ACSAC 2022

Trebiz: Byzantine Fault Tolerance with Byzantine Merchants

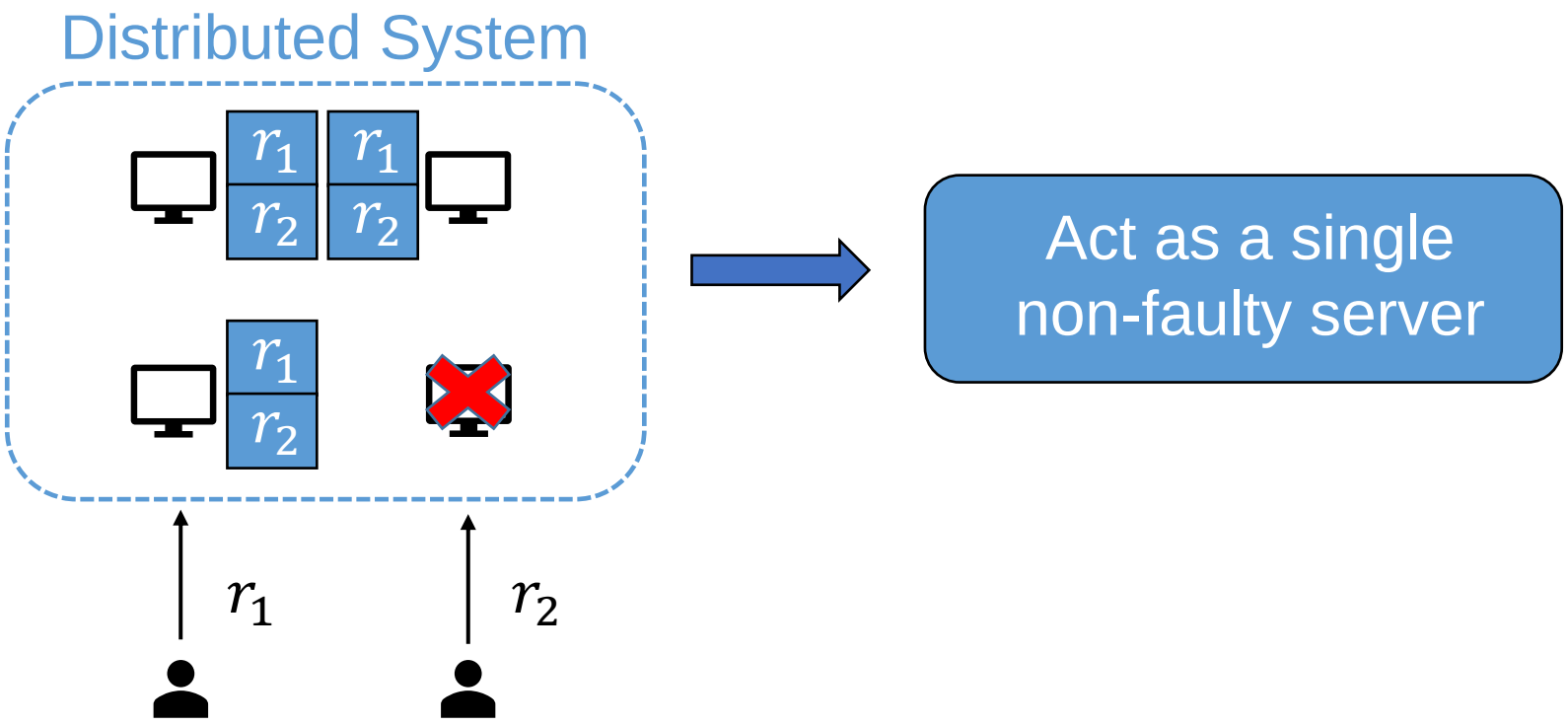
Xiaohai Dai*, Liping Huang*, Jiang Xiao*, Zhaonan Zhang*, Xia Xie+, and Hai Jin*

***National Engineering Research Center for Big Data Technology and System**
***Services Computing Technology and System Lab, Cluster and Grid Computing Lab**
***School of Computer Science and Technology, Huazhong University of Science and Technology, China**
+School of Computer Science and Technology, Hainan University, China

Outline

Background

- **Consensus**
  - All correct replicas can reach an agreement on requests even there are some faulty replicas

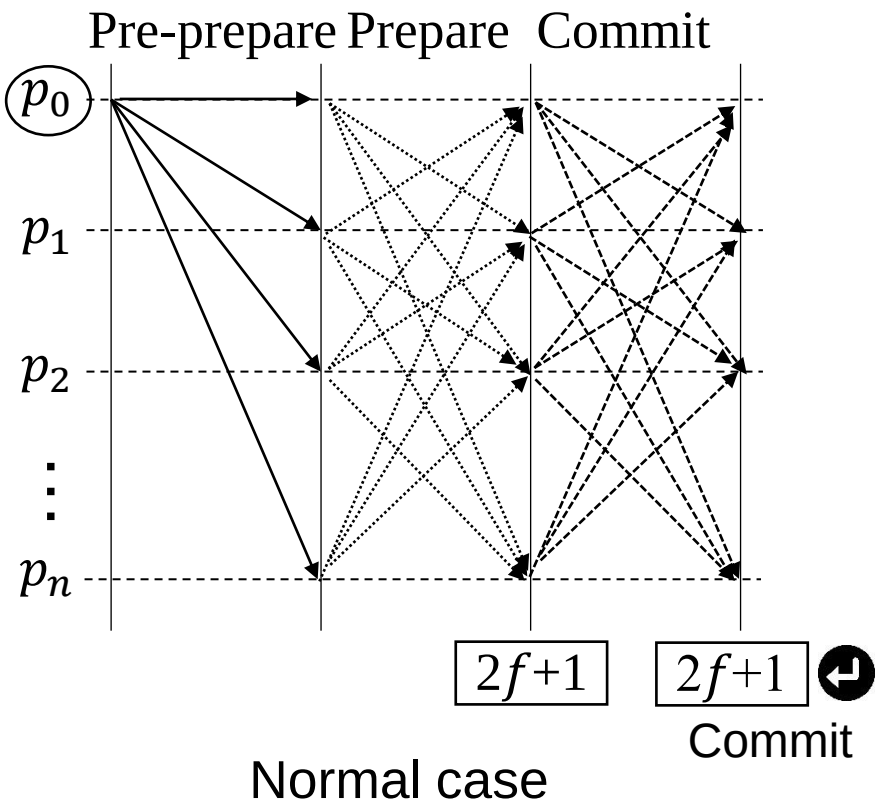


Background

- **BFT (Byzantine Fault Tolerant) consensus**
  - **Fault model:**
    - $N = 3f + 1$ replicas
    - Up to f replicas can be Byzantine faulty (behave arbitrarily)
  - **Safety:** Any two correct replicas should commit the same request at the same index
  - **Liveness:** A request proposed by a correct client will eventually be committed by all the correct replicas
- **PBFT (Practical Byzantine Fault Tolerance)**
  - Widely adopted by BFT consensus (e.g., Hyperledger Fabric, ELASTICO, and ByzCoin)

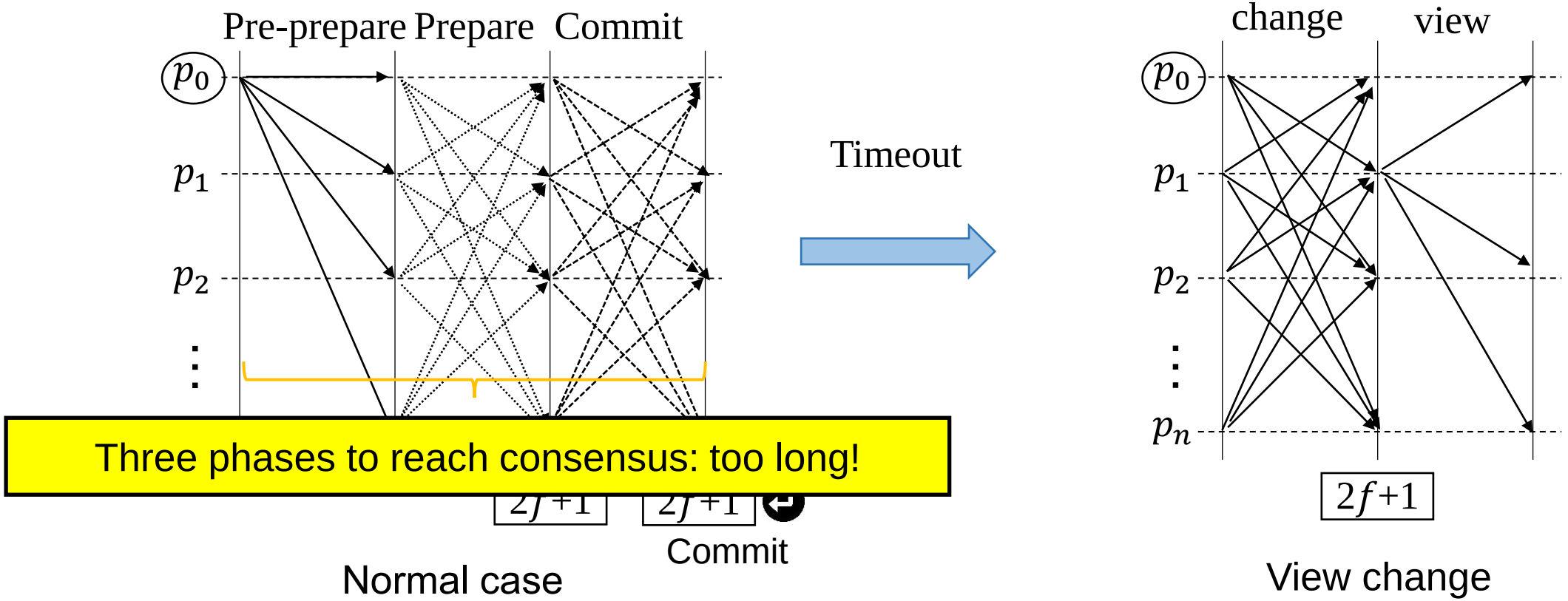
Background

- **PBFT (Practical Byzantine Fault Tolerance)**
  - $N = 3f + 1$, and replicas are denoted as $p_0, p_1, \ldots, p_n$
  - Work in successive views



Background

- **PBFT (Practical Byzantine Fault Tolerance)**
  - $N = 3f + 1$, and replicas are denoted as $p_0, p_1, \ldots, p_n$
  - Work in successive views

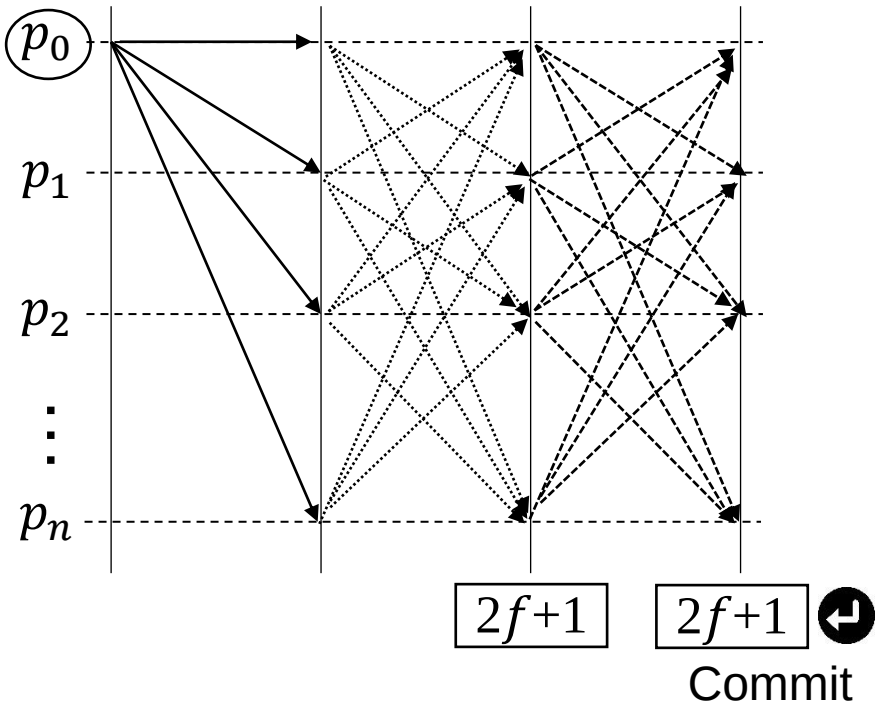


Three phases to reach consensus: too long!

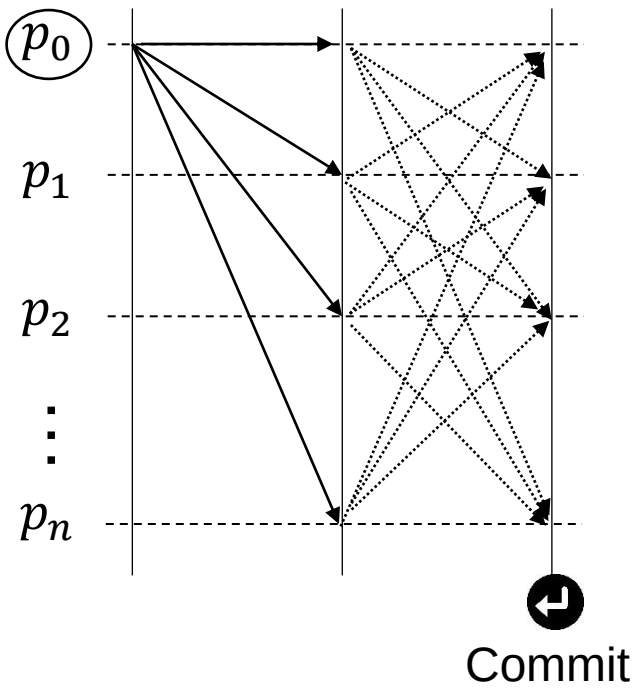
Background

- **Many works aim to reduce the commit phases**

Three phases commit → **Two phases commit**



Normal case



Normal case

Motivation

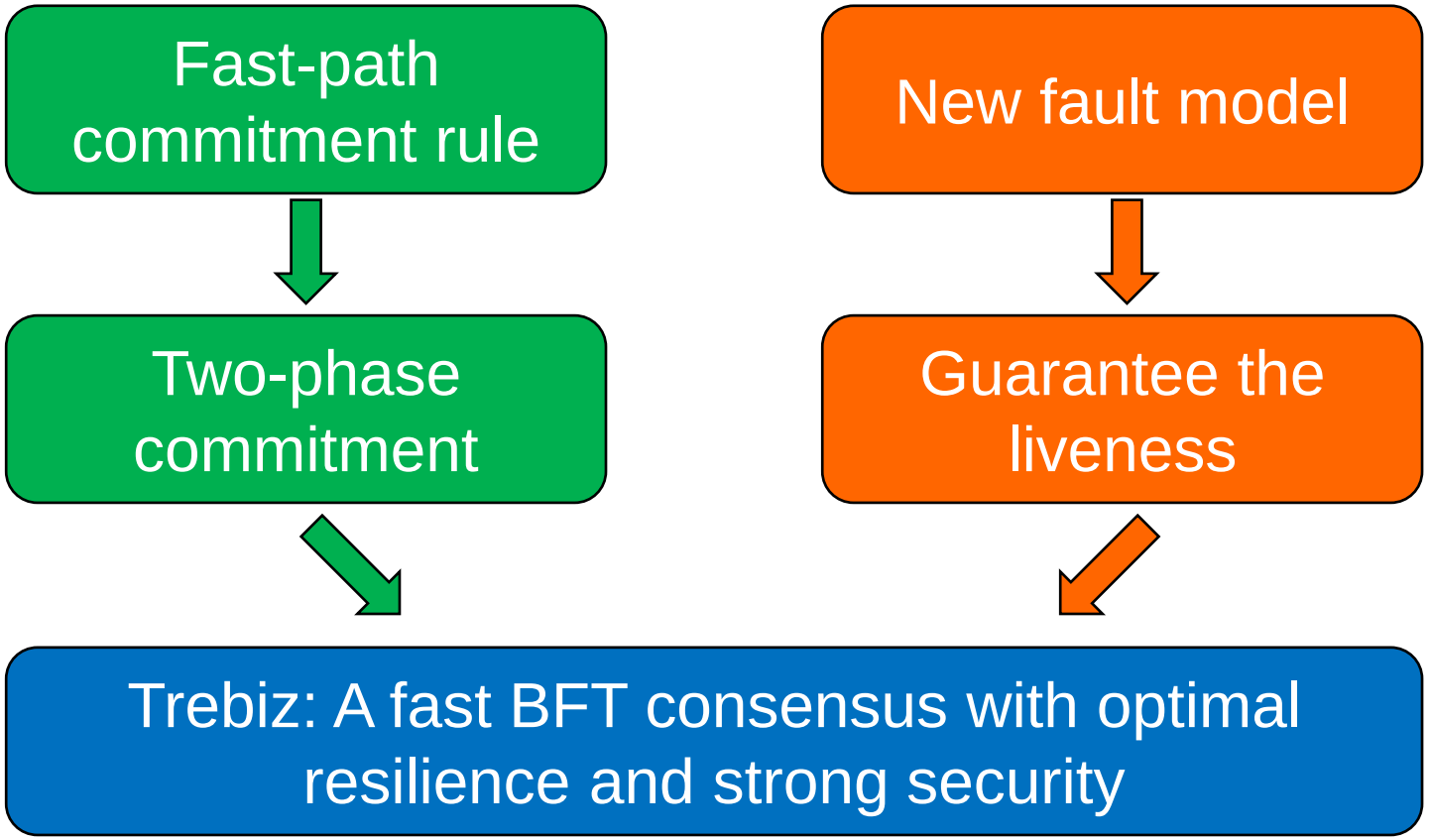
- **Many works aim to reduce the commit phases**

	Description	Con.
Zyzzyva[SOSP'07]	Rely on correct clients	Insecure
FastBFT[TC'18]	Rely on trusted hardware	Insecure
SBFT[DSN'19]	Need extra correct replicas	Lower resilience (< 1/3)

Can we reduce the commitment phases (i.e., latencies) without sacrificing security or resilience?

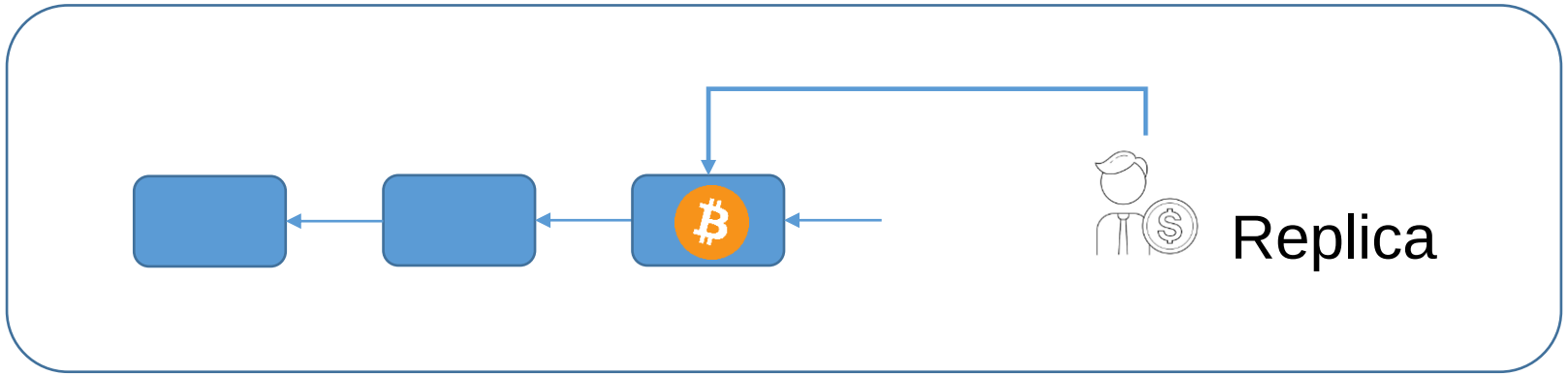
Trebiz Design

- **Main ideas**



Trebiz Design

- **New fault model**
  - Replicas tend **to keep the whole system running** rather than pushing it to a halt
    - May break **safety** but will not break **liveness**
    - Named as “Byzantine Merchant”



System halts or crashes → All replicas suffer a loss

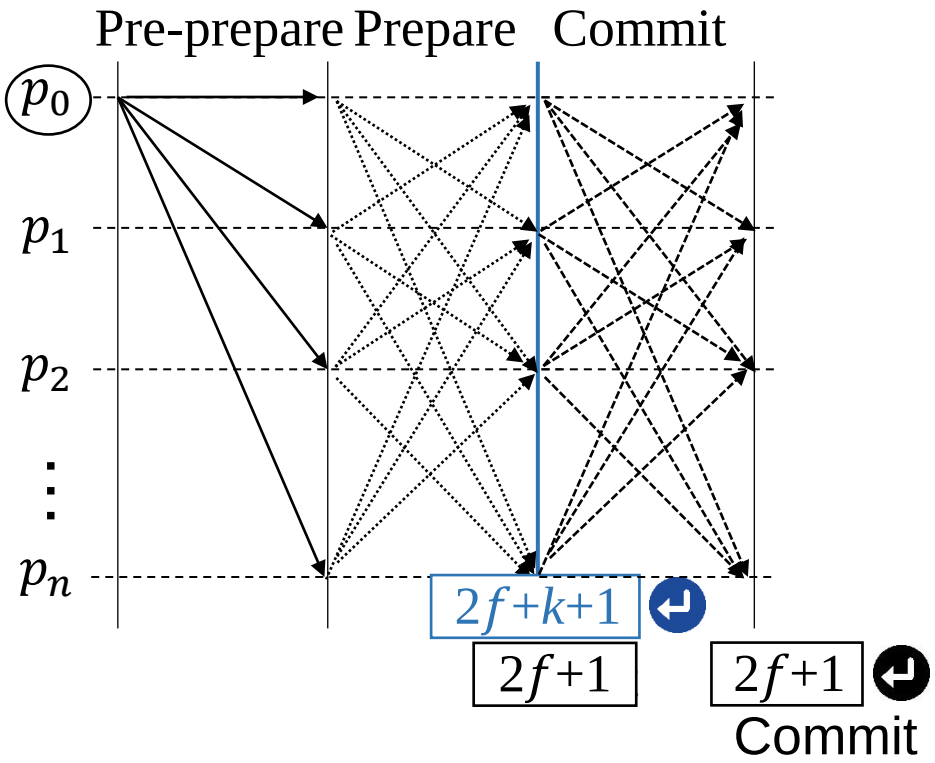
Trebiz Design

- **New fault model**
  - Divide Byzantine fault into Byzantine General and Byzantine Merchant
  - $N = 3f + 1 = n_c + n_g + n_a + n_p$
  - Up to f replicas can be Byzantine faulty
  - $n_g + n_a + n_p \leq f$

Category		Description	Number
Correct		Obey the protocol	n_c
Byzantine General (BG)		Deviate from the protocol arbitrarily	n_g
Byzantine Merchant	Active (ABM)	Break safety but maintain liveness **actively**	n_a
	Passive (PBM)	Break safety but maintain liveness **passively**	n_p

Trebiz Design

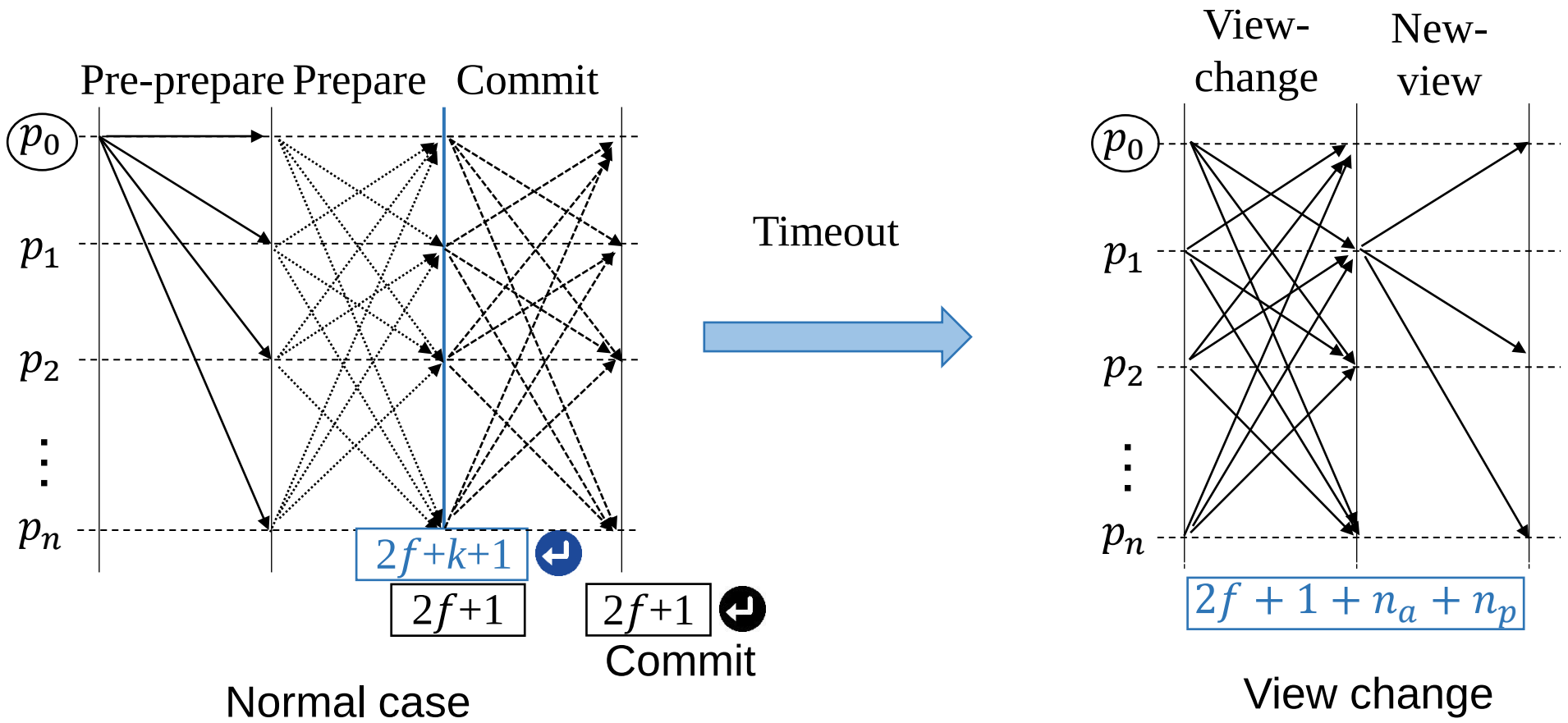
- **Normal case of Trebiz**
  - $k = f - n_a - \lfloor n_p/2 \rfloor$
  - Commit directly if receive $2f + k + 1$ prepare msgs, otherwise continue as PBFT



Normal case

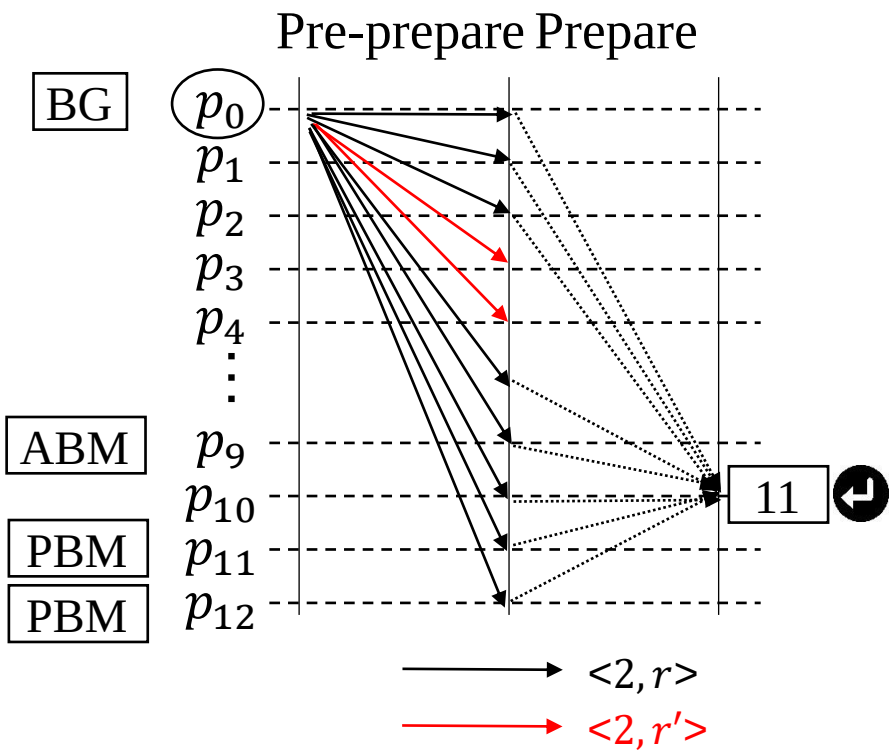
Trebiz Design

- **View change of Trebiz**
  - New leader waits for $2f + 1 + n_a + n_p$ view-change msgs to create new-view msg



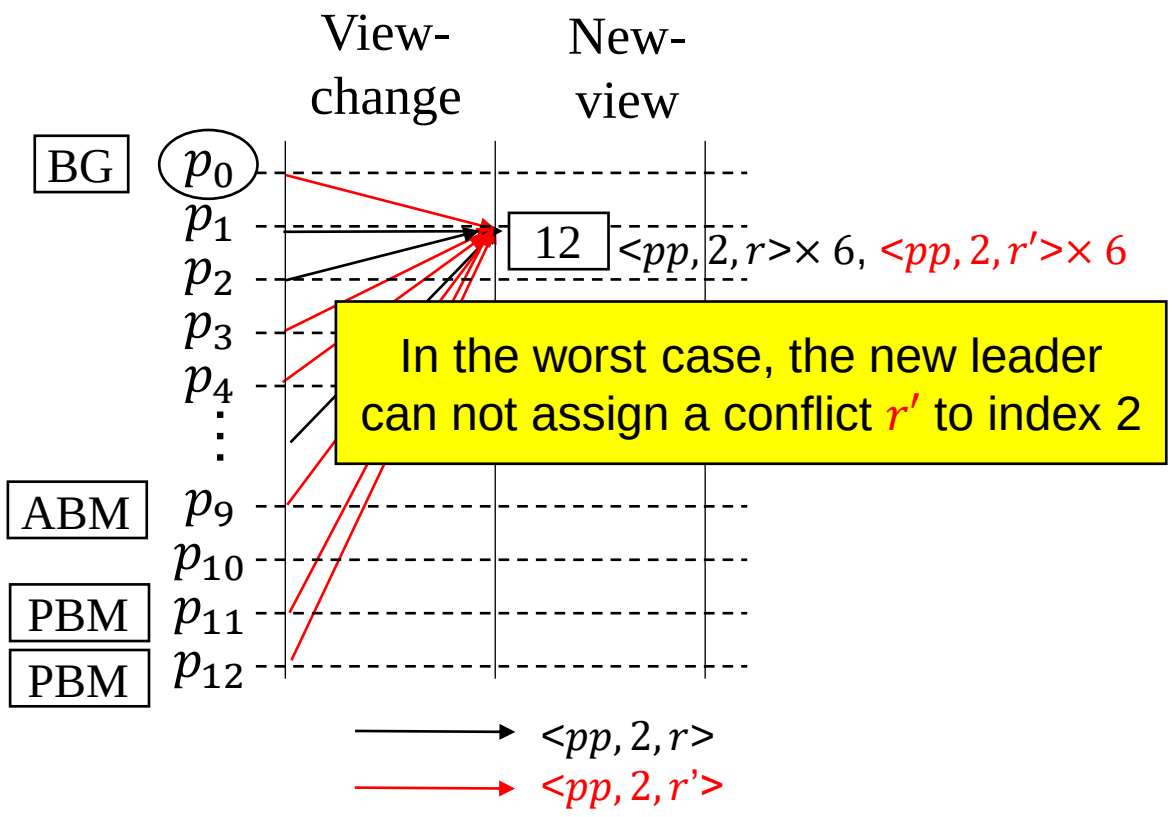
Trebiz Design

- **An example for the fast-path commitment**
  - $k = f - n_a - \lfloor n_p/2 \rfloor = 2$
  - BM tries to break **safety**



Normal case (f=4, n_g=1, n_a=1, n_p=2)

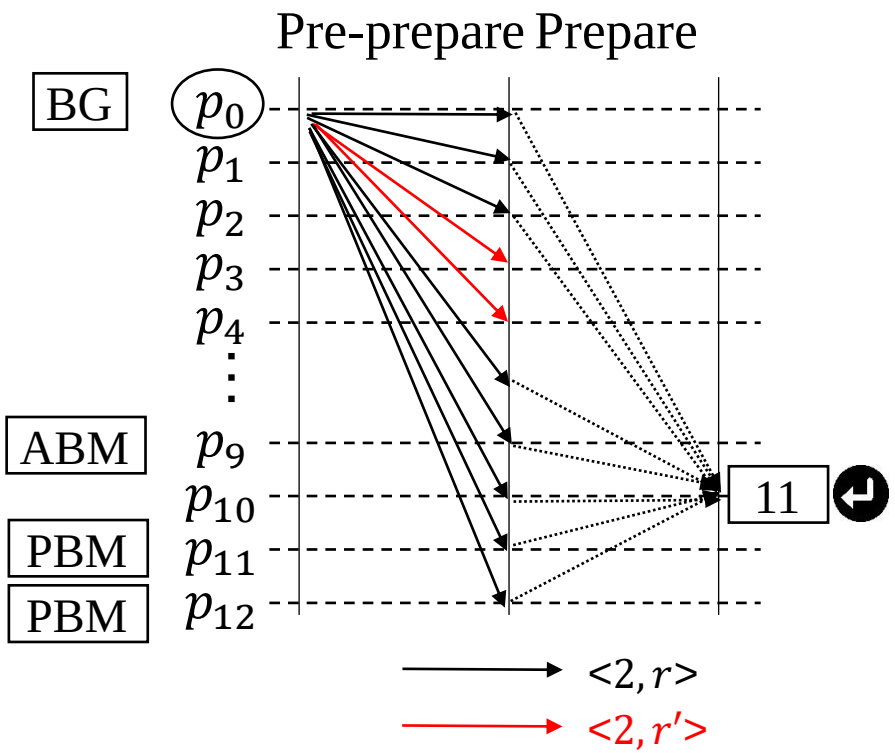
Timeout



View change (f=4, n_g=1, n_a=1, n_p=2)

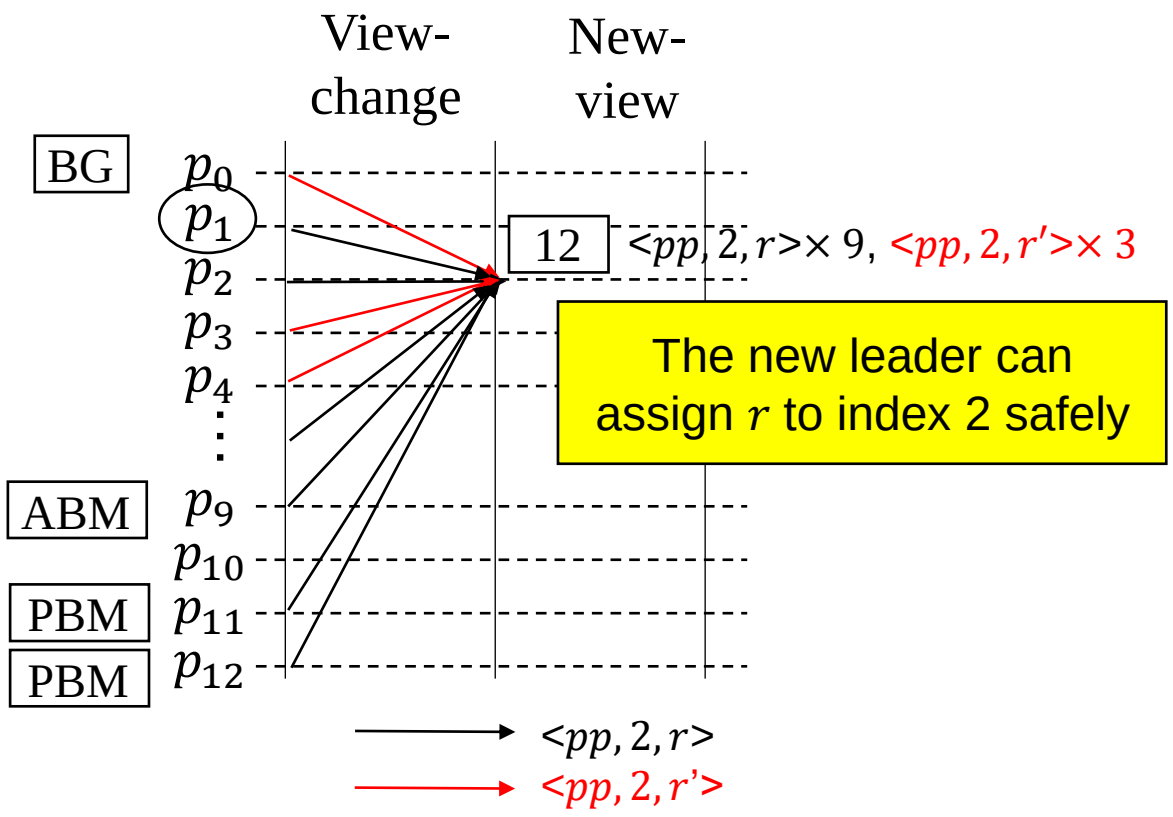
Trebiz Design

- **An example for the fast-path commitment**
  - $k = f - n_a - \lfloor n_p/2 \rfloor = 2$
  - BM tries to maintain **liveness**



Normal case (f=4, n_g=1, n_a=1, n_p=2)

Timeout



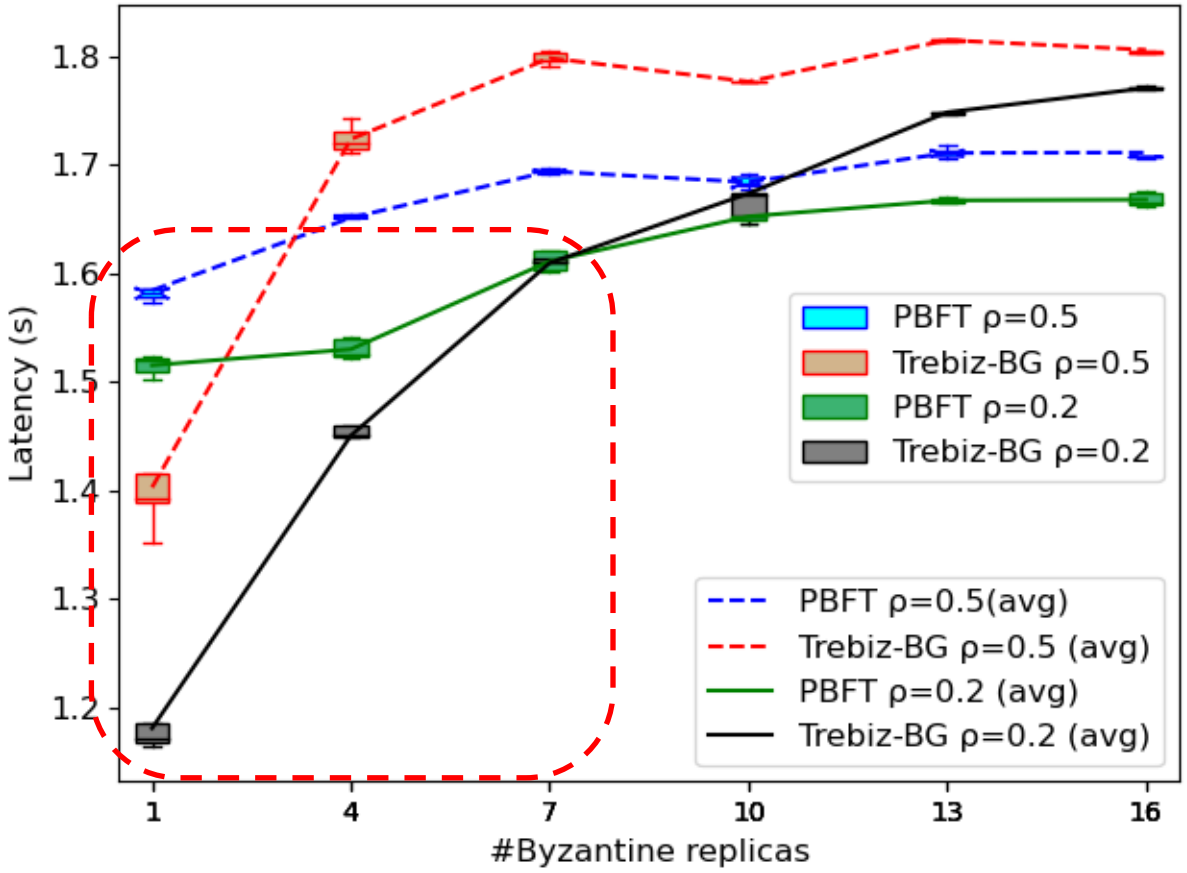
View change (f=4, n_g=1, n_a=1, n_p=2)

Evaluation

- **Implementation and experimental setting**
  - Prototype implementation
    - Implement PBFT in Golang, and implement Trebiz on top of PBFT
  - Experimental environment
    - Alibaba cloud ECS.c6.2xlarge instance
    - 8 vCPU, 16 GB memory and 100 Mbps bandwidth
    - Spanning 5 continents
  - Malicious behavior simulation of malicious replicas
    - Be non-responsive with a probability ρ

Evaluation

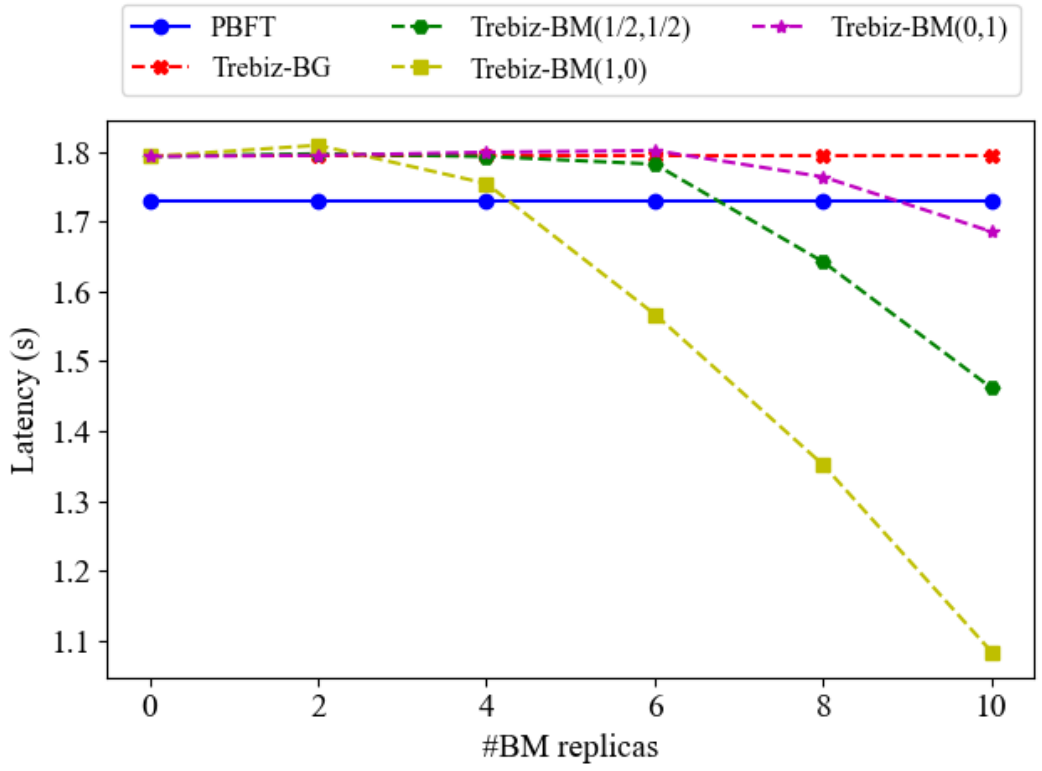
- **Latency comparison between PBFT and Trebiz-BG**
  - Trebiz-BG outperforms PBFT when the number of Byzantine replicas is small



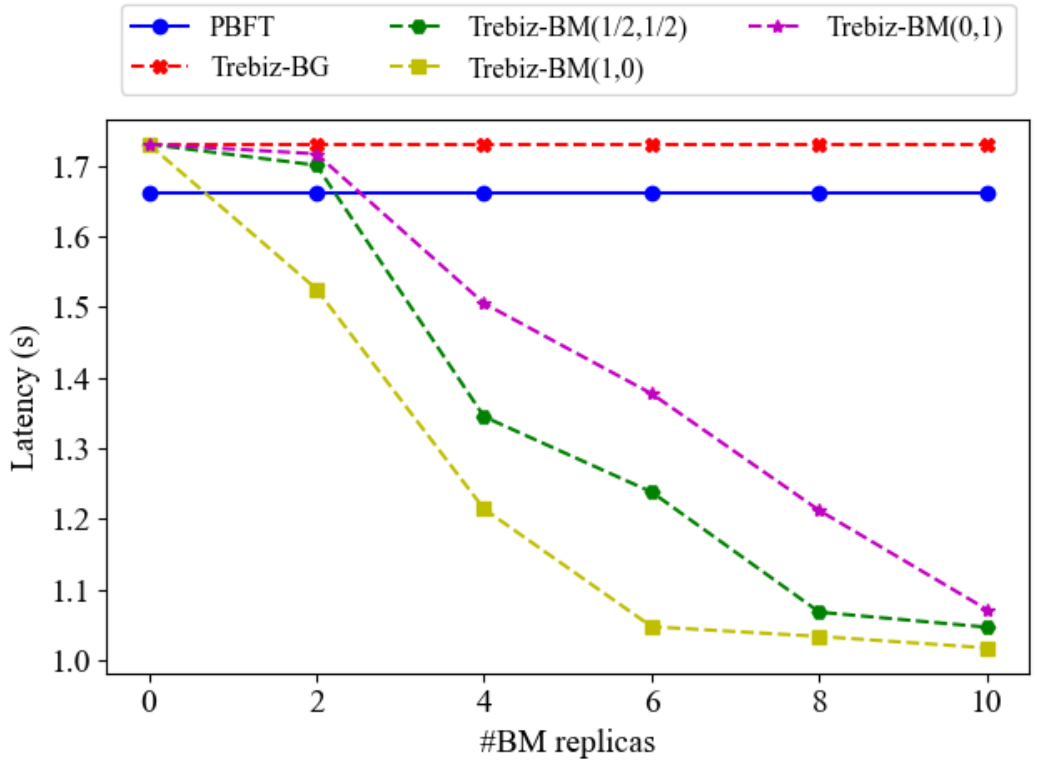
Evaluation

- **Latency comparison**
  - More BM, lower latency in Trebiz-BM

Trebiz-BM (t_a, t_p)
- t_a and t_p are the portions of ABM and PBM



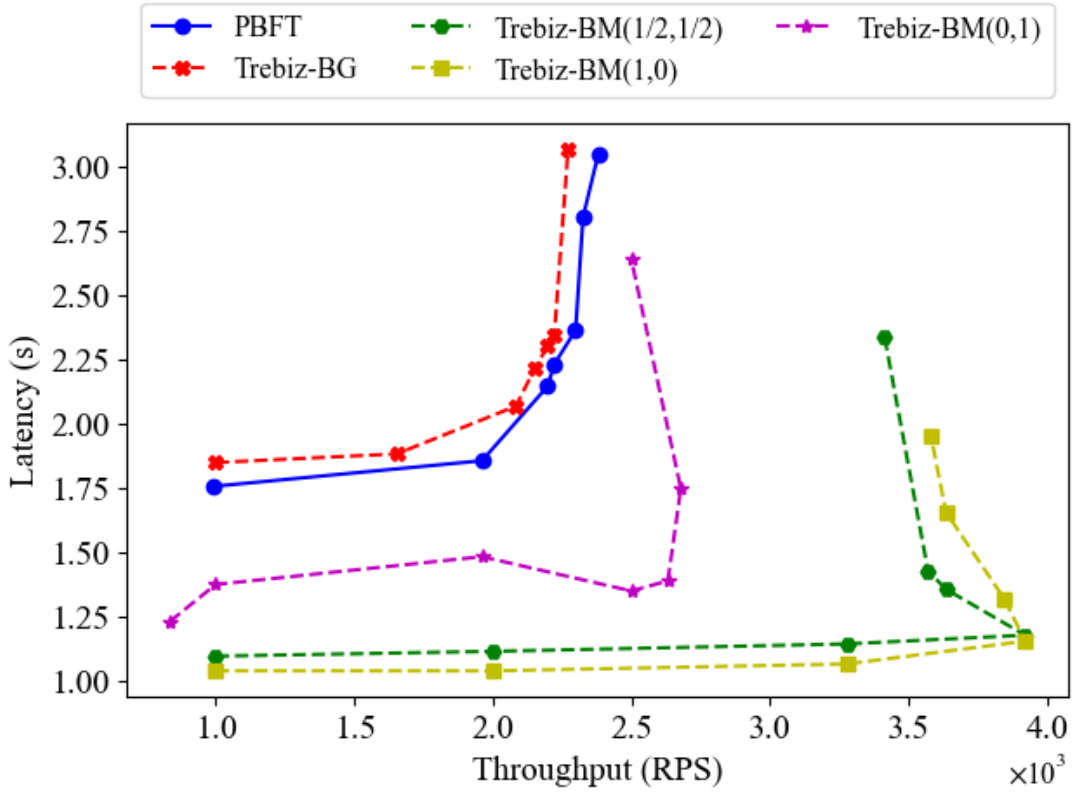
(a) $\rho = 0.5$



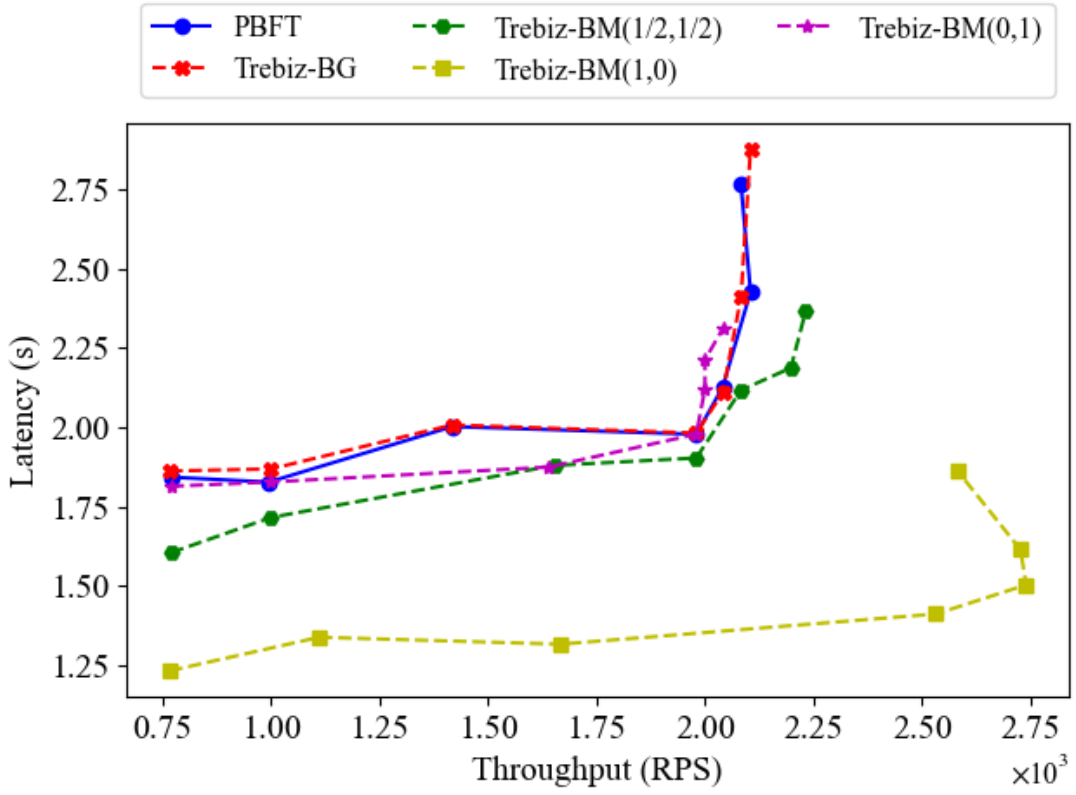
(b) $\rho = 0.2$

Evaluation

- **Latency vs. throughput**
  - Higher throughput and lower latency in Trebiz-BM



(a) $\rho = 0.5$



(b) $\rho = 0.2$

Conclusion

- Take a re-understanding of fault models by considering the economical interests and propose a new fault model, which classifies Byzantine replicas into BG, ABM, and PBM

- Propose a fast consensus protocol named Trebiz, which achieves fast-path commitment with optimal resilience and strong security

- Implement a prototype of Trebiz and conduct extensive experiments to demonstrate its feasibility and efficiency

Thanks!
Q&A

More questions, please feel free to contact us:
hustlp@hust.edu.cn