

CSCI 6760 HWP 3: Reliable communications over unreliable channels

Assigned: Thursday, Mar 7

Due: Tuesday, Apr 9, 2:00 PM

(LATE SUBMISSIONS WILL NOT BE ACCEPTED)

Over the course of the semester, we spent considerable time developing network technologies that provide reliable communication channels (TCP) over unreliable channels (IP). For this home work project, we will build such a reliable communication mechanism over an unreliable UDP based stream (along with dummynet interface to control and tweak the network parameters).

You will develop an reliable network mechanism that transfers large files (e.g. 25 MB). You have to deal with cross-flows (from other UDP and TCP traffic). You will plot the network throughput, the time taken to transfer the file and the amount of data transferred (both the forward direction and acknowledgment traffic). You will use dummynet to simulate interesting network scenarios.

The grading will be done as follows: If you implement basic ARQ functionality, then you will receive 60% points. Every additional feature that you implement will get you 10% more points. You could receive more than 100% points for more complex (and correct) implementations. Untested (but potentially correct implementations) will not receive any credit. At the end, we will compare your results with other implementations by your colleagues. The best implementation (as measured by the overall throughput and amount of acknowledgment traffic) will receive an extra credit (of 10% points). Specifically, we will:

1. Develop a program that will reliably (in-order, delivered once semantics) transfer 25 MB of data from one machine to another machine (within the class network cluster). You will use a maximum packet size of 100 bytes. You will use UDP as the underlying network mechanism and implement TCP functionality. You need not implement the TCP connection open (SYN) and connection shutdown (FIN) mechanisms. Your implementation should adaptively measure the RTT. Implement some/all of the following techniques:
 - Slow start
 - Fast recovery
 - SACK
 - DSACK
 - delayed ACK
 - other techniques used in TCP or discussed in class

Clearly identify the features that you had implemented to receive proper credit.

The next few features are necessary for the proper testing and simulation setup of your program and as such carry no specific grade:

2. Develop a UDP program that will continuously use half the available bandwidth (for example, for a 1 Mbps link, send 625 - 100 byte packets per second ($625 * 100 * 8 = 500k\text{bps}$)). You will run this program simultaneously to provide the cross flows.
3. Use dummynet to simulate the following network conditions between the source and destination (full duplex settings; 8 combinations):
 - (a) 100 kbps and 1 Mbps.

- (b) delays of 100 msec and 1 sec.
 - (c) packet loss of 1% and 10%.
4. Run two instances of your program along with the UDP program and measure the instantaneous throughput and plot the throughput measured with time (in 100 msec granularity). You developed such a throughput measurement tool for your HWA #1. Also measure the amount of traffic on the wire (both directions, including retransmissions, ACK packets etc).

Submission

Please submit your project, along with a succinct report called **REPORT.txt** (plain text is fine) describing your approach, the merits of your approach and compilation instructions. You will turn in your complete project as a single tar file. On gemini, please use `/home/profs/surendar/bin/turnin NETWORKS HWP3 <your tar file>` to submit your assignment. You can submit your assignment multiple times. I will only use the latest submission. To see the files that you had submitted, try `turnin NETWORKS HWP3`. **Remember, I will randomly choose students who will be asked to explain their approach in person.**