CSCI/ECEN 5673: Distributed Systems
Spring 2022

Homework 1 Solutions

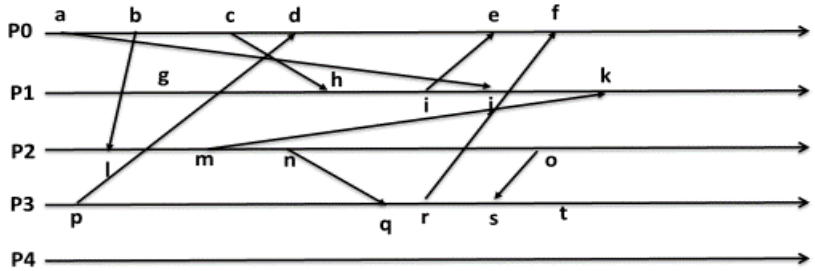
1. Read the paper titled *The Rise and Fall of CORBA* by Michi Henning.
   https://cacm.acm.org/magazines/2008/8/5336-the-rise-and-fall-of-corba/fulltext

   Suppose you are tasked with designing and implementing a middleware service that would be used by a large number of distributed systems developers. What lessons from CORBA development effort would you incorporate in developing this middleware service?

   These are mainly discussed under the title Can We Learn from the Past? In the paper. Four suggestions are provided:

   - Standards consortia need iron-cast rules to ensure that they standardize existing best practice.
   - No standard should be approved without a reference implementation.
   - No standard should be approved without having been used to implement a few projects of realistic complexity.
   - To create quality software, the ability to say "no" is usually far more important than the ability to say "yes".

2. Consider the following figure that shows five processes ***(P0, P1, P2, P3, P4)*** with events ***a, b, c,*** ... and messages communicating between them. Assume that initial logical clock values are all initialized to ***0***.



a) **[15 Points]** Provide logical clock (C) values of each event shown.

a: 0
b: 1
c: 2
d: 3
e: 5
f: 7
g: 0
h: 3
i: 4
j: 5
k: 6
l: 2
m: 3
n: 4
o: 5
p: 0
q: 5
r: 6
s: 7
t: 8

b) **[15 Points]** Provide vector clock (V) values of each event shown.

a: [10000]
b: [20000]
c: [30000]
d: [40010]
e: [53010]
f: [63330]
g: [01000]
h: [32000]
i: [33000]
j: [34000]
k: [35000]
l: [20100]
m: [20200]
n: [20300]
o: [20400]
p: [00010]
q: [20320]
r: [20330]
s: [20440]
t: [20450]

c) **[10 Points]** Identify two events *ai* and *aj* to show that C(*ai*) < C(*aj*) does not necessarily imply *ai* → *aj*.

g, l
p, b

There are many such valid events.

d) **[10 Points]** Assuming P0 < P1 < P2 < P3 < P4, provide a total ordering of all events constructed from the logical clock C. Is this total order unique, i.e., is it possible to construct a different total order than the one you constructed?

a,b,g,p,c,l,d,h,m,i,n,,e,j,o,q,k,r,f,s,t

In general, the total order is not unique. For example, events b and g can be swapped with one another for another total ordering. However, once we fix the criteria for resolving ties, i.e. how to order the events that have the same logical clock values, the total order becomes unique.

e) **[10 Points]** Suppose process P4 sends a message m (send event is *aa* and the corresponding receive event is *bb*). Show how *aa* → *t*, *aa* → *o*, and *aa* and *b* are concurrent. Identify an event a_i ($a_i \neq aa$) such that a_i → *bb*.

bb should occur on P2 anywhere after event l and before event o

aa→bb (send and receive events)
bb→o (events on same process, P2)
So aa→o (transitive property)

o→s (send and receive events)
s→t (events on same process, P3)
So aa→t (transitive property)

Also aa and b are concurrent in this case.

Finally, l→bb (events on same process, P2)

3. Browse the NTP project webpage (*http://www.ntp.org*).

(a) [15 Points] Explain how NTP computes filter dispersion.

Grade for completion. Clock synchronization from the NTP website with graphs is one possible answer.

(b) [15 Points] Is NTP still at a security risk[yes/no]? If no, explain how NTP is being protected from vulnerabilities? If yes, what are some best practices that one can use while using NTP that might reduce the impact of security risks?

Yes, NTP is still at a security risk. Many instances of spoofed NTP servers and NTP abuse lead to DDoS attacks. There is also a possibility of replay attacks due to lack of synchronization.

Best Practices:
1) Presence of three or more time sources
2) In areas with more devices to be synchronized, stratum 2 or stratum 3 servers can be used as secondary servers.
3) Set up internal NTP service on the latest revision of stable code and also standardizing its use.
4) Try using public NTP servers for external hosts.
5) Try setting up private NTP severs or your own NTP hierarchical service and standardize that to UTC time.
6) Restrict commands on standard stratum servers.

4. A clock is being synchronized with a time server using the Cristian's probabilistic clock synchronization algorithm. Request send time is 4:20:15.200, the server time is 4:19:10.800 and the time when the response is received is 4:20:15.300

a) **[5 Points]** What is the client's estimate of the server time?

Elapsed time: 4:20:15.300 – 4:20:15.200 = 0.100
Client's estimate of the current server time: 4:19:10.800+0.5 * 0.100
= 4:19:10.850

b) **[5 Points]** Is the client's clock running faster or slower than the server clock? What will the client do after estimating the server time?

The clients clock is faster than the servers clock.
The client will increase the timer interrupt interval to make its clock tick slower, which in turn will result in getting its clock synchronized with the server's clock.