CS7610: Lab 3 report

Madhukara S Holla (sholla.m@northeastern.edu)

This program implements a distributed membership service where peers (nodes) dynamically join and leave the group, with one designated leader maintaining the membership list. The system also includes a failure detection mechanism and leader election when the leader crashes.

Initialization

The program starts by reading a **hostsfile**, which contains information about all possible peers in the network. Each peer assigns itself a unique ID based on its position in the file and identifies the leader, who is always the first peer.

All peers establish communication over TCP on port 8080 for reliable message exchange and use UDP on port 8081 for heartbeat messages.

Membership Management

1. Each peer maintains a **Membership** struct containing the current **view_id** and a list of alive peers.
2. When a new peer joins, it sends a **JOIN** message to the leader, who coordinates the membership update by sending a **REQ** message to all other peers.
3. Once all peers acknowledge the request with **OK** messages, the leader increments the **view_id** and broadcasts a **NEWVIEW** message to update the membership list.

Failure Detection

1. Peers exchange heartbeat messages periodically over UDP to detect failures.
2. If a peer misses two consecutive heartbeats from another peer, it declares the peer dead and reports it.
3. If the leader crashes, a new leader is automatically elected based on the lowest peer ID.

Leader Election and Recovery

1. When the leader crashes, the peer with the lowest ID becomes the new leader.
2. The new leader sends a **NEWLEADER** message to all peers to check for any pending operations, such as membership updates that were not completed before the leader crashed.
3. The new leader finalizes any pending operations and sends a **NEWVIEW** message to ensure all peers have an updated membership list.

Key Considerations

1. **Concurrency**: The program uses Goroutines to handle multiple tasks concurrently, including listening for incoming messages, sending heartbeat messages, and running the failure detector.
2. **Synchronization**: Mutex locks are used to ensure consistency when updating membership lists and handling failure detection.

Testcase 1

```
docker compose -f .\docker-compose-testcase-1.yml up
Attaching to five, four, one, three, two

one   | {peer_id: 1, view_id: 0, leader: 1, memb_list: [1]}

one   | {peer_id: 1, view_id: 1, leader: 1, memb_list: [1,2]}
two   | {peer_id: 2, view_id: 1, leader: 1, memb_list: [1,2]}

one   | {peer_id: 1, view_id: 2, leader: 1, memb_list: [1,2,3]}
two   | {peer_id: 2, view_id: 2, leader: 1, memb_list: [1,2,3]}
three | {peer_id: 3, view_id: 2, leader: 1, memb_list: [1,2,3]}

one   | {peer_id: 1, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
two   | {peer_id: 2, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
three | {peer_id: 3, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
four  | {peer_id: 4, view_id: 3, leader: 1, memb_list: [1,2,3,4]}

one   | {peer_id: 1, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
two   | {peer_id: 2, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
three | {peer_id: 3, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
four  | {peer_id: 4, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
five  | {peer_id: 5, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
```

Testcase 2

```
docker compose -f .\docker-compose-testcase-2.yml up
Attaching to five, four, one, three, two
```

```
one      | {peer_id: 1, view_id: 0, leader: 1, memb_list: [1]}
one      | {peer_id: 1, view_id: 1, leader: 1, memb_list: [1,2]}
two      | {peer_id: 2, view_id: 1, leader: 1, memb_list: [1,2]}
one      | {peer_id: 1, view_id: 2, leader: 1, memb_list: [1,2,3]}
two      | {peer_id: 2, view_id: 2, leader: 1, memb_list: [1,2,3]}
three    | {peer_id: 3, view_id: 2, leader: 1, memb_list: [1,2,3]}
one      | {peer_id: 1, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
two      | {peer_id: 2, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
three    | {peer_id: 3, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
four     | {peer_id: 4, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
one      | {peer_id: 1, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
two      | {peer_id: 2, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
three    | {peer_id: 3, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
four     | {peer_id: 4, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
five     | {peer_id: 5, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}

five     | {peer_id:5, view_id: 4, leader: 1, message:"crashing"}
five exited with code 0

four     | {peer_id:4, view_id: 4, leader: 1, message:"peer 5 unreachable"}
three    | {peer_id:3, view_id: 4, leader: 1, message:"peer 5 unreachable"}
two      | {peer_id:2, view_id: 4, leader: 1, message:"peer 5 unreachable"}
one      | {peer_id:1, view_id: 4, leader: 1, message:"peer 5 unreachable"}
```

Testcase 3

```
docker compose -f .\docker-compose-testcase-3.yml up
Attaching to five, four, one, three, two

one      | {peer_id: 1, view_id: 0, leader: 1, memb_list: [1]}
one      | {peer_id: 1, view_id: 1, leader: 1, memb_list: [1,2]}
two      | {peer_id: 2, view_id: 1, leader: 1, memb_list: [1,2]}
one      | {peer_id: 1, view_id: 2, leader: 1, memb_list: [1,2,3]}
two      | {peer_id: 2, view_id: 2, leader: 1, memb_list: [1,2,3]}
three    | {peer_id: 3, view_id: 2, leader: 1, memb_list: [1,2,3]}
one      | {peer_id: 1, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
two      | {peer_id: 2, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
three    | {peer_id: 3, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
four     | {peer_id: 4, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
one      | {peer_id: 1, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
two      | {peer_id: 2, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
three    | {peer_id: 3, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
four     | {peer_id: 4, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
five     | {peer_id: 5, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}

five     | {peer_id:5, view_id: 4, leader: 1, message:"crashing"}
five exited with code 0
two      | {peer_id:2, view_id: 4, leader: 1, message:"peer 5 unreachable"}
one      | {peer_id:1, view_id: 4, leader: 1, message:"peer 5 unreachable"}
two      | {peer_id: 2, view_id: 5, leader: 1, memb_list: [1,2,3,4]}
three    | {peer_id: 3, view_id: 5, leader: 1, memb_list: [1,2,3,4]}
four     | {peer_id: 4, view_id: 5, leader: 1, memb_list: [1,2,3,4]}
one      | {peer_id: 1, view_id: 5, leader: 1, memb_list: [1,2,3,4]}

four     | {peer_id:4, view_id: 5, leader: 1, message:"crashing"}
four exited with code 0
two      | {peer_id:2, view_id: 5, leader: 1, message:"peer 4 unreachable"}
one      | {peer_id:1, view_id: 5, leader: 1, message:"peer 4 unreachable"}
two      | {peer_id: 2, view_id: 6, leader: 1, memb_list: [1,2,3]}
one      | {peer_id: 1, view_id: 6, leader: 1, memb_list: [1,2,3]}
three    | {peer_id: 3, view_id: 6, leader: 1, memb_list: [1,2,3]}

three    | {peer_id:3, view_id: 6, leader: 1, message:"crashing"}
three exited with code 0
two      | {peer_id:2, view_id: 6, leader: 1, message:"peer 3 unreachable"}
one      | {peer_id:1, view_id: 6, leader: 1, message:"peer 3 unreachable"}
two      | {peer_id: 2, view_id: 7, leader: 1, memb_list: [1,2]}
one      | {peer_id: 1, view_id: 7, leader: 1, memb_list: [1,2]}

two      | {peer_id:2, view_id: 7, leader: 1, message:"crashing"}
two exited with code 0
one      | {peer_id:1, view_id: 7, leader: 1, message:"peer 2 unreachable"}
one      | {peer_id: 1, view_id: 8, leader: 1, memb_list: [1]}
```

Testcase 4

Leader crashing after sending REQ to delete the last node (peer 5 in this case)

```
docker compose -f .\docker-compose-testcase-4.yml up
Attaching to five, four, one, three, two

one   | {peer_id: 1, view_id: 0, leader: 1, memb_list: [1]}
one   | {peer_id: 1, view_id: 1, leader: 1, memb_list: [1,2]}
two   | {peer_id: 2, view_id: 1, leader: 1, memb_list: [1,2]}
one   | {peer_id: 1, view_id: 2, leader: 1, memb_list: [1,2,3]}
two   | {peer_id: 2, view_id: 2, leader: 1, memb_list: [1,2,3]}
three | {peer_id: 3, view_id: 2, leader: 1, memb_list: [1,2,3]}
one   | {peer_id: 1, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
two   | {peer_id: 2, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
three | {peer_id: 3, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
four  | {peer_id: 4, view_id: 3, leader: 1, memb_list: [1,2,3,4]}
one   | {peer_id: 1, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
two   | {peer_id: 2, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
three | {peer_id: 3, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
four  | {peer_id: 4, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}
five  | {peer_id: 5, view_id: 4, leader: 1, memb_list: [1,2,3,4,5]}

one   | {peer_id:1, view_id: 4, leader: 1, message:"crashing"}
one exited with code 0
four  | {peer_id:4, view_id: 4, leader: 1, message:"peer 1 (leader)
unreachable"}
five  | {peer_id:5, view_id: 4, leader: 1, message:"peer 1 (leader)
unreachable"}
three | {peer_id:3, view_id: 4, leader: 1, message:"peer 1 (leader)
unreachable"}
two   | {peer_id:2, view_id: 4, leader: 1, message:"peer 1 (leader)
unreachable"}

four  | {peer_id:4, view_id: 4, leader: 2, message:"peer 1 unreachable"}
five  | {peer_id:5, view_id: 4, leader: 2, message:"peer 1 unreachable"}
three | {peer_id:3, view_id: 4, leader: 2, message:"peer 1 unreachable"}
two   | {peer_id:2, view_id: 4, leader: 2, message:"peer 1 unreachable"}

four  | {peer_id: 4, view_id: 5, leader: 2, memb_list: [2,3,4,5]}
three | {peer_id: 3, view_id: 5, leader: 2, memb_list: [2,3,4,5]}
five  | {peer_id: 5, view_id: 5, leader: 2, memb_list: [2,3,4,5]}
two   | {peer_id: 2, view_id: 5, leader: 2, memb_list: [2,3,4,5]}

three | {peer_id: 3, view_id: 6, leader: 2, memb_list: [2,3,4]}
four  | {peer_id: 4, view_id: 6, leader: 2, memb_list: [2,3,4]}
two   | {peer_id: 2, view_id: 6, leader: 2, memb_list: [2,3,4]}
```

```
   five   | {peer_id:5, view_id: 5, leader: 2, message:"peer 4 unreachable"}
   five   | {peer_id:5, view_id: 5, leader: 2, message:"peer 2 (leader)
unreachable"}
   five   | {peer_id:5, view_id: 5, leader: 3, message:"peer 3 (leader)
unreachable"}
   five   | {peer_id:5, view_id: 5, leader: 2, message:"peer 4 unreachable"}
   five   | {peer_id:5, view_id: 5, leader: 2, message:"peer 2 (leader)
unreachable"}
   five   | {peer_id:5, view_id: 5, leader: 3, message:"peer 3 (leader)
unreachable"}
```