TCP Congestion Control

Holden Wells, Hayden Pott, Cameron McCarthy

Introduction

TCP congestion is a frequent problem in networks. Two algorithms, TCP Taho and TCP Reno, have different methods of dealing with this congestion. To analyze their differences, we were given a Wireshark pcap file and were tasked with simulating their response. Both methods both increase their sending rate exponentially until reaching a threshold. It then increases linearly until a packet loss event occurs, then they both decrease their sending rate and decrease the threshold. TCP Reno cuts its sending rate in half. TCP Tahoe resets its sending rate to the max segment size.

Methodology

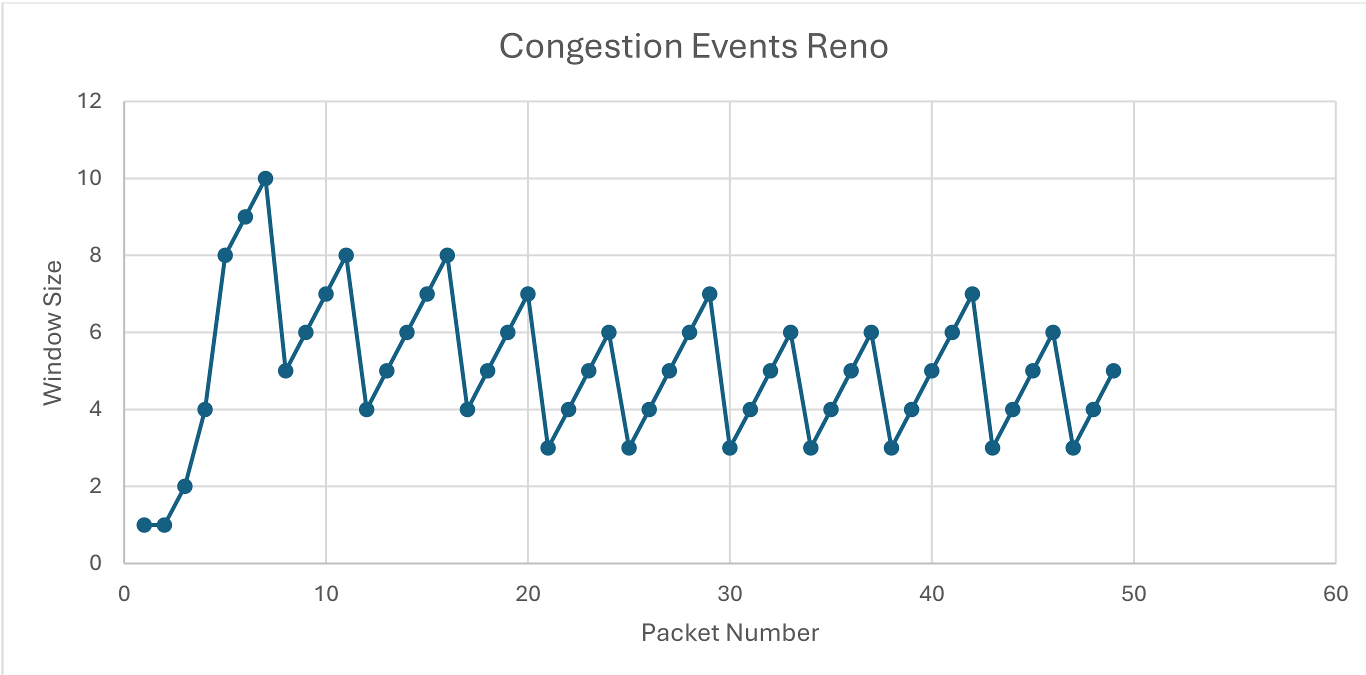
To begin, the pcap file was converted into a csv file using the tshark command:

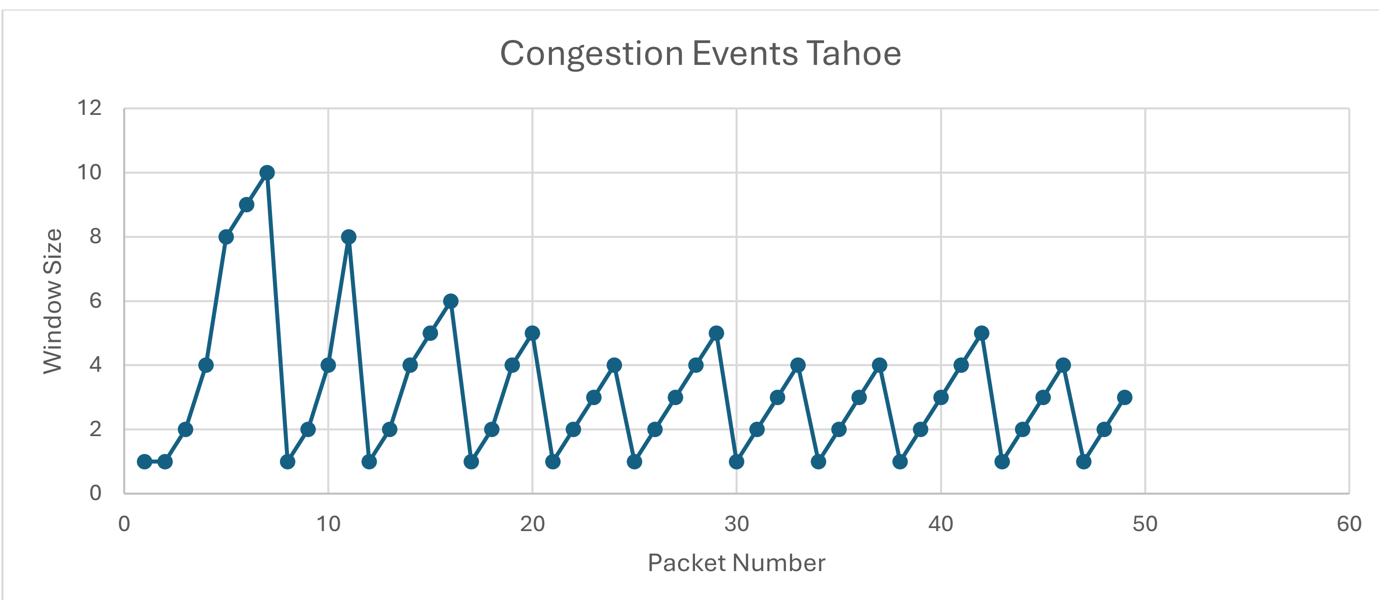
```
    tshark -r <file_name> -T fields -Eseparato r=',' -e
frame.number -e frame.time_relative -e ip.src -e ip.dst -e
ip.proto -e frame.len -e tcp.len -e tcp.time_delta -e tcp.flag s -
e tcp.ack -e tcp.seq -e tcp.window_size_value -e _ws.col.info
```

Our program, written in C, first parses the CSV for the data. Then detects triple duplicate ACKS, retransmission, and tcp window size to determine congestion events. Then it analyzes it based on the appropriate algorithm, Tahoe or Reno. Finally, it outputs the results into two separate CSV files, one for Tahoe and one for Reno.

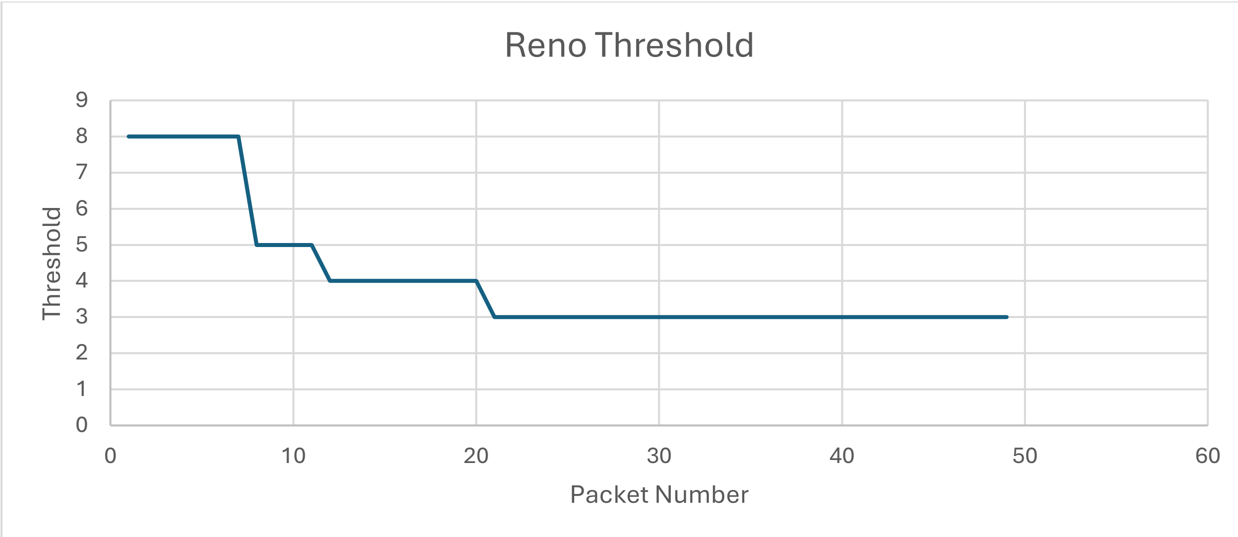
Results and Analysis

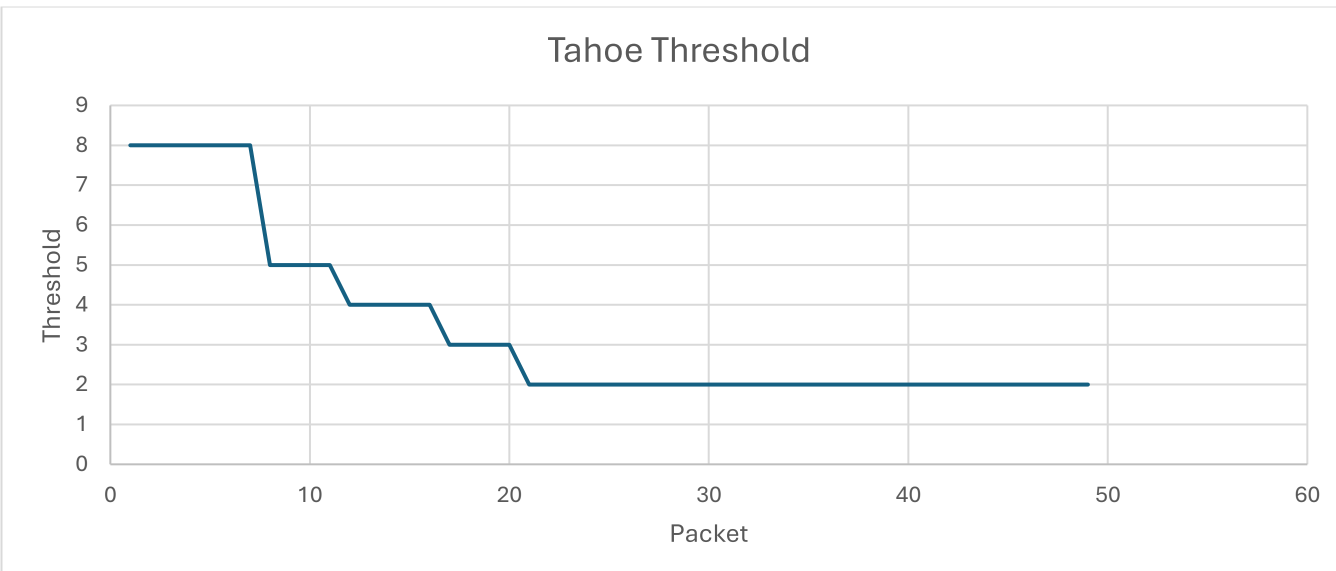
The results of Reno and Tahoe are shown below. To make it simpler, the first 50 entries are used for the graph, as there are thousands of packets. As you can see below, the window size increases until there is congestion, then drops back down in both cases.



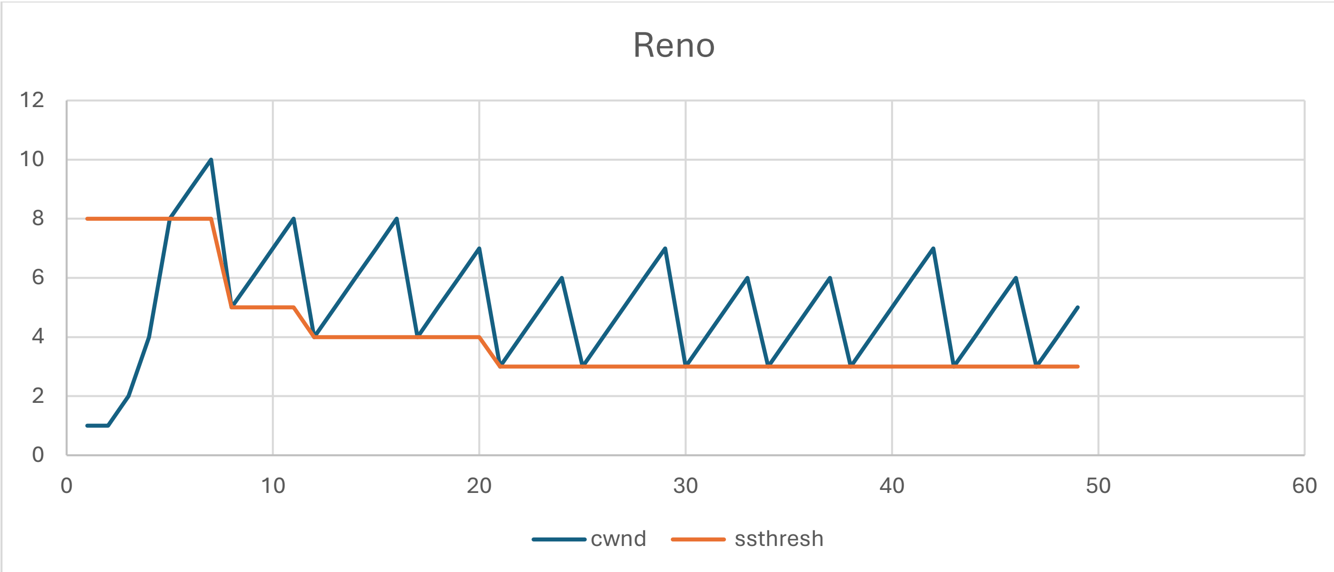


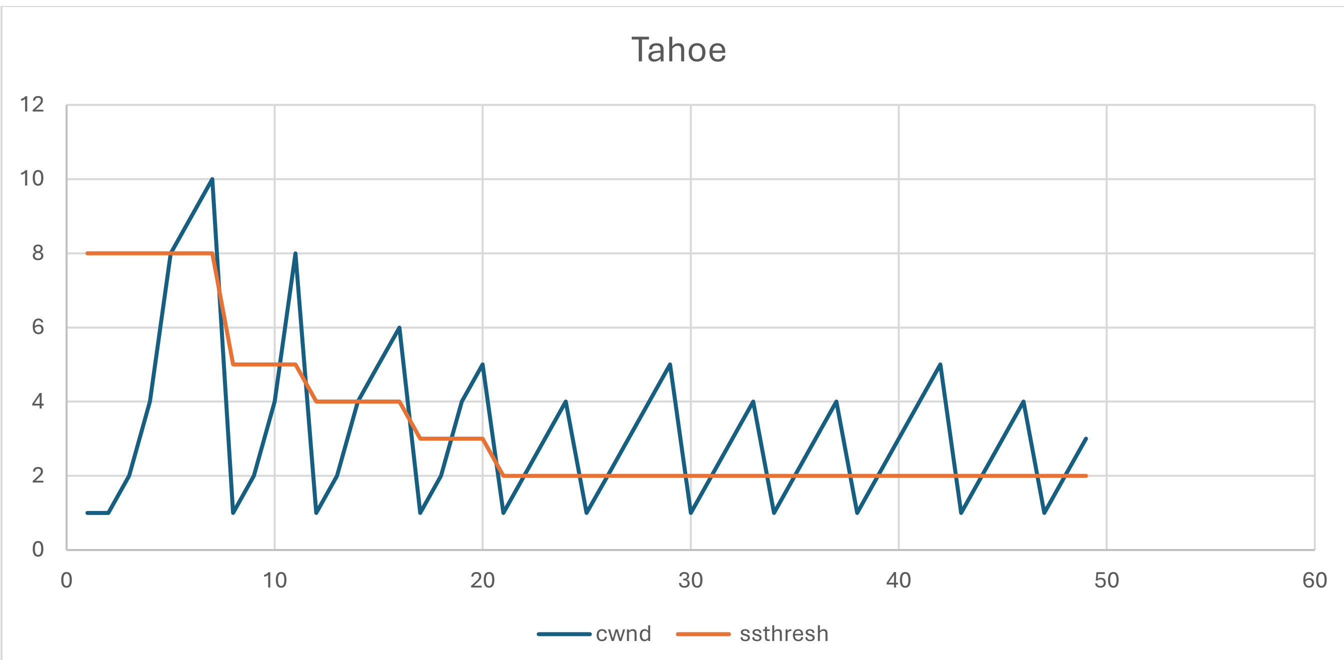
Below, threshold size is measured for both algorithms.





Here it is overlayed. Here you can see explicitly that when the threshold it reached, it stops increasing exponentially and starts increasing linearly





Discussion

In this situation, TCP Reno appears to have done better. TCP Reno allows for fast retransmission on packet loss. TCP Reno window size does not drop nearly as low as Tahoe. Therefore, TCP Reno's throughput remains higher throughout the packet loss. TCP has a much slower recovery.