

# **CS 425: Homework #1**

Due on 09/27/2011 2:00PM

**Kurt Rudolph**  
rudolph9

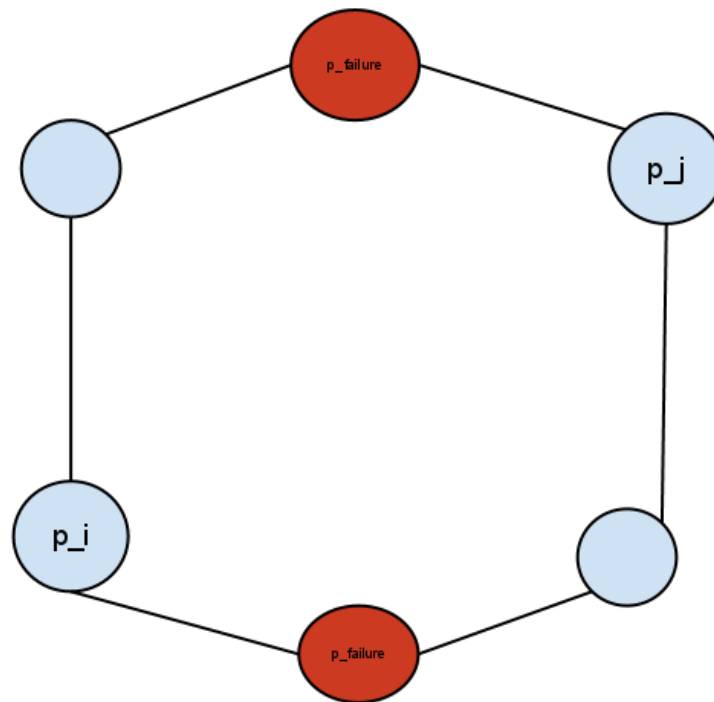
## Problem 1

Ring heart-beating may not detect simultaneous multiple failures of processors.

- (1) What is the maximum number of simultaneous processor failures that can be detected by ring heart-beating protocol?

### Solution

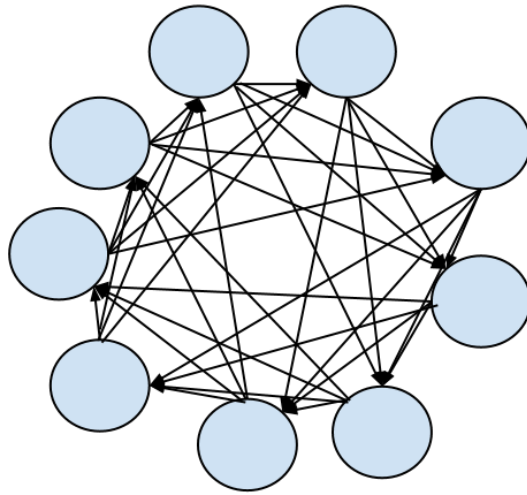
Let  $N$  equal the number of processes within the distributed system. The maximum number of process which can be detected by the ring heart-beating protocol is  $\lfloor \frac{N}{2} \rfloor$ . However if  $|p_{failure}| > 1$  as seen in the diagram below, the nodes detecting failure  $p_i$  will have no means of communicating the failure to all the nodes in the system.



- (2) Modify the ring heart-beating protocol to detect up to 4 simultaneous processor failures.

### Solution

A modified version of the ring heart-beating protocol capable of detecting up to 4 simultaneous process failures is as following. Placing all nodes in a ring formation where each node has a proceeding and preceding node, for each node in the system connect to the proceeding 4 nodes, guaranteeing failure detection up to 4 nodes, as in the diagram below.



## Problem 2

(Problem 2.14 from the book) Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed, or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents.

- (1) Describe the classes of failure exhibited by each service.

### Solution

Service A:

- (a) Omission Failure A message inserted in an outgoing message buffer never arrives at the other end's incoming message buffer.
- (b) Arbitrary Failure Process/channel exhibits arbitrary behavior: it may send/transmit arbitrary messages at arbitrary times, commit omissions; a process may stop or take an incorrect step.

Service B:

- (a) Omission Failure A message inserted in an outgoing message buffer never arrives at the other end's incoming message buffer.
- (b) Clock Failures Process's local clock exceeds the bounds on its rate of drift from real time.
- (c) Performance Failure A message's transmission takes longer than the stated bound.

- (2) Can service B be described as a reliable communication service?

### Solution

The term *reliable communication* is defined in terms of validity and integrity as follows:

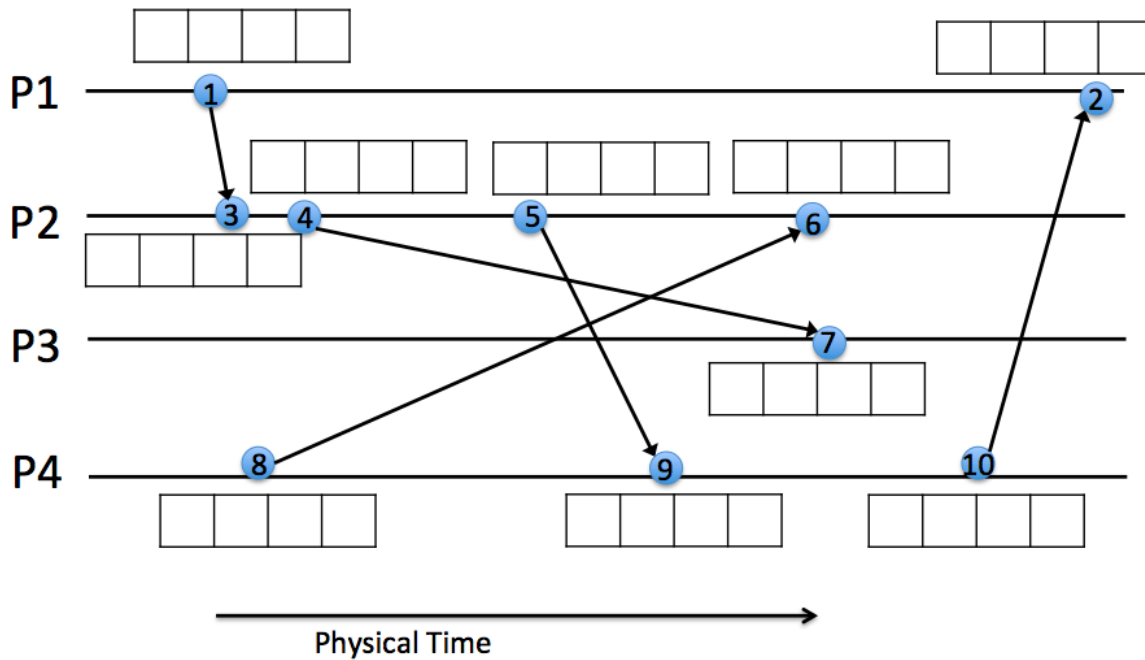
*validity*: any message in the outgoing message buffer is eventually delivered to the incoming message buffer.

*integrity*: the message received is identical to one sent, and no messages are delivered twice.

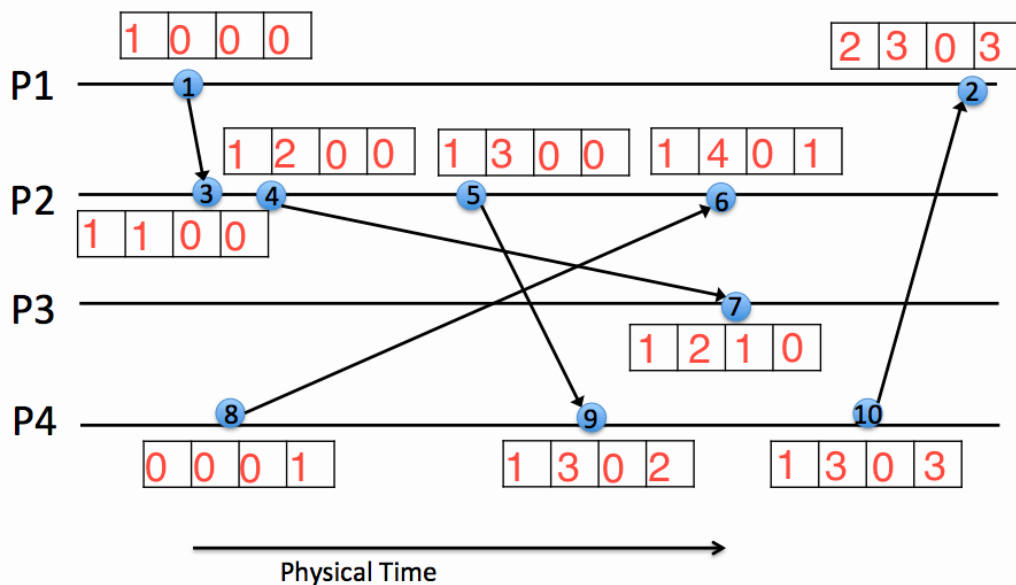
Because service B adheres to each of these constraints, service B may be considered a reliable communication service.

### Problem 3

P1, P2, P3, and P4 are four processes. Write down the vector logical timestamps in the boxes attached to each event.

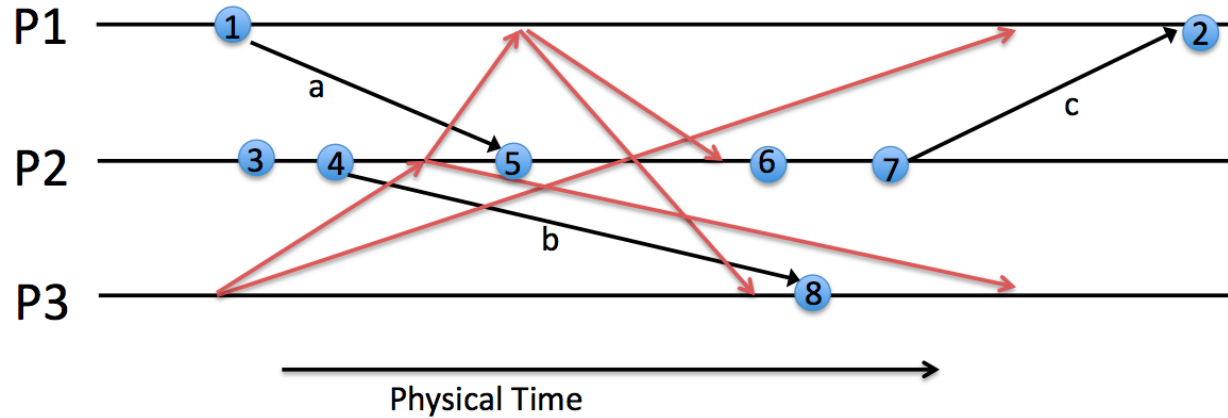


Solution

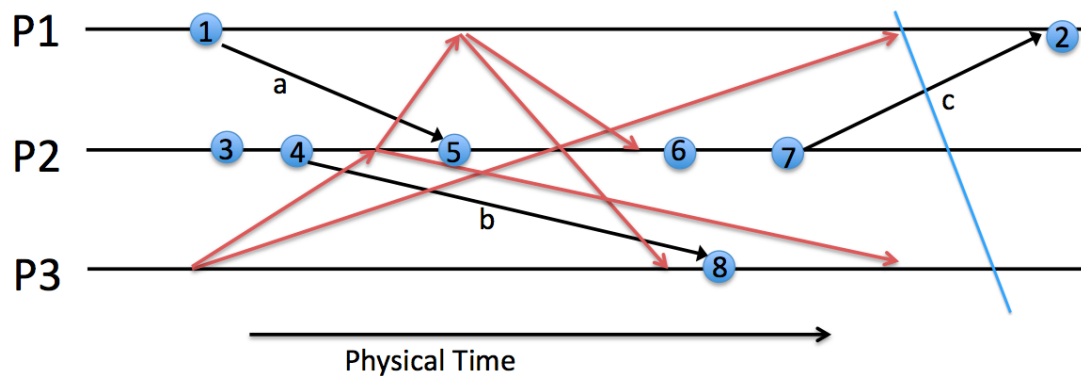


## Problem 4

Process P3 initiates the snapshot algorithm. The black arrows are messages sent and received. The red arrows are marker messages. Find out the consistent cut corresponding to this global snapshot and mark the states of each process and channel.

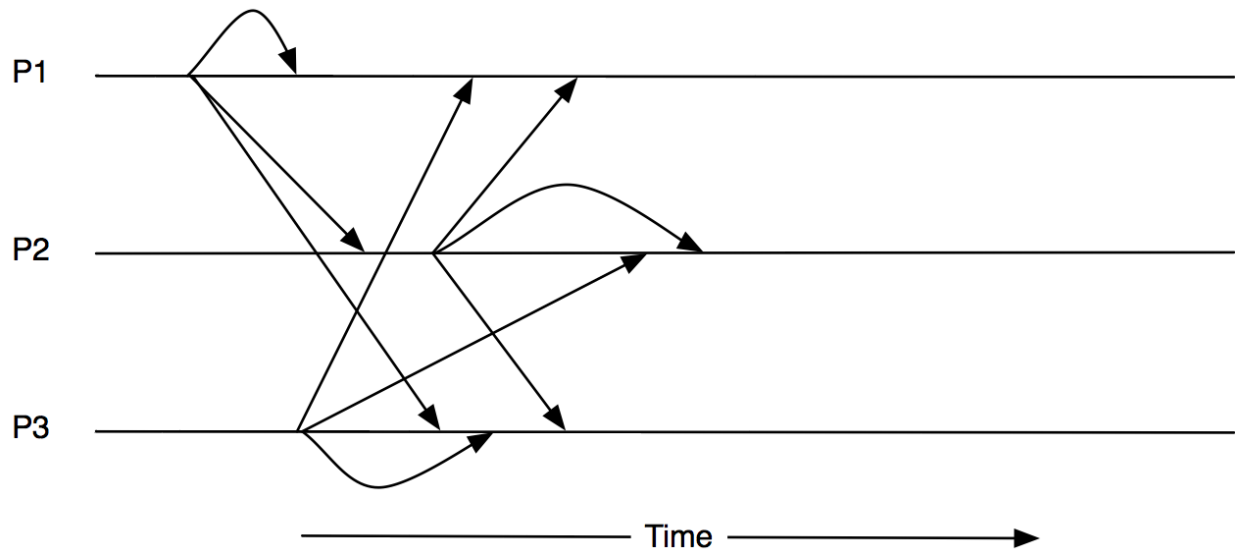


Solution



## Problem 5

Consider multicast messages sent and received using the order illustrated below. What ordering does this example follow? (a) FIFO (b) Causal (c) Total (d) FIFO-Total (e) Causal-Total.

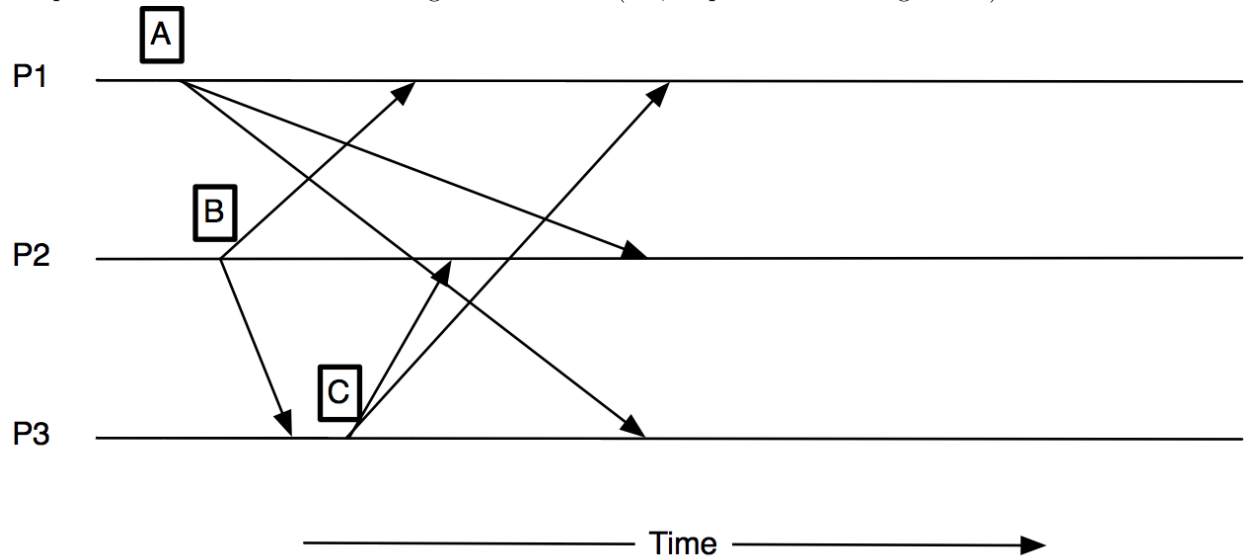


## Solution

The example follows (c) Total. This is indicated by the second to last message delivery.

## Problem 6

The figure below illustrates message flow in a multicast group that uses the Isis total ordering algorithm. Processes P1, P2, P3 each multicast message A,B,C, respectively. The figure shows the transmission and reception times of each initial message transmission (i.e., step 1 of the Isis algorithm).



- (a) Show the priority queues at each process at the completion of the messages shown and the proposed priority that each process assigns to each message.

### Solution

P1

- (1) B
- (2) C

P2

- (1) C
- (2) A

P3

- (1) B
- (2) A

- (b) What is the (total) order that the messages will be delivered in?

### Solution

P3:B, P1:B, P2:C, P1:C, P2:A, P3:A

- (c) Prove or disprove: *The Isis total ordering system always produces a FIFO total order.*



**Solution**

False, by definition total ordering states that if a correct process delivers message  $m$  before it delivers  $m'$ , then any correct process that delivers  $m'$  will deliver  $m$  before  $m'$ . The definition allows message delivery to be ordered arbitrarily as long as the order is the same at different processes. Hence total ordering does not preserve FIFO ordering.