COMPUTER NETWORKS

ASSIGNMENT 2

SUBMITTED TO: **Dr Umar Farooq**

SUBMITTED BY: **AMINA QADEER**

Roll no: **359607**

CE-42-A

DATE: 28 NOVEMBER 2022

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING

TCP Taho and TCP Reno perform well for usual network conditions. However, their congestion avoidance mechanisms result in noticeably degraded performance for high-speed networks.

Q. 1) Can you calculate the time needed to achieve a data rate of 1 Gbps, using TCP Reno, where RTT is 2000 ms, current congestion window (CWND) size is 1 MSS, and last timeout had occurred at CWND = 128 MSS.

CWND= (Bandwidth*RTT)/(MSS*8)

CWND= 171

Time= 226 sec

The findings from Q.1 are sufficient to indicate the performance degradation faced by TCP in high-speed networks. Therefore, several alternatives have been developed.

Introduction:

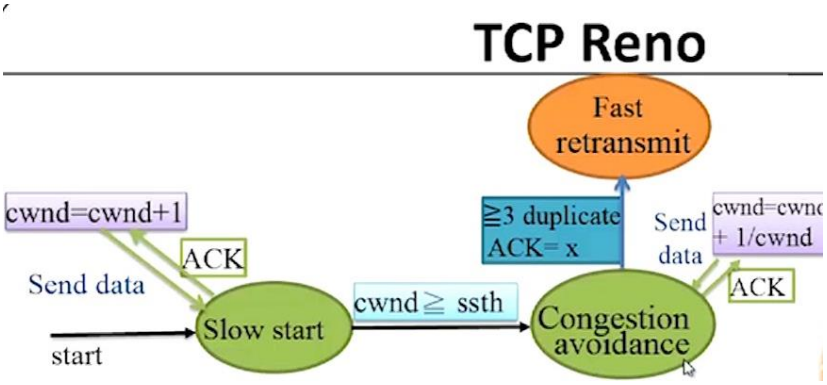
TCP (Transmission Control Protocol) is a connection-oriented communication protocol that helps in the exchange of messages between devices in a network. TCP takes messages from one application/server and divides it into packets, which is then forwarded by the devices in the network.

Congestion happens if the packets beyond the capacity of the network, which is a state in which the traffic of message is too heavy that it affects the response time. Time needed for a packet to reach the destination is increased. There are quite a few negative effects of congestion one of them is that the delay increases affecting the performance negatively. The important aspect of TCP is that it can handle the congestion with help of network congestion-avoidance algorithm. To execute them we are provided with different variants of TCP which are as follows:

TCP detecting and reacting to packet loss:

- TCP New Reno
  New Reno is the extension of Reno in this if congestion occurs, it retransmits the packets and enters a new mechanism that is fast recovery. Its advantage is that it detects multiple packet loss. Moreover, it does not leave the fast recovery until it receives acknowledgment of all packets, present in the window. New Reno has modified congestion avoidance mechanism and has slow start hence, few retransmits only. However, it takes one RTT for detecting packet loss. TCP Reno reacts differently to detecting a packet loss due to either timeout or 3 duplicate acknowledgements.
  In case of time out:
  1.cwnd is set to 1
  2.after a slow start window grows exponentially to threshold and goes linear onwards.
  In case of loss indicated by 3 duplicate acknowledgments:

1.cwnd is cut in half window and then grows linearly.
2.Duplicate acknowledgments means network holds capacity of delivering segments of packet.



- TCP TAHO

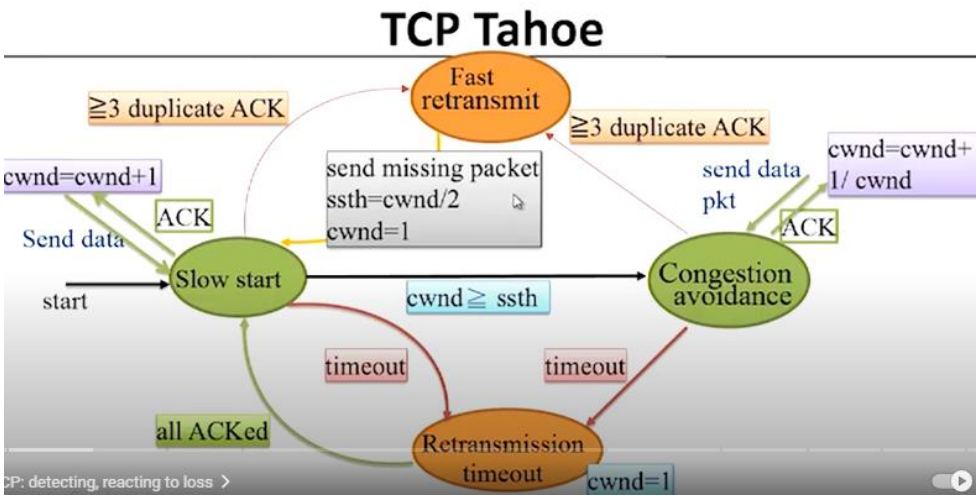
  In case of packet loss, TCP Tahoe always set the cwnd to 1 either the cause was 3 dup acknowledgements or time out.

  Mechanism starts with a slow start. After every acknowledgment TCP Tahoe increments the cwnd size by 1 and keep sending data. This process continues until window reaches a threshold. Now its time for congestion avoidance mechanism to play its role. This time we increment cwnd by one, after gaining acknowledgment for entire cwnd unlike the start. Data is sent to packet after acknowledgments.

  During this process, if timeout occurs. TCP runs retransmission of packets. Cwnd is set to 1. Once we get all acknowledgments for what we have retransmitted, we return to slow start and the loop goes on.

  Meanwhile if we are in congestion avoidance phase (additive increase) If we get 3 duplicate acknowledgments before timeout, fast retransmission of plackets will take place. You send the missing packets before waiting for time out to occur. Threshold value is set to half the window size. And congestion window size is set to 1. The process goes on to start and repeat itself.

- Timeout → retransmission
- Threshold → congestion avoidance
- 3 duplicate acks →Fast retransmission



Q. 2) Can you compare and criticize the following TCP versions for their performance in high-bandwidth networks, wireless/loss-prone networks, and high-delay networks (impact of RTT on performance)

TCP congestion control and avoidance algorithms (CCAs) are an important connection tuning consideration, especially with high bandwidth/high latency broadband networks.

Based on how they determine network congestion to manage the congestion window (cwnd) buffer, there are typically three different types of CCAs:

Loss-based algorithms modify their buffer size in response to packet loss to assess congestion. Most CCAs, such as TCP Reno, New-Reno, and CUBIC, employ this measure.

delay-based algorithms, such as TCP Vegas, employ RTT fluctuations to identify congestion and adjust the buffer size.

Model-based: Modern algorithms like BBR, which are both rate- and delay-based and add packet pacing, employ a variety of parameters to identify congestion. Unlike BBRv1, BBRv2 additionally incorporates packet loss into the model.

FAST TCP: especially targeted at long-distance, high latency links

TCP CUBIC is one of the most popular congestion control/avoidance algorithms currently in use. A less aggressive and more methodical version of BIC, CUBIC has a window that is a cubic function of the amount of time since the last packet loss. TCP window growth is divided into two halves. The window soon increases in size in the first section to its pre-packet loss size. After the initial section, there is a pause that gives the network time to settle. Following this delay, the second CUBIC component searches for more bandwidth initially slowly and then quickly.

TCP WESTWOOD

The main novel concept of TCP Westwood is to monitor the pace of returning ACKs at the sender side to assess the bandwidth consumed by the connection. Compared to TCP Reno,

this enables a quicker recovery of the connection. When using wireless lines, the TCP Westwood mechanism is especially useful because conventional congestion algorithms sometimes mistake intermittent losses brought on by radio channel issues for congestion and thereby reduce the TCP window without necessity.

TCP Reno has been modified into Westwood, which is designed for lossy networks. It works well with wireless networks, and wireless networks with lossy links show the most benefit. While TCP Reno is equally susceptible to random loss and congestion loss and is unable to distinguish between the two, TCPW is also not very sensitive to random packet loss.

TCP Vegas is known as one of the best variants. It uses triple duplicate acknowledgments which always result in packet retransmission. Vegas has also introduced a new re-transmission mechanism for lost packets. It also calculates the round-trip transmissions and investigates the time of every transmission to re-transmit them if time is more than expected increasing the pace of detecting packets losses. emphasizes packet delay as a signal to decide the rate at which to deliver packets rather than packet loss. TCP-Vegas identifies network congestion based on growing RTT values of the packets in the connection, identifying it in advance of packet loss. This contrasts with TCP-Reno, which only detects congestion after it has occurred via packet losses.
TCP Vegas often overruns connections when the RTT is high and limits throughput when the RTT is low. TCP Vegas cannot get along well with other congestion management algorithms since it can throttle the connection and identify congestion faster than other techniques that rely mostly on packet loss.

CONCLUSION:
Because queueing delay gives a sharper indication of congestion and scales more organically with network capacity than packet loss probability does, TCP Vegas and FAST TCP strive to increase throughput and fairness over its predecessors, of which TCP Reno is the most well-known.
Because queueing delay is used by both FAST TCP and TCP Vegas as a congestion indicator, their window updating algorithms depend on propagation delay, which is calculated using a metric called baseRTT. The shortest round-trip time (RTT) to date is the metric used to describe it. Because the routers' queues might occasionally remain full, base RTT estimation of the actual propagation delay may be off, leading to unfairness and excessive queue variation.
It can be concluded from the information about different variants that there is none which can overcome the problem of congestion. All have their own pros and cons however, looking at all the parameters it can be seen that Vegas and newly CUBIC provides more efficiency and give better results.