LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY

A UGC Autonomous Institution | Approved by AICTE | Recognized by Govt. Of Telangana
Accredited by NBA& NAAC 'A' Grade | Included of 2F of UGC

Department of Computer Science And Engineering

Academic Year: 2026-2027

Student Roll Number: 160923733152

Student Name: MOHAMMED UFRAAN

Year: 04 **Semester:** 07 **Branch & Section:** CSE-4A2

Course Name: DISTRIBUTED SYSTEMS

Course Code: U23CS701

Assignment Number: 01

Date of Assignment: 24/8/26 **Due date:** 29/8/26 **Date of Submission:**

Marks Obtained: 10

Remarks:

Signature of Faculty with date

Mohammed Ufraan
160923733152

LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Department of Computer Science Engineering-

Name of the Subject: DISTRIBUTED SYSTEMS

Assignment – 1 Date : 24/08/26

Answer all the Questions. **CSE-C** Max.Marks:**10**

Q.No.	Question	MARKS
1.	Explain inter-process communication mechanisms in Distributed Systems. Discuss the different methods used for IPC.	5M
2.	What is Remote Method Invocation (RMI) and Explain the Bully algorithm with a suitable example	5M

Bloom's Taxonomy Levels (BTL)		
BTL1 –Remember	BTL2 Understand	BTL3 –Apply
BTL4-Analyze	BTL5 –Evaluate	BTL6 –Create

NOTE:

1.Answer all Questions.

2.Answer to the assignment questions are to be written on A4 size papers only.

3.Each page of the assignment paper should contain full roll number of the student.

4.The hard copy of the assignment is to be submitted on or before 29/08/26.

Mohammed Ufraan
160923733152

01.) Explain inter-process communication mechanisms in Distributed Systems. Discuss the different methods used for IPC.

Ans: Inter-Process communication refers to mechanisms that allow processes running on the same or different machines in a distributed system to exchange data and coordinate their activities.
It is fundamental to distributed systems for enabling cooperation between autonomous processes.

Key Characteristics:
* Processes run independently on different hosts.
* Communication through network protocols.
* Handles heterogeneity (different platforms, languages).
* Subject to network delays and failures.

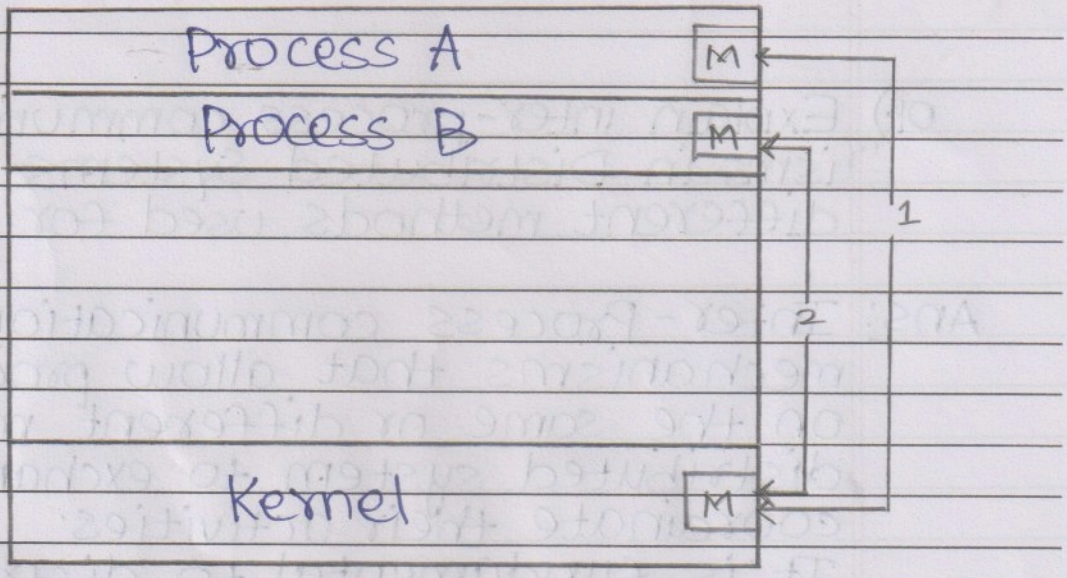
Main IPC Methods:

1. MESSAGE PASSING:

Processes communicate by explicitly sending and receiving messages.

Characteristics:
* Asynchronous: sender doesn't wait for reply.
* Synchronous: sender waits for acknowledgement.
* Point-to-point or multicast.
* Example: Message queues, email systems.

Mohammed Ufraan
160923733152



Message Passing

Advantages:
* Asynchronous & natural for distr. systems
* Loose coupling between processes.
* Works over networks.

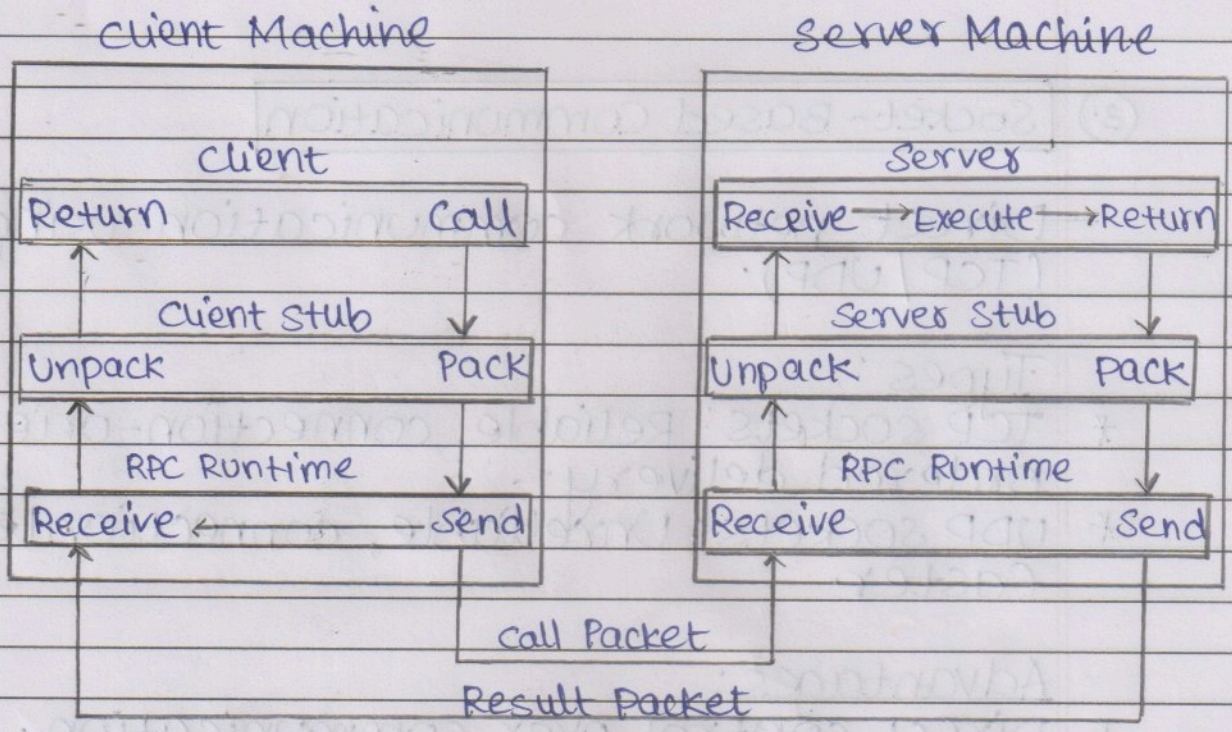
Disadvantages:
* Slow communication overhead.
Completely in synchronisation.

(2.) REMOTE PROCEDURE CALL

A process calls a function / procedure on a remote machine as if it were local.

How it Works:
* Client invokes remote procedure with arguments.
* Stub marshals arguments and sends to server.
* Server executes procedure.
* Results returned to client.

Mohammed Ufraan
160923733152



Remote Procedural Call (RPC)

Example:

* Sun RPC
* ONC RPC
* JSON RPC

Advantages:

* Familiar programming model.
* Hides network complexity
* Synchronous communication.

Disadvantages:

* Tight coupling.
* Network failures visible to application.
* Cannot pass complex data structures easily.

Mohammed Ufraan
160923733152

(3.) Socket-Based Communication

Direct network communication using sockets (TCP/UDP).

Types:
* TCP sockets: Reliable, connection-oriented, ordered delivery.
* UDP sockets: Unreliable, connectionless, faster.

Advantages:
* Direct control over communication.
* Flexible data formats.
* Wide platform support.

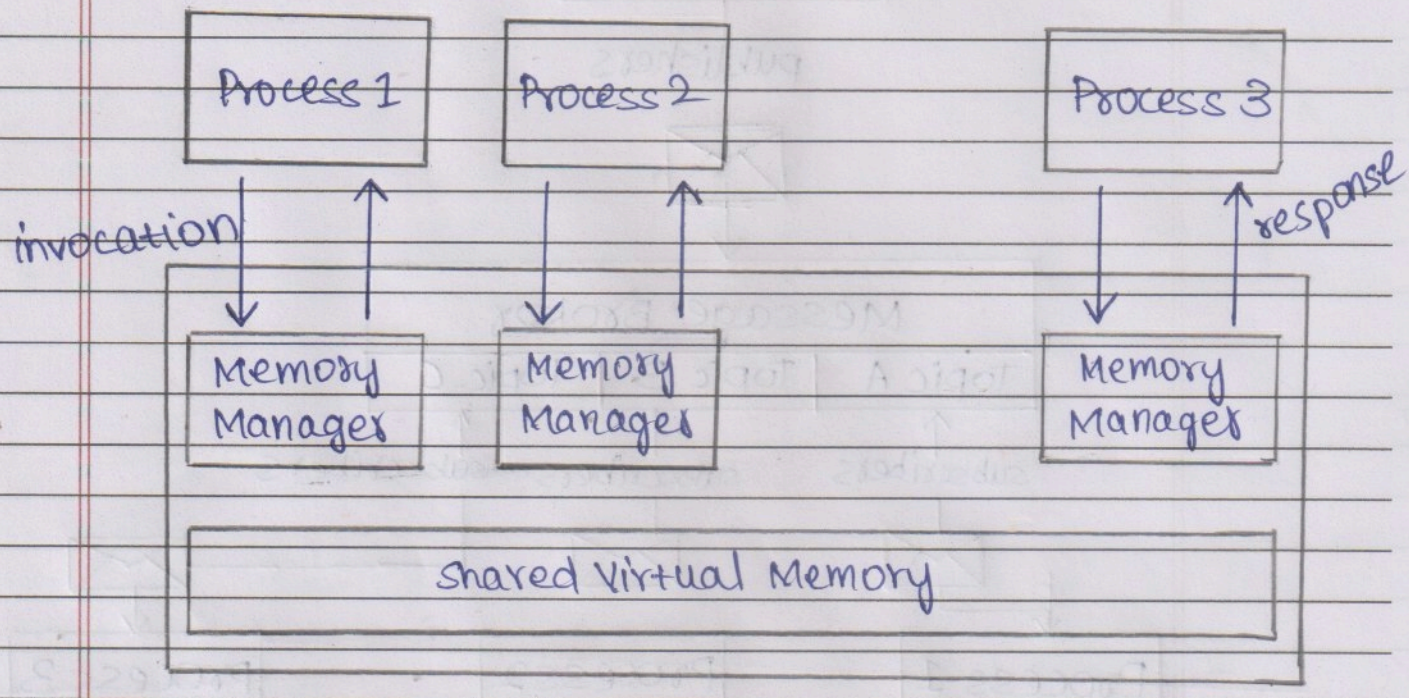
Disadvantages:
* Low-level programming.
* Error-handling complexity.
* Data serialisdtion overhead.

(4.) Shared Memory

Processes access common memory regions for data exchange.

Methods:
* Shared variables protected by locks.
* Semaphores and monitors for synchronisation.
* Memory-mapped files.

Mohammed Ufraan
160923733152



Distributed Shared Memory

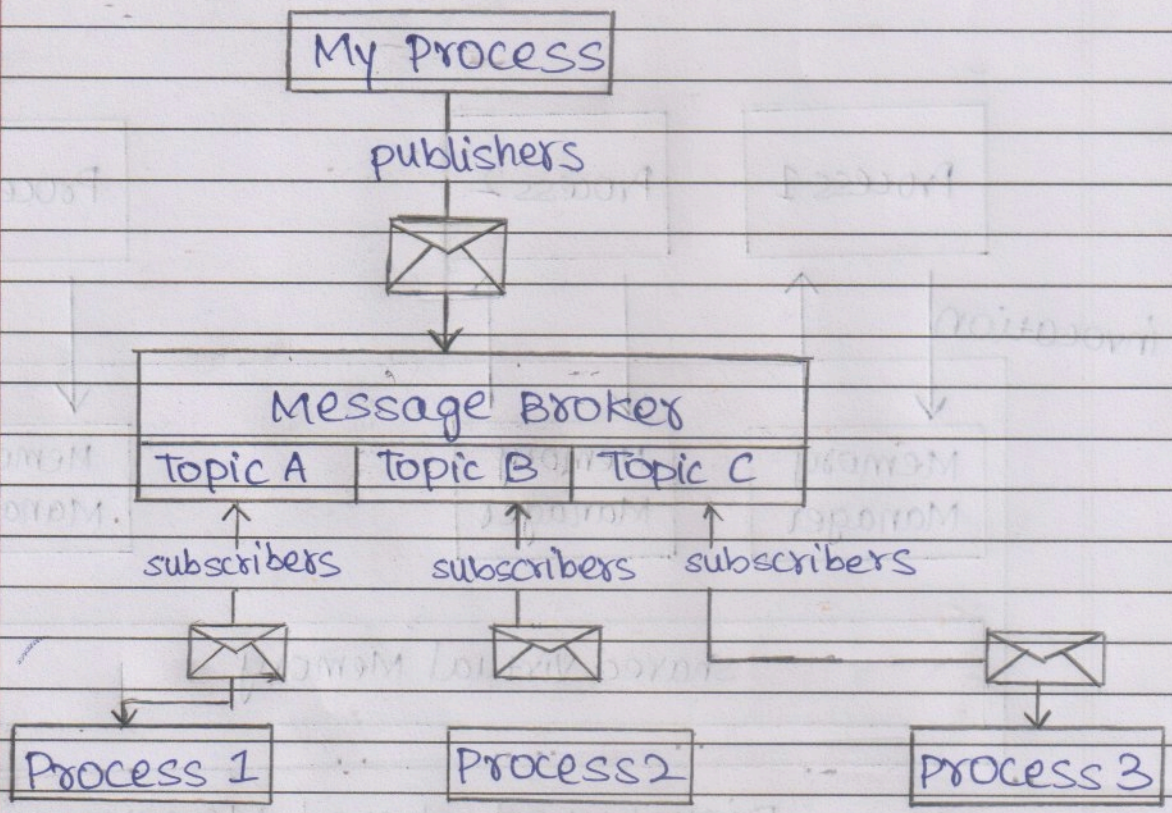
Limitations:
* Only works on same machine (limited scalability).
* Cache coherence issues.
* Difficult to maintain consistency across machines.

⑤ Publish-Subscribe Communication

Processes publish events; other process subscribe to receive notifications.

Characteristics:
* Asynchronous communication.
* Loose coupling b/w publisher and subscriber.
* Event broker/bus intermediary

Mohammed Ufraan
160923733152



Pub-Sub Architecture Diagram

Example:
* Rabbit MQ.
* Apache Kafka.

Advantages:
* Highly decoupled.
* Scalable.
* Supports one-to-many communication.

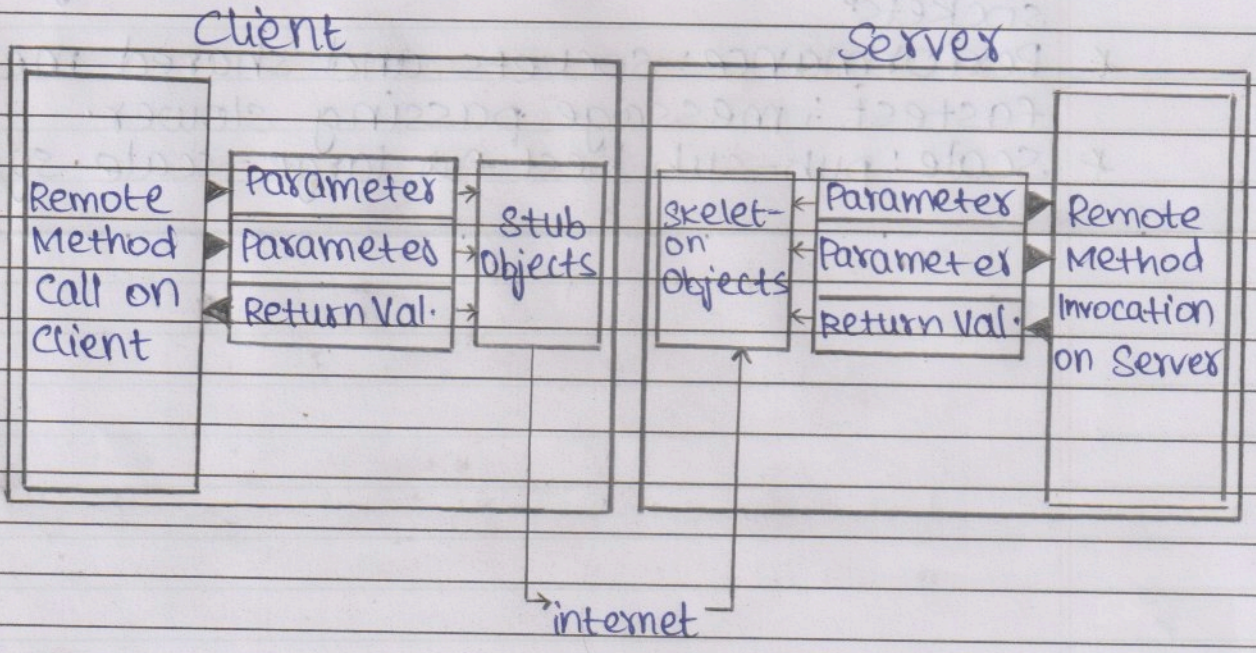
⑥ Remote Method Invocation

Object-oriented version of RPC where objects on different machines invoke methods on each other.

Mohammed Ufraan
160923733152

Features:

* Works with objects and serialization.
* Automatic marshalling of objects.
* Language and platform specific (Java RMI).



Java RMI Architecture

Comparison Table:

Method	Coupling	Speed	Reliability	Use Case
Message Passing	Loose	Slow	High	Asyn apps
RPC	Tight	Medium	Medium	Simple Calls
Socket	Tight	Medium	Medium	Direct Control
Shared Memory	N/A	Fast	High	Same Machine
Pub-Sub	Loose	Medium	High	Event Systems
RMI	Medium	Medium	Medium	Object Systems

Mohammed Ufraan
160923733152

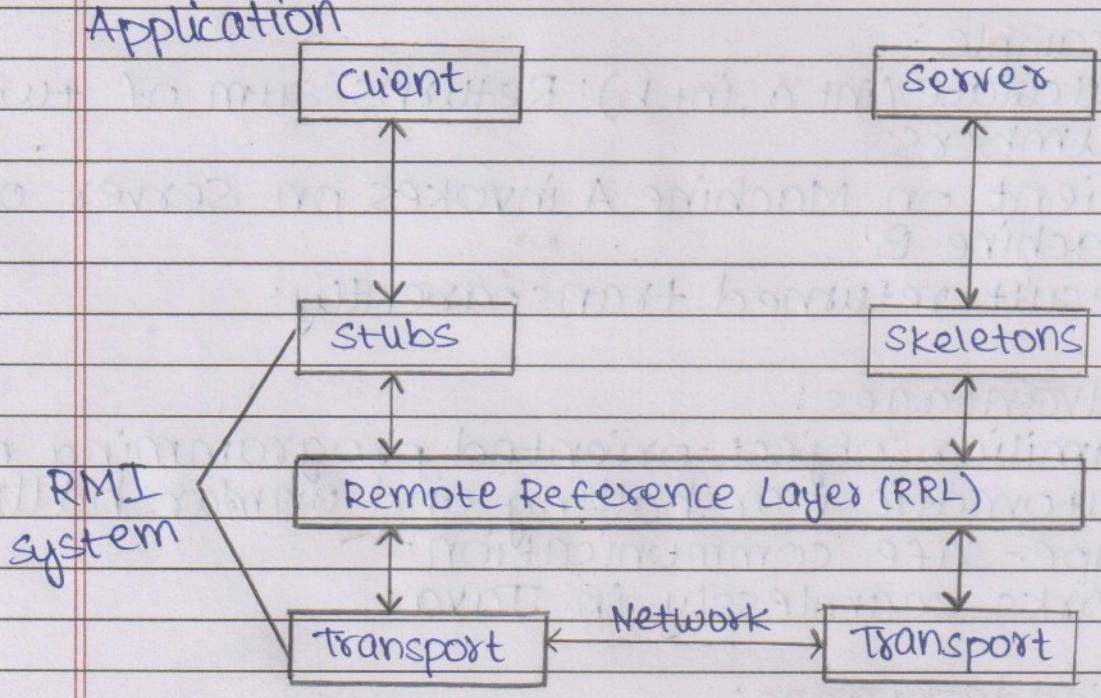
Selection Criteria:

* Synchronous vs Asynchronous: RPC/RMI for sync; message parsing for async.
* Coupling Requirements: Loose coupling uses RPC/sockets.
* Performance: sockets and shared memory fastest; message passing slower.
* Scale: pub-sub best for large-scale systems.

Mohammed Ufraan
160923733152

02.) What is Remote Method Invocation (RMI) and explain the Bully algorithm with a suitable example.

Ans: Remote Method Invocation is an object-oriented mechanism that allows a java object on one machine across a network, transparently treating remote calls like local method calls.



Remote Method Invocation

→ Stub and Skeleton:

* Stub (client-side proxy): Marshals method arguments and sends to remote object; unmarshals return values.
* Skeleton (server-side proxy): Receives method invocation, invokes actual method, returns results.
* Automatically generated from remote interface.

Mohammed Ufraan
160923733152

→ **Remote Interface:**
* Extends java.rmi.Remote
* Declares methods that can be invoked remotely
* Must throw RemoteException.

→ **RMI Registry:**
* Naming service for locating remote objects.
* Binds object names to references.
* Clients look up remote objects by name.

Example:
* Calculate(int a, int b): Returns sum of two numbers
* Client on Machine A invokes on server on Machine B.
* Result returned transparently.

→ **Advantages:**
* Familiar object-oriented programming model.
* Automatic marshalling and unmarshalling.
* Type-safe communication.
* Works seamlessly in Java.

→ **Disadvantages:**
* Java specific (not language-independent).
* Network failures exposed to application.
* Performance overhead due to serialization.
* Tight coupling between client and server.

Mohammed Ufraan
160923733152

The Bully Algorithm is a distribution algorithm for coordinator (leader) election in a distribution system. It elects the process with the highest ID as the coordinator when the current coordinator fails.

Key Assumptions:

* Each process has a unique ID (processes ranked by ID).
* All processes know IDs of other processes.
* System is synchronous with known timeout.
* Process that fails can recover later.

Algorithm Steps:

STEP (1) When a Process Detects Coordinate Failure:

* Starts election process by sending ELECTION message.
* Sends to all processes with higher ID.

STEP (2) When a process receives ELECTION Message:

* Sends back OK message (acknowledging candidacy).
* Starts its own election (sends ELECTION to higher IDs).

STEP (3) When a process sends no OK Responses:

* It becomes the new coordinate. Broadcasts COORDINATOR message to all
* lower-ID processes.

STEP (4) When a Process Recovers:

* It starts election process
* Since it may have highest ID, can become coordinator.

Mohammed Ufraan
160923733152

Worked Example:

System:
5 processes with IDs : P1, P2, P3, P4, P5 (P5 has highest ID).

Initial State:
P5 is coordinator.

	P1	P2	P3	P4	P5
Status	Running	Running	Running	Running	Running

STEP (1) **P5 (Coordinator) Fails:**
* Process P2 detects timeout from P5 and initiates election.

STEP (2) **P2 sends ELECTION message:**
* P2 sends ELECTION to P3, P4, P5 (higher IDs)
* Election initiated by P2 :
  ELECTION → {P3, P4, P5}

STEP (3) **Responses:**
* P3 receives ELECTION from P2 : Sends OK to P2, starts election to P4, P5.
* P4 receives ELECTION from P2 : sends OK to P2, starts election to P5.
* P4 receives ELECTION from P3 : sends OK to P3, starts election to P5.
* P5 is down : No response.

Mohammed Ufraan
160923733152

STEP (4.) P4's Election:

* P4 sends ELECTION to P5 (higher ID). P5 doesn't respond (down).
* Since no OK response, P4 becomes coordinator and broadcasts COORDINATOR message to all.

Final State:

	P1	P2	P3	P4	P5
Status	Running	Running	Running	Running	Running

Advantages:

* Simple and easy to understand.
* Guarantees highest-ID process becomes coordinator.
* Handles multiple simultaneous failures.
* Recovers from process failures.

Disadvantages:

* High message overhead ($O(n^2)$ in worst case).
* Not fair: higher IDs always win.
* Poor scalability for large systems.
* Requires knowledge of all process IDs.

Comparison:

Algorithm	Messages	Fair	Complexity
Bully	$O(n^2)$	No	Simple
Ring	$O(n)$	Yes	Medium
Candidate	$O(n \log n)$	Yes	Complex.