POLITECNICO
MILANO 1863

cini
Cybersecurity
National Lab

PBFT vs Proof-of-Authority:
Applying the CAP Theorem to Permissioned Blockchain

Instructor: Wei Zheng

Reporter: Xinbo Zhang

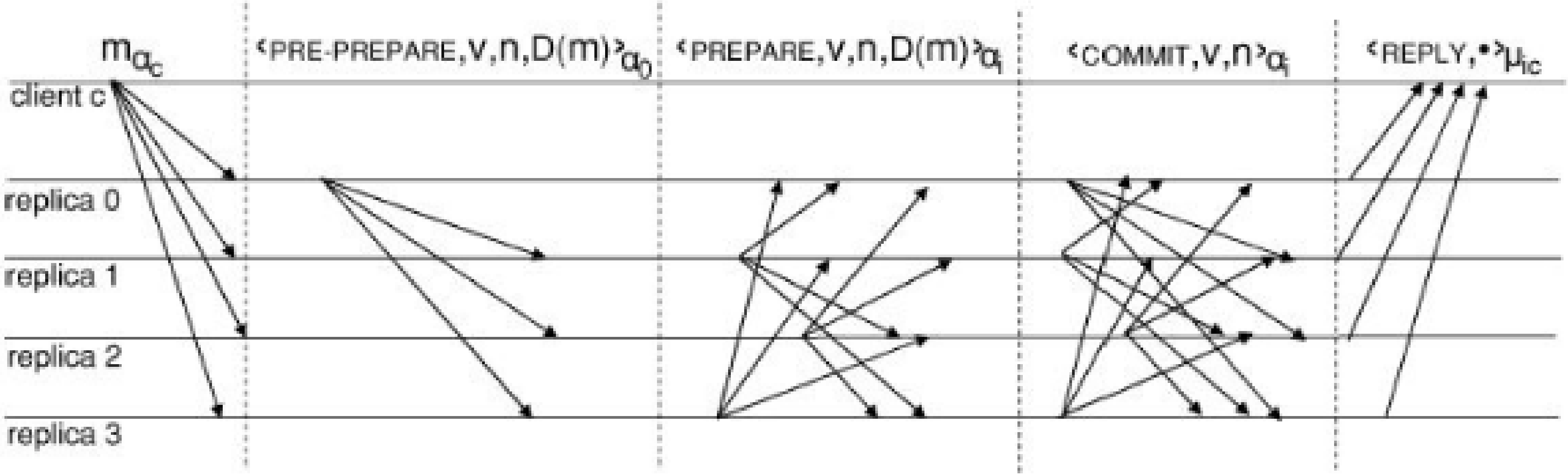
Author : Stefano De Angelis, Leonardo Aniello, Roberto Baldoni,
Federico Lombardi, Andrea Margheri, and Vladimiro Sassone

ITASEC18

ITALIAN CONFERENCE ON CYBERSECURITY

Outline:

- Review
  - Blockchain
  - Proof-of-Work
- PBFT Restatement
- Proof-of-Authority
  - Aura
  - Clique
- Comparison
  - CAP Theorem Analysis
  - Performance Analysis
- Conclusion
- Recap


m_{α_c}
⟨PRE-PREPARE,v,n,D(m)⟩$_{\alpha_0}$
⟨PREPARE,v,n,D(m)⟩$_{\alpha_i}$
⟨COMMIT,v,n⟩$_{\alpha_i}$
⟨REPLY,•⟩$_{\mu_{ic}}$
client c
replica 0
replica 1
replica 2
replica 3

Review: Blockchain

- Blockchain is one of the most disruptive technologies of recent years, firstly appeared as a decentralized public ledger for the Bitcoin cryptocurrency.
- Blockchain is a linked data structure replicated over a peer-to-peer network, where transactions are issued to form new blocks.
- Blockchain systems like Bitcoin are called permissionless (public), any node on the Internet can join and become a miner.

Review: Proof of Work

- Distributed consensus is achieved via Proof of Work (PoW), a computational intensive hashing-based mathematical challenge.
- Came at a huge cost: Lacking of performance.
- Together with the absence of privacy and security controls on data, this has led to the so-called permissioned blockchain.
- An additional authentication and authorization layer on miners is in place.

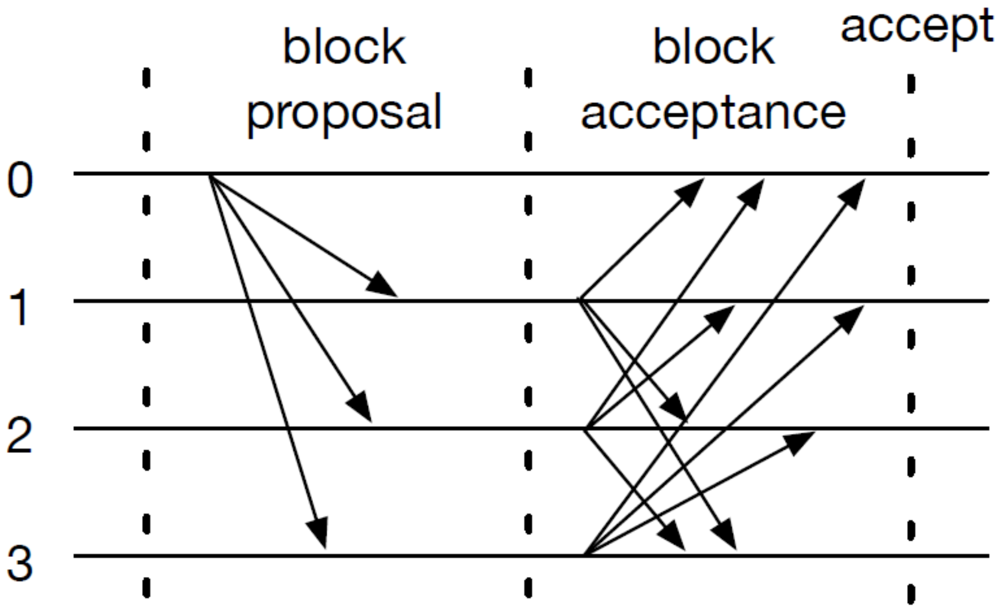
PBFT Restatement:

- Being the operating environment more trusted, permissioned blockchains rely on message-based consensus schema, rather than on hashing procedures.
- In such setting, dominant candidates are Byzantine fault tolerant (BFT) algorithms such as the Practical BFT (PBFT).
- Indeed, BFT-like algorithms have been widely investigated for permissioned blockchains with the aim of outperforming PoW while ensuring adequate fault tolerance.

Proof of Authority:

- For permissioned blockchain
- Authorities: N
- Honest nodes: N/2 + 1
- Consensus in PoA algorithms relies on a mining rotation schema
- Time is divided into steps, each of which has an authority elected as mining leader
- The authorities run a consensus to order the transactions issued by clients
- PoA was originally proposed as part of the Ethereum ecosystem for private networks and implemented into the clients Aura and Clique

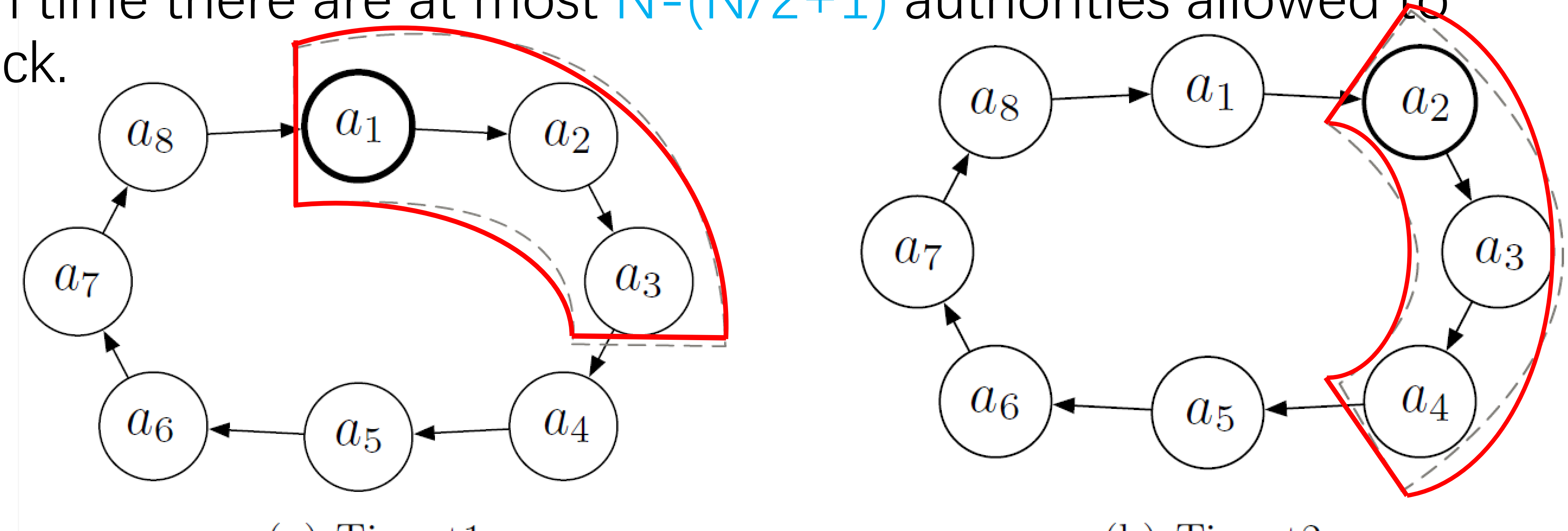
Proof of Authority: Aura



- Aura (Authority Round)
- Mining rotation schema:
  - s = t/step_duration
  - l = s mod N
- Block proposal:
  - The leader l includes the transactions in a block b, and broadcasts it to the other authorities
- Block acceptance:
  - Then each authority sends the received block to the others
- If authorities do not agree on the proposed block during the block acceptance, a voting is triggered.

Proof of Authority: Clique

- Mining rotation schema:
- Clique computes the current step and related leader using a formula that combines the block number and the number of authorities.
- Each authority is only allowed to propose a block every N/2+1 blocks.
- At any point in time there are at most N-(N/2+1) authorities allowed to propose a block.

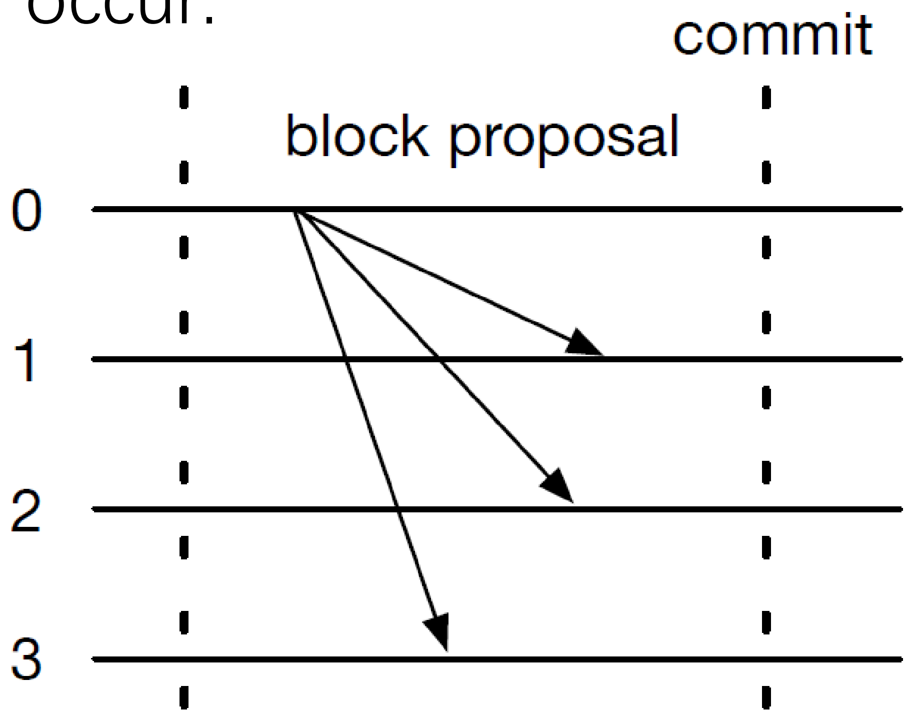


(a) Time t1

(b) Time t2

Proof of Authority: Clique

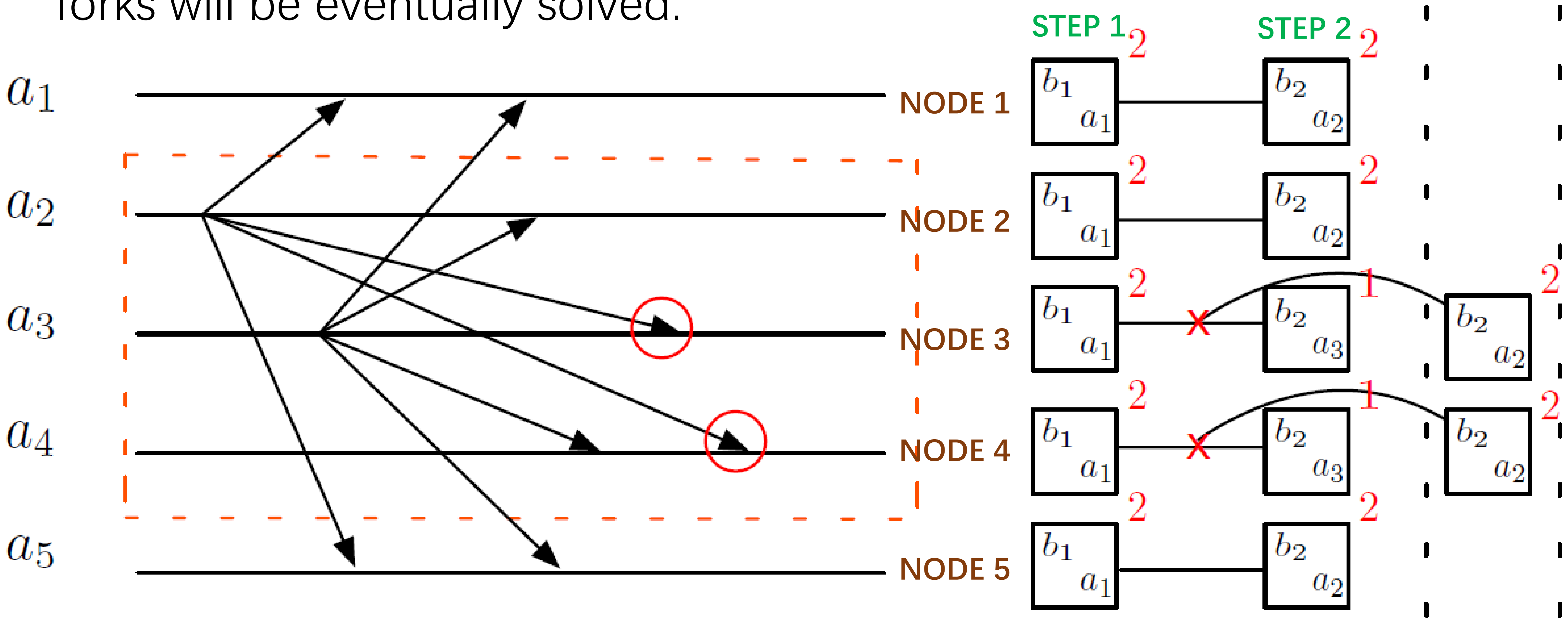
- At each step the leader broadcasts a block and all the authorities directly commit it to the chain.
- As more authorities can propose a block during each step, forks can occur.



- However, fork likelihood is limited by the fact that each non-leader authority proposing a block delays its block by a random time, hence the leader block is likely to be the first received by all the authorities.

Proof of Authority: Clique

- If forks happen, the GHOST protocol is used, which is based on a block scoring approach: leaders' blocks have higher scores, thus ensuring that forks will be eventually solved.

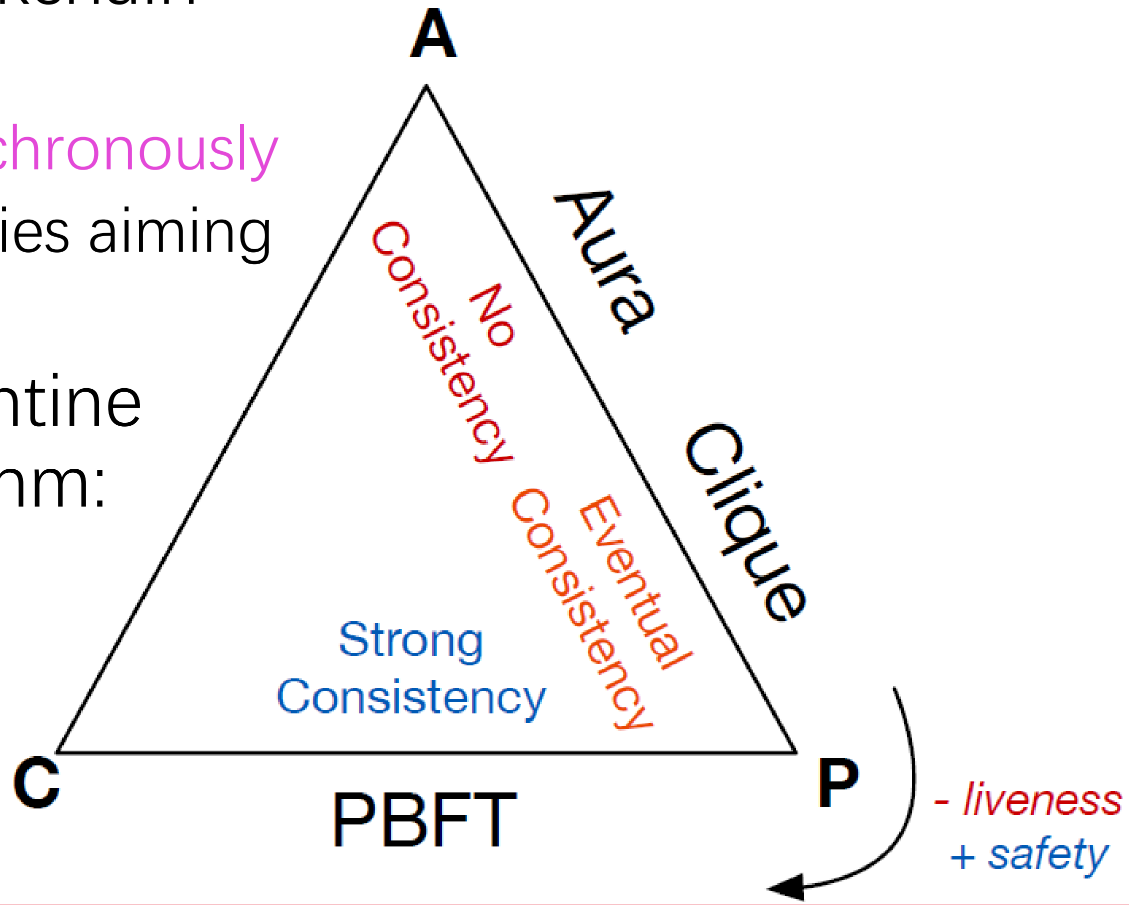


Comparison: CAP Theorem

- The CAP Theorem states that in a distributed data store only two out of the three following properties can be ensured:
- Consistency (C)
    - A blockchain achieves consistency when forks are avoided, corresponds to achieving the total order and agreement properties of atomic broadcast.
    - When consistency cannot be obtained, it divided 2 scenarios eventual consistency or no consistency.
- Availability (A)
    - A blockchain is available if transactions submitted by clients are served and eventually committed.
- Partition Tolerance (P)
    - When a network partition occurs, authorities are divided into disjoint groups in such a way that nodes in different groups cannot communicate each other.

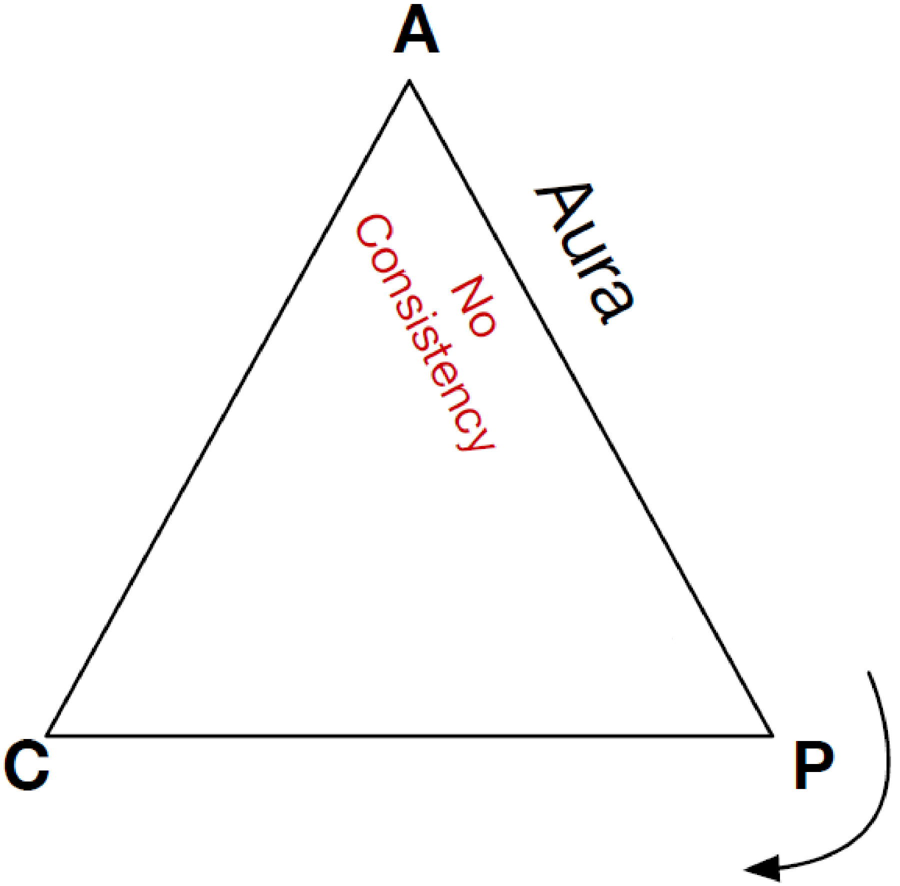
Comparison: CAP Theorem

- Since a blockchain must tolerate partitions, hence CA option is not considered, we analyze the algorithms with respect to CP and AP options.
- An Internet-deployed permissioned blockchain has to tolerate these adverse situations
  - (i) periods where the network behaves asynchronously
  - (ii) a bounded number of Byzantine authorities aiming at hampering availability and consistency.
- The maximum number of tolerated Byzantine nodes depends on the consensus algorithm:
  - N/2 for PoA
  - N/3 for PBFT.



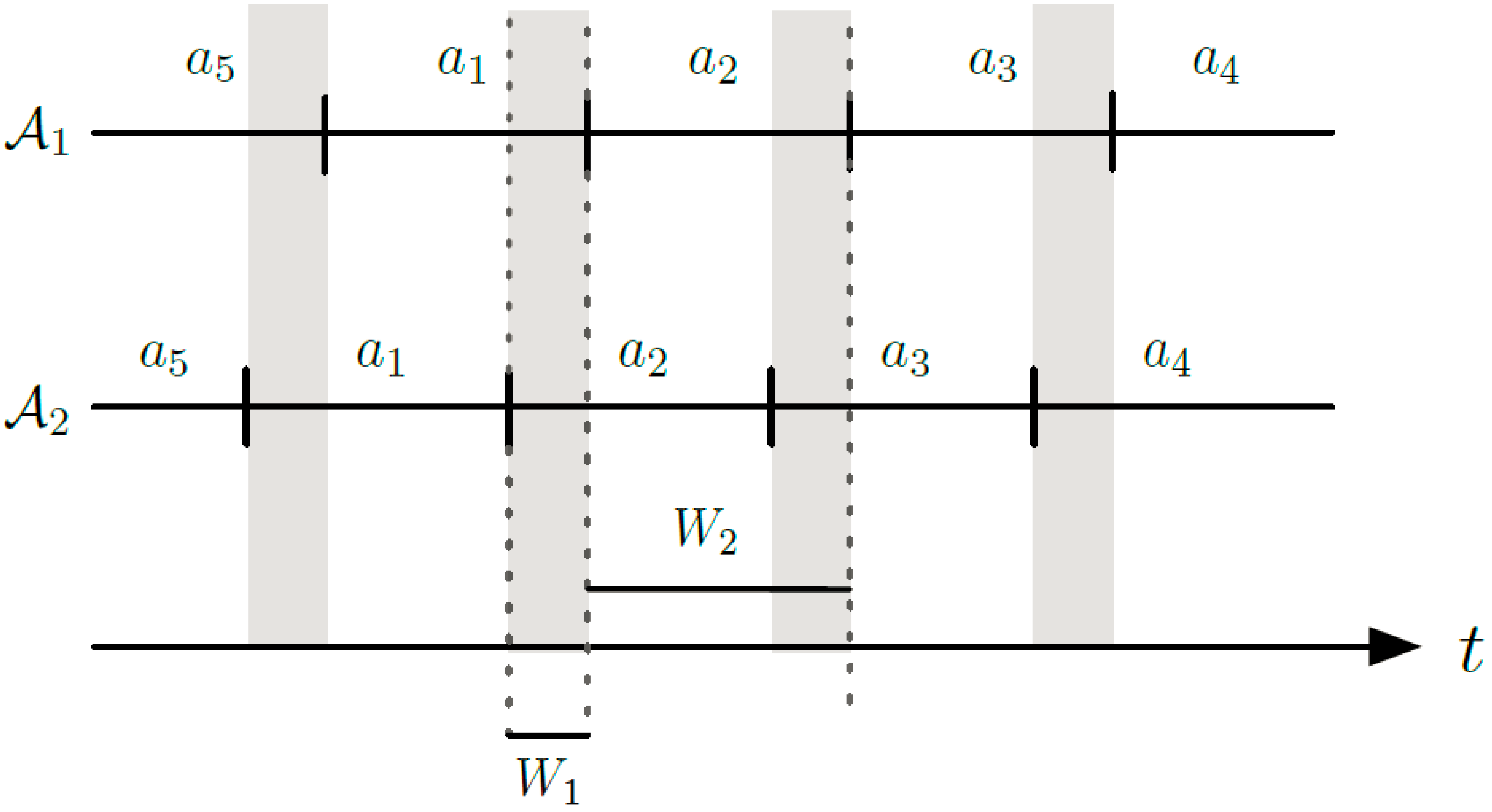
Comparison: CAP Theorem of Aura

- Aura: Being based on UNIX synch time, authorities' clocks can drift and become out-of-synch.



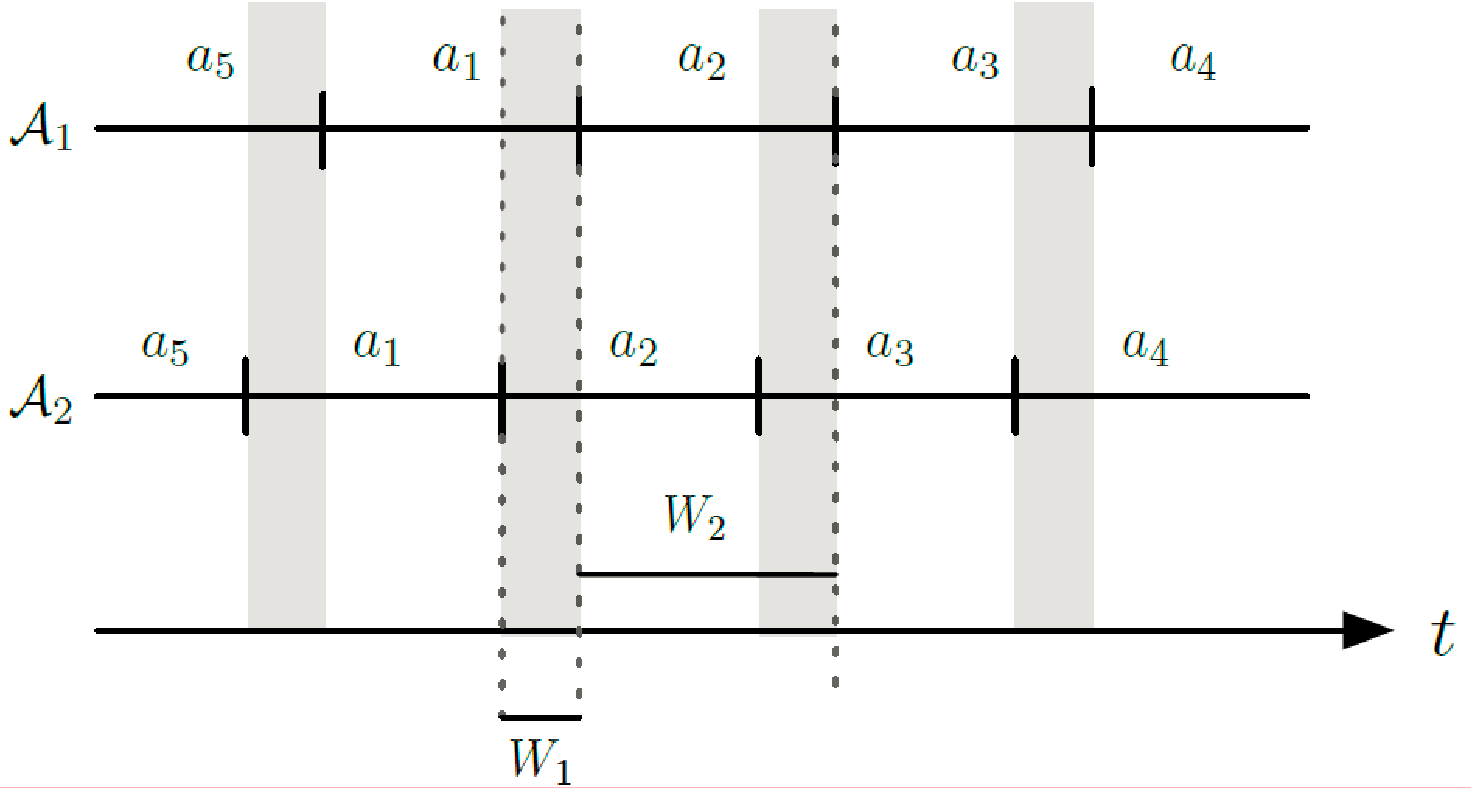
- *liveness*

+ *safety*



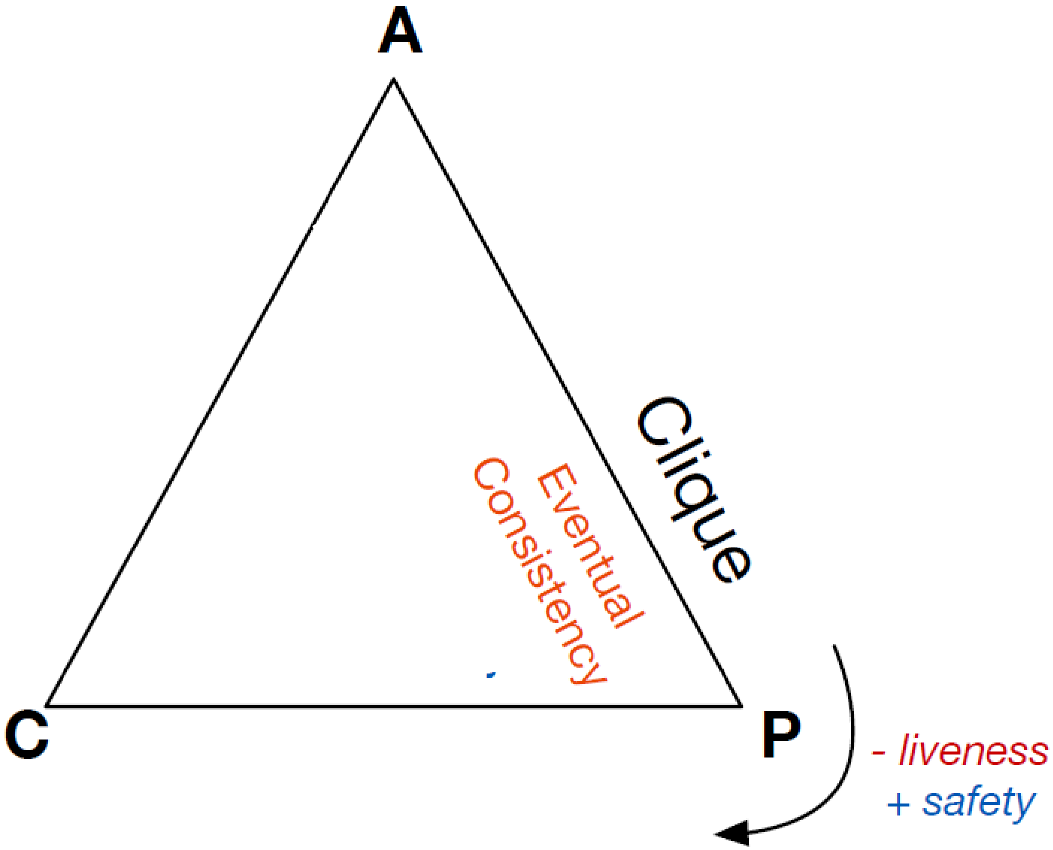
Comparison: CAP Theorem of Aura

- N1 = |A1| and N2 = |A2| (N1+N2==N && N%2!=0) are the number of authorities in the two sets.
- Authorities in A1 = {a1, a3, a5} is later than A2 = {a2, a4}.
- The grey time windows: A1 disagrees with A2 on who is the current leader.



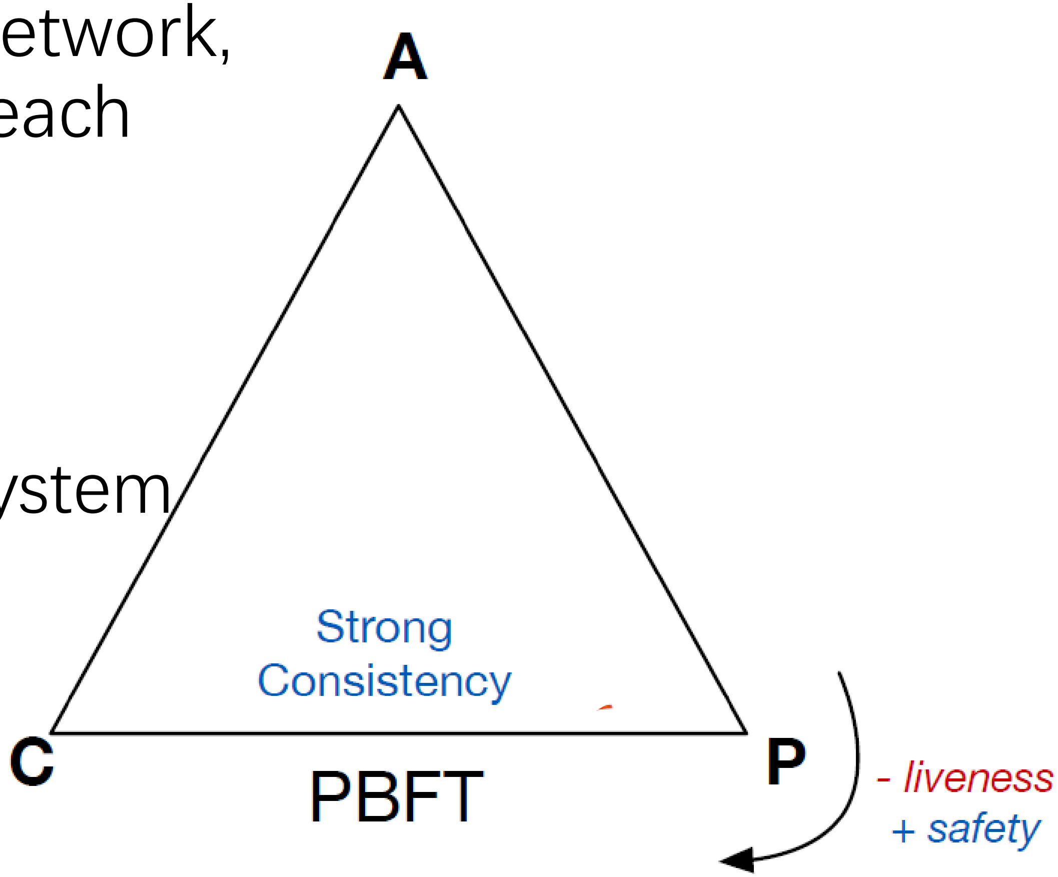
Comparison: CAP Theorem of Clique

- Clique:
- By design, Clique allows more than one authority to propose blocks with random delays.
- This permits coping with leaders that could not have sent any block due to either network asynchrony or benign/Byzantine faults.
- Resulting forks are anyway resolved by the GHOST protocol, hence we have eventual consistency.
- This PoA algorithm can thus be classfied as AP, with eventual consistency guarantees.



Comparison: CAP Theorem of PBFT

- PBFT:
- As long as less than 1/3 of nodes are Byzantine, PBFT has been proved to guarantee consistency, i.e. no fork can occur.
- Because of the eventual synchrony of the network, the algorithm can stall and blocks cannot reach finality.
- In this case, consistency is preserved while availability is given up.
- PBFT can then be easily classified as a CP system according to the CAP theorem.

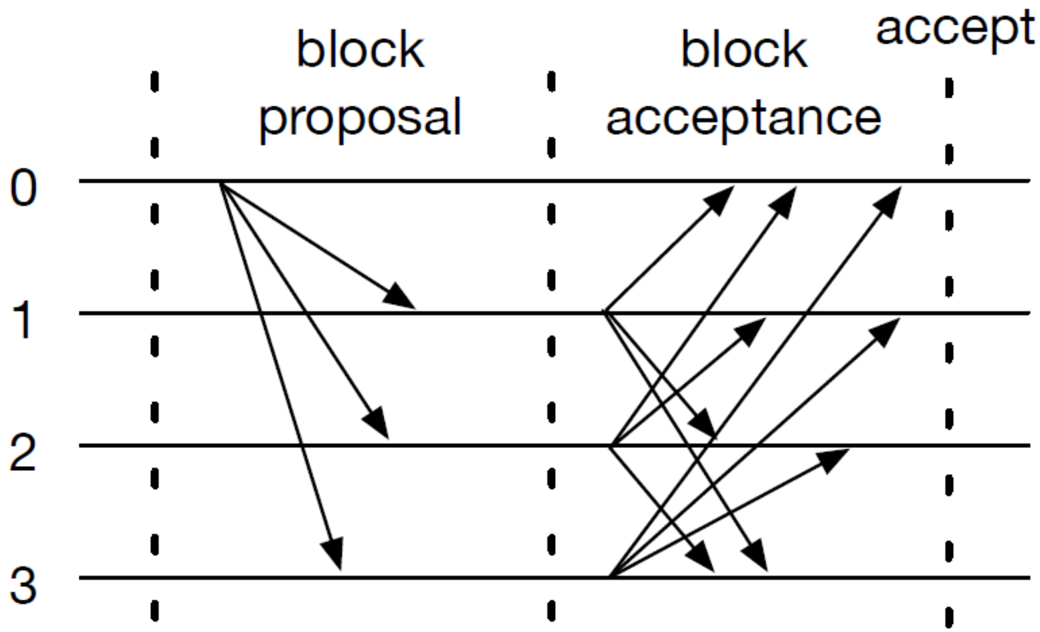


Comparison: Performance Analysis

- The analysis here reported is qualitative and only based on how the consensus algorithms work in terms of message exchanging.
- The performance metrics usually considered for consensus algorithms are transaction latency and throughput.
- Latency:
  - In the specific case of permissioned blockchains, we measure the latency of a transaction t as the time between the submission of t by a client and the commit of the block including t.
  - We can compare the algorithms in terms of the number of message rounds required before a block is committed.
- Throughput:
  - Future work.

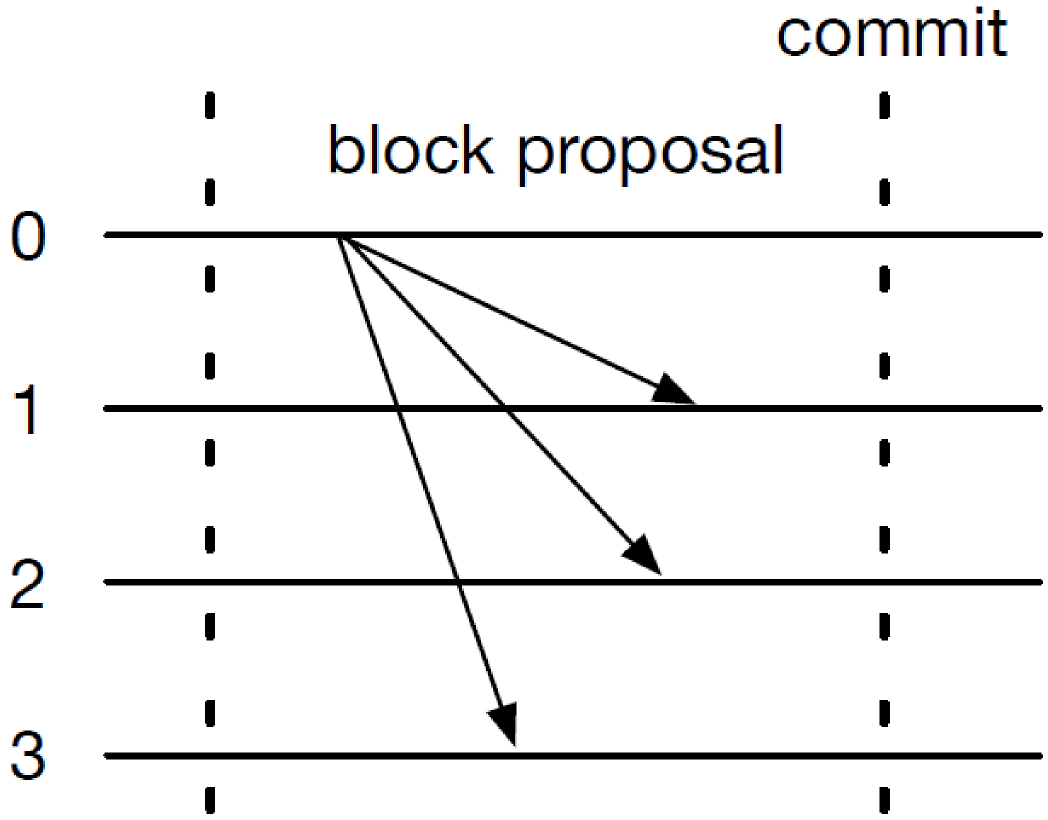
Comparison: Performance Analysis

- In Aura, each block proposal requires 2 message rounds:
  - In the 1st round the leader sends the proposed block to all the other authorities
  - In the 2nd round each authority sends the received block to all the other authorities.
- A block is committed after a majority of authorities have proposed their blocks, hence the latency in terms of message rounds in Aura is 2(N/2+1) (N is the number of authorities).



Comparison: Performance Analysis

- In Clique, a block proposal consists of a single round, where the leader sends the new block to all the other authorities.
- The block is committed straight away, hence the latency in terms of message rounds in Clique is **1**.



Conclusion:

- PoA algorithms can give up consistency for availability when considering the presence of Byzantine nodes.
- PBFT keeps the blockchain consistent at the cost of availability, this behaviors is much more desirable when data integrity is a priority.
- Despite one of the most praised advantages of PoA algorithms are their performance, the qualitative analysis shows that in terms of latency the expected loss of PBFT is bounded, and can be offset by the gain in consistency guarantees.

Recap:

- Analyzed 2 of the main PoA algorithms
  - Aura & Clique
- Weighted by the CAP theorem
  - Consistency
  - Availability
  - Partition tolerance guarantees
- Reported a qualitative latency analysis
- PoA for permissioned blockchains, deployed over the Internet with Byzantine nodes, do not provide adequate consistency guarantees for scenarios where data integrity is essential.
- PBFT can fit better such scenarios (strong consistency), despite a limited loss in terms of performance.

Q&A:

- Thanks for your listening!