Thalagala B.P. 180631J

Home Work : Transport Layer

December 15, 2021

---

1. Why are transport layer services called end-to-end?

Transport layer provides a reliable logical connection between two application processes run in two end stations which is achieved through the protocols implemented inside the Operating Systems. These protocols make sure that the data communication between two application processes reside in two end stations happens in such a way that the relationship among data packets of a given application instance is preserved even though all the applications use a common network interface to send data packets. Port number is used to distinguish different processes of a given host.

2. How do transport layer services differ from network layer services?

Network layer provides host-to-host reliable communication. Regardless of the processes which packets belong, network layer ensures that each packet reaches from its point of origin to its final destination. This is achieved thorough the protocols implemented inside the networking devices such as routers and switches. Network layer uses IP addresses of the source and destination to deliver packets to the desired station.

3. In what way are transport layer services similar with data link layer services?

Transport Layer(TPL) and Data Link Layer(DLL) both provides similar types of services such as Error control, Flow control and Connection establishments. However, most of the properties of the DLL services are predefined(fixed) and there is no flexibility as it provides node-to-node data transmission inside a single link. Whereas the services of TPL can be provided with much flexibility through negotiations done between end hosts as the TPL is implemented inside the Operating Systems of end stations. Therefore, the TPL services are much more sophisticated than the DLL services even though are similar.

4. What are the types of transport services?

Connection-oriented	Reliable	Multicast
Connectionless	Not reliable	

5. What is the theoretical maximum number of Internet connection? Note: IPv4 address and a port together represent the transport address

Number of bits in an IPv4 address	$= 32$
Number of bits in port number	$= 16$
$\therefore$ Number of sources/destinations	$= 2^{32} \times 2^{16}$
$\therefore$ Number of Internet connections	$= (2^{32} \times 2^{16}) \times (2^{32} \times 2^{16}) = 2^{96}$

Because a given source out of $2^{32} \times 2^{16}$ sources can have $2^{32} \times 2^{16}$ destinations.

6. What are the transport layer primitives? How are they related with connection-less service?

TPL Primitives

- **Listen** - Wait for an incoming connection
- **Connect** - Establish a connection
- **Send** - Send data
- **Receive** - Wait for data to arrive
- **Disconnect** - Release connection

In connection-less services there is no need of establishing a connection prior to send data, as the routing happens on per-packet basis. Therefore from the above primitives, the primitives related to connection establishment(*Connect*) and connection release(*Disconnect*) are not required in connection-less service. Other primitives are required.

7. How is a connection established before starting a communication? It may not happen at once. What are the possible issues that can exist? How is it solved?

To establish a connection, client first sends a connection request and server should accept that connection request. This connection establishment process also known as 3-way Handshaking as it makes use of following three segments

1. `SYN`: (Synchronization) segment sends by sender to initialize connection which includes some random number which will be used by sender to start numbering segments.
2. `SYN + ACK`: (Acknowledgment and Synchronization) sends by receiver to inform the sender that it receives connection request(first SYN) and the random number that the receiver will be using to start numbering its segments.
3. `ACK`: (Acknowledgment) sends by the sender again to acknowledge the receivers connection request as the TCP connections are full duplex.

This will happen at once only if the network is reliable. However network may not reliable sometimes and packets may get delayed, lost or duplicated.

In order to overcome those issues, sequence number is associated with each Protocol Data Unit(PDU-Segment) of the transport layer. These sequence numbers are assigned by both the parties for their segments respectively starting from some random integer specified when sending the synchronization segments.

8. Draw diagrams for following scenarios (you may use a diagramming tool and include an image in the document, or you may draw on a piece of paper, take a photo and include it in the document)

8.1. Connection establishment

- Normal case of a three-way handshake
- Old CONNECTION REQUEST appearing out of nowhere
- Duplicate CONNECTION REQUEST and duplicate ACK

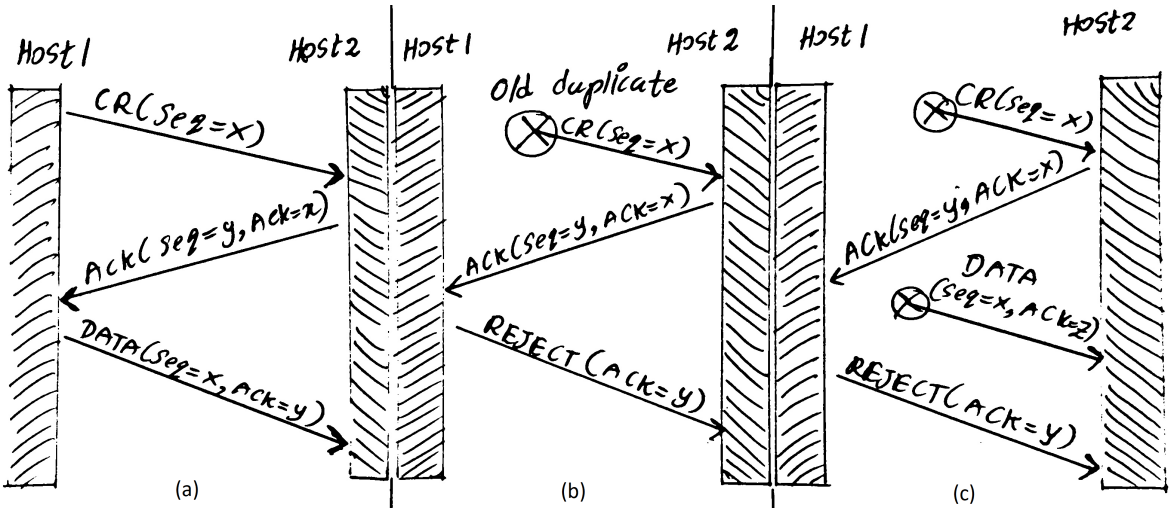

Figure 1: **Connection establishment**: (a) Normal case of a three-way handshake, (b) Old CONNECTION REQUEST appearing out of nowhere and (c) Duplicate CONNECTION REQUEST and duplicate ACK

8.2. Connection Release

- Normal case of a three-way handshake
- Final ACK lost
- Response lost
- Response lost and subsequent DRs lost

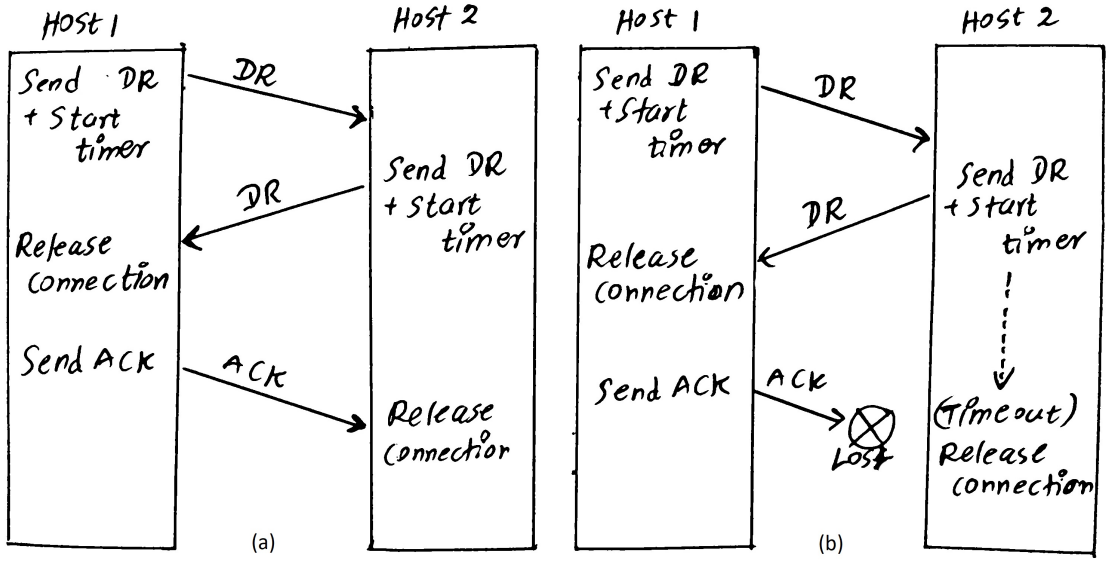

Figure 2: **Connection Release**: (a) Normal case of a three-way handshake and (b) Final ACK lost

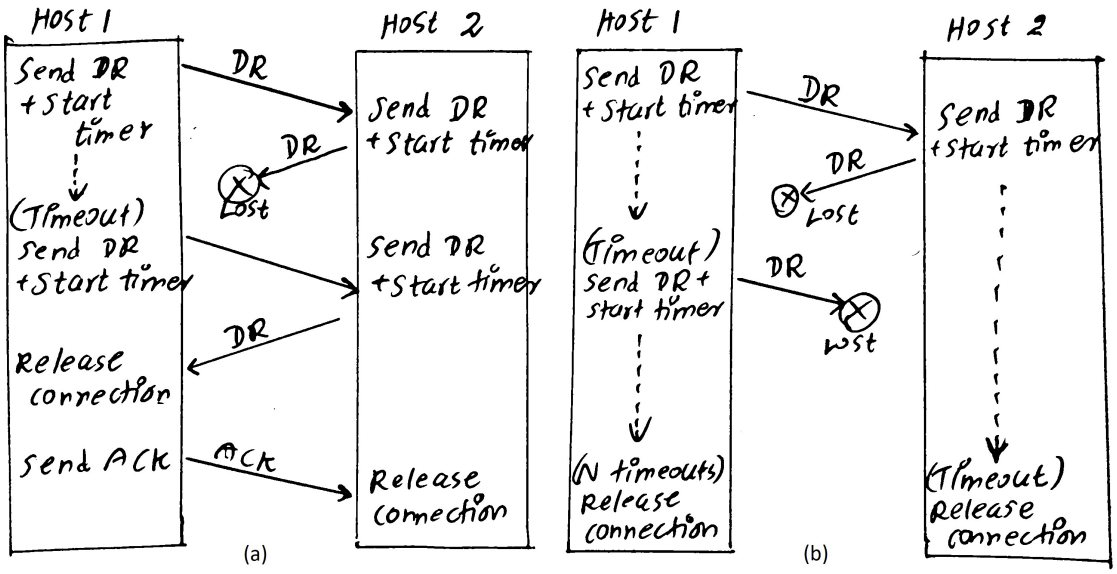

Figure 3: **Connection Release**: (a) Response lost and (b) Response lost and subsequent DRs lost

9. What is the objective of having flow control mechanism?

When the sender sends data in a rate that can not be handled by the receiver packets get lost, as the receiver is not capable of processing the packets in the same rate as it receives. In a situation like this data flow must be controlled through negotiations between end stations to ensure a reliable and efficient delivery of data. Otherwise retransmissions have to be done frequently for lost packets which ultimately reduces the efficiency of communication.

10. How does buffer help controlling the flow?

Rather than sending the receiving data directly to the processing unit of the receiver, a buffer can be introduced in between to store the incoming packets before processing. In this way receiver does not have to process data in the same rate it receives in real time and receiver can process them in its own rate without any packet loss as long as there is sufficient buffer to store incoming packets.