Chapter 5 End-to-End Protocols

TCP (Continued)

- Connection establishment
  - Before a client attempts to connect with a server, the server must first bind to and listen at a port; this is called a *passive open*.
  - Once the passive open is established, a client may initiate an *active open*.
  - The three-way handshake (see slides) occurs during connection establishment:
    - Each side selects an initial sequence number at random.
    - A timer is scheduled for SYN and SYN+ACK segments so that they can be retransmitted upon timeout.
    - The connection functions correctly even if the 3rd message is lost: the server will move to the ESTABLISHED state when the first data segment arrives.
  - Detection of old duplicate SYN segment (see slides)

- Connection termination
  - Each side independently closes its half of the connection by sending a FIN segment:
    - If one side closes the connection, it can no longer send data, but it still can receive data from the other side.
  - A timer is scheduled for FIN segment, FIN segment is retransmitted upon timeout.
  - Timed_Wait is used to prevent confusion due to delayed duplicate FIN packet from the other side being delivered during a subsequent connection.
TCP’s Sliding Window Algorithm

- Provides reliable delivery, in-order delivery, and flow control
- Reliable and ordered delivery
  - Send buffer stores data that has been sent but not yet ACKed, as well as data that has been written by the sending application but not transmitted
  - Receive buffer holds data that arrives out of order, as well as data that is in the correct order but that the application process has not had the chance to read
- Flow control
  - MaxRcvBuffer denotes the size of the receive buffer
  - MaxSendBuffer denotes the size of the send buffer
  - Receive side must keep LastByteRcvd – LastByteRead ≤ MaxRcvBuffer to avoid overflowing its buffer
  - Receiver advertises a window size of AdvertisedWindow = MaxRcvBuffer – ((NextByteExpected-1) – LastByteRead)
    - AdvertisedWindow indicates the amount of free space remaining in the receive buffer.
  - Sender must ensure the number of outstanding bytes is no larger than AdvertisedWindow, that is, LastByteSent – LastByteAcked ≤ AdvertisedWindow
    - Sender computes an effective window that limits how much data it can send: EffectiveWindow = AdvertisedWindow – (LastByteSent – LastByteAcked)
    - If EffectiveWindow > 0, sender can send more data.
  - Sender must ensure the application process does not overflow the send buffer, i.e., LastByteWritten – LastByteAcked ≤ MaxSendBuffer
    - TCP blocks the sending process if (LastByteWritten – LastByteAcked) + y > MaxSenderBuffer, where y is the number of bytes the sending process tries to write to TCP.
- **When AdvertisedWindow = 0**
  - The sending side periodically sends a probe segment with one byte of data.
  - Each probe segment triggers a response that contains the current advertised window; this allows the sending side to learn that the advertised window is no longer 0.

- **Protecting against wraparound**
  - Receiver cannot distinguish between two packets having different incarnations of the same sequence number → TCP needs to make sure the sequence number does not wrap around within the Maximum Segment Lifetime (MSL) where MSL = 120 seconds.
    - Time until wraparound = $2^{32} \times \frac{8}{\text{bandwidth}}$ → it depends on the network bandwidth (see Table 5.1)
  - TCP uses the 32-bit timestamp option to effectively extend the sequence number space.
    - TCP reads the system clock when it is about to send a segment, and puts this time in the segment’s header.
    - TCP accepts or rejects a segment based on a 64-bit identifier that has the SequenceNum field in the low-order 32 bits and the timestamp in the high-order 32 bits.
    - The timestamp serves to distinguish between two different incarnations of the same sequence number.

- **Keeping the pipe full**
  - The advertised window need allow a full RTT x bandwidth product’s worth of data to be transmitted (i.e., keep the pipe full).
    - 16-bit AdvertisedWindow field allows receiver to advertise a window of only 64KB, which is not big enough for high-speed networks (see Table 5.2).
  - TCP uses the window scale option to effectively increase the size of the advertised window.
    - The option defines a scaling factor for the advertised window that allows the two sides to agree that the
AdvertisedWindow field counts larger chunks (e.g., 16-byte units) of data the sender can have unACKed.

- In other words, the option specifies how many bits each side should left-shift the AdvertisedWindow field before using its contents to compute an effective window.
- The scaling factor has a maximum value of 14, so the maximum window size is $2^{30}$ byte = 1 gigabyte.