University of Leeds Computer Systems (COMP1440)

Coursework 2 Deadline: 10.00, 9th December 2014

This exercise aims to give you some practical experience of network diagnostic programs and network programming in Python and to give you better understanding of network layered architectures, and some practical experience of network simulation.

Useful Resources

- Example Python code is available on the module area on the VLE

- Example ns2 simulation scripts are available on the module area on the VLE

Task 1 (15 marks)

As you write programs to use network services, you will most frequently find yourself writing network clients. The example program `connect.py` shows how to communicate with sockets on the client side. Inspect this program then run it.

1. Modify the program `connect.py` so that it connects to a server of your choice. What is the name of this server?

2 You are required to measure the elapsed time to connect to the server. To do this, Python provides support for keeping track of elapsed time directly from within the program. The function `time()` returns the current time according to the computer's system clock. A standard pattern for recording the elapsed time is:

```
start = time()  # record the starting time
# ... perform some task
stop = time()   # record the finish time
elapsed = stop - start # measured in seconds (or fraction
thereof)
```

Include this piece of code in the relevant place in `connect.py` (you need to include in your submission a copy of the amended program).

3 Run the program 5 times and record the elapsed time for each run.

4 Explain the discrepancy between the elapsed time for each run.

5 Calculate the average elapsed time to connect to the server.

Task 2 (15 marks)

Suppose the network layer provides the following service. The source host accepts from the transport layer a segment of maximum size 1,000 bytes and a destination host address. The network layer then guarantees delivery of the segment to the transport layer at the destination host. Suppose many network application processes can be running at the destination host.

1. Design the simplest possible transport layer protocol that will get application data to the

desired process at the destination host. Call this protocol STP (Simple Transfer Protocol).

2. Assume the operating system in the destination host has assigned a two-byte port number to each running application process. What is the maximum size of the chunk of data STP handles between the sending and receiving processes?

3. In your protocol, does the transport layer have to do anything in the core of the computer network?

Consider now a planet where everyone belongs to a family of five, every family lives in its own house, each house has a unique address, and each person in a house has a unique name. Suppose this planet has a mail service that delivers letters from source house to destination house. The mail service requires that

- the letter be in an envelope, and that
- the address of the destination house (and nothing more) be clearly written on the envelope.

Suppose each family has a delegate family member who collects and distributes letters for the other family members. The letters do not necessarily provide any indication of the recipients of the letters.

1. Using the solution to Question1as inspiration, propose a protocol that the delegates can use to deliver letters from a sending family member to a receiving family member.

2. In your protocol, does the mail service ever have to open the envelope and examine the letter in order to provide its service?

Task 3 (15 marks)

Inspect the network simulator ns2 example script `simple.tcl` and answer the following questions:

1. How many nodes are there in the network? Draw the network topology and specify the bandwidth and latency on each link.

2. What is the name of the application running on each node?

3. What transport protocol does each application use?

4. What is the simulation duration?

5. Run`simple.tcl`andexplainwhatthesimulationdoes.

6. Amend the network topology to include (anywhere) an extra node in the network. Draw the network topol- ogy and specify the necessary parameters on the link. Re-run the simulation and discuss your results.

Submission

1. Prepare a typeset `pdf` document containing your answers to Tasks 1–3.

Convert other document formats (e.g. Word, OpenOffice) to `pdf` before submission. Submit through the VLE under **Submission** before the stated deadline.

2. Submit your answers on paper in the normal way, via the coursework postbox.

   Remember to attach a completed coursework header sheet.

Weight

This coursework accounts for 8% of the assessment.