# NETWORKS LAB

ASSIGNMENT 4

**GROUP 3**

*APPLICATION ID : **3***


Group Members:

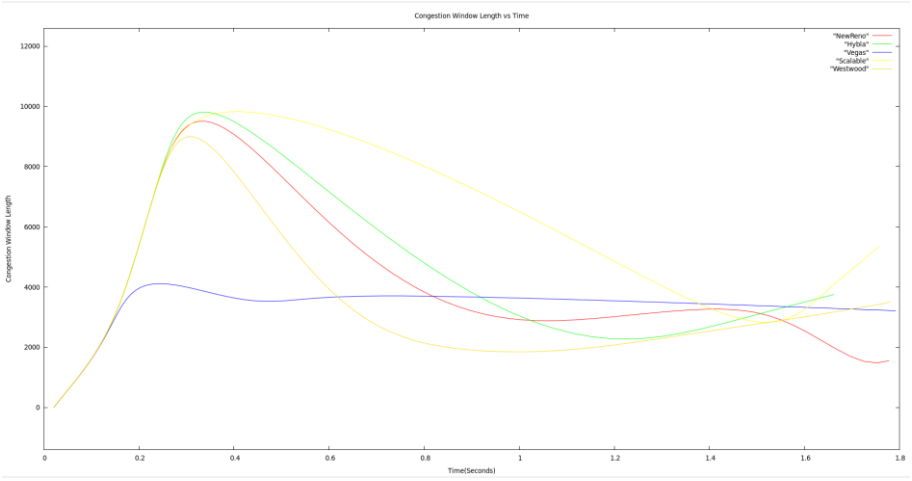
| | |
|---|---|
| **Aman Mishra** | **170101005** |
| **Keerti Harpavat** | **170101031** |
| **Priyanshu Singh** | **170101049** |


**Q. 1**

Transmission Control Protocol (TCP) uses a network congestion-avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease (AIMD) scheme, along with other schemes including **slow start** and **congestion window**, to achieve congestion avoidance. The **TCP congestion-avoidance algorithm** is the primary basis for congestion control in the Internet. Per end-to-end principle, congestion control is largely a function of internet hosts, not the network itself. There are several variations and versions of the algorithm which are:

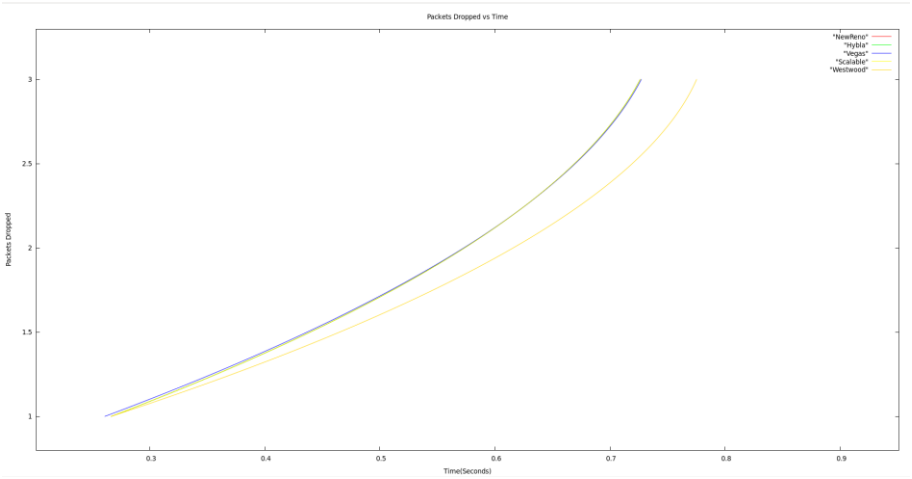
1. **TCP NEW RENO**: TCP New Reno, improves retransmission during the fast-recovery phase of TCP Reno. During fast recovery, to keep the transmit window full, for every duplicate ACK that is returned, a new unsent packet from the end of the congestion window is sent. For every ACK that makes partial progress in the sequence space, the sender assumes that the ACK points to a new hole, and the next packet beyond the ACKed sequence number is sent.
2. **TCP HYBLA**: TCP Hybla aims to eliminate penalties to TCP connections that incorporate a high-latency terrestrial or satellite radio links. Hybla improvements are based on analytical evaluation of the congestion window dynamics.
3. **TCP WESTWOOD**: TCP Westwood (TCPW) is a sender-side-only modification to TCP New Reno that is intended to better handle large bandwidth-delay product paths (large pipes), with potential packet loss due to transmission or other errors (leaky pipes), and with dynamic load (dynamic pipes).
4. **TCP SCALABLE**: Type of Transmission Control Protocol which is designed to provide much higher throughput and scalability. Standard TCP recommendations as per RFC 2581 and RFC 5681 call for congestion window to be halved for each packet lost. Effectively, this process keeps halving the throughput until packet loss stops. Once the packet loss subsides, slow start kicks in to ramp the speed back up.
5. **TCP VEGAS**: It is a TCP congestion avoidance algorithm that emphasizes packet delay, rather than packet loss, as a signal to help determine the rate at which to send packets. TCP Vegas detects congestion at an incipient stage based on increasing Round-Trip Time (RTT) values of the packets in the connection unlike other flavours such as Reno, New Reno, etc., which detect congestion only after it has actually happened via packet loss.

The attached graph is in accordance with the different stages of congestion control, like slow start and congestion avoidance.




**Q. 2**

The attached graph has different coloured lines for representing packet drops for different algorithms, some of which are overlapping.




**Q. 3**

The attached graph has different coloured lines for representing bytes transferred for different algorithms.