TCP CONNECTION ESTABLISHMENT AND TERMINATION

Before any host can send data, a connection must be established. TCP establishes the connection using a three-way handshake procedure shown in Figure 8.22. The handshakes are described in the following steps:

1. Host A sends a connection request to host B by setting the SYN bit. Host A also registers its initial sequence number to use (Seq_no fl x).
2. Host B acknowledges the request by setting the ACK bit and indicating the next data byte to receive (Ack_no fl x + 1). The ``plus one'' is needed because the SYN bit consumes one sequence number. At the same time, host B also sends a request by setting the SYN bit and registering its initial sequence number to use (Seq_no fl y).
3. Host A acknowledges the request from B by setting the ACK bit and con firming the next data byte to receive (Ack_no fl y + 1). Note that the sequence number is set to x + 1. On receipt at B the connection is established.

If during a connection establishment phase, one of the hosts decides to refuse a connection request, it will send a reset segment by setting the RST bit. Each SYN message can specify options such as maximum segment size, window scal- ing, and timestamps.

Because TCP segments can be delayed, lost, and duplicated, the initial sequence number should be different each time a host requests a connection.

TCP CONNECTION TERMINATION

TCP provides for a graceful close that involves the independent termination of each direction of the connection. A termination is initiated when an application tells TCP that it has no more data to send. The TCP entity completes transmis- sion of its data and, upon receiving acknowledgment from the receiver, issues a segment with the FIN bit set. Upon receiving a FIN segment, a TCP entity informs its application that the other entity has terminated its transmission of data. For example, in Figure 8.27 the TCP entity in host A terminates its trans- mission first by issuing a FIN segment. Host B sends an ACK segment to acknowledge receipt of the FIN segment from A. Note that the FIN segment uses one byte, so the ACK is 5087 in the example.

After B receives the FIN segment, the direction of the flow from B to A is still open. In Figure 8.27 host B sends 150 bytes in one segment, followed by a FIN segment. Host A then sends an acknowledgment. The TCP in host A then enters the TIME_WAIT state and starts the TIME_WAIT timer with an initial value set to twice the maximum segment lifetime (2MSL).

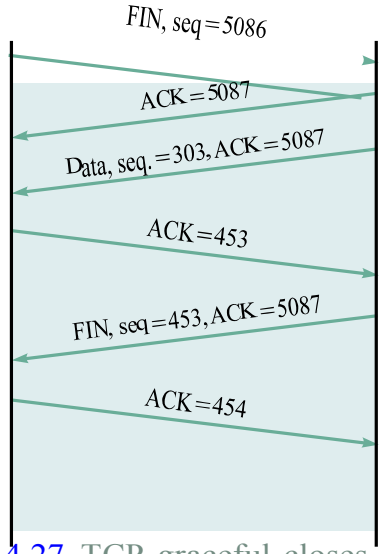

FIGURE 4.27 TCP graceful closes

TIME_WAIT timer is restarted at 2MSL. When the TIME_WAIT timer expires, host A closes the connection and then deletes the record of the connection.

The TIME_WAIT state serves a second purpose. The MSL is the maximum time that a segment can live inside the network before it is discarded. The TIME_WAIT state protects future incarnations of the connection from delayed segments. The TIME_WAIT forces TCP to wait at least two MSLs before setting up an incarnation of the old connection. The first MSL accounts for the maximum time a segment in one direction can remain in the network, and the second MSL allows for the maximum time a reply in the other direction can be in the network. Thus all segments from the old connection will be cleared from the network at the end of the TIME_WAIT state.

TCP provides for an abrupt connection termination through reset (RST) segments. An RST segment is a segment with the

RST bit set. If an application decides to terminate the connection abruptly, it issues an ABORT command, which causes TCP to discard any data that is queued for transmission and to send an RST segment. The TCP that receives the RST segment then notifies its application process that the connection has been terminated.

Source : http://elearningatria.files.wordpress.com/2013/10/cse-vi-computer-networks-ii-10cs64-notes.pdf