COMP 7005
Computer Systems Technology
September 2022
Network Security

Project

Due: Nov 29th, 2022 @ 5:29pm

This is pairs project.

Objective

Simulate a lossy network.

Tasks

- Create a reliable protocol based on TCP (that means that the TCP data portion holds your protocol)
- Create a sender that reads from the keyboard or a file and writes to a TCP socket.
- Create a receiver that reads from a TCP socket and writes to the console.
- Create a proxy that sits between the two hosts that:
  - Randomly drops data from the sender
  - Randomly drops acks from the receiver
- The receiver must send an acknowledgement (not TCP, your own acknowledgement in the TCP data) back to the sender (via the proxy)
- If the sender does not get the acknowledgement in a reasonable time (as determined by you) then it resends the packet.

Constraints

- The sender must take command line arguments for the IP address of the proxy (or receiver if the proxy is removed), the port, and, optionally, the file to read
- The proxy must take command line arguments for the IP address of the receiver, the port to send to, and the port to listen on
- The receiver must tame a command line argument for the port to receive from
- The proxy must have some way to set the % chance of the data packets (from the sender) to drop (command line, GUI, or some other means)
- The proxy must have some way to set the % chance of the ack packets (from the receiver) to drop (command line, GUI, or some other means)
- All 3 programs must maintain a list of statistics showing how many packets have been sent and received.

- You must provide a design for the sender, receiver, and proxy
- You must provide videos (tmux us a good idea) of the sender, receiver, and the proxy working
- You can obtain bonus marks for:
    - The proxy has a % chance to delay a data or ack packet
    - A GUI that graphs the data
    - Dynamically changing the % chance to drop (or delay) at runtime
    - Implementing a window based-protocol in instead of send and wait

Submission Requirements

Use the following directory structure:

Directory	**Purpose**
source	Any source code files
report	Report files in .pdf format
video	Video(s) demonstration of your working project
pcap	Any relevant packet captures

Notes

- Follow the appropriate report format (samples).
- The demo video should cover each one of your test cases. In other words, it will be similar to an actual lab demo, except you will prepare a video of each test instead of me standing beside you observing each test.
- You can have a separate video for each test (**preferred**) or a combined video of all test cases.
- During the test, you will capture network traffic on any relevant machines and then submit the pcap files as specified above.
- Please set the appropriate packet capture filter to limit the size and scope of the data collected to be what is necessary.

Format

You must hand in a pax.Z file to the assignment submission folder on Learning Hub (https://lear.bcit.ca).

You can hand in as many versions as you like. The last submission, based on the timestamp, will be the one to be marked.

```
pax -w report/ source/ video/ pcap/ -f project-v#.pax
compress -f project-v-#.pax
```

Hand in the resulting project-v-#.pax.Z file.

Note: failure to follow the submission requirements may result in a loss of marks, up to 100%.

Evaluation

Item	Value
Server	20
Client	10
Proxy	30
Report	20
Data (pcap, videos)	20
Total	**100**

Notes

- Please put a chart at the start of the report that clearly shows any bonus or “make it better” tasks.