Prism: Scaling Bitcoin by 10,000×

Lei Yang [1] Vivek Bagaria [2] Gerui Wang [3]
Mohammad Alizadeh [1] David Tse [2] Giulia Fanti [4]
Pramod Viswanath [3]

[1]MIT CSAIL

[2]Stanford University

[3]University of Illinois Urbana-Champaign

[4]Carnegie Mellon University

The Stanford Blockchain Conference 2020

Bitcoin: high security, low throughput, and long latency

- Security: 50% adversary

Bitcoin: high security, low throughput, and long latency

- Security: 50% adversary
- Throughput: 7 tps
- Confirmation Latency: hours

Bitcoin: high security, low throughput, and long latency

- Security: 50% adversary
- Throughput: 7 tps
- Confirmation Latency: hours

How much better can we do theortically

Bitcoin: high security, low throughput, and long latency

- Security: 50% adversary
- Throughput: 7 tps
- Confirmation Latency: hours

How much better can we do theortically and practically?

Bitcoin: high security, low throughput, and long latency

- Security: 50% adversary
- Throughput: 7 tps
- Confirmation Latency: hours

How much better can we do theortically and practically?
And how?

This talk

The Prism consensus protocol

System implementation

Evaluation results and findings

This talk

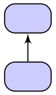
The Prism consensus protocol

System implementation

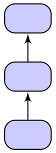
Evaluation results and findings

Bitcoin: mine on the longest chain

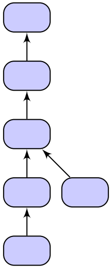
Bitcoin: mine on the longest chain



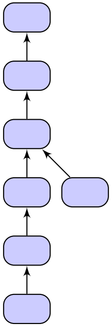
Bitcoin: mine on the longest chain



Bitcoin: mine on the longest chain



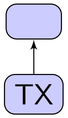
Bitcoin: mine on the longest chain



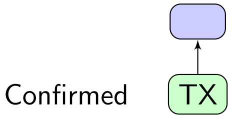
Mining rate f = 1 block per 10 min

Bitcoin: *k*-deep confirmation rule causes high latency

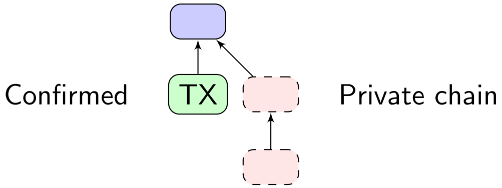
Bitcoin: *k*-deep confirmation rule



Bitcoin: *k*-deep confirmation rule

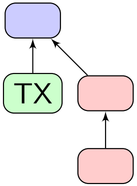


Bitcoin: *k*-deep confirmation rule

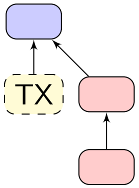


Bitcoin: *k*-deep confirmation rule

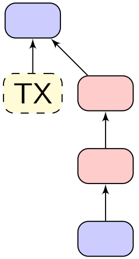
Confirmed



Bitcoin: *k*-deep confirmation rule

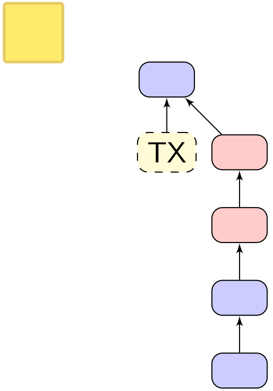


Bitcoin: *k*-deep confirmation rule



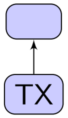
Longest chain

Bitcoin: *k*-deep confirmation rule

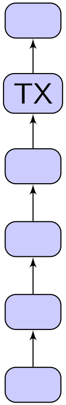


Longest chain

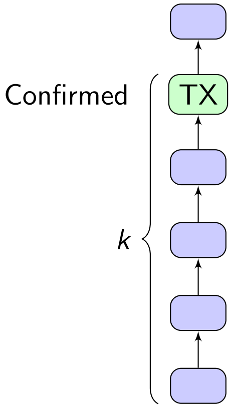
Bitcoin: *k*-deep confirmation rule



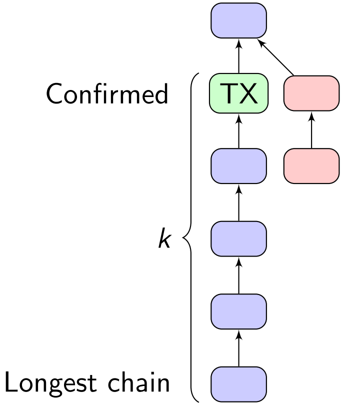
Bitcoin: *k*-deep confirmation rule



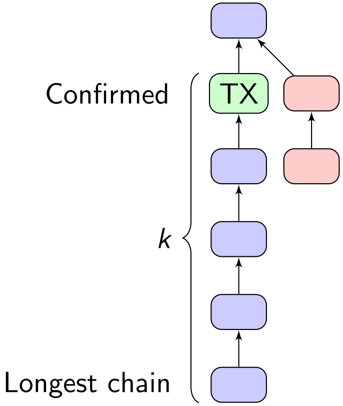
Bitcoin: k-deep confirmation rule



Bitcoin: k-deep confirmation rule



Bitcoin: *k*-deep confirmation rule

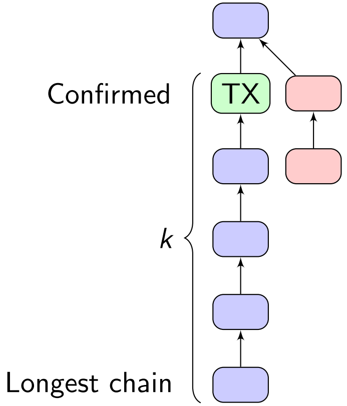


30% adversary power

k	ϵ
0	1.0000000
5	0.1773523
10	0.0416605
15	0.0101008
20	0.0024804
25	0.0006132
30	0.0001522
35	0.0000379
40	0.0000095
45	0.0000024
50	0.0000006

Bitcoin: *k*-deep confirmation rule

T, B, C, D
Numer of blocks per second:T / B
Propagation Delay: D + B / C
“active” blocks: TD/B + T/C

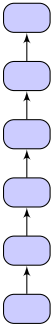


30% adversary power

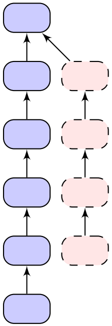
For 10^{-3} attack probability, wait 250 mins!

k	ϵ
0	1.0000000
5	0.1773523
10	0.0416605
15	0.0101008
20	0.0024804
25	0.0006132
30	0.0001522
35	0.0000379
40	0.0000095
45	0.0000024
50	0.0000006

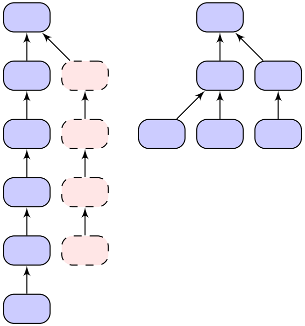
Increasing the mining rate harms the security



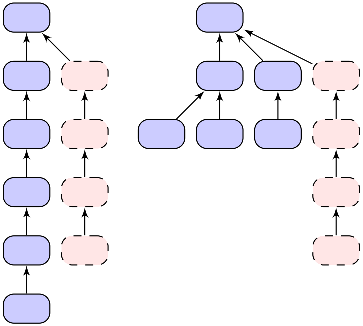
Increasing the mining rate harms the security



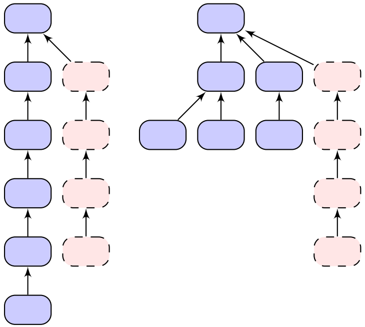
Increasing the mining rate harms the security

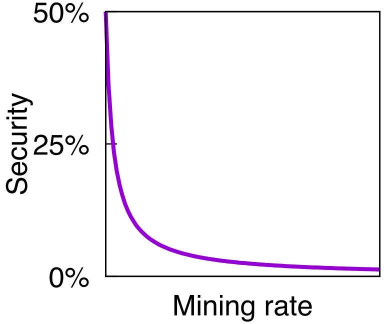


Increasing the mining rate harms the security



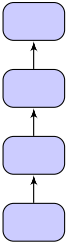
Increasing the mining rate harms the security



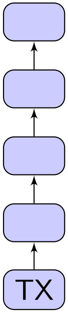


Same when increasing the block size

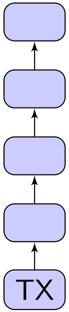
Two roles of a block: proposing and voting



Two roles of a block: proposing and voting

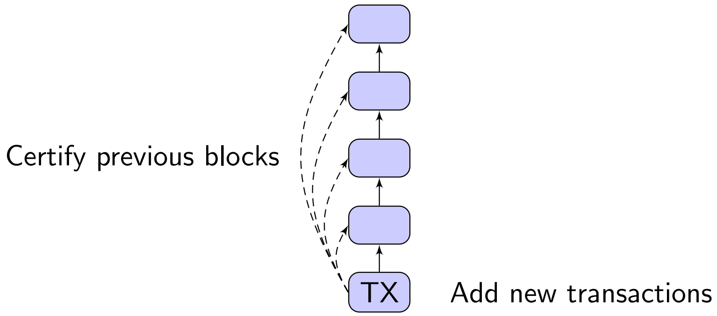


Two roles of a block: proposing and voting



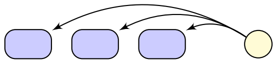
Add new transactions

Two roles of a block: proposing and voting

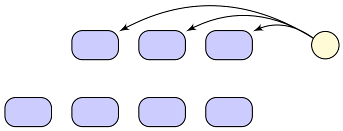


Scale proposing with lots of transaction blocks

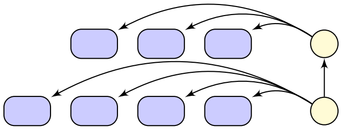
Scale proposing with lots of transaction blocks



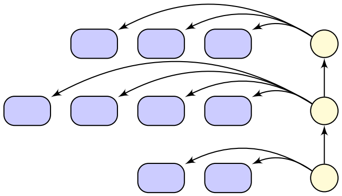
Scale proposing with lots of transaction blocks



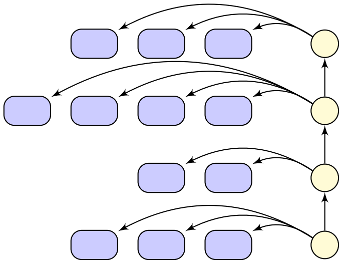
Scale proposing with lots of transaction blocks



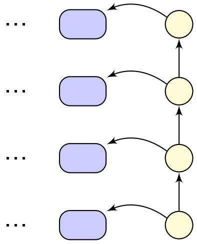
Scale proposing with lots of transaction blocks



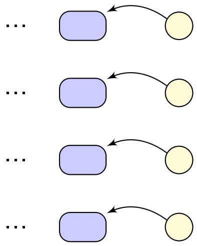
Scale proposing with lots of transaction blocks



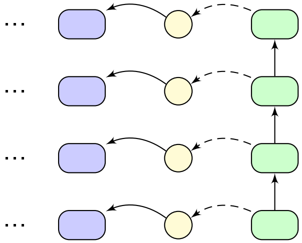
Scale voting with lots of voter chains



Scale voting with lots of voter chains

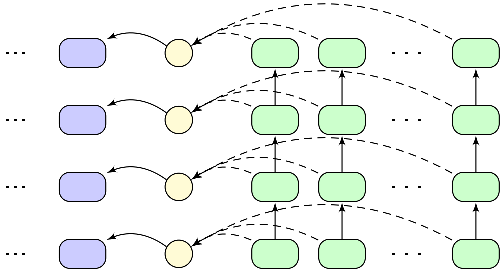


Scale voting with lots of voter chains



- 1 voter chain: 25-deep

Scale voting with lots of voter chains



- 1 voter chain: 25-deep
- 1000 voter chains: 2-deep

Prism is provably fast and secure

Adversary power $\beta < 0.5$

- Security: consistency and liveness

Prism is provably fast and secure

Adversary power $\beta < 0.5$

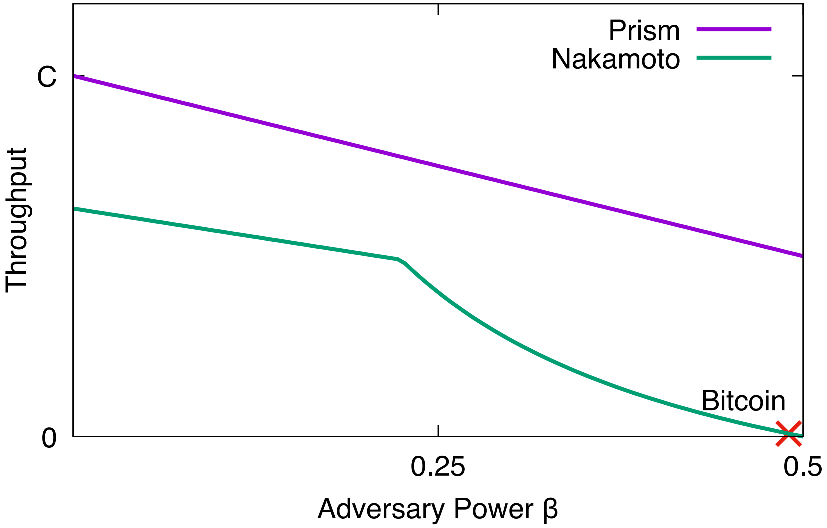
- Security: consistency and liveness
- Throughput: $(1-\beta)C$

Prism is provably fast and secure

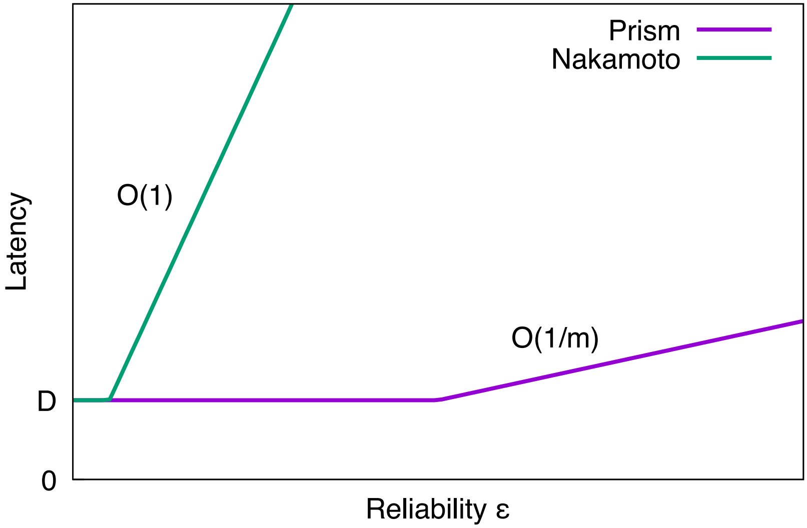
Adversary power $\beta < 0.5$

- Security: consistency and liveness
- Throughput: $(1 - \beta)C$
- Confirmation Latency: $O\left(D\right) + O\left(\frac{-D\log(\epsilon)}{m}\right)$

Prism throughput approaches the network bandwidth



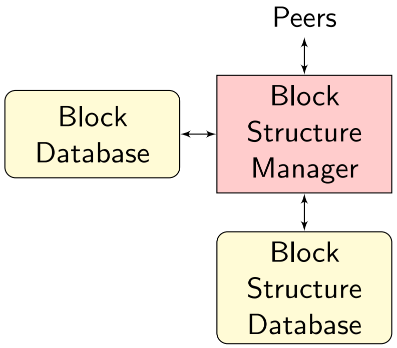
Prism latency approaches the network latency



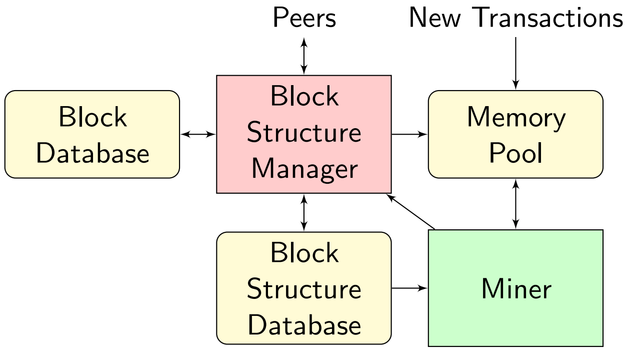
Implementing Prism in Rust

- 10,000 lines of Rust
- UTXO model
- Pay-to-public-key transactions
- Code available at t.leiy.me/prism-code

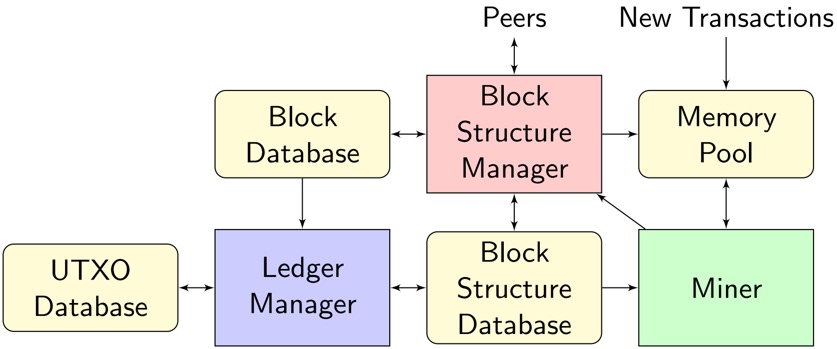
Blockchain client: consensus and ledger keeping



Blockchain client: consensus and ledger keeping



Blockchain client: consensus and ledger keeping

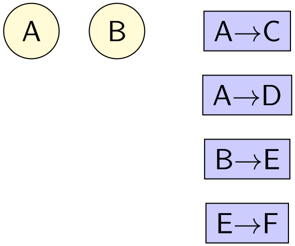


High throughput brings challenges

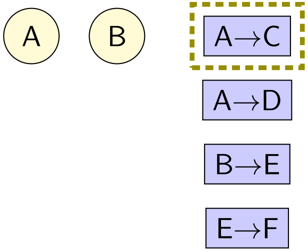
In real time:

- 70,000 tps
- 400 blocks/s
- 1000 chains

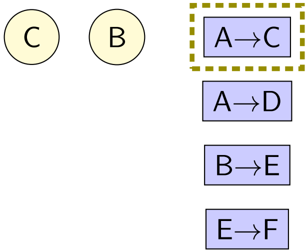
Enable parallel transaction execution with scoreboarding



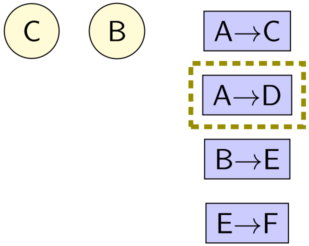
Enable parallel transaction execution with scoreboarding



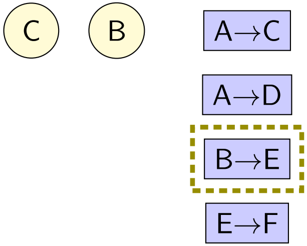
Enable parallel transaction execution with scoreboarding



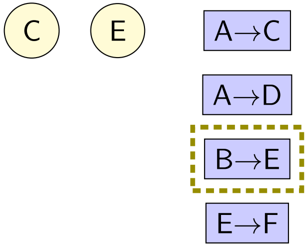
Enable parallel transaction execution with scoreboarding



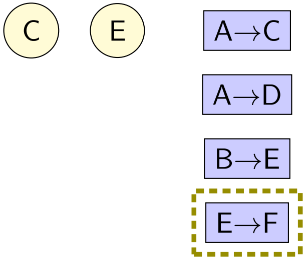
Enable parallel transaction execution with scoreboarding



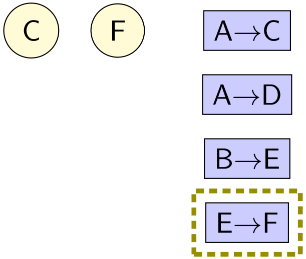
Enable parallel transaction execution with scoreboarding



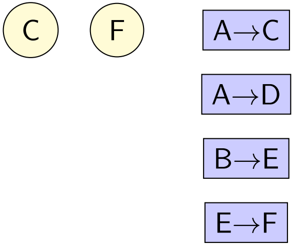
Enable parallel transaction execution with scoreboarding



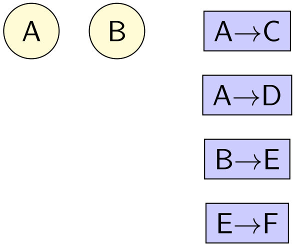
Enable parallel transaction execution with scoreboarding



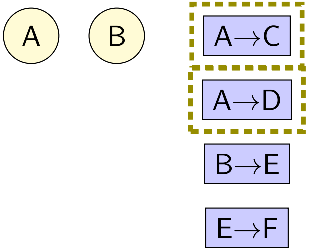
Enable parallel transaction execution with scoreboarding



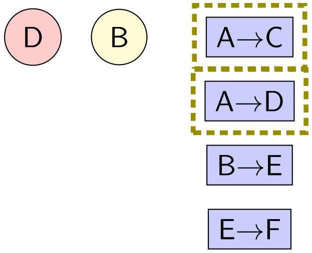
Enable parallel transaction execution with scoreboarding



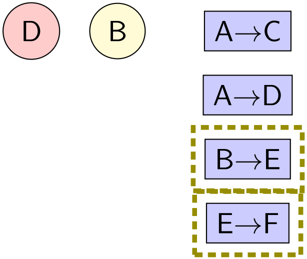
Enable parallel transaction execution with scoreboarding



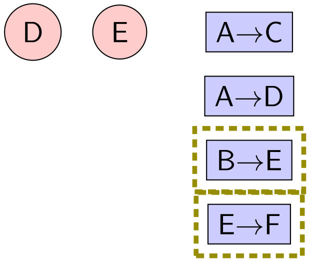
Enable parallel transaction execution with scoreboarding



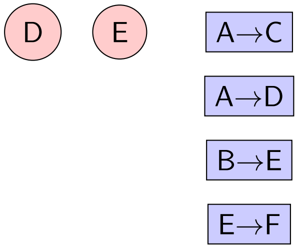
Enable parallel transaction execution with scoreboarding



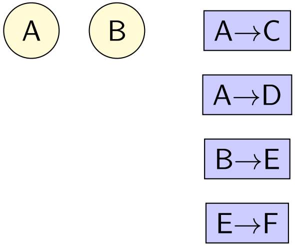
Enable parallel transaction execution with scoreboarding



Enable parallel transaction execution with scoreboarding

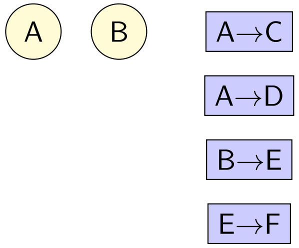


Enable parallel transaction execution with scoreboarding



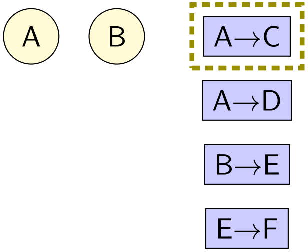
Scoreboard:

Enable parallel transaction execution with scoreboarding



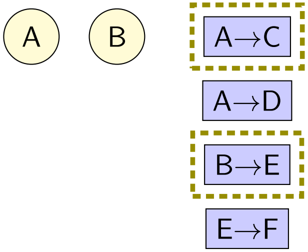
Scoreboard: A, C

Enable parallel transaction execution with scoreboarding



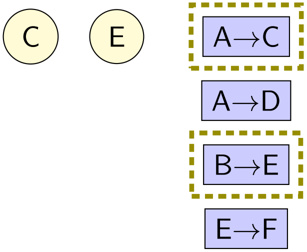
Scoreboard: A, C

Enable parallel transaction execution with scoreboarding



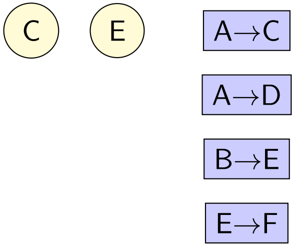
Scoreboard: A, B, C, E

Enable parallel transaction execution with scoreboarding



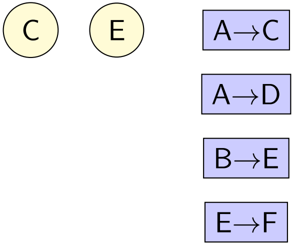
Scoreboard: A, B, C, E

Enable parallel transaction execution with scoreboarding



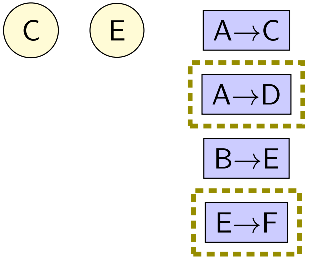
Scoreboard:

Enable parallel transaction execution with scoreboarding



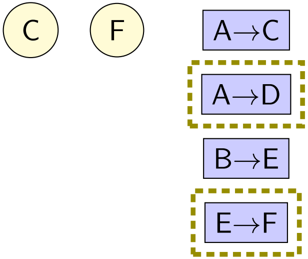
Scoreboard: A, D, E, F

Enable parallel transaction execution with scoreboarding



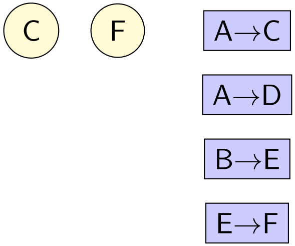
Scoreboard: A, D, E, F

Enable parallel transaction execution with scoreboarding



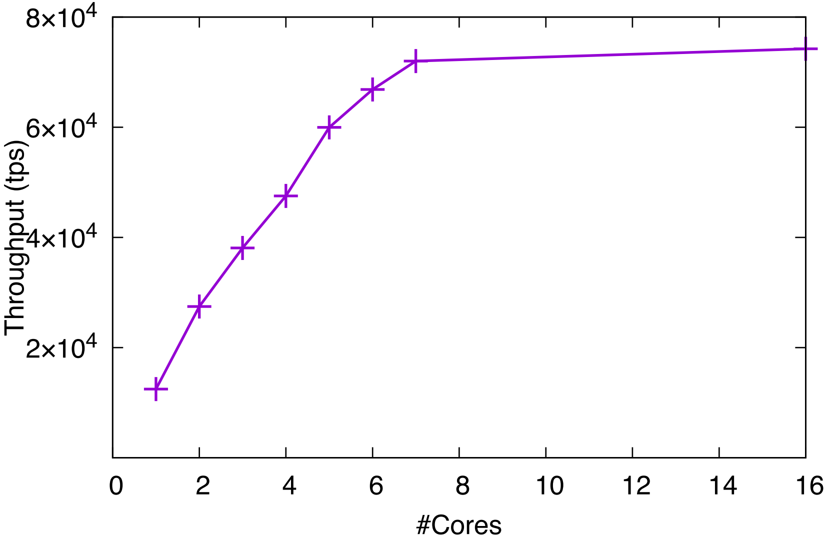
Scoreboard: A, D, E, F

Enable parallel transaction execution with scoreboarding

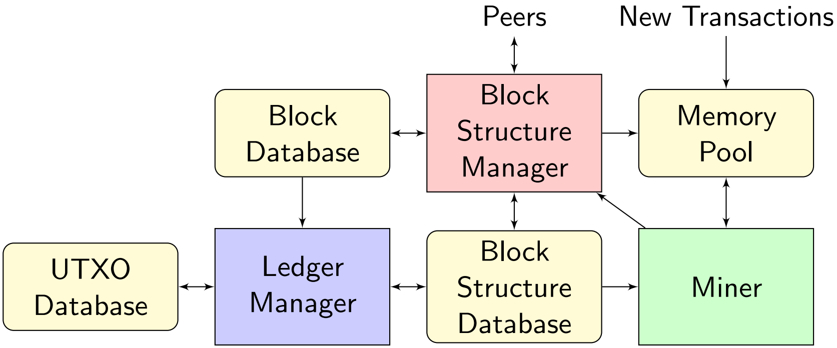


Scoreboard:

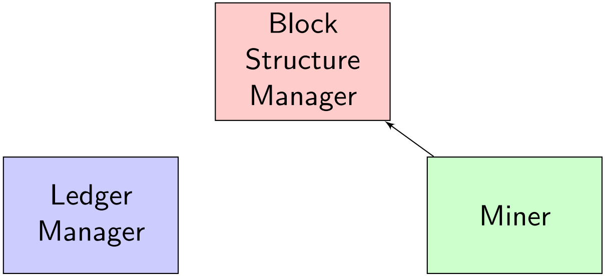
Enable parallel transaction execution with scoreboarding



Handle high block rate with async ledger update

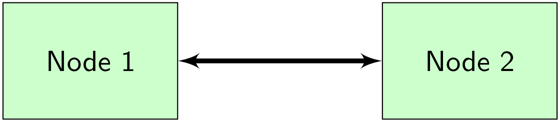


Handle high block rate with async ledger update

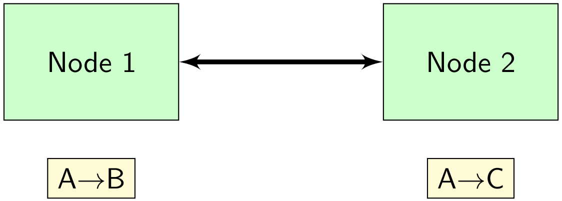


- Ledger updates are “infrequent”
- Sanitize later

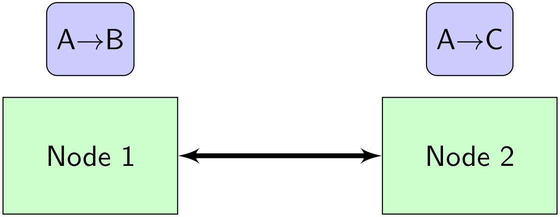
Reduce spams using random jittering



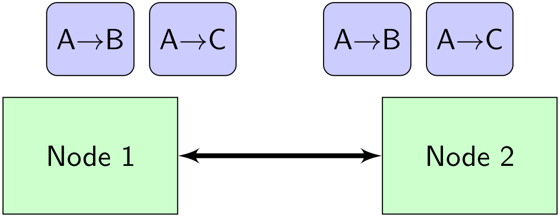
Reduce spams using random jittering



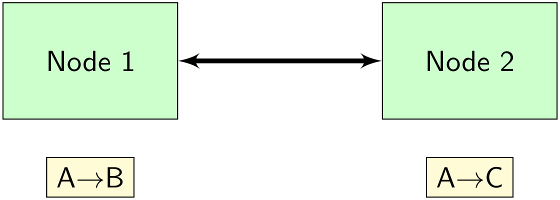
Reduce spams using random jittering



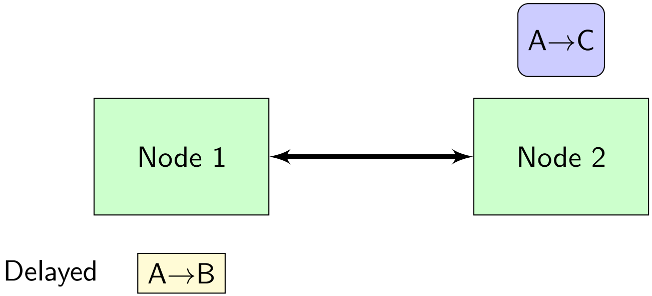
Reduce spams using random jittering



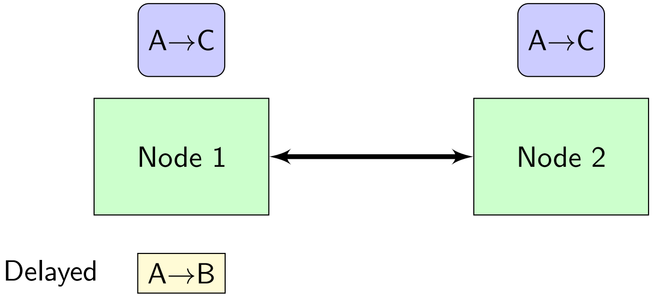
Reduce spams using random jittering



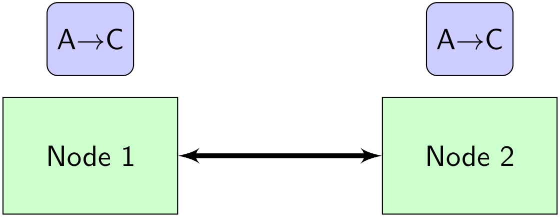
Reduce spams using random jittering



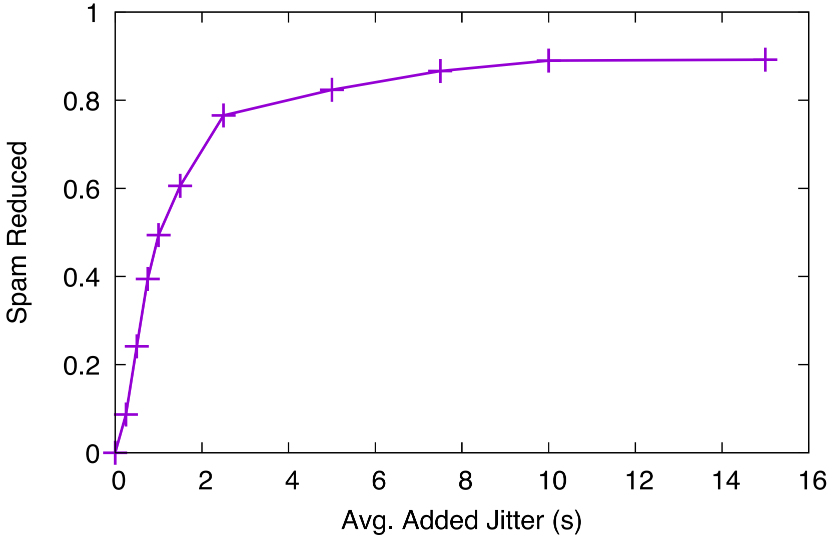
Reduce spams using random jittering



Reduce spams using random jittering



Reduce spams using random jittering



The Prism consensus protocol

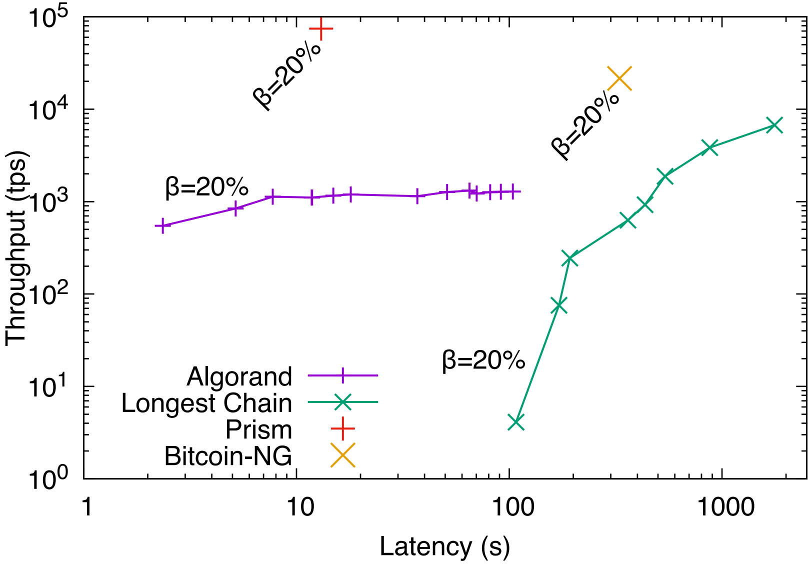
System implementation

Evaluation results and findings

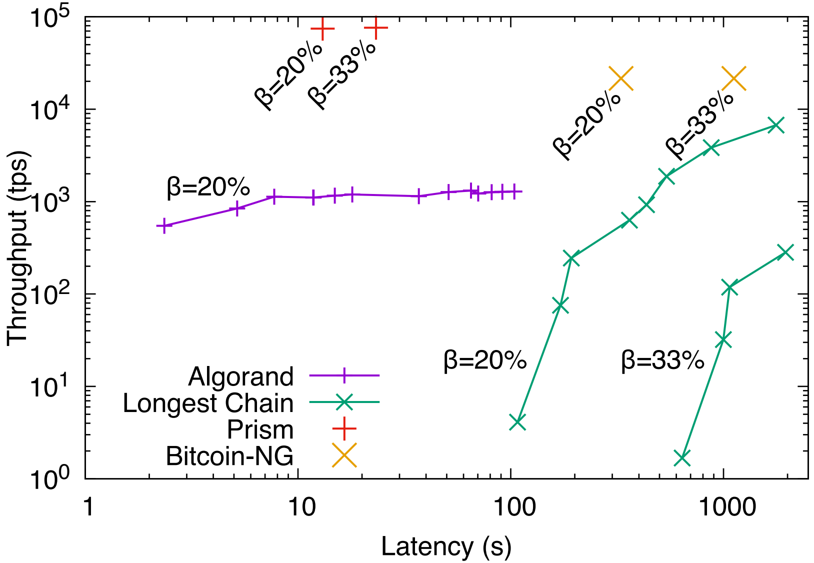
Testbed setup

- 100 - 1000 AWS EC2 instances
- Random 4-regular graph
- 120ms propagation delay
- 400 Mbps bandwidth

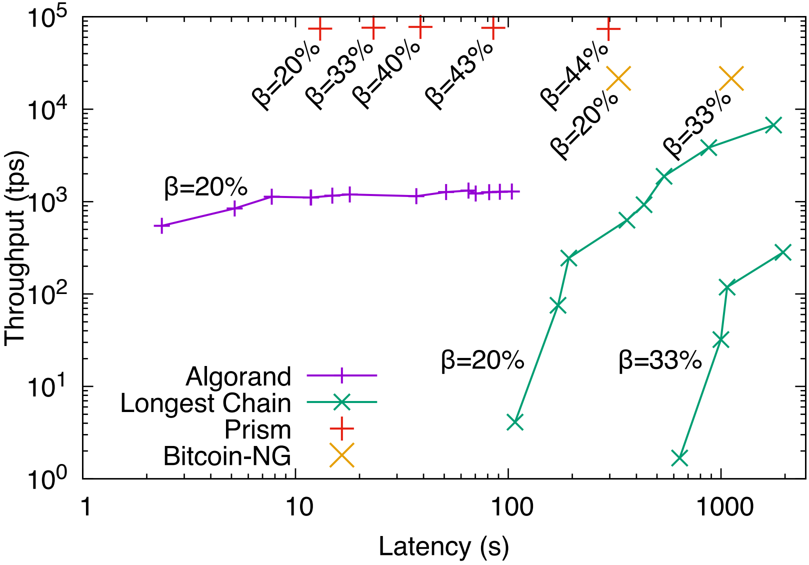
Comparison with Algorand, Bitcoin-NG, Nakamoto



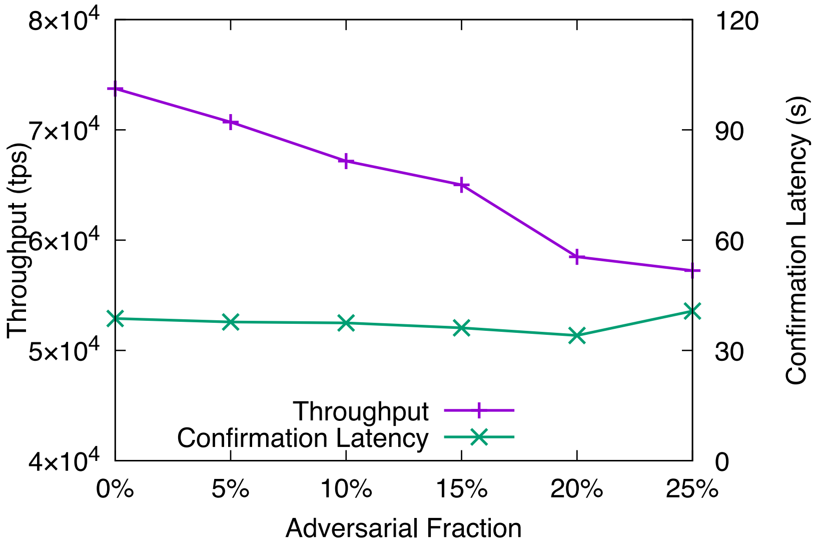
Comparison with Algorand, Bitcoin-NG, Nakamoto



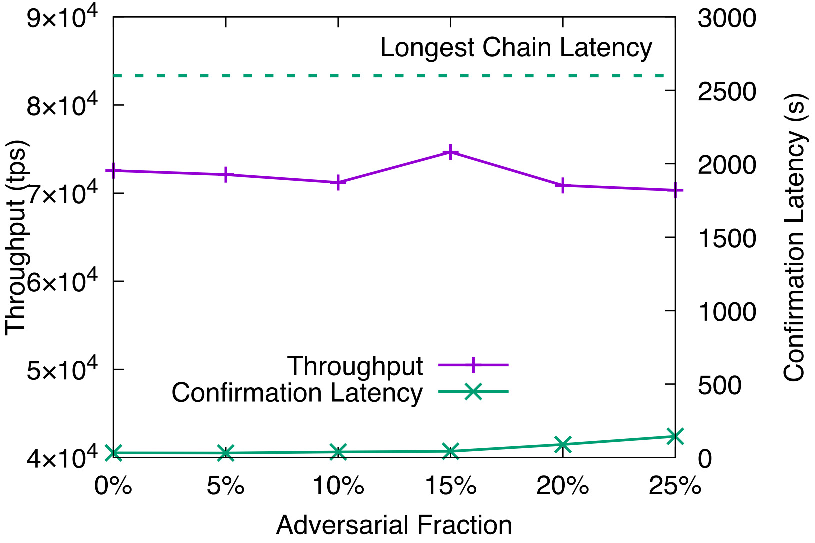
Comparison with Algorand, Bitcoin-NG, Nakamoto



Prism is robust against censorship attacks



Prism is robust against balancing attacks



Our implementation is efficient

CPU

- Signature check: 22%
- RocksDB: 53%
- Data serialization: 7%

Bandwidth

- Transaction blocks: 99.5%
- Voter blocks: 0.4%
- Proposer blocks: 0.1%

Takeaways

- Prism approaches the physical limit by deconstructing and scaling each part
- Prism is proven with a real implementation
- Building a high performance blockchain client requires careful design

Resources

- Code: t.leiy.me/prism-code
- Theory Paper: Deconstructing the Blockchain to Approach Physical Limits
- System Paper: t.leiy.me/prism-paper
- Online Demo: t.leiy.me/prism-demo