

COMP3150 – Computer Networks

Group Project Topics 2016

Deadlines and Important Dates

Tuesday November 15th, 2016 Checkpoint 1 (2 marks)

Expected: objectives established,
basic code drafted and possibly some screenshots.
Analysis of project must be totally completed.

Tuesday November 22nd, 2016 Checkpoint 2 (3 marks)

Expected: Good progress made based on established objectives. Amount of work remaining must be elaborated. Functioning of group members must also be described. Coding almost completed. Working program components.

Tuesday November 29th, 2016

Presentations. Hard copies of presentation, report and peer evaluation sheets are due at the start of the session. Also submit an electronic copy of all documents to the lecturer via My Learning. A multimedia projector would be provided. Each group should have a laptop.

Group Project Information

Each group is to consist of at most 4 members. A summary of the planned project work and group member names are to be submitted to the lecturer within two weeks of receiving the group project. Each project is to be typewritten and presented in a scholarly manner with table of contents, abstract, proper formatting etc. There would be a 8-10 minute presentation of projects. Projects should have a minimum of 15 pages of content. The content does not include the main programming code, but can include important snippets which are also explained. All program listings must be in the document (Appendices). All important lines of code must be well documented. Groups will be asked to do a demo of any software developed.

Choose one of the projects below. (Other projects are possible - students can do a proposal and forward to the lecturer for approval).

PROJECTS

1. Write software to implement some of the basic commands of the FTP protocol.
Client/Server principles must be illustrated.

2. Write software to implement some of the basic commands of the Telnet protocol. Client/Server principles must be illustrated.
3. Write a UDP program of your choice using client/server principles. Program “reliability” into your software at the application layer.
4. Write programs to simulate **rdt 2.0** as seen in the text and discussed during lectures. Be sure to document the principles behind your simulation. [Client/Server communication must be used.]
5. Write program that works like a gateway to repackage an IPV4 datagram into an IPV6 datagram. Also demonstrate how an IPV6 datagram can be changed into the IPV4 version.
Note: For extra fields in IPV6, you may include your own relevant data.
6. Write TCP/UDP programs to compare the performance of TCP and UDP. Comment on the statistics collected. Also, indicate how many UDP segments (if any) were dropped by the network. Clearly indicate the role of client and server in your report.
7. Develop an educational client/server game that uses threads and the TCP protocol. Illustrate how clients can be blocked and tracked at the server. The complexity of the games has to be discussed with the lecturer.
8. Research how Python code can be used in network programming. Develop a short question/answer game in several versions. The versions are: Java TCP, Java UDP, Python TCP, Python UDP. Produce a user manual to show how the different programs vary. Document the differences between the Java and Python versions for each program (This can be normal documentation or via an animated file).

Marking Scheme

Layout	2
Proper user manual	8 (How is the software to be executed? Screenshots to be included)
Documentation	2
Fully working program demo	12
(State program objectives before the practical demo is done)	
Check Points (2, 3)	5
*Program design	5
Report Overall Quality	11
Presentation	5

TOTAL**50 MARKS**

*Program design refers to how the software is laid out with try..catch blocks, checking error conditions, proper use of methods etc.

Note: Individual assessments may be done on the Group projects by the lecturer.