Prism: Deconstructing the Blockchain to Approach Physical Limits

Team members:

Gadde Nitish Samanth - 180050031
Tekuri Sai Akhil - 180050112
Urabavi Revanth Kumar - 180050114

Performance measures for Bitcoin

Security -50% adversary

Transaction Throughput <20tx/s

Confirmation Latency -hours

Problem: Scalability

We want to increase the throughput of bitcoin and decrease the confirmation latency

We have different protocols like

- Bitcoin
- Ethereum
- Stellar
- Algorand
- Thundercoin

Drawbacks

If we increase threshold for block generation which means decrease in Interarrival time, forking increases. This may lead to double spend attack and Security (defined in Bitcoin-NG) decreases.

If we increase block size, validation time increases which leads to increase in propagation delay and forking.

We are not using the complete hashing power. We can use hashing power for generating different types of blocks. There has always been a tradeoff between throughput and confirmation latency.

Different Protocols

We have seen different protocols to either increase throughput or decrease confirmation latency

Eg: Bitcoin-NG

Anchors/Links

Rapid Chains

Inspiration

We employ a similar idea of creating different type of entities.

In satoshi's bitcoin, PoW, ordering of transactions and voting for previous blocks in the chain happens in a single step.

In Bitcoin-NG, PoW and ordering of transactions is separated.

In the same way, for PRISM we separate out
- Leader election,
- Transaction ordering
- Voting for blocks in the chain

Physical Limits

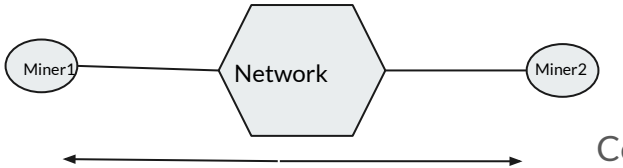
Network capacity

Let say Network capacity = C

Maximum Transaction Throughput <= C

Capacity

Speed-of-light propagation delay



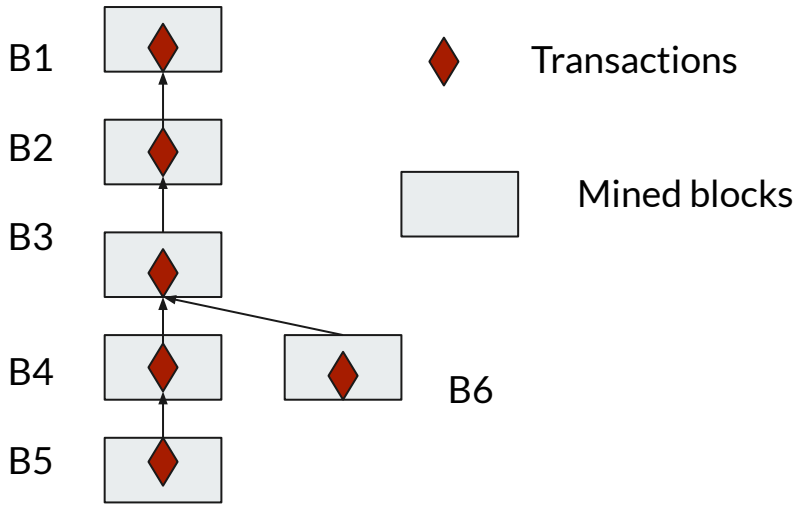
D Delay

Confirmation time > D

Bitcoin

For B1: B2,3,4,5 are acting as votes

For B2: B3,4,5 are acting as votes



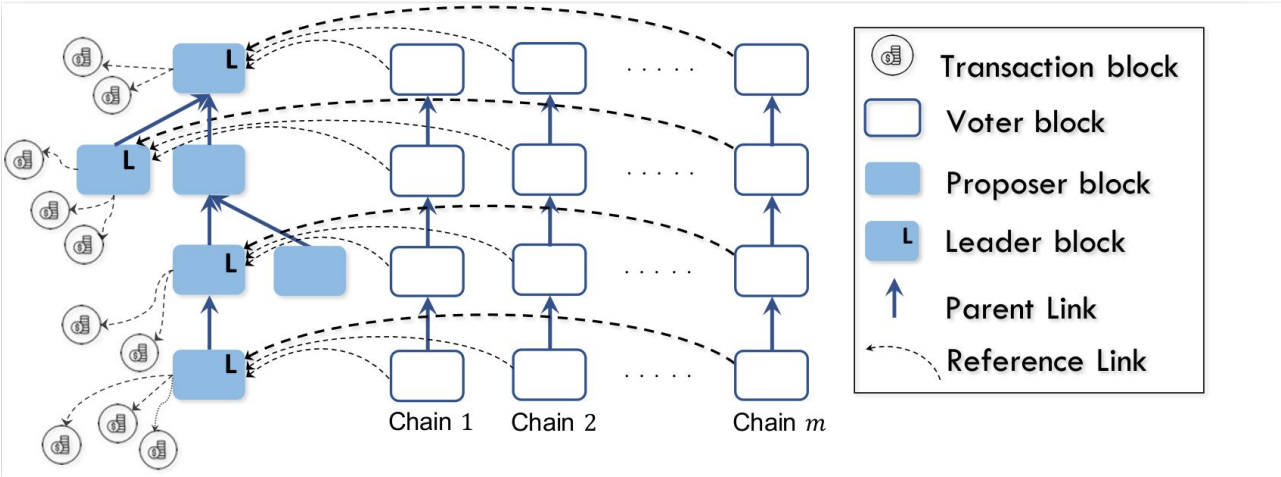
Prism

Leader Election → Proposer blocks

Transaction ordering → Transaction blocks

Voting for blocks → Voting chains

High Level of Prism



Blocks

Transaction Blocks:

- Transaction blocks contain transactions
- Transaction Blocks do not have any specific order
- Transaction Blocks have PoW
- Same transaction may appear in multiple blocks
- Although transaction blocks are not considered part of the proposer blocktree, each transaction block has a proposer block as its parent.

Blocks

Proposer blocks

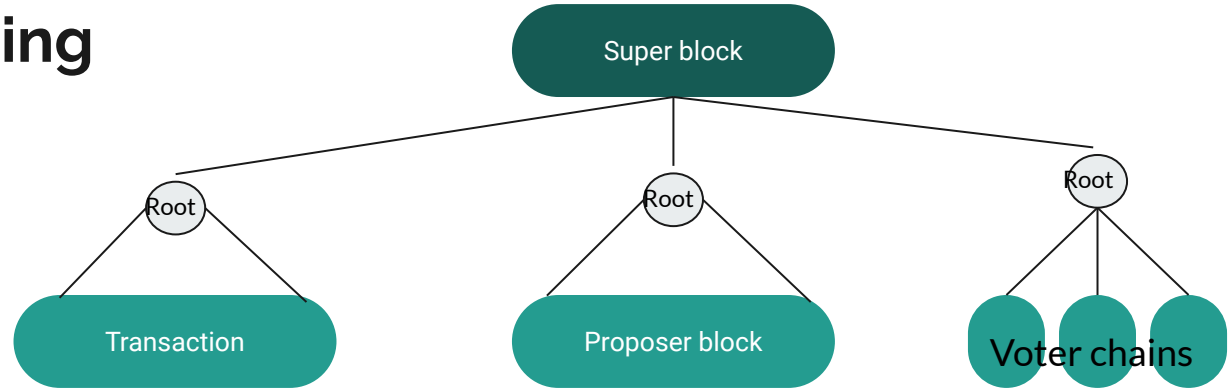
- Contain hashes of Transaction Blocks
- Small Size, so propagate faster than Transaction blocks
- Mining Rule: Mine on longest or heaviest chain.
- These help in ordering of transactions by having a leader at each level
- Rule to choose leader at level l: Block which gets the highest number of votes among voting chains

Blocks

Voter Blocks

- Small Size, Propagation is faster
- Low fork rate
- Voter block votes on all levels in the proposer tree that are unvoted by the voter block's ancestors. Each longest chain from each voter blocktree can cast at most one vote for each level in the proposer blocktree.
- Mining Rule: Mine on longest chain

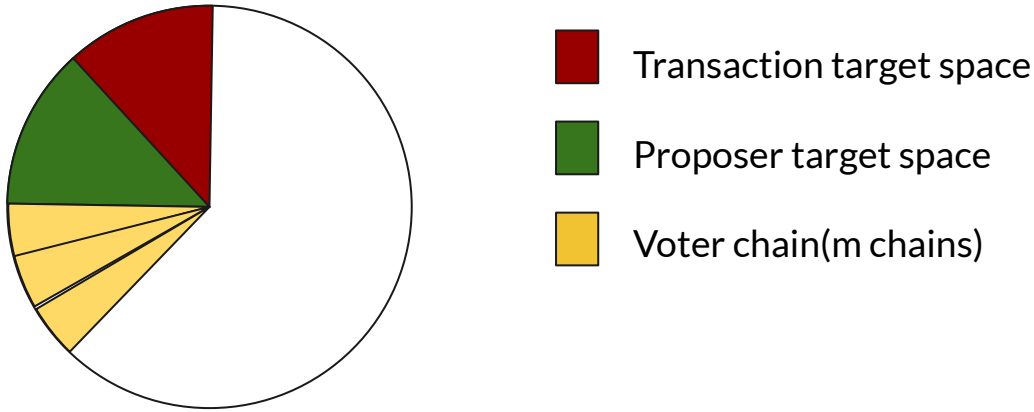
Mining



Simultaneously mine for Transaction block, proposer block, m-voting blocks

How to generate blocks

Target space:



Contribution

The main contribution of this work is a new blockchain protocol, Prism, which, in the face of any powerful adversary with power $\beta < 0.5$, can simultaneously achieve:

(1) Security: a total ordering of the transactions, with consistency and liveness guarantees.

(2) Throughput: a throughput $\lambda = 0.9(1 - \beta)C$ transactions per second.

(3) Latency: confirmation of all honest transactions (without public double spends) with an expected latency of

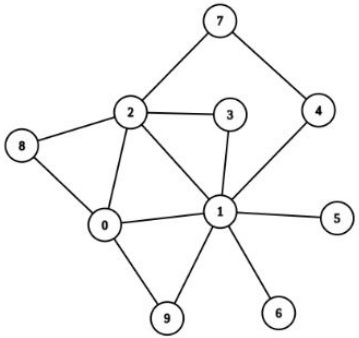
$$E[\tau] < \max\{c_1(\beta)D, c_2(\beta)(B_v/C)\log(1/\varepsilon)\} \text{ seconds}$$

with confirmation reliability at least $1 - \varepsilon$ .. Here,$c_1(\beta)$ and $c_2(\beta)$ are constants depending only on β

Simulation

We have created a P2P network with power law distribution using 2018 Bitcoin paper on Distributed Systems.

Here is a sample network distribution for 10 nodes.



Details

We have taken 2 classes of nodes
HIGH CPU nodes
LOW CPU nodes

The hashing power of HIGH CPU nodes is double the hashing power of LOW CPU nodes.
The time delay of broadcasting of block depends on the content of it.

Each transaction block contain maximum of 20 transactions
Each proposer block can refer maximum of 5 transaction blocks

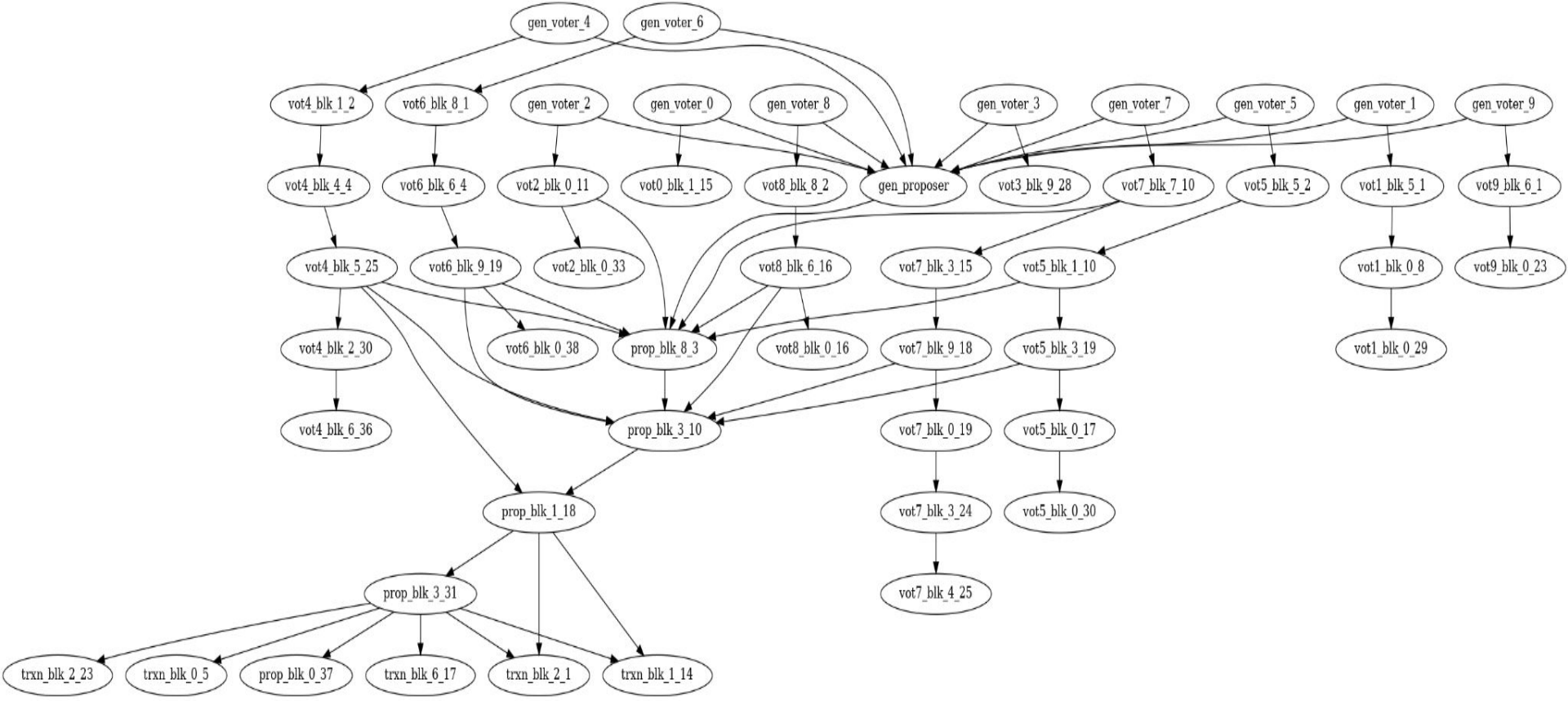
A block is created at every interarrival time specified on average. The probability for the block to be transaction block is ⅔, proposal block is ⅙ and voter block is ⅙.

Prism blocks

Enter the number of nodes(n): 10
Enter the percent of slow nodes(z): 20
Enter the mean interarrival time of transactions(T_tx): 2
Enter the percent of High CPU nodes: 80
Enter block interarrival time(in sec): 5
Enter no of voter chains: 10

We have created a tree structure using these values.

- A proposer block points to its children and refers to its transaction blocks
- A voter block points to its children and refers to its voted proposer block


gen_voter_4
gen_voter_6
vot4_blk_1_2
vot6_blk_8_1
gen_voter_2
gen_voter_0
gen_voter_8
gen_voter_3
gen_voter_7
gen_voter_5
gen_voter_1
gen_voter_9
vot4_blk_4_4
vot6_blk_6_4
vot2_blk_0_11
vot0_blk_1_15
vot8_blk_8_2
gen_proposer
vot3_blk_9_28
vot7_blk_7_10
vot5_blk_5_2
vot1_blk_5_1
vot9_blk_6_1
vot4_blk_5_25
vot6_blk_9_19
vot2_blk_0_33
vot8_blk_6_16
vot7_blk_3_15
vot5_blk_1_10
vot1_blk_0_8
vot9_blk_0_23
vot4_blk_2_30
vot6_blk_0_38
prop_blk_8_3
vot8_blk_0_16
vot7_blk_9_18
vot5_blk_3_19
vot1_blk_0_29
vot4_blk_6_36
prop_blk_3_10
vot7_blk_0_19
vot5_blk_0_17
prop_blk_1_18
vot7_blk_3_24
vot5_blk_0_30
prop_blk_3_31
vot7_blk_4_25
trxn_blk_2_23
trxn_blk_0_5
prop_blk_0_37
trxn_blk_6_17
trxn_blk_2_1
trxn_blk_1_14

Explanation

In prism, if a block is created for every t secs, then the proposal block is created for every 6t secs on average. So, we have to compare the throughput of PRISM and bitcoin-blockchains by setting the block interarrival time as t and 6t respectively.

Analysis

Prism

Enter the number of nodes(n): 10
Enter the percent of slow nodes(z): 20
Enter the mean interarrival time of transactions(T_tx): 2
Enter the percent of High CPU nodes: 80
Enter block interarrival time(in sec): 5
Enter no of voter chains: 10
Stop time = 100

Throughput = 280 transactions

Bitcoin

Enter the number of nodes(n): 10
Enter the percent of slow nodes(z): 20
Enter the mean interarrival time of transactions(T_tx): 2
Enter the percent of High CPU nodes: 80
Enter block interarrival time(in sec): 30
Stop time = 100

Throughput = 173 transactions

The throughput for PRISM is higher than that of bitcoin from the simulations.

Thank You