**CS5352.751/752, Distributed Computing, Summer I, 2020**

**Assignment 2**

Zebo Xiong, A04907051

Issued: 07/16/2020

Due: 07/23/2020

1. (5 + 5 + 10 = 20 pts) This problem is pertaining to the NTP protocol.
2. Explain the meaning of offset oi between two NTP servers.

We cannot get the real “o” between two servers’ clock (such as NTP server A and NTP server B, no matter which stratums they exist in).

However, an oﬀset “o-i”, which is an estimate of the actual oﬀset between the two clocks, can be calculated with other information such as T(i-3), T(i-2), T(i-1) and T(i)

1. Why the values of t and t′ cannot be measured? Why we can obtain the value of t + t′ accurately?

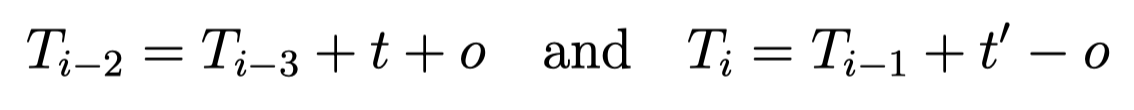
We cannot measured the t and t prime because we do not know the two servers’ offset in advance. Thus, we cannot capture the real send and receiving time point (Otherwise the t and t prime can be easily gotten by the server time difference).

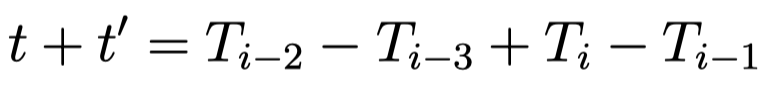
However, we can get the time points on each server (such as server A and sever B).

A close up of a logo

Description automatically generated

Thus, we get the four time points during the procedure-call or symmetric mode. Then we have below equation:



After we combine the two equations we get: t+t’ = T(i) + T(i-2) – T(i-3) - T(i-1), or, 

1. The propagation delay di plays a significant role in the accuracy of NTP protocol. Explain using your own words why.

Without calculating the propagation delay “di”, we cannot start estimating the offset between two server.

Once we confirm the “di”, we can get “oi” which is the half of “di”. Afterward, we can find the range of the real “o”.

It shows that “oi” is an estimate of o, and di” is a measure of the accuracy of this estimate on “oi”.

Additionally, NTP employed “data filtering algorithm” which use eight continuous pairs < oi , di >. The value oi of that corresponds to the minimum value di is chosen as “oi” 🡪 The whole process is working with propagation delay “di”

Also, NTP use peer-selection to synchronize peer severs. Without the propagation delay information for each sever, this step will not be processed properly.

1. (12 + 13 = 25 pts) The problem is pertaining to the concepts of logical time and clocks.
2. It was stated in class that for the vector timestamps, property P3 (if e ̸= e′ then L(e) ̸= L(e′)) may not always be true. Do you agree with this statement? Please explain your conclusion.

Perperty P3 may not always be true for “Lamport’s method without process ID”. As shown in below figure:

A close up of a map

Description automatically generated

The event “a” and event “e” has the same Lamport timestamp. However the property is always true for vector timestamps.

For the “Vector clocks”, each process content the vector with size N – which is the number of total processes.

Each process actually carries all the other processes’ timestamp. It means that each process has the whole picture of the global state.

According to the VC mechanism, not only the sender’s vector timestamp will be updated, but also the receiver’s vector. Thus when the events are different, the Vectors will also be different.

Reference from lecture note:

A close up of a map

Description automatically generated

1. For the diagram shown in Fig. 1, assume that each process starts with a vector (0, 0, 0). Process p1 always increases its component by 1, process p2 always increases its component by 2, and process p3 always increases its component by 10. Write down the complete vector timestamps for all the events in the diagram.

A picture containing sitting, map, photo, table

Description automatically generated

a1 🡪 (1,0,0)

b1 🡪 (0,2,0)

c1 🡪 (0,0,10)

a2 🡪 (2,0,0)

b2 🡪 (0,4,0)

c2 🡪 (1,0,20)

a3 🡪 (3,2,0)

b3 🡪 (0,6,10)

c3 🡪 (1,0,30)

a4 🡪 (4,2,30)

1. (25 + 30 = 55 pts) Modify the Java UDP example given in Chapter 4, p.152-153 as follows.
2. 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.

Compile: javac UDPServer\_1.java

Javac UDPClient\_1.java

Run server: java UDPServer\_1

Run client: java UDPClient\_1

The server screen shot:

A screenshot of a cell phone

Description automatically generated

The client screen shot:

A screenshot of a cell phone

Description automatically generated

1. In this second modification, the client will still send a character string to the server. However, that character string is the name of a cam- mand. After receiving that cammand name, the server will try to excute that command and send output of the command back to the client. If the server cannot excute the command, it will issue an error message back to the client. The client should just print each line of replies, plus the IP number of replying computer on its standard output.

Compile: javac UDPServer\_2.java

javac UDPClient\_2.java

Run server side: java UDPServer\_2

Run client side: java UDPClient\_2 ls localhost

The server side screen shot:

A screenshot of a social media post

Description automatically generated

The client side screen shot:

A screenshot of a cell phone

Description automatically generated

Notes: You should implement and test the programs on the department Linux machines. The client/server should be able to run on the same computer, or on different computers. For this problem, if you are not already familiar with Java programming language, you have to read a Java textbook or read the JDK1.8.0 (or the latest version) manual about basic control (iteratios) and file I/O. The manual can be found from <http://www.oracle.com/technetwork/java/index.html>