CS:4980 Advanced Computer Networks

Programming Project # 2

Due Date: Tuesday December 1, 2015 at noon in ICON Dropbox.

Submission Instructions: Submit all required files in a single .zip file.

Late submissions are not allowed unless you have prior permission to do so.

You are not allowed to copy code/data from the Internet. You cannot discuss your code/data with each other (except your project partner).

You are asked to implement a reliable data delivery service with congestion control over the unreliable UDP transport using socket API. This project can be completed in Python, C/C++, or Java. You will implement the following variations of this service.

Question 1. Implement an application-layer reliable data delivery service over UDP sockets using *sequence numbers*, *acknowledgements*, *timeouts* (fixed), and *retransmissions*. Implement a fixed-sized (5 packets) sliding window. Set packet size at 1000 bytes.

[10 points]

Question 2. Further implement a dynamic sliding window at the sender. Assume that the receive window (rwnd) at the receiver is sufficiently large. Thus, the sliding window size is determined by cwnd at the sender. Implement *slow start* and *congestion avoidance* from TCP Tahoe. Start with the initial congestion window of 1 packet. Also implement dynamic timeouts using the RTT estimation method discussed in the textbook.

[10 points]

What to analyze?

- Compute and report per-packet delay (time of acknowledgement arrival - time of packet dispatch) in a text file. Plot the timeseries of packet delay.

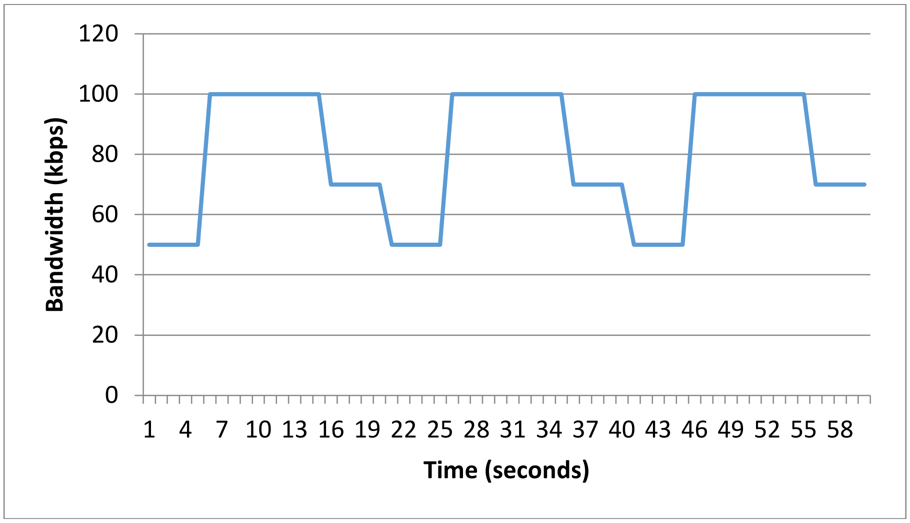
- Compute and report instantaneous throughput every 1 second. Recall that throughput is number of bytes successfully acknowledged by the receiver.

Test your client and server by transferring a large file (https://www.dropbox.com/s/diwh8ltre7dx6ms/John%20Oliver%20-%20Quotations.mp4) from server to client for varying packet loss and delay. You should use netem to introduce link delays and losses.

A) Plot timeseries of per-packet delay and timeseries of throughput for varying link delay (10 ms, 50 ms, 100 ms) and link loss (0%, 0.1%, 1%). Also report the following performance metric:

$\log_{10}$(average throughput in bits per second)−$\log_{10}$(average RTT in milliseconds)

B) Plot timeseries of per-packet delay and timeseries of throughput for the following dynamic link conditions (use train.sh, download at: https://www.dropbox.com/s/huqwk7vxm41mek4/train.sh). Run it on shell with sudo access: bash train.sh. It provides the following bandwidth profile.



Question 3**.** Propose and implement your own congestion control with the aim of outperforming TCP Tahoe. Briefly describe your idea using suitable text and figures. Plot timeseries of per-packet delay and timeseries of throughput for train.sh bandwidth profile. Also report the following performance metric:

$\log_{10}$(average throughput in bits per second)−$\log_{10}$(average RTT in milliseconds)

If you outperform TCP Tahoe in terms of this metric on train.sh bandwidth profile, you will be awarded 3 extra credit points. If you also outperform TCP Tahoe on test.sh bandwidth profile (kept secret), you will be awarded additional 3 extra credit points.

If your proposed congestion control outperforms TCP Tahoe and other student submissions on test.sh bandwidth profile, you will be awarded A+ in the course and you won't have to bother about the final exam :)

[Extra Credit]

What to submit?

1. Submit client and server **source codes** and/or **binaries** for each of the 3 implementations. Include a **readme file**. We should be able to run your code/binaries from the command line. Your code will be tested on Ubuntu 14.04.3 LTS.

2. Submit a **detailed report** (in PDF format) showing sample outputs and plots.

3. Include the following **signed plagiarism statement** as a scanned PDF. Your submission will be rejected without the inclusion of this signed statement.

Good luck!

I certify that all work submitted in this document is my own work. I have completed all of the assignments on my own without assistance from others except as indicated by appropriate citation. I have read and understand the university policy on plagiarism and academic dishonesty. I further understand that official sanctions will be imposed if there is any evidence of academic dishonesty in this work. Please provide your identity to certify that the above statements are true.

Group Member 1:

Name (Printed) Signature Date

____

Group Member 2:

Name (Printed) Signature Date

____