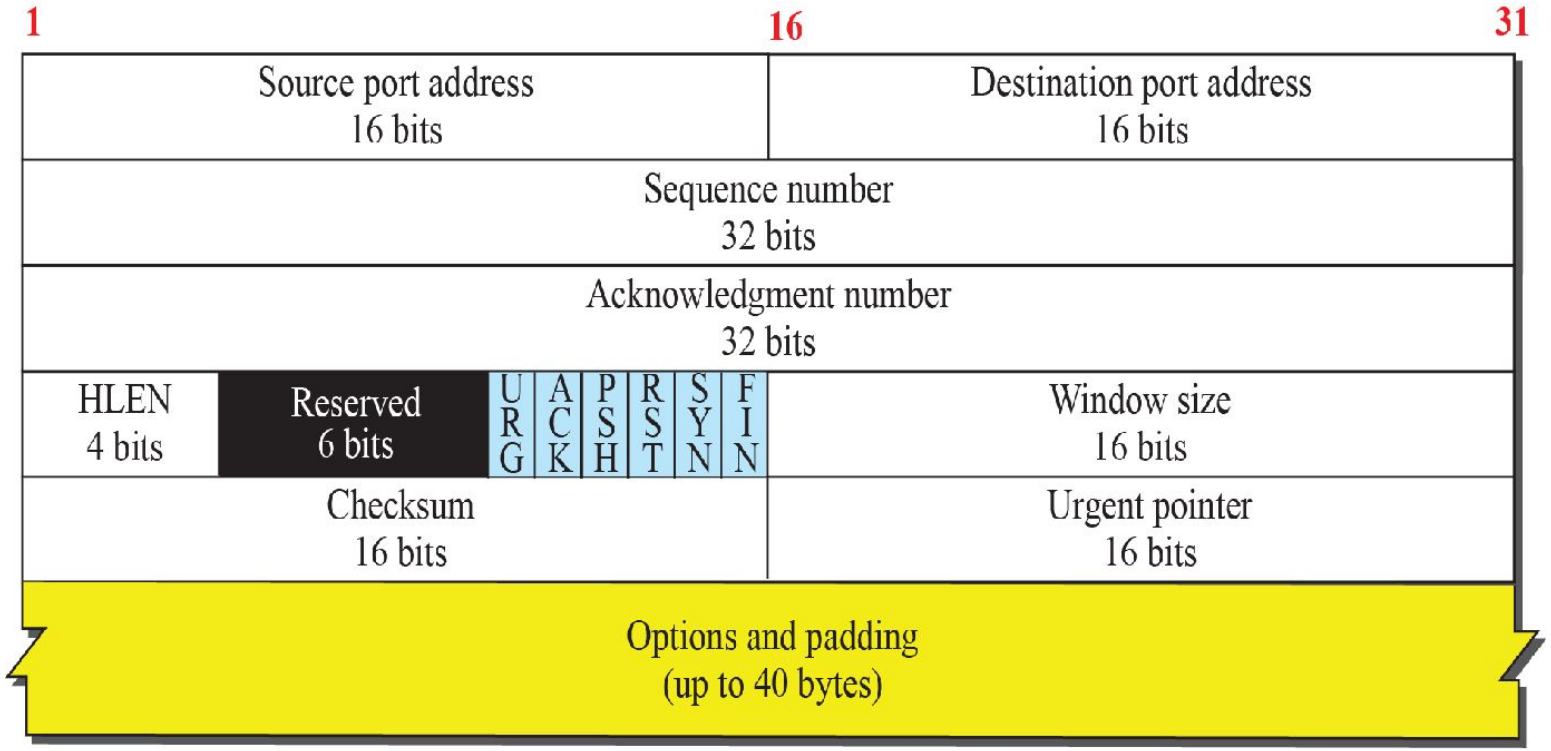
Then, the segments are encapsulated in an IP datagram

Figure 15.4 *TCP segments*



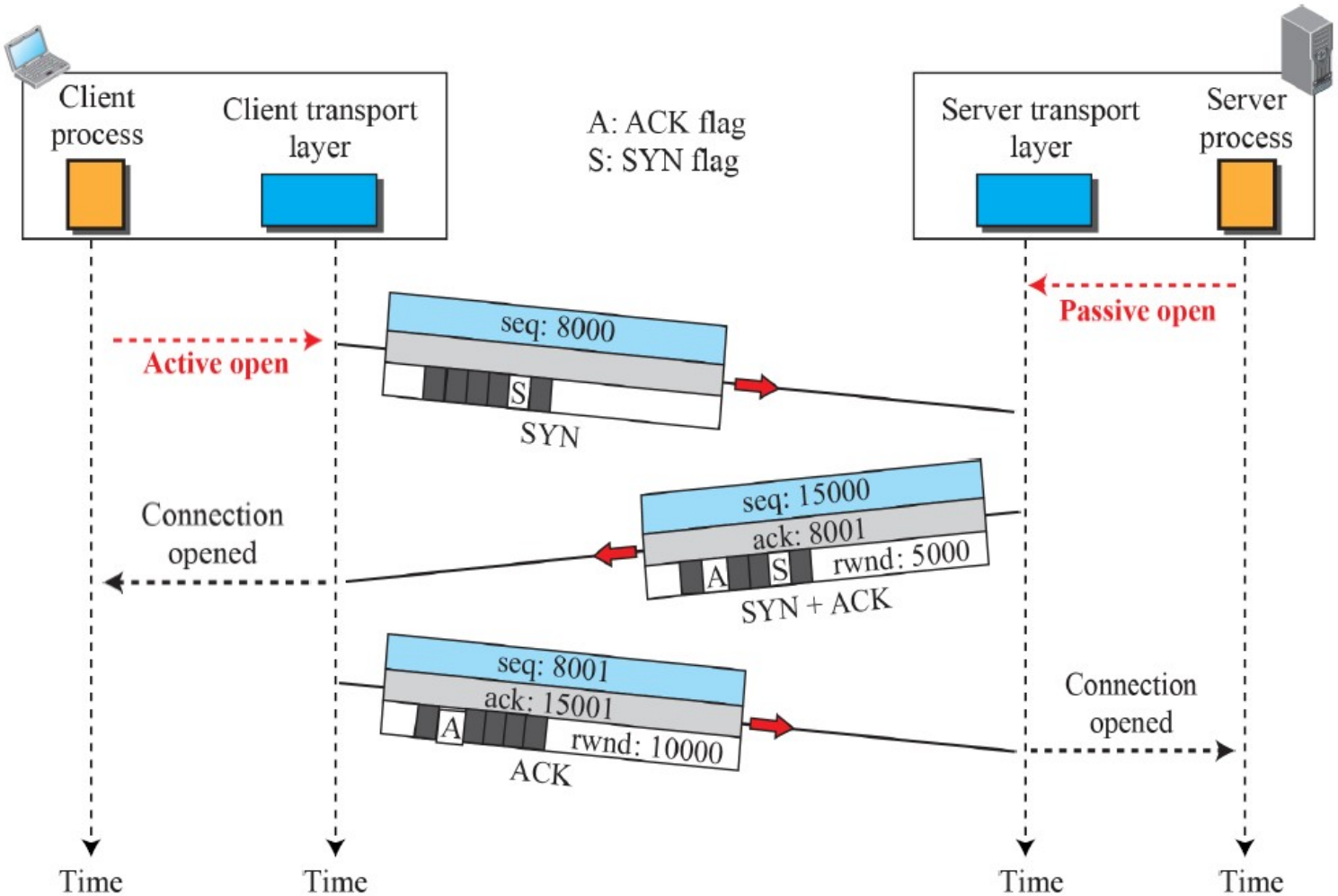
TCP segment format





b. Header

Connection establishment using three-way handshaking



TCP FEATURES

Numbering System

Flow Control

Error Control

Congestion Control

Numbering Bytes

TCP uses sequence number and acknowledgement number to keep track of the segment being transmitted or received

The numbering does not necessarily start from 0

- It starts randomly
- Between 0 and 2^32 – 1 for the number of the first byte
- Byte numbering is used for flow and error control

Sequence Number

TCP assigns a sequence number to each segment that is being sent

The sequence number for each segment is the number of the first byte carried in that segment

Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data is sent in five segments, each carrying 1000 bytes?

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)

Flow Control

The receiver controls how much data are to be sent by the sender

Prevent the receiver from being overwhelmed with data

Error Control

TCP implements an error control mechanism

To provide reliable service

Congestion Control

TCP takes into account congestion in the network

Thus, the amount of data sent by a sender is controlled both by

- The receiver (flow control) n
- The level of congestion in the network

SCTP

- Stream Control Transmission Protocol (SCTP) is a reliable, message-oriented transport layer protocol.
- SCTP has mixed features of TCP and UDP.
- SCTP maintains the message boundaries and detects the lost data, duplicate data as well as out-of-order data.
- SCTP provides the Congestion control as well as Flow control.
- SCTP is especially designed for internet applications.

SCTP Services

1. **Process-to- Process communication**
2. **Multi- Stream Facility**
3. **Multihoming**
4. **Full- Duplex Communication**

5. Connection- Oriented Service : The **SCTP** is a connection oriented protocol, just like TCP with the only difference that, it is called **association in SCTP**.

If User1 wants to send and receive message from user2, the steps are :

Step1: The two **SCTPs** establish the connection with each other.
Step2: Once the connection is established, the data gets exchanged in both the directions.
Step3: Finally, the association is terminated.

5. Reliability
SCTP uses an acknowledgement mechanism to check the arrival of data.

Process-to- Process communication : SCTP uses all important ports of TCP.

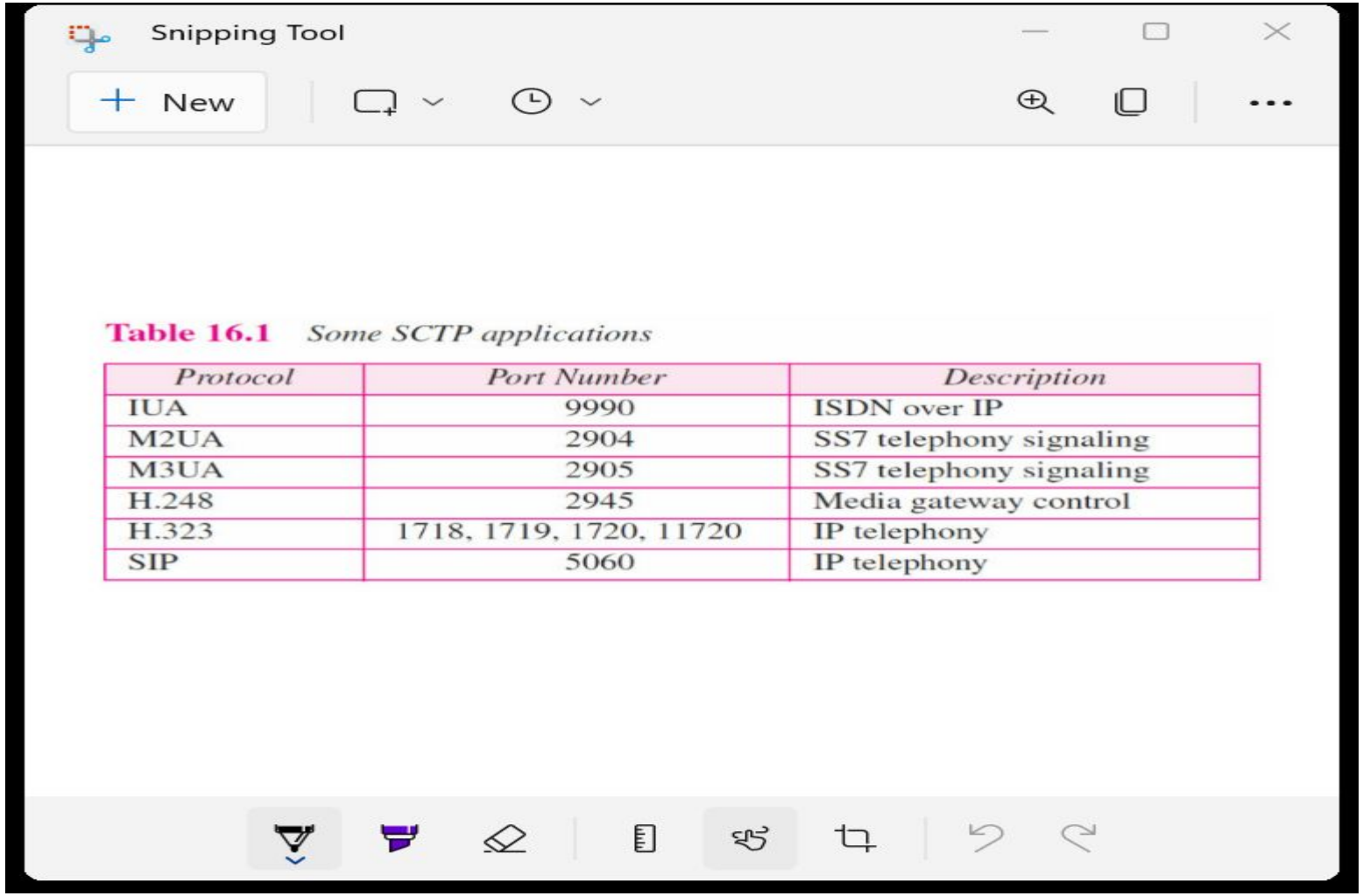
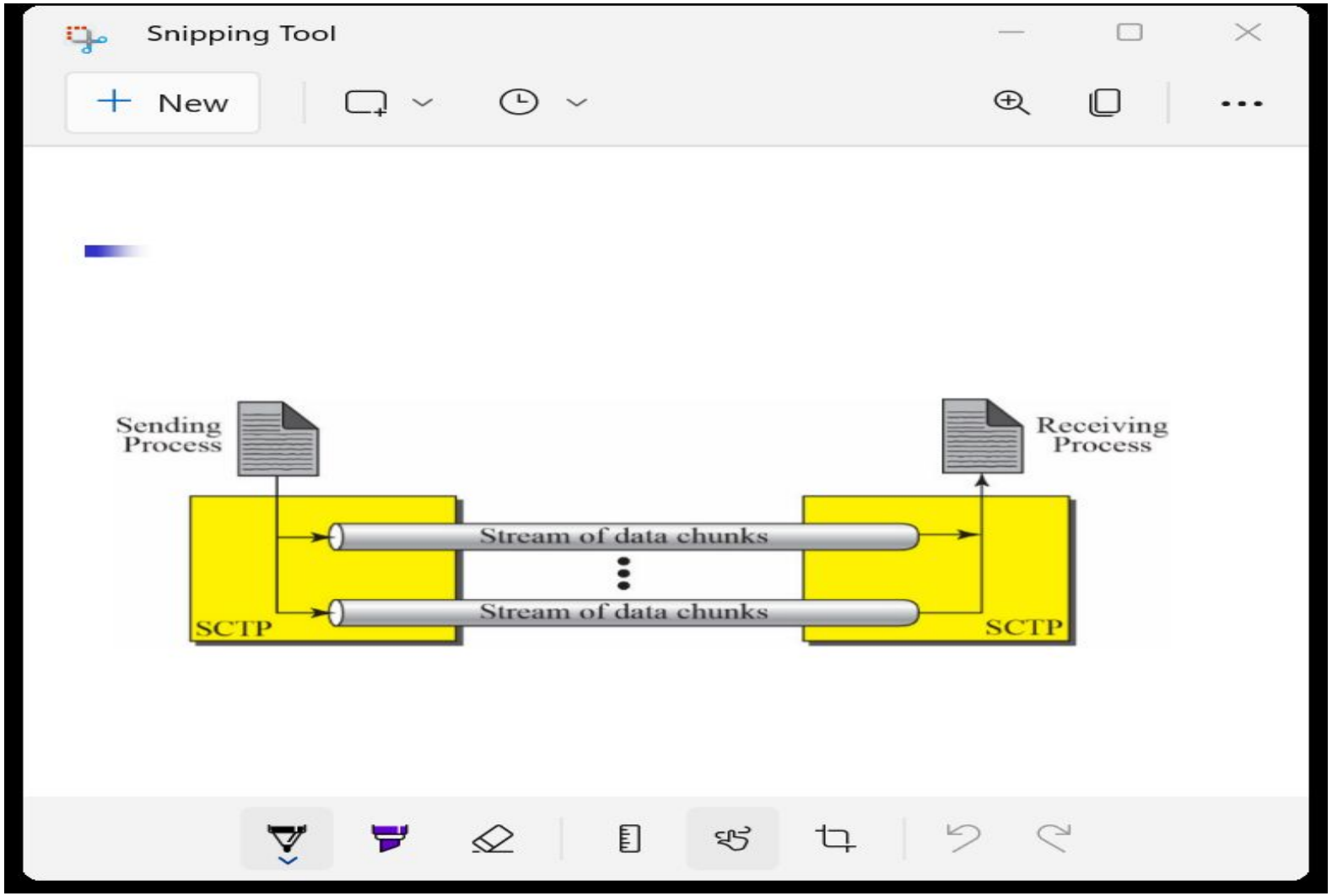


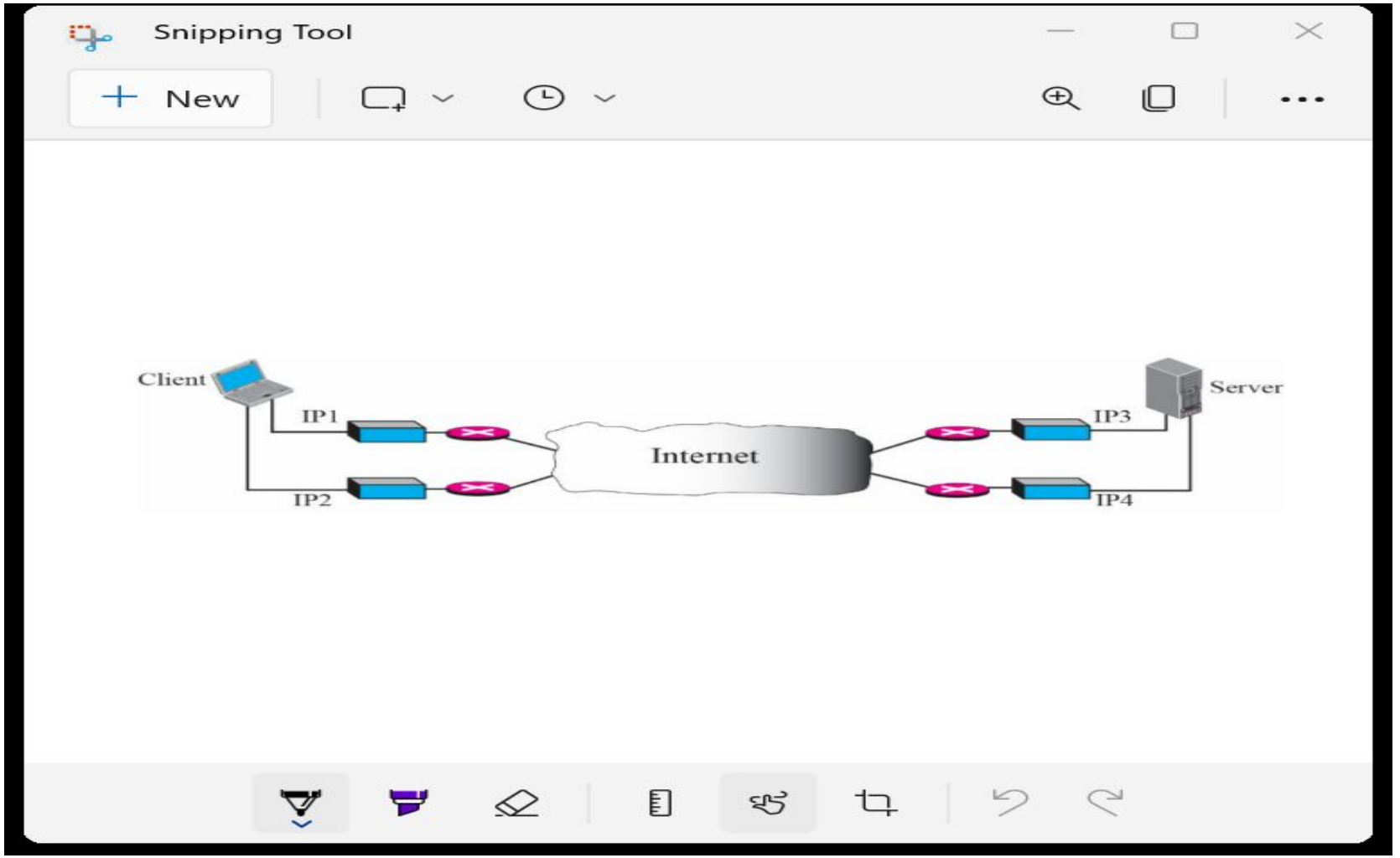
Table 16.1 *Some SCTP applications*

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

Multi- Stream Facility : SCTP provides multi-stream service to each connection, called as association. If one stream gets blocked, then the other stream can deliver the data.



Multihoming



Features of SCTP

1. Transmission Sequence Number (TSN)
The unit of data in SCTP is a **data chunk**. Data transfer in SCTP is controlled by numbering the data chunks. In SCTP, TSN is used to assign the numbers to different data chunks.

2. Stream Identifier (SI)
The **SI** is a **16 bit number** and starts with 0. In SI, there are several streams in each association and it is needed to identify them. Each data chunk needs to carry the SI in the header, so that it is properly placed in its stream on arrival.

3. Packets
In SCTP, the data is carried out in the form of **data chunks** and control information is carried as **control chunks**. Data chunks and control chunks are packed together in the packet.

4. Multihoming
Multihoming allows both ends (sender and receiver) to define **multiple IP addresses** for communication. But, only one of these can be defined as primary address and the remaining can be used as alternative addresses.

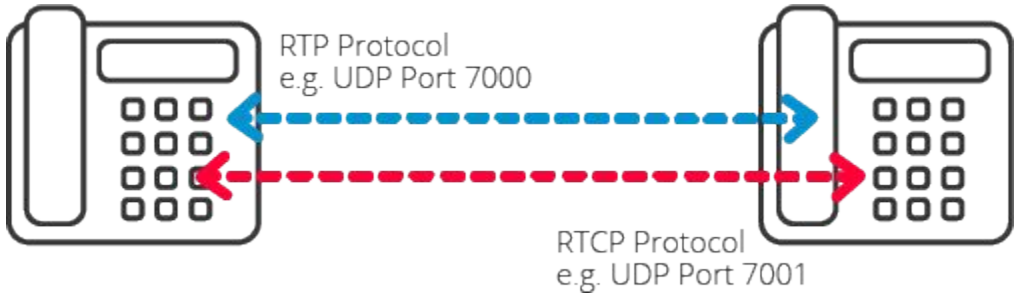
SCTP Packet Structure

RTP

Real-time Transport Protocol defines a standard packet format for delivering audio and video over the Internet.

It is defined in RFC 1889. It was developed by the Audio Video Transport Working group and was first published in 1996.

RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services and web-based push-to-talk features.



RTP is used in conjunction with the RTP Control Protocol (RTCP).

RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams.

RTP is originated and received on even port numbers and the associated RTCP communication uses the next higher odd port number.

Quality of Service (QoS),

Quality of service (QoS) is the use of mechanisms or technologies that work on a network to control traffic and ensure the performance of critical applications with limited network capacity.

It enables organizations to adjust their overall network traffic by prioritizing specific high-performance applications.

QoS is particularly important to guarantee the high performance of critical applications that require high bandwidth for real-time traffic.

QoS characteristics:

Bandwidth

Delay

Reliability

Jitter