CPEG 572 Data and Computer Communications

Report No. 8

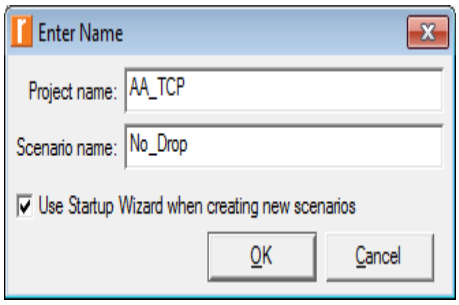
Summary:

This lab is created to show the congestion control algorithms implemented that is done by TCP. This lab has many scenarios to reproduce theses algorithms, and we are comparing and analyzing the consequences.

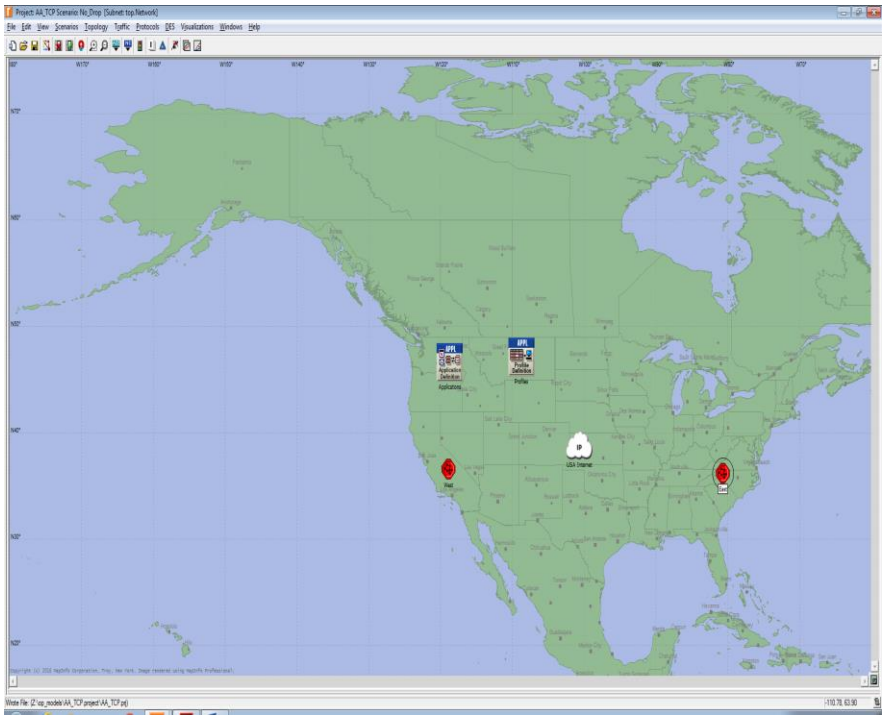
Implementation:

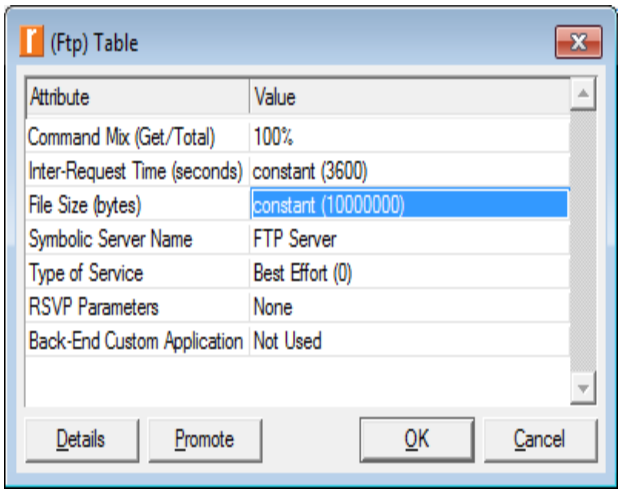
The TCP is steady for the internet to assure the delivery of stream of bytes. Also, it has flow -control mechanism to allow how much data is received by the receiver, and that data is sent by the sender. The TCP is using the congestion control to do how much capacity is applicable in the network that means how the number of packets are transmitted. Furthermore, TCP uses additive increase/multiplicative decrease to reduce the congestion widow the level of congestion is going to the top and vice versa. Show start is another mechanism is used by TCP to increase the congestion window "rapidly" from a cold start in TCP connections. So, in this lab we will create the network that handles TCP as its end-to-end transmission protocol and analyze the size of the congestion window with another different mechanisms.

We will implement the network which has Application Config, Profile Config, ip32_cloud, and two subnets.

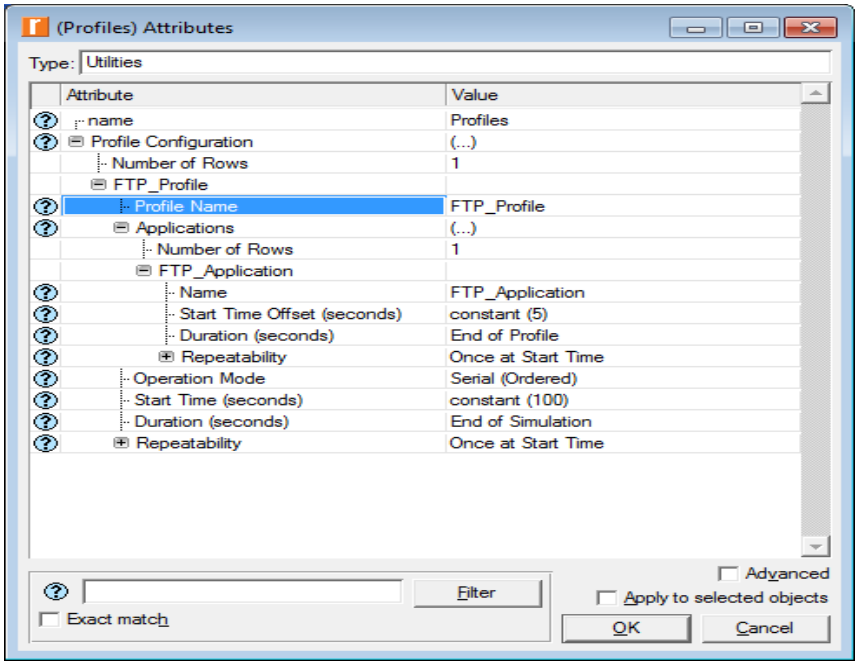


New Project

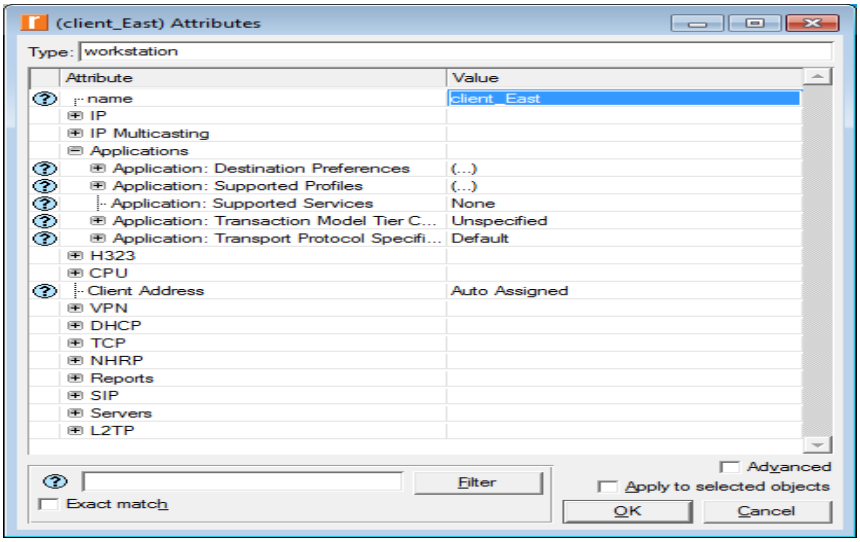




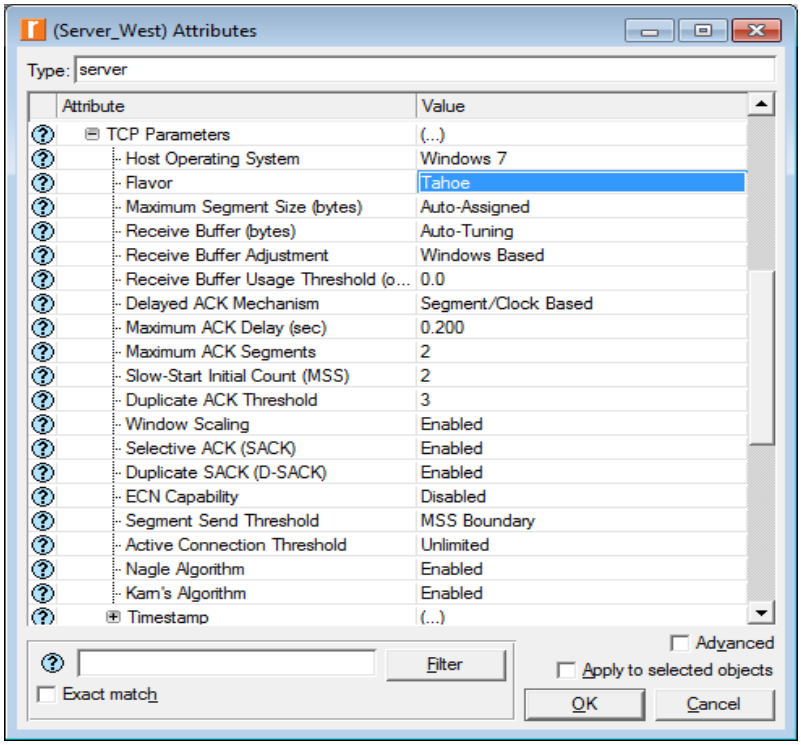
Application Config



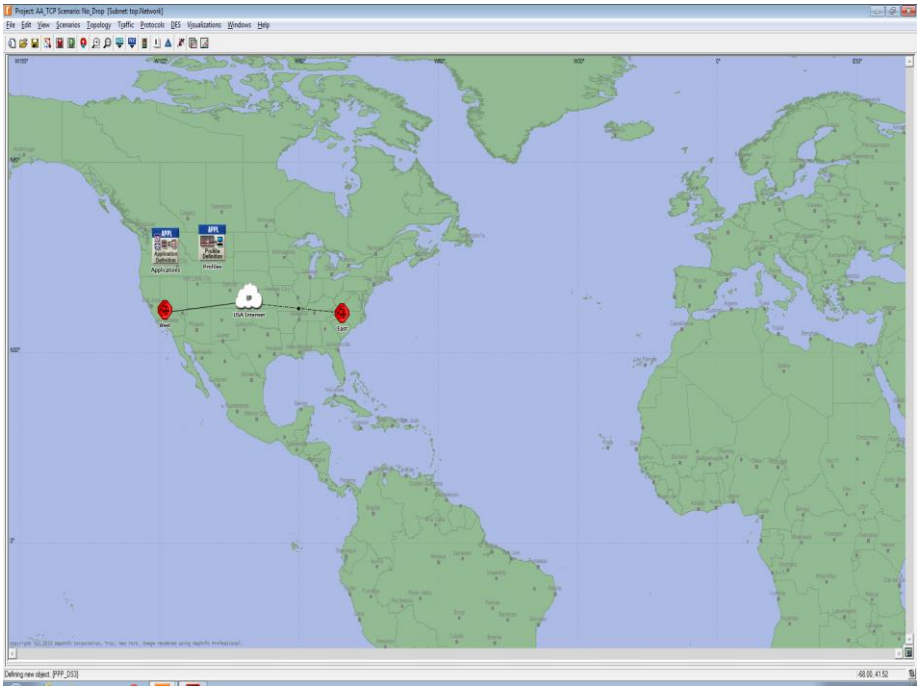
Profile Config



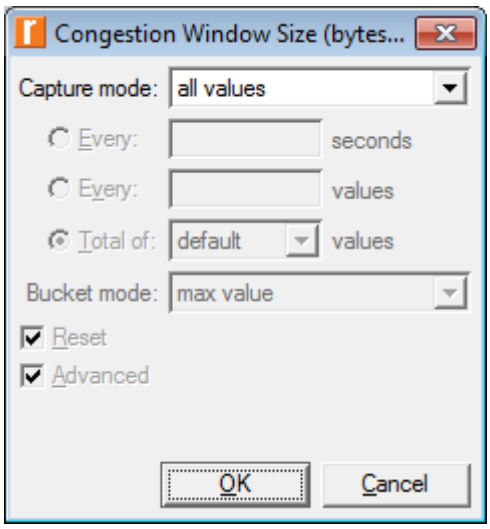
East Subnet Config



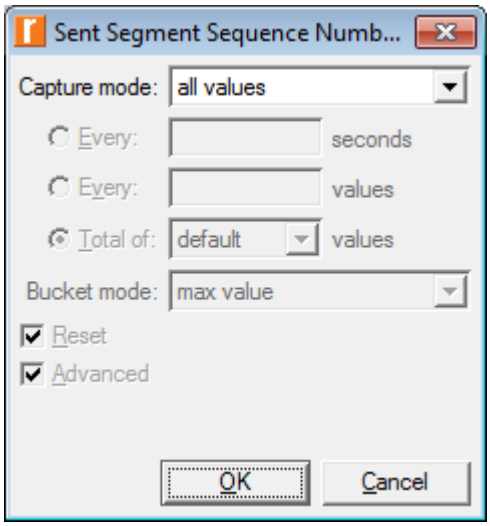
West Subnet Config



Connect Subet to Cloud

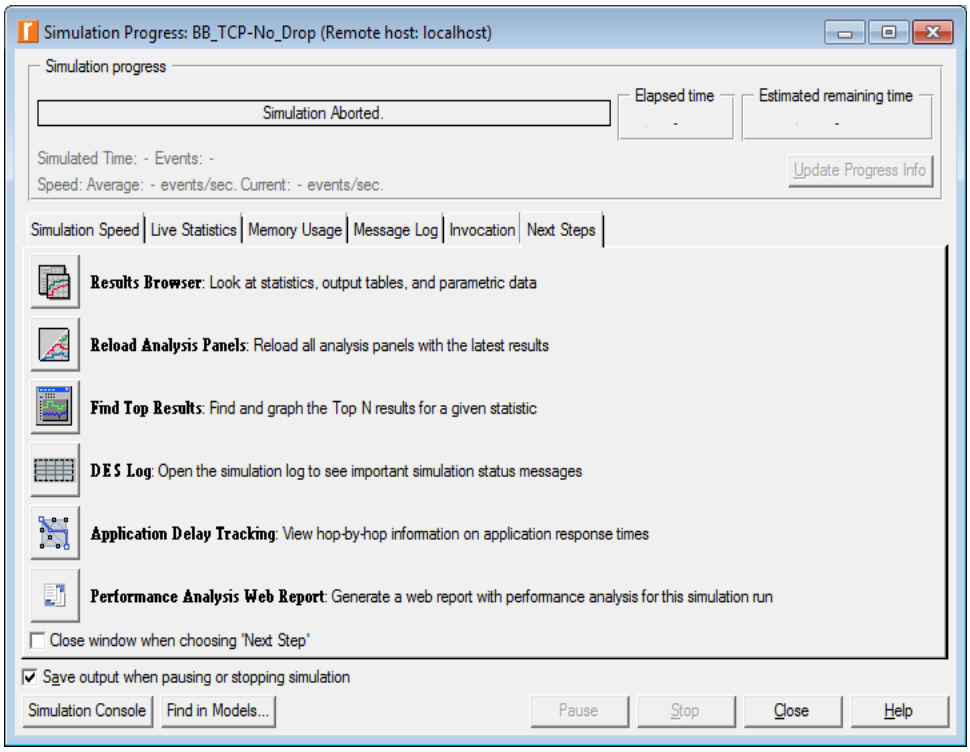


Congestion window size

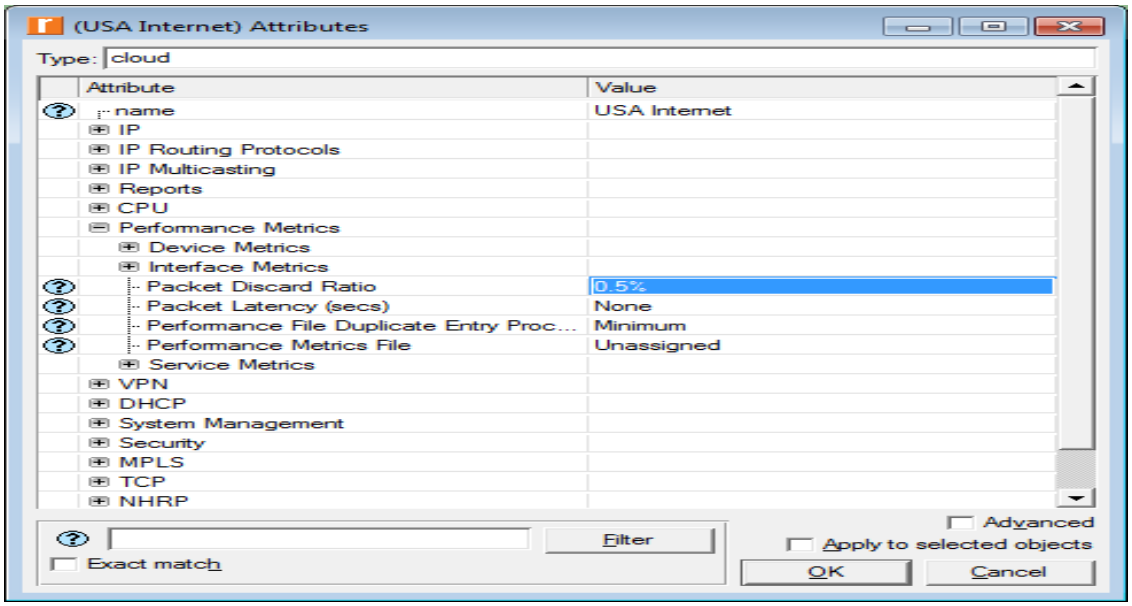


Sent Segment Sequence

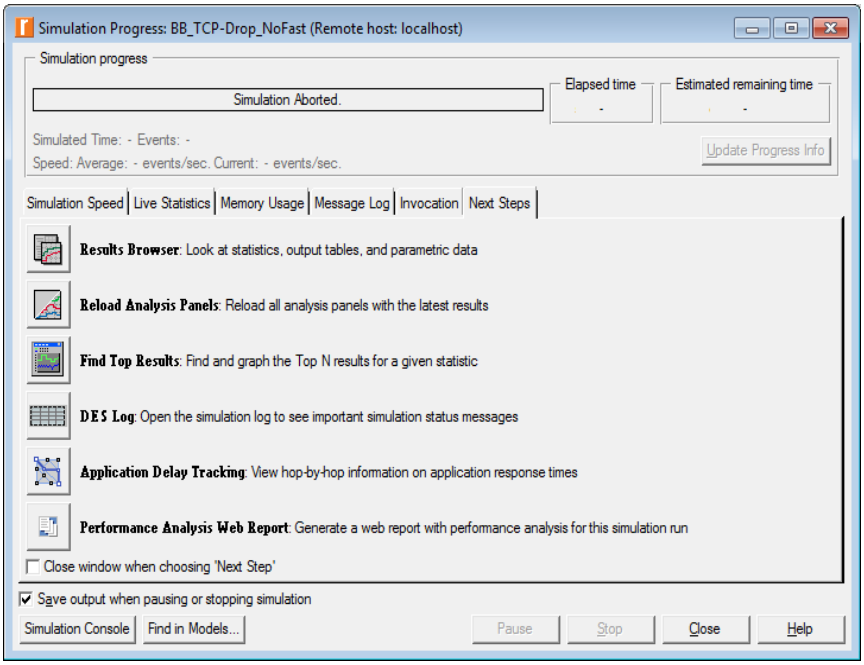
Configure Simulation:



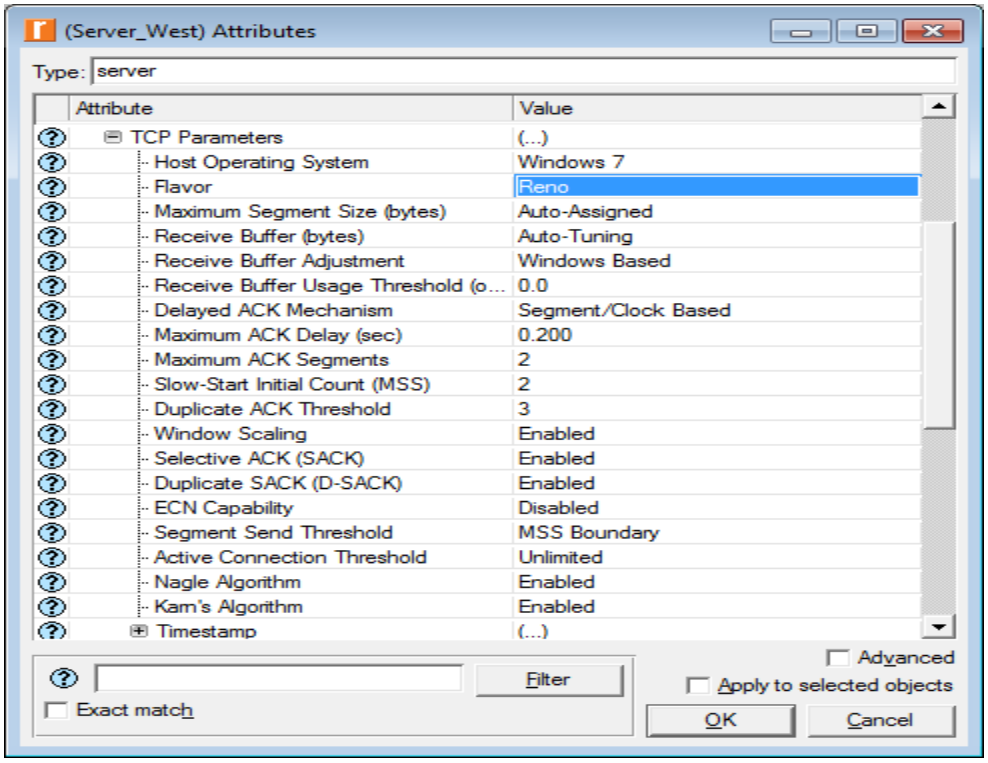
Execution of 1st scenario



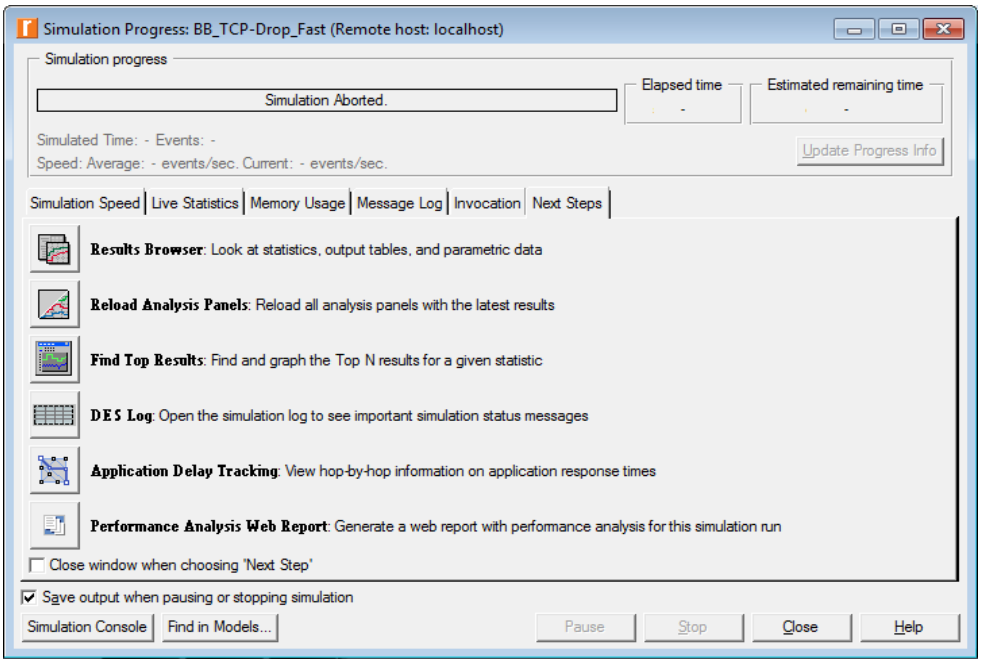
Duplicating scenario, NOFAST



Execution of 2nd scenario

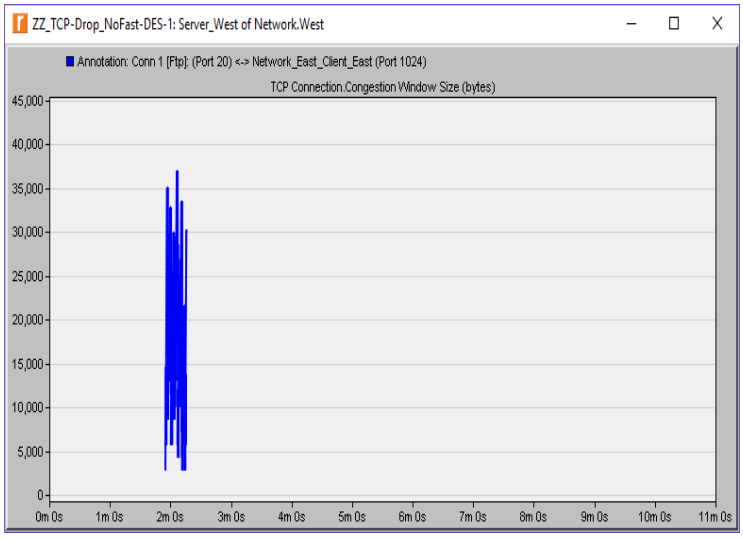


Duplicating Scenario, FAST

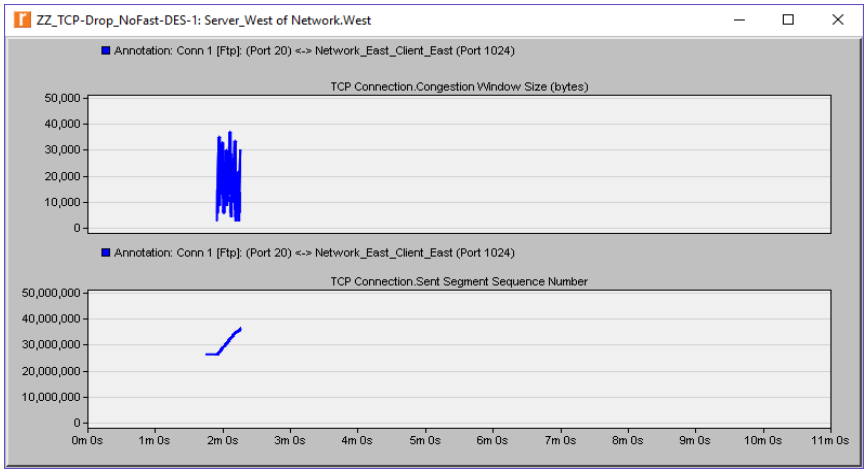


Execution of 3rd scenario

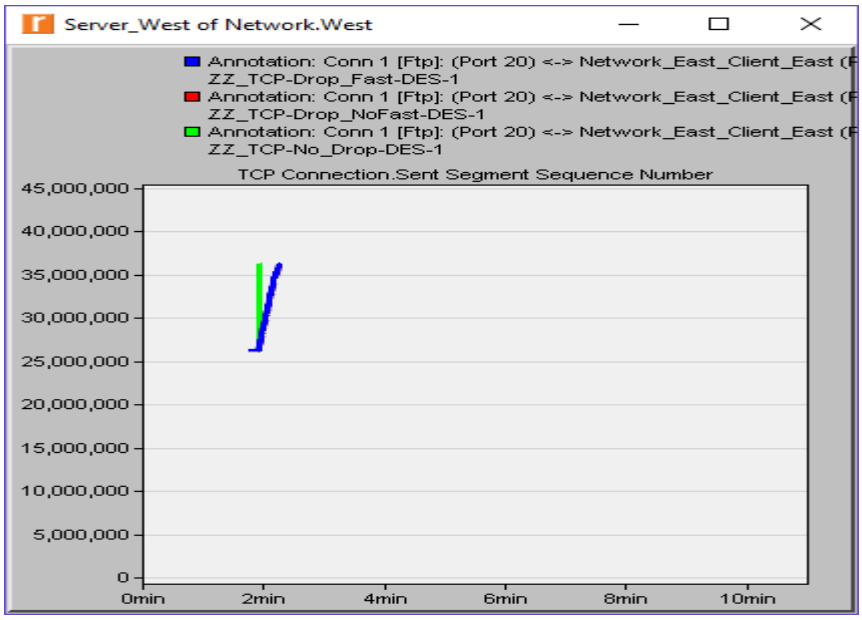
Results:



Congestion window size



Comparison of congestion window sizes



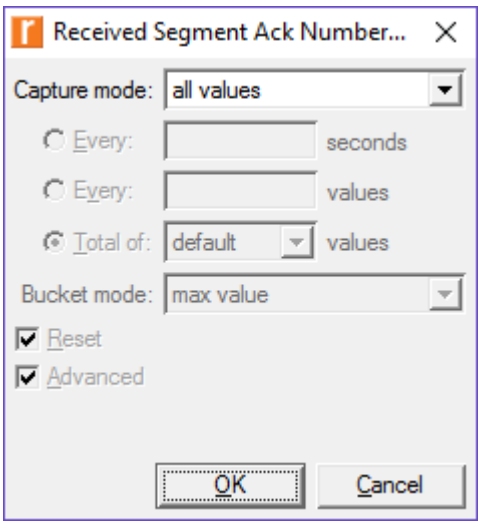
Comparison of sent sequence number

Answers:

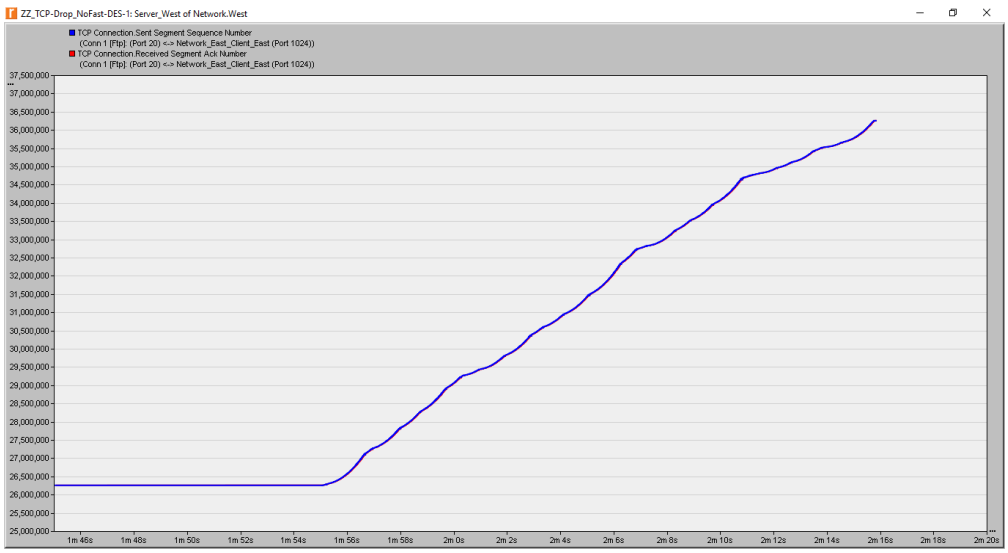
1. Why does the Segment Sequence Number remain unchanged (indicated by a horizontal line in the graphs) with every drop in the congestion window?
   It happened due to congestion in the network and packets were starting to drop by decreasing congestion window in size when it detects timeouts which it is done when the packet were assigned to the IP cloud to 5%

2. Analyze the graph that compares the Segment Sequence numbers of the three scenarios. Why does the Drop_NoFast scenario have the slowest growth in sequence numbers?
   Since the Drop_NoFast scenario grows slowly because the packet is 5% that means there are timeouts in the network every once in a while. The Drop_Fast when the retransmit is done, there are are three round trips worth of time is through before the packets are retransmi. Thus, the recovery from dropped is more speedy.

3. In the Drop_NoFast scenario, obtain the overlaid graph that compares Sent Segment Sequence Number with Received Segment ACK Number for Server_West. Explain the graph.



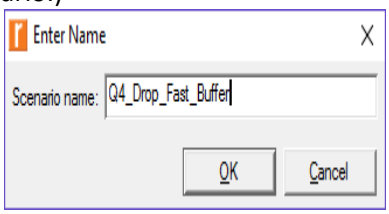
Capture mode: All values



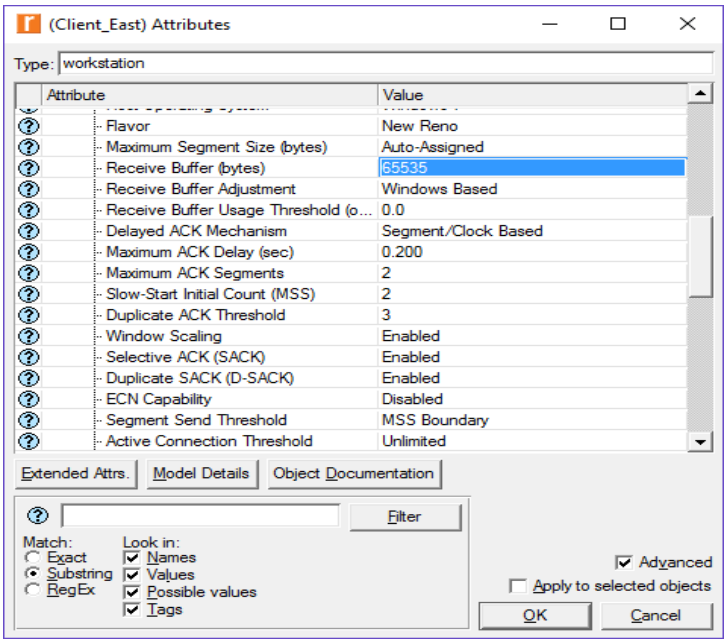
Comparison: sent sequence number vs receiver segment ACK number

The two lines are very close to each other, almost above each other. That means the client acknowledges at the same time

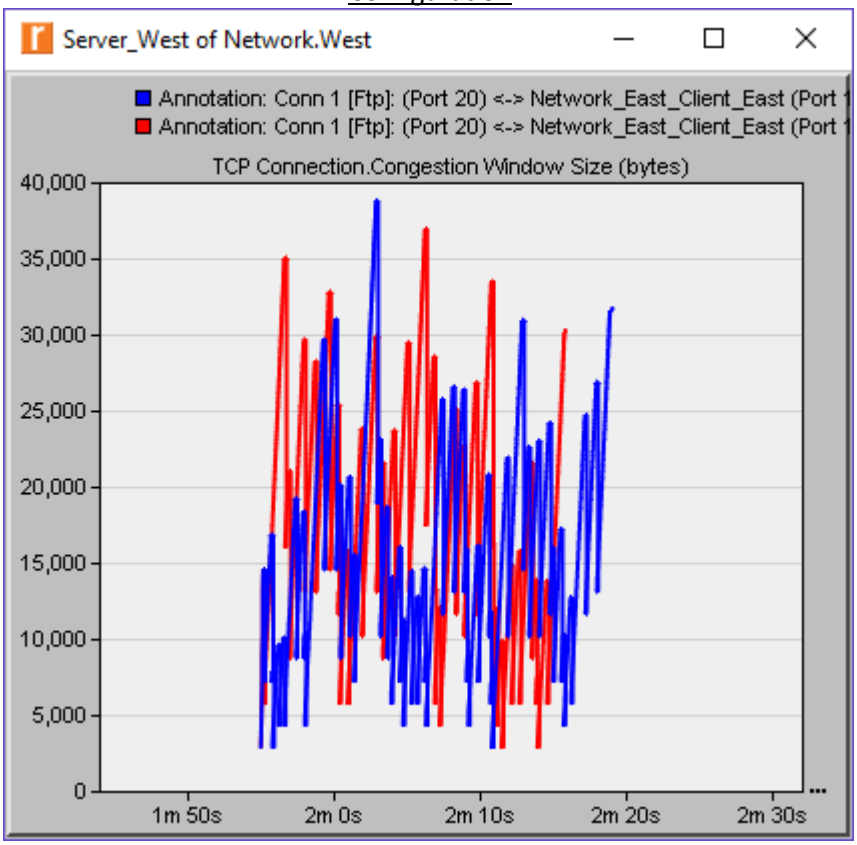
4. Create another scenario as a duplicate of the Drop_Fast scenario. Name the new scenario Q4_Drop_Fast_Buffer. In the new scenario, edit the attributes of the Client_East node and assign 65535 to its Receiver Buffer (bytes) attribute (one of the TCP Parameters). Generate a graph that shows how the Congestion Window Size (bytes) of Server_West gets affected by the increase in the receiver buffer (compare the congestion window size graph from the Drop_Fast scenario with the corresponding graph from the Q4_Drop_Fast_Buffer scenario.)



Scernario: Q4_Drop_Fast_Buffer



Configuration



Comparison: Congestion Window Size

The red line is Congestion Window Size in Q4_Drop_Fast_Buffer scenario and the blue line is for Congestion Window Size in Drop_Fast scenario. It is obvious the blue line has reached the top before the red line that mean the results in the entire session completing in about a quarter of the time than with the smaller receive buffer.

Conclusions:

This lab shows the simulation of the fundamental of the congestion control mechanisms in TCP. We examined three scenarios that showed how the Congestion Window Size is growing and the differences between them and how they were transmitted. Furthermore, it was noticed that on the segment sequence number and segment acknowledgement number, and we compared the Congestion Window Size between two scenarios after changing Receiver Buffer for those scenarios.