Beyond Introduction

Distributed Systems

Nabendu Chaki
University of Calcutta, India

1

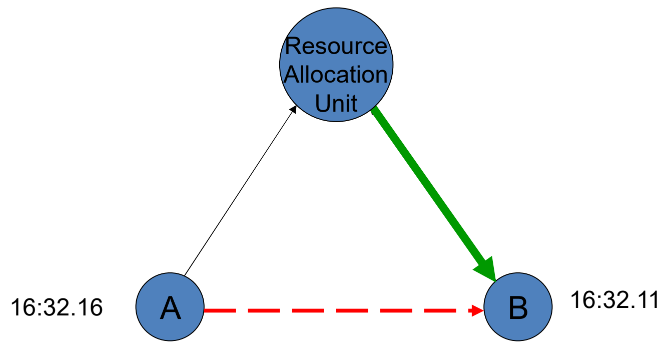
State Recording

- Local State
  - State of a component
- Global State
  - Collection of local states for all the components recorded at the same instance of time
  - Snapshot recording
- How do you ensure that same instance of time?
  - Where is the problem?

30 April 2024

2

Why do we need to Time Stamp?



30 April 2024

3

Clock Model

- Every processor has its own clock
- Clocks tend to drift
- Events occurring in different nodes gets time-stamps from different clocks
- How do you the find the event order?
- What will be the CLOCK MODEL?

Mukesh Singhal and Niranjan G. Shivaratri, Advanced Concepts in Operating Systems: Distributed, Database, and Multiprocessor Operating Systems, McGraw-Hill Education Pvt Limited.

4

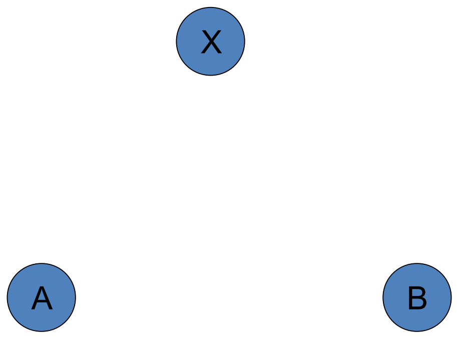
Options with the clock

- Synchronize clocks of different nodes
  - Clocks naturally drift
  - What happens after a finite duration?
  - Synchronization of clocks involves a high overhead, but does not guarantee a full-proof solution
- Assume a Master clock in some node
  - The clock node???

30 April 2024

5

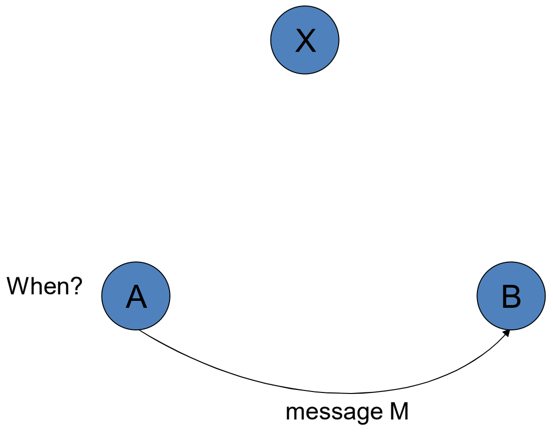
The Master Clock Option



30 April 2024

6

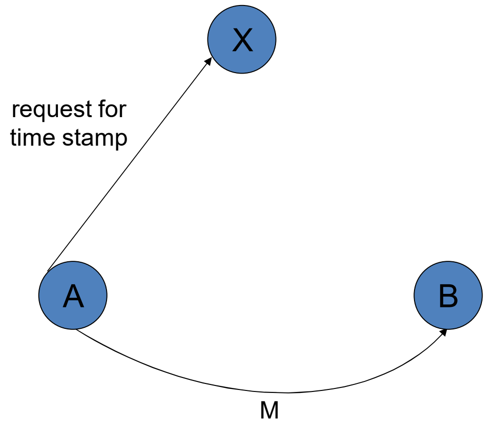
The Master Clock Option



30 April 2024

7

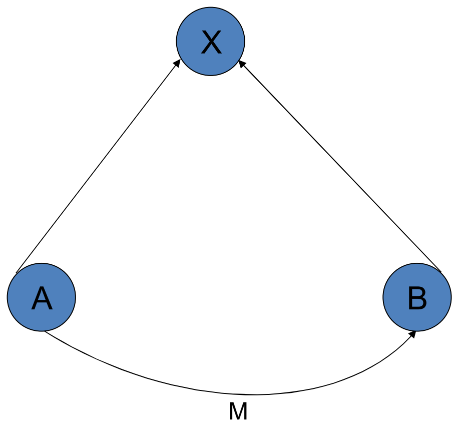
The Master Clock Option



30 April 2024

8

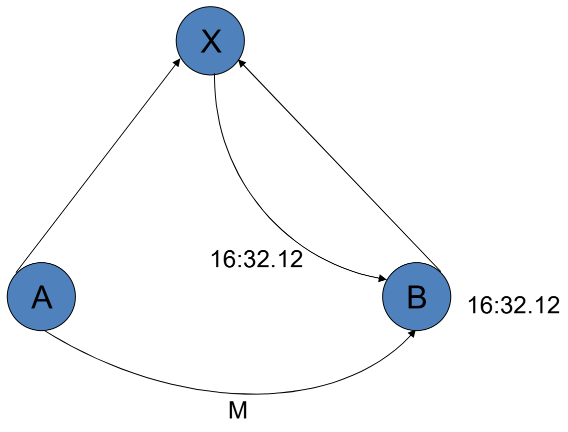
The Master Clock Option



30 April 2024

9

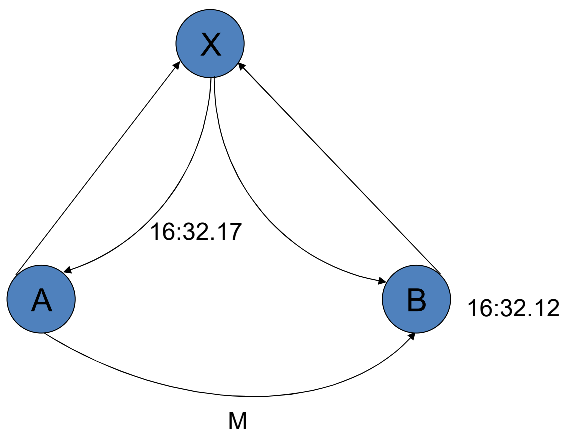
The Master Clock Option



30 April 2024

10

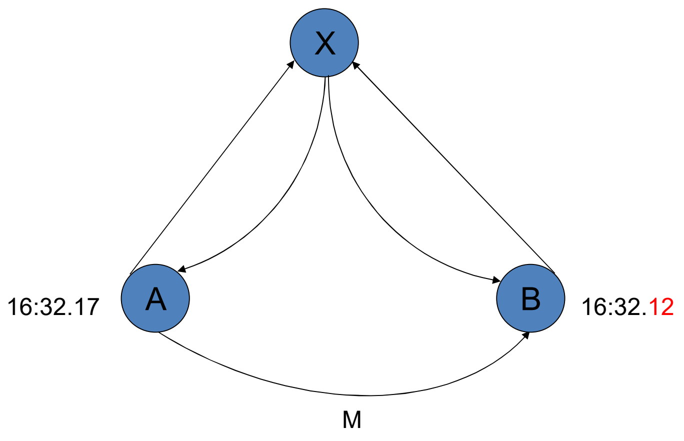
The Master Clock Option



30 April 2024

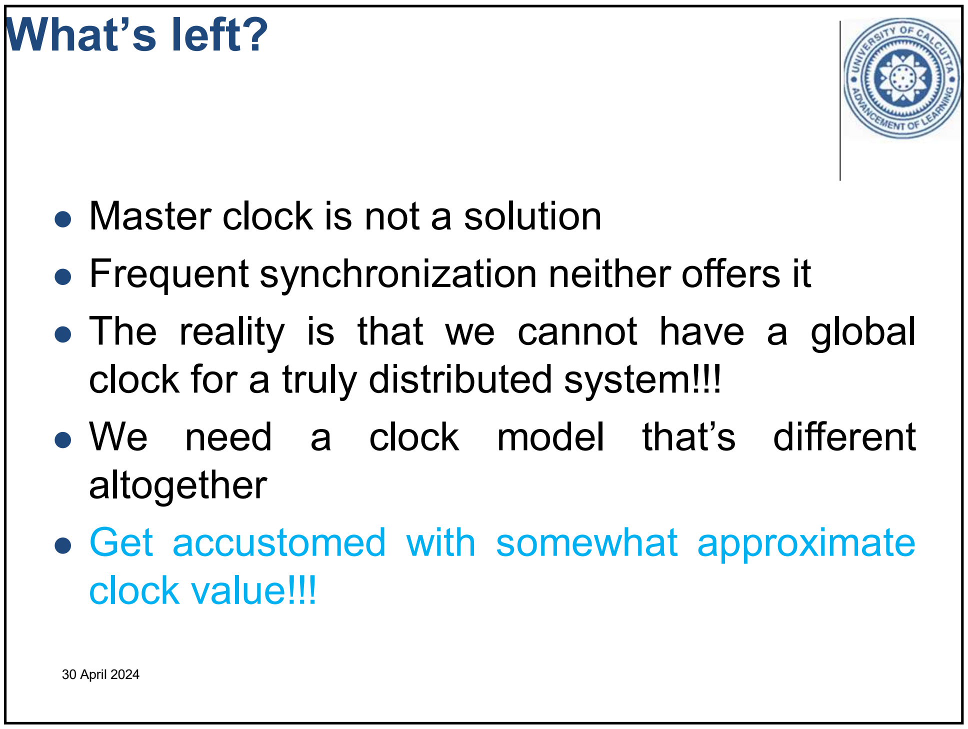
11

The Master Clock Option



30 April 2024

12

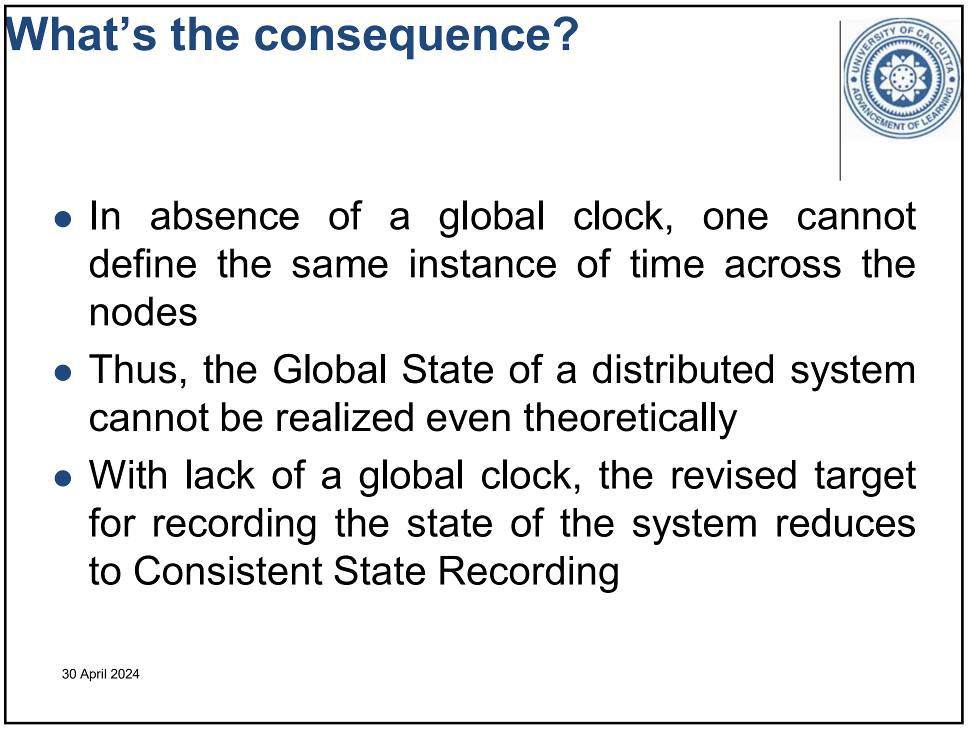


What's left?

- Master clock is not a solution
- Frequent synchronization neither offers it
- The reality is that we cannot have a global clock for a truly distributed system!!!
- We need a clock model that's different altogether
- Get accustomed with somewhat approximate clock value!!!

30 April 2024

13



What's the consequence?

- In absence of a global clock, one cannot define the same instance of time across the nodes
- Thus, the Global State of a distributed system cannot be realized even theoretically
- With lack of a global clock, the revised target for recording the state of the system reduces to Consistent State Recording

30 April 2024

14

Cause-Effect Pair of Events

- Create a file – Open the file
- Open a file – read the file
- Send a message – receive the message
- Get inputs – process the inputs
- These had been some examples of event-pairs that are related by the cause and its effect - the effect cannot precede cause

30 April 2024

15

Consistent State

CAUSE	EFFECT	STATE
√	√	Consistent
-	-	Consistent
√	-	Consistent
-	√	Inconsistent

30 April 2024

16

Consistent State Recording

- In a consistent state recording, for every effect that has been recorded for the system, the corresponding cause must also be recorded
- This is obvious when the effect and cause are located in the same node
- When the effect and cause are in two threads in two different nodes, then a message must be sent across the two nodes

30 April 2024

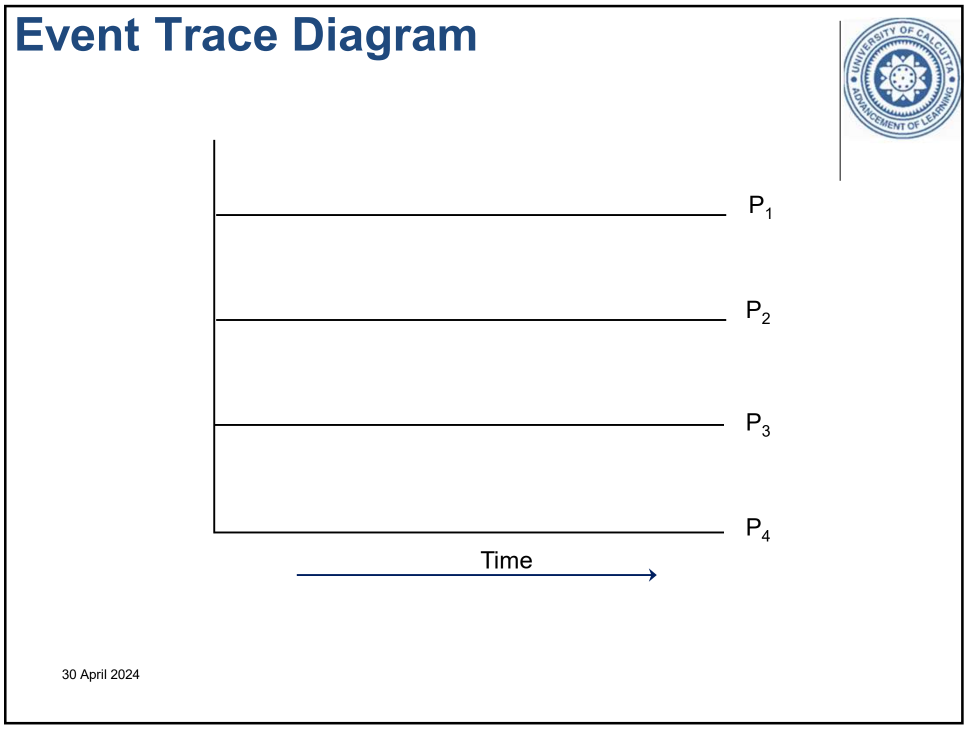
17

Consistent State Recording

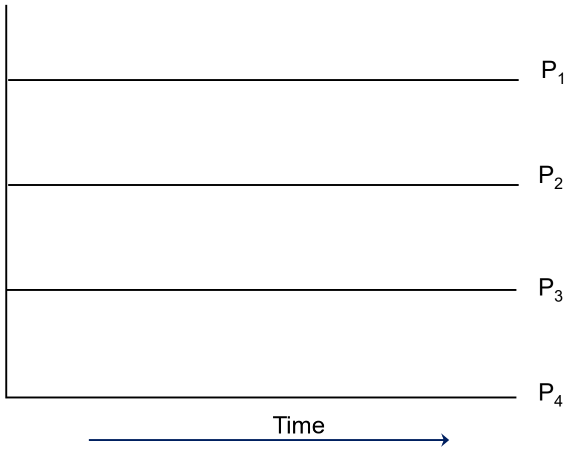
- Considering the message passing mode of communication for a distributed system, the condition for consistent state recording is revised as:
  - In a consistent state recording (CSR), for every message that is recorded as received, the corresponding state recording in the sender location must reflect that the message has been sent

30 April 2024

18

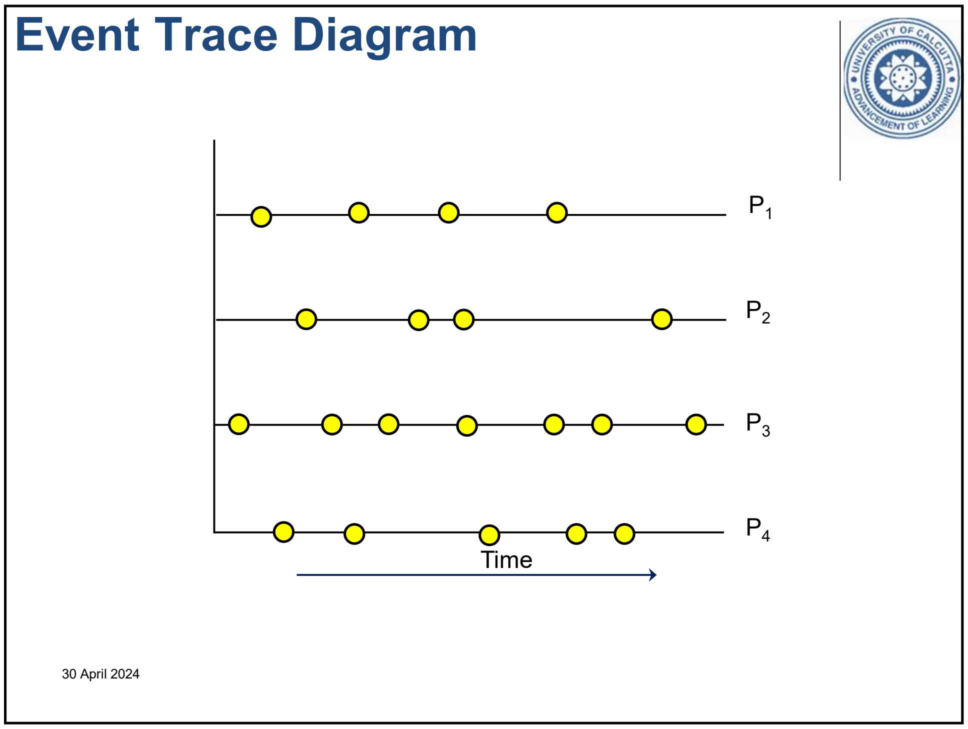


Event Trace Diagram

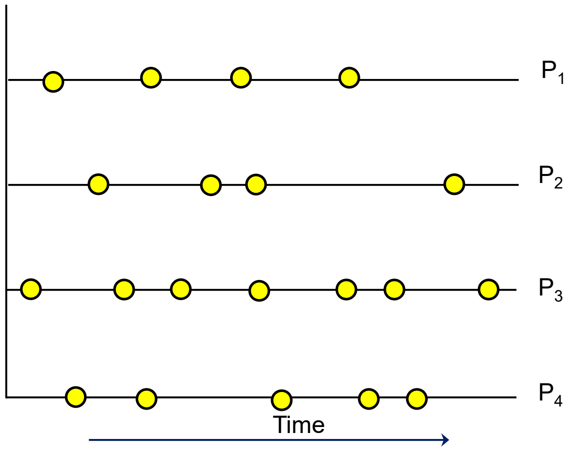


30 April 2024

19



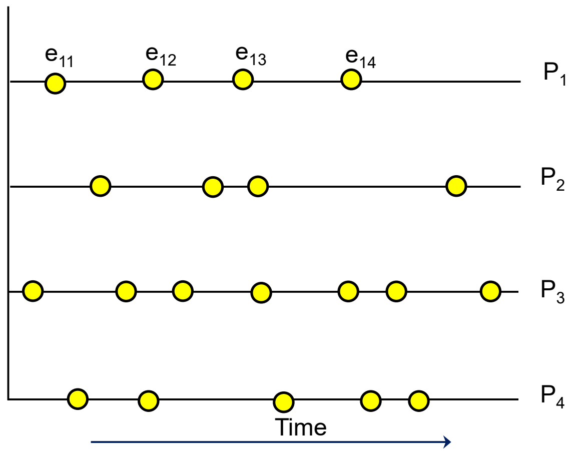
Event Trace Diagram



30 April 2024

20

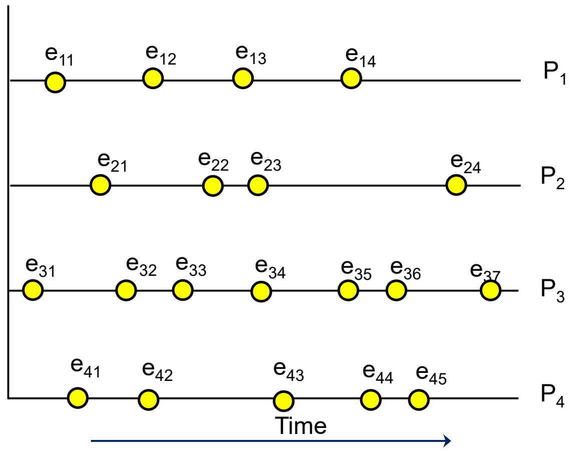
Event Trace Diagram



30 April 2024

21

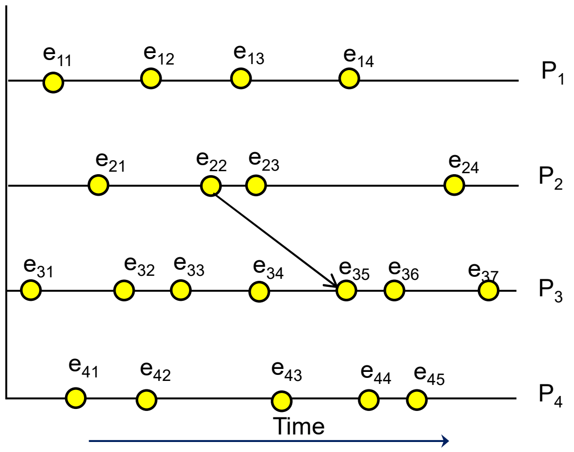
Event Trace Diagram



30 April 2024

22

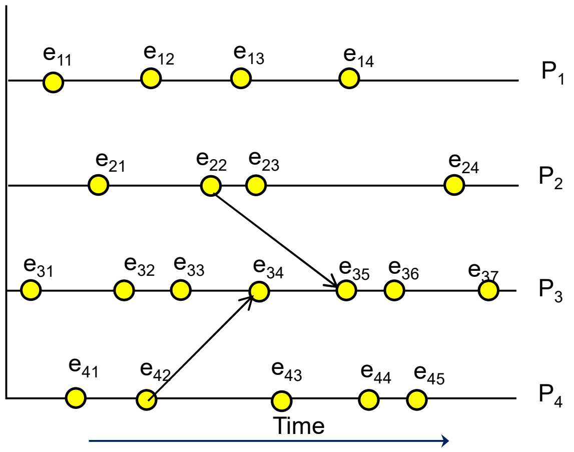
Event Trace Diagram



30 April 2024

23

Event Trace Diagram



30 April 2024

24

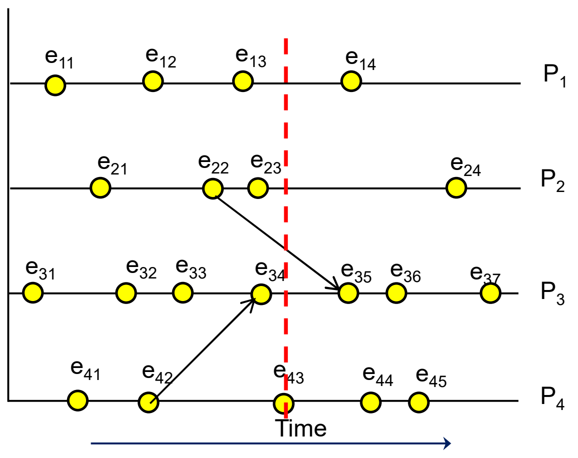
Cut in Event Trace Diagram

- A cut is an imaginary line that connects the points on the time-line of an event trace diagram in which the states for different nodes are recorded

30 April 2024

25

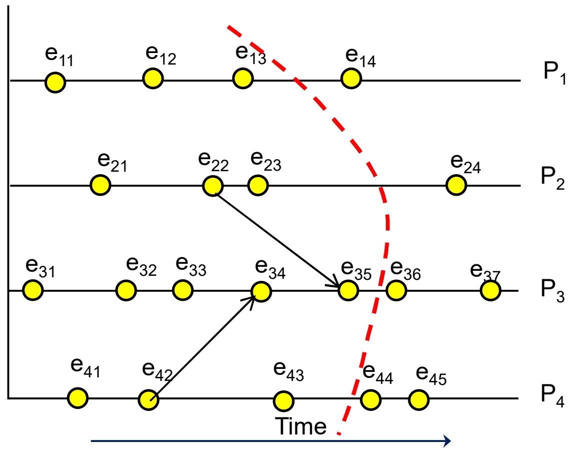
Ideal Cut for Global State Recording



30 April 2024

26

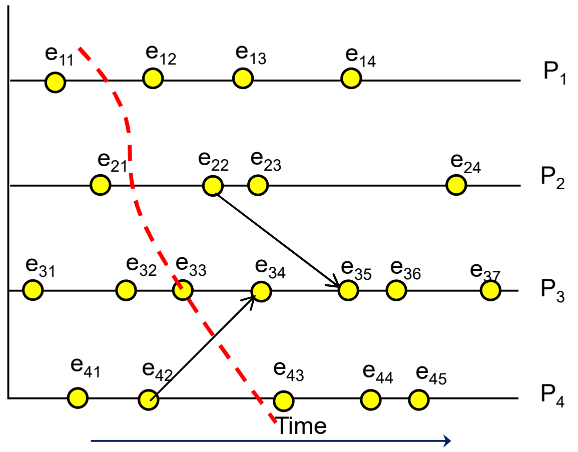
Cut 1: No Intersection



30 April 2024

27

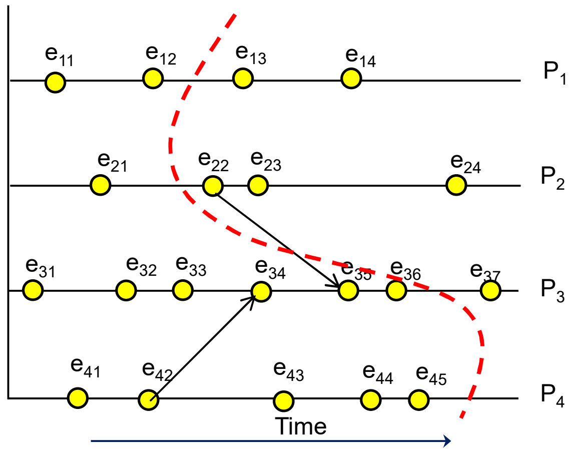
Cut 2: Forward Intersection



30 April 2024

28

Cut 3: Backward Intersection



30 April 2024

29

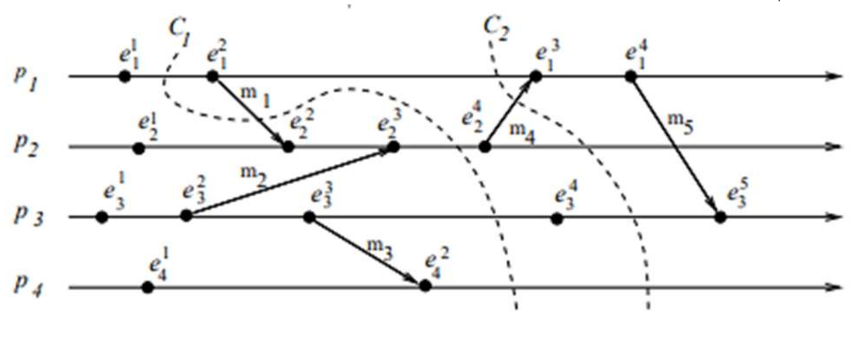
Cut and Consistency

- There are four possibilities
  - Both send and receive events are in past
    - No intersection, consistent state
  - Both send and receive events are in future
    - No intersection, consistent state
  - Send is in past and receive event is in future
    - Forward intersection, consistent state
  - Receive is in past and Send is in future
    - Backward intersection, inconsistent state
    - Problem in state recording algorithm

30 April 2024

30

Cut 4: Scenarios

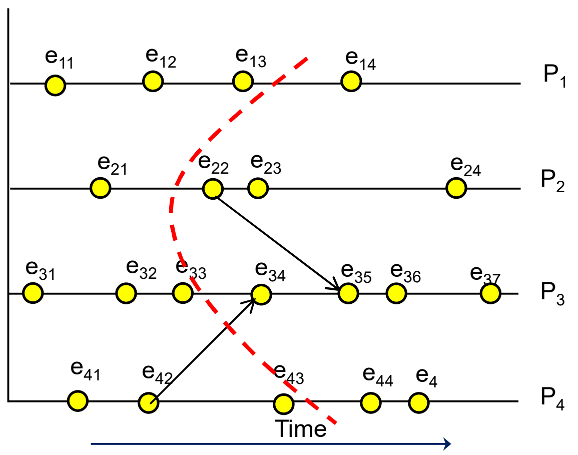


Time

30 April 2024

31

Cut 4: A Scenario



Time

30 April 2024

32

Condition of Consistency

- Two processes P_i and P_j are said to be mutually consistent if all the messages recorded as received by either of the two processes are also recorded as sent
- The state of a system is consistent if every possible pair of processes are mutually consistent

Thanks for your kind attention

Questions??