

The Transport Layer

The Transport Layer is responsible for end-to-end communication. In other words, node-to-node protocols that operate over the underlying subnetwork, providing end-to-end services to the layers above.

- Ideally, the Network Layer is transparent to the Transport layer.
- The service model in the transport layer are the types of things that you see in the socket layer interface:
 - Connection-oriented or connectionless protocols.
 - Reliable or unreliable protocols.
 - Multiplexing methods.

Universal Datagram Protocol (UDP)

An unreliable, connectionless protocol.

Source Port	Destination Port
Checksum	Length
Data	

UDP Packet Format

- Checksum is the same as the IP checksum with the first four bytes of the UDP header.
- Length is the number of data bytes.

Transport Control Protocol (TCP)

A reliable, connection-oriented stream protocol.

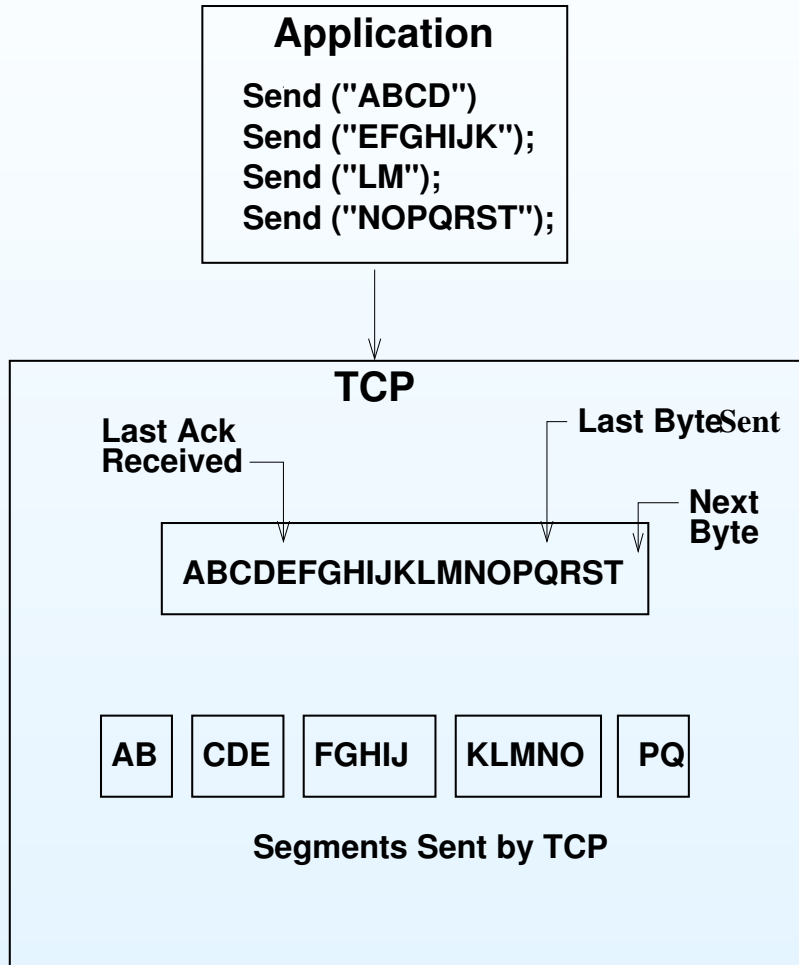
Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Header Length	0	Flags	Advertised Window Size
Checksum		Urgent PTR	
Options ...			
Data			

TCP Packet Format

TCP Packet Fields

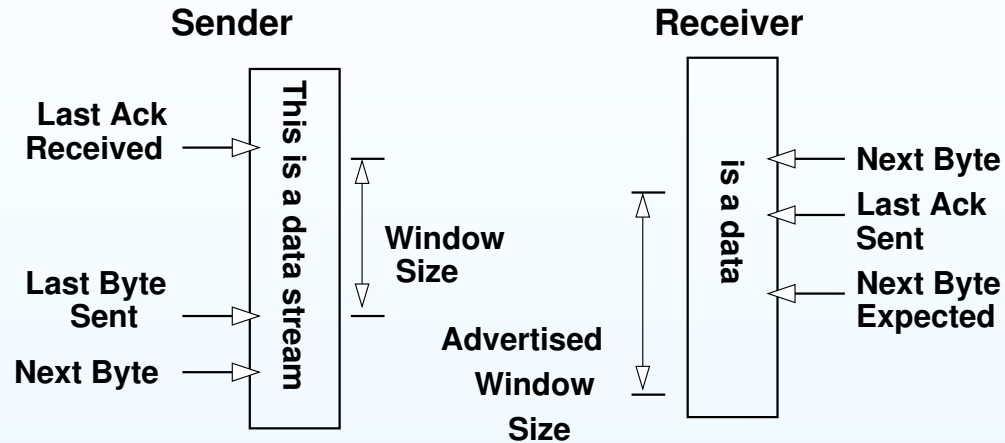
- Source and Destination Ports for multiplexing processes.
- Sequence and Acknowledgement numbers for sliding window protocol.
- Header length in 32 bit words.
- Flags
 - SYN - connect request
 - FIN - disconnect request
 - RESET - reset connection
 - PUSH - upper layer forced
 - URG - urgent pointer valid
 - ACK - ACK field valid
- Checksum is the IP checksum plus the first 20 bytes of header.
- Length is length of the data in bytes.

TCP Segments



TCP Segmentation

TCP Window Management



TCP Window Management

MSS = Maximum Segment Size is the largest acceptable segment size.

WS = Window Size (sender)

AWS = Advertised Window Size (receiver)

Next Byte = next location to put (sender), get(receiver) data.

LBS = Last Byte Sent

LAR = Last Ack Received

L = Message Length

TCP SWP Example

- MSS = 8 bytes
- AWS = 20 bytes
- Data: (with space = '#')

Humpty#	Dumpty#sat#	on#	a	#wall,#Humpty#	Dumpty#had#a#	great#fall.
---------	-------------	-----	---	----------------	---------------	-------------

Urgent Data

If you want to send urgent data or data to be read out-of-order,

- Set the URG flag on the send - flags = MSG_OOB
- The TCP sender will put the data in the buffer and send it with the Urgent Pointer pointing to the end of the data.
- The urgent data may be in a packet with non-urgent data, but it will always be at the beginning.
- The *Urgent PTR* field points to the end of the urgent data in the packet.

Interesting Things

- If a segment timer expires, a different segment could be sent.
- Sequence numbers are randomized on the connect. For example, at 100 Mbps, it takes about 6 minutes to wrap around.