Table of Contents

Election Algorithm

Many distributed algorithm require one process to acts as coordinator, initiator, or otherwise perform some special role.

The goal of an election algorithm is to ensure that when an election starts concludes with all process agreeing on who the new coordinator is to be.

Types Of Election Algorithm

1. Token ring algorithm
2. Bully Algorithm

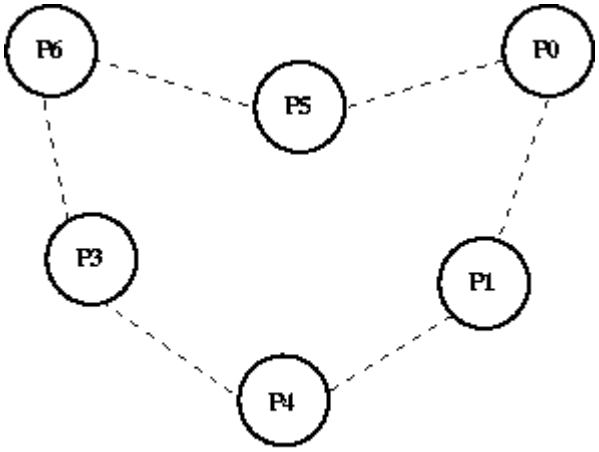
1. Token ring algorithm

When any process notice that the coordinator is not functioning, it builds an ELECTION MESSAGE containing its own process number and sends the message to its successor.

Example:

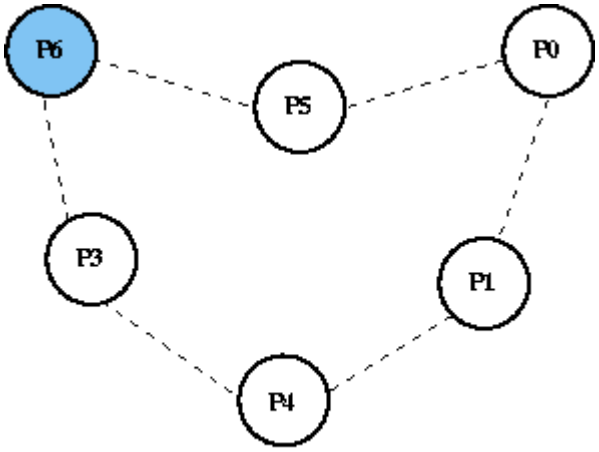
We start with 6 processes, connected in a logical ring. Process 6 is the leader, as it has the

highest number.



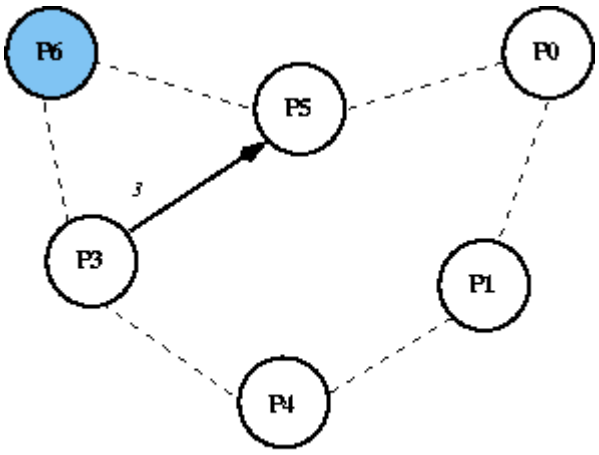
Token Ring Election Algorithm: Step 0

Process 6 fails.



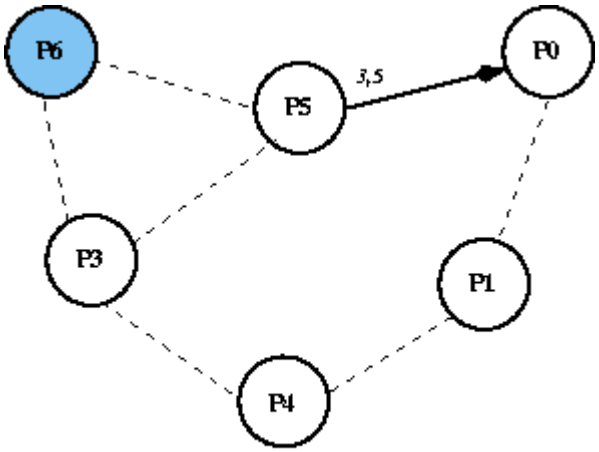
Token Ring Election Algorithm: Step 1

Process 3 notices that Process 6 does not respond. So it starts an election, sending a message containing its id to the next node in the ring.



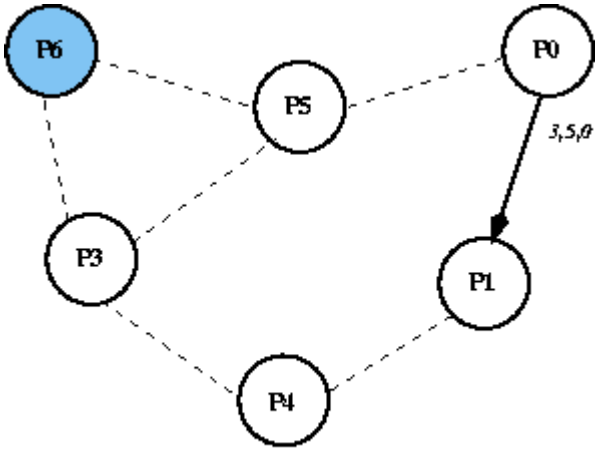
Token Ring Election Algorithm: Step 2

Process 5 passes the message on, adding its own id to the message.



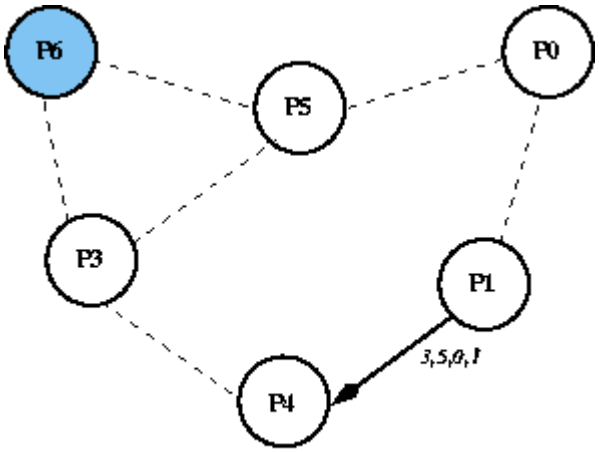
Token Ring Election Algorithm: Step 3

Process 0 passes the message on, adding its own id to the message.



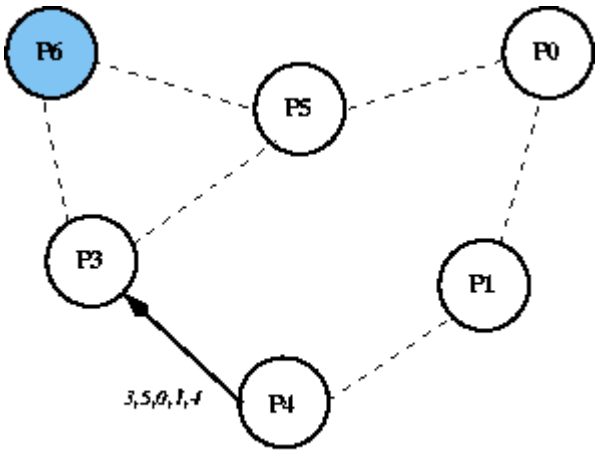
Token Ring Election Algorithm: Step 4

Process 1 passes the message on, adding its own id to the message.



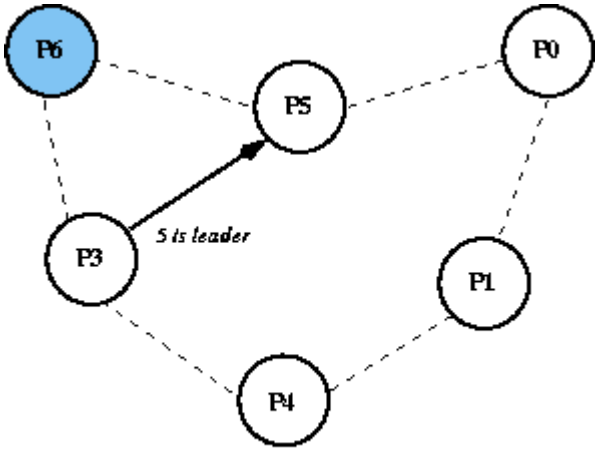
Token Ring Election Algorithm: Step 5

Process 4 passes the message on, adding its own id to the message.



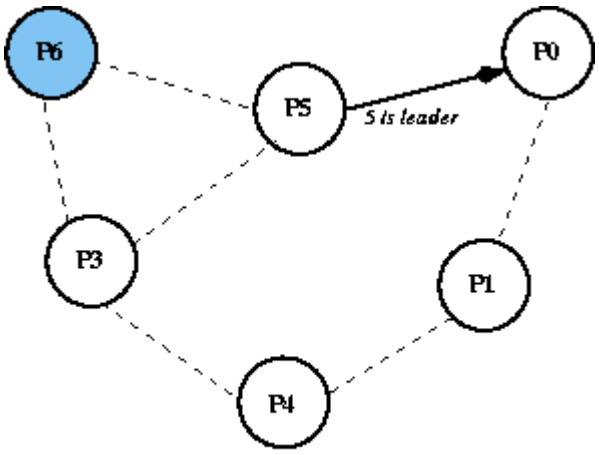
Token Ring Election Algorithm: Step 6

When Process 3 receives the message back, it knows the message has gone around the ring, as its own id is in the list. Picking the highest id in the list, it starts the coordinator message "5 is the leader" around the ring.



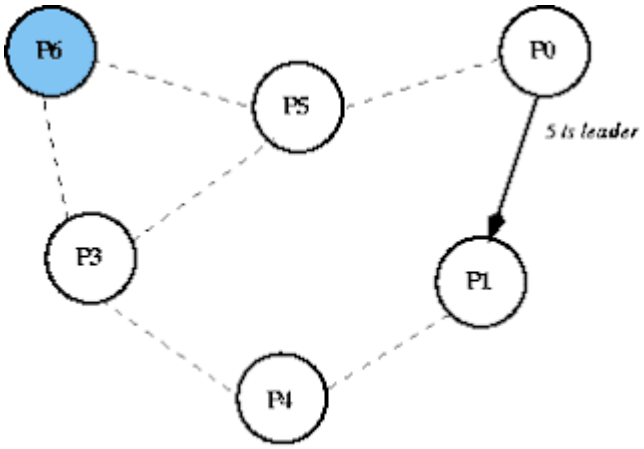
Token Ring Election Algorithm: Step 7

Process 5 passes on the coordinator message.



Token Ring Election Algorithm: Step 8

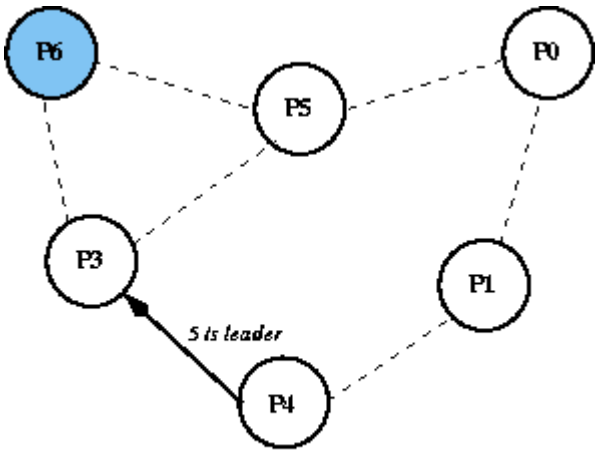
Process 0 passes on the coordinator message.



Token Ring Election Algorithm: Step 9

Process 1 passes on the coordinator message.

Process 4 passes on the coordinator message.



Token Ring Election Algorithm: Step 11

Process 3 receives the coordinator message, and stops it.

---

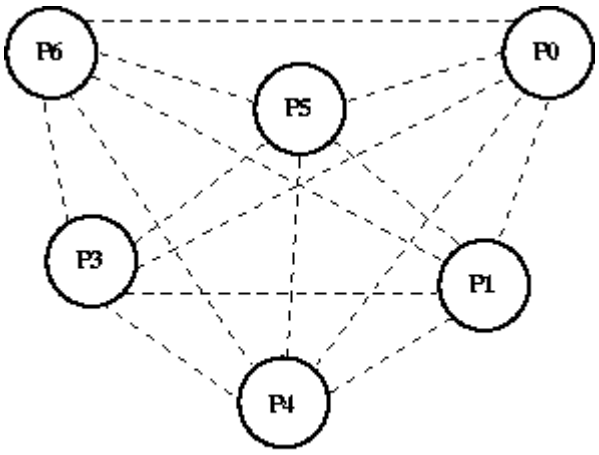
2. Bully Algorithm

When any process notices that the coordinator is no longer responding to request, it initiates an ELECTION.

The process holds an ELECTION message as follows:

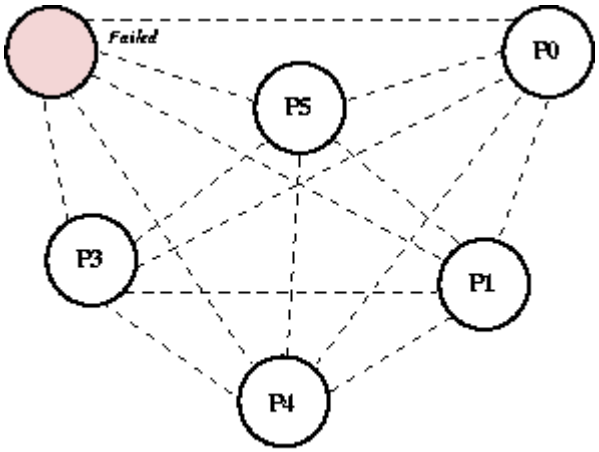
Example:

We start with 6 processes, all directly connected to each other. Process 6 is the leader, as it has the highest number.



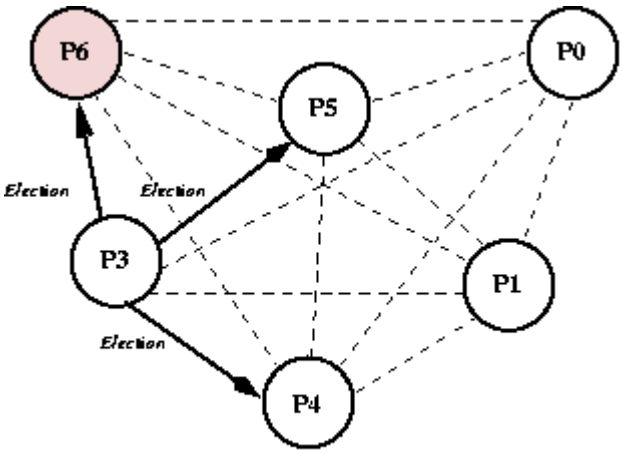
Bully Algorithm: Step 0

Process 6 fails.



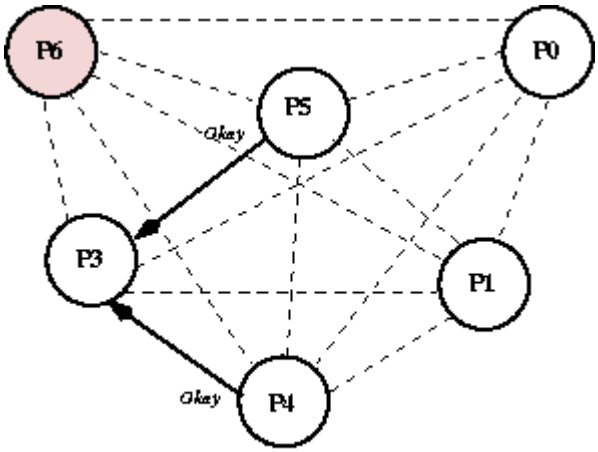
Bully Algorithm: Step 1

Process 3 notices that Process 6 does not respond. So it starts an election, notifying those processes with ids greater than 3.



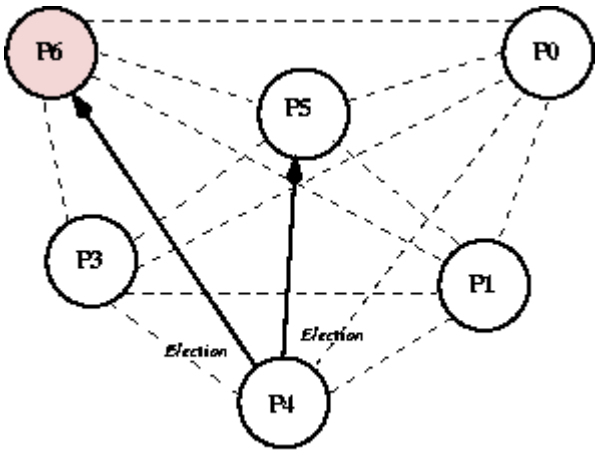
Bully Algorithm: Step 2

Both Process 4 and Process 5 respond, telling Process 3 that they'll take over from here.



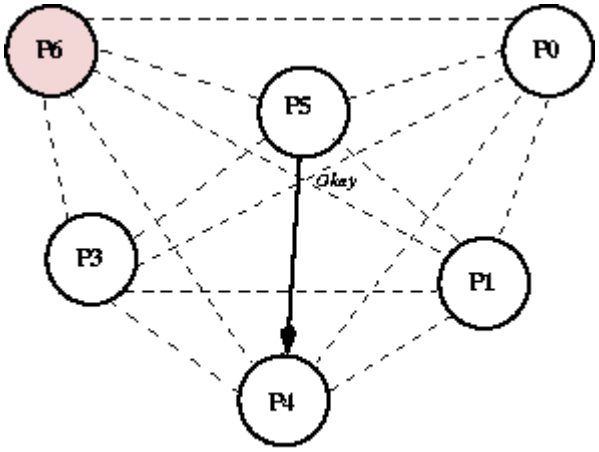
Bully Algorithm: Step 3

Process 4 sends election messages to both Process 5 and Process 6.



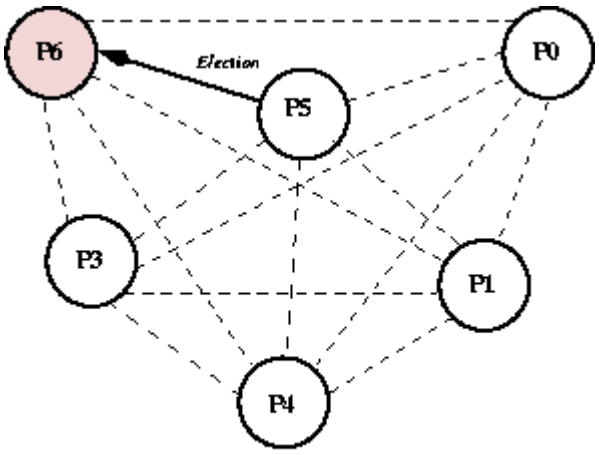
Bully Algorithm: Step 4

Only Process 5 answers and takes over the election.



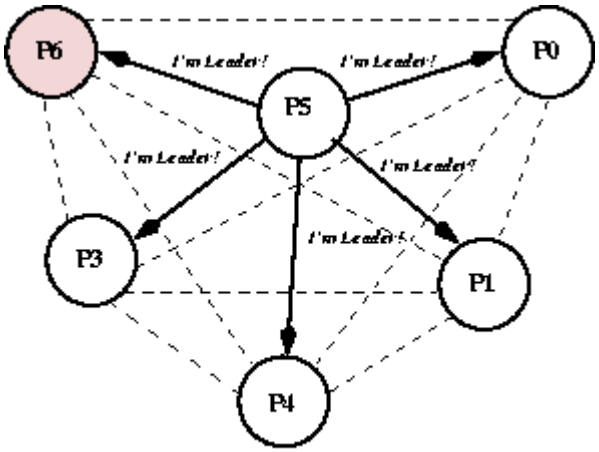
Bully Algorithm: Step 5

Process 5 sends out only one election message to Process 6.



Bully Algorithm: Step 6

When Process 6 does not respond Process 5 declares itself the winner.



Bully Algorithm: Step 7

Related posts:

1. Advantages Disadvantages of DS
2. Distributed computing models
3. Goals of DS
4. Hardware software concepts
5. Issues in designing ds

6. Design and Implementation Issues DS
7. Structure of share memory space
8. DSM Architecture & its Types
9. File Application & Fault tolerance
10. File service architecture
11. Desirable features of good distributed file system
12. Distributed shared memory
13. RPC messages
14. RPC Implementation
15. RPC mechanism
16. Client server communication
17. Datarepresentation and Marshalling
18. Communication between distributed objects
19. Load distributing algorithm
20. Task migration and its issues
21. Deadlock issues in deadlock detection & resolution
22. Distributed Scheduling-Issues in Load Distributing
23. Characterstics of Multimedia Data
24. Case Study of Distributed System
25. Distributed multimedia system
26. Distributed DBMS
27. Advantages of DDBMS over centralised DBMS