TRANSPORT LAYER

The transport layer is Layer 4 of the Open Systems Interconnection (OSI) communications model. It is responsible for ensuring that the data packets arrive accurately and reliably between sender and receiver. The transport layer most often uses TCP or User Datagram Protocol (UDP). In the TCP/IP network model, the transport layer comes between the application and network layers.

In the OSI model, the transport layer sits between the network layer and the session layer. The network layer is responsible for taking the data packets and sending them to the correct computer. The transport layer then takes the received packets, checks them for errors and sorts them. Then, it sends them to the session layer of the correct program running on the computer. The session layer now takes the well-formatted packets and uses them for the application's data.

Working of Transport Layer

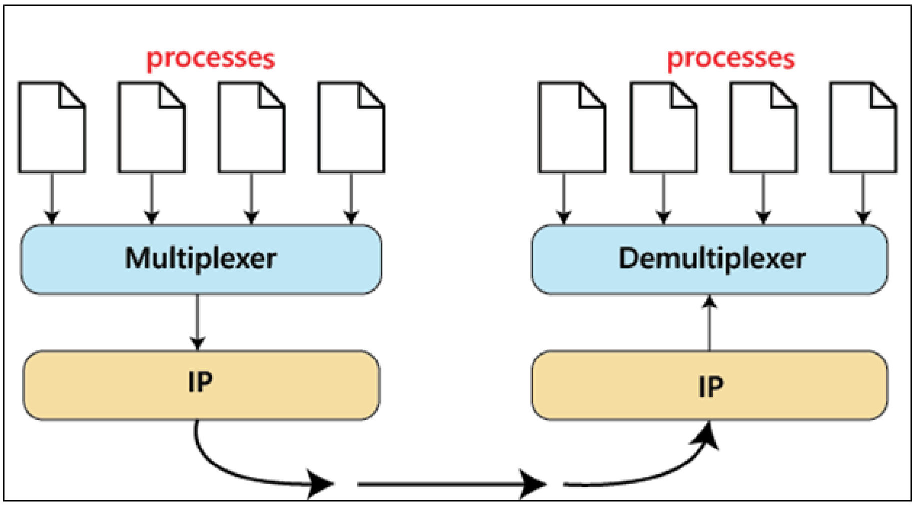
The transport layer takes services from the Application layer and provides services to the Network layer.
At the sender's side:
The transport layer receives data (message) from the Application layer and then performs Segmentation, divides the actual message into segments, adds the source and destination's port numbers into the header of the segment, and transfers the message to the Network layer.
At the receiver's side:
The transport layer receives data from the Network layer, reassembles the segmented data, reads its header, identifies the port number, and forwards the message to the appropriate port in the Application layer.

Responsibilities of a Transport Layer

- The Process to Process Delivery
- End-to-End Connection between Hosts
- Multiplexing and Demultiplexing
- Congestion Control
- Data integrity and Error correction
- Flow control

1. The Process to Process Delivery

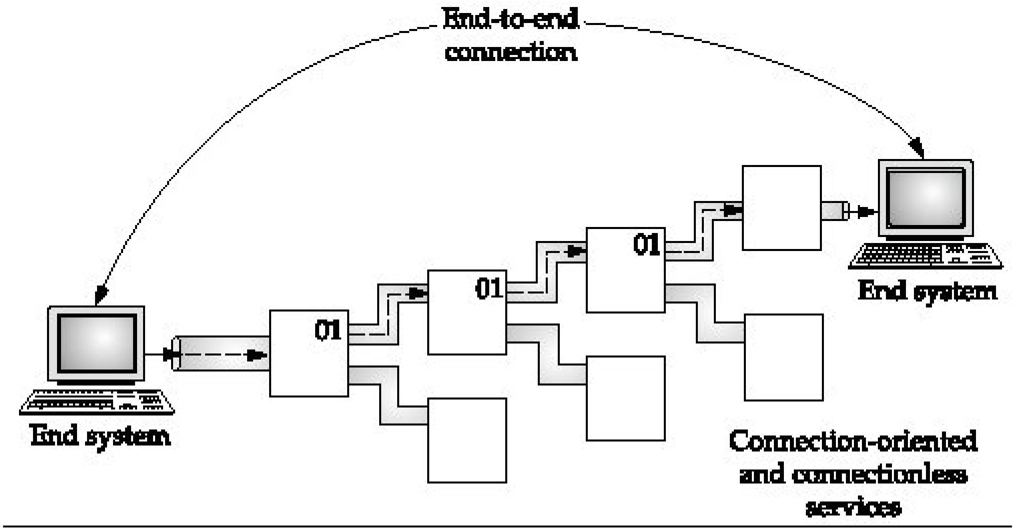
While Data Link Layer requires the MAC address (48 bits address contained inside the Network Interface Card of every host machine) of source-destination hosts to correctly deliver a frame and the Network layer requires the IP address for appropriate routing of packets, in a similar way Transport Layer requires a Port number to correctly deliver the segments of data to the correct process amongst the multiple processes running on a particular host. A port number is a 16-bit address used to identify any client-server program uniquely.



Process to Process Delivery

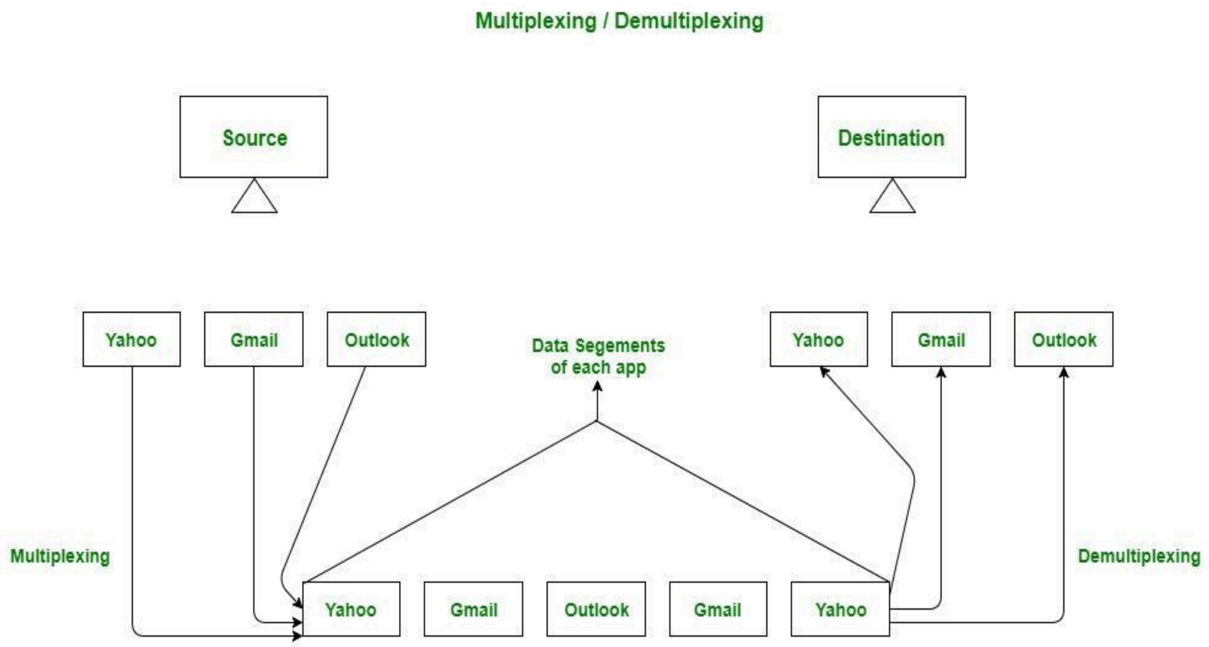
2. End-to-end Connection between Hosts

The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP. TCP is a secure, connection-orientated protocol that uses a handshake protocol to establish a robust connection between two end hosts. TCP ensures the reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol that ensures best-effort delivery. It is suitable for applications that have little concern with flow or error control and requires sending the bulk of data like video conferencing. It is often used in multicasting protocols.



3. Multiplexing and Demultiplexing

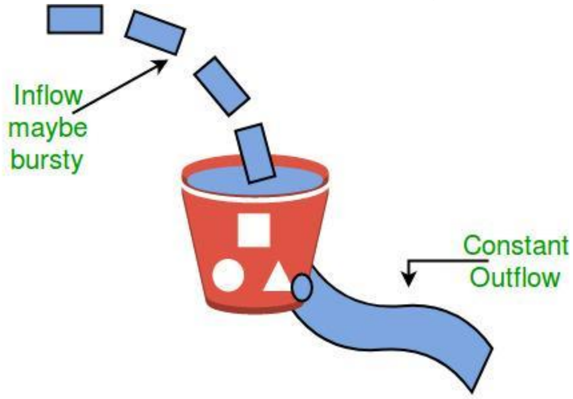
Multiplexing(many to one) is when data is acquired from several processes from the sender and merged into one packet along with headers and sent as a single packet. Multiplexing allows the simultaneous use of different processes over a network that is running on a host. The processes are differentiated by their port numbers. Similarly, Demultiplexing(one to many is required at the receiver side when the message is distributed into different processes. Transport receives the segments of data from the network layer distributes and delivers it to the appropriate process running on the receiver's machine.



Multiplexing and Demultiplexing

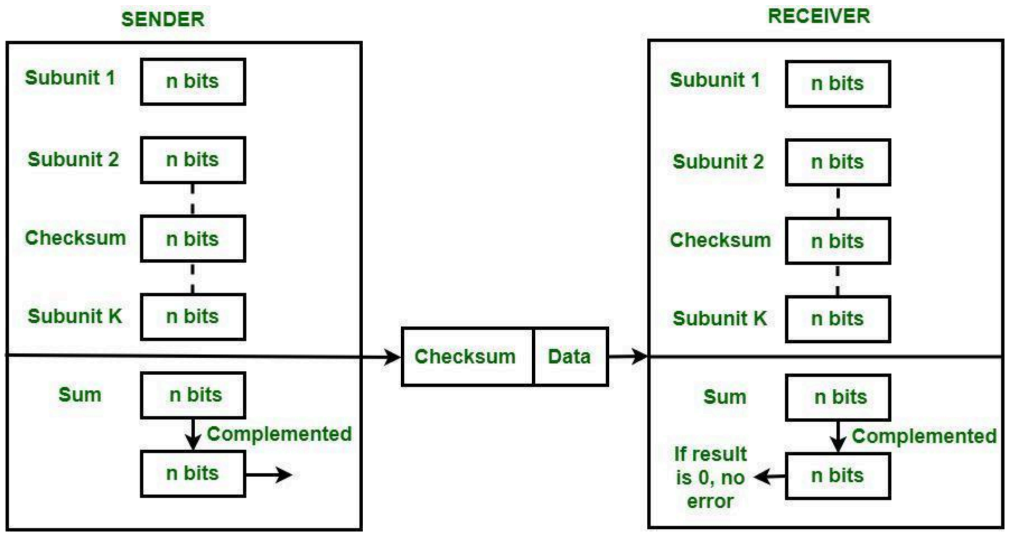
4. Congestion Control

Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occurs. As a result, the retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides Congestion Control in different ways. It uses open-loop congestion control to prevent congestion and closed-loop congestion control to remove the congestion in a network once it occurred. TCP provides AIMD – additive increases multiplicative decrease and leaky bucket technique for congestion control.



5. Data integrity and Error Correction

The transport layer checks for errors in the messages coming from the application layer by using error detection codes, and computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.



Error Correction using Checksum

6. Flow Control

The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

Protocols of Transport Layer

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Stream Control Transmission Protocol (SCTP)
- Datagram Congestion Control Protocol (DCCP)
- AppleTalk Transaction Protocol (ATP)
- Fibre Channel Protocol (FCP)
- Reliable Data Protocol (RDP)
- Reliable User Data Protocol (RUDP)
- Structured Steam Transport (SST)
- Sequenced Packet Exchange (SPX)