DECOR+HOP: A Scalable Blockchain Protocol

Sergio Demian Lerner

Abstract

[1] Cryptocurrencies, based on and led by Bitcoin, have shown promise as infrastructure for pseudonymous online payments, cheap remittance, trustless digital asset exchange, and smart contracts. However, Bitcoin-derived blockchain protocols have inherent scalability limits that trade-off between throughput and latency and withhold the realization of this potential. This paper presents DECOR+HOP , a new blockchain protocol designed to scale. Based on Bitcoin's blockchain protocol, DECOR+HOP is Byzantine fault tolerant, is robust to extreme churn, and shares the same trust model obviating qualitative changes to the ecosystem. In addition to DECOR+HOP , we introduce several novel metrics of interest in quantifying the security and efficiency of Bitcoin-like blockchain protocols. We implement DECOR+HOP and perform large-scale simulation at 15% the size of the operational Bitcoin system, using emulated clients of both protocols. These experiments demonstrate that DECOR+ scales near optimally, with bandwidth limited only by the capacity of the individual nodes and latency limited only by the propagation time of the network.

1 Introduction

Bitcoin has emerged as the first widely-deployed, decentralized global currency, and sparked hundreds of copycat currencies. Overall, cryptocurrencies have garnered much attention from the financial and tech sectors, as well as academics, achieved wide market penetration in underground economies [32], reached a $12B market cap and attracted close to $1B in venture capital [13]. The core technological innovation powering these systems is the *Nakamoto consensus* protocol for maintaining a distributed ledger known as the blockchain. The blockchain technology provides a decentralized, open, Byzantine fault-tolerant transaction mechanism, and promises to become the infrastructure for a new generation of Internet interaction, including anonymous online payments [12], remittance, and transaction of digital assets [14]. Ongoing work explores smart digital contracts, enabling anonymous parties to programmatically enforce complex agreements [26, 49].

Despite its potential, blockchain protocols face a significant scalability barrier [45, 30, 17, 4]. The maximum rate at which these systems can process transactions is capped by the choice of two parameters: block size and block interval. Increasing block size improves throughput, but the resulting bigger blocks take longer to propagate in the network. Reducing the block interval reduces latency, but leads to instability where the system is in disagreement and the

[1] Portions of this abstract, the introduction, the model and the Bitcoin explanation in this paper has been borrowed in good faith from the Bitcoin NG paper, by Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer and Robbert van Renesse. I'm grateful to them for writing such clear descriptions. By copying big but non-novel chunks of their paper I'm helping the reader realize that the two papers present comparable technologies for scaling Bitcoin, so the reader of both papers can skip over those sections in this paper and focus on the innovative parts. Also copy-pasting allowed me to finish the paper for ScalingBitcoin on time before the deadline. Finally, researchers should not waste time in re-writing paper introductions, when good introductions have already been written. At least that is what open source is about.

blockchain is subject to reorganization. To improve efficiency, one has to trade off throughput for latency. Bitcoin currently targets a conservative 10 minutes between blocks, yielding 10 minute expected latencies for transactions to be encoded in the blockchain.[2] The block size is currently set at 1MB, yielding only 1 to 3.5 transactions per second for Bitcoin for typical transaction sizes.

Proposals for increasing the block size are the topic of heated debate within the Bitcoin community [41].

In this paper, we present DECOR+HOP a scalable blockchain protocol, based on the same trust model as Bitcoin. DECOR+HOP latency is limited only by the propagation delay of the network, and its bandwidth is limited only by the processing capacity of the individual nodes. DECOR+ achieves this performance improvement by sending the Bitcoin header before the transaction payload and by allowing miners to share the profit in case competing blocks are generated. This approach is not a significant departure from Bitcoin's operation, and DECOR+HOP maintains Bitcoin's security properties.

DECOR+HOP comprises two parts: DECOR+ is a reward sharing strategy. And HOP (acronym of Header Only Propagation) is a combination of different changes in block propagation protocol, including header first propagation (HFP) and Mining on unverified parents (MUP) also strangely referred as "SPV Mining". Both changes have been discussed many times in forums. Header first propagation splits block propagation in two stages: header only propagation and block data propagation. Header is propagated using a push model, not requiring a round-trip. Block data is propagated normally, but it can also be optimized by compressing the transaction information using ILBTs, synched indexes, as in The Fast Relay Network, or as a list of transaction hashes. MUP allows miners to be able to catchup with competing miners with a delay independent of the block size. Even if a faulty implementation of MWV has recently lead to a block chain fork, we argue that the problem was not MUP itself, but the fact that no time-out for MWV was being used, and this is imperative. Also we show that MUP is the only protection the network has to an attack to the Fast Relay Network, which is centralized.

Evaluating the performance and functionality of new consensus protocols is a challenging task. To help perform this quantitatively and provide a foundation for the comparison of alternative consensus protocols, we introduce several metrics to evaluate implementations of the Nakamoto consensus. These metrics capture performance metrics such as protocol goodput and latency, as well as various aspects of its security, including its ability to maintain consensus and resist centralization.

We evaluated the performance of DECOR+HOP on a discrete-event simulator consisting of 1000 nodes, amounting to over 15% of the current operational Bitcoin network [35]. This testbed enables us to run emulated clients, but using realistic Internet latencies. We compare DECOR+ with the original Bitcoin client, and demonstrate the critical tradeoffs inherent in the original Bitcoin protocol. Controlling for network bandwidth, reducing Bitcoin's latency by decreasing the block interval and improving its throughput by increasing the block size both yield adverse effects. In particular, fairness suffers, giving large miners an advantage over small miners. This anomaly leads to centralization, where the mining power tends to be used under a single controller, breaking the basic premise of the decentralized cryptocurrency vision. Additionally, mining power is lost, making the system more vulnerable to attacks. In contrast, DECOR+HOP improves latency and throughput to close to the maximum allowed by network conditions and node processing limits, while avoiding the fairness and mining power utilization problems. In summary, this paper makes three contributions. First, it outlines the DECOR+HOP scalable blockchain protocol, which achieves significantly higher throughput and lower latency than Bitcoin while maintaining the Bitcoin trust assumptions. Second, it introduces quantitative metrics for evaluating Nakamoto consensus protocols. These metrics are designed to ground the ongoing

[2] On average, assuming no backlog, both block interval and the average time to wait for a block starting at any time are ten minutes. This is a non-intuitive property of the memoryless exponential distribution.

discussion over parameter selection in Bitcoin-derived currency. Finally, it quantifies, through large-scale experiments, DECOR+HOP 's robustness and scalability.

2 Model and Goal

The system is comprised of a set of nodes $\mathcal{N}$ connected by a reliable authenticated peer-to-peer network. Each node can poll a random oracle [5] as a random bit source. Nodes can generate key-pairs, but there is no trusted public key infrastructure.

The system employs an associated puzzle system, defined by a cryptographic hash function H. The solution to a puzzle defined by the string y is a string x such that $H(y|x)$ — the hash of the concatenation of the two — is smaller than some target. Each node i has a limited amount of compute power, called *mining power*, measured by the number of potential puzzle solutions it can try per second. A solution to a puzzle constitutes a *proof of work*, as it statistically indicates the amount of work a node had to perform in order to find it. Most proof-of-work blockchains, with the exception of DECOR+HOP are vulnerable to selfish mining by attackers larger than 1/4 of the network [21]. The nodes are to implement a replicated state machine (RSM) [28, 44]. Properties of the system can be compared to those of classical consensus [40]:

3 Bitcoin and its Blockchain Protocol

Bitcoin is a distributed, decentralized crypto-currency [6, 7, 8, 37], which implicitly defined and implemented the Nakamoto consensus. Bitcoin uses the blockchain protocol to serialize transactions of the Bitcoin currency among its users. The replicated state machine maintains the balance of the different users, and its transitions are transactions that move funds among them. This state machine is managed by the system nodes, called miners.

Each user commands *addresses*, and sends Bitcoins by forming a transaction from her address to another's address and sending it to the nodes. More explicitly, a transaction is from the output of a previous transaction, to a specific address. An output is *spent* if it is the input of another transaction. A client owns x Bitcoins at time t if the aggregate of unspent outputs to its address is x. Transactions are protected with cryptographic techniques that ensure only the rightful owner of a Bitcoin address can transfer funds from it. Miners accept transactions only if their sources have not been spent, thereby preventing users from double-spending their funds. The miners commit the transactions into a global append-only log called the *blockchain*.

The blockchain records transactions in units of blocks. Each block includes a unique ID, and the ID of the preceding block. The first block, dubbed *the genesis block*, is defined as part of the protocol. A valid block contains (1) a solution to a cryptopuzzle involving the hash of the previous block, (2) the hash (specifically, the Merkle root) of the transactions in the current block, which have to be valid, and (3) a special transaction, called the *coinbase*, crediting the miner with the reward for solving the cryptopuzzle. This process is called Bitcoin *mining*, and, by slight abuse of terminology, we refer to the creation of blocks as *block mining*. The specific cryptopuzzle is a double-hash of the block header whose result has to be smaller than a set value. The *problem difficulty*, set by this value, is dynamically adjusted such that blocks are generated at an average rate of one every ten minutes.

Mining When a miner creates a block, she is compensated for her efforts with Bitcoins. This compensation includes a per-transaction fee paid by the users whose transactions are included, as well as an amount of new Bitcoins that did not exist before.

Forks Any miner may add a valid block to the chain by simply publishing it over an overlay network to all other miners. If multiple miners create blocks with the same preceding block,

the chain is *forked* into *branches*, forming a tree. Other miners may subsequently add new valid blocks to any of these branch. When a miner tries to add a new block after an existing block, we say it *mines on* the existing block. If this block is a leaf of a branch, we say he mines on the branch.

To resolve forks, the protocol prescribes on which chain the miners should mine. The criterion is that the winning chain is the *heaviest one*, that is, the one that required (in expectancy) the most mining power to generate. All miners add blocks to the heaviest chain of which they know, using the first branch it has heard of as tie-breaker, also called the first-seen rule.[3] The heaviest chain a node knows is the serialization of RSM inputs it knows, and hence describes the RSM's state. The formation of forks is undesirable, as they indicate that there is no globally-agreed RSM state.

Branches outside the main chain are called pruned, and blocks not in the best chain are called stale.[4] Transactions in pruned blocks are ignored. They can be placed in the main chain at any later time, unless a contradicting transaction (that spends the same outputs) was placed there in the meantime.

Block dissemination over the Bitcoin overlay network takes seconds, whereas the average mining interval is ten minutes. Therefore, accidental bifurcation is rare. It occurs on average once about every 60 blocks [16]. This value was true in 2013 and is still quite accurate today, in spite of the widespread use of the Fast Relay Network, which allow blocks to disseminate in hundreds of milliseconds.

4 First Problem: First seen rule and harmful competition

Miners that obey the Bitcoin protocol use the heaviest branch rule to choose the branch to mine on top of. If a miner is running the standard Bitcoin software, a self-mined block is not treated differently from a received block. Then, in case a miner hears about a block solved which competes with a locally solved block, it will generally prefer his own block, as a result of the first seen rule. Mining will continue in the selfish branch. There is nothing dishonest on this strategy since the miner has not enough information about which of the two forks is the one which the majority of the remaining miners are mining on top of. Nevertheless the division of hashing power in forks is against the common good: it reduces the network security and increases the network confirmation time due to reorganizations. In other words, clients need to wait more time to obtain the same confidence that a block will not be reverted. Also the first seen rule can lead to competing branches or longer length, if block on competing branches are produced at the near same time. The first seen rule favors the bigger miners, allows selfish mining and therefore is unfair.

5 Second Problem: Double-betting Strategy

The double-betting is a mining strategy that is also a consequence of the first-seen rule. Most mining pools create a candidate header in a centralized location, and then broadcast the header to all mining clients. Informing all clients that a new header is ready takes time in practice (the results cited below suggest that average time is 8 seconds). Then clients will actually keep mining on old parents even if the pool administration software has switched to a different branch. If a mining client solves a block during that period of time, the solved will be discarded by the pool software. However, it is evident that the best strategy for the pool administrator

[3] Choosing a longest branch at random is suggested in [21]. The operational client currently chooses the first branch it has heard of, making it more vulnerable (see [21] for details). However, it's clear that the best greedy strategy for mining pools is not first-seen rule, but the double-bet strategy as will be described in the following section.

[4] Often confusingly referred to as orphans in informal discussions, despite their having a parent in the block tree.

is to mine in the local branch, even if it is considered stale for the remaining miners. If a new block is added to the local branch before the remaining miners extend their branch, the reward is near doubled, since the local branch will be broadcast and accepted by the remaining nodes. Therefore, we conclude that if miners are not currently double-betting, they soon will, and mining pool delays will be a source of unfairness.

6 Third Problem: Latencies in Mining pools software stacks

Currently most miners are using the Matt Corallo Fast Relay Network to propagate blocks faster (an average of 300 msec between miners), so the appearance of stale blocks should be a rare event, at a rate of approximately 0.05% of the block rate. But in practice the actual stale block rate detected by block chain explorers is about 1.3%. It seems that client miners of the top mining pools receive notifications of a change in branch much slower than the inter-miner block propagation time. By registering simultaneously as a miner with several major mining pools, Gregory Maxwell has found that the average delay which the pool miner receives an update from the first pool to detect the change in the tip is 8 seconds in average, and the average time is takes for the second to last pool to inform the miner is 31 seconds. This data agrees with the current stale block rate.

The source of the bottleneck is not yet completely understood, and it's probably a conjunction of several independent delays in the pool administrator software stack. At the end, such delays only favors the bigger pools, and those pools which have optimized their (most times secret) software stacks. This is clearly unfair and it's a pressure for centralization.

7 Natural Reversal Probability

Any PoW based block-chain has a natural reversal probability for each branch length. The natural reversal probability is defined as the probability at a node that a best branch is reversed by cause of a fork and a reorganization, but where the network has no attackers, and everyone is rational, but acts in an uncoordinated way. We described how the first seen rule and the heaviest branch selection rule are the two main contributors to the Bitcoin natural reversal probability. But by dumping the first seen rule there is a trade-off between natural reversal probability and fairness that miners can make. If all miners could choose the same branch (of equally heavy branches) as the best-chain, then there would be no harmful competition. To choose the same branch, miners need to have the same block-chain state information (e.g. the full list of competing blocks). But as competing blocks are propagated and discovered by nodes, the best branch would flip between branches every time a new competing block is received, causing more branch reversals. Since propagation time is much lower than average block interval, these reversals are only one block in length, so only the natural reversal probability of length one increases. All remaining probabilities decrease considerably. As a consequence, increasing the block rate two-fold clearly increases fairness and decreases natural reversal probability for any length. The problem we solve in this paper is how to incentive competing miners to discard a self-solved block in favor of another miner's block. Our proposal creates reward distribution model where harmful competition events become irrelevant based on economic reasons. Also we'll show that at the same time we solve the Selfish Mining problem, rising the security of the network (back) to 50%.

8 DECOR+HOP

DECOR+HOP is a small variation of the standard Bitcoin blockchain protocol, but allows for better latency and bandwidth without sacrificing other properties.

DECOR+HOP is the combination of several protocol modifications:

- DECOR+: A new reward sharing scheme
- HFP: Push-model Header-first propagation of block
- MUP: Mining on unverified parents until a timeout.

Even if DECOR+ can be implemented without the remaining modifications, we find by simulations that the combination of them increases considerably the security and convergence of the network. Also, because DECOR+ requires the publication of uncle headers, adding the GHOST protocol requires little code changes, and it increases convergence in case of high rate blocks.

9 DECOR+ (DEterministic COnflict Resolution)

Currently in Bitcoin, when two or more miners have solved blocks at equal height, there is a clear conflict of interests. Each competing miner wants his block to be selected by the remaining miners as the block-chain tip, while the remaining miners generally would not mind which one is chosen. However all the remaining independent miners and honest users would prefer that all of them choose the same one, because this reduces the natural reversal probability. Also we want that the miners in conflict be able to choose the same parent also, and DECOR+ sets the right economic incentives for that choice, without requiring further interaction between miners. We define the block-chain conflict time as the average time interval where the network does not share a single block-chain state.

DECOR+, a is reward strategy that incentivizes economically resolving the conflict such that:

1. The conflict is resolved deterministically when all parties have access to the same block-chain state information.
2. The chosen resolution is the one that maximizes all miners revenue, both for miners in conflict and for the rest.
3. Resolving the conflict takes negligible time.

When DECOR+ is combined with header-first block propagation flooding, and assuming a random network graph, then the following property also hold:

1. The conflict time varies with the logarithm of the network diameter, and does not depend on the block size.

Because miners need to be well connected to the network to be able to propagate blocks efficiently, because resolving a DECOR+ conflict takes negligible time, and because the network propagated headers fast, the probability that a block solved during a conflict is low and can be ignored. Nevertheless, in case not all conflicting parties see the same network state, and party does not know about the existence of a competing block, DECOR+ punish this miner with a relatively low monetary punishment, so miners need not take special precautions.

In a nutshell DECOR+ strategy is to share the block reward between all miners that have solved a block of the same height. To help the explanation we'll first assume for a moment that all block rewards and fees are almost equal so each miner receives almost the same net payment for a block (there is no difficulty adjustment, nor subsidy halving). We'll also simplify the explanation by limiting ourselves to a conflict between two miners, and we'll later show this

is the most common case. Whenever two miners (Alice and Bob) mine two competing blocks (a block conflict) both decide to mine on top of the block with the highest reward, and if both have the exact same reward, the one with lower hash. This will be the conflict block selection rule. For miners to be able to compare competing blocks, all conflicting blocks headers that are not too old (e.g. no more than 6 steps back of the chain tip) are forwarded by the network. If a miner Carol (which could be also Alice or Bob) solves a following block, she can decide to include in her block a reference to the uncle block header that was left out of the main chain, can collect an extra prize when the coinbase of the conflicting block matures . The prices and rewards are computed according to the following steps:

1. The full reward of a conflicting block must pay a *forward pressure fee. This is variable and depends a difficulty adjustments occurring before the coinbase matures. If not such event occurs, this fee set to zero (more on this later). The forward pressure fee is burned.*
2. *After the forward pressure fee is subtracted, if the conflicting block and sibling headers do not obey the selection rule a punishment fee of* 20% *is subtracted. The punishment fee is burned. The aim of the punishment fee*
3. *After the punishment fee is subtracted,* 10% *of the remaining is the publishers fee, and it is shared in equal parts between the miners that included uncle headers.*
4. *After the publishers fee is subtracted, the remaining is the reward share and it is split in equal parts between the miners that solved the sibling blocks (including the miner of the block in the chain and the miner that solved the uncle headers).*

Example 1:

- Alice mines block A1 at height 1, with reward 28 BTC (25 BTC subsidy + 2 BTC fees)
- Bob also mines a block B1 a height 1, with reward 26 BTC (25 BTC subsidy + 1 BTC fees)
- Carol receives both blocks, and decides to mine on top of A1, because it has higher reward, obeying the selection rule.
- Carol mines the block C1 with parent block A1, and includes a reference to B1's header

After A1 block matures, A1's reward is shared in the following way:

- Since there is no difficulty adjustment in the middle of the maturity period, no forward pressure fee is paid.
- Since Carol had obeyed the selection rule, no punishment fee it paid.
- From the remaining 26 BTC, 10% (2.6 BTC) is payed to Carol, because she is the only miner which has included a block uncle.
- The remaining 23.4 BTC is split between Alice and Bob, and each one gets 11.7 BTC as reward.

The forward pressure creates an incentive to move forward in the block-chain in the cases where the network difficulty has increased to more than 200% of the difficulty in the previous interval (the maximum decrease is to 400%). In that case, miners can only create siblings or the last 6 blocks (prior siblings are not accepted). Then the forward pressure fee for the last 6 blocks is set to 50%. If the subsidy is low compared to transactions fees, there exist the possibility that one or more miners are willing to re-mine the last block to be able to create more siblings for the block before the last 6 blocks. This situation may already be a problem in Bitcoin, but should be analyzed further in the case DECOR+ is implemented. Setting the forward pressure fee to 50% for all immature blocks prior a difficulty adjustment greater than 200% should solve the problem.

The exact percentages for each fee can be changed. For example, to incentivize miners to have better network connectivity the punishment fee can be increased. To incentivize miners to broadcast uncle headers, the publishers fee can be increased. If the the sharing steps are performed as described (first forward pressure fee, then punishment fee, then publishers fee, and last sibling blocks sharing) then the right incentives will prevail.

9.1 DECOR+ when mining on top of unverified block headers

When DECOR+ is implemented with mining on top of unverified block headers in Bitcoin, there exists a drawback. A miner cannot apply the selection rule between two competing headers without the coinbase transaction that specifies the reward (subsidy+fees). This is however a rare case, since mining an empty block because of missing transactions is rare, then having two competing headers with missing transactions is much less probable. Nevertheless, several solutions are possible, such as transferring the coinbase transaction and a SPV proof along the header.

9.2 Implementation as a Soft-fork

When implemented in Bitcoin as a soft-fork, a reference to an uncle header can be stored in the scriptpub of an output of the coinbase transaction (e.g. in OP_RETURN payload), or in the coinbase field. To allow reward sharing, the coinbase transaction must pay new rewards to scriptpub OP_DROP OP_TRUE, so all coinbases scriptpub can be easily grabbed by miners. However a soft-fork prevents anyone from grabbing them. The block where a coinbase becomes mature must include a special Coinbase Split Transaction (CST) that splits the matured reward exactly as the DECOR+ rule establishes, based on previously published uncles and the dropped scriptpub scripts. A block missing the CST or having an erroneous CST is considered invalid.

9.3 Selfish Mining

Not possible. Explain why.

10 HOP

Bitcoin forwards each block by packing the block header with all the transactions contained in the block. This strategy, while being the most easy to analyze, is known to perform badly both regarding block propagation latency and bandwidth usage, which is doubled. Bitcoin miners partially solved this problem using the Fast Relay Network (FRN): this is a centralized backbone that relays blocks in a compressed form, and it is maintained by a single user. The FRN is provides more fairness to miners, as it prevents bigger mining pools for taking advantage of better network connectivity. However, for the same reason, the FRN can be a target of attack by a big pool. Since the FRN is a non-profit community service, the attack can take several forms, from DoS attacks to any human engineering techniques. To protect the FRN to be attacker, there must be a non-centralize network capable of comparable low latencies. We propose embedding the Fast Relay Network embedded into the network protocol using Push-model Header-first propagation of blocks (HFP) and Mining on unverified parents until a timeout (MUP).

10.1 Header-first propagation of blocks

We propose that blocks are sent in two stages: in the first stage only the block header is sent. Then the full block is sent (any block compression algorithm such as ILBTs also helps). Block

headers are verified at every node and if correct, they are pushed into peers without an INV round-trip. Headers up to 6 block old (as specified by its height) are broadcast. This gives plenty of time for miners to include stale headers as uncles in blocks. 6 hour Since the transactions in a block are generally already known to the network, there is no benefit in transmitting them again. Using 2SBP the channel capacity is doubled, allowing more transactions to be stored in each block. After each node has received the block header and the transaction hash list associated with the block header, the node attempts to reconstruct the block in order to verify it (fig. 3). The peer will fetch from a peer any transaction contained in the block but missing in his transaction pool.

10.2 Mining on unverified parents

Nodes can then start mining an empty block (coinbase only) on top of a header even if the transactions are still missing during a fixed interval. After that interval, they must resume mining with whatever block they were mining before. Most of the times no block is solved before the transactions arrive, and the does not solve an empty block. Even if the average block interval is reduced to 30 seconds, our simulations show that empty blocks are produced with very low probability and they do not affect the bandwidth and block-chain storage usage, because of their small size.

10.3 Simulations

We've simulated the block propagation using a discrete event simulation built specifically for this purpose. The simulator simulates the interaction between a small set of top-miners, each one in a random graph where the hop distance between them is near the average distance between nodes in the network. Even if this is not the worst case, since it is the best interest for top-miners to be well-connected, we assume miners perform not worse than the average. The simulated events are the creation of a block in one of locations and the propagation of the block to each of the other miner locations. The following results show the simulation with a 1 minute block interval and 30 TPS. The key result is that the network handles the increased transaction rate without problems and fairness is maintained.

11 Conclusion

This paper presents DECOR+HOP , a new blockchain protocol designed to scale. Based on Bitcoin's blockchain protocol, DECOR+HOP is Byzantine fault tolerant, is robust to extreme churn, and shares the same trust model obviating qualitative changes to the ecosystem. Simulations show that DECOR+HOP applyed to Bitcoin can whitstand a increase of the block-chain rate up to 1 minute average interval using a block size of 1 megabyte and therefore increase transaction rate to 30 tps whitout sacrifying network stability nor fairness.

References

[1] Andresen, G. O(1) block propagation. https://gist.github.com/gavinandresen/#file-blockpropagation-md, retrieved July. 2015.

[2] Aspnes, J. Randomized protocols for asynchronous consensus. *Distributed Computing 16*, 2-3 (2003), 165–175.

[3] Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timn, J., and Wuille, P. Enabling blockchain innovations with pegged sidechains. http://cs.umd.edu/projects/coinscope/coinscope.pdf, 2014.

[4] Bamert, T., Decker, C., Elsen, L., Wattenhofer, R., and Welten, S. Have a snack, pay with Bitcoins. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on* (2013), IEEE, pp. 1–5.

[5] Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security* (1993), ACM, pp. 62–73.

[6] Bitcoin community. Bitcoin source. https://github.com/bitcoin/bitcoin, retrieved Mar. 2015.

[7] Bitcoin community. Protocol rules. https://en.bitcoin.it/wiki/Protocol_rules, retrieved Sep. 2013.

[8] Bitcoin community. Protocol specification. https://en.bitcoin.it/wiki/Protocol_specification, retrieved Sep. 2013.

[9] BlockTrail. BlockTrail API. https://www.blocktrail.com/api/docs#api_data, retrieved Sep. 2015.

[10] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. Research perspectives on Bitcoin and second-generation cryptocurrencies. In *Symposium on Security and Privacy* (San Jose, CA, USA, 2015), IEEE.

[11] Buterin, V. Slasher: A punitive proof-of-stake algorithm. https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/, January 2015.

[12] CNNMoney Staff. The Ashley Madison hack...in 2 minutes. http://money.cnn.com/2015/08/24/technology/ashley-madison-hack-in-2-minutes/, retrieved Sep. 2015.

[13] CoinDesk. Bitcoin venture capital. http://www.coindesk.com/bitcoin-venture-capital/, retrieved Sep. 2015.

[14] Colored Coins Project. Colored Coins. http://coloredcoins.org/, retrieved Sep. 2015.

[15] Corallo, M. High-speed Bitcoin relay network. http://sourceforge.net/p/bitcoin/mailman/message/31604935/, November 2013.

[16] Decker, C., and Wattenhofer, R. Information propagation in the Bitcoin network. In *IEEE P2P* (Trento, Italy, 2013).

[17] Decker, C., and Wattenhofer, R. A fast and scalable payment network with Bitcoin Duplex Micropayment Channels. In *Stabilization, Safety, and Security of Distributed Systems - 17th International Symposium, SSS 2015, Edmonton, AB, Canada, August 18-21, 2015, Proceedings* (2015), Springer, pp. 3–18.

[18] Dwork, C., Lynch, N. A., and Stockmeyer, L. J. Consensus in the presence of partial synchrony. *J. ACM 35*, 2 (1988), 288–323.

[19] Eyal, I., Birman, K., and van Renesse, R. Cache serializability: Reducing inconsistency in edge transactions. In *35th IEEE International Conference on Distributed Computing Systems, ICDCS 2015, Columbus, OH, USA, June 29 - July 2, 2015* (2015), pp. 686–695.

[20] Eyal, I., and Sirer, E. G. Bitcoin is broken. http://hackingdistributed.com/2013/11/04/bitcoin-is-broken/, 2013.

[21] Eyal, I., and Sirer, E. G. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security* (Barbados, 2014).

[22] Garay, J. A., Kiayias, A., and Leonardos, N. The Bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II* (2015), pp. 281–310.

[23] Garcia-Molina, H. Elections in a distributed computing system. *Computers, IEEE Transactions on 100*, 1 (1982), 48–59.

[24] Hearn, M., and Spilman, J. Rapidly-adjusted (micro)payments to a pre-determined party. https://en.bitcoin.it/wiki/Contract, retrieved Sep. 2015.

[25] Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. Eclipse attacks on Bitcoin's peer-to-peer network. In *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015.* (2015), pp. 129–144.

[26] Kosba, A., Miller, A., Shi, E., Wen, Z., and Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. Cryptology ePrint Archive, Report 2015/675, 2015. http://eprint.iacr.org/.

[27] Kroll, J. A., Davey, I. C., and Felten, E. W. The economics of Bitcoin mining or, Bitcoin in the presence of adversaries. In *Workshop on the Economics of Information Security* (2013).

[28] Lamport, L. Using time instead of timeout for fault-tolerant distributed systems. *ACM Transactions on Programming Languages and Systems 6*, 2 (Apr. 1984), 254–280.

[29] Le Lann, G. Distributed systems-towards a formal approach. In *IFIP Congress* (1977), vol. 7, Toronto, pp. 155–160.

[30] Lewenberg, Y., Sompolinsky, Y., and Zohar, A. Inclusive block chain protocols. In *Financial Cryptography* (Puerto Rico, 2015).

[31] Litecoin Project. Litecoin, open source P2P digital currency. https://litecoin.org, retrieved Nov. 2014.

[32] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013* (2013), pp. 127–140.

[33] Miller, A., and Jansen, R. Shadow-Bitcoin: Scalable simulation via direct execution of multi-threaded applications. *IACR Cryptology ePrint Archive 2015* (2015), 469.

[34] Miller, A., and Jr., L. J. J. Anonymous Byzantine consensus from moderately-hard puzzles: A model for Bitcoin. https://socrates1024.s3.amazonaws.com/consensus.pdf, 2009.

[35] Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., and Bhattacharjee, B. Preprint: Discovering Bitcoins public topology and influential nodes. http://cs.umd.edu/projects/coinscope/coinscope.pdf, 2015.

[36] Moraru, I., Andersen, D. G., and Kaminsky, M. Egalitarian Paxos. In *ACM Symposium on Operating Systems Principles* (2012).

[37] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf, 2008.

[38] Nayak, K., Kumar, S., Miller, A., and Shi, E. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. *IACR Cryptology ePrint Archive 2015* (2015), 796.

[39] Pazmiño, J. E., and da Silva Rodrigues, C. K. Simply dividing a Bitcoin network node may reduce transaction verification time. *The SIJ Transactions on Computer Networks and Communication Engineering (CNCE) 3*, 2 (February 2015), 17–21.

[40] Pease, M. C., Shostak, R. E., and Lamport, L. Reaching agreement in the presence of faults. *J. ACM 27*, 2 (1980), 228–234.

[41] Peck, M. E. Adam Back says the Bitcoin fork is a coup. http://spectrum.ieee.org/tech-talk/computing/networks/the-bitcoin-for-is-a-coup, Aug 2015.

[42] Poon, J., and Dryja, T. The Bitcoin Lightning Network. http://lightning.network/lightning-network.pdf, February 2015. Draft 0.5.

[43] Sapirshtein, A., Sompolinsky, Y., and Zohar, A. Optimal selfish mining strategies in Bitcoin. *CoRR abs/1507.06183* (2015).

[44] Schneider, F. B. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys 22*, 4 (Dec. 1990), 299–319.

[45] Sompolinsky, Y., and Zohar, A. Accelerating Bitcoin's transaction processing. fast money grows on trees, not chains. In *Financial Cryptography* (Puerto Rico, 2015).

[46] Sompolinsky, Y., and Zohar, A. Secure high-rate transaction processing in Bitcoin. In *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers* (2015), pp. 507–527.

[47] Stathakopoulou, C. A faster Bitcoin network. Tech. rep., ETH, Zürich, January 2015. Semester Thesis, supervised by C. Decker and R. Wattenhofer.

[48] Swanson, E. Bitcoin mining calculator. http://www.alloscomp.com/bitcoin/calculator, retrieved Sep. 2013.

[49] The Ethereum community. Ethereum white paper. https://github.com/ethereum/wiki/wiki/White-Paper, retrieved July. 2015.

[50] Wikipedia. List of cryptocurrencies. https://en.wikipedia.org/wiki/List_of_cryptocurrencies, retrieved Oct. 2013.