Computer Network

Lecture-42

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Congestion Control and Quality of Service

Congestion Control in TCP

Congestion Window

The sender's window size is determined not only by the receiver but also by congestion in the network.

The sender has two pieces of information: the receiver advertised window size(rwnd) and the congestion window size(cwnd). The actual size of the window is the minimum of these two i.e.

Actual window size = minimum (rwnd, cwnd)

Congestion Policy

TCP's general policy for handling congestion is based on three phases: slow start, congestion avoidance, and congestion detection.

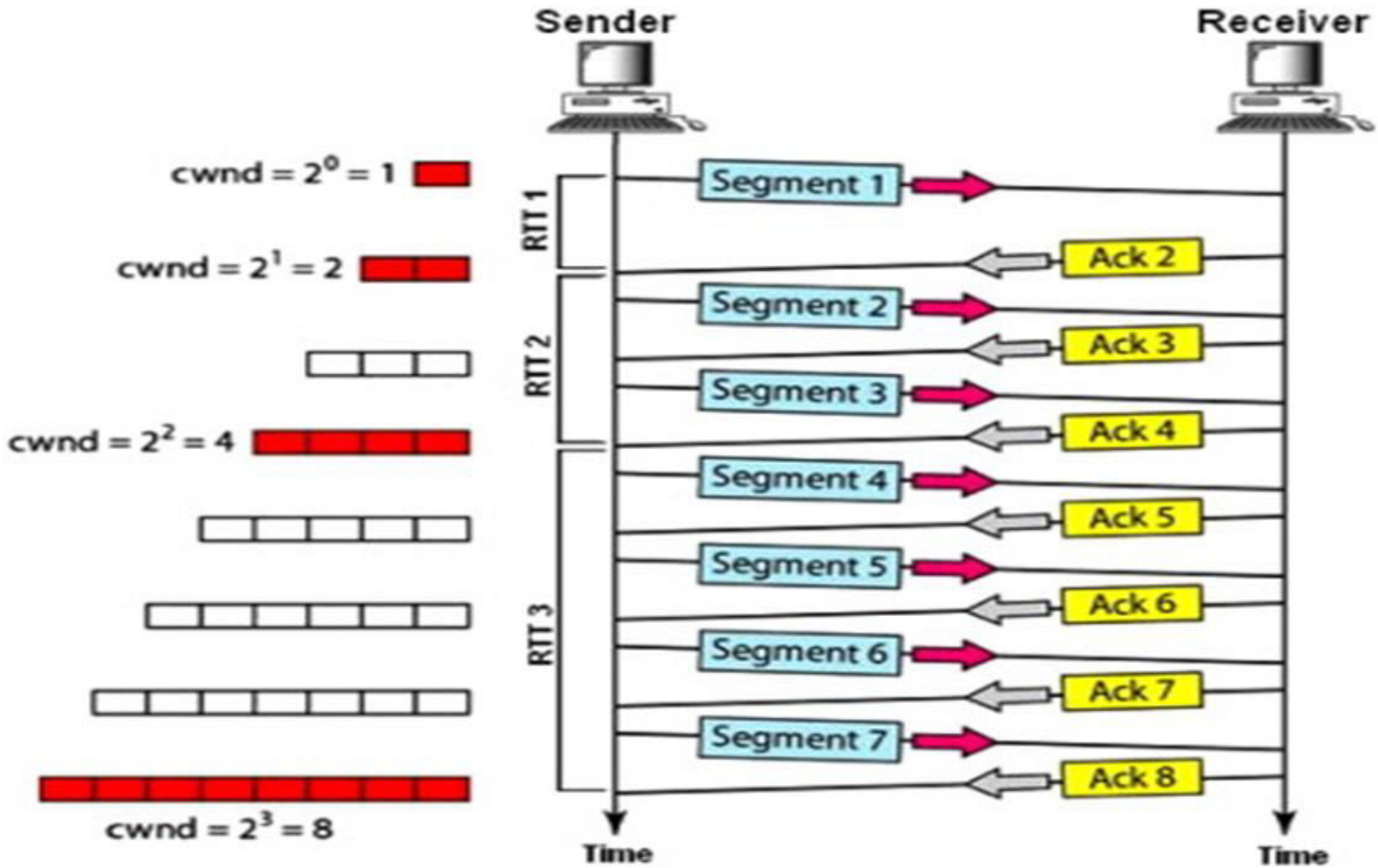
Congestion Control and Quality of Service

Slow Start: Exponential Increase:

This algorithm is based on the idea that the size of the congestion window (cwnd) starts with one maximum segment size (MSS). The size of the window increases one MSS each time an acknowledgment is received. As the name implies, the window starts slowly, but grows exponentially. It is shown in the following figure:-

In this figure, we have assumed that rwnd is much higher than cwnd, so that the sender window size always equals cwnd. We have assumed that each segment is acknowledged individually.

Congestion Control and Quality of Service



Congestion Control and Quality of Service

The sender starts with cwnd =1 MSS. This means that the sender can send only one segment. After receipt of the acknowledgment for segment 1, the size of the congestion window is increased by 1, which means that cwnd is now 2. Now two more segments can be sent. When each acknowledgment is received, the size of the window is increased by 1 MSS. When all seven segments are acknowledged, cwnd = 8.

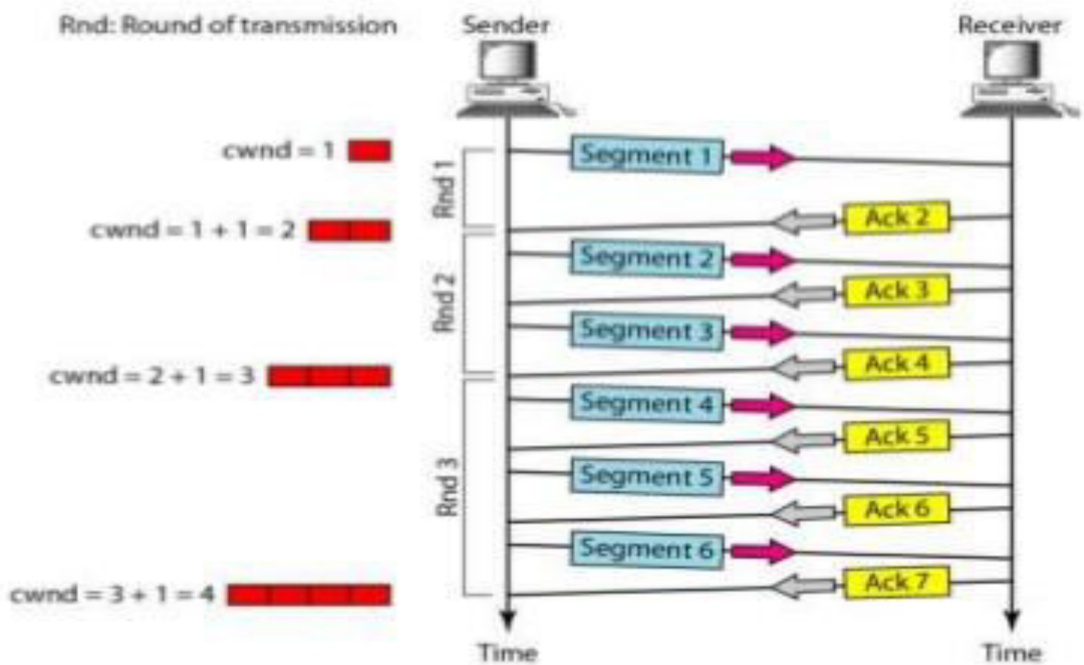
Congestion Avoidance: Additive Increase

TCP defines another algorithm called congestion avoidance, which undergoes an additive increase instead of an exponential one. When the size of the congestion window reaches the slow-start threshold, the slow-start phase stops and the additive phase begins. In this algorithm, each time the whole window of segments is acknowledged (one round), the size of the congestion window is increased by 1. It is shown in the following figure:-

Congestion Control and Quality of Service

Congestion Control in TCP

Congestion avoidance, additive increase:



Congestion Control and Quality of Service

Congestion Detection: Multiplicative Decrease

If congestion occurs, the congestion window size must be decreased. The only way the sender can guess that congestion has occurred is by the need to retransmit a segment. However, retransmission can occur in one of two cases: when a timer times out or when three ACKs are received. In both cases, the size of the threshold is dropped to one-half, a multiplicative decrease. Most TCP implementations have two reactions:

1. If a time-out occurs, there is a stronger possibility of congestion; a segment has probably been dropped in the network, and there is no news about the sent segments.

In this case TCP reacts strongly:

a. It sets the value of the threshold to one-half of the current window size.

b. It sets cwnd to the size of one segment.

c. It starts the slow-start phase again.

Congestion Control and Quality of Service

2. If three ACKs are received, there is a weaker possibility of congestion; a segment may have been dropped, but some segments after that may have arrived safely since three ACKs are received. This is called fast transmission and fast recovery. In this case, TCP has a weaker reaction:

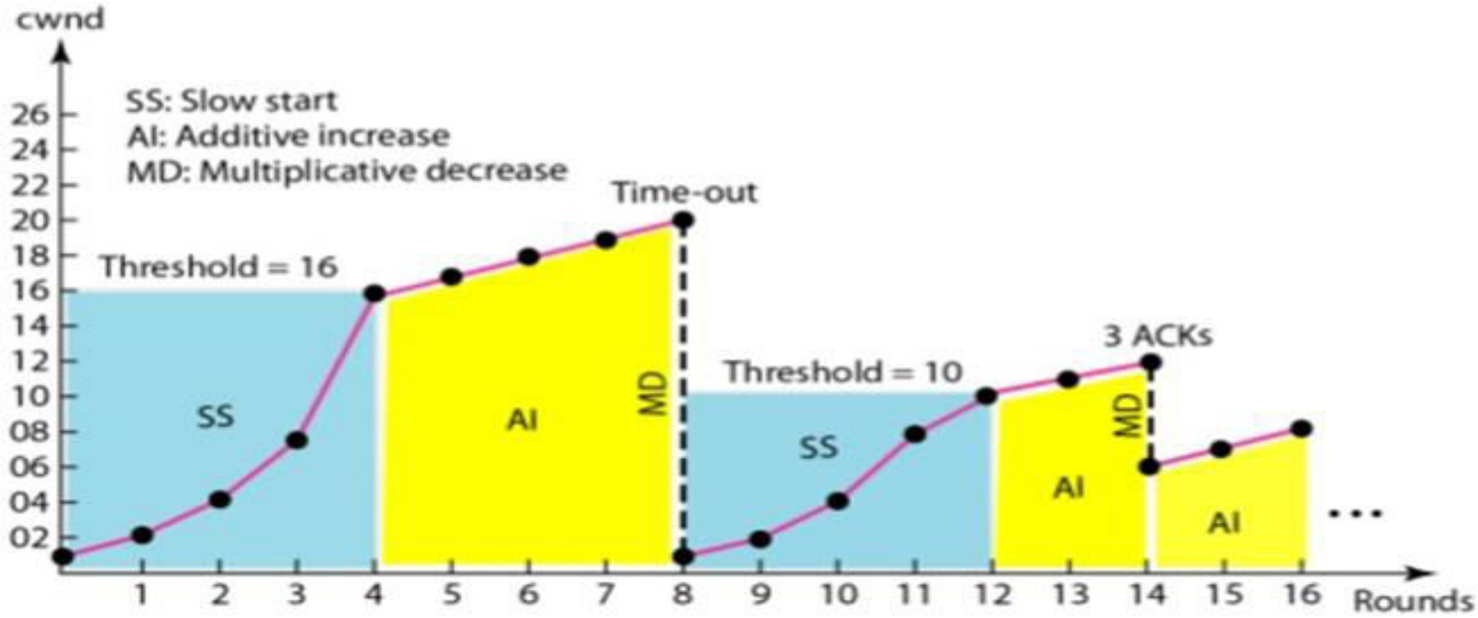
   a. It sets the value of the threshold to one-half of the current window size.

   b. It sets cwnd to the value of the threshold (some implementations add three segment sizes to the threshold).

   c. It starts the congestion avoidance phase.

Congestion Control and Quality of Service

Example:



In this figure, the time-out occurs when the window size is 20. At this moment, the multiplicative decrease procedure takes over and reduces the threshold to one-half of the previous window size. The previous window size was 20 when the time-out happened so the new threshold is now 10.

Congestion Control and Quality of Service

TCP moves to slow start again and starts with a window size of 1, and TCP moves to additive increase when the new threshold is reached. When the window size is 12, a three-ACKs event happens. The multiplicative decrease procedure takes over again. The threshold is set to 6 and TCP goes to the additive increase phase this time. It remains in this phase until another time-out or another three ACKs happen.