CS7610: Lab 2 report

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

This program implements a **PassToken application** and integrates the Chandy-Lamport algorithm for capturing consistent snapshots in a distributed system.

Initialization

1. The program checks for command line arguments to obtain the path to the hosts file containing peer information.
2. It reads the configuration, initializes the **Process** struct with metadata, and establishes connections with other peers.
3. A **bi-directional TCP connection** is used for communication between peers on port 8080.

Payload

- Token: Encoded as an integer payload where the **first two bits represent the message type** (TOKEN).
- Marker: Encoded as an integer payload where the first two bits represent the message type (MARKER) and the **remaining 30 bits store the snapshot ID**. This allows concurrent snapshots by distinguishing between different marker messages using the snapshot ID.

Snapshot storage

1. Each process stores a map of Snapshot ID : Snapshot state (a struct that stores snapshot information)
2. Each Snapshot state instance has information about the state value and token at the time of initiation, and maps of closed channels and messages recorded on each channel.

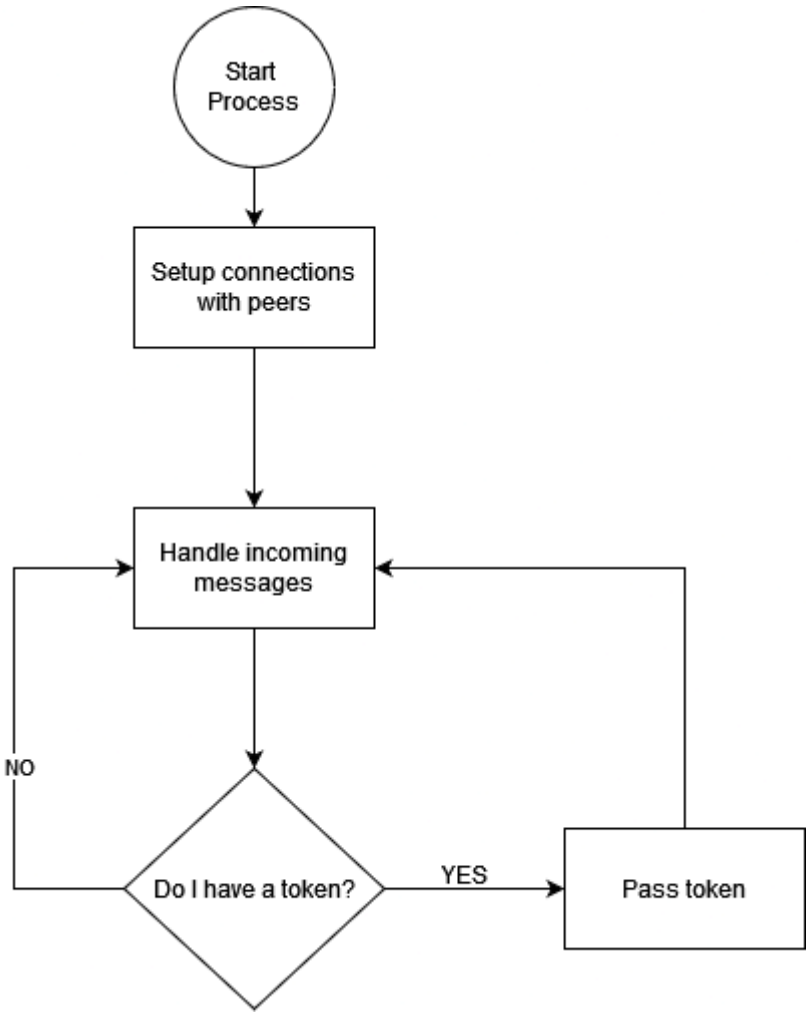
Message Handling

1. When a message is received in a process, its type is determined. A message of Token type updates the state of the process.
2. A message of Marker type initializes a snapshot or closes the channel according to the Chandi-Lamport algorithm.

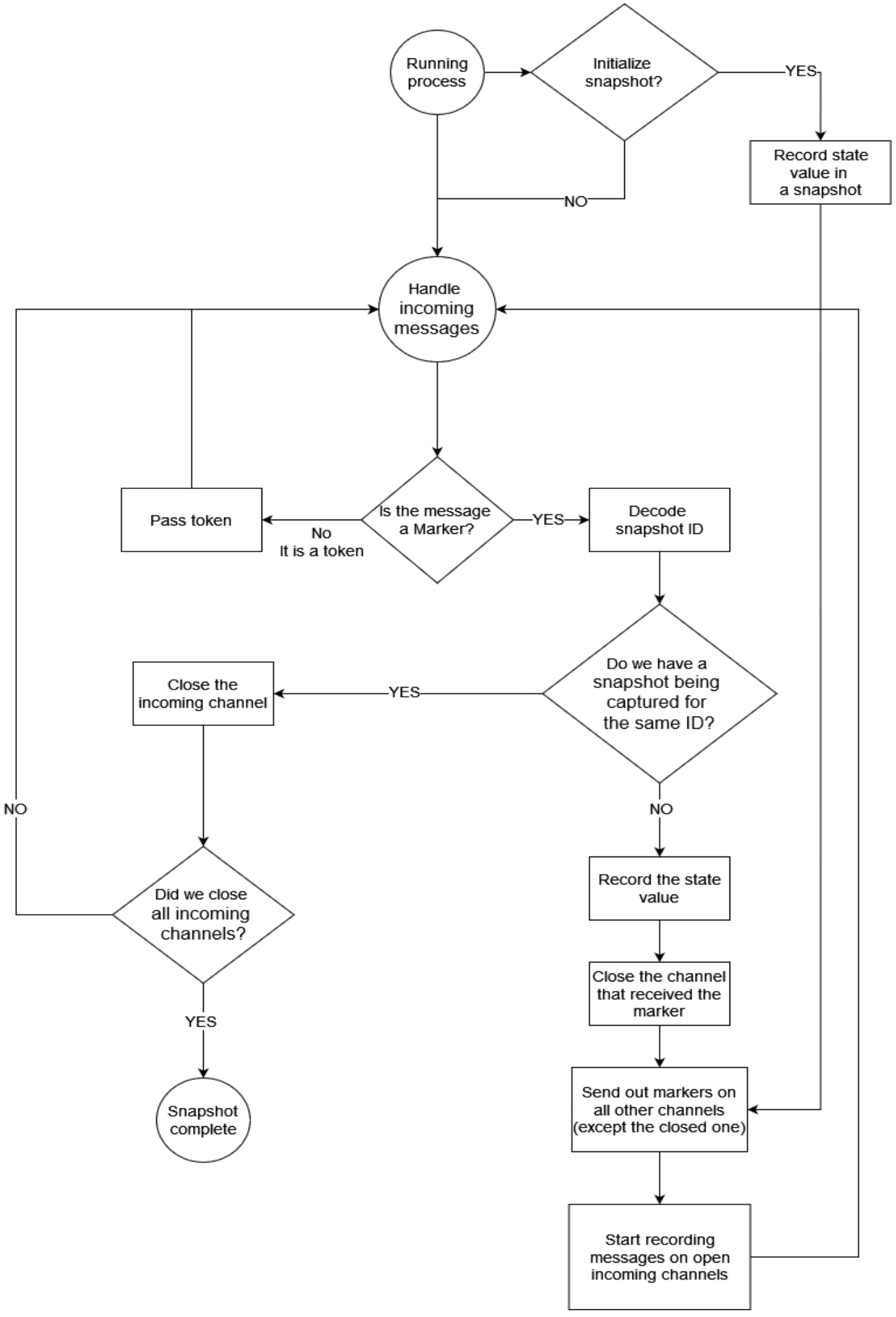
Key Considerations

1. **Goroutines:** The program uses multiple goroutines for handling tasks concurrently.
    a. Accepting a connection from another process
    b. Handling incoming messages
    c. Monitoring the process state and triggering the snapshot
    d. Sending markers to other processes
2. **Locks and Synchronization:**
    a. Read-Write mutex is used for managing process metadata. Write lock is used when updating the state value or passing the token. A read-only lock is used when reading state information.
    b. A generic mutex is used for managing snapshot data. Locks are used when marking channels as closed or adding a message to the queue.

State diagram: Token passing



State diagram: Snapshot mechanism



Testcase 1

```
docker compose -f .\docker-compose-testcase-1.yml up
Attaching to peer1, peer2, peer3, peer4, peer5
peer1  | {proc_id: 1, state: 0, predecessor: 5, successor: 2}
peer5  | {proc_id: 5, state: 0, predecessor: 4, successor: 1}
peer4  | {proc_id: 4, state: 0, predecessor: 3, successor: 5}
peer2  | {proc_id: 2, state: 0, predecessor: 1, successor: 3}
peer3  | {proc_id: 3, state: 0, predecessor: 2, successor: 4}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 1}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 1}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 1}
peer2  | {proc_id: 2, sender: 1, receiver: 2, message:"token"}
peer2  | {proc_id: 2, state: 1}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer2  | {proc_id: 2, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 1}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 2}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 2}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 2}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer2  | {proc_id: 2, sender: 1, receiver: 2, message:"token"}
peer2  | {proc_id: 2, state: 2}
```

Testcase 2

Output from Peer1

```
peer1  | {proc_id: 1, state: 0, predecessor: 5, successor: 2}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 1}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 2}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"started"}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 3}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 4}
peer1  | {proc_id:1, snapshot_id: 1, sender:1, receiver:4, msg:"marker", state:4,
has_token:NO}
peer1  | {proc_id:1, snapshot_id: 1, sender:1, receiver:5, msg:"marker", state:4,
has_token:NO}
peer1  | {proc_id:1, snapshot_id: 1, sender:1, receiver:2, msg:"marker", state:4,
has_token:NO}
peer1  | {proc_id:1, snapshot_id: 1, sender:1, receiver:3, msg:"marker", state:4,
has_token:NO}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 5}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 6}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"channel closed", channel:4-1, queue:[]}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"channel closed", channel:3-1, queue:[]}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"channel closed", channel:2-1, queue:[]}
peer1  | {proc_id: 1, sender: 1, receiver: 2, message:"token"}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"channel closed", channel:5-1,
queue:[TOKEN,TOKEN,TOKEN,TOKEN]}
peer1  | {proc_id:1, snapshot_id: 1, snapshot:"complete"}
peer1  | {proc_id: 1, sender: 5, receiver: 1, message:"token"}
peer1  | {proc_id: 1, state: 7}
```

Testcase 3

Output from Peer3

```
peer3  | {proc_id: 3, state: 0, predecessor: 2, successor: 4}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 1}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 2}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 3}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 4}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"started"}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"channel closed", channel:1-3, queue:[]}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 5}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
peer3  | {proc_id: 3, state: 6}
peer3  | {proc_id: 3, sender: 3, receiver: 4, message:"token"}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"channel closed", channel:4-3, queue:[]}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"channel closed", channel:2-3,
queue:[TOKEN,TOKEN]}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"channel closed", channel:5-3, queue:[]}
peer3  | {proc_id:3, snapshot_id: 1, snapshot:"complete"}
peer3  | {proc_id:3, snapshot_id: 1, sender:3, receiver:2, msg:"marker", state:6,
has_token:NO}
peer3  | {proc_id:3, snapshot_id: 1, sender:3, receiver:4, msg:"marker", state:6,
has_token:NO}
peer3  | {proc_id:3, snapshot_id: 1, sender:3, receiver:5, msg:"marker", state:6,
has_token:NO}
peer3  | {proc_id:3, snapshot_id: 1, sender:3, receiver:1, msg:"marker", state:6,
has_token:NO}
peer3  | {proc_id: 3, sender: 2, receiver: 3, message:"token"}
```

Testcase 4

Output from Peer5

```
.
.
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 4}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"started"}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"channel closed", channel:1-5,
queue:[]}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 5}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 6}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"channel closed", channel:4-5,
queue:[TOKEN,TOKEN]}
peer5  | {proc_id:5, snapshot_id: 1, sender:5, receiver:1, msg:"marker",
state:6, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 1, sender:5, receiver:2, msg:"marker",
state:6, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"channel closed", channel:2-5,
queue:[]}
peer5  | {proc_id:5, snapshot_id: 1, sender:5, receiver:3, msg:"marker",
state:6, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"channel closed", channel:3-5,
queue:[]}
peer5  | {proc_id:5, snapshot_id: 1, snapshot:"complete"}
peer5  | {proc_id:5, snapshot_id: 1, sender:5, receiver:4, msg:"marker",
state:6, has_token:NO}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 7}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 8}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 9}
.
.
.
peer5  | {proc_id: 5, state: 31}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"started"}
```

```
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"channel closed", channel:3-5,
queue:[]}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 32}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 33}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id:5, snapshot_id: 2, sender:5, receiver:2, msg:"marker",
state:33, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 2, sender:5, receiver:3, msg:"marker",
state:33, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 2, sender:5, receiver:4, msg:"marker",
state:33, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 2, sender:5, receiver:1, msg:"marker",
state:33, has_token:NO}
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"channel closed", channel:4-5,
queue:[TOKEN,TOKEN]}
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"channel closed", channel:2-5,
queue:[]}
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"channel closed", channel:1-5,
queue:[]}
peer5  | {proc_id:5, snapshot_id: 2, snapshot:"complete"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 34}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
peer5  | {proc_id: 5, sender: 4, receiver: 5, message:"token"}
peer5  | {proc_id: 5, state: 35}
peer5  | {proc_id: 5, sender: 5, receiver: 1, message:"token"}
```

Testcase 5

Output from Peer4

.
.
.

```
peer4  | {proc_id: 4, state: 3}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"started"}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"channel closed", channel:2-4,
queue:[]}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 4}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"started"}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"channel closed", channel:1-4,
queue:[]}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 5}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"channel closed", channel:3-4,
queue:[TOKEN,TOKEN]}
peer4  | {proc_id:4, snapshot_id: 2, sender:4, receiver:2, msg:"marker",
state:5, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 2, sender:4, receiver:3, msg:"marker",
state:5, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"channel closed", channel:1-4,
queue:[]}
peer4  | {proc_id:4, snapshot_id: 2, sender:4, receiver:5, msg:"marker",
state:5, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 2, sender:4, receiver:1, msg:"marker",
state:5, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"channel closed", channel:5-4,
queue:[]}
peer4  | {proc_id:4, snapshot_id: 2, snapshot:"complete"}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 6}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"channel closed", channel:2-4,
queue:[]}
peer4  | {proc_id:4, snapshot_id: 1, sender:4, receiver:1, msg:"marker",
state:6, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"channel closed", channel:3-4,
queue:[TOKEN,TOKEN]}
peer4  | {proc_id:4, snapshot_id: 1, sender:4, receiver:2, msg:"marker",
state:6, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 1, sender:4, receiver:3, msg:"marker",
state:6, has_token:NO}
```

```
peer4  | {proc_id:4, snapshot_id: 1, sender:4, receiver:5, msg:"marker",
state:6, has_token:NO}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"channel closed", channel:5-4,
queue:[]}
peer4  | {proc_id:4, snapshot_id: 1, snapshot:"complete"}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 7}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
peer4  | {proc_id: 4, sender: 3, receiver: 4, message:"token"}
peer4  | {proc_id: 4, state: 8}
peer4  | {proc_id: 4, sender: 4, receiver: 5, message:"token"}
```