A Weak Consensus Algorithm and Its Application to High-Performance Blockchain

Qin Wang[2,4] , Rujia Li[1,3]

1 Southern University of Science and Technology, Shenzhen, China
2 Swinburne University of Technology, Melbourne, Australia
3 University of Birmingham, Birmingham, United Kingdom
4 HPB Foundation, DUO Tower, Singapore

May, 2021.

Consensus Algorithms

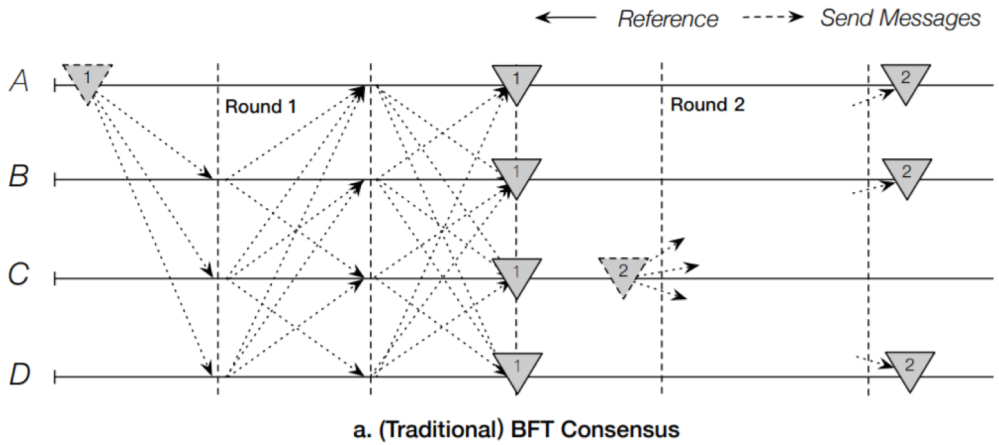
- Achieving consensus is a fundamental issue in distributed systems.
- Consensus algorithms play an important role in building fault-tolerant distributed systems
- Consensus algorithms are widely adopted in the blockchain system.

Consensus Targets

A consensus algorithm is a procedure through which all the peers of the distributed network reach a common agreement about the present state.

- Termination: All correct processes eventually have the decisions.
- Validity : if all correct nodes propose a valid transaction before starting a consensus instance, then the block decided in this instance is not empty;
- Agreement: All correct processes select the same proposal. Specifically, for a given consensus instance, if a correct node decides the block B, then all correct nodes decide B.

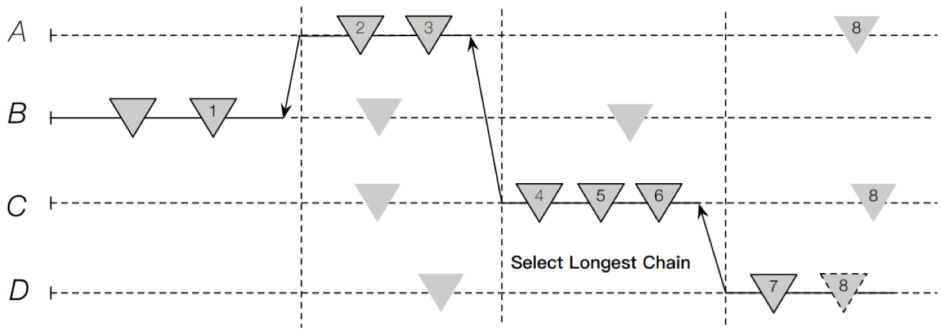
Byzantine Fault Tolerant Protocol



a. (Traditional) BFT Consensus

- The leader sends a proposal to replicas, and replicas distribute their replies.
- After receiving valid replies over the predefined threshold, a replica broadcasts his status (whether ready for the new state) to others.
- The decision is made once upon the received commit messages exceeds the threshold.

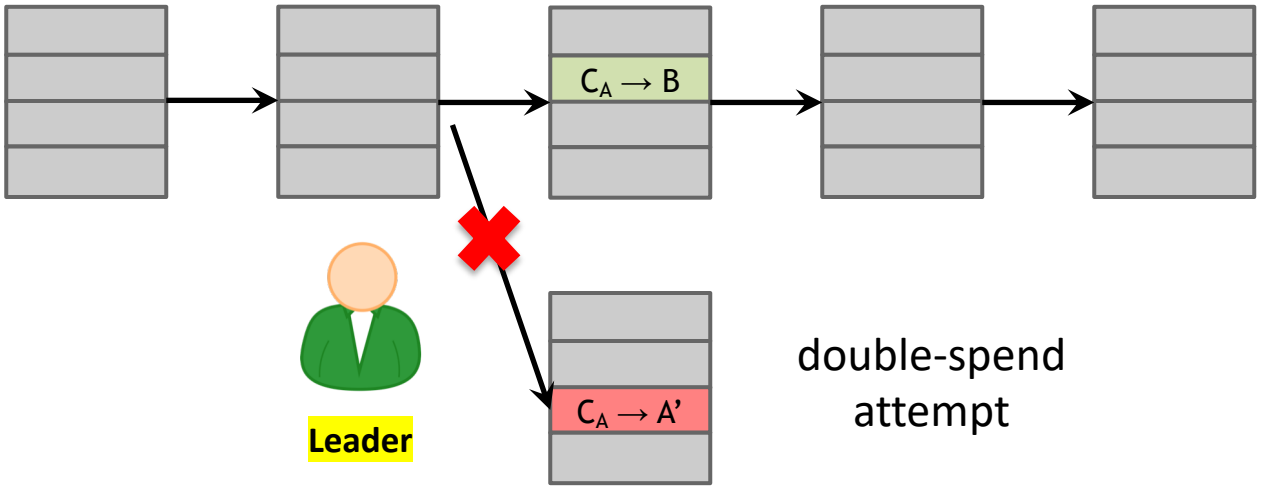
Nakamoto Consensus



b. Nakamoto Consensus

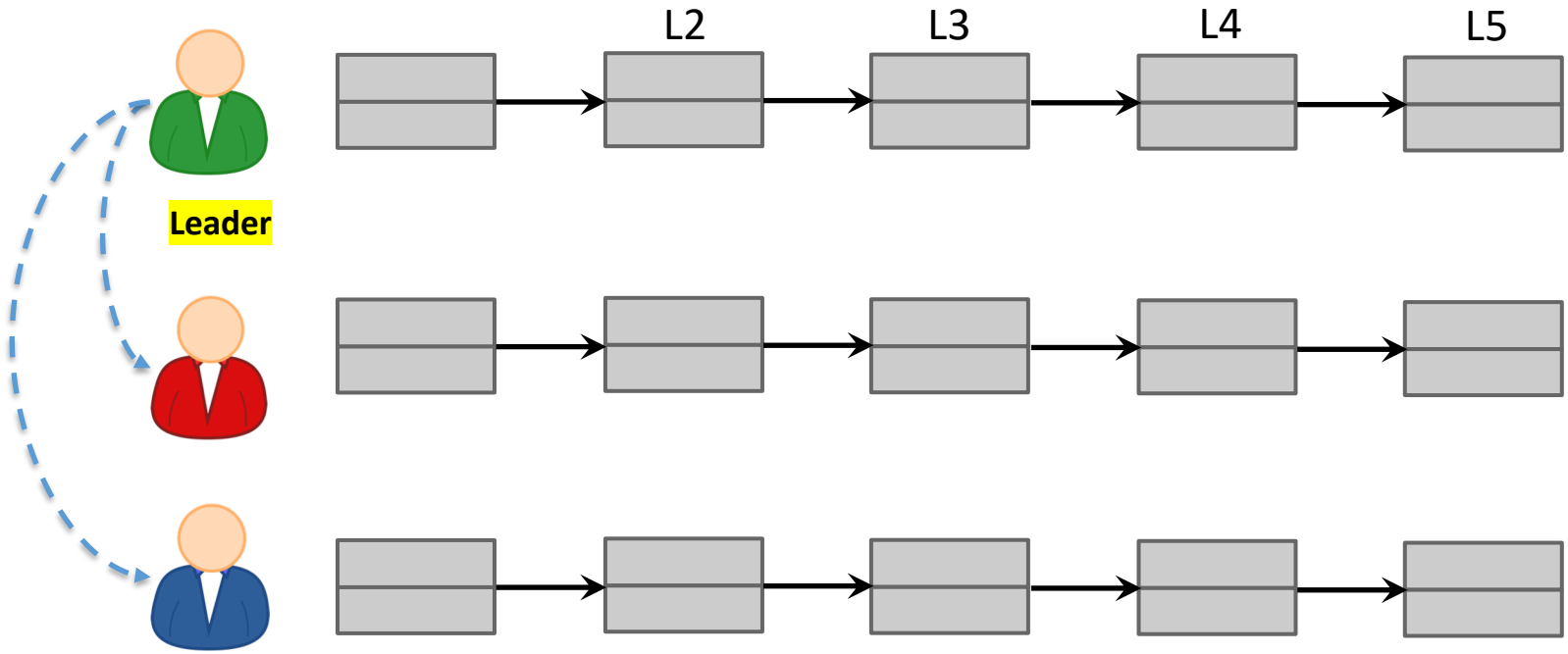
- Blocks generated by miners are randomly attached to their ancestors. Only the chain who has most descendants survives, whereas other competitive subchains are abandoned.
- The fastest miner receives the block reward, thus creating new Bitcoin, as well as an incentive to keep participating in the network.

Same Principle

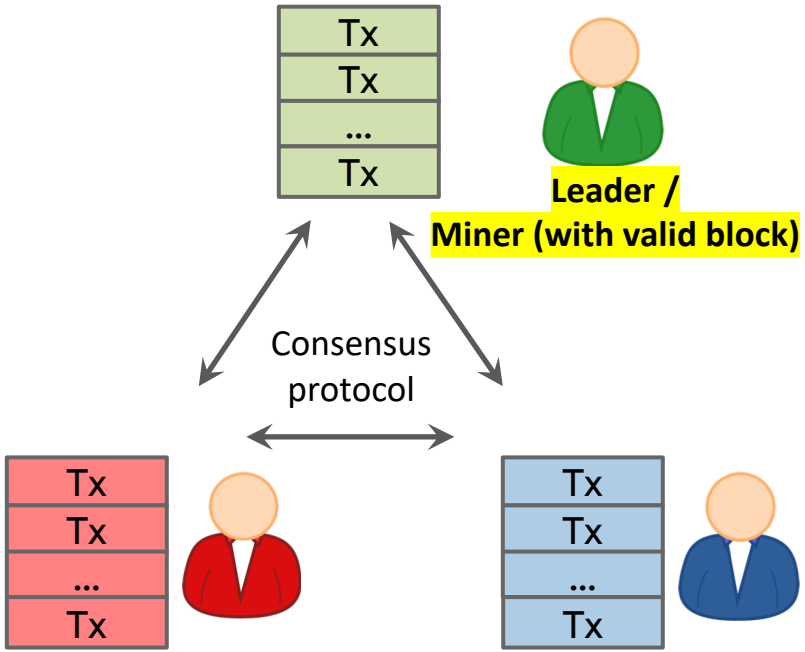


Only one block generated by the leader is deemed as confirmed at one round.

Same Principle

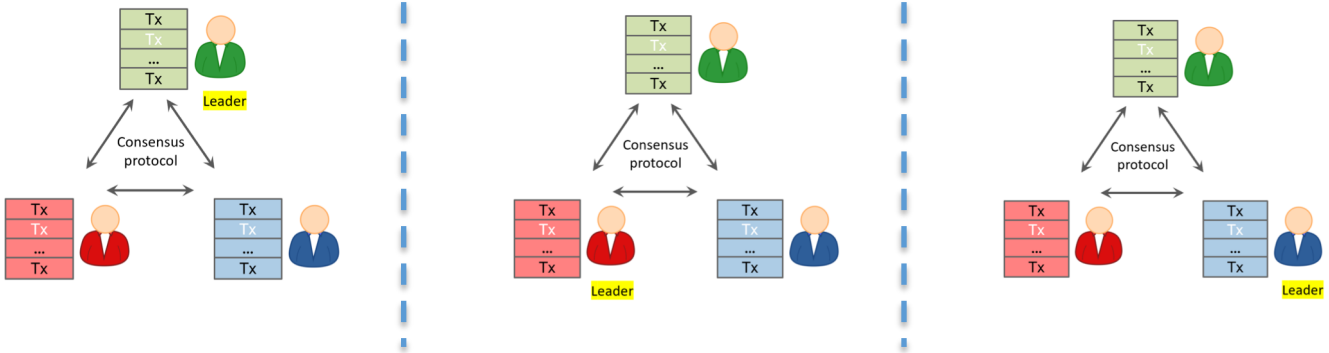


Performance and Scalability Issues



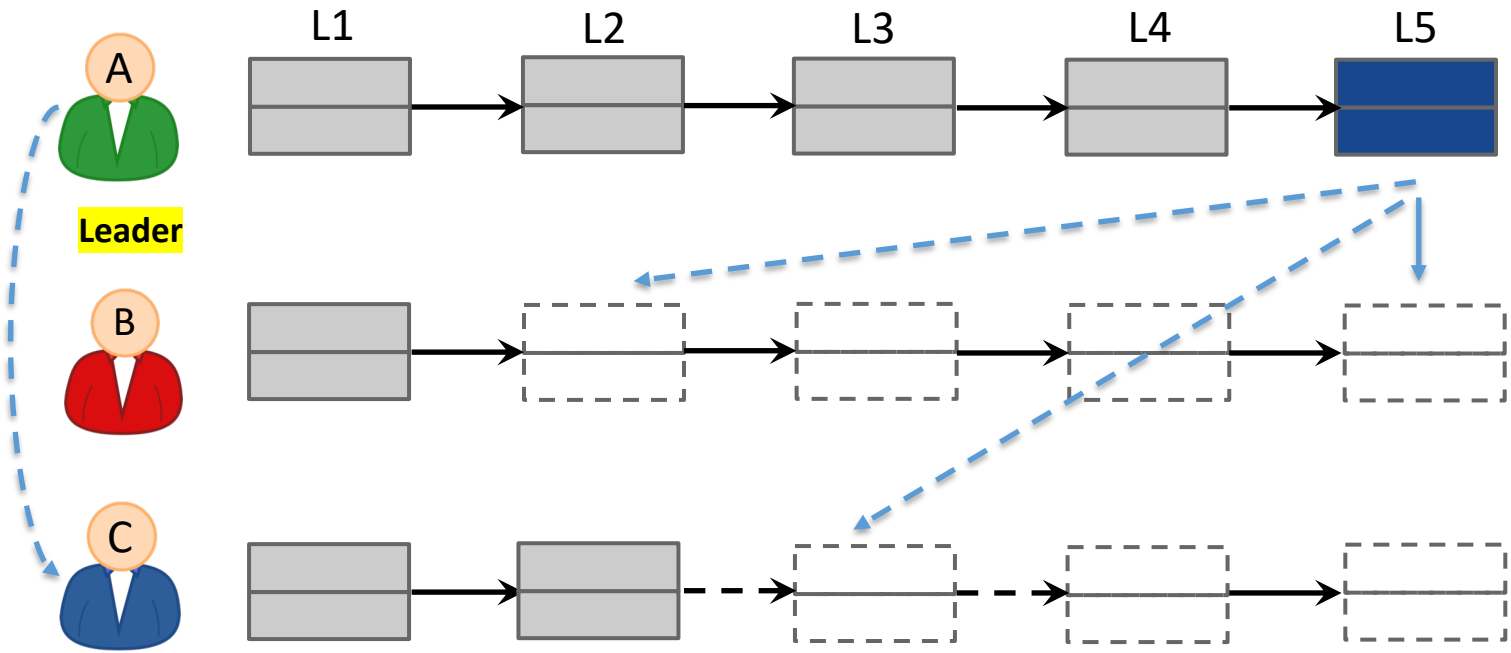
- In each round, only one node is granted and one block is generated.
- Increasing the number of the blockchain node is helpless for the scalability.
- Conflicts of competitive states greatly constrains their overall performance.

A Root Cause

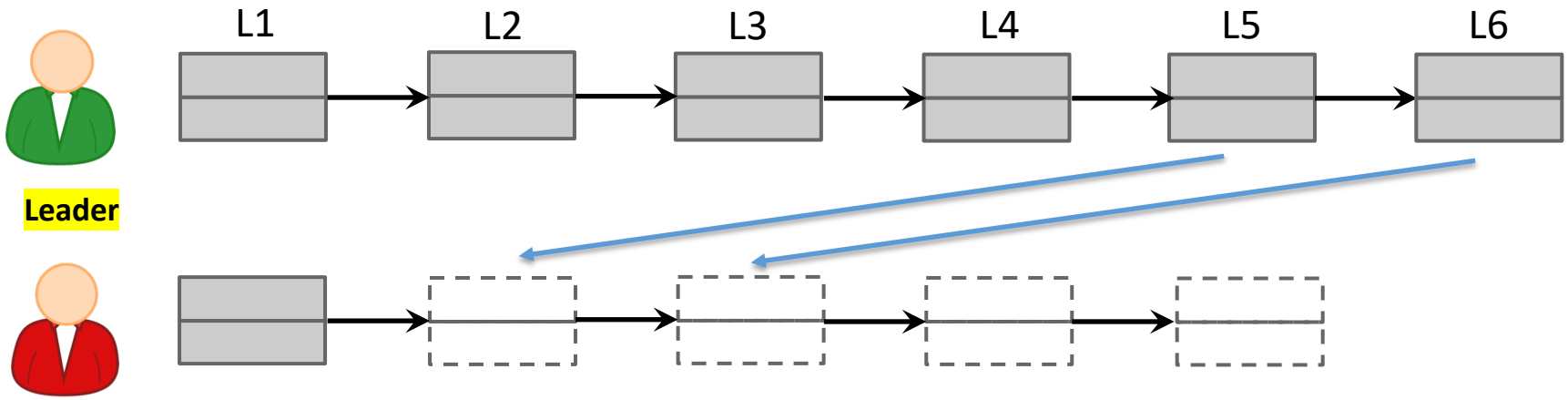


Only one leader is selected at each round.

Delayed Synchronization

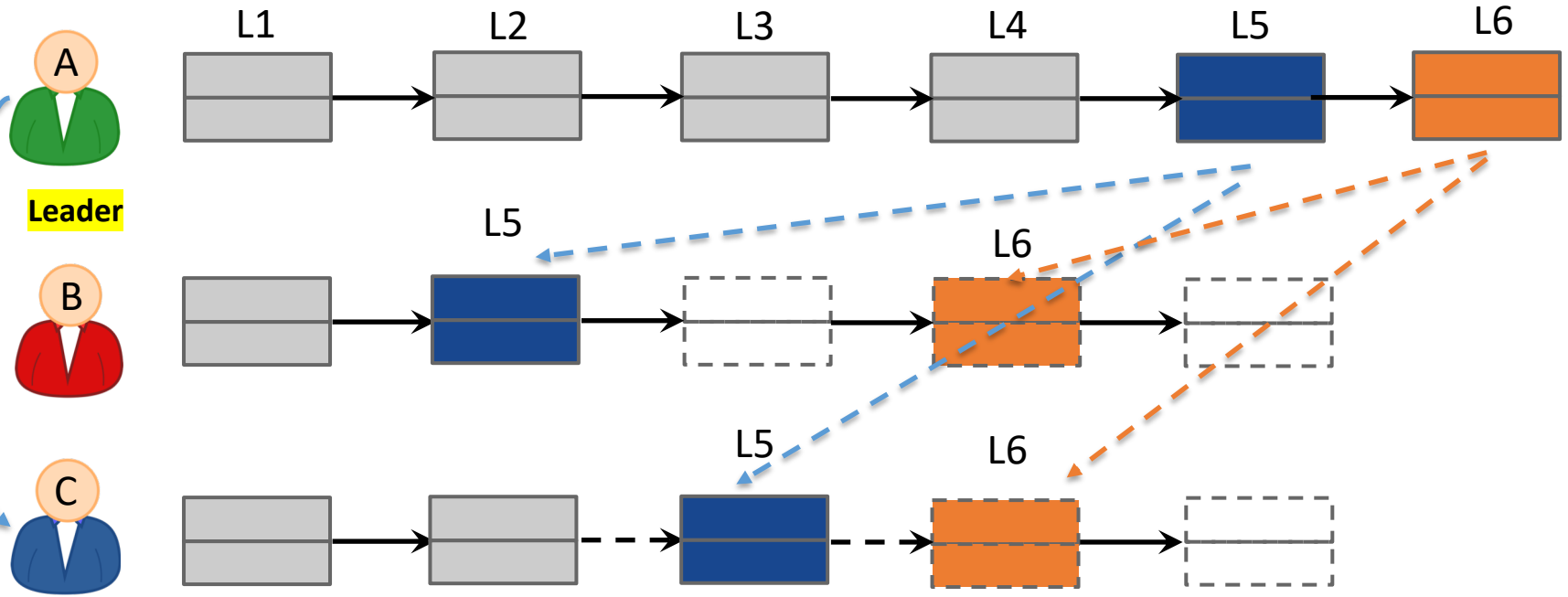


Delayed Synchronization



Is it possible to propose a consensus algorithm to improve the performance by weakening the guarantee of consistency?

Weaken the guarantee of consistency



Challenges

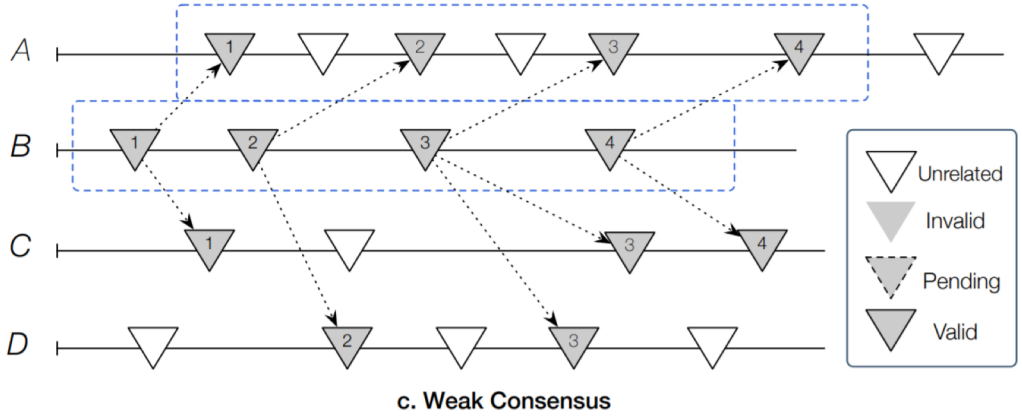
Disordered transactions, on the contrary, indicate ambiguous states where users may feel confused when invoking the blockchain service.

For example, Alice sends a transaction to Bob. If this transaction is stored in more than one block, Bob cannot know which position provides a valid transaction.

Exceptional Cases:

- Blockchain – based log system
- Blockchain – based certificate system

Weak Consensus Algorithm

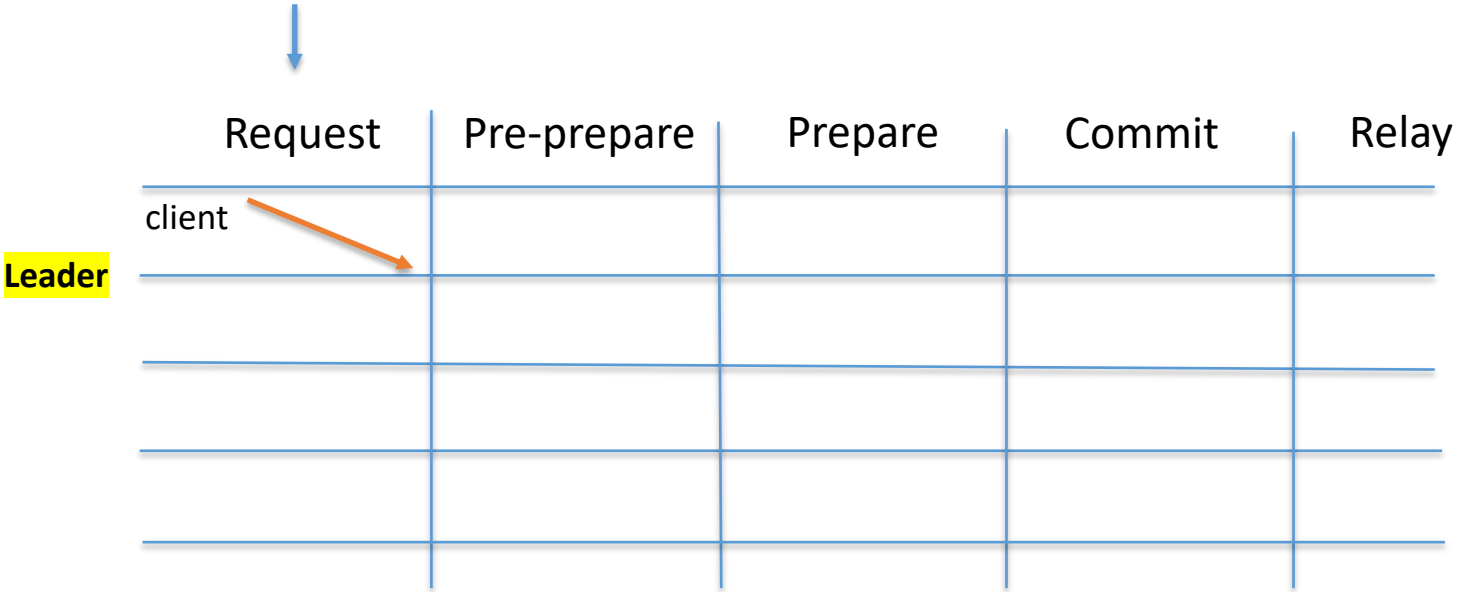


c. Weak Consensus

Ensuring that the sequence of (B1 → B2 → B3 → B4) can be correctly maintained across chains, *no matter how many blocks (generated by other nodes) are inserted between them.*

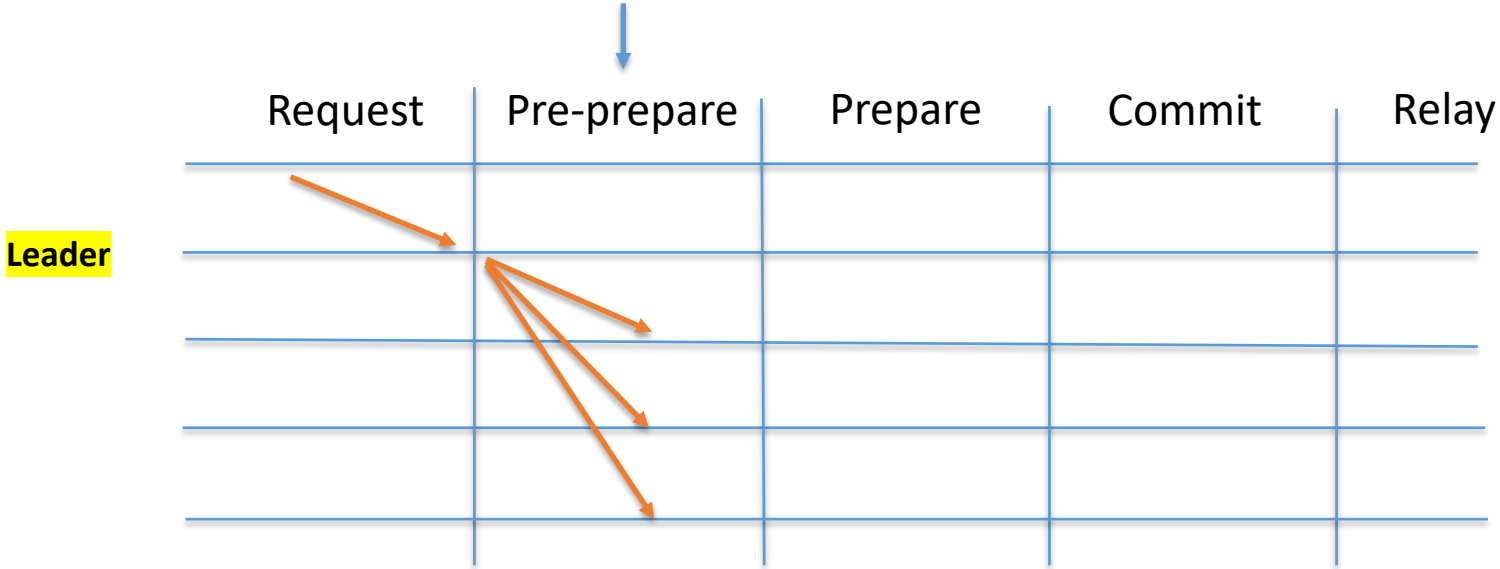
How Does Algorithm Work ?

- Client sends a request to any primary node.
- The node validates the message and propose a sequence number for it.



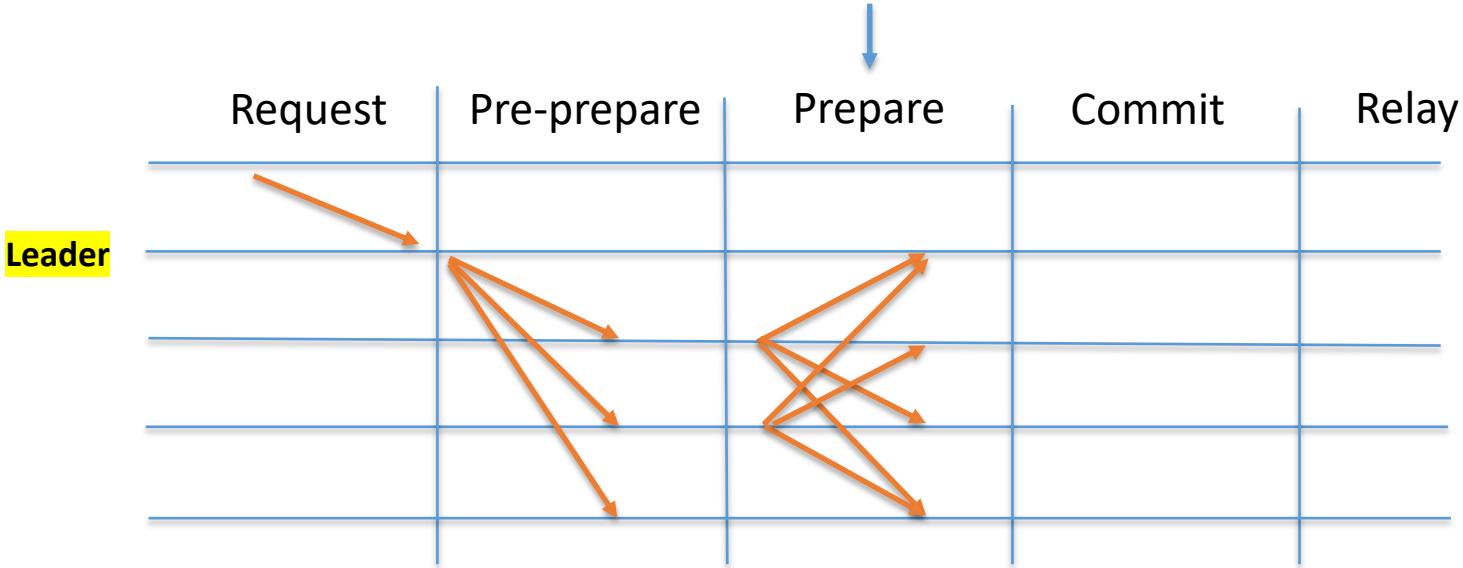
How Does Algorithm Work ?

- The primary puts the such messages into the local chain;
- The primary creates a Pre-prepare message;
- The primary broadcasts the signed message to peers.



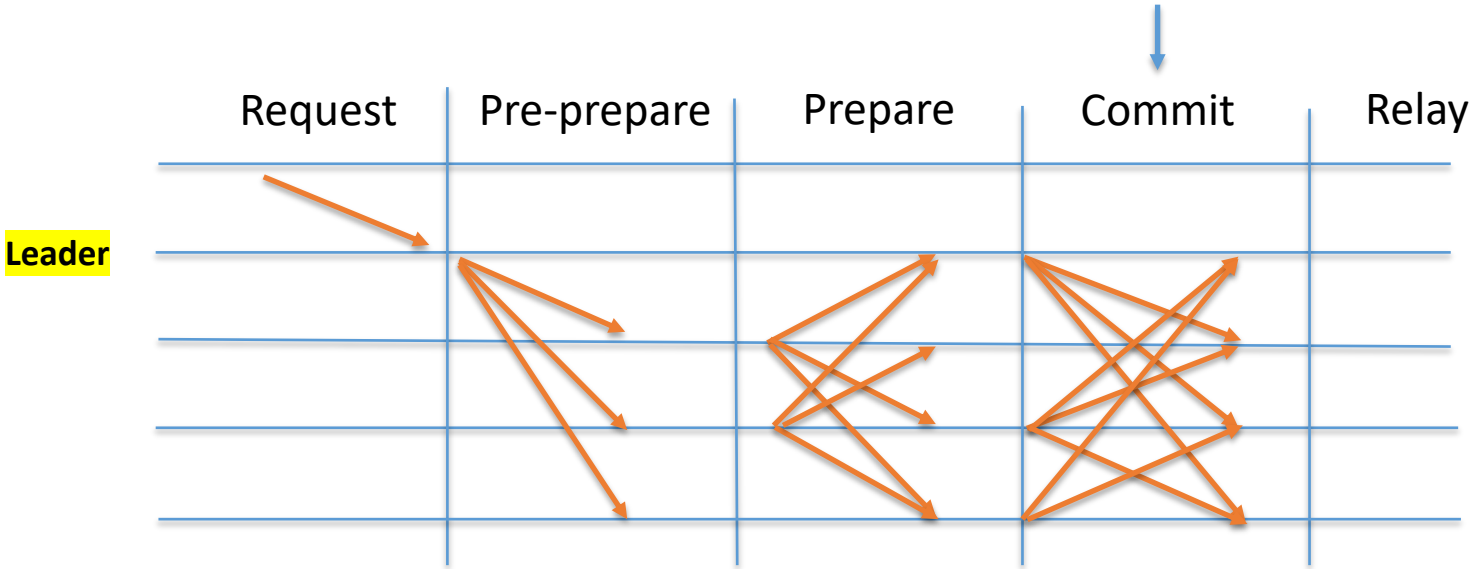
How Does Algorithm Work ?

- If passing the verification, the node updates his local-stored state,
- The node broadcasts the replied Prepare. Otherwise, the node aborts it.



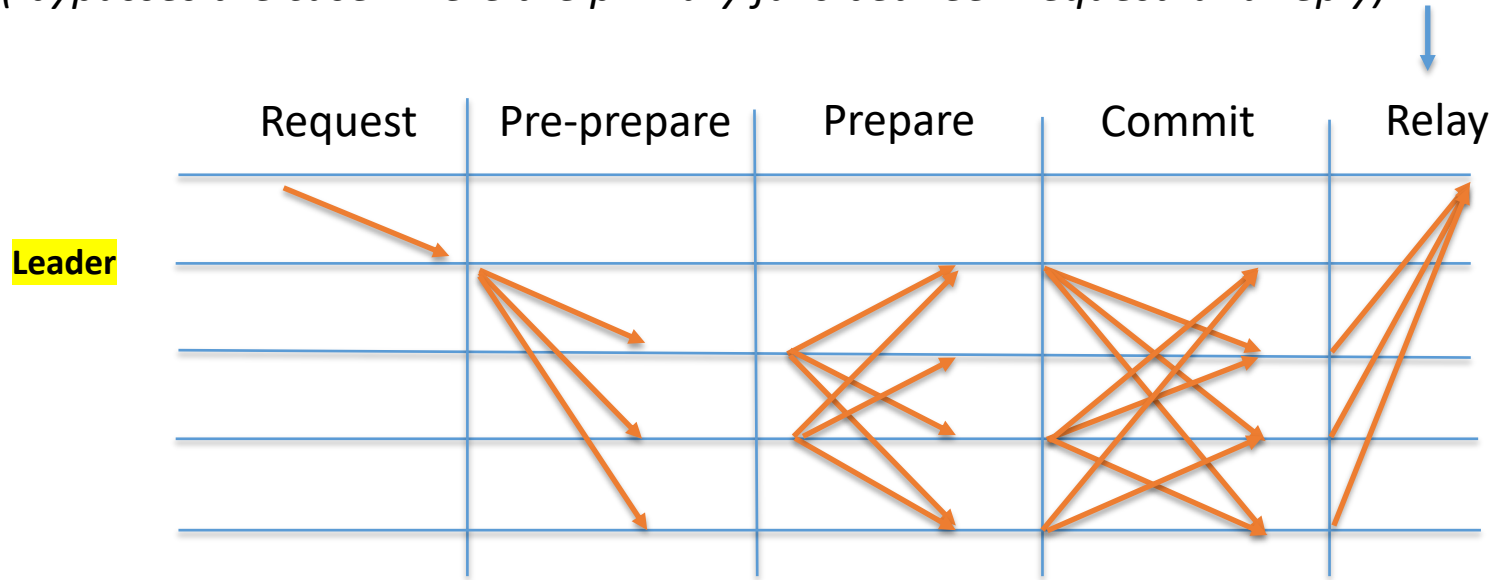
How Does Algorithm Work ?

- If exceeding 2f + 1 Prepare messages, the node broadcast a Commit message.



How Does Algorithm Work ?

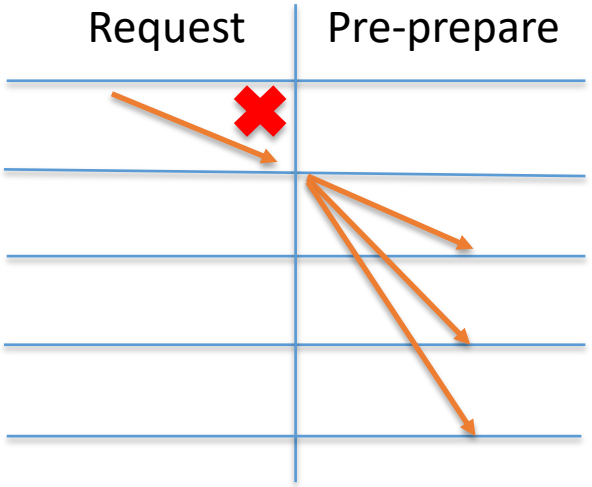
- When collecting more than 2f +1 Commit messages, the node transits the state and replies to clients with updated states.
  (bypasses the case where the primary fails between request and reply)



Complementary Mechanism

What if the node is crashed?

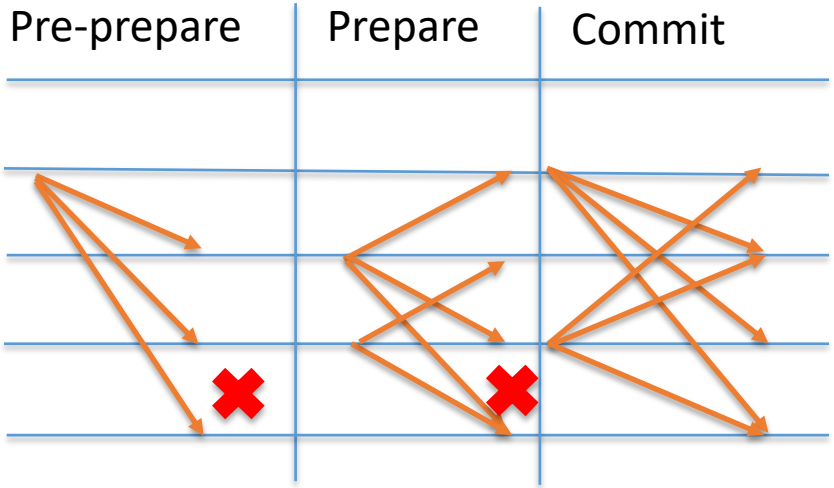
- The node uses a timeout mechanism. When this timeout expires, the request is resent to other replicas.



Complementary Mechanism

What if the nodes are (Byzantine) faulty?

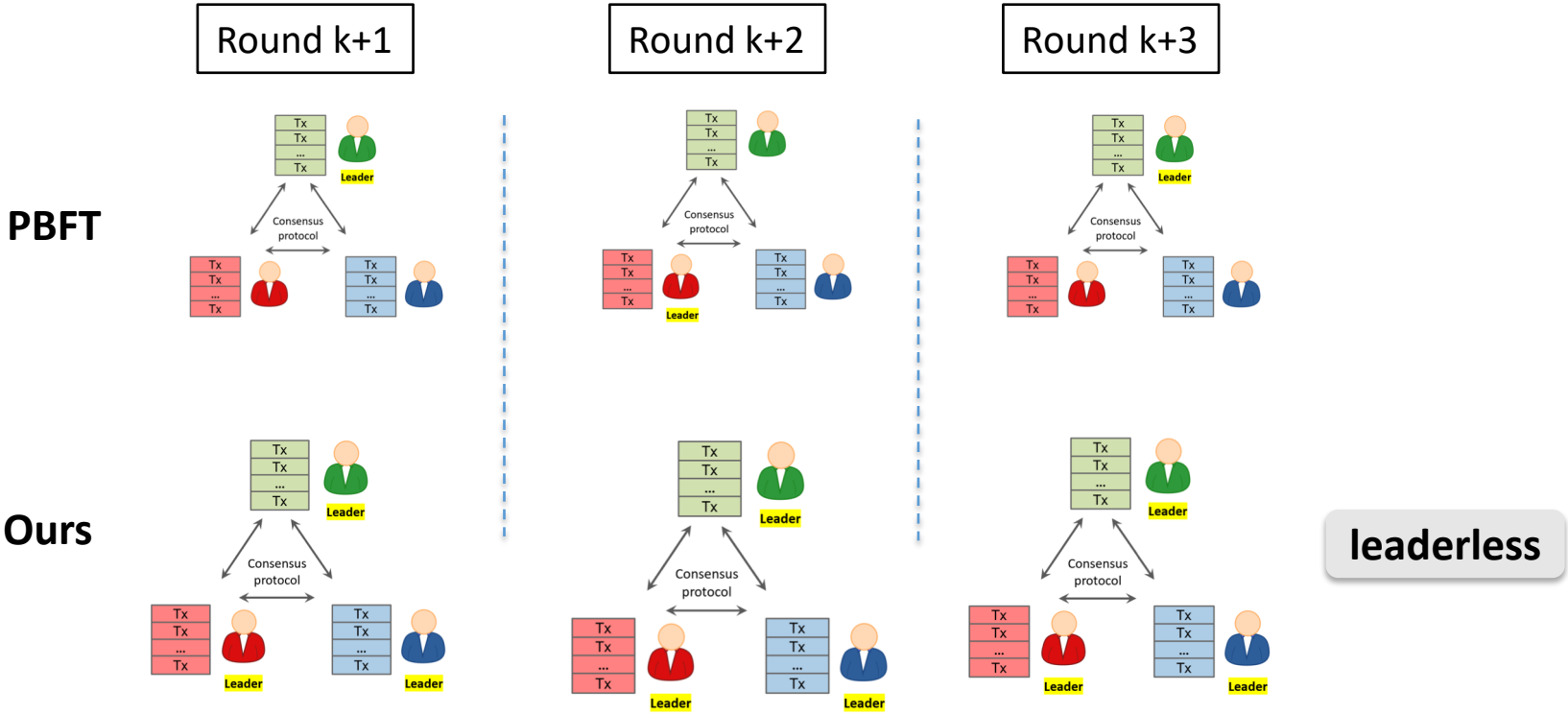
- If the message fails due to the lack of enough confirmation, the procedure of rebroadcast will be launched, and the counter increases each time of a retry.



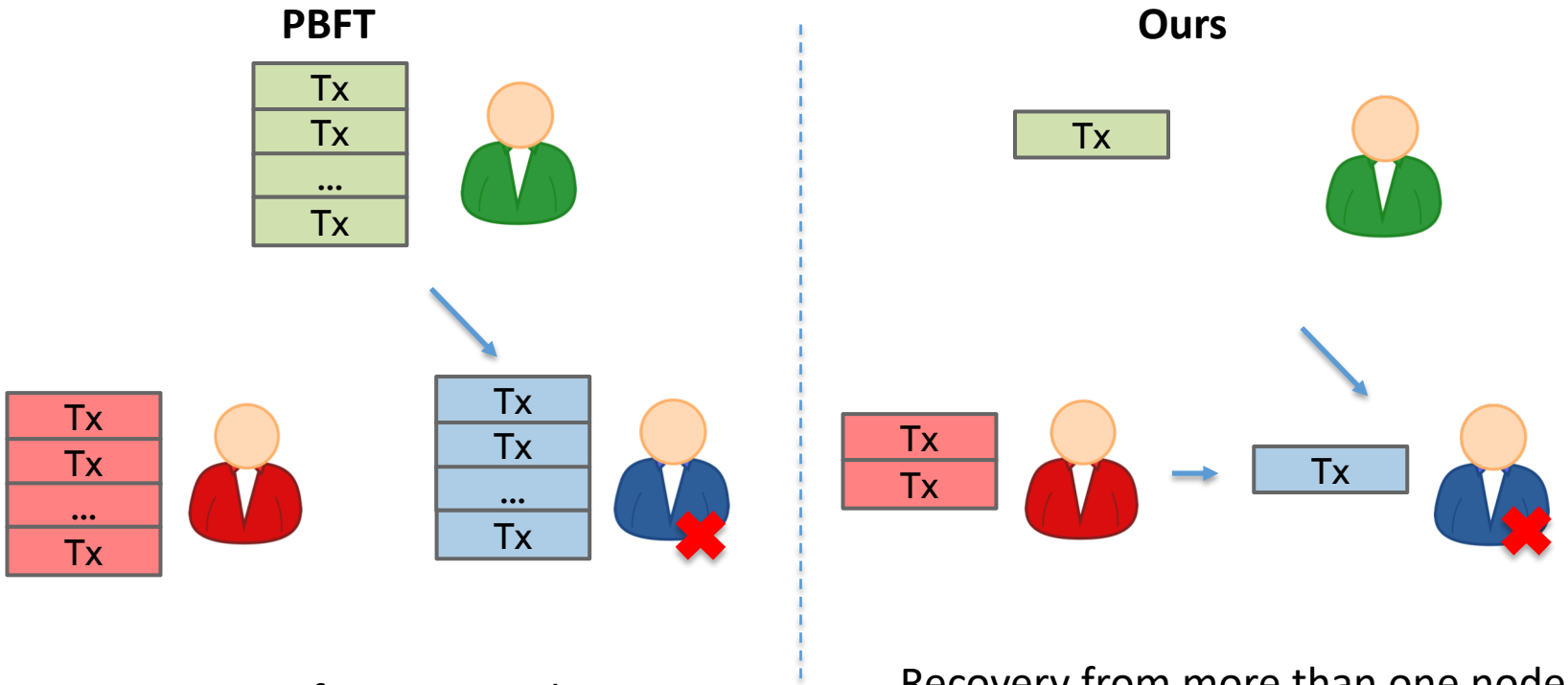
Differences with PBFT

Weak consistency	• weaken the assumption of strong consistency.
Leaderless	• asynchronously leaderless Byzantine agreement protocol. • independently proceed but mutually interact with each other.
State recovery	• collect states from a quorum of nodes, instead of a single one.

Differences with PBFT



State Recovery



Recovery from any nodes

Recovery from more than one node
(e.g. recovery from 2n/3 nodes)

Security Analysis

Relative persistence

If the relative position of two state y and x is accepted by the node N_i in iteration r and by the node N_j in r + 1, respectively, their decisions on the relationship are the same.

Liveness

If a correct relationship is committed, every honest node will eventually accept it.

Implementation & evaluation

Implementation

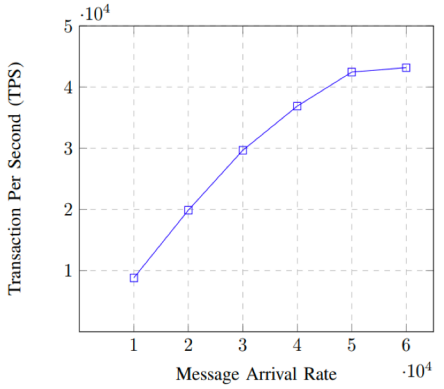
- Go language with 32,000+ lines of code
- Full functionalities
  - account configuration,
  - consensus mechanism,
  - peer to peer network,
  - user interface ….
- Deploy on 8 Dell R730 rack servers in a local cluster, with dual 2.1 GHz Opteron CPUs.
- OS: Ubuntu 16.04.1 LTS version

Implementation & Evaluation

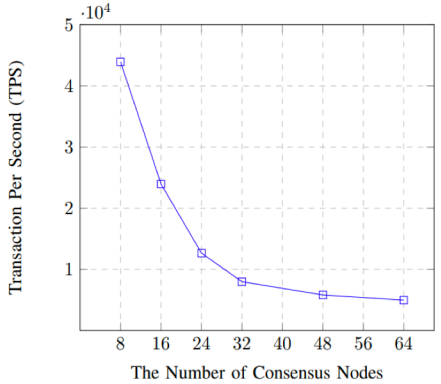
Performance

Nodes	8	16	32	64
Ethereum	355	311	268	91
Ours	43k	29k	10k	5k

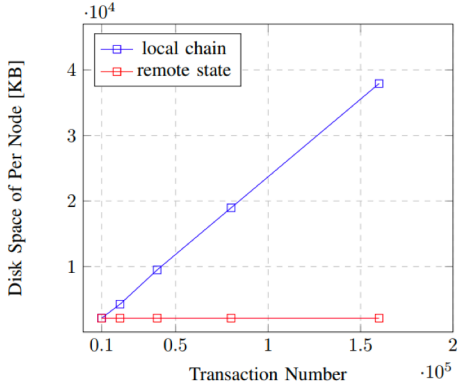
Scalability & Disk space



(a) TPS under Different Message Arrival Rates



(b) TPS under Different Size of Nodes



(c) Space Usage on Each Sphinx Node

Summary

- Identify the reasons of performance bottleneck.
- Propose a weak consensus algorithm.
- Apply it to blockchain system with high performance.
- Full implementation with evaluations:
  - peak value: 43K TPS under 8 full nodes.

References

- Vukolić, Marko. "Rethinking permissioned blockchains." Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. 2017.
- Bano, Shehar, et al. "SoK: Consensus in the age of blockchains." Proceedings of the 1st ACM Conference on Advances in Financial Technologies. 2019.
- Castro, Miguel, and Barbara Liskov. "Practical byzantine fault tolerance." OSDI. Vol. 99. No. 1999. 1999.
- Garay, Juan, Aggelos Kiayias, and Nikos Leonardos. "The bitcoin backbone protocol: Analysis and applications." Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2015.
- Bonneau, Joseph, et al. "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies." 2015 IEEE symposium on security and privacy. IEEE, 2015.
- Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.
- Bagaria, Vivek, et al. "Prism: Deconstructing the blockchain to approach physical limits." Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019.
- Wang, Qin, et al. "SoK: Diving into DAG-based Blockchain Systems." arXiv preprint arXiv:2012.06128 (2020).
- Fitzi, Matthias, et al. "Parallel Chains: Improving Throughput and Latency of Blockchain Protocols via Parallel Composition." IACR Cryptol. ePrint Arch. 2018 (2018): 1119.
- Kiffer, Lucianna, Rajmohan Rajaraman, and Abhi Shelat. "A better method to analyze blockchain consistency." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018.

Thanks

A Weak Consensus Algorithm and Its Application to High-Performance Blockchain

Qin Wang[2,4] , Rujia Li[1,3]
1 Southern University of Science and Technology, Shenzhen, China
2 Swinburne University of Technology, Melbourne, Australia
3 University of Birmingham, Birmingham, United Kingdom
4 HPB Foundation, DUO Tower, Singapore
May, 2021.