

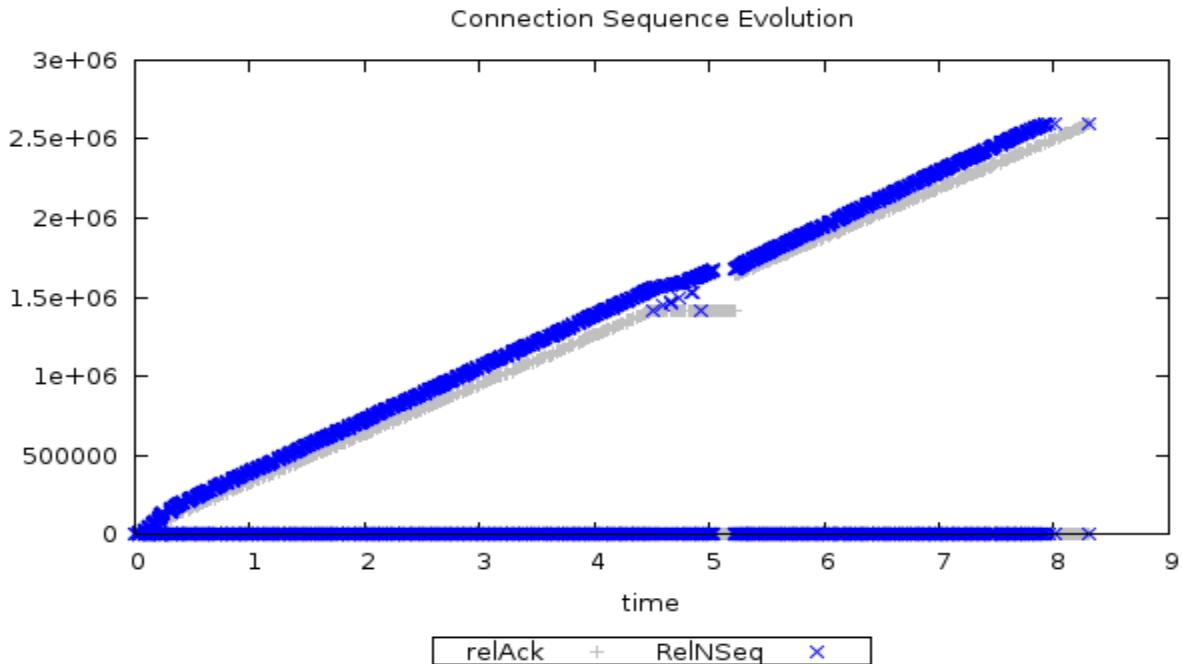
# Flow control during fast recovery

Example connection2

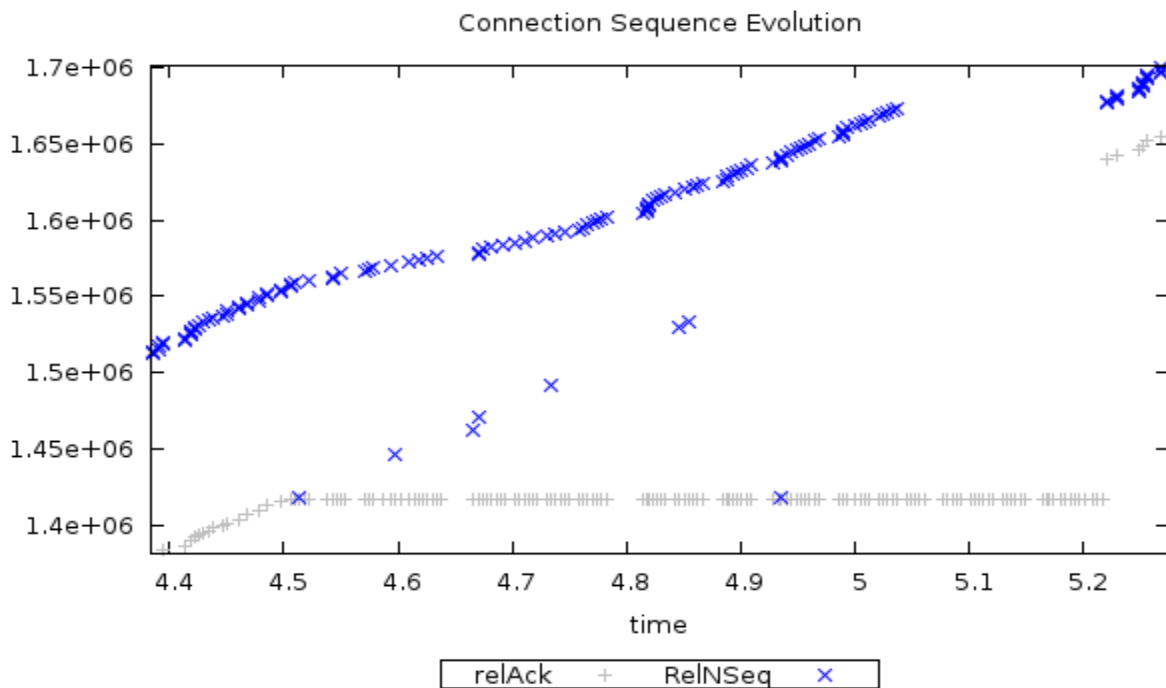
Alejandro Popovsky, Universidad de Palermo, Argentina, 2013-Aug-12

This is a second example of a connection where flow control limitation is reached during fast recovery. This time the receiver “auto-tunes” its receive window, but not as dynamically as it would need to do during fast recovery, as seen below.

The following graph shows the connection sequence and ack evolution:



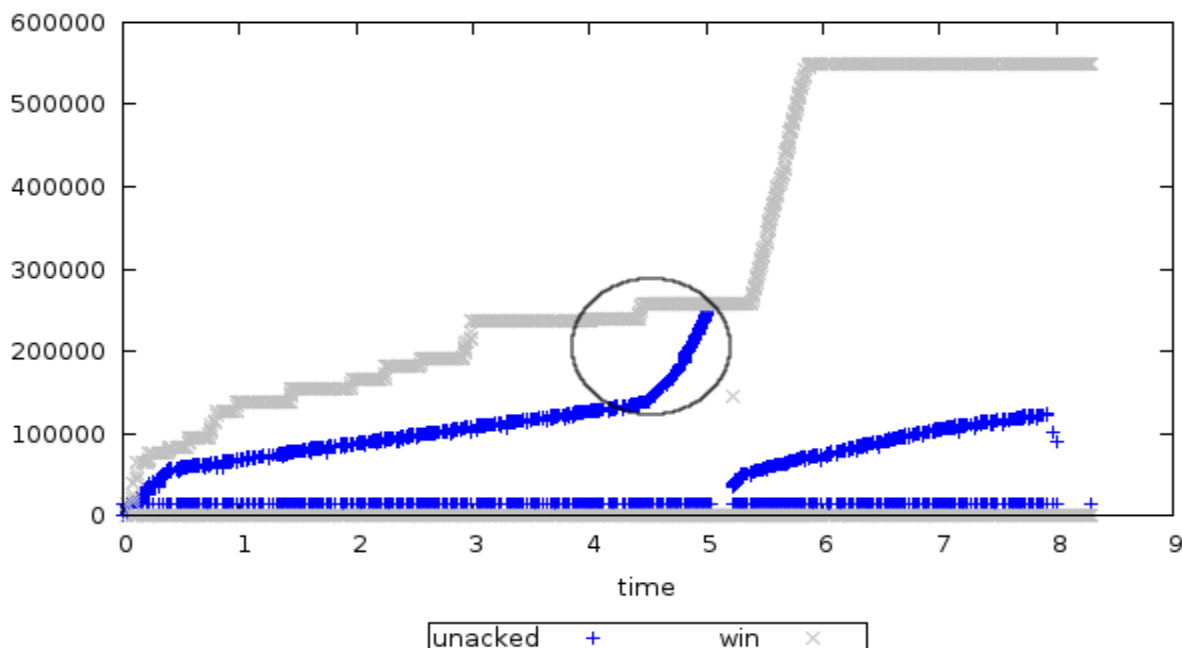
The graph was captured very close to the sender, so time reference is almost the same as the sender's. If we zoom on the fast recovery section:



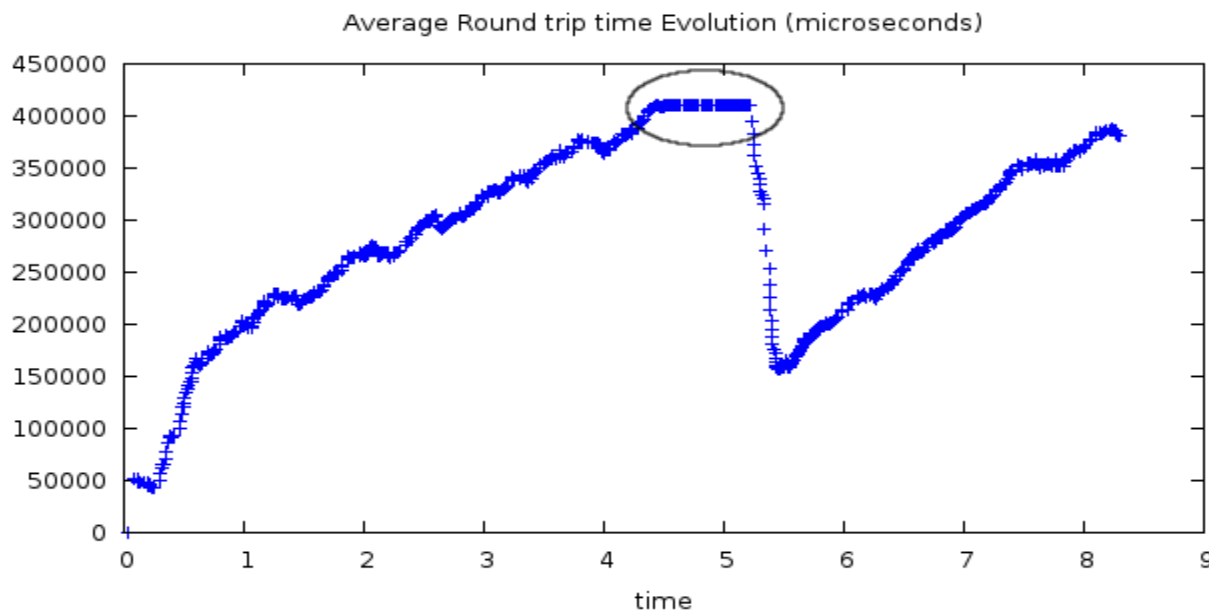
We can see the sequence number growing while the ack sequence remains the same (duplicate acks), until the sequence number stops growing. This is because the sender has filled his perceived space in the receiver window.

The retransmissions fill the gaps learned from the selective acknowledgments piggybacked in the duplicate acks. And when the ack of the filled sequence arrives, the sender can resume normally.

Lets see the number of unacked bytes and receive window evolution:



We can see the the receiver is adapting its receive window as the unacked byte count grows, and the average round trip time grows also:



The bottleneck router or firewall in the path has a large output queue buffer and the sender is slowly growing its congestion window exploring the capacity of the path. This growth in congestion window is not increasing the data rate at the receiver, because the output rate from the bottleneck interface remains constant.

*But during this fast recovery sequence, the unacked bytes count grows faster, while the round trip time remains constant, so the receiver stops growing its receive window, until the perceived window space at the sender goes to zero, causing a stall at the sender.*

The pcap dump for the connection can be obtained from:

<http://www.palermo.edu/ingegneria/comm/exampleDumpFlowCtrlFastRecovery2.pcap>

