National University of Singapore
School of Computing

CS2105 **DIY Exercise 2** Semester 2 13/14

You may run the following on Linux, Mac OS X, or Microsoft Windows. The behaviour may be slightly different (but should not impact your learning).

You need a network connection for this exercise. Bad network connection (e.g., wifi with weak signal) works better for this exercise.

HAVE FUN, AND BE CURIOUS

`Wireshark` is a GUI tool for capturing and analyzing network packets. It is freely available on http://www.wireshark.org/. A video of short wireshark tutorial has been posted – you should check it out if you are not familiar with wireshark.

Download and install wireshark on your computer. You will need it for this exercise and future exercises.

It is best if you stop running as many networked programs as possible, and close any many browser windows/tab as possible.

1. Launch Wireshark. In the field "Filter:", enter

   ```
   tcp.port == 80
   ```

   and click "Apply."

   Under "Edit" → "Preferences.." → "Protocols" → "TCP", make sure that the option "Relative sequence numbers" is turned off, and "Analyze TCP sequence numbers" is turned on.

   Start capturing packets by typing Ctrl-E.

   If you are running any application that uses HTTP, you should see some packets being listed, ignore them.

2. On a terminal, run

   ```
   telnet www.nus.edu.sg 80
   ```

   Wait until the connection closes. Type Ctrl-E in Wireshark to stop capturing.

   Wireshark have captured the TCP packets exchanged between your computer and www.nus.edu.sg. A sequence of exchanges between two host/ports is called a *conversation*. You can right click on a packet that belongs to that conversation and choose "Conversation Filter" → "TCP", to show only that conversation. You may want to colorize the conversation so that the text is easier to read.

   You should see some packets used for connection establishment and for connection termination. Examine various header fields in the TCP headers of these packets and relate them to what you have learnt about (i) the functions of these fields and (ii) how TCP establishes/terminates connection.

3. At your display filter, you should now see a filter with four conditions, two for IP addresses and two for TCP port numbers. This display filter is set for you when you chose to filter a conversation.

   Now edit the field "Filter:" to remove the conditions involving TCP ports.

   Start capturing again with Control-E (if you are asked to save a previous capture, just answer the question and continue).

   On a terminal, run

```
curl www.nus.edu.sg
```

You should now see a longer conversation that includes some HTTP request/response exchanges between your computer and www.nus.edu.sg.

Wait until the connection closes. Type Ctrl-E in Wireshark to stop capturing.

(a) Look through the TCP sequence numbers and ACK numbers exchanged carefully. Observe how the sequence numbers and ACK numbers changes.

(b) Look for packets that are marked as "TCP Out-Of-Order," "TCP Previous segment lost," "TCP Dup ACK," "TCP Retransmission," or "TCP Fast Retransmission" (You may not necessary find such packets in your trace, especially if you are on a good connection). Observe the sequence numbers and ACK numbers of such packets and the packets before/after.

(c) Roughly count the number of ACK packets and data packets. What is the ratio of the number of ACK packets to data packets?

Why isn't the ratio equal to one? (Hint: Read up about how TCP actually acknowledges segments in the textbook).

THE END