OSDI 2016, November 3 2016, Savannah, GA, USA

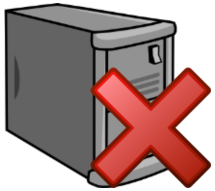
XFT:
Practical Fault Tolerance beyond Crashes

Shengyun Liu (NUDT, China)
Paolo Viotti (EURECOM, France)
Christian Cachin (IBM Research – Zurich)
Vivien Quéma (INP Grenoble, France)
Marko Vukolić (IBM Research – Zurich)

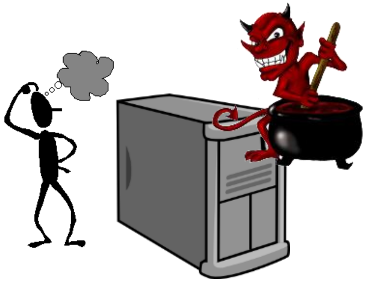
Fault tolerance

Building systems that tolerate **machine** and **network** faults

What machine faults?



Crash faults (CFT)



Non-crash (a.k.a. Byzantine) faults (BFT)

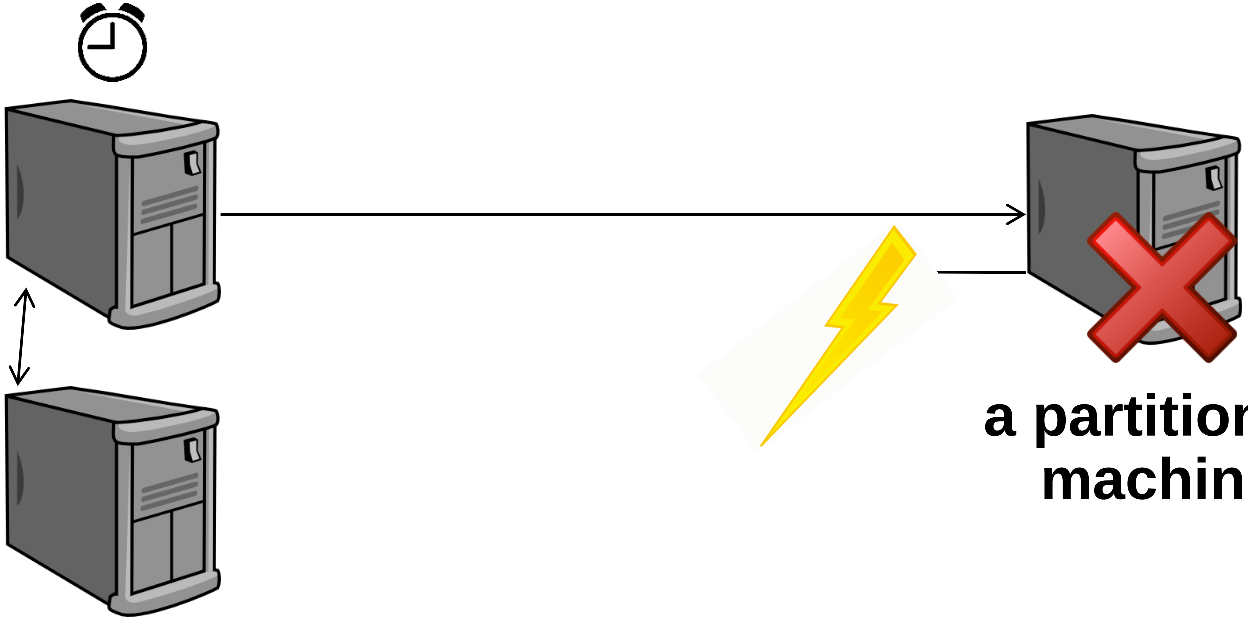
Data losses, omissions, data corruptions, bugs, misconfigurations, hardware faults, cosmic rays, incorrect firmware, operator errors, …

…and malicious behavior

Network faults (a.k.a network partitions, asynchrony)

Reflect the inability of **correct** machines

to communicate in a timely manner (i.e., synchronously)



a partitioned machine

This paper in one slide: XFT (cross fault tolerance)

in absence of non-crash faults

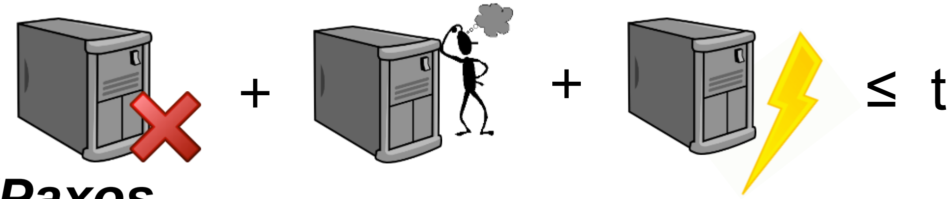
the same fault-tolerance guarantees as asynchronous CFT

(i.e., with the same thresholds)

in presence of non-crash faults

> 0

fault-tolerance guarantees as long as
the number of faulty or partitioned machines is within a threshold

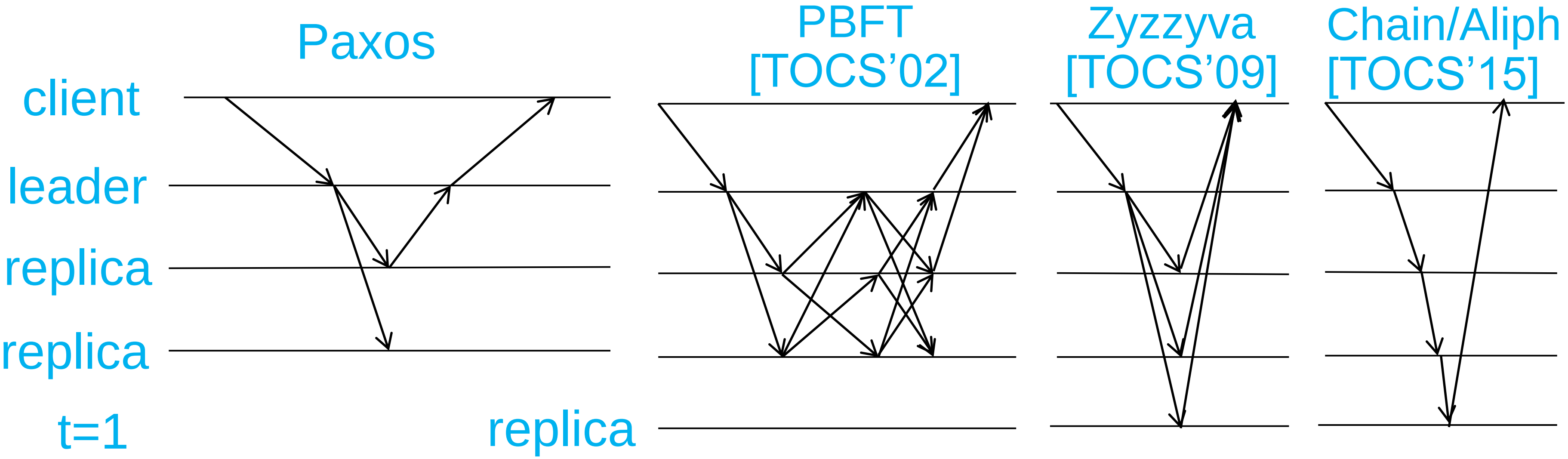


XFT showcase: XPaxos

- The first state-machine replication (SMR) protocol in the XFT model
- (almost) **as efficient as optimized CFT Paxos**

CFT vs. BFT deterministic SMR

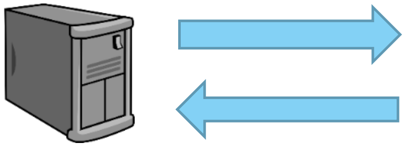
CFT SMR state-of-the-art (e.g., Paxos, RAFT, Zab)	**BFT SMR state-of-the-art (e.g., PBFT, Zyzzyva, Chain/Aliph)**
2t+1 replicas	3t+1 replicas
Efficient message patterns	Inefficient message patterns



NB: These are only common-case message patterns

FT guarantees: **CFT SMR**

machine faults	no	yes
Non-crash faults	none 🙁	none 🙁
Crash faults	**Consistency** 🙂 w. any number of machine faults **Availability** 🙂 w. up to n/2 machine faults **Performance/cost** 🙂 very good (production use)	**Consistency** 🙂 w. any number of faulty or partitioned machines **Availability** 🙂 w. up to n/2 faulty or partitioned machines



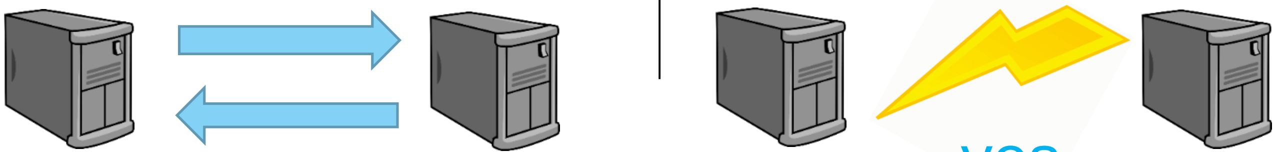
no

Network faults?

yes

FT guarantees: BFT SMR

machine faults	no	yes
Non-crash faults	**Consistency & Availability** w. up to n/3 machine faults 😐	**Consistency** 🙂 w. up to n/3 machine faults and any no. of partitioned machines **Availability** 😐 w. up to n/3 faulty or partitioned machines
Crash faults ❌	**Consistency** 🙂 w. any number of machine faults **Availability** 😐 w. up to n/3 machine faults **Performance/cost** ☹ poor (compared to CFT)	**Consistency** 🙂 w. any number of faulty or partitioned machines **Availability** 😐 w. up to n/3 faulty or partitioned machines



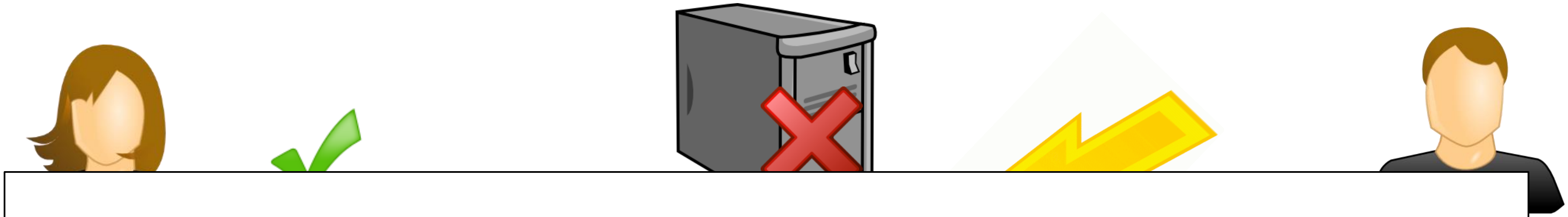
no

Network faults?

yes

The Cost of Asynchronous BFT (Infamous 3t+1)

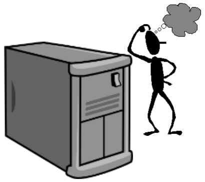
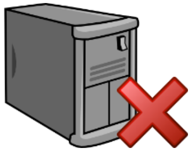
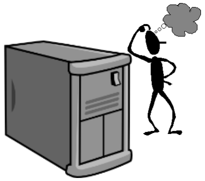
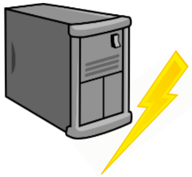
x = "I like you"



W

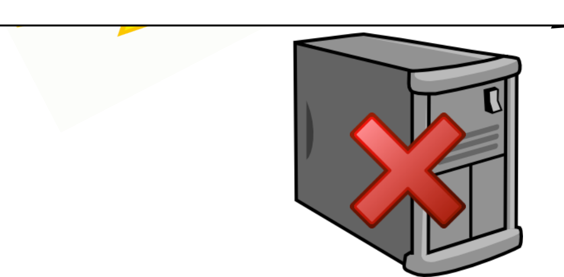
ull

The cost of BFT comes from providing consistency when

n/3 >  > 0 and  +  +  ≥ n/2

Such a particular adversary is in many use cases

irrelevant



t = 1

XFT (cross fault tolerance)

in absence of non-crash faults

the same fault-tolerance guarantees as asynchronous CFT

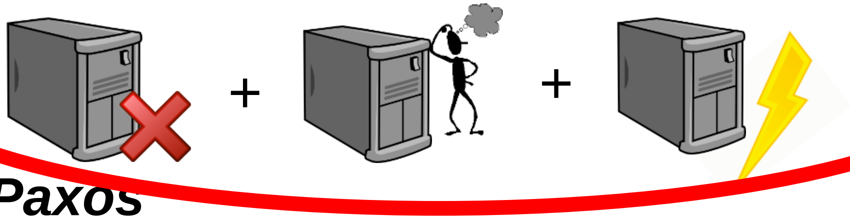
(i.e., with the same thresholds)

> 0

in presence of non-crash faults

fault-tolerance guarantees as long as

the number of faulty or partitioned machines is within a threshold

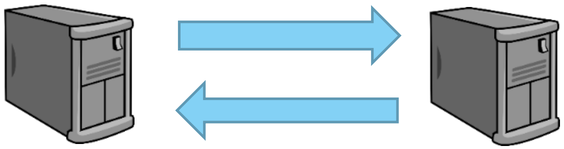


XFT showcase: XPaxos

- The first state-machine replication (SMR) protocol in the XFT model
- (almost) **as efficient as optimized CFT Paxos**

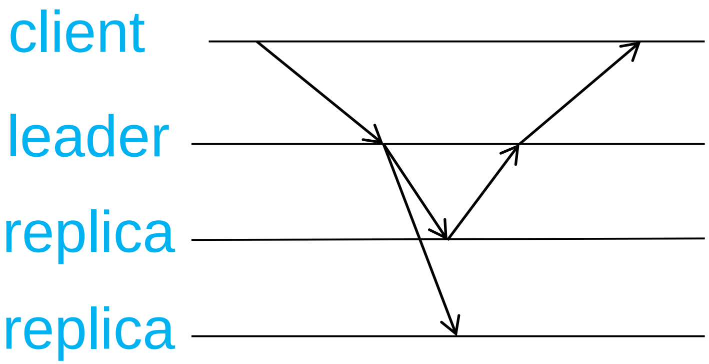
XPaxos: XFT SMR

machine faults		Network faults? no	Network faults? yes
	Non-crash faults	🙂 **Consistency & Availability** w. up to n/2 machine faults	**Consistency** 😐 w. up to n/2 faulty or partitioned machines **Availability** 🙂 w. up to n/2 faulty or partitioned machines
	Crash faults ❌	**Consistency** 🙂 w. any number of machine faults **Availability** 🙂 w. up to n/2 machine faults **Performance/cost** 🙂 very good (compared to CFT)	**Consistency** 🙂 w. any number of faulty or partitioned machines **Availability** 🙂 w. up to n/2 faulty or partitioned machines

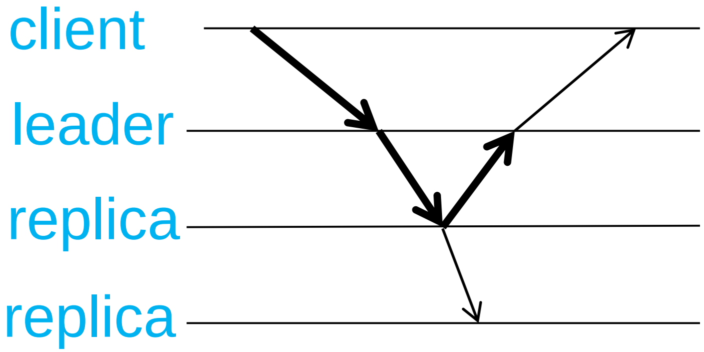


XPaxos message pattern (common case)

Paxos (t=1)

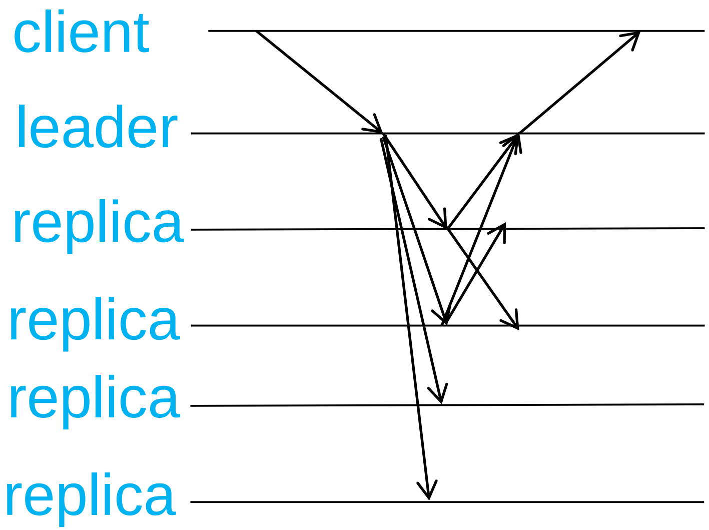


XPaxos (t=1)

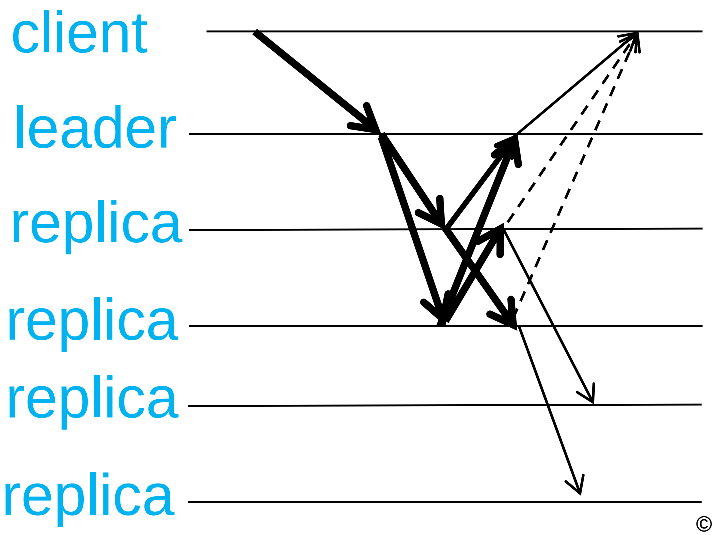


→ Digitally signed messages

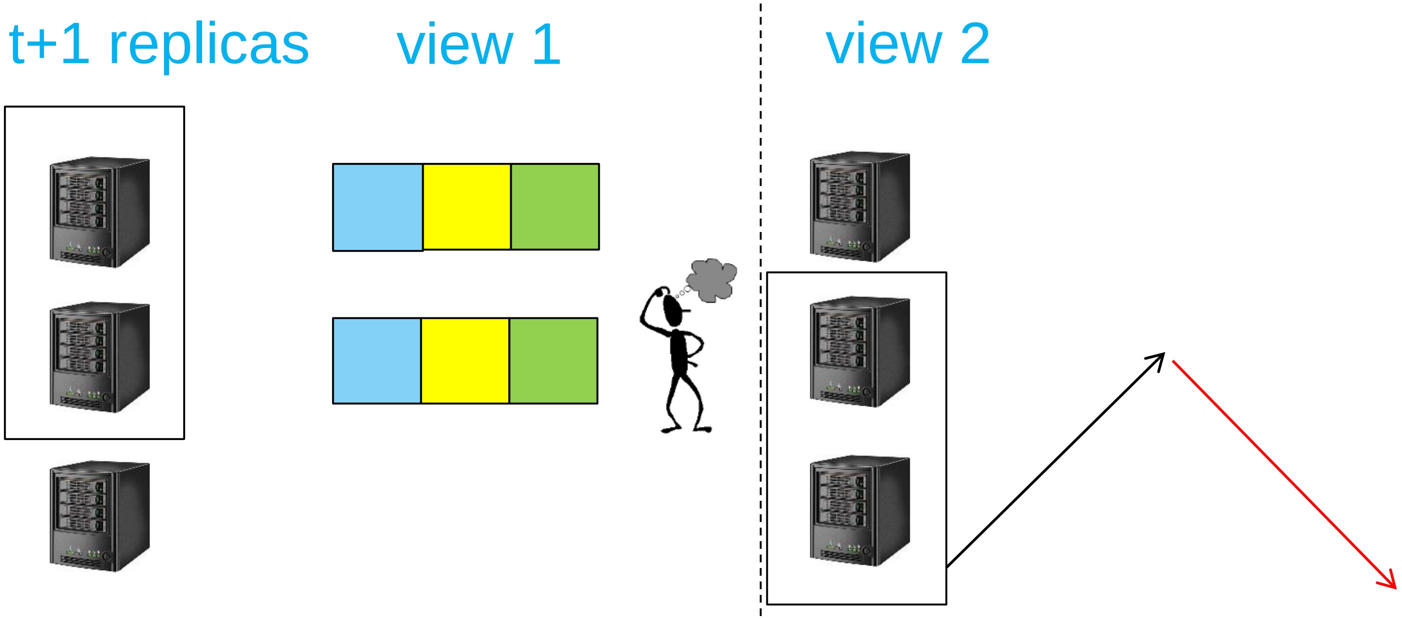
Paxos (t>1, here t=2)



XPaxos (t>1, here t=2)



View-change sketch: a problem



t+1 replicas

view 1

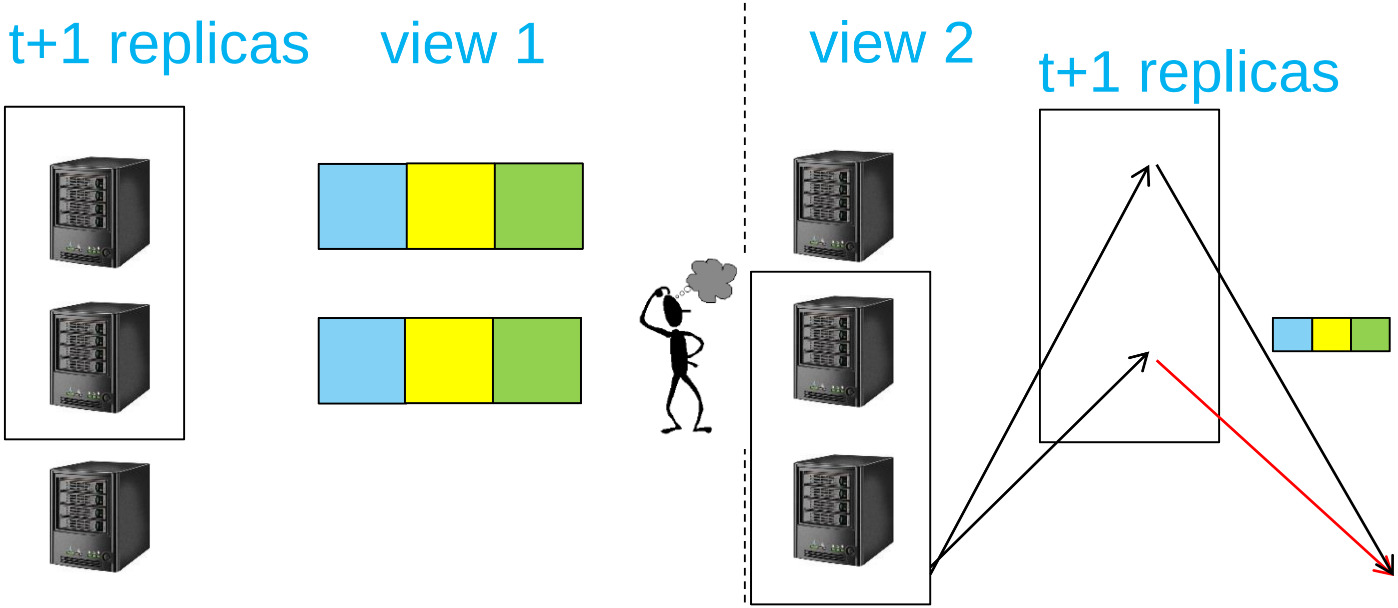
view 2

t+1 replicas

What happened in view1?

Nothing!

View-change sketch: XPaxos solution



t+1 replicas

view 1

view 2

t+1 replicas

t+1 replicas

what happened in view1?

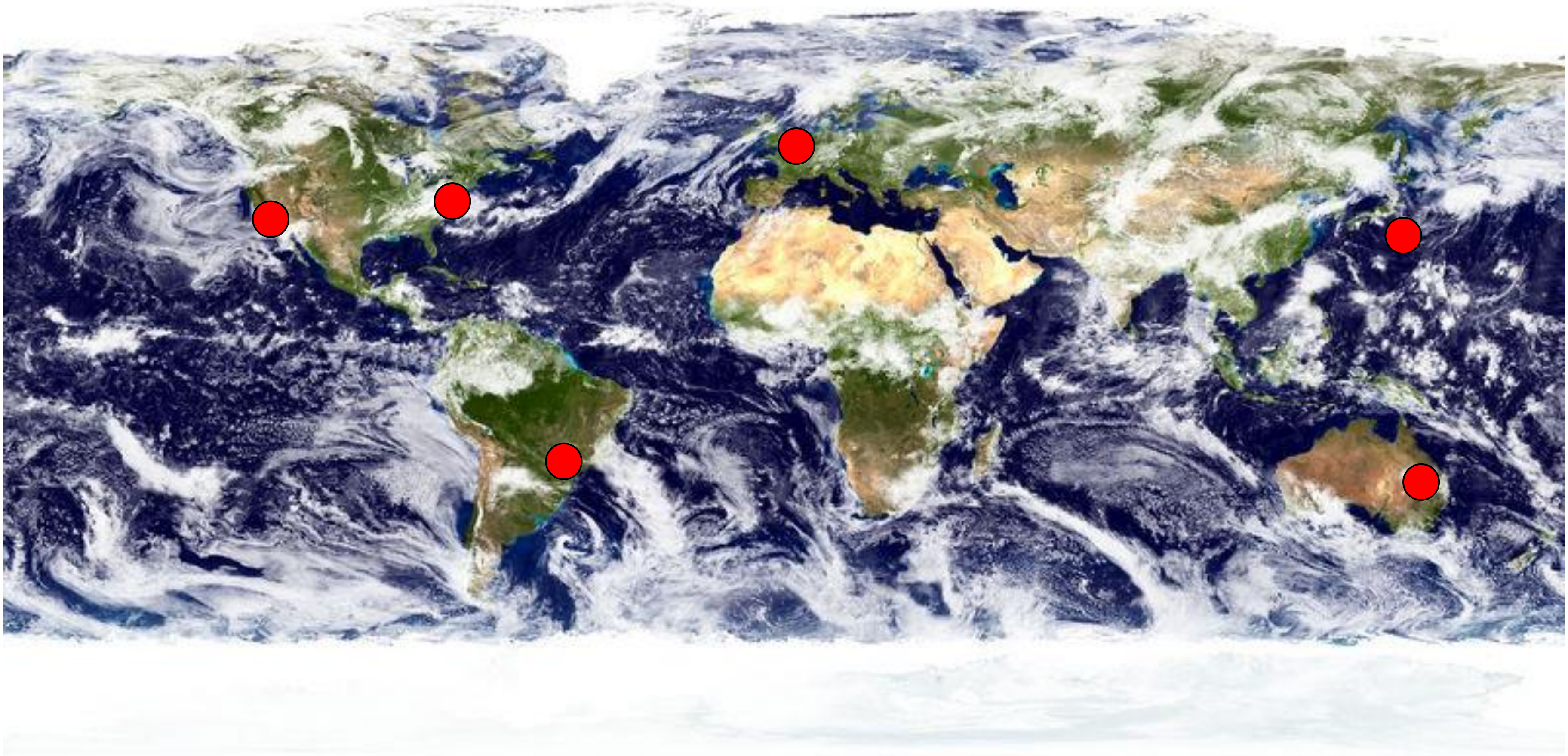
Nothing!

Wait for at least 1 response

+

Connection <u>timeout</u> to all t+1 replicas (including at least one correct)

Deployment: geo-replication playground



Choosing the timeout (for view-change)

Machines TCP ping-ing each other every 100ms for 3 months

- Amazon AWS EC2 micro VMs in 6 regions
  - US West (CA), US East (VA), Ireland (EU), Brazil (BR), Tokyo (JP), Sydney (AU)

Round-trip Latency [ms]	avg	99%	99.9%	99.99%	max
min	85 [CA-VA]	130 [CA-JP]	1082 [CA-VA]	1097 [CA-VA]	5208 [JP-AU]
max	401 [AU-BR]	516 [AU-BR]	1474 [AU-BR]	2495 [JP-BR]	169749 [VA-EU]

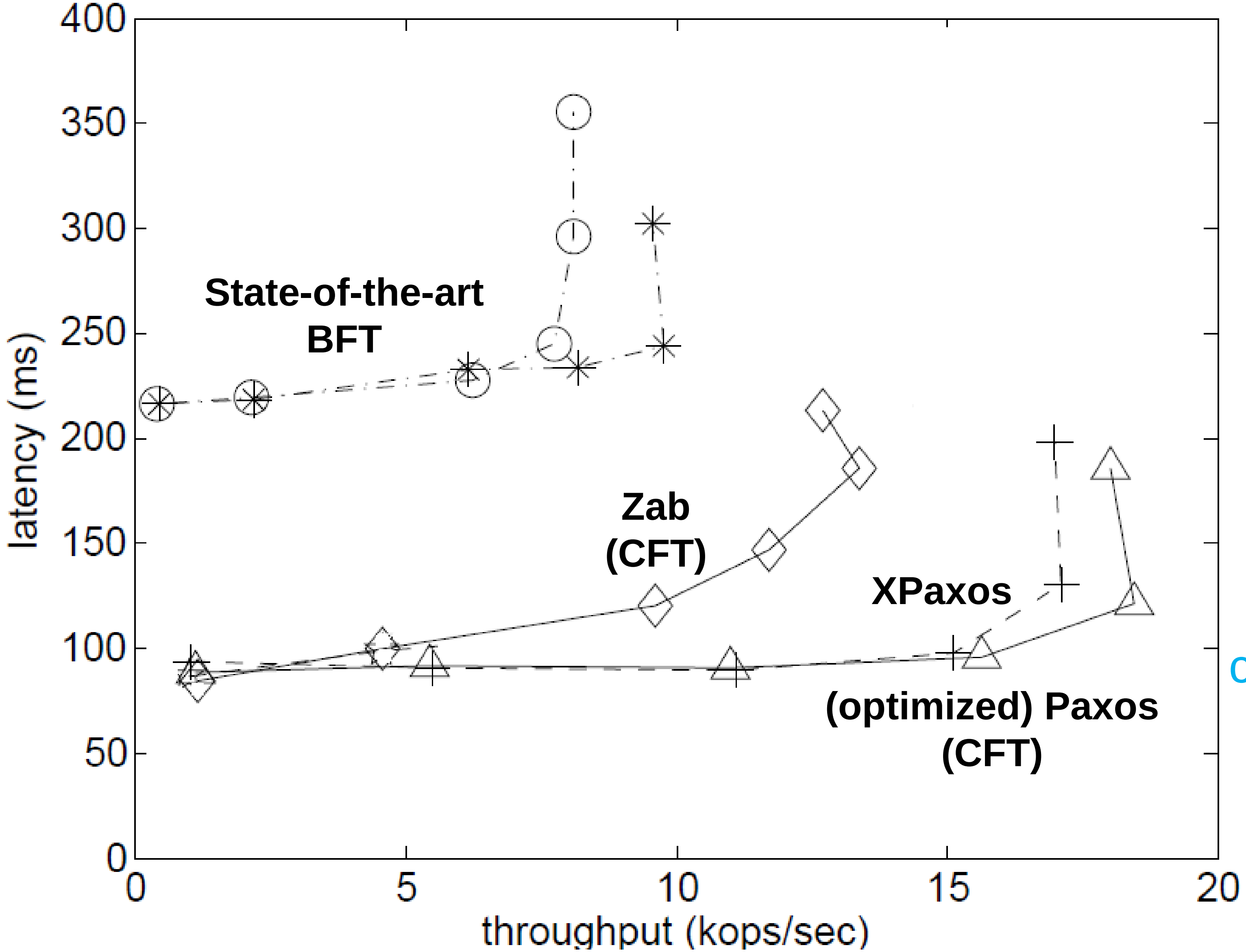
< 2.5s

- IBM Softlayer
  - Mexico (MX), San Jose (CA), Washington (DC), London (UK), Tokyo (JP), Sydney (AU)

Round-trip Latency [ms]	avg	99.99%	Max
min	65 [CA-MX]	1077 [CA-DC]	3476 [UK-DC]
max	305 [UK-AU]	1440 [UK-AU]	127869 [JP-DC]

< 1.5s

Performance (ZooKeeper on Amazon EC2 testbed)



t=1

write-only workload

1kB requests

closed-loop

Where/when to use XFT?

- Tolerating “accidental” non-crash (Byzantine) faults

- Wide-area networks and geo-replicated systems

- When adversary cannot control the network at will

- “Permissioned” blockchain

HYPERLEDGER PROJECT

www.hyperledger.org

Thank you!

IBM Research - Zurich is hiring

Keywords: distributed systems
fault-tolerance
consistency
blockchain