TCP Behaviour and Congestion Control

Networked Systems (H) 2025-2026 – Laboratory Exercise 5
Prof Colin Perkins, School of Computing Science, University of Glasgow

1 Introduction

The laboratory exercises for Networked Systems (H) will introduce you to network programming in C using the Berkeley Sockets API, and help you understand the operation and structure of the network. The exercises will help you practice C programming, building on the Systems Programming (H) course, and introduce network programming in C. Other exercises will illustrate key points in the operation of the network. The laboratory exercises are intended to complement the material covered in the lectures. Some expand on the lectures to give you broader experience in a particular subject. Others exercises cover material, such as network programming in C, that's better taught by doing than by lecturing.

This laboratory exercise reviews TCP behaviour and congestion control. **The assessed coursework for Networked Systems (H) is based on the material in this laboratory exercise**.

2 Preliminaries

Retrieve the file `lab05.zip` from Moodle or from the course website, and unzip it to produce the file `lab05.pcap`.

The `tcpdump` command can be used to capture the packets that comprise a TCP connection and store them into a packet capture (.pcap) file. The `lab05.pcap` file is an example of such a recorded TCP connection, capturing the download of a single file from a web server using HTTPS. The `tcpdump` command can also be used to inspect the contents of a packet capture. Using one of the `stlinux` machines, or on your own machine if you have `tcpdump` installed, run the following command to display the contents of the `lab05.pcap` file:

```
tcpdump -r lab05.pcap
```

The resulting output will be 1,389 lines long, with the first few lines looking like the following:

```
17:28:25.104219 IP mangole.dcs.gla.ac.uk.52088 > yali.mythic-beasts.com.https:
                Flags [SEW], seq 2687265183, win 65535,
                options [mss 1460,nop,wscale 6,nop,nop,TS val 3652396230 ecr 0,sackOK,eol],
                length 0
17:28:25.117936 IP yali.mythic-beasts.com.https > mangole.dcs.gla.ac.uk.52088:
                Flags [S.E], seq 2648910570, ack 2687265184, win 65160,
                options [mss 1380,sackOK,TS val 1395313631 ecr 3652396230,nop,wscale 7],
                length 0
17:28:25.118029 IP mangole.dcs.gla.ac.uk.52088 > yali.mythic-beasts.com.https:
                Flags [.], ack 1, win 2052, options [nop,nop,TS val 3652396243 ecr 1395313631],
                length 0
```

The output above has been wrapped to fit this page: the actual output will be longer unwrapped lines, each starting with a timestamp. To make the output easier to read, you can send the output of the `tcpdump` command to a pager ("`tcpdump -r lab05.pcap | more`") or redirect it to a file ("`tcpdump -r lab05.pcap > filename`").

Review the documentation for the tcpdump command. This can be accessed using the command "man tcpdump" on the stlinux machines, or online at https://www.tcpdump.org. Ask the lecturer or one of the lab demonstrators if you are unsure how to read the output of tcpdump.

You will see that there are many options that can be passed to tcpdump to change how it captures and displays .pcap files. For example, the command "tcpdump -v -r lab05.pcap" will display more detail about the packet headers and contents, while the command "tcpdump -X -r lab05.pcap" will display the full contents of the packets captured.

The Wireshark application (https://www.wireshark.org) is an alternative to tcpdump that can capture and display similar information.

3 TCP Connection Handshake

The first part of this exercise involves using tcpdump to review the lab05.pcap file to understand the initial three-way handshake that starts a TCP connection. Run "tcpdump -ttttt -n -S -r lab05.pcap". This displays information about the packets in the .pcap file with timestamps calculated relative to the start of the connection ("-ttttt"), without converting IP addresses to DNS names ("-n"), and as absolute rather than relative sequence and acknowledgement numbers ("-S"). Look at the records for the first three packets. Observe that:

- The first packet has the SYN bit set in the TCP header (the "Flags [...]" in the output includes "S"), and includes an initial sequence number ("seq 2687265183"), a receive window, and some TCP options. It has zero length (i.e., it does not carry any data).
- The second packet also has the SYN bit set in the TCP header, and includes the initial sequence number used for the server-to-client direction, a receive window, some TCP options, and is of length zero. This packet also includes an acknowledgement for the first packet ("ack 2687265184"), that is, it's a SYN-ACK packet.
- The third packet does not have any flags set in the TCP header ("Flags [.]") and merely acknowledges the second packet.

These first three lines of the tcpdump output show the TCP three-way handshake used to establish the connection. The remainder of the output shows the packets carrying data that within the TCP connection, followed by connection shutdown packets. Read https://www.tcpdump.org/manpages/tcpdump.1.html#lbAM, then study the output of tcpdump, and discuss with the lecturer or lab demonstrators, to be sure you understand what it is showing.

4 Starting an HTTP Connection

Run the "tcpdump -ttttt -n -S -r lab05.pcap" command again, and consider the packets in lines 4-13 of the output. These packets comprise the TLS handshake, starting with packet 4 ("seq 2687265184:2687265507") that contains the TLS ClientHello.

The connection continues with an acknowledgement (packet 5), followed by packets 6-8 that contain the ServerHello message, all sent from the server to the client. Consider why the server might send a TCP packet containing only an acknowledgement, followed a short time later by the ServerHello, rather than immediately sending the ServerHello. Consider also what aspects of the ServerHello cause it to need three TCP packets to send. Discuss with the lecturer or lab demonstrators to be sure you understand.

Following on from this, consider what is the role of packets 9-13 in the connection. With reference to the blog post on "The Illustrated TLS 1.3 Connection", in the reading relating to Lecture 3, explain what are the contents of these packets.

5 TCP Congestion Control

Run the following command: "`tcpdump -ttttt -r lab05.pcap > packets.dat`". This formats the packets in the `tcpdump` file with timestamps displayed relative to the start of the connection, and saves the result in the file `packets.dat`. Once you have done this, edit the `packets.dat` file to:

1. remove packets sent from the client to the server, keeping only packets sent in the server-to-client direction;
2. remove packets sent in the initial and closing handshakes (i.e., remove packets in the `SYN`/`SYN-ACK`/`ACK` handshake at the start of the file, and in the corresponding `FIN`/`FIN-ACK`/`ACK` handshake at the end of the file); and
3. remove packets that do not contain data (i.e., remove any lines that contain `ack` but not `seq`).

The result should be a file containing one line for each packet sent from the server to client that contains data. You may edit the file manually, or you can write/use a program to make the changes.

Process the file to extract the timestamp and the last sequence number from the sequence number range from each line. Save the processed results in the file `lab05-timeseq.dat`. For example, if a line contains the text:

```
00:00:00.093456 IP yali.mythic-beasts.com.http > 192.168.0.66.56735: Flags [.], seq 1:1389, ack 122...
```

then the corresponding line in the `lab05-timeseq.dat` file would be:

```
00:00:00.093456 1389
```

Prepare a plot of the sequence number of the received packet vs. the time at which that packet was received for the `lab05-timeseq.dat` file. The x-axis of this plot will represent the time, in seconds since the start of the connection in the range from 0.00s - 0.25s; the y-axis will represent the final sequence number of the range covered by the packet. For each line in the `lab05-timeseq.dat` file, place a mark on the appropriate point on the graph. You may use any graph plotting tool you want to prepare this graph.

Review the graph you have produced to explain what is the behaviour of the TCP connection that you see. In particular, consider what aspects of TCP congestion control and reliable data transfer are illustrated in the connection. Discuss with the lecturer and lab demonstrators if necessary.

- + -