

# Functional Overview of Transport Layer or Duties of the Transport Layer

The Transport layer is the first layer from the bottom that is implemented and processed only on the communicating end nodes and not on intermediate nodes. While the physical, data link and network layer protocols are implemented and processed on end nodes as well as on intermediate nodes (Routers), the transport and application layers are implemented and processed only inside end computers (source and destination nodes). This means that intermediate network nodes like Routers are not involved in processing of the transport layer headers inside an IP datagram.

## Positioning of Transport Layer :

The transport layer sits between the application layer and the network layer. On the sending node, the transport layer takes data from different applications, segments the data, adds the transport header and hands over the segment to the underlying network layer, to be delivered to the receiving node. On the receiving node, the transport layer receives segments from the underlying network layer, process the transport header and then hands over the application data to an application layer process.

## Functions of Transport Layer :

The transport layer protocols are primarily responsible for

- **End-to-End application data delivery** between the source and destination computers using the underlying unreliable best-effort IP based networks. It provides either a byte stream or message based service to the application layer protocols. On the sending side, application data is split into a byte stream or into a series of message blocks, a transport layer header added and then transport layer segments are sent to the underlying IP layer in the sending node, to be delivered to the network. Once the IP network delivers the transport layer segments to the receiving machine, the transport layer process at the receiving machine processes the transport headers for reliability, flow and error controls and then hands over the application data to the appropriate application process.
- **Application layer protocol Multiplexing/Demultiplexing** for multiple applications communicating between end nodes. Multiplexing/Demultiplexing is provided using transport layer port numbers for providing multiple services like Email, Web browsing, file transfer over the same IP stack implementation on end nodes. For example, HTTP uses the reserved TCP port number 80, SMTP uses the reserved TCP port number 25, FTP control connection uses the reserved port number 21 etc. Similarly, DNS uses the reserved UDP port number 53, SNMP uses the reserved UDP port numbers 161 and 162. Normally,

the transport layer header contains fields named as source port number and destination port number to uniquely identify the sending and the receiving processes respectively.

- **End-End Reliability** for making sure that every byte of sender's application data reaches the receiver's application, in order, in spite of travelling over a wide variety of unreliable telecommunication links. Reliability is taken care by using mechanisms like checksums, sequence numbers, acknowledgement numbers, timeouts and retransmissions.
- **End-End Flow control** for making sure that the sender sends data at a rate that the receiver can process and store. This is usually achieved by the receiver advertising its current receiver buffer size continuously to the sender.
- **End-End Congestion control** for making sure that the sender does not introduce congestion in the intermediate network links and router buffers. Mechanisms like feedbacks from intermediate routers/receiver and monitoring receiver packet loss are used for controlling the sender rate so as not to congest the network links and router buffers.
- **Provides unique communication end points in the form of Sockets** for applications to use the Networking stack of the machine to communicate externally
- TCP, UDP are the most popular transport layer protocols