CSc 361: Computer Communications and Networks (Spring 2016)

Assignment 2: TCP Traffic Analysis

Spec Out: Feb. 5 2016
Final Due: 3:30 pm, Feb. 26, 2016

1 Goal

The purpose of this project is to learn about the Transmission Control Protocol (TCP). You are required to write a C program with the pcap library to analyze the TCP protocol behavior.

2 Requirements

You will be given a TCP trace file (cap-feb-6 in connex resource). During the period traced, a single web client accesses web pages from different web sites on the Internet. This trace is to be used to for your own test. Your code might be tested with another trace file, which will be disclosed after your final submission.

You need to write a C program with the pcap library for parsing and processing the trace file, and tracking TCP state information. In particular, the program processes the trace file and computes summary information about TCP connections. Note that a TCP connection is identified by a 4-tuple (IP source address, source port, IP destination address, destination port), and packets can flow in both directions on a connection (i.e., from host A to host B, and from host B to host A). Also note that the packets from different connections can be arbitrarily interleaved with each other in time, so your program will need to extract packets and associate them with the correct connection.

The summary information to be computed for each TCP connection includes:

- the state of the connection. Possible states are: S0F0 (no SYN and no FIN), S1F0 (one SYN and no FIN), S2F0 (two SYN and no FIN), S1F1 (one SYN and one FIN), S2F1 (two SYN and one FIN), S2F2 (two SYN and two FIN), S0F1 (no SYN and one FIN), S0F2 (no SYN and two FIN), and so on, as well as R (connection reset due to protocol error). Getting this state information correct is the most important part of your program. We are especially interested in the complete TCP connections for which we see at least one SYN and at least one FIN. For these complete connections, you can report additional information, as indicated in the following.

- the starting time, ending time, and duration of each complete connection

- the number of packets sent in each direction on each complete connection, as well as the total packets

- the number of data bytes sent in each direction on each complete connection, as well as the total bytes. This byte count is for data bytes (i.e., excluding the TCP and IP protocol headers).

Besides the above information for each TCP connection, your program needs to provide the following statistical results for the whole trace data:

- the number of reset TCP connections observed in the trace
- the number of TCP connections that were still open when the trace capture ended
- the number of complete TCP connections observed in the trace
- Regarding the complete TCP connections you observed:
  - the minimum, mean, and maximum time durations of the complete TCP connections
  - the minimum, mean, and maximum RTT (Round Trip Time) values of the complete TCP connections
  - the minimum, mean, and maximum number of packets (both directions) sent on the complete TCP connections
  - the minimum, mean, and maximum receive window sizes (both sides) of the complete TCP connections.

For the output, please strictly follow the output format of this project (outputformat.pdf in connex resource).

3 Deliverables and Marking Scheme

For your final submission of your assignment, you are required to submit your source code to connex. You should include a readme file to tell TA how to compile and run your code. At the last lab session that you attend, you need to demo your assignment to TAs. Nevertheless, before the final due date, you can still make changes on your code and submit a *change.txt* file to connex to describe the changes after your demo.

The marking scheme is as follows (refer to outputformat.pdf in connex resource as well):

Components	Weight
Make file	5
Total number of connections	15
Connections' details	30
General Statistics	20
Complete TCP connections:	20
Code style	5
Readme.txt and change.txt(if any)	5
Total Weight	100

4 Plagiarism

This assignment is to be done individually. You are encouraged to discuss the design of your solution with your classmates, but each person must implement their own assignment.

5 Extra Info: Code Quality

We cannot specify completely the coding style that we would like to see but it includes the following:

1. Proper decomposition of a program into subroutines (and multiple source code files when necessary)—A 500 line program as a single routine won't suffice.
2. Comment—judiciously, but not profusely. Comments also serve to help a marker, in addition to yourself. To further elaborate:
   (a) Your favorite quote from Star Wars or Douglas Adams' Hitch-hiker's Guide to the Galaxy does not count as comments. In fact, they simply count as anti-comments, and will result in a loss of marks.
   (b) Comment your code in English. It is the official language of this university.
3. Proper variable names—`leia` is not a good variable name, it never was and never will be.
4. Small number of global variables, if any. Most programs need a very small number of global variables, if any. (If you have a global variable named `temp`, think again.)
5. **The return values from all system calls and function calls listed in the assignment specification should be checked and all values should be dealt with appropriately.**

The End
