



Election Algorithms:

Distributed system is a collection of independent computers that do not share their memory. Each processor has its own memory and they communicate via communication networks. Communication in networks is implemented in a process on one machine communicating with a process on other machine. Many algorithms used in distributed system require a coordinator that performs functions needed by other processes in the system. **Election algorithms** are designed to choose a coordinator.

Election algorithms choose a process from group of processors to act as a coordinator. If the coordinator process crashes due to some reasons, then a new coordinator is elected on other processor. Election algorithm basically determines where a new copy of coordinator should be restarted.

Election algorithm assumes that every active process in the system has a unique priority number. The process with highest priority will be chosen as a new coordinator. Hence, when a coordinator fails, this algorithm elects that active process which has highest priority number. Then this number is sent to every active process in the distributed system.

We have two election algorithms for two different configurations of distributed system.

1. **The Bully Algorithm**
2. **The Ring Algorithm**

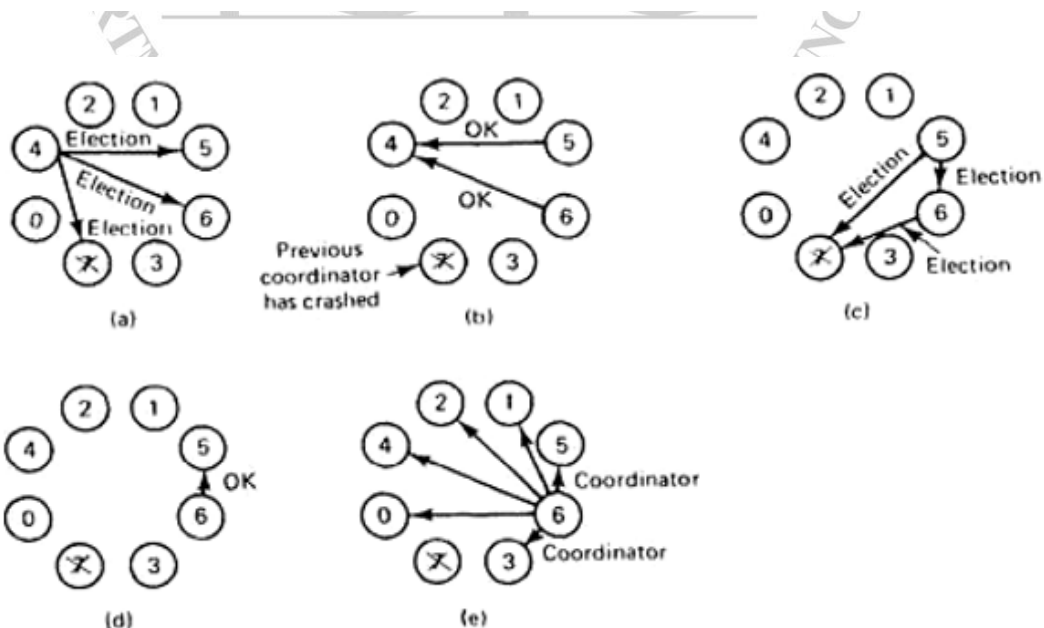
NOTE:-

- **Distributed algorithms require one process to act as a coordinator or initiator. To decide which process becomes the coordinator different types of algorithms are used.**
- **Election algorithms are meant for electing a coordinator process from among the currently running processes in such a manner that at any instance of time there is a single coordinator for all processes in the system.**
- **The goal of an election algorithm is to ensure that when an election starts it concludes with all the processes agreeing on who the coordinator should be.**
- **Therefore, whenever initiated, an election algorithm basically finds out which of the currently active processes has the highest priority number and then informs this to all other active processes.**

1. Bully Algorithm :-

- This algorithm was proposed by Garcia-Molina.
- When the process notices that the coordinator is no longer responding to requests, it initiates an election. A process, P, holds an election as follows:

- (I) P sends an ELECTION message to all processes with higher numbers.
- (II) If no one responds, P wins the election and becomes the coordinator.
- (III) If one of the higher-ups answers, it takes over. P's job is done.
 - a. A process can get an ELECTION message at any time from one of its lower numbered colleagues.
 - b. When such a message arrives, the receiver sends an OK message back to the sender to indicate that he is alive and will take over. The receiver then holds an election, unless it is already holding one.
 - c. All processes give up except one that is the new coordinator. It announces its victory by sending all processes a message telling them that starting immediately it is the new coordinator.
 - d. If a process that was previously down comes back up, it holds an election. If it happens to the highest numbered process currently running, it will win the election and take over the coordinator's job. Thus the biggest guy in town always wins, hence the name "bully algorithm".
 - e. Example:



UNIT 3 – ELECTION ALGORITHM

- In fig(a) a group of eight processes taken is numbered from 0 to 7. Assume that previously process 7 was the coordinator, but it has just crashed. Process 4 notices it first and sends ELECTION messages to all the processes higher than it that is 5, 6 and 7.
- In fig (b) processes 5 and 6 both respond with OK. Upon getting the first of these responses, process 4's job is over. It knows that one of these will become the coordinator. It just sits back and waits for the winner.
- In fig(c), both 5 and 6 hold elections by each sending messages to those processes higher than itself.
- In fig(d), process 6 tells 5 that it will take over with an OK message. At this point 6 knows that 7 is dead and that (6) it is the winner. If there is state information to be collected from disk or elsewhere to pick up where the old coordinator left off, 6 must now do what is needed. When it is ready to take over, 6 announces this by sending a COORDINATOR message to all running processes. When 4 gets this message, it can now continue with the operation it was trying to do when it discovered that 7 was dead, but using 6 as the coordinator this time. In this way the failure of 7 is handled and the work can continue.
- If process 7 is ever restarted, it will just send all the others a COORDINATOR message and bully them into submission.

2. Ring Algorithm :-

- This algorithm uses a ring for its election but does not use any token. In this algorithm it is assumed that the processes are physically or logically ordered so each processor knows its successor.
- 1. When any process notices that a coordinator is not functioning, it builds an ELECTION message containing its own process number and sends the message to its successor. If the successor is down the sender skips over the successor and goes to the next member along the ring until a process is located.
- 2. At each step the sender adds its own process number to the list in the message making itself a candidate to be elected as coordinator.
- 3. The message gets back to the process that started it and recognizes this event as the message contains its own process number.

UNIT 3 – ELECTION ALGORITHM



4. At that point the message type is changed to COORDINATOR and circulated once again to inform everyone who the coordinator is and who are the new members. The coordinator is selected with the process having highest number.
5. When this message is circulated once it is removed and normal work is preceded.

