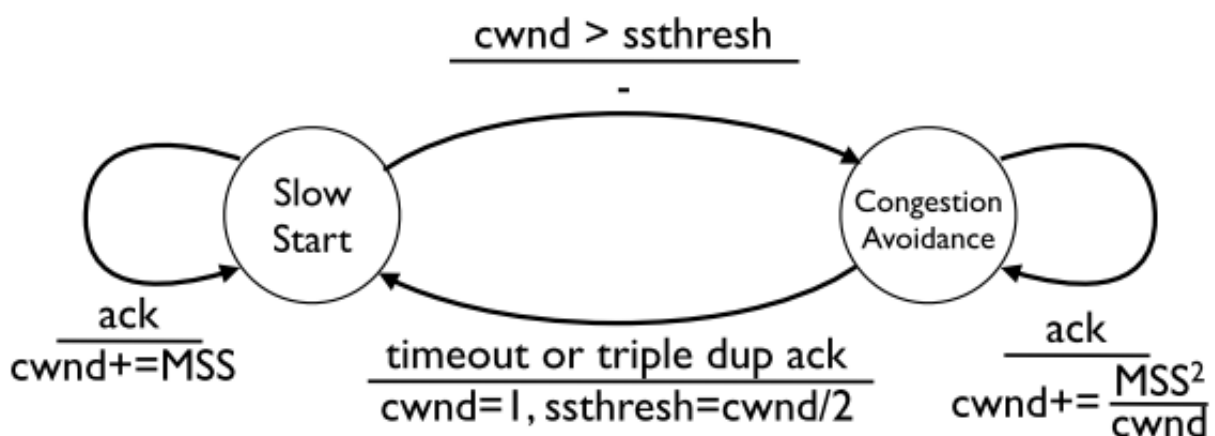


# TCP Tahoe

- In the old TCP implementation (the pre-Tahoe):
  - the transmitter initially sends a number of segments that equals the size of the window. This will incur packet loss and the network is used at less than its capacity.
  - for each packet sent, a timer is executed
- the TCP Tahoe [FSM](#) has two states: **Slow Start** and **Congestion Avoidance**.

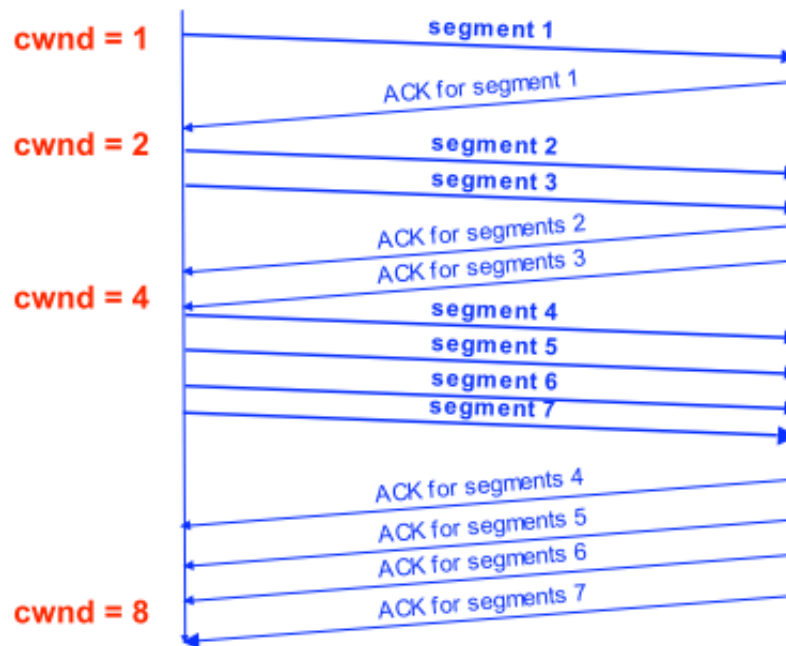


TCP Tahoe FSM – © Stanford University

- In the TCP Tahoe implementation, after a Retransmission TimeOut (RTO) or upon receiving duplicate ACKs, the transmitter gets in the “slow start” state.

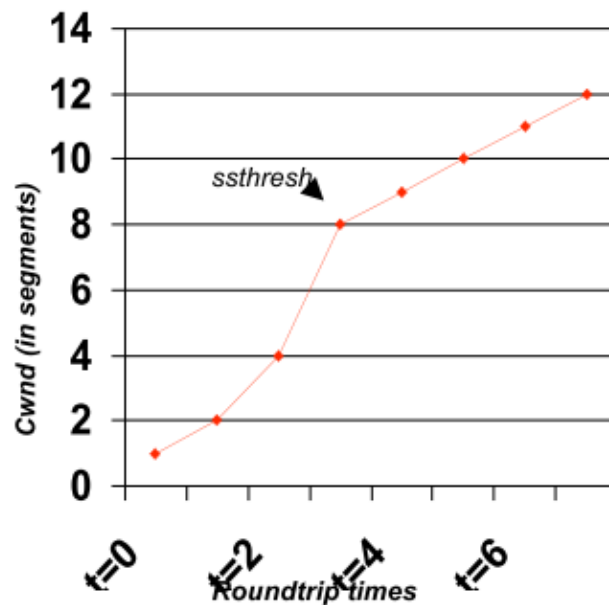


contains one MSS. When it receives a “good ACK” for the segment, it increases the window size to 2 and sends 2 segments (a “good ACK” is a ACK that requests the next higher byte of data that has not already been acknowledged). When TCP receives an ACK for each segment sent, it increases the window size by 2 MSSs. So more generally, at each RTT, TCP increases the congestion window size by 1 MSS for each ACK received per segment sent (window size: 1, 2, 4, 8...)



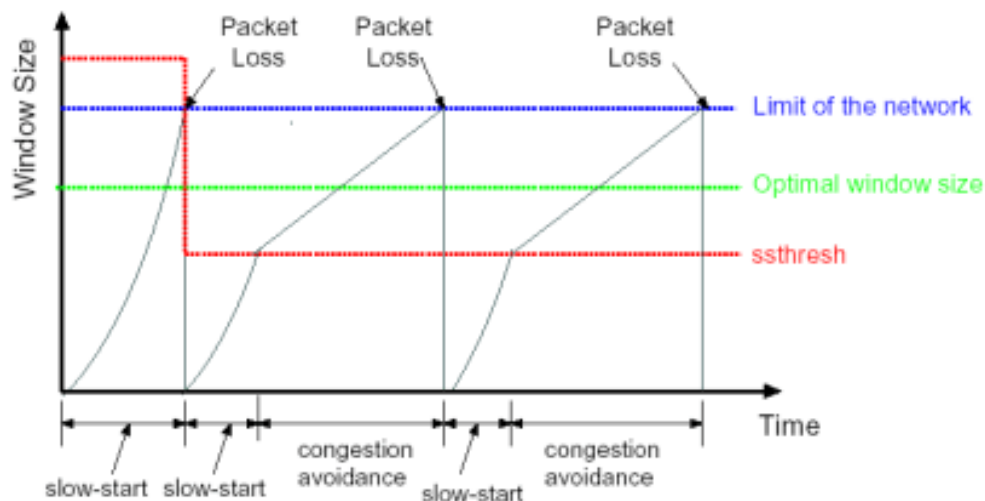
Simple exchange of segments that demonstrates Slow Start state © Kamil Sarac

- This process continues until **cwnd** > **ssthresh** (ssthresh is the *Slow Start Threshold*). At this point, TCP behaves additively (in AIMD manner, increasing the window by one MSS each RTT). This is the **Congestion Avoidance** state



TCP switches to AIMD at *sssthresh* © Kamil Sarac

- If one of the following events happen...
  - a RTO, or
  - three (or more) duplicate ACKs received
- ... then a packet loss is assumed and:
  - **$sssthresh \leftarrow cwnd/2$**
  - **$cwnd \leftarrow 1$** , and
  - TCP sender retransmits the missing segment (this is TCP **Fast Retransmit**).
  - TCP gets back to Slow Start state.
- The cycle repeats:



## TCP Tahoe cyle © www.soi.wide.ad.jp

- In the congestion avoidance state:
  - at each RTT, TCP increases the congestion window size by  **$2 * MSS / \text{Window size}$**  for every ACK received.
  - the growth of the window size is less rapid than in Slow Start
- What happens when **cwnd = ssthresh**? TCP can behave either in Slow Start state or in Congestion Avoidance state.
- To explain the initial exponential burst in the Window size graph, of a TCP Tahoe implementation, let's consider a new TCP connection: the ssthresh is defined at a very high value. So cwnd increases exponentially until there are dup ACKs or a RTO. At that point, ssthresh is defined as cwnd/2, and TCP Tahoe enters Slow Start again. Once cwnd > ssthresh, TCP Tahoe enters AIMD...So that burst appears only in the initial probe.
- When there are dup ACKs, the window size does not grow.
- The flow control window is a parameter that sets an upper bound on the amount of data the transmitter can send. It does not take into consideration the state of the network. In fact, the sender might be able to send at rates higher than the network rate. The sender's window can be much smaller than the flow control window. Recall this formula from the article "[Network-based And Host-based Congestion Control](#)":

**Sender's window size = min (advertized window, congestion window)**

- AIMD works perfectly for stable networks.
- The main difference between Slow Start state and Congestion Avoidance state is how TCP increases cwnd.