Internet of Things 8 (2019) 100109

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Review article

Capacity of blockchain based Internet-of-Things: Testbed and analysis

Xu Wang[a,*], Guangsheng Yu[a], Xuan Zha[a,b], Wei Ni[c], Ren Ping Liu[a], Y. Jay Guo[a], Kangfeng Zheng[d], Xinxin Niu[d,e]

[a] Global Big Data Technologies Centre, University of Technology Sydney, Australia
[b] China Academy of Information and Communications Technology, Beijing, China
[c] Data61, Commonwealth Scientific and Industrial Research Organisation, Australia
[d] School of Cyberspace Security, Beijing University of Posts and Telecommunications, China
[e] Data Security Center, State Key Laboratory of Public Big Data, China

ARTICLE INFO

Article history:
Received 5 March 2019
Revised 6 September 2019
Accepted 6 September 2019
Available online 09 September 2019

Keywords:
Blockchain
IoT
Blockchain capacity
Markov process

ABSTRACT

An integration of Internet-of-Things (IoT) and blockchain becomes increasingly important to secure IoT data in an anti-tampering manner. Challenges arise from the immense scale of IoT and the resultant impact of network partitioning on blockchain. We design a new testbed to evaluate the impact, where resource-limited IoT devices, acting as light nodes, are an integral part of a blockchain. Our testbed is built on the Ethereum platform with non-trivial modifications on key modules. The partitioning of IoT is emulated by probabilistically dropping blocks travelling among the miners. We also propose a new discrete-time Markov chain model to validate our testbed and analyze the impact of block mining rates and network conditions on the capacity of public blockchains. The model is first formed to be non-ergodic with an infinite state space. By exploiting the eventual consistency property of blockchain, the model is collapsed to be ergodic and approximated with a finite state space and significantly improved tractability. Both the testbed and analysis reveal the blockchain capacity can be improved by accelerating the block mining rates which, however, increases stale blocks. Our analysis provides an asymptotic upper bound for the blockchain capacity.

1. Introduction

The Internet of Things (IoT) is set to ubiquitously connect a huge number of devices (embedded with sensors and actuators) to the Internet, digitizing the physical world into computer-based data systems [1–3]. Data integrity is vulnerable in IoT (e.g., to tampering), given the sheer scale of IoT networks, non-homogeneous network structure, limited device computing power, and the immense volume of data generated across the networks [4–6].

Being a distributed, incorruptible and tamper-resistant ledger database, blockchain has the potential to address the critical security issues of IoT, particularly on data integrity and reliability [7,8]. Interests in applying blockchain to IoT networks have already emerged in academia and industry, with the goal of providing security [9–11].

* Corresponding author.
E-mail addresses: Xu.Wang-1@uts.edu.au (X. Wang), Guangsheng.Yu@uts.edu.au (G. Yu), Xuan.Zha@student.uts.edu.au (X. Zha), Wei.Ni@data61.csiro.au (W. Ni), RenPing.Liu@uts.edu.au (R.P. Liu), Jay.Guo@uts.edu.au (Y.J. Guo), kfzheng@bupt.edu.cn (K. Zheng), xxniu@bupt.edu.cn (X. Niu).

https://doi.org/10.1016/j.iot.2019.100109
2542-6605/

1.1. Problem Statement

The capacity of a blockchain, defined as the average number of transactions successfully recorded per second in a final accepted blockchain [12], is restricted by the consistency requirement of blockchains, such as eventual consistency (across all chains maintained at different miners). As reported in [11] and [13], a well-known problem of applying public Proof-of-Work (PoW) blockchain into IoT applications is the limited blockchain capacity which can hardly meet the rapid growth and expansion of the IoT. Apart from the PoW consensus protocol, new consensus protocols, such as IOTA [11] and Practical Byzantine Fault Tolerance (PBFT) [14], have been proposed to improve the capacity of blockchain through centralized coordination. This, however, would be at the expense of significantly increased requirement of trust in the coordinators (which provide an audit to the distributed ledger) [15], and increased susceptibility to single node failures [16]. In this sense, PoW blockchains still have the merits of low trust requirement on the miners and strong tolerance to node failures. The PoW blockchains are important to many IoT applications, where trust cannot be established between parties.

In PoW blockchains, the generations of blocks need to be slowed down, and the generations also need to take place at different time, to avoid inconsistent records at different miners. This penalizes capacity though. The final blockchain can address inconsistency by keeping one of the conflicting blocks and discarding the others. Unfortunately, this would lead to a further loss of capacity [17]. Moreover, the consensus would be increasingly violated, as the demand for capacity grows, as qualitatively studied in [18,19]. The in-depth understanding of the interaction between capacity and consistency in PoW based blockchains has yet to be developed.

Although blockchain-based IoT applications have been studied [20–22], their capacities have not been numerically measured over unreliable IoT networks. Markov models have been developed to study the capacity and consistency of public PoW blockchains in partitioned/asynchronous networks [23–26]. However, the models are designed for the PoW blockchains for cryptocurrency and fail to analyze the blockchain for IoT applications, which requires high capacity. To be specific, the models only capture the blockchain growth between two players connected by a link with a fixed propagation delay where no more than one block can be mined by different miners simultaneously. As a result, a testbed of IoT blockchain, together with a blockchain growth model, is important to qualify the blockchain capacity for IoT applications.

1.2. Our Contributions

This paper analyzes the connection between the capacity and the consistency requirement of blockchain, under different distributions of block mining rates across the network. A new testbed platform is designed to evaluate the tolerance of IoT blockchains to network partitioning and unreliable network links, where the resource-limited IoT devices, acting as light nodes, are an integral part of the blockchain. Our testbed is built on the open-source Ethereum blockchain platform, with non-trivial modifications on the *Core, Miner, Consensus*, and *Eth* modules. The potential partitioning of IoT is emulated by probabilistically dropping the blocks travelling among the miners in the application layer of the testbed.

We also propose a new discrete-time Markov chain (DTMC) model which captures the impact of unreliable network links on the capacity and quantitatively analyzes the PoW based blockchain capacity. The model is first formed to be non-ergodic with an infinite state space. By exploiting the eventual consistency nature of blockchain, we propose to collapse the states of the model to be ergodic, and further approximate the ergodic model with a finite state space and significantly improved tractability.

The key contributions of this paper can be summarized as follows:

1. A new testbed is designed and developed to evaluate the tolerance of IoT blockchains to network partitioning and unreliable network links, where resource-limited IoT devices, acting as light nodes, are an integral part of the blockchain.
2. A new DTMC model is proposed to evaluate the impact of unreliable network links on the capacity of public PoW blockchains and analyze the capacity.
3. By exploiting the eventual consistency nature of blockchain, our DTMC model is collapsed to be ergodic and approximated with a finite state space, to significantly improve analytical tractability. The analysis from the ergodic model with a finite state space provides an upper bound for the blockchain capacity and becomes asymptotically accurate with the growing state space.

Both the experimental testbed and analysis reveal the strong impact of the mining rates and network conditions on the capacity of the blockchain. Our analysis provides an upper bound for the blockchain capacity and becomes asymptotically accurate with the increasing scale of the finite Markov model. The numerical and experimental results indicate that the blockchain capacity can be improved by accelerating the block mining rates of miners which however increases the stale block rate. Reducing the partition probabilities of the miners helps increase the blockchain capacity and reduce the stale block rate.

1.3. Organization and notations

The rest of the paper is organized as follows. In Section 2, the related works are surveyed, followed by design and testbed of Built-in IoT blockchain in Section 3. In Section 4, the eventual consistency blockchain capacity model is presented. The

Table 1
Abbreviations.

Abbreviation	Full form
API	Application Programming Interface
BaaS	Blockchain as a Service
CAP	Consistency, Availability and Partition tolerance
DTMC	Discrete-time Markov Chain
IoT	Internet-of-Things
LAN	Local Area Network
LES	Light Ethereum Subprotocol
NFC	Near-Field Communication
P2P	Peer-to-Peer
PBFT	Practical Byzantine Fault Tolerance
PoW	Proof-of-Work
SPV	Simplified Payment Verification
UTXO	Unspent Transaction Output

Table 2
Notations and definitions.

Notation	Description
n	The number of miners
W	The duration of a timeslot
δ	The growth rate of the blockchain at each timeslot
ζ	The number of transactions per block
R	Blockchain capacity
O_r	The ratio of stale blocks
μ	The size of a block header
H	The storage growth rate of light nodes
T_b (T_e)	The timestamp at the beginning (end) of the observation
N_b (N_e)	The number of blocks at the beginning (end) of observation
c_i	The successful block mining rate of the ith miner
c	The successful block mining rate of the miners
a_i	The non-partition probability of the ith miner
α	The non-partition probability of the miners
$\boldsymbol{v}$	The partition status of the blockchain network
$p_{\boldsymbol{v}}$	The probability at which the network is in the state $\boldsymbol{v}$
$\boldsymbol{l}$	Blockchain network state
$\boldsymbol{u}$	Collapsed blockchain network state
$\boldsymbol{\pi}$	The steady-state distribution
ω_u	The expected blockchain capacity at the state $\boldsymbol{u}$

numerical results of the proposed model and experimental results are presented in Section 5, followed by conclusions in Section 6.

Abbreviations and notations used in this paper are collated in Tables 1 and 2, respectively.

2. Related work

There are two popular designs of blockchain for IoT, namely, "*Built-in blockchain*" and "*Blockchain as a service*", as illustrated on the right-hand side (RHS) of Fig. 1.

In the case of Blockchain as a service (BaaS), the IoT devices, such as Near-Field Communication (NFC) tags and wireless sensors, collect sensory data and feed into a blockchain through blockchain agents [20,21]. The IoT devices are not part of the blockchain. The agents can translate the sensory data as transactions and broadcast them into the blockchain network [20]. The agents can also secure the transactions using their own private keys, while the IoT devices do not have the keys. The redundancy of the sensory data can be suppressed by using traditional aggregation algorithms [27] at the agents. However, the agents are susceptible to single-point failure. As the proxy between the IoT devices and the blockchain, the agents can carry out the man-in-the-middle attacks, e.g., injection, tampering and forging.

In the case of Built-in blockchain, all IoT devices can operate as blockchain nodes and become part of a blockchain network which consists of three types of nodes, namely, miner, full node and light node. The miners are typically fixed servers and workstations which mine transactions into blocks and record all blocks. The full nodes record all the complete blocks, including block headers and payload, but do not mine blocks. Powerful IoT devices, such as smart vehicles, can act as full nodes. Other IoT devices, such as mobile devices, can join the blockchain network by operating as light nodes.

Exploiting a simplified payment verification (SPV) technology [22], the light nodes can verify transactions without mining blocks or storing complete blocks. For example, the Ethereum SPV nodes have been deployed on smart bicycles, where Ethereum is a programmable blockchain platform [28]. A light node only needs to keep the chained block headers and the

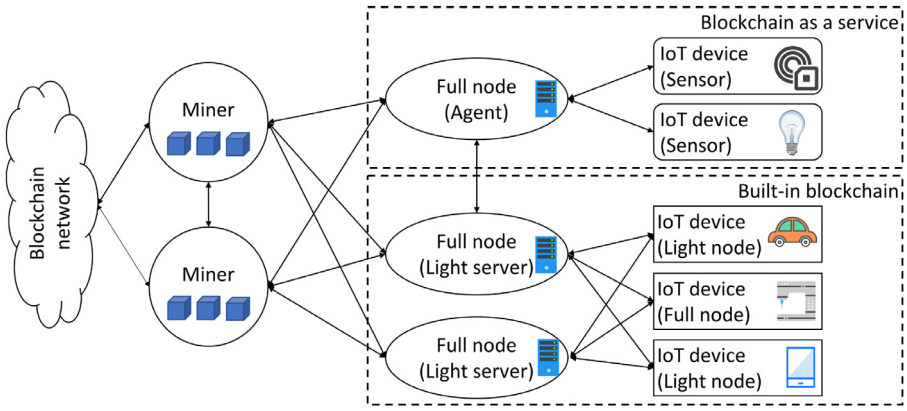


Fig. 1. A general architecture of blockchain-based IoT. The low-power devices can use the blockchain service with the help of agents. The powerful IoT devices can operate as blockchain nodes and join the blockchain network.

Merkle branch linking to the transaction to be verified. Although the node cannot validate the transaction, it can check whether the blockchain network has accepted the transaction by comparing the Merkle branch linking to the transaction [29]. The light nodes need to fetch blockchain data from full nodes or miners, e.g., a light server in Ethereum [29].

All the IoT devices of a *Built-in blockchain* solution have their own blockchain accounts. They can sign transactions containing sensory data, transfer tokens, and verify transactional messages. As a result, the raw data from the IoT devices can be secured by the private keys (from their blockchain accounts) and permanently recorded in blockchain. They can connect to multiple light servers to avoid the man-in-the-middle attack and single-point failure.

The blockchain capacity is currently restrained by consistency, and would be penalized by non-ideal system characteristics of IoT networks, such as incomplete network connectivity and inherent network partitioning. The well-known CAP theorem asserts that *distributed systems cannot simultaneously provide three vital guarantees, i.e., Consistency, Availability and Partition-tolerance* [18]. Further, the PACELC theorem indicates that *if there is a Partitioning, the distributed system has to give up either of Availability and Consistency; but Else, if the network is connected, the system has to trade off between Latency and Consistency* [19]. Both the CAP and PACELC theorems are particularly applicable to IoT blockchains which are prone to partition and can incur availability (or capacity) losses and delays for achieving consistency. However, the theorems are qualitative, and do not quantify the trade-offs between capacity and consistency [30]. Many existing blockchain approaches [22,31,32] have given away substantial capacity for the purpose of preventing inconsistency.

Markov models have been developed to study the capacity and consistency of public PoW blockchains in partitioning/asynchronous networks [23–26]. They only analyze blockchain growth between two players, i.e., an adversarial player and an honest one, and thus cannot be applied to analyze the general blockchain growth among multiple independent miners. Another limitation of the existing Markov models is the assumption that no more than one block can be mined across the entire network at any moment [25,26]. The assumption is only valid in the case of low block mining rates (or low blockchain capacity) and can hardly be applicable in IoT blockchains with high blockchain capacity. Meanwhile, models [23,25,26] only capture a constant delay during block propagation and fail to characterize uncertain partitioning in IoT networks. In this paper, the proposed DTMC model is general and can qualify the blockchain capacity across multiple miners operating different block mining rates and connected by unreliable links.

3. Testbed design for built-in IoT blockchain

In this section, we design and implement a new testbed of IoT blockchain to evaluate the impact of the physical network conditions on the capacity and consistency of IoT blockchain. Our testbed is a *Built-in blockchain*, where the IoT devices are designed to be light nodes.

Ethereum is selected as the development platform of our testbed for the following reasons.

1. Ethereum is one of the most popular public blockchain platforms, only next to Bitcoin in terms of their market capitalization [33]. However, Bitcoin is limited to the applications of cryptocurrency adopting the unspent transaction output (UTXO) transaction format and does not provide the flexibility and versatility to support IoT applications.
2. Ethereum allows the light IoT devices to join the blockchain in the form of light nodes and therefore ensures end-to-end trust in IoT applications by employing the SPV technology [22].
3. The official open-source implementation of the Ethereum protocol, i.e., go-ethereum, can be compiled for various platforms, such as 64-bit macOS, 64-bit Linux and Raspbian in our testbed [34].
4. As an open-source public blockchain platform, Ethereum provides substantial research and development opportunities. It has been widely adopted in the literature, especially for IoT applications [35–37].

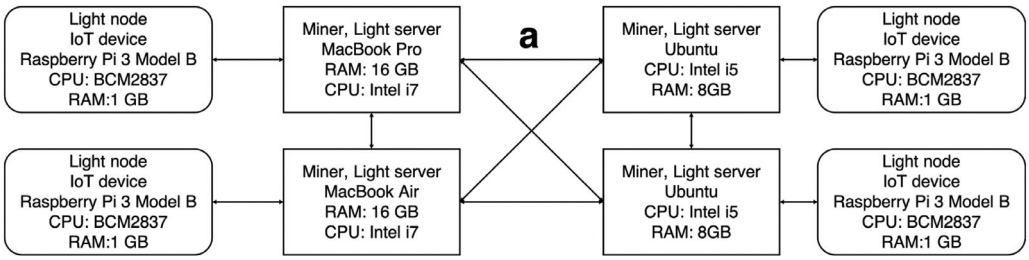


Fig. 2. The topology of our testbed, where four miners are connected/disconnected in every timeslot according to predefined non-partition probabilities **a**. The IoT devices, operating as blockchain light nodes, uniformly connect the miners (light servers).

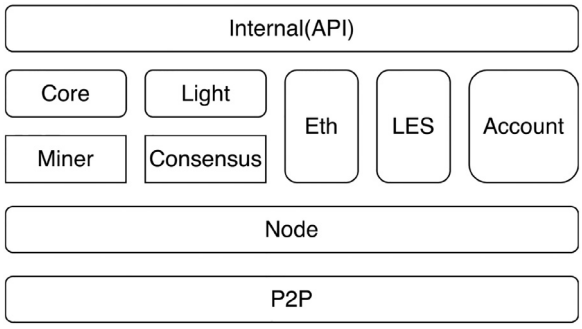


Fig. 3. The structure of key modules in Ethereum.

As a distributed ledger, Ethereum is used to record IoT sensory data as transactions in our testbed. Other Ethereum features, such as smart contract and token, can also be important to some IoT applications. Nevertheless, they do not affect the blockchain capacity and, therefore, are not considered nor implemented in our testbed.

Non-trivial efforts have been spent customizing the Ethereum to develop our testbed with the purpose of understanding the connection between the blockchain capacity, the distribution of block mining rates, and the network partitioning. Following software changes have been made to Ethereum in the testbed.

1. The block mining rate of every miner is carefully calibrated by tuning its CPU time inside the customized *Miner* module and adjusting the difficulty criteria.
2. Two new software functions, namely, "stopSync" and "startSync", are developed to emulate unstable network connectivity (or in other words, network partitioning) which prevails in practical IoT platforms.
3. We replace the incumbent "greedy heaviest observed subtree" principle of Ethereum by developing the "longest chain" principle which has been dominating the designs of consensus protocols in blockchains, e.g., Bitcoin [22] and its variations [31,38].

More details are provided in the rest of this section.

3.1. Testbed design

Fig. 2 shows the architecture of our new testbed, which consists of eight devices, including $n = 4$ miners and 4 IoT devices. The eight devices run three different operating systems: macOS, Ubuntu and Raspbian. All the eight devices join the same wireless LAN. The miners are a MacBook Pro, a MacBook Air and two desktop PCs, and they mine transactions into blocks in parallel. They also act as light servers to connect the light nodes – the IoT devices. The MacBook Pro runs macOS 10.12.6 on Intel i7 CPU and 16 GB memory. The MacBook Air runs macOS 10.12.6 on Intel i7 CPU and 8 GB memory. The desktop PCs have Intel i5 CPU and 8 GB memory and run Ubuntu 16.04. Four IoT devices are implemented by Raspberry Pi devices. They send transactions recording the temperature and humidity to the miners. Four Raspberry Pi 3 Model B devices run the Raspbian system with BCM2837 CPU and 1 GB RAM.

Most existing blockchain technologies have focused on the service layer with no consideration on physical restrictions of the network, devices, or bandwidth [20,39]. In practice, the links between the miners can be unreliable with errors and delays [40]. The miners can temporarily lose connection and be partitioned from others. Let $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ denote the predefined non-partition probabilities of the n miners. The i-th miner connects the blockchain network with probability a_i at any timeslot. To emulate this, we propose to disconnect the miners from each other in the blockchain (i.e., the application layer), according to the predefined non-partition probabilities.

Our testbed is a *Built-in blockchain* based on Ethereum. Fig. 3 shows the key modules of Ethereum, as follows.

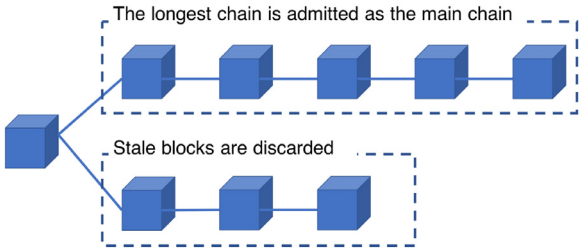


Fig. 4. The longest chain accounts for the greatest PoW effort and is admitted as the main chain. The other conflicting blocks are discarded.

```
geth --identity "miner" --networkid 16 --port 25052 --nodiscover --
syncmode "full" --lightserv 25 --lightpeers 1
```

(a) Light Server.

```
geth --identity "light" --networkid 16 --port 25052 --nodiscover --
syncmode "light"
```

(b) Light Node.

Fig. 5. Start command.

- The *P2P* module implements a kademlia-like P2P protocol and supports encryption and authentication [41].
- The *Node* module defines the structure and behaviours of an Ethereum node. Every node instance is a vertex in the P2P network.
- The *Account* module implements account-related functions, e.g., account creation and signature.
- The *Miner* module generates new blocks and outputs the blocks to the *Consensus* module.
- The *Consensus* module decides the admitted blocks and exports the result back to the *Miner* module.
- The *Core* and *Light* modules are enabled in full nodes and light nodes, respectively. They validate blocks and block headers from the *Miner* module and decide whether a chain-reorganization should be applied.
- The *Eth* and *LES* modules control incoming and outgoing messages based on pre-defined protocols in full nodes. Light nodes only used the *LES* module.
- The *Internal* module configures public and private Application Programming Interfaces (APIs) for users.

Particularly interested in the impact of network partitioning on IoT blockchain, the *Core, Miner, Consensus, Eth* and *Internal* modules are updated in our new testbed.

The PoW block mining algorithm is implemented at the *Miner* and *Consensus* modules. The miners generate blocks independently by solving the PoW problems which are formulated locally based on the local chains of the miners, the transactions to be mined, and the required difficulty level. The miners can try different nonces in the block headers during mining, until the hashes of the headers meet predefined difficulty criteria. The successful block mining rate of the ith miner, denoted by c_i, depends on the computing resource of the ith miner and the predefined criteria. In our testbed, we adjust c_i, by tuning the CPU time of each miner inside the customized *Miner* module and calibrating the difficulty criteria when the testbed is reset.

Different blocks at the same height of the blockchain cause forks of the chain, even in the case where the blocks record the same transactions. This is because the blocks are based on different PoW problems, and have different PoW solutions and different miners' signatures. The longest chain principle is applied to decide on local chains and eliminate forks at the *Core* module of every miner node, shown in Fig. 4. The principle specifies that each miner admits the longest chain to the best of the miner's knowledge, and discards the rest. The longest chain principle has been practically adopted in Bitcoin [22] and its variations [31,38], and is reasonable since the longest chain typically accounts for the greatest PoW effort and investment of the system.

New APIs, namely, "stopSync" and "startSync", are added to the *Internal* module and implemented in the *Eth* module to simulate the partition and recovery of the miners. If the "stopSync" command is called, the incoming and outgoing blocks of a miner are stored in a cache. The cached blocks are delivered to the processing queue, once the "startSync" command is called.

In the new testbed, four miners, acting as light servers simultaneously, and four light nodes run the same customized source code on different platforms. By adopting the docker technology [34], the customized source code is compiled for compatible executable programs on different operating systems, namely, macOS, Ubuntu and Raspbian, in the testbed. The nodes play different roles according to their start commands [42], as shown in Fig. 5. Specifically, the light service function is activated by adding the "lightserv" and "lightpeers" parameters to the start command, as shown in Fig. 5(a). The miner function is activated by calling the "miner.start" API after a blockchain node is started. The node acts as a light node by

Block Number	1
Block hash	0x3be8c42d4a236bebb6370e7811c74abe5a1f99c5f8a0c9e8d8237489fafc507c
TimeStamp	1526789283
Miner	0xa128a4b2375ab09245227938ed19b6c88ecf536d
Gas Limit	7992189
Size	537

(a) The block header of the first block.

Block Number	307
Block hash	0xf34d9227e6b46f7615e204098782e52c6dea0f5197111a18354591003207a3fc
TimeStamp	1526799231
Miner	0x1895fcf51d64bb64716879ee3a61f4ff503d5489
Gas Limit	5927232
Size	539

(b) The block header of the 307th block after 10^3 timeslots.

Fig. 6. The first block header and the last block header of the blockchain-based IoT in 10^4 s (10^3 timeslots). The uniform block mining rates of the four miners are 0.25 blocks per timeslot. The non-partition probabilities of the miners are 0.1.

adding the "syncmode light" parameter to the start command, as shown in Fig. 5(a). The links of the testbed are established by calling the "addPeer" API [36], according to the topology specified in Fig. 2.

During our test, the Raspberry Pis first interpret the physical world into digital data, including temperature data and humidity data, with Raspberry Pi Sense HATs [43]. The Raspberry Pis then send the sensory data to the light servers in the form of blockchain transactions, where the "input" fields of the transactions are filled with the sensory data. After receiving the transactions, the light servers, also acting as the blockchain miners, mine the transactions into blocks independently by running the PoW block mining algorithm. The miners share the mined blocks and solve inconsistency if more than one block is mined by different miners simultaneously, by adopting the longest chain rule. The miners push the latest blocks to the IoT devices by taking the role of light servers. The IoT devices can verify their received blocks and the data embedded in the blocks, as the blockchain light nodes.

Apart from environmental sensing, the proposed blockchain based IoT testbed can be applied in various IoT applications, e.g., smart city, industrial IoT, and healthcare applications [3,44,45]. Take the IoT healthcare application [45] for an example. The IoT gateways can join the blockchain network as the light nodes. They collect the health data of patients and upload the data to the blockchain in the form of transactions. The miners can be run by multiple hospitals and serve as a trustworthy database on top of the blockchain. As a result, health data stored on the blockchain are trustworthy to all blockchain participants, e.g., different hospitals.

3.2. Experimental results

In the experiment, we configure the testbed to run on the basis of $W = 10$ s per slot for 10^4 seconds, i.e., 10^3 timeslots in total. During each timeslot, the synchronization and mining take place in sequel at the miners. This is due to the fact that a blockchain starts with the genesis block at one miner and proceeds by synchronizing with other miners before any mining starts [12]. We configure the *Miner* and *Consensus* modules to allow the testbed to mine on average one block per timeslot. The four miners, configured to have the same block mining rate c, generate blocks in parallel, i.e., $c = 0.25$. In other words, every miner needs 4 timeslots (40 s) on average to generate a block. Their non-partition probabilities are set to be $a_i = \alpha = 0.1$. The Raspberry Pi devices connect to the miners and upload transactions to the blockchain. In the presence of non-negligible network partitioning, the capacity of the blockchain can be evaluated based on the first and last blocks, shown in Fig. 6. The block mining time is recorded in the Unix timestamp format [46], as shown in the *TimeStamp* field.

The growth rate of the blockchain for each timeslot W, denoted by δ, can be obtained by counting the number of blocks generated within a time window $[T_b, T_e)$ and evaluating the average number for a timeslot W, i.e.,

$$\begin{aligned}\delta &= \frac{N_e - N_b}{T_e - T_b} \times W \\ &= \frac{307 - 1}{1526799231 - 1526789283} \times 10 \\ &= 0.308, \end{aligned} \tag{1}$$

where T_b and T_e are the timestamps at the beginning and the end of the observation, respectively; N_b and N_e are the numbers of blocks in the blockchain at the beginning and the end of the observation, respectively. The experimental results are presented in Fig. 6, where 306 blocks have been generated within 9948 s (or 994 slots).

Block Number	582
Block hash	0xa9d1f39c8aba93b301f408de7300df071f105072f84d52ec1d1b179ae8d0801d
TimeStamp	1526877105
Miner	0x03c54ef759bab0539f23dfe83a2c8f707f7fa484
Gas Limit	7384182
Size	36661
Transactions	351

Fig. 7. A block with the gas limit of 7384182 stores 351 transactions.

Under this setting, the main chain achieves the growth rate of 0.308 blocks per timeslot, which is far less than the total block mining rate of all the miners (1 block per timeslot).

The experiment also shows that a single block has a limited size and can store finite transactions. Fig. 7 shows the header of the 582nd block in our test chain, where the IoT devices send a large number of transactions to fill in the block. We can see that the gas limit of this block is 7,384,182, where the gas measures how much "work" an action takes to perform in Ethereum [47]. As a result, the block stores $\frac{7384182}{21000} = 351$ transactions, where 21,000 gas is the minimum cost of an Ethereum transaction [47]. Another example is that a Bitcoin block is no larger than 1 MB with the minimum transaction of 166 bytes [48].

The capacity of our testbed, denoted by R, is defined as the number of transactions added into the eventually consistent blockchain ledger per timeslot. The blockchain capacity determines the scale of IoT applications and can be given by

$$R = \delta \times \zeta, \tag{2}$$

where δ is the growth rate of the blockchain, and ζ is the constant number of transactions per block. Without loss of generality, we normalize $\zeta = 1$. (In our testbed, $\zeta = 351$ transactions per block, as shown in Fig. 7.) Therefore, the capacity of the blockchain can be evaluated as the growth rate of the main chain.

Discarded blocks, known as stale blocks, are detrimental to the security and performance of a blockchain [24]. With the blockchain growth rate δ, the ratio of stale blocks, denoted by O_r, is given by

$$O_r = \frac{\sum_{i=1}^{n} c_i - \delta}{\sum_{i=1}^{n} c_i}. \tag{3}$$

The blockchain growth rate determines the storage requirement of light nodes running by IoT devices. The size of a block header, without the transactions, is invariable with a fixed structure. The storage growth rate of light nodes, denoted by H, can be given by

$$H = \mu\delta, \tag{4}$$

where μ is the size of a block header. For example, μ is around 80 bytes for a Bitcoin block or 500 bytes in the case of Ethereum blockchain.

Besides storage, IoT devices are sensitive to resource consumption. H can be used to evaluate the lowest communication, computation and energy cost. This is based on the fact that the block headers are the minimum requirement of blockchain nodes. The block headers are fetched from light servers in a wireless fashion and verified by data checking and hash operations.

4. Analysis of Blockchain capacity in the presence of partitioning

We propose a DTMC model to analyze the growth of a public blockchain among n miners connected with unreliable links, e.g., the miners belonging to different LANs which are geopolitically apart. In public blockchains, blocks are independently generated by decentralized miners and propagated across the blockchain network with unreliable network links. The blocks at the same height are different blocks, and a single block among them can be eventually kept. During every timeslot, miners work on the synchronization and mining as described in Section 3.1. At the end of every timeslot, the state of the DTMC model is observed and recorded in the DTMC model. Let $\boldsymbol{l} = [l_1, l_2, \ldots, l_n]$ denote a state of the blockchain network. $l_i \geq 0$, is the length of the local chain at the ith miner. For example, the states of three miners can be visualized as a three-dimensional integer grid with non-negative elements, as illustrated in Fig. 8.

At the beginning of a time slot, a miner broadcasts its own chain (i.e., the longest chain to the knowledge of the miner) to all the other non-partitioned miners if it is not partitioned. If multiple chains of different lengths are observed, a miner keeps the longest and disposes of the others. If a miner is separated due to network partitioning, it may not receive the longer chains and therefore sticks to its local chain. After this, the miner independently mines a new block in attempt to extend the chain it keeps. The proposed DTMC can, therefore, be constructed by cascading these two phases at each timeslot. The transition probability from $\boldsymbol{l}$ to $\boldsymbol{l}'$ per timeslot is given by

$$\Pr\{\boldsymbol{l}'|\boldsymbol{l}\} = \sum_{\bar{\boldsymbol{l}}} \Pr\{\bar{\boldsymbol{l}} \big| \boldsymbol{l}, \mathcal{S}\} \Pr\{\boldsymbol{l}' \big| \bar{\boldsymbol{l}}, \mathcal{M}\}, \tag{5}$$

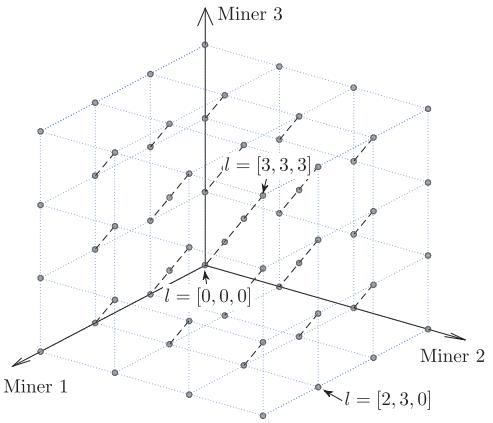


Fig. 8. The state space of the proposed DTMC model, where three miners are considered. For example, $\boldsymbol{l} = [3, 3, 3]$ denotes that the local chains of the three miners have three blocks.

where $\Pr\{\bar{\boldsymbol{l}}\big|\boldsymbol{l}, \mathcal{S}\}$ is the transition probability from $\boldsymbol{l}$ to $\bar{\boldsymbol{l}}$, resulting from synchronization, indicated by $\mathcal{S}$; $\Pr\{\boldsymbol{l}'\big|\bar{\boldsymbol{l}}, \mathcal{M}\}$ is the transition probability from $\bar{\boldsymbol{l}}$ to $\boldsymbol{l}'$, resulting from block mining of the individual miners, indicated by $\mathcal{M}$.

Let $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$ denote the partition state of the blockchain network. $v_i = 1$ denotes that the ith miner is not partitioned, and $v_i = 0$ denotes that the ith miner is partitioned. The probability that the blockchain network is in the partition state $\boldsymbol{v}$, denoted by $p_{\boldsymbol{v}}$, can be given by

$$p_{\boldsymbol{v}} = \prod_{\substack{i=1 \\ v_i=1}}^{n} a_i \prod_{\substack{i=1 \\ v_i=0}}^{n} (1 - a_i). \tag{6}$$

This is because every miner is independently partitioned according to its network connection.

With a given blockchain state $\boldsymbol{l}$ and a partition state $\boldsymbol{v}$, the blockchain state after synchronization, denoted by $\boldsymbol{l}'$, can be obtained and written as

$$\boldsymbol{l}' = f(\boldsymbol{l}, \boldsymbol{v}), \tag{7}$$

where $l'_i = l_i$ if the ith miner is partitioned. $l'_i = \hat{l}$ if the ith miner is not partitioned, where $\hat{l}$ is the length of the longest chain from the non-partitioned miners.

Note that $\boldsymbol{l}$ can transit to $\boldsymbol{l}'$ with different partition states of the blockchain network. Take the blockchain with three miners for an example, $\boldsymbol{l} = [3, 3, 0]$ can transit to $\boldsymbol{l}' = [3, 3, 3]$ with $\boldsymbol{v} = [1, 0, 1]$ or $\boldsymbol{v}' = [0, 1, 1]$, where the third miner can obtain the longest chain from either the first miner or the second miner, respectively. As a result, $\Pr\{\boldsymbol{l}'\big|\boldsymbol{l}, \mathcal{S}\}$ can be obtained by summing all the partition probabilities $p_{\boldsymbol{v}}$ where $\boldsymbol{v}$ meets the condition that $\boldsymbol{l}' = f(\boldsymbol{l}, \boldsymbol{v})$. $\Pr\{\boldsymbol{l}'\big|\boldsymbol{l}, \mathcal{S}\}$ can be written as

$$\Pr\{\boldsymbol{l}'\big|\boldsymbol{l}, \mathcal{S}\} = \sum_{\substack{\boldsymbol{v} \\ \boldsymbol{l}' = f(\boldsymbol{l}, \boldsymbol{v})}} p_{\boldsymbol{v}}. \tag{8}$$

Since every miner independently generates and attaches blocks to its local chain, $\Pr\{\boldsymbol{l}'\big|\boldsymbol{l}, \mathcal{M}\}$ can be written as

$$\Pr\{\boldsymbol{l}'\big|\boldsymbol{l}, \mathcal{M}\} = \prod_{i=1}^{n} \Pr\{l'_i\big|l_i, \mathcal{M}\}, \tag{9}$$

where $\Pr\{l'_i\big|l_i, \mathcal{M}\}$ is the transition probability that the local chain of the ith miner grows from a relative length l_i to l'_i, and can be given by

$$\Pr\{l'_i\big|l_i, \mathcal{M}\} = \begin{cases} c_i, & \text{if } l'_i = l_i + 1; \\ (1 - c_i), & \text{if } l'_i = l_i; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Remark 1. The DTMC model in Fig. 8 is non-ergodic, and does not have the steady-state and cannot be efficiently analyzed through the steady-state probability (as typically done in Markov analysis). This is because the states of the DTMC model can only transit "monotonically" from state $\boldsymbol{l}$ to $\boldsymbol{l}'$ with $l'_i \geq l_i$ for $i = 1, \ldots, n$, due to the monotonic growth of the blockchains. State $\boldsymbol{l}'$ by no means can transit to state $\boldsymbol{l}$.

With a focus on *eventual consistency*, we propose to collapse the states having the same relative lengths in Fig. 8, $\boldsymbol{l}$, to a single state $\boldsymbol{u} = \boldsymbol{l} - \min(\boldsymbol{l}) \times \mathbf{1}_n$, without compromising the Markov property of the proposed DTMC model. This is due to the fact that the public blockchain, as an eventual consistency system, has its capacity grow against the longest local chain

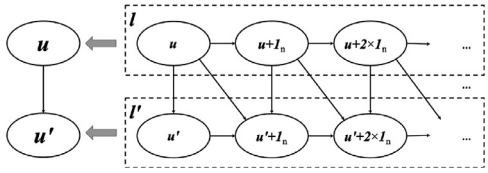


Fig. 9. An illustration of collapsing states $\boldsymbol{l} = \boldsymbol{u} + k \times \mathbf{1}_n$ for $k = 0, 1, \ldots,$ into a single state $\boldsymbol{u}$, and collapsing states $\boldsymbol{l}' = \boldsymbol{u}' + k' \times \mathbf{1}_n$ for $k' = 0, 1, \ldots,$ into a single state $\boldsymbol{u}'$.

at every moment. Here, $\min(\boldsymbol{l})$ takes the minim entry of state $\boldsymbol{l}$, and $\mathbf{1}_n$ is the n-dimensional row vector with all the entries of 1. In Fig. 8, the states, which have the same relative lengths inside a state and can be collapsed, are along the same chain of states with increment $\boldsymbol{1}_n$ between consecutive states. In other words, the states are along one of the dash lines in Fig. 8.

It is reasonable to collapse the states with the same relative lengths into one, as illustrated in Fig. 9. This is because, from (7) and (10), we can see that only the relative lengths within each of the states before and after a transition are used to evaluate the transition probability. As a result, we can conclude

$$\Pr\{\boldsymbol{u}' + k' \times \mathbf{1}_n | \boldsymbol{u} + k \times \mathbf{1}_n\} = \Pr\{\boldsymbol{u}' + (k' - k) \times \mathbf{1}_n | \boldsymbol{u}\}, \forall k, \ \ k' \in \mathbb{Z}^+, \tag{11}$$

where $\mathbb{Z}^+$ stands for positive integers, and $k' \geq k$ since the main chain grows monotonically.

Note that $k = 0, 1, 2, \ldots, \infty$ is used to enumerate all the Markov states which can be collapsed into a single state in the proposed model (so is k'). All of the states $\boldsymbol{l}(\boldsymbol{u}, k) = \boldsymbol{u} + k \times \mathbf{1}_n$ for $k = 0, 1, 2, \ldots, \infty$ can be represented by the single state $\boldsymbol{u}$. By this means, we are able to transform the originally non-ergodic Markov chain model with states of $\boldsymbol{l}(\boldsymbol{u}, k)$ $\forall \boldsymbol{u}, k$ to the ergodic Markov chain model with states of $\boldsymbol{u}$. The ergodicity of the new model allows for the evaluation of the stationary distribution of the model and, in turn, the analysis of the blockchain growth rate and the ratio of stale blocks, as will be described shortly in Remark 2.

By collapsing $\boldsymbol{l}$ to $\boldsymbol{u} = \boldsymbol{l} - \min(\boldsymbol{l}) \times \mathbf{1}_n$ and $\boldsymbol{l}'$ to $\boldsymbol{u}' = \boldsymbol{l}' - \min(\boldsymbol{l}') \times \mathbf{1}_n$, the transition probability from $\boldsymbol{u}$ to $\boldsymbol{u}'$ can be written as

$$\begin{aligned}
\Pr\{\boldsymbol{u}' | \boldsymbol{u}, \mathcal{C}\} &= \frac{\sum_{k=0}^{\infty} \left(p_{\boldsymbol{u}+k\times\mathbf{1}_n} \sum_{k'=k}^{\infty} \Pr\{\boldsymbol{u}' + k' \times \mathbf{1}_n | \boldsymbol{u} + k \times \mathbf{1}_n \} \right)}{\sum_{i=0}^{\infty} p_{\boldsymbol{u}+k\times\mathbf{1}_n}} \\
&= \frac{\sum_{k=0}^{\infty} \left(p_{\boldsymbol{u}+k\times\mathbf{1}_n} \sum_{k'=k}^{\infty} \Pr\{\boldsymbol{u}' + (k' - k) \times \mathbf{1}_n | \boldsymbol{u} \} \right)}{\sum_{k=0}^{\infty} p_{\boldsymbol{u}+k\times\mathbf{1}_n}} \\
&= \sum_{k''=0}^{\infty} \Pr\{\boldsymbol{u}' + k'' \times \mathbf{1}_n | \boldsymbol{u}\},
\end{aligned} \tag{12}$$

where $p_{\boldsymbol{u}+k\times\mathbf{1}_n}$ is the probability of state $\boldsymbol{u} + k \times \mathbf{1}_n$, and "$\mathcal{C}$" indicates that $\boldsymbol{u}$ and $\boldsymbol{u}'$ are collapsed states, and $k'' = k' - k$. The second equality of (12) is achieved by exploiting (11). $\Pr\{\boldsymbol{u}' + k'' \times \mathbf{1}_n | \boldsymbol{u}\}$ in the last equality of (12) can be evaluated with (5).

As shown in (12), $\Pr\{\boldsymbol{u}' | \boldsymbol{u}, \mathcal{C}\}$ is independent of individual states along a chain of states (which are collapsed into a single state), due to the specific characteristic given by (11), and therefore the transition probabilities between the collapsed states can be acquired. As a result, the collapsed DTMC model can be readily constructed with no ambiguity.

The remaining valid states of $\boldsymbol{u}$ are on the n number of $(n-1)$-dimensional spaces specified by $\min(\boldsymbol{u}) = u_i = 0$ for $i = 1, \ldots, n$. u_i is the ith element of $\boldsymbol{u}$. $\boldsymbol{u}_0 = [0, 0, \ldots, 0]$ stands for the last time when all the miners have the consistent chain lengths.

Remark 2. The collapsed DTMC model is ergodic. This is due to the fact that every state can transit to $\boldsymbol{u}_0$ with different probabilities based on (12), while $\boldsymbol{u}_0$ can transit to any other states. As a result, every state can transit to every other state and itself. According to [49, Theorem 16], the existence and uniqueness of the steady-state probabilities of the infinite states can be confirmed.

The capacity of the public blockchain system, denoted by R, can be given by

$$R = \boldsymbol{\pi} \cdot \boldsymbol{\omega}, \tag{13}$$

where "$\cdot$" stands for the dot product of two vectors. The u-th entry of $\boldsymbol{\pi}$ is the steady-state probability of state $\boldsymbol{u}$. The uth element of $\boldsymbol{\omega}$, denoted by ω_u, is the expected capacity at state $\boldsymbol{u}$, and can be given by

$$\omega_u = \sum_{\boldsymbol{l}} \left(\Pr\{\boldsymbol{l} | \boldsymbol{u}, \mathcal{S}\} \left(1 - \prod_{\substack{i=1 \\ l_i = \max(\boldsymbol{l})}}^{n} (1 - c_i) \right) \right), \tag{14}$$

where $\Pr\{\boldsymbol{l}\big|\boldsymbol{u}, \mathcal{S}\}$ can be evaluated by using (8). This is because the main chain increases at most one block in the blockchain pursuing eventual consistency, in the case that at least one of the miners, which has the longest local chain after synchronization, generates a block.

To evaluate $\boldsymbol{\pi}$, we can use a finite Markov model to asymptotically approximate the infinite collapsed DTMC. We assume that a block can propagate and reach all the miners within the time, during which up to $k > 0$ new blocks are admitted to the main chain. This assumption is legitimate in practice. For example, Bitcoin assumes a waiting time period of 6 blocks before a block can be confirmed and admitted in the final blockchain [22].

In the case that the maximum length difference is not reached, i.e., $\max(\boldsymbol{u}) < k$, the state transitions are the same as they are in the infinite collapsed DMTC model. In the case that the maximum length difference is reached, i.e., $\max(\boldsymbol{u}) = k$, the state transits to state $\boldsymbol{u}'$, where u_i' is given by

$$u_i' = \begin{cases} u_i - 1, & u_i > 0; \\ 0, & u_i = 0. \end{cases} \tag{15}$$

This is based on the assumption that the block generated k blocks earlier has been successfully delivered to all miners. We can see that $\max(\boldsymbol{u}') = \max(\boldsymbol{u}) - 1 = k - 1$, indicating that the state is not beyond the finite state space. The public blockchain capacity can be obtained by (13). Take $\boldsymbol{u} = [1, 3, 0]$ for example, and set $k = 4$. In the case that the second miner produces a block after an unsuccessful synchronization, i.e., transiting to [1,4,0], the system reaches the maximum length difference with $\max(1, 4, 0) = 4$. By applying (15), [1,4,0] can transit to [0,3,0]. As a result, [1,3,0] transits to [0,3,0] instead of [1,4,0], in the finite approximation of the infinite collapsed DTMC model.

Note that the proposed analytic model is explicitly designed to capture the distributed block mining process of PoW blockchains at different miners, and the resolution of inconsistency (or in other words, forks) at individual miners in the presence of network partitioning. Following are the reasons:

1. The Markov chain model is defined as such that each state collects the lengths of the chains that all the miners maintain locally to the best of their knowledge on the longest chains. The lengths of the chains, or in other words, the elements of the states, grow with the time, as blocks are increasingly mined into the chains.
2. By exploiting the eventual consistency property of blockchain, we propose to group the states based on the difference between elements (or the inconsistency of the chains at different miners). Each of the groups is collapsed into a single state focusing on the inconsistency. By doing this, the state space of the Markov chain is dramatically reduced, and the originally non-ergodic Markov chain is transformed to be ergodic, facilitating the evaluation of the stationary distribution of the states and hence the capacity of the blockchain.

For these reasons, the proposed analytic model is able to explicitly characterize and analyze the behavior of PoW blockchain, and provide effective analysis of the capacity of the blockchain.

5. Experimental and numerical validation

Validated by the proposed DTMC model, our blockchain testbed is able to effectively evaluate the impact of mining capabilities and network conditions of the miners on the blockchain capacity.

We validate the finite approximation of the proposed infinite collapsed DTMC model, where four and five miners are analyzed and simulated. The fifth miner connects with others in the same way as illustrated in Section 3.1. The miners are configured to have the same block mining rate, c. We set $c = \frac{1}{n}$ allowing that the miners can generate a block per timeslot on average. Two networks, with high and low possibilities of undergoing partition, i.e., $a_i = \alpha = 0.1, 0.5$, are simulated, and their effects are compared in the case of four miners. The experiment results in 10^3 timeslots are shown in Fig. 10. We can see that the capacity, evaluated by using the finite approximation of the proposed infinite DTMC model, converges to the simulation result (i.e., the capacity of the infinite Markov model). We also see that the approximation needs more states with the decreasing non-partition probabilities and/or the increasing number of miners. This is because the decreasing non-partition probabilities or the increasing number of miners can enlarge the length difference among the miners, and therefore require a larger value of k (i.e., more states) to capture the inconsistency.

Fig. 11 shows the effect of the non-partition probability on the blockchain capacity and the ratio of stale blocks, O_r, based on (13) and (3), where the four miners have the same block mining rate, i.e., $c = 0.1,\ 0.25,\ 0.5$, respectively. The miners are connected in a P2P fashion. We can see that R increases with the growth of the uniform non-partition probability α, while O_r declines. We can also see that R is lower bounded by the maximum capacity of miners and upper bounded by $\left(1 - \prod_{i=1}^{n}(1 - c_i)\right)$ which is the capacity of the blockchain in a partition-free network. For example, the blockchain capacity is greater than 0.1 and less than $1 - (1 - 0.1)^4 = 0.3439$ in the case of $c = 0.1$. This is consistent with the CAP theorem and indicates that our model is able to quantitatively analyze the effect of network partitions. Meanwhile, the blockchains with low block mining rates are able to maintain a low O_r. As shown in the figure, the blockchain with $c = 0.1$ has the lowest O_r, especially when the miners are strongly connected, e.g., $\alpha = 0.9$. We also see that although accelerating the block mining of miners can improve the blockchain capacity, it could result in a higher ratio of stale blocks, as O_r increases with the growth of c from 0.1 to 0.5. In contrast, increasing the non-partition probability can improve the blockchain capacity and reduce the stale blocks at the same time.

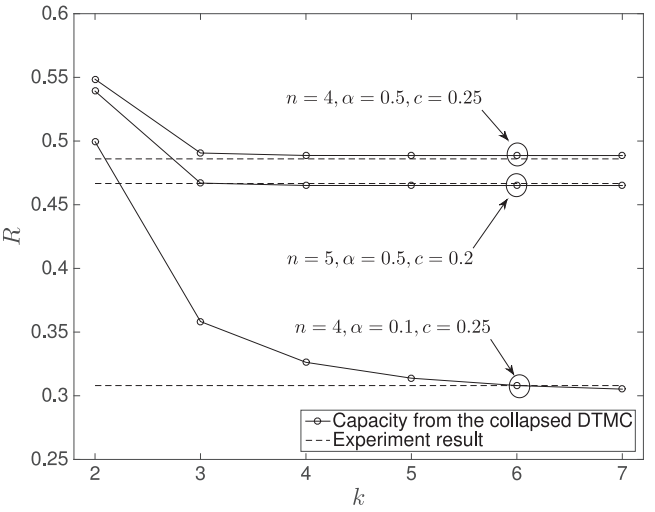
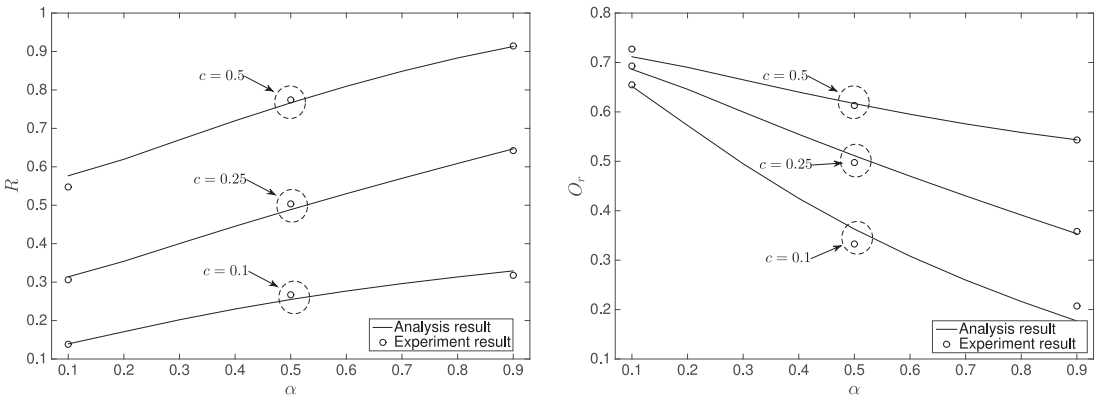


Fig. 10. The capacity of the infinite collapsed DTMC with five and four miners, where the miners are connected with each other in a P2P network. The uniform non-partition probabilities, denoted by α, and the block mining rate per miner, denoted by c, are shown in the figure. The reference lines are obtained from experiments on the testbed running 10^3 timeslots.



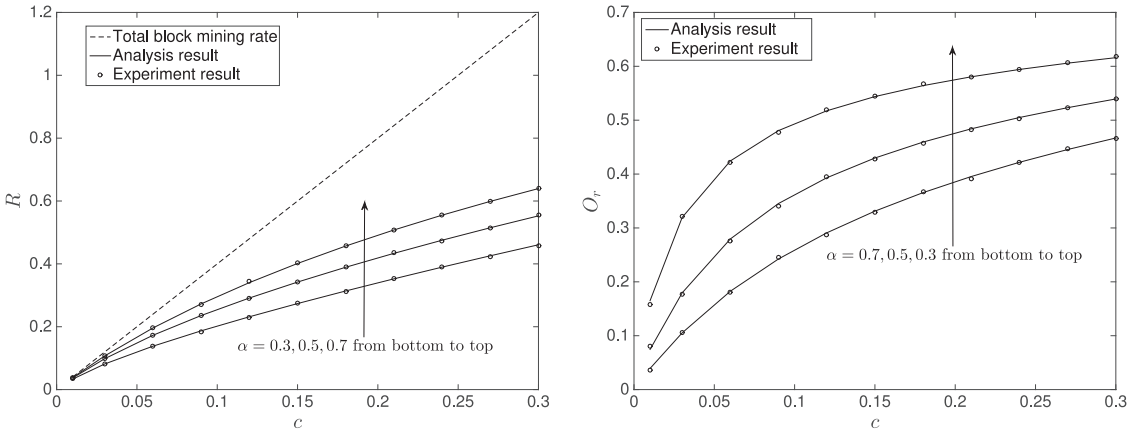
(a) The blockchain capacity. (b) The ratio of stale blocks.

Fig. 11. The capacity and the ratio of stale blocks with the growth of the non-partition probabilities α in the blockchain with four miners, where different block mining rates, i.e., $c = 0.1, 0.25, 0.5$ are considered.

Fig. 12 shows the improvement of the mining rate on the blockchain capacity, R, and the ratio of stale blocks, O_r, where the four miners are connected in a P2P fashion and have even non-partition probabilities, i.e., $\alpha = 0.3,\ 0.5$ and 0.7 from bottom to top. The total block mining rate is indicated by the dashed line in Fig. 12(a) for reference purpose. Fig. 12(a) numerically confirms the finding that the blockchain capacity can be improved by accelerating the block mining rates, especially in well-connected blockchain networks, e.g., the network with $\alpha = 0.7$. We also see that the achieved blockchain capacity, i.e., R, is always below the total block mining rate, because only a single block among the blocks at the same height and generated by different miners can be eventually accepted. According to Fig. 12(b), the improvement of the mining rate increases stale blocks, especially in highly-partitioned blockchain networks, e.g., the network with $\alpha = 0.3$.

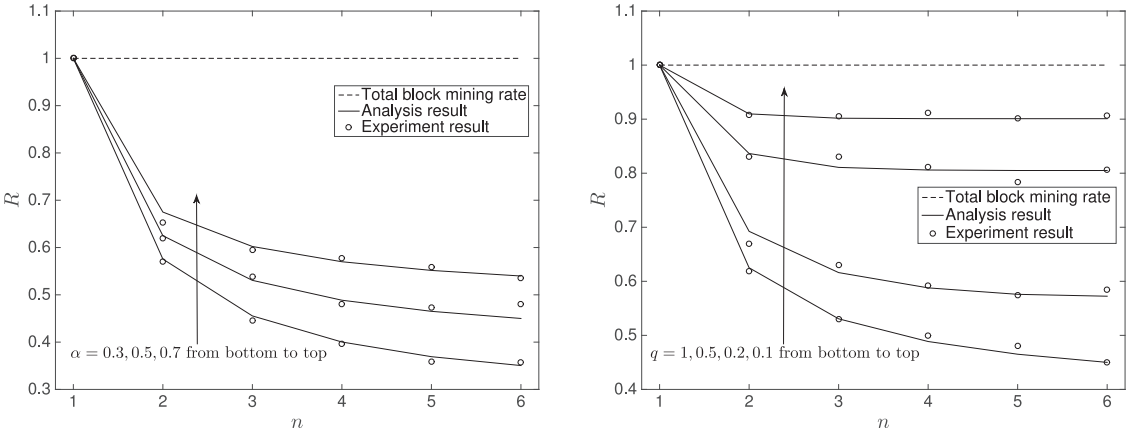
Fig. 13 plots the blockchain capacity with the growing number of miners based on the collapsed DTMC, where the total block mining rate is set to be 1, as indicated by the dash lines in the figure for reference purpose. In Fig. 13(a), miners have the same block mining rates, i.e., $c = \frac{1}{n}$. We can see that the blockchain capacity declines with the enlarging scale of the network. This is because the miners increasingly suffer from inconsistency and dispose of blocks due to inconsistency.

In Fig. 13(b), the block mining rates of the miners constitute a geometric sequence, i.e., $c_i = c_1 \times q^{i-1}$; q is a preconfigured parameter: c_1 is set to $c_1 = \frac{1-q}{1-q^n}$, if $q \neq 1$; and $c_1 = \frac{1}{n}$, if $q = 1$. n is the number of miners. The uniform non-partition probability is set to $\alpha = 0.5$. The value of q is from 0.1 to 1. We can see that, in the case that the block mining rates are all uniform, i.e., $q = 1$, the lowest blockchain capacity is provided. In other words, R is lower bounded in the case where c_i takes the same value. We further see that the blockchain capacity remains almost unchanged when the number of miners



(a) The blockchain capacity. (b) The ratio of stale blocks.

Fig. 12. The capacity and the ratio of stale blocks with the growth of the block mining rate c in the blockchain with four miners, where different non-partition probabilities, i.e., $\alpha = 0.3, 0.5, 0.7$ are considered.



(a) The blockchain capacity with the growth of the number of miners, where miners have the same block mining rates and non-partition probabilities. Three non-partition probabilities, i.e., $\alpha = 0.3, 0.5, 0.7$, are considered.

(b) The block mining rates of miners constitute a geometric sequence, i.e., $c_i = c_1 \times q^{i-1}$, q is the preconfigured parameter, and c_1 is set to $c_1 = \frac{1-q}{1-q^n}$, $q \neq 1$, and $c_1 = \frac{1}{n}$, $q = 1$.

Fig. 13. The blockchain capacity with the growth of the number of miners, where the total block mining rate is set to be 1, shown as the reference lines.

exceeds 5 and $q \neq 1$. This is because $c_i = \frac{(1-q)q^{i-1}}{1-q^n}$ is too small for $i > 5$ and has little impact on the blockchain capacity. In other words, the miners with the high block mining rates are able to dictate the blockchain capacity, and hijack the blockchain growth process (see the 51% attack [50]).

Fig. 14 shows the effect of the joint distribution of block mining rates and non-partition probabilities on the blockchain capacity, in the case that the block mining rates of four miners are set to $c = [0.1, 0.2, 0.3, 0.4]$. Suppose that the total non-partition probability of the four miners is α. Three joint distributions are considered, the positively correlated one, i.e., $\mathbf{a} = \alpha \times [0.1, 0.2, 0.3, 0.4]$, the uniform non-partition probabilities, i.e., $\mathbf{a} = \alpha \times [0.25, 0.25, 0.25, 0.25]$, and the negatively correlated one, i.e., $\mathbf{a} = \alpha \times [0.4, 0.3, 0.2, 0.1]$. We can see that the three distributions have similar effects when the miners are poorly connected, i.e., $\alpha = 0.5$. This is because the blockchain capacity is mainly determined by the fastest miner, i.e., $c_4 = 0.4$ in the figure. The difference enlarges with the growth of α. We note that the positively correlated one can always achieve the maximum blockchain capacity of the three. This observation indicates that it is better to improve the non-partition probabilities of the miners with high block mining rates to effectively enhance the blockchain capacity.

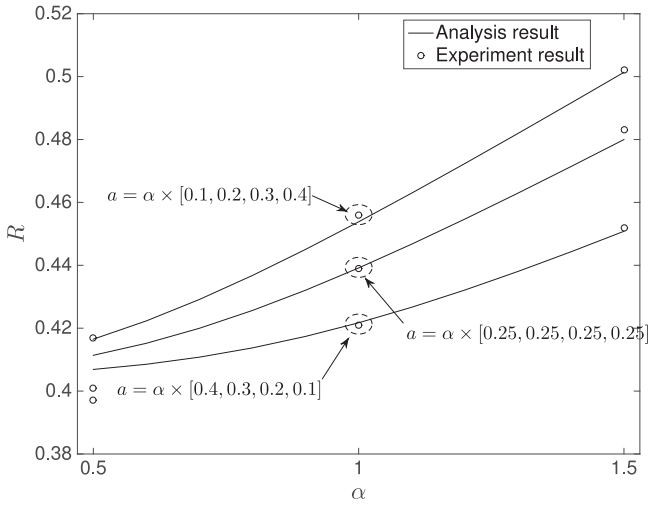


Fig. 14. The capacity of the blockchain with four miners in a heterogeneous network, where the block mining rates of miners are set to $\mathbf{c} = [0.1, 0.2, 0.3, 0.4]$. The non-partition probabilities are shown in the figure with the growth of α. Every dot is the experimental result in 10^3 timeslots.

6. Conclusion

In this paper, a new IoT blockchain testbed was developed based on Ethereum to analyze the capacity of blockchains in the presence of IoT network partitioning. The block mining algorithm and consensus protocol of Ethereum were updated to support arbitrary and heterogeneous mining rates of miners. A new DTMC model was also proposed to cross-validate our testbed design and analyze the impact of block mining rates and network conditions on the capacity of public blockchains. By exploiting the eventual consistency property of blockchain, the model was transformed to be ergodic and approximated with a finite state space and significantly improved tractability. Both our testbed and analytic model showed that the blockchain capacity can be improved by accelerating the block mining rates which, however, increases stale blocks. Our analysis provides an asymptotic upper bound for the blockchain capacity.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Gubbi, R. Buyya, et al., Internet of Things (IoT): a vision, architectural elements, and future directions, Future Gen. Comput. Syst. 29 (7) (2013) 1645–1660.
[2] H. Wang, R.P. Liu, W. Ni, W. Chen, I.B. Collings, Vanet modeling and clustering design under practical traffic, channel and mobility conditions, IEEE Trans. Commun. 63 (3) (2015) 870–881.
[3] M. Ammar, G. Russello, B. Crispo, Internet of things: a survey on the security of IoT frameworks, J. Inf. Secur. Appl. 38 (2018) 8–27.
[4] X. Zha, W. Ni, R.P. Liu, K. Zheng, X. Niu, Secure data transmission and modelling in vehicular ad hoc networks, in: Proceedings of the IEEE Globecom Workshops (GC Wkshps'15), 2015, pp. 1–6.
[5] X. Wang, W. Ni, K. Zheng, R.P. Liu, X. Niu, Virus propagation modeling and convergence analysis in large-scale networks, IEEE Trans. Inf. Forensics Secur. 11 (10) (2016) 2241–2254.
[6] F. Al-Turjman, S. Alturjman, Confidential smart-sensing framework in the IoT era, J. Supercomput. 74 (10) (2018) 5187–5198.
[7] N. Kshetri, Can blockchain strengthen the internet of things? IT Profess. 19 (4) (2017) 68–72.
[8] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.
[9] X. Wang, X. Zha, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Survey on blockchain for Internet of Things, Comput. Commun. 136 (2019) 10–29.
[10] G. Zyskind, O. Nathan, A.S. Pentland, Decentralizing privacy: using blockchain to protect personal data, in: Proceedings of the IEEE Security and Privacy Workshops (SPW'15), IEEE, 2015, pp. 180–184.
[11] IOTA, IOTA, 2017, https://www.iotatoken.com.
[12] M. Vukolić, The quest for scalable blockchain fabric: proof-of-work vs. BFT replication, in: Proceedings of the International Workshop on Open Problems in Network Security, Springer, 2015, pp. 112–125.
[13] D. Schiener, IoT and blockchain: a relationship that makes sense?, 2017, https://blog.iota.org/a-primer-on-iota-with-presentation-e0a6eb2cc621.
[14] G. Yu, X. Wang, X. Zha, J.A. Zhang, R.P. Liu, An optimized round-robin scheduling of speakers for peers-to-peers-based byzantine faulty tolerance, in: 2018 IEEE Globecom Workshops (GC Wkshps), 2018, pp. 1–6.
[15] The coordinator, 2018, https://domschiener.gitbooks.io/iota-guide/content/chapter1/current-role-of-the-coordinator.html.
[16] R. Han, V. Gramoli, X. Xu, Evaluating blockchains for IoT, in: Proceedings of the 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS' 18), 2018, pp. 1–5.
[17] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: a technical survey on decentralized digital currencies, IEEE Commun. Surveys Tut. 18 (3) (2016) 2084–2123.
[18] S. Gilbert, N. Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT News 33 (2) (2002) 51–59.

[19] D. Abadi, Consistency tradeoffs in modern distributed database system design: cap is only part of the story, Comput. 45 (2) (2012) 37–42.
[20] A. Dorri, S.S. Kanhere, R. Jurdak, Towards an optimized blockchain for IoT, in: Proceedings of the 2rd International Conference on Internet-of-Things Design Implementation, ACM, 2017a, pp. 173–178.
[21] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: the case study of a smart home, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017b, pp. 618–623.
[22] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, 2008, https://bitcoin.org/bitcoin.pdf.
[23] Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: Proceedings of the International Conference on Financial Cryptography Data Secur. (FC '15), Springer, 2015, pp. 507–527.
[24] A. Gervais, et al., On the security and performance of proof of work blockchains, in: Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS '16), ACM, 2016, pp. 3–16.
[25] L. Kiffer, R. Rajaraman, a. shelat, A better method to analyze blockchain consistency, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, ACM, New York, NY, USA, 2018, pp. 729–744.
[26] R. Pass, L. Seeman, A. Shelat, Analysis of the blockchain protocol in asynchronous networks, in: J.-S. Coron, J.B. Nielsen (Eds.), Proceedings of the Advances in Cryptology – EUROCRYPT 2017, Springer International Publishing, Cham, 2017, pp. 643–673.
[27] F. Al-Doghman, Z. Chaczko, J. Jiang, A review of aggregation algorithms for the internet of things, in: Proceedings of the 25th International Conference on Systems Engineering (ICSEng'17), 2017, pp. 480–487.
[28] C. Jaffe, C. Mata, S. Kamvar, Motivating urban cycling through a blockchain-based financial incentives system, in: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and 2017, ACM International Symposium on Wearable Computers UbiComp '17, ACM, New York, NY, USA, 2017, pp. 81–84.
[29] J. McKinney, Light client protocol, 2017, https://github.com/ethereum/wiki/wiki/Light-client-protocol.
[30] M.R. Rahman, L. Tseng, S. Nguyen, I. Gupta, N. Vaidya, Characterizing and adapting the consistency-latency tradeoff in distributed key-value stores, ACM Trans. Auton. Adapt. Syst. 11 (4) (2017) 20.
[31] Bitcoincash, 2018, https://www.bitcoincash.org.
[32] V. Buterin, Ethereum: a next-generation smart contract and decentralized application platform (2014), https://github.com/ethereum/wiki/wiki/White-Paper.
[33] Top 100 cryptocurrencies by market capitalization, 2018, https://coinmarketcap.com.
[34] Ligi, Cross Compiling Ethereum, 2019, https://github.com/ethereum/go-ethereum/wiki/Cross-compiling-Ethereum.
[35] L. Zhou, L. Wang, Y. Sun, P. Lv, Beekeeper: a blockchain-based IoT system with secure storage and homomorphic computation, IEEE Access 6 (2018) 43472–43488.
[36] X. Wang, X. Zha, G. Yu, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Attack and defence of ethereum remote APIS, in: Proceedings of the IEEE Globecom Workshops (GC Wkshps), 2018, pp. 1–6.
[37] S. Huh, S. Cho, S. Kim, Managing IoT devices using blockchain platform, in: Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT' 17), 2017, pp. 464–467.
[38] A. Ouaddah, A.A. Elkalam, A.A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in IoT, in: Europe and MENA Cooperation Advances in Inf. and Communication Technologies, Springer, 2017, pp. 523–533.
[39] Y. Rahulamathavan, R.C.-W. Phan, S. Misra, M. Rajarajan, Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption, in: Proceedings of the 2017 Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems, Odisha, India, 2017.
[40] X. Zha, W. Ni, X. Wang, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, The impact of link duration on the integrity of distributed mobile networks, IEEE Trans. Inf. Forensics Secur. 13 (9) (2018) 2240–2255.
[41] P. Maymounkov, D. Mazieres, Kademlia: a peer-to-peer information system based on the XOR metric, in: Proceedings of the International Workshop on Peer-to-Peer System, Springer, 2002, pp. 53–65.
[42] karalabe, bas vk, zelig c, et al., Command Line Options, 2019, https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options.
[43] Sense HAT, 2019, https://www.raspberrypi.org/products/sense-hat/.
[44] K. Biswas, V. Muthukkumarasamy, Securing smart cities using blockchain technology, in: Proceedings of the 18th IEEE International Conference on High Performance Computer and Communications ; 14th IEEE International Conference on Smart City; 2nd IEEE International Conference on Data Science and Intelligent Systems (HPCC/SmartCity/DSS'16), 2016, pp. 1392–1393.
[45] F. Al-Turjman, S. Alturjman, Context-sensitive access in industrial internet of things (IIOT) healthcare applications, IEEE Trans. Ind. Inform. 14 (6) (2018) 2736–2744.
[46] B. Sintay, Unix timestamp, 2013, http://www.unixtimestamp.com.
[47] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper (2014). https://ethereum.github.io/yellowpaper/.
[48] Petertodd, et al., 2014, https://en.bitcoin.it/wiki/Maximum_transaction_rate.
[49] T. Konstantopoulos, Markov chains and random walks, 2009, https://pdfs.semanticscholar.org/251b/3a804be62da4e0835fcffa704cacb03ce310.pdf.
[50] D. Bradbury, The problem with Bitcoin, Comput. Fraud Secur. 2013 (11) (2013) 5–8.