Distributed Systems - COMP SCI 3012
Collaborative Session 3

Gia Bao Hoang - a1814824

1. Overview:

The Adelaide Suburbs Council requires a robust and resilient voting system to elect the next council president. Given the unique communication challenges faced by members M1, M2, and M3, a Paxos protocol is being employed to ensure a fault-tolerant voting process.

2. Communication:

2.1. Mode of Communication:

- Communication between council members is carried out via sockets.

2.2. Communication Reliability:

- As members have variable response times, timeouts will be implemented to ensure progress.

- If no response is received from a member within a specific timeframe, the system will proceed with the available responses.

3. Communication Protocol:

3.1. Proposal Requests:

1. **Prepare Request**: Proposer sends out a message with a unique proposal number.

2. **Accept Request**: If the proposer gets a majority of promises, they send out the value they wish to propose with the proposal number.

3.2. Response Types:

1. **Promise**: Sent by acceptors in response to a prepare request. Can include previously accepted values.

2. **Accepted**: Sent by acceptors when they accept a proposal.

3. **NACK**: Sent when a request is rejected because of a higher-seen proposal number.

3.3. Decision Phase:

- Once a proposal is accepted by a majority, the proposer sends out a "Decide" message, declaring the president.

4. Normal Operation Mode:

1. A proposer (member wanting to be president) sends out a "Prepare" request with a unique proposal number.

2. Acceptors respond with a "Promise" or a "NACK".

3. If a majority of promises is achieved, the proposer sends out an "Accept Request".

4. Acceptors respond with "Accepted" or "NACK".

5. Once a majority accepts, the proposer sends out a "Decide" message.

5. Possible Breaking Points:

1. **Variable Response Times**: Members have differing response rates. The system should not halt if it doesn't receive immediate responses.

2. **Lost Messages**: Messages might not reach M3 or could be delayed for M2.

3. **Concurrent Proposals**: Two members might send out proposals simultaneously, causing potential contention.

5.1. Reproduction:

- Simulate a delay in M2's and M3's responses.

- Drop some messages intended for M3 to mimic lost messages.

- Send simultaneous proposals from M1 and M2.

6. Multiple-Client Mode:

6.1. Breaking Points:

1. **Contention**: Multiple members might propose themselves concurrently.

2. **Network Saturation**: A large number of messages exchanged might clog the network.

7. Testing Harness Pseudo-code:

```pseudo

function testPaxosProtocol() {

    simulateMembers(M1, M2, M3, M4-M9)

    M1.sendPrepareRequest()

    waitForResponses()

    validatePromisesReceived()


    M2.sendPrepareRequest()

    waitForResponses()

    validatePromisesReceived()


    M1.sendAcceptRequest()

    waitForResponses()

    validateAcceptancesReceived()


    M3.sendPrepareRequest()

    dropSomeMessagesFor(M3)

    waitForResponses()

    validatePromisesReceived()


    M1.sendDecideMessage()

    validateDecisionAcrossAllMembers()
```

```
}

function simulateDelayedResponsesFor(member) {
    // Introduce artificial delays in member's responses
}

function dropSomeMessagesFor(member) {
    // Drop some messages intended for member
}
```

7.1. Testing Scenarios:

1. **Normal Operation**: All members respond timely and proposals are made one after the other.

2. **Delayed Responses**: M2 and M3 have delayed responses, and the system's timeout resilience is tested.

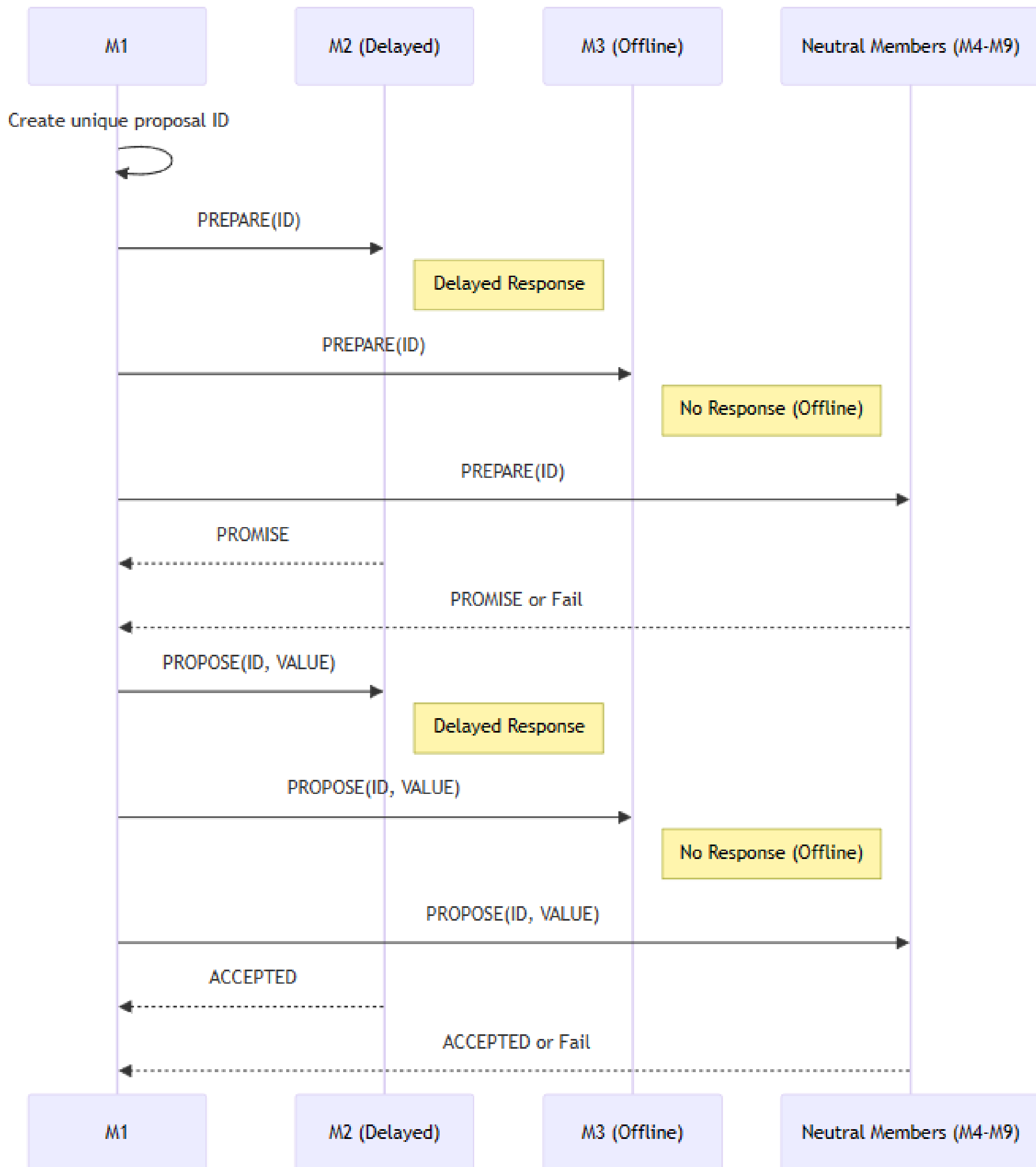
3. **Lost Messages**: Messages intended for M3 are dropped to test the system's fault tolerance.

4. **Concurrent Proposals**: M1 and M2 simultaneously send proposals to test contention handling.

The testing harness will automate these scenarios, validate the expected outcomes, and report any discrepancies or failures.

8. Design Sketch:

The illustrated sequence diagram depicts the Paxos voting protocol as implemented for the election of the Suburbs Council President. In this scenario, M1 takes the initiative to create and disseminate a unique proposal. The diagram highlights the interactions between M1, M2 (which has delayed responses), M3 (which is offline), and the Neutral Members (M4-M9). The two-phase commitment of the Paxos protocol, encompassing the PREPARE-PROMISE and PROPOSE-ACCEPT stages, is showcased, emphasizing the direct communication patterns and unique characteristics of each member.



With this design, the council can implement a robust Paxos-based voting system that remains resilient against varying response times, lost messages, and concurrent proposals.