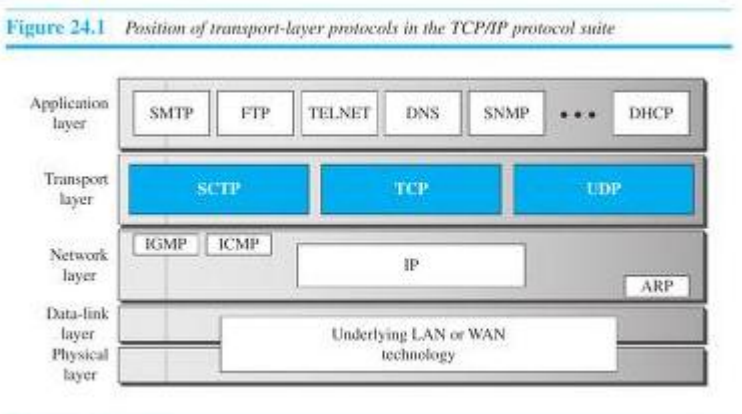
TRANSPORT LAYER

Introduction

The figure shows the position of three protocols in the TCP/IP.



Services

1.UDP-User Datagram Protocol

It is an unreliable transport layer protocol

2.TCP-Transmission control Protocol

It is reliable connection-oriented protocol

Port Numbers

One of the responsibility of transport layer protocol is to create a process to process communication,these protocol use port numbers to accomplish this.

Port numbers provide end to end addresses at the transport layer and allow multiplexing and demultiplexing at this layer.

USER DATAGRAM PROTOCOL

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. it performs very limited error checking. If UDP is so powerless. UDP is a very simple protocol using a minimum of overhead.

Well-Known Ports for UDP

Table shows some well-known port numbers used by UDP. Some port numbers can be used by both UDP and TCP.

Table 23.1 *Well-known ports used with UDP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users

User Datagram

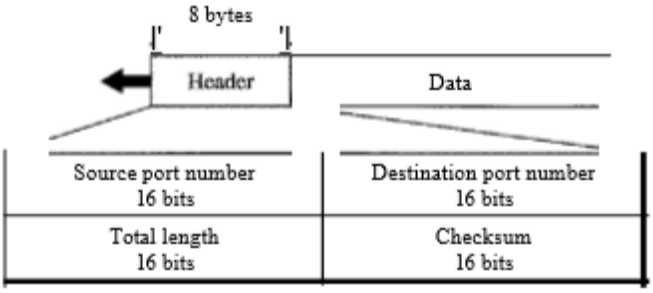
UDP packets, called user datagrams, have a fixed-size header of 8 bytes. Figure shows the format of a user datagram. The fields are as follows:

Source port number: This is the port number used by the process running on the source host. Itis 16 bits long, which means that the port number can range from 0 to 65,535.

Destination port number: This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number.

Length:This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes

Figure 23.9 *User datagram format*



A user datagram is encapsulated in an IP datagram. There is a field in the IP datagram that defines the total length.

UDP length =IP length – IP header's length

Checksum. This field is used to detect errors over the entire user datagram (header plus data).

UDP Services or UDP Operation

UDP uses concepts common to the transport layer.

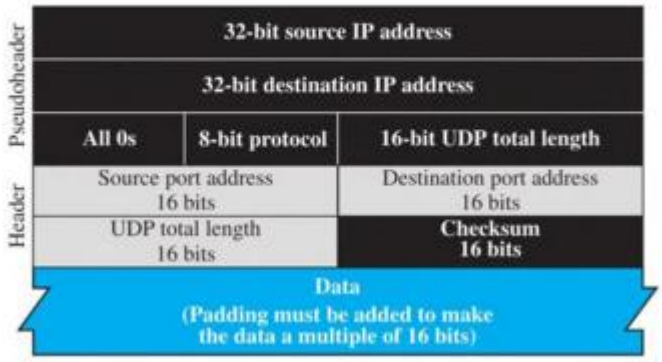
Process to process communication

It provides process to process communiocation using socket addresses,a combination of IP addresses and port numbers.

Checksum

the checksum includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer. The pseudoheader is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with Os.

Figure 24.3 *Pseudoheader for checksum calculation*



Connectionless Services

UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no connection establishment and no connection termination. The user datagrams are not numbered.

Flow and Error Control

UDP is a very simple, unreliable transport protocol. There is no flow control and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know ifa message has been lost or duplicated.

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

Queuing

In UDP, queues are associated with ports.

Congestion control

UDP is connectionless protocol.It doesnot provide congestion control.

Multiplexing and Demultiplexing

Multiplexing done at sender site and demultiplexing at destination site.

Use of UDP

The following lists some uses o fthe UDP protocol:

UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.

UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.

UDP is a suitable transport protocol for multicasting.

UDP is used for management processes such as SNMP

UDP is used for some route updating protocols such as Routing Information Protocol (RIP) .

UDP Applications

UDP Features

1.Connection-less service

UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no connection establishment and no connection termination. The user datagrams are not numbered.Client application needs to send a short request to a serverand to receive a short response.

2.Lack of Error control

UDP does not provide error control.It provides an unreliable service.

3.Lack of Congestion control

UDP does not provide congestion control.UDP does not create additional traffic in an error-prone network.

Transmission Control Protocol(TCP)

TCP is a connectionoriented 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 is called a connection-oriented, reliable transport protocol.

TCP Services

Process-to-Process Communication

Like UDP, TCPprovides process-to-process communication using portnumbers.

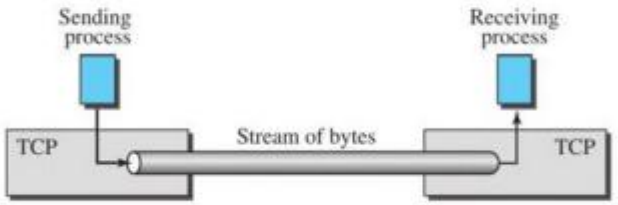
Table 23.2 *Well-known ports used by TCP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
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	FIP, Data	File Transfer Protocol (data connection)
21	FIP, Control	File Transfer Protocol (control connection)
23	TELNET	Tenninal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

Stream Delivery Service

TCP, unlike UDP, is a stream-oriented protocol. TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their data across the Internet.

Figure 24.4 *Stream delivery*



Sending and Receiving Buffers

TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction.

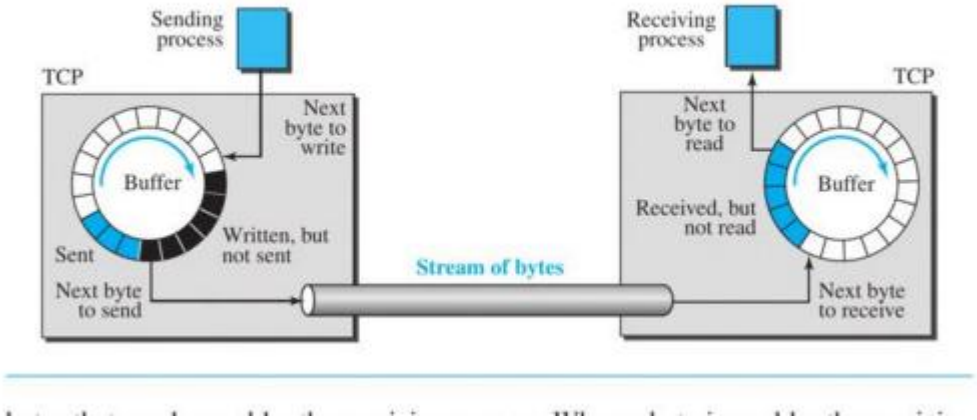
Below Figure shows the movement of the data in one direction.

At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer).

The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment.

The colored area contains bytes to be sent by the sending TCP.

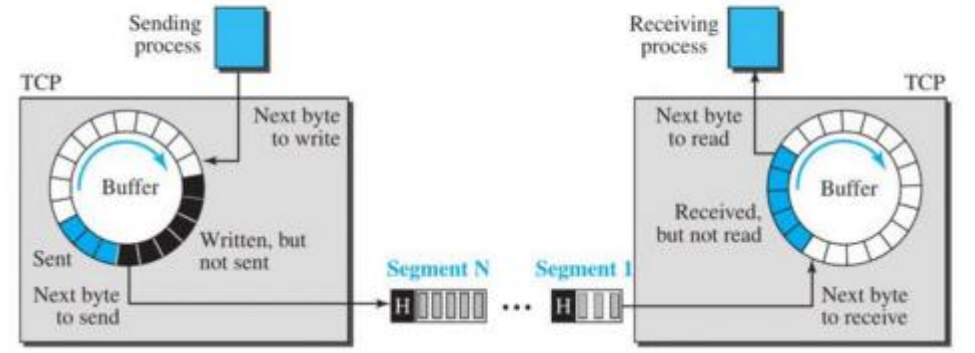
Figure 24.5 *Sending and receiving buffers*



Segments

At the transport layer, TCP groups a number of bytes together into a packet called a segment.

Figure 24.6 *TCP segments*



Note that the segments are not necessarily the same size.

Full-Duplex Communication

TCP offers full-duplex service, in which data can flow in both directions at the same time. EachTCP then has a sending and receiving buffer, and segments move in both directions.

Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.

2. Data are exchanged in both directions.

3. The connection is terminated.

This is a virtual connection, not a physical connection.

Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

TCP Features

Numbering System

There are two fields called the sequence number and the acknowledgment number.

Sequence Number :After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent.The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

Acknowledgment Number : communication in TCP is full duplex; when a connection is established, both parties can send and receive data at the same time. The sequence number in each direction shows the number of the first byte carried by the segment. Each party also uses an acknowledgment number to confirm the bytes it has received. However, the acknowledgment number defines the number of the next byte that the party expects to receive.

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.

Flow Control

TCP, unlike UDP, provides flow control. The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data.

Error Control

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection.

Congestion Control

TCP, unlike UDP, takes into account congestion in the network. The amount ofdata sent by a sender is not only controlled by the receiver (flow control), but is also detennined by the level ofcongestion in the network.

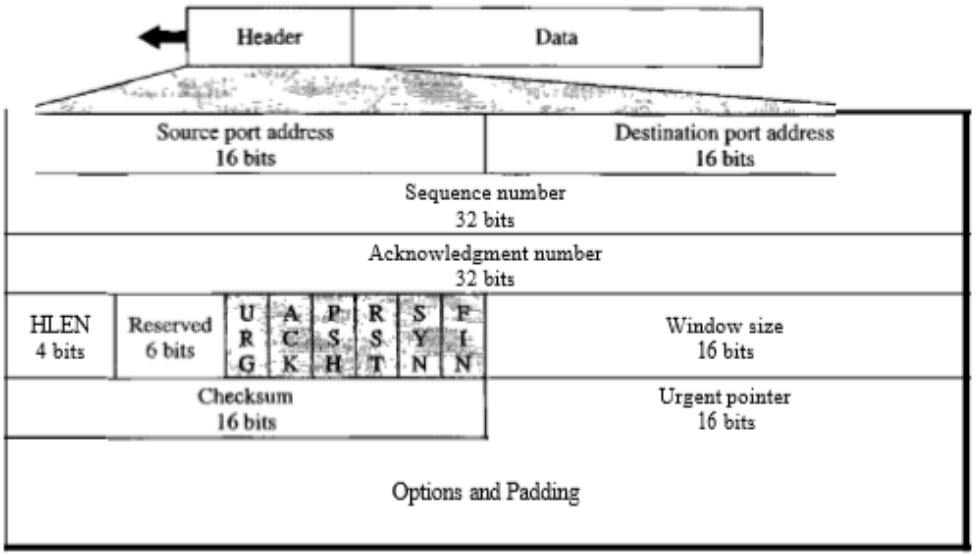
Segment

A packet in TCP is called a segment.

Format

The format of a segment is shown in Figure 23.16.

Figure 23.16 *TCP segment format*



Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

Destination port address. This is a 16-bit field that defines the port number ofthe application program in the host that is receiving the segment.

Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment.

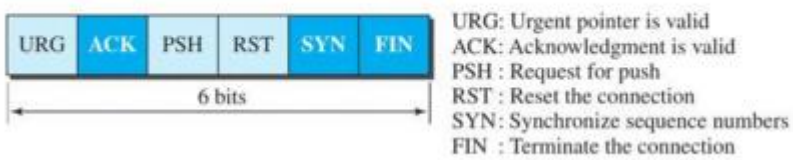
Acknowledgment number. This 32-bit field defines the byte number that the receiver ofthe segment is expecting to receive from the other party.

Header length. This 4-bit field indicates the number of4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes.

Reserved. This is a 6-bit field reserved for future use.

Control. This field defines 6 different control bits or flags as shown in Figure 23.17. One or more ofthese bits can be set at a time.

Figure 24.8 *Control field*



Window size. This field defines the size of the window, in bytes, that the other party must maintain.

Checksum. This 16-bit field contains the checksum.

Urgent pointer. This l6-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.

Options. There can be up to 40 bytes of optional information in the TCP header.

A TCP Connection

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

Connection Establishment

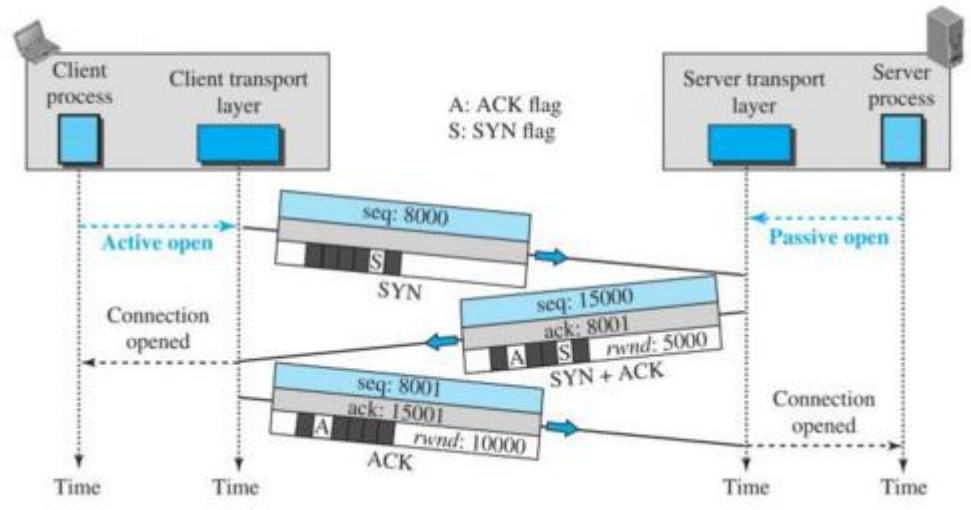
TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.

Three-Way Handshaking The connection establishment in TCP is called three-way handshaking.

In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a passive open.

The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server.

Figure 24.10 *Connection establishment using three-way handshaking*



The three steps in this phase are as follows.

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization ofsequence numbers. Itconsumes one sequence number. When the data transfer starts, the sequence number is incremented by 1.

A SYN segment cannotcarry data, but it consumes one sequence number.

2.The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK. This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.

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

3.The client sends the third segment. This is just an ACK segment. It acknowledges the receipt ofthe second segment with the ACK flag and acknowledgment number field.

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

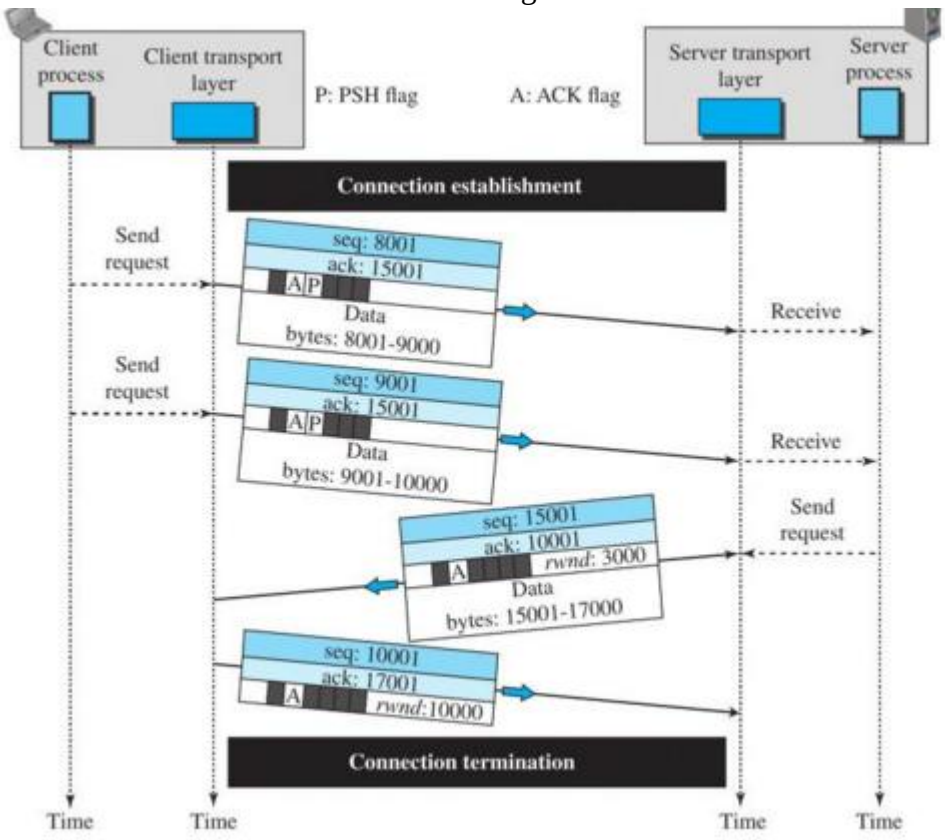
SYN Flooding Attack

The connection establishment procedure in TCP is susceptible to a serious security problem called ***SYN flooding attack.*** This happens when one or more malicious attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses in the datagrams. The

Server can verify that the connection request is coming from valid IP addresses,by using cookie.

Data Transfer

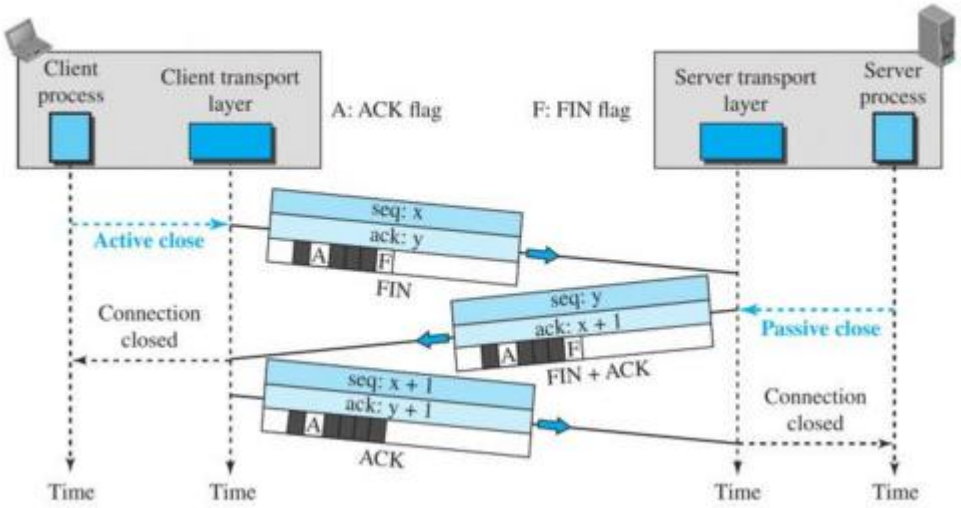
After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments.



Pushing Data: We saw that the sending TCP uses a buffer to store the stream ofdata coming from the sending application program. The sending TCP can select the segment size. The receiving TCP also buffers the data when they arrive and delivers them to the application program when the application program is ready.
The sending TCP must also set the push bit (PSH).
Urgent Data :TCP is a stream-oriented protocol. This means that the data are presented from the application program to TCP as a stream ofbytes.
The solution is to send a segment with the URG bit set. The sending application program tells the sending TCP that the piece ofdata is urgent.
Connection Termination
Any ofthe two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client.
Three-Way Handshaking Most implementations today allow three-way handshaking for connection termination as shown in Figure:

1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set. The FIN segment consumes one sequence number ifit does not carry data.
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirm the receipt ofthe FIN segment from the client and at the same time to announce the closing of the connection in the other direction. The FIN +ACK segment consumes one sequence number ifitdoes not carry data.
3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.
   Half-Close :In TCP, one end can stop sending data while still receiving data. This is called a half-close. Although either end can issue a half-close, it is normally initiated by the client.
   The client half-closes the connection by sending a FIN segment. The server accepts the half-close by sending the ACK segment. The data transfer from the client to the server stops. The server, however, can still send data. When the server has sent all the processed data, it sends a FIN segment, which is acknowledged by an ACK from the client. After half-closing of the connection, data can travel from the server to the client and acknowledgments can travel from the client to the server.

Figure 24.12 *Connection termination using three-way handshaking*



Connection Reset

TCP at one end may deny a connection request, may abort an existing connection, or may terminate an idle connection. All of these are done with the RST (reset) flag.

SCTP-Stream Control Transmission Protocol

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol.It combine some features of UDP and TCP .

SCTP Services

Some important services provided by SCTP are as stated below:

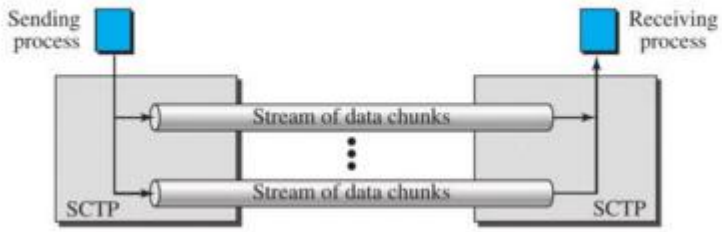
1. Process-to- Process communication

SCTP uses all important ports of TCP.

2. 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.

Figure 24.38 *Multiple-stream concept*



3.Full- Duplex Communication

SCTP provides full-duplex service (the data can flow in both directions at the same time).

4. 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.

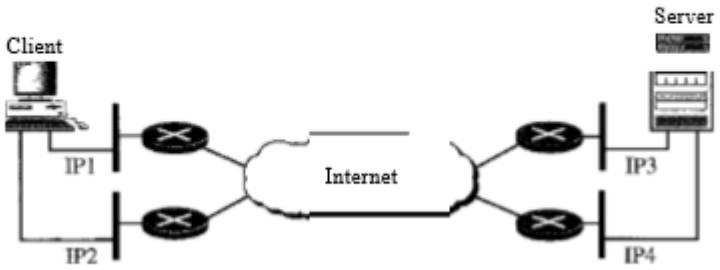
6.Multihoming

The sending and receiving host can define multiple IP addresses in each end for an association.

In the figure ,the client is connected to two local networks with two IP addresses.The server is also connected to two networks with two IP addresses.The client and the server can make

an association using four different pairs of IP addresses.

Figure 23.28 *Multihoming concept*



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.

Features of SCTP

Some important features of SCTP are as stated below:

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.Each data chunk needs identifiers:TSN,SI.

Figure 24.40 *Comparison between a TCP segment and an SCTP packet*