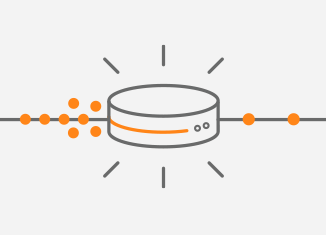
**CS 4713**

**INTRODUCTION TO THE INTERNET: ARCHITECTURE AND PROTOCOLS**



**ASSIGNMENT 3**

**CONGESTION CONTROL OVER CELLULAR NETWORKS**

**The assignment has been adapted from PA1 of Stanford’s CS244**

**Deadline: Friday, 7th December 2018 at 11:50pm**

**Note:** You can work on this assignment with a partner who is also in the class, and submit the assignment in pairs. In case you submit the assignment as a pair, please make sure when you submit code files and technical report, names and roll numbers of both the students are written. Please note both the students should be able to explain any program code or report they submit.

**Note:** You may turn the assignment in up to two days late, and each day will incur a 10% deduction from your score.

**Course policy about plagiarism is as follows:**

* Students must not share any scripts or code with other students.
* Students must be prepared to explain any program code or report they submit.
* Students must indicate with their submission any assistance received.
* All submissions are subject to plagiarism detection.
* Students are strongly advised that any act of plagiarism will be reported to the Disciplinary Committee.

**INTRODUCTION**

In this assignment, you will compete to build a congestion-control algorithm to maximize "power" (throughput / delay) on an emulated cellular network.

You will start with a [simple UDP sender and receiver](https://github.com/keithw/sourdough/tree/master/datagrump), built on C++ networking classes that do most of the grunt work of opening sockets and reading and writing datagrams.

By modifying the behavior of the congestion controller (implemented in [controller.cc](https://github.com/keithw/sourdough/blob/master/datagrump/controller.cc)), you can control how the sender decides when to send datagrams. The more datagrams it sends, the more your throughput will be, at the risk of building up an in-network standing queue and resulting delay.

**SETUP**

You have been provided a VM with the assignment and relevant dependencies all setup.

Boot the VM and perform the following test run.

**Username:** osboxes.org

**Password:** osboxes.org

cd sourdough

./autogen.sh && ./configure && make (compile the contest code)

sudo sysctl -w net.ipv4.ip\_forward=1 (enable IP forwarding for Mahimahi to work)

cd datagrump

./run-contest [username]

This command will run the congestion-controller in controller.cc for about two minutes over an emulated Verizon LTE (4G) connection. While it runs, you will see an animation of the network's capacity, the sending rate of the algorithm, and the in-network queuing delay. The animation will run for ~140 seconds, after which the results will be displayed. After the run completes, it will upload your results to the contest website.

You should use [rollnumber1\_rollnumber2] as your username when running the test script.

You shouldn’t need to edit files other than controller.hh and controller.cc. However, you are free to modify any file in the datagrump directory EXCEPT the run-contest script.

**TASKS**

Please write up your answers to the below exercises in a report, to be submitted with any program files you edit:

* Task 1 [25 marks]: Vary the fixed window size by editing controller.cc to see what happens. Make a 2D graph of throughput vs. 95-percentile signal delay (similar to what is seen on the contest analysis URLs) as you vary this value. What is the best single window size that you can find to maximize the overall "score" (log throughput/delay)? How repeatable are the measurements taken with the same window size over multiple runs?
* Task 2 [50 marks]: Implement a simple [AIMD scheme](http://en.wikipedia.org/wiki/Additive_increase/multiplicative_decrease), similar to TCP's congestion-avoidance phase. How well does this work? What constants did you choose?
* Task 3 [25 marks]: **The Contest**. Try different approaches and work to maximize your score on the final evaluation. Be wary about "overtraining": after the contest is over, we will collect new network traces and then run everybody's entries over the newly-collected evaluation trace. In your report, please explain your approach, including the important decisions you had to make and how you made them. Include illustrative plots.

**BONUS**

The top three pairs to score in the contest will get a bonus worth 25 marks. These can be used to make up for marks lost in the three assignments. **Good Luck!**