Non-Distributed Excercises

CS60002: Distributed Systems

Sliding Window Protocol

TRUE OR FALSE (with explanation)

1. **In the sliding window protocol, we must use a large window size when the bandwidth of the communication channel is small.**

2. **In the sliding window protocol, a packet is never re-transmitted by process P after it has been received by the process Q.**

Network Routing

TRUE OR FALSE (with explanation)

1. **While the distributed version of the Floyd-Warshall algorithm is in execution, temporary cycles may be created in a routing path between some source and destination.**

2. **NetChange routing algorithm uses an upper-bound on the number of nodes to guarantee termination.**

Deadlock Prevention

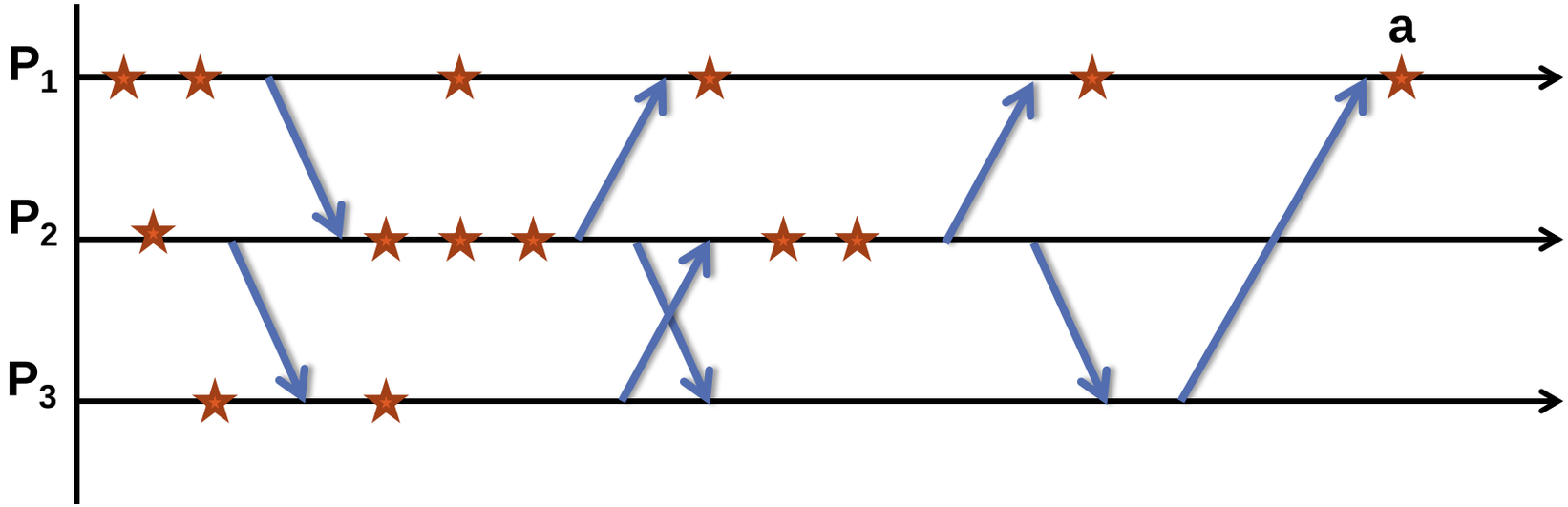
TRUE OR FALSE (with explanation)

1. **Buffer graph for deadlock-free packet switching may have cycles if it has to support two-way communication between a pair of nodes.**

2. **Forward state controller for deadlock-free packet switching uses fewer buffers than a forward count controller.**

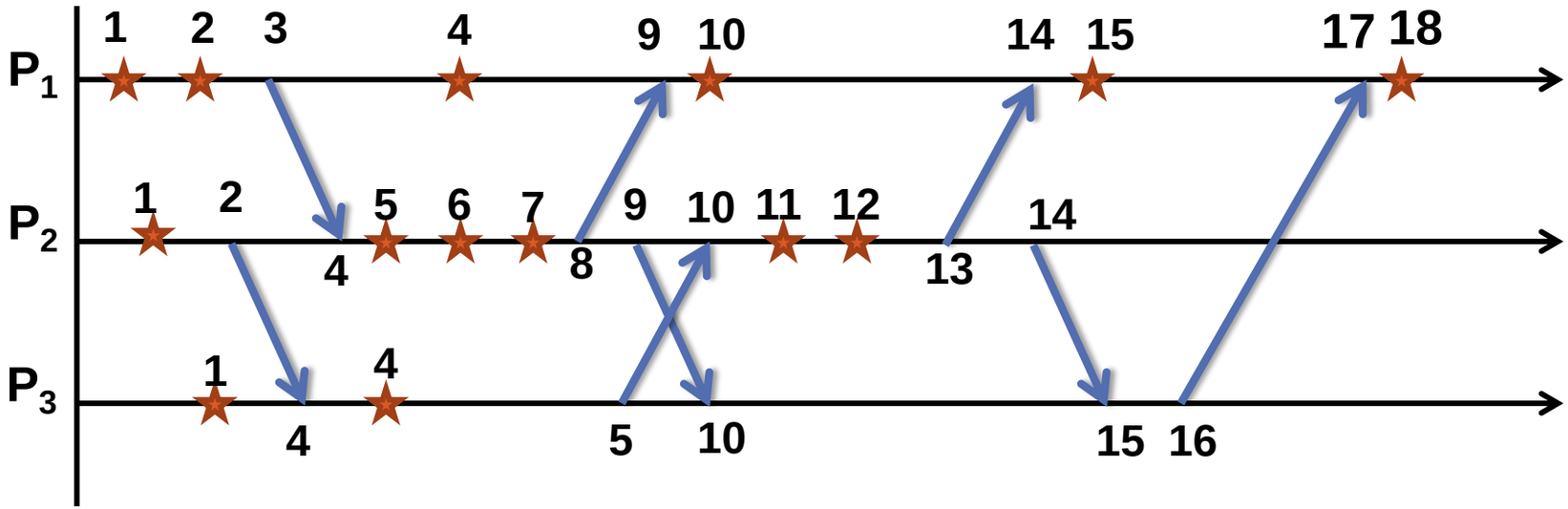
Logical Clocks

- With respect to the space-time diagram given below (where stars indicate local computation events), what would be the Lamport's logical timestamp and vector clock timestamp of the event *a*?



Logical Clocks

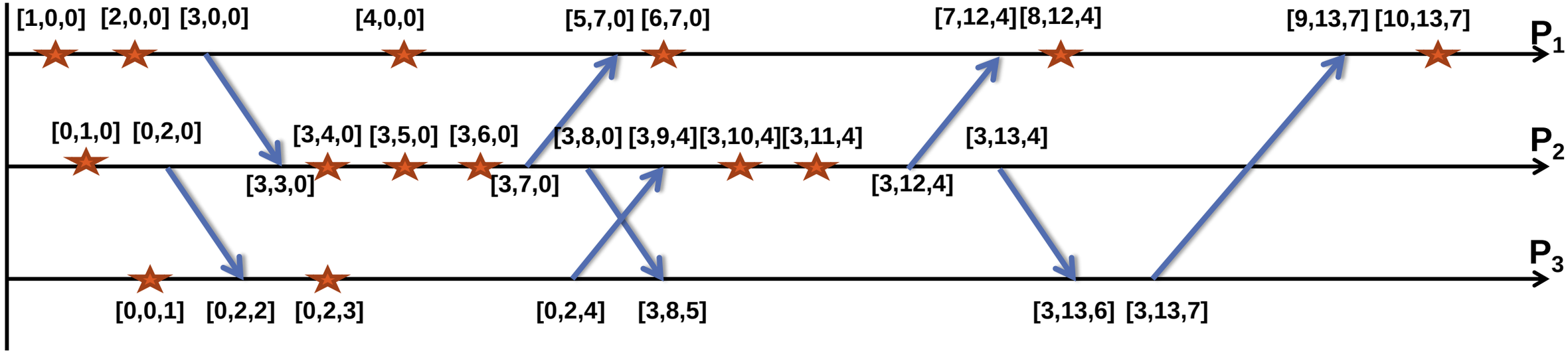
- With respect to the space-time diagram given below (where stars indicate local computation events), what would be the Lamport's logical timestamp and vector clock timestamp of the event *a*?



LAMPORT'S LOGICAL TIMESTAMP

Logical Clocks

- With respect to the space-time diagram given below (where stars indicate local computation events), what would be the Lamport's logical timestamp and vector clock timestamp of the event *a*?



VECTOR CLOCKS

- Suppose in Lamport's Logical Clock, each node I increments its clock by a fixed positive constant $d_i > 0$ on an event (instead of incrementing by 1). Will the clocks work correctly if the d_i's are different for different i? Justify your answer.

Global Snapshots

- Suppose that two processes *i* an *j* take five snapshots each of their local states independently at arbitrary times, with no communication for snapshot capture between them (i.e., no explicit snapshot capturing algorithm is run, processes just record their states whenever they want). Channel states are implicitly captured in the process states using message logs.
  - Is it possible to have a scenario in which no pair of snapshots (one taken from *i* and one taken from *j*; there are 5*5=25 such pairs) is consistent? Justify your answer with a space-time diagram.

Distributed Mutual Exclusion

- **Show the following with an example using space-time diagrams involving two processes only:**
  - **Why does Lamport's mutual exclusion algorithm require FIFO channels?**
  - **Why Raicairol-Carvalho's mutual exclusion algorithm does not ensure timestamp-ordered entry into the critical section.**