The University of Queensland
School of Information Technology and Electrical Engineering
Semester 1, 2014

COMS3200 – Assignment Two

Due: 2pm Friday May 23, 2014

Weighting: 16% of final grade

Introduction

This assignment has two parts: Part A and Part B:

- Part A addresses a variety of issues related to the behaviour and performance of the communication protocols.
- Part B is related to networking programming. In this part of the assignment students are required to develop the networked application with the behaviour described in Assignment 1 but this time using UDP for communication between its components.

This is an **individual assignment**. You may discuss aspects of the design, programming and the assignment specification with fellow students, but should not actively help (or seek help from) other students with the actual design and coding of the assignment solution. It is cheating to look at another student's code and it is cheating to allow your code to be seen or shared in printed or electronic form. You should note that submitted code will be subject to checks for plagiarism and collusion. If we detect plagiarism or collusion, formal misconduct proceedings will be initiated against you. A likely penalty for a first offence would be a mark of 0 for the assignment. If you're having trouble, seek help from the tutor or the lecturer – don't be tempted to copy another student's code. You should read and understand the statements on student misconduct in the course profile and on the school website: http://www.itee.uq.edu.au/index.html?page=138114

Part A (6%)

Q1 [1%]
Find the performance of the selective-repeat protocol working under the following assumptions:

- frames are 800 bytes long
- data rate is 10 Mbps
- propagation delay is 5 µsec/km
- 3 bits are allocated for the frame sequence number
- the distance is 5 km

Assume that no frames are lost. Assume that transmission time for ACK is negligible. Show your working by including all the drawings and calculations used to arrive at the solution.

Q2 [1%].
Find the performance of the Go-back-n protocol working under the following assumptions:

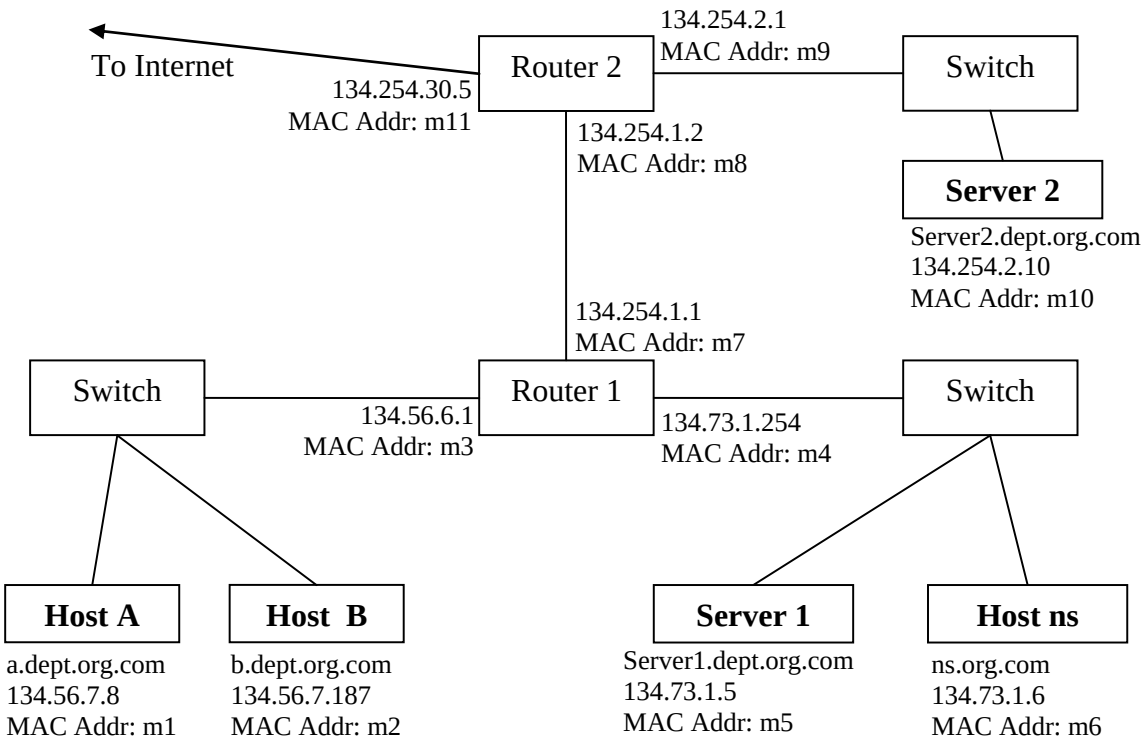
- frames are 800 bytes long
- data rate is 10 Mbps
- propagation delay is 5 µsec/km

- 3 bits are allocated for the frame sequence number
- the distance is 100 km

Assume that no frames are lost. Assume that transmission time for ACK is negligible. Show your working by including all the drawings and calculations used to arrive at the solution.

Q3 [1%]
Repeat the performance calculation from Q1 and Q2 with the assumption that 7 bits (not 3) are allocated for the frame sequence number.

Q4 [3%]
Consider the following internetwork, representing a subset of the org.com network:



Hosts have the names, IP addresses and (simplified) MAC addresses shown. *Host ns* serves as the authoritative DNS server for org.com. Each host is aware of its subnet mask, the IP address of the gateway (i.e. the particular router interface address), and the IP address of the DNS server. Assume that at the beginning of the communication exchange described below no host has ARP or DNS information cached (other than *Host ns* knowing DNS information for the whole domain), however once it acquires such information it will be cached. You may assume that the routers have complete ARP caches – i.e. they know the MAC addresses for all of the network interfaces on the networks they are directly connected to.

Consider a client process on host B sending a request to Server 1. The address (Server1.dept.org.com) is given to the client process as an argument. Assume that TCP is used at the transport layer for communication with the server. Please also note that communication with the DNS server uses UDP. Write down the **sequence of *frames* that are <u>received</u> by Router 1** (on any

interface) as this request is met. Do not show any other frames. Do not show frames transmitted by Router 1.

You should clearly specify the source and destination MAC addresses and, if the frame contains an IP packet, the source and destination IP addresses (or equivalent host name). You should specify the protocol(s) transmitted within the frame and give a brief description of the contents of the packet. For TCP segments, indicate if any TCP flags are set (SYN etc). You may assume that no transport or IP level fragmentation is necessary and that no packets are lost. You may also assume that the client request will be successful and a response will come from the server.

As an example of the level of detail required, the first 3 frames received by Router 1 are provided for you. Your answer should be a table with the columns shown below. You do not need to repeat the frames given below.

Source MAC address	**Dest. MAC address**	**Source IP address (or host)**	**Dest. IP address (or host)**	**Protocol(s)**	**Contents**
m2	* (Broadcast)	Not Applicable	Not Applicable	ARP	Who owns 134.56.6.1?
m2	m3	B	ns	DNS / UDP / IP	What's the IP address for Server1.dept.org.com?
m6	m4	ns	B	DNS / UDP / IP	IP address is 134.73.1.5
…					

Assessment Criteria for Part A

Provided your assignment is submitted using the submission instructions it will be marked as indicated in the assignment questions. The marks are allocated for correctness. Partial marks will be allocated for answers that are partially correct.

Part B (10%) - 100 marks

The aim of Part B of the assignment is for students to gain experience in network programming using socket level primitives for the UDP protocol. In this part of the assignment students are required to develop the networked application with the functionality described in Assignment 1 but this time using UDP for communication between the application's components. As UDP is connectionless you should scan the Assignment 1 specification for anything related to connections (e.g. requests to print message about "listening on a port" or 'unable to connect") and ignore them in Assignment 2. To this end, you are required to implement the Name Server, two clients (QueryClient) and two servers (LoansServer, CatalogServer) in a library loans system. The QueryClient send requests to the LoansServer and the CatalogServer.

Specification

You must implement all the components in Java using socket level primitives. All communication is to be UDP based.

Java Requirements

Your implementation must consist of four classes: `LoansServer`, `CatalogServer`, `NameServer` and `QueryClient` which will be in the files `LoansServer.java`, `CatalogServer.java`, `NameServer.java` and `QueryClient.java`. Each of these classes will contain a `public static void main(String args[])` method. You may use additional classes and Java files as needed but at a minimum your solution must include all of these files. Your Java classes should not be part of any package.

Other requirements

All communication is to be UDP based. All the communication has to be reliable and the reliability should be ensured by the communicating processes (at the application layer), i.e. the sender process needs to set timeout for an ACK arrival and retransmit the message if the timeout expires. The receiver has to acknowledge received messages. You are only allowed to use **one** port for sending and receiving UDP messages (one port per process), i.e. you cannot let the client contact the server on the server's UDP port, get the server to create a new UDP port and send a message back to the client. The port number should be allocated by the system, i.e. the process should not specify the port number (except for the Name Server). Multi-threading is allowed for implementation of the communication reliability, i.e. the sending process may create a separate thread that reacts to timeout expiry and retransmits the required packet. The thread uses the same port as the process that instantiated the thread.

As the local area network on which you test your application is reliable and does not lose packets it is necessary to simulate packet loss to test introduced reliability. Please simulate the packet loss by using a random number generator and deciding based on the generated value whether the packet is sent or not. The following pseudo-code shows how packet loss can be simulated:

```
generate uniform random number x from 0 to1;
    if(x>=0.5) transmit packet;
    set timeout
```

Hint

- You may wish to start from the UDP based client/server example available on Blackboard. You are free to use this code however you like.

Penalties that may be applied

- Code must be modified by marker to permit compilation and/or marking (-1 to -50 depending on the severity of the change required. Deduction is at the discretion of the course coordinator whose decision is final.)

Assessment Criteria

Provided your assignment is submitted following the instructions below, it will be marked according to the following criteria. You must pay careful attention to the details of any required behaviour. Part marks may be awarded for a given criteria if the specification is partially met. All marking will be performed in a UNIX environment, specifically moss.labs.eait.uq.edu.au and it is expected that your code will work in this environment. Note that some criteria can only be tested for if other criteria are met. You will need to demonstrate Part B if requested by the tutor.

QueryClient (22 marks)

- Code compiles successfully (2 marks)
- Coding style (neat code layout and code is commented) (2 marks)
- Client correctly deals with invalid command line arguments (1 marks)
- Client correctly prints out error message and exits with the correct status when unable to contact Name Server (2 marks)
- QC correctly looks up required Server information from the Name Server (2 marks)
- There are two (or more) QCs running in the application (4 marks)
- QC correctly sends requests to the LoanServer (3 marks)
- QC correctly sends requests to the CatalogServer (3 marks)
- QC correctly prints responses (3 marks)

Name Server (18 marks)

- Code compiles successfully (2 marks)
- Coding style (neat code layout and code is commented) (2 marks)
- Server correctly deals with invalid command line arguments (1 marks)
- Server correctly handles invalid messages (2 mark)
- Server correctly sends error messages when process is not registered (2 marks)
- Server correctly stores valid information without any error (2 marks)
- Server correctly responds to registration and lookup requests (5 marks)
- Server doesn't crash for communication cases (2 marks)

LoansServer (30 marks)

- Code compiles successfully (2 marks)
- Coding style (neat code layout and code is commented) (2 marks)
- Server correctly deals with invalid command line arguments (1 marks)
- Server correctly handles invalid messages (2 mark)
- Server correctly prints out error message and exits with the correct status when unable to contact Name Server (2 marks)
- Server correctly handles valid requests (12 mark)
- Server correctly prints messages to standard error (7 marks)
- Server doesn't crash for communication cases (2 marks)

CatalogServer (30 marks)

- Code compiles successfully (2 marks)
- Coding style (neat code layout and code is commented) (2 marks)
- Server correctly deals with invalid command line arguments (1 marks)
- Server correctly handles invalid messages (2 mark)
- Server correctly prints out error message and exits with the correct status when unable to contact Name Server (2 marks)
- Server correctly handles valid requests (12 marks)
- Server correctly prints messages to standard error (7 marks)
- Server doesn't crash for communication cases (2 marks)

Submission Instructions

The assignment (Part A and Part B) is due at 2pm on Tuesday May 23 and should be submitted as a zipped file through Blackboard. You are advised to keep a copy of your assignment.

Late Submission
Late submission will be penalized by the loss of 10% of your assignment mark per working day late (or part thereof). In the event of exceptional personal or medical circumstances that prevent on-time hand-in, you should contact the lecturer and be prepared to supply appropriate documentary evidence (e.g. medical certificate). Late submissions must be submitted by email to the lecturer or tutor (Blackboard submission will be unsuccessful, i.e., will not be recorded properly).
.

Modifications to Part A and Part B Requirements

Note: it is possible that there are inconsistencies in the above requirements and/or that not all details have been specified. Please ask if you unsure of the requirements. Please monitor your email, the course newsgroup, or the course website for clarifications and/or corrections to the above information. It will be assumed that students see such email or postings by the end of the next business day. Requirements changes/clarifications emailed and/or posted by one of the teaching staff before 4 pm Monday May 19 are considered to be part of the assignment requirements.

Academic Merit, Plagiarism, Collusion and Other Misconduct
You should read and understand the statement on academic merit, plagiarism, collusion and other misconduct contained within the course profile and the School website. You should note that this is an **individual assignment**. **All submitted source code will be subject to plagiarism and/or collusion detection.** Work without academic merit will be awarded a mark of 0.

Assignment Return: Assignment feedback arrangements will be advised later.