COL 334/672 Assignment 3

Deadline: 11:55 P.M 22 October 2021

In this assignment, we are going to use the NS-3 (www.nsnam.org) simulator, which is a discrete event, packet level network simulator for Internet systems.

Part 1: 35 points

In this part, we are going to analyze different congestion control protocols. The goal is to study how the congestion window at the sender varies with time for different protocols.

Create a simple topology of two nodes N1 and N2, and join them using a point-to-point link as shown in Figure 1. Create a TCP source at N1 and a TCP sink at N2. Fix the application data rate to be **1 Mbps**. The data rate of the link between N1 and N2 is **8 Mbps** and propagation delay is **3ms**. At N2 use RateErrorModel as the error model with error rate as **0.00001**. Packet size is **3000 bytes**.

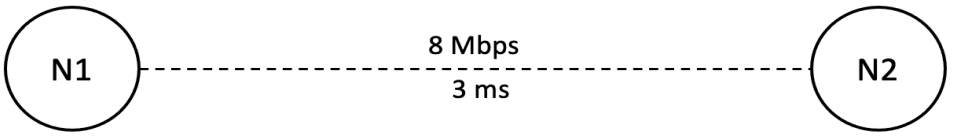

Figure 1

Connection will start at **t=1s** and end at **t=30s**. In our study, we will analyse the following protocols.

a. Newreno
b. Highspeed
c. Veno
d. Vegas

(Note: You are not required to code these protocols from scratch, use the inbuilt libraries present in ns-3). Read the usage section from this link on how to change congestion control protocol in ns3.

1. For each protocol, generate a plot having Congestion window size on the y-axis and time on the x-axis(till t=30s).
2. For each protocol, find the number of dropped packets in total. What is your inference?
3. For each protocol, describe what you observed (4-5 sentences per protocol is enough). You can talk about the trend you observed in the above plot, the algorithms they used for different phases etc.

Part 2: 15 points

In this part, we are going to analyze the effect of the bandwidth/application data rate on the congestion window at the sender. We'll use the default **TcpNewReno** as the congestion protocol.

Create a simple topology of two nodes N1 and N2, and join them using a point-to-point link as shown in Figure 2. Create a TCP source at N1 and a TCP sink at N2. The channel delay is **3ms**. At N2 use RateErrorModel as the error model with error rate as **0.00001**. The connection will start at **t=1s** and ends at **t=30s.** Packet size is **3000 bytes**.

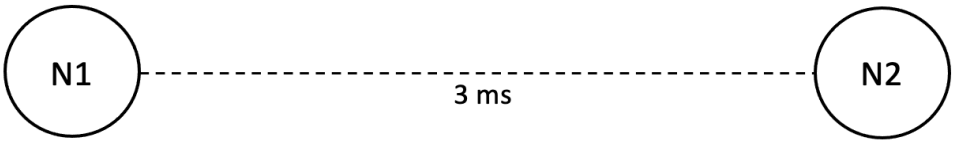

Figure 2

a) Plot the congestion window size vs time graph for the TCP connection at different Channel Data Rates (***2Mbps, 4Mbps, 10 Mbps, 20Mbps, 50 Mbps***) between N1 and N2. Use **Application data rate as 2Mbps**. You need to create a plot for each Channel Data Rate. Explain the trends that you observe.

b) Plot the congestion window size vs time graph for the TCP connection at different Application Data Rates (***0.5 Mbps, 1Mbps, 2Mbps, 4Mbps, 10 Mbps***). Use Channel data rate between N1 and N2 as **6 Mbps**. You need to create a plot for each Application data rate. Explain the trends that you observe.

Part 3: 50 points

In this part, we will learn how to create our own congestion control protocol.

NS-3 by default uses TcpNewReno as the congestion control strategy (Link). It consists of the following functions of interest to us

1. **TcpNewReno::SlowStart:**
   "ns-allinone-3.29/ns-3.29/src/internet/model/tcp-congestion-ops.cc"

2. **TcpNewReno::CongestionAvoidance:**
   "ns-allinone-3.29/ns-3.29/src/internet/model/tcp-congestion-ops.cc"

Our goal will be to create a new congestion control algorithm called TcpNewRenoCSE. For this, you need to create 2 new files TcpNewRenoCSE.c and TcpNewRenoCSE.h in *"ns-allinone-3.29/ns-3.29/src/internet/model"*. You can refer to the following link on how to write a new congestion control algorithm in NS-3.

You need to reuse most of the code of TcpNewReno from *“ns-allinone-3.29/ns-3.29/src/internet/model/tcp-congestion-ops.cc”* and *“"ns-allinone-3.29/ns-3.29/src/internet/model/tcp-congestion-ops.h"* and the algorithmic changes need to be done only for the SlowStart and CongestionAvoidance functions in your TcpNewRenoCSE.c.

The way you implement it by using inheritance of base class TcpCongestionOps / TcpNewReno /or any other way is your decision. The important point is that all functionalities in TcpNewRenoCSE except the below two should be of TcpNewReno.

1. TcpNewReno::SlowStart
2. TcpNewReno::CongestionAvoidance

In **Slow Start phase of TcpNewRenoCSE**, increase the congestion window size$Cwnd$ as:

$$Cwnd = Cwnd + (SegmentSize)^{1.9}/Cwnd$$

In the **Congestion Avoidance** phase of TcpNewRenoCSE, update the congestion window as

$$Cwnd = Cwnd + 0.5 * SegmentSize$$

In order to understand the behavior of the network using the newly designed congestion control protocol, we will use the topology described in Figure 3 below.

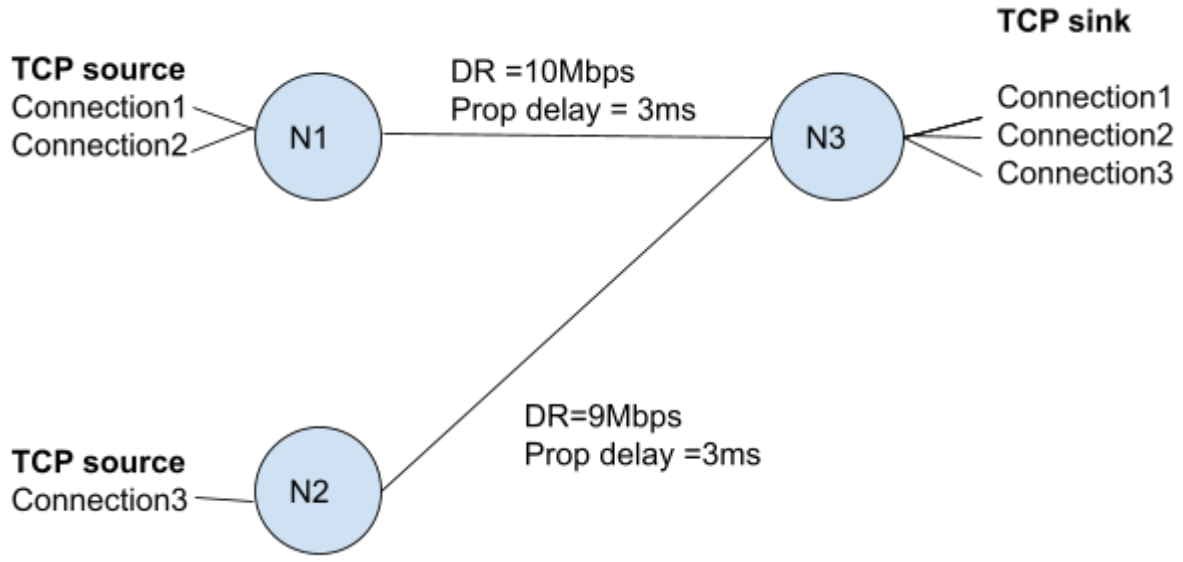

Figure 3

In Figure.3, N1-N2 and N1-N3 are connected via a point-to-point link. DR means Data Rate of the channel. Here we have TCP source at N1 and N2 and TCP sink at N3.

We will use the below 3 configurations of the network described in Fig.3 to analyze the behavior of congestion control algorithms:

- **Configuration1:** All senders use *TCPNewReno*.
- **Configuration2:** Connection3 at TCP source uses the new congestion protocol *TCPNewRenoCSE*. The remaining senders use the default protocol *TCPNewReno*.
- **Configuration3:** All senders use *TCPNewRenoCSE*.

For all three configurations above, the following hold:

i) Application data rate: 1.5 Mbps

ii) Connection 1 starts at time 1sec, Connection 2 starts at time 5sec, Connection 3 starts at time 15sec.

iii)Connection 1 ends at time 20sec, Connection 2 ends at time 25sec, Connection 3 ends at time 30sec.

iv) Packet size is 3000 bytes.

You need to answer the following questions:
For each **configuration** do the following at **each TCP source.**

1. Plot Congestion window size vs time (from t=1 to t=30 seconds)
2. Analyze the number of dropped packets for each connection separately.
3. How does the congestion avoidance phase vary on the same sender when using *TCPNewRenoCSE* vs *TCPNewReno*? Explain the observed trends. How does it impact the entire network?

What to Submit

1. For each part, write your answers, observations, plots and hypotheses in a single report file.
2. Submission folder should be a single .zip or .tar.gz file containing a report file and code for each question. Follow the directory structure as given below.

EnrollmentNumber/-
Report.pdf
Q1/-
First.cc (In case more than 1 file then make it First_1.cc, First_2.cc ...)
Q2/-

Second.cc (In case more than 1 file then make it
Second_1.cc,Second_2.cc ...))

Q3/-

Third.cc (In case more than 1 file then make it
Third_1.cc,Third_.cc ....))

Congestion/-

TcpNewRenoCSE.c
TcpNewRenoCSE.h

In case you are using any other helper code/libraries to generate plots, submit that code for generating plots.

3. The plots can be generated either using the .cc files or any other tool you wish to choose such as Python. But make sure, that plot is generated just using commands and not by copy pasting data values manually in excel or something like that during the demo. (**Tip** : Save the output values in a separate file and write a script to process that file and generate plots).
4. For each code file/package/libraries you submitted in the assignment, clearly describe their purpose in the report (1 to 2 sentences are enough for each).