2019 IEEE International Conference on Blockchain (Blockchain)

Evaluating The Impact of Network Latency on The Safety of Blockchain Transactions

1st Luming Wan
Department of Computer Science
University of Otago
Dunedin, New Zealand
lwan@cs.otago.ac.nz

2nd David Eyers
Department of Computer Science
University of Otago
Dunedin, New Zealand
dme@cs.otago.ac.nz

3rd Haibo Zhang
Department of Computer Science
University of Otago
Dunedin, New Zealand
haibo@cs.otago.ac.nz

***Abstract*—Blockchain came to prominence as the distributed ledger underneath Bitcoin, which protects the transaction histories in a fully-connected, peer-to-peer network. For safety against double-spending, a bitcoin transaction is only considered to be unrevokable after it is within a chain of at least six mined blocks on the blockchain—termed "six confirmations" for short. Besides the use for securing financial activity, blockchain technology is now also merged with various emerging applications, including Internet of Things (IoT) and vehicular ad-hoc networks (VANET). However, such emerging applications may have comparatively low Internet connectivity due to incorporating mobile devices, that may range away from network infrastructure. This paper investigates the impact of network latency on blockchain forking behavior and possible violations of the aforementioned six confirmations convention for transaction approval. To speed up our simulations, we simplify the blockchain's data structure, and avoid the extensive computation required in proof-of-work (PoW). Through simulation, we show that the six confirmations convention is sensitive to the peer-to-peer network latency, and also show how quickly it is violated with an easier difficulty of PoW mining. Unsurprisingly, the speed at which all nodes on the blockchain converge is shown to be severely affected by the latency of the underlying peer-to-peer network. It is also shown the extent to which nodes with more frequent Internet connectivity can gain an unfair advantage in terms of proof-of-work mining rewards. We thus recommend that networks with heterogeneous latency profiles across nodes monitor fairness, and potentially preclude some nodes from being miners.**

***Index Terms*—Blockchain, network latency, block convergence, six confirmations, heterogeneous network latency**

I. Introduction

Bitcoin, the peer-to-peer payment system and cybercurrency introduced by Satoshi Nakamoto [1], has gained astonishing success in both its peaks in market price, and the number of its cybercurrency users. Its success is mostly due to its underlying security technology—namely blockchain—which is a distributed ledger that securely stores bitcoin transaction history. Bitcoin transactions can be easily traced and are hard to tamper, given the protection of blockchain. Bitcoin's success has focused attention on both blockchain-based cybercurrency and other blockchain-based applications.

The key feature of blockchain is its decentralised security over a peer-to-peer untrusted network. This is achieved by ensuring the majority of cybercurrency users converge on consistent blockchain data. Proof-of-Work (PoW) schemes make changing the blockchain expensive, and block hash chaining ensures that any change to blocks (and thus transactions) is detectable. Therefore, tampered blocks can be easily detected with low computational complexity, by agreement of the distributed inspection of the chains on many nodes. In this paper, we define *block convergence* as all nodes agreeing on an identical blockchain.

The block convergence of blockchain can be easily disrupted by increased network latency. Ideally, nodes should hear about freshly mined blocks as quickly as possible. Faster spreading to the network of a new block enables it to be validated earlier by other nodes, allowing them to update their blockchains with the new validated block. Thus the blockchain will stabilise back to a globally converged state within shorter timeframes. However, as network latency increases, the blockchain has a much higher chance to form forks due to multiple new blocks being mined by different nodes. The variety of new blocks will result in honest nodes being unsure as to which of the forks will end up being established as the longest chain, thus destabilising global consensus. The divergence of these new blocks expose the corresponding involved transactions to the risk of ending up on a blockchain fork that is not adopted. In scenarios with variable network latency for different nodes, nodes with lower latency will gain disproportionate control though selfish mining. For some emerging technologies that are designed for a high-latency network communication environment, the security of blockchain could become fragile with delayed communication between blockchain maintainers.

There are few studies investigating the impact of network latency on the performance of blockchain. Most existing blockchain-based systems are designed under the assumption of stable peer-to-peer connectivity. The delay of synchronisation is not a focus in their work. Existing research has instead well examined the influence from the typical delays that exist in real-world blockchain systems, and has explored non-security perspectives of blockchain performance, such as the rate of blocks being exchanged, or the throughput of transactions into the blockchain. Gencer *et al.* [2] and Decker *et al.* [3] investigate the impact of network latency to transaction throughput and the ratio of lost ledger replica information. However, their experiments are conducted in environments

978-1-7281-4693-5/19/$31.00 ©2019 IEEE
DOI 10.1109/Blockchain.2019.00033

with network latencies ranging from milliseconds to seconds. To the best of our knowledge, there is not existing research that measures comprehensively how much latency would interfere with the security of blockchain, or that studies the effect of latencies that range up to the minute or hour level. We also provide guidance on the feasibility of deploying blockchain to environments with high or highly-variable network latency.

In this paper, we focus on analysing the speed of block convergence and the violation of the six confirmation convention of blockchain when nodes may have the network latency configured from 10 minutes, which is the target average Bitcoin block generating time adaptively adjusted by PoW mining difficulty, up to two hours.

The contributions of this paper are:

1) We analyse the trade-off between block convergence speed of blockchain with either various population size or PoW mining difficulty, under different extents of network latency.
2) We measure the safety of applying the six confirmations convention at different levels of network latency.
3) We also conduct several simulations that model heterogeneous network latency. We quantify the extent to which nodes with lower latency gain additional unfair advantage for PoW mining.

The rest of this paper is structured as follows: Section II introduces some state-of-the-art studies on the effects of network latency, and the six confirmation convention, on blockchain. Section III defines the term *block convergence* in detail, and introduces how the block convergence is maintained by real-world Bitcoin blockchain users. Section IV introduces the six confirmation convention. Section V illustrates the experimental setup for our blockchain simulations. Section VI shows the impact of network latency on various blockchain properties, and section VII summaries the conclusions of our research.

II. Related work

Blockchain was firstly introduced by Satoshi Nakamoto in his Bitcoin paper [1] in 2009, as blockchain provides an advanced decentralised peer-to-peer security for Bitcoin transactions without the need of authorised third parties. Beyond cybercurrency activities protection, the convergence of Blockchain technology with various emerging network applications brings numerous opportunities as well as challenges. The suitability of applying Blockchain technology to different scenarios has been reviewed in publications from Crosby *et al.* [4] and Lo *et al.* [5]. Some researchers designed blockchain-based infrastructure for Internet-of-Things (IoT) [6], [7], and the privacy and legality concern behind Blockchain technology is also discussed. Sharma *et al.* [8] examined emerging VANETs in smart city applications, and a distributed architecture for VANET based on blockchain is further introduced. They also introduced the combination of Software Defined Network (SDN) with Blockchain technology for cloud storage in later contribution [9], to achieve a minimal end-to-end delay between IoT devices. Dorri *et al.* [10] designed a lightweight scalable blockchain for VANET.

Few publications explore the impact of network disconnection on the performance of blockchain. Gencer *et al.* [2] conducted a comprehensive measurement study of decentralisation metrics for two leading cyptocurrencies, bitcoin and Ethereum. Their results indicate that Bitcoin users are geographically closer together than Ethereum users, so typically has a network latency less than 100ms. They also showed that many of the Bitcoin nodes are run in datacenters. Decker and Wattenhofer [3] discovered that network latency critically affects the propagation of blocks smaller than 20KB.

Many blockchain overview and survey publications [11], [12] have mentioned the six confirmations convention to reduce the risk of revocation of transactions on blockchain forks that end up being discarded. A new cryptocurrency was claimed in [13] namely ByzCoin, which optimises transaction commitment and verification while still guaranteeing safety and liveness under Byzantine faults. ByzCoin only needs one block confirmation to approve a transaction, which hugely improves the transaction throughput compared to the original blockchain. Rosenfeld [14] calculated the possibility of double-spending can be raised within the Bitcoin blockchain based on changing the numbers of confirmations required, and the computation power of attackers. To the best of our knowledge, our work is unique in conducting a comprehensive survey on the impact brought by large-scale network latency to the fundamental and important properties of blockchain, especially targeting block convergence and the feasibility of applying the six confirmations rule.

III. Block Convergence

Blockchain has such effectiveness in practice because of the global unification of the blockchain data that maintained by the majority of the mining population. In this paper, we define *block convergence* of blockchain as the state in which every single node in the network maintains exactly the same chain copy, with a consistent sequence on both the blocks on the longest chain and the transactions inside each block. The blockchain managed by different Bitcoin users need to periodically achieve the state of block convergence, by synchronising new validated blocks, or updating itself to a longer blockchain from other nodes network-wide. Blockchain provides a positive environment for nodes to align themselves towards block convergence, which is achieved through PoW mining scheme and blockchain consensus protocol supported by stable and fast Internet connectivity. PoW mining is computation that is *hard to solve* by blockchain miners themselves, but *easy to validate* for other users. For each newly mined block, miners have to solve a hash puzzle with an agreed difficulty level. The simplicity of hash-based PoW leads to extremely low computational difficulty in validating the entire blockchain. Any modification to block content leads to a detectable change on the PoW hash, and re-solving a PoW hash puzzle remains as highly computational expensive as the original published solution. Blockchain consensus protocol requires all the nodes broadcast both unapproved transactions and new generated blocks. Compared to the relatively long

time for blockchain block generation (around 10 minutes, on average), synchronising a new block is much faster for nodes with stable Internet connectivity. Hence, nodes in the network can quickly adapt back to a block converged state, and those nodes' blockchains will be stable most of time. Because of this, an extreme powerful decentralised security of blockchain is guaranteed, which brings success for most of the real world blockchain-based cryptocurrencies and applications.

It is widely believed that by controlling at least 51% of the net computational power, malicious attackers can successfully take control over the blockchain. Actually, effective attacks can be easier, since it has been proved in [15]–[17] that attackers have the ability to perform Byzantine fault attacks controlling only one third of the total computation power. In any case, it is desirable to have as many nodes as possible maintain an identical blockchain, to guarantee the decentralised immutability of the blockchain.

Bitcoin blockchain will not quiesce into a long-term, stable block convergence state, because so many incoming transactions are waiting to be added into blockchain, Nonetheless, it is still necessary to get some insights on block convergence. This is because the integrity of blockchain mainly comes from the global block convergence state. PoW and the consensus protocol underneath blockchain are all designed for the purpose of reaching block convergence for all the blockchain nodes. However, block convergence is highly sensitive to many environmental factors. Slower block convergence causes degradation of the decentralised security of blockchain, and raises risks of malicious attacks. In this paper, we mainly focus on investigating the degree of negative influence to block convergence that comes from exploring increasing peer-to-peer network latency.

IV. Six Confirmations Convention

To reduce the risk of double-spending, a Bitcoin transaction typically is only treated as being permanently confirmed when it is six blocks deep in the longest accepted (known) blockchain [11], [12]. Although transactions are cryptographically signed within new blocks when they are firstly entered into a blockchain, the transaction is not allowed to be cleared for the transacting parties: because the block is not buried deep enough inside a blockchain, it has a comparatively high chance of ending up on a chain that is replaced by a longer chain, when forks are resolved. Thus to treat the transaction as cleared, typically both transacting parties will wait for six subsequent blocks to be appended to the chain containing the transaction. In real-world Bitcoin blockchain, the longest chain will usually only be one block ahead of other blockchain forks due to the high availability of network connectivity. Six confirmations makes committed transactions extremely unlikely to be revoked from the blockchain.

There is not a special reason of choosing six as the default number of confirmations, and it can be flexibly changed by updating the consensus rule of blockchain software. However, Meni Rosenfeld [14] summarised the possibility of successfully launching double spending attacks under various combinations of different numbers of block confirmations and the computation power of attackers. His calculations conclude that with six confirmations for transaction commitment, the risk of double-spending is negligible ($< 0.1\%$) even under the unlikely situation that attackers amass up to 10% of the entire computation power of the blockchain network.

Of course, a transaction followed by six blocks is not automatically safe from being revoked. As noted above, the transaction's block could still be discarded (rolled back) due to soft/hard forks caused by consensus change or software updates. For example, when an additional constraint on blockchain consensus rule needs to be applied to the new versions of software, a soft fork is triggered for nodes to upgrade to the newest blockchain. The blocks generated by the old version miners will be treated as invalid to the nodes with new version blockchain consensus. Continued rejection of the mined old-version blocks will force blockchain nodes to catch up to the latest released version of blockchain.

Case Study: More than 15 soft fork events are recorded in Blockchain Improvement Protocol (BIP) [18]. BIP16 [19], [20] is one of the most famous soft forks of the Bitcoin blockchain, and happened in 2012. The purpose of the soft fork was to officially standardise additional validation rules to the new transactions, namely "pay to script hash" (P2SH). The newest version of consensus rule can only be activated if more than 55% of the users update to the newest software within seven days. However, the evaluation point was hugely delayed, and the miners who used the new software version had their blockchains get stuck at a block height of 170,060 because the remaining 45% of miners produced invalid blocks for several months after P2SH was released. Later BIP34 [21] prevented this problem by raising the activation threshold to 95%.

The above case study demonstrates that a large number of blocks containing transactions can be revoked because of manually updating consensus or software. Besides this, network latency, while not a common problem in today's blockchain community with good network connectivity, still has the potential to violate six confirmations of blockchain in a much easier way compared to the soft/hard fork scenarios. In this paper, we investigate the impact of various network latency levels to how convergence occurs within blockchain, and the safety of applying the convention of waiting for six subsequent blocks for these different network latency settings.

V. Experimental Setup

The goal of our simulation is to investigate the impact of network latency on the safety of blockchain. The blockchain we implemented uses a much simplified structure compared to the original Bitcoin blockchain. The PoW mining is modelled in a much less computational approach, which eliminates what would have been an impractical workload for our blockchain simulator. To the best of our knowledge, there has not been a blockchain simulation that has focused on investigating the impact to blockchain security caused by a large ranges of network latency.

A. How to Reach Block Convergence?

To measure the speed of block convergence, we define the concept of block convergence in our simulation as how much time is needed for all the nodes to finally obtain exactly the same blockchain, with all pre-generated transactions included. Therefore, the number of transactions has to be limited as well as their generating period. After the transaction generation, PoW winners still create empty blocks without transactions included, and nodes still communicate and synchronise their blockchain with others as they do throughout the rest of simulation time. The simulation is terminated when all the nodes reach the block convergence state.

B. Nodes and Communication

Due to hardware and simulation time limitations, it is better to simulate the activities within a competing mining pool, rather than a large area of blockchain community. Therefore, the node population size is relatively small in our simulation, with the configuration options of 20, 30, 40, 50 and 60, respectively. Also it is constant without the consideration of nodes dynamically involving and exiting the network during a given simulation. Nodes are initialised with a network disconnection interval which is randomly chosen from an appropriate time range. The latency time setting for each node might be different, and this is further introduced in SectionV-C.

It is hard to quantify the impact of random peer-to-peer network latency on the safety of blockchain, because unnormalised latency will bring unpredictable bias to the final statistics. To address this problem, we model the peer-to-peer communication as nodes periodically connecting to an additional global reachable node which denotes Internet connectivity. Except randomly picking a latency time at the very start of simulations, nodes will have a constant Internet disconnection intervals during the rest of simulation time to avoid further random contact inside population. The subset of nodes that connect to the Internet node indicates the availability of stable peer-to-peer connectivity among all of them within the current time unit (i.e., 600 seconds). When a pair of nodes have Internet connectivity, they perform the following three tasks:

1) **Uncommitted transaction exchanging:** Both nodes exchange their transactions inside *mempool*. Of course, the already received transactions will not be accepted by a node twice.
2) **Blockchain validation:** nodes validate each other's blockchains by inspecting the hash correctness of each block.
3) **Blockchain updating:** nodes always replace their blockchain with a longer one, if such a blockchain is exchanged with them.

C. Network Latency Setting

Let l denote the current network latency, which is the disconnection time for a node away from Internet. Let p be the network population, and *time_unit* represents the time unit for simulation, which we set to 600 seconds—equal to the target average block generation interval for the original bitcoin blockchain. For the simulations with homogeneous latency configuration, the latency time of a node is randomly chosen from the range of [*l, l+time_unit*].

To explore the impact of different degrees of heterogeneity of network latency on the performance of blockchain, it is helpful to avoid large variance within the statistics introduced by random distribution of peer-to-peer latency. Therefore, we normalise nodes latency into different heterogeneous network latency groups (HNL). A simulation is held under the any of the configuration options of HNL1, HNL2, HNL3 and HNL4, respectively. In each case the n in HNLn indicates how many heterogeneous groups the entire population will be evenly divided into. Each group of nodes have their latency chosen from an appropriate range, which is given as follows:

1) HNL1: homogeneous group. Network latency of nodes are always within the range of [*l, l+time_unit*].
2) HNL2: two heterogeneous groups. The network latency of first half of the population is set within the range [*l, l + time_unit*], while the second half of the nodes has [*2l, 2l + time_unit*].
3) HNL3: three heterogeneous groups, with the network latency settings of [*0.5l, 0.5l + time_unit*], [*l, l + time_unit*], and [*2l, 2l + time_unit*]. The entire population is equally divided into these three groups.
4) HNL4: four heterogeneous groups, with the network latency settings of [*0.5l, 0.5l + time_unit*], [*l, l + time_unit*], [*2l, 2l + time_unit*], and [*4l, 4l + time_unit*]. Each group has the same size of population.

D. Blockchain Structure and Transaction Storage

All nodes are initiated with an empty chain and an unapproved transaction list, namely *mempool*. The uncommitted transactions will be temporarily stored in *mempool* before they are added into blocks on the blockchain. Since the removal of transactions due to fulfilled *mempool* is unnecessary in our simulation, our *mempool* is unbounded. Similarly, there is no limitation on the capacity of a block for the approved transactions. Every time a mining winner generates a new block, all the transactions in *mempool* are included into the new block, and *mempool* is emptied entirely of transactions.

The blockchain structure is much simplified from the original one for Bitcoin. Consistent with the original Bitcoin blockchain, nodes still maintain a linear time-stamped structure blockchain, and the blocks are interconnected by hash consecutiveness. However, a block maintains fewer components, specifically:

1) **Index number:** the index of the current block.
2) **Approved transaction list:** the bitcoin transactions.
3) **Previous block hash:** the overall hash value of previous block.
4) **Block generator:** the ID of the node who generates this block.

Since the consistency of transaction content is out of the scope of this paper, it is not necessary to store extra bits for the

transaction hashes using Merkel Trees. Instead, they are simply stored in the block without any encryption or compression. We only need to keep track of the timestamp of when the longest blockchain contains all the generated transactions, and the longest chain is obtained by all the nodes. Therefore, how the transactions are stored is not relevant in our simulation.

E. Selection of Parties within a Transaction

The payer and payee of a simulated Bitcoin transaction is randomly chosen from the node population. Due to the need to reach a final block convergence state, it is necessary to constrain the number of transactions and their generation period. Based on the current simulator's speed, a total of 3,000 transactions was selected, and their invocation times are evenly distributed within 90,000 seconds, i.e., 20 transactions per *time_unit*. Due to the small population size within our selected configurations, 3,000 transactions is sufficient to support our simulation generating useful results. Beyond the period of transaction generation, nodes still keep generating blocks and synchronising their blockchain with others. The simulation is terminated until nodes reach the state of block convergence for their blockchain.

F. Selection of PoW Winners

The difficulty of PoW mining is also a critical factor to the performance of blockchain. We model our PoW mining difficulty by controlling the number of winners picked during each *time_unit*. Unsurprisingly, to speed up simulation, nodes are not required to solve actual PoW mining challenges. Instead, PoW winners are randomly selected from the nodes' population in each *time_unit*. There are three, six, or nine winners as configuration options, which indicates the number of winners needs to be selected within each *time_unit*. The selected winners will generate a new block and append it to their blockchain. All their pending transactions inside their *mempool* are approved and added into the new block.

Despite avoiding actual PoW computation, the growing of blockchains is still correctly simulated. All the nodes have equal probability to be selected as a PoW winner, which indicates an equal computation power for all the nodes on their PoW mining. The number of winners can be treated as the difficulty level of PoW mining. The more the number of winners, the simpler for mining difficulty. The number of winners is constant during an experiment. This approach of modelling PoW mining enables us to accurately quantify the impact of mining difficulties to blockchain performance in a lightweight manner.

G. Synchronisation of Blockchain Forks

Since the real PoW mining is not implemented in our simulation, a single blockchain cannot be measured in term of PoW difficulty. Therefore, the length of blockchain is the only criteria for deciding which blockchain fork should persist. A node always updates its blockchain to a longer chain during a peer-to-peer communication. However, we possibly meet blockchain synchronisation conflict due to the existence of

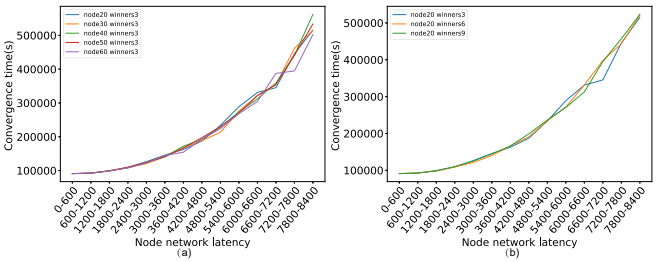

Fig. 1. Convergence speed of different size of population with (a) homogeneous network latency, and (b) different number of winners.

multiple same-length blockchain forks, which is caused by the selection of multiple winners within each *time_unit*. In this case, a node always chooses the first, longest chain it meets, and ignores all the other same-length blockchains that it might discover, later. In addition, blockchain forking might lead to some committed transactions inside the removed blocks have to be re-approved, because they were not committed into the longest chain. These transactions will be recycled back to the *mempool* and wait to be signed into a future block by simulated PoW mining.

VI. Experimental Results

In this section, we evaluate the impact of different network latency settings on blockchain security in terms of block convergence speed, safety of the six confirmations convention, length of the longest chain, and the maximum and average number of blocks that get rolled back, and thus revoke transactions. We also examine the impact of heterogeneous connectivity to the security of blockchain. In order to counteract the variance in behaviour caused by to desired randomness within the simulator, each reported statistic is determined based on 40 runs of the simulator.

A. Convergence Speed

The convergence speed versus homogeneous network latency under different number of population and winners is shown in Fig.1 (a) and (b), respectively. The y-axis represents the convergence time, which is the total amount of time required to reach the block convergence state under different settings on network latency. Note that the last transaction is generated at 90000s of simulation time, thus the period after it, i.e., *convergence time* $-$ 90000, is the time spent on reaching the block convergence. As shown in the figures, the convergence time of blockchain is proportional to the peer-to-peer network latency. With almost any amount of network connectivity (i.e., the network latency is in the range of 0–600), the block convergence state can be immediately reached by nodes before the next block generation time. However, it needs around an hour ($\approx$ 94000 $-$ 90000) for nodes to reach the global block convergence state when the network latency is extended to 600–1200 seconds, and almost one day when the network latency is set to 4200–4800 seconds. After this, the convergence speed is further exponentially increased, and

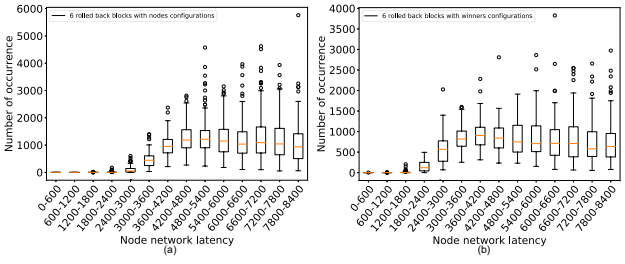

Fig. 2. The occurrence times of six rolled-back blocks for revoked transactions versus (a) the number of nodes, and (b) the number of winners.

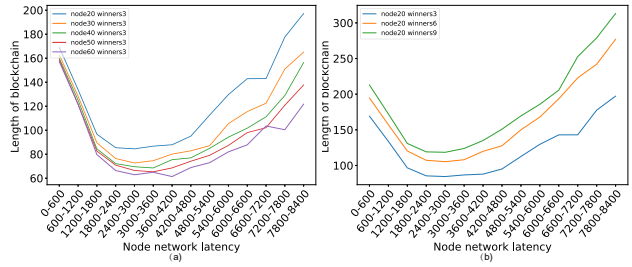

Fig. 3. The length of blockchain versus network latency (a) with different number of nodes, and (b) with different number of winners

it takes almost four days when the peer-to-peer latency is set to two hours (7200–7800 seconds).

It can also be observed from Fig.1 (a) and (b) that neither the node population nor the number of winners has an obvious impact on the convergence time. This is because there is plenty of randomness in the simulator, which is introduced by random winner selection, selection of transaction parties, and different network latency between nodes. These random factors add variance to the convergence time in each run of the simulation.

B. Six Confirmations Feasibility

Unsurprisingly, the six confirmations convention can easily be violated if the PoW mining difficulty is reduced, as nodes could mine a long, local blockchain entirely on their own. Also, a long network latency is also a critical factor to fail six confirmations. However, it is not clear that under which level of network latency the six confirmations convention has negligible risk of being violated. Fig.2 indicates the safety of the six confirmations convention with all the number of nodes and winners settings. Fig.2 (a) shows the distribution range for the number of occurrences of six blocks revocation for all the simulations with various population size configurations together, and Fig.2 (b) covers all the simulations for changing the number of winners selected in each mining round. The number of occurrences of six rolled-back blocks means how many times that six blocks revocation happened during a simulation, which is collected from all the blockchain forking events. Having no occurrence of six rolled-back blocks indicates we do not violate the six confirmations convention for the corresponding network latency range. As shown in Fig.2 (a), the phenomenon of six rolled-back blocks start to appear since the range of 1200–1800 seconds for network latency. However, the failure of six confirmations convention appears at all latency levels for the simulations with various the numbers of winner settings as shown in Fig.2 (b). This demonstrates that the six confirmation convention could be easily violated by having more simultaneous winners in each time unit, or in other words, a simpler difficulty of PoW mining.

C. Length of Blockchain

Fig.3 displays the number of blocks needed on the longest blockchain to reach global block convergence. It can be seen that the length of blockchain decays until the latency setting of 2400–3000s, but then it grows afterwards. This is because when nodes have continuous connectivity, the new generated blocks can be spread immediately to the network. However, when the disconnection time is slightly longer, the propagation of new generated blocks is disrupted. Thus these new blocks could be replaced by a longer chain during next time of nodes' communication, which results in the shortening of blockchain size. With the network latency further expanded, the time spent on reaching block convergence is significantly augmented, so that more blocks will be generated within the extended period, thus having a much longer blockchain.

Fig.3 (a) displays the dependency of length of the blockchain with different population size. The figure shows that the length of blockchain is inversely proportional to the population of nodes. This is because a larger network population results in more communications in the network, which leads to more opportunity for nodes to have blockchain synchronisation take place.

Fig.3 (b) exhibits the relation between the length of the blockchain and the number of winners. It can be seen that the length of blockchain is perfectly proportional to the number of winners at any network latency. This is because the more winners are selected in each time, the higher possibility of having more blockchain forks growing at the same time, and also a higher risk of suffering synchronising conflict due to multiple same length blockchain forks exist.

D. Maximum and Average Number of Rolled-back Blocks

Fig.4 shows the maximum and average number of rolled-back blocks under various experimental configurations. The maximum rolled-back blocks as shown on the y axes of Fig.4 (a) and (b) indicates the largest number of blocks being revoked we met in the simulations, and the average rolled-back blocks for Fig.4 (c) and (d) is the average number of rolled-back blocks collected from all the blockchain merging. It can be seen in the figures that the increasing trends of both maximum and average number of rolled-back blocks are exactly the same. As shown in Fig.4 (a) and (c), the maximum and average number of rolled-back blocks are inversely proportional to the population size. This is because with a larger population, nodes have more chance to communicate and synchronise their blockchain with others. Thus fewer new blocks, which are generated within nodes' time of disconnection, will be replaced because of more frequent merging of blockchain forks.

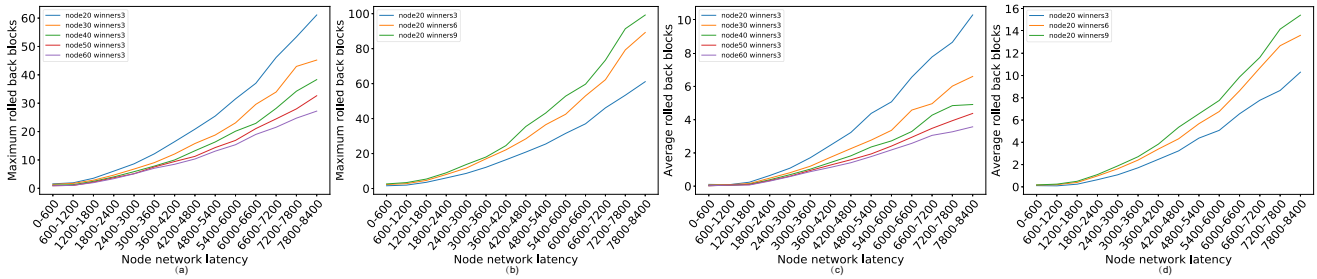

Fig. 4. The maximum number of rolled-back blocks with (a) different size of population, (b) different number of winners, (c) the average number of rolled-back blocks with different size of population, and (d) different number of winners.

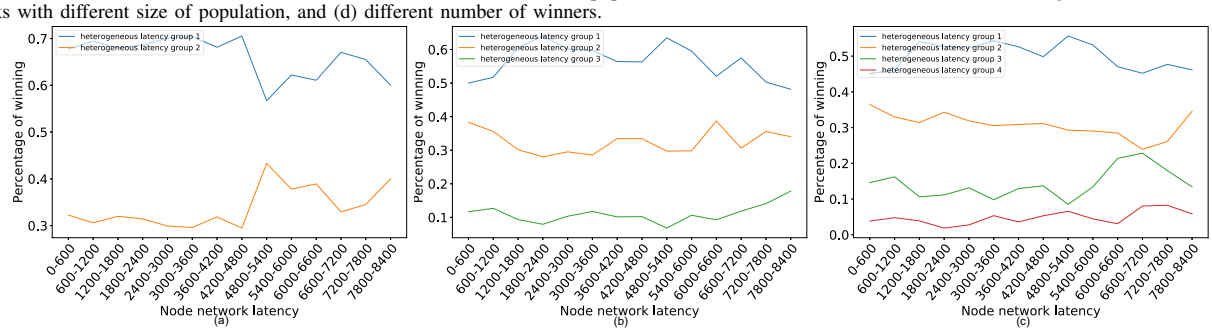

Fig. 5. The percentage of blocks belongs to each heterogeneous group members on the longest blockchain, with the heterogeneous latency groups setting of (a) HNL2, (b) HNL3, (c) HNL4

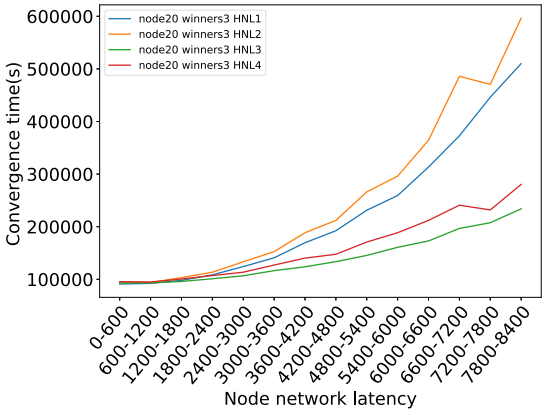

Fig. 6. The convergence speed of various heterogeneous latency group configurations

Fig.4 (b) and (d) demonstrate—unsurprisingly—that more winners per time unit lead to more rolled-back blocks. With a simpler PoW mining difficulty, there will be more nodes that can solve hash puzzles within a time unit, so that more blocks can be generated within the same period. Multiple equal-length blockchain forks could be simultaneously grown, and consequently, more blocks will be replaced when the merging of blockchain forks occurs in the near future.

E. Impact of Heterogeneous Connectivity

This section investigates the impact to blockchain security brought by different heterogeneous network connectivity. This is evaluated in terms of proportion of winning for each heterogeneous latency group, the convergence speed, and the length of blockchain, as depicted in Fig.5, Fig.6, and Fig.7, respectively. The proportion of winning of a heterogeneous latency group indicates the percentage of blocks on the longest accepted blockchain that are generated by the members of this latency group. The network population is set to 20 and there are only three winners within each time unit for the simulations mentioned in this section.

Fig.5 shows the proportion of blocks generated by each heterogeneous latency group of nodes on the longest blockchain. The blocks are mainly generated by the nodes with the lowest latency. and that group still creates more than 50% of the blocks in the experiments of HNL3 and HNL4, as shown in Fig.5 (b) and (c). The nodes with the largest network latency generate only 30%, 10% and 5% of blocks, respectively. Note that all the nodes have equal computational power in our PoW mining model. Therefore, our experiments demonstrate that the nodes with low network latency have a much higher chance to mine the longest chain, thus able to gain the majority of blockchain reward from PoW winning.

Fig.6 displays the convergence speed of blockchain under various heterogeneous network latency settings. It can be seen that the convergence speed of the heterogeneous groups HNL3 and HNL4 is up to three times faster than HNL1 and HNL2. This is because the nodes with lower network latency, which the disconnection time is within the range of [*0.5l, 0.5l + time_unit*], are able to boost the speed of blockchain convergence.

Fig.7 explores the dependency between the length of blockchain and the heterogeneous latency groups. It can be observed that the blockchain length for all heterogeneous groups are shrunk until the network latency reaches 2400 seconds, and then it further grows with a larger latency. The

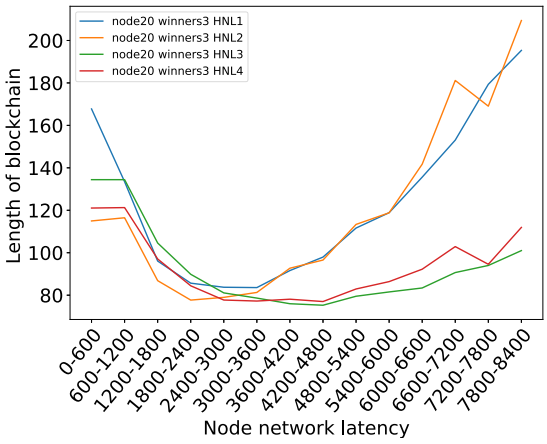

Fig. 7. The length of blockchain versus various heterogeneous latency group configurations

reason behind this is identical to the homogeneous latency simulation as shown in Fig.3. In addition to that, most of time heterogeneous group HNL3 and HNL4 obtain a shorter blockchain than HNL1 and HNL2. This is because there are some more active nodes that speed up the convergence speed, as shown in Fig.6, so that fewer blocks are generated for the simulations of HNL3 and HNL4, which in turn leads to a shorter blockchain for these two heterogeneous groups.

From the above experimental statistics, the following summarises the unfair advantages that could be obtained by nodes with low network latency:

1) Nodes with network latency can focus most of their PoW mining work on the longest chain, thus having a much higher chance to earn blockchain mining rewards.
2) The blockchain is updated much faster for the low latency nodes. Hence, their transactions have a higher chance to be well protected.

VII. Conclusion

In this paper, we investigate the impact of a wide range of network latency configurations on blockchain security. We define a notion named *global block convergence* to quantify blockchain security within our simulation results. We mainly analyse the speed of block convergence and determine how the safety of the six blocks confirmation convention of blockchain is affected by large peer-to-peer network latencies. From the simulations, it is obvious that the time spent on block convergence is proportionally increased with the extension of network latency, however, there is no clear dependency for the block convergence speed with either the network population size or the mining difficulty. In addition, the safety of the six confirmations convention of blockchain for transaction commitment is highly sensitive to both the difficulty of PoW mining and the peer-to-peer latency.

Finally, we show that variance in network latency is an important factor to monitor: we quantify the extent to which nodes that consistently experience lower network latency gain a significant unfair advantage from PoW mining—in some cases low-latency simulation groups dominate more than half of the blocks on the globally longest blockchain.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 03 2009.

[2] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer, "Decentralization in Bitcoin and Ethereum networks," *CoRR*, vol. abs/1801.03998, 2018. [Online]. Available: http://arxiv.org/abs/1801.03998

[3] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *IEEE P2P 2013 Proceedings*, Sep. 2013, pp. 1–10.

[4] C. Michael, P. Pattanayak, S. Verma, V. Kalyanaraman *et al.*, "Blockchain technology: Beyond bitcoin," *Applied Innovation Review*, vol. 2, 2016.

[5] S. K. Lo, X. Xu, Y. Chiam, and Q. Lu, "Evaluating suitability of applying blockchain," pp. 158–161, 11 2017.

[6] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 839–858.

[7] N. Fabiano, "Internet of Things and blockchain: Legal issues and privacy. the challenge for a privacy standard," in *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on*. IEEE, 2017, pp. 727–734.

[8] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A distributed blockchain based vehicular network architecture in smart city," *Journal of Information Processing Systems*, vol. 13, no. 1, p. 84, 2017.

[9] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[10] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

[11] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 104–121.

[12] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3416–3452, Fourthquarter 2018.

[13] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 279–296. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kogias

[14] M. Rosenfeld, "Analysis of hashrate-based double spending," *CoRR*, vol. abs/1402.2009, 2014. [Online]. Available: http://arxiv.org/abs/1402.2009

[15] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186. [Online]. Available: http://dl.acm.org/citation.cfm?id=296806.296824

[16] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[17] S. Zoican, M. Vochin, R. Zoican, and D. Galatchi, "Blockchain and consensus algorithms in Internet of Things," in *2018 International Symposium on Electronics and Telecommunications (ISETC)*, Nov 2018, pp. 1–4.

[18] Blockchain improvement protocol. [Online]. Available: https://github.com/bitcoin/bips/blob/master/README.mediawiki

[19] G. Andresen. Pay to script hash. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki

[20] P. McCorry, E. Heilman, and A. K. Miller, "Atomically trading with Roger: Gambling on the success of a hardfork," in *IACR Cryptology ePrint Archive*, 2017.

[21] G. Andresen. Block v2, height in coinbase. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki