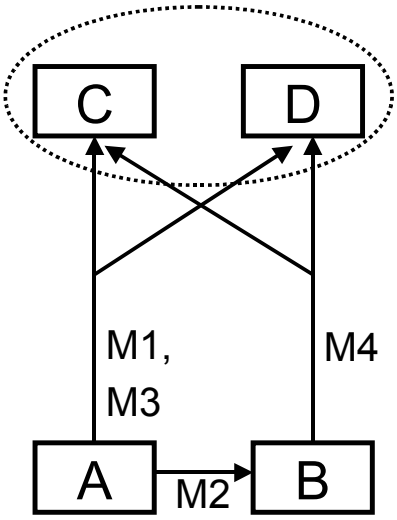
CptS 464/564 Sample Mid-term

Write the answers to the following questions. If you need more space, use the back of a sheet and clearly mention this on the front. (You generally should not; answers are not expected to take longer than the space provided, but extra length is not penalized so long as it is coherent and not unfocused rambling.) If you are not sure about some part of the question, state any assumptions you are making about the question, then answer it based on those assumptions. If the assumptions are reasonable, there will be no points off.



1) [20 points] Ordering

In the example to the left, multicasts are sent to the group designated by the oval, and containing participants C and D.
In our scenario, (in real time order) A sends M2 to B, then sends M1 to the group, then M3 to the group. After receiving M2, B sends M4 to the group.

a) [5 points] Give all examples of legal orderings of {M1,M3,M4} that could be delivered to the group, assuming FIFO ordering.

b) [5 points] Give all examples of legal orderings of {M1,M3,M4} that could be delivered to the group, assuming causal ordering.

c) [5 points] Suppose you observe this system and see that participant C receives messages in the order {M1,M4,M3} and participant D receives messages in the order {M4,M1,M3}. What can you conclude about the underlying message delivery system: (mark all that apply)

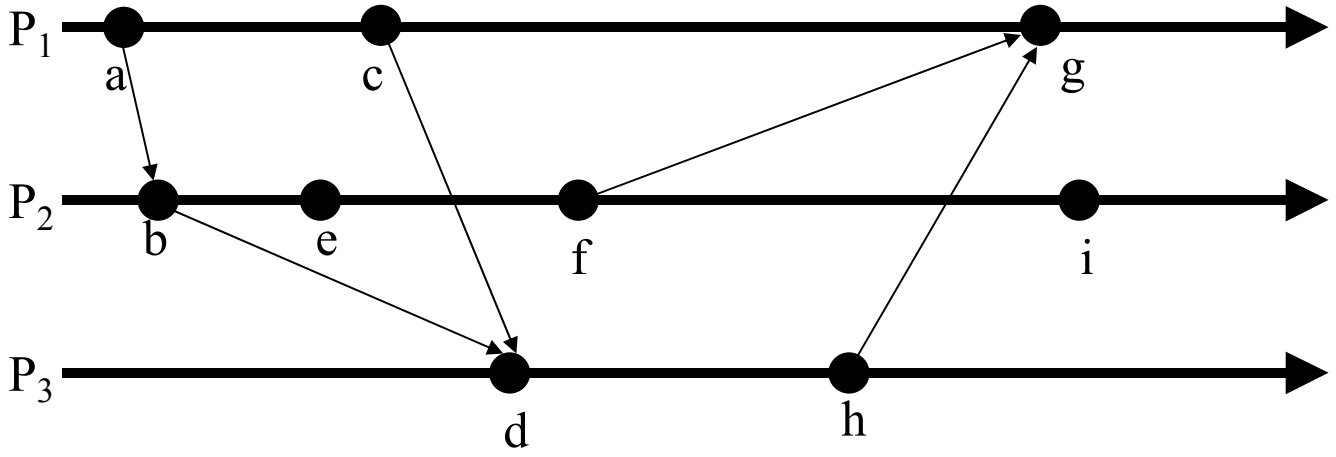
___ It is causal ___ It is FIFO ___ It is total

___ It is not causal ___ It is not FIFO ___ It is not total

d) [5 points] Give all examples of legal orderings of {M1,M3,M4} that could be delivered to the group, assuming causal ordering and that M2 was never sent.

2) [20 points] Cuts

In the following diagram, P_i denotes process *i*, a dot on a process's timeline denotes an event at the process, a thin arrow between two dots denotes a message sent between two processes. Physical time increases from left to right, in the direction of the thick horizontal arrows. (Note: a thin arrow denotes a message between two processes, and not any interaction with a process whose time line the arrow crosses; i.e., the message from **c** to **d** is from P_1 to P_3, and does not involve P_2.)



a) [4 points] Draw an inconsistent cut on the figure above, and mark it "IC"

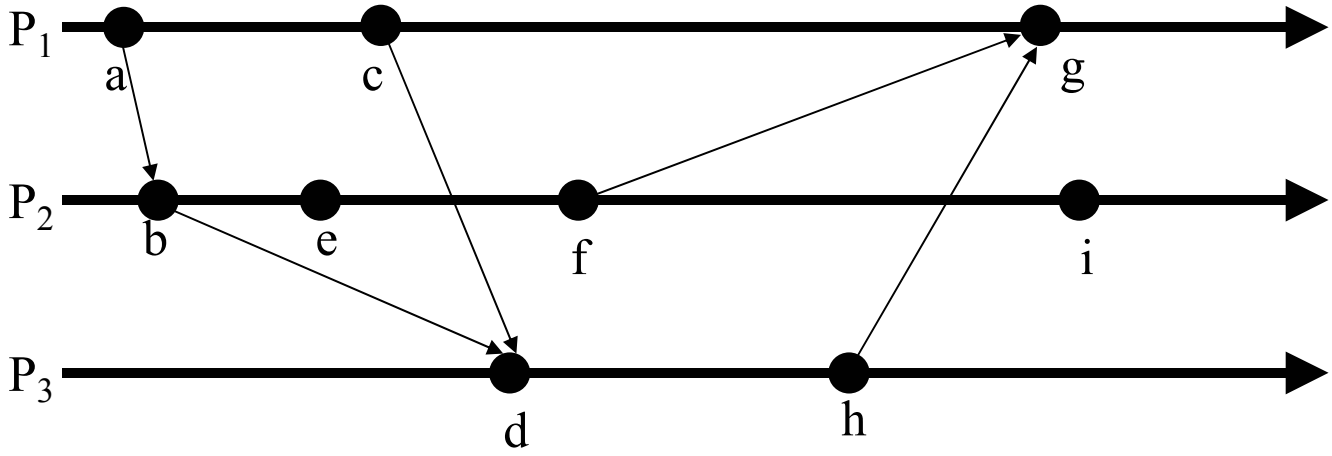
b) [4 points] Draw a strongly consistent cut on the figure above, and mark it "SCC"

c) [4 points] Draw a consistent, but not strongly consistent, cut on the figure above, and mark it "CC"

d) [8 points] The <u>frontier</u> of a cut is defined as the latest events from each process represented in the cut's states. For example, the frontier {c,e,d} represents a strongly consistent cut. Give two examples of a frontier of an inconsistent cut that can be made by changing only one of the events in the frontier {c,e,d}.

3) [25 points] Logical Time

In the following diagram, P_i denotes process *i*, a dot on a process's timeline denotes an event at the process, a thin arrow between two dots denotes a message sent between two processes. Physical time increases from left to right, in the direction of the thick horizontal arrows. (Note: a thin arrow denotes a message between two processes, and not any interaction with a process whose time line the arrow crosses; i.e., the message from **c** to **d** is from P_1 to P_3, and does not involve P_2.)



a) [8 points] On the diagram above, for each event write the Logical Clock (LC) time at its processor to the left of the dot for the event.

b) [17 points] Fill out the empty cells in the table to give the relations between each event: "→" denotes "happened before", "←" denotes "happened after", and "||" denotes "concurrent". (For examples of this notation, because there is a message from 'a' to 'b', cell [a,b] is filled with "→" and cell [b,a] is filled with "←". Also, 'g' and 'i' are concurrent, and are so marked.)

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
a	—	→							
b	←	—							
c			—						
d				—					
e					—				
f						—			
g							—		||
h								—	
i							||		—

4) [10 or 20 points] Local vs. Remote Procedure Calls

a) [10 points] Discuss the truth or falsity of the statement "With enough available bandwidth, remote procedure call can be made to behave just like local procedure call".

b) [564 only; 10 points] Discuss the truth or falsity of the statement "The interface of an object is independent of the context in which that object is used".

5) [12 points] A and B are participants in a distributed system.

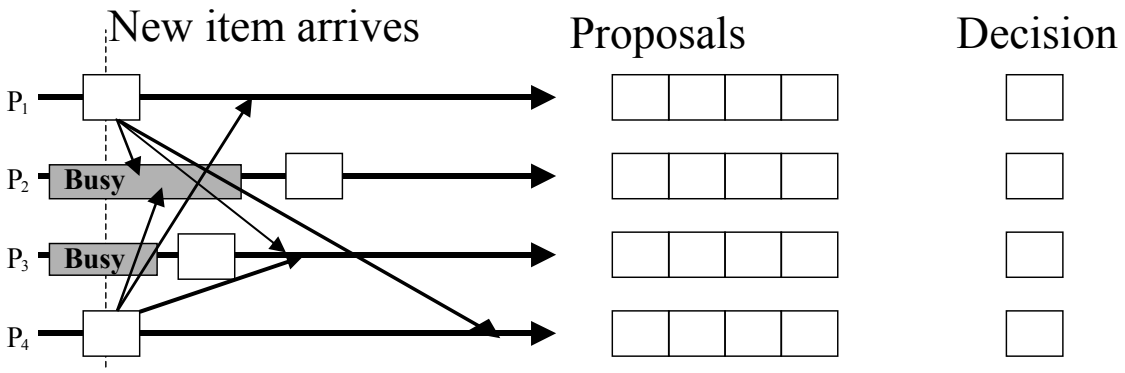
a) [3 points] When A sends a short message to B, an external observer measures the time it takes as 24 milliseconds. What term is used for this property of the network connecting A and B?

b) [3 points] When A sends long messages to B, the observer sees that B receives a megabyte of data per second. What term is used for this property of the network connecting A and B?

c) [6 points] The observer sees that the local clocks of A and B are never more than 20 milliseconds apart, though she is not able find any bound on the difference between the clocks and an international standard clock. We would say that the clocks are _____synchronized and that the _____of the synchronization is 20 milliseconds.

6) [20 points] Distributed Consensus

Consider the consensus example below, using an algorithm identical to that described in the textbook and discussed in class, with a set of assembly line machines reaching consensus on which one will handle the next item. In the example below, when the new item arrives (denoted by the dashed line), P_1 and P_4 are not busy and send in an immediate proposal, but P_2 and P_3 are busy then and send in the proposal at the time indicated by the blank box on their timelines.



a) [6 points] Write in the blank boxes on the timeline diagrams the values that each of the processes will propose.

b) [6 points] Draw in arrows depicting messages that P_2 and P_3 would send in this scenario

c) [6 points] Write in the proposals' values in the order received at each process in the box labeled "Proposals".

d) [2 points] Write in the consensus decision that each process will make, assuming it uses the "minimum" function to decide, under the heading "Decision".

7) [10 points] Concurrency Control

In any system that provides concurrency control one must be wary of the possibility of deadlock. Consider the following program consisting of a Producer process and a Consumer Process. The lock variable mutex starts in the unlocked state, bufferEmpty in the unlocked state, and bufferFull in the locked state.

```
shared int buffer;
Producer () {
   int item;
   while (true) {
      produce_item (&item);
      lock(mutex);
      lock(bufferEmpty);
      buffer = item;
      unlock(bufferFull);
      unlock(mutex);
   }
}
```

```
Consumer () {
   int item;
   while (true) {
      lock(bufferFull);
      lock(mutex);
      item = buffer;
      unlock(bufferEmpty);
      unlock(mutex);
      consume_item(&item);
   }
}
```

When executed, the program deadlocks. (This question appeared on a previous exam but we skipped the applicable section of the book this time.)

a) [5 points] Which statement is each of the processes attempting to execute when the deadlock occurs?

b) [5 points] What simple change could be made to the source code to prevent this deadlock?

8) [9 or 19 points] Partitions

a) [9 points] Describe the primary partition technique, the problem it solves, and how it solves it.

b) [564 only; 6 points] Several different ways of determining a primary partition were discussed. What are the fundamental requirements of any technique for determining whether or not a partition is primary?

c) [564 only; 4 points] When a partition heals, what steps must be taken before normal operation of the system can resume?

9) [14 <u>or 19</u> points] Fundamental Tradeoffs

a) [6 points] List four characteristics that distinguish a fully asynchronous framework from a synchronous framework.

b) [4 points] What are the advantages of a fully asynchronous framework?

c) [4 points] What are the advantages of a synchronous framework?

d) [<u>564 only</u>; 5 points] Why might a synchronous framework be hard to realize in practice?

10) [10 points] Reasons to Distribute Applications

The decision to build an application using distributed systems techniques is often motivated by a desire to improve availability and performance. Describe an application for which availability and performance could be improved using distribution. Describe the specific distribution techniques you would use and explain how they contribute to availability and performance of the system.