Blockchain Technologies

Scaling blockchain

HARD-CODED LIMITS IN BITCOIN

- The divisibility of the currency(100 M *satoshis* per bitcoin)
- The total number of bitcoins(21M total bitcoins maximum)
- The block reward structure(50,25,12.5 ... bitcoin mining reward)

These affect economic balance of power too much to change now

If that changes, it will have significant financial implications and the community has basically agreed that those aspects, whether or not they were wisely chosen, will not change.

HARD-CODED LIMITS IN BITCOIN

- Limits on the average time per block (10 min. average creation time per block)
- The size of blocks (1 M bytes in a block)
- The number of signature operations in a block(20,000 signature operations per block)

THROUGHPUT LIMITS IN BITCOIN

- 1 M bytes/block (10 min)
- >250 bytes/transaction
- 7 transactions/sec ☹

Compare to:

- VISA: 2,000-24,000 transactions/sec
- PayPal: 50-100 transaction/sec

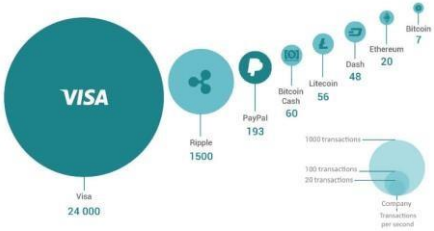
CRYPTOGRAPHIC LIMITS IN BITCOIN

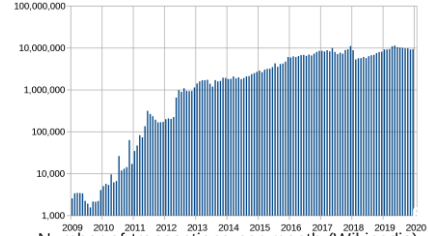
- Only 1 signature algorithm (ECDSA/P256)
- Hard-coded hash functions
- Crypto primitives might break by 2040...

REMINDER: BITCOIN IS NOT SCALABLE!

- As we have already seen, BitCoin is not an scalable payment system.
- The following bottlenecks are the most important ones:
  - Throughput
  - Delay
  - Storage

	Security	Throughput	Latency
Bitcoin	50% Adversary	~ 7 transactions/s	~ hours
Etherium	50% Adversary	~ 20 transactions/s	~ tens of minutes



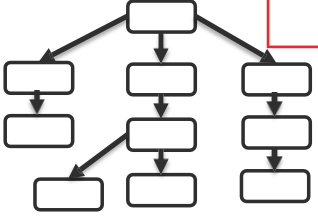


Number of transactions per month (Wikipedia)

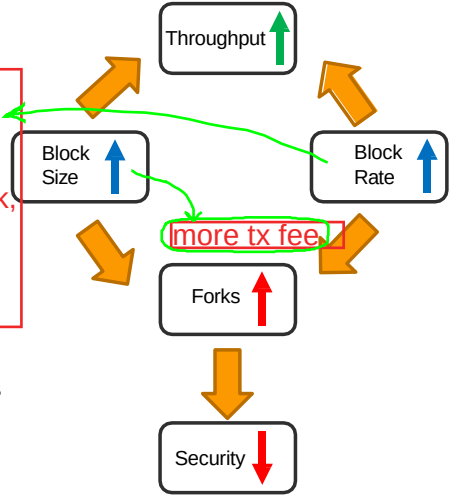
NO EASY WAY TO SOLVE THE SCALABILITY PROBLEM

- We have already analyzed that tuning protocols parameters does not solve the problem.
  - Increasing block rate
  - Increasing block size

It means blockchain fork,

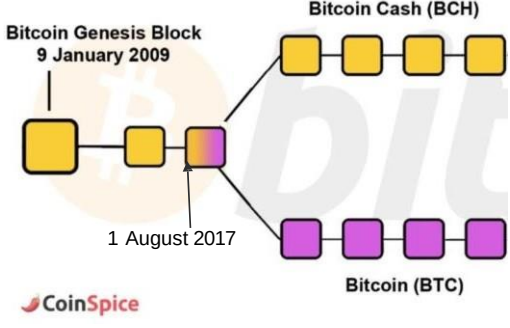


The growth rate of honest tree depth is reduced

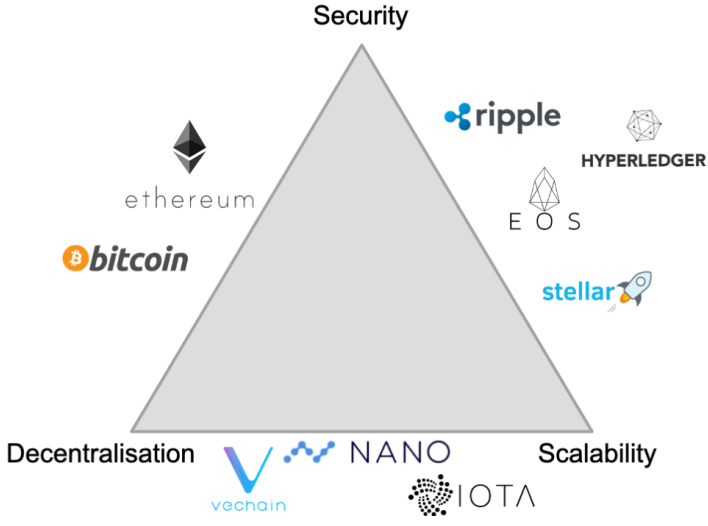


HARD FORK EXAMPLE: BITCOIN CASH

- Bitcoin cash is a hard fork in which the block-size is increased from 1MB to 32 MB.
- It results in 32 times increase in throughput.



THE SCALABILITY TRILEMMA.



Blockchain Trilemma – FINGO Blog

THE BITCOIN STORAGE BOTTLENECK

- Bitcoin full nodes need to keep all the transactions history, but this history is growing in volume.
- This makes people reluctant to involve in mining, which reduces security.

Google blockchain size graph

All Images News Videos More Settings Tools

About 1,470,000 results (0.44 seconds)

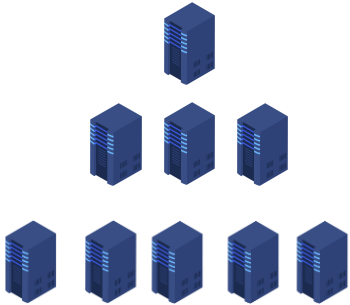
Size of the Bitcoin blockchain from January 2009 to March 27, 2021 (in gigabytes)

	Blockchain size in gigabytes
Mar 25, **2021**	327.61
Mar 24, **2021**	327.41
Mar 23, **2021**	327.21
Mar 22, **2021**	327

Converges to centralization of com

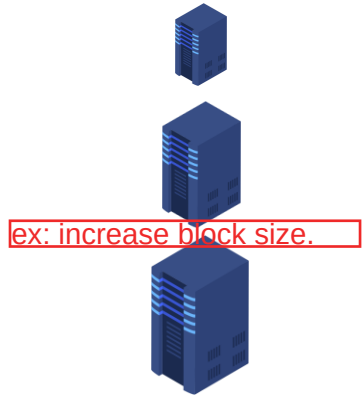
HORIZONTAL VS. VERTICAL SCALING

Horizontal Scaling
add more machines of the same computational power



ex: increase block rate.

Vertical Scaling
add more RAM/ CPU power to each existing machine



ex: increase block size.

LAYERS OF SCALING

- Layer 1 scaling refers to changing the blockchain itself (On-chain scaling)

ex: change consensus protocol

- Layer 2 scaling refers to pushing computation off the blockchain (Off-chain scaling)

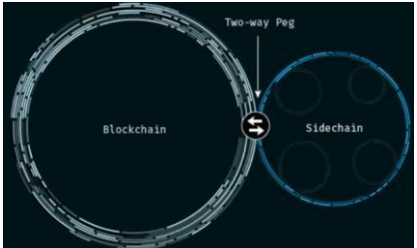
ex: move less important transactions to secondary blockc

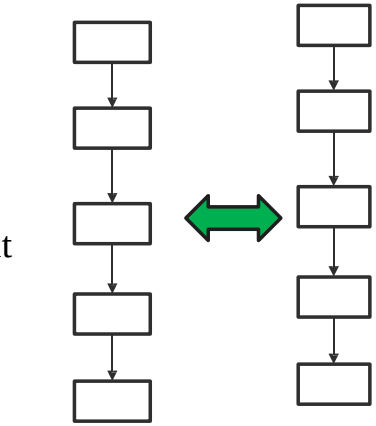
LET'S TRY HARDER TO SCALE BITCOIN

- In this lecture we present some other ideas to scale bitcoin
  - Side Chains
  - Payment Channels and the Lightning Network
  - Sharding
  - GHOST protocol
  - PRISM

SIDE CHAINS

- Using sidechains for less important transactions (e.g., buying a cup of coffee)
- Since we need less security here, block rate can be increased.
- It uses the main chain as a reference.
- Coins can be transferred between chains (current solutions mainly rely on central exchanges)

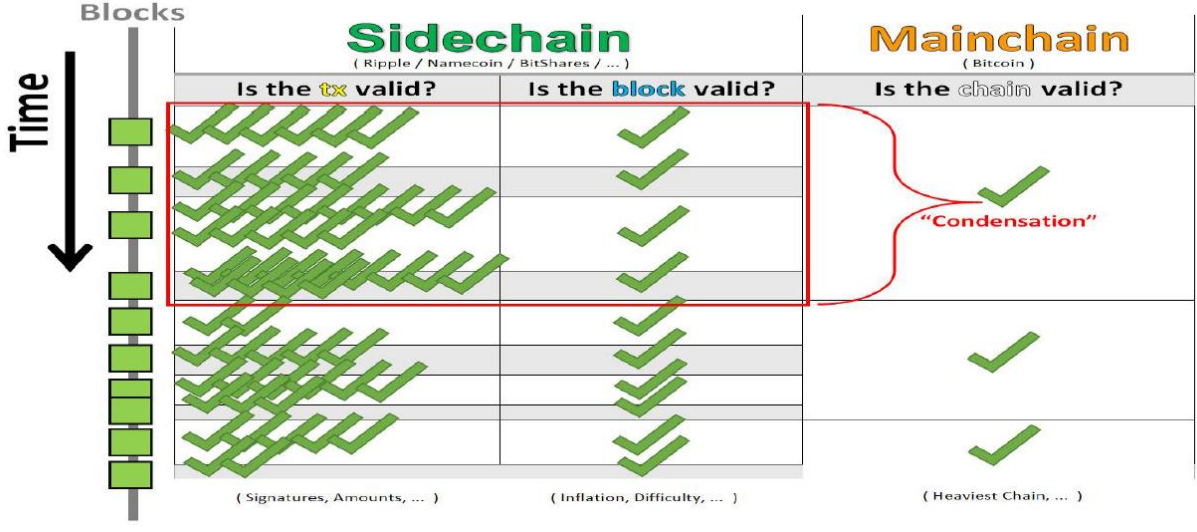




Main Chain

Side-Chain
Lower Security Requireme

ex: polygon is the layer 2 for ETH.



PAYMENT CHANNELS

implement using smart contracts.only two transac

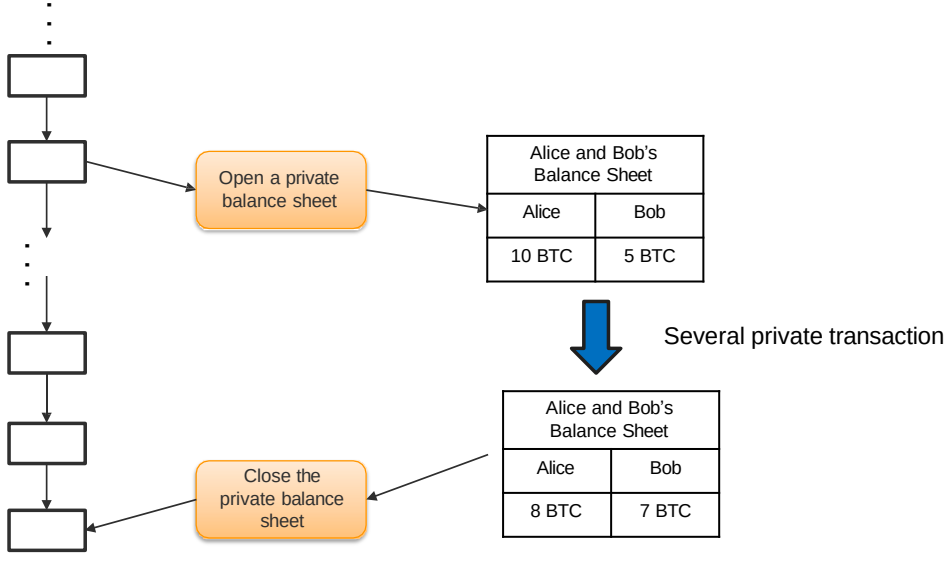
- The main idea: Instead of registering every transaction on the main chain, Alice and Bob maintain a *private balance sheet.*
- The main chain is only consulted when one party wants to settle.
- But they do not trust each other!

Alice and Bob's Balance Sheet	
Alice	Bob
10 BTC	5 BTC

Alice pays Bob 2 BTC

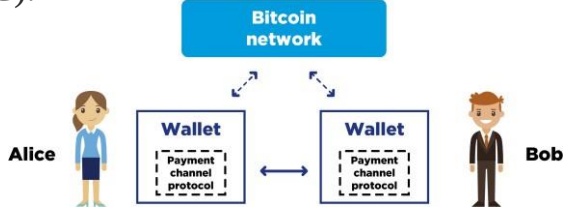
Alice and Bob's Balance Sheet	
Alice	Bob
8 BTC	7 BTC

PAYMENT CHANNELS



PAYMENT CHANNELS

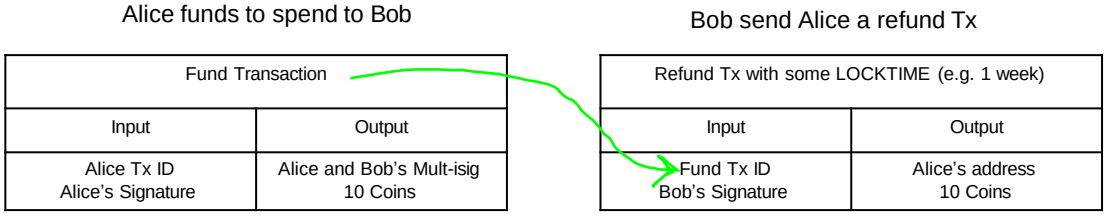
- Payment Channels are a framework to create block-chain enforceable contracts between the two parties to prevent cheating while maintaining the private sheet.
- It is built upon a technique called Hash Time-Locked Bidirectional Payment Channel (HTLC).



- Let’s first review the one-way channel.

1-WAY CHANNELS

- A channel is just a multi-sig output, where for a transaction to be valid 2-of-2 signatures are required.

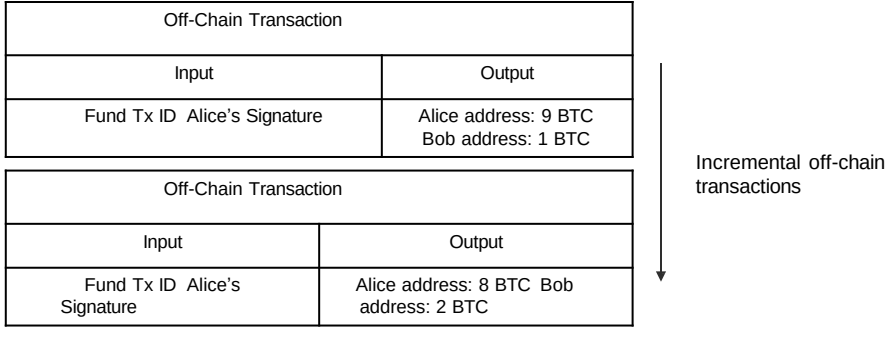


Alice doesn't broadcast the tx until Bob broadcasts.

- Bob should send the refund Tx to Alice before Alice sends the fund Tx to Bob.
- Nothing is broadcast on the network.

1-WAY CHANNELS

- Now Alice can spend the fund incrementally to buy something (e.g. coffee) from Bob.
- Bob does not need to broadcast the transaction on the network immediately.

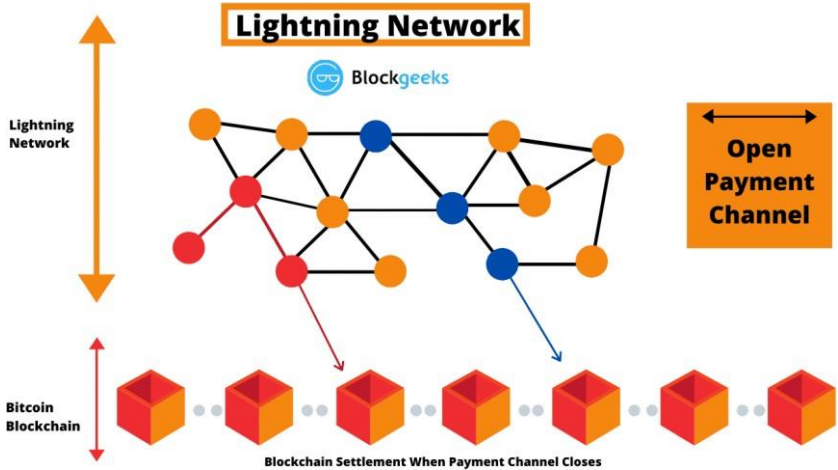


1-WAY CHANNELS

- Via the one-way channel, just Alice can send coin to Bob.
- Bob cannot send any money back in a credible way.
- Useful for some applications like paying fee for steaming video services.
- Bob should be careful to sign one of the half-signed transactions before the next week (when the locktime expires)!
- From the bitcoin network point of view just the fund transaction and one of these incremental transactions are seen.
- The channel is also alive only until the locktime expires.

THE LIGHTENING NETWORK

- A network of Payment Channels.

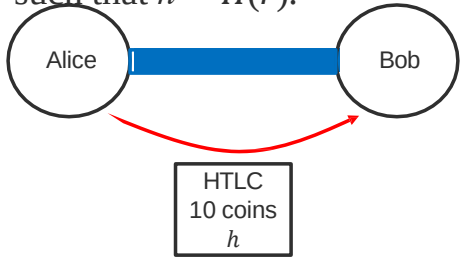


MULTI-HOP PAYMENT VIA HTLC

- There should be a guarantee that intermediate nodes do not cheat.
- Define

H: a hash function
r: a random number
$h = H(r)$

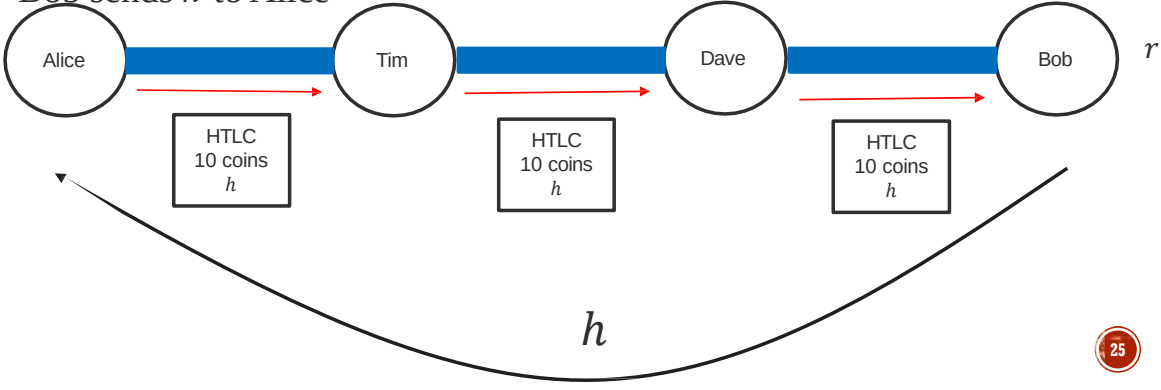
- If Alice sends HTLC of 10 coins with hash h to Bob, then Bob can receive it from Alice if in addition to his signature he provides the random number r such that $h = H(r)$.



Bob generates r privately, randomly and sends h to Alice

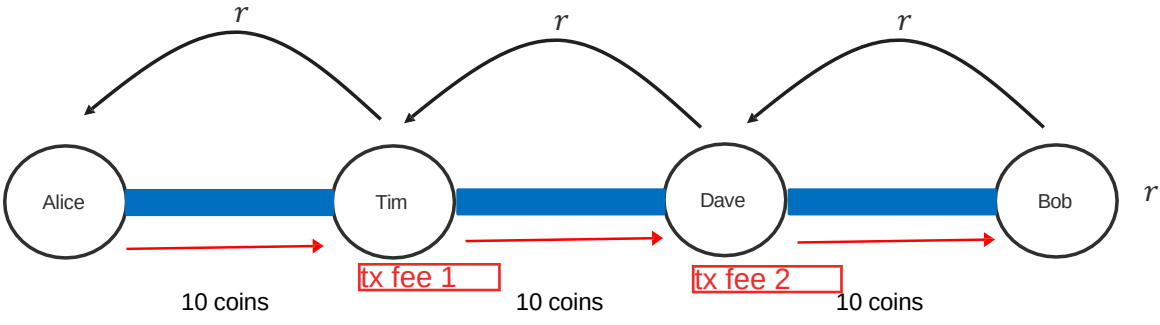
CHANNEL COMPOSITION

- Alice wants to send 10 coins to Bob.
- Bob chooses a random number r and computes $h = H(r)$.
- Bob sends h to Alice



CHANNEL COMPOSITION

- Then, by revealing r backwards the money is actually transferred.

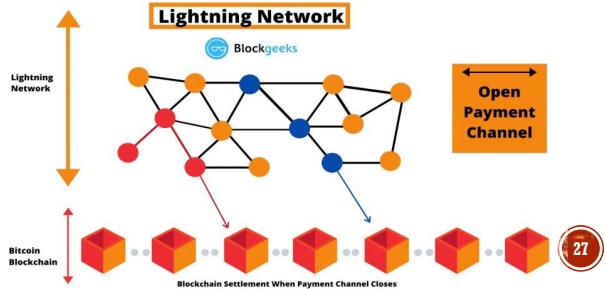


I have a lot of questions here.

MAIN LIGHTNING ROUTING CHALLENGES

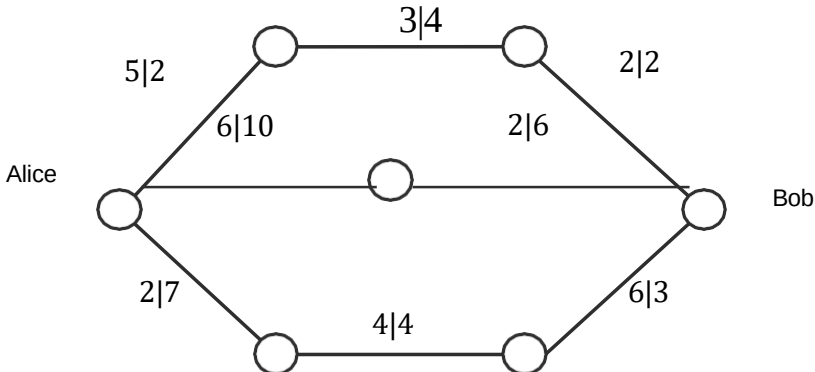
- The main challenge in routing is satisfying balance constraints.
- In practice, nodes only know channel capacities, but not the balances.
- One solution is finding shortest paths respecting the capacity constraints, and if balances are violated trying alternative routes.
- Continue this until the payment succeeds.

Also path with



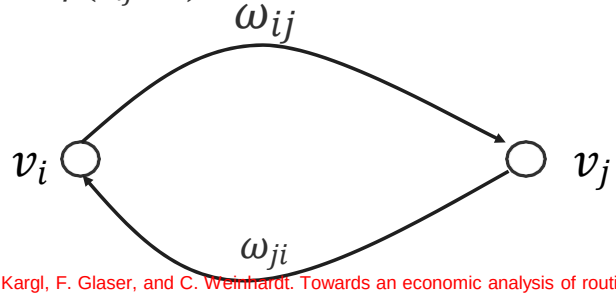
SPLITTING PAYMENTS

- Assume Alice wants to send 6 coins to Bob.
- She should split the payment into three 2-coin payments, via different paths.
- The challenge here is that Bob should only be able any fund when he has received the whole payment.



SINGLE-PATH SINGLE-TRANSACTION ROUTING FORMULATION

- Let's model the payment channel network by a graph $G = (V, E)$, where nodes are linked by directional edges, e.g. e_{ij}.
- The capacity from node v_i to node v_j is ω_{ij}.
- When a node v_i routes a transaction tx to the node v_j demands a financial reward $\rho(e_{ij}, tx)$.



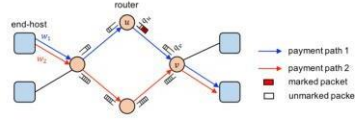
F. Engelmann, H. Kopp, F. Kargl, F. Glaser, and C. Weinhardt. Towards an economic analysis of routing in payment channel networks. In Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, page 2. ACM, 2017.

FORMULATION LIMITATIONS

- It only considers a single transaction.
- The solution may imbalance many links, which is a bad situation for coming transactions.
- Multi-path solutions are not included.
- Source node should know the network topology.
- Does not consider the dynamics of transactions.

ANOTHER ROUTING PROPOSAL: SPIDER NETWORKS

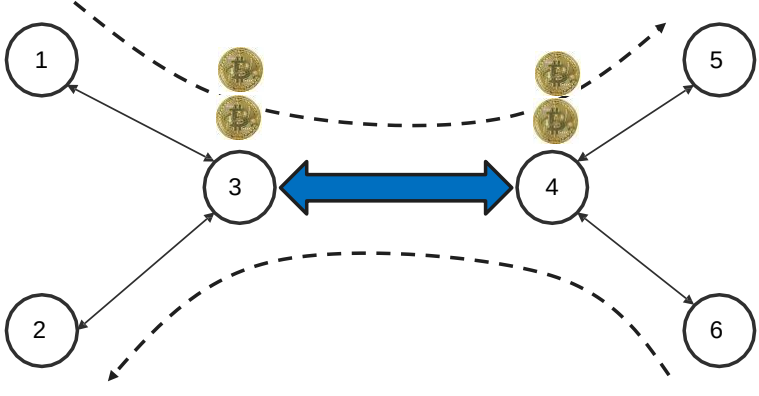
- Main Idea: Treating the payment network like a communication network.
- First each transaction is split into some transaction units (packetized payments)
- Treating the problem like a packet-switched network (instead of a circuit switched network.)
- Each node maintains a queue of un-relayed transactions.



V. Sivaraman, S. B. Venkatakrishnan, M. Alizadeh, G. Fanti, and P. Viswanath. Routing cryptocurrency with the spider network. 2018.

EXAMPLE: THE BENEFIT OF PACKETIZED PAYMENTS

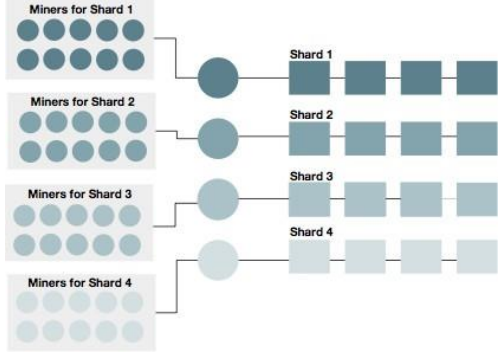
Node 1 wants to send 10 coins to node 5



Node 6 wants to send 10 coins to node 2

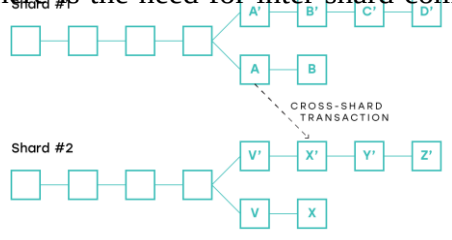
SHARDING

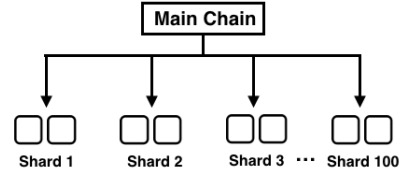
- Sharding is a well-known technique for scaling databases.
- The main idea is to partition the ledger into several disjoint sub-ledgers.
- Miners work on different shards.

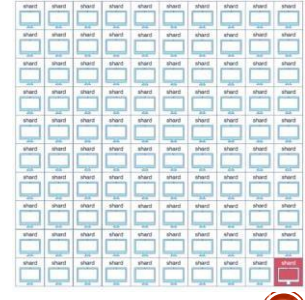


SHARDING CHALLENGES

- Adversary can take the control of a single shard and poison the whole network.
- How to assign nodes (miners) to shards? Need for a center to do this?
- If the assignment is static, adversary can gradually bribe the nodes in a single shard.
- The probabilistic and dynamic assignment seems to be the best option.
- There is the need for inter-shard communications.

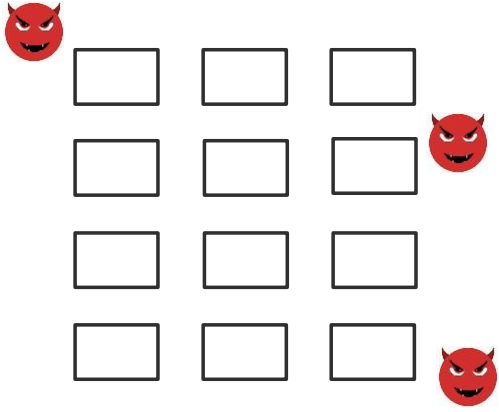


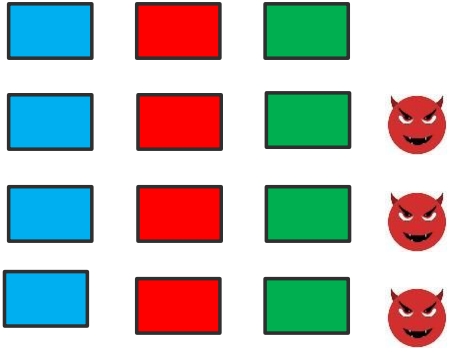




POLYSHARD: HOW TO USE CODING

Full replication versus shading



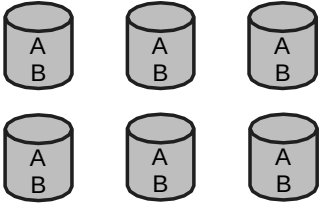


Security is damaged!

S. Li, M. Yu, C. Yang, A. S. Avestimehr, S. Kannan and P. Viswanath, "PolyShard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 249-261, 2021.

EXAMPLE: BENEFIT OF CODING IN STORAGE

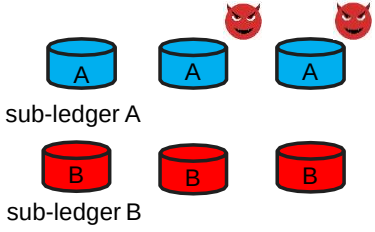
Full replication



storage ☹

security ☺

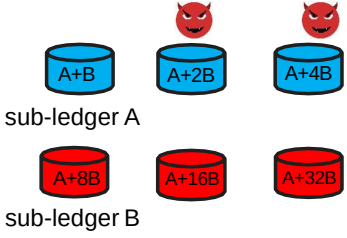
Naïve Sharding



storage ☺

security ☹

Coded Sharding



storage ☺

security ☺

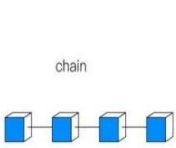
THE COMPUTATION ASPECT

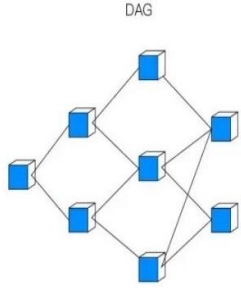
- In the bitcoin consensus framework miners do computations of ledgers. For example, checking the validity of signatures.
- The challenge with the coding solution is doing the computation on coded sub-ledgers. This is called coded verification.
- It is shown in their paper that this can be done if the computations are multi- variate polynomials.
- Every binary computation can be turn into a more complex polynomial computation (i.e., in a larger field size).
- One weak aspect of Poly-shard is the communication cost.

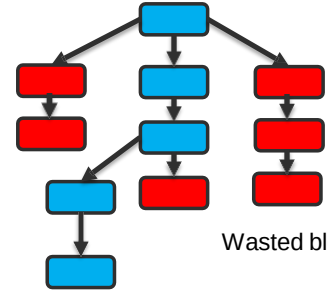
S. Li, M. Yu, C. Yang, A. S. Avestimehr, S. Kannan and P. Viswanath, "PolyShard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 249-261, 2021.

MAIN APPROACH IN DAG-BASED PROTOCOLS

- In Nakamoto Consensus many blocks are wasted.
- Also, in longest chain protocols forks reduce security, preventing increase of block rate.
- In DAG-based protocols, blockchain is treated as a DAG, and DAG topology determines the state of the ledger.

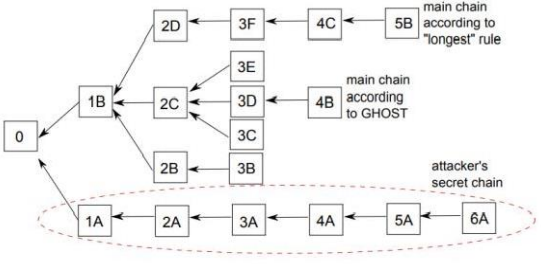


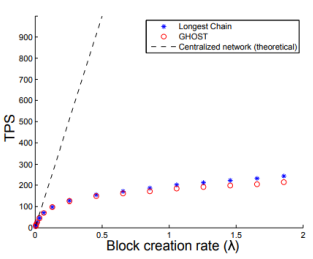


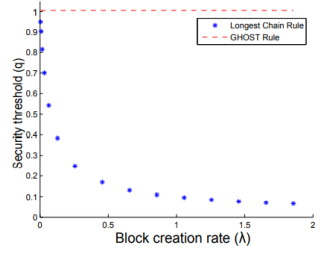


GHOST: GREEDY HEAVIEST OBSERVED SUBTREE

- The main idea is to change the BlockChain into a BlockTree by welcoming forks.
- Instead of following the longest chain, honest nodes follow heaviest subtrees
- That is because forking now is not considered just an adversarial action.
- This idea is used in Ethereum.



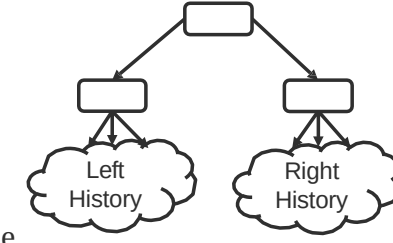




Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. Financial Cryptography and Data Security, 2015

THE BALANCE ATTACK ON GHOST

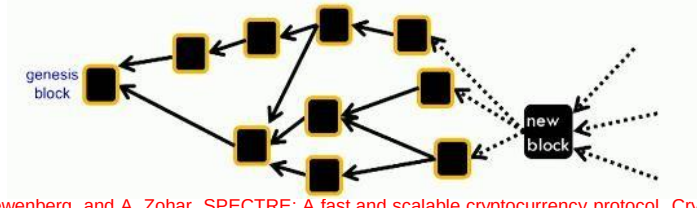
- GHOST is robust to 49% hash power private attack (for proof refer to their paper).
- Suppose that there are two blocks mined after the genesis block.
- The goal of the adversary is to maintain theses two possible version of history balanced.
- Adversary will privately mine on both sub-trees, and publishes blocks at appropriate time when the good time comes.
- Since the difference of the numbers of blocks is of order $\sqrt{\lambda_h}$, adversary has a relatively easy job.



Christopher Natoli and Vincent Gramoli. The balance attack against proof-of-work blockchains: The r3 testbed as an example. arXiv preprint arXiv:1612.09426, 2016

SPECTRE

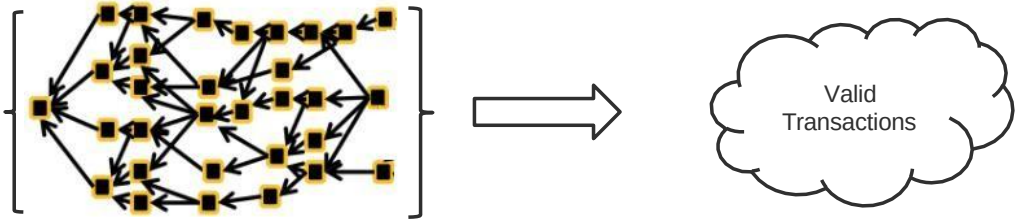
- The same as GHOST, the idea here again is to allow increasing block rate.
- However, in addition to forks, we allow (require) each new block to point to many other blocks.
- This changes BlockTrees (in GHOST) to BlockDAGs.
- It embeds much more information than previous structures.
- In fact each honest miner generates blocks pointing to the tips of the DAG
- From this topology, we are looking for agreement on a ledger.



Y. Sompolinsky, Y. Lewenberg, and A. Zohar. SPECTRE: A fast and scalable cryptocurrency protocol. Cryptology ePrint Archives

SPECTRE

- Each miner maps its local view of the graph topology to a set of valid transactions.

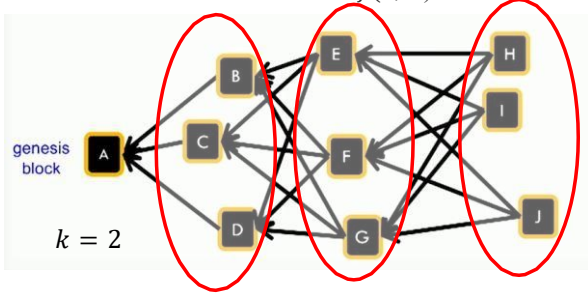


- For conflicting transactions, the one with more votes is the correct one.
- Vote to a specific block: the number of blocks which have a path to this block.

PHANTOM

- DAG generation rule is the same as SPECTRE.
- The key observations:
  - Two honest blocks are not connected if they are created approximately at the same time.
  - About k other honest blocks are created simultaneously where k depends on block generation rate λ and network delay Δ.

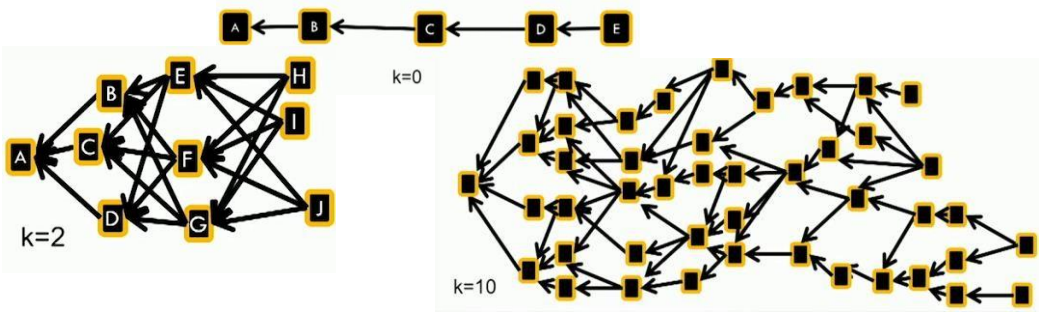
$$k = f(\lambda, \Delta)$$



Yonatan Sompolinsky and Aviv Zohar, A. Phantom, a scalable blockdag protocol. https://eprint.iacr.org/2018/104.pdf.

PHANTOM

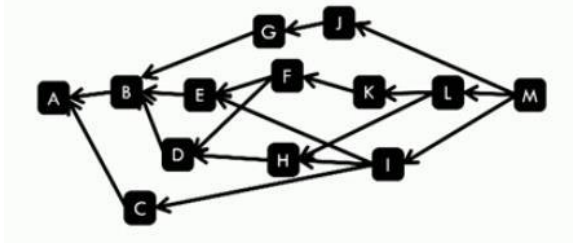
- The effect of changing the block rate:



- k is a known parameter.

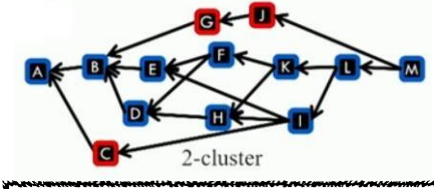
PHANTOM

- The goal: if we assume that the adversaries can do any attachment strategy, what will be the cluster of honest nodes?



PHANTOM

- The goal: if we assume that the adversaries can do any attachment strategy, what will be the cluster of honest nodes?
- The answer is detecting the *k*-clusters
- Definition: a *k*-cluster is a subset of blocks such that every block is connected to all other blocks, except for at most *k* blocks.
- But adversaries can also form *k*-clusters.
- Since we assume honest node have the main hash power, largest *k*-cluster should belong to honest nodes
- Then, based on a topological order on the cluster's blocks, one can check the transaction.

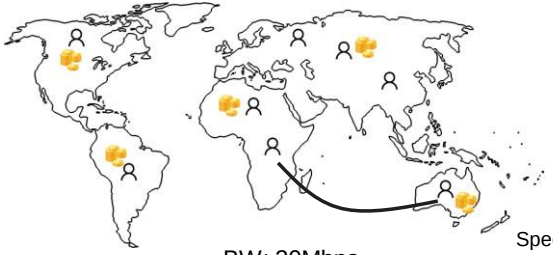


Algorithm (NP hard version):
1. Search for **largest *k*-cluster**
2. Order its blocks via some topological ordering
3. Iterate over blocks in the prescribed order, and accept transactions consistent with history

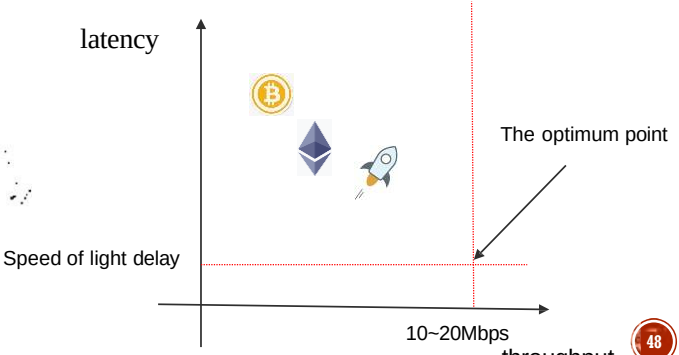
Yonatan Sompolinsky and Aviv Zohar, A. Phantom, a scalable blockdag protocol. https://eprint.iacr.org/2018/104.pdf.

PRISM

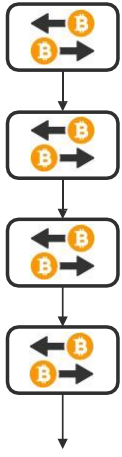
- Let us first determine the fundamental physical limits of any cryptocurrency's performance metrics.
- Available BW in current internet backbone is around 10~20 Mbps.
- Any communication cannot surpass the speed of light!



BW: 20Mbps
Delay: 1s



DECONSTRUCTING NAKAMOTO'S PROTOCOL



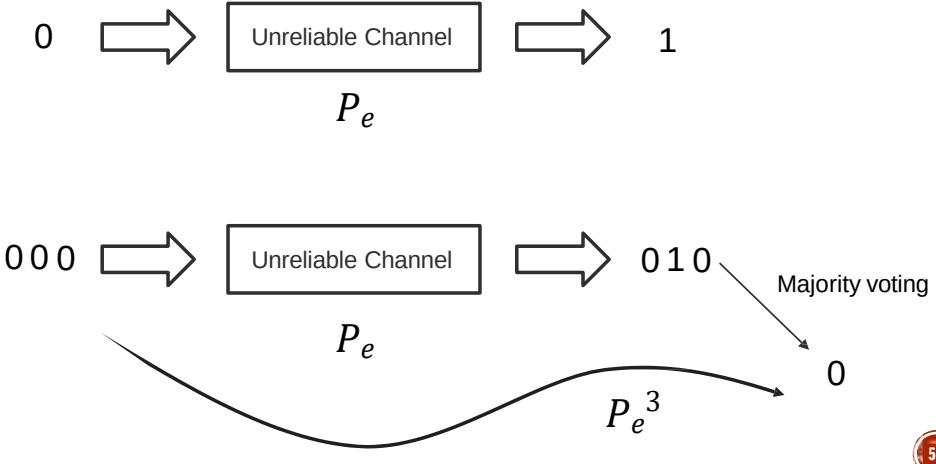
Proposing

Voting

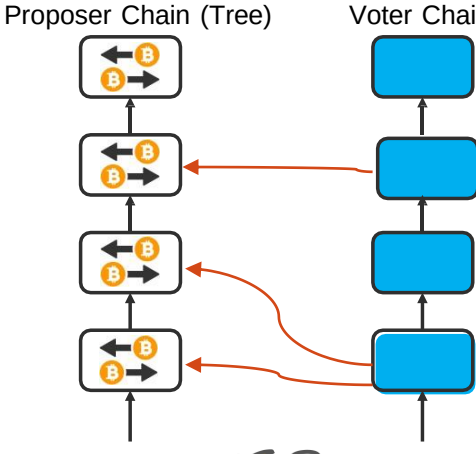
Each block has two roles

V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2019, pp. 585–602.

VERY SIMILAR TO THE REPETITION CODING IDEA!



DECONSTRUCTING NAKAMOTO'S PROTOCOL



Each voter block votes on all previous proposer blocks not voted on yet

Ledger Construction Rules:

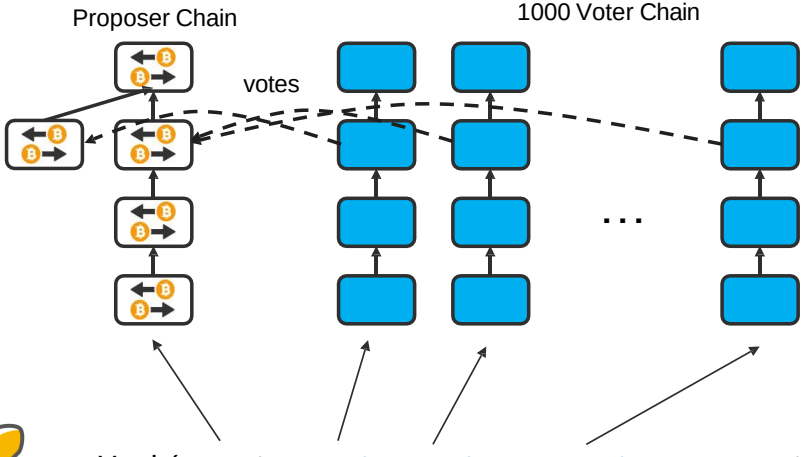
1. On Voter Tree: Select votes along the longest chain.
2. Order transactions based on votes.

Each full node mines on both the chains.

Latency is the same as before yet!

Still 6-block confirmation rule is required on the Voter Tree

DECONSTRUCTING NAKAMOTO'S PROTOCOL



No need for k-deep confirmation rule

Hash(parent, parent, parent, …, parent, nonce, contents)