

Lab 02- Simulating TCP Tahoe: Understanding Congestion Control Mechanisms

Matthew Belanger

February 1, 2025

1 Lab Questions

1.1 Task 1: Running the TCP Tahoe Simulation

1.1.1 What happens to cwnd during slow start?

cwnd (the congestion window) increases for each time ACK is successfully received until a packet is lost (ACK was not received) then cwnd goes back to its original size (2) and tries to transmit packets again. When the algorithm detects that the window size has exceeded the threshold then we move into congestion avoidance.

1.1.2 How does TCP Tahoe handle packet loss?

Tahoe reduces the slow start threshold to half the window size (or 2 in the edge case), then sets the window size to 1, then returns to slow start state. The cwnd is how much data we can send without receiving an ACK, so in theory, reducing the window increases the rate at which we check for ACK messages slowing down our throughput giving the communication a chance to "catch back up".

1.1.3 What triggers the transition from slow start to congestion avoidance?

When the algorithm detects that the window size has exceeded the threshold then we move into congestion avoidance.

1.2 Task 2: Analyzing the Simulation Code

1.2.1 How is cwnd growth modeled during slow start and congestion avoidance?

The congestion window increases by one for each successfully sent packet in slow start. In congestion avoidance the window increases by the inverse of the current window size.

so the idea is that the window increase slower after it becomes larger than the threshold value.

1.2.2 What role does ssthresh play in TCP Tahoe?

Slow start threshold is the maximum size of the congestion window before we move from slow start to congestion avoidance.

1.3 Task 3: Modifying the Simulation

1.3.1 How does the packet loss rate affect TCP Tahoe's performance?

packet loss decreases performance because Tahoe will start to send less data per message which increases overhead.

1.3.2 How does increasing the maximum congestion window size impact the simulation?

When packet loss is low, the maximum congestion window size caps the amount of data we can send in a single message. When packet loss is high, the maximum congestion window size has little affect because Tahoe will reduce the window size algorithmically when a packet is lost.

1.4 Task 4: Exploring Scenarios

1.4.1 How does a higher packet loss rate affect cwnd dynamics?

when packet loss is low 'cwnd' peaks at a larger value because Tahoe does not need to reduce the window size as frequently so 'cwnd' can increase without being reset.

1.4.2 What differences do you observe when modifying ssthresh values?

in the long run it has no affect because after a packet is lost Tahoe overwrites it based on 'cwnd'

1.5 Task 5: Extending the Simulation

1.5.1 How does your chosen algorithm differ from TCP Tahoe?

TCP Reno is similar to Tahoe but it introduces new state 'fast recovery' when a duplicate sequence number is recieved 3 times. 'fast recovery' provides a special case that allows Reno to attempt resolve the congestion without immediately tanking performance like Tahoe

1.5.2 Which algorithm performs better under high packet loss?

Reno performs better because the 'cwnd' does not always return to two when packet loss occurs so we maintain better performance.