COL334 - Assignment 4 - Part A

Name : Ujjwal Mehta | Entry No. : 2020CS10401

Description of Part A:

In this part we have created our new Tcp protocol namely TcpNewRenoPlus which is an extension of TcpNewReno protocol with changes reflected in the **Slow Start phase** and the **Congestion Avoidance phase** and we have tested and compared this new extension and with original TcpNewReno protocol on the topology described in the assignment.

Code Explanation:

The code for this part consists of 4 Files namely, **TcpNewRenoPlus.cc** , **TcpNewRenoPlus.h** , **partA.cc** and **plot_partA.py** where the first 2 files contain the code of our new TcpNewRenoPlus protocol which is implemented via deriving the TcpNewRenoPlus class from the parent class TcpNewReno (base class) and here we have implemented the TcpNewRenoPlus class methods in a similar way as present in the **tcp-congestion-ops.cc** file just the only changes are in the methods **TcpNewRenoPlus::SlowStart** and **TcpNewRenoPlus::CongestionAvoidance** where we do updation for our congestion window as follows:

Slow Start:

$Cwnd = Cwnd + ((SegmentSize)^{1.91}/Cwnd)$

Congestion Avoidance:

$Cwnd = Cwnd + 0.51 * SegmentSize$

Finally after creating these files inside the "ns-allinone-3.29/ns-3.29/src/internet/model/" directory and updating the wscript file, we can use this new TcpNewRenoPlus protocol by setting this Tcp protocol as default in our network topology file via the following command:

```
Config::SetDefault ("ns3::TcpL4Protocol::SocketType", StringValue
("ns3::TcpNewRenoPlus"));
```

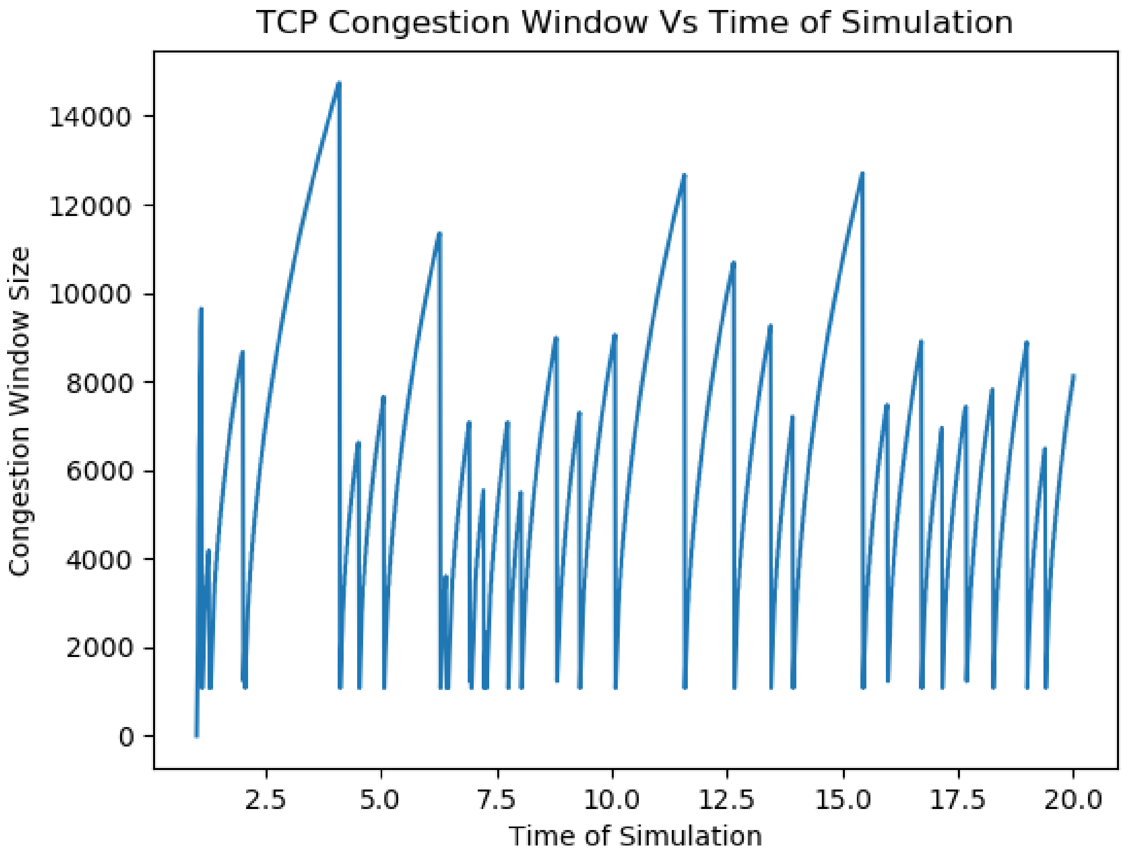
After this, in order to create the given network topology , I have taken the help of example code from seventh.cc file and modified it to build the given topology by using 2 point to point links, storing the linked devices in 2 net devices, assigning the Ip address to each net device and then finally creating 3 tcp-sink applications at Node 3, 2 tcp-source applications at Node 1 and 1 tcp-source application at Node 2.

Finally upon running the given topology file, 3 files namely **"socket1.txt", "socket2.txt" and "socket3.txt"** are created which contains the congestion window size at the corresponding tcp-connection source along with the time stamp. Then we save all the plot files using **plot_partA.py** file which uses **Matplotlib Library** for plotting after reading and storing all the corresponding socket.txt files.

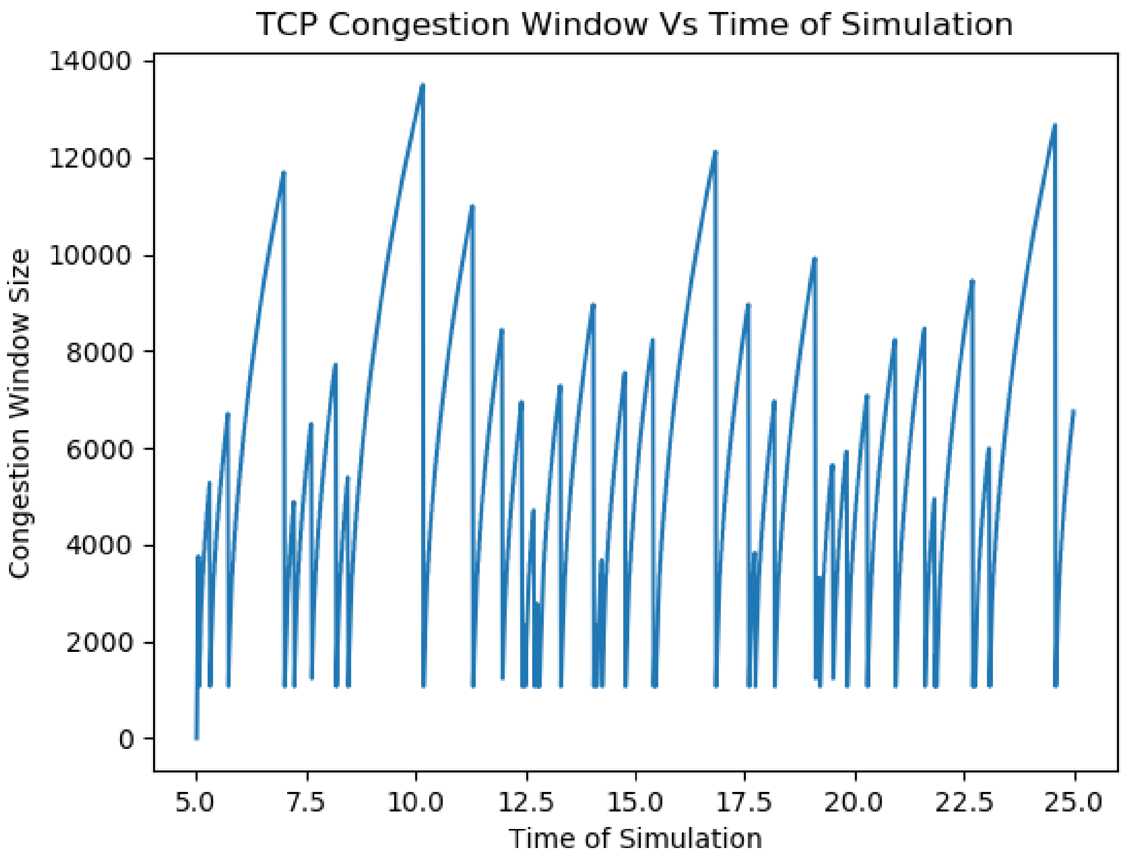
Plot of Congestion Window Size Vs Time:

(a) TcpNewReno

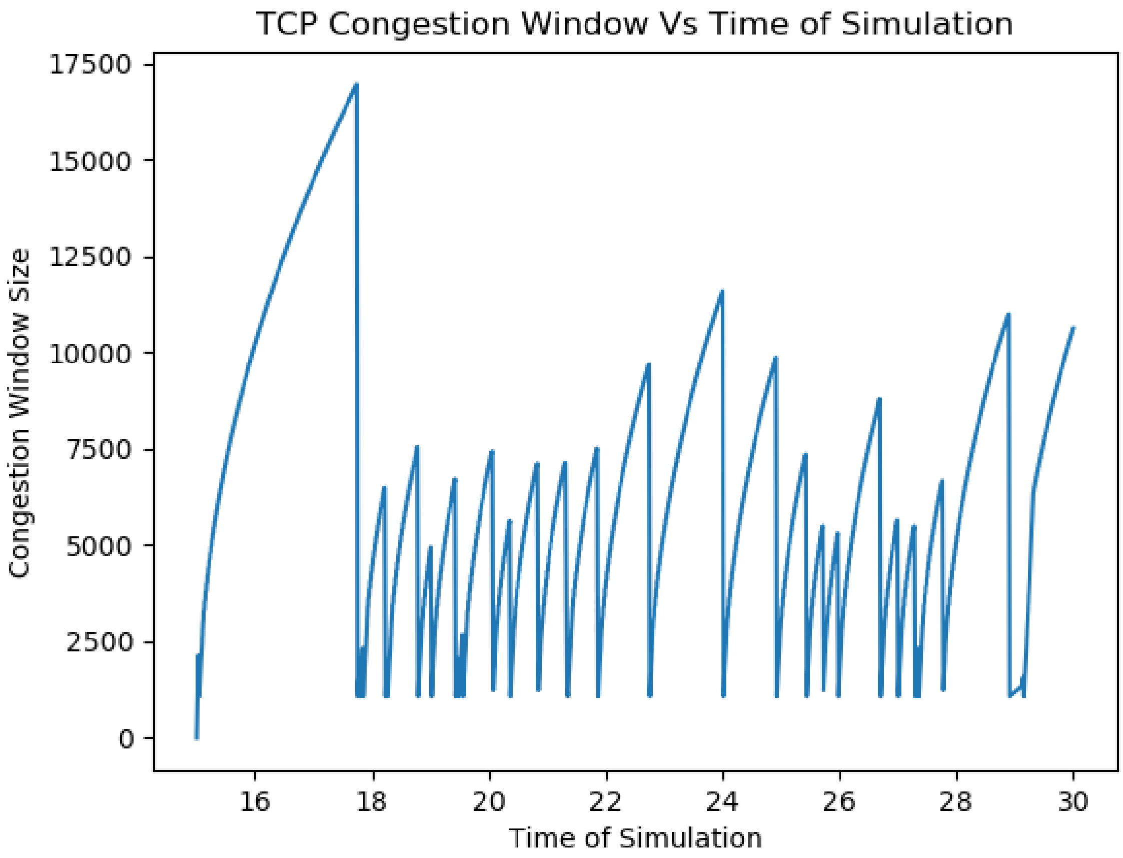
1. **Connection-1**
   For tcp-connection 1 source, we get the maximum window size as 14749. The corresponding plot is as follows:



2. **Connection-2**
   For tcp-connection 2 source, we get the maximum window size as 13500. The corresponding plot is as follows:

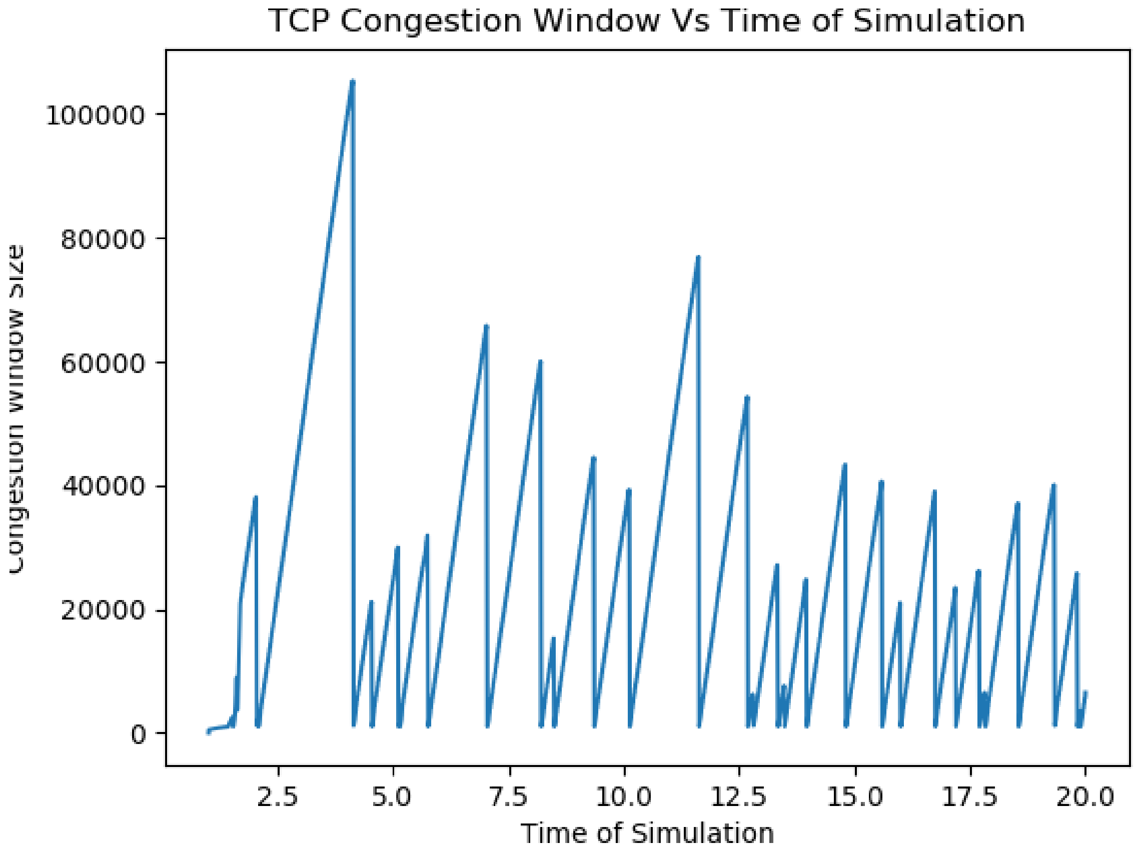


3. **Connection-3**
   For tcp-connection 3 source, we get the maximum window size as 16958. The corresponding plot is as follows:

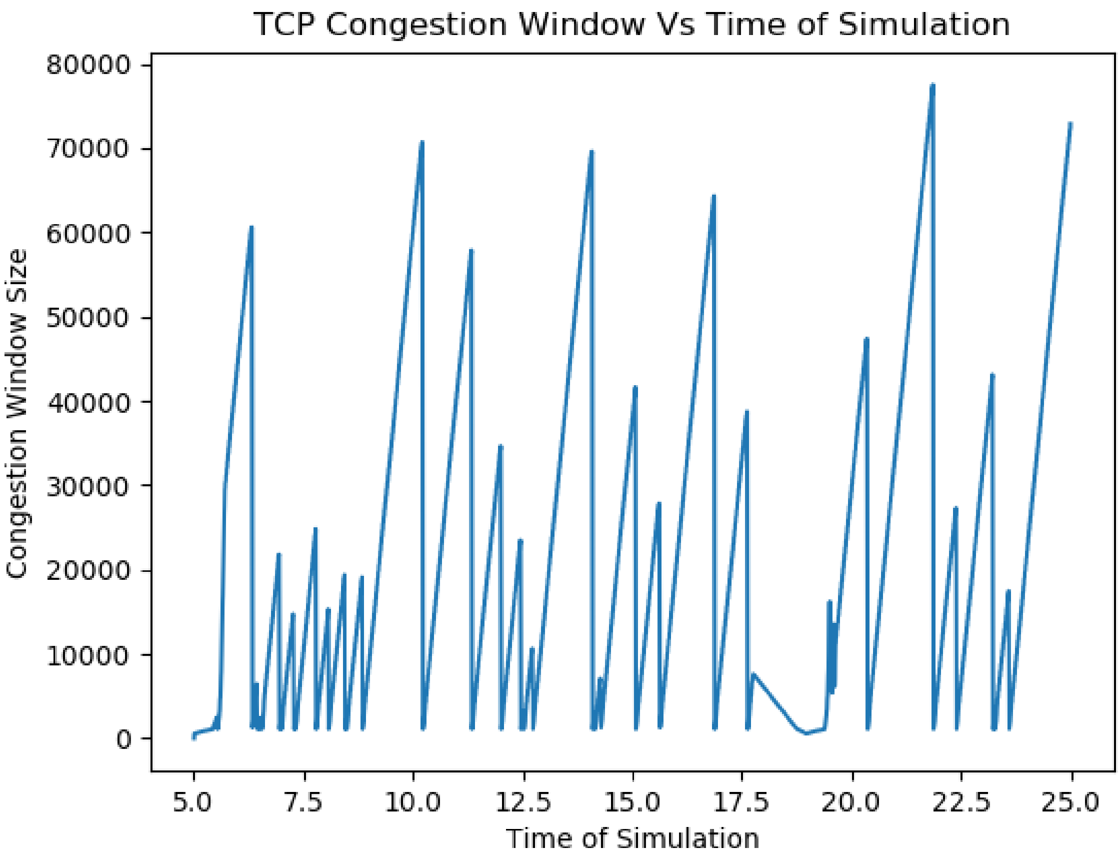


(b) TcpNewRenoPlus

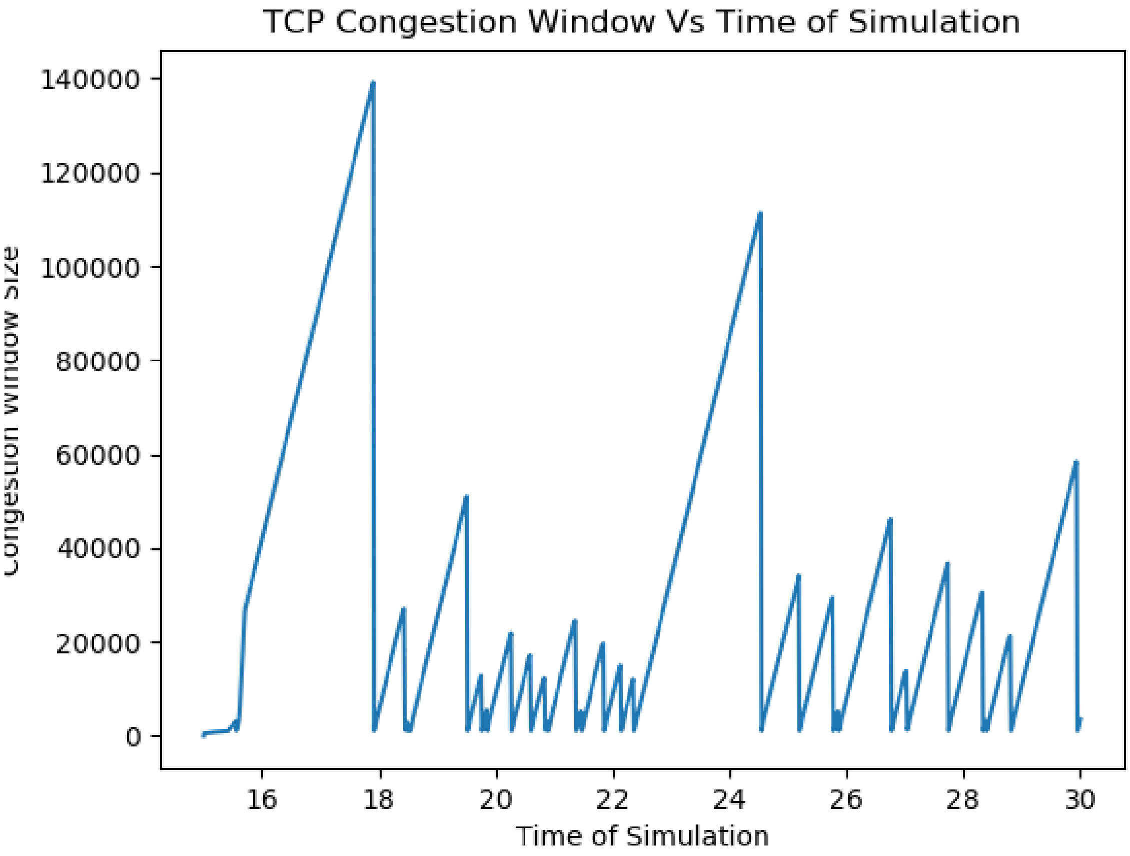
1. **Connection-1**
   For tcp-connection 1 source, we get the maximum window size as 105358. The corresponding plot is as follows:



2. **Connection-2**
   For tcp-connection 2 source, we get the maximum window size as 77512. The corresponding plot is as follows:



3. **Connection-3**
   For tcp-connection 3 source, we get the maximum window size as 139210. The corresponding plot is as follows:



Observations, Explanations and Impact on the Network:

Here our some of the observations that we can see based on the above graphs that are:

1. We can see that on an average the maximum congestion window size is greater in case of Tcp NewRenoPlus than Tcp NewReno and the explanation is that this happens because during the congestion avoidance phase in case of Tcp NewRenoPlus the congestion window keeps on increasing linearly whereas the Tcp NewReno increase rate is inversely dependent on the congestion window size hence we get a much larger value of congestion window size.
2. We can also observe that the rate of rise of congestion window size in slow start phase is also quite more in Tcp NewReno as compared to the Tcp NewRenoPlus and this happens because of increase rate varying inversely with the congestion window size in case of Tcp NewRenoPlus slow start phase.
3. We can see a large number of peaks in case of connection 1 and connection 2 which is a direct result of packet drops and thus both the slow start and congestion avoidance phases are achieved.