ELEC/TELE/PHTN 4123

Networked Communications Elective Topic, S1 2017

created by Prof. D. Taubman

Australia's Global University | Faculty of Engineering | School of Electrical Engineering and Telecommunications

Organization and Marking of the Elective Design Topics (General)

The elective design topics are all undertaken between Weeks 10 and 12 in the laboratory. Unlike the core design topics, the elective topics are undertaken by a team of 3 or (preferably) 4 students. Some of the marks are awarded to the team as a whole, while others are awarded individually.

It is highly desirable for all students in an elective design team to belong to the same tutorial group. While this might not always be achievable, the following factors should be considered when forming teams:

1. There will be no rearrangement of tutorial membership to accommodate the elective design topics. Each student is expected to attend his/her originally assigned tutorial, regardless of elective design team membership.
2. During the tutorials in Week 11 and Week 12, students are expected to discuss ideas, problems and their understanding of the elective design topic they are working on.
3. Roughly half of the tutorial time will be spent discussing these matters in a team huddle. The other half of the tutorial time will be spent exchanging ideas and problem understanding between teams. This requires students to be competent in explaining the design problem that their team faces; it also requires students in other teams to pay close attention and contribute useful suggestions or clarifying questions. It is expected that this monitored interaction between design teams will bring considerable benefits to learning and outcomes.
4. If a team of (say) 4 students is split across two tutorial groups, the responsibilities outlined above effectively become more challenging for the individual students. For example, if only 2 students from a team belong to a given tutorial, they become fully responsibility for explaining the problems and progress of their team to the tutorial. It is not sufficient for individual team members to be familiar only with some aspect of the work that is their primary responsibility. All students in each elective design team need to be familiar with the problems and approaches being employed by the entire team.

As with the other design topics, your individual participation in tutorials is assessed separately and contributes to the tutorial component of your final mark (20% over the duration of the course).

The laboratory component of your assessment for the elective design topics is worth 17% of the marks for the course. This is represented by a mark in the range 0 to 85, which consists of two components: a design outcomes component (up to 50 marks); and an understanding component (up to 35 marks). The design outcomes component is awarded to the team as a whole, whereas the understanding component is individually assessed. All assessment of the laboratory component of the elective design topics takes place in Week 12 only.

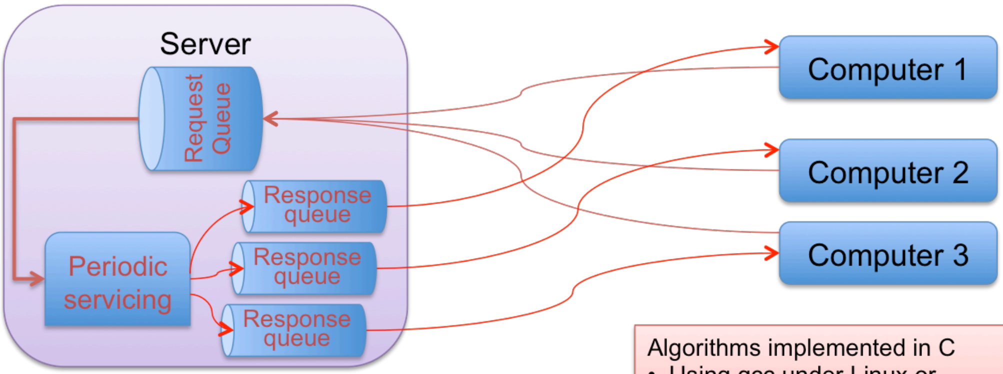
Additionally, 12% of your assessment for the course as a whole is based upon a report that you write as a team and submit by email to d.taubman@unsw.edu.au, no later than 5pm Thursday in Week 13. A draft version of the report must be available from the start of the laboratory session in Week 12. This will form part of the basis for assessment of understanding, as individual team members are interviewed.

An overview of the report structure may be found on the course web-site. Two particularly important elements are the high level decomposition of the design problem into tasks, assigned to individuals, and the technical documentation of each of the design tasks (managed by the individual to whom they have been assigned).

Networked Communications Design Topic

Context and Objectives

The objective of this design project is to design a client (or clients) which can reliably recover a hidden message which exists on a server. Communication with the server is conducted via UDP. Each UDP message sent by the client issues a request for a single character of the text message, while each UDP message delivered by the server identifies the character which corresponds to a specific client request. In this project, three computers must run the client algorithm (or perhaps different client algorithms) in order to each recover the message held on the server. Once the message has been completely recovered on a computer it should report this event, and the ultimate performance is assessed in terms of the time taken for all three computers to report that they have recovered the message. The configuration is illustrated below:



Protocol released by Monday of Week 10

Algorithms implemented in C
- Using gcc under Linux or
- Microsoft Visual Studio under Windows

Computers may talk to each other

Details of the communication protocol implemented by the server will be released by Monday of Week 10. In the meantime, however, there are several things you should note:

1. The server creates a separate response queue for each client, and these response queues have a limited delivery rate – e.g., 1kB/s.
2. The server will not create more than one response queue for any given client IP address. Moreover, it will not destroy a response queue until all of its messages have been delivered.

3. The server has a single shared request queue, which is serviced episodically – e.g., once every 10 milliseconds, generating responses and passing them to the response queues. The service interval may, in general, vary over time, but successive service intervals are expected to be similar to within +/- 10%.
4. All queues have limited capacity, but the capacity is not something your client algorithm can know ahead of time. If a queue is full, new messages (request or response).
5. Apart from overflow, or inability to create a new response queue because three are still in progress, there are no other conditions which should corrupt a request or response.

Constraints

The client algorithm must be implemented entirely in C or C++, without reference to any external libraries other than the standard system and networking library functions. You may use GCC under Linux, or Microsoft Visual C++ under Windows.

It is perfectly allowable for the three client computers to communicate with one another, using any desired method, so long as such communication is implemented using C or C++, as with the rest of the client algorithm.

You may use your own laptops, if desired, rather than the computers in the lab.

Design Outcome Marks

The hard requirement for this project is that you must be able to recover the message completely, even if this takes a long time. This is reflected in the marking scheme. The second basis for the marking scheme is the time taken for all three computers to recover the message completely, without any errors. This is measured in terms of the average character rate; averaging may be performed over multiple trials. Finally, the marking scheme reflects the general desirability of an elegant design, where simpler approaches are preferred unless a complex solution brings clear benefits.

Your design outcome marks will be awarded as follows:

- Message recovered reliably by all three computers: (____/15)
- Message recovery time (full marks for X chars/s, X to be provided): (____/15)
- Design principles, elegance, etc.: (____/12)
- Clarity of code (must meet objectives first): (____/8)

Note: In the event that your design exceeds the message recovery speed target of **X** chars/s, bonus marks may possibly be awarded to compensate for less than perfect performance in other assessment criteria associated with this design topic.

Understanding Marks

- Understanding of system design, trade-offs and choices: (____/10)
- Understanding of individual sub-systems designed: (____/15)
- Understanding of how things could be further improved: (____/10)

Note: the first two items mentioned above must be supported by the draft report provided in the laboratory in Week 12. The draft report must contain at least a close approximation to the final source code. Lab notebooks and print-outs must provide remaining details by the time of marking. You cannot expect your assessor to look at your code directly, but print-outs (not a computer screen) of key sections of the code can be used together with your lab notebooks.