Large Scale Systems
CS 410 / 510

Lecture 12:

PBFT Intricacies and View Change

Suyash Gupta

Assistant Professor

Distopia Labs and ONRG

Dept. of Computer Science

(E) suyash@uoregon.edu

(W) gupta-suyash.github.io

UNIVERSITY OF OREGON

Assignment 3 is Out!

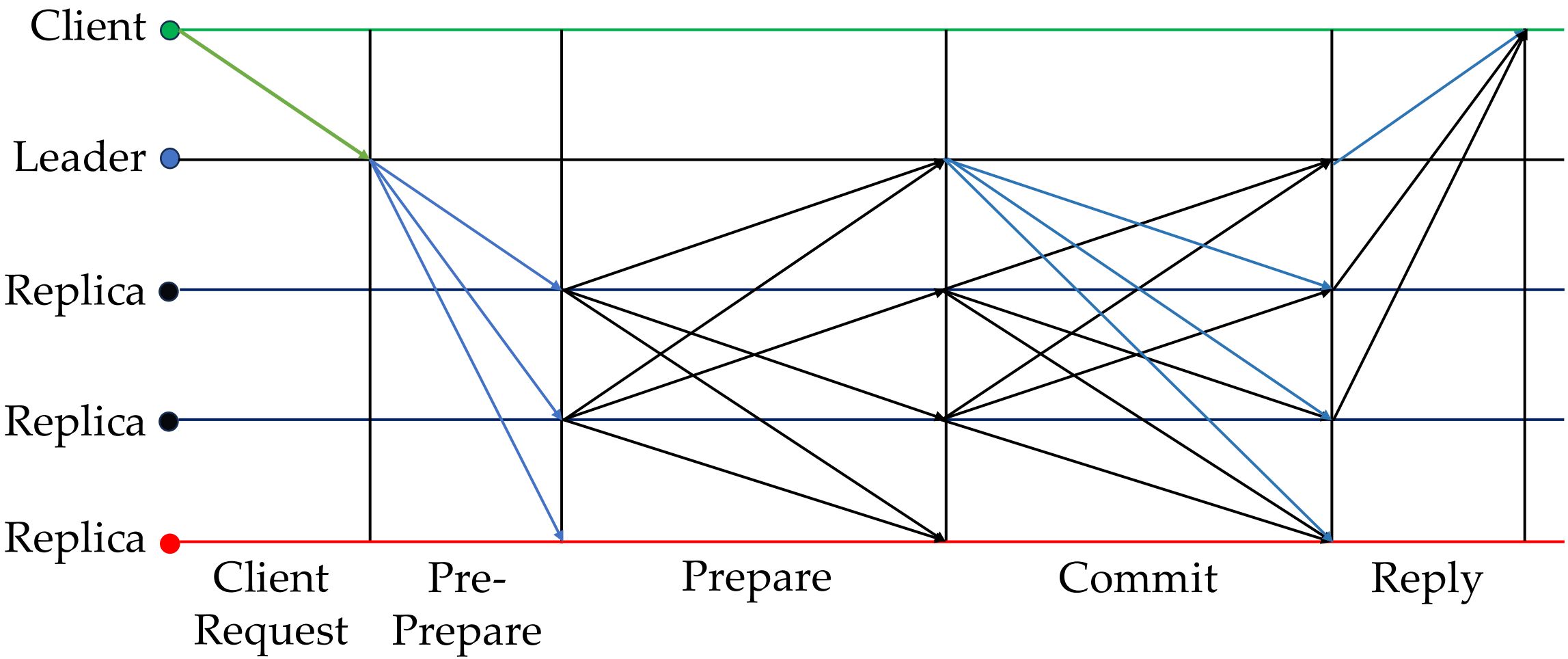
- Assignment 3 is due on **May 18, 2026** at **11:59pm PST**.
- Please start working with your groups.

Last Class

- Last class we looked at:

- Byzantine Failures
- Started looking at PBFT

PBFT Protocol



Prepare and Commit Phases

- What role do the Prepare and Commit Phases play?

- Prepare Phase:
  - Helps to establish **non-equivocation.**
  - Ensure that there is only one transaction agreed by a quorum of **2f+1** replicas.
  - Intuitively → Each replica checks if the proposal it received from the leader was also received by a quorum of replicas.

- Commit Phase:
  - Helps to guarantee **persistence**.
  - Ensures that even if the leader fails, an agreed transaction persists and is not lost!
  - Intuitively → Each replica checks that at least a quorum of good replicas also know that the same proposal was received a quorum of replicas.

Execution and Reply

- When can a replica execute the client transaction?
  - Post commit phase.

- How many replicas need to execute and send response to the client?
  - Ideally, all honest replicas should execute.

  - However, client needs at least **f+1** matching responses.
    - **Why?**
    - Because less than **f** responses are not sufficient to guarantee that at least **one** honest replica executed.

Sequence Numbers

Sequence Numbers

- Sequence numbers help in **identification** and **ordering**.

- Both clients and leaders assign sequence numbers.
  - They have access to a counter and use it to fetch/assign a sequence number.
  - Can also use clocks to fetch the current timestamp.

- Why are clients required to assign sequence number?
  - To prevent replay attacks.

- Why are replicas required to assign sequence number?
  - To order client requests.

Leader Identification

- Will PBFT always have the same leader?

Leader Identification

- Will PBFT always have the same leader?
  - No, the current leader can fail and needs to be changed.

- How many total leadership changes can occur in PBFT?

Leader Identification

- Will PBFT always have the same leader?
  - No, the current leader can fail and needs to be changed.
- How many total leadership changes can occur in PBFT?
  - Infinitely many.
- Then, how can we recognize who is the current leader and distinguish between the current leader's old and new reigns?

Leader Identification

- Will PBFT always have the same leader?
  - No, the current leader can fail and needs to be changed.
- How many total leadership changes can occur in PBFT?
  - Infinitely many.
- Then, how can we recognize who is the current leader and distinguish between the current leader's old and new reigns?
  - View numbers.

View Number

- Each replica is assigned a unique identifier (**ID**).
  - Generally, these identifiers are monotonically increasing.
  - In a system of **n** replicas, the replica identifiers are: **0,1,2,…,n**.

- Each view is also assigned a unique identifier – **view number**.
  - View numbers are also monotonically increasing, 0,1,2,….

View Number

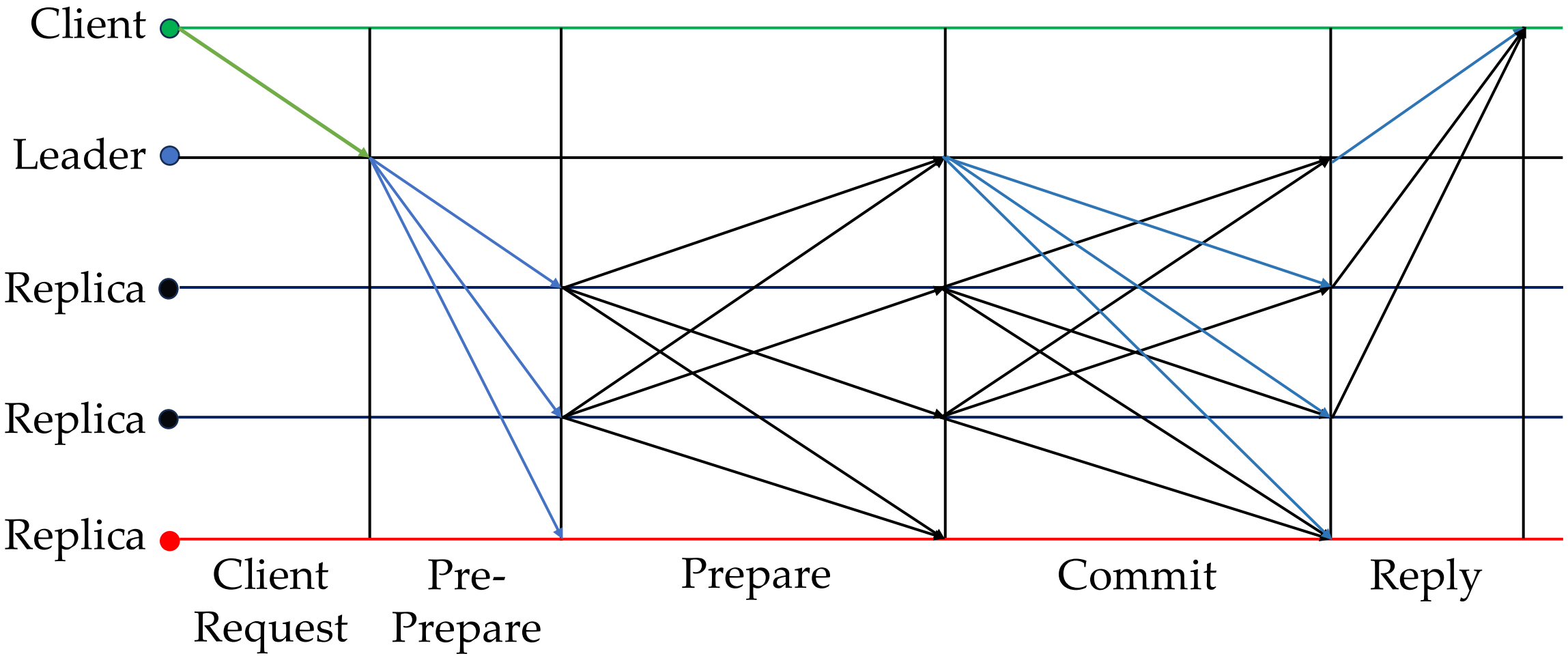
- Each replica is assigned a unique identifier (**ID**).
  - Generally, these identifiers are monotonically increasing.
  - In a system of **n** replicas, the replica identifiers are: **0,1,2,…,n**.

- Each view is also assigned a unique identifier – **view number**.
  - View numbers are also monotonically increasing, 0,1,2,….

- How do we know who is the leader for view **v**?
  - **Modulo Operation**
  - **v % n = ID**.
  - For example, if v = 15 and n = 7, then the leader is replica 1.

Message Identification

- During the Prepare and Commit phases, do the replicas need to identify which proposal are they voting for?



Message Identification

- During the Prepare and Commit phases, do the replicas need to identify which proposal are they voting for?
  - Yes, replicas do need to identify due to delays or failures.

- So, what should replicas send to track the proposal?

Message Identification

- During the Prepare and Commit phases, do the replicas need to identify which proposal are they voting for?
  - Yes, replicas do need to identify due to delays or failures.

- So, what should replicas send to track the proposal?
  - They should send with their vote, the full signed proposal by the leader.

- Is there anything better we can do?

Message Identification

- During the Prepare and Commit phases, do the replicas need to identify which proposal are they voting for?
  - Yes, replicas do need to identify due to delays or failures.

- So, what should replicas send to track the proposal?
  - They should send with their vote, the full signed proposal by the leader.

- Is there anything better we can do?
  - Leader should send signed proposal digest along with the proposal.
  - Digests are simply hash of the proposal.
  - Digests are significantly smaller in size in comparison to the proposal.

Hashing

- We represent a hash of message **x** as
  - **H = hash(x)**,
  - Where **hash()** is an existing hash function.

- What is the key property do we expect from a hash function?

Hashing

- We represent a hash of message **x** as
  - **H = hash(x)**,
  - Where **hash()** is an existing hash function.

- What is the key property do we expect from a hash function?
  - Collision resistance.
  - Given a message **x'**, the hash function **hash()** should ensure the following:
    - **H' = hash(x')** and **H ≠ H'**

- Several Collision Resistant Hash functions exist in the wild:
  - SHA 256, Double SHA.
  - Check out NIST recommendations.

Fairness

- How to guarantee fairness in the PBFT protocol?

Fairness

- How to guarantee fairness in the PBFT protocol?
- To guarantee fairness, an honest leader should order all the transactions in the order they were received.
- If Alice's transaction was received by the leader before Bob's transaction, then the leader should order Alice's transaction before Bob.

Garbage Collection

- What is meant by garbage collection in a BFT system?

Garbage Collection

- What is meant by garbage collection in a BFT system?

- As the system processes billions of transactions, we need to clear up the space to allow the system to run uninterrupted for long periods of time.
  - Clearing up the space implies removing old requests.
  - Ensuring all the honest replicas are at the same state.

Garbage Collection

- What is meant by garbage collection in a BFT system?

- As the system processes billions of transactions, we need to clear up the space to allow the system to run uninterrupted for long periods of time.
  - Clearing up the space implies removing old requests.
  - Ensuring all the honest replicas are at the same state.

- **Solution:**
  - Periodically checkpoint the system.
  - Clear up all the space before the last checkpoint.

Checkpointing in PBFT

Checkpointing in PBFT

- Simple Checkpoint Protocol:
  - Periodically, all replicas initiate the *checkpoint phase*.

Checkpointing in PBFT

- Simple Checkpoint Protocol:
  - Periodically, all replicas initiate the *checkpoint phase*.
  - Create a digest of the current state.

Checkpointing in PBFT

- Simple Checkpoint Protocol:
  - Periodically, all replicas initiate the *checkpoint phase*.
  - Create a digest of the current state.
  - Create a Checkpoint message that includes the digest and the highest sequence number till now and send this message to all the replicas.

Checkpointing in PBFT

- Simple Checkpoint Protocol:
  - Periodically, all replicas initiate the *checkpoint phase*.
  - Create a digest of the current state.
  - Create a Checkpoint message that includes the digest and the highest sequence number till now and send this message to all the replicas.
  - Once a replica receives matching Checkpoint message from **n-f** replicas, it marks the checkpoint phase as complete

- By the word “periodically”, we mean to run checkpoint protocol after
  - Every **x** requests have committed, or
  - Some **t** time has passed.

Failures in PBFT

Impact of Non-leader Byzantine Replica

Impact of Non-leader Byzantine Replica

- A non-leader Byzantine replica can drop, delay, or ignore a message.

- However, if the network is reliable, then a non-leader Byzantine replica **cannot affect** safety and liveness.
  - The system will make progress and remain safe till there is a good leader.

Impact of Byzantine Leader

Impact of Non-leader Byzantine Replica

- A Byzantine replica can do everything that a non-leader can and more:
  - Equivocate
  - Suppress clients.

- If the leader is malicious, we need to take steps:
  - Detect the malicious behavior.
  - Replace

Impact of Non-leader Byzantine Replica

- A Byzantine replica can do everything that a non-leader can and more:
  - Equivocate
  - Suppress clients.

- If the leader is malicious, we need to take steps:
  - Detect the malicious behavior.
  - Replace

- Leader Replacement → View Change

- *Note:* A true leader detection and replacement only takes place under the assumption that the network is reliable.

Detecting Malicious Leader

- Who can detect that a leader is acting maliciously?

Detecting Malicious Leader

- Who can detect that a leader is acting maliciously?
  - Replicas
  - Clients

Client Detection

- How can a client detect that the leader is malicious?

Client Detection

- How can a client detect that the leader is malicious?
  - When the client sends its transaction to the leader, it **starts a timer**.
  - If the **timer expires** and the client has not received f+1 matching response for its request, it **assumes** that the leader is malicious.
  - Next, it **complains** to all the replicas that it has not received sufficient matching responses.
  - Client **hopes** that good replicas will ensure that it receives responses.

Non-Leader Replica Detection

- How can a non-leader replica detect that the leader is malicious?

Non-Leader Replica Detection

- How can a non-leader replica detect that the leader is malicious?
  - Receive a complaint from the client.
  - Track leader by itself.

Non-Leader Replica Detection

- How can a non-leader replica detect that the leader is malicious?
  - Receive a complaint from the client.
  - Track leader by itself.

- **Complaint from client:**
  - On receiving a complain, a replica starts a timer and forwards the client complaint to the leader.

  - If the timer expires and the leader has not proposed any transaction from the complaining client, the replica assumes that the leader is malicious.

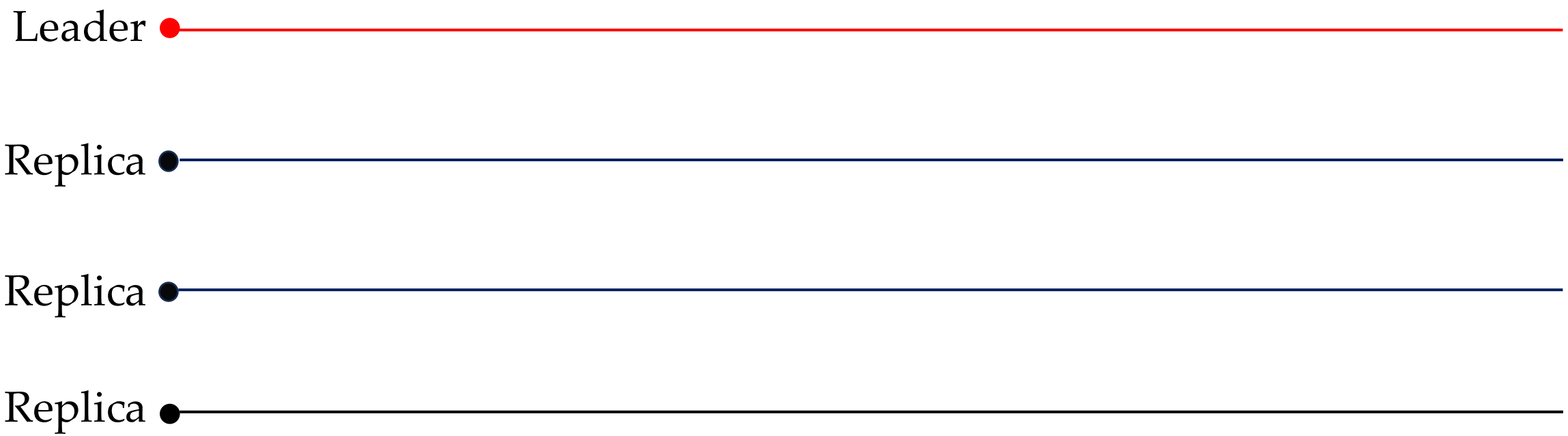
Non-Leader Replica Detection

- How can a non-leader replica detect that the leader is malicious?
  - Receive a complaint from the client.
  - Track leader by itself.

- **Complaint from client:**
  - On receiving a complain, a replica starts a timer and forwards the client complaint to the leader.
  - If the timer expires and the leader has not proposed any transaction from the complaining client, the replica assumes that the leader is malicious.

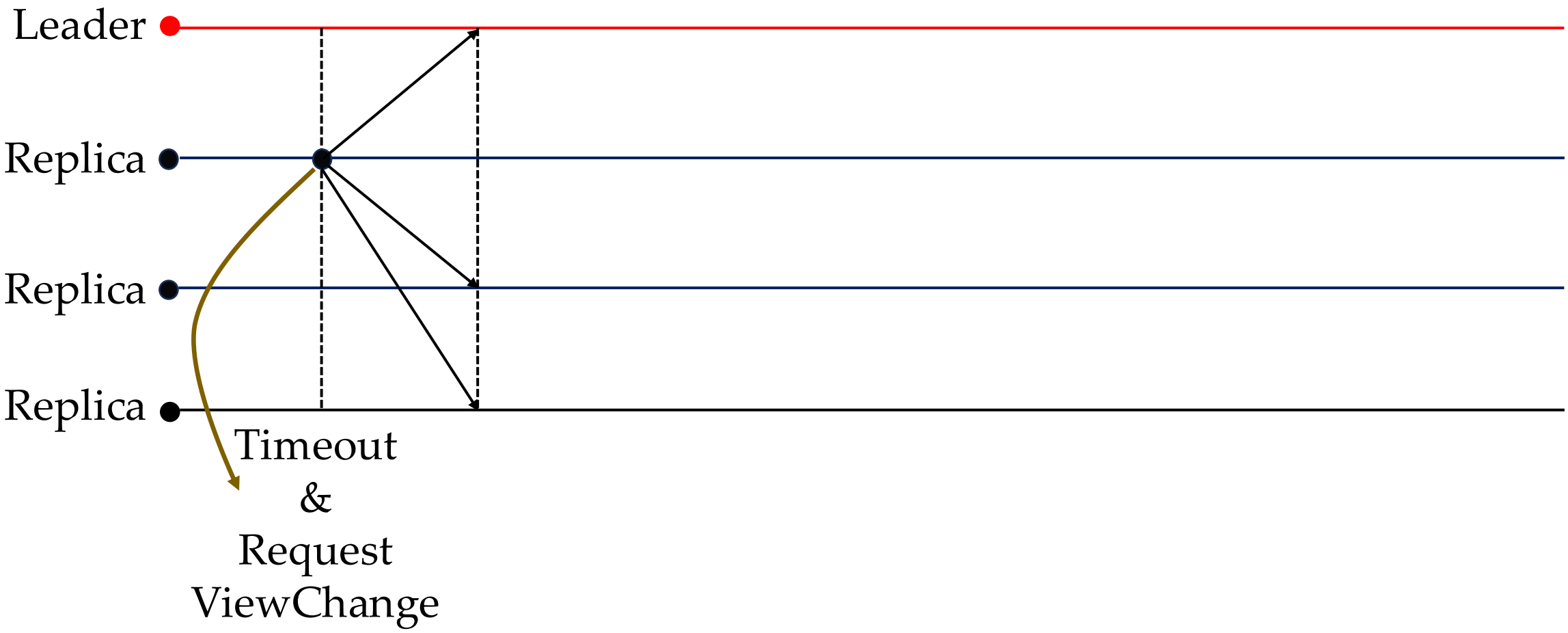
- **Track by itself:**
  - When a replica receives a Pre-prepare message, it starts a timer.
  - If the timer expires and the replica did not receive n-f Commit messages for the Pre-prepare message, it assumes that the leader is malicious.

View Change Protocol Flow

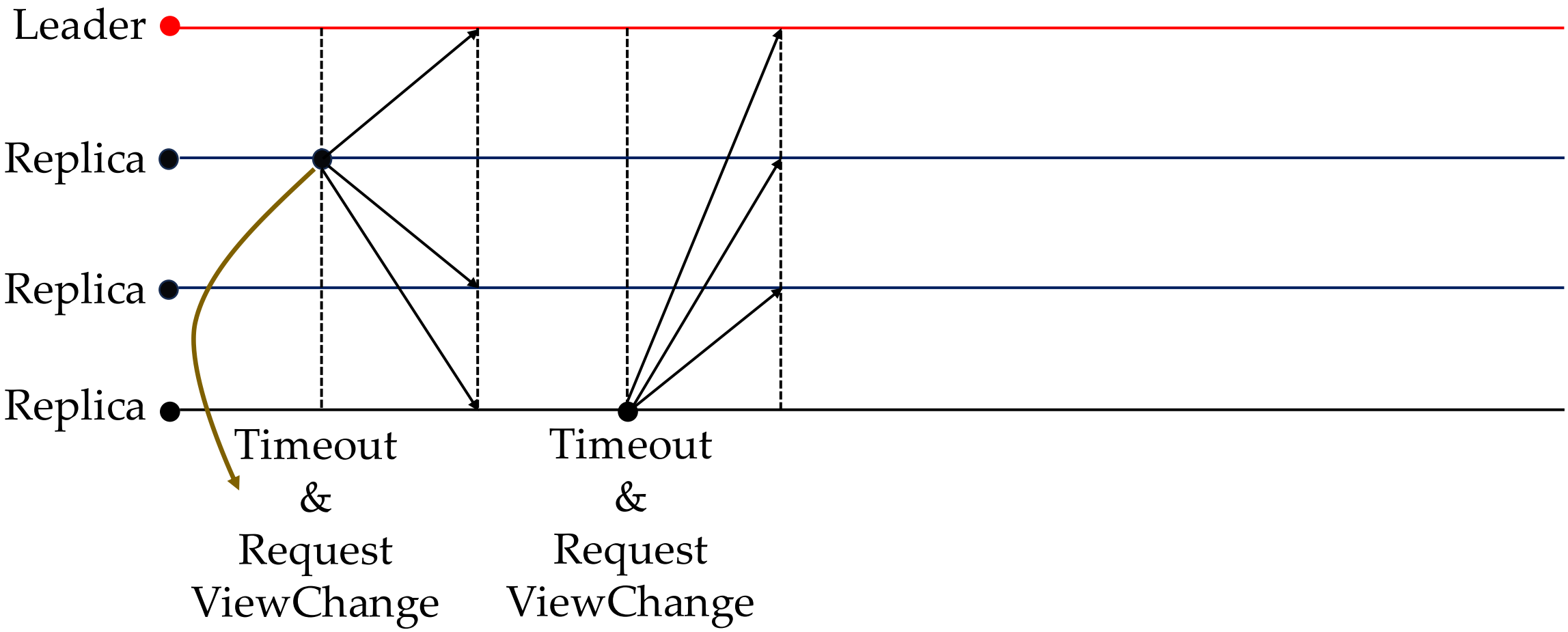


View Change Protocol Flow

When a replica timeouts, it sends a ViewChange message.

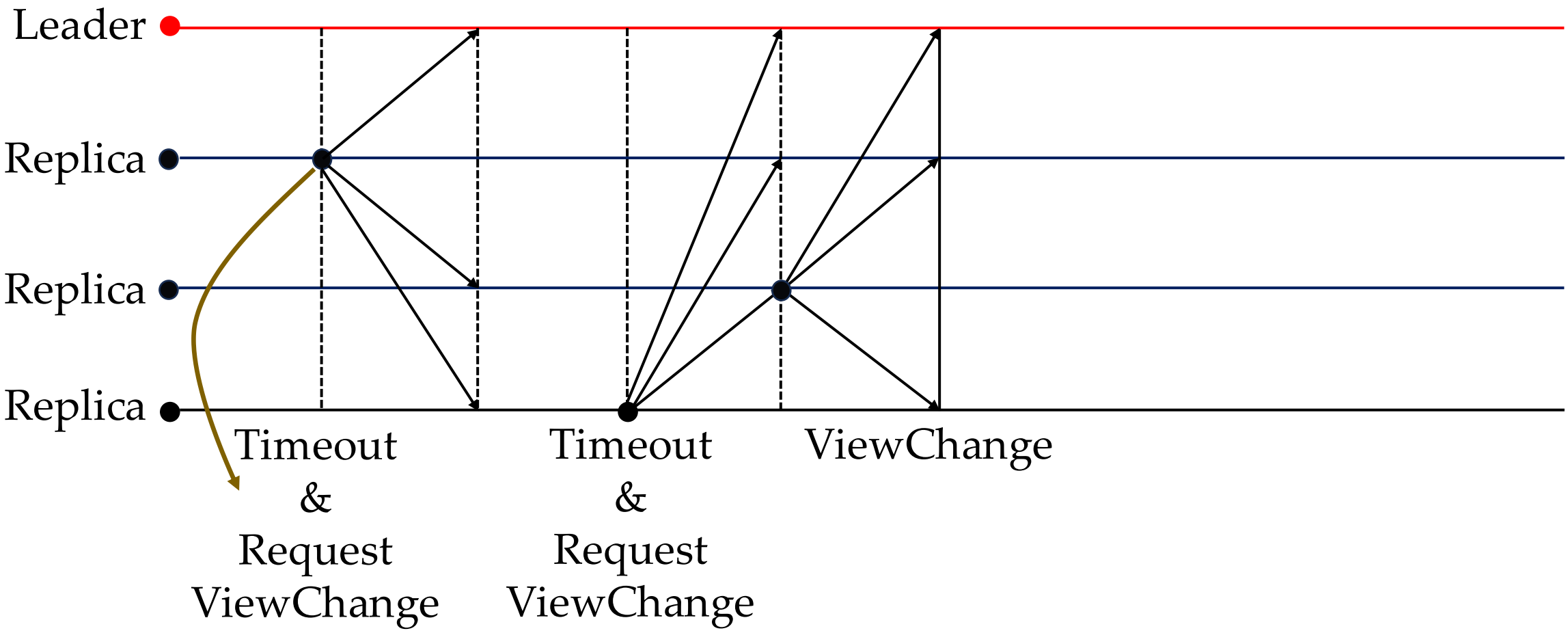


View Change Protocol Flow



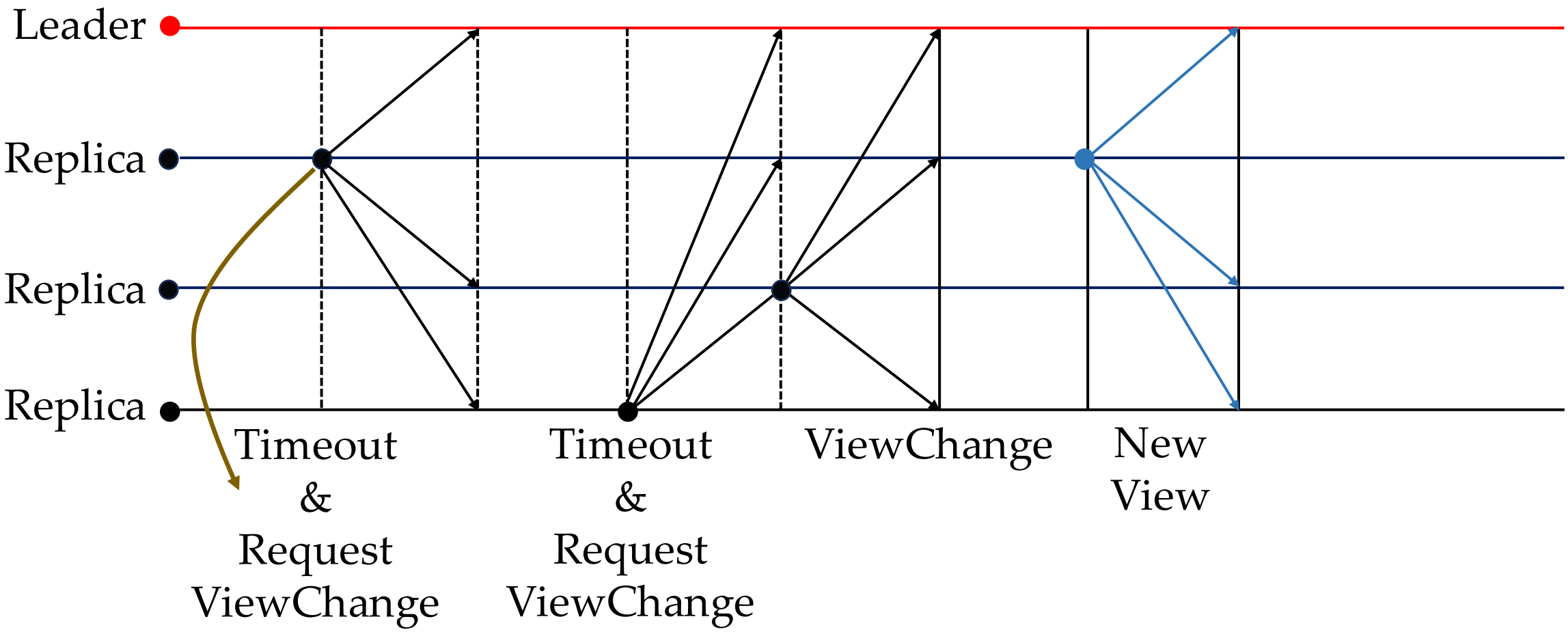
View Change Protocol Flow

When a replica receives ViewChange messages from **f+1** replicas, it also joins mutiny.



View Change Protocol Flow

When the next leader receives **n-f** ViewChange messages, it starts the NewView.



View Change Message

- What information should a replica include in its View Change message?

View Change Message

- What information should a replica include in its View Change message?

- The View Change message should help the next leader to form a **common state**.

- It includes:
  - The last agreed checkpoint.
  - For every sequence number since the last checkpoint:
    - **n-f** signed Prepare messages.
    - Pre-prepare messages for those sequence numbers for which **n-f** Prepare messages do not exist.

New View Messages

- What information should a replica include in its View Change message?
- The NewView message tries to bring all the replicas to the common state.
- It includes:
  - **n-f** ViewChange messages.
  - The last agreed checkpoint.
  - The list of agreed up on sequence numbers and proposal digests since the last checkpoint.

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.
- What happens if the next leader is also Byzantine?

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.
- What happens if the next leader is also Byzantine?
  - Replica will again timeout and start the next ViewChange.

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.
- What happens if the next leader is also Byzantine?
  - Replica will again timeout and start the next ViewChange.
- Say replicas are out of sync and some are timing out too fast. What can we do?

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.
- What happens if the next leader is also Byzantine?
  - Replica will again timeout and start the next ViewChange.
- Say replicas are out of sync and some are timing out too fast. What can we do?
  - Post timeout, replicas do an exponential backoff to increase their timer value.
- How many consecutive view changes are possible under a reliable n/w?

View Change Phase

- What if a replica receives a proposal from the old leader after it has timeout?
  - Post timeout, a replica ignores all proposals from the old leader.
  - It sends the ViewChange message to the next leader, moves to the next view, and restarts the timer.
- What happens if the next leader is also Byzantine?
  - Replica will again timeout and start the next ViewChange.
- Say replicas are out of sync and some are timing out too fast. What can we do?
  - Post timeout, replicas do an exponential backoff to increase their timer value.
- How many consecutive view changes are possible under a reliable n/w?
  - At most **f**.

New View

- What should replicas do after receiving a New View Message?

New View

- What should replicas do after receiving a New View Message?
  - Ensure they have the common state.

New View

- What should replicas do after receiving a New View Message?
  - Ensure they have the common state.

- What should the new leader do after sending the New View message?

New View

- What should replicas do after receiving a New View Message?
  - Ensure they have the common state.

- What should the new leader do after sending the New View message?
  - Re-run consensus on all the proposals since the last checkpoint.
  - If replicas have already executed some proposal, they do not need to re-execute if there is no state change.
  - Finally, start proposing new proposals.