Election Algorithms

Algorithms to select a leader/coordinator. The process with the highest identifier is generally elected as the coordinator.

Bully Algorithm

Consider N processes, $\{P_0, \ldots, P_{N-1}\}$ and let $id(P_k) = k$
When any processes notices that the coordinator is no longer responding to requests, it initiates an election.

A process P_k holds an elections as follows:

- P_k sends an ELECTION message to all processes with higher identifiers: $P_{k+1}, P_{k+2}, \ldots$
- If no one responds, P_k wins the election.
- If one of the higher ups answers, it takes over and P_k's job is done. It sends an OK message back to P_k to indicate that it is alive and will take over (recursive process starts unless the higher up is already holding an election).

The biggest guy in the room always wins and thus the name "bully algorithm".

If 2 processes send OK, they both will hold an election.
Whenever a processes wins it sends a message COORDINATOR to rub it in their noses.

Ring Algorithm

We assume each process knows who it successor is.

- When any process notices that the coordinator is no longer responding, it sends a message ELECTION containing its ID to its successor.
- If the successor is down, it goes over the next node in the ring, and one after that until a running process is located. Each sender adds its own ID int the message making itself the candidate for the election.
- Eventually this message gets back to process that started it all (ring) and recognizes with its own ID in the message.
- At that point the message type is changed to COORDINATOR and circulated once again to notify all who the new coordinator (the one with the highest ID in the list) is and who the member of the new ring are.

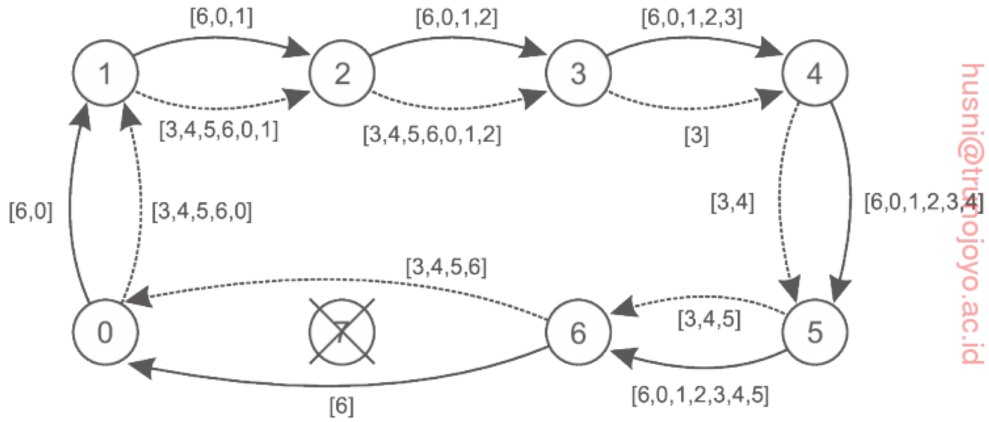

Figure 6.21: Election algorithm using a ring. The solid line shows the election messages initiated by P_6; the dashed one those by P_3.