CS3200: Computer Networks

Lecture 27

IIT Palakkad

16 Oct, 2019

Transmission Control Protocol (TCP)

- TCP (Transmission Control Protocol) was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork.
- TCP was formally defined in RFC 793 in September 1981.
- TCP service is obtained by both the sender and the receiver creating end points, called **sockets**.
- Port numbers below 1024 are reserved for standard services that can usually only be started by privileged users (e.g., root in UNIX systems).

Transmission Control Protocol (TCP)

- **inetd (Internet daemon)** in UNIX, attach itself to multiple ports and wait for the first incoming connection.

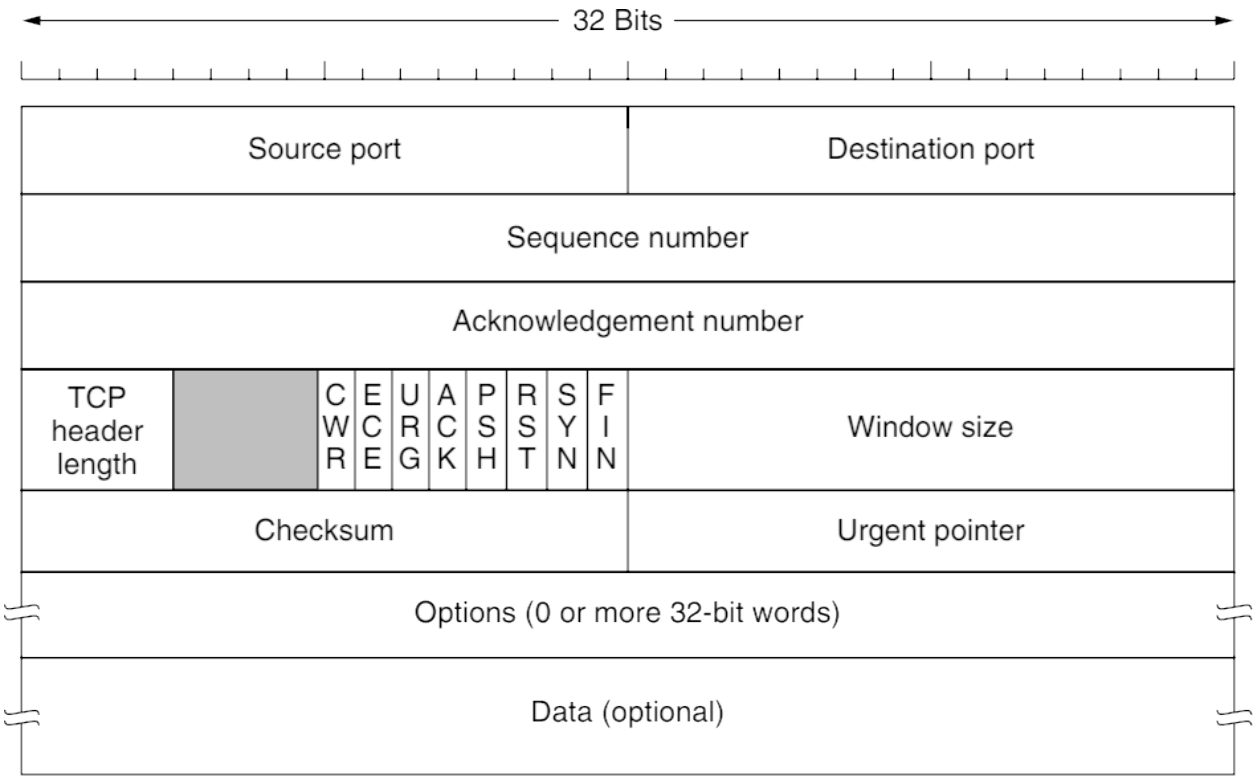
- All TCP connections are full duplex and point-to-point. Full duplex means that traffic can go in both directions at the same time. Point-to-point means that each connection has exactly two end points.

- A TCP connection is a byte stream, not a message stream. For example, if the sending process does four 512-byte writes to a TCP stream, these data may be delivered to the receiving process as four 512-byte chunks, two 1024-byte chunks, one 2048-byte chunk, or some other way.

The TCP protocol

- The sending and receiving TCP entities exchange data in the form of segments.
- A **TCP segment** consists of a fixed 20-byte header (plus an optional part) followed by zero or more data bytes.
- The basic protocol used by TCP entities is the sliding window protocol with a dynamic window size. When a sender transmits a segment, it also starts a timer.

TCP segment header



TCP segment header

- The *Source port* and *Destination port* fields identify the local end points of the connection.
- The *Sequence number* and *Acknowledgement number* fields perform their usual functions.
- *TCP header length* tells how many 32-bit words are contained in the TCP header.
- CWR and ECE are used to signal congestion when ECN (Explicit Congestion Notification) is used.
- ECE is set to signal an ECN-Echo to a TCP sender to tell it to slow down when the TCP receiver gets a congestion indication from the network.
- CWR is set to signal Congestion Window Reduced from the TCP sender to the TCP receiver so that it knows the sender has slowed down and can stop sending the ECN-Echo.

TCP segment header

- URG is set to 1 if the Urgent pointer is in use. The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found.
- The ACK bit is set to 1 to indicate that the *Acknowledgement number* is valid.
- The PSH bit indicates PUSHed data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received (which it might otherwise do for efficiency).
- The RST bit is used to abruptly reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection.
- The SYN bit is used to establish connections.
- The FIN bit is used to release a connection.

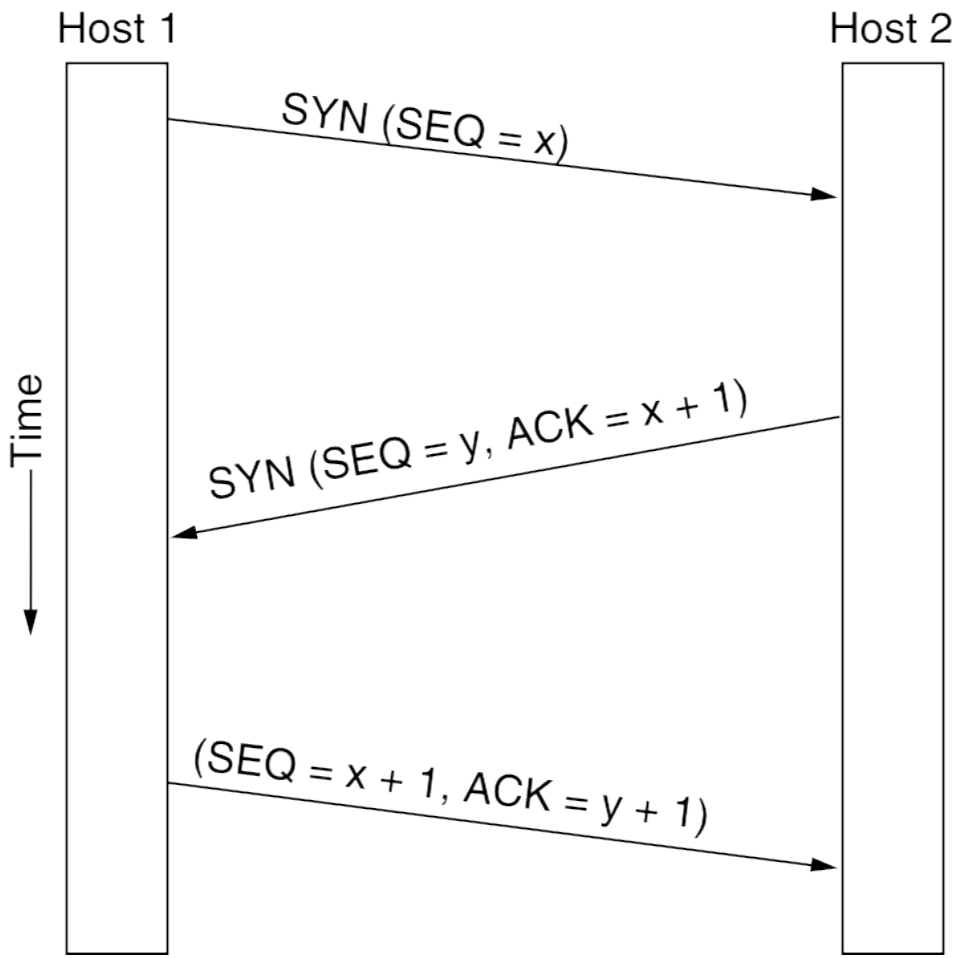
TCP segment header

- The *Window size* field tells how many bytes may be sent starting at the byte acknowledged.
- A **Checksum** is also provided for extra reliability. It checksums the header and data.
- The **window scale** option allows the sender and receiver to negotiate a window scale factor at the start of a connection. Both sides use the scale factor to shift the Window size field up to 14 bits to the left, thus allowing windows of up to 230 bytes.
- The **timestamp** option carries a timestamp sent by the sender and echoed by the receiver. Can be used for computing RTT, as well as discard data that arrive late.
- The **SACK (Selective ACKnowledgement)** option lets a receiver tell a sender the ranges of sequence numbers that it has received.

TCP connection establishment

- Connections are established in TCP by means of the three-way handshake.
- To establish a connection, the server, passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives.
- The client, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect.
- The server can can either accept or reject the connection. If it accepts, an acknowledgement segment is sent back.

TCP connection establishment



TCP connection establishment

- A vulnerability with implementing the three-way handshake is that the listening process must remember its sequence number as soon it responds with its own SYN segment.
- This means that a malicious sender can tie up resources on a host by sending a stream of SYN segments and never following through to complete the connection. This attack is called a **SYN flood**.
- One way to defend against this attack is to use **SYN cookies**. Instead of remembering the sequence number, a host chooses a cryptographically generated sequence number, puts it on the outgoing segment, and forgets it.

TCP connection release

- To release a connection, either party can send a TCP segment with the FIN bit set, which means that it has no more data to transmit.
- When the FIN is acknowledged, that direction is shut down for new data. However, data may continue to flow in the other direction.
- When both directions have been shut down, the connection is released.
- Normally, four TCP segments are needed to release a connection: one FIN and one ACK for each direction. However, it can some times be three.
- If a response to a FIN is not forthcoming within two maximum packet lifetimes, the sender of the FIN releases the connection. The other side will eventually notice that nobody seems to be listening to it anymore and will time out as well.