WELLESLEY

CS349-01: Distributed Computing

Assignment 3

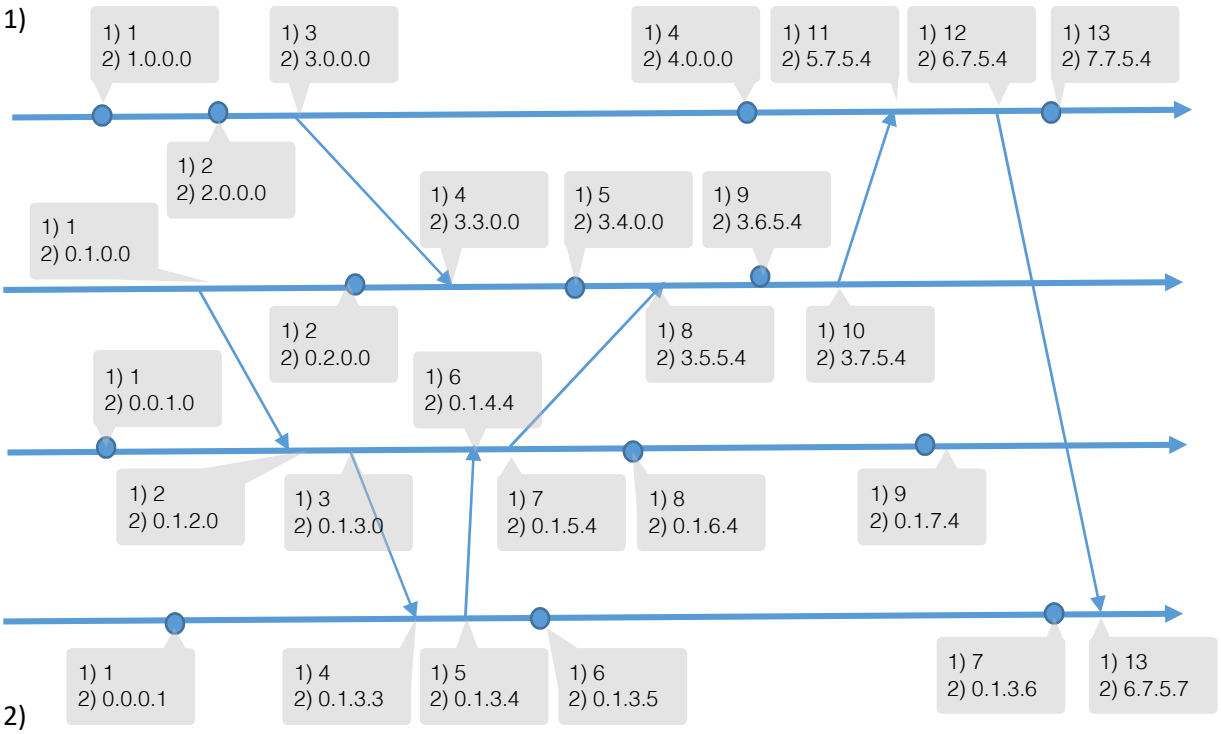
Written problems (Individual)

Problem 1

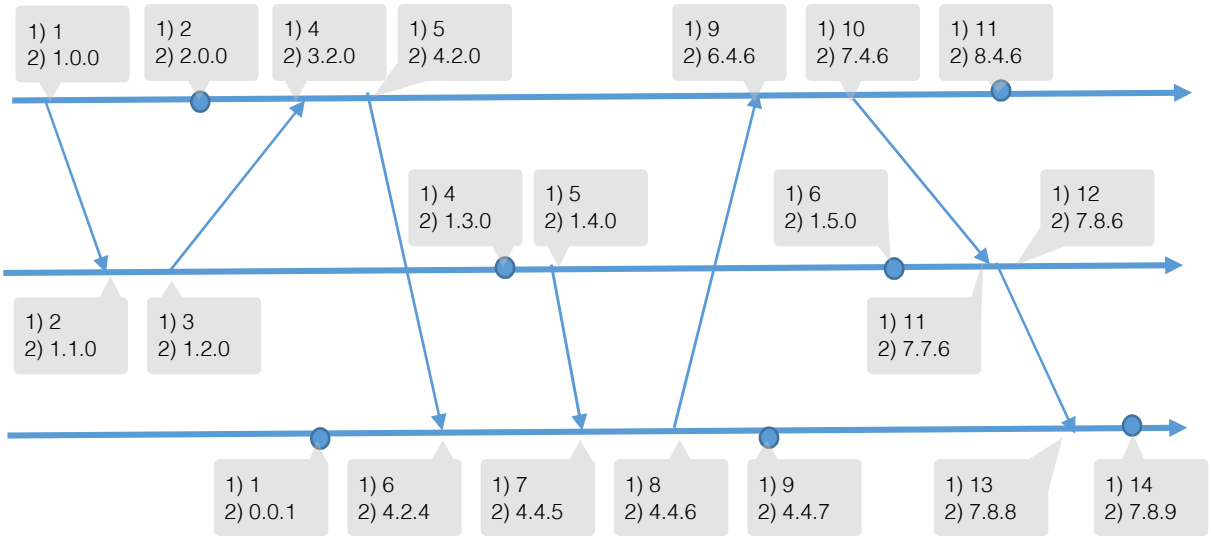
For each of the systems of distributed processes shown below, compute the following:

1. Scalar times of all events using Lamport algorithm
2. Vector times of all events

1)



2)



Problem 2:

Exercise 3.3 If events e_i and e_j respectively occurred at processes p_i and p_j and are assigned vector timestamps VT_{e_i} and VT_{e_j}, respectively, then show that

$$e_i \rightarrow e_j \Leftrightarrow VT_{e_i}[i] \prec VT_{e_j}[i].$$

When calculating vector timestamps for a receiving node (like e_j), we first increment the corresponding entry in that processes previous timestamp. We then take the element wise maximum of this new vector and the timestamp of the sender (VT_{ej}). Because of this element-wise maximum operation, we know that each element in VT_{ej} is ≥ to the corresponding element in VT_{ei}. We also know that the element in the new timestamp (VT_{ej}) corresponding to the receiving process (p_j) is strictly greater than the matching entry in VT_{ej} because it is at least one greater than in that process's previous timestamp, which is the maximum value any other sending process (p_i) could have a record of. Thus, if a message is sent from e_j to e_j, it is known that the vector timestamp VT_{ei} is strictly less than VT_{ej}.

Problem 3:

Exercise 4.3 Consider a distributed system where every node has its physical clock and all physical clocks are perfectly synchronized. Give an algorithm to record global state assuming the communication network is reliable. (Note that your algorithm should be simpler than the Chandy–Lamport algorithm.)

Let the snapshot process be initiated by all processes simultaneously at agreed-upon increments of time (for example: every time the number of seconds is devisable by 3 and ms = 000). Whenever a snapshot is triggered, each process records it's own state, then sends a "stop recording" message to every other process (requires a fully connected network). After doing so, each process records all incoming messages on each channel until it receives one of these "stop recording" messages, capturing all data which was in transit at the time of the snapshot's commencement (the channel states).