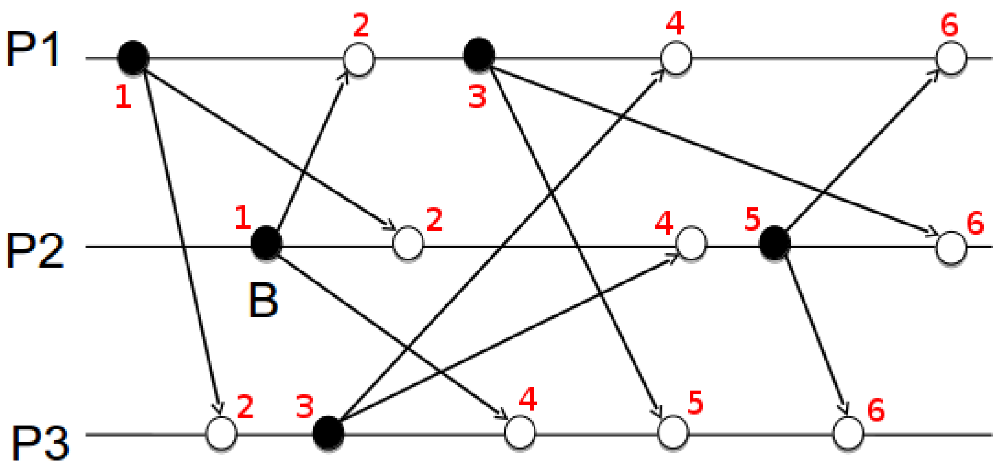
DSM: Tutorial 3

Question 1

P_1, P_2 and P_3 broadcast to each other using a **reliable, FIFO, causal** broadcast protocol. P_2 broadcasts $m1$ and $m2$. P_3 broadcasts $m3$ after delivering $m1$ and before delivering $m2$.

- P_1 may deliver the messages as $\langle m1, m2, m3 \rangle$ or $\langle m1, m3, m2 \rangle$.
- Using atomic broadcast, only $\langle m1, m3, m2 \rangle$ would be possible.

Question 2



- **FIFO**: satisfied, because every message sent from a process has a **higher clock** value than all messages it has **previously sent**.
- **Causal**: satisfied, because every message sent from a process has a **higher clock** value than all messages it has **previously received**.

Question 3

Prove that for all processes P_i and P_j, $C_i[j] \leq C_j[j]$.

According to the vector clock algoritm clock values never decrease: values are either incremented on broadcast, or overwritten by higher incoming values. $C_i[j]$ starts at 0 and is only updated when a vector clock V is received; new values for $V[j]$ are only created by P_j.

$C_i[j]$ is only ever the current value or a previous value of $C_j[j]$, and therefore $C_i[j] \leq C_j[j]$.

Question 4

Prove that if M_A causes M_B then for the clocks they contain $V_A < V_B$.

When M_A is delivered by some process P_i, that process will update C_i with any higher values from the vector clock V_A that was recevied. At this point, $V_A \leq C_i$.

C_i may increase further as a result of other events, but it will not decrease.

When M_B is later broadcast by P_i (creating the causal connection) $C_i[i]$ will be incremented to create V_B and therefore $V_A < V_B$.

Question 5

Prove that if two messages M_A and M_B are not causally related (they are concurrent) then for the clocks they contain neither $V_A < V_B$ nor $V_B < V_A$.

For no causal connection to exist, some process P_i must broadcast M_A before delivering M_B and some process P_j must broadcast M_B before delivering M_A.

Because broadcasting M_A increased $C_i[i]$ (and clock values are only increased by their 'owner') it is true that $V_B[i] < V_A[i]$.

Similarly, because broadcasting M_B increased $C_j[j]$ it is true that $V_A[j] < V_B[j]$.

V_A and V_B now both contain at least one value that is higher than in the other vector, so neither $V_A < V_B$ nor $V_B < V_A$ can be true.

Question 6

- When sending a message, a process includes **its own clock value from the last message** it broadcast.
- When receiving a message, a process knows which message it should have already received and transitively knows whether there are any messages to wait for before deliverying the one just received.
  - Hosts can essentially build up a linked-list of received messages for each sender, treating the 'root' part of the list as stable and considering any non-contiguous messages as unstable until the gaps are filled in.