**RFC 5681 – Summarization and Comparison with RFC 2001 and RFC 2581**

**Introduction:**

RFC 5681 describes four of the congestion control algorithms. This document obsoletes the older RFCs (RFC 2581 and RFC 2001) which are based on these congestion control algorithms. In this summarization, we will compare all the three RFCs and understand the reason for the change.

**Congestion Control Algorithms – Slow Start:**

When a new connection is setup, and the data transmission is started, TCP would send huge chunks of data into the network. This could cause congestion if there are intermediate devices which are slower than the others. There are possibilities that the intermediate devices run out of space and starts dropping the packets. An algorithm was devised to address this issue which is called Slow Start. This algorithm adds congestion window (cwnd) into the TCP of the sender. This is initially set to one segment, and whenever there is an ACK from the receiver, the size of the cwnd is increased by one segment. Congestion Window is the sender side limit, and the Receivers Advertised Window(rwnd) is the receiver side limit. The size of cwnd increases exponentially until there is a congestion in the network.

**Congestion Control Algorithms – Congestion Avoidance:**

Congestion Avoidance algorithm deals with the lost data due to the congestion in the network. This loss of data during the transmit can be found out when there is a timeout and when receiving duplicate acknowledgements. Though congestion avoidance and slow start are two different algorithms, they work together. Whenever there is a congestion in the network, the rate at which the data is transmitted into the network should be reduced which is taken care when slow start algorithm is used. These two algorithms require two variables to be added to the TCP whenever there is a new connection. The first one is the congestion window and the second one being slow start threshold size (ssthresh). When Congestion Window < Slow Start Threshold, the slow start algorithm is implemented and when the other way is true then congestion avoidance algorithm is applied. If both the values are same, then the sender can choose any one of the two algorithms. Congestion Window`s size increases exponentially when slow start algorithm is used. However, when congestion avoidance algorithm is used, Congestion Window`s size increases linearly.

**Comparison of above algorithms with newer RFCs:**

In addition to the description provided by RFC 2001 in the above paragraphs for a slow start and congestion avoidance algorithms, RFC 2581 suggests methods on how to increase the size of the initial window of the cwnd. It also suggests that during the congestion avoidance, the size of cwnd can be increased by a technique called Byte Counting. RFC 5681 which obsoletes RFC 2581 explicitly suggests Appropriate Byte Counting [RFC 3465] while not denying the use of above-suggested methods in RFC 2581. Also, RFC 2581 states that the ssthresh`s initial value MAY be arbitrarily high. This has been updated to SHOULD by RFC 5681.

**Congestion Control Algorithms – Fast Retransmit:**

Whenever there is data which is out of its order, then there will be an immediate acknowledgement from the receiver. The TCP sender on receiving this ACK does not know if it is due to the loss of data or due to the change in the order of the data. Hence it closes watches the duplicate ACKs. If more than two duplicate ACKs are received, then there are more chances that the data has been lost. Hence it retransmits the data on the network.

**Congestion Control Algorithms – Fast Recovery:**

When the missing data is retransmitted by the TCP using Fast Retransmit algorithm, the congestion avoidance algorithm is implemented rather than the slow start. This algorithm is called Fast Recovery. This is because there will still be data in the buffer which are flowing between the sender and receiver and implementing slow start will reduce the flow which isn’t an ideal way of handling this situation. Similar to above algorithms, Fast Retransmit and Fast Recovery Algorithms also work together.

**Comparison of above algorithms with newer RFCs:**

RFC 2581 suggests that whenever there is a detection of loss of data due to the retransmission timer, the ssthresh`s value must not be set more than the maximum value between half the flight size and twice the value of SMSS. This has been clarified in RFC 5681 stating that the ssthresh`s value must be equal to half of the flight size on the first retransmission and moving forward the value should be held constant. Also, RFC 5681 suggests the use of Limited Transmit (RFC 3042), limits the number of duplicate acknowledgements which artificially inflates during loss recovery, the value of the restart window is set based on the minimum value between the initial window and congestion window rather than only from the initial window.

**Additional Considerations:**

**Restarting Idle Connections:**

After an extended period of idle time, TCP congestion control algorithms tend to send an inappropriate chunk of traffic through the network. This is because it cannot use ACK clock as all the ACKs would have drained. The recommendation here is that, after an extended period of idle time, slow start algorithm should be used by the TCP which also starts the ACK clock.

**Generating Acknowledgements:**

TCP receiver can delay the acknowledgements which are being. This process is called delayed acknowledgements. When doing so, the delay should not be excessive. Also, the acknowledgements should be sent immediately when the received segments are out of order, and when the missing data is received, the same process has to be followed to notify the sender.

**Loss Recovery Mechanisms:**

There are possibilities where multiple losses occur in a single window of data. The recommendation here is that the implementors of TCP should implement advanced loss recovery algorithm to take care of this scenario.

**Security Considerations:**

Currently, on receiving an ACK, the size of the cwnd is increased based on a constant number. The recommendation here is that the size of the cwnd should be increased based on the number of acknowledgements received. On successful implementation of this recommendation, we could avoid the congestion collapse and preserve the network stability.