Net Neutrality

Unexpected Solution to Blockchain Scaling

ALEKSANDAR KUZMANOVIC

CLOUD-DELIVERY NETWORKS COULD DRAMATICALLY IMPROVE BLOCKCHAINS' SCALABILITY, BUT CLOUDS MUST BE PROVABLY NEUTRAL FIRST.

There is a growing expectation, or at least a hope, that blockchains possess a disruptive potential in numerous domains because of their decentralized nature (i.e., no single entity controls their operations). Decentralization comes with a price, however: blockchains do not scale—they are incapable of processing a large, or even moderate, number of transactions in a timely manner. For example, bitcoin processes three transactions per second.

The root of the problem—and the limiting factor for blockchains—is a trustless peer-to-peer network model, in which information must be suboptimally propagated to—and validated at—every hop in the network. Undoubtedly, cloud-delivery networks (e.g., Akamai or YouTube), which resolved similar performance challenges in other domains

(e.g., web and video delivery), could help scale blockchains as well. The problem is that such large centralized infrastructures disturb the decentralized nature of blockchains, hence eliminating their disruptive potential. The question is, can cloud-delivery networks be used to scale blockchains without upsetting their decentralized nature? The answer is positive, and the key to the solution lies in an advanced version of an existing concept: net neutrality.

Blockchain and the cryptocurrency revolution initiated by bitcoin in 2008[8] are thriving.The market capitalization of prominent cryptocurrencies, while highly volatile, continues to be measured in hundreds of billions of dollars. A unique feature of blockchains is the lack of centralized administration. They rely on third-party mediation, (i.e., a global peer-to-peer network of participants who validate and certify all transactions). Given the purely distributed and decentralized design of blockchains, many people believe that such systems have a disruptive potential in other areas beyond cryptocurrencies, including health care, government, manufacturing, retail, insurance, The Internet of things, the sharing economy, etc. Numerous high-tech companies, big and small, are closely watching the blockchain space, analyzing how the new technology could affect their existing or future operations.

A major problem for blockchains is scalability. The blockchain system throughput is measured in the number of TPS (transactions per second) a system can support. Bitcoin's current average throughput of three TPS compares to 2,000 TPS average throughput in Visa's centralized system, 4,000 TPS daily peak, and 56,000 TPS

maximum capacity. Without scalability, cryptocurrency systems will hardly become mainstream, and blockchains are unlikely to realize their disruptive potential in any other areas.

WHAT IS A BLOCKCHAIN?

A blockchain is a public distributed ledger that stores all past transactions and is fundamentally a type of database created and shared by multiple (tens of thousands) nodes connected in a peer-to-peer network. To achieve consensus regarding the correct copy of the database, certain rules about writing to the database must be imposed. Although the rules may vary, they generally include the following:

➡ *Transactions must be valid.* A transaction, which typically passes some amount of cryptocurrency from one user to another, must contain digital signatures from the participants for authentication purposes.

➡ *Transactions must be added in sequence.* Transactions are not added to the ledger individually; rather, they are added in batches, known as *blocks*. For example, the bitcoin blockchain requires that each new block contains a solution of a hashing "puzzle" that is unique to the combination of the last block of transactions on the chain and the current block being added.

➡ *Adding blocks to the blockchain is expensive and competitive.* Parties who want to add blocks to the blockchain must invest either cryptocurrency or the computing power necessary to solve a cryptographic puzzle (e.g., the hashing puzzle required by bitcoin). Such a party is called a *miner*, and the process of adding new blocks to a blockchain is referred to as *mining*.

➥ *The longest blockchain available is the up-to-date version.* When combined with the previous rules, this makes a blockchain very expensive to forge successfully. Even copying an existing blockchain and attempting to modify the last few blocks can quickly become prohibitively expensive. Once blocks get sufficient confirmations on the network, deleting or modifying a block becomes mathematically improbable. Effectively, transactions can only be added to the blockchain; they can never be deleted.

➥ *Independent verification is required.* A node should be able to verify independently that all the previous rules have been complied with when it inspects a copy of the blockchain database. If each user can verify the blockchain independently, this allows all users to come to a consensus about the correct blockchain.

➥ *Adding blocks to the blockchain is rewarding.* Because writing blocks to the blockchain is hard, not all nodes will participate in the process. Many users will create transactions but then just request that they be written to the network, often offering a fee as an incentive. In addition, miners are rewarded with the ability to distribute new cryptocurrency to themselves whenever they win a round of the mining process and get the chance to add a block to the blockchain.

➥ *Forks can happen, but they are resolved via the longest blockchain rule.* Reaching a consensus on the blockchain is not immediate, and sometimes a *fork* (a different copy of the database) may arise in the blockchain, where different versions of the blockchain's public ledger coexist, and diverge after a common history. By selecting the longest blockchain on the network, however, nodes

work to resolve these forks.

THE BLOCKCHAIN SCALABILITY PROBLEM

Before explaining the blockchain scalability problem, let's first see how it manifests in reality. Figures 1 and 2 show the transaction backlogs for bitcoin and ethereum, two leading cryptocurrencies. You can see that tens of thousands of transactions are regularly waiting to be processed by a blockchain. To increase the likelihood of being selected by miners and get "on-chain," users increase the size of the fees they (voluntarily) include in their transactions. As a result, fees are far from negligible, and they can grow considerably during times of high

FIGURE 1: **BITCOIN TRANSACTIONS BACKLOG**

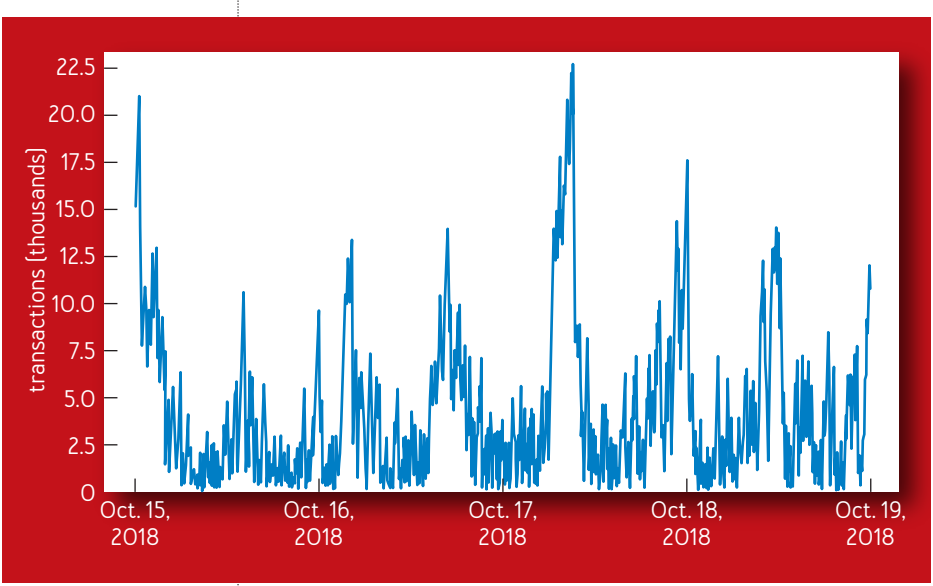
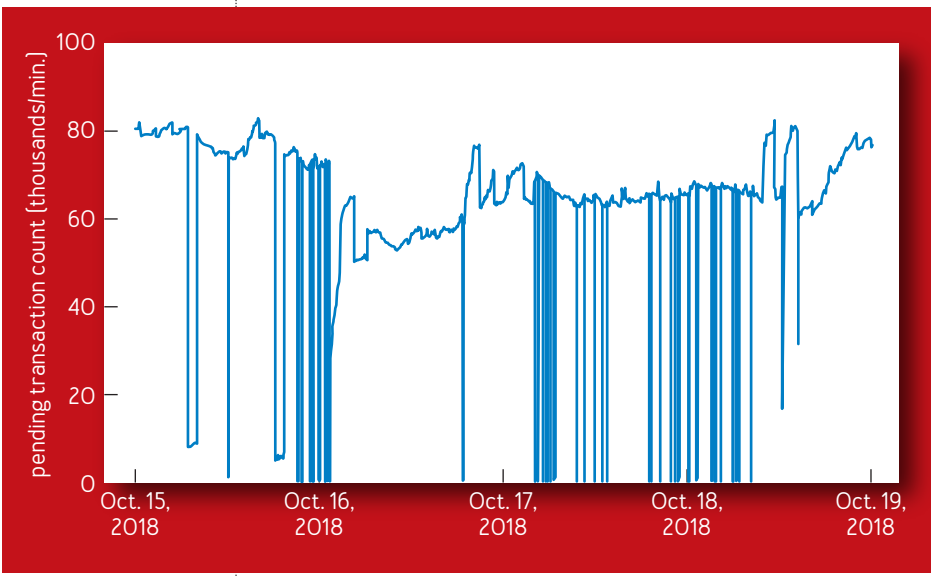


FIGURE 2: **ETHEREUM TRANSACTIONS BACKLOG**



congestion.

To understand where the bottleneck is, let's compute the blockchain throughput first. The system throughput depends directly on two parameters: the block size B (i.e., the number of bytes that can contain transactions in each block), and the interblock time interval T (i.e., the average time required for the system to mine a new block). In bitcoin, B = 1 MB and T ~ 600 seconds, which allows approximately three TPS. On-chain throughput can be improved through the following options: increase B to include more transactions; reduce T so that blocks are mined at a higher rate; or both. The problem is that these parameters cannot be arbitrarily changed, as detailed later.

Obviously, it is the blockchain's distributed nature that

causes the problems. Indeed, if blocks and transactions were to be instantly propagated among nodes, immense blocks could be mined at a rapid pace, until the limitation of designated processing units and flash storage arrays was reached.[4] In reality, however, blockchain nodes—tens of thousands of them or more—are distributed around the world. Hence, *the network is the bottleneck*.

Nodes in a blockchain network communicate in a peer-to-peer fashion. This, unfortunately, works against the goal of high-throughput, low-latency communication in the following ways:

- The information is transmitted from one node to another; hence, it takes multiple hops for the information to be propagated through the entire network. Given that each node in the network is distrustful of every other node, the information being propagated must be independently validated at every hop. This typically involves a cryptographic operation at each hop, which adds latency and hurts throughput.
- The performance *variance* of nodes in a blockchain network is high, which means that a single slow node on the critical path can inflate the propagation time.
- Finally, nodes in a peer-to-peer network are randomly formed; hence, they are not organized for optimal propagation. This means that data travels through suboptimal paths in the network.

As a result, the average time needed to propagate a 1 MB block to 90 percent of the nodes in the bitcoin network is 11.6 seconds, which was the average propagation time observed in March, 2017.[1] This is, unfortunately, just a part of the problem. It has been shown, both in theory[7] and in

practice,[4,5] that increasing the block size B by a factor of X also increases the time required for a block to propagate by the same factor X. Similarly, decreasing the interblock interval T by a factor of X has exactly the same effect. This means that the block-propagation time increases proportionally with each of these two parameters.

For example, increasing the block size tenfold would increase the block-propagation times by tenfold as well, making them longer than 100 seconds. Likewise, increasing the block size by a factor of 100 would lead to block-propagation times longer than 1,000 seconds. Such a propagation time exceeds the time between blocks, *causing a fork every time a new block is mined*. Indeed, in this scenario, forks will not be resolved by the mining of the following block, and instead the blockchain will unravel to forks, and forks-of-forks, and forks-of-forks-of-forks, until nodes and miners do not know which fork is the "true" chain—and the blockchain breaks. This is the blockchain scalability problem caused by the networking bottleneck.

CLOUD-DELIVERY NETWORKS

Cloud-delivery networks were very successful in resolving performance problems on the Internet. Such networks distribute content via an immense infrastructure that can consist of hundreds of thousands of servers worldwide (e.g., Akamai). In addition, they perform extensive network and server measurements and use them to redirect clients to nearby servers. This helps the Internet operate at the immense scale it does.[6] As an example, YouTube alone has more than a billion users, and a whopping 70 percent of North American Internet traffic in peak evening hours

comes from streaming video and audio sites such as Netflix and YouTube. This would not be possible without cloud-delivery networks.

This is in striking contrast to the state of affairs with blockchains. Indeed, as explained earlier, propagating a 1 MB block through a blockchain network is a time-consuming task, and increasing the block size much more could lead to unrecoverable problems. Yet cloud-delivery networks manage to send terabytes of data every second, and that is considered ordinary. Can such networks be used to scale blockchains?

Undoubtedly, cloud networks could improve the performance of blockchains. The issue is trust. In a blockchain ecosystem, a node does not trust its immediate peers, so how will it trust a cloud network, which is far more powerful than any individual node? Cloud-delivery networks are centralized systems that can censor transactions, blocks, or miners of a blockchain network. For example, the cloud-delivery network administrators may reject blocks that contain transactions among unauthorized parties, or blocks mined by unauthorized miners, according to their own policies, business interests, or legal requirements.

Thus, the key question is whether it is possible to make cloud-delivery networks trustless, such that they can be used to scale blockchain networks, without the ability to exercise censorship and other powers previously mentioned here. This concept is called *provable net neutrality*. Without diving into formalism, this article outlines the key properties associated with this concept.

First, the network should not be able to censor

information based on the content of blocks. Second, the network should not be able to censor nodes. Third, nodes should be able to audit these properties continuously, and in case of network misbehavior, to abandon and replace the network. How do you enable such properties?

A PROVABLY NEUTRAL BLOCKCHAIN-DISTRIBUTION NETWORK

Consider a cloud-distribution network that aims to enable blockchain systems (not necessarily cryptocurrencies only) to scale to thousands of on-chain transactions per second. Moreover, it aims to provide scalability to numerous cryptocurrencies and blockchains simultaneously, using a global infrastructure to support distributed blockchain systems in a provably neutral fashion. This is known as a BDN (blockchain distribution network). This section outlines the system's trust model and then describes the key mechanisms necessary to fulfill the neutrality properties.

Reversed trust model

BDN's trust model is based on two observations: first, long block-propagation times will not ever allow trustless peer-to-peer blockchains (e.g., bitcoin), to scale substantially; second, small centralized systems scale very well by placing trust in a small subset of participants and passing them the control over the transactions included in the blockchains (e.g., Ripple and EOS).

Such centralization, however, defeats the single most notable aspect of blockchains: the distribution and decentralization of control over transactions. Providing control over a blockchain's transactions to a limited

number of participants allows participants to collude, censor, and discriminate among users, nodes, and miners. A limited participant set also reduces the number of nodes a malicious actor has to compromise to control the system.

BDN addresses this tradeoff by reversing the direction of trust in centralized systems. While centralized systems place trust in a subset of nodes to enable scalability, BDN enables scalability by using a small set of servers that place trust in the entire network instead. The resulting system can enable scaling, yet nodes need not place any trust in the BDN. Instead, the BDN blindly serves the nodes, without knowledge of the blocks it propagates, their origin, or their destination. Moreover, its behavior is constantly audited by the nodes it serves, and it is incapable of discriminating against individual nodes, blocks, and transactions. While such a design places the BDN at a disadvantage compared with the nodes it serves, its robustness allows it to withstand dishonest and malicious behavior.[7]

Provable network neutrality

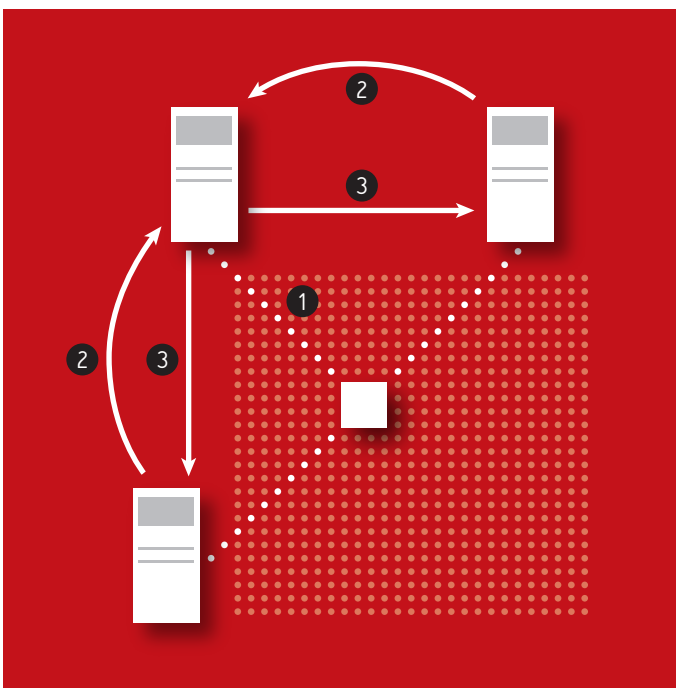
In short, BDN can only propagate all blocks to all the blockchain nodes fairly, and it is incapable of discrimination because of the auditing performed by the blockchain nodes, still connected in a peer-to-peer fashion.

Encrypted blocks

To prevent BDN from stopping the propagation of any block based on its content, blocks are propagated after being encrypted (step 1 in figure 3). BDN's encryption also alters the block size, hiding the number of transactions and their total size. After the block has been propagated, the

receiving peer nodes inform the sender by sending a hash of the block (step 2 in figure 3). Finally, a block's encryption key is revealed and is propagated directly over the blockchain peer-to-peer network (step 3 in figure 3). The encryption key's tiny size, only several bytes, allows it to quickly propagate directly over the peer-to-peer network, and BDN is powerless to stop it.

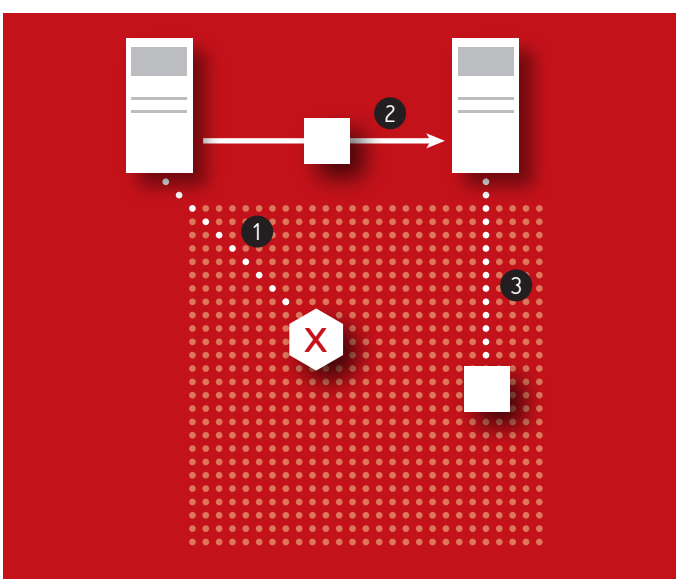
FIGURE 3: **ENCRYPTED BLOCK**



Indirect relay

To ensure BDN is not preventing individual nodes from propagating their blocks, nodes do not have to propagate blocks directly to BDN. In case a block is not propagated by BDN (step 1 in figure 4), the sending node will propagate it to a peer on the peer-to-peer network (step 2 in figure 4), which will relay it to BDN (step 3 in figure 4), obscuring the block's origin from BDN. For example, a node that mined a block in China could relay it to a node in Europe, which then sends the block via BDN. In addition to relaying blocks indirectly to BDN, nodes may request their peers to relay to them incoming blocks arriving from BDN. This ensures that BDN cannot discriminate against nodes through late

FIGURE 4: **INDIRECT RELAY**

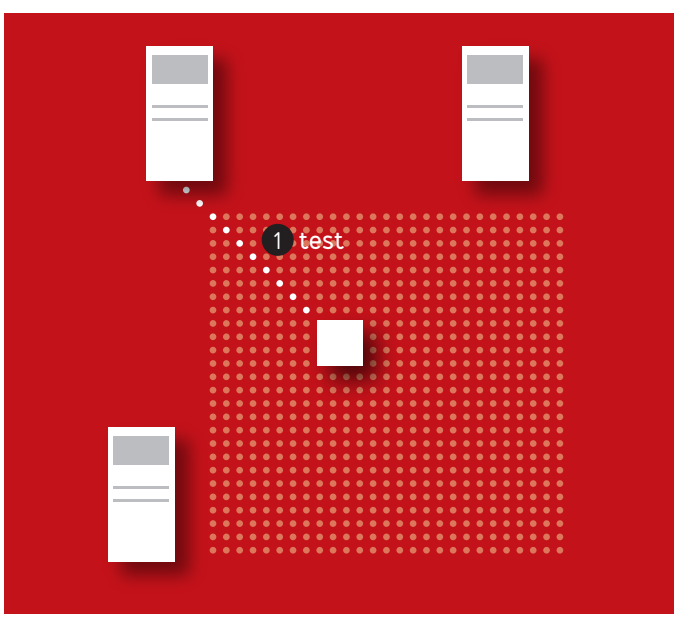


delivery of blocks since nodes are not required to interact directly with BDN in order to benefit from its service.

Auditing via test blocks

While BDN is oblivious to which node originates each block, it may attempt to prevent or stall blocks arriving from some subset of nodes, affecting all the blocks they relay. In order to detect and prevent such behavior, nodes must be capable of continuously monitoring BDN's service. Such monitoring is achieved by allowing nodes to send encrypted invalid blocks, *test blocks*, directly to BDN (figure 5) and measuring the time required for peers to report the arrival of the test blocks. BDN is unable to

FIGURE 5: **TEST BLOCK**



employ discriminatory policies over valid blocks alone and faithfully propagate test blocks, since the two are indistinguishable until their keys are published.

Thus, by using traffic encryption and indirect traffic relaying, and by explicitly auditing BDN, blockchain nodes are capable of restricting the BDN's ability to misbehave, effectively decoupling a BDN operator's authority from the BDN infrastructure. If a BDN ceases to deliver blocks completely, or delivers blocks only to a small subset of nodes, the blockchain nodes can abandon the BDN.

Since nodes are constantly using test blocks to infer the best source from which to receive blocks, any node that BDN discriminates against will simply be receiving blocks from its peers. Thus, if BDN is maliciously discriminating against many or all peers, peers will simply form their own peer-to-peer network until a different system takes its place. Additionally, if the discrimination is caused by a large-scale system failure, the peers will return to using BDN once the failure is resolved.

PERFORMANCE

In essence, BDN deploys *a broadcast primitive*, meaning it enables efficient transmission of data from a single source node to all other nodes in a blockchain network. In contrast to a peer-to-peer network, where each blockchain node is connected to numerous other nodes, often spread around the world, a blockchain node replaces this one-to-many communication with one-to-one communication. This is because a blockchain node connects to a single BDN server.

With large TPS rates, using a single connection vs. many

connections helps with scaling. Necessarily, blockchain nodes still need to be connected in a peer-to-peer network to audit the BDN effectively. The bulk of the data, however, is transmitted to and from the BDN. Following are several ways in which BDN helps scale blockchains.

Transactions caching

In a blockchain system, such as bitcoin or ethereum, transactions are received by each node twice: once as raw transactions when initially propagated through the network, and the second time when they are included in blocks. A BDN can effectively distribute transactions through the cloud, index them, and then utilize indexes (instead of raw transactions) when transmitting blocks. This effectively compresses the block size by more than 100 times, given that the raw transaction is approximately 500 bytes long, while an index can be four bytes or less.

Transactions caching is an existing idea in the blockchain ecosystem, and it has been adopted by certain projects,[3] but it has been deployed only by endpoints, not by the network. As a result, given that not all transactions in a pure blockchain system reach all endpoints,[8] even a slight desynchronization can lead to significant increases in the block size (not all transactions are "compressed"); hence, the performance suffers. In contrast, BDN effectively transmits and indexes blockchain transactions.

Cut-through routing

In contrast to blockchain nodes, BDN *cannot* check the validity of blocks flowing through the network, because they are encrypted. This helps with swift transmission of

blocks through the network. In particular, before all bits of a block are received by a BDN node, the BDN can already start transmitting received bits of a block to the rest of the network. This is called *cut-through routing*, and it has been widely adopted in network switches for decades. Still, it can significantly speed up the data transmission, particularly when blocks are large.

Transactions incast problem

Transactions need to be broadcast in a blockchain network. In the absence of a BDN, at higher TPS rates this creates a so-called *incast* problem: the *same* transactions are received at a high rate from multiple sources. This can significantly affect a node's resources and impact overall blockchain performance. BDN eliminates this problem given that the bulk of data, including transactions, is propagated to and from a single BDN server.

RELATED BLOCKCHAIN SCALING ATTEMPTS

Alternative approaches to scaling blockchains are described below.

Off-chain scaling solutions

One alternative approach (e.g., the Lightning Network), which uses *off-chain transactions*, aims to reduce some of the redundancy on the main blockchain. Generally speaking, an off-chain scaling solution will open up a payment channel between two parties (i.e., have the parties exchange funds while keeping track of intermediate balances) and then post a settlement transaction on the blockchain.

Such a solution is agnostic to BDN's proposition. Indeed,

an off-chain scaling solution still fundamentally requires on-chain capability. Also, the potential scaling benefits are *multiplicative*. If the underlying blockchain can support 1,000 times the number of transactions as before thanks to BDN, and if off-chain transactions increase the throughput by another factor of 1,000, then that blockchain's throughput has increased by six orders of magnitude.

On-chain scaling solutions

On-chain scaling solutions typically involve modifying the consensus protocol in some way to achieve higher throughput. One such approach, known as *sharding*, splits the blockchain into several smaller shards, which are maintained and interleaved such that the blockchain's original security properties are preserved while requiring only a full node to track one shard instead of the full blockchain. Numerous other ideas exist in this space.[2] While these approaches show potential, their robustness, security, and usability in practice remain to be seen.

Still, all on-chain scaling solutions will perform strictly better with a faster network layer, and this is where BDN improves their performance. Indeed, in every distributed consensus protocol, every protocol-compliant node must reach the same decision. Thus, every such peer must obtain information about each transaction in the system, independently from the consensus protocol. BDN focuses on this particular problem, which is fundamentally a broadcast problem, since every valid piece of information must be propagated to every peer in the system. BDN is thus agnostic to a native consensus protocol, and it is capable of boosting the performance, often dramatically, of *any* blockchain.

CONCLUSION

Provably neutral clouds are undoubtedly a viable solution to blockchain scaling. By optimizing the transport layer, not only can the throughput be fundamentally scaled up, but the latency could be dramatically reduced. Indeed, the latency distribution in today's data centers is already biased toward *microsecond* timescales for most of the flows, with millisecond timescales residing only at the tail of the distribution. There is no reason why a BDN point of presence would not be able to achieve a similar performance.

Adding dedicated optical infrastructure among such BDN points of presence would further alleviate throughput and reduce latency, creating the backbone of an advanced BDN. The key to this vision, however, lies in establishing trust by the blockchain ecosystem into the underlying networking infrastructure. This, in turn, is achieved by decoupling authority from infrastructure via a provably neutral network design.

Related articles

- Research for Practice
  Cryptocurrencies, Blockchains, and Smart Contracts
  Arvind Narayanan and Andrew Miller
  https://queue.acm.org/detail.cfm?id=3043967
- Better, Faster, more Secure
  Who's in charge of the Internet's future?
  Brian Carpenter
  https://queue.acm.org/detail.cfm?id=1189290
- A Purpose-built Global Network: Google's Move to SDN
  A discussion with Amin Vahdat, David Clark, and Jennifer Rexford
  https://queue.acm.org/detail.cfm?id=2856460

References

1. Bitcoinstats.com. Data propagation; www.bitcoinstats.com/network/propagation/.
2. Cachin, C., Vukolic, M. 2017. Blockchain consensus

protocols in the wild. arXiv; https://arxiv.org/pdf/1707.01873.pdf.
3. Clifford, A., Rizun, P. Suisani, A., Stone, A., Tschipper, P. 2016. Towards massive on-chain scaling: presenting our block propagation results with Xthin; https://medium.com/@peter_r/towards-massive-on-chain-scaling-presenting-our-block-propagation-results-with-xthin-da54e55dc0e4.
4. Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., Song, D., Wattenhofer, R. 2016. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*. Springer. 106–125.
5. Decker, C., Wattenhofer, R. 2013. Information propagation in the Bitcoin network. In *Proceeding of the 13th IEEE International Conference on Peer-to-Peer Computing*; https://ieeexplore.ieee.org/document/6688704.
6. Internet Live Stats; http://www.internetlivestats.com/one-second/.
7. Klarman, U., Basu, S., Kuzmanovic, A., Sirer, E. G. 2018. bloXroute: a scalable trustless blockchain distribution network; https://bloxroute.com/wp-content/uploads/2018/03/bloXroute-whitepaper.pdf.
8. Nakamoto. S. 2008. Bitcoin: a peer-to-peer electronic cash system. Bitcoin.org; https://bitcoin.org/bitcoin.pdf.

Aleksandar Kuzmanovic *is a professor of computer science at Northwestern University. His recent research includes content-delivery networks, net neutrality, and blockchains. He is a cofounder of bloXroute Labs, a blockchain scaling startup, where he serves as a chief architect.*

Copyright © 2019 held by owner/author. Publication rights licensed to ACM.