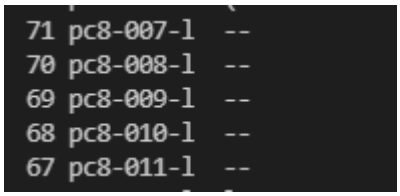
Unix User ID

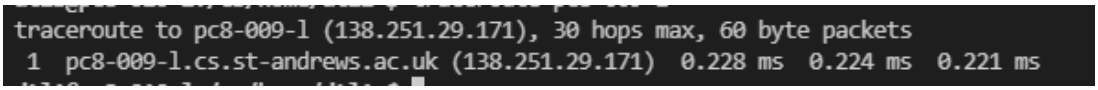
21929

Part 1

Initially I ran the user_login_check.sh script to find two lab clients that didn't have any logged in users.



I decided to use pc8-009-l to run slurpe-3 and use pc8-010-l to run my slurpe-probe.

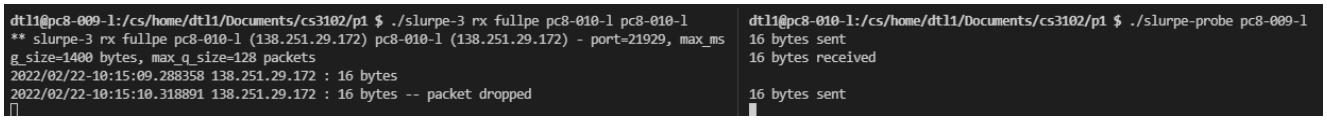


Then I ran traceroute to check there were no nodes between the two machines.

I examined the code in the wk03 Lectures directory, and with help from the code in "udp-client-1.c", I designed my slurpe-probe program to send a single packet and exit. I used my unix user id (21929) to try sending this packet to my slurpe-3 program. Once that was working and I could see slurpe-3 receiving my packet I modified my slurpe-probe program to include the header file "UdpSocket.h".

Using the UdpBuffer struct within UdpSocket, I modified my program to create and send one of these struts, so it was now sending packets in a binary format. Next, I created a while loop to send multiple packets in a row and sleep the CPU between them, and because the creation of the packet is within the loop I had to use malloc and free to dynamically allocate memory. The next step was to implement the function "recvUDP" to receive packets being sent back by slurpe-3, after this I made my slurpe-program print the return values of "sendUdp" and "recvUdp" so I could debug my program more easily.

Next was to introduce in the network path emulation. Because of the packet loss emulation I noticed that my probe program would hang when a packet was dropped, (see below).



I realised that this was because recvUdp was blocking. So using the code found in "udp-chat-2.c" I set the i/o to be non-blocking and this fixed the issue. I then modified my program to output "packet dropped" when it didn't receive back a packet it had sent. (see below).

```
2022/02/22-10:19:23.093697 138.251.29.172 : 16 bytes                                16 bytes received
2022/02/22-10:19:24.093847 138.251.29.172 : 16 bytes
2022/02/22-10:19:25.093995 138.251.29.172 : 16 bytes                                16 bytes sent
2022/02/22-10:19:26.094134 138.251.29.172 : 16 bytes                                16 bytes received
2022/02/22-10:19:27.094285 138.251.29.172 : 16 bytes
2022/02/22-10:19:28.094428 138.251.29.172 : 16 bytes -- packet dropped              16 bytes sent
2022/02/22-10:19:29.094568 138.251.29.172 : 16 bytes                                packet dropped
2022/02/22-10:19:30.094706 138.251.29.172 : 16 bytes
2022/02/22-10:19:31.094867 138.251.29.172 : 16 bytes                                16 bytes sent
2022/02/22-10:19:32.094999 138.251.29.172 : 16 bytes                                16 bytes received
```

My last step for part 1 was to include the "clock_check.c" file so I could print the start, finish and packet arrival times during the execution of my program. I also added column headings and all the extra necessary information. (see below).

```
** slurpe-probe pc8-009-l : src/dst port 21929, size 16
** sending 10 packets, 1 every 1 second(s)
** clock resolution: 1 ns, start time:  1645528592 s 983457512 ns

packet arrival time     packet number   packet size (bytes)
1645528593.983604322            1               16
1645528594.983755487            2               16
1645528595.983898131            3               16
1645528596.984049102            4               16
1645528597.984199736            5               16
1645528598.984349064            6               16
                ----- packet dropped -----
1645528600.984640610            8               16
1645528601.984780651            9               16
1645528602.984932772            10              16
end time: 1645528603 s 985075090 ns, packets sent: 10, packets lost: 1, packet loss: 0.10
```

Here is an example run of sending 10 packets to slurpe-3 rx fullpe.

Part 2

For part 2 I tried five separate runs of slurpe-3 and my probe program. The four five consisted of no path emulation, just delay, just loss, just rate and fullpe. And for each run my probe program sent 100 probes.

Then I copied the output from my program for each of these runs into Microsoft Excel so I could analyse the measurements. In total I created 8 sets of measurements from this part and my Excel document contains 8 sheets.

No parameters

Time taken: 101 s 1486185 ns

Just delay

Time taken: 101 s 0149397 ns

Just loss

Time taken: 101 s 1465063 ns

Packets lost: 7

Packet loss: 7%

Just rate

Time taken: 101 s 015101 ns

Fullpe

Time taken: 101 s 1473392 ns

Packets lost: 7

Packet loss: 7%

I did another run for loss and instead sent 500 packets for a more accurate measurement of packet loss.

The final line for this output was:

end time: 1645531228 s 677315883 ns, packets sent: 500, packets lost: 43, packet loss: 0.09

With the run of 100 packets giving a packet loss value of 7% and a run of 500 packets giving a value of 9% you can see that the value is tending towards 10% which is to be expected as mentioned in the spec.

A big problem, however, is that the tame taken for the delay and rate runs is not longer than the time taken for no parameters, which is not what is expected, I thought I try these again with 200 probes to see if I could get a more accurate result.

No parameters – 200 probes

Time taken: 201 s 0299921 ns

Just delay – 200 probes

Time taken: 201 s 0295518 ns

But again they were very similar, I don't really have an idea of why I'm not seeing any delay.

Critique

There are a few potential sources of errors within my submission, maybe one or more of these could explain why I wasn't getting expected results for part 2. They are as follows:

- Probe packet contents – I wasn't entirely sure on what would be appropriate data to include in the probe, so I settled for a random number of 123. Perhaps more / less data is required.
- Probe packet size – perhaps the size wasn't large enough for slurpe-3 to perform its path emulation.
- Part 2 runs – I only ran probe tests of 100 and 200, maybe more was needed for more accurate results, but I still think that I should've seen longer times for delay and rate even at these numbers of probes.
- Program output – maybe outputting packet arrival time isn't enough, perhaps a time difference measurement was needed.