

CASE STUDY NO : 5

Date	/ /20
Page No.	26

Case Study On :- Study of Election Algorithm.

Election algorithm and distributed processing-

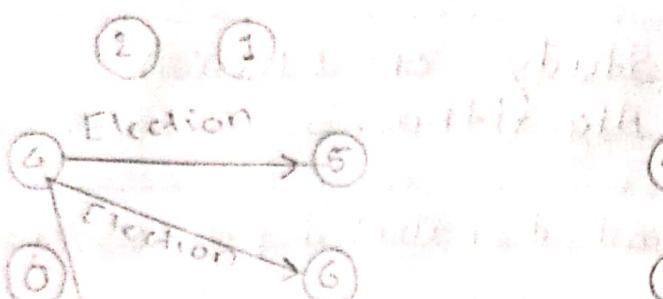
Distributed Algorithm is a algorithm that runs on a distributed system. Distributed system is a collection of independent computers that do not share their memory. Each processor has its own memory and they communicate via communication networks. Communication in networks is implemented in a process 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 :

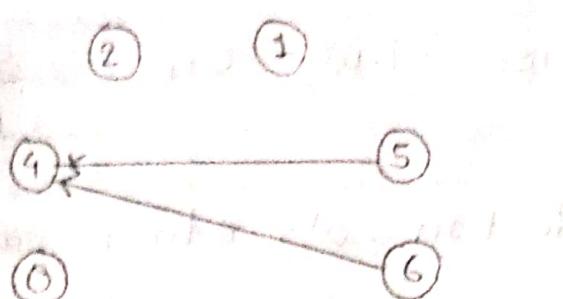
Election algorithms choose a process from group of processors to act as a coordinator. If the coordinator process crashes due to some reasons, then a new coordinator is elected on other processor. Election algorithm basically determines where a new copy of coordinator should be restarted.

3: OA Consensus

[a]



[b]



(2) (1) (0) (3) (4) (5) (6)

Election (2) (1) (0) (3) (4) (5) (6)

[c]

[d]

(2) (1)

(0) (5)

(4) (3)

(6)

(2) (1)

(0) (5)

(4) (3)

(6)

(2) (1)

(0) (5)

(4) (3)

(6)

Building after right 1

Building after right 2

Building after right 3

Building after right 4

Building after right 5

Building after right 6

Building after right 7

Building after right 8

Building after right 9

Building after right 10

Building after right 11

Building after right 12

Election (2) (1) (0) (3) (4) (5) (6)

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 numbers. Then this number is send to every active process in the distributed system.

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

1. The Bully Algorithm :-

This algorithm applies to system where every process can send a message to every other process in the system.

As a first example, consider the bully algorithm devised by Garcia-Molina (1982). When a process notices that the coordinator is no longer responding to request, it initiates an election. A process, P, holds an election as follows.

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.

If a process that was previously down back up, it holds an election. If it happens to be 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."

Algorithm =

Suppose process P sends a message to the coordinator.

1. If coordinator does not respond to it with in a time interval T, then it is assumed that coordinator has failed.
2. Now process P sends election message to every process with high priority number.
3. It waits for responses, if no one responds for time interval T then process P elects itself as a coordinator.
4. Then it sends a message to all lower priority number processes that it is elected as their new coordinator.

5. However, if an answer is received within time T from any other process q .

- I) Process P again waits for time interval T' to receive another message from q that it has been elected as coordinator.
- II) If q doesn't respond within time interval T' then it is assumed to have failed and algorithm is restarted.

2. The Ring Algorithm -

This algorithm applies to systems organized as a ring (logically or physically). In this algorithm we assume that the link between the process are unidirectional and every process can message to the process on its right only. Data structure that this algorithm uses is active list, a list that has priority number of all active processes in the system.

Algorithm -

1. If process p_1 detects a coordinator failure, it creates new active list which is empty initially. It sends election

message to its neighbour on right and adds number 1 to its active list.

2. If process P2 receives message elect from processes on left, it responds in 3 ways:

- I) If message received does not contain 1 in active list then P1 adds 2 to its active list and forwards the message.
- II) If this is the first election message it has received or sent, P1 creates new active list with numbers 1 and 2. If then followed election message 1 followed by 2.
- III) If process P1 receives its own election message 1 then active list for P1 now contains numbers of all the active processes in the system. Now Process P1 detects highest priority number from list and elects it as the new coordinator.

