Data Communication and Net-Centric Computing (COSC1111/2061)

Assignment 2 (S2 2015)

Aims

- To investigate the real-world data communication.

Instructions

- This assignment contributes 25% towards your overall result of DCNC with a bonus of up to 5 additional marks.
- You are expected to work in a group (maximum 4 members) of your choice:
  - Individual submission would be acceptable;
  - High standard will be expected to group works, especially for the report part.
- Failure to comply with the instructions may lead to mark deduction.

Submission

- Date of submission: 11:59PM, Friday, 9th October 2015.
- The assignment is to be submitted via Weblearn.
- Only one submission for each group. Please add a Readme.txt file stating all team members names, student IDs and contribution % of each person. Please put the report, and code (Tasks 2.1, 2.3) into one folder and compress it to a single compress file for submission.
- The report should be in PDF format following the word template provided.

Late submission policy:

- There will be a late submission period of 5 working days for this assignment.
- Late submissions will incur a penalty of 10% per day, unless prior arrangements are made with the lecturer or the head tutor regarding an extension.
- Submissions received after the late submission period expires will not be assessed and will hence receive zero mark.

Each group chooses one of the following tasks (either Task 1 or Task 2):

Task 1

Task 1.1 Investigate code division multiple access (CDMA) and wireless communications. (20 marks)

The following table shows the development of mobile communications:

Generation	Functions	Principles
1G	voice-oriented analog technology	Frequency Division Multiple Access
2G	voice-oriented digital technology	Time Division Multiple Access
3G	high-speed voice-oriented integrated with data services	Code Division Multiple Access
4G	IP-based: voice, data, multimedia service	Code Division Multiple Access

You are required to ***write a technical report*** using the given template, discussing the following points:

- The functionalities of different generations of wireless communication technology.
- How CDMA, also termed as “cellular technology”, works.
- How the communication is seamlessly maintained while users are moving.
- How “roaming” service is implemented (understand the cost of roaming).

Note that you need to discuss technical details of the functionalities and discuss each generation of cellular technology using concepts such as bandwidth, data rate, etc. The report in Task 1.1 must not exceed 6000 words (prefer 4 pages).

Task 1.2 Wireless network scanner **(5 marks)**

Perform a War-Walking activity at RMIT and at any other venue of your choice (please specify) with your laptop, sniffing the details of available wireless networks. Record the wireless networks available in two different spots and investigate:

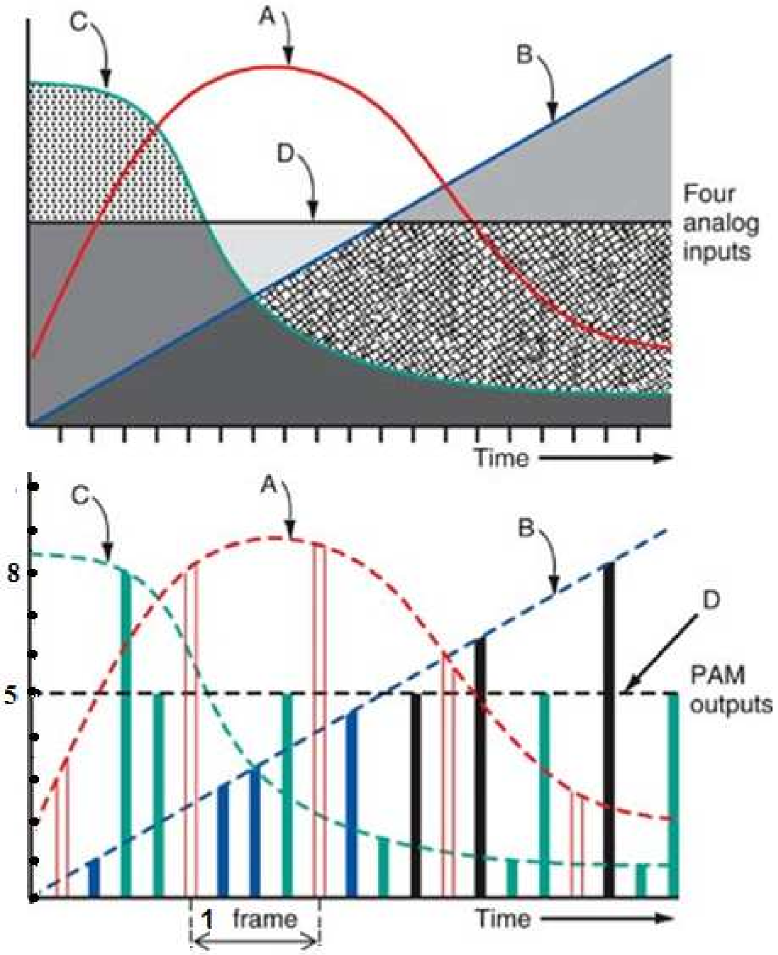
- Dual band WiFi and its advantages.
- Security situation of the discovered wireless networks.

Write a report (1 page) answering the above questions including screenshots of scanning activities.

Task 1.3 Bonus Component (5 marks) (there is no word limitation on the length for the bonus part)

Take a 4-Channel TDM Multiplexing shown in the following figure as an example, if a multiplexer combines the four 100 kbps source using a time slot of 1 byte; 1 synchronising bit is added to each frame.

- Draw a picture to describe the TDM.
- What is the frame rate, frame duration and the number of bits in each frame.
- Discuss and calculate the details of digitizing the analog signals using Pulse Code Modulation (PCM) encoding (hint: sampling, quantization and encoding should be discussed).



References for Task 1:

1. Chapter 8, 11, 14, edition 9th Data and Computer Communications, William Stallings.
2. Chapter 8, 9 (9.6), 10, 17, edition 10th Data and Computer Communications, William Stallings.
3. Lecture Notes Chapter 5 FDM (page 5.6-5.10) and TDM (page 5.12-5.18).
4. DCNC Workshop 3, week 8.

Task 2

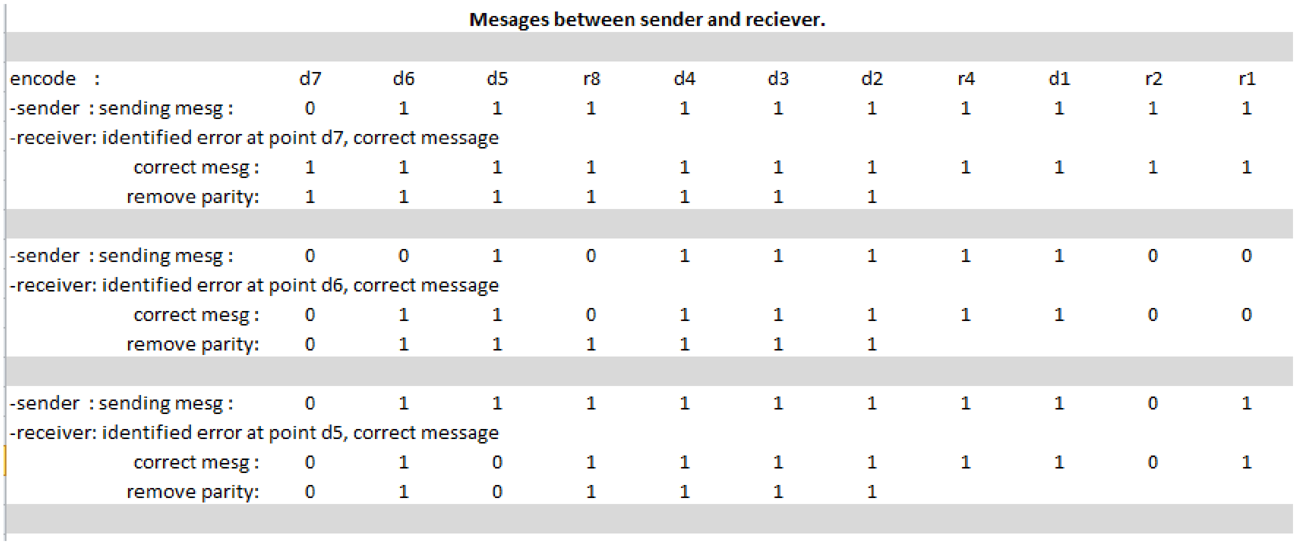
Task 2.1 Simulate the error detection hamming code at Client and Server sides using network communication technology, includes: (20 marks)

- Send messages from client to server (one client and one server);
- Send messages from client to client through a server (three tier architecture);
- Send messages from client/server to another client/server.

You may use the attached client and server Java code as start-up. However, you have the freedom to use any programming languages, such as C/C++, Java, or others you prefer.

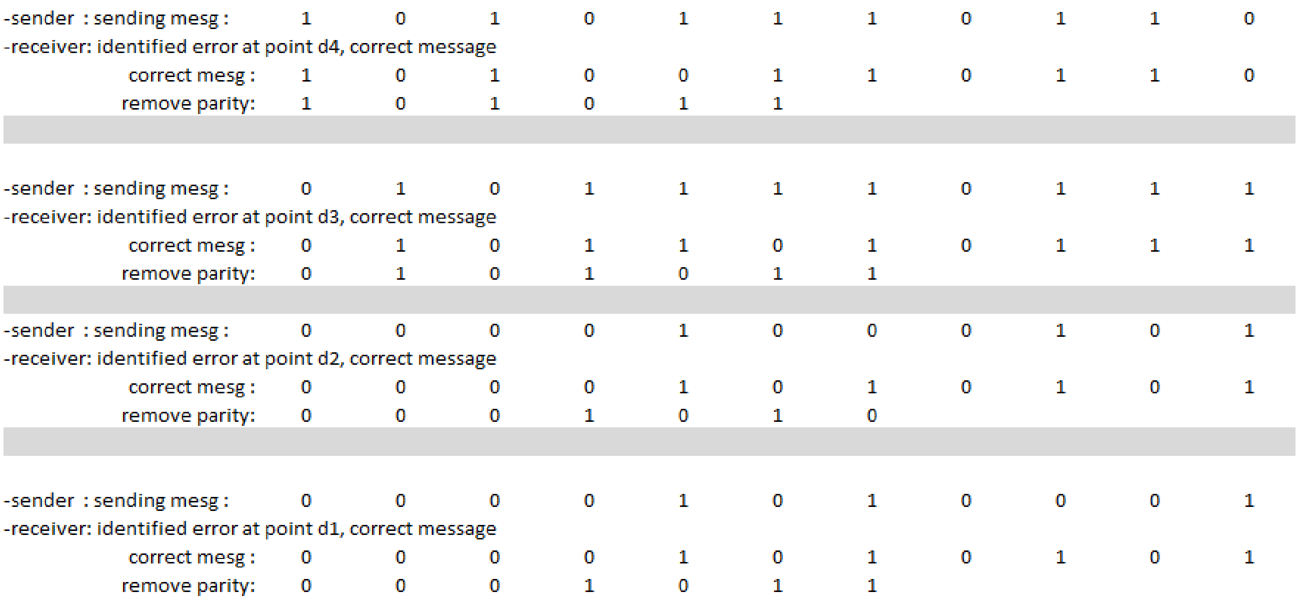
What to submit:

- The source code including inline comments (in separate files).
- Describe how the overall program works including screenshots to capture a complete successful run (client & server). The report must not exceed 5 pages (prefer 4 pages).
- You may use the trial version of Snagit or similar tools to capture the screenshots of the process as part of the submission, (available at https://www.techsmith.com/download/snagit/).



Mesages between sender and reciever.

encode :	d7	d6	d5	r8	d4	d3	d2	r4	d1	r2	r1
-sender : sending mesg :	0	1	1	1	1	1	1	1	1	1	1
-receiver: identified error at point d7, correct message											
correct mesg :	1	1	1	1	1	1	1	1	1	1	1
remove parity:	1	1	1	1	1	1	1				
-sender : sending mesg :	0	0	1	0	1	1	1	1	1	0	0
-receiver: identified error at point d6, correct message											
correct mesg :	0	1	1	0	1	1	1	1	1	0	0
remove parity:	0	1	1	1	1	1	1				
-sender : sending mesg :	0	1	1	1	1	1	1	1	1	0	1
-receiver: identified error at point d5, correct message											
correct mesg :	0	1	0	1	1	1	1	1	1	0	1
remove parity:	0	1	0	1	1	1	1				



-sender : sending mesg :	1	0	1	0	1	1	1	0	1	1	0
-receiver: identified error at point d4, correct message											
correct mesg :	1	0	1	0	0	1	1	0	1	1	0
remove parity:	1	0	1	0	1	1					
-sender : sending mesg :	0	1	0	1	1	1	1	0	1	1	1
-receiver: identified error at point d3, correct message											
correct mesg :	0	1	0	1	1	0	1	0	1	1	1
remove parity:	0	1	0	1	0	1	1				
-sender : sending mesg :	0	0	0	0	1	0	0	0	1	0	1
-receiver: identified error at point d2, correct message											
correct mesg :	0	0	0	0	1	0	1	0	1	0	1
remove parity:	0	0	0	1	0	1	0				
-sender : sending mesg :	0	0	0	0	1	0	1	0	0	0	1
-receiver: identified error at point d1, correct message											
correct mesg :	0	0	0	0	1	0	1	0	1	0	1
remove parity:	0	0	0	1	0	1	1				

Task 2.2 Analyze network protocol (5 marks)

Use your favourite browser to visit the following webpage: http://www.rmit.edu.au/

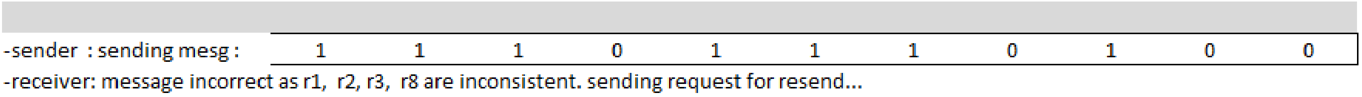
Use any network protocol analyser to capture all HTTP packets and investigate:

- Find the request HTTP packet number & request line for RMIT global-brand.jpg image.
- Find out the HTTP version number used by both client & server.
- Which languages does the browser support?
- When was the received webpage last updated?
- What is the total size, in bytes, of the object (image file) downloaded?

Write a report (1 page) answering the above questions including screenshots of your analyser tool.

Task 2.3 Bonus Component (5 marks) (there is no length limitation for the bonus part)

Depending on the error received, to correct the error or request for retransmission.



```
-sender  : sending mesg :     1     1     1     0     1     1     1     0     1     0     0
-receiver: message incorrect as r1,  r2, r3,  r8 are inconsistent. sending request for resend...
```

References for Task 2:

1. Chapter 22, Appendix B3, B4, edition 9th Data and Computer Communications, William Stallings.
2. Chapter 2, 15, Appendix B3, B4, edition 10th Data and Computer Communications, William Stallings.
3. Lecture Notes Chapter 3, (page 3.23-3.27).
4. DCNC Workshop 1, week 4; Workshop 2, week 6.

Appendix for task 2:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

/**
 * A TCP server that runs on port 9090.  When a client connects, it
 * sends the client the current date and time, then closes the
 * connection with that client.
 */

public class DateServer {

    /**
     * Runs the server.
     */
    public static void main(String[] args) throws IOException {
        ServerSocket listener = new ServerSocket(9090);
        try {
            while (true) {
                Socket socket = listener.accept();
                try {
                    PrintWriter out =
                        new PrintWriter(socket.getOutputStream(),
true);
```

```
                    out.println(new Date().toString());
                } finally {
                    socket.close();
                }
            }
        }
        finally {
            listener.close();
        }
    }
}
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

/**
 * A TCP server that runs on port 9090.  When a client connects, it
 * sends the client the current date and time, then closes the
 * connection with that client.
 */

public class DateServer {

    /**
     * Runs the server.
     */
    public static void main(String[] args) throws IOException {
        ServerSocket listener = new ServerSocket(9090);
        try {
            while (true) {
                Socket socket = listener.accept();
                try {
                    PrintWriter out =
                        new PrintWriter(socket.getOutputStream(),
true);
                    out.println(new Date().toString());
                } finally {
                    socket.close();
                }
            }
        }
        finally {
            listener.close();
        }
    }
}
```