Robust and High Performance Consensus Protocols

PhD Private Defense

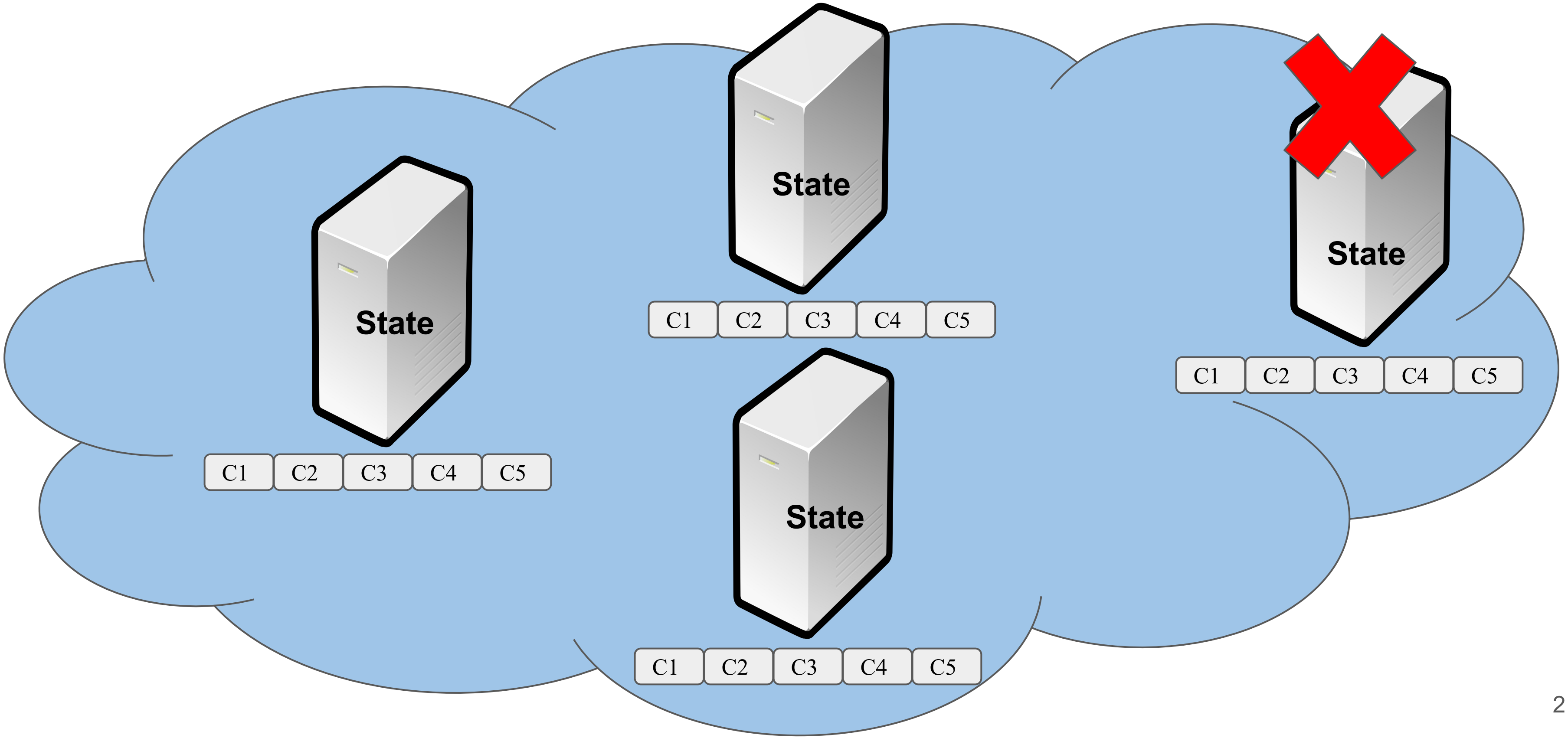
Pasindu Tennage

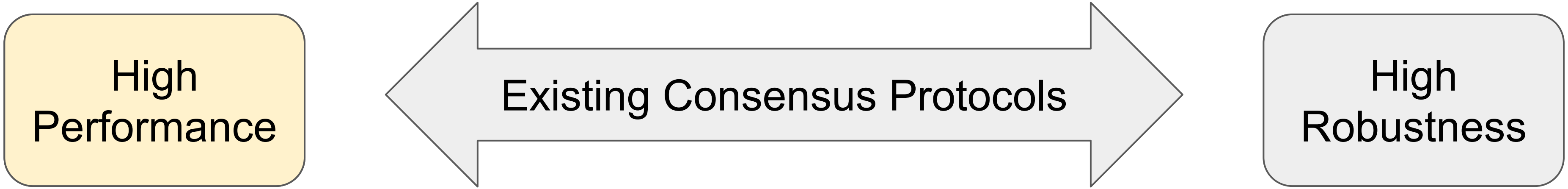
Thesis director: Bryan Ford
Thesis co-director: Lefteris Kokoris-Kogias

DEDIS

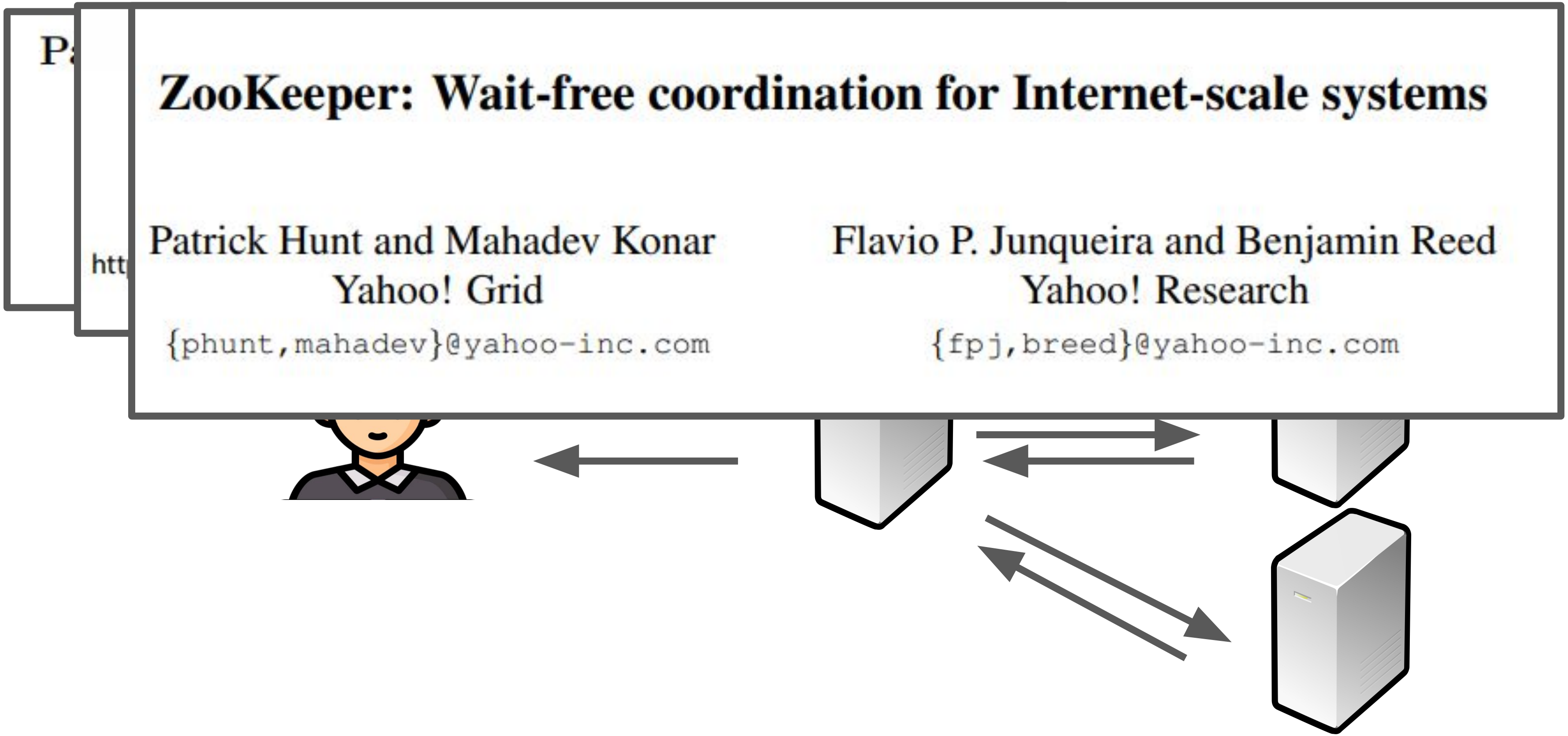
EPFL

Consensus


State
C1 C2 C3 C4 C5
State
C1 C2 C3 C4 C5
State
C1 C2 C3 C4 C5
State
C1 C2 C3 C4 C5


High
Performance
Existing Consensus Protocols
High
Robustness

High Performance using Leader-based Consensus

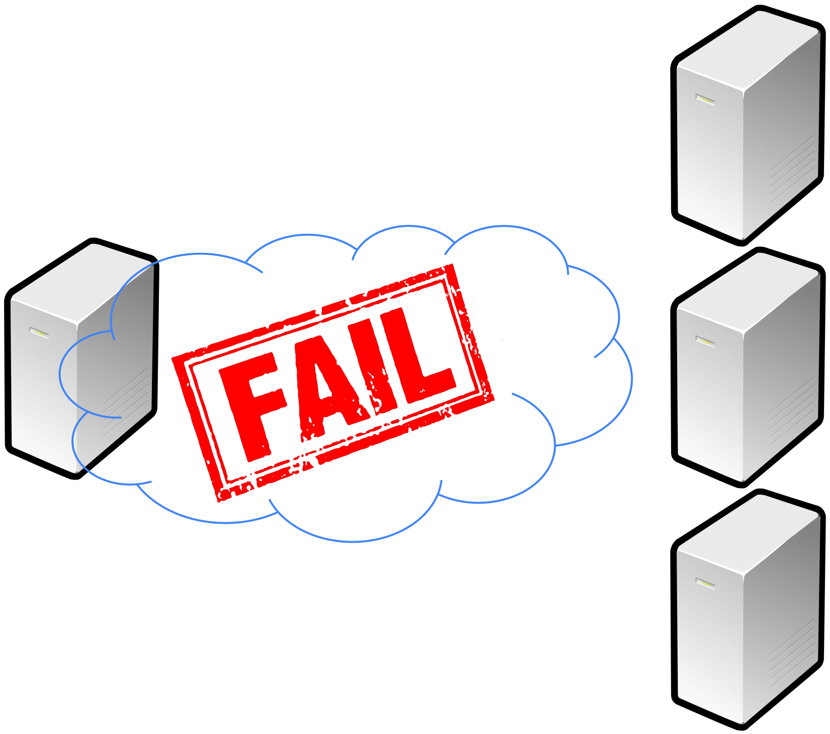


ZooKeeper: Wait-free coordination for Internet-scale systems

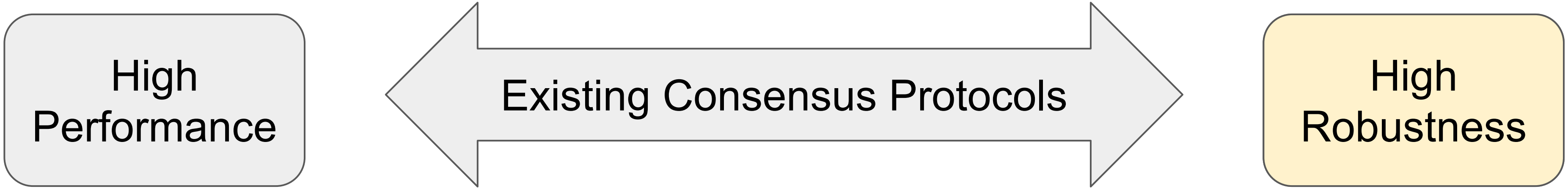
Patrick Hunt and Mahadev Konar
Yahoo! Grid
{phunt,mahadev}@yahoo-inc.com

Flavio P. Junqueira and Benjamin Reed
Yahoo! Research
{fpj,breed}@yahoo-inc.com

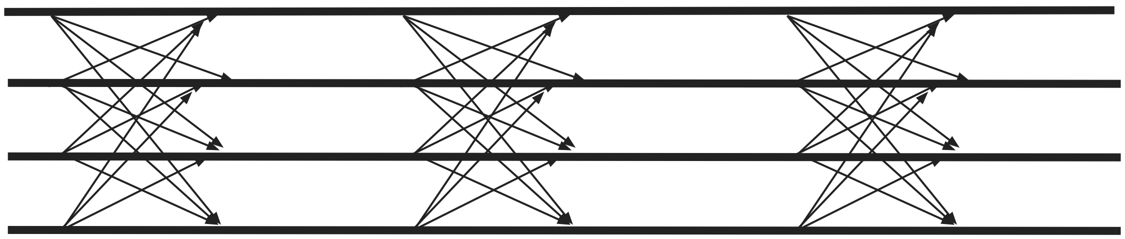
Robustness Problem of Leader Based Protocols



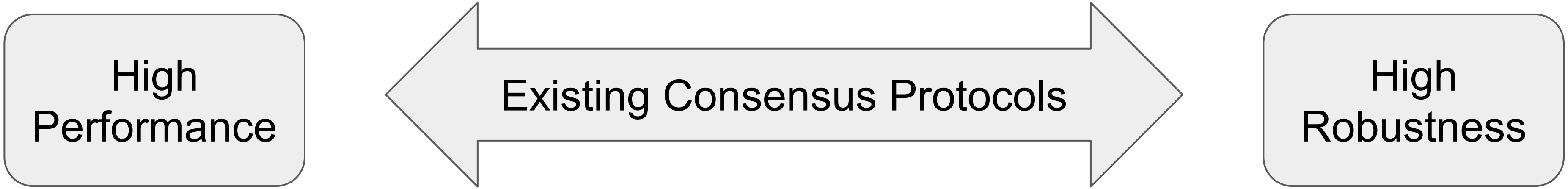
- Network partition.
- Link failures.
- DDoS attacks.
- Leader crash.


High
Performance
Existing Consensus Protocols
High
Robustness

Robust randomized consensus protocols



- Less efficient.
    - $O(n^2)$ / $O(n^3)$
- Hard to understand.
- Rarely deployed.



Can we have the best of both worlds?

Thesis goals

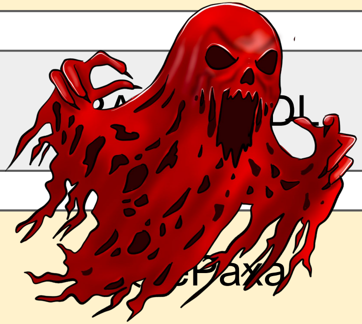
Explore the robustness and performance challenges of existing protocols

Design and evaluate new protocols that achieve both robustness and high performance

Thesis Contributions

Baxos

Explores mechanisms to avoid the impact of leader-targeted attacks



Explores mechanisms to avoid leader performance bottleneck and the impact of network asynchrony

Explores mechanisms to avoid the tyranny of timeouts

Mahi-Mahi

Explores mechanisms to avoid high latency and high resource consumption in blockchain consensus protocols

Publications

QuePaxa	Published in SOSP 2023
Mahi-Mahi	Under review in ICDCS 2025
RACS-SADL	Under review in IEEE CLOUD 2025

Thesis Scope

In Scope

- Total Ordering.

Out of Scope

- Node / committee reconfiguration.
- Transaction execution.
- Sharding.
- Distributed transactions.

Outline

- Baxos
- QuePaxa
- Mahi-Mahi
- Summary
- Future Work

Baxos: Backing off for robust consensus

Pasindu Tennage*, Cristina Basescu, Lefteris Kokoris-Kogias, Ewa Syta, Philipp Jovanovic, Bryan Ford

Baxos Outline

- Problems with leader based protocols.
- Baxos design.
- Evaluation.

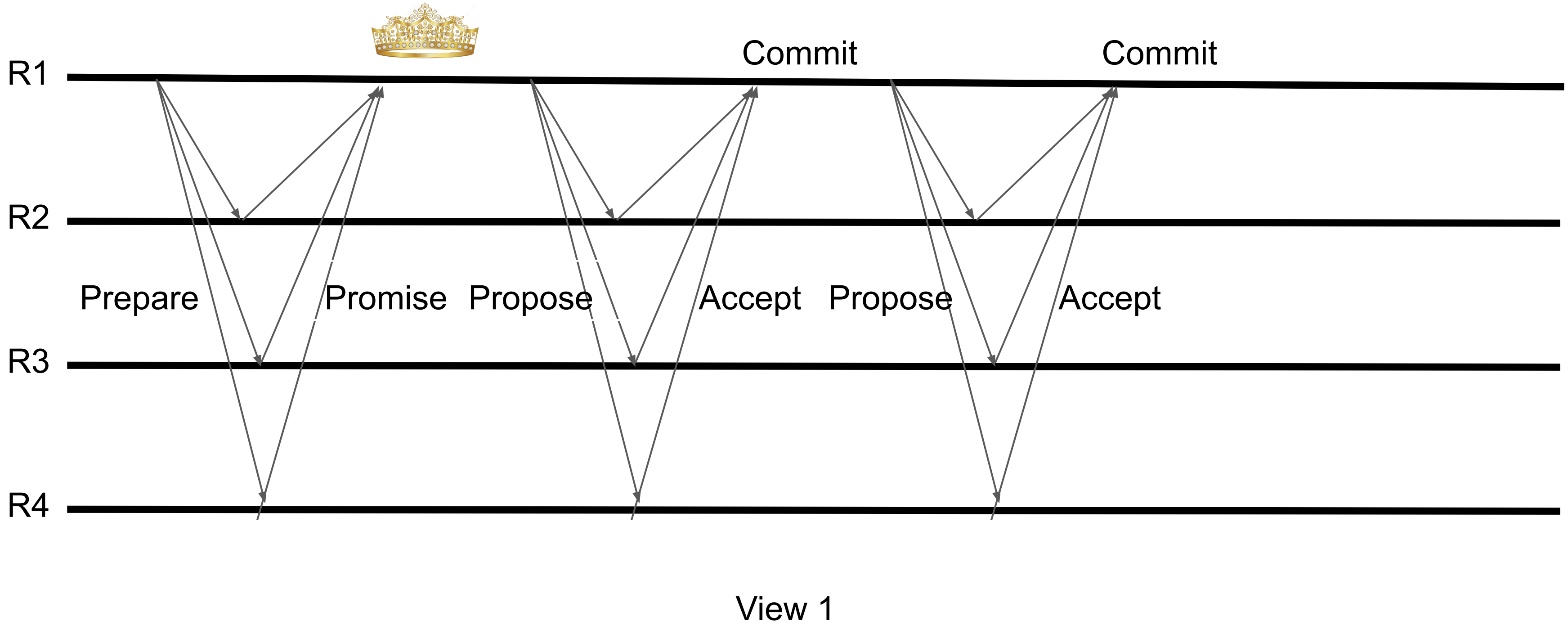
Problems with leader-based protocols

Cost of view change

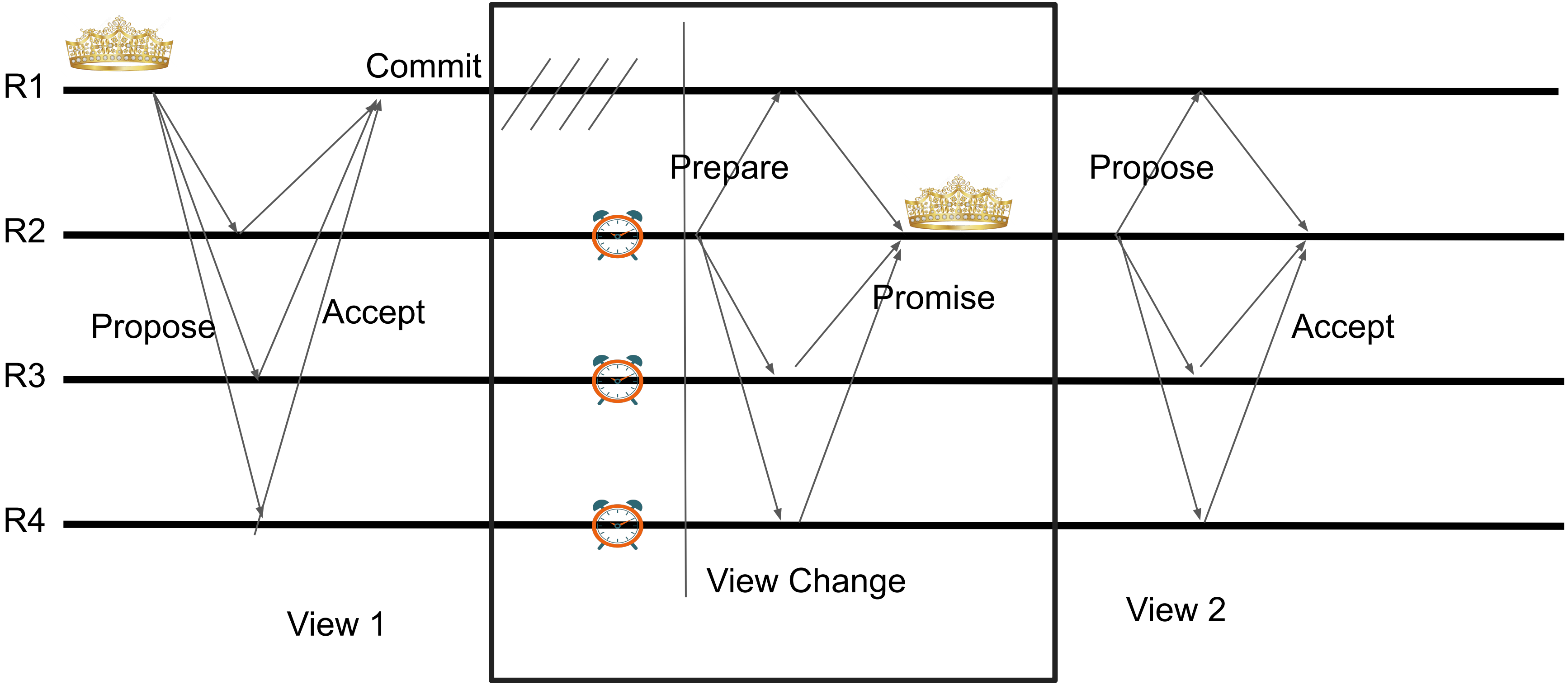
Leader-targeted attacks

Variability in resource usage

Normal case operation of Multi-Paxos



Timeout based view change in Multi-Paxos



Problems with view change

No commands committed during view change

Complex and error prone

- Catch-up.
- Synchronizer.
- Ignored in prototypes

Problems with leader-based protocols

Cost of view change

Leader-targeted attacks

Variability in resource usage

Leader-targeted attacks

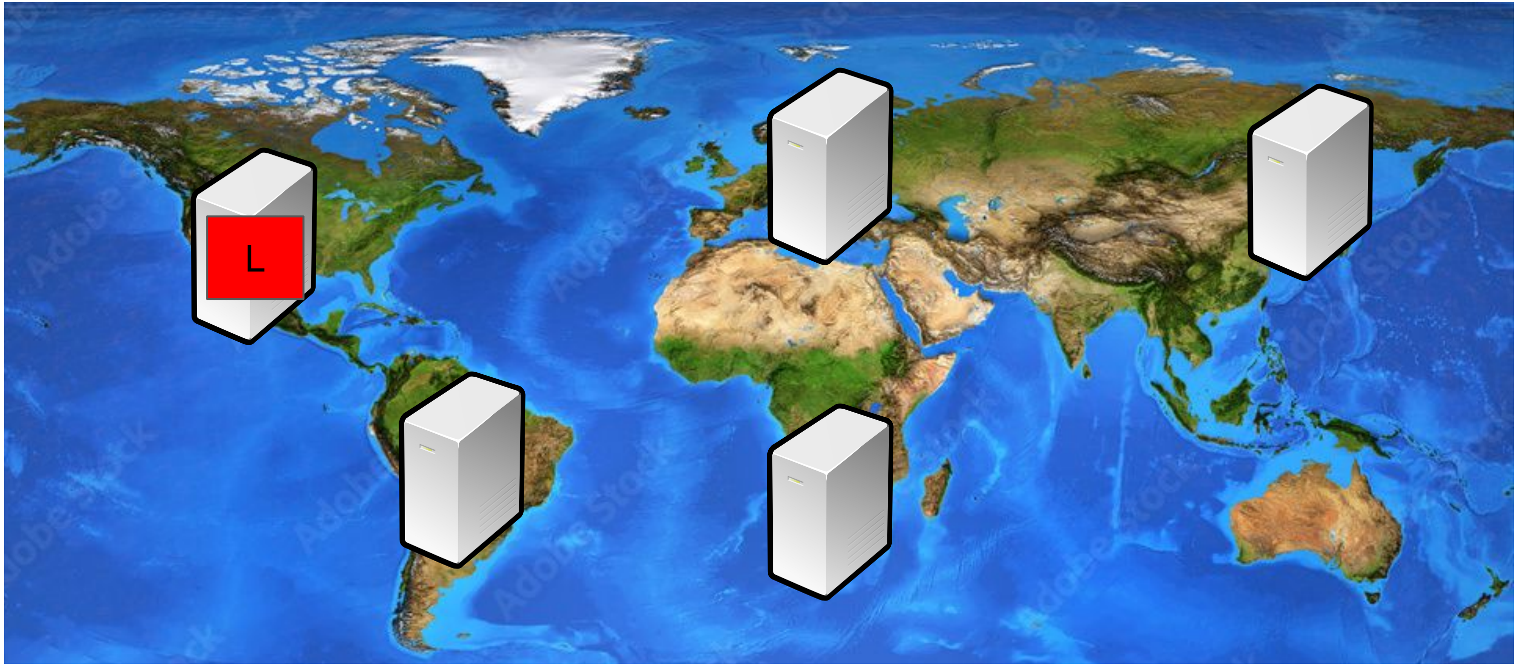
Problems with leader-based protocols

Cost of view change

Leader-targeted attacks

Variability in resource usage

Resource utilization variability



Baxos Overview

Based on Paxos

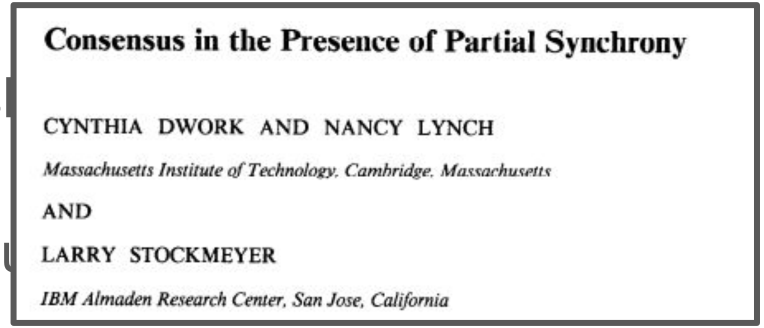
Replaces view change with random exponential backoff

Threat Model

- Up to ***f*** out of ***2f+1*** nodes can crash

- The network is partially synchronous

- Network attacker
  - Can find and attack the current leader.



Consensus in the Presence of Partial Synchrony

CYNTHIA DWORK AND NANCY LYNCH

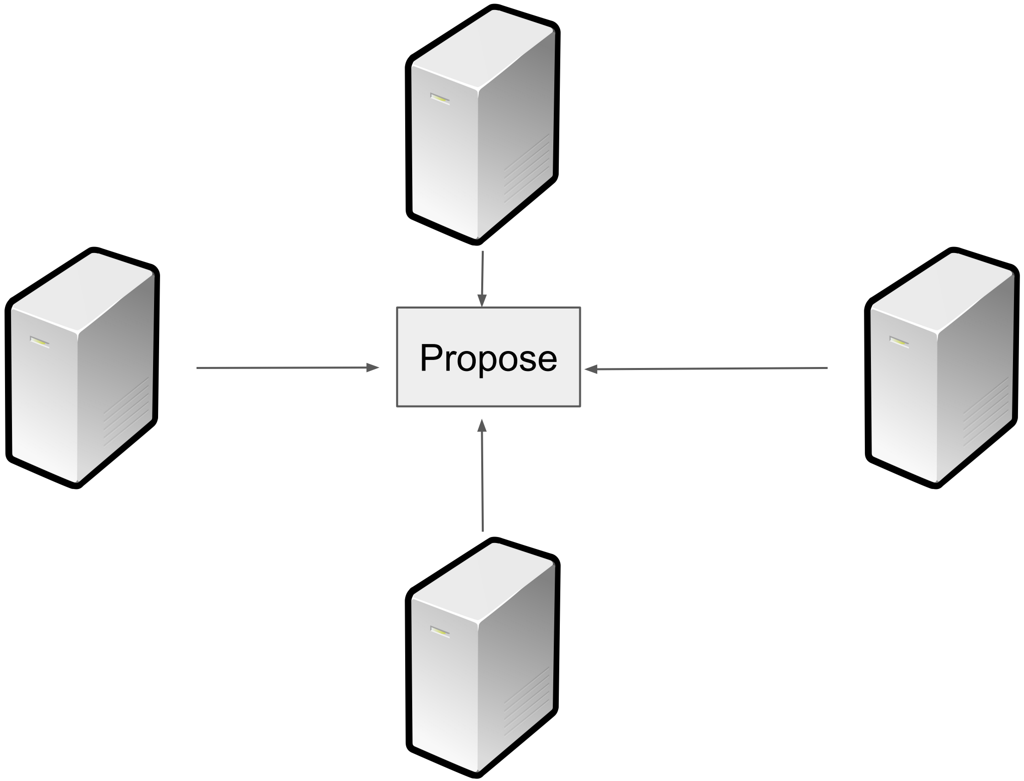
Massachusetts Institute of Technology, Cambridge, Massachusetts

AND

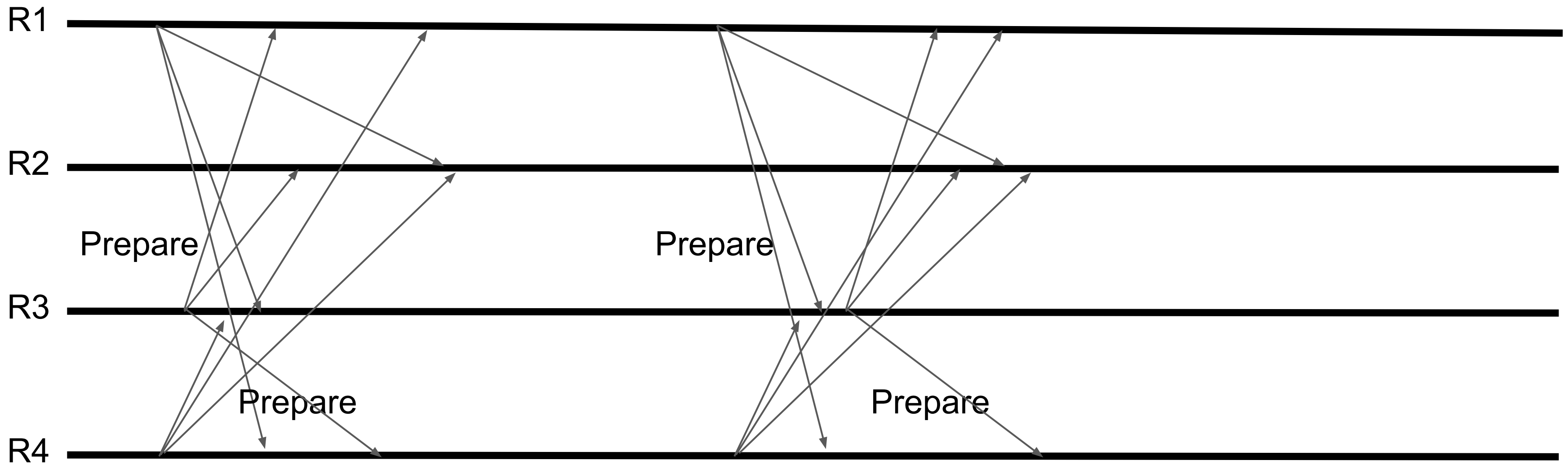
LARRY STOCKMEYER

IBM Almaden Research Center, San Jose, California

Baxos allows all replicas to propose



Contention under concurrent proposals



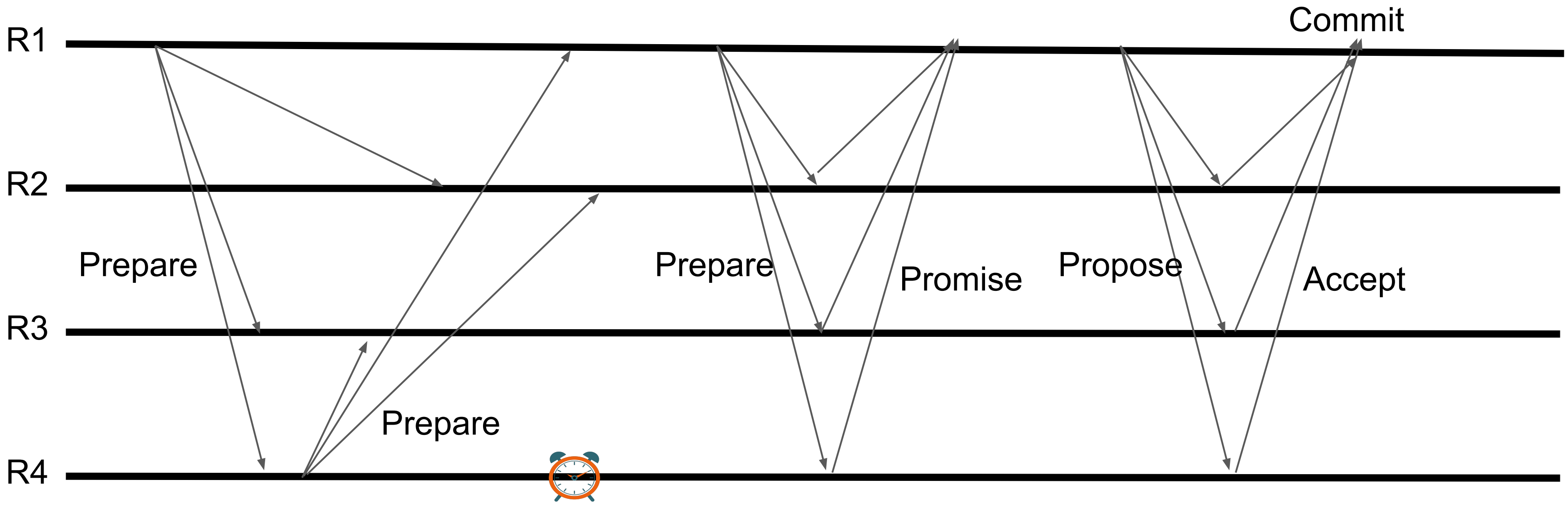
Random exponential backoff

Manage access to shared resources in networks (CSMA CD/CA)

Backing off before retrying to avoid contention

Can we apply REB to consensus to handle contention?

Baxos uses Random exponential Backoff

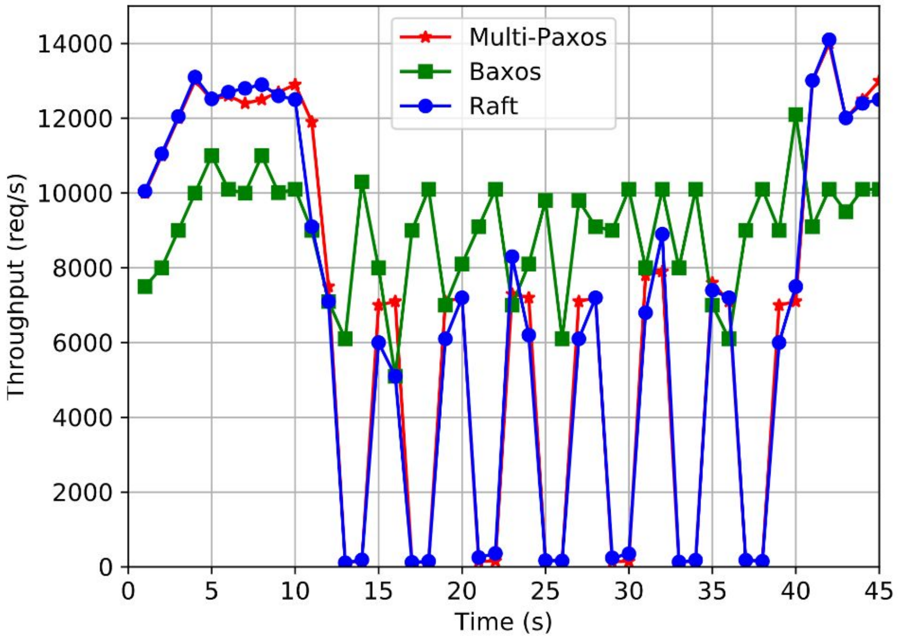


Multi-Paxos vs Baxos

Multi Paxos	Baxos
Uses Paxos core	Uses Paxos core
Only the leader proposes	Every node proposes
Uses view change	Uses REB

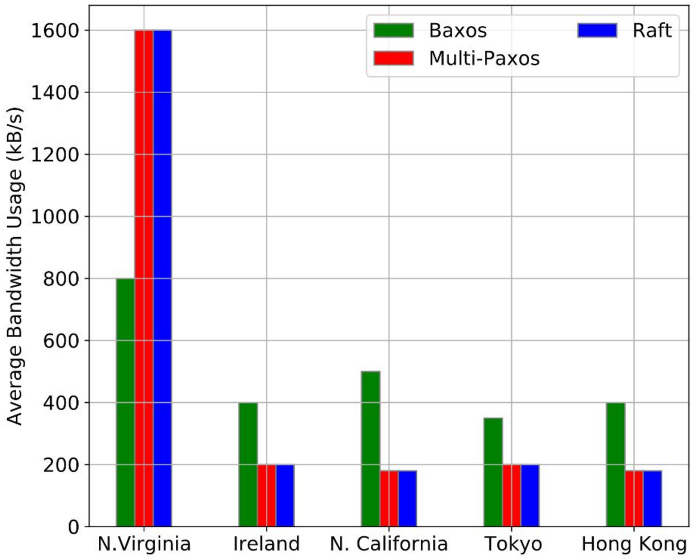
Baxos Evaluation

Robustness of Baxos



Baxos is resilient against leader-targeted attacks

Resource utilization of Baxos



Baxos has uniform resource utilization across replicas

Baxos Summary

Avoid view changes

Robust against leader-targeted attacks

Uniform resource usage

Outline

- Baxos
- **QuePaxa**
- Mahi-Mahi
- Summary
- Future Work

QuePaxa: Escaping the tyranny of timeouts

Pasindu Tennage*, Cristina Basescu, Lefteris Kokoris-Kogias, Ewa Syta, Philipp Jovanovic, Vero Estrada, Bryan Ford

QuePaxa Outline

- Tyranny of timeouts.
- QuePaxa.
- Evaluation.

Tyranny of Timeout Problems in Consensus

Timeout based view change	Conservative timeouts	Manually configured timeouts

Timeout based view change

View change succeeds only when the network is synchronous

⇨

Loss of liveness under asynchronous networks

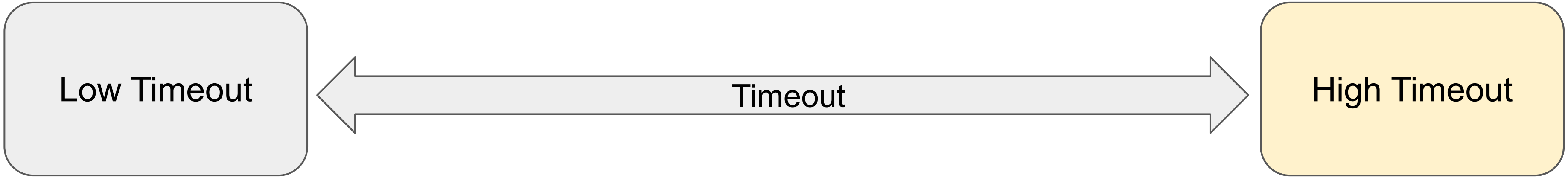
Tyranny of Timeout Problems in Consensus

Timeout based view change

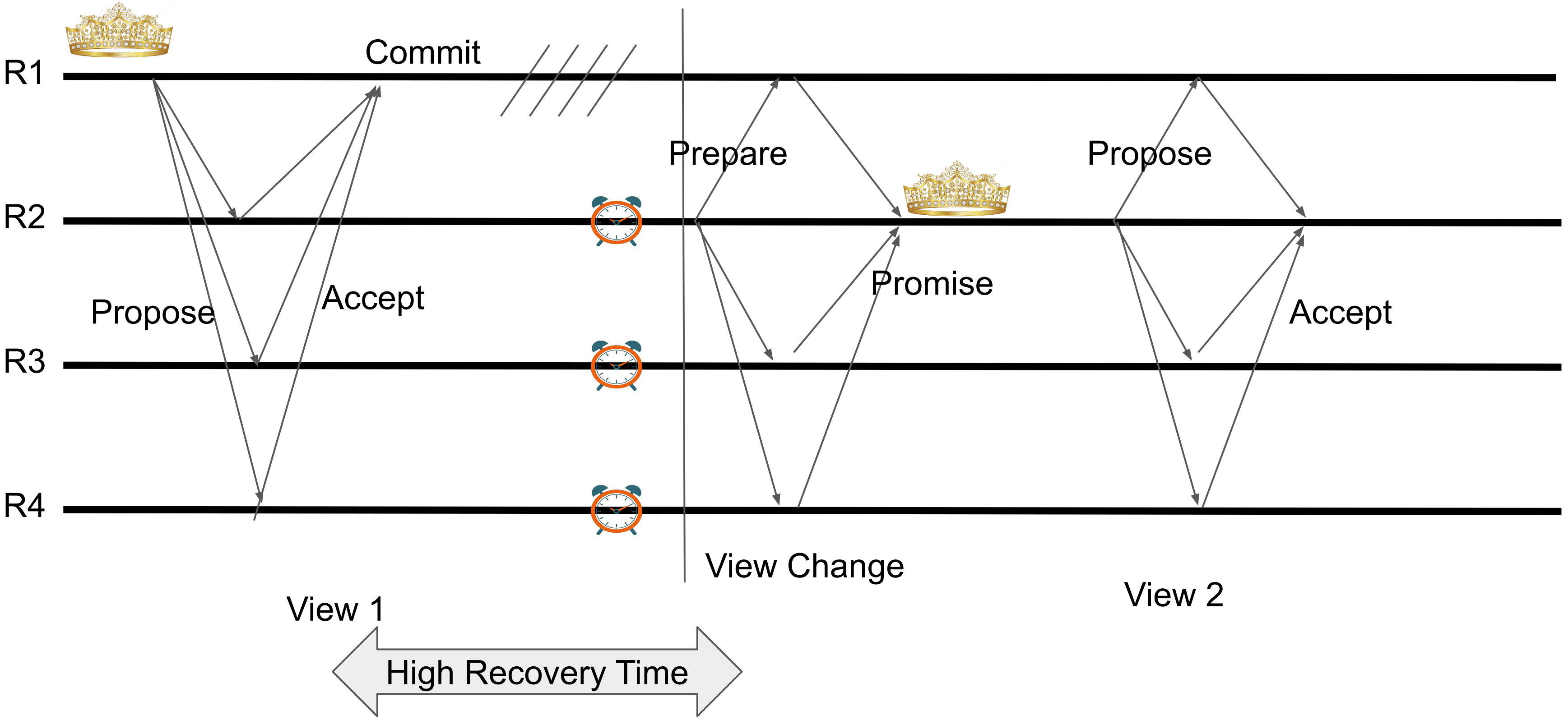
Conservative timeouts

Manually configured timeouts

Choosing Timeouts in leader based protocols



Timeout based view change [Multi-Paxos]



Choosing Timeouts in leader based protocols

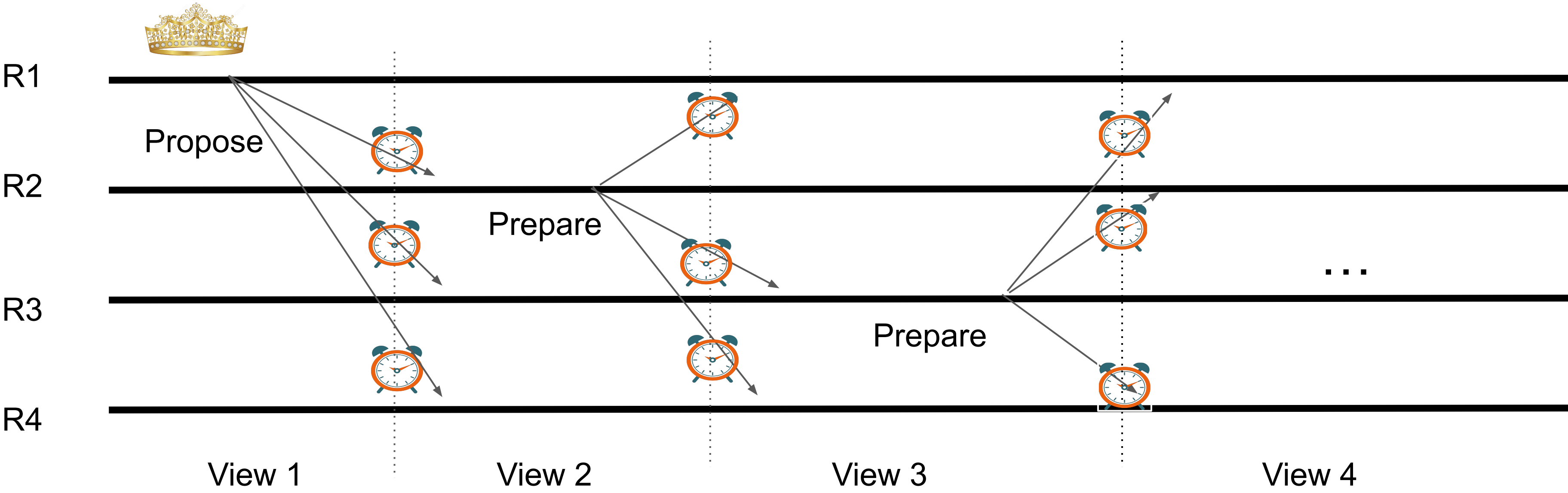
Low Timeout

Timeout

High Timeout

High Recovery Time

Liveness loss with low timeouts



Choosing Timeouts in leader based protocols

Low Timeout

Timeout

High Timeout

Liveness Loss

High Recovery Time

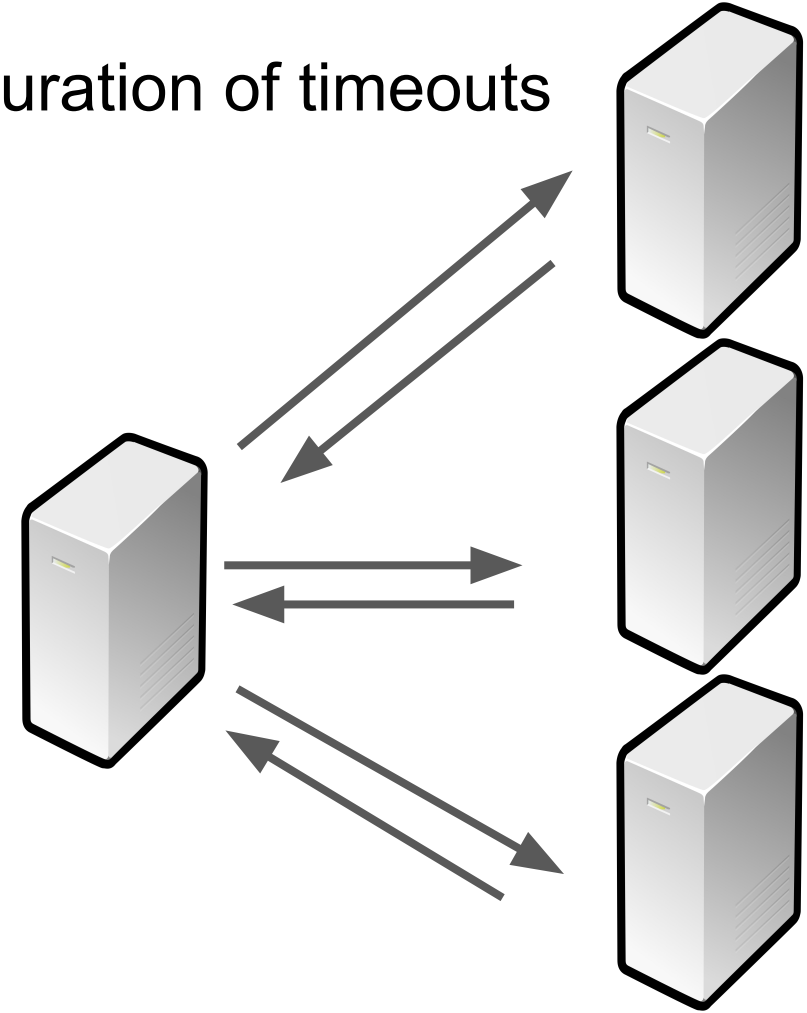
Tyranny of Timeout Problems in Consensus

Timeout based view change

Conservative timeouts

Manually configured timeouts

Manual configuration of timeouts

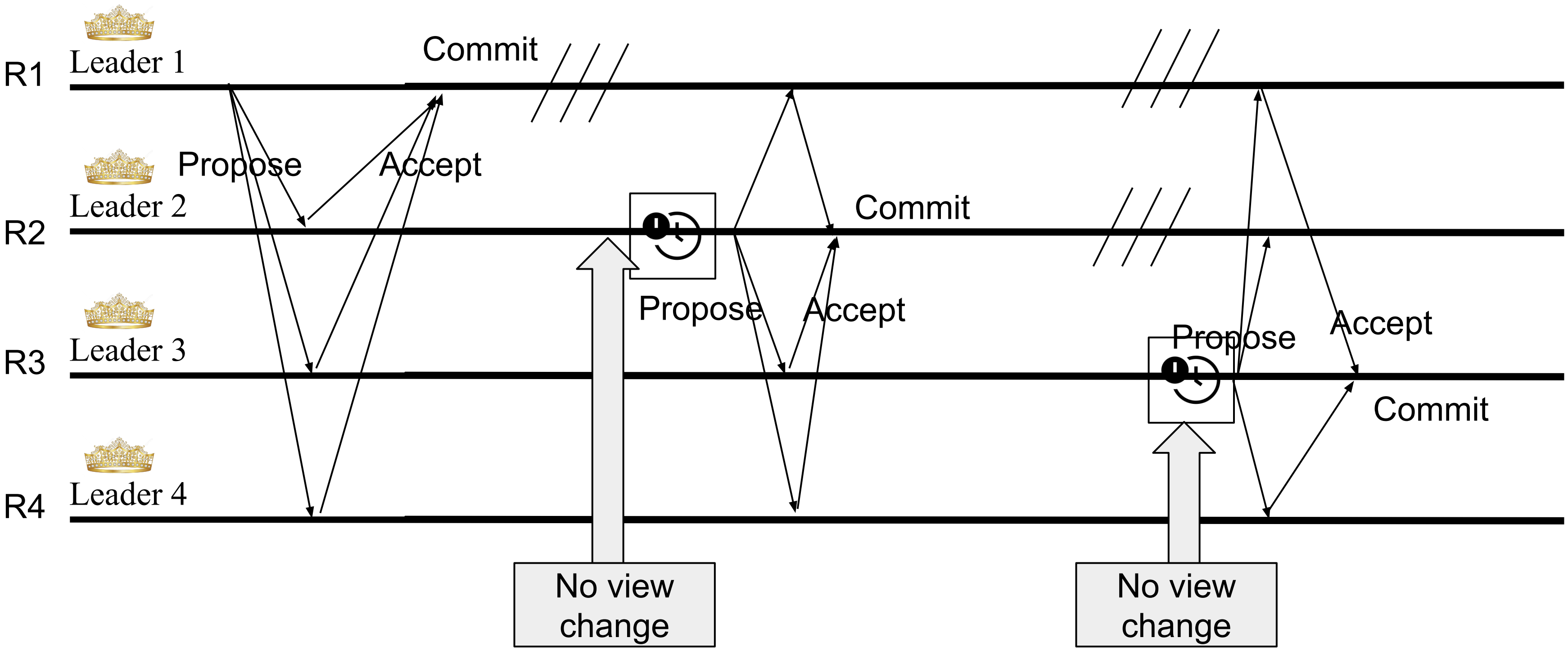


- Slow but functioning leader.
- Timeout does not adapt to changing network delay.

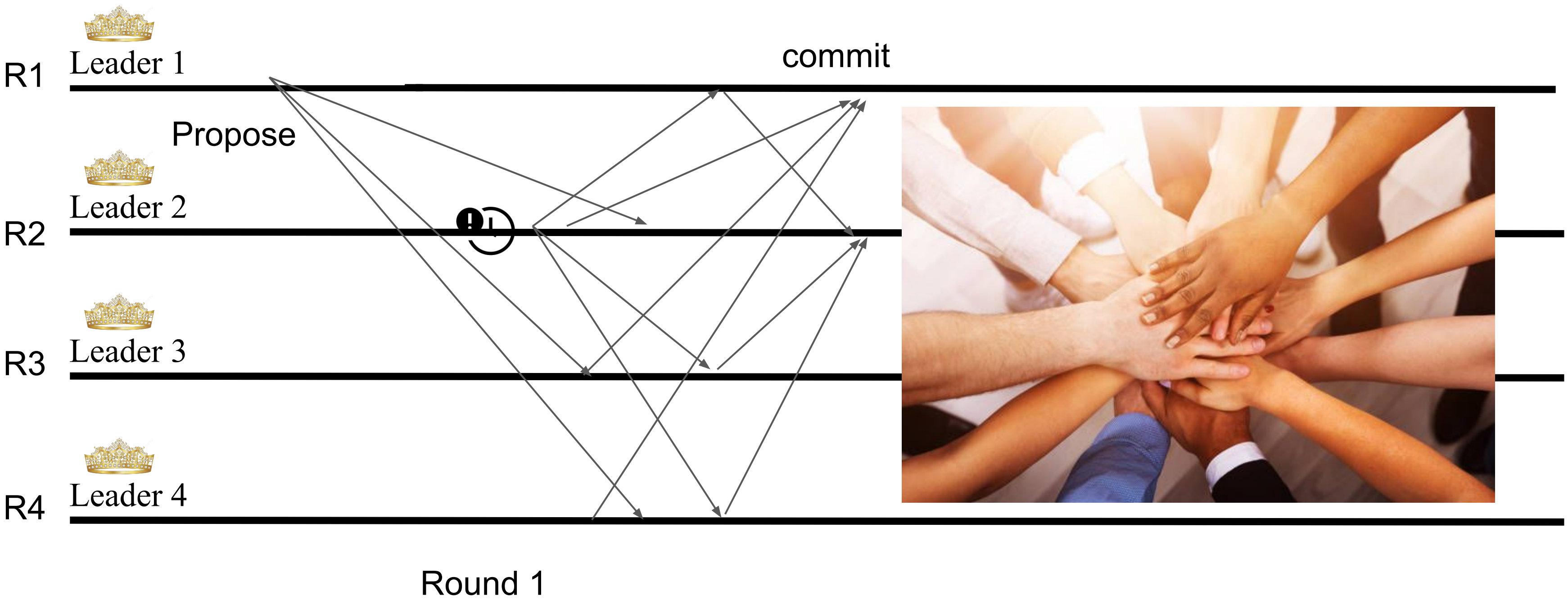
Are timeouts necessary for progress?

Can we eliminate the impact of timeout for liveness?

An alternative approach? (Hedging)



What if multiple leaders could **cooperate** instead of **interfere**?



QuePaxa Contributions

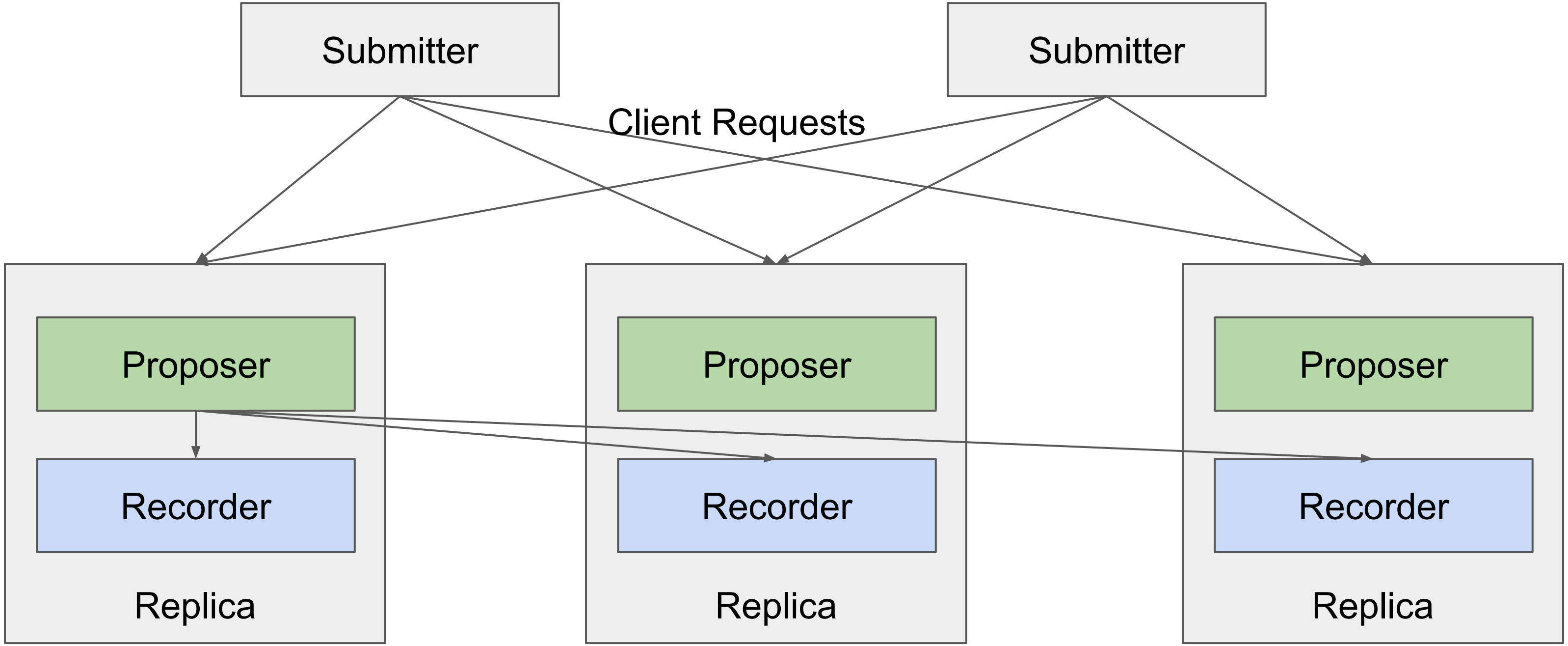
Optimal Performance under synchrony and asynchrony

Enables Hedging

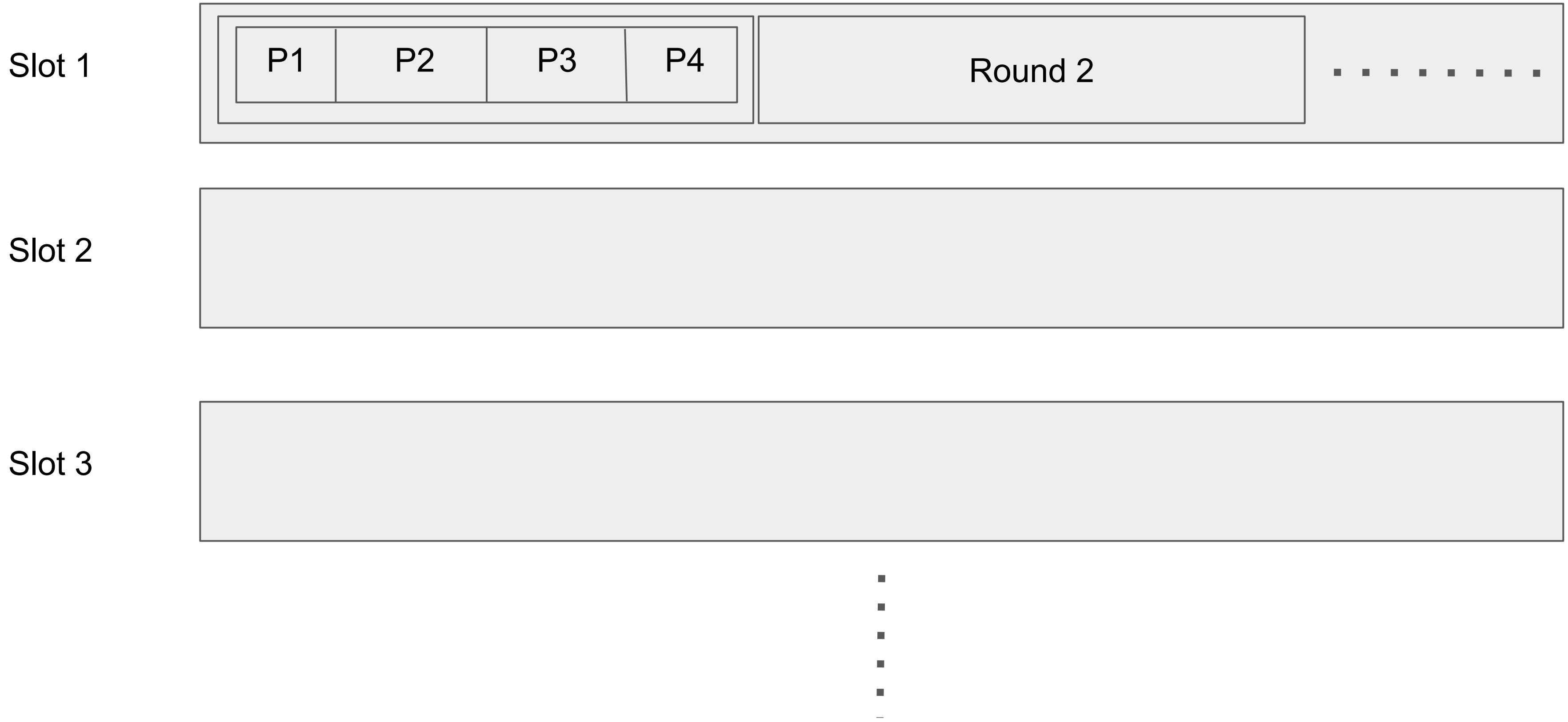
Threat Model

- Up to ***f*** out of ***2f+1*** nodes can crash.
- The network is **asynchronous** – there exists **no bound Δ** on message transmission delay.
- Network attacker
  - Can reorder and delay messages.
  - Cannot see internal replica state and message contents.

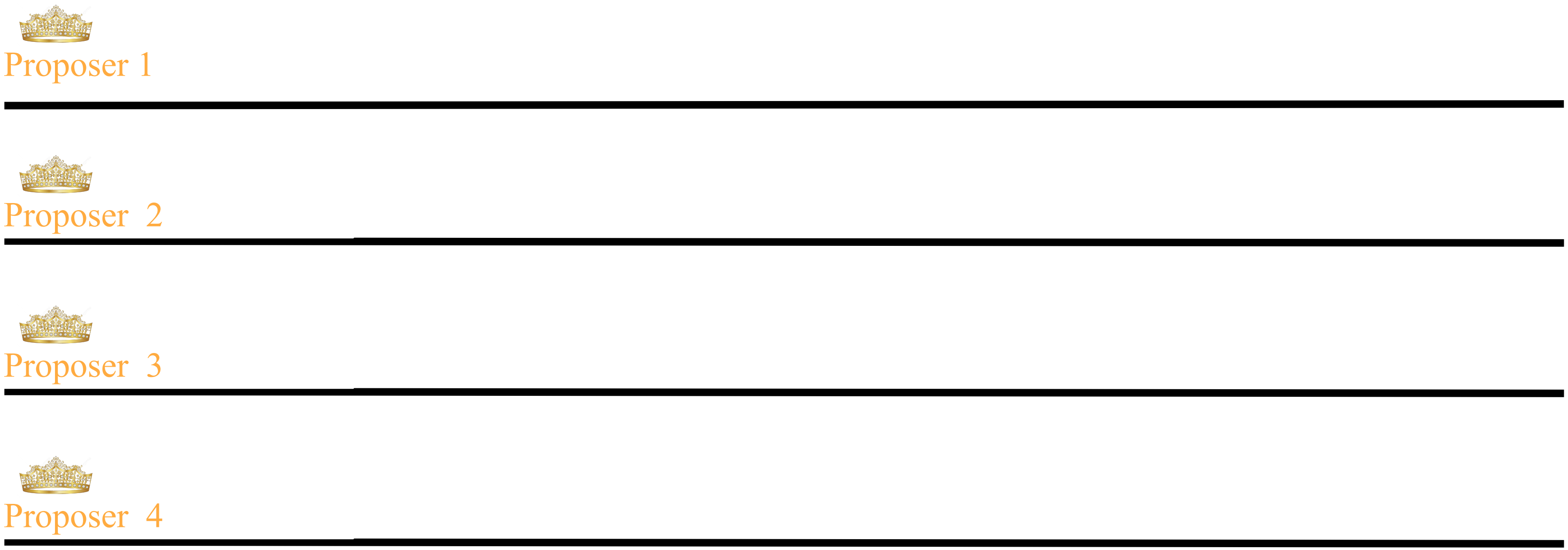
QuePaxa Architecture



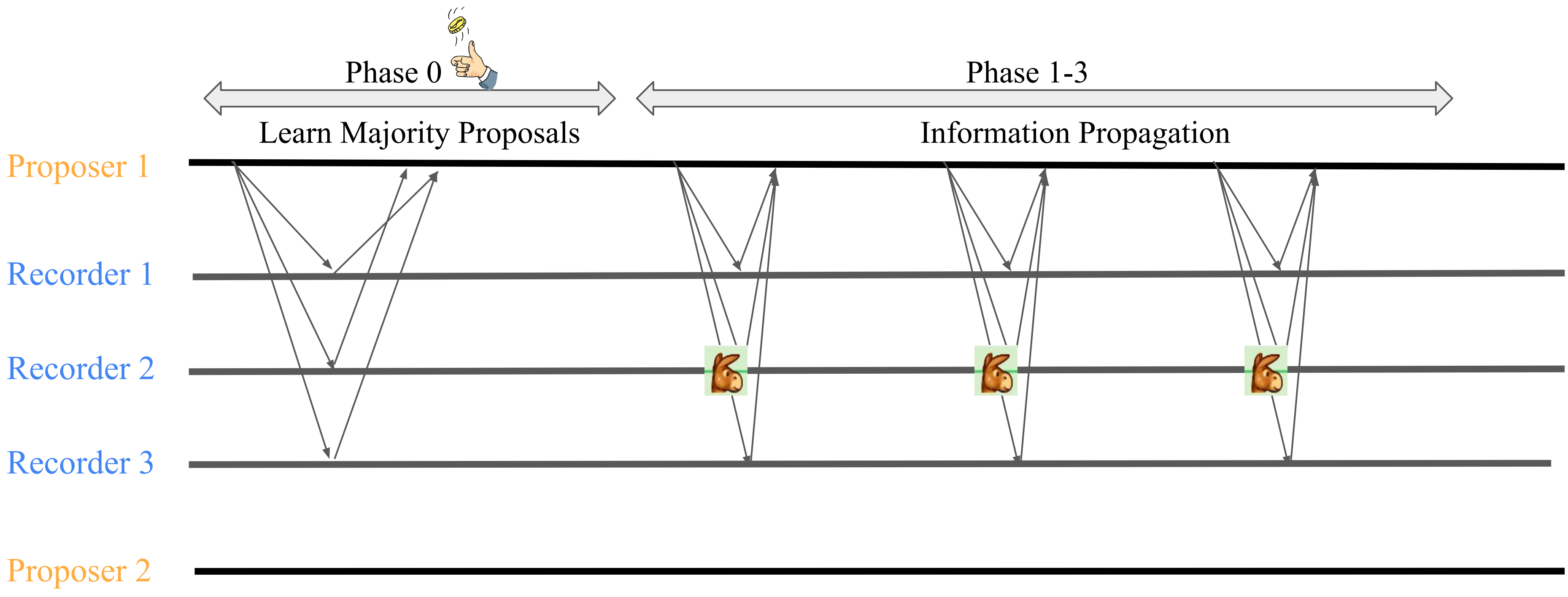
QuePaxa Log Structure



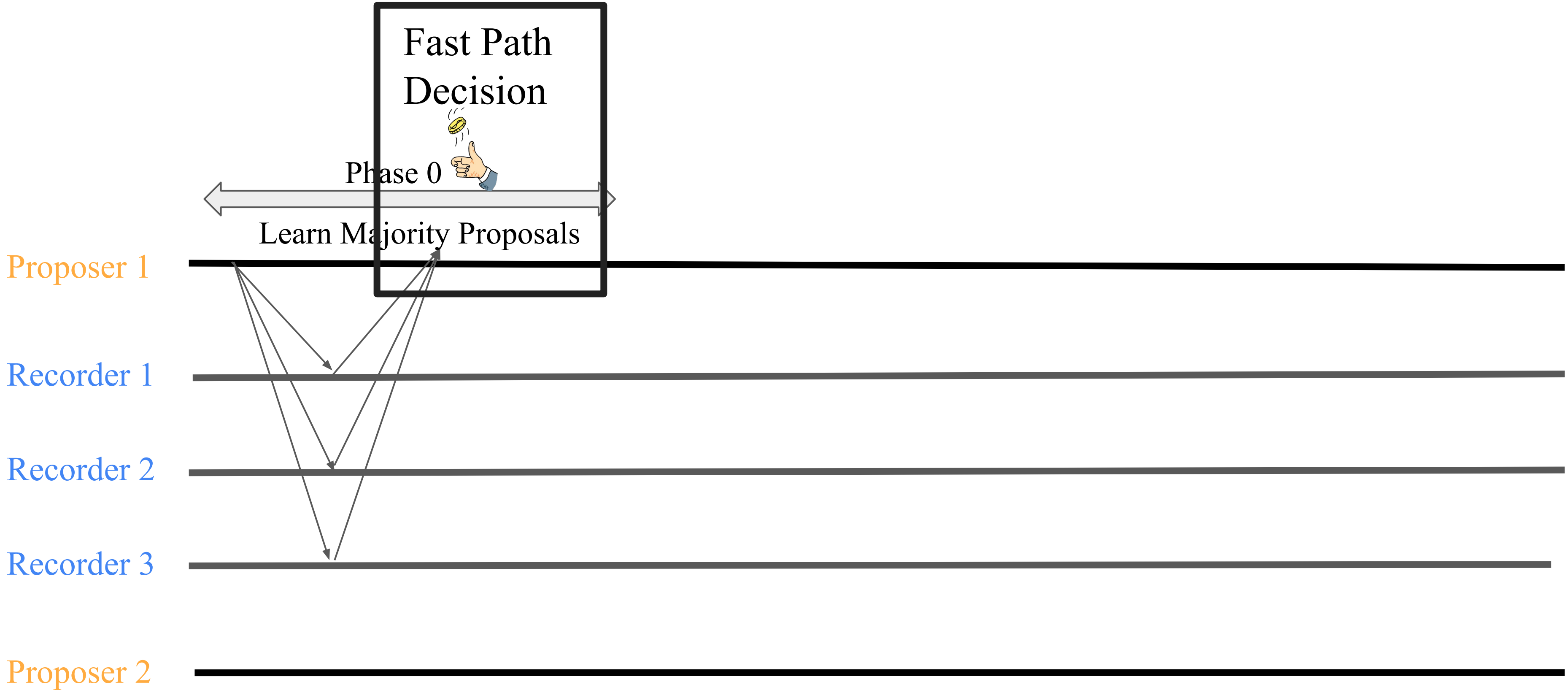
QuePaxa Proposer Sequence



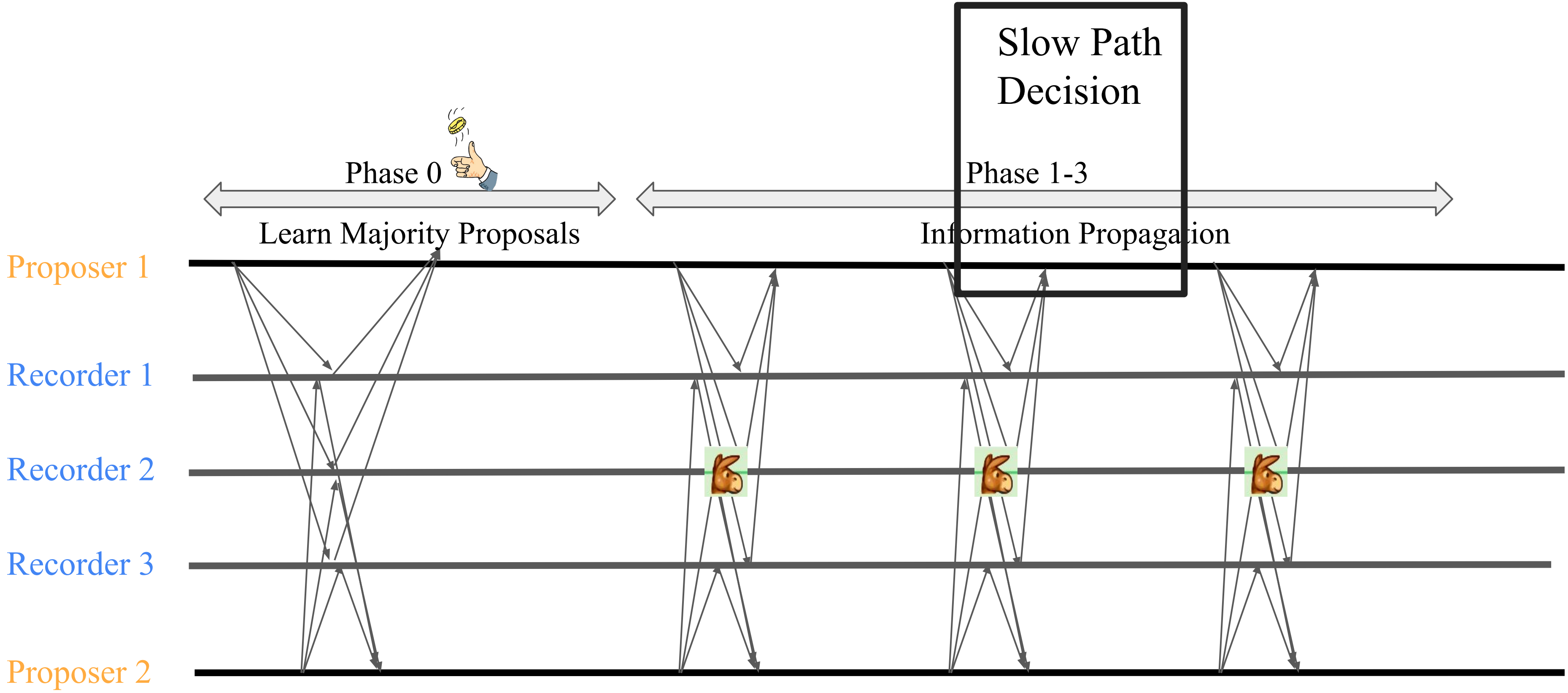
QuePaxa Protocol Diagram



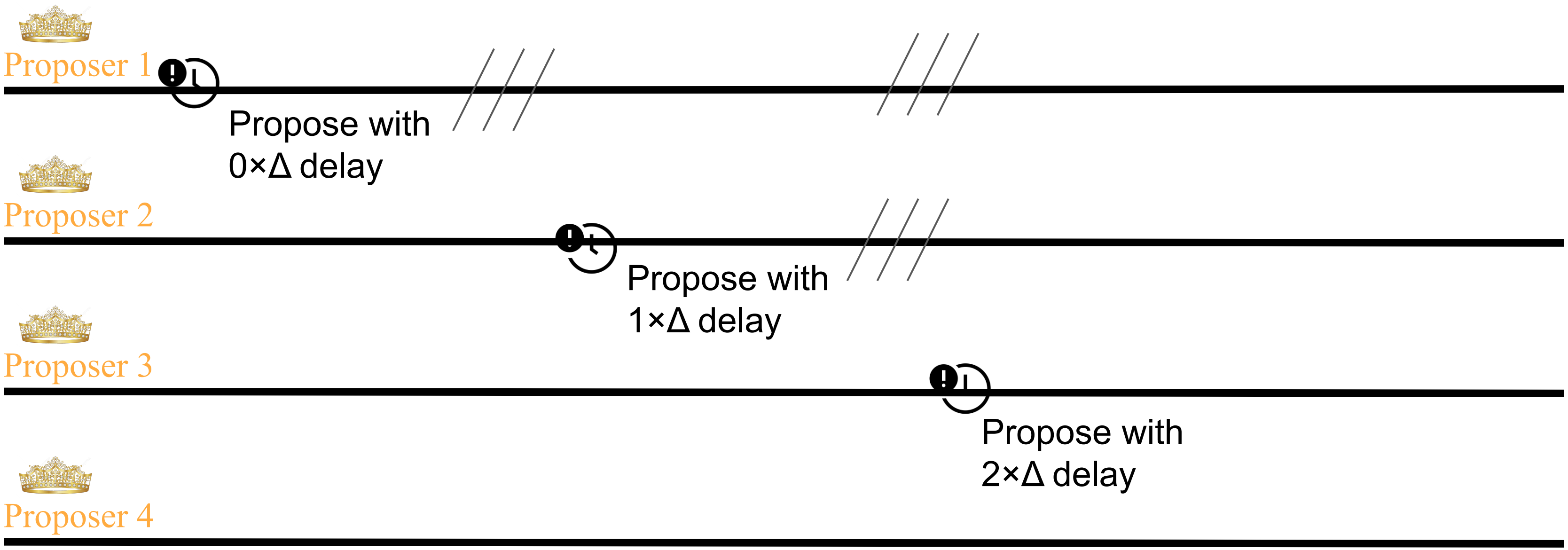
QuePaxa Synchronous Execution



QuePaxa Asynchronous Execution



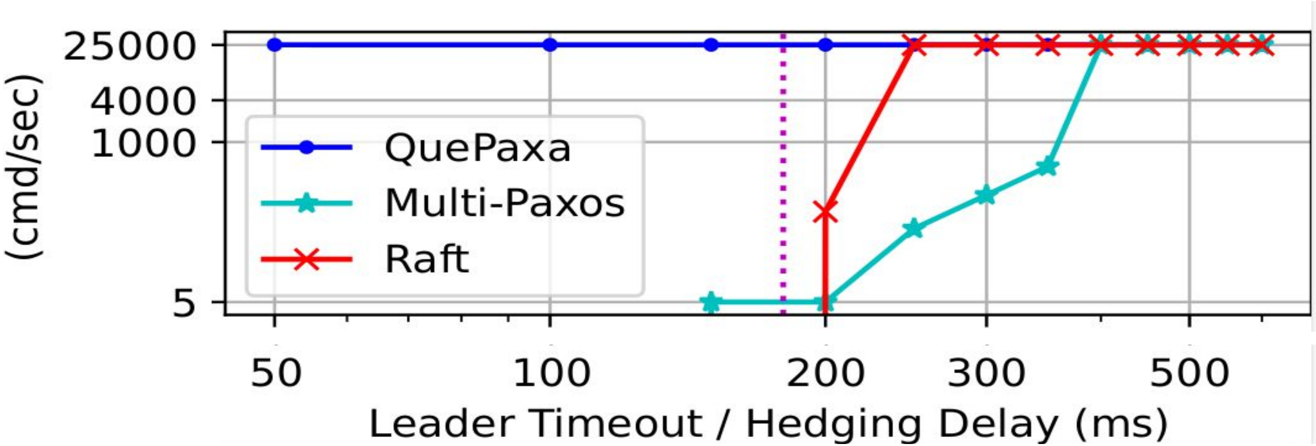
Hedging in QuePaxa



Evaluation

Effect of Hedging in Quepaxa

Throughput

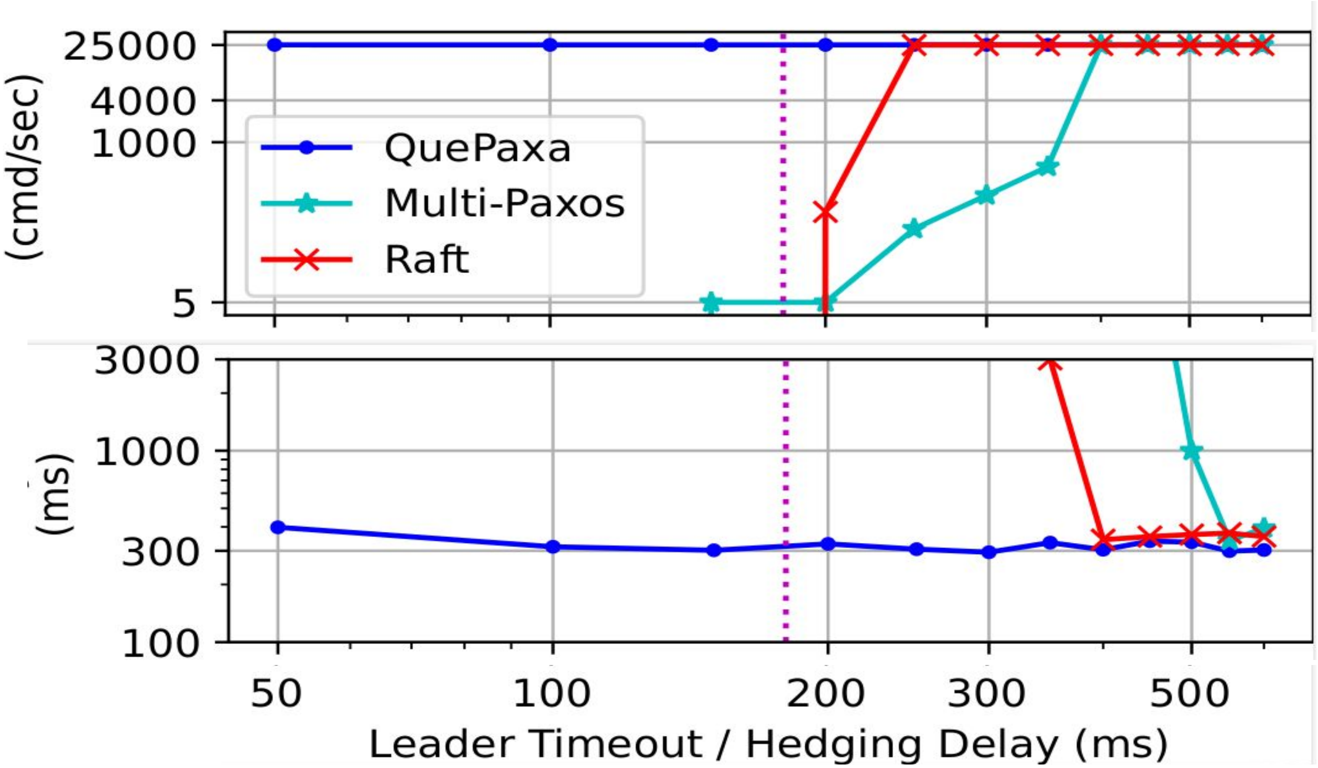


Liveness of QuePaxa does not depend on the timeout

Effect of Hedging in Quepaxa

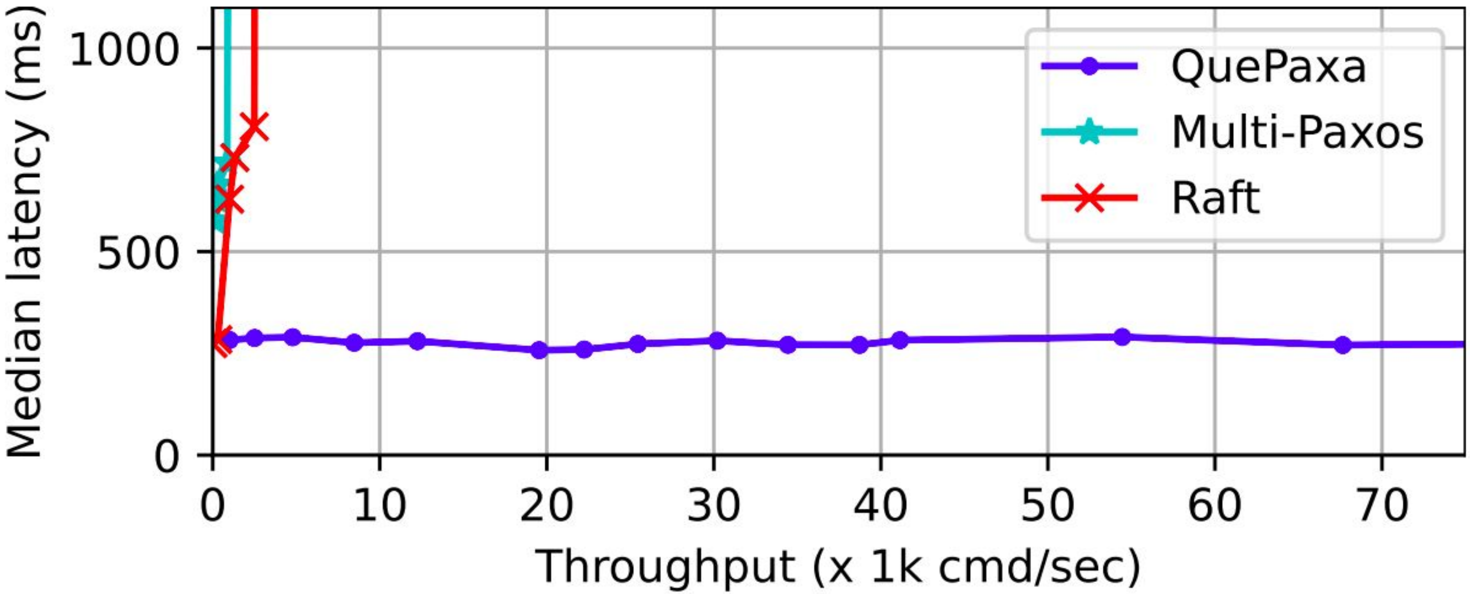
Throughput

Recovery Time



QuePaxa has low recovery time

Performance under adversarial networks



QuePaxa is resilient to adversarial attacks and asynchronous network conditions

QuePaxa Summary

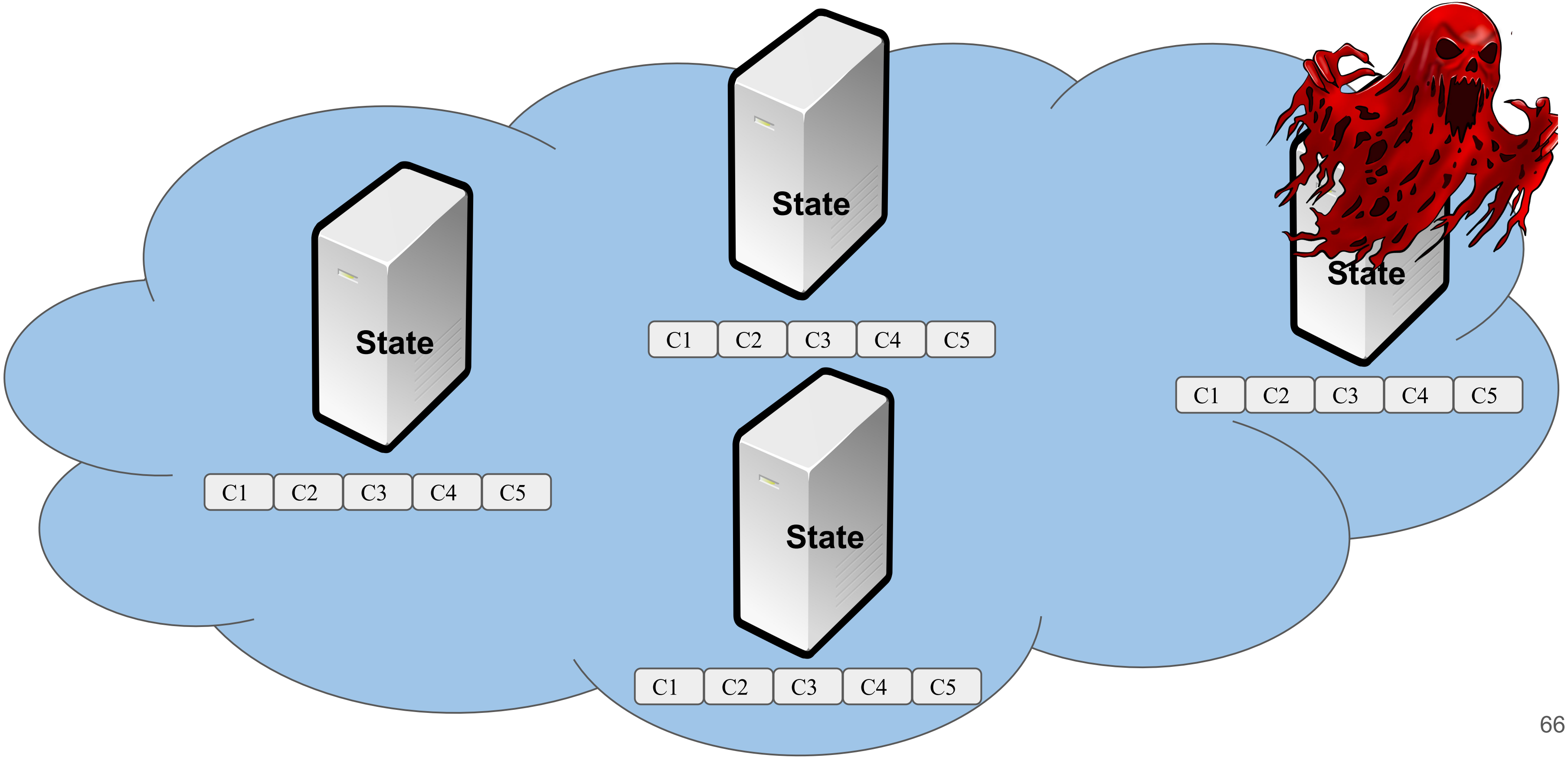
Optimal Performance under synchrony and asynchrony

Enables Hedging

Mahi-Mahi - Low latency DAG based BFT

Pasindu Tennage, Philipp Jovanovic, Lefteris Kokoris Kogias, Bryan Kumara, Alberto Sonnino , Igor Zablotchi

BFT Consensus



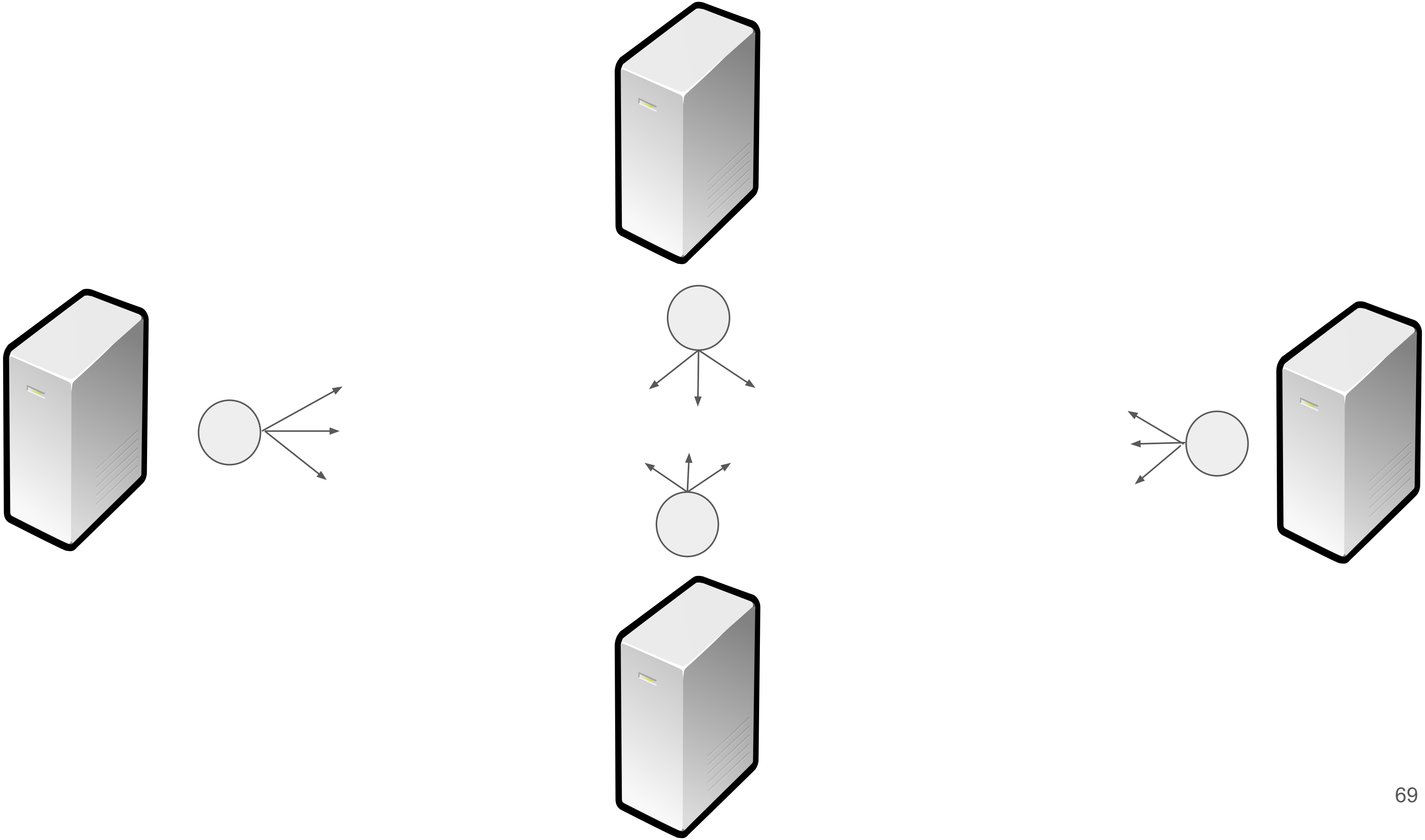
Mahi Mahi outline

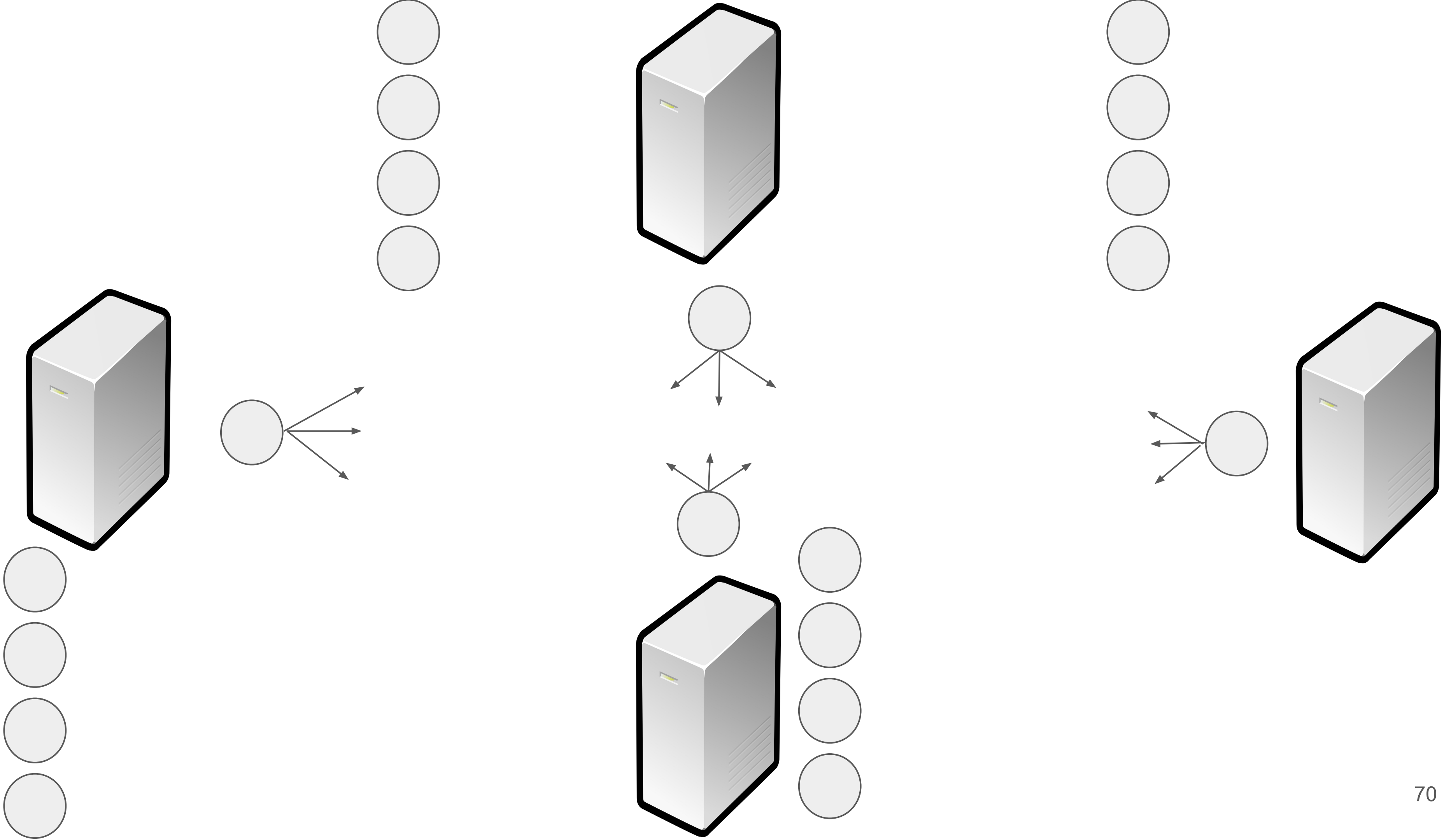
- Distributed Acyclic Graph (DAG) overview.
- Limitations of DAG protocols.
- Mahi-Mahi design.
- Evaluation.

Why DAGs?

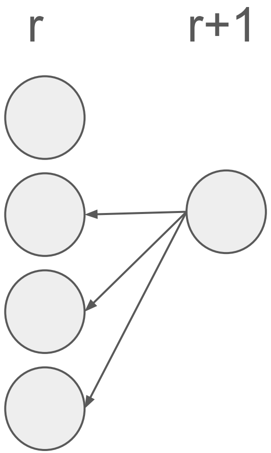
Single message type

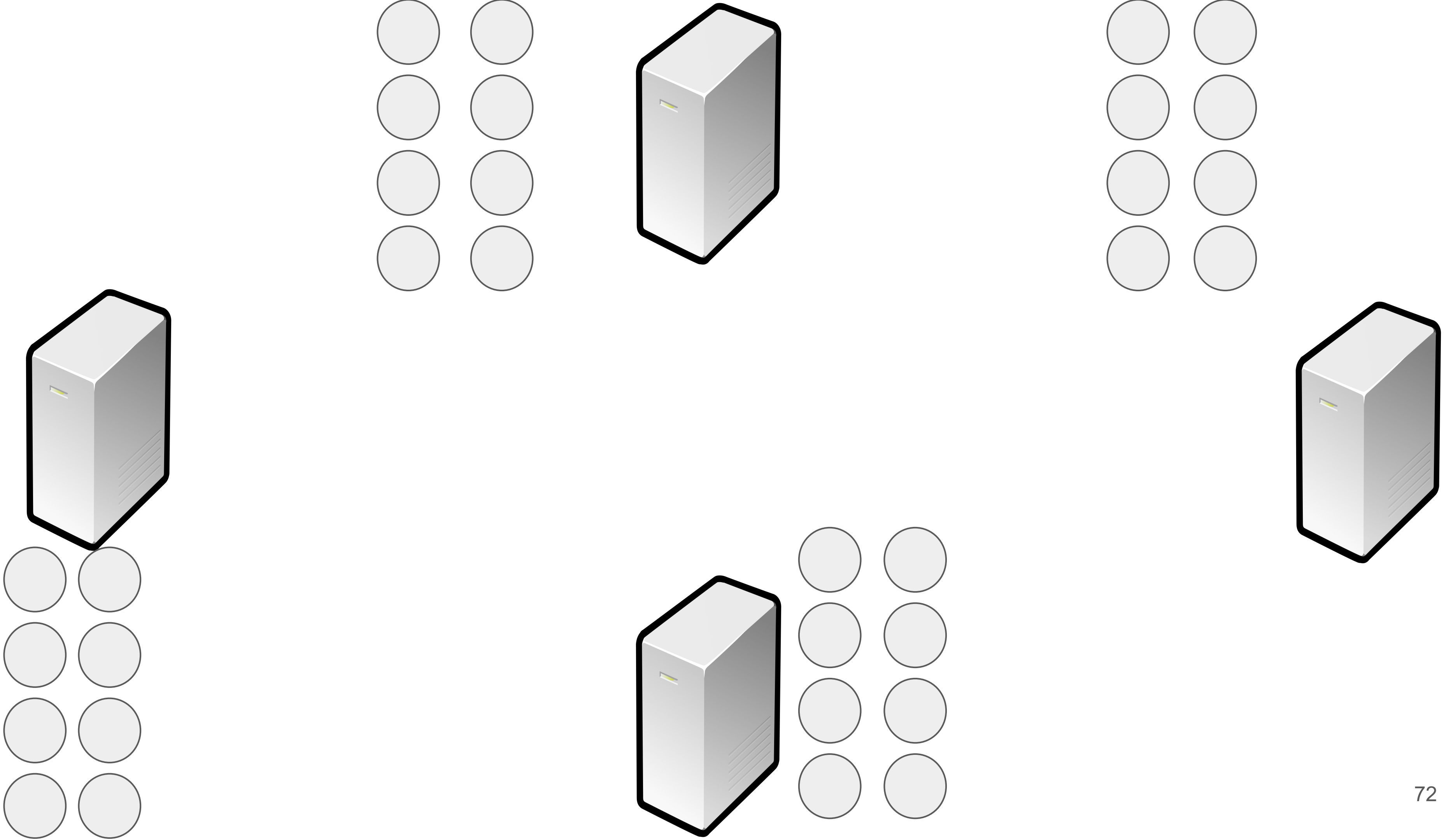
Load balancing



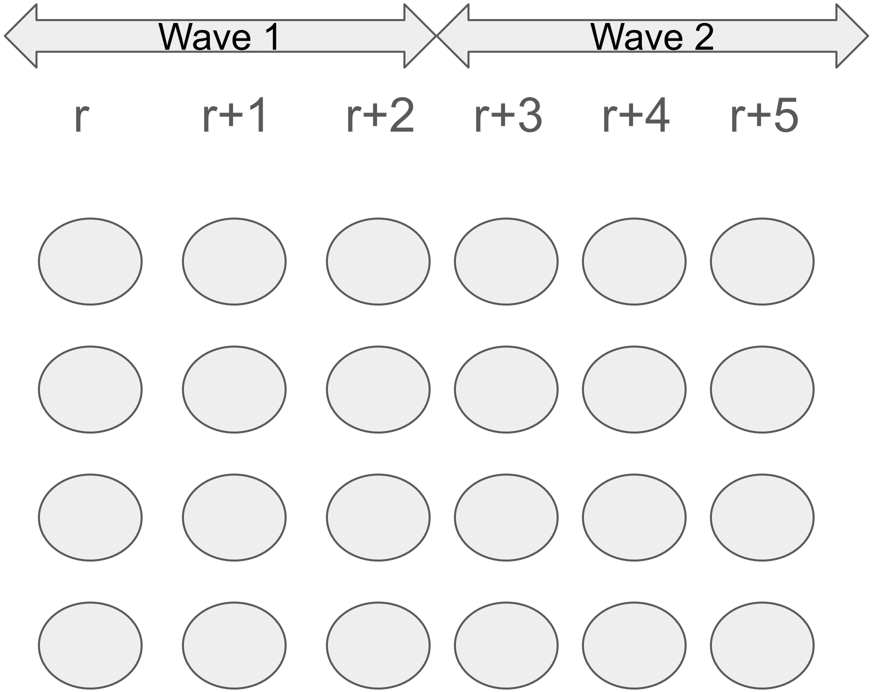


2f+1 Hash Links

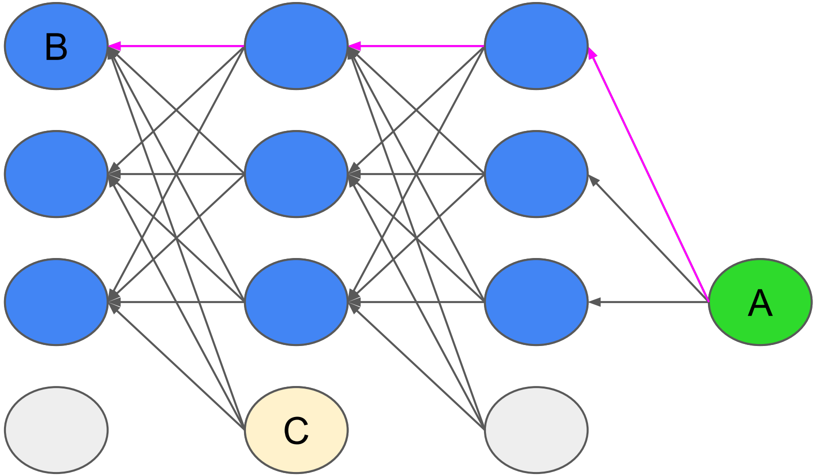




Local View of the DAG

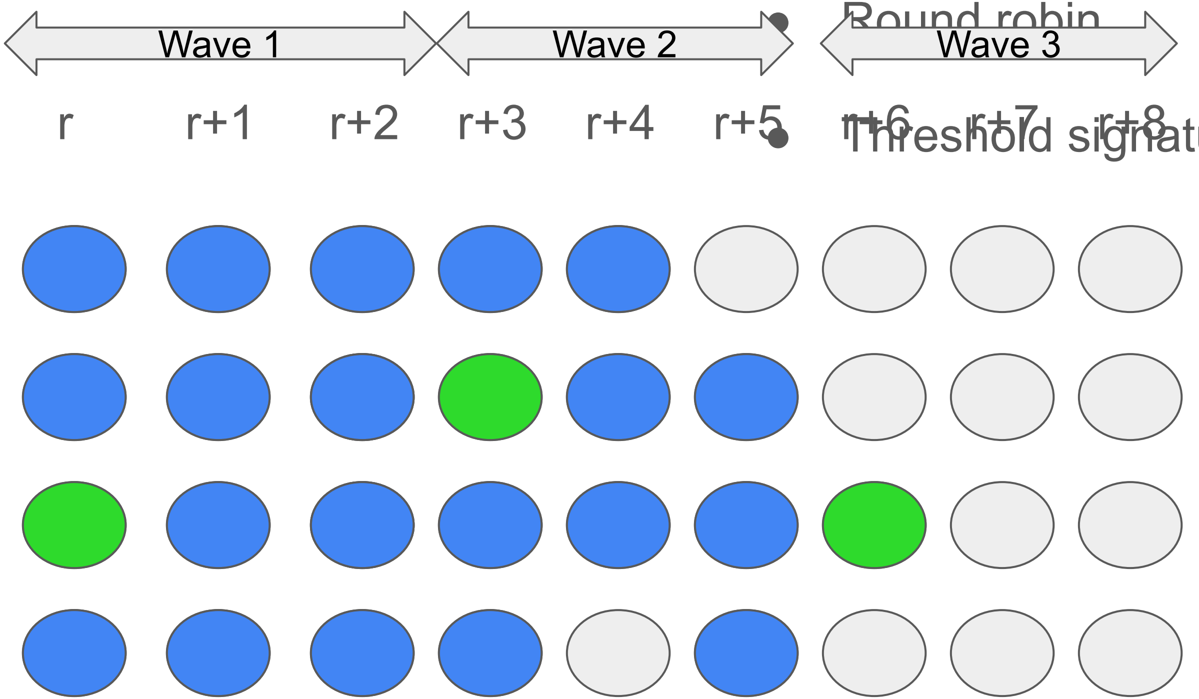


Casual history in Local DAG View



Consensus in DAG based BFT

- Round robin
- Threshold signature based.



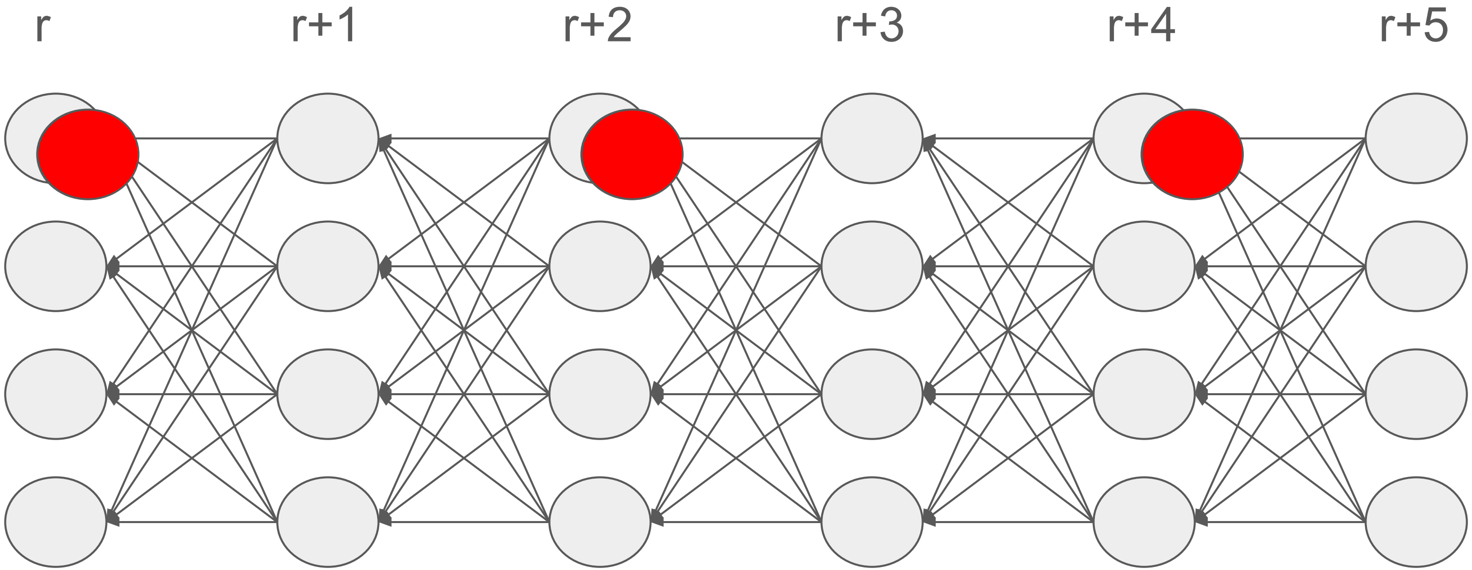
Problems with existing DAG based protocols

Cost of certification

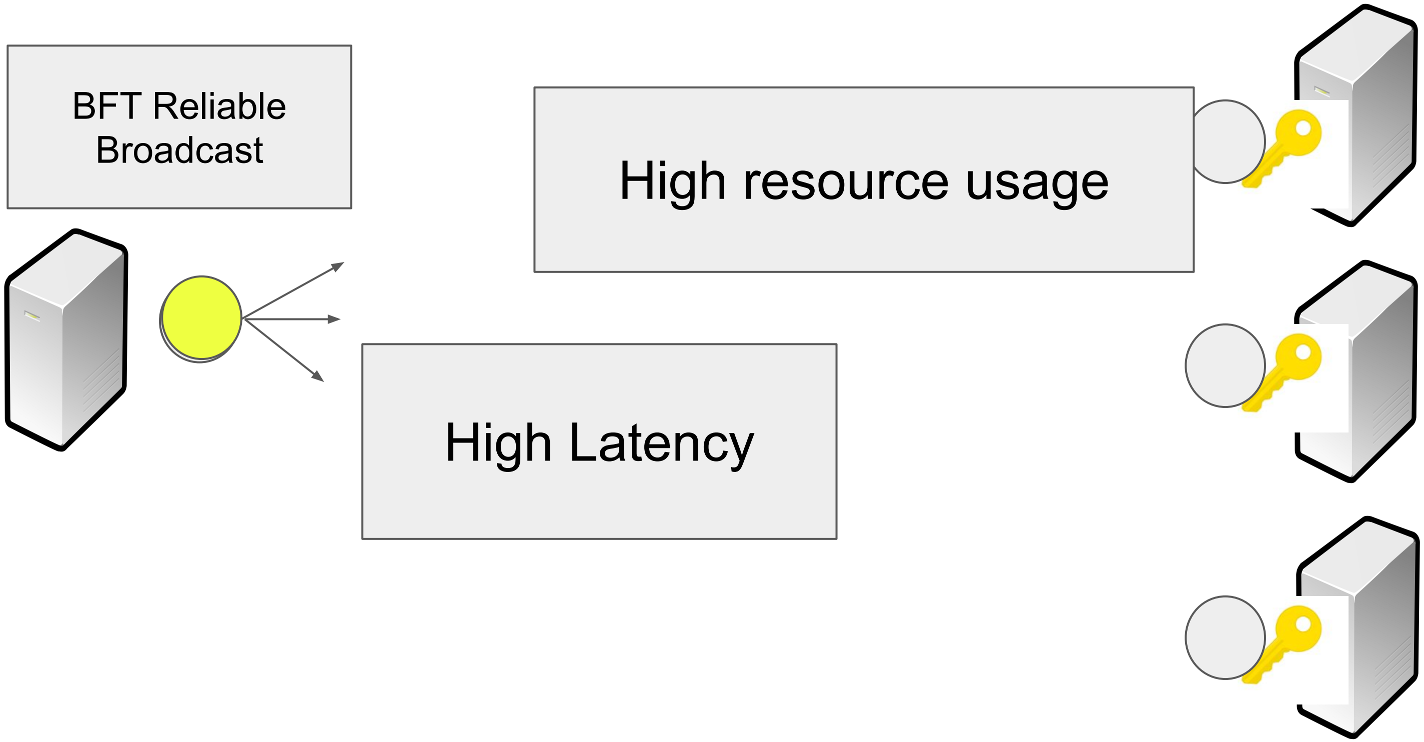
High commit delay due to wave by wave design

High commit delay due to crashed validators

Equivocation in DAG based Consensus



Handling equivocation using certificates



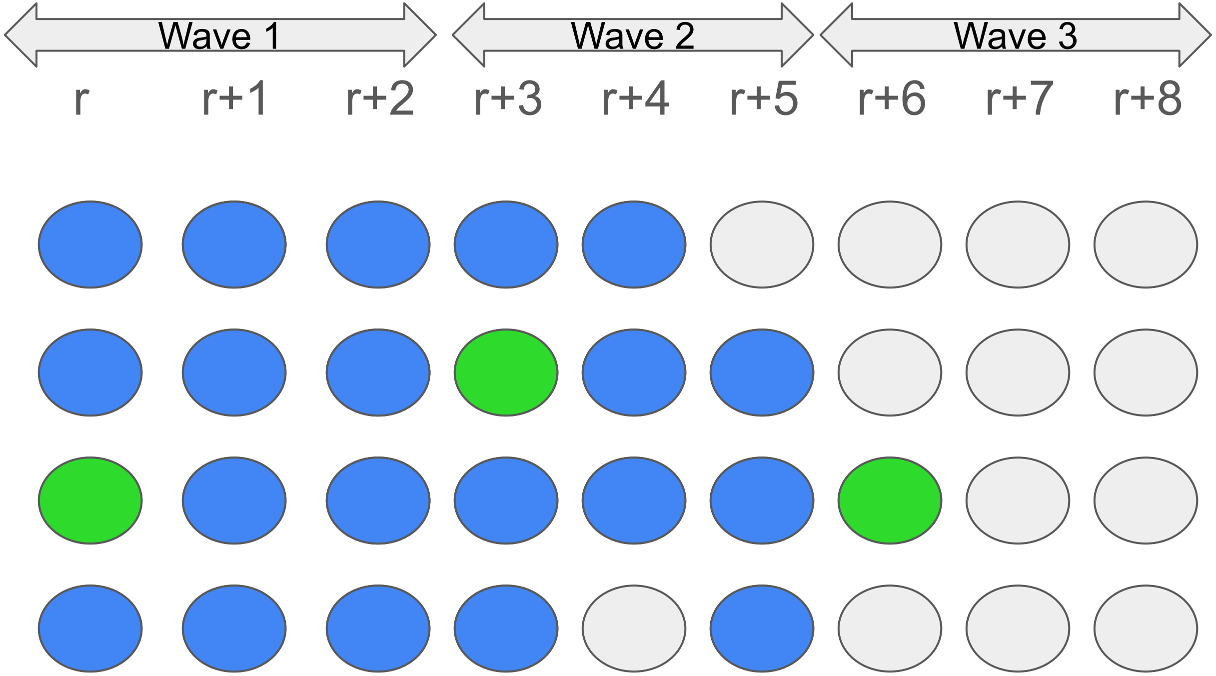
Problems with existing DAG based protocols

Cost of certification

High commit delay due to wave by wave design

High commit delay due to crashed validators

Relative distance to the next committed leader



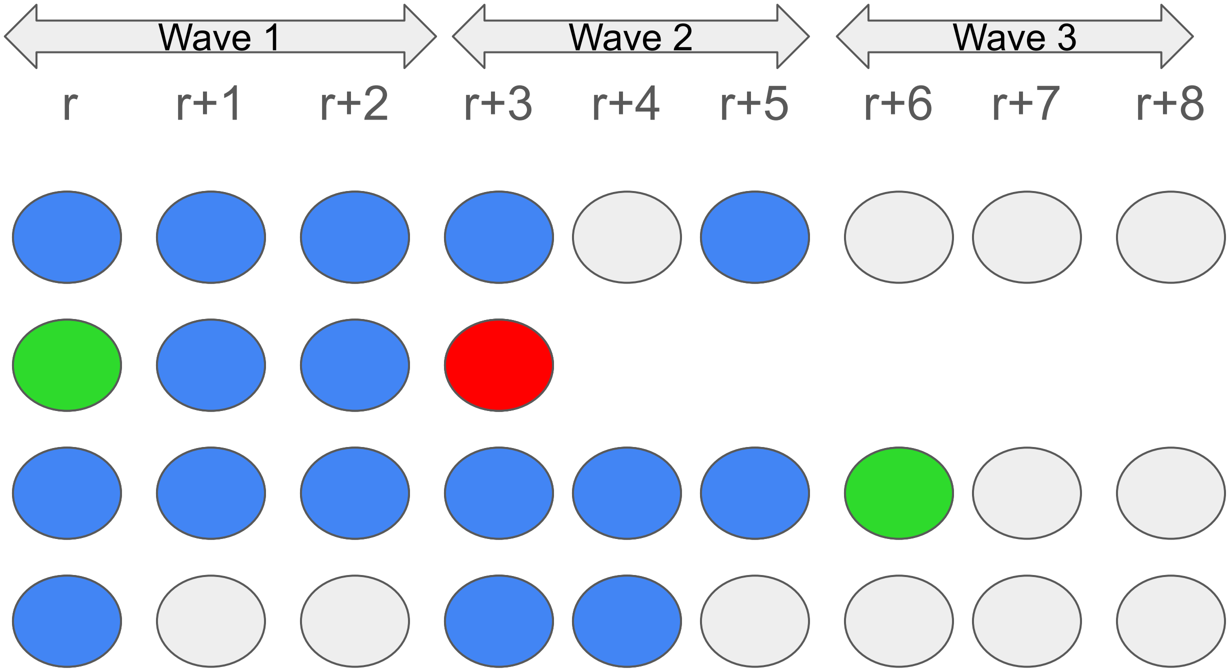
Problems with existing DAG based protocols

Cost of certification

High commit delay due to wave by wave design

High commit delay due to crashed validators

Crash failures increase commit latency



Mahi-Mahi

Reduce resource consumption	Reduce commit latency	Have minimal impact from the crashed validators

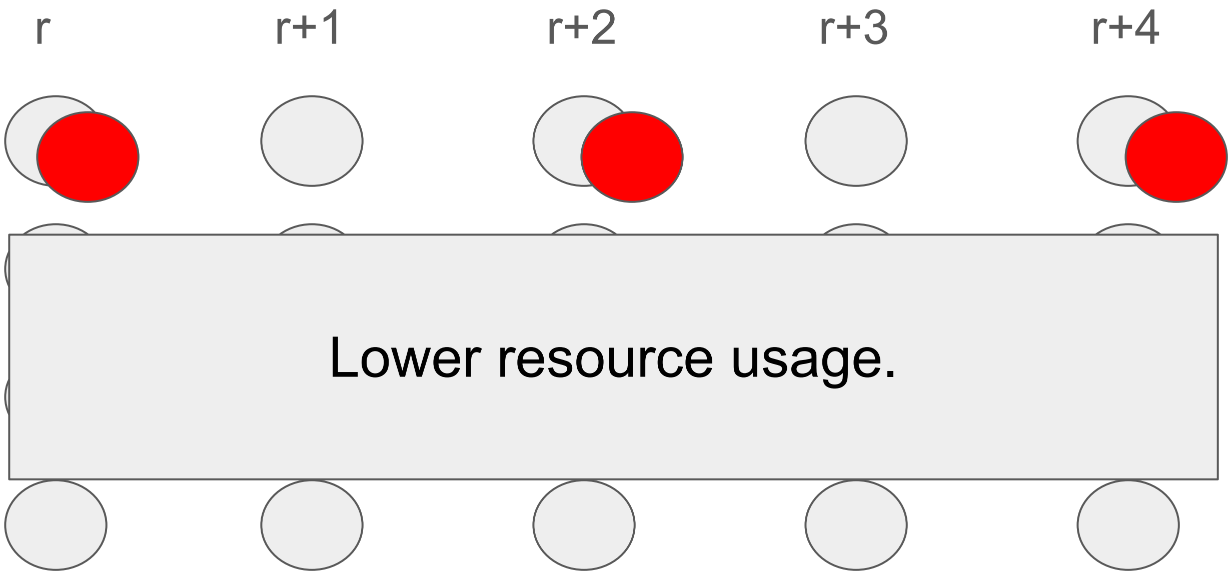
Threat Model

- Up to ***f*** out of ***3f+1*** nodes are **malicious**.

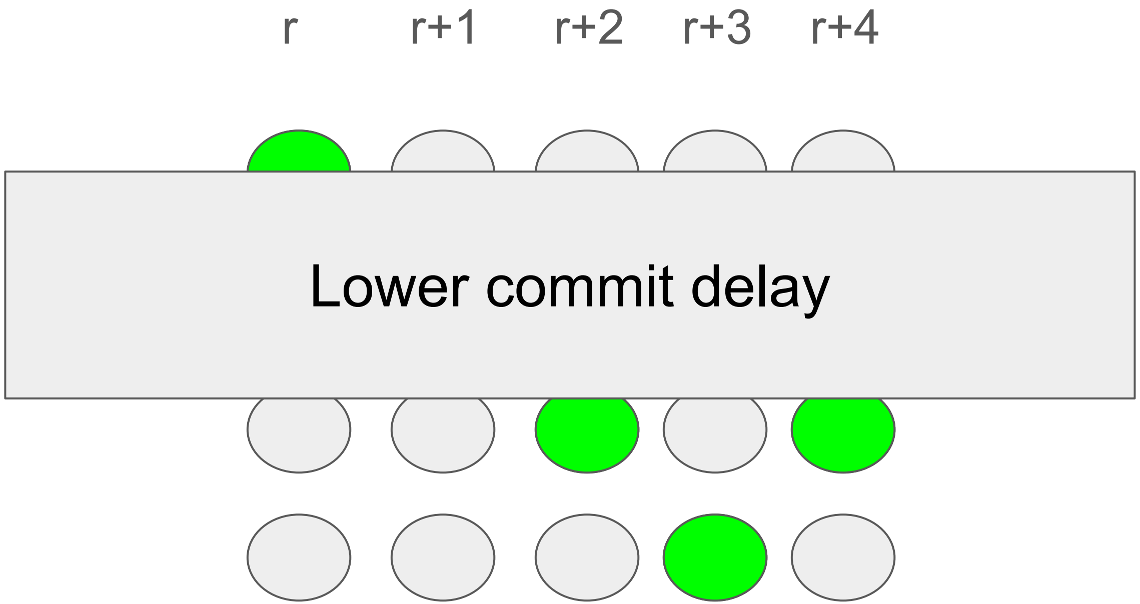
- The network is **asynchronous** – there exists **no bound Δ** on message transmission delay.

- Network attacker
  - Can delay and reorder messages.
  - Cannot intercept messages from honest nodes.

Mahi-Mahi uncertified DAG Wave

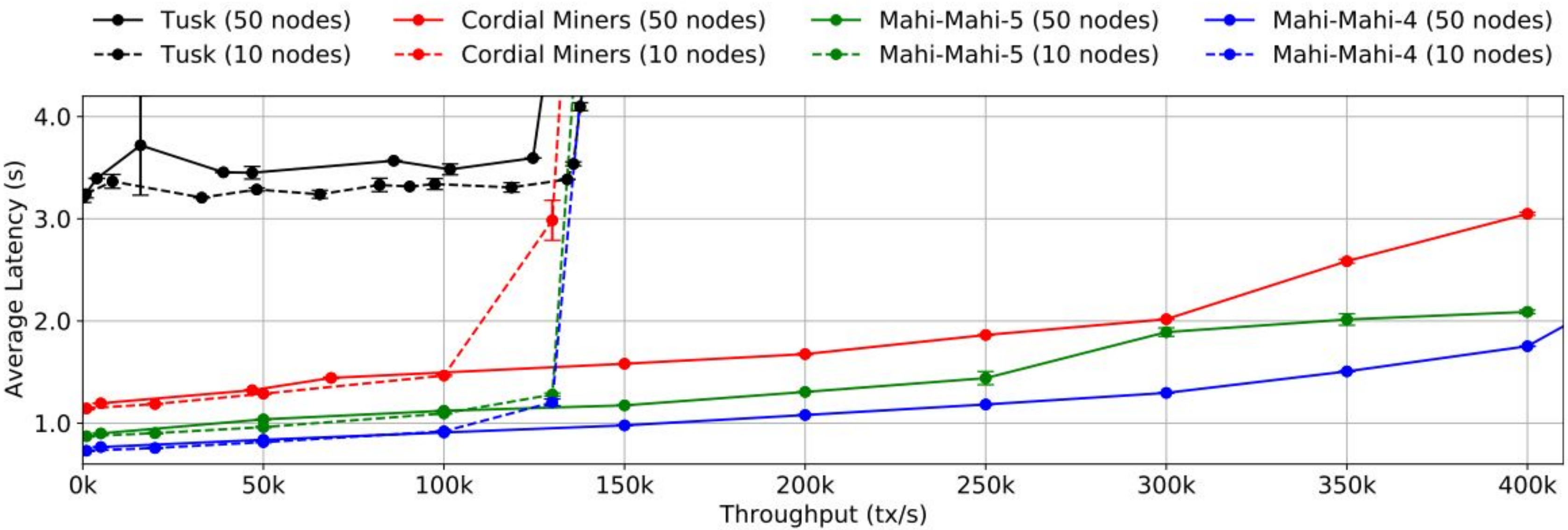


Leader blocks in each round



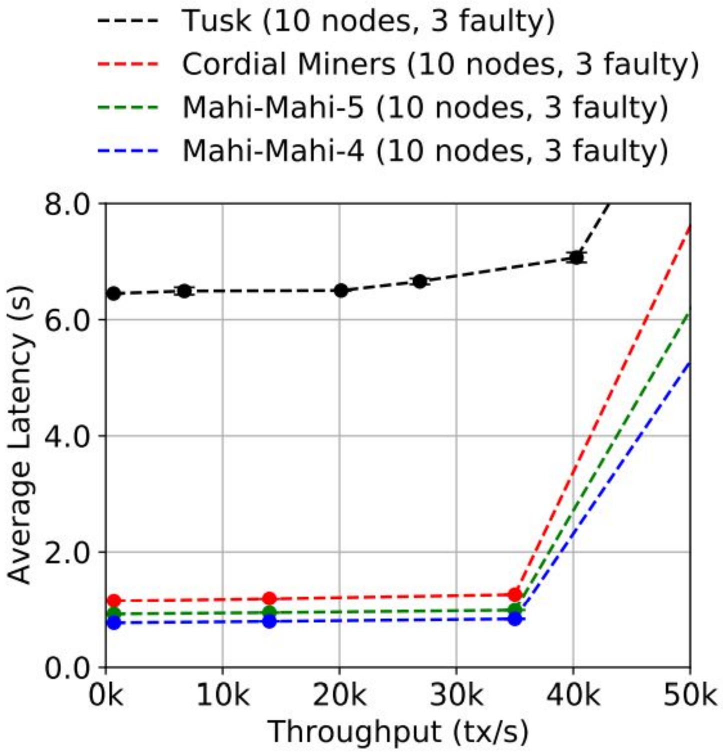
Evaluation

Normal Case Performance



Mahi-Mahi achieves higher throughput with lower latency

Performance under crash faults



Mahi-Mahi has minimal impact from crashed validators

Mahi-Mahi Summary

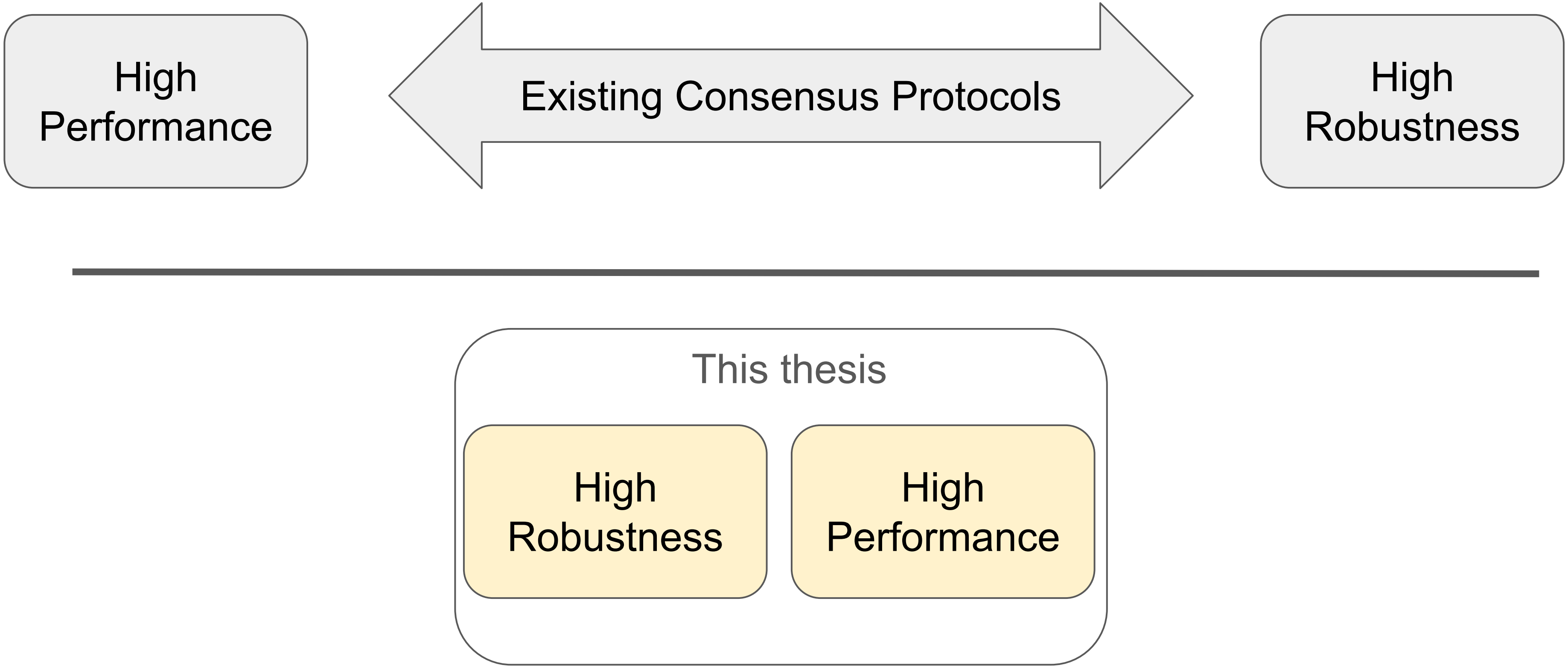
Reduce resource consumption

Reduce commit latency

Have minimal impact from the crashed validators

Summary of Thesis Contributions

Baxos	REB as a replacement for leader election in Multi-Paxos to achieve high robustness
RACS-SADL	Avoid leader bottleneck and asynchronous liveness
QuePaxa	Optimum performance under synchronous and asynchronous networks, support hedging
Mahi-Mahi	Low commit delay with low resource utilization for DAG based BFT


High Performance
Existing Consensus Protocols
High Robustness
This thesis
High Robustness
High Performance

Future Directions

- Measuring adversarial performance.
- Merging SADL and RACS to reduce latency.
- Tuning consensus for high performance.

Summary of Thesis Contributions

Baxos	REB as a replacement for leader election in Multi-Paxos to achieve high robustness
RACS-SADL	Avoid leader bottleneck and asynchronous liveness
QuePaxa	Optimum performance under synchronous and asynchronous networks, support hedging and tuning
Mahi-Mahi	Low commit delay with low resource utilization for DAG based BFT