Blockchain Layer Zero: Characterizing the Bitcoin Network through Measurements, Models, and Simulations

Elias Rohrer
Distributed Security Infrastructures
Technical University of Berlin
elias.rohrer@tu-berlin.de

Florian Tschorsch
Distributed Security Infrastructures
Technical University of Berlin
florian.tschorsch@tu-berlin.de

***Abstract*—In recent years, research has shown the networking layer's significant influence on the scalability, security, and privacy of blockchain systems. Such large-scale networks however exhibit a degree of complexity that demands model-based simulations as real-world experiments are often not possible. In this work, we methodically characterize blockchain networks by reference to the paradigmatic Bitcoin peer-to-peer network, explore the state-of-the-art protocols, and emphasize this key design space. To this end, we conducted a longitudinal measurement study on the Bitcoin network, from which we extract a comprehensive network model and implement it as part of the `bns` network simulation framework. We validate the model in comparison to real-world measurements as well as to results from related work. Moreover, we experimentally show how network utilization and miners' geographical location impact the block propagation characteristics.**

I. Introduction

Even though they have become increasingly relevant infrastructures, open blockchain networks, such as Bitcoin [1] and Ethereum [2], are still struggling to get ready for global mass adoption. Research on blockchain systems in recent years often aimed at addressing the bottlenecks of the consensus layer by scaling-out and adding additional layers of abstraction, such as transaction off-chaining. However, prior empirical work has shown the properties of the peer-to-peer network layer to have a significant influence on the security and performance of blockchain systems [3, 4]. It even has been identified as *the* impeding factor for transaction scalability [5]. Moreover, analytical works focusing on the consensus layer proved that the consistency properties of Nakamoto-style blockchain protocols hold in the partially synchronous network model, that is, when all blocks reach all participants in bounded time [6, 7, 8]. Consequently, consistency, security, and chain-quality properties suffer when network-induced propagation delay increases.

Given the complexity of large-scale peer-to-peer networks, network-layer behavior is often studied in controlled environments such as network simulations. While relying on empirical datasets and simulation models is generally inevitable, prior entries studying blockchain systems based on network simulation often revert to assumptions that are of simplistic nature.

In this work, we methodically characterize blockchain networks in reference to the paradigmatic Bitcoin peer-to-peer network, explore the state-of-the-art protocols, and highlight this vital design space. Our main contributions comprise a comprehensive measurement study, designing a blockchain network simulation model, and a simulation study validating the model as well as providing additional insights.

In a first step, we study the Bitcoin network empirically through an extended measurement study. To this end, we deployed measurement nodes in seven geographically distributed locations all over the world and conducted long-term measurements on the provisioned bandwidths and latencies of Bitcoin peers. We moreover recorded the regional distributions of peers and miners in the network, as well as block sizes and the corresponding validation delays incurred. This comprehensive measurement study confirms and extends prior work on blockchain networks.

Second, we extract parameters from this data set that induce a geographically clustered network model, which builds the foundation for realistic network simulations. We implement this model in `bns`, a modular simulation framework for blockchain networks that allows for configurable networking stacks. It is based on ns-3 and offers a layered design closely resembling the internet architecture. To the best of our knowledge, `bns` is the first networking-centric simulation framework allowing to capture complex network effects of blockchains.

Lastly, we validate the introduced simulation model by comparing the results of simulated network experiments to real-world measurements. We show that the simulations using `bns` framework in comparison to other blockchain simulators [4, 9] resemble the real-world ground truth more accurately. Moreover, we show the capabilities of `bns` by evaluating the influence of different block propagation mechanisms and block sizes on the network utilization. We confirm that high network utilization or even congestion significantly increases the rate of stale blocks, which has been shown to negatively impact the security of the consensus layer [4]. Furthermore, we study the impact of miners' geographic locations on the block propagation process and show some geographic regions to be disadvantaged in the block race.

The remainder of this paper is structured as follows. Section II gives an overview of the Bitcoin network protocol. In Section III, we present the results of our measurement study on the Bitcoin peer-to-peer network. Section IV introduces the `bns` network simulator enabling the simulation of blockchain networks in accordance to a geographic topology model. In Section V, we validate the underlying network model and study the impact of the miners' geographic location on the block propagation. Section VI discusses related work before the paper concludes with Section VII.

II. State of the Bitcoin Network Protocol

The backbone of the Bitcoin network is comprised of nodes that form an open and unstructured peer-to-peer overlay network: everyone who wants to participate in the network can setup a so-called *full node*, i.e., run a software which implements the Bitcoin protocol and replicates the entire blockchain. We base our account of the protocol on the behavior of the reference client, Bitcoin Core [10].

The Bitcoin peer protocol is an unencrypted TCP-based network protocol in which nodes pick their neighbors in a randomized fashion: every node establishes 8 outgoing connections and, if reachable and configured, accepts up to 117 incoming connections, resulting in a maximum connection count of 125. By convention, nodes that accept incoming connections from other peers are called *servers*, and *clients*, if they only establish outgoing connections. The Bitcoin software keeps a local database of known peer addresses from which it randomly draws candidates for outgoing connections. If this database is empty, e.g., when the software is started for the first time, it is bootstrapped by querying a number of community-run DNS servers whose addresses are hard-coded in the client software. They return peer IP addresses as the contents of `SRV` resource records. Peer addresses are also gossiped to network neighbors and can likewise be requested by and from each network participant.

After a TCP connection is established, the nodes exchange `version`/`verack` messages which transport crucial peer data and additionally serve as basic handshake messages. Moreover, the exchange of peer address data, transaction forwarding, and block propagation is initiated. Every network participant may insert new transactions to the network, which are then propagated via Bitcoin's gossip protocol and are validated by every full node along the way. In particular, new transactions are first announced to neighboring nodes through `inv` (read: *inventory*) messages, which in this case are only sent after an exponentially distributed delay for privacy reasons. This staggered propagation scheme is known as *diffusion spreading* [11].

Transactions are validated, collected, and bundled into blocks by miner nodes which start "mining" the block, i.e., start calculating the solution to a cryptographic puzzle whose difficulty parameter is set based on network consensus. The solution, the proof-of-work, is included into the block header, which is prepended before the new block is disseminated in the peer-to-peer network. In earlier versions of the Bitcoin protocol, blocks were announced via immediately forwarded `inv` messages, and receiving nodes would request blocks and headers independently with `getheaders` and `getdata` requests. These would then again answered by corresponding `headers` and `block` messages.

However, since protocol version 70012 was introduced with Bitcoin Core v0.12.0, the default block propagation scheme changed [12]: blocks are announced directly by sending `headers` messages, which reduce the propagation delay. Yet, when more than one block has to be announced, the client falls back to the `inv`-based announcement scheme. Nodes enable this new protocol by sending a `sendheaders` message after the initial handshake. In addition, a scheme for *compact block relay* was introduced [13], which allows to send block announcements in a more bandwidth efficient way. In particular, it allows nodes to only retrieve the transaction data they are missing from an announced block, which can severely reduce the bandwidth overhead of block propagation, but is prone to induce an additional latency overhead.

While other blockchain networks may exhibit their own idiosyncrasies, they often coincide with Bitcoin's general networking paradigm, i.e., block and transaction propagation through broadcast in an unstructured overlay topology. For example, even though Ethereum's [2] peer discovery is based on a Kademlia [14] overlay construction, the actual `eth` propagation protocol follows a schematism similar to that of Bitcoin [15, 16]. However, at the time of writing, it chooses to forgo the request-response scheme over unsolicited block propagation and also does not implement discussed protocol extensions, such as a compact block relay mechanism [17]. As there are still many similarities, the Bitcoin approach to networking might rightfully be classified as paradigmatic for blockchain networks overall.

III. Bitcoin Measurement Study

In order to characterize the network conditions currently found in the Bitcoin network, we conducted a network measurement study that allows us to categorize the network peers along the lines of seven regional clusters: North America (NA), South America (SA), Europe (EU), Oceania (OC), Asia (AS), Africa (AF), and China (CN)[1]. In the following, we present and discuss our methodology and the measurement results.

A. Data Rates

In order to characterize how fast Bitcoin network peers are able to propagate blocks, we conducted a longitudinal measurement study recording the bandwidth distribution in the Bitcoin network. The measurements took around a month, starting from April 1, 2020. In particular, since it is typically the limiting factor of internet access, we are interested in the upload bandwidth of Bitcoin nodes. For this, we developed a measurement utility that was deployed on seven nodes as close as possible to the geographical center of the seven regional clusters. In particular, we deployed nodes at the

[1]The separate cluster for China is justified on the basis of earlier research that highlights its special role for blockchain networks [18, 19, 20].

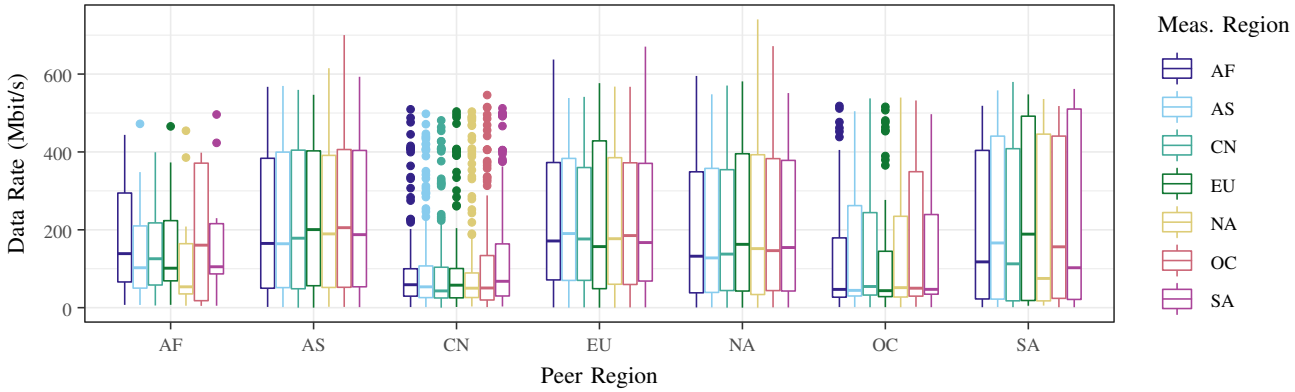

Fig. 1. Measured Data Rates per Region

following Amazon AWS regions: `us-west-1` (NA), `sa-east-1` (SA), `eu-central-1` (EU), `ap-southeast-2` (OC), `ap-south-1` (AS), `me-south-1` (AF), and `ap-east-1` (CN).

Once started, the measurement tool connects to one Bitcoin peer at a time and requests as many blocks as possible in a given time frame. In order to ensure bandwidth saturation, we configured the client to establish three concurrent connections to each of the node addresses publicly available from the Bitnodes [21] database. After an initial offset to account for connection establishment, the download traffic was recorded and analyzed for five minutes utilizing `libpcap` [22], before moving on to the next address. During the study's runtime, each of the seven measurement nodes processed the node list in a randomized order to minimize the risk of our study interfering with regular network operation. As we're interested in the upload line speed of each peer, we processed the incoming data in intervals of 30 ms and recorded peak data rates. Moreover, the addresses were clustered based on the GeoLite2 [23] geolocation database.

Every measurement node initiated connections to an average of 4,155 peers that could be successfully reached (from a total of 7,111 known peers). The resulting data rates are shown in Figure 1 for each peer region and in dependence of the measurement region. We observe that generally the data rates follow a wide spread of up to 740 Mbit/s as well as down to close to 0 Mbit/s. Moreover, the measured network bandwidth in NA, SA, EU, and AS regions are highest and similarly distributed, as they exhibit average peak rates of around 200 Mbit/s, 222 Mbit/s, 218 Mbit/s, and 224 Mbit/s, respectively. This suggests that peers in these regions supply the core infrastructure of the Bitcoin peer-to-peer network. The peers in the AF and OC regions are fewer and not as well connected, featuring average peak rates of 161 Mbit/s and 144 Mbit/s. Interestingly, the measured rates of peers located in the CN region are lowest with a mean peak rate of only around 106 Mbit/s. This observation is particularly notable, as the regional distribution of blockchain networks have been discussed in literature for quite a while [18, 20] and these measurement results are in line with prior research that suggests that the so-called "Great Firewall" of China may pose a significant bandwidth bottleneck, which the authors link to detrimental miner behavior, such as creating empty blocks [19].

B. Latencies

Besides bandwidth, inter-peer latencies have a major impact on the characteristics of message propagation in peer-to-peer networks. Therefore, we conducted an extensive measurement study with the goal of capturing latency distributions between the different geographical regions of the Bitcoin network. In April 2020, we deployed seven additional measurement nodes in the NA, SA, EU, AF, AS, CN, and OC regions. After deployment, each measurement node ran a script sending 100 ICMP `ping` requests to each of the publicly available IP addresses of Bitcoin nodes and recorded the average round-trip time (RTT) value. Afterwards, we collected the results and grouped them according to their source-destination regions i.e., from where the measurement was conducted and where the measured nodes were located.

From the 7,121 queried IP addresses, around 2,500 did not respond to the ping requests, be it because they were not online anymore, or were blocking ICMP requests. This leaves us with measurement results for 4,602 Bitcoin peers. The measured average RTT values are shown in Figure 2: we observed a mean RTT of 180 ms, which however exhibits a large standard deviation of 92 ms, as values range from less than 1 ms to the maximum outliers well surpassing 1,000 ms. We attribute measurements close to 0 ms to nodes in physical proximity (maybe even in the same data center). Moreover, peers located in the EU and NA regions are reachable the quickest, both featuring an overall mean RTT of 176 ms, while AF peers are the slowest to respond as they do so within 304 ms on average. In contrast to the bandwidth measurements, the latency of peers located in the CN region are not exhibiting significantly sub-par performance. However, our observation that ICMP packets seem not to experience similar effects actually further supports the hypothesis that the low data rates shown before are caused by the performance deficits induced by the Great Firewall's deep packet inspection.

C. Peer and Mining Distribution

In order to investigate the current regional distributions of network peers and mining power, a Bitcoin node was

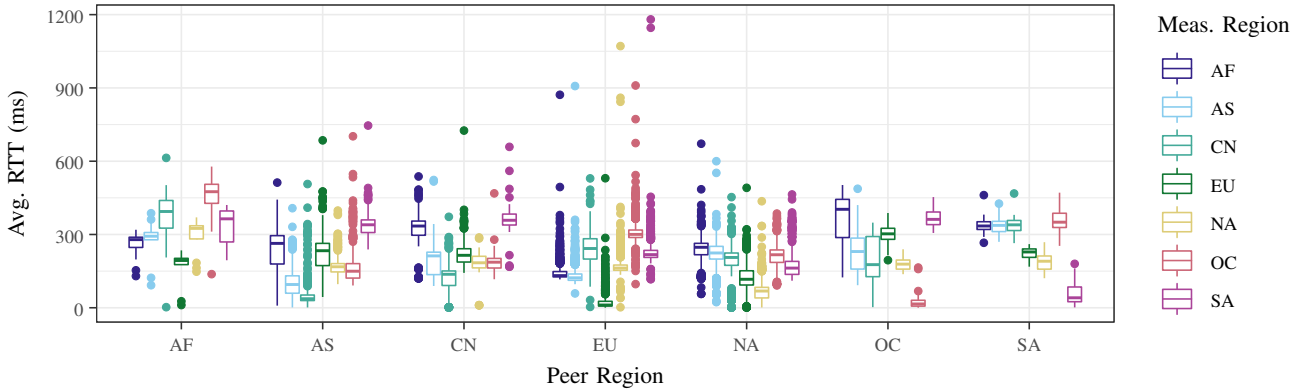

Fig. 2. Measured Inter-regional Latencies

TABLE I
REGIONAL PEER AND MINING DISTRIBUTION

Region	EU	NA	AS	OC	SA	CN	AF
Peer Share (%)	49.1	41.4	4.7	1.8	1.3	1.0	0.6
Mining Share (%)	85.2	10.1	0.8	0.0	0.0	3.9	0.0

deployed in the network of our university. It was configured to run a modified version of the Bitcoin Core software that allowed for an unbound number of incoming and outgoing node connections. We let this node run for several days to acquire a high number of connected neighbor nodes, which eventually fluctuated around 2,500 connections, thus covering a large share of the Bitcoin network. From September 1, 2020 to September 30, 2020, we recorded incoming block announcements from all neighbor nodes. Given the good connectivity of our measurement node, we assume the node receives new block announcements from the miner directly or from a source close to the miner. Hence, we attribute the first observed announcement of a new block to the geographic region associated with the IP address we received it from, which corresponds to a *first-spy estimator* [24].

As can be seen in Table I, the distribution of network peers over the regional clusters is heavily skewed, as the highest share of peers is located in the EU region with 49.1 %, while the lowest share is located in the AF region with only 0.6 % of peers. The result also reveal a mining power distribution which is even more skewed than the peer distribution. Notably, the EU region provides 85.2 % of observed blocks, while only 49.1 % are located in this region. Moreover, the CN region exhibits an overproportional share of mining power of 3.9%, while providing only 1.0% of network nodes. The current picture is however a rather big change from a pilot study following the same methodology we conducted in the end of 2019, in which we observed an even more drastic inequality in mining power distribution: in November 2019, peers from the CN region still provided 77.6% of mined blocks, while only accounting for 3.9% of the network. While we observe this significant change, we currently have no clear indication on what exactly led to the shift of mining power distribution.

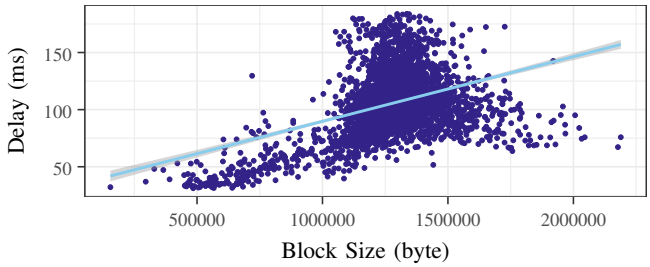

Fig. 3. Observed Block Sizes and Validation Delays

D. Block Sizes and Validation Delay

In order to get an understanding on how much data the network needs to process during block propagation, starting September 1, 2020, we utilized our measurement node to record newly published blocks over a period of one month. During this time, 4,087 blocks were published, for which the sizes can be seen in Figure 3. While the range of block sizes spanned everything from 200 bytes to 2.09 MB, the mean observed block size was 1.17 MB, which shows that Bitcoin is currently not constantly hitting its capacity limit.

Moreover, as each node only forwards new blocks after it validated its transactions and the proof-of-work, we recorded the time our measurement node took for validation. As seen in Figure 3, we observed a linear correlation between block size and validation delay. We furthermore validated this using Pearson's product-moment correlation test, which yielded a correlation coefficient $r = 0.41$ and $p < 2.2e-16 \ll 0.05$, which allows us to reject the null hypothesis, i.e., suggests that there is indeed a significant correlation between block size and validation delay.

E. Data Set and Measurement Ethics

We make the source code of our measurement tools, the measurement data, and the inferred regionally clustered model accessible to the public.[2] Note that some of the recorded data, such IP addresses, are most likely also available from other public sources such as Bitnodes [21]. However, for the sake of measurement ethics and in accordance with the Menlo report [25], we try to minimize our interference with the live

[2]See the companion repository: https://git.tu-berlin.de/rohrer/blz-data

network and hence treat such potentially identifying information as sensitive data. We therefore refrain from publishing the raw data set and instead publish data only in sanitized form.

IV. Bitcoin Network Model

In the following, we methodically model blockchain networks based on our measurements of the Bitcoin network.

A. The bns Simulation Framework

The bns simulation framework[3] is based on the ns-3 network simulator [26], whose discrete event-based architecture enables realistic and time-independent simulations of large computer networks. bns follows a modular approach that allows for expandability and customizability of the networking stack. The implementation is oriented towards the paradigmatic Bitcoin node logic and, as a baseline, currently implements a TCP-based networking stack that resembles Bitcoin's unstructured peer-to-peer overlay for block propagation, i.e., each node by default establishes 8 outgoing connections to randomly chosen peers. The capabilities of the bns simulation framework were first proven when it was utilized to evaluate the Kadcast broadcast protocol [27].

1) Architecture: The main component of bns is implemented as a C++ program that creates different network scenarios using the ns-3 simulator. To this end, it spawns a configurable number of ns3::Node objects and configures network links between them. On each node, a blockchain-specific ns3::Application is installed, which is then run during the simulation process. As the different layers closely resemble the real internet architecture, and all nodes and applications behave as independent actors, this simulation method captures detailed network effects and dynamics. In particular, and in contrast to previous works, TCP streams flow over shared links, thereby inducing queueing delays, and possibly even network congestion leading to dropped packets and retransmissions.

2) Configurability and Parametrization: The simulator can be parametrized to reflect different blockchain systems and in order to experiment with different parameter sets. We chose the default parameter set of bns in reference to the Bitcoin network which therefore is able to mimic its wire protocol, i.e., it implements the inventory and header-based announcement schemes, as well as a stochastic model for block propagation based on compact blocks. In order to provide a unified and controlled simulation environment, we base the parametrization of the simulator on the network model derived from our measurement study presented in Section III. Specifically, simulated block sizes are sampled from our measurements and are optionally multiplied with a *block size factor* in order to simulate different block sizes. Likewise, blocks are only forwarded after a validation delay Δ_v calculated based on their size B_s corresponding to the linear equation $\Delta_v = \alpha + \beta B_s$, where α and β are taken from the measured validation delays, as discussed in Section III-D. Similarly, each miner schedules block generation events based on her hash power, overall resulting in a Poisson process spawning a block every ten minutes on average. This may be modified by applying a *block interval factor* to simulate other block intervals, e.g., a factor of 0.025 would result in Ethereum's 15 second target interval. As default, we simulate 10 mining pools that together provide nearly the entire hash power of the Bitcoin network [28]. To this end, bns supports the simulation of block propagation according to different network topologies, which we describe in the following.

[3]See the simulator repository: https://git.tu-berlin.de/rohrer/bns-public

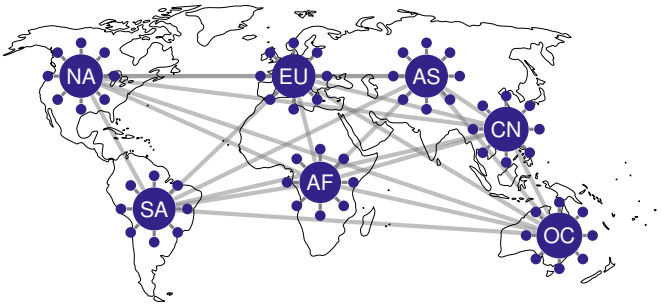

Fig. 4. Geographic topology model consisting of seven regional hubs and associated peers.

B. Geographic Topology Model

In order to capture the protocol behavior in complex and realistic settings, we employ a regionally clustered network model in ns-3. This network architecture is derived from the measured data set and creates an underlay topology that relies on seven regional node clusters. As can be seen in Figure 4, each of the clusters follows a hub-and-spoke model, i.e., nodes are arranged around a regional router, and all regions are fully interconnected. The regional distribution of peers and miners is conducted according to our previously discussed findings. Moreover, data rates of nodes are drawn from a piecewise-linear distribution created based on our measurements, while inter-hub links are not assumed to be bottlenecks, i.e., are provisioned with really high data rates.

Since our ping measurements were conducted end-to-end, they capture the inter- and intra-regional components of latencies. In order to parametrize individual segments of a peer-to-peer path, we first create individual piecewise-linear distributions for each regional combination. We then establish intra-regional links in each region r and, as before, estimate individual link latencies $\hat{l}_{r,i}$ by sampling from the intra-regional distribution, divided by four. In the next step, we create a model for each of the inter-regional links between all regions r_0 and r_1. For this, we first calculate the means of the provisioned intra-regional latencies $\overline{L_r}$, as well as the mean measured peer-to-peer inter-regional latencies $\overline{L_{r_0,r_1}}$. We then calculate the estimated inter-regional link latency as $\hat{l}_{r_0,r_1} = \overline{L_{r_0,r_1}}/2 - \overline{L_{r_0}} - \overline{L_{r_1}}$, which is finally assigned to the corresponding edge.

V. Network Experiments

In the following, we present empirical experiments showing the validity of the introduced network model as well as the

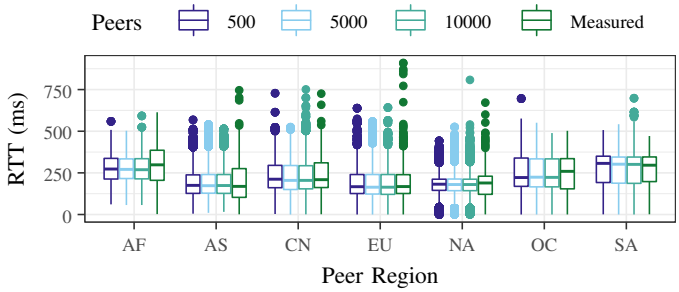

Fig. 5. Simulated latencies for different network sizes in comparison to measured values.

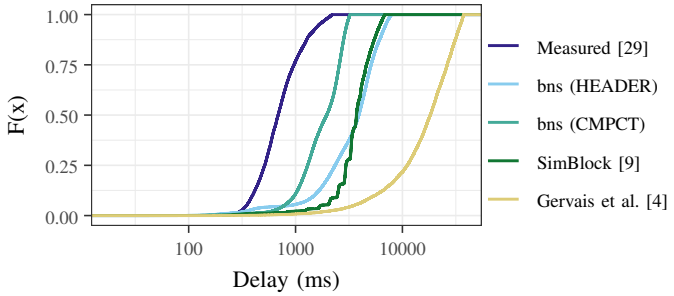

Fig. 6. Block propagation times as simulated by `bns` in comparison to real-world measurements and results by other blockchain simulators.

capabilities of the `bns` simulation framework.

A. Latency Model Validation

In order to investigate whether the results gathered from simulated network scenarios of different magnitudes still fit the real-world conditions, we validate the latency model of the `bns` simulation framework. To this end, we implemented a `ping` application that was deployed on random nodes in each of the seven regions of the geographic topology model. Each instance of the application was configured to retrieve latency measurements to random nodes located in all seven regions, a process which was repeated fifty times for each regional combination and fifty times overall in order to ensure the statistical significance of the results. The simulations were furthermore run in scenarios of different magnitudes, i.e., in networks with 500, 5,000, and 10,000 peers.

The simulation results are shown in Figure 5 in comparison to the real-world latencies we retrieved as part of our measurement study. We observe that in all cases the latency distribution is very stable and independent of the number of network peers. This indicates that the `bns` simulation framework is able to yield expressive results, even when only smaller scenarios are considered. Moreover, the simulated latency distribution resembles the measured real-world distribution very well, with means diverging only about 1-6 ms in most cases.

B. Propagation Model Validation

To further investigate whether the simulated geographically clustered network topology is able to produce valid results that closely resemble the properties of real-world blockchain networks, we compare block propagation times of `bns` with related work as well as independent real-world data. To this end, we simulated network scenarios with 500 nodes utilizing

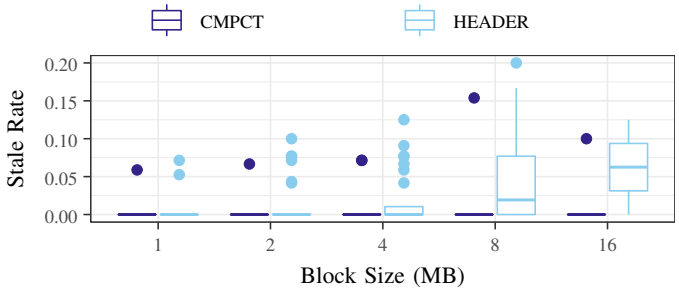

Fig. 7. Rate of stale blocks in dependence of block size and propagation method.

`bns`, the SimBlock [9] simulator, as well as the simulator introduced by Gervais *et al.* [4]. We moreover retrieved block propagation data collected by Neudecker [29, 30] as ground truth.

Figure 6 shows the simulation results in logarithmic scale as cumulative distribution function, which allows a visual comparison of the data. In order to quantify how well the simulated data sets approximate the measured data, we additionally calculated the *root mean squared error (RMSE)*. We observe that SimBlock (RMSE: 3,345 ms) yields results very similar to `bns` (RMSE: 3,590 ms) when the header-based propagation method is used. However, when compact block relaying is enabled, `bns` resembles the measured real-world data most closely, resulting in an RMSE of 1,415 ms. This does not come as a surprise, since Bitcoin's introduction of compact blocks [13] indeed reduced the average block propagation delay. The characteristics of the data retrieved from Gervais *et al.*'s simulator diverge from the real-world data set the most (RMSE: 20,066 ms). We conclude that the network model underlying the `bns` simulator enables valid simulations of blockchain network behavior.

C. Impact of Network Utilization

We furthermore investigate what impact different block sizes and propagation schemes have on the network utilization and the resulting block propagation process. To this end, we simulated the compact block and header-based propagation of 1, 2, 4, 8, and 16 MB blocks in scenarios with 500 nodes and analyzed the block propagation delay.

As to be expected, the average block propagation delay increases with the size of transferred blocks and is significantly decreased when compact block relaying is enabled. Starting from 4 MB, we observe a high network utilization for header-based relaying that increasingly leads to network congestion and in turn induces further packet losses and retransmits. Especially in edge cases and for lower-bandwidth peers, this leads to higher delays until blocks are received, which is reflected by increased standard deviations of the average block propagation delays. This behavior is of particular relevance, because when miners receive new blocks too late, they waste their mining power on producing stale blocks, which has been shown to negatively impact the security of the consensus layer [4]. Figure 7 therefore shows the rates of stale blocks, i.e., blocks that are finally not included in the blockchain, in

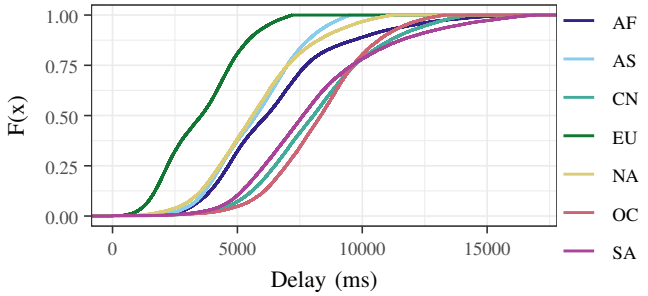

Fig. 8. Propagation delays in dependence of miner's geographic location.

dependence of the block size and propagation method. While the stale rate for header-based propagation remains negligible for 1 and 2 MB blocks, it rises for block sizes of 4 MB and above. Stemming from the high network utilization observed for 16 MB blocks, the average stale rate even surpasses 6%.

While these results show that the Bitcoin network currently could not handle header-based propagation of larger blocks without incurring security penalties, they also highlight that compact block propagation significantly improves the block propagation delay and network utilization. As shown in Figure 7, the simulation results indicate that Stale rates can in fact kept negligible for sizes of up to 16 MB and beyond, when these blocks are propagated through the compact block scheme.

D. Regional Influence on Block Propagation

In order to evaluate the regional influence, we configured `bns` based on Bitcoin parameters and ran simulations with a single miner deployed in each of the regions. These simulations were run in network scenarios with 500 peers and were repeated 150 times to ensure statistical significance.

Figure 8 shows the incurred delay in order to disseminate a new block to 90% of network peers in dependence of the miner's geographical location. Not surprisingly, miners in the EU, NA, and AS regions are able to propagate their blocks the quickest, exhibiting an average delay of 3.5 s, 5.7 s, and 5.7 s, respectively. In comparison, miners located in the AF, SA, and CN regions take more than 84% longer than EU miners to propagate their blocks in the network. Miners located in the OC region take the longest to propagate their blocks, resulting in an average block propagation delay of 8.5 s, 133% longer than miners in the EU region.

These results clearly show that the provisioned bandwidth and geographical location have a significant impact on miners' block propagation times. As such specific network characteristics may only be simulated based on a fine-grained network model, this scenario highlights the capabilities of the `bns` framework.

VI. Related Work

In recent years, a large body of literature studied various properties of real-world blockchain networks. While early entries were mainly concerned with the network topology and block propagation behavior [3, 31], more recent contributions measured the latency, bandwidth [29, 32], and mining power distributions [18, 33, 34], as well as the node churn [29, 35] exhibited by the Bitcoin network. Moreover, the peer-to-peer networks of Ethereum [17, 32, 36], Zcash [20], and Monero [37] have been explored in literature. However, most entries do not entail recent measurement results that consider the current state of blockchain network provisioning. To this end, our work provides a comprehensive model of the Bitcoin network based on an updated data set.

Furthermore, the behavior of blockchain networks can be studied based on a broad spectrum of models, tools, and simulations, reaching from testbeds with actually deployed prototypes to highly-abstracted simulated processes. Previously, Miller and Jansen [38, 39] proposed a simulator based on the actual Bitcoin Core source code, while abstracting lower-level network behavior. Contrastingly, the BlockSim simulator [40] highly abstracts from the application and network behavior, promising to enable a more lightweight simulation. The contributions most closely related to our work however are the Bitcoin network simulator introduced by Gervais *et al.* in [4] as well as the SimBlock [9, 41, 42] simulator. While the former makes use of ns-3's discrete-event network simulation, it relies on a simplified model of the network topology, such as establishing network links based on a random graph model congruent to the Bitcoin node's TCP connections. It thereby tends to yield idealized results. While SimBlock on the other hand adopts a more realistic geographical distribution of nodes and implements more recent protocol updates, such as compact blocks, it abstracts from the lower-level network protocols. To this end, both entries do not allow to consider more complex network effects, such as congestion with resulting queuing delays and packet losses. In contrast, the `bns` simulation framework builds upon the ns-3 simulator and models the link, network, transport, and application layers of blockchain nodes independently. This allows to study a larger variety of (sometimes interdependent) effects and—as our experiments confirm—yields more accurate simulation results.

VII. Conclusion

In this work, we characterized blockchain networks with reference to the paradigmatic Bitcoin network. To this end, we conducted a longitudinal measurement study from which we extracted a comprehensive model of the network behavior. We implemented this model as part of the `bns` network simulation framework and showed its validity and capability through empirical experiments.

References

[1] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008.
[2] Ethereum Project. "A next-generation smart contract and decentralized application platform." (2014), [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper.
[3] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *P2P '13: Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing*, Trento, Italy, Sep. 2013, pp. 1–10.
[4] A. Gervais *et al.*, "On the security and performance of proof of work blockchains," in *CCS '16: Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, Oct. 2016.

[5] K. Croman *et al.*, "On scaling decentralized blockchains - a position paper," in *BITCOIN '16: Proceedings of the 3rd Workshop on Bitcoin Research*, Christ Church, Barbados, Feb. 2016, pp. 106–125.

[6] J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *EUROCRYPT '15: Proceedings of the 34th International Conference on the Theory and Applications of Cryptographic Techniques*, Sofia, Bulgaria, Apr. 2015, pp. 281–310.

[7] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *EUROCRYPT 17: Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Paris, France, 2017, pp. 643–673.

[8] L. Kiffer, R. Rajaraman, and A. Shelat, "A better method to analyze blockchain consistency," in *CCS '18: Proceedings of the 25nd ACM SIGSAC Conference on Computer and Communications Security*, Toronto, ON, Canada, Oct. 2018, pp. 729–744.

[9] Y. Aoki *et al.*, "Simblock: A blockchain network simulator," in *INFOCOM '19: Proceedings of the 2019 Conference on Computer Communications Workshops*, Paris, France, Apr. 2019, pp. 325–329.

[10] Bitcoin Core. "Homepage." (2019), [Online]. Available: https : / / bitcoincore.org.

[11] G. C. Fanti and P. Viswanath, "Deanonymization in the bitcoin P2P network," in *NIPS '17: Proceedings of 30th Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA, Dec. 2017.

[12] S. Daftuar. "Bip 130: Sendheaders message." (May 2015), [Online]. Available: https : / / github . com / bitcoin / bips / blob / master / bip - 0130 . mediawiki.

[13] M. Corallo. "Bip 152: Compact block relay." (Apr. 2016), [Online]. Available: https : / / github . com / bitcoin / bips / blob / master / bip - 0152 . mediawiki.

[14] P. Maymounkov and D. Mazires, "Kademlia: A peer-to-peer information system based on the XOR metric," in *IPTPS '02: Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, Cambridge, MA, USA, 2002, pp. 53–65.

[15] Ethereum Project. "Devp2p network protocols." (2019), [Online]. Available: https://github.com/ethereum/devp2p.

[16] S. A. Henningsen *et al.*, "Eclipsing ethereum peers with false friends," in *S&B '19: Proceedings of IEEE Security & Privacy on the Blockchain*, Jun. 2019, pp. 300–309.

[17] S. K. Kim *et al.*, "Measuring ethereum network peers," in *IMC '18: Proceedings of the Internet Measurement Conference*, Boston, MA, USA, Oct. 2018, pp. 91–104.

[18] A. Gervais *et al.*, "Is bitcoin a decentralized currency?" In *SP '14: Proceedings of the 35th IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2014, pp. 54–60.

[19] B. Kaiser, M. Jurado, and A. Ledger, "The looming threat of china: An analysis of chinese influence on bitcoin," *CoRR*, vol. abs/1810.02466, 2018.

[20] E. Daniel, E. Rohrer, and F. Tschorsch, "Map-z: Exposing the zcash network in times of transition," in *LCN '19: Proceedings of the 44th IEEE International Conference on Local Computer Networks*, Osnabrck, Germany, Oct. 2019.

[21] Bitnodes. "Homepage." (2019), [Online]. Available: https : / / bitnodes . earn.com.

[22] T. T. Group. "TCPDUMP/LIBPCAP public repository." (2020), [Online]. Available: https://www.tcpdump.org/.

[23] I. MaxMind. "Geoip geolite2 database." (2019), [Online]. Available: https://dev.maxmind.com/geoip/geoip2/geolite2/.

[24] S. B. Venkatakrishnan, G. C. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, 2017.

[25] M. Bailey *et al.*, "The menlo report," *IEEE Secur. Priv.*, vol. 10, no. 2, pp. 71–75, 2012.

[26] T. R. Henderson *et al.*, "Network simulations with the ns-3 simulator," *SIGCOMM Demonstration*, vol. 14, no. 14, p. 527, 2008.

[27] E. Rohrer and F. Tschorsch, "Kadcast: A structured approach to broadcast in blockchain networks," in *AFT '19: Proceedings of the first ACM conference on Advances in Financial Technologies*, Zurich, Switzerland, Oct. 2019.

[28] blockchain.com. "Hashrate distribution." (2019), [Online]. Available: https://blockchain.com/pools?timespan=4days.

[29] T. Neudecker, "Characterization of the bitcoin peer-to-peer network (2015-2018)," Karlsruhe, Tech. Rep. 1, 2019, 29 pp.

[30] K. I. of Technology DSN. "Bitcoin monitoring." (2020), [Online]. Available: https://dsn.tm.kit.edu/bitcoin/.

[31] J. A. D. Donet, C. Prez-Sola, and J. Herrera-Joancomart, "The bitcoin p2p network," in *BITCOIN '14: Proceedings of the 1st Workshop on Bitcoin Research*, Barbados, Mar. 2014, pp. 87–102.

[32] A. E. Gencer *et al.*, "Decentralization in bitcoin and ethereum networks," in *FC '18: Proceedings of the 22nd International Conference on Financial Cryptography and Data Security*, Santa Barbara, Curaao, Feb. 2018.

[33] L. Wang and Y. Liu, "Exploring miner evolution in bitcoin network," in *PAM '15: Proceedings of the 16th International Conference on Passive and Active Measurement*, New York, NY, USA, Mar. 2015, pp. 290–302.

[34] C. Wang, X. Chu, and Q. Yang, "Measurement and analysis of the bitcoin networks: A view from mining pools," *CoRR*, vol. abs/1902.07549, 2019. arXiv: 1902.07549.

[35] M. A. Imtiaz *et al.*, "Churn in the bitcoin network: Characterization and impact," in *ICBC '19: Proceedings of the 2019 International Conference on Blockchain and Cryptocurrency*, Seoul, Korea (South), May 2019, pp. 431–439.

[36] P. Silva *et al.*, "Impact of geo-distribution and mining pools on blockchains: A study of ethereum," in *DSN '20: Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Valencia, Spain, Jun. 2020, pp. 245–252.

[37] T. Cao *et al.*, "Exploring the monero peer-to-peer network," in *FC '20: Proceedings of the 24th International Conference on Financial Cryptography and Data Security*, Kota Kinabalu, Malaysia, Feb. 2020, pp. 578–594.

[38] A. Miller and R. Jansen, "Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications," in *CSET '15: Proceedings of the 8th Workshop on Cyber Security Experimentation and Test*, Washington, DC, USA, Aug. 2015.

[39] R. Jansen and N. Hopper, "Shadow: Running tor in a box for accurate and efficient experimentation," in *NDSS '12: Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, 2012.

[40] M. Alharby and A. van Moorsel, "Blocksim: A simulation framework for blockchain systems," *SIGMETRICS Performance Evaluation Review*, vol. 46, no. 3, pp. 135–138, 2018.

[41] R. Banno and K. Shudo, "Simulating a blockchain network with simblock," in *ICBC '19: Proceedings of the 2019 International Conference on Blockchain and Cryptocurrency*, Seoul, Korea (South), May 2019, pp. 3–4.

[42] R. Nagayama, R. Banno, and K. Shudo, "Identifying impacts of protocol and internet development on the bitcoin network," in *ISCC '20: Proceedings of the 2020 Symposium on Computers and Communications*, Rennes, France, Jul. 2020, pp. 1–6.