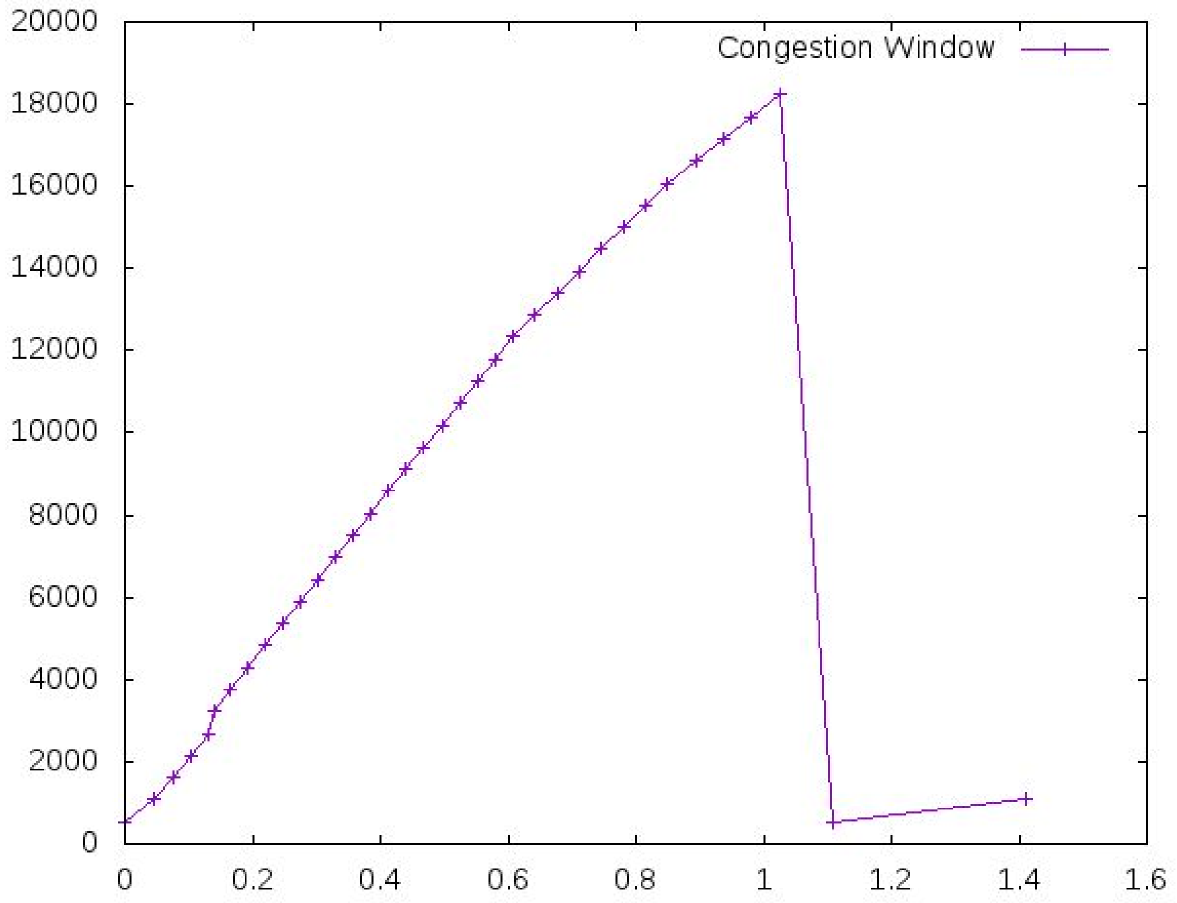
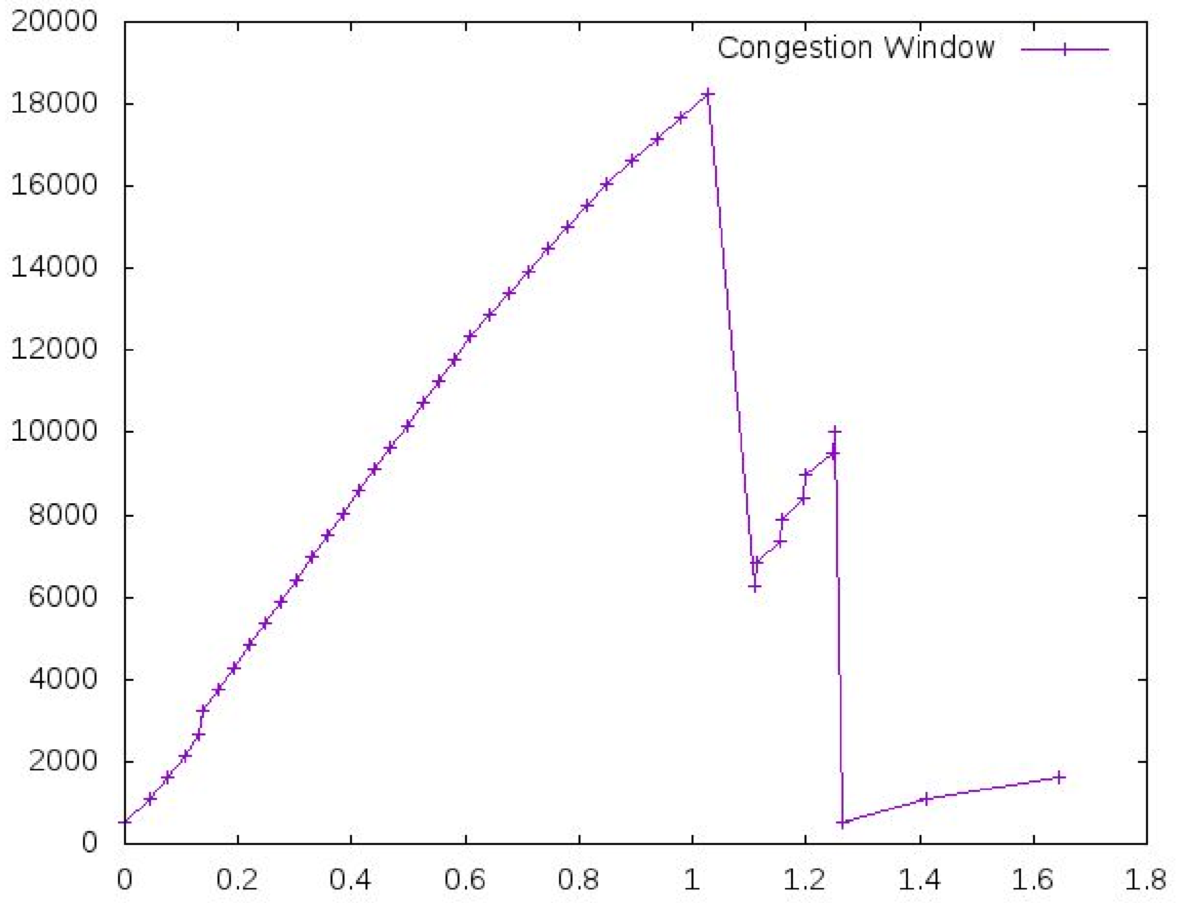


Tahoe-Congestion Window vs time

Explanation:-

TCP Tahoe is the earliest version of TCP that introduces the well-known congestion control mechanism.The idea is for a TCP source (sender) that has no knowledge about the network status to probe the path by gradually increasing its sending rate dictated by the congestion window. At the beginning of a connection, the source employs the slow-start phase in which the congestion window is increased exponentially. Once the window reaches a certain threshold known as the slow start threshold, to prevent congestion, the TCP source transfers from the slow-start phase to the congestion avoidance phase during which the window is increased linearly. The source continues to stay in its congestion avoidance until a timeout occurs or a certain number of duplicate acknowledgments (DUPACK) are received that is interpreted as a signal for congestion over the path. Upon such an event, the TCP source enters the fast retransmit, halves its slow-start threshold, resets the congestion window to one full-sized segment, and immediately retransmits the missing segment. The source ends its congestion control cycle by going back to the slow-start phas

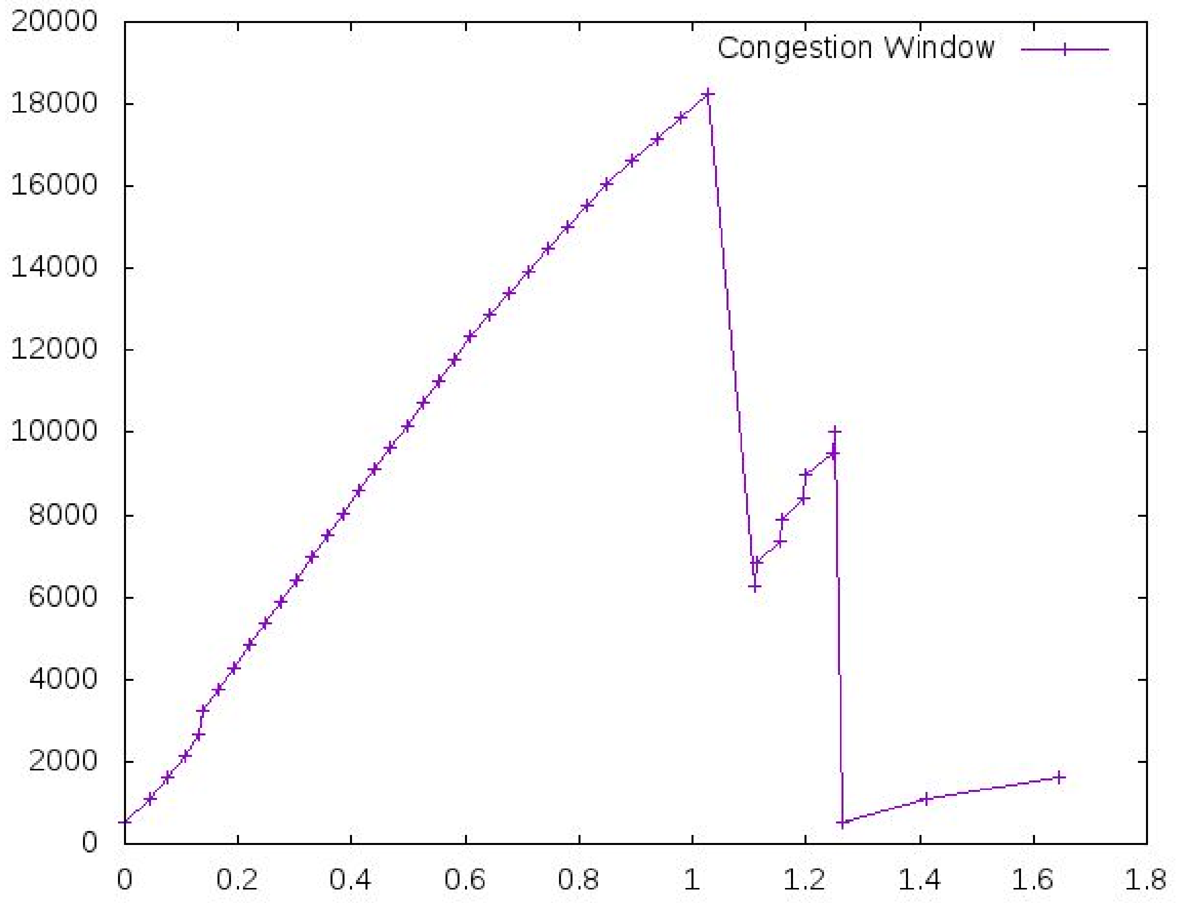


Reno-congestion window vs time

Explanation:

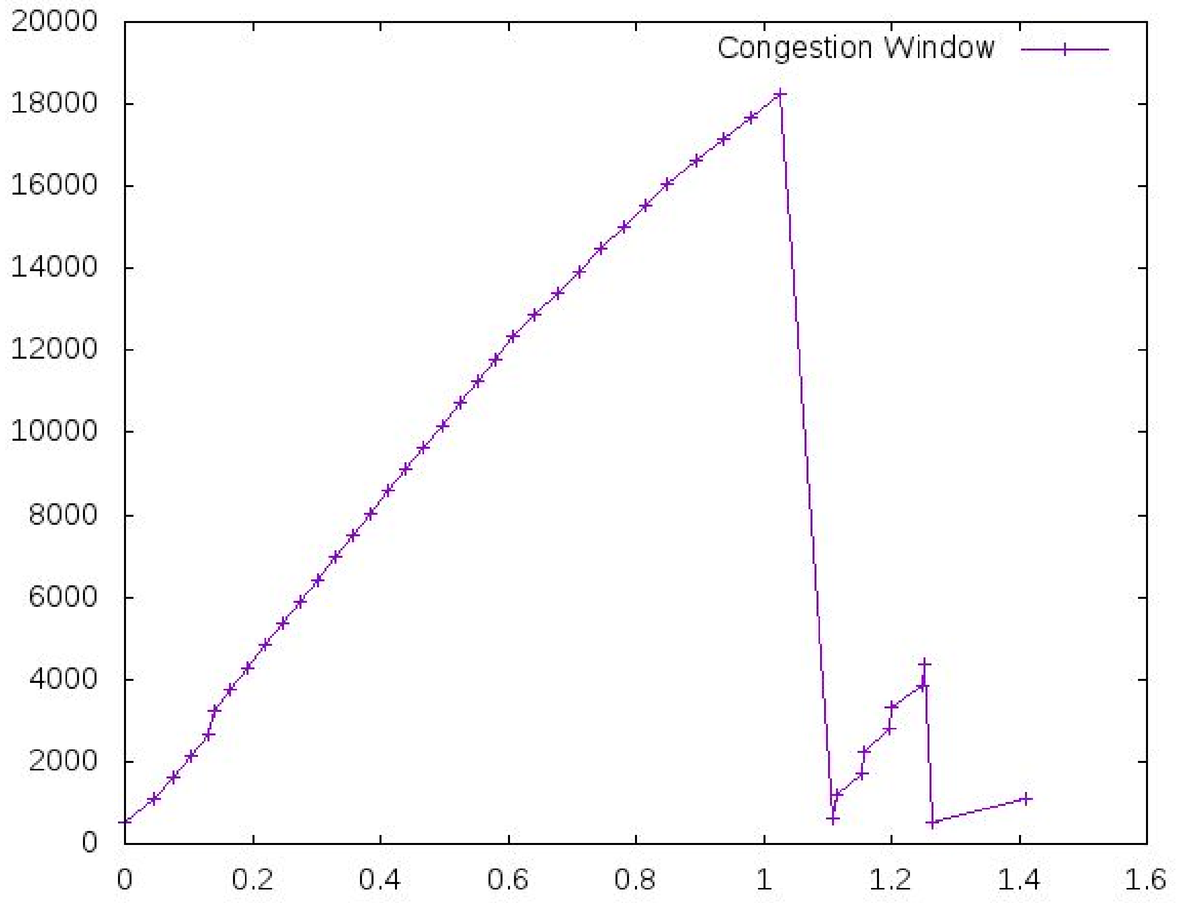
The TCP Reno protocol improves upon the performance of TCP Tahoe's congestion control algorithm, particularly for high bandwidth × delay product networks. TCP Reno uses a novel mechanism called fast recovery . Based on the TCP's ACK based method, the receipt of any ACK with the inclusion of a DUPACK on the sender's side indicates the arrival of data at the receiver and is interpreted as bandwidth available in the network. To take advantage of the available bandwidth, instead of reseting to the slow-start phase after leaving the fast retransmit phase as in TCP Tahoe, TCP Reno employs the fast recovery mechanism.

This allows the source to transmit a new segment on the receipt of each DUPACK assuming the source is not constrained by the sender's congestion window nor the receiver's advertised window. When a new ACK receives, the source transfers back to the congestion avoidance phase eliminating the reseting of the congestion window and the slow start stall times in the process.



NewReno-congestion window vs time

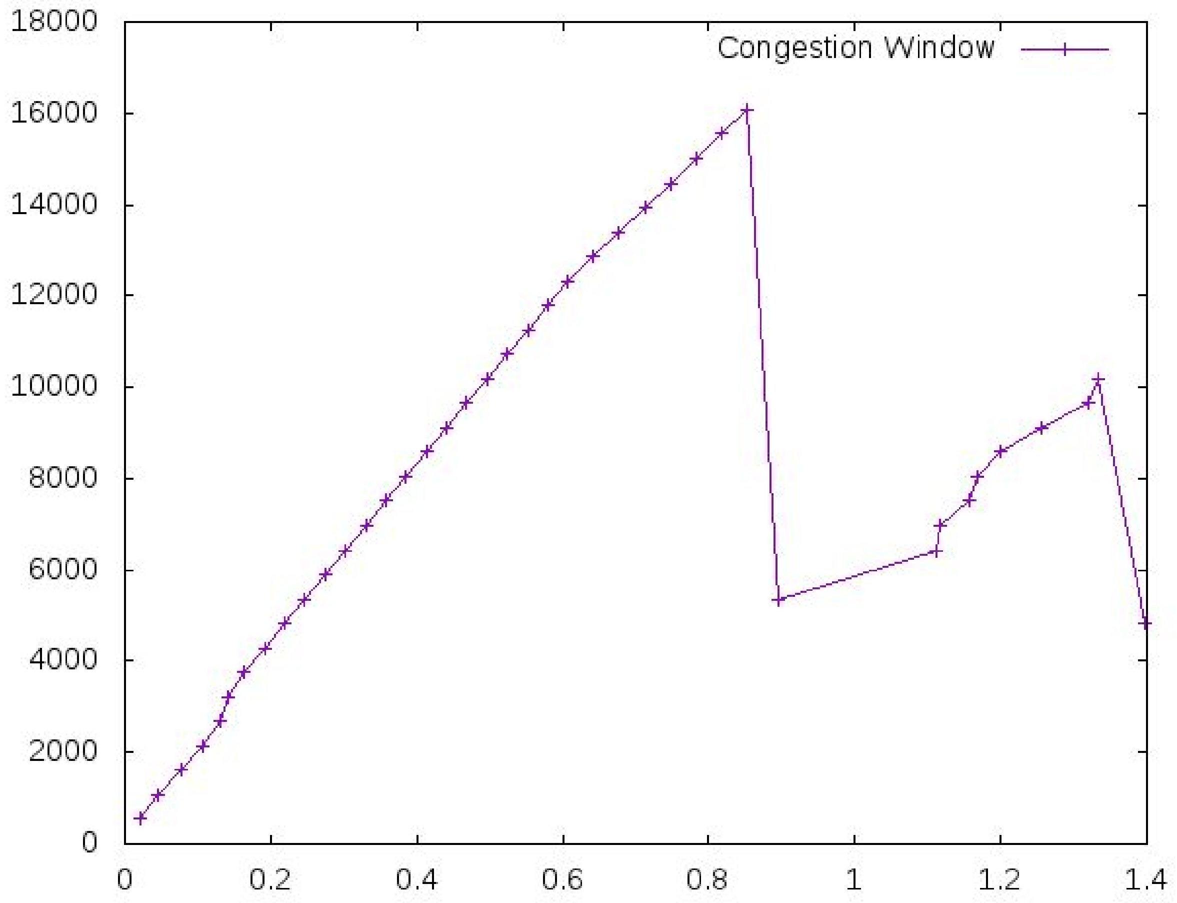
Explanation:
TCP NewReno consists of a slight modification to Reno's fast recovery algorithm resulting in higher performance, especially in the presence of multiple segment losses per sending window. Unlike Reno, NewReno distinguishes between a full new ACK and a partial new ACK during its fast recovery phase. A full new ACK or a full ACK is an ACK that acknowledges all the outstanding segments before the sender enters the fast recovery while a partially new ACK or a partial ACK acknowledges only a fraction of the sent data. TCP NewReno treats a partial ACK as an indication that the segment following the ACK has been lost. It remains in the fast recovery retransmitting all of the missing segments until a full ACK is received. This optimization allows TCP NewReno to recover from multiple losses faster than TCP Reno by not having to wait for a retransmission timeout or re-enter fast retransmit.



Westwood-congestion window vs time

Explanation:

TCP Westwood [14] is a sender side modification to TCP Reno that adjusts the sending rate based on a novel bandwidth estimation algorithm to improve performance in heterogeneous networks. The bandwidth estimation algorithm constantly monitors the rate of ACK reception while keeping an account on the number of DUPACKs and new ACKs.TCP Westwood allows the inclusion of DUPACKs in the calculation as each DUPACK indicates some data segments have reached the receiver. The amount of data acknowledged between ACK receptions is then used to compute the bandwidth of the link for the considered ACK interval. The calculated bandwidth is then passed through a Tustin approximation low pass filter to filter out the high frequency components.



vegas-congestion window vs time

Explanation:

TCP Vegas is a modified version of Westwood with a slightly different bandwidth estimation algorithm to reduce Westwood's aggressiveness in the presence of ACK compression. ACK compression is an Internet phenomenon in which ACKs arrive at a much closer spacing than when they were generated due to queuing delay in the network. Because Westwood estimates the network's bandwidth based on ACK interarrival times, ACK compression causes West-wood to overestimate the bandwidth, resulting in fairness

problems in the existence of multiple TCP flows. To alleviate the effect of ACK compression, Vegas modifies the bandwidth estimation mechanism to perform the sampling every RTT instead of every ACK reception. The result is a more accurate bandwidth measurement that ensures better performance when comparing with Reno or NewReno but still be fair when sharing the network with other TCP connections.