**CS3506 - Wireshark Lab 2: TCP**

In this lab, we’ll explore several aspects of the Transport Control Protocol:

* TCP traffic generated by a POST HTTP message;
* the TCP ACK, slow start and congestion control mechanisms;
* the format and contents of a TCP packet.
* Data contents of a TCP packet

Ensure your PC is running Linux and remember that you have to run Wireshark as the administrative user, so at the command line type “**sudo wireshark**” (without the quotation marks).

You may first also need to run “**xhost +**” to ensure that the X Window System allows Wireshark access to the display.

**TCP - E-mail trace**

Download a captured trace called **captured\_mail.pcap** for an email transaction from the Canvas page and open it in your Wireshark. Keep the Wireshark window open.   
Print and fill out this assignment

Let’s begin by loading the trace. Before analysing the behaviour of the TCP connection in detail, let’s take a highlevel view of the trace.

You should see a series of TCP and SMTP messages between a client and an email server. It starts with the initial three-way handshake containing a SYN, a SYN-ACK and an ACK message.

*Now answer questions 1.1-1.9*

**TCP - HTTP**

Download **http-v-https.pcap** from the Canvas page and open it in your Wireshark.

What you should see is a series of TCP, TLS and HTTP messages between a client and a web server.

*Note*: Wireshark’s power lies in filters. You can create a filter for every field in a packet. You can select a field, right click on it and select “apply as a filter”. Now Wireshark will only display the packets that have the same value for this field. Here are some filters that might be of use for this lab:

tcp.stream eq 0

tcp contains FROM

tcp contains BONUS

tcp.analysis.retransmission

ip.addr == 143.239.211.230 && tcp.port == 80

This capture file shows the download of the same file over HTTPS and then HTTP so ultimately you have the same data transferred in both interactions.

*Now answer questions 2.1-2.3*