

HMC CS125 LAB #3

DUE DATE: OCT. 9TH 2PM

ANDREW GOODNEY - AGOODNEY@HMC.EDU

1. GROUP PROJECT: FAST, RELIABLE FILE TRANSFER

This project will build on the last assignment. In it you will explore the real-world implications of how TCP performs under less than perfect conditions. The project is in two parts, first, a short exercise using `scp`. Second, you will develop a file transfer program. I suggest doing the following experiments using the FBSD9-64-STD (FreeBSD 9.x) operating system, however you may like to do the program development on a Linux image.

1.1. `scp`. The ‘secure copy program’ `scp` is a standard tool on modern UNIX-like machines. It is used to copy files between machines, securely and reliably. However, as we will see, it does not always provide good throughput.

- Create an experiment in DETER with two computers. Link the computers together with a 100Mb/s link. Be sure to include some initial delay in the link so DETER will allocate you a delay node.
- Once the experiment is started, use the DETER website to set the delay on the link to zero. What is the RTT delay between the nodes now? What is the bandwidth delay product?
- On each node create and mount a temporary local file system:
 `> sudo mkextrafs /mnt`
- On one node, create a 200MB file in `/mnt`:
 `> cd /mnt`
 `> dd if=/dev/urandom of=data.bin bs=1m count=200`
- Explain briefly what the above commands do.
- Copy the file to the other node:
 `> scp data.bin USER@nodeB:/mnt` (where USER is your DETER username)
- What transfer throughput do you get?
- Explore the problem: use the DETER website to increase the delay on the link. What happens to the throughput? Can you improve the performance using the kernel parameters (i.e. default and maximum TCP window sizes)? Make sure to explore RTT delays at least up to 100ms.
- Does `scp` seem to have some sort of built-in limitation? Can you guess what it is? Hint: `scp` uses the SSH protocol to transfer data.

- Now set your delay back to zero. At the same time add a small amount of packet-loss to the link. Start at 0.1% (i.e. 1 packet dropped in 1000) and test at various loss levels up to 5%. Graph throughput vs. loss. Discuss these results. Does this seem extreme? Can you explain why this happens?
- Estimate (or measure) the delay and loss from Los Angeles to Switzerland. If you had a physicist for a colleague, and he wanted to download some data from the new atom smasher at CERN, would you expect him to come to you for help? Can you help him? If so, how?

Create a document with results and answers to all of the above questions.

1.2. File Transfer Utility. As we've learned above, TCP, while totally reliable and robust, doesn't always give us good throughput. In this section you'll design a IP based file-transfer utility. The design and implementation of the utility is up to your group, however it must full-fill only three requirements: it must use IP (so it can be routed), it must transfer the file reliably (with no errors) and it must be implemented with a command-line interface similar to `scp`.

The link speed between the sender and receiver must be **100Mbps** and the test file size must be at least **1GBytes**. You should emulate the delay and the loss rate of the link using the delay node. You should test your system under various different conditions. However two settings that you must expose your system for the assignment are:

- The Delay (RTT) of 10ms with the Loss rate of 1%
- The Delay (RTT) of 200ms with the Loss rate of 20%

Describe, in detail, the concept(s) behind your file transfer utility, results, and the analysis in the document that must be submitted on October 9th by 2pm. Also, on October 9th and 14th we will have presentations from each group. The presentations will give details on how you solved the problems, problems you encountered and the results you were able to obtain. Some hints:

- There are several closed and open source projects out there that do this. They typically use UDP at their base. You may use them as inspiration, however the end work must be your own. You must use UDP and only UDP for the transport of the file data. However, you may use TCP for control or metadata, but all of the file data, including retransmissions, must use UDP.
- You can implement the program in any language you'd like (C, C++, Java, Python, Ruby, etc.) as long as it works on the DETER nodes, and your submission comes with clear, concise instructions on how to build and test your program.
- Think about why `scp` has issues. Identify these weaknesses as inspiration for solving the problem. Think about selective re-transmission, parallel flows, forward error correction...
- Learn how to use the following UNIX tools, they are your friends: `tcpdump`, `tcpreplay`, `nc` (aka `netcat`), `nmap`, `netstat`, `iperf`
- There will be a prize for the team that achieves the highest throughput on the **200ms, 20%** test case!

- You may need to learn about network programming on UNIX using sockets, and/or `libpcap`.
See:
 <http://beej.us/guide/bgnet/>
 <http://www.prasannatech.net/2008/07/socket-programming-tutorial.html>
 <http://gnosis.cx/publish/programming/sockets2.html>
- To measure the throughput achieved, the `time` utility will be used. Familiarize yourself with this utility and understand how to use it to measure the total throughput achieved by your FTP utility.