CS5352.0501/0502, Distributed Computing, Summer I, 2009
Assignment 2

Issued: 06/22/2009
Due: 06/29/2009

Please hand in the entire assignment through ftp, same as for Assignment 1. For each programming problem, hand in

- Your source program;
- The execution results;
- Put a copy of your program through ftp using username cs5352 and password 04DistComp09 to the directory

  ftp://glaciers.cs.txstate.edu/pub/teaching/cs5352/sum-I-09/hw2/your-lastname

  For each program, you should have a separate file that specifies how to compile and execute your program.

  Please note that ftp upload can only be done in normal FTP service (not SSH FTP) and can only be performed from computers on campus.

Each programming problem must be well-commented. It should include a brief explanation about what you have done and special techniques/tricks you have used.

1. $(5 + 5 + 10 = 20$ pts) This problem is pertaining to the NTP protocol.

   (1) Explain the meaning of offset o_i between two NTP servers.

   (2) Why the values of t and t' cannot be measured? Why we can obtain the value of $t + t'$ accurately?

   (3) The propagation delay d_i plays a significant role in the accuracy of NTP protocol. Explain using your own words why.

2. $(12 + 13 = 25$ pts) The problem is pertaining to the concepts of logical time and clocks.

   (1) It was stated in class that for the vector timestamps, property P_3 (if $e \neq e'$ then $L(e) \neq L(e')$) may not always be true. Do you agree with this statement? Please explain your conclusion.

   (2) For the diagram shown in Fig. 1, assume that each process starts with a vector $(0, 0, 0)$. Process p_1 always increases its component by 1, process p_2 always increases its component by 2, and process p_3 always increases its component by 10. Write down the complete vector timestamps for all the events in the diagram.

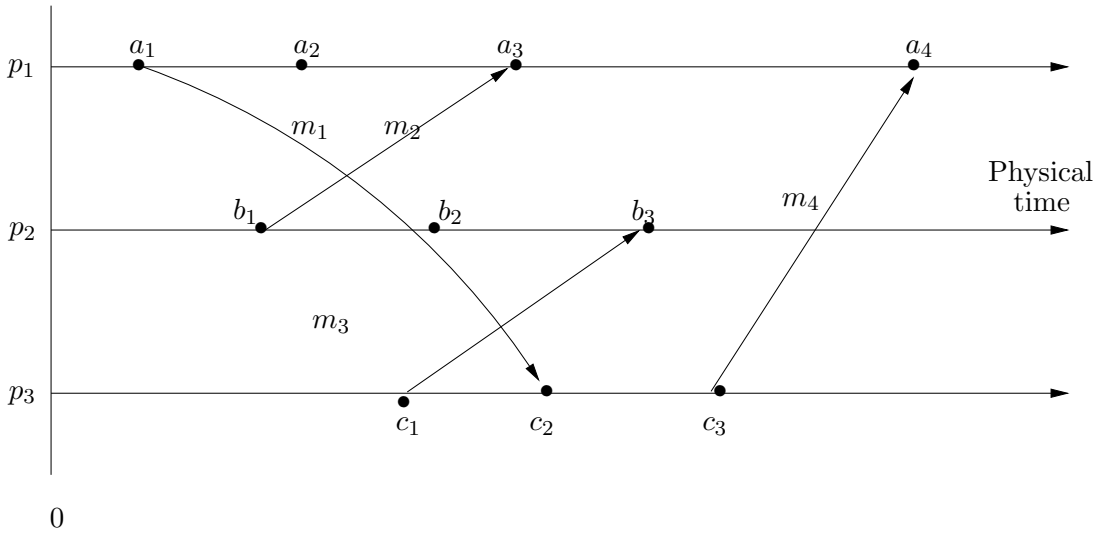

Figure 1: Three processes for Problem 2 and 3

3. (25 + 30 = 55 pts) Modify the Java UDP example given in Chapter 4, p.138-139 as follows.

(1) The client will iterate in a loop. During each iteration, it will get a character string from standard input and send it to the server. The server will also iterate in a loop. During each iteration it will receive a character string and send it back to the client.

(2) In this second modification, the client will still send a character string to the server. However, that character string is the name of a file on the server's host. After receiving that file name, the server will try to open the file and send each line of the file back to the client. If the server cannot open the file, it will issue an error message back to the client. The client should just print each line of replies on its standard output.

Notes: For this problem, if you are not already familiar with Java programming language, you have to read a Java textbook or read the JDK1.5.1 (or the latest version) manual about basic control (iteratios) and file I/O. The manual can be found from java.sun.com.