Chainweb: A Proof-of-Work Parallel-Chain Architecture for Massive Throughput

Will Martino, will@kadena.io Monica Quaintance, monica@kadena.io
Stuart Popejoy, stuart@kadena.io

DRAFT v15

1 Abstract

Traditional Proof-of-Work (PoW) as demonstrated in the Bitcoin network is a masterstroke of design that created the first truly trustless public network. However, PoW has critical limitations that curtail its potential to empower a truly massive distributed economy and is justly criticized for its slowness, consumption of large amounts of energy, and congestion pricing in the form of high transaction fees. Existing PoW networks are uniformly characterized by very low throughput, and while improvements through protocol modifications have been made none have provided a practical path to approach the throughput levels of modern fiat-currency payment networks[1]. Moreover, efforts that seek to replace PoW with new consensus models like Proof-of-Stake (PoS) or integrate the protocol with off-chain networks and processes (payment channels, side chains) degrade the assurance, censorship resistance, and trustless nature of the original PoW design. To resolve these current limitations of PoW we present Chainweb, a parallel chain PoW architecture that combines hundreds or thousands of individually mined peer chains into a single network, capable of achieving throughput in excess of 10,000 transactions per second. Peer chains incorporate each other's Merkle roots to enforce a single super chain that offers an effective hash power that is the sum of the hash rate of each individual chain. Each chain in the network mines the same cryptocurrency which can be transferred cross-chain via a trustless, two-step Simple Payment Verification (SPV) at the smart contract level. In addition to massively increasing throughput, Chainweb also provides a significant increase in security, significantly reducing confirmation depth. We include the results of a loose probabilistic security analysis of several network configurations with up to 3243 individual chains, with a more comprehensive security study to follow in a companion paper.

[1] `https://cointelegraph.com/explained/bitcoin-scaling-problem-explained`

2 Overview

This paper will be divided into three sections; the first will summarize the Chainweb architecture by providing a short survey of prior work and detailing structural concepts that allow Chainweb to function. The second section details the Chainweb protocol, the structures that follow from the architecture and their implications for security, and assert a loose closed-form probability for a double-spend attack of the Chainweb network, with more comprehensive proofs to follow in a companion paper. Finally, we consider some of the implications of Chainweb as well as describing future suggested work.

3 Chainweb Architecture

3.1 Prior and Related Work

New developments in consensus designs like Proof-of-Stake (PoS) offer dramatic improvements to the traditional single-chain Bitcoin-style protocol in terms of throughput and confirmation latency. These advances have led networks such as Ethereum network to aspire to move away from PoW as soon as possible.[2] While PoS offers the attractive advantages of deterministic confirmation and a significant boost in performance, PoS is nonetheless fundamentally bounded by the causally consistent execution speed of the application layer, which creates a hard upper bound on throughput.[3] Though sharded PoS networks have been proposed that may further increase throughput, these changes move PoS towards a centralized system that begins to greatly resemble existing trustful financial networks.[4]

A potentially graver problem with PoS is the risk to the continued legal functioning of cryptocurrencies as predicated on the probabilistic censorship-resistance of the original PoW design, a feature that PoS designs fundamentally sacrifice by requiring distinct actors to stake funds in order to validate transactions. The exemption of money-transmitter (MTA) regulation to PoW miners (at least in the United States) stems directly from the probabilistic nature of confirmation and the lack of distinct rights for a given miner in the system: no single miner can be seen as confirming any transaction, since blocks must accumulate toward some indefinite confirmation depth, and no unique miner has the ability to influence the acceptance of a given transaction over any other.[5] In direct contrast, MTA

[2] https://www.coindesk.com/shifting-changing-ethereums-casper-code-takes-shape/

[3] King, Nadal 2012 http://www.peercoin.net/assets/paper/peercoin-paper.pdf

[4] Gao, Nobuhara 2017 http://journals.sfu.ca/apan/index.php/apan/article/download/225/pdf_137

[5] https://www.fincen.gov/news/news-releases/fincen-publishes-two-rulings-virtual-currency-miners-and-investors

regulations will easily apply to any staking design that designates distinct parties who participate in the deterministic confirmation of a given transaction.[6] A possible solution creates a PoS safe harbor like those found elsewhere[7] but would damage the egalitarian blockchain ethos by requiring a central authority to "erase" transactions. We maintain that staking designs put validators at risk of being subject to money transmitter regulation and enforcement as their unique identity and funds are essential to the effectuation of transfers in the system.

3.1.1 Payment Channels and Side Chains

Payment channels such as the Lightning Network are an additional approach to increasing throughput. In payment channels, funds are sequestered from the main network and are used in a series of smaller payments (or commitments) between a specified set of actors, with the ability to net out the payments on the main network at any time.[8] While there are attractive features of payment channels, we note that they do not fundamentally change the overall throughput of the system, as side-chain commitments are fundamentally not equivalent to transfers on the main network. For example, participants can steal from each other, requiring the victim to post the correcting transaction onto the main network before a timeout or lose the funds.[9] Also, the fact that funds must be pre-allocated for a channel imposes significant liquidity constraints which limits their availability to all but the largest stakeholders or those engaged in complicated multi-party arrangements.[10]

Side-chains, such as Liquid[11], have the same liquidity issues as payment channels but use an external, trustful consensus ledger to effect transactions. While this model is certainly effective for a subset of interactions, deployment to a full network introduces unnecessary compromises in several key elements of a blockchain because the involvement of external trust fundamentally changes the security model, permissioned architectures have scaling problems, and the MTA regulatory issues of PoS are still present should one actor validate a transaction on behalf of others. A PoW approach may be slower than protocols that use trust-based deterministic consensus, but PoW has no effective limit on the number of participants, while closed architectures face unavoidable performance problems as the number of consensus nodes increase.[12]

[6] https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ\#what-is-proof-of-stake

[7] e.g. DMCA Safe Harbor https://en.wikipedia.org/wiki/Online_Copyright_Infringement_Liability_Limitation_Act#Safe_harbor_provision_for_online_storage_-_\%C2\%A7_512(c)

[8] http://lightning.network/lightning-network-paper-DRAFT-0.5.pdf

[9] https://www.deadalnix.me/2017/02/27/segwit-and-technologies-built-on-it-are-grossly-oversold/ "With LN, the person you are transacting with can steal your funds at any time and it is up to you to publish a proof that you are being stolen from, within a given time frame."

[10] https://www.coindesk.com/lightning-technical-challenges-bitcoin-scalability/

[11] https://www.coindesk.com/its-happening-blockstream-launches-liquid-sidechain-in-beta/

[12] Martino 2016 http://kadena.io/docs/Kadena-ConsensusWhitePaper-Aug2016.pdf

3.1.2 PoW Parallel Chain Prior Art

Parallel-chain PoW architectures have been previously proposed to reap the benefits of security or deployment flexibility rather than for scalability. A proposal in 2014 sought to "decrease the odds of 51 percent attacks ever occurring" by "linking blockchains together to create a blockrope" with "each chain referencing the other" by incorporating each other's block hashes into their headers.[13] Another proposal sought to use a parallel chain as a "live beta" for Bitcoin by transferring coins from the main chain to a "beta" chain "in some UTXO-recognized way" so that it could be used "for real value transactions" to better exercise the beta.[14] While parallel-chain approaches were recognized as a possible scaling solution, the increased bandwidth usage was seen as a significant problem,[15] and the increased coordination required for hard-forks of multiple chains undesirable for Bitcoin.[16]

3.2 Chainweb Architectural Features

The novel architecture of Chainweb is predicated upon two separate, yet related features that operate at distinct layers of the Chainweb stack: 1) cross-chain cryptocurrency transfers via on-chain SPV smart contracts and 2) parallel-chain binding at the hashing level via peer-chain Merkle root inclusion. The former, which occurs in the application (smart contract) layer, leverages the latter to create valid Merkle proofs of currency transfer.

In this section, we will first detail how to enable globally "mass-conserving"[17] cross-chain transfers of cryptocurrency via SPV. This implementation is necessary to avoid per-chain floating currencies that would require dedicated exchange markets to move value between individual chains.

Next, we describe the protocol by which parallel chains are bound together to form a Chainweb network. The protocol itself does not impose an upper bound on network size, and is instead constrained theoretically by existing global IP infrastructure and bandwidth and practically by necessity-Chainweb configurations with throughput in excess of 100,000 transactions per second are not currently necessary.

[13] Blanger 2016, "Mt. Gox, Failure and Opportunity" `https://web.archive.org/web/20140610064048/http://www.belmarca.com/2014/02/26/mt-gox-failure-and-opportunity/`

[14] `https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-October/003351.html`

[15] `https://bitcointalk.org/index.php?topic=1060430.0`

[16] Back et al 2014 `https://blockstream.com/sidechains.pdf`

[17] A euphemism from physics used here to describe a net-zero transfer system like Bitcoin

3.2.1 Simple Payment Verification (SPV) History and Application

In the Bitcoin paper, Nakamoto introduces SPV as a way "to verify payments without running a full network node." This transaction is accomplished by obtaining "the Merkle branch linking the transaction to the block it's timestamped in", and "linking" it to a Merkle root obtained from the block header stream of the "longest chain." The need to determine the longest chain adds some insecurity to the process, for instance if "the network is overpowered by an attacker." Nakamoto concludes by observing that "businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification."[18]

SPV is mainly used for user wallets, such as Electrum.[19] The insecurity of so-called "thin clients" leads most server-side deployments requiring verification to run full nodes[20], in line with Nakamoto's observation. The original "pegged-sidechain" design used SPV to manage cross-chain transfer of ownership, but sidechain projects have since adopted other pegging techniques, with a common concern being the inability to process an SPV proof without changes to Bitcoin script.[21] The Ethereum-based Raiden network incorporates Merkle proofs tangentially but is mainly focused on "hash-locked transfers" similar to those used in the Lightning network.[22]

Chainweb manages a single currency across multiple chains, which requires the ability to move liquidity across chains in a trustless manner, destroying coin on one and creating it on the other. To enable this, SPV proofs of deletion on one chain are provided to the creating chain to be validated by built-in functionality in the application layer. The capture and incorporation of peer Merkle roots in Chainweb serves to provide a trustless oracle of the longest chain of a given peer to the SPV validation process. For transfers from peers that are not directly referenced by the creating chain, the final Merkle proof must also represent the intermediate Merkle roots that are captured in a directly-referenced peer root.

3.2.2 Chainweb Inter-Chain Transfers

A Chainweb transfer moves coin by deleting it in an account on one chain and creating it in an account on the other. Here we describe the steps in the transfer process.

In this description, coin is deleted on chain 1 from account A and then created on chain 2 in account B. *Receipts* are transaction records that are validated in SPV Merkle proofs;

[18] Satoshi Nakamoto 2008, https://bitcoin.org/bitcoin.pdf

[19] `http://docs.electrum.org/en/latest/faq.html\#does-electrum-trust-servers`

[20] `https://en.bitcoin.it/wiki/Thin_Client_Security\#Thin_Clients`

[21] `https://uploads.strikinglycdn.com/files/ec5278f8-218c-407a-af3c-ab71a910246d/RSK\%20White\%20Paper\%20-\%20Overview.pdf`

[22] `https://raiden-network.readthedocs.io/en/stable/spec.html`

values recovered from receipts are stored in protocol-reserved fields to prevent spoofing. An SPV proof cannot be validated until the block is recorded in the opposite chain (and on any intermediate chains), so the initiator of the second step must wait until the corroborating Merkle roots are published and confirmed.

1. **Chain 1 - Delete:** User signs and publishes transaction, calling $cx - delete$ with arguments A (delete account on 1), Y (create chain), B (create account on 2), Q (transfer quantity). $cx - delete$ performs the following:
   (a) Enforce A keyset against signature.
   (b) Enforce sufficient funds to delete Q in A.
   (c) Delete Q from A.
   (d) Receipt records X, Y, B, Q, T (transaction ID) in protocol-reserved fields.
2. **Chain Y - Create:** Anybody publishes transaction, calling $cx - create$ with SPV proof and receipt of deletion transaction on chain 1. $cx - create$ performs the following:
   (a) Validate SPV proof of deletion transaction, recovering X, Y, B, Q, T from receipt.
   (b) Enforce unique usage of (T, X).
   (c) Enforce Y identifies to this chain, and B is a valid account on chain.
   (d) Create Q in B.

Some observations on this process:

Create must first prove delete. The main requirement of a Chainweb transfer is to create coin only once you can prove the corresponding deletion has provably occurred. Thus, a create must accept and validate an SPV proof of a previous delete for the same quantity.

A delete must only allow one create. The tuple of (T, X) forms a unique token that can only be used once by a given chain to create coins. The deletion's transaction ID can only be consumed once, and this feature is enforced by the chain on which the creation occurs tracking which Transaction IDs have been consumed. As the delete step dictates onto which chain the create step can occur, mass is conserved.

The owning user need only publish and sign the first. An opportunity thus arises for businesses to offer clearing services to handle the subsequent steps. Note that full on-chain automation of this process is not feasible as it would require chains to be aware of transaction data on other chains, which is not part of the Merkle-root exchange.

UTXO or Account-Model. This process works equally well for either UTXO or Account-Model style verification, though the public blockchain released by Kadena will operate with Account-Model verification.

4 The Chainweb Protocol

The Chainweb design embraces parallelism as a way to linearly increase throughput with each additional chain added to the network. A scaled Chainweb network can run thousands of chains simultaneously, increasing throughput to satisfy the requirements of modern electronic financial payment systems. Greater bandwidth requirements are necessary and even desirable, which we see reflected in the increased utilization of current blockchain networks.[23] As a PoW architecture, Chainweb operates within the MTA exception under which miners currently operate and avoids the security and liquidity tradeoffs of payment channels or side-chains. Chainweb also serves to mitigate the worrisome energy footprint of current mining operations by distributing competition across many chains and reducing spurious competitive mining. The increase in attack-resistance offered by the multiple-chain architecture also significantly lowers the required per-chain hashrate, while the use of hashrate to support additional chains serves to increase throughput and utilization, not just security. Chainweb makes significantly more efficient use of hashrate than a single-chain PoW design.

4.1 Overview

The Chainweb network is comprised of multiple independent peer blockchains minting distinct coins of the same currency. Each chain incorporates a subset of other peers' Merkle roots in its own block hashes. The capture of foreign roots serves two purposes. First, it allows a given chain to validate that its peer chains are maintaining a consistent fork by locating its own previous Merkle roots in those obtained from its peer chains. Second, it provides a trustless oracle of peer Merkle roots, which is necessary to allow application-layer transfer code to validate provided Merkle proofs to guarantee cross-chain transfers of funds.

The cross-referencing of Merkle roots serves to increase hostile-fork resistance as a function of the number of roots referenced. To replace a given block, an attacker must fork all chains that directly or indirectly reference that block beginning at the point that the reference occurs. Once full coverage is reached (i.e. every chain in the web references a block directly or indirectly) the attacker must fork every chain in the network. Effectively, the hashrate

[23] Bloomberg 11/14/17, "Bitcoin's High Transaction Fees Show Its Limits" `https://www.bloomberg.com/view/articles/2017-11-14/bitcoin-s-high-transaction-fees-show-its-limits`

of the network is the sum of the hashrate of each peer, the combined power of which an attacker must overwhelm in order to guarantee acceptance of a fraudulent fork. The protocol itself consists of three elements: generation of probabilistic assurance, determining required peer references, and validating peer references.

4.2 Terminology

In a traditional blockchain, where there is only one chain, each new block must only reference the header of its previous block. In Chainweb, each parallel blockchain must additionally reference the headers of other chains (peers) at the same block height as its previous block. The base graph is used to structure the interaction of chains in the web, and can be thought of as the instructions for how to braid the chains together. A layer is the set of each chain's block at a given block height (cross-section of the braid).

4.3 Peer Header Relationships as Defined by the Base Graph

Each vertex in the base graph represents a chain in the network. The order of the graph (number of vertices) defines the total chain count while also describing the overall throughput of the network. The degree of the graph defines how many previous headers of peers a given chain must reference, while the edges of a vertex indicate the specific peers for which a given chain must reference the previous block. The diameter of the graph defines the maximum number of inter-chain hops required to construct a Merkle proof between any pair of chains, how many subsequent layers a given block requires to be fully braided and, in terms of block height, the maximum a given chain can advance (or fall behind) any peer in the web.

When a block is processed for any given chain, the peer headers found in that block's header are committed to a storage location available to the application (smart contract) layer. In the Kadena public blockchain implementation this storage location will be a table guarded by a cryptocharter[24] committed at genesis.[25] The available peer headers are part of the consensus level and thus they are trustlessly assured at the application layer. Therefore, users can construct a Merkle proof between any two chains that covers, at most, a one

[24] A smart contract that syntactically requires an upgrade governance mechanism. In Pact, this mechanism can be autonomous (e.g. the mechanism in the Ethereum EVM where a hard fork is required to upgrade a given contract), centralized (e.g. a specific set of signature capabilities that is required to enact an upgrade), decentralized (e.g. the cryptocharter tally's votes for an upgrade transaction's hash) or a mixture of the aforementioned.

[25] For Kadena Public, the fundamental cryptocurrency itself is defined by a cryptocharter committed in the genesis block. Moving the definition of the coin to the Pact smart contract layer allows for its formal verification.

less than diameter number of cross-chain hops, as the last hop is available via query in the application layer.

The design of Chainweb seeks to drastically increase the efficiency of PoW by increasing the throughput of the network while keeping the hashrate constant, so we focus on graphs that are found in the best known solutions to the Degree-Diameter Problem for Undirected Regular Graphs found in graph theory.[26] As these solutions are the largest order graphs for a given degree and diameter, they maximize the throughput of the overall network while minimizing the number of hops required to construct a cross-chain Merkle proof.

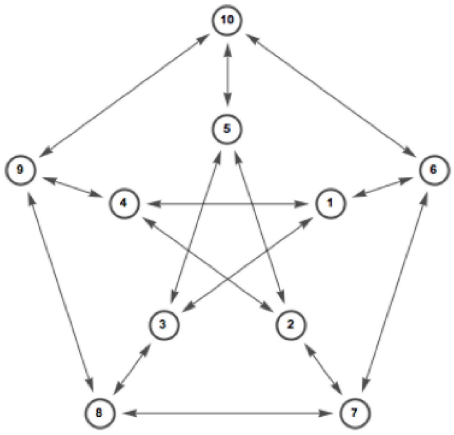

Figure 1: Petersen Graph (Order 10, Degree 3, Diameter 2)

Using the Petersen graph[27] [Figure 1] as the base graph and taking the example block of *(Chain 1, Layer N)* as a starting point, we see the below header references for previous layers. Using these references, one can construct a Merkle proof which the SPV system can directly validate for any transaction in the network that has occurred in *layer* $\leq N-2$. This process doubles as a mechanism to near-exponentially increase cryptographic assurance.

This process quickly becomes difficult to represent without the use of three-dimensional graphics. As such, Figure 3 visualizes the base graphs, full braids, and the forward propagation of Merkle roots across subsequent layers for the four smallest-possible Chainweb configurations. However, the Chainweb protocol is fully general for, at the very least, all undirected, regular base graphs.

[26] `https://en.wikipedia.org/wiki/Table_of_the_largest_known_graphs_of_a_given_diameter_and_maximal_degree`

[27] Petersen, Julius (1898), "Sur le thorme de Tait", L'Intermdiaire des Mathmaticiens, 5: 225-227

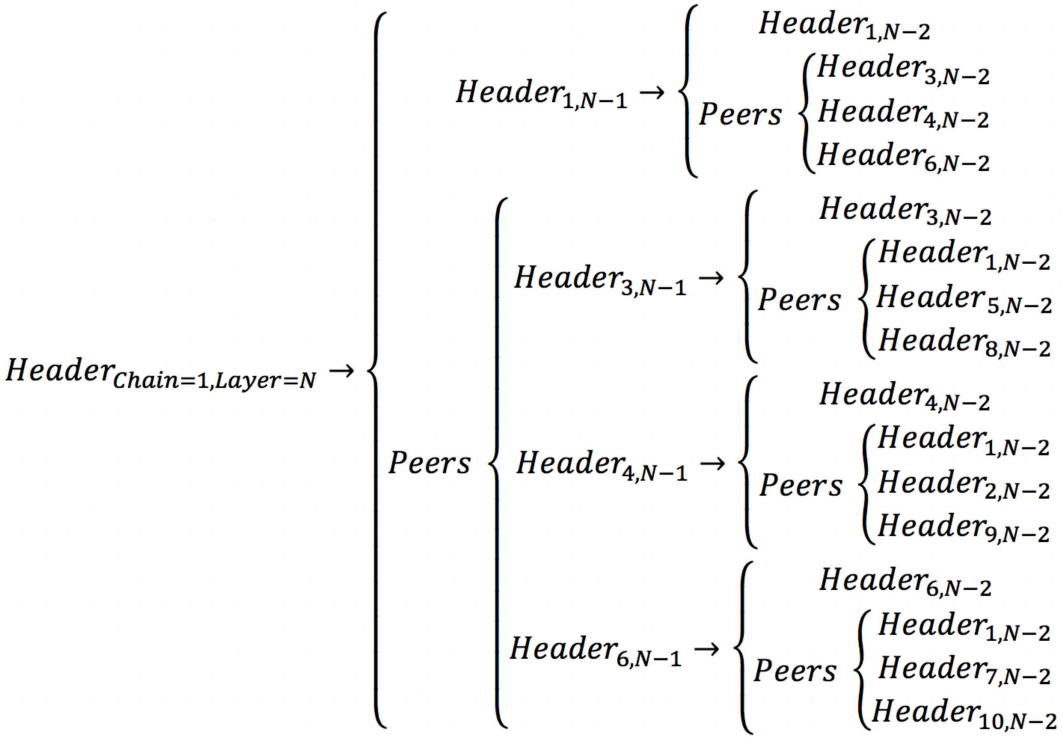

Figure 2: Propagation of header references

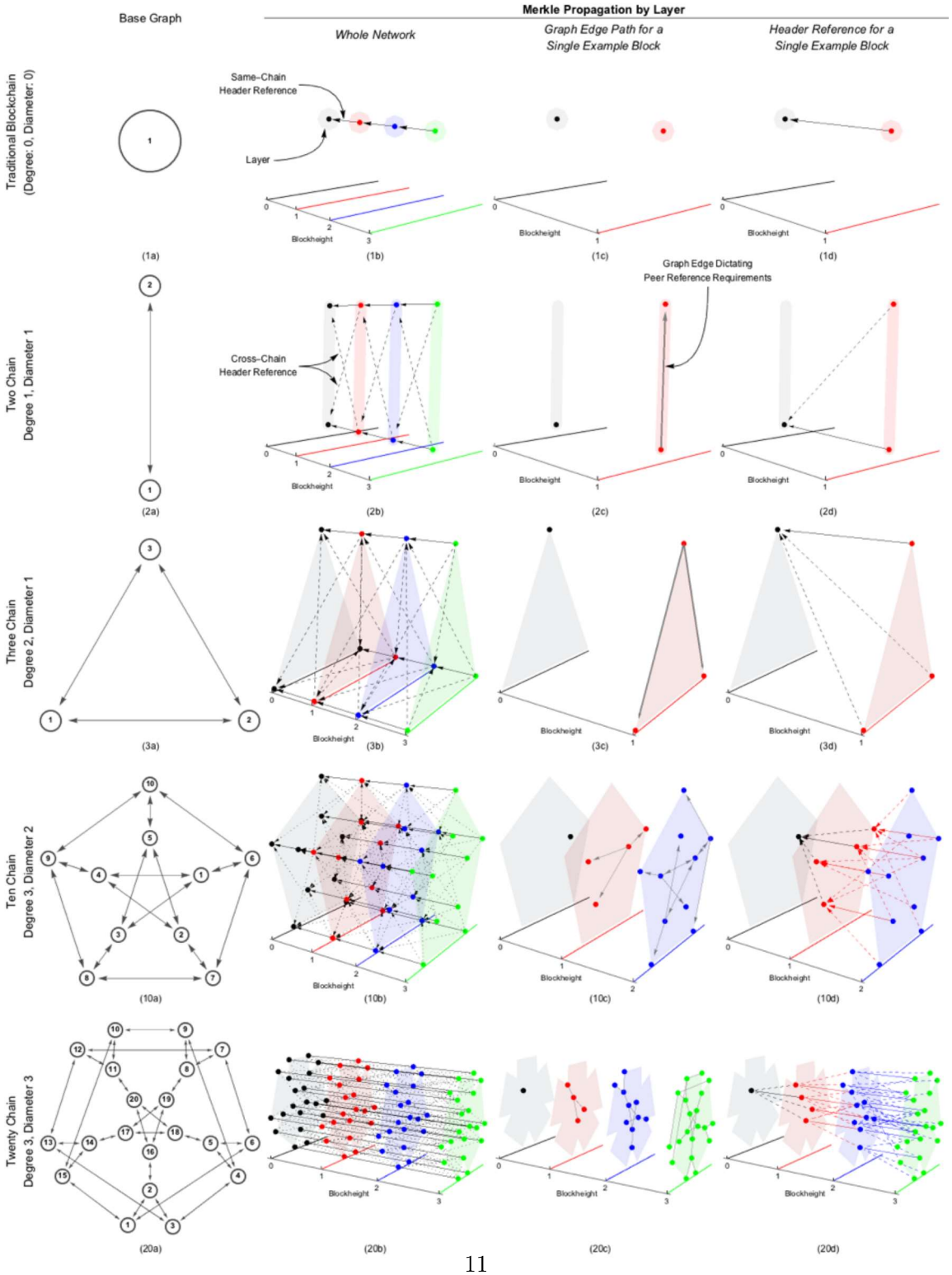

Figure 3: Sample Chainweb configurations

4.4 Peer Reference Protocol Requirements

Beyond the peer references dictated by the base graph, Chainweb must guarantee that, to replace any given block in the network, all blocks that currently exist within the future Merkle cone of that block must be replaced. This feature is required to ensure that Chainweb network cross-chain transfers conserve cryptocurrency mass. Two peer header rules, each of which dictate how two header reference paths for a pair of neighboring chains must terminate as the same header, enforce the same history requirement.

Same Chain Rule requires that the header of a chain and the headers of referenced peers agree on the ancestry of the header of the chain.

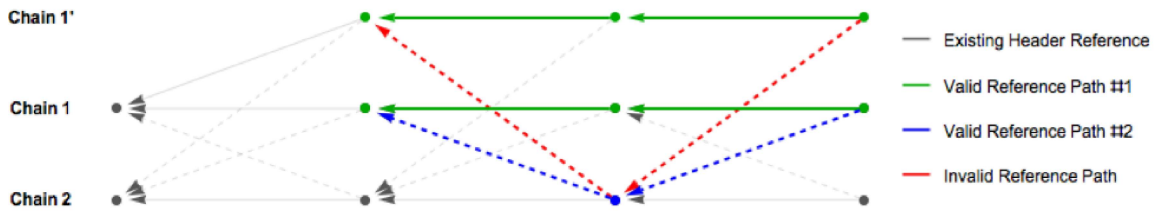

Figure 4: Same chain rule

Same Peer Rule requires that the header and the headers of referenced peers agree on the ancestry of the referenced peer.

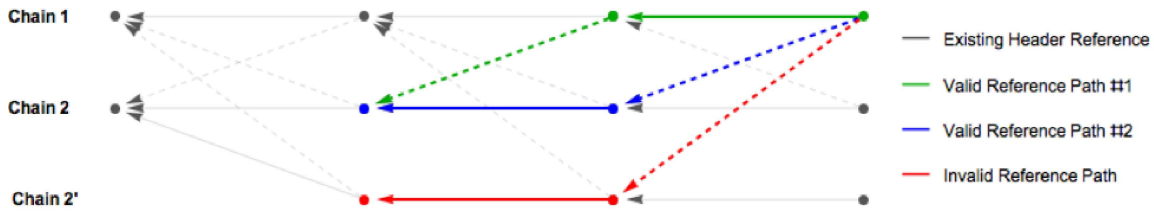

Figure 5: Same peer rule

These rules are applied to every peer reference found in any given header, with failure in any rule creating an invalid block header. Thus, to replace a given block in the network one must replace every block that directly or indirectly references that block.

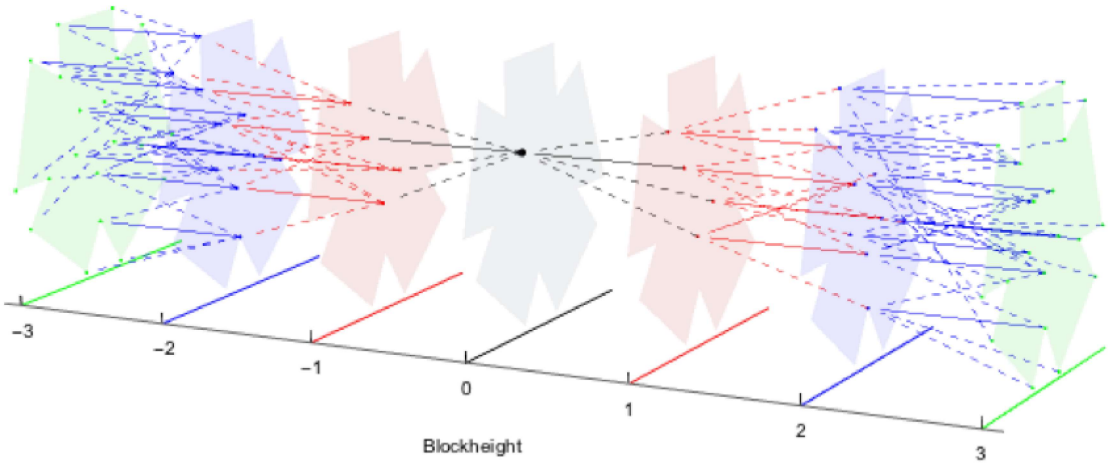

Figure 6: Past and Future Merkle cones for $Block_{Chain20,Height0}$ of the Twenty Chain Graph

4.5 Chainweb Structures

4.5.1 Past and Future Merkle Cones

For any given block, the reference propagation structure is defined by the *Merkle cone* of that block; the past Merkle cone describes what past peer transactions are trustlessly provable at a given point and the future Merkle cone describes how a given transaction becomes trustlessly provable to the entire network as subsequent layers are formed. Any transactions that fall outside of the Merkle cone of a block are unprovable via direct SPV.

Merkle cones are a novel and fundamental feature of Chainweb with wide-ranging implications, chiefly that they are the bridge between the diameter of the graph and the confirmation latency of a given chainweb configuration. This is due to the relationship of three features: the future Merkle cone of a given block that directly defines what a double-spend attacker must mine to replace that block; the diameter of the graph that defines the length of the cone; and the configuration of the base graph that defines the mass of the cone.

As every layer has a probability mass, so too do Merkle cones. Known as Merkle mass (or μ_2 for short), it is the sum of the mass of the sequential layer intersections of the cone and it increases at a nearly exponential rate[28] as the future Merkle cone of a block

[28]Interestingly, the rate at which μ_2 increases is not necessarily uniform for all chains when degree-diameter problem base graphs are used, though the variance is normally less than 1%.

is being constructed by the network. This measure dominates the security model of any Chainweb configuration since an attacker must overwhelm the full mass to successfully double spend.

A non-fully-propagated (i.e. partially-constructed) future Merkle cone also casts a shadow on the next layer. The shadow consists of the peer headers that can be constructed for that layer solely based on the headers found in the partially constructed cone. We believe the size of the shadow is important from the perspective of an attacker as it dictates how far ahead they can potentially work in private without needing to wait for the honest network to catch up and provide the requisite peer headers. In such an instance where the lack of peer headers stall a double-spend attacker, a more detailed strategy is required as it is optimal for an attacker to participate in honest mining dedicated to the peer headers they require to continue their double-spend attack.

4.5.2 Mining Resource Allocation

Layers are formed by the mining of individual chains wherein each chain, being a peer, has the same difficulty level. Difficulty is adjusted from a whole-network perspective and are enacted at a given layer height. The chance that a miner could find duplicate solutions, only one of which can be used, for the same block at nearly the same time is non-zero and thus a strategy that maximizes the expected return of mining is needed. Therefore, on average each chain receives the same fraction of total network hashrate, as this allocation is the equilibrium of individual miners selfishly attempting to minimize mining duplicates (both personal and network) and thus waste.

4.5.3 Diameter-Bounded Network Advancement

Mining a chain is a stochastic process that depends on the progress of its peers and thus, from time to time, the production of the next block in a given chain will stall. In such an instant, the global hashrate naturally pools toward that chain, increasing its speed of advancement and allowing it to catch up. This pooling occurs because the peer chains that reference the stalled block are unable to advance without said block to reference. These peer stalled blocks begin to stall subsequent blocks, all found in the next layer of the cone of said block, and so on. The absent block carves out a hole in the braid where its future Merkle cone should be. As the stall continues, fewer and fewer blocks are able to be mined. Should the problem persist for long enough, the stalled block will be the only minable block in the entire network.

Using the Petersen base graph example, if a given chain should fall two blocks (the diameter) behind its peers, no other chains can be mined and the entirety of the network

hashrate is thus forced to be applied to chain 1, which increases the average rate of mining success by 10x for the current block. Once that block is found and the stalled chain is now only 1 block behind, only chains $\in (1, 3, 4, 6)$ are minable, resulting in an initial 2.5x increase in the rate of mining success which decreases to 1x (average) as those solutions are found.

Therefore, we find that it is in the best interests of any miner to allocate mining resources at a per-chain level in a manner that keeps the rate of new block production for every chain as even as possible. This strategy is purely selfish, as the existence of a lagging chain decreases their own expected return.

4.5.4 Chainweb Attack Analysis

In this paper we present a loose closed-form probability analysis for double-spend attacks which falls in line with the tradition of the Nakamoto analysis. Further proofs with tighter bounds will be contained in our companion paper.

Even when restricting the security analysis to undirected, regular base graphs, the addition of graph theory structures such as degree, diameter, and order have profound implications for the security model of the network. The future Merkle cone bridges the domains of graph theory and PoW, the solution to which captures the number of blocks an attacker must mine to replace her fraudulent block in the network. As the layer depth of a given block increases, the peer headers required increases at a nearly exponentially rate until a layer depth equivalent to the diameter is reached, after which the required headers increase linearly.

The Chainweb structure increases the network security in a similar manner to how orphaned block headers are used in GHOST or DECOR+, though the inclusion in Chainweb occurs at a much faster rate while simultaneously increasing network throughput. Much like in a traditional blockchain where a malicious fork must be mined sequentially, a Chainweb attacker must mine the peers found in subsequent layers that fall in a block's future Merkle cone in a sequential fashion.

For the purposes of this paper, we will only cover the theme of the probability analysis and that the results for a full fork of the braid "in private" (i.e. without publishing) is almost immediately impossible. The intuition for this conclusion is straightforward: the greater the number of mining solutions needed, the greater the difficulty in maintaining a hostile fork. In our companion paper we will show that it is similarly infeasible for an attacker to mine a block's future Merkle cone. The intuition for the second is similar as the attacker, though initially at an advantage, quickly finds the cone infeasible to maintain due to the number of mined solutions increasing at a near exponential rate of subsequent layers.

The probability of an attacker working in private being able to mine a fork of the full braid is described as follows:

$$1 - \sum_{b=0}^{\mu_z} \frac{\lambda^b e^{-\lambda}}{b!} \left[1 - \left(\frac{q}{p} \right)^{\mu_z - b} \right]$$

5 Considerations and Future Work

As alluded to previously, optimal mining resource allocation strategies are necessary, but a full analysis, as well as the full analysis of the security model, will require the construction of a detailed simulator. More than likely an evolutionary approach will also be necessary to discover the optimal mining and attack strategies. This analysis is left for subsequent papers.

5.1 Bandwidth

Overall network bandwidth utilization increases linearly with the size of the network into the GB/s range for large base graphs. Two primary classes of data streams that comprise this bandwidth utilization are worthy of inspection: the mass of individual chain block streams versus the Chainweb whole-network header stream.

Block streams, for which there is an individual stream per chain, consist of the header and block for a given chain. The latter consumes the vast majority of bandwidth, but fortunately only operators replicating the entire network need to consume all of the block streams, who will already be running a large cluster to perform the replication, as the full base graphs will be infeasible to operate on a single server.

Meanwhile, the full Chainweb header stream is lightweight and fully captures the assurance of the full network. As such, it is possible for an individual to trustlessly operate a single chain in the network by accessing just its block stream along with the header stream.

5.2 Decentralization, Subset Replication and Mining

Taking the notion of individual chain replication one step further, consider a business that deploys a smart contract to a subset of chains for the purposes of provisioning load. The bandwidth and infrastructure requirements of the company grow linearly as their deployed chains increase in number. Moreover, should they wish to validate transactions involving their contracts more quickly they can devote mining resources directly to this subset.

Given that the subset only requires a fraction of the global hashrate, we believe this strategy is both feasible and optimal. Unlike the interests of miners, the financial interests of a business align more with servicing their smart contracts and less in mining rewards, thereby making a sub-optimal mining strategy rational. We hypothesize that, over time, the uptick in the block production rate will be noted by the rest of the miners and mining resources will be partially reallocated to other chains as this maximizes a miner's expected return. Individuals can also mine subsets of the network, or form smaller pools, in a similar manner.

5.3 The Role of Large Mining Pools

Some degree of centralization, whether that be in the form of staked validators or in the form of large mining pools, are an unavoidable reality for any successful cryptocurrency. Continuing with the logic found in the previous section, we find that in Chainweb centralization provides several noteworthy and novel services to the network as a whole by committing their resources elsewhere and informing others: load balancing the hashrate; catching up lagging chains; finding layer consensus by providing whole network header streams publicly; and rejecting invalid blocks (such as ones that, if included, would violate mass-conservation) injected through a single-chain mining overload.

Though the logic for the first two services has already been discussed, the third is a more novel hypothesis derived from the following: if the mining rewards are evenly distributed in a layer and the cryptographic assurance (i.e. likelihood of inclusion) for a given block builds rapidly, then withholding mined blocks from the network poses a significant risk to the inclusion of the withheld block while yielding little advantage in return due to the header of the withheld block being only one of many headers required to construct the next block in a chain.

Therefore, miners are incentivized to distribute newly found blocks as quickly as possible. As the assurance builds rapidly, and some miners may only be mining on a subset of the network, we believe that large pools also provide honest header streams to subscribers and help distribute their view of what the leading edge of the entire network looks like. This distribution, in turn, quickly produces new layer consensus.

5.4 Personal Replication and Single Step Cross-Chain Payments

While it is infeasible for most individuals to maintain a full network replica of networks with large base graphs, it is also unnecessary. The security provided by the header stream is enough to assure an individual that their single-chain replication is correct.

Consider the case of Alice and Bob who each have accounts on separate chains in 1,100-chain Chainweb network (degree 8, diameter 4). Neither has the resources to devote to a full network replica, but each operates the chain on which their account is found.

If Alice wished to purchase a good from Bob, how long would it take for the payment to be confirmed? The answer is always the number of layers equal to the diameter of the base graph, which is this case is 4 layers. Even though cross-chain cryptocurrency transfers require two distinct transactions (at least twice the diameter's number of layers) to confirm, the first (delete) step is enough to assure that the transaction will take place. The second step, where the coins are created on the other chain, can be done at any time. Because Bob is the only individual to whom the created coins can go, and no one is able to alter his right to create, only Bob requires confirmation that Alice's delete step is well formed.

5.5 Target Block Times and Layer Consensus

Subsequent studies wherein various Chainweb configurations are simulated are needed to fully explore this subject. The primary aspect to explore is how shorter block times (sub-minute) impact the ability to achieve new layer consensus in a given configuration. If the chain fork rate is too high, it will degrade overall network throughput and efficiency. Base graph degree will likely impact new-layer consensus. Simulations are needed to find the right balance between confirmation latency (base graph diameter and block time) and new layer consensus efficiency (block time and degree).

We estimate that 30-second block times are feasible, and 15- and even 5-second block times may be possible as well. Given that production Chainweb implementations will likely favor diameter-4 or -5 base graphs, 30-, 15- and 5-second block times would represent average confirmation latencies of 120-150 seconds, 60-75 seconds, and 20-25 seconds respectively, while supporting network throughput of over 10,000 transactions per second. Under load and assuming that 5 second block times are feasible, a Chainweb network could achieve a lower practical latency (e.g. the time from 100,000 transactions between Alice and Bob entering the memory pool to each being confirmed) than any network using known deterministic consensus mechanisms.

6 Conclusion

In conclusion, Chainweb provides significant advances over existing approaches in scalable public blockchain. It provides unparalleled increases in PoW throughput while keeping the global hashrate, and thus energy required, constant. The confirmation latency of Chainweb is also significantly decreased from traditional PoW and is potentially even

lower than that of PoS systems. Chainweb achieves these advances while maintaining the core trustless, decentralized nature of PoW. This protocol enables greater practical decentralization and enables the creation of an ecosystem where enterprises, individual users, and large mining pools can co-exist peacefully by acting selfishly. Chainweb avoids liquidity and centralization problems associated with using staked channels for scaling while also staying in the existing global regulatory context. We present in Chainweb as a solution by which PoW can be scaled such that it supports true decentralized economy.