CSCI 558L – Laboratory Assignment #4A: Fast, Reliable File Transfer (Group)

Instructor: Young H. Cho

T.A.: Siddharth Bhargav

Due date: September 15, 2012 at 11:55pm

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. This lab has two parts, first, a short exercise using scp. Second, you will develop a file transfer program. We suggest doing the following experiments using the FBSD8-STD (FreeBSD 8.x) operating system, however you may like to do the program development on a Linux image.

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 50ms.
- 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.

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%

For this part, your team must be divided into 2 subgroups, building two different methods of FTP to meet the minimum performance of 20Mbps transfer rate. The goal of this task is to encourage a healthy competitive development environment for everyone. You are encouraged to help each other within the team to get the best result. However, we suggest that each subgroup develop the core concepts independently to eventually merge the new ideas into a single implementation in Part B.

As a part of the assignment, each group must post the result of your system on the Laboratory 4 Results and Discussion forum every day. Your final result must be posted on September 15th by 11:55pm. You also must demonstrate your system to your T.A. during his office hour on Sept 17th.

Describe, in detail, the concept(s) behind your file transfer utility, results, and the analysis in the document that must be submitted on September 15th by 11:55pm. The scoring will be based on the submitted document, T.A.s verification of the system.

3. Tournament Script

In order to test your results from 2, you must create .ns file with 2 nodes acting as senders and 1 node as the receiver. The shared link between the senders and the receiver should be configured as 100 Mbps link with RTT of 200ms and loss rate of 20%.

Then a script or a program should be created to deploy two sets of FTP processes simultaneously. The program should be responsible in checking for the data integrity and fairness as well as measure performance of each independently. But most importantly, the program must fairly determine the winner of the competition. You will use this program to demo two FTP methods created by the team.

We will use the best script/program of all of the groups for Part B of the lab. If your script is chosen, you will receive extra credit for the lab.

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 are not restricted to UDP, you can use any IP transport protocol if you need to (except TCP). You may however use TCP for control or metadata, but all of the file data, including retransmissions, must use a protocol other than TCP.
- 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. (Note: I could implement this using a shell script and existing UNIX command line tools, so if you haven't done a lot of UNIX network programming, get creative.)
- 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
- 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