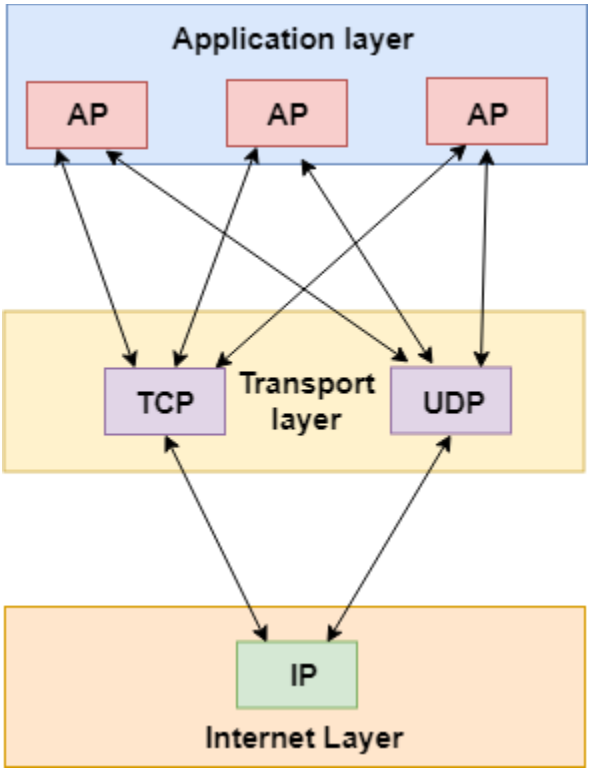
Iv unit

Transport Layer

- The transport layer is a 4th layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.

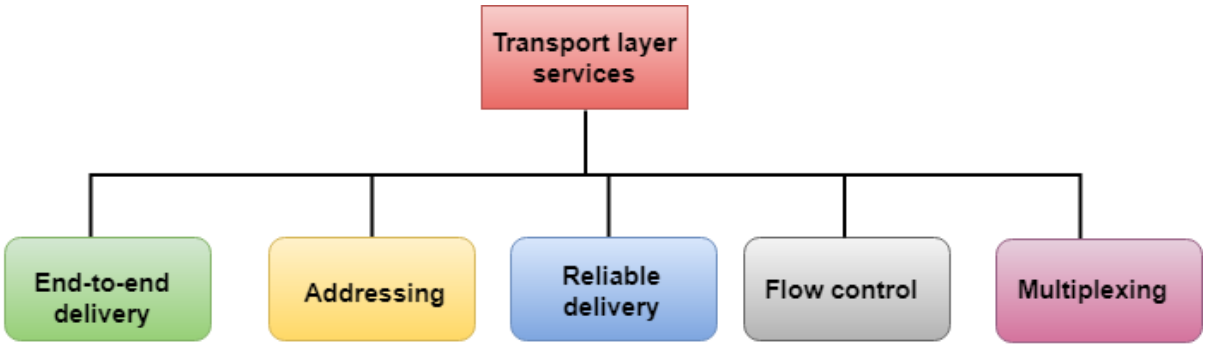


Services provided by the Transport Layer

The services provided by the transport layer are similar to those of the data link layer. The data link layer provides the services within a single network while the transport layer provides the services across an internetwork made up of many networks. The data link layer controls the physical layer while the transport layer controls all the lower layers.

The services provided by the transport layer protocols can be divided into five categories:

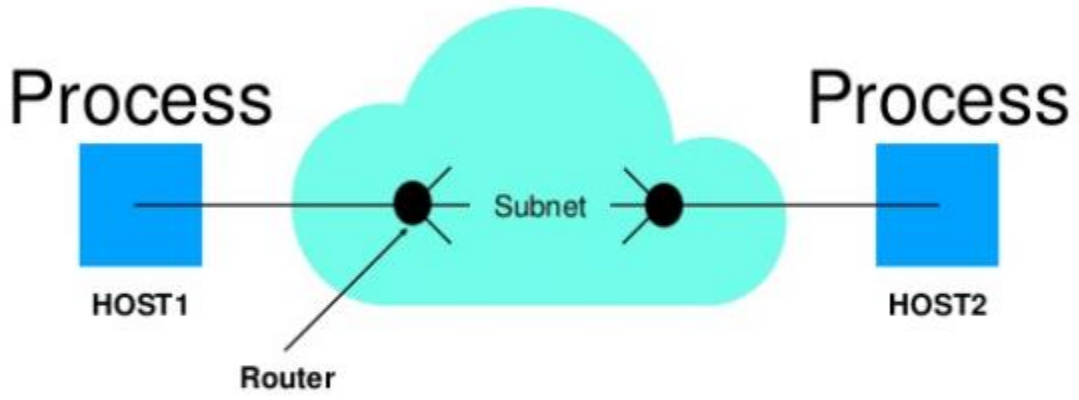
- End-to-end delivery
- Addressing
- Reliable delivery
- Flow control
- Multiplexing



End-to-end delivery:

TRANSPORT LAYER

- To provide reliable, cost effective data transfer from source to destination
- This layer deals with end to end transfer of data
- Here transport entity deals with other host's transport entity.
- Transport layers deals with processes running on the host.



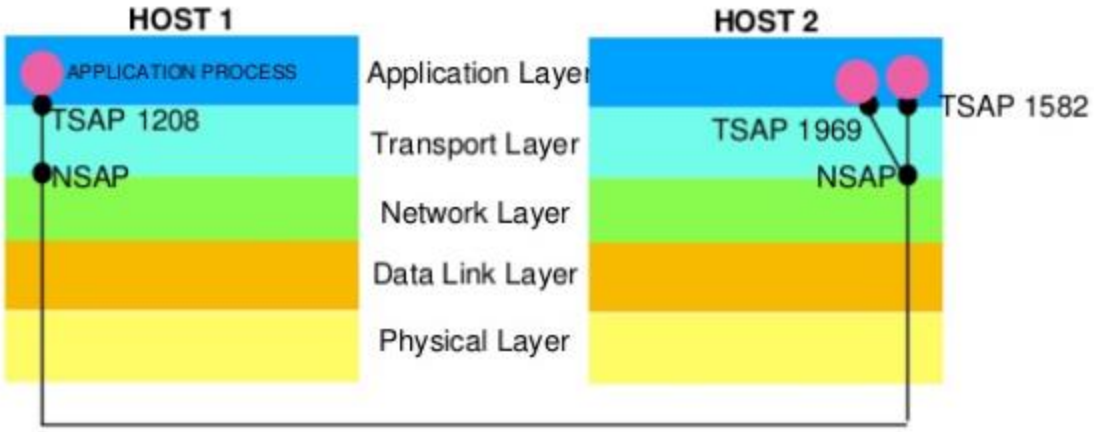
END TO END TRANSMISSION OF PACKET FROM SOURCE TO DESTINATION

Elements of Transport Protocol

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

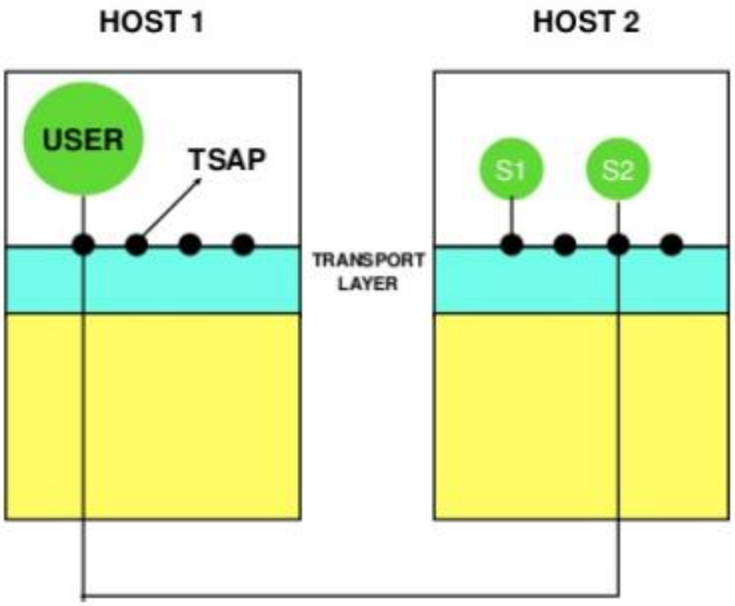
Addressing

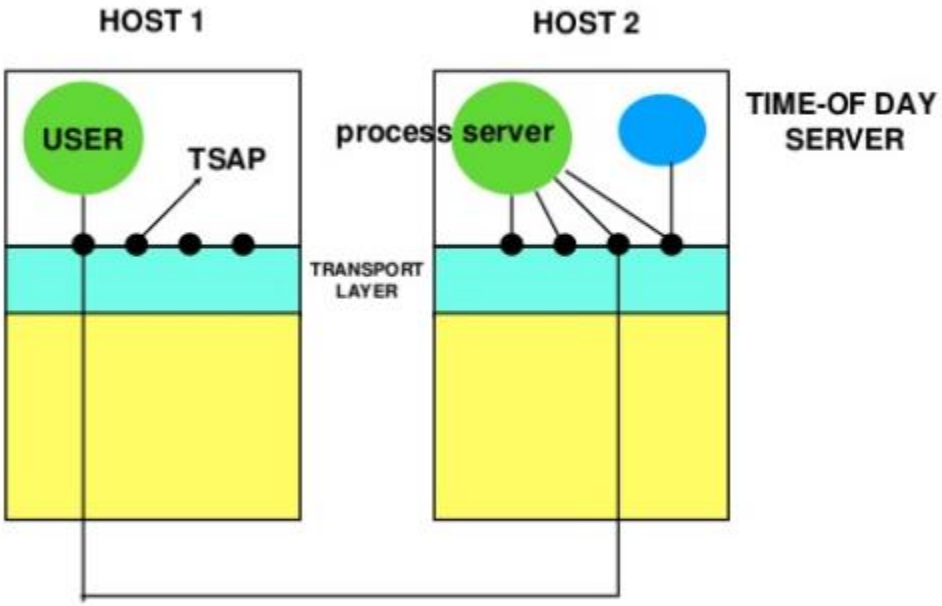
- Application Process is connected to the TSAP
- Entity connects to the NSAP.
- There are multiple processes running within the host.

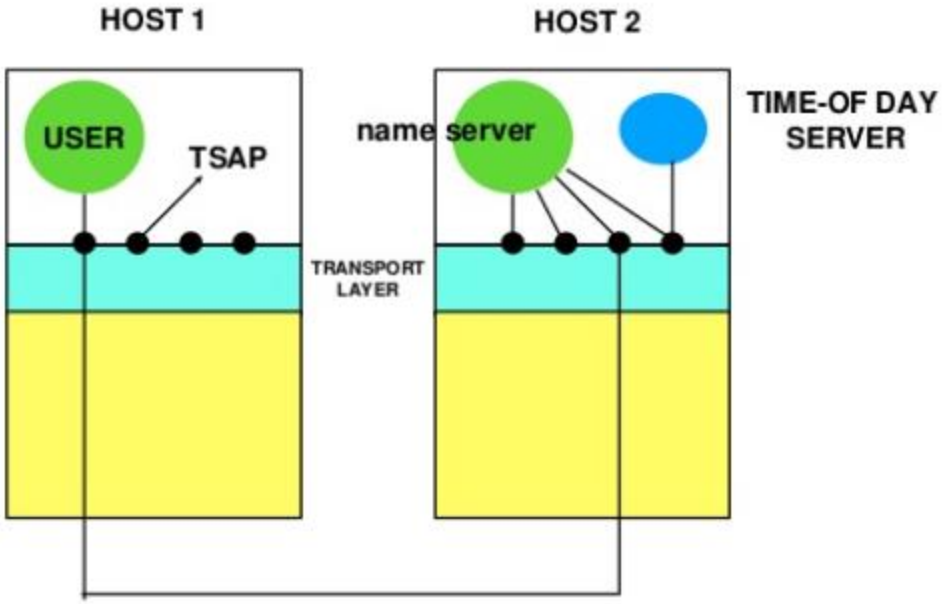


To access a specific service , we have to mention a specific Port Address.
SAP - Service Access Point
TSAP- Transport SAP
NSAP - Network SAP

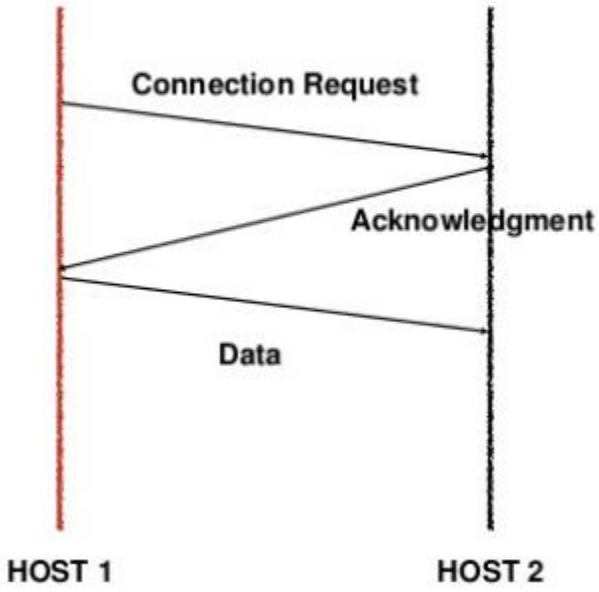
Connection Establishment

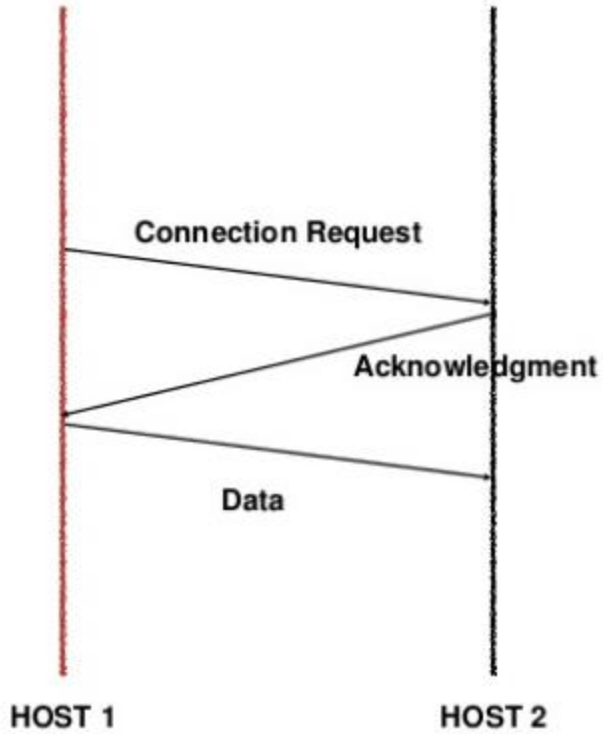






Connection is being established by 3 way handshake

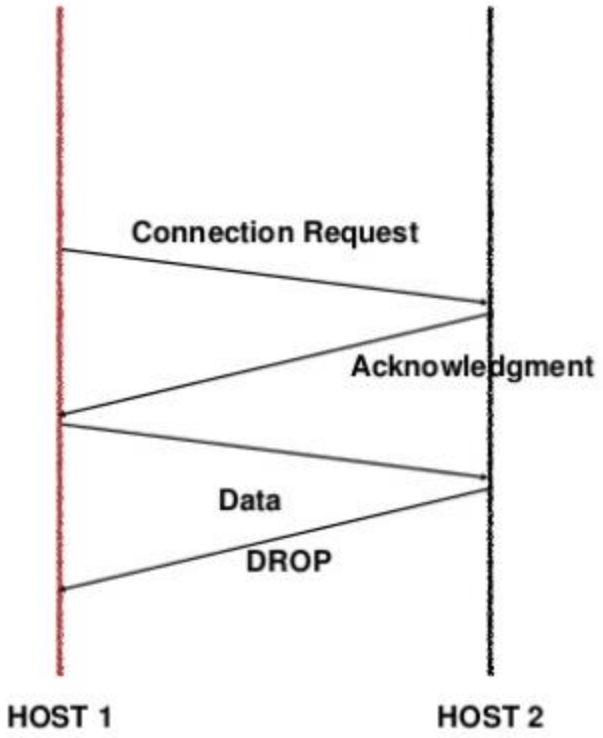




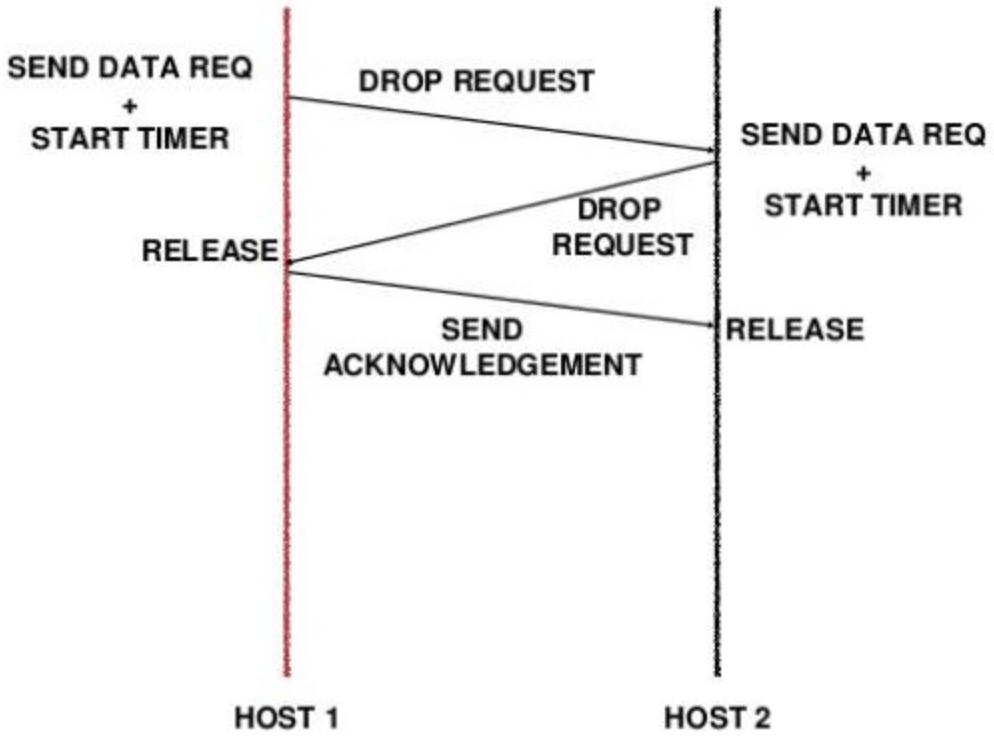
CONNECTION RELEASE

- Disconnection connection between two users.
- Asymmetric Release
- Symmetric Release

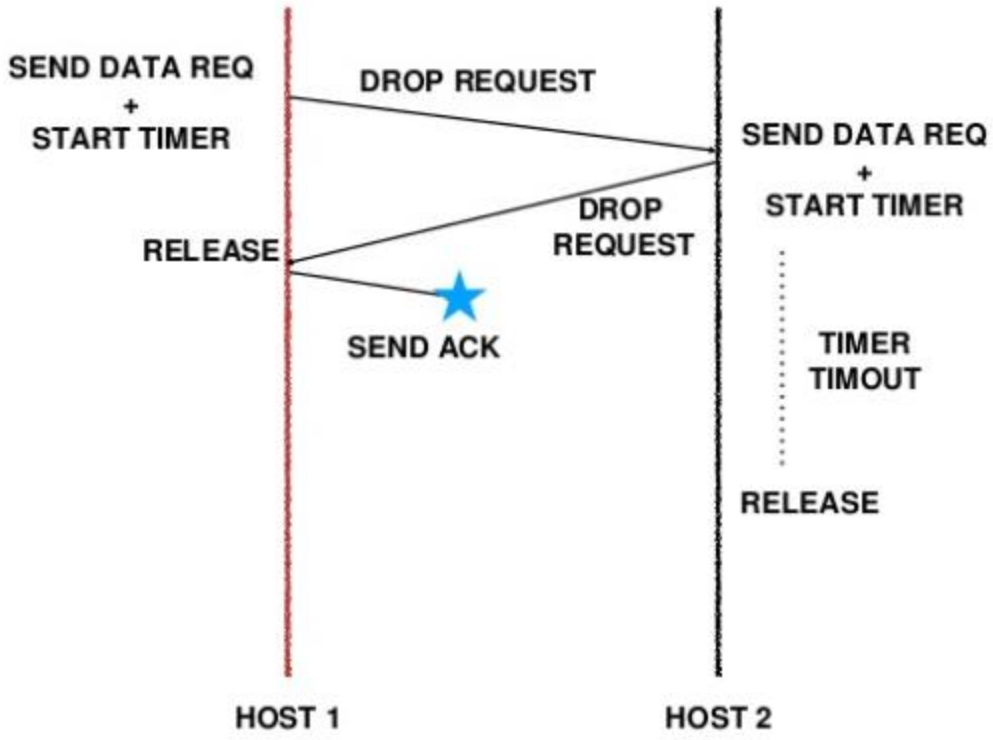
ASYMMETRIC RELEASE



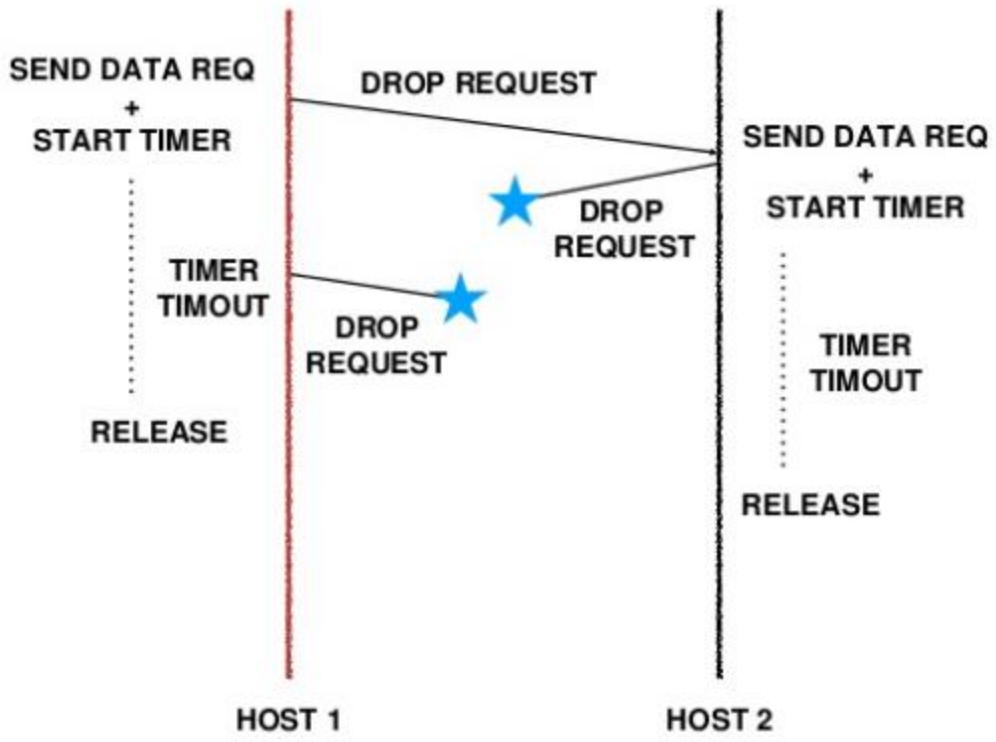
SYMMETRIC RELEASE



SYMMETRIC RELEASE

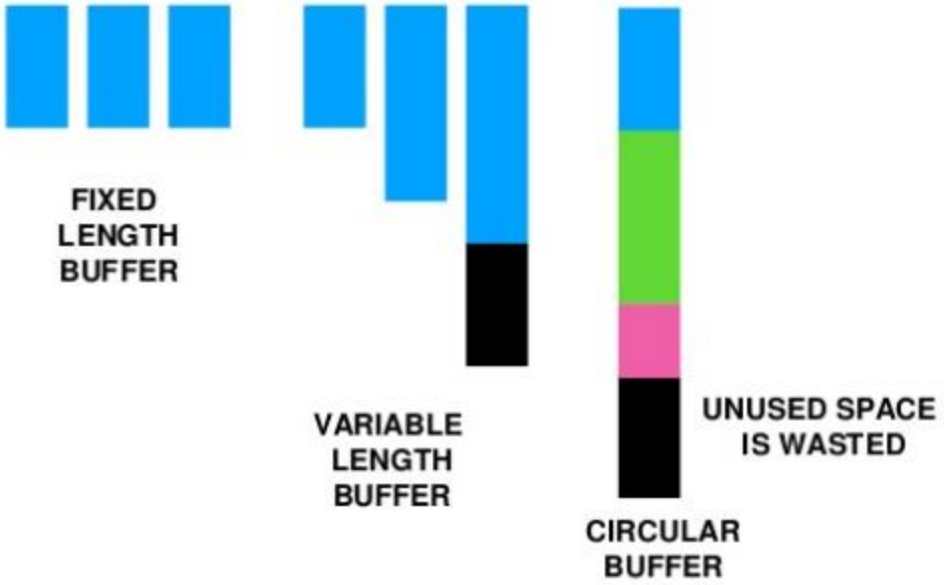


SYMMETRIC RELEASE



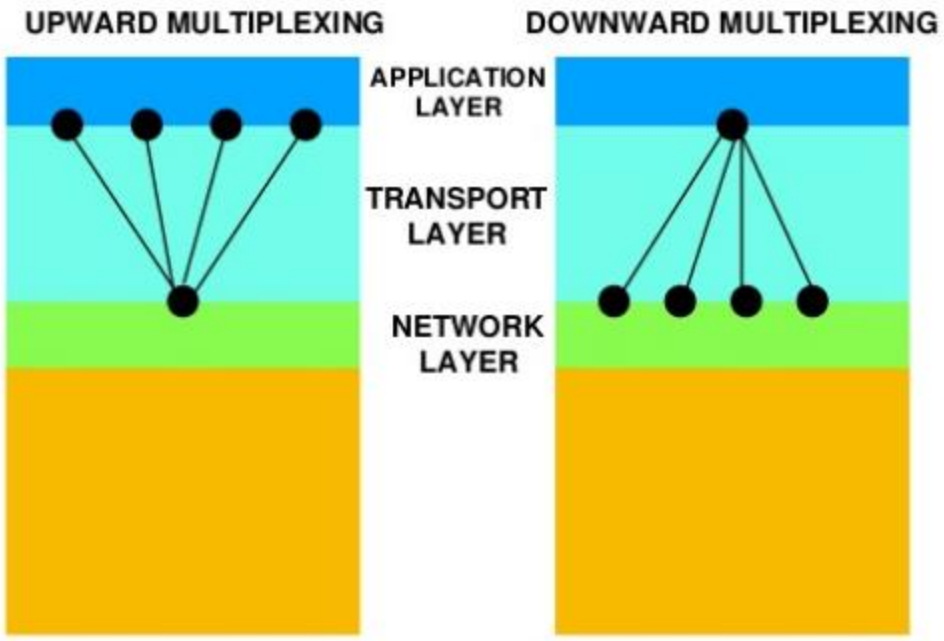
FLOW CONTROL

- To control buffer, Transport Layer manages buffer.



MULTIPLEXING

- UPWARD
- DOWNWARD



CRASH RECOVERY

STRATEGY USED BY SENDING HOST

- Always Retransmit
- First Acknowledgment then write
- Retransmit in S0(Sent 2 messages, Ack of both received). No outstanding Packet is present
- retransmit in S1 (Sent 2 messages, Ack of only 1 received). Here Outstanding Packet is present

TCP and UDP in Transport Layer

Layer 3 or the Network layer uses IP or Internet Protocol which being a connection less protocol treats every packet individually and separately leading to lack of reliability during a transmission. For example, when data is sent from one host to another, each packet may take a different path even if it belongs to the same session. This means the packets may/may not arrive in the right order. Therefore, IP relies on the higher layer protocols to provide reliability.

TCP (Transmission Control Protocol)**:**

TCP is a layer 4 protocol which provides acknowledgement of the received packets and is also reliable as it resends the lost packets. It is better than UDP but due to these features it has an additional overhead. It is used by application protocols like HTTP and FTP.

UDP (User Datagram Protocol)**:**

UDP is also a layer 4 protocol but unlike TCP it doesn't provide acknowledgement of the sent packets. Therefore, it isn't reliable and depends on the higher layer protocols for the same. But on the other hand it is simple, scalable and comes with lesser overhead as compared to TCP. It is used in video and voice streaming.

TCP Vs UDP –

1. **Session Multiplexing:**
   A single host with a single IP address is able to communicate with multiple servers. While using TCP, first a connection must be established between the server and the receiver and the connection is closed when the transfer is completed. TCP also maintains reliability while the transfer is taking place.

   UDP on the other hand sends no acknowledgement of receiving the packets. Therefore, provides no reliability.

2. **Segmentation:**
   Information sent is first broken into smaller chunks for transmission. Maximum Transmission Unit or MTU of a Fastethernet is 1500 bytes whereas the theoretical value of TCP is 65495 bytes. Therefore, data has to be broken into smaller chunks before being sent to the lower layers. MSS or Maximum Segment Size should be set small enough to avoid fragmentation. TCP supports MSS and Path MTU discovery with which the sender and the receiver can automatically determine the maximum transmission capability.

   UDP doesn't support this; therefore it depends on the higher layer protocols for data segmentation.

3. **Flow Control:**
   If sender sends data faster than what receiver can process then the receiver will drop

the data and then request for a retransmission, leading to wastage of time and resources. TCP provides end-to-end flow control which is realized using a sliding window. The sliding window sends an acknowledgement from receiver's end regarding the data that the receiver can receive at a time.
UDP doesn't implement flow control and depends on the higher layer protocols for the same.

4. **Connection Oriented:**
   TCP is connection oriented, i.e., it creates a connection for the transmission to take place, and once the transfer is over that connection is terminated.
   UDP on the other hand is connectionless just like IP (Internet Protocol).

5. **Reliability:**
   TCP sends an acknowledgement when it receives a packet. It requests a retransmission in case a packet is lost.
   UDP relies on the higher layer protocols for the same.

Real Time Transport Protocol (RTP)

A protocol is designed to handle real-time traffic (like audio and video) of the Internet, is known as **Real Time Transport Protocol (RTP)**. RTP must be used with UDP. It does not have any delivery mechanism like multicasting or port numbers. RTP supports different formats of files like MPEG and MJPEG. It is very sensitive to packet delays and less sensitive to packet loss.

Applications of RTP :

1. RTP mainly helps in media mixing, sequencing and time-stamping.
2. Voice over Internet Protocol (VoIP)
3. Video Teleconferencing over Internet.
4. Internet Audio and video streaming.

RTP Header Format :

The diagram of header format of RTP packet is shown below:

The header format of RTP is very simple and it covers all real-time applications. The explanation of each field of header format is given below:

- **Version :**
  This 2-bit field defines version number. The current version is 2.
  1. **P –**
     The length of this field is 1-bit. If value is 1, then it denotes presence of padding at end of packet and if value is 0, then there is no padding.
  2. **X –**
     The length of this field is also 1-bit. If value of this field is set to 1, then its indicates an extra extension header between data and basic header and if value is 0 then, there is no extra extension.
  3. **Contributor count –**
     This 4-bit field indicates number of contributors. Here maximum possible number of contributor is 15 as a 4-bit field can allows number form 0 to 15.
  4. **M –**
     The length of this field is 1-bit and it is used as end marker by application to indicate end of its data.
  5. **Payload types –**
     This field is of length 7-bit to indicate type of payload. We list applications of some common types of payload.
  - **Sequence Number –**
    The length of this field is 16 bits. It is used to give serial numbers to RTP packets. It helps in sequencing. The sequence number for first packet is given a random number and then every next packet's sequence number is incremented by 1. This field mainly helps in checking lost packets and order mismatch.
  - **Time Stamp –**
    The length of this field is 32-bit. It is used to find relationship between times of different RTP packets. The timestamp for first packet is given randomly and then time stamp for next packets given by sum of previous timestamp and time taken to produce first byte of current packet. The value of 1 clock tick is varying from application to application.
  - **Synchronization Source Identifier –**
    This is a 32-bit field used to identify and define the source. The value for this source identifier is a random number that is chosen by source itself. This mainly helps in solving conflict arises when two sources started with the same sequencing number.

Contributor Identifier –

This is also a 32-bit field used for source identification where there is more than one source present in session. The mixer source use Synchronization source identifier and other remaining sources (maximum 15) use Contributor identifier.