Election Algorithms

Lecturer:
Prof. William Fornaciari
Politecnico di Milano
fornacia@elet.polimi.it
www.elet.polimi.it/~fornacia

Summary

- Problem definition
- Bully algorithm
- Ring algorithm

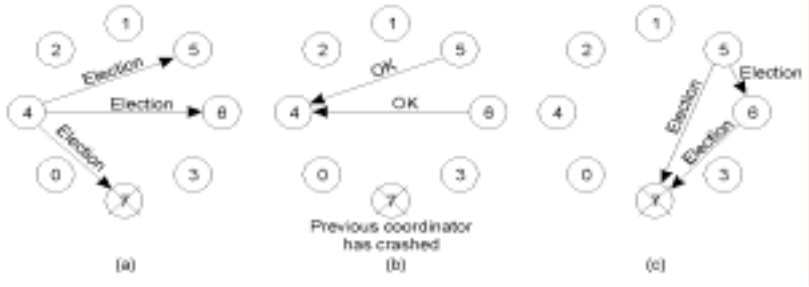
The problem

- Discover a coordinator among a group of similar processes, in charge for a generic service
- Assumptions
  - Each process has a unique id (i.e. machine # if only one P per node)
  - Each process knows the id of the others
  - The processes are unaware of the running/fault state of the others
- Election
  - Goal: ensure that the election achieves an agreement among all the processes (about the new coordinator)
  - Typ. the P with the highest id is elected

Bully algorithm

- A P suspecting the missing of a coordinator, can start an election
  - P sends an ELECTION msg to all the P with higher ids
    - If no one answer, P is the winner and becomes the coordinator
    - If a P with a bigger id answers, P looses
  - If a P receives msg from those having smaller id, an answer is sent to block the processes
- Finally only a P (the new coordinator) will remain and it will inform the other by sending a msg
- If a process wakeup or it is restarted, the first action is to trigger an election

Bully Algorithm



- Process 4 holds an election
- Process 5 and 6 respond, telling 4 to stop
- Now 5 and 6 each hold an election

Ring Algorithm

- Hp: processes are physically or logically ordered, each of them knows the following in the ring
- When a P suspects a coordinator fault
  - Sends an ELECTION msg containing its Pid to the next; if it is down the msg is sent to the following of the ring
  - Each member of the network receives and propagates a msg to the next, adding its id
- When a msg return to a P who sent it (verif. by inspecting the list), the msg is turned into COORDINATOR and circulated, to report:
  - New coordinator: P of the list with highest id
  - Network members still active
- Multiple messages can circulate over the network

Ring Algorithm