**CSCI 367 - Computer Networks I**

TCP – Transmission Control Protocol (Layer 4/Transport Layer)

References:

<https://en.wikipedia.org/wiki/Transmission_Control_Protocol>

<https://networklessons.com/cisco/ccie-routing-switching-written/tcp-header>

<https://serverfault.com/questions/329845/how-to-forcibly-close-a-socket-in-time-wait>

1. A TCP segment is encapsulated in the IP datagram’s payload.
2. A TCP segment consists of a header section and a payload section.
3. The header of a TCP segment can range from 20-60 bytes. 40 bytes are for options. If there are no options, a header is 20 bytes else it can be of upmost 60 bytes.  The following are the TCP header fields:

**Source Port Address**   
A 16-bit field that represents the port number of the application on the host that’s sending the data.

**Destination Port Address**  
A 16-bit field that represents the port number of the application on the destination host.

**Sequence Number**  
A 32-bit field that represents the byte number of the first byte that is sent in a particular TCP segment. It is used to reassemble the message at the receiving end, particularly if the segments are received out of order.

**Acknowledgement Number**  
A 32-bit field that represents the byte number that the receiver expects to receive next. It is an acknowledgement for the previous bytes that were received successfully.

**Header Length (HLEN)**  
This is a 4-bit field that indicates the length of the TCP header by the number of 4-byte words in the header. For example, if the header is 20 bytes (the minimum length of TCP header), then this field is set to 5 (5 x 4 = 20).

**Control flags**  
These are 6 1-bit control bits that control connection establishment, connection termination, connection abortion, connection flow control, and mode of transfer:

URG: Urgent pointer is valid

ACK: Acknowledgement number is valid (used in case of cumulative acknowledgement)

PSH: Request for push

RST: Reset the connection

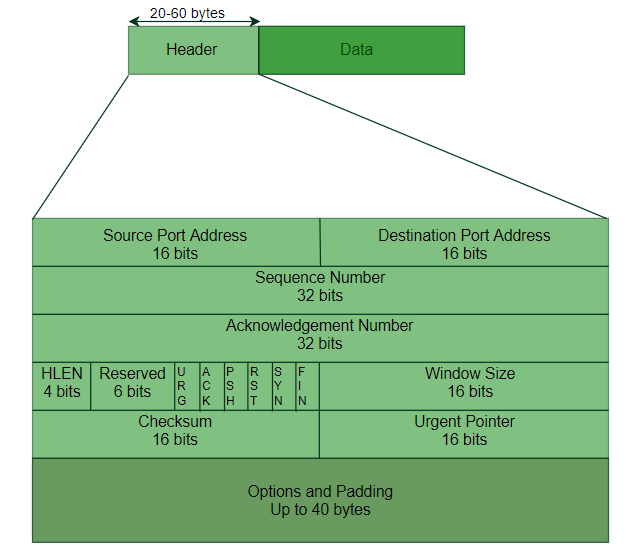
SYN: Synchronize sequence numbers

FIN: Terminate the connection

**Window size**   
This field tells the window size of the sender in bytes. The window size is used as a signal to control the connection’s data flow rate.

**Checksum**  
This field holds the checksum for error control. It is mandatory in TCP as opposed to UDP.

**Urgent pointer**  
This field (valid only if the URG control flag is set) is used to point to data that is urgently required and that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte. 



1. The payload section of a TCP segment contains the Layer-5 application packet. The Layer-5 application packet is processed by the application program.
2. The Transmission Control Protocol (TCP) is the most common Layer-4 Transport layer protocol.
3. TCP works together with IP and provides a reliable transport service between processes using the network layer service provided by the IP protocol.
4. The Transmission Control Protocol (TCP) is designed to be a bidirectional, ordered, and reliable data transmission protocol between two end points (applications):
   1. TCP is a bidirectional communication protocol. After a connection is established between a client and a server, there isn’t a significant difference between the client and the server. Both are simply hosts communicating in a bidirectional manner.
   2. The term reliable means that if the sender doesn’t receive an acknowledgement (ACK) packet from the receiver the sender will retransmit packets. TCP guarantees reliability by sending back Acknowledgment (ACK) packets for a single or a range of packets received from the sender.
5. A TCP connection is established by a 3-packet connection establishment sequence, known as the 3-way handshake.
6. The various services provided by the TCP to the application layer are as follows:

**Process-to-Process Communication**  
TCP provides a process-to-process communication. The transfer of data takes place between individual host processes. A TCP port number represents the process. Port numbers are 16 bits long that help identify which process is sending or receiving data on a host.

**Stream Oriented**  
Data is sent and received as a stream of bytes. However, the network layer that provides the network service for the TCP segments, sends fixed sized packets of data, not streams of bytes. Hence, TCP groups a number of bytes together into a *segment*, adds a header to each of these segments and then delivers these segments to the network layer. At the network layer, each of these segments is encapsulated in an IP packet for transmission.

**Full-duplex Service**  
Communication can take place between hosts in both directions and at the same time.

**Connection-oriented Service**  
Unlike UDP, TCP provides a connection-oriented service. It defines 3 different phases:

* Connection establishment
* Data transfer
* Connection termination

**Reliability**  
TCP uses checksums for error detection of corrupted packets. Using an acknowledgement policy and timers, TCP attempts to recover lost packets by retransmitting the packets. The acknowledgement policy uses byte sequence numbering and byte acknowledgement numbering to ensure reliability.

**Congestion Control**

TCP uses the Window Size field as a data transfer control mechanism. The TCP Window Size field contains the receiving connections current buffer size. The sending host uses the receiver’s TCP Window Size value to determine the rate of data transfer.

**Multiplexing**  
TCP includes multiplexing and de-multiplexing at the sender and receiver ends respectively as a number of logical connections can be established between port numbers over a physical connection.

**Byte number, Sequence number and Acknowledgement number**  
All transmitted bytes are numbered. The start number is arbitrary. The TCP segment’s sequence number field contains the byte number of the first byte that is being sent. The receiver reassembles the TCP segments based on the segment’s sequence number. If TCP segments arrive out of order, the receiver reorders them before passing the data to the application layer. The receiver sends back a TCP acknowledgement (ACK) packet for an individual packet or for a sequence of packets. The TCP acknowledgement number in an ACK packet is the next byte number that the receiver expects to receive from the sender.

1. The following is a simple diagram illustrating TCP/IP communication between two network hosts:

Graphical user interface, text, application

Description automatically generated

1. The following is a diagram illustrating application ports for two network hosts:

**Network Application Ports**

Router

216.155.1.42



164.67.235.92



216.155.1.43



216.155.1.41



164.67.235.91



164.67.235.93



Networks

4000

2025

6072

Port Numbers

13

80

23

Port Numbers

Client Computer

Server Computer

Daytime Server Port

HTTP Server Port

Telnet Server Port

Client Port for Daytime Server Data

Client Port for HTTP Server Data

Client Port for Telnet Server Data

Switch

Switch

216.155.1.1

164.67.235.1

1. The client or the server can initiate the connection termination.
   1. The active-closer is the socket that sends a TCP packet with the FIN flag set. The other peer is considered the passive-closer.
   2. The active-closer sends a FIN packet. After the active closer sends a FIN packet, the active-closer’s TCP state goes to FIN-WAIT-1.
   3. After the passive-closer receives the active-closer’s FIN packet, the passive-closer sends back an ACK packet, and the passive-closer’s state goes to CLOSE-WAIT.
   4. After the active-closer receives an ACK from the passive-closer, the active-closer’s state goes to FIN-WAIT-2.
   5. The passive-closer sends it’s FIN packet. After the passive-closer sends it’s FIN packet, the passive-closer’s TCP state goes to LAST-ACK.
   6. After the active-closer receives the passive-closer’s FIN packet, the active-closer sends back an ACK packet, and the active-closer’s state goes to TIME-WAIT.
   7. After the passive-closer receives the active-closer’s ACK packet, the passive-closer’s state goes to CLOSED.
   8. After the active-closer’s TIME-WAIT period expires, the active-closer’s state goes to CLOSED.
2. TCP connection termination request and response protocol is defined in RFC 793. The protocol defines the TIME-WAIT state to be as follows:
   1. TIME-WAIT - represents waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
   2. RFC 793 sets the TIME-WAIT TIME-OUT to be twice the Maximum Segment Lifetime (2MSL).
      1. The MSL is the maximum timeout period.
      2. By default, TIME-OUT is set to 2 minutes.
      3. Therefore, 2MSL is 4 minutes.
      4. If there is no ACK, the active closer must wait 4 minutes.
   3. <http://www.tcpipguide.com/free/t_TCPConnectionTermination-2.htm>
   4. https://www.vskills.in/certification/tutorial/tcp-connection-establish-and-terminate/

TCP Connection Termination State Diagram

TCP Connection Termination - GeeksforGeeks

TCP Connection Termination - GeeksforGeeks

Diagram, schematic

Description automatically generated