Assignment: Understanding TCP Congestion Control Algorithms using NS-3

Deadline: 27th October 2024, 11:59 PM

Goal:

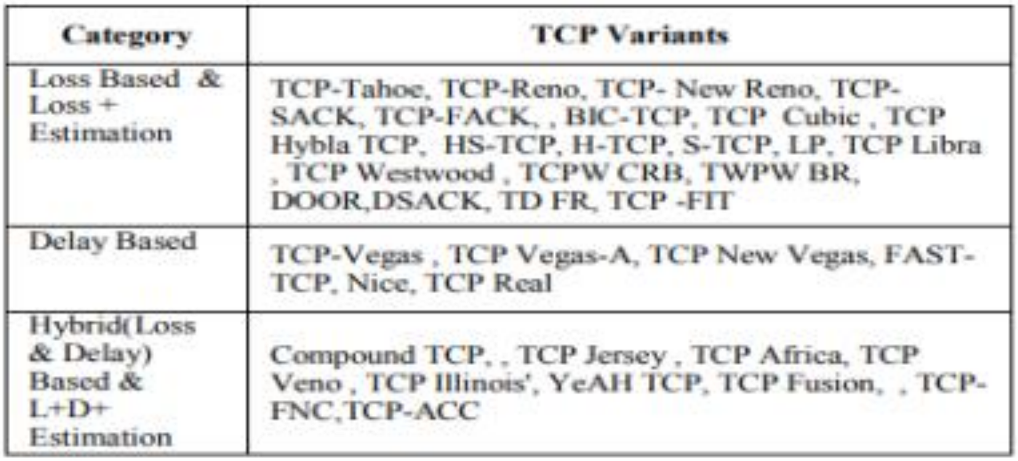
Study and understand different TCP variants using NS-3 and by reading research articles and observing how different algorithms react to network congestion while also trying to implement a new congestion control algorithm from scratch and evaluating it further for different metrics and fairness.

Part 1: Study of TCP Congestion Control Algorithms [20 Points]

Explain and compare any four TCP congestion control algorithms (TCP Variants).

Out of four TCP congestion control algorithms, one must be TCP Cubic, and at least one algorithm should be from loss-based, delay-based, and hybrid categories. Definitions of these categories are given below, and some of the TCP variants are shown in the table.

1. Loss-based TCP variants use packet loss as an indication of Congestion.
2. Delay-based TCP variants consider packet delay rather than packet loss as congestion in the network.
3. In the hybrid type of TCP variant, both packet loss and delay of the packet are considered as congestion in the network.



Category	TCP Variants
Loss Based & Loss + Estimation	TCP-Tahoe, TCP-Reno, TCP- New Reno, TCP-SACK, TCP-FACK, , BIC-TCP, TCP Cubic , TCP Hybla TCP, HS-TCP, H-TCP, S-TCP, LP, TCP Libra , TCP Westwood , TCPW CRB, TWPW BR, DOOR,DSACK, TD FR, TCP -FIT
Delay Based	TCP-Vegas , TCP Vegas-A, TCP New Vegas, FAST-TCP, Nice, TCP Real
Hybrid(Loss & Delay) Based & L+D+ Estimation	Compound TCP, , TCP Jersey , TCP Africa, TCP Veno , TCP Illinois', YeAH TCP, TCP Fusion, , TCP-FNC,TCP-ACC

Note: Refer to research papers for different algorithms. The report should have the following points.

- ➔ Algorithm Details [**6 points**]
- ➔ Suitable for which scenarios and where it fails [**7 points**]
- ➔ Compare all four variants in the end and write inferences [**7 points**]

Note for Part 1: Do not copy and paste from the paper; understand the algorithms. This assignment will be evaluated through a presentation by students. TAs can randomly pick some students and ask them to present the submitted report for evaluation.

Part 2: Understanding TCP Congestion Window using NS-3 [32 Points]

PART A [10 points]

For each of the given congestion control algorithms, perform the simulations and answer the following questions. (Demo_2.cc file is provided)

a. Newreno

b. Highspeed

c. Veno

d. Vegas

Q1. Plot the cwnd vs time graph, and describe what you observed, like slow start and congestion avoidance, in detail. **[2 points]**

Q2. Find the average throughput for each of the congestion control algorithms using tshark from the pcap files generated, and state which algorithm performed the best. **[2 points]**

Q3. How many times did the TCP algo reduce the cwnd and why? **[2 points]**

Q4. Check the effect of changing the bandwidth and latency of point-to-point connection and explain its effect on average throughput. **[2 points]**

Q5. Explain in short what is the effect of changing the default MTU size. **[1 point]**

Q6. Plot the rtt vs time graph and explain your inferences and observations. **[1 point]**

PART B [22 points]

In this part, you need to implement a new TCP Congestion Control Algorithm (CCA) and evaluate its performance and fairness (you can use Demo_2.cc and demo.cc for your reference).

a. You need to create a new CCA named Tcp<first_name> and you need to try to improve TcpNewReno's performance. **[1 point]**

b. You need to focus only on the SlowStart phase and in order to improve it you need to replace Reno's slow start with TCP Hystart (whose implementation can be found in src/internet/model/tcp-cubic.cc which is ns-3's Tcp Cubic implementation). **[6 points]**

c. You aim and try to make the slow start phase more intelligent with better exit points by doing this (more about Hystart at Reference a) but once you have done this you need to evaluate your algorithm's performance, so consider the dumbbell topology in demo.cc, make all the flows TCP and evaluate the standard metrics like throughput, congestion window, ssthresh, rtt variation between your algo and NewReno. **[4 points]**

d. Also, calculate the value of Jain's fairness index (more about this at Reference b) by varying the number of flows as 4,8,16,20. So in the first run of your experiment, all the flows will be using TcpNewReno so that becomes your baseline readings for comparison and in the second run of experiment, half of the flows will be using TcpNewReno while the other half of the flows will be using your new CCA. Create a table like shown below and populate it with the fairness values achieved in each case. **[5 points]**

Number of flows	4	8	16	20
NewReno & Your_CCA	<fairness_index>	<fairness_index>	<fairness_index>	<fairness_index>
NewReno	<fairness_index>	<fairness_index>	<fairness_index>	<fairness_index>

e. Plot the necessary graphs for all the metrics (throughput, congestion window, ssthresh, rtt) collected. **[3 points]**

f. Finally, try to justify and explain the results of the comparison thus obtained. **[3 points]**

References:

- Sangtae Ha and Injong Rhee. 2011. Taming the elephants: New TCP slow start. Comput. Netw. 55, 9 (June, 2011), 2092–2110. https://doi.org/10.1016/j.comnet.2011.01.014
- Jain's Fairness Index (https://doi.org/10.48550/arXiv.cs/9809099)

**** ns-3 version should be >=3.36**

Deliverables in a tar ball on GC:

- **Submission Guidelines**: For each part, create a separate folder. Upload all the folders and Assignment Report in GC as a tarball with the file name as <your roll no>_<your name>.tar
- Readable Report [**3 Marks for report quality**] enumerating steps followed with screenshots for each of the important steps.
    - Put the screenshots in the report for better clarity.
    - Your report should **necessarily** have screenshots of your final results, and you are expected to explain in the report which function was used, or which line of code set the particular socket to use a specific CCA or how the congestion window was traced so everything needs to be carefully explained in your report.
    - Explain everything that has been asked in your report highlight the important parts, and attach every codefile, script or anything that you used to complete the tasks.

- For all the plots, write the inferences that you have observed.

Check Web sources for additional information.