Coursework – Network Application Development Coursework Weight: 60%

In this coursework, you will develop a number of small networking-based applications. These are designed to increase your competency in developing socket-based applications, as well as increasing your familiarity with a number of key technologies and measures. These are in widespread use, and commonly deployed to evaluate networks and to provide services over them.

During this practical, you will become familiar with the concept of network sockets and begin to understand how they are used. Sockets are a programming abstraction designed to assist us in building applications that use the network. We can treat these the same as any other resource; writing to a socket sends a packet into the network, whilst reading from a socket provides us with the contents of the packet. We can do this in much the same way as we would read and write to any file found on the local filesystem.

Practical Lab Structure

The coursework of Network Application Development is split into a number of smaller tasks: Task 1.1 ICMP Ping Client, Task 1.2 Traceroute Client, Task 2.1 Web Server and Task 2.2 Web Proxy. Importantly, the tasks build upon each other; the work you do in Task 1.1 will be fundamental to Task 1.2, and similarly, the work completed in Task 2.1 will greatly assist you in Task 2.2.

You are encouraged to progress as far as possible with each task. Do note however, that Task 1 and Task 2 are independent; attempting both of them is advised, even if you do not fully complete each.

Python Library Usage

You are not expected to use any external libraries for this practical; doing so is strictly prohibited. All tasks can be achieved fully with the use of standard Python libraries.

We are also aware of a number of network and IP-orientated libraries that are included within the standard Python distribution. These could potentially be used in different ways to assist in your implementation. However, as we are trying to build your understanding around the fundamentals of computer networks, we ask that you do not use these for this practical either.

The teaching team believe it is vitally important that you grasp the technical details behind many of these libraries, which do a good job of abstracting and obscuring the details. It is of course perfectly acceptable to use these libraries in any future software development you may do, whether this be as part of an upcoming course module or even after graduation.

By following the provided structure and guidance, you will not need to use any of these. If you are in any doubt about whether or not you can use a particular library, please contact the course tutors to confirm.

Submission and Assessment

The submission for all work completed in this practical is due by the end of **Week 9** (Sunday). Please submit 4 distinct Python scripts, named according to each task. Even though you can reuse code from earlier tasks in the later tasks, it will simplify the marking procedure if you submit each solution independently.

What you will need to submit:

- Python scripts for 4 tasks named “ICMPPing.py”, “Traceroute.py”, ”WebServer.py” and “ProxyServer.py”.
- If you finished any of the additional functions, please submit the relevant Python scripts, for example, “WebClient.py”.

You will be awarded the majority of marks for completing each task. The submitted Python scripts will account for 80% of the total marks, where they should meet the requirements given in the Task documentation. There will also be a small proportion of marks available for a consistent code style and useful commenting. Resilient code, using *try* and *except* statements to catch errors is also preferred. Also, having descriptive comments can help to increase the readability of code. Hence, the descriptive comments in your code and resilient code will be rewarded accordingly, which in total accounts for up to 10% of the total marks.

During the marking session (scheduled for **Week 10**), you will be expected to demonstrate the functionality of these scripts. You will mainly be assessed on functionality but also expected to be able to walk-through and explain your code. 2 of the 4 files will be chosen for demonstration. You need to give brief explanations for these scripts and will be asked questions during your demonstration process, which will account for 10%. As we will also be providing you with a few small snippets of code (to use in your own solution), you will not be expected to explain these in great detail. However, a general understanding of how these functions work will be beneficial to your overall learning and comprehension.

The additional functions will be marked where each accounts for 1% and will be added to your total mark. The total mark will not exceed 100%.