

CSC 457/Assignment#0

Name - Phyo Thiha

Date - Jan 21, 2013

Description: Report of the experiments carried out for the assignment

Code location: under "/localdisk/phyo_net/Assignment0" at "f13.cs.rochester.edu"

Note: A good amount of comments were provided in the code to help TA alleviate some pain in grading. :)

To compile, please type "make" in the commandline.

To run, I've given examples commands in each of the sections below as well as in the header of the relevant code files.

Report:

=====

Task#1: (10 points) Measure the round-trip transmission delay of a TCP message between two machines. Provide measurement results for the following TCP message sizes: 1-byte, 10-byte, 100-byte, 1,000-byte, 10,000-byte. You should include the connection setup time in measured delays. For comparison purposes, all measurements should be done on Linux machines of the undergraduate lab or the graduate software lab.

We used "hrtimer_x86.c" for accurate measurement of time in reference to CPU clock of x86 systems. This gives us the time resolution as good as down to microsecond, but we'll report our results in terms of second (in contrast to popular unit microsecond used by Unix programs such as "ping").

"tcpclient.c" is the program that will establish connection with the server and send a determined amount of data in bytes. The code measures the time taken (round-trip time a.k.a RTT) including the connection setup time using the "hrtimer" mentioned above. "tcpserver.c" is the program that will listen on a socket of the host machine at a defined port. We ran "tcpserver.c" on "cycle1.csug.rochester.edu" <128.151.69.85> and ran the client on "f13.cs.rochester.edu" <128.151.67.113>. This will send a certain number of characters ('a' in this case) as determined by the commandline argument to "tcpclient.c" program to the server. When we ran "tcpclient.c", it'll send the data 200 times and calculate the average. However, this number itself fluctuates significantly from run to run, so we ran "tcpclient.c" program 10 times each for different message size, and calculate the overall averages. The results are as shown below:

On "cycle1.csug.rochester.edu":

```
$ ./tcpserver 8000
```

On "f13.cs.rochester.edu" (for sending 1 byte of data):

```
$ ./tcpclient 128.151.69.85 8000 1
```

Message Size(bytes) Round-trip Time(secs)

=====	=====
1	0.000651119
10	0.000700392
100	0.00168678
1000	0.0019259

10000

0.00283898

From the experiment data above, we can conclude that the RTT increase is positively correlated to the amount of data sent.

Task#2: (10 points) Establish TCP connections with Internet service sites (e.g., a search engine or an e-commerce site) and measure the round-trip transmission delays. Please report the results for a site in the United States and a foreign site. For the foreign site, you are encouraged to find one with longest delay possible. There is no need to use messages of different sizes (you cannot control the return message size anyway). To ease your task, it may not be necessary to establish a valid message exchange (i.e., a return error message is sufficient for measuring the round-trip delay).

We learned that the round-trip time (RTT) is positively correlated to the physical proximity of the locations between two IP addresses <Ref:

<http://www.caida.org/research/performance/rtt/walrus0202/>>. Therefore, we selected a few IPs which are located progressively farther from Rochester, NY: Virginia (US), California (US), Norway and Iraq. Here, we tried to measure the time between just sending and receiving the data (that is, between send() and recv() function calls) vs. that between creating socket and establishing the connection along with send and receive time (that is, starting from "socket(AF_INET, SOCK_STREAM, 0)" call in the code, "tcpclient.c").

We also tried two different number of trials for each of the variation above. The purpose is to see if RTT will faster or slower as the number of trials are increased from 20 to 200. It turns out that we cannot draw such conclusion with certain from the data obtained. In other words, the RTT for certain IPs seems to have decreased whereas some of them increased.

The correlation between physical proximity and RTT is obvious, especially when we include the connection time (see 1 b. under results section). For example, it takes longer to establish connection, and send/receive data between Rochester (NY) and Mountain View (CA) than it takes between Rochester (NY) and Chantilly (VA) for Google's main webpage. Similarly, RTT increased significantly if we send and receive data with foreign sites. We decided to try Norway and Iraq, which are ~3600 miles and ~6000 miles from Rochester respectively. Compared to RTT results from a site within the U.S. (e.g., Google), the sites from Norway and Iraqi University take longer to reach.

Note: We also did a bit of research on which countries have the slowest internet access and found a few resources. From there, we determined that Nepal and Burma could be good candidate for testing. However, there was significant delay (up to >300 seconds) and connection errors in trying to reach a couple of sites from those countries. Therefore, we decided to move on with sites from Norway and Iraq, which are foreign countries. The links about internet speeds are listed below:

http://www.huffingtonpost.com/2012/05/08/slowest-internet-countries_n_1475397.html

<http://www.cs.umd.edu/projects/hpsl/html-papers/latency-tr.html>

<http://carlos.bueno.org/2010/07/internet-cartography.html>

<http://www.theverge.com/2012/8/9/3230626/akamai-global-internet-speed>

<http://www.caida.org/research/performance/rtt/walrus0202/>

The detail and results for Task#2 are shown below:

IPs tried (closest from Rochester to farthest in physical location):

=====

74.125.224.96	<google.com> Virginia	Chantilly
173.194.32.33	<google.com> California	Mountain View
93.94.10.13	<norway.no> Norway	
188.39.92.163	<en.uobaghdad.edu.iq> Iraq	

Distance from Rochester to (base on Wolfram Alpha search engine):

=====

Virginia:	395 miles
California:	2355 miles
Norway:	3670 miles
Iraq:	6024 miles

Results

=====

1 a)

To run: `./tcpclient <ip> 80 1`

Note: Measurements are only between send() and recv() commands.

That is, creating socket and connection times are NOT included.

IP	Avg. Round-trip Time (secs)	# of Trial	DataSize(bytes)
---	-----	-----	-----
74.125.224.96	0.239404	20	1
	0.254737	200	1
173.194.32.33	0.41508	20	1
	0.312382	200	1
93.94.10.13	0.397753	20	1
	0.370038	200	1
188.39.92.163	0.307284	20	1
	0.322844	200	1

1 b)

To run: `./tcpclient <ip> 80 1`

Note: This time, the time taken for creating the socket and connection times are included.

IP	Avg. Round-trip Time (secs)	# of Trial	DataSize(bytes)
---	-----	-----	-----
74.125.224.96	0.469405	20	1
	0.375384	200	1
173.194.32.33	0.647941	20	1
	0.674597	200	1

93.94.10.13	0.86763	20	1
	0.742682	200	1
188.39.92.163	0.726933	20	1
	0.608844	200	1

Credit: Thank you, Amal, for helping me troubleshoot some kinks in my program.