BLOCKCHAINS

ARCHITECTURE, DESIGN AND USE CASES

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

PRAVEEN JAYACHANDRAN
IBM RESEARCH,
INDIA

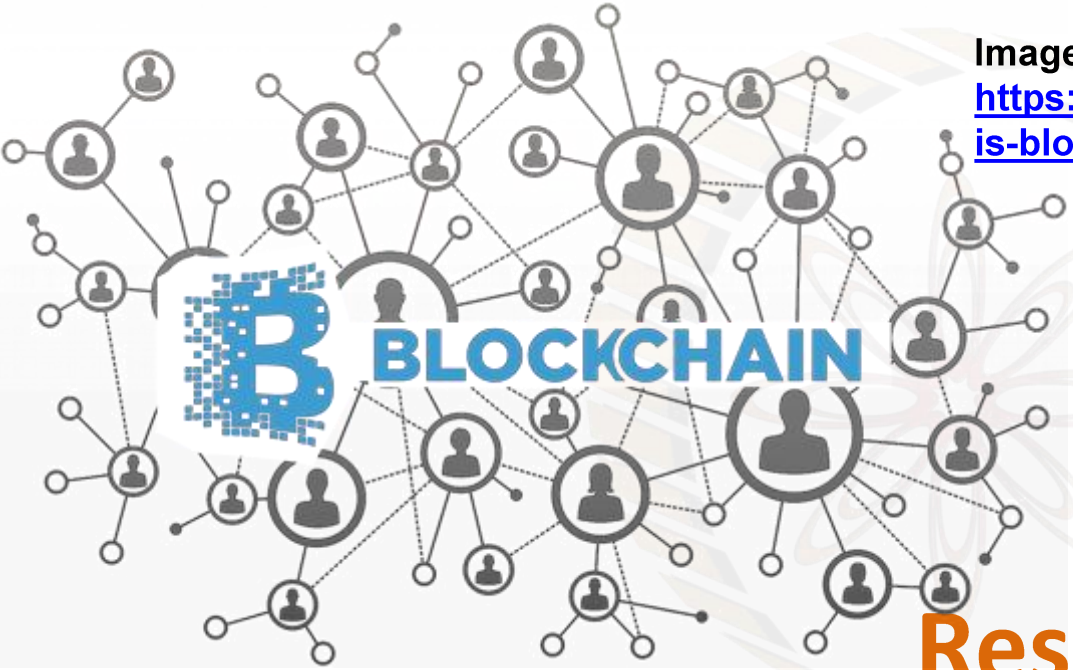


Image Source:
https://steemit.com/blockchain/@tariq3njr/what-is-blockchain-technology

Research Aspects - IV
ByzCoin

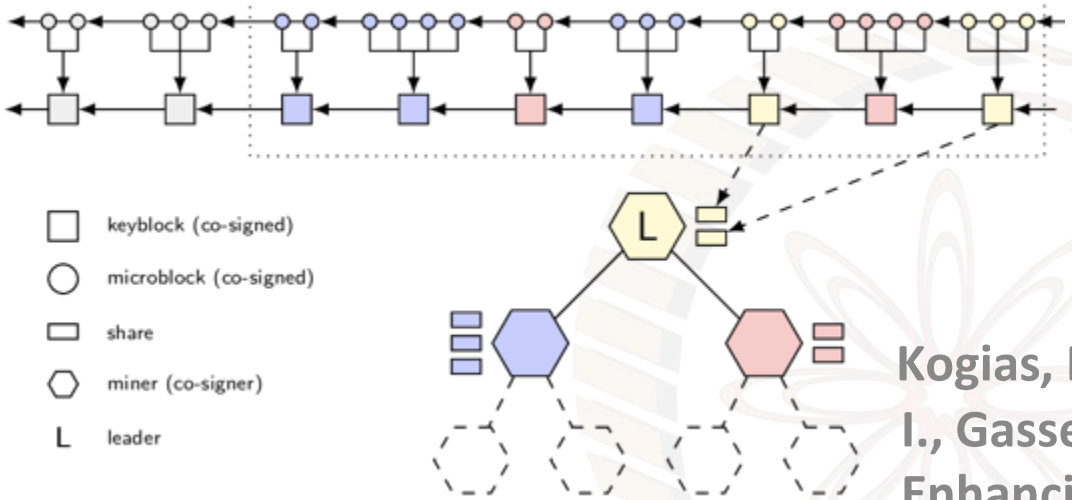


Image Source: http://hackingdistributed.com/2016/08/04/byzcoin/

Kogias, E. K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., & Ford, B. (2016, August). Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium 2016*

Byzcoin

Requirements for Blockchain Consensus

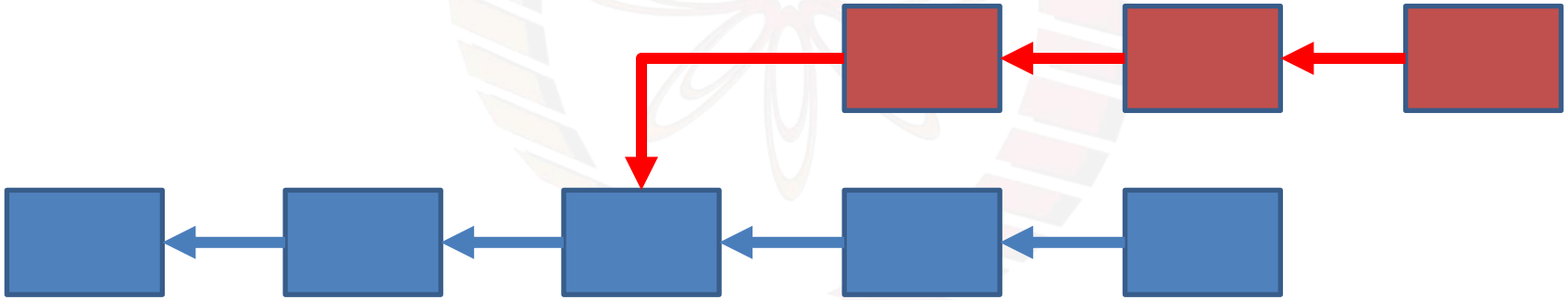
- **Byzantine fault tolerant** – the system should work even in the presence of malicious users while operating across multiple administrative domains
- Should provide **strong consistency guarantee** across replicas
- Should **scale well to increasing workloads** in terms of transactions processed per unit time
- Should **scale well to increasing network size**

Problems with Bitcoin

- There is **no verifiable commitment** of the system that a block would exist
  - Probability of successful fork attack decreases as the size of the Blockchain increases

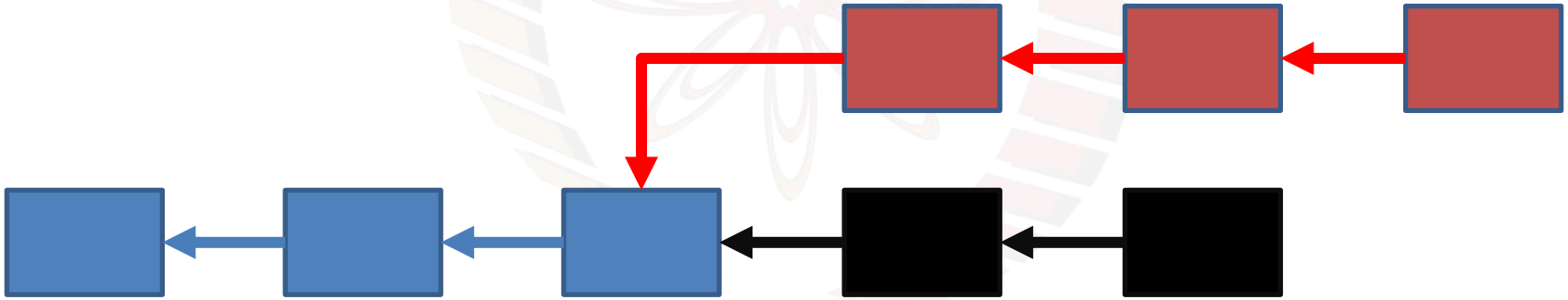
Problems with Bitcoin

- There is **no verifiable commitment** of the system that a block would exist



Problems with Bitcoin

- There is **no verifiable commitment** of the system that a block would exist



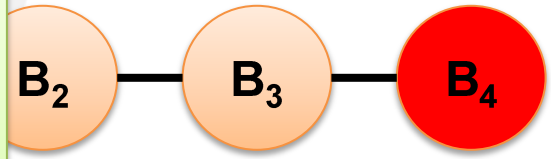
Problems with Bitcoin-NG

- A faulty key block is verified only after end of the round
  - A faulty miner can introduce a number of correct microblocks following a faulty microblock in the system - certainly a overhead for the application - **a fork alleviates the problem further**

Problems with Bitcoin-NG

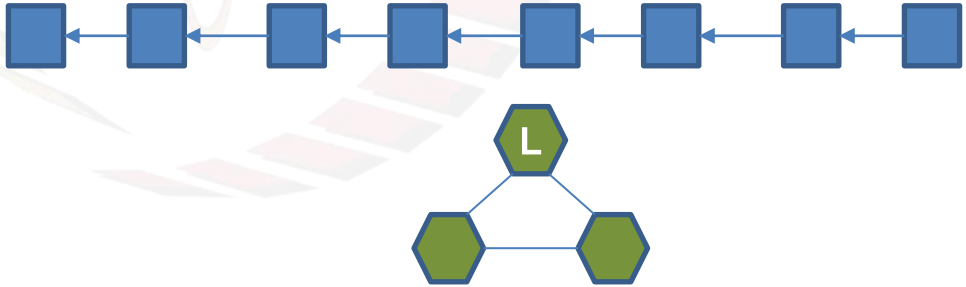
- A faulty key block is verified only after end of the round
  - A faulty miner can introduce a number of correct microblocks following a faulty microblock in the system - certainly a overhead for the application - **a fork alleviates the problem further**

Solve this problem by a set of PBFT verifier - who will verify a block and then only the block is added in the Blockchain



PBFTCoin - A Strawman Design

- **Assumption**: **3f+1** fixed ”trustees” are there, who will run the PBFT to withstand **f** failures
  - Avoid the probabilistic strong consistency - introduces low latency in the system
  - No forks in the system - Blocks are added only after verification from the trustees



Problems of PBFT

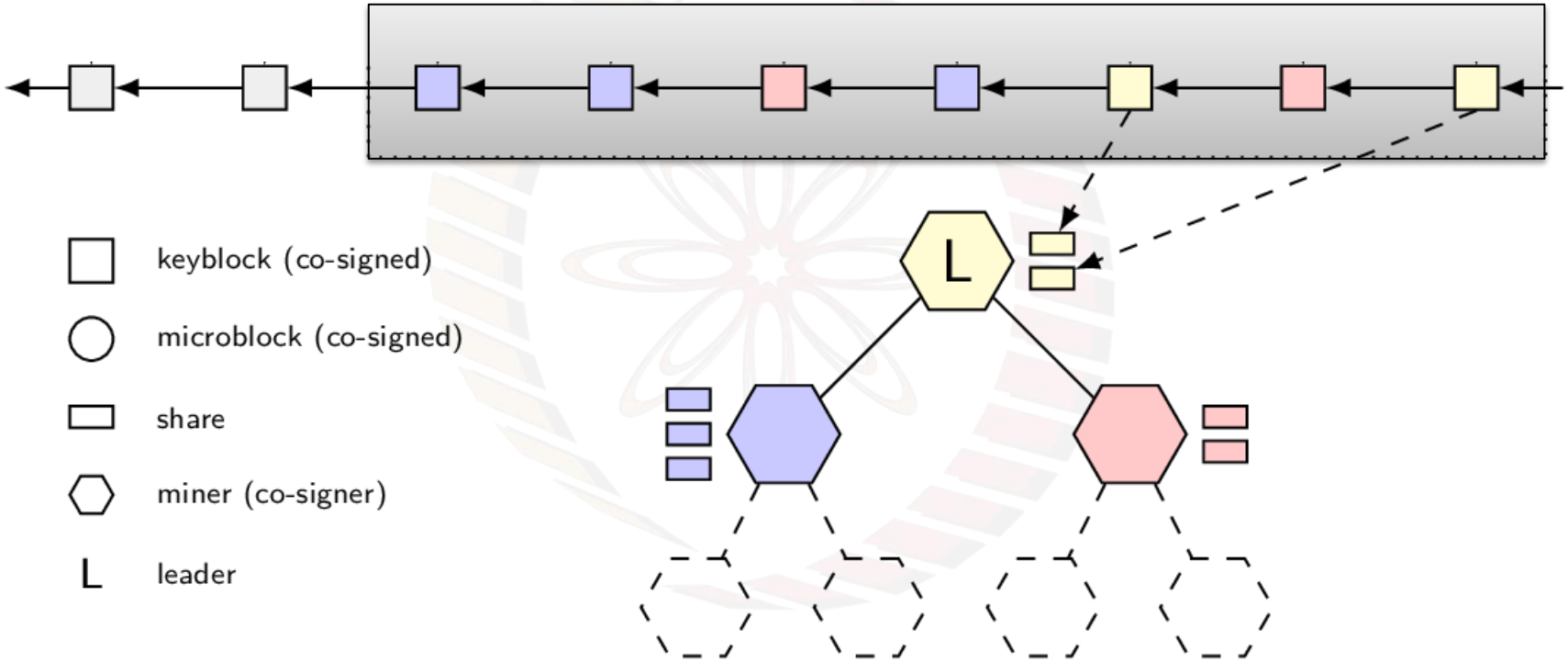
- PBFT requires a **static consensus group** (because of message passing)
- **Scalability** (in terms of nodes) is a problem for PBFT
  - $O(n^2)$ communication complexity
  - $O(n)$ verification complexity
  - Absence of third-party verifiable proofs (PBFT uses MAC - need to share the keys among the miners)
- **Sybil attack** - create multiple pseudonymous identities to subvert the **3f+1** requirements of PBFT

Open the Consensus Group

- Use PoW based system to give a *proof of membership* of a miner as a part of the trustees

- Maintains a “balance of power” within the BFT consensus group
  - Use a fixed-size sliding window
  - Each time a miner finds a new block, it receives a *consensus group share*
  - The share proves the miner’s membership in the trustee group

Open the Consensus Group



Replace MAC with CoSi

- Substitute MACs with public-key cryptography
  - Elliptic curve based cryptography (ECDSA) provides more efficiency
  - Third-party verifiable
  - PoW Blockchain as PKI
  - Use specific topology (ring or chain based) for collective verification

Improve Efficiency

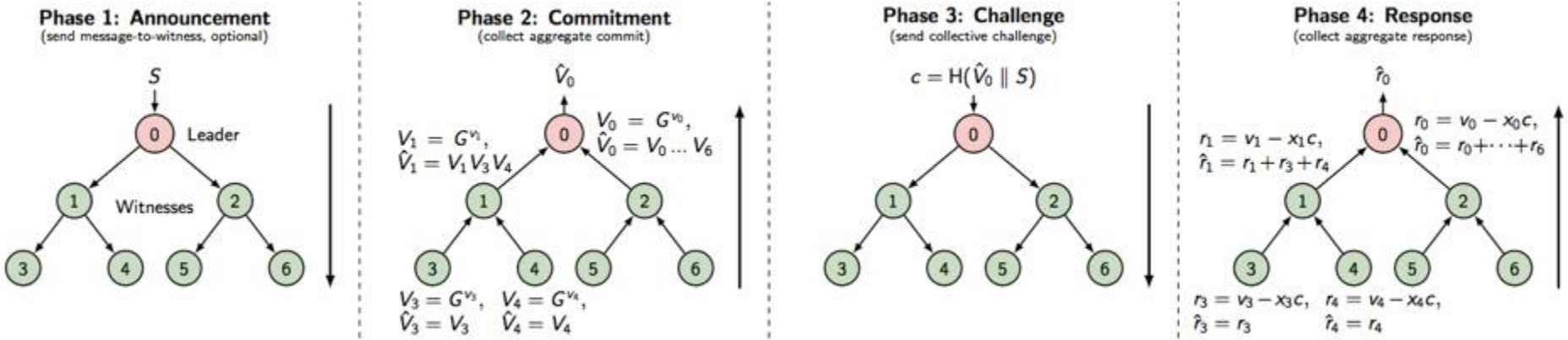
- **Improve O(n) communication complexity**
  - Use tree based multicast protocol - share information with O(log n)

- **Improve O(n) complexity for verification**
  - Use Schnorr multisignatures or BLS for verification
  - Verification can be done in O(1) through signature aggregation

- Multisignatures + Communication trees - **CoSi**

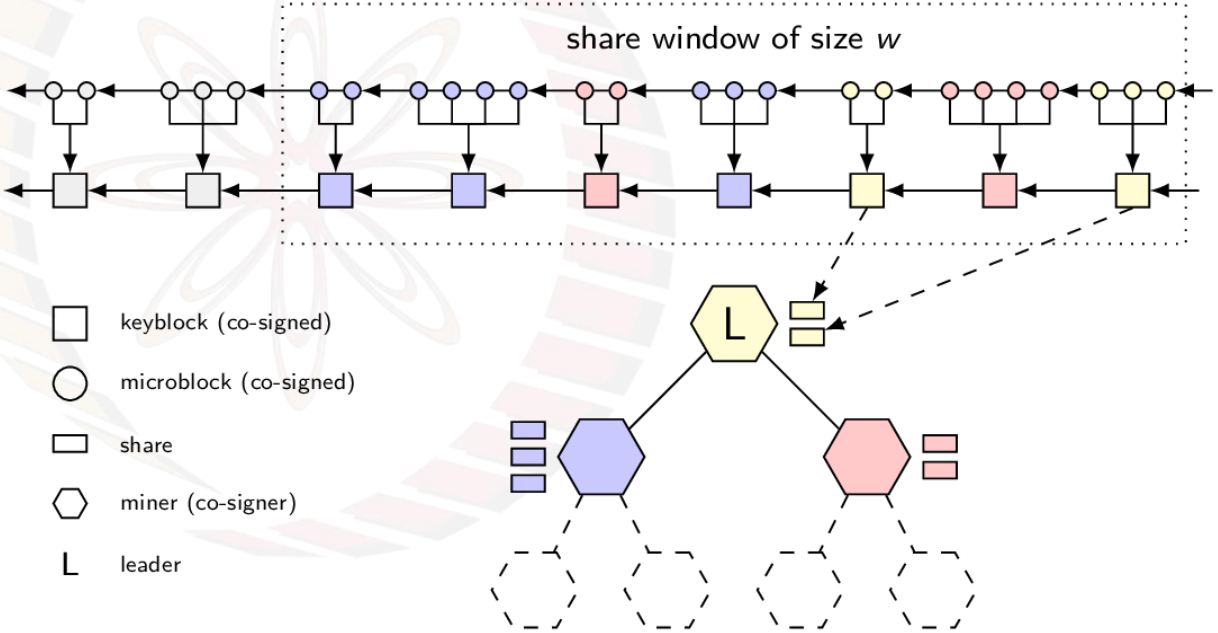
CoSi as BFT Protocol

- CoSi is not a BFT protocol
- PBFT can be implemented over two subsequent CoSi rounds - Prepare and Commit

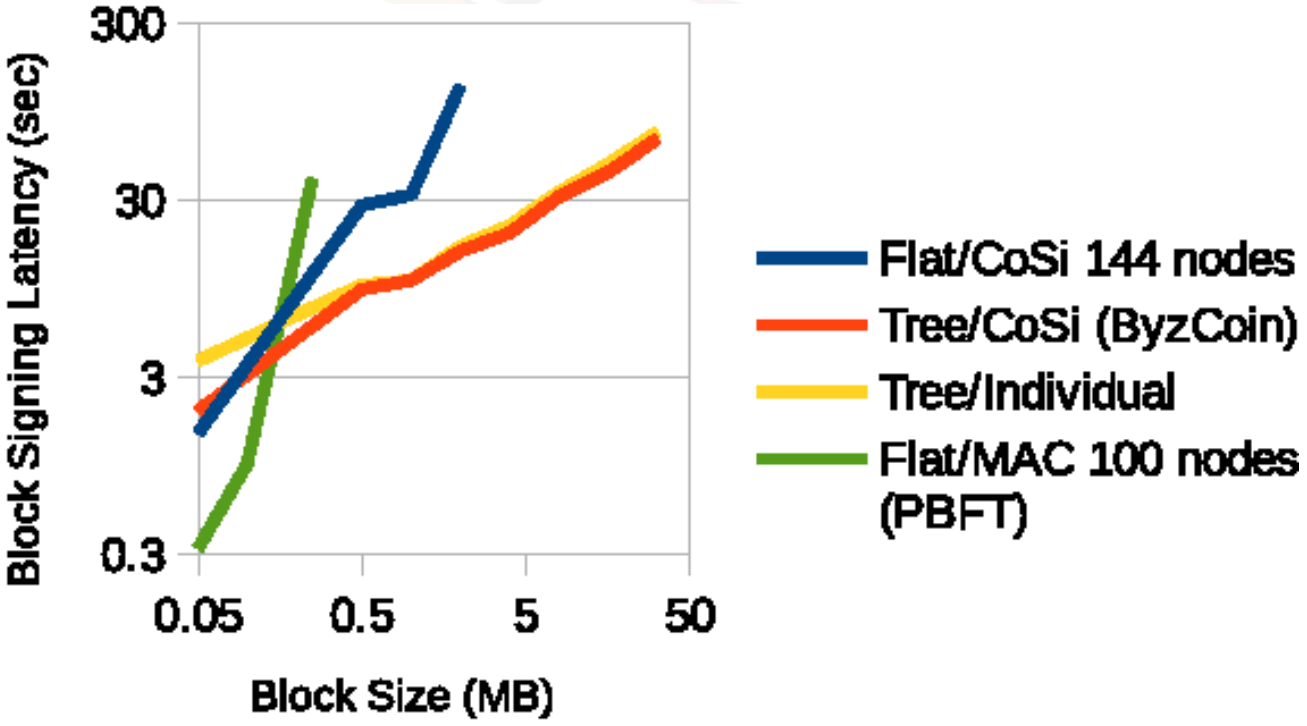


Further Improvement

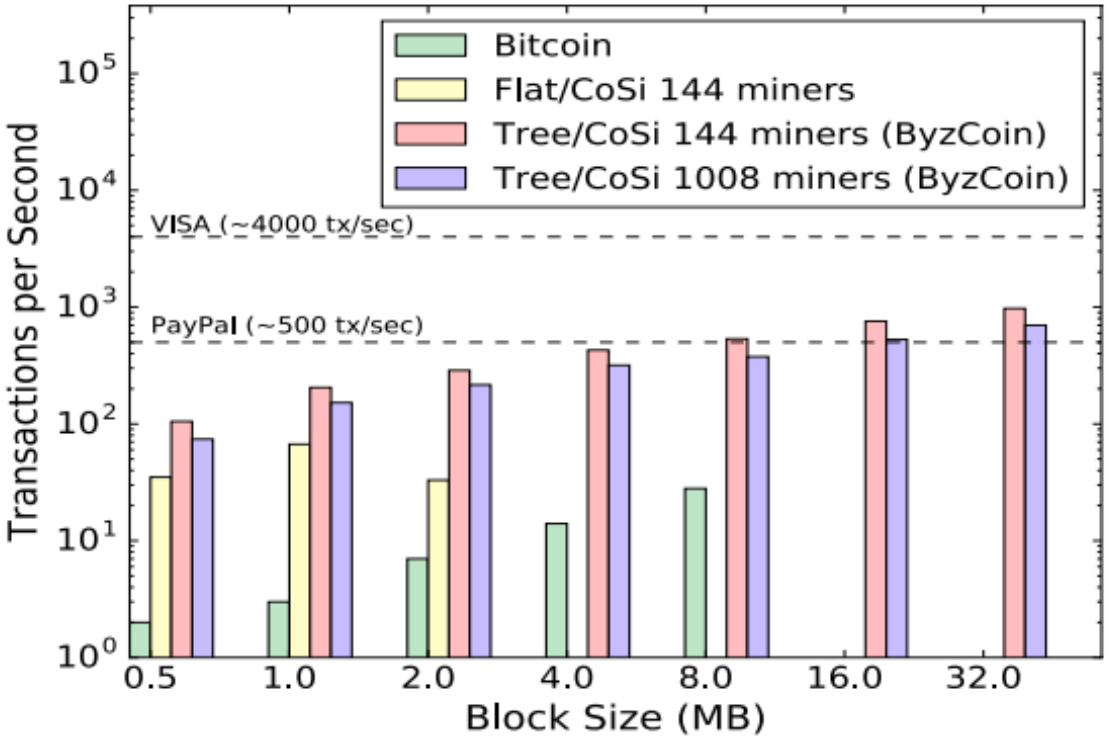
- Inherit Bitcoin NG's idea of separating out **transaction verification** and **leader election**



ByzCoin Performance



ByzCoin Performance



Further Read and Explore

- https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kogias

- Byzcoin source: https://github.com/dedis/cothority/tree/v0/protocols/byzcoin

thank you!