RFC SUMMARY

RFC-9293 TCP

A SUMMARY OF RFC-9293

JORDAN PATTERSON

What is TCP and when was it first specified?

- TCP is an important and widely implemented transport-layer protocol that has evolved as the internet and its uses have grown. TCP was first specified in RFC 793 in September 1981. The current RFC, 9293, replaces RFC 793 and several others, bringing all information regarding TCP into one document.

What are the key concepts of TCP?

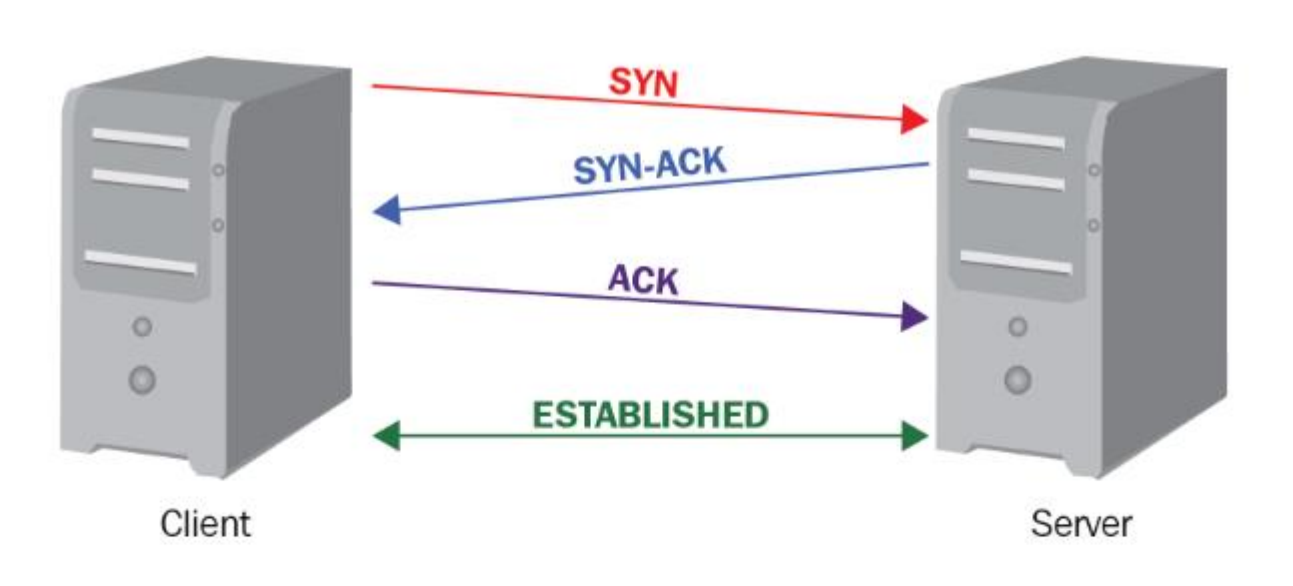
- TCP provides reliable, in-order, byte-stream service to applications.
- The application byte-stream is conveyed over the network via TCP segments, with each TCP segment sent as an IP datagram.
- TCP can detect packet losses via sequences numbers and errors via per-segment checksums. Losses and errors are corrected via retransmission.
- TCP supports unicast. Some anycast applications can use TCP without modification but there are risks of instability due to lower layer forwarding behavior.
- TCP is connection oriented.
- Data flow is bidirectional. An application may choose to send data unidirectionally if it chooses.
- TCP uses port numbers to identify application services and to multiplex distinct flows between hosts.

TCP Header

- The TCP header consists of several fields, including Source Port, Destination Port, Sequence Number, Acknowledgment Number, Data Offset, Control Bits, Window, Checksum, Urgent Pointer, and Options. This is then followed by the user data.
- There are 8 different control bits, also known as Flags. They are CWR, ECE, URG, ACK, PSH, RST, SYN, and FIN

Establishing a TCP Connection

- A TCP connection is established by a Three-Way Handshake. The connection is initiated by one host and responded to by another. One host sends a SYN, the other responds with a SYN-ACK, and the first host then confirms with an ACK.



TCP Connection States

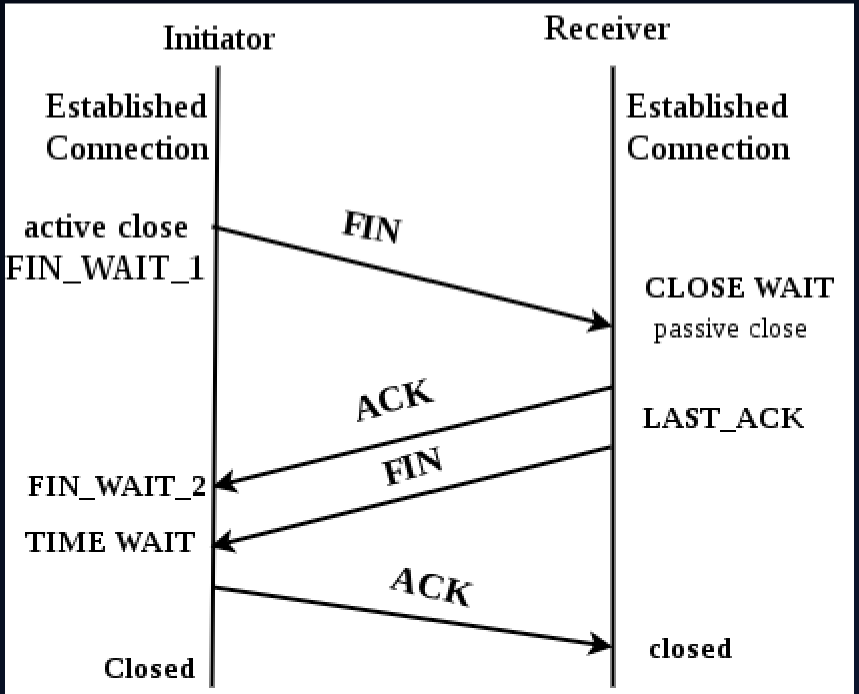
- A TCP connection will go through several states during its lifetime. These different states are the result of events such as user calls (OPEN, SEND, RECEIVE, CLOSE, ABORT, STATUS), incoming segments (containing SYN, ACK, RST, FIN flags), and timeouts.

TCP Connection States

- LISTEN - Waiting for a connection request from any remote TCP peer and port.
- SYN-SENT - Waiting for a matching connection after having sent a connection request.
- SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
- ESTABLISHED - An open connection. This is the normal state for the data transfer phase of the connection.
- FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP peer or the acknowledgement of the connection termination request that was already sent.
- FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP peer.
- CLOSE-WAIT - Waiting for a connection termination request from the user.
- CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP peer.
- LAST-ACK - Waiting for ab acknowledgement of the connection termination request previously sent to the remote TCP peer.
- TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP peer received the acknowledgement of its connection termination request.
- CLOSED - The connection state is closed.

Closing a TCP Connection

- A TCP Connection is closed when one of the hosts sends the FIN flag in the TCP Header. The process is similar to the Three-Way Handshake used to establish a connection but in this case the connection will close.



TCP Privacy and Security

- TCP lacks built-in cryptographic capabilities and so it relies on external protocols like IPsec and TLS for security. While non-cryptographic enhancements exist, IPsec is the most effective for comprehensive protection.