Northeastern University – Silicon Valley

CS 6650 Scalable Dist Systems

Homework #3 [100 points]

INSTRUCTIONS: ***Please provide clear explanations in your own sentences, directly answering the question, demonstrating your understanding of the question and its solution, in depth, with sufficient detail. Submit your solutions [PDF preferred]. Include your full name. Do not email the solutions.***

Study **Chapter 14 and 15 from** Coulouris Book Clocks and Time, Global States, Consensus

Answer the following questions using explanation and diagrams as needed. No implementation needed.

1. 14.4 [5 points]

A client attempts to synchronize with a time server. It records the round-trip times and timestamps returned by the server in the table below.

Which of these times should it use to set its clock? To what time should it set it? Estimate the accuracy of the setting with respect to the server's clock. If it is known that the time between sending and receiving a message in the system concerned is at least 8 ms, do your answers change?

Round-trip (ms)	*Time (hr:min:sec)*
22	*10:54:23.674*
25	*10:54:25.450*
20	*10:54:28.342*

The timestamp with the least round-trip time must be chosen. In this case, it would be 20 ms.

Time to be set = 10:54:28:342 + (00:00:00:020 / 2)
= 10:54:28:352

Accuracy = t_round / 2 = 20 ms / 2
= ± 10 ms.

When min time = 8 ms,

Accuracy = 20 - 2* 8 / 2
= ± 2 ms.

2. 14.7 [10 points]
   An NTP server B receives server A's message at 16:34:23.480, bearing a timestamp of 16:34:13.430, and replies to it. A receives the message at 16:34:15.725, bearing B's timestamp, 16:34:25.7. Estimate the offset between B and A and the accuracy of the estimate.

Accuracy di = Ti–2 – Ti–3 + Ti– Ti–1

= 16:34:23:480 - 16:34:13:430 + (16:34:15:725 - 16:34:25:7)

= 00:00:10:050 + (-00:00:09:975)

= 00:00:00:075.

Offset oi = Ti–2 – Ti–3 + Ti–1 – Ti2

= 16:34:23:480 - 16:34:13:430 + 16:34:25:7 - 16:34:15:725

= 00:00:10:050 + 00:00:09:975 / 2

= 00:00:20:025 / 2

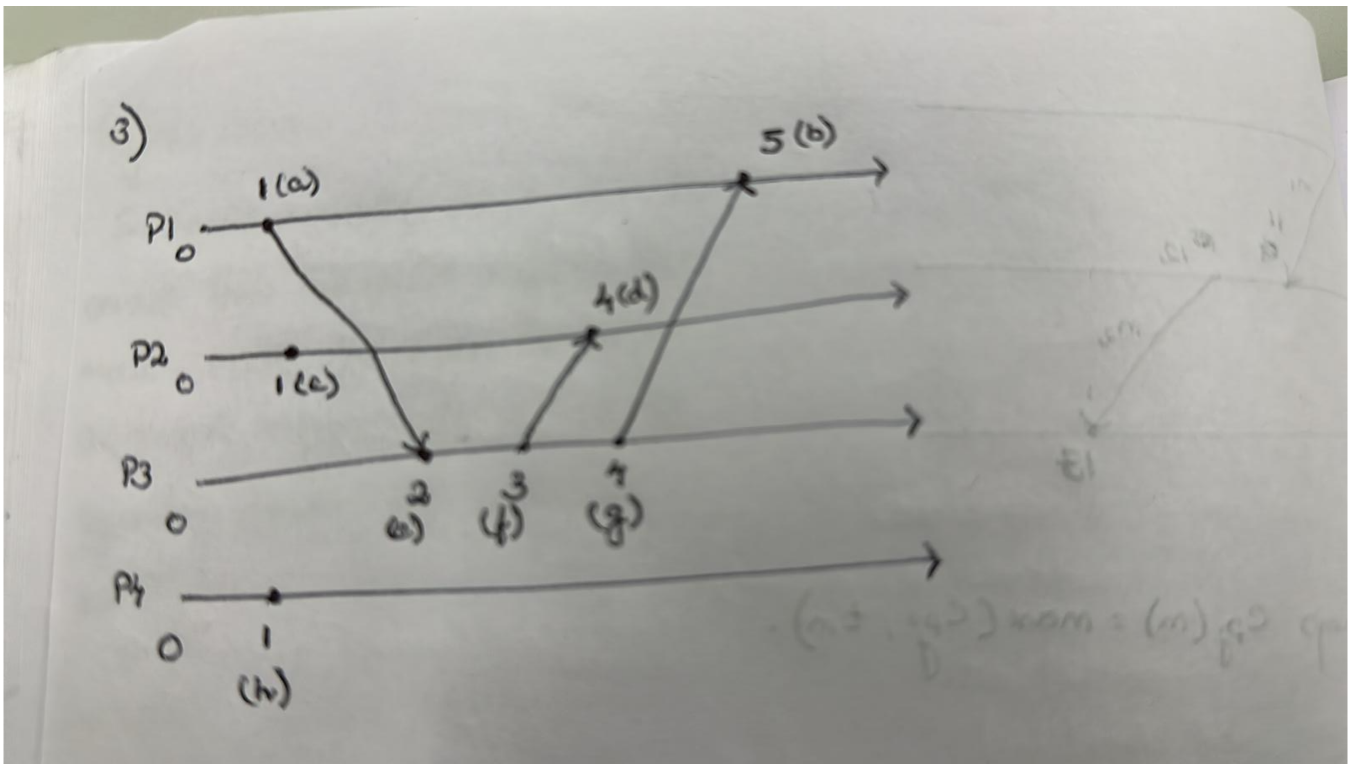
= 00:00:10:013.

3. A system of four processes, (P_1, P_2, P_3, P_4), performs the following events: [25 points]

   a. P_1 sends a message to P_3 (to event e).
   b. P_1 receives a message from P_3 (from event g).
   c. P_2 executes a local event.
   d. P_2 receives a message from P_3 (from event f).
   e. P_3 receives a message from P_1 (from event a).
   f. P_3 sends a message to P_2 (to event d).
   g. P_3 sends a message to P_1 (to event b).
   h. P_4 executes a local event.

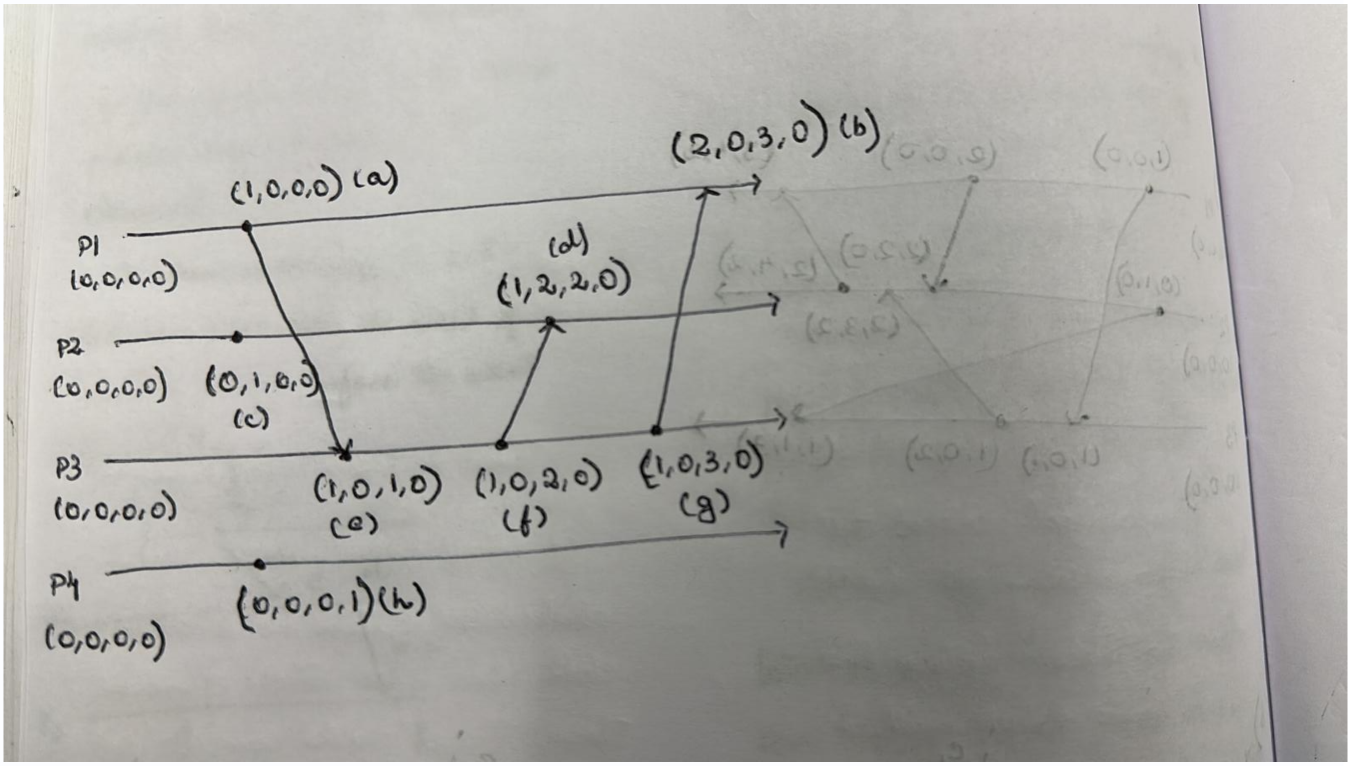
   When taking place on the same processor, the events occur in the order listed.
   Assign Lamport timestamps to each event. Assume that the clock on each processor is initialized to 0 and incremented before each event. For example, event *a* will be assigned a timestamp of 1.

a. 1	b.	c.	d.
e.	f.	g.	h.



a. 1
b. 5
c. 1
d. 4
e. 2
f. 3
g. 4
h. 1

a. Assign vector timestamps to each event in question 2. Assume that the vector clock on each processor is initialized to (0,0,0,0) with the elements corresponding to (P_1, P_2, P_3, P_4). For example, event *a* will be assigned a timestamp of (1, 0, 0, 0).



b. Which events are concurrent with event *d*?
Events b, g and h are concurrent to d.

4. You are synchronizing your clock from a time server using Cristian's algorithm and observe the following times: [15 points]
    - timestamp at client when the message leaves the client: 6:22:15.100
    - timestamp generated by the server: 6:21:10.700
    - timestamp at client when the message is received at client: 6:22:15.250

- To what value do you set the client's clock?
  Time to set = t + t_round/2
  T_round = 6:22:15:250 - 6:22:15:100 / 2 = 0:00:00:075
  t = 6:21:10:700

  Time set = 6:21:10:775

- If the best-case *round-trip* message transit time is 124 msec (0.124 sec), what is the error of the clock on the client?
  Accuracy = t_round / 2
  = 124 / 2
  = 62 ms.

  Error would be ± 62 ms.

5. 14.14 [15 points]

Two processes *P* and *Q* are connected in a ring using two channels, and they constantly rotate a message *m*. At any one time, there is only one copy of *m* in the system. Each process's state consists of the number of times it has received *m*, and *P* sends *m* first. At a certain point, *P* has the message and its state is 101. Immediately after sending *m*, *P* initiates the snapshot algorithm. Explain the operation of the algorithm in this case, giving the possible global state(s) reported by it.

- ➔ P's state = 101
- ➔ P sends the message m to Q
- ➔ P then initiates the snapshot algorithm by sending the marker message.
- ➔ At P,
    - ◆ P records its state = 101.
    - ◆ P sends the marker to its channel PQ.
- ➔ At Q,
    - ◆ Receives the message m from Q. updates state to 102.
    - ◆ Q then receives the marker message in channel PQ.
    - ◆ Since Q has not recorded its state,
        - It records state of channel PQ = null
        - Q then records its own state. (Q = 102).
        - Q now sends the marker message to channel QP.
- ➔ At P,
    - ◆ Since P has recorded its state,
        - P records the state of the channel QP = {m}.

Thus, the global state can be recorded and would be reported as,

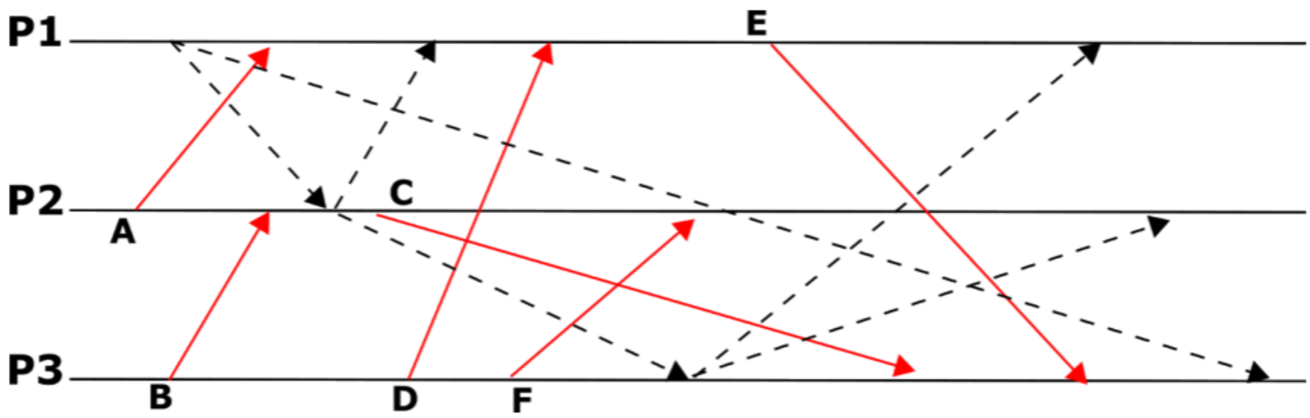
P = 101, Q = 102, PQ = null, QP = {m}

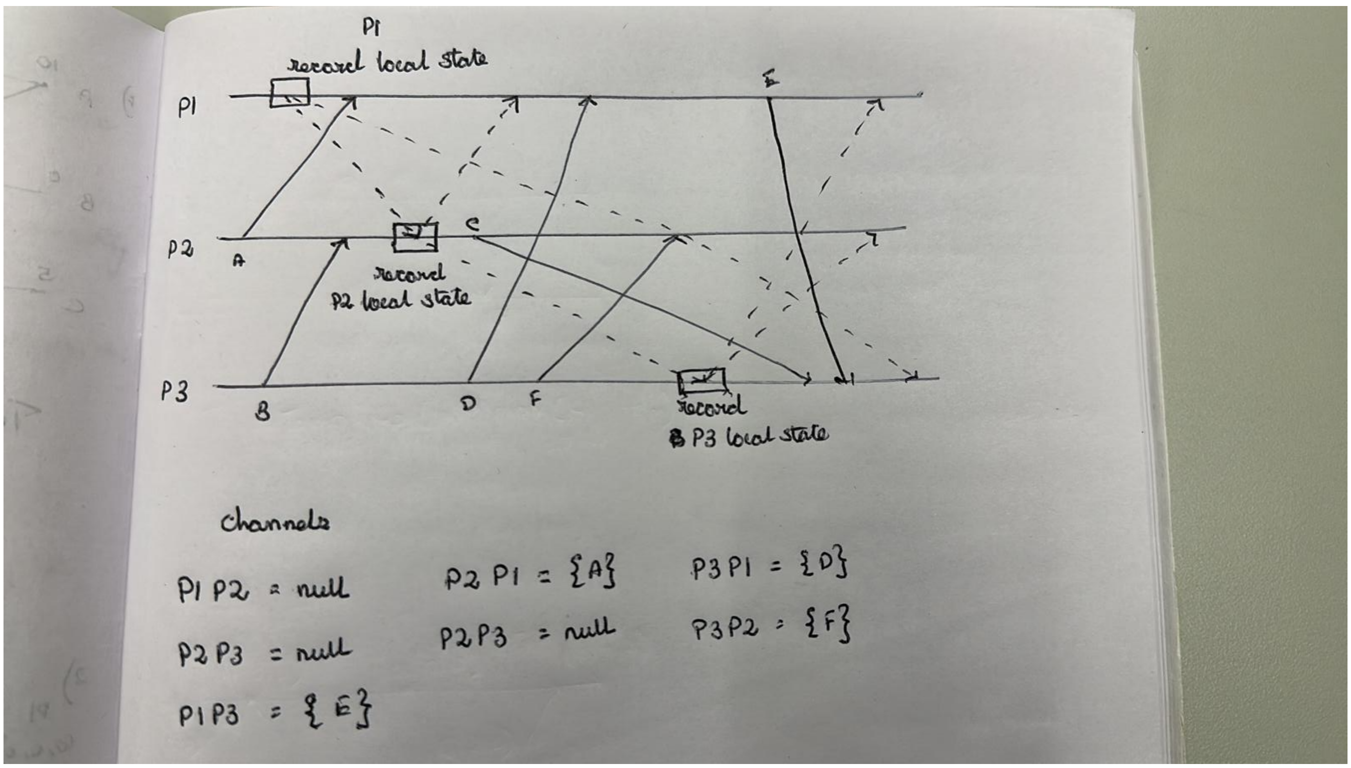
6. 15.9 [10 points]

Suggest how to adapt the bully algorithm to deal with temporary network partitions (slow communication) and slow processes.

During the temporary network partition, each subnet or partition can use the bully algorithm to elect the process with the largest ID in their respective partition as the leader. Once the network connections are re-established, the leaders of the partition can run the bully algorithm again to elect a new leader. (process with the largest ID).

7. Using Chandy-Lamport algorithm, show when each process records its local state (you can annotate the figure) and list the channel states for each process captured in the snapshot. Black dotted lines are marker messages. Red lines are messages (A to F) [20 points]


P1
P2
P3
E
C
A
B
D
F


P1
record local state
P1
P2
P3
A
B
C
D
F
E
record
P2 local state
record
P3 local state
Channels
P1 P2 = null
P2 P1 = {A}
P3 P1 = {D}
P2 P3 = null
P2 P3 = null
P3 P2 = {F}
P1 P3 = {E}