Distributed Systems

a.y. 2023/2024
Distributed Systems:

Election
Leader election algorithms

- The problem
  - N processes, may or may not have unique ids (UIDs)
  - for simplicity assume no crashes
  - they must choose unique master co-ordinator amongst them
  - election called after failure has occurred
  - one or more processes can call election simultaneously

- (LE1) Every process knows P, identity of leader, where P is unique process id (usually maximum) or is yet undefined.
- (LE2) All processes participate and eventually discover the identity of the leader (it cannot be undefined).
Election Algorithms

- The Bully Algorithm
  1. $P$ sends an *ELECTION* message to all processes with higher numbers.
  2. If no one responds, $P$ wins the election and becomes coordinator.
  3. If one of the higher-ups answers, it takes over. $P$’s job is done.
(a) Process 4 holds an election.
(b) Processes 5 and 6 respond, telling 4 to stop.
(c) Now 5 and 6 each hold an election.
The Bully Algorithm

- (d) Process 6 tells 5 to stop.
- (e) Process 6 wins and tells everyone.
The election of coordinator $p_2$, after the failure of $p_4$ and then $p_3$. Eventually.....
A Ring Algorithm
Leader election algorithms

- **(LE1)** Every process knows $P$, identity of leader, where $P$ is unique process id (usually maximum) or is yet undefined.

- **(LE2)** All processes participate and eventually discover the identity of the leader (it cannot be undefined).
A Ring Algorithm

Previous coordinator has crashed

No response

Election message
End of lecture