Byzantine Fault-Tolerant Distributed Computing: Reconfiguration and Consensus

Student: Andrei Tonkikh

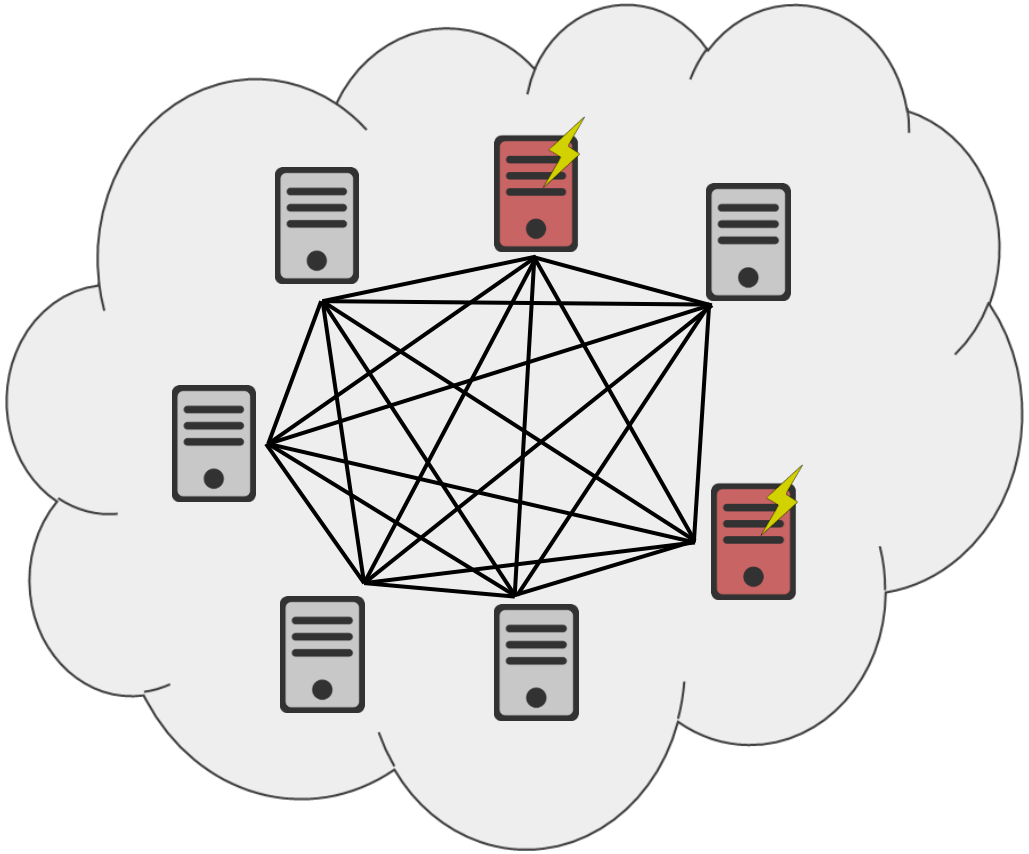
Research Supervisor: Petr Kuznetsov

May 2021

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY
SAINT PETERSBURG

Byzantine fault tolerance

- **Processes** are connected by a complete network;
- **Correct** processes follow the protocol;
- **Byzantine** (also called **faulty**) processes are controlled by a **malicious adversary**.



Network models

- **Synchronous**
  - The network is **always** predictable (known upper bound on message delays)
- **Partially synchronous**
  - The network is **usually** predictable (the upper bound may sometimes be violated)
- **Asynchronous**
  - The network is **unpredictable** (no upper bound)

Network models

- **Synchronous**
  - The network is **always** predictable
- **Partially synchronous**
  - The network is **usually** predictable

Consensus can be solved

- **Asynchronous**
  - The network is **unpredictable**

Consensus cannot be solved

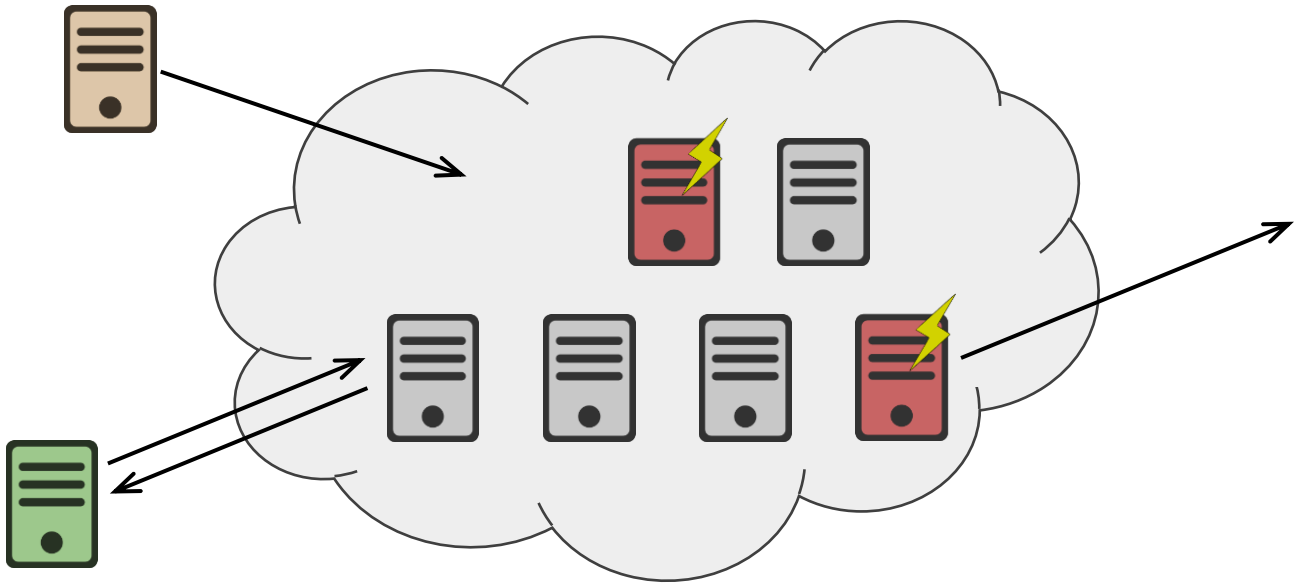
Network models

- **Asynchronous**
  - The network is **unpredictable**

} Consensus cannot be solved

Reconfiguration

- Replicas need to be **replaced**, **added**, or **removed**;
- Hard without consensus;



Reconfiguration

- Replicas need to be **replaced**, **added**, or **removed**;
- Hard without consensus;
- Well-studied with crash failures, but **no Byzantine fault-tolerant solutions**.

Network models

- **Synchronous**
  - The network is **always** predictable
- **Partially synchronous**
  - The network is **usually** predictable

Consensus can be solved

- **Asynchronous**
  - The network is **unpredictable**

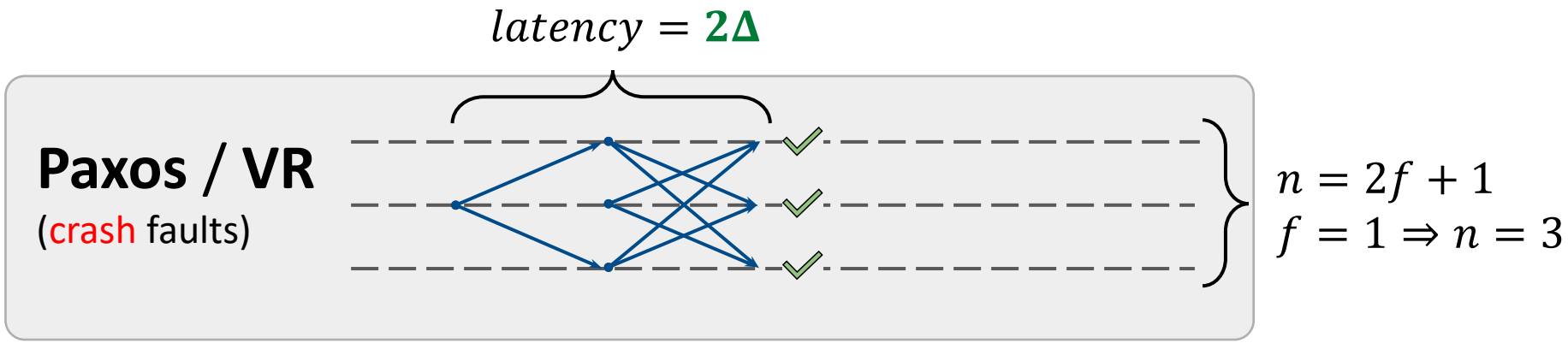
Consensus cannot be solved

Network models

- **Partially synchronous**
  - The network is **usually** predictable

} **Consensus** can be solved

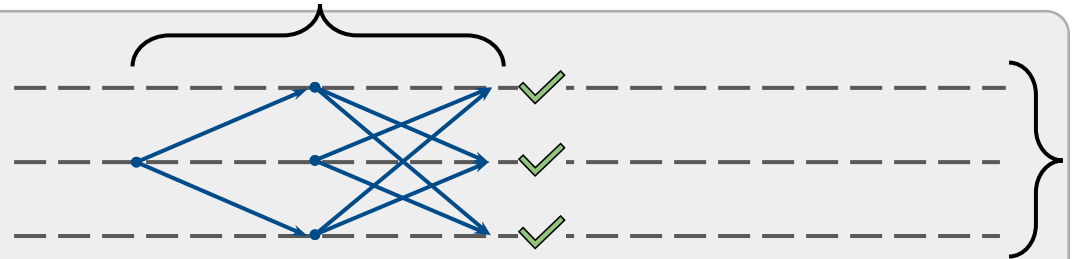
Optimistic latency & resilience



Optimistic latency & resilience

$latency = \mathbf{2\Delta}$

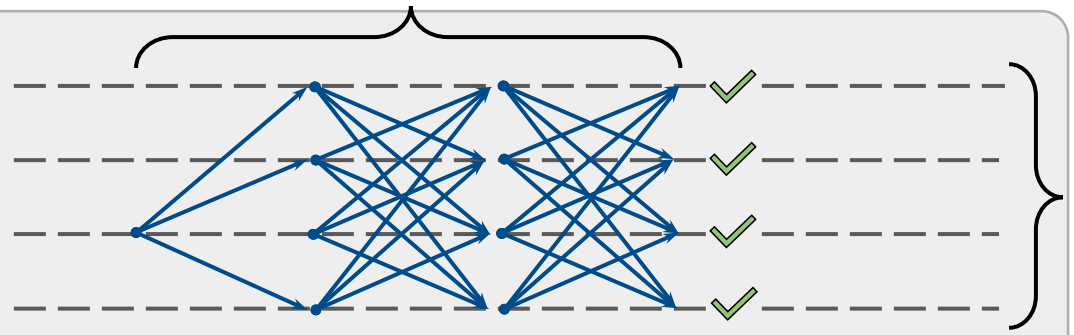
Paxos / VR
(crash faults)



$n = 2f + 1$
$f = 1 \Rightarrow n = 3$

$latency = 3\Delta$

PBFT
(**Byzantine** faults)



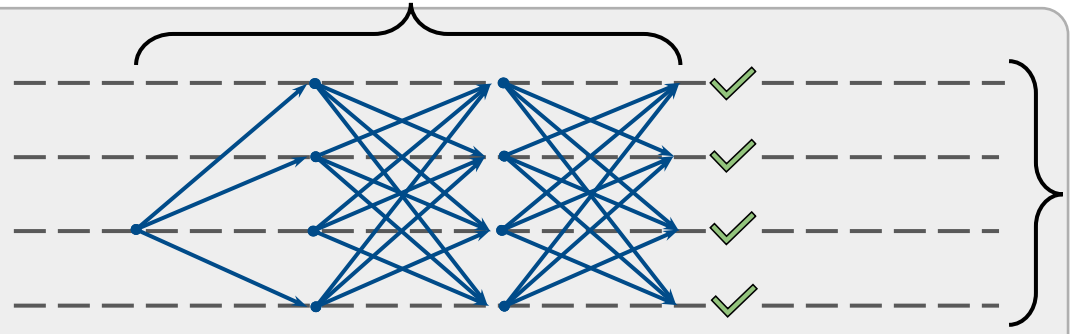
$n = \mathbf{3}f + \mathbf{1}$
$f = 1 \Rightarrow \boldsymbol{n} = \mathbf{4}$

Optimistic latency & resilience

$latency = 3\Delta$

PBFT

(**Byzantine** faults)



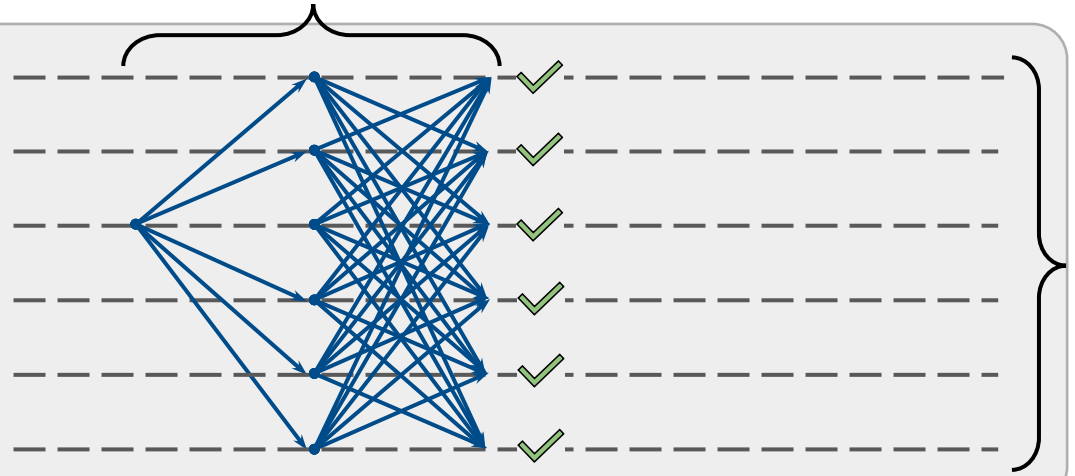
$n = 3f + 1$

$f = 1 \Rightarrow n = 4$

$latency = 2\Delta$

FaB Paxos / Zelma

(**Byzantine** faults)



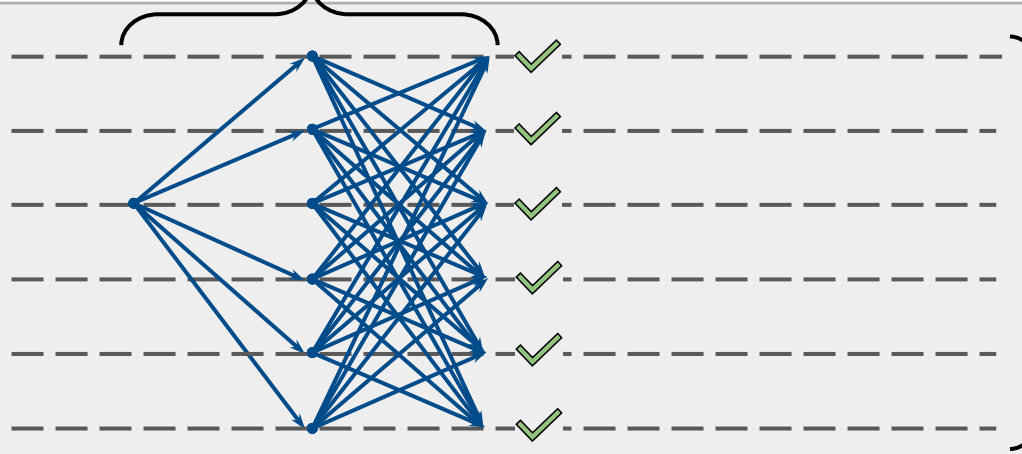
$n = 5f + 1$

$f = 1 \Rightarrow n = 6$

Optimistic latency & resilience

$latency = \mathbf{2\Delta}$

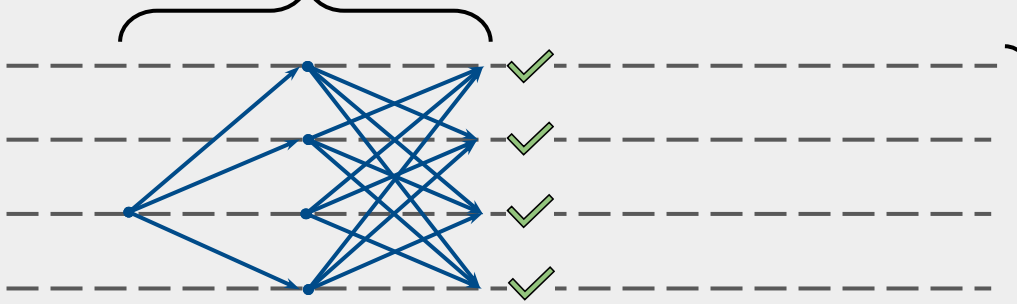
FaB Paxos / Zelma
(**Byzantine** faults)



$n = 5f + 1$
$f = 1 \Rightarrow n = 6$

$latency = \mathbf{2\Delta}$

This work
(**Byzantine** faults)



$n = \mathbf{5f - 1}$
$f = 1 \Rightarrow \boldsymbol{n = 4}$

Goal:

- Study important problems in different models of Byzantine fault-tolerant distributed computing: **asynchronous reconfiguration** and **fast Byzantine consensus**.

Objectives:

- **Formalize the problem** of asynchronous reconfiguration in the model with Byzantine faults;
- **Find a solution** for the problem;
- Rigorously **prove the correctness** of the resulting protocol;
- Create a fast Byzantine **consensus algorithm** that requires just $n = 5f - 1$ replicas to tolerate f failures and prove its correctness;
- Prove a matching **lower bound** on the resilience of fast Byzantine consensus.

Reconfiguration
formalization & solution

- A framework that allows you to **easily obtain specifications** of reconfigurable objects out of their static specifications is proposed;
- The problem of reconfiguration is split into two parts: **state transfer** and **conflict resolution**;
- State transfer is implemented using **forward-secure digital signatures** in a novel way;
- Conflict resolution is resolved by utilizing two instances of **Dynamic Byzantine Lattice Agreement** – a new object type proposed in this work.

Reconfiguration
applications

- Reconfigurable Atomic Max-Register ⇒ **reconfigurable key-value storage**;
- As proposed in [CGK+20], asynchronous reconfiguration can be used to implement efficient **permissioned cryptocurrency** solutions.

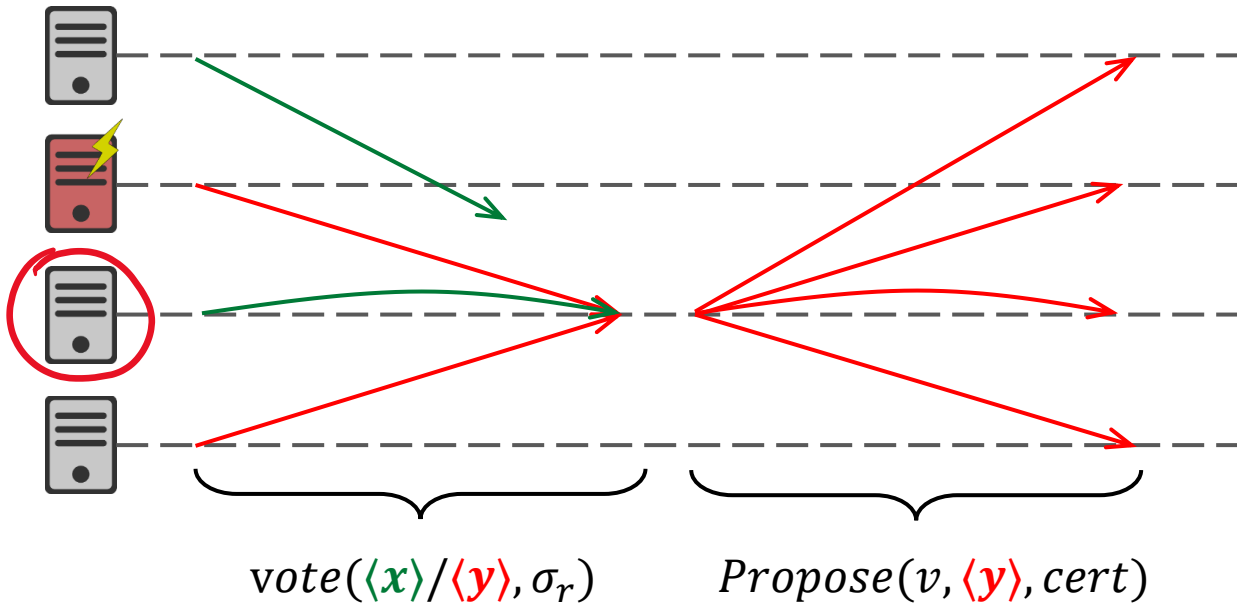
[CGK+20] D. Collins, et al. Online payments by merely broadcasting messages. *(DSN 2020)*

Reconfiguration
following research

- In [KPPT21], techniques proposed in this thesis are applied to implement a **permissionless asynchronous cryptocurrency**;
- In [SKRT21], some of the ideas of this work are applied to implement **self-healing distributed objects**.

[KPPT21] P. Kuznetsov, et al. Permissionless and Asynchronous Asset Transfer. *(arXiv preprint, 2021)*
[SKRT21] L. F. De Souza, et al. Accountability and Reconfiguration: Self-Healing Lattice Agreement. *(arXiv preprint, 2021)*

Fast consensus with $n = 5f - 1$
safety failure example



Key observation: the new leader can deduce that the previous one is Byzantine.

Fast Byzantine consensus
lower bound

There is no fast Byzantine consensus algorithm that can work with just $n = 5f - 2$ processes.

	$\{p\}$	P_1	P_2	P_3	P_4	P_5	
ρ_5		s_1	s_2	s_3	s_4		indistinguishable for P_3
ρ_4		s_1	s_2	s_3		t_5	indistinguishable for P_1, P_2, and P_5
ρ_3		s_1	s_2		t_4	t_5	indistinguishable for P_1, P_4, and P_5
ρ_2		s_1		t_3	t_4	t_5	indistinguishable for P_3
ρ_1			t_2	t_3	t_4	t_5	
size	1	f	$f-1$	$f-1$	$f-1$	f	

Fast Byzantine consensus
summary

Protocol	Optimistic latency	# replicas	# replicas when $f = 1$
PBFT [CL02]	3Δ	$\boldsymbol{n = 3f + 1}$	**4**
FaB Paxos [MA06] / Zelma [AGMM18]	$\boldsymbol{2\Delta}$	$n = 5f + 1$	6
This Work [KTZ21]	$\boldsymbol{2\Delta}$	$\boldsymbol{n = 5f - 1}$	**4**

green – optimal for BFT **red** – suboptimal **black** – optimal for this latency

[CL02] Castro, M., & Liskov, B. (2002). "Practical Byzantine fault tolerance and proactive recovery."
[MA06] Martin, J. P., & Alvisi, L. (2006). "Fast byzantine consensus."
[AGMM18] Abraham, I., et al. (2018). "Revisiting fast practical byzantine fault tolerance: Thelma, Velma, and Zelma."
[KTZ21] Kuznetsov, P., Tonkikh, A., & Zhang, Y. X. (2021). "Revisiting Optimal Resilience of Fast Byzantine Consensus."
(to appear in PODC 2021)

Conclusion

- Two important problems in Byzantine fault-tolerant distributed computing were addressed: **asynchronous reconfiguration** and the **resilience of fast partially-synchronous consensus**.
- For the first problem, there were **no known solutions prior to this work**.
  - The results were presented on DISC 2020.
- For the second problem, a **new upper bound** was proposed and a **matching lower bound** was proven.
  - The results are accepted to PODC 2021.