

CSC/ECE 573

Spring 2023

Homework #2

Instructions

- You must do this assignment in groups as assigned.
- Submit your final codes and other documents (mentioned later in the description) via Moodle.
- The due date and time for this assignment: March 19, 2023, 11:45pm ET.

TCP Analysis (60 Points: 50 Implementation and Experiment, 10 Report)

Objective:

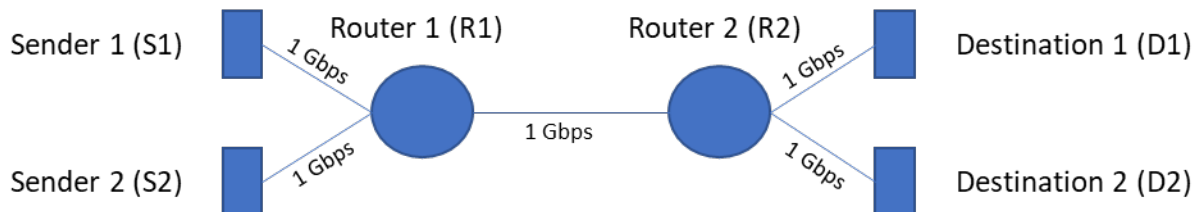
This project aims at giving students an understanding of the behaviors of two different TCP protocols when congestion occurs in network. These two TCP protocols are Cubic and DCTCP. Student will learn how each protocol behaves individually in the network and how both coexist with each other. The performance of these protocols will be evaluated in a simple network topology. At the end of this projects, students will have basic understanding of how performance of these two TCP protocols changes in different network condition and they will also get hands-on experience on ns-3.

Description:

In this project, you will assess the performance of TCP Cubic and DCTCP in terms of throughput and average flow completion time. You will implement a dumbbell topology in the ns-3 simulator and run multiple experiments to evaluate the performance of both TCP versions for the given metrics. You will demonstrate how well these two protocols coexist with each other and which one performs better in terms of responding to the congestion in the network. The simulator allowed in this project is ns-3. Details of network topology, measurement metrics, experiments, and deliverables are explained below.

Topology: [5 points]

You are required to set up a simulation of dumbbell topology as shown in the figure below:



There are two sender servers S1 and S2 which are connected to router R1 with 1 Gbps link each. There are two destination servers D1 and D2 which are connected to router R2 with 1 Gbps link each. Both R1 and R2 are connected with 1 Gbps link as well. Please note that for the given simple topology, the term router and switch can be treated as same.

- **Point distribution**
 - **Correct topology setup [5 points].**
 - **Any incorrect parameter [-5 points].**

Configuration: [5 points]

- S1 will send traffic to D1 only
- S2 will send traffic to D2 only
- All the servers i.e., S1, S2, D1, and D2 should be configured to support TCP Cubic (also known as TCP BIC in ns-3) and DCTCP. This parameter can be set according to the requirements of the experiments as described in section 2.6.

- **Point distribution**
 - **Correct configuration setup [5 points].**
 - **Any incorrect configuration [-5 points].**

Measurement Metrics:

There are two metrics that you will study:

1. **Throughput** tells how much data is transferred per unit time from source to destination over a given period of time. Its unit is bits per second (bps).
2. **Average Flow Completion Time** measures the time taken by the sender to send the desired amount of data to the receiver. It is measured in seconds.

The methodology of measuring these metrics is open-ended. Develop your way to measure them as accurately as possible. However, for both metrics, take the results of three runs for each experiment and then calculate the average and standard deviation for each experiment. More details on experiments are provided next.

Experiments: [25 points]

Once the topology is set and configurations are done, run the following experiments and measure *throughput* and *average flow completion time*. In each experiment, use bulk sender application in ns-3 to generate the traffic. Use bulk sender to send 50 MB of data from source to destination in each of the following experiments:

- Exp-1:** S1 sends traffic to D1 using TCP Cubic. S2 and D2 are not used in this experiment.
- Exp-2:** S1 sends traffic to D1 and S2 sends traffic to D2. Both senders will use TCP Cubic and start sending data to respective destinations simultaneously.
- Exp-3:** S1 sends traffic to D1 using DCTCP. S2 and D2 are not used in this experiment.
- Exp-4:** S1 sends traffic to D1 and S2 sends traffic to D2. Both senders will use DCTCP and start sending data to respective destinations simultaneously.
- Exp-5:** S1 sends traffic to D1 using TCP Cubic whereas S2 sends traffic to D2 using DCTCP. Both senders will start sending data to respective destinations simultaneously.

Run each experiment three times and then calculate average and standard deviation for the two metrics (i.e., throughput and average flow completion time).

Note: Implement your code such that it executes all of the above five experiments one after the other and generates the results in a .csv file as specified in the next section titled "Output Format". This is mandatory because we are going to test your code and it must generate results for all the experiments that you have implemented.

- **Point distribution**

- Each experiment has 5 points. Conducting all the experiment will give you [25 points].
- Deduction for any experiment not performed [-5 points].
- If the 'Note' given above is not followed [-15 points].
- Failure to compile code due to any issue [-20 points].
- Error at the time of execution, during runtime, or exit [-15 points].
- Exception at the time of execution, during runtime, or exit [-15 points].

Output Format: [15 marks]

You will also submit the results from your execution in a csv file. The title of csv file *must* be tcp_<unityid> e.g., tcp_mshahza.csv. The format of the file *must* be as follows:

```
exp,r1_s1,r2_s1,r3_s1,avg_s1,std_s1,unit_s1,r1_s2,r2_s2,r3_s2,avg_s2,std_s2,unit_s2,
th_1,520.12,547.85,690.98,586.32,91.70,Mbps,,,,,
th_2,<,<,<,<,<,<,<,<,<,<,<,<,<,<
th_3,<,<,<,<,<,<,<,<,<,<,<,<,<,<
th_4,<,<,<,<,<,<,<,<,<,<,<,<,<,<
th_5,<,<,<,<,<,<,<,<,<,<,<,<,<,<
afct_1,20,17,26,21,4.58,sec,,,,,
afct_2,<,<,<,<,<,<,<,<,<,<,<,<,<,<
afct_3,<,<,<,<,<,<,<,<,<,<,<,<,<,<
afct_4,<,<,<,<,<,<,<,<,<,<,<,<,<,<
afct_5,<,<,<,<,<,<,<,<,<,<,<,<,<,<
```

- ✓ Column 'exp' represents 5 experiments. Rows th_# implies data for five throughput experiments. Row afct_# implies data for average flow completion time for five experiments.
- ✓ Columns 'r1_s1', 'r2_s1', 'r3_s1', 'avg_s1', 'std_s1', 'unit_s1' represent the 3 runs, average, standard deviation, and unit for each of the experiments involving sender S1. These columns will be filled for all the five experiments because S1 is part of all the experiments.
- ✓ Columns 'r1_s2', 'r2_s2', 'r3_s2', 'avg_s2', 'std_s2', 'unit_s2' represent the 3 runs, average, standard deviation, and unit for each of the experiments involving sender S2. These columns will be only filled for Exp-2, Exp-4, and Exp-5 because sender S2 is only used in these experiments. For experiment Exp-1 and Exp-3, keep these columns blank.
- ✓ For columns r1_, r2_, r3_, avg_, and std_, replace <> with the numerical values obtained from the experiments. Do not write unit in these columns.
- ✓ In column unit_, replace <> with appropriate units. For throughput use Gbps, Mbps, Kbps, or bps. For average flow completion time, use millisec or sec.
- ✓ There are no spaces before and after the commas.
- ✓ For example, two rows th_1 and afct_1 are filled to show how to present your data. The values shown are hypothetical. Your values will vary. Rows th_1 and afct_1 are for 'Exp 1' which only includes sender S1. Therefore, values for sender S2 will be blank.

- **Points distribution**

- Correct file generated according to the given format [15 points].
- No file generation after execution of code [- 30 points].
- File generated but it is empty [-25 points].
- File generated but has incorrect format [-15 points].

Submission Instruction:

We will test your code for successful compilation, execution, termination, and csv file. Your results will be compared with our implementation for correctness.

Your program will be tested on Ubuntu 16.04 LTS operating system. It is highly recommended to use Ubuntu 16.XX or 18.XX for this project. If you are using Windows and MAC machines, then install VMware workstation or virtual box on your machine and setup a virtual machine (VM) for Ubuntu 16.04.

[VCL](#) is not recommended for this project as this will require you to install ns-3 every time you make a reservation for a VM. This is time consuming process. Therefore, it is better to setup environment on your local machine.

For submission, students must make a folder named **HW2** and place the following three files in it:

1. Code file named as tcp_<unityid>.cc
2. Csv file named as tcp_<unityid>.csv
3. Readme file as readme_<unity_id>.txt.
 - Readme file should contain clear instructions about the following:
 - How to compile the code? Provide command.
 - How to execute the compiled code? Provide command.
 - It should contain any extra information regarding your implementation such as delay used for point-to-point links etc.
 - Which packages and dependencies are required by your code for the successful compilation and execution of the code? Provide a list of all those packages and dependencies. Failure to provide this may lead to unsuccessful compilation and execution of your code due to which your points will be deducted as per the criteria mentioned in section 2.6.

Once you have placed the three files in folder HW2, go to the section below about final submission instructions.

Documentation and Help:

ns-3:

- Installation Guide
 - <https://www.nsnam.org/wiki/Installation>
- Documentation
 - <https://www.nsnam.org/documentation/>
- Code Search and details
 - <https://www.nsnam.org/doxygen/>
- For a quick start, after installation, look for examples in folder *examples/tutorials/* and *example/tcp/*
- You can search for quick setup tutorials on YouTube as well.

Report (10 points)

The template for the report is provided in the attached file. Please follow the instructions below while writing the report:

1. The report must be to-the-point. There is no need of long details.
2. Only follow the format of the template and fill in the required data.
3. There are 10 points for each project in the report.
4. Report must be submitted in .pdf format. Name the report as <unityid>.pdf.
5. Once the report is complete, go to section 4 below for final submission instructions.

Final Submission

Once you have created the folder HW2 and added the respective files in it, zip it along with the report into a single .zip file. Name that .zip file as <unityid>.zip and submit it on Moodle. Only one member of a group should submit. Your submitted .zip file contains following:

- ✓ HW2 folder
- ✓ <unityid>.pdf

Good Luck!