CIS–3152 Lab #3
Transmission Control Protocol

Peter C. Chapin[*]
Vermont Technical College

Last Revised: January 28, 2015

Introduction

In this lab you will use Wireshark to observe the TCP segments that flow between a simple client and a simple server. You can use the daytime client and server provided as a class sample. To observe network traffic between the client and server you have two choices.

1. You can run both client and server on HackBox and capture traffic on the loopback interface. Be aware, however, that some of the parameters of the loopback interface are atypical. The data you observe for this lab will reflect those atypical values.

2. You can run the client on HackBox and the server on lemuria. However this will only be possible if you are on campus.

It doesn't matter which of these two options you choose, but you should make a note of your choice in your report.

1 Procedure

Proceed as follows.

1. Run the client and connect it to the server. Use Wireshark to watch the traffic between the two programs. Note: You may find it useful to filter the displayed packets so that only those on the connection of interest are shown.

2. Record the following information about every segment exchanged (in some cases the information is consistent from segment to segment; if that is so, you can just make a note of that fact).

   - Source and destination IP addresses (in the IP header) and port addresses (in the TCP header).
   - Which TCP flags are set.
   - Sequence numbers and acknowledgment numbers.
   - Window size.
   - Any options that are set.

   Record also the amount of data in each segment. Wireshark normally displays "relative" sequence numbers. It notes the initial sequence numbers used by both endpoints of the connection and displays sequence numbers that are relative to that value. Thus Wireshark shows the first byte of the connection as byte number 1. However, if you dig into the segment header itself you can find the true sequence numbers in use. It is those values that I want you to record for this lab.

3. Do the numbers you collected above make sense? In particular, are the sequence numbers and acknowledgment numbers correct? Does the window size change (or not) in an appropriate way? *Do the math!*

4. Run the client under the control of the debugger. Right after it has received an end-of-file indication but before it closes the socket use `netstat`

[*] PChapin@vtc.vsc.edu

`-nt` on both the client and server machines to observe the state of the TCP connection at each endpoint.

5. Run the server under the control of the debugger. Right after `accept` has returned but before any data is sent to the client use `netstat -nt` on both the client and server machines to observe the state of the TCP connection at each endpoint.

2 Report

Write a report that describes your observations and explains what you saw. In your report be sure to show the data you collected on the segments you observed. Were there any unexplained or unexpected results? Do the TCP states you saw make sense?