Large Scale Systems
CS 410 / 510

Lecture 18:
Streamlined Consensus

Suyash Gupta

Assistant Professor

Distopia Labs and ORNG

Dept. of Computer Science

(E) suyash@uoregon.edu

(W) gupta-suyash.github.io

UNIVERSITY OF OREGON

Assignment 4 is Out!

- Assignment 4 is due on **June 2, 2025** at **11:59pm PST**.
- Please start working with your groups.

Presentations

- Each group will present their MiniSpanner on **June 4, 2025 in class**.
- The class will start **10min earlier** on **8:20am** to accommodate all the groups.
- Each group will get **25-30min** to present their progress.

Reading Material

- Online reading

Last Class

- Last class we looked at:

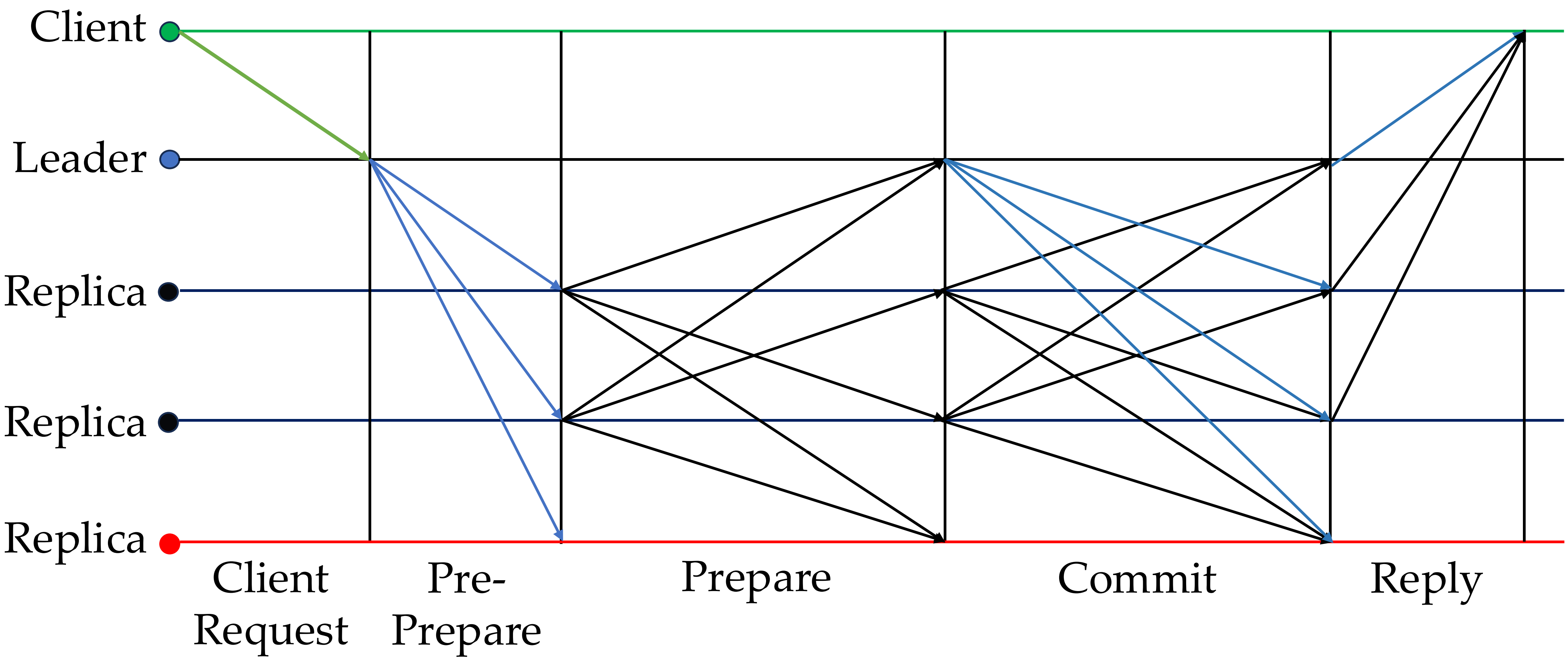
- Scaling Consensus across Globe
- GeoBFT
- RingBFT

Optimizing PBFT

- Today, our goal is to continue optimizing PBFT.

- Until now, we added to PBFT:
    - Pipelining
    - Out-of-Order Message Processing
    - Speculation
    - Parallelism
    - Geo-scaling
    - Sharding

PBFT Protocol



What can we do next?

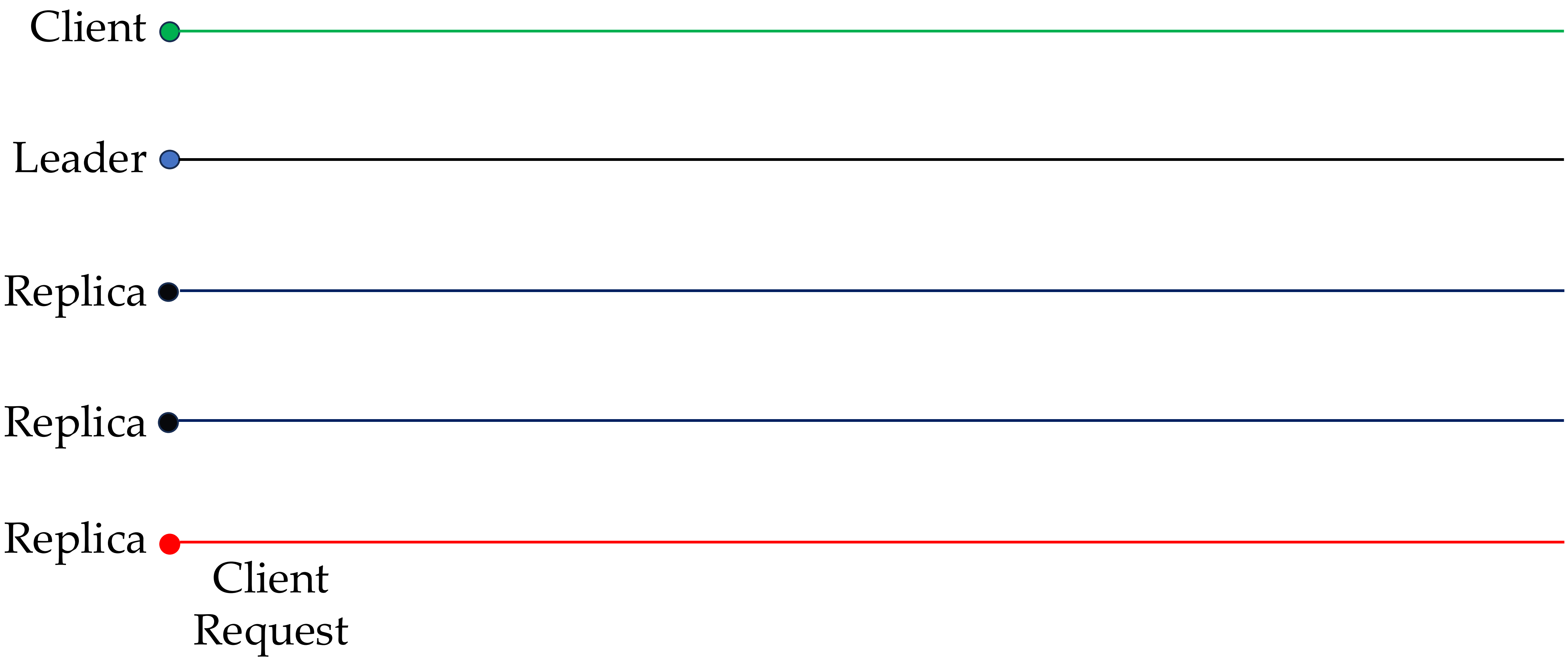
What can we do next?

- We can **linearize** the phases of PBFT.

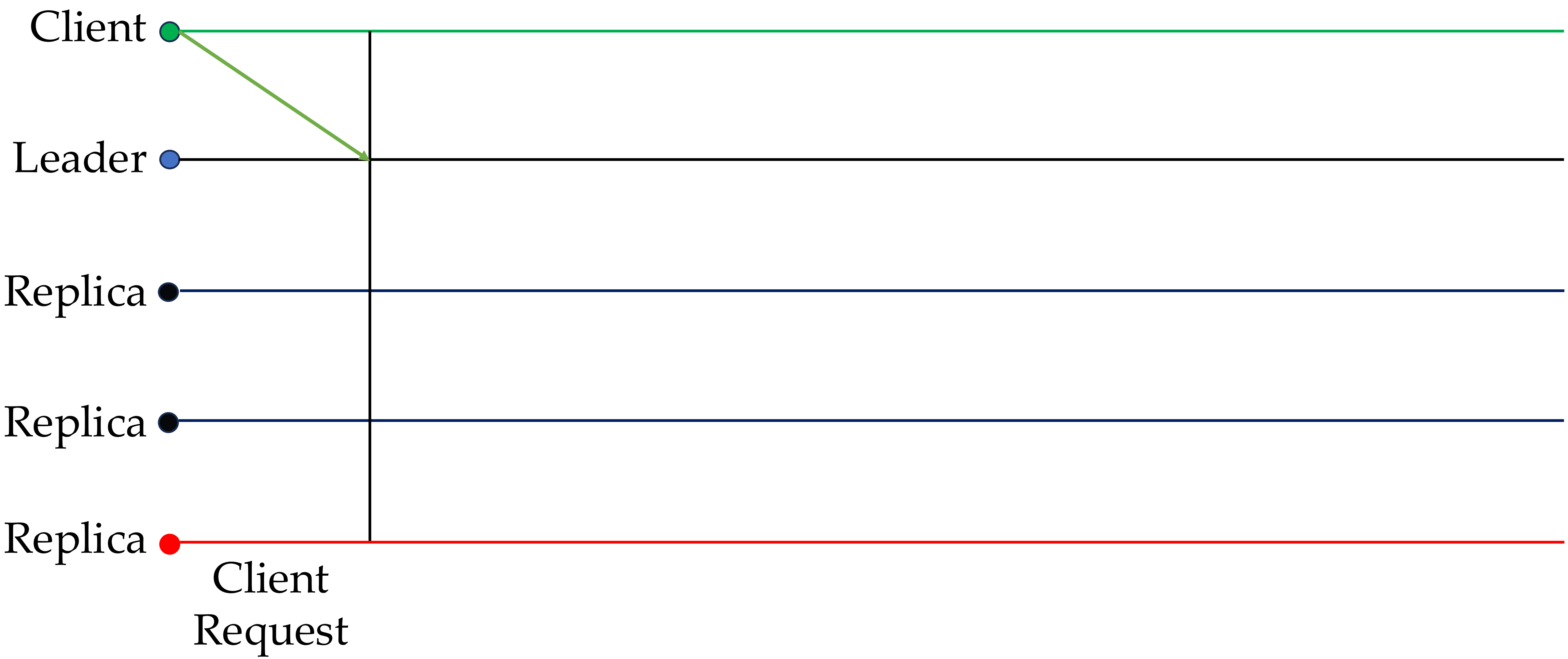
- Split each phase with quadratic communication complexity into 2 phases of linear complexity.

- Instead of All-to-All → All-to-One + One-to-All.

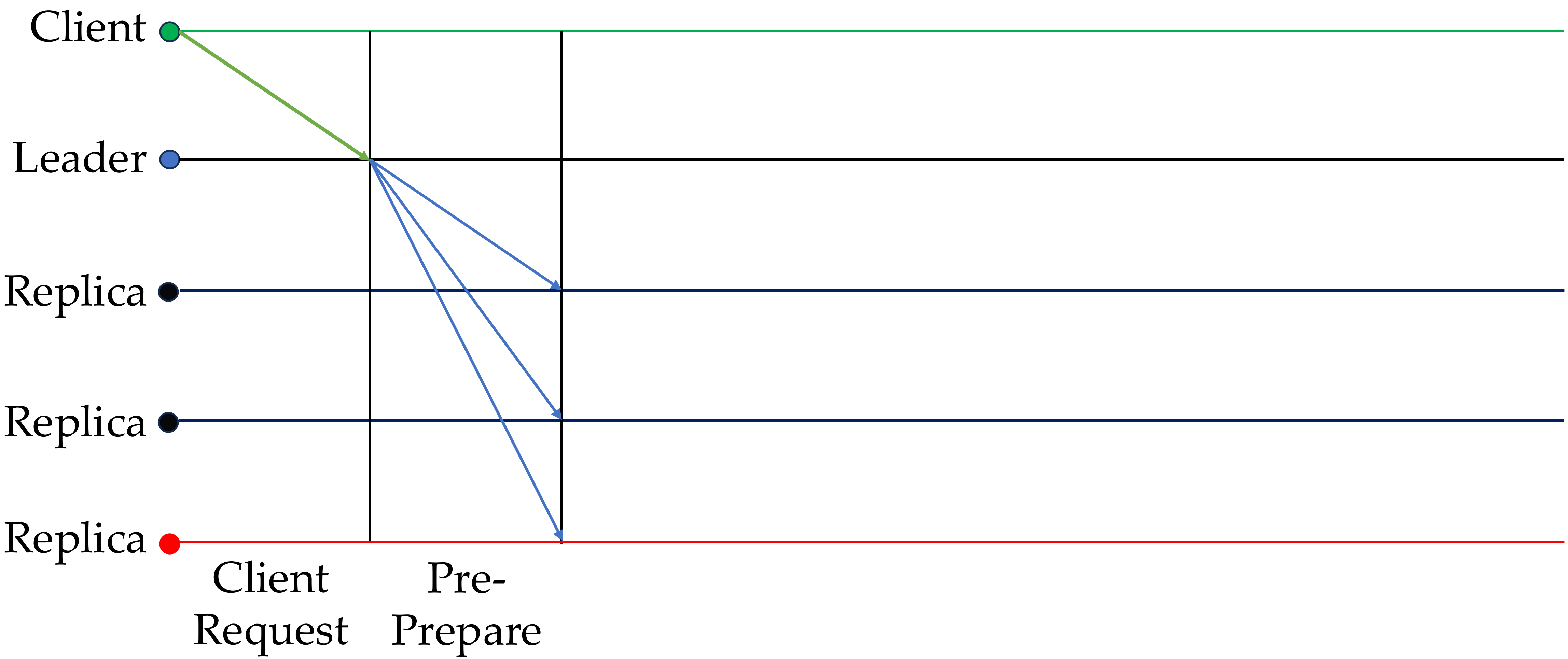
PBFT Linearized



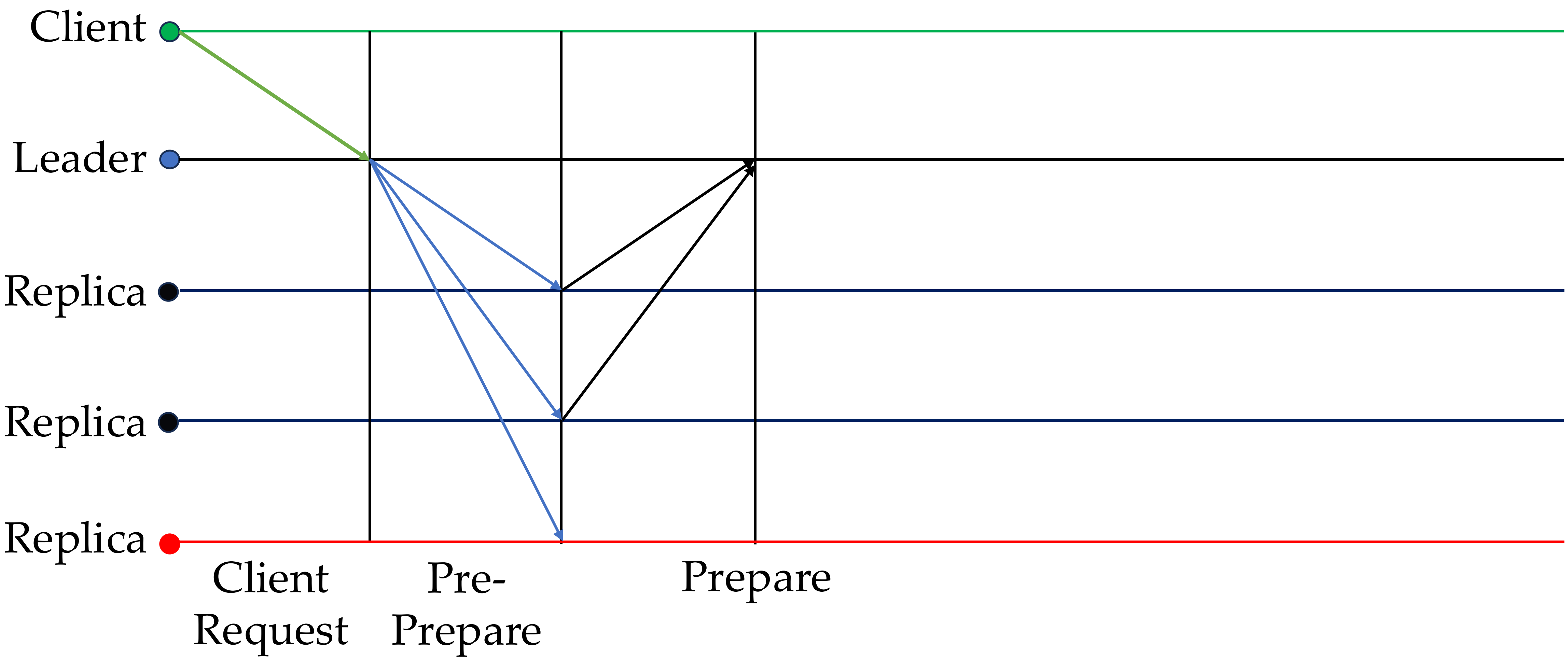
PBFT Linearized



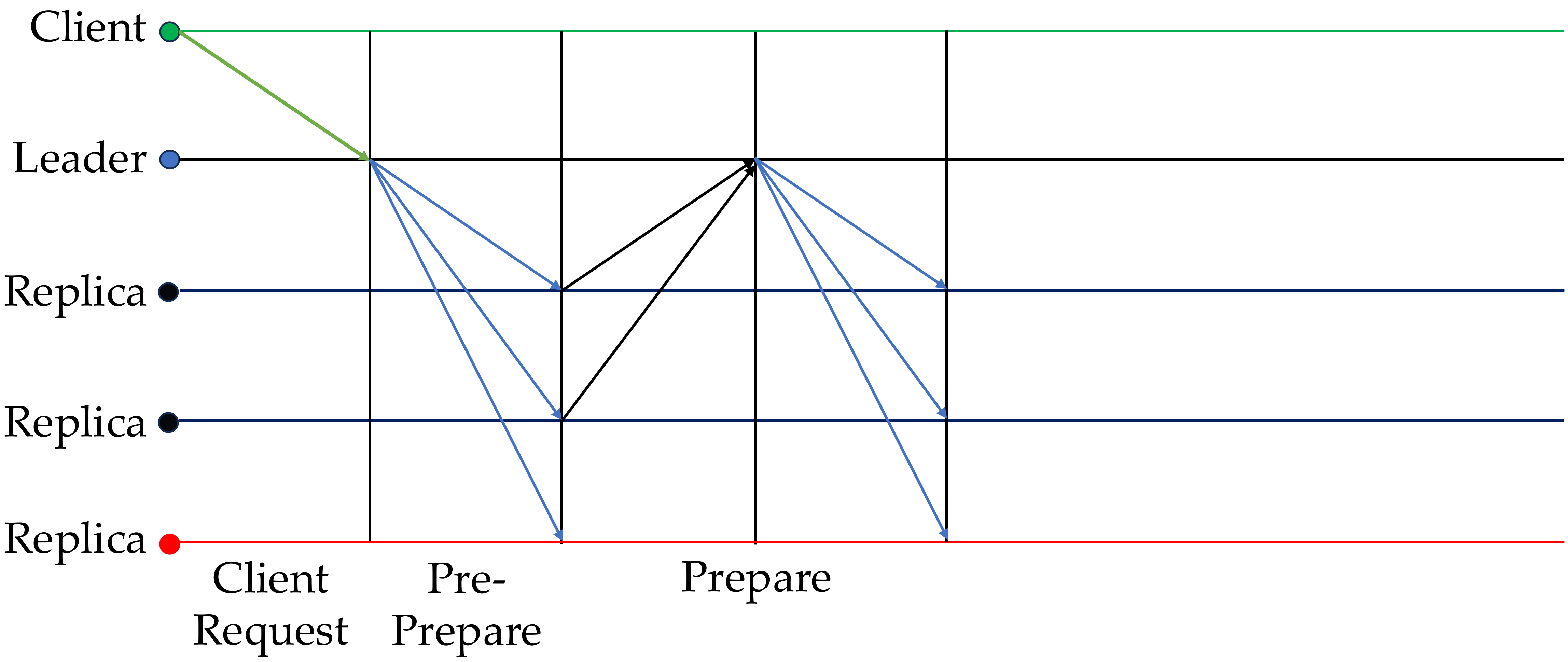
PBFT Linearized



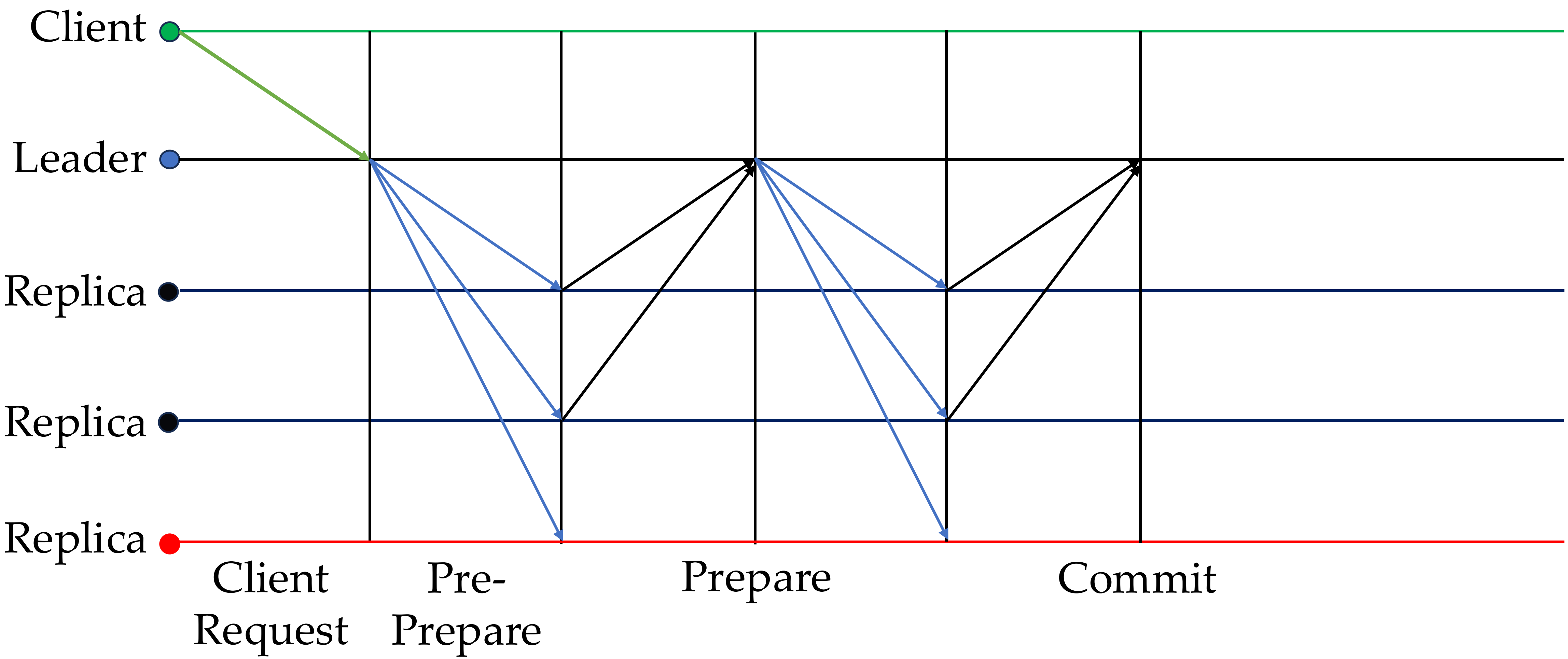
PBFT Linearized



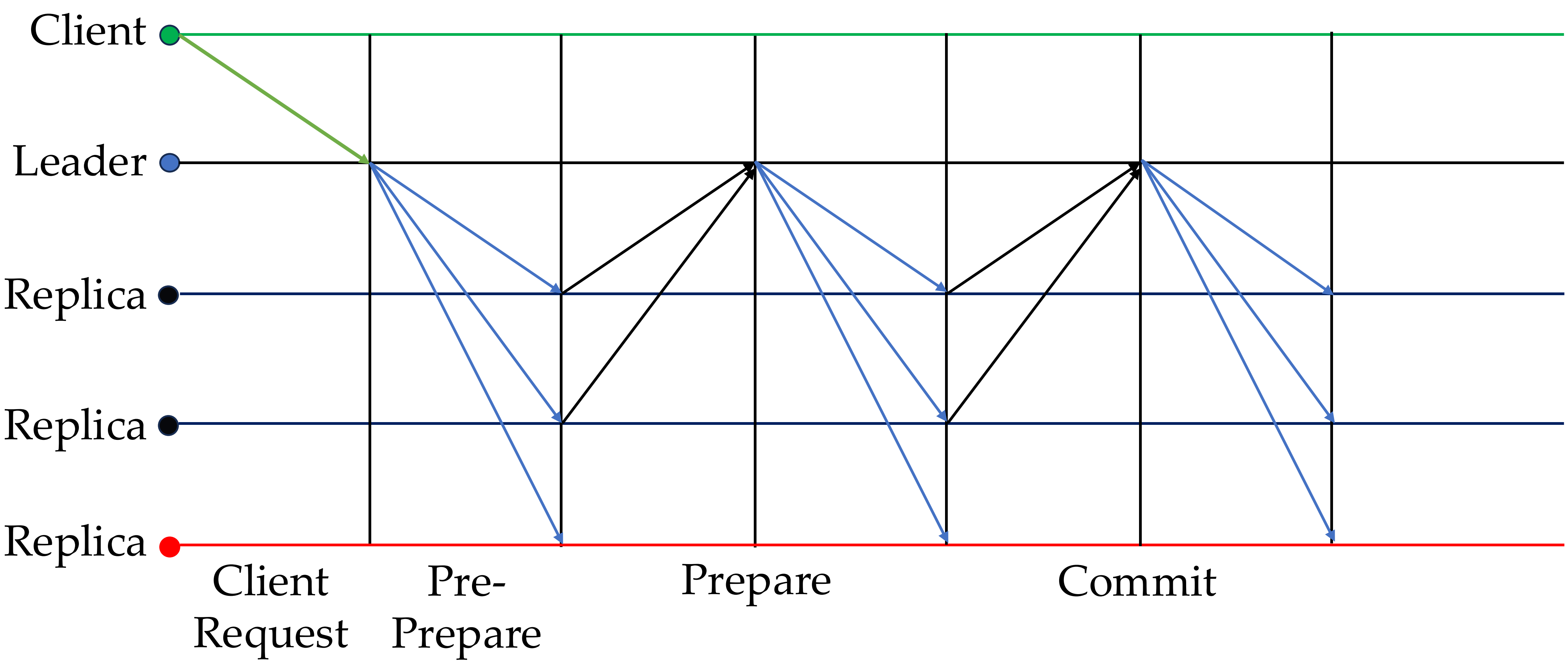
PBFT Linearized



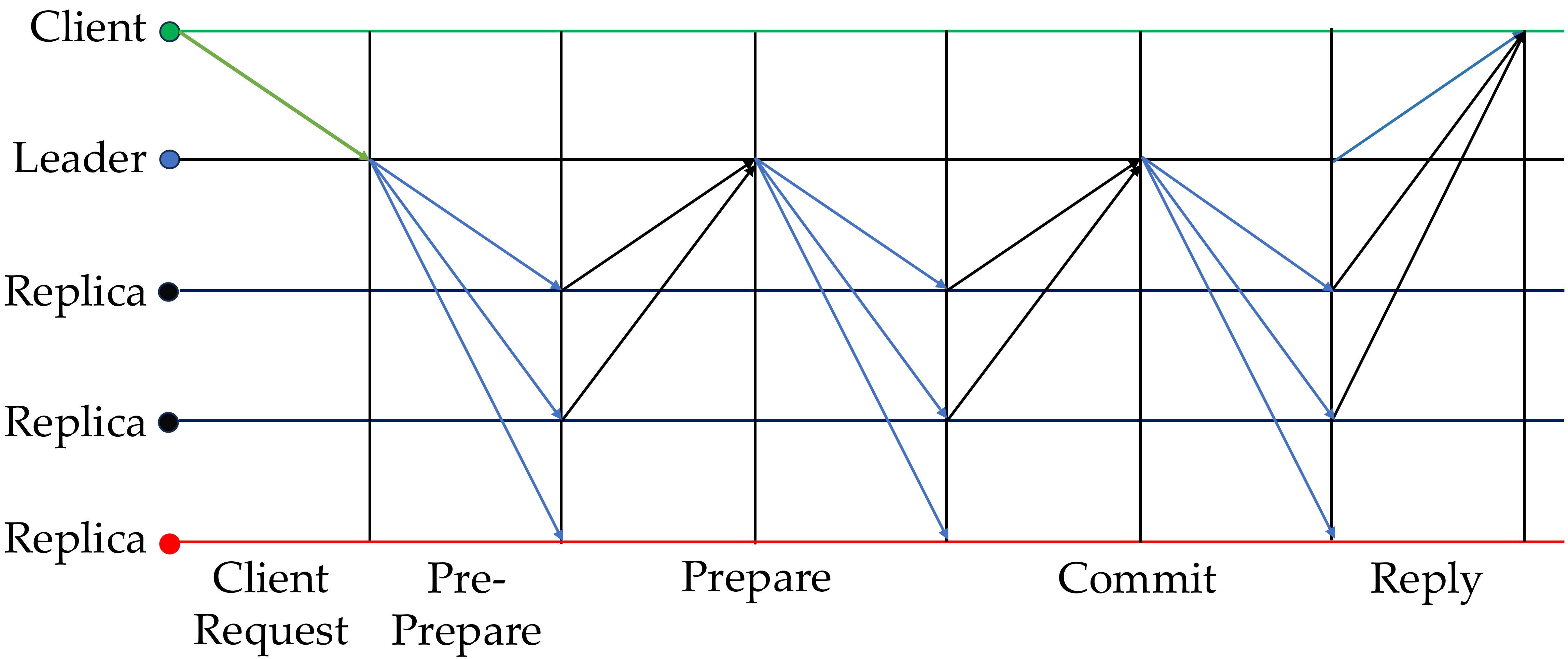
PBFT Linearized



PBFT Linearized



PBFT Linearized



Challenges for Linearized PBFT?

- Even though we linearized PBFT, did we actually save any communication?

Challenges for Linearized PBFT?

- Even though we linearized PBFT, did we actually save any communication?

- The leader needs to still send each replica a certificate!
    - A certificate comprises of n-f Prepare or Commit messages.

- The communication is still quadratic → n messages and each of size O(n).

Challenges for Linearized PBFT?

- Even though we linearized PBFT, did we actually save any communication?

- The leader needs to still send each replica a certificate!
  - A certificate comprises of n-f Prepare or Commit messages.

- The communication is still quadratic → n messages and each of size O(n).

- Can we do something better?

Threshold Signature-Scheme

Threshold Signature-Scheme

- Each replica creates a threshold-share
    - Signs the digest of the message using a special key.

Threshold Signature-Scheme

- Each replica creates a threshold-share
  - Signs the digest of the message using a special key.

- Next, each replica sends this share along with its message to the leader.

Threshold Signature-Scheme

- Each replica creates a threshold-share
  - Signs the digest of the message using a special key.

- Next, each replica sends this share along with its message to the leader.

- The leader combines n-f shares to create a threshold signature.
  - The size of the signature is O(1).
  - Anyone can verify to check if the share was created with the help of n-f replicas.

Threshold Signature-Scheme

- Each replica creates a threshold-share
    - Signs the digest of the message using a special key.

- Next, each replica sends this share along with its message to the leader.

- The leader combines n-f shares to create a threshold signature.
    - The size of the signature is O(1).
    - Anyone can verify to check if the share was created with the help of n-f replicas.

- This whole idea is enabled using threshold cryptographic schemes that allow creating keys that support aggregation.

Linearized PBFT with Threshold Cryptography

- With threshold cryptography, linearization of PBFT is complete.
- Now, we have reduced communication cost from quadratic to linear.
- However, the design is now more computationally expensive as threshold cryptography is expensive.
- This idea was introduced in a protocol → **HotStuff [PODC'19].**
- The **n-f** threshold signature is often referred to as **QC** → **Quorum Certificate**.

Challenges for Linearized PBFT?

- Any other remaining challenges?

Challenges for Linearized PBFT?

- Any other remaining challenges?
    - Yes!

- HotStuff also discussed about the challenges of having one leader:
    - Censorship of Clients.
    - Leader Slowness.
    - Expensive view change.

Client Censorship

- A single fixed leader can censor clients → silently drop their requests.

- May avoid proposing requests of specific clients unless forced.

Leader Slowness

- A clever Byzantine leader may propose client requests slowly.
- If the leader knows that it will not be replaced if it proposes a minimum number of requests within a specific time period, then it will not propose any extra requests.

View Change Costs

- We know that view change is expensive.

- During view change, no consensus can take place.

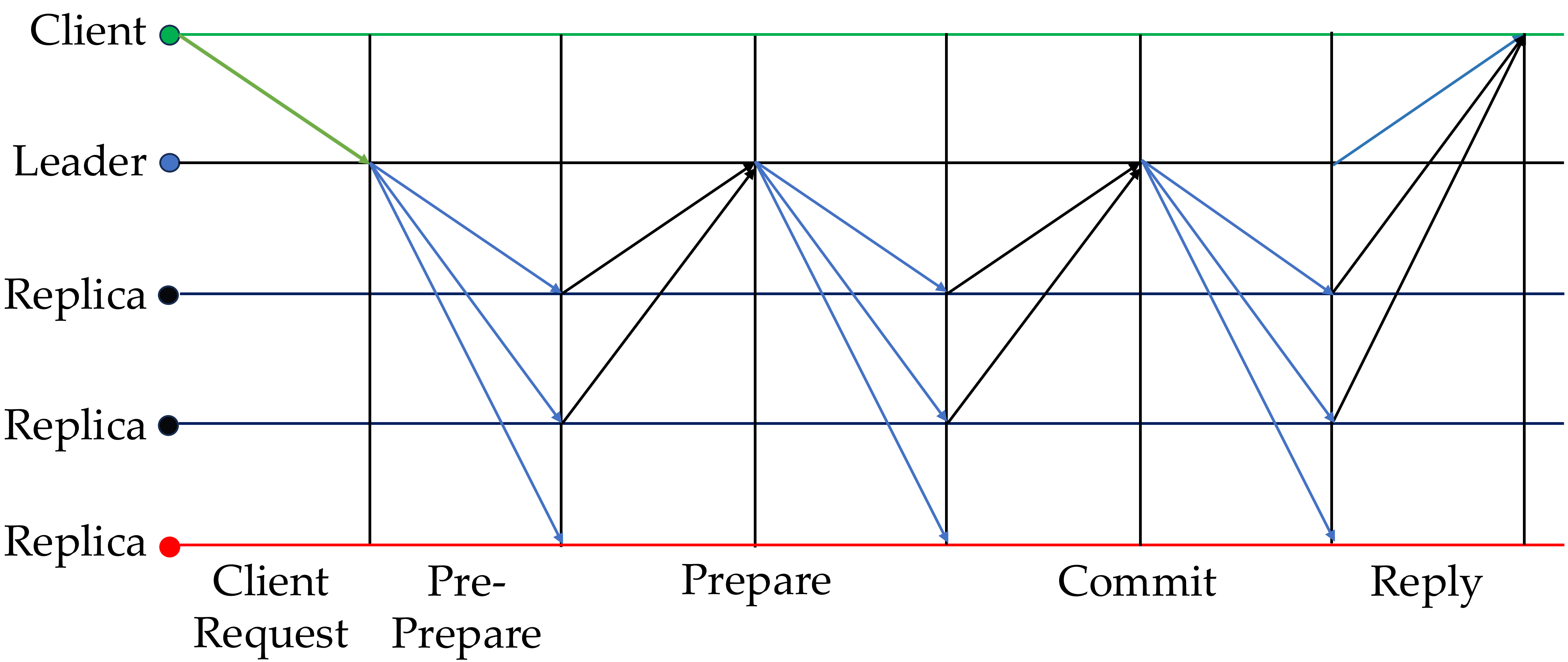
HotStuff’s Solution

HotStuff’s Solution

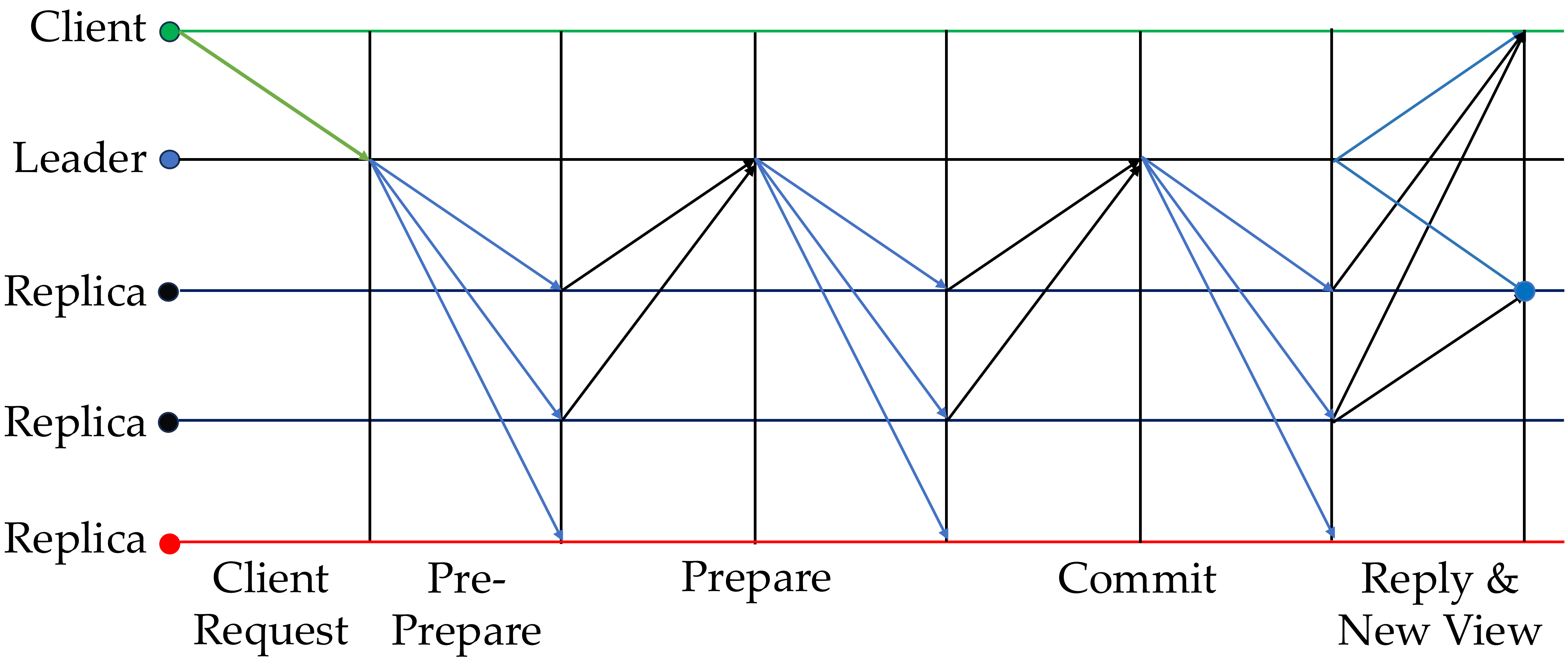
- Switch leader after every consensus!

- Each leader lasts for only one round of consensus.

HotStuff-2 (a faster variant of HotStuff)



HotStuff-2 (a faster variant of HotStuff)



Challenges with HotStuff-2

Challenges with HotStuff-2

- Changing leader at the end of each consensus enforces that each leader can propose only one batch.

- Propose one batch → Switch.

- No possibility for applying out-of-order message processing!

- Extreme drop in throughput.

How can we increase HotStuff-2's throughput?

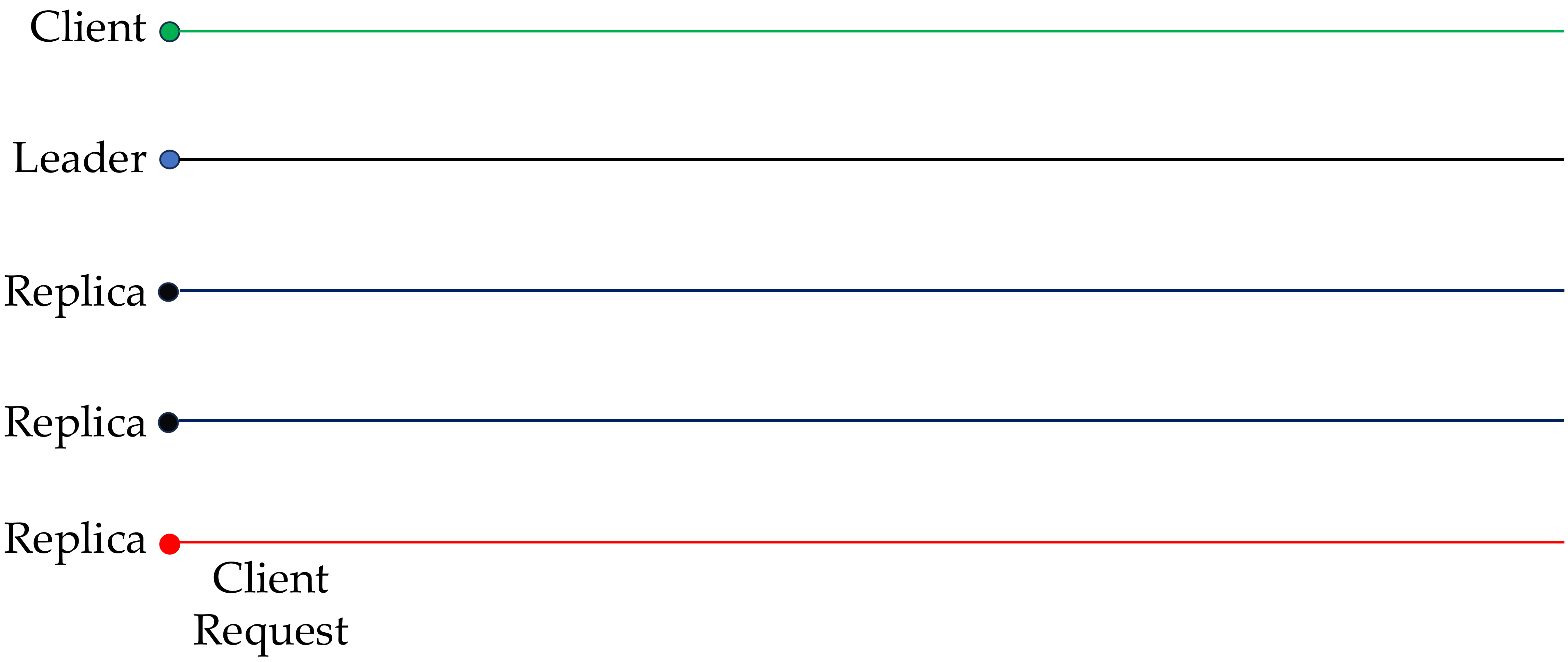
Streamlined HotStuff-2

- Allow switching leader in every alternate phase.
    - Essentially pipelining

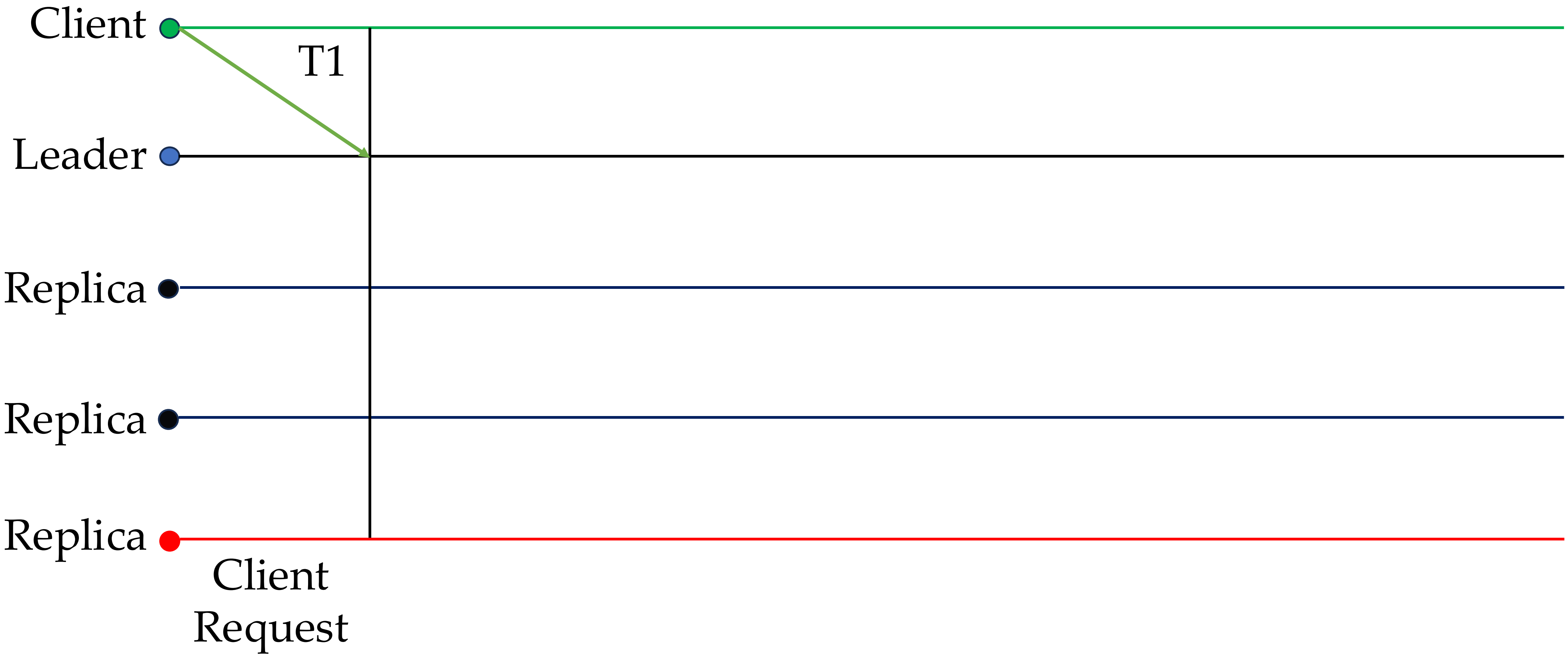
- Twice the throughput.

- No change in the number of phases necessary to commit a transaction.

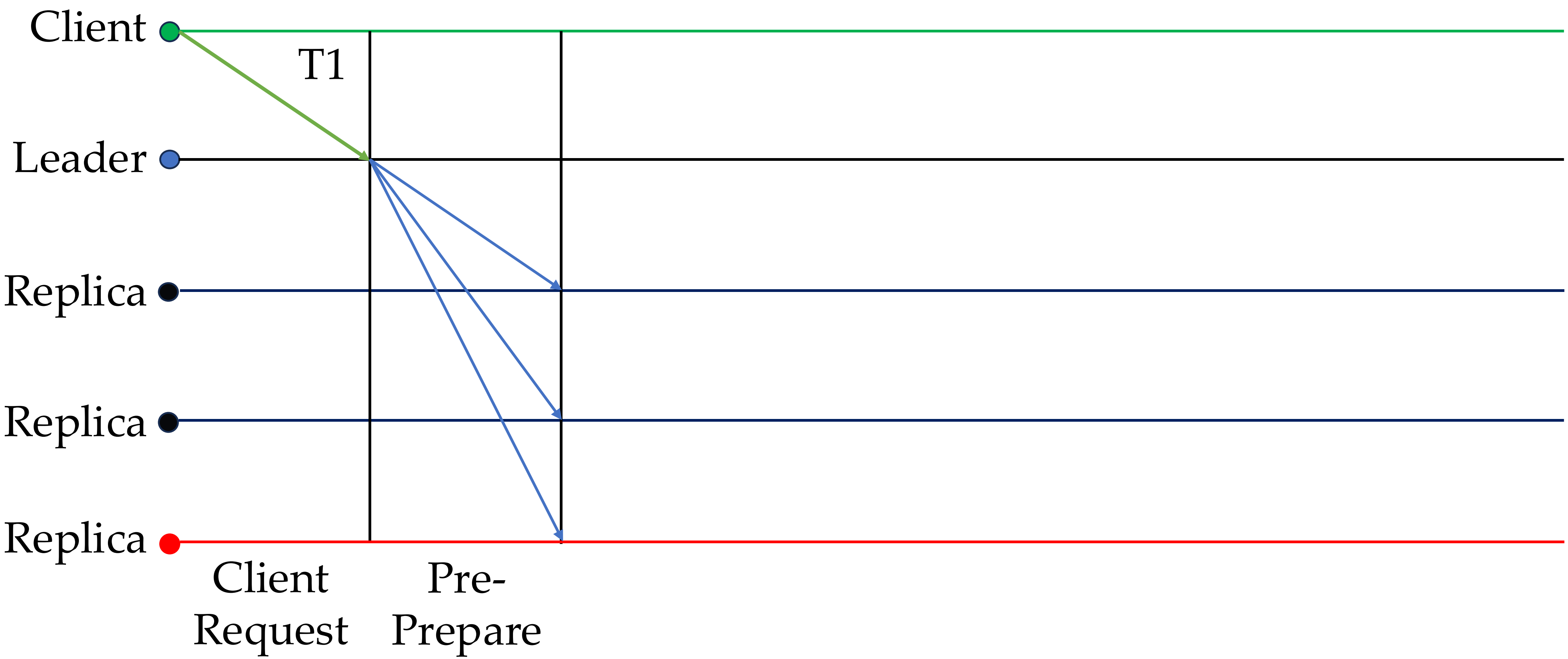
Streamlined HotStuff-2



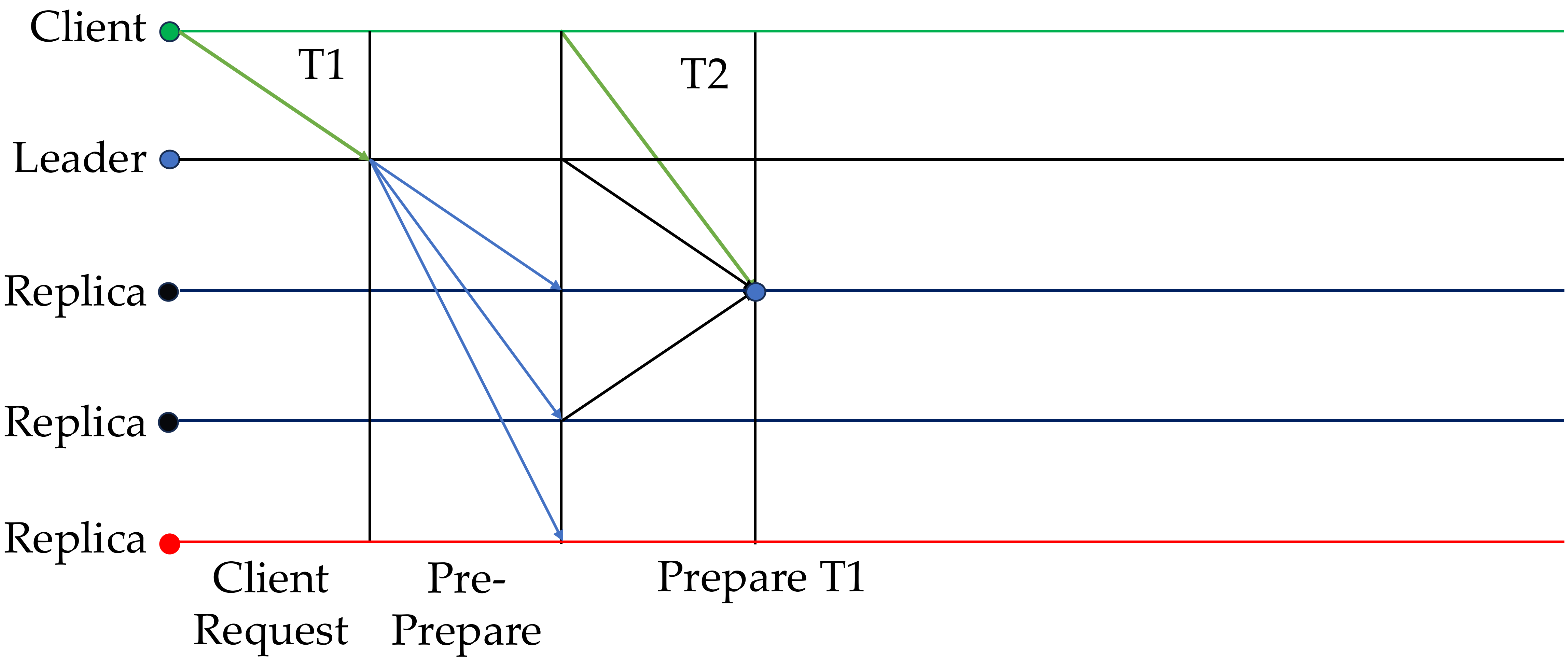
Streamlined HotStuff-2



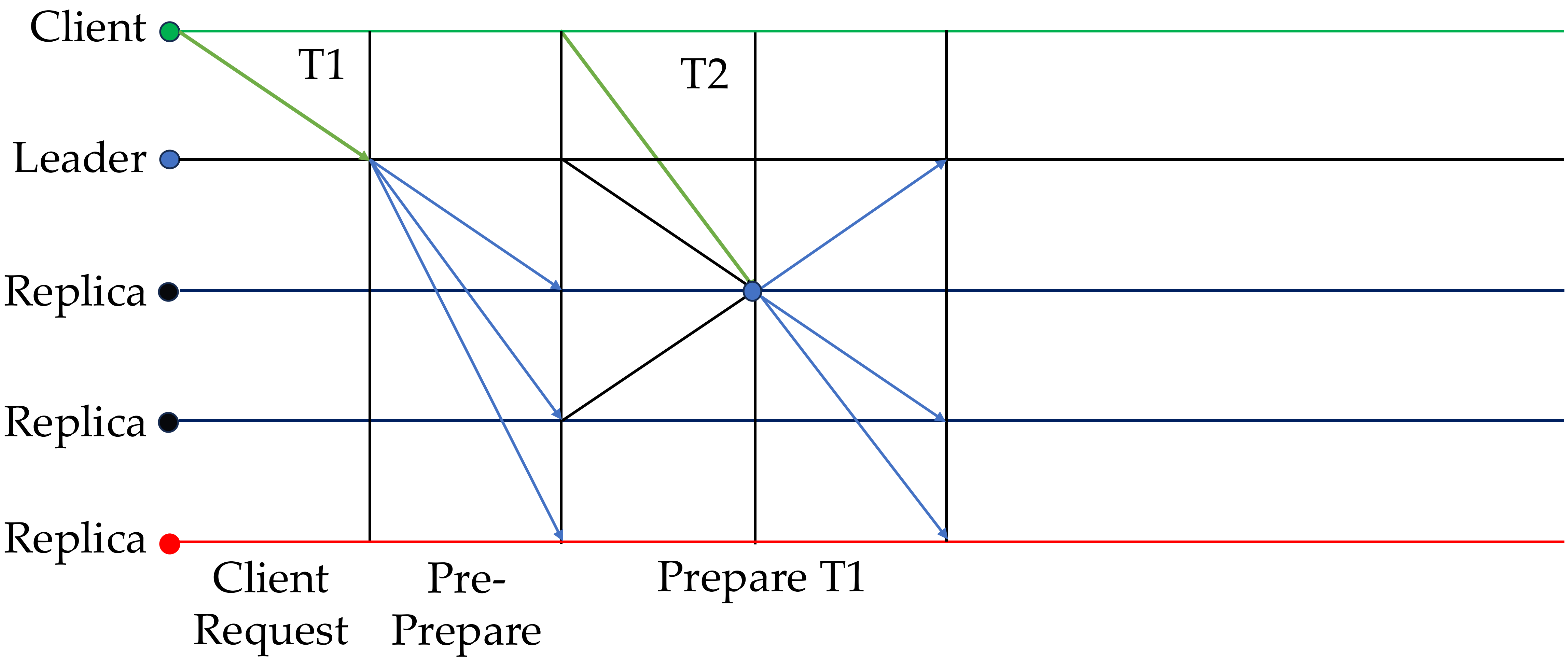
Streamlined HotStuff-2



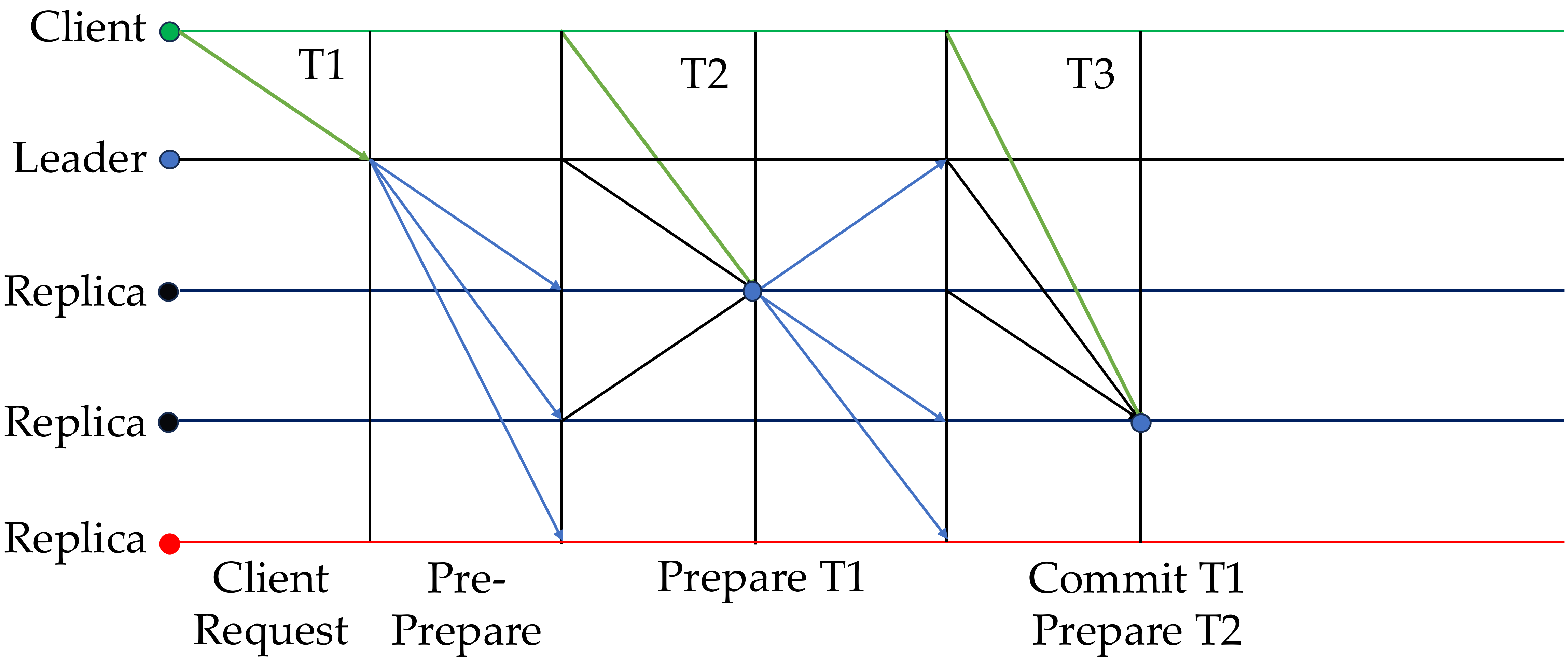
Streamlined HotStuff-2



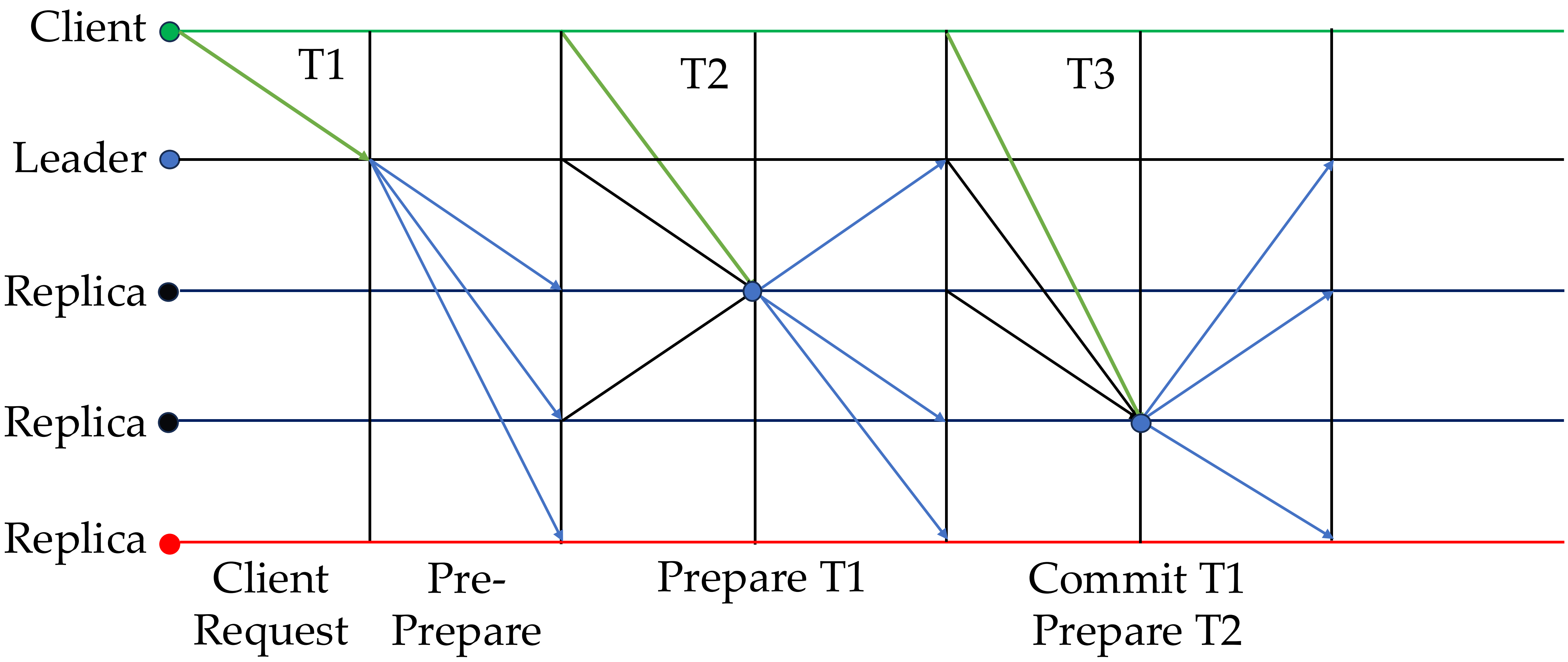
Streamlined HotStuff-2



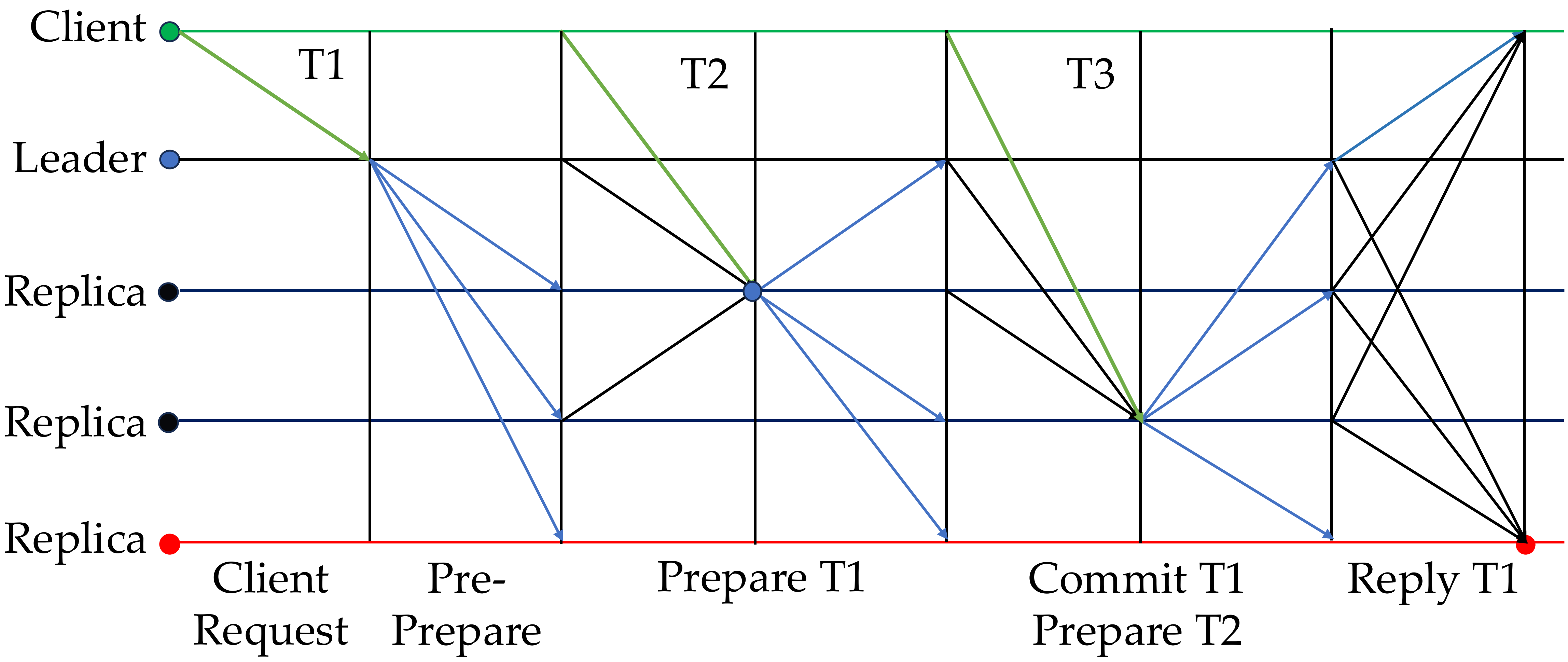
Streamlined HotStuff-2



Streamlined HotStuff-2



Streamlined HotStuff-2



Streamlined HotStuff-2

- The first QC is formed and stored by each replica at the end of Prepare phase.

- The second QC → at the end of Commit phase.

- Storing QC is same as locking the QC.
  - Essentially a guarantee by the replica that it will not support any QC formed at a lower view.

Can we do better?

HotStuff-1

- **HotStuff-1 [SIGMOD’2025]** → Speculation + HotStuff-2.
  - More challenging than PoE as regular view changes!

- HotStuff-1 + Slotting:
  - Increases throughput by giving leaders multiple proposal slots.