Speculative Byzantine Fault Tolerance

# ZYZZYVA

By Océan Gillaux

University of Stavanger, MID110, April 2010

# Plan

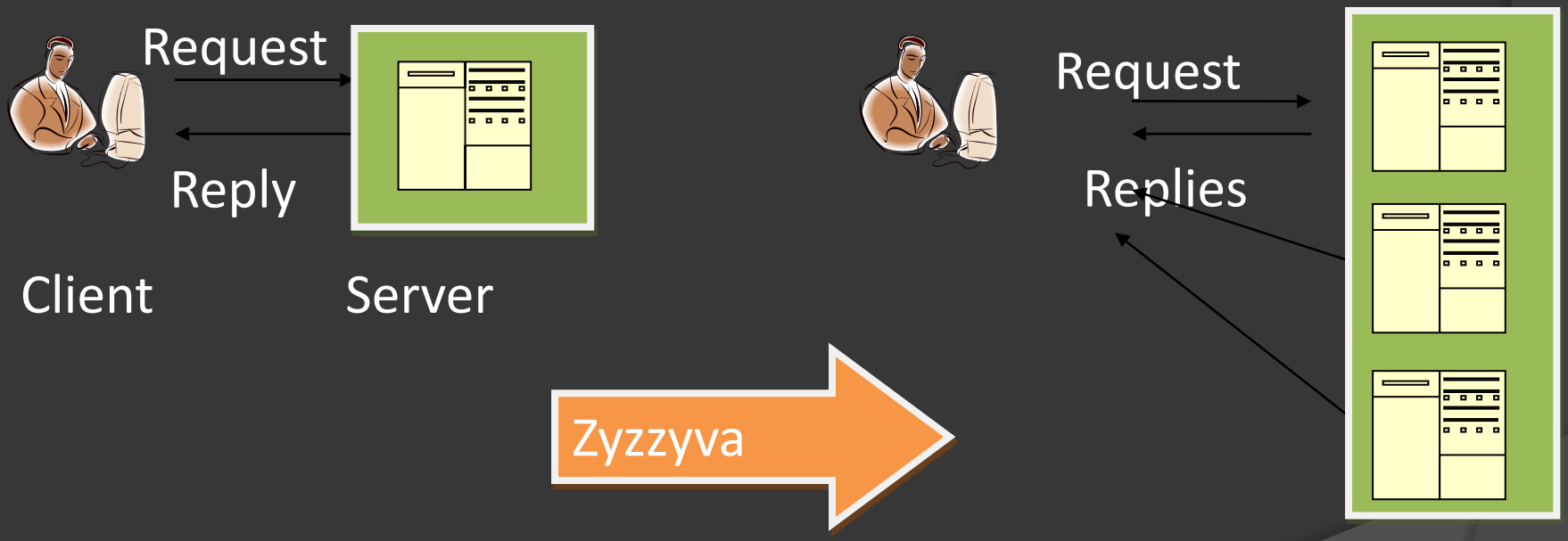
- Zyzzyva: Last word of dictionary
- Requirements & Introduction
- Byzantine problem
- Zyzzyva Protocol
- Evaluation
- Conclusion

# Requirements

- Fault Tolerance ?
  - Servers Problems:
    - Hardware
    - Software
    - Hacking
  - Access 24/7
- Application see centralized services

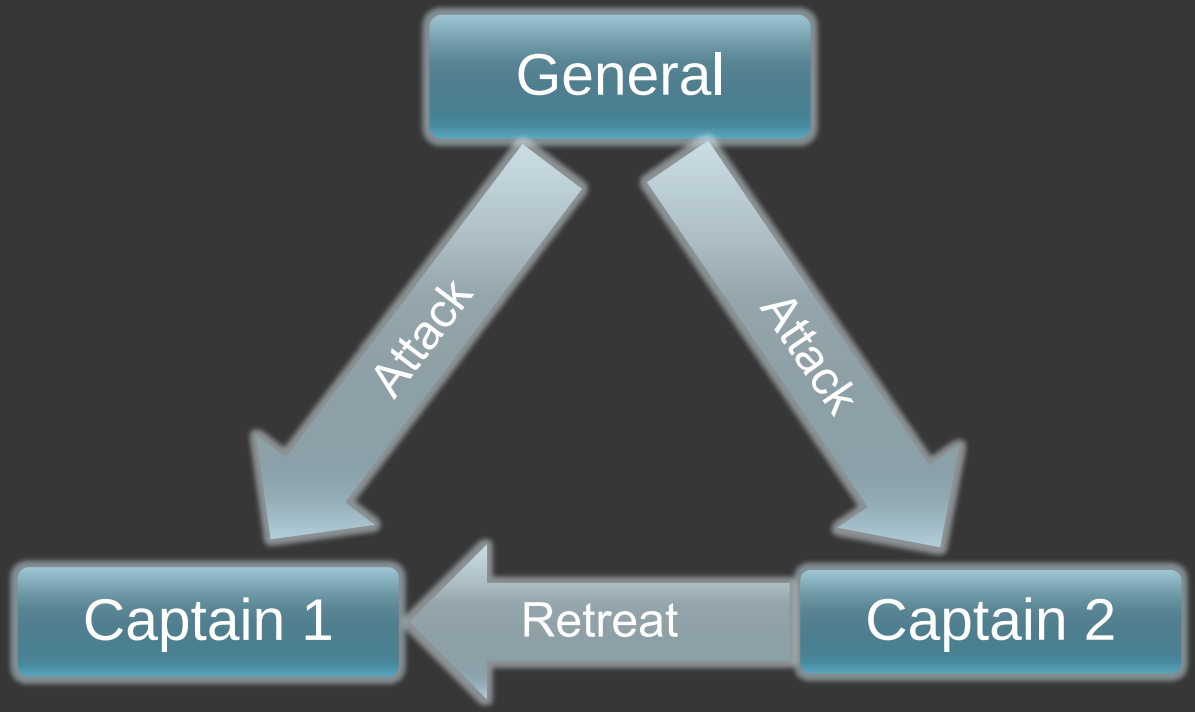
# Solution

- Add Servers




- Problem reliability: Byzantine General's problem

# Byzantine General's problem

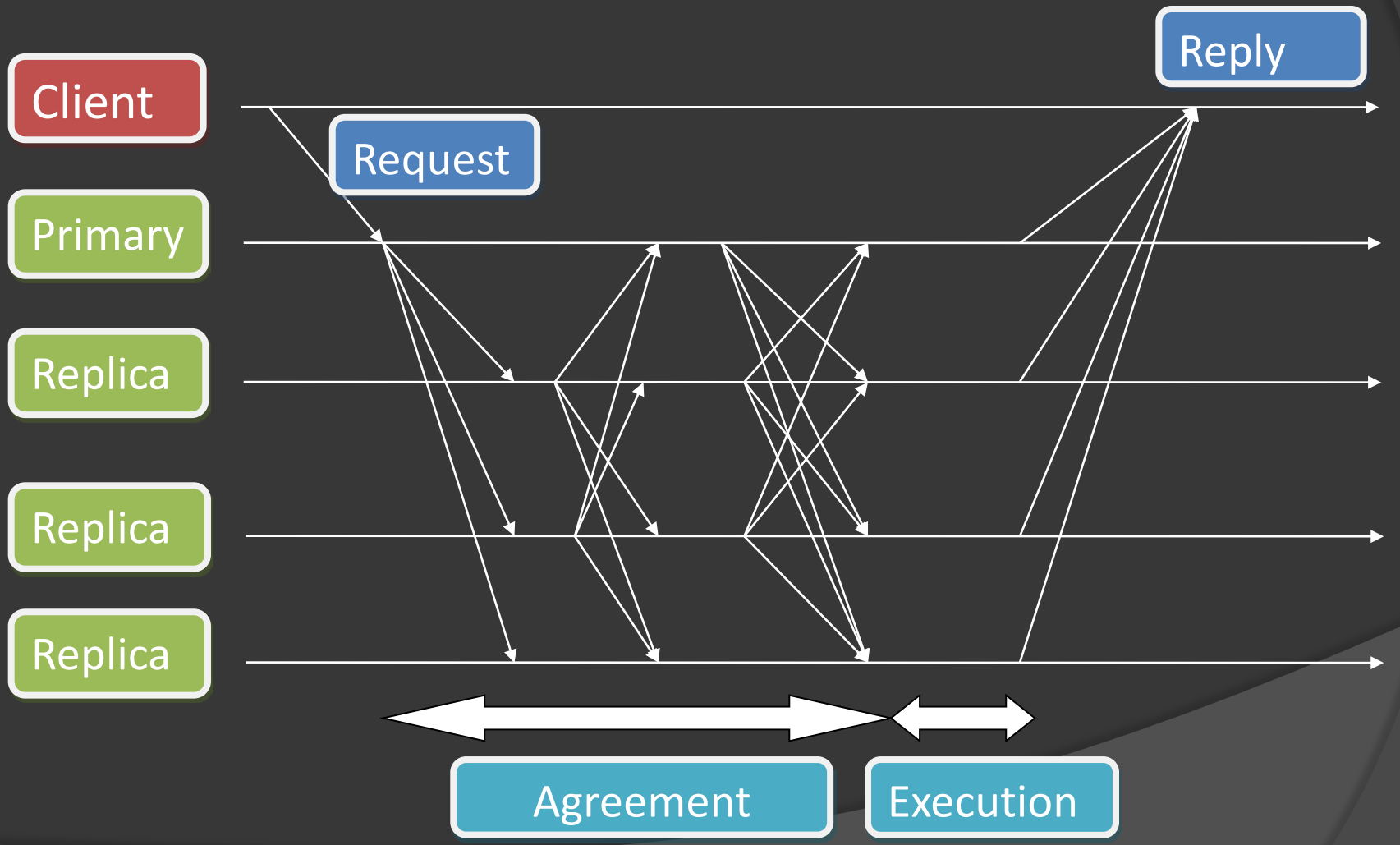



- Captain 2 is a liar
- Minimum 2m+1 loyal for 1 liar

# Security

- We admit that adversary cannot break cryptographic techniques
- Zyzzyva uses the concept of private/public key

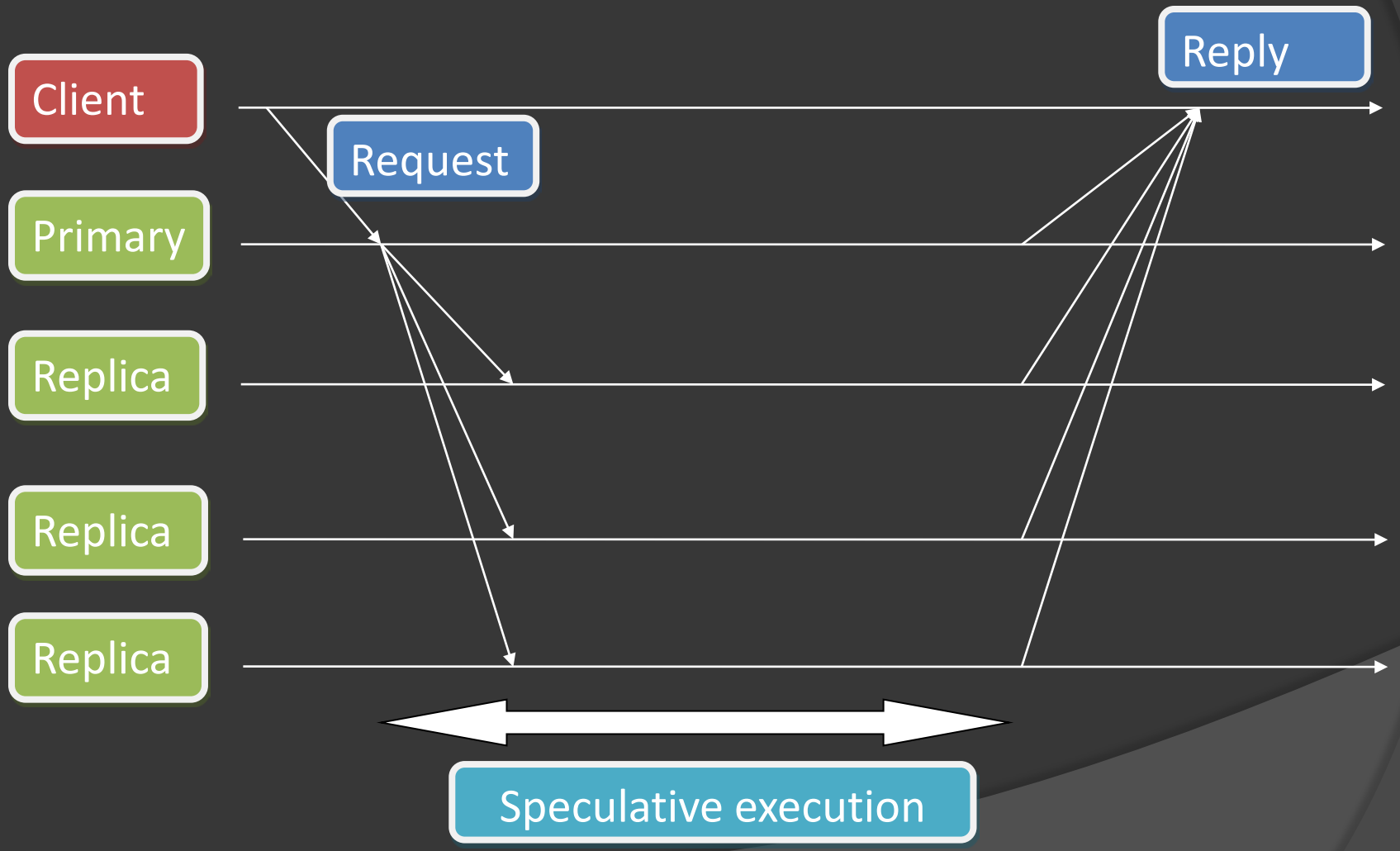
# Introduction: Byzantine Fault Tolerance



# Introduction: Byzantine Fault Tolerance

- Long phase of agreement
- Cost important
- Many messages

# Introduction: Zyzzyva



# Introduction: Zyzzyva

- Replica make **speculation** to send the response:
  - It is faster
- The client **verifies** if the reply is stable

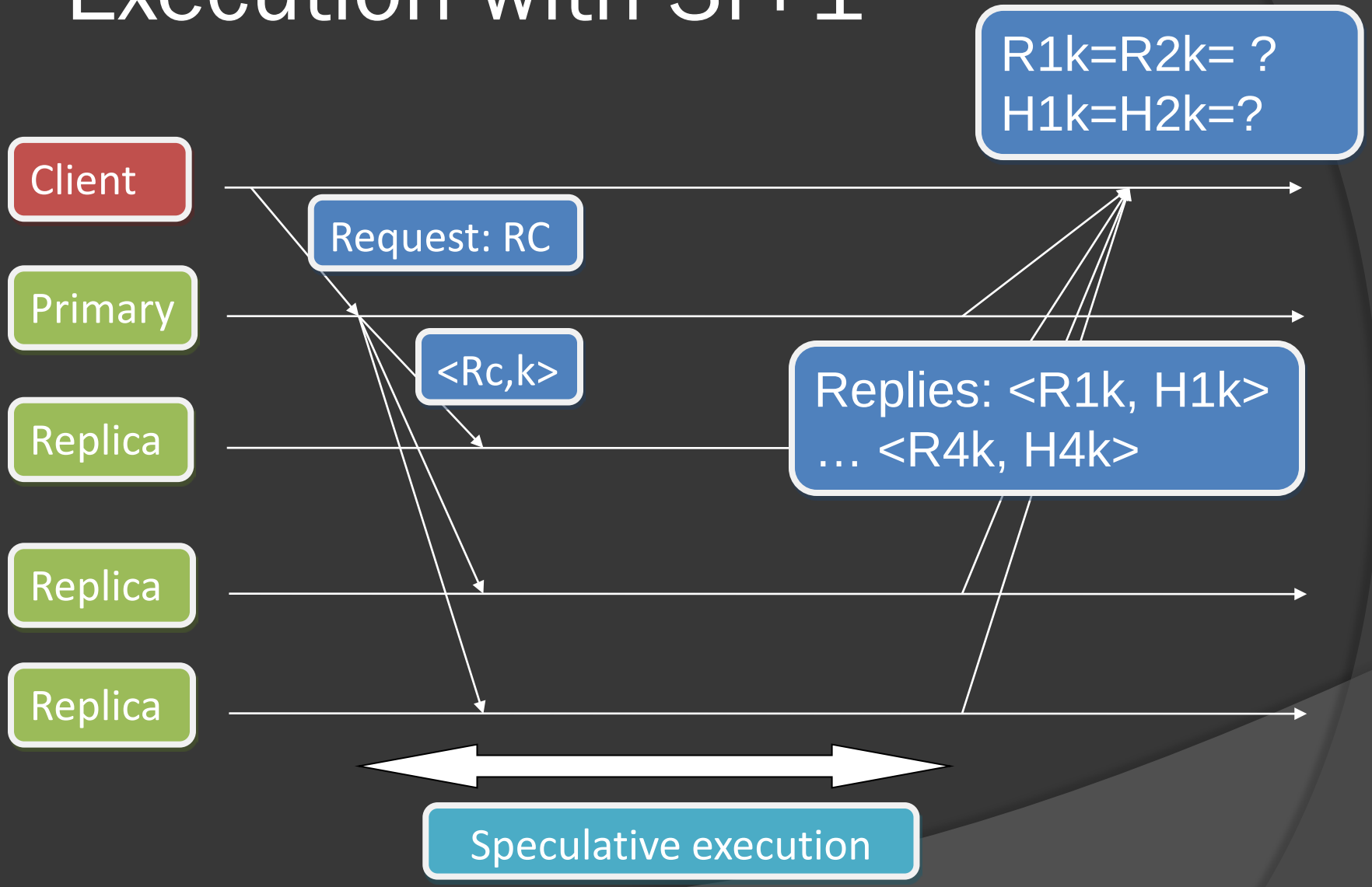
# Zyzzyva Protocol

- 3 sub-protocols
  - Agreement protocol
  - View-change protocol
  - Checkpoint protocol

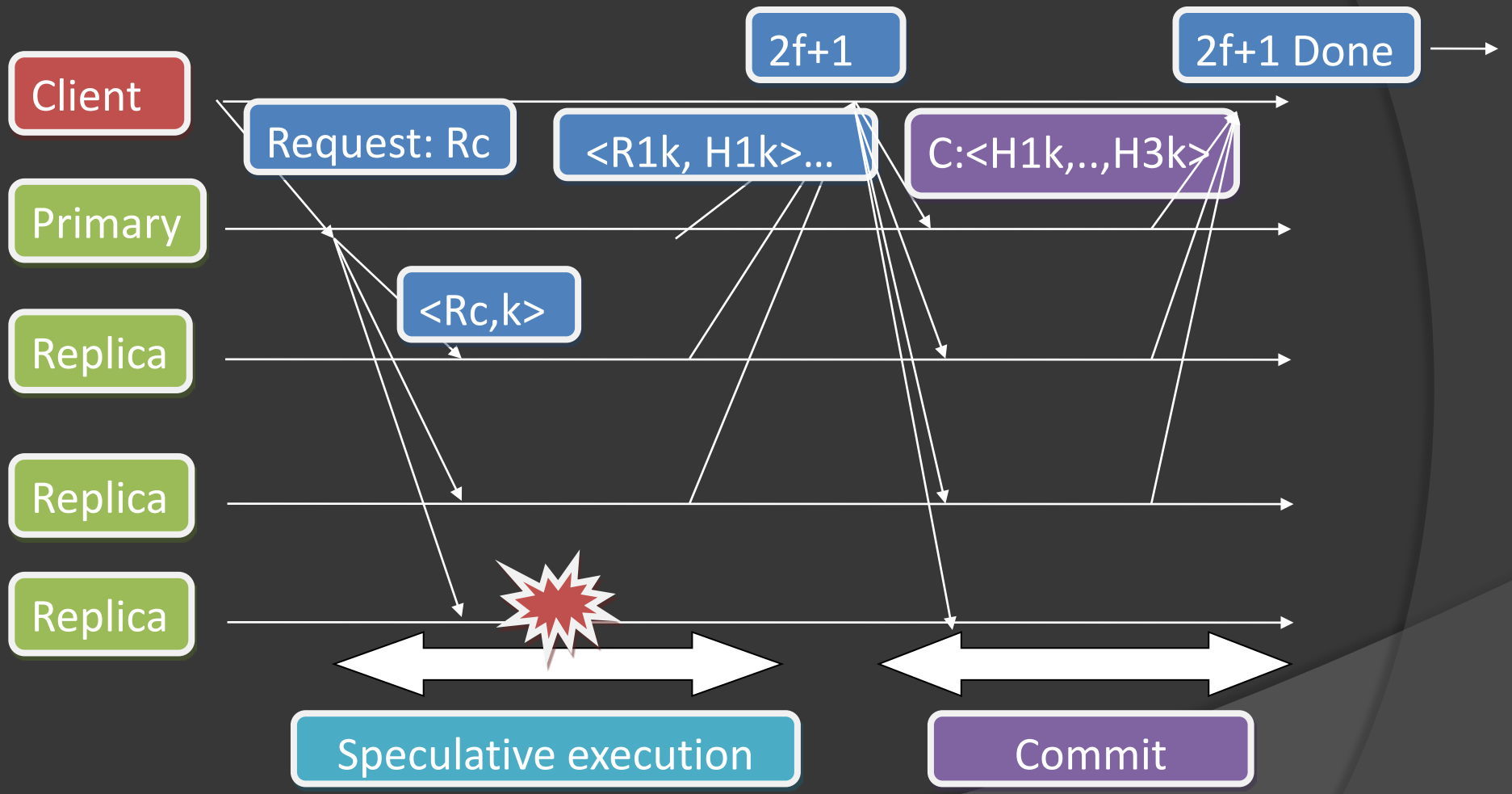
# Agreement Protocol

- How the client check stable reply?
  - History included in the message
  - Matching responses

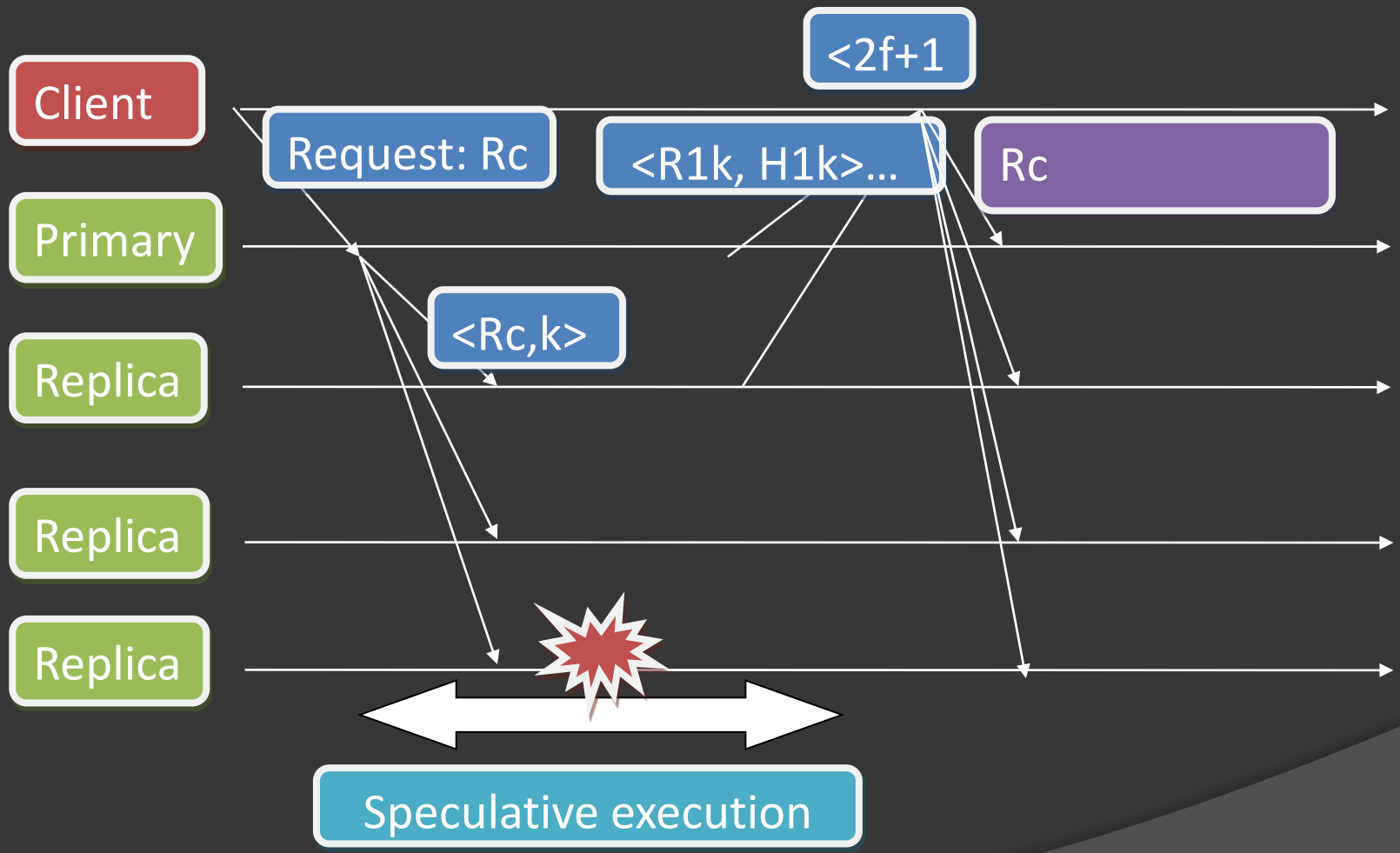
# Execution with 3f+1



# One faulty: 2f+1 replies



# Less 2f+1 responses



# Checkpoint Protocol

- History is important
  - Manage the history
  - Replica maintains only 1 checkpoint
  - Only last information could be necessary

# View Change

- Election new Primary AND guarantees the history
- Concept “I hate the primary”
  - Replica can make a mutiny
  - View-change message

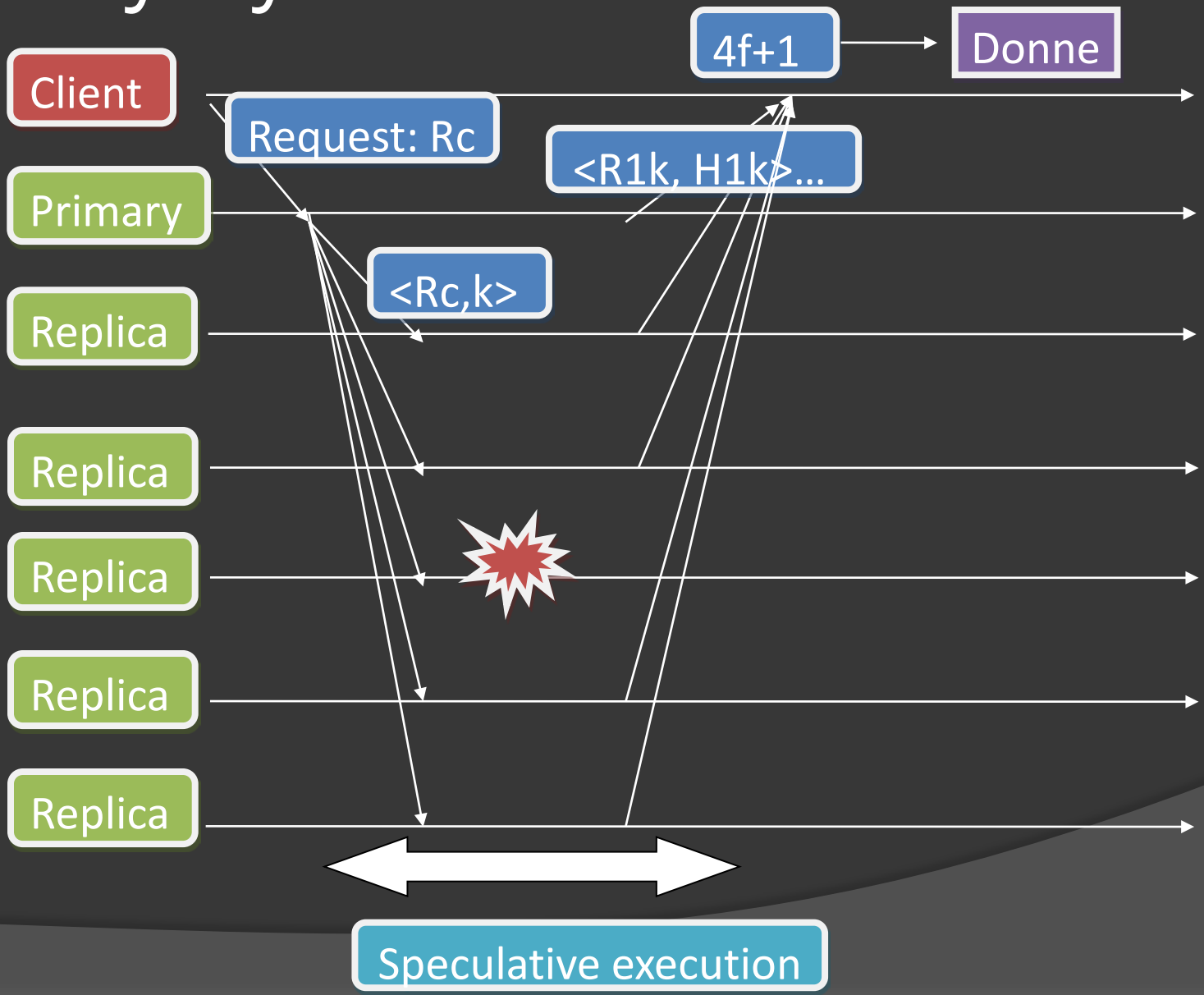
# Client

- Important Roles in Zyzzyva
  - Can a faulty client block zyzzyva?
    - Not commit message
    - Only affect own process
  - Can a faulty client compromised zyzzyva?
    - Commit bad history
    - Security encryption

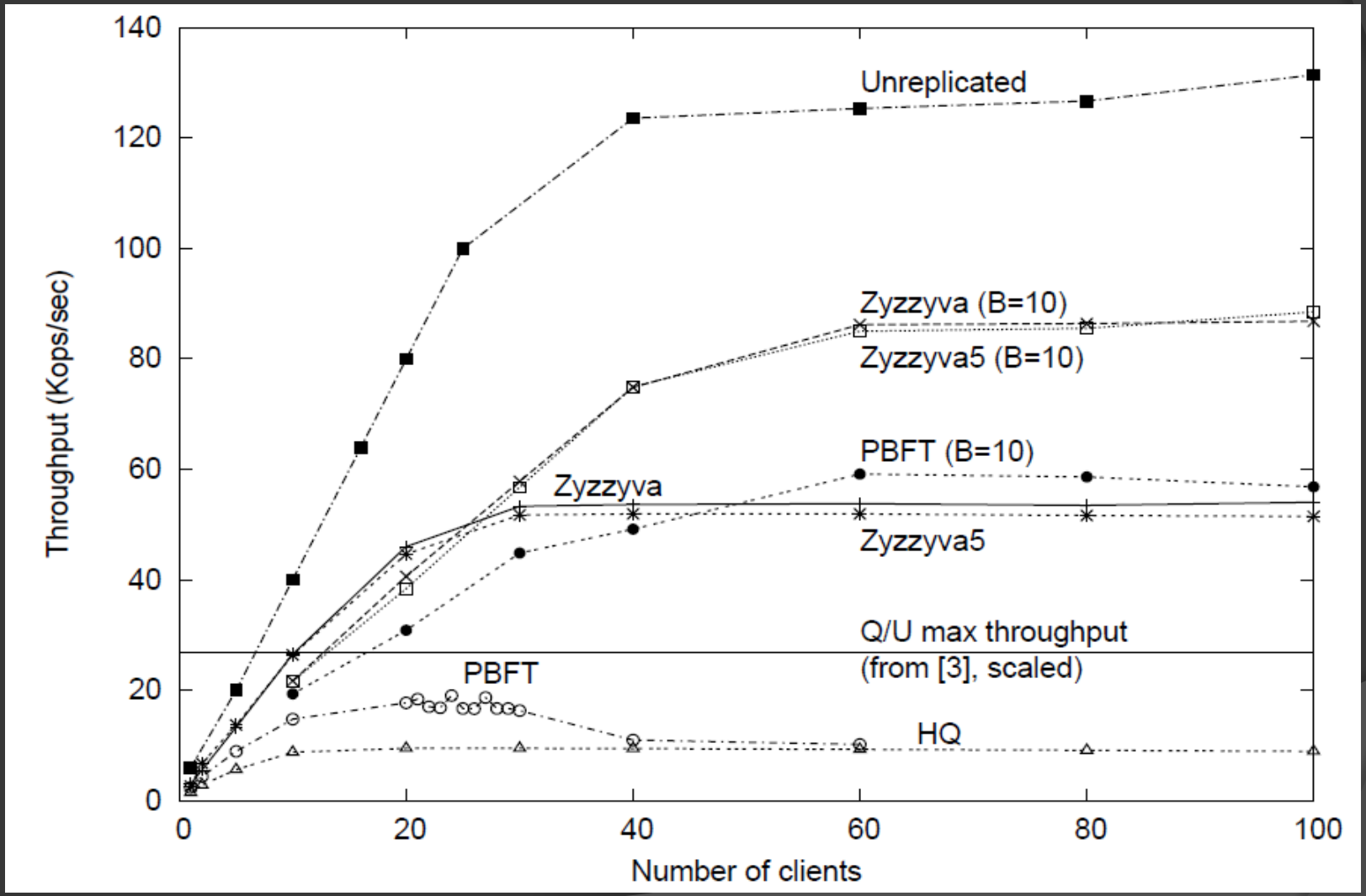
# Optimization

- Replacing signatures with MACs
- Separating agreement from execution
- Request Batching
- Zyzzyva5

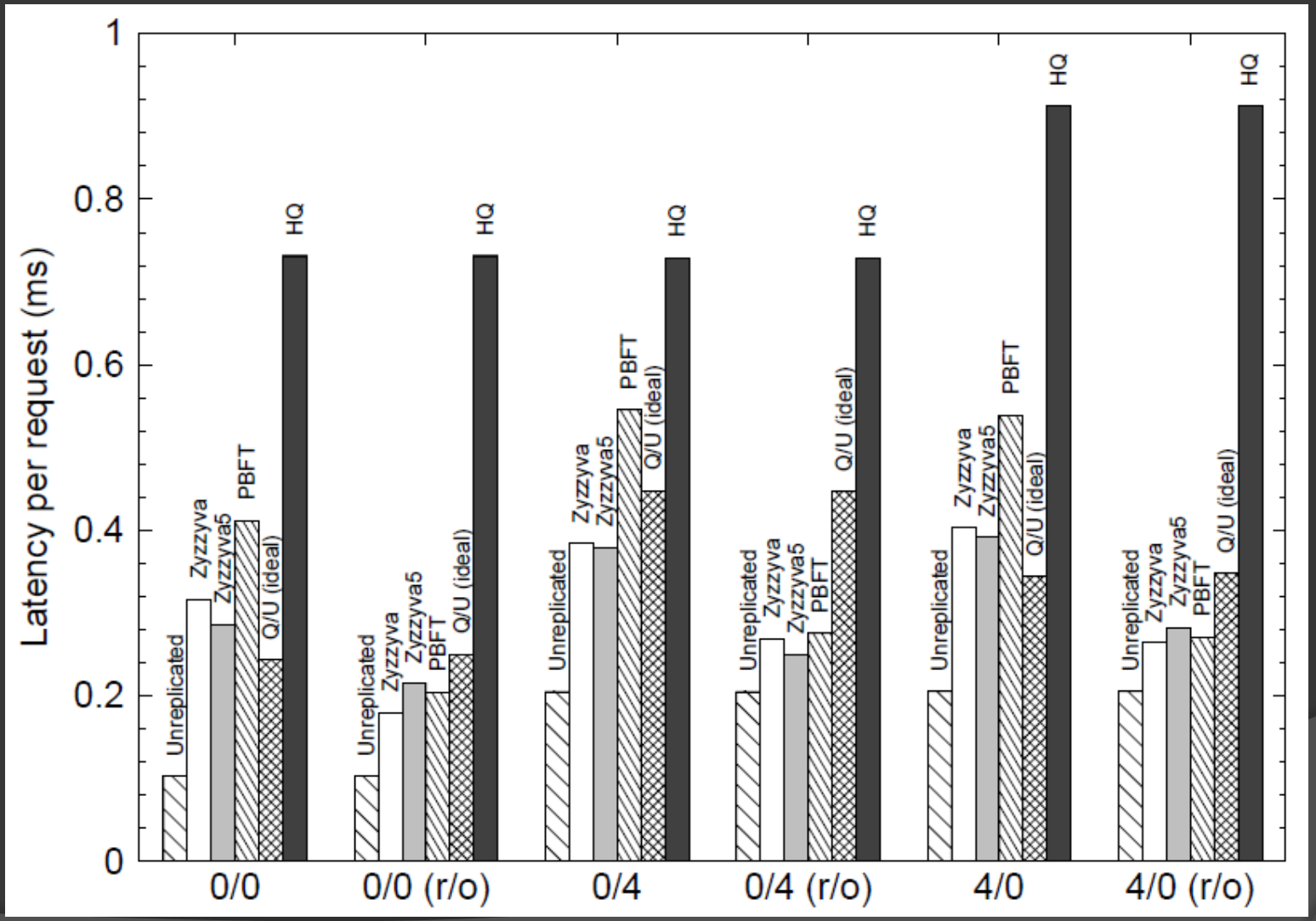
# Zyzzyva5: 5f+1



# Evaluation


Throughput (Kops/sec)
Number of clients
0
20
40
60
80
100
120
140
Unreplicated
Zyzzyva (B=10)
Zyzzyva5 (B=10)
PBFT (B=10)
Zyzzyva
Zyzzyva5
Q/U max throughput
(from [3], scaled)
PBFT
HQ

# Evaluation


Latency per request (ms)
0
0.2
0.4
0.6
0.8
1
Unreplicated
Zyzzyva
Zyzzyva5
PBFT
Q/U (ideal)
HQ
0/0
0/0 (r/o)
0/4
0/4 (r/o)
4/0
4/0 (r/o)

# Conclusion

- In exploiting speculation, Zyzzyva has a good performance over existing BFT services. Zyzzyva approaches the theoretical lower bounds for any BFT.

Thank you

Questions ?