

## Networks — CSCI 125, Fall 2014

Instructors: Goodney

Lab 3

Due: 2:00PM on Thursday, October 9

**Name:** Bruce Yan (byan@hmc.edu)

**DETER ID:** hmc125ah, hmc125-ANGELA

**Worked with:** Angela Zhou

### Problem 1 - Group Project: Fast, Reliable File Transfer:

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.

- We created an experiment in DETER with two computers together with a 100Mb/s link and 50ms initial delay. Below is the .ns file we used.

---

```
# lab3-part1.ns

# This is a simple ns script. Comments start with #.
set ns [new Simulator]
source tb_compat.tcl

# Define the 4 nodes in the topology
set nodeA [$ns node]
set nodeB [$ns node]

# Define the link and the LAN that connect the nodes
# This sets the link speed between nodeA and nodeB
# There is a 25ms round-trip delay
set link0 [$ns duplex-link $nodeB $nodeA 100Mb 50ms DropTail]

# Set the OS to Ubuntu1204-64-STD (modern)
tb-set-node-os $nodeA FBSD9-64-STD
tb-set-node-os $nodeB FBSD9-64-STD

# Enable routing (static) between all the nodes
$ns rtproto Static

# Instruct the simulator to start
# Go!
$ns run
```

---

- a
- b
- c

### Problem 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.