CMSC 417 Computer Networks **Fall 2012**

Programming Assignment 3

Assigned: Oct 11th *Due: Oct 24th, 11:59:59 PM.*

1 Description

In this assignment, you will analyze network transactions using network packet traces that we have captured for you. Your goal is to identify TCP flow and congestion control behavior, find retransmissions, and estimate round trip times.

The network traces, you will mainly work on, were collected via downloading three files from our web server. The traces were captured from different vantage points that exhibit different network characteristics. For example, the traces were captured from an airplane, a Comcast residential connection, and Verizon and Sprint cellular networks. You will also work on an additional trace collected by a server while sending files to different hosts.

The following table describes the traces you will work on:

Trace group name	Description
spr_by_window	trace-group from Sprint's 3G network. The position of the computer that captured the trace was next to a window.
spr_desk	trace-group from Sprint's 3G network. The computer used to capture the trace was on a desk.
comcast	trace-group from a home that has Comcast XFinity.
airplane	trace-group from a satellite link that was captured during a flight to California.
vzw	trace-group that was captured from Verizon's cellular network.
rtt_trace	trace of a web-server. (Use this trace only for RTT calculation)

Table 1: Description of the trace groups provided for this assignment

You should analyze at least three traces from different groups listed in the first five rows of Table 1. Analyze more traces from other groups to get bonus points. You will also use the rtt_trace for analyzing the round-trip time as will be illustrated shortly.

In order to analyze a trace, you need to write a program that will take a trace as input, and will collect all the information that you need in order to answer the questions below. Note that trace collection is not perfect, and some traces might have missing SYN packets. You can find background information about packet sniffing in the Sniffer FAQ [6]. There are many tools that enables you to analyze network traces. The most popular one is Tcpdump [1] which is installed on linux-lab machines. In order to use it, type `/usr/sbin/tcpdump -r <filename>`.

2 Programming with *pcap*

To analyze the traces, you will need to develop a small tool that uses the packet capture (pcap) API to extract information for the provided packet traces. Unix-like systems implement pcap in the libpcap [1] library (WinPcap [4] is the library that implements pcap in Windows).

To learn how to use the API, read carefully through `http://www.tcpdump.org/pcap.htm`. Make sure that you actually understand the code tutorial and are not just blindly copying portions of the code that seem to fit your needs. If you use some of the code snippets that you will find on this tutorial or any other tutorial, you should **cite** your sources in both the write up and the code files that you will submit.

3 Write-Up

Along with your pcap reader tool, you should also submit a write up (at least 3 pages, with 10 points Times Roman font), in which you provide answers for the questions below. Some of the questions will require you to produce plots. Therefore, if you need more paper to explain your answers, or to zoom in on some of your plots please do. However, **do not** append the packet reader code as an appendix to your write up.

Feel free to use any of the following tools to produce your plots: jgraph [2], gnuplot [3] or Matlab. If you try to use MS Excel you might end up having some problems for some of the plots since the number of values it can plot is limited. Do not ask the TAs for how to fix problems with Excel or tools not listed above. We have provided you with a simple jgraph sample file and with a README file on how to use the program. However, feel free to use any plotting tool you're comfortable with. The following list provides resources for both jgraph and gnuplot:

1. jgraph: `http://web.eecs.utk.edu/ plank/plank/jgraph/jgraph.html`
2. gnuplot: `http://gnuplot.sourceforge.net/demo/`

4 Questions

4.1 Receiver traces

For each trace you choose for analysis, you should answer the following questions:

1. List both the IP addresses and port numbers for both connection endpoints.
2. How many data packets were received from the file sender? How many acknowledgement packets were sent by the file receiver? What was the file downloading throughput in bytes/sec?
3. Based on the metrics above, compare the performance of the trace groups from which you selected the traces.
4. Did the connection experience losses? Did these losses slow down the connection or were there enough packets in the queue to absorb the losses? You should be able to answer these questions by interpreting the sequence number plot, looking for periods of slow start, idle periods, .. etc. Compare your different traces.

5. If there were any losses, did they occur when the window was large or did the size of the window have little effect? We don't expect a statistical analysis but a visual qualitative interpretation of your plots is sufficient.

4.2 Sender traces

In the following questions, you will work on the rtt_trace file. The rtt_trace file consists of 20 different TCP transactions. We instructed 20 nodes to download a file from our web server. The web server captured these 20 transactions in the `rtt_trace`. You should study at least one of the nodes. Table 2 shows all the nodes that took part in these transactions. In order to select which host to analyze use the following technique: Given that your class account login id is cs417xyz, the id of the host you should select can be computed as xyz modulo 20. Therefore, if your login id is cs417050, then you should choose host number 10 from Table 2 (50%20=10). The questions you should answer are the following:

- Analyze and plot the round trip time for at least one TCP transaction in the `rtt_trace`. Feel free to compute and plot the round trip time for more than one node for bonus points. Note that it is not necessary to implement Karn's [7] algorithm to ignore samples during retransmission events.
- How many data packets were retransmitted? How many duplicated acknowledgement packets were received at the sender side?
- Plot the sizes of the congestion window (the amount of unacknowledged data in the network at a time) over time. What can you infer from the plot? How does the congestion control policy affect the performance of the transaction?
- If there is unfilled space in the receiver's buffer, perhaps the sender's window is the limit. What is the size of the sender's buffer?

4.3 Guidelines

Use your judgement to explain the plots and answer the questions of the assignment. Make sure you answer all questions, but please **do not** list them in your write up. A list of the **plots** you should generate for each one of your traces follows:

1. Sequence number plot.
2. Unused receiver buffer space plot.
3. Congestion or actual window plot.
4. Transmission rate plot.
5. Round trip time plot.

A list of the **metrics** you should calculate or estimate *include*:

1. overall performance: time, bytes and throughput.
2. losses, detection and how they are correlated with the window size.

ID	Host
0	ec2-107-22-104-157.compute-1.amazonaws.com
1	ec2-54-242-88-149.compute-1.amazonaws.com
2	ec2-23-20-133-128.compute-1.amazonaws.com
3	ec2-174-129-46-232.compute-1.amazonaws.com
4	ec2-54-242-91-171.compute-1.amazonaws.com
5	ec2-50-19-155-132.compute-1.amazonaws.com
6	ec2-23-20-6-16.compute-1.amazonaws.com
7	ec2-54-242-80-150.compute-1.amazonaws.com
8	ec2-23-22-149-241.compute-1.amazonaws.com
9	ec2-23-20-224-215.compute-1.amazonaws.com
10	ec2-50-16-88-27.compute-1.amazonaws.com
11	ec2-23-22-102-244.compute-1.amazonaws.com
12	ec2-54-242-46-145.compute-1.amazonaws.com
13	ec2-23-22-199-249.compute-1.amazonaws.com
14	ec2-50-19-199-215.compute-1.amazonaws.com
15	ec2-23-20-139-174.compute-1.amazonaws.com
16	ec2-107-20-40-35.compute-1.amazonaws.com
17	ec2-23-23-45-129.compute-1.amazonaws.com
18	ec2-23-23-35-14.compute-1.amazonaws.com
19	ec2-23-20-23-174.compute-1.amazonaws.com

Table 2: Host mapping

What should you submit?

1. A write up that answers the questions of the assignment.
2. In your write up, state clearly which traces you chose to analyze.
3. All the code you used to gather the data you needed to answer the questions.
4. A README file in which you explain how the TA will use your code to reproduce the data you gathered.
5. You can use Wireshark [8] to **verify** some of your answers.
6. MS-WORD write ups are **NOT** acceptable; you can still *use* MS-WORD, but send your write up in a **pdf** format.

This assignment is based on a prior networking programming assignment[5] by Neil Spring.

References

[1] *Packet Capture.* http://www.tcpdump.org/

[2] *Jgraph.* J.Plank http://www.cs.utk.edu/ plank/plank/jgraph/jgraph.html

[3] *GnuPlot*. http://www.gnuplot.info/

[4] *Winpcap* http://www.winpcap.org/install/default.htm

[5] Neil Spring, *Networking Project 3* http://www.cs.washington.edu/education/courses/cse461/00au/pa3-handout.pdf

[6] R.Graham, *Sniffer's FAQ*, http://newdata.box.sk/2001/jan/sniffing-faq.htm

[7] Karn's Algorithm, http://en.wikipedia.org/wiki/Karn's_Algorithm

[8] *Wireshark*, http://www.wireshark.org