A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus and Traffic Modeling

LAPHOU LAO, The Hong Kong Polytechnic University, China
ZECHENG LI, The Hong Kong Polytechnic University, China
SONGLIN HOU, The Hong Kong Polytechnic University, China
BIN XIAO[*], The Hong Kong Polytechnic University, China
SONGTAO GUO, Chongqing University, China
YUANYUAN YANG, Stony Brook University, America

Blockchain technology can be extensively applied in diverse services, ranging from online micro-payment, supply chain tracking, digital forensics, health-care record sharing to insurance payment. Extending the technology to the Internet of things (IoT), we can obtain a verifiable and traceable IoT network. Emerging research in IoT applications exploits blockchain technology to record transaction data, optimize current system performance or construct next-generation systems, which can provide additional security, automatic transaction management, decentralized platforms, and offline-to-online data verification, etc. In this paper, we conduct a systematic survey of the key components of IoT blockchain and examine a number of popular blockchain applications.

In particular, we first give an architecture overview of popular IoT-blockchain systems by analyzing their network structures and protocols. Then, we discuss variant consensus protocols for IoT blockchains, and make comparisons among different consensus algorithms. Finally, we analyze the traffic model for P2P and blockchain systems and provide several metrics. We also provide a suitable traffic model for IoT-blockchain systems to illustrate network traffic distribution.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Networks → Network architectures**; **Network protocols**; • **Theory of computation → Models of computation**.

Additional Key Words and Phrases: Blockchain, IoT, architecture, consensus, traffic modeling

ACM Reference Format:
Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. 2019. A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus and Traffic Modeling. 1, 1 (November 2019), 32 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

[*] Corresponding Author

Authors' addresses: Laphou Lao, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, laphou.lao@connect.polyu.hk; Zecheng Li, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, cszcli@comp.polyu.edu.hk; Songlin Hou, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, csslhou@comp.polyu.edu.hk; Bin Xiao, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China, csbxiao@comp.polyu.edu.hk; Songtao Guo, Department of Electronic and Information Engineering, Chongqing University, China, guosongtao@cqu.edu.cn; Yuanyuan Yang, Department of Computer Science, Stony Brook University, New York, America, yuanyuan.yang@stonybrook.edu.

XXXX-XXXX/2019/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Blockchain technology has the potential to become ubiquitous in financial services and industries. As the core mechanism of publicized cryptocurrencies, such as Bitcoin [Nakamoto 2008] and Ethereum [Wood 2014], blockchain has demonstrated its intrinsic characteristics, including decentralization, anonymity, and auditability. A large number of blockchain applications has been deployed. The worldwide estimated expenditure related to these applications will reach nearly $12 billion by 2020 [Linask 2018]. Apart from cryptocurrencies, blockchain can also be used in diverse applications, such as the Internet of Things (IoT).

IoT applications allow direct communications and interactions between devices over the Internet. Current IoT devices comprise smartphones, smart home appliances, vehicles, and indoor and outdoor sensors. Before 2020, the number of IoT devices will grow to 20 billion [Jackson 2018]. However, with the involvement of a huge number of devices, traditional IoT applications are facing challenges in many aspects, including data integrity, security, and robustness.

Blockchain provides a practical solution to address many of the limitations of traditional IoT applications. Blockchain can ensure IoT data integrity without a third party, while saving bandwidth and computational power of IoT Devices. Moreover, blockchain can provide a secure and scalable framework for an IoT network so that sensitive information can be delivered without a centralized server. Equipped with identification and authentication functions, blockchain can track sensor data measurements and transfer data among IoT peers without a central server. Furthermore, blockchain can reduce the operational cost of the IoT. By allowing smart devices to make automated micro-transactions, two parties can reach an agreement much faster and at a lower cost. A surge in numbers of applications combining blockchain and the IoT can be seen, with nearly 20% of IoT deployments adopting blockchain services by 2019 [Bieler 2018]. For example, IOTA [Sikorski et al. 2017], one of the world's major cryptocurrencies, enables micropayment transactions between IoT devices.

However, it is still challenging to deploy IoT applications on blockchain systems. First, the architecture of an IoT-blockchain system needs to support an enormous number of IoT devices. Second, the consensus, a mechanism to ensure data integrity among peers in the blockchain, needs to be specifically designed for IoT blockchain due to the limited storage and the computing capability of IoT devices. Third, traffic modeling of a blockchain network is needed to realize high system performance in IoT blockchains. A deep understanding of a traffic model can reveal clues for optimizing communication processes and protocols.

Some previous work has discussed the integration of IoT and blockchain. In [Christidis and Devetsikiotis 2016], smart contracts of IoT blockchain are investigated and certain unsolved issues are pointed out. It concludes that the IoT-blockchain combinations can lead the way for new business models and applications. Marco *et al.* conduct a systematic literature review of 18 use cases of the blockchain and investigated system integrity, anonymity, and adaptability [Conoscenti et al. 2016]. Alfonso *et al.* categorize IoT blockchains into different domains and survey their challenges [Panarello et al. 2018]. A study of a smart home presents the efficiencies of the integration of blockchain with home devices [Dorri et al. 2017a].

Different from previous work that provides surveys of IoT networks [Da Xu et al. 2014; Whitmore et al. 2015] and blockchain systems [Zheng et al. 2018] separately, a survey of IoT applications using blockchain technology is presented in this paper. This paper also provides a survey of traffic analysis of IoT blockchain systems, which has not been done before. With a solid survey of many IoT blockchains, we are in a new position to present several key observations and propose new architecture and traffic models. Specifically, we delve into three critical aspects of IoT blockchain, i.e., architecture, consensus and traffic modeling.

To summarize, this paper makes the following contributions.

- We survey IoT-blockchain systems, and analyze their architectures. We also propose a general IoT-blockchain architecture.
- We make comparisons between current consensus algorithms and communication protocols, and show their strengths and shortcomings when applied in IoT blockchains.
- We analyze the current traffic models of P2P and blockchain systems. Moreover, we propose a traffic model of IoT-blockchain systems.

The rest of this article is structured as follows. Section II gives the background of blockchains and IoT systems. In Section III, we survey the architectures of IoT blockchains. The consensus mechanisms are discussed in Section IV. The traffic model analysis is presented in Section V.

2 BACKGROUND

2.1 Blockchain

2.1.1 Overview. As the name suggests, blockchain can be considered as blocks chained together. Since the advent of a novel type of cryptocurrency (e.g., Bitcoin), it is also referred to in public as digital ledgers. Blockchain is a comprehensively distributed architecture that emphasizes features such as data consistency, transparency, user privacy, resistance to retroactive alteration *etc.* Unlike other centralized systems, blockchain-based systems use a decentralized network, typically a peer-to-peer (P2P) network [Schollmeier 2001], to distribute data processing tasks to different nodes. By using a mechanism called consensus, information stored and data generated in each node can be synchronized.

In the traditional data sharing system, user information and privacy are often revealed, mishandled, stole, and accessed without authorization [Chen and Zhao 2012; Liu et al. 2018; Sun et al. 2018]. Blockchain technology improves data protection by eliminates the need for the central server. It uses the peer-to-peer network and consensus protocol to verify user data. No single node can act as a gatekeeper to control others' data. Also, blockchain improves data ownership by allowing user owners to define who can access their files. It uses asymmetric cryptography to ensure the safety of user data. Moreover, the zero-knowledge proof [Ben-Sasson et al. 2013] is used in blockchain to provide advanced data privacy of data sharing in blockchain.

Blockchains can be mainly categorized as public and private blockchains [Zheng et al. 2017]. Public blockchains [Nakamoto 2008; Wood 2014] allow anyone to participate in the blockchain network. The participants help the public blockchain to verify transactions and provide truthful services. Public blockchains are commonly used for cryptocurrencies. Private blockchains [Brown et al. 2016; Pongnumkul et al. 2017] only allow authentic nodes to join the blockchain network. The owner of the private blockchain can modify and delete entities on the blockchain. Private blockchains have been proposed for various kinds of business applications. Consortium blockchains and permissioned blockchains are also categorized as private blockchains. In consortium blockchains [Kang et al. 2017; Li et al. 2017], every participant in the blockchain network is a member of the same group, company, or industry. Permissioned blockchains like Hyperledger Fabric [Cachin 2016], which give special authority to certain nodes, have been proposed for business use in many cases. Apart from this, there are Hybrid blockchains [Ateniese et al. 2018] that combine the benefits of public blockchains and private blockchains. They protect user data by a private blockchain manager and use the consensus of public nodes to validate transactions. Hybrid blockchains are commonly used in IoT system and supply chain for data protection and access control.

Apart from the differences in access permissions, the enabling components of blockchains, such as consensus algorithms and communication protocols, can also be different. For consensus algorithms, public blockchain system Bitcoin adopts Proof-of-Work (PoW) [Nakamoto 2008], while other public blockchain systems, such as Peercoin and ShadowCash, choose Proof-of-Stake (PoS) [King and Nadal 2012]. Consensus algorithms, such as Byzantine Fault Tolerant (BFT) [Vukolić 2015], are usually used in permissioned blockchains. Many other algorithms, such as PoA, RBFT in consensus can also be found in different blockchains.

Gossip and Kademlia are two frequently used communication protocols chosen to support the message transfer in different nodes. Gossip, which is used in Bitcoin, mimics the spread of epidemic diseases by transmitting information to the whole network with every node communicating only with its neighbors. Kademlia, on the other hand, uses a distributed hashtable (DHT) to specify the network structure and the list of peers every node should communicate with.

2.1.2 Blockchain System Structure. The system structures of Bitcoin and Ethereum are representative of the public blockchains. The Bitcoin network has a hybrid network architecture containing supernodes and common nodes. The nodes in the network have four main functions: wallet, mining, block database, and network routing [Antonopoulos 2014]. Network routing is shared by all nodes. Other functions are specialized and different from node to node. Some nodes may contain only a part of these functions. The nodes that contain all functions are called core nodes.

Every node will participate in validating and broadcasting the transaction and block information, and discovering and maintaining a connection with other nodes. Nodes containing all of the transaction and block information are called full nodes. Other nodes, containing part of the block database information, are called light nodes. The wallet node is usually a PC or mobile application designed to manage users' accounts and initiate transactions.

2.1.3 Applications. Commercially, blockchain has been proved to be a successful practice, such as the most famous blockchain system Bitcoin [Nakamoto 2008], which has revolutionized today's digital trading system. With the underlying blockchain technology, there are also other applications in the financial field. For example, Alipay (HK) users can send money to the Philippine wallet Gcash using the blockchain technique [Chen 2018].

Many famous projects based on blockchains, including Ethereum [Wood 2014], Hyperledger [Androulaki et al. 2018] and IOTA [IOTA 2019], are under active development. Some advanced functions, such as smart contract provided by Ethereum [Wood 2014], have also contributed to accelerate the advent of large decentralized applications [Suryanarayana and Taylor 2004], such as decentralized medical claim systems [Ekblaw et al. 2016] and decentralized voting systems [Pilkington 2016]. Apart from its success in commerce, blockchain can also be used in lots of other fields, such as the Internet of Things (IoT).

2.2 IoT

2.2.1 Overview. With the boom in wireless technologies, Internet of Things applications have become more pervasive than ever before. The Internet of Things can be characterized as physical devices with network functions, computers, and items embedded with connectivity functions [Atzori et al. 2010]. Together with rapid advancements in software and hardware, IoT related technologies have already become indispensable in today's society. While many IoT applications were initially designed for specialized use, most relevant technologies have been migrated and adapted for public use [Li et al. 2011a].

2.2.2 Applications. IoT applications can be found in many areas for different purposes. For example, a large number of electronics companies are designing devices for smart homes [Stojkoska and Trivodaliev 2017]. *Xiaomi*, a popular electronics company, has already released nearly 100 different electronic products for smart homes. The IoT is also used in smart cities to provide more convenience to the public [Zanella et al. 2014]. Shared bicycles and shared power banks are deployed in many places, e.g., almost all first-tier cities in China. IoT technologies are also applied to public transportation. Electronic tickets (or E-tickets) can save passengers a great deal of time [Xie and Shugan 2001]. Mobile payments provide much more convenience, as an effective replacement for cash.

2.3 IoT Blockchain

2.3.1 Overview. IoT blockchains are blockchain systems that are customized and optimized to enable IoT applications. IoT applications have been developed and applied in many fields [Miorandi et al. 2012]. However, most of these applications are prone to problems, such as data leakage and systematic failure. To mitigate these problematic effects, blockchain can be used to provide higher security and stability for traditional IoT applications.

Despite the benefit that blockchain brings to traditional IoT applications, there are still many barriers in its actual implementation. The major problems come from the limitations of IoT devices. Issues such as task distribution, power consumption, and computational ability need to be taken into account before blockchain can be effectively applied in most IoT applications. To cope with these issues, many attempts to adopt blockchain in IoT applications have been seen in recent years [Bahga and Madisetti 2016; Huh et al. 2017; Sharma et al. 2017]. Although IoT blockchain applications are in many aspects far from mature, we are optimistic that blockchain can hopefully confer more capabilities and opportunities in the IoT world with an increase in real-world practices and studies on IoT blockchain.

2.3.2 Applications. Many IoT applications now adopt blockchain for various purposes [Bahga and Madisetti 2016; Huh et al. 2017; Sharma et al. 2017]. Generally, there are three major categories, including digital payment, smart contract service, and data storage.

Digital Payment Digital payment is the first and most widely used field for blockchain [Swan 2015]. While it initially works on a distributed network supported by mostly high performance machines, special optimization now supported by major blockchains such as Bitcoin and Ethereum is used for devices with trivial computation power, such as smartphones and pocket PCs. Instead of being assigned massive computational work, low-end devices usually work as light nodes, which do not keep the complete chain in their local storage or participate in the most power consuming processes, like mining. These features have made mobile payment with the blockchain technique much more accessible than before.

Smart Contract The smart contract serves as a mechanism to make blockchain systems flexible and scalable to cope with contract-related tasks [Christidis and Devetsikiotis 2016]. In a narrow sense, smart contracts are considered rules or agreements implemented in a program to automate transactions. Broadly speaking, smart contracts are viewed as general procedures to build automation systems and control IoT devices. Currently, many companies provide IoT solutions using a smart contract [Buterin et al. 2014]. For example, LeewayHertz [LeewayHertz 2019] is a company that provides solutions for IoT startups and enterprises using Ethereum smart contract. A drug supply chain network on blockchain is developed by TraceRx [TraceRx 2019]. Similar projects have been launched in companies like Techracers[Teachracers 2019], Attores[Attores 2019], etc.

Data Storage Blockchain can be utilized as a distributed secure database in data storage applications. Project DokChain [DokChain 2019] provides a viable solution in healthcare improvement. It uses permissioned chain (Hyperledger Sawtooth) for protecting the privacy of customer's data. Also, it uses blockchain as a distributed database to group financial and clinical data to simplify the treatment process. Factom [Factom 2019] is a company providing blockchain capabilities for traditional applications. It provides APIs to integrate data into its blockchain to ensure data integrity. The blockchain data storage also needs to consider the data authenticity when a user queries a blockchain system. A middleware solution is given in [Peng et al. 2019] to ensure both the query efficiency and data authenticity.

2.3.3 *Challenges.*

Despite the benefits of IoT blockchain, there are some challenges of adopting blockchain technology in IoT system [Atlam et al. 2018; Dorri et al. 2016; M.Padma et al. 2019; Ramachandran and Krishnamachari 2018; Reyna et al. 2018]. They can be summarized into two categories.

Resource Constraints

General blockchain requires high computational power, high bandwidth, and low delays. Most blockchain systems use Proof-of-Work (PoW) as their underlying consensus mechanism. However, the mining process in PoW requires huge computational power. Most IoT devices have simple hardware specifications and low processing power. It is not capable or time consuming for IoT devices to perform the mining tasks of blockchain.

Apart from this, blockchain needs to perform data encryption frequently. However, the encryption speed and time will be different because different IoT devices have different computational power. Moreover, other processes, consistency algorithms, and routine testing require huge processing power, which overloads the low power capacity of IoT devices.

Moreover, in blockchain, the transactions and blocks are not stored in a central server. But some nodes need to keep a copy of the full ledger in their storage. The size of the ledger will increase over time. However, the majority of IoT devices have low hardware storage capacity. Low power IoT devices only have 10KB to 100KB memory for storing data and memory [Bormann et al. 2014]. But blockchain requires massive storage for storing the entire chain. For example, Bitcoin needs over 200 GB of memory, Etherum requires over 1.5 T of memory. It is not capable of storing a copy of the full blockchain for IoT components.

Additionally, the consensus process in blockchain requires the exchange of information between nodes frequently to reach an agreement to maintain the correctness of blockchain and generate new blocks. This process requires high bandwidth and low network latency. However, IoT devices are always strict in a limited bandwidth.

Scalability

According to the blockchain trilemma by Vitalik Buterin, it is hard to achieve high scalability in a secure and fully decentralized blockchain network. However, the IoT system is expected to involve numerous IoT devices. Also, IoT devices always require to join and leave the network frequently in many application scenarios. For example, PBFT consensus mechanism is considered as a suitable protocol for IoT system. However, the PBFT algorithm can only work well in a fixed-size network, in which the network members cannot easily change. That is not scalable for the numerous IoT devices management.

2.4 Current Trend in IoT Blockchain Development

With the rapid growth of IoT and blockchain technology, the development trend of IoT blockchains also continues to change. The current trend is mainly reflected in four aspects: popularity, range of applications, development of underlying technology, and business models.

2.4.1 *Popularity.*

In the past 10 years, the IoT and blockchain industries both grew rapidly [Whitmore et al. 2015; Zheng et al. 2017]. In the beginning, IoT was only applied in certain fields, such as manufacturing, logistics, and transportation. Then, more and more emerging industries became involved in the wave of IoT, such as smart homes, smart cities, and asset digitalization. A growing trend of research in related fields also indicates a growing interest in academia [Khan and Salah 2018; Kshetri 2017; Zheng et al. 2018]. Different kinds of consensus algorithms have been presented and structural improvements have been proposed for IoT blockchains [Samaniego and Deters 2016; Zhang and Wen 2017].

2.4.2 Range of Applications. IoT blockchain has also become more widely applied than ever before. The initial uses of blockchain have focused on decentralized currency systems [Gervais et al. 2014]. One of the most famous blockchain systems, Bitcoin, was first designed to create a decentralized currency system without any supervision. With the development of blockchain technologies, blockchain systems with more advanced features, such as smart contract in Ethereum, have enabled a wider variety of applications and services other than financial use. The integration of IoT into blockchain systems provides even more possibilities for applications. Logistics companies such as SF Express focus on using blockchain to implement asset tracking. Hardware and electric product companies, e.g. MI, will use blockchain to enhance the interaction between humans and IoT devices. Energy companies use blockchain to implement energy sharing and energy transaction business. IoT blockchains can also be deployed in the edge computing environment where IoT devices contain limited resources and a resource allocation strategy is proposed in [Huang et al. 2019].

2.4.3 Development of Underlying Technology. The rapid development of many underlying technologies also contributes to the rapid growth of IoT and blockchain. In view of the scale of IoT connectivity, with the development of communication technologies such as LoRa, NB-IoT [Sinha et al. 2017] and 5G [Andrews et al. 2014], communications using IoT devices in China has increased rapidly. By the end of June 2018, the number of Chinese Internet accesses reached 465 million, twice the amount of the previous year. To meet different blockchain application requirements, newly-proposed structures are equipped with optimization functions. For example, many new blockchain systems adopt fast consensus algorithms for quick transaction confirmation. Public chains like IOTA [Sikorski et al. 2017] and EOS [Kakushadze and Russo Jr 2018] solve the low transaction rate problem of traditional implementations. Private chains like Hyperledger [Cachin 2016] are more suitable for consortium business.

2.4.4 Business Models. Many companies are seeking ways to integrate blockchain techniques with their business models to enhance profits. For example, the E-Business company, JD [JDChain 2019], deploys a blockchain system to solve problems occurring in logistics and electronic invoices. There are also several start-up companies, such as Slock.it [sloct.it 2018], which use existing blockchain systems such as Ethereum to implement P2P business. In this way, IoT devices will connect to the current blockchain system and users can use smart contract to interact with these devices. However, the blockchain system performance may severely affect the supported business, e.g., how fast the consensus algorithm validates transactions, which deserves more attention when designing industrial applications.

3 ARCHITECTURE

This section presents the architecture of IoT, blockchain, and blockchain-based IoT. In addition, we investigate the current IoT blockchain applications, give a general architecture of IoT blockchain, and conduct several case studies.

3.1 IoT Network Architecture

The traditional IoT architectures are well discussed in previous work [Al-Fuqaha et al. 2015; Da Xu et al. 2014; Said and Masud 2013; Whitmore et al. 2015]. From the pool of proposed architectures, there are two types of IoT architecture: three-layer architecture [Al-Fuqaha et al. 2015; Khan et al. 2012; Said and Masud 2013; Wu et al. 2010; Yang et al. 2011] and five-layer architecture [Al-Fuqaha et al. 2015; Atzori et al. 2010; Chaqfeh and Mohamed 2012; Said and Masud 2013; Tan and Wang

2010; Wu et al. 2010]. The three-layer architecture, which is illustrated in Fig. 1 (a), comprises three basic layer.

Physical layer: The physical layer consists of a variety of IoT devices including sensors, RFID tags, NFC devices, and mobile phones. This layer is responsible for connecting different devices, exchanging messages, and gathering information to the upper layer.

The mobile phone is an integrated IoT device which includes many IoT elements, including camera, microphone, proximity sensor, ambient light sensor, accelerometer, gyroscope, magnetic field sensor, temperature sensor, RFID, and NFC. There are billions of mobile phone users generate massive data at every single moment around the world. Social network services and social applications rely on mobile phones. User information like messages, voices, photos, and location information is collected by mobile phones and uploaded onto social platforms.

Radio-Frequency Identification (RFID) is a wireless radio communication technology. RFID systems exchange data between RFID tags and RFID readers by electromagnetic fields. Each RFID tag contains a unique identifier, which allows RFID readers to distinguish it from others. The RFID technology is widely used in the indoor pedestrian localization, indoor object positioning, and commodity tracking in logistics systems. The IoT system employs the RFID technology for physical security, device management, and monitoring service.

Near Field Communication (NFC) is a communication protocol which allows devices to communicate within short ranges. NFC provides an easy communication channel and data sharing approach without the need to set up a network. Also, NFC technology can be utilized to connect unpowered objects, which provides a more flexible approach to accessing IoT devices around us. As a large amount of NFC-enabled devices are deployed, NFC technology is playing an essential role in the IoT world.

Other IoT hardware devices include various kinds of wireless sensors for environmental monitoring. A large wireless sensor network can consist of numerous wireless sensors. They work together to perform tasks like data collection, data monitoring, and data analysis in wide ranges. Common wireless sensors include infrared sensors, temperature sensors, ultrasonic sensors, pressure sensors, chemical sensors, and smoke sensors. They perform the environmental monitoring of light, temperature, sound pressure, food quality, and air quality.

Network layer: The network layer transfers information which collected or processed by IoT devices. Data is conveyed through diverse technologies such as 4G, 5G, RFID, WiFi, and Bluetooth. Specifically, the network layer connects edge nodes, hypervisors, and user applications.

Application layer: The application layer can be recognized as an abstraction of IoT services provided to users. A user can request different services through APIs. After receiving user requests, the application layer either processes the information locally or calls the underlying APIs to handle the requests.

With the rapid development of the IoT, many auxiliaries and business models have emerged. The rudimentary three-layer architecture model cannot provide a sufficiently accurate abstraction. Accordingly, a more concrete five-layer architecture is proposed to fulfill the description of IoT, which is exhibited in Fig. 1 (b).

In the proposed five-layer architecture, the physical layer is renamed the sensing layer, because of the widely-deployed sensors in IoT systems. In addition, the network layer was expanded to the network layer together with the middleware layer. The new network layer still undertakes the task of data transmission while the middleware layer is responsible for data processing, data storage, and service management.

Middleware layer: Since there are numerous heterogeneous devices in the lower layer of the IoT system. They may have different operating systems, query formats, and data formats. It is difficult for software developers to develop an integrated IoT application which is compatible with all IoT devices. Middleware layer sits between IoT devices and IoT applications to handle the compatibility problem. It handles users' requests and gives efficient responses. The functionalities provided by the middleware layer consist of primary message delivery, API management, data and service integration, and data formatting. Network companies like Cisco [Cisco IoT 2019], Huawei [HUAWEI IOT 2019], Oracle [Oracle IoT 2019], and Mulesoft [MuleSoft IoT 2019] offer IoT middlewares.

Moreover, the business layer was added to the top of application layer.

Business layer: As the IoT develops, the applications supported by an IoT system have increased dramatically, forming a complicated ecosystem. A single application layer cannot accurately describe it. Therefore, the business layer was proposed to represent a higher-level abstraction of the IoT ecosystem. It is responsible for the management of the entire IoT system. Also, the business layer builds business models, graphs, and performs data analysis.

Depended on the analysis results, the information provided in this layer offers guidance and recommendations to help system managers in decision-making and future development.

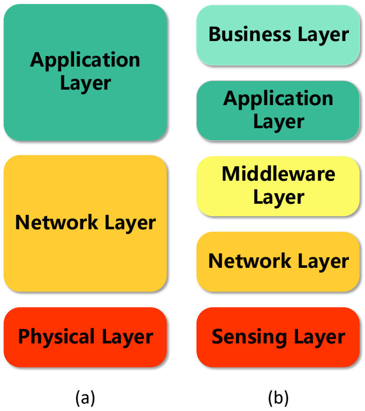


Fig. 1. IoT general architecture: (a) three-layer architecture; (b) five-layer architecture.

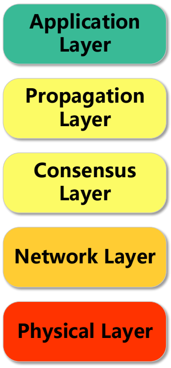


Fig. 2. Five-layer architecture of blockchain.

3.2 Blockchain Architecture

The architectures of blockchain have been broadly discussed in a previous work [Lin and Liao 2017; Swan 2015; Zheng et al. 2018]. We propose a five-layer architecture model, which summarizes previous discussions. This model is illustrated in Fig. 2.

The physical layer consists of various kinds of blockchain nodes. They interconnect with each other to form the blockchain network. In general, there are two types of nodes in blockchain. Full nodes and light nodes.

Full nodes: Full nodes keep a copy of the full blockchain in their storage. They participate in creating, verifying, and adding new blocks to blockchain. Accordingly, full nodes maintain the security, consensus, and data correctness on blockchain. Network participants require high hardware storage to become a full node. Some full nodes can be miners that contribute their computation power and resources to get a relative amount of reward. Miner completes a hash calculation and broadcasts his result to the blockchain network. Then, the result will be verified by others under a consensus protocol. Once the result is confirmed, the miner gets his reward.

Light nodes: Light nodes don't require high hardware specifications. They don't keep a copy of the full blockchain but just know the status of the last block. Light nodes obtain the blockchain information from full nodes.

The network layer provides reliable data delivery service. It allows the blockchain nodes to communicate with each other. In general, blockchain network is based on Peer to Peer (P2P) network. In the network, there are no central servers and central authority. Network participants reach an agreement to ensure the integrity and security of the network under a consensus protocol. Accordingly, blockchain achieves decentralization and avoid centralized vulnerabilities.

The consensus layer is responsible for ensuring the consistency of the stored data and incentivizes participants to find a new block. The consensus mechanism is the important part in blockchain to ensure data correctness and integrity. We discussed the consensus mechanisms in depth in Section IV.

The propagation layer consists of communication protocols that define the rules of how messages [Wen et al. 2014] and blocks [Decker and Wattenhofer 2013] are propagated in the network. Bitcoin and Hyperledger fabric adopt the Gossip as their communication protocol while Ethereum employs the Kademlia algorithm as the basis of its communication protocol.

Gossip protocol: Gossip protocol represents a kind of epidemic-like peer-to-peer communication method. In a gossip-based system, updates spread like an infectious disease and eventually infect nearly all nodes. Therefore, gossip protocol utilizes the terminology of epidemiology to denote different nodes types. A node which owns an update and is about to spread to others is called the *"infective"* node. In contrast, a node which waits to accept an update from others is denoted as the *"susceptible"* node. Specifically, a node that owns an update but is not going to spread it is called the *"removed"* node.

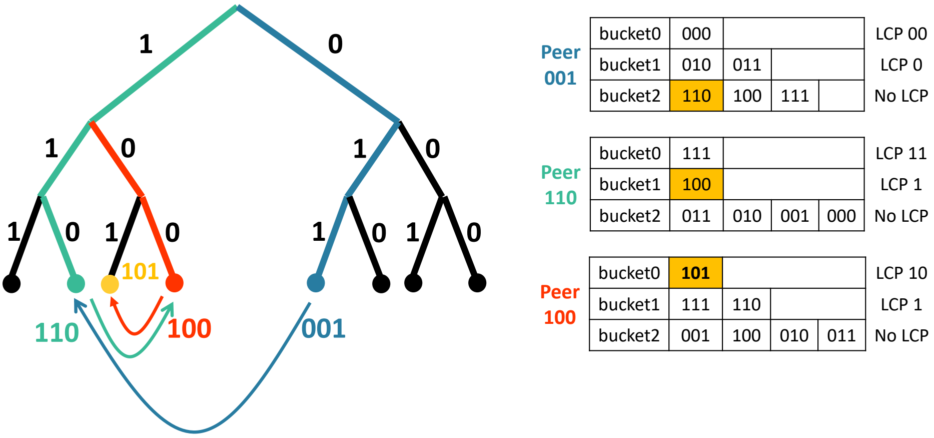
Gossip protocol has two different modes of propagation: anti-entropy and rumor-mongering. The anti-entropy method reduces the entropy of the system by eliminating the differences between distinct nodes and making the whole system orderly. Specifically, a node with an update periodically chooses one or several nodes at random and compares contents. If discrepancy of contents is found among the nodes, they will resolve the differences through data-pulling or data-pushing to get each one updated. Rumor-mongering adopts a different method. It merely propagates the updates to a random node on a regular basis. The whole system can achieve eventual consistency as time goes by.

The two modes of propagation require different types of nodes. In anti-entropy, only susceptible and infective nodes are needed. Therefore, anti-entropy is also recognized as the *SI* model or simple epidemics model. In contrast, rumor-mongering needs nodes of all the three types and is recognized as the *SIR* model or complex epidemics model.

Kademlia: Kademlia is an algorithm designed for decentralized peer-to-peer networks. It is used in Ethereum to optimize network routing and to help locate target nodes. In Kademlia, each node has its unique node ID. The Kademlia algorithm uses node ID to perform fast peer lookup and locate files or resources.

In the Kademlia network, each node is recognized as a leaf on a binary tree. The logical distance between two node IDs can be calculated by applying the XOR operation. The smaller result represents the closer logical distance between two nodes. The nodes with the longest common prefix (LCP) will share a subtree. K-buckets can be expressed as subtrees. The K in K-bucket refers to the maximum number of leaves in the subtree. The routing table in Kademlia stores information in a K-bucket list. Fig. 3 gives an example of the binary tree, routing table, and K-bucket in Kademlia.

The process of peer lookup in Kademlia is illustrated in Fig. 3. Assume the node 001 hopes to locate a target node 101 (find the IP address of node 101). The Kademlia algorithm will first calculate the logical distance of 001 and 101 by XOR operation and get operation result 100, which means



Peer 001					
bucket0	000				LCP 00
bucket1	010	011			LCP 0
bucket2	110	100	111		No LCP

Peer 110					
bucket0	111				LCP 11
bucket1	100				LCP 1
bucket2	011	010	001	000	No LCP

Peer 100					
bucket0	101				LCP 10
bucket1	111	110			LCP 1
bucket2	001	100	010	011	No LCP

Fig. 3. Illustration of peer lookup in Kademlia.

Table 1. Summary of Blockchain Applications

Blockchain Application	Consensus Mechanism	TPS (tx/sec)	Releases date
Bitcoin	PoW	7	Jan 2009
Litecoin	PoW	56	Oct 2011
Bitshares	DPoS	17	Jul 2014
NEO	DBFT	1000	Feb 2014
Ethereum	Pow/PoS	15	Jul 2015
Hashgraph	Hashgraph	10000	Jul 2017
Tangle	DAG	800	Apr 2018
Ripple	Ripple	1700	May 2018
EOS	DPoS	3000	Jun 2018
QTUM	PoS	70	Jul 2018
Futurepia	DDPoS	300000	Sep 2018
Casper	PoS	10000	Jan 2019
Monoxide	PoW with Chu-ko-nu Mining	15.6	Feb 2019

the largest common prefix of the IDs is 1. The algorithm will then try to find the target node from 2-bucket($K = 3 - 1 = 2$) in the routing table of peer 001. If the target node is found in 2-bucket, then this process finishes.If not, the algorithm will send a lookup request to 110 (The first record in 2-bucket). Then the lookup operation will be performed recursively until the target node 101 is found.

The application layer is responsible for implementing functions according to the application requirement. Blockchain applications have been widely used as cryptocurrency in financial markets. Also, there is a clear trend that blockchain applications will internationally adopt in health care, IoT, and supply chain shortly. We summarize and compare the consensus mechanism, performance in transaction per second, and releases date of representative blockchain applications in Table 1.

3.3 General IoT-Blockchain Architecture

After illustrating the typical architectures of IoT and blockchain, we conclude a five-layer general architecture of IoT-blockchain from the existing IoT blockchains. It is given in Fig. 4. The architecture of existing IoT blockchains are summarized in Table 3. This architecture combines the features of traditional IoT systems and blockchain systems. The physical layer of IoT blockchain is the same as the physical layer of IoT [Gubbi et al. 2013; Lee and Lee 2015]. It includes the sensors, smart home devices, RFID tags, mobile phones, and other IoT devices that related to the IoT-blockchain application. The network layer of IoT blockchain is similar to the network layer of the traditional blockchain layer. It is responsible for internetworking, routing, and multicasting between IoT devices [Jiang et al. 2016; Maymounkov and Mazieres 2002; Milojicic et al. 2002]. In our survey, the majority of IoT-blockchain applications are based on the P2P network. Few of them uses the commercial network. The blockchain layer represents the blockchain functions in IoT blockchain, such as consensus, data storage, and data sharing. It can be a specific blockchain platform, commercial blockchain, or public blockchain solution [Crosby et al. 2016; Nakamoto 2008; Wood 2014]. The middleware layer of IoT blockchain is different from the middleware layer of IoT. The main purpose of this layer is not data formatting or integration for heterogeneous IoT devices. But it responsible for blockchain management, blockchain service integration, and providing additional security service [Alphand et al. 2018; Bahga and Madisetti 2016; Huh et al. 2017]. The application layer of IoT-blockchain architecture is similar to IoT system and traditional blockchain architectures. It provides interactions, abstraction services, and API functions to users. In general, IoT-blockchain application is included as one of the blockchain application. Also, it can be known as an IoT application that implements blockchain technology [Dorri et al. 2017a; Huckle et al. 2016].

In this architecture, IoT devices are responsible for data generation. Blockchain can serve as a secure distributed database that safely keeps records and avoids malicious modification. Once a block is confirmed and added as a part of the blockchain, the transactions contained in the block as well as transactions in all previous blocks get tamper-proof. Additionally, all historical records can be easily retrieved without loss of information. The authorized user can browse all transactions and verify the authenticity of each transaction. For most applications, blockchain is used as an add-on storage technique. In addition, blockchain can support cryptocurrency and optimize digital trading in IoT systems.

3.4 Current IoT-Blockchain Applications and Their Architectures

IoT-blockchain applications are still at an early stage. However, the integration of IoT and blockchain is rapidly evolving and growing. We summarize and compare the service, blockchain, consensus, IoT devices, and company size of some typical IoT-blockchain application in Table 2. We also investigate and present the characteristics and architectures of the representative IoT-blockchain applications below. The architectures of these applications are similar to the general IoT-blockchain architecture that we proposed in Fig. 4. They are categorized into two types:

3.4.1 Specific Application. Specific application refers to the standalone software or system that used IoT and blockchain as an essential component in their commercial operations. The core layers of this category are the application layer and the physical layer. The IoT devices and the functions of the application define the nature of the IoT-blockchain application. The blockchain layer and the network layer are responsible for storage and communication. Some IoT-blockchain applications with a simple architecture do not include the middleware layer.

Dorri *et al.* introduce LSB blockchain structure that emphasizes the security and privacy in a smart home [Dorri et al. 2017b]. The architecture of LSB is shown in Table 3. Physical layer includes

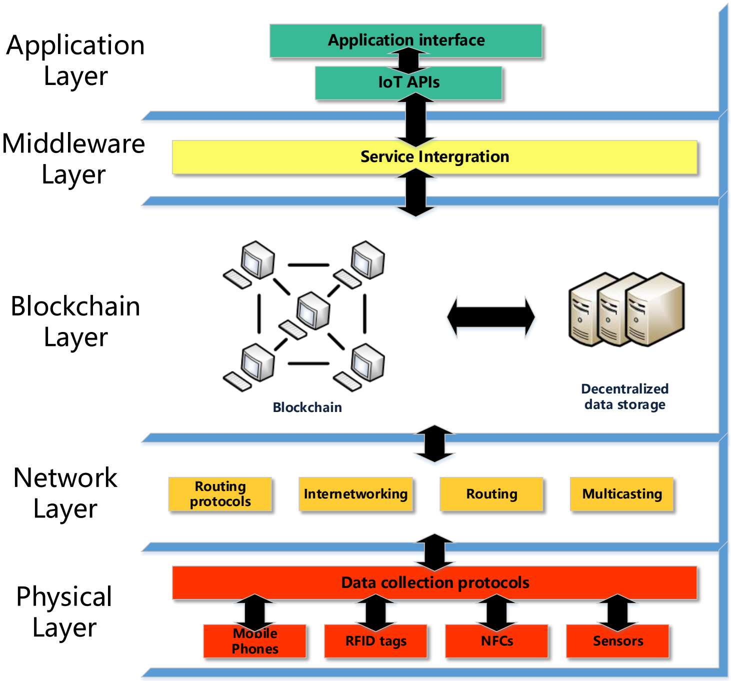


Fig. 4. 5-layer architecture for an IoT blockchain.

various types of Smart Devices. High resource smart devices are responsible for managing a public blockchain to ensure user privacy and security. Low smart resource IoT devices are responsible for end-to-end communication and processing entering and outgoing requests. The smart devices are registered on the blockchain network. Middleware layer provides overlaid blockchain management that can reduce the management overhead of the blockchain. Smart home applications achieve home automation.

LO3 Energy [LO3 2018] introduces a P2P marketplace for solar energy. The architecture of LO3 is shown in Table 3. The physical layer of LO3 includes electrical grid and solar panels. They itemize their extra energy yield and upload it to blockchain through a low latency network. Exergy token system is used as a middleware to enhance the system performance and the efficiency of network participants' authorization. The customer can purchase energy by commercial energy shopping applications.

"Slock.it" [Prisco 2016] is an existing IoT-blockchain application in the market. It operates with electronic locks that can unlock by an appropriate token. The client who wishes to sell their properties can set a price on an electronic lock. The customer can browse the goods and pay the requested amount in cryptocurrencies to unlock the lock. The architecture of "Slock.it" is shown in Table 3. It has a simple architecture that consists of distributed applications, Ethereum blockchain, commercial network, and electronic locks.

3.4.2 Application Platform as a Service. Application platform as a service refers to the support software that connects everything in an IoT-blockchain system. It integrates IoT devices and blockchain technique and provides a straightforward management platform for developers. It facilitates communication, data flow, and device management. The core layer of this category is the middleware layer. The nature of middleware defines the functions of the platform service. In this category, the elements of the application layer and the physical layer are not specified. The blockchain layer and the network layer need to support the functions of the middleware. For example, allow data integrity check in the smart contract.

Sagirlar *et al.* propose a new IoT-blockchain platform "Hybrid-IoT" [Sagirlar et al. 2018]. The platform implements consensus based on PoW and BFT algorithms. The authors define sub-blockchains and inter-blockchains as the area structures of IoT-blockchain. Fig. 5 shows the architecture of hybrid-IoT. In the architecture, two PoW sub-blockchains are connected by a BFT inter-connector framework.

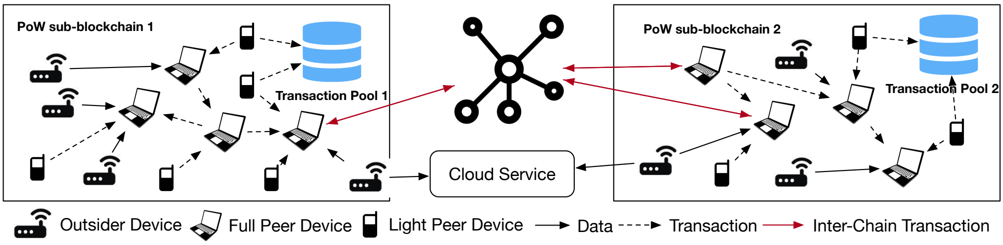


Fig. 5. Architecture of Hybrid-IoT.

Bahga *et al.* introduce a platform for the industrial IoT (BPIIoT) [Bahga and Madisetti 2016]. This platform allows users develop applications (DApps) with the features of blockchain. Its architecture is shown in Table 3. In the platform, IoT devices need to register on blockchain network. Users can develop applications on single-board computers (SBC) to control and manage the blockchain network and IoT devices.

"JD.COM" e-business company publishes a blockchain platform "JD Blockchain Open Platform" [JD 2018] that focuses on providing IoT solutions with the blockchain technology. This blockchain platform provides blockchain gateway service, blockchain node service, and blockchain consensus network service. The architecture of JD Blockchain platform is shown in Table 3. The platform uses BFT-like consensus algorithm. Also, an authentication protocol is used to control the number of accesses to the blockchain network. The system has three kinds of peers: consensus peers, gateway peers, and IoT devices. Gateway peers function in the middleware layer to integrate inputs and protocols from lower layers.

Liu *et al.* [Liu et al. 2017] propose a framework to implement IoT data integrity and security based on a blockchain system. The authors describe critical components of the proposed decentralized data integrity service framework. The framework achieves full decentralization with a data integrity verification protocol. It provides reliable data integrity verification to the users in an IoT system without requiring a third-party auditor. The architecture of this framework is shown in Table 3. In the framework, the IoT device is responsible for generating data and writing data to Ethereum. The user can check the data integrity provided by data integrity service through a data user application.

Alphand *et al.* [Alphand et al. 2018] propose an architecture "IoTChain" which combines OSCAR architecture [Vučinić et al. 2015] and ACE authorization framework [Seitz et al. 2017]. In IoTChain, each registered user has an authorized token. It identifies the particular privilege of a set of resources. When a user wishes to access an object, the user needs to send a transaction with the required data to the smart contract address. Then the smart contract will generate an authorized token for the data user. This architecture uses blockchain instead of the centralized ACE authorization

Table 2. Comparison between IoT-Blockchain Applications

IoT-Blockchain	Service	Blockchain	Consensus	IoT devices	Company size
Filament [Filament 2018]	Transaction service to embedded IoT	Hardware-based Consortium Blockchain	PoW	Blocklet USB Enclave, Blocklet Chip	40 milions market cap
Xage [Xage 2018]	Security service	Fabric	Fabric consensus	Broker, Enforcement Point	300 milions market cap
UniquID [UniquID 2018]	Integrated service to IoT and blockchain	Litecoin	PoW	Sensors, Actuators, Appliances	Open source project
LeewayHertz [LeewayHertz 2019]	IoT-blockchain solutions	Public blockchain	PoW	Robots, Audio devices	More than 10 years in operations
ElectriCChain [ElectriCChain 2018]	Process data of solar panel	SolarCoin	PoS	Solar panel	Open source project
Atonomi [Atonomi 2019]	IoT-blockchain solutions	Atonomi	Atonomi consensus	Smart devices, Smart home	Leading provider of IoT data security
LO3 Energy [LO3 2018]	Solar energy marketplace	Public blockchain solution	PoW	Grid Edge, Solar plane	1 million in revenue annually
Slock.it [sloct.it 2018]	Commission shop	Ethereum	PoW	Electronic lock	1.5 millions in revenue annually
JD.com [JDChain 2019]	Blockchain platform	BFT blockchain	BFT	IoT devices	1.7 trillions market cap

Table 3. Summary of IoT-Blockchain Architectures

IoT Blockchain	Application Layer	Middleware Layer	Blockchain Layer	Network Layer	Physical Layer
Smart home	Smart home application	Overlay blockchain management	Commercial blockchain	P2P network	Smart device
LO3 Energy	Energy shopping application	Exergy token system	Public blockchain solution	low latency network	Grid Edge, Solar plane
Slock.it	DApp	None	Ethereum	Commercial network	Electronic locks
Hybrid-IoT	IoT application	Hybrid-IoT platform	PoW blockchain, BFT blockchain	P2P network	Full peer, Light peer, Sensor
BPIIoT	Manufacturing DApps	Single-board computers	Blockchain network bridge	P2P network	Industrial IoT device
JD.COM	JD.com	Blockchain gateway service	BFT blockchain	P2P network	IoT devices
IoT data Service Framework	Data user application	Data integrity service framework	Ethereum	P2P network	IoT devices
IoTChain	Authorized access	OSCAR, ACE framework	Ethereum	Commercial network	IoT devices

server. The architecture of IoTChain is shown in Table 3. In this architecture, the IoT device is responsible for data generation. The data owner is responsible for uploading the data to blockchain. The OSCAR architecture and ACE authorization framework are responsible for ensuring the safety of user data.

4 CONSENSUS

The consensus mechanism helps multiple participants in a network to reach a necessary agreement. Consensus mechanisms should be fault-tolerant in providing reliable services. In the decentralized environment of blockchain, a consensus mechanism is a set of rules that hinges on the contributions of multiple participants to collectively maintain a synchronized state. A superior consensus mechanism should achieve validity and consistency to function appropriately, while efficiency and cost-effectiveness need to be maintained to ensure high performance.

4.1 Common Consensus Mechanisms

In the blockchain, a consensus mechanism is necessary to provide truthful services among participants without mutual trust. To keep locally stored ledgers consistent with others, various consensus mechanisms are implemented for different scenarios. We divide them into four categories based on their implementations.

4.1.1 Byzantine Fault Tolerance Series. Byzantine Fault Tolerance (BFT) [Driscoll et al. 2003] is a solution to solve the problem of reaching agreement in a system where processors can fail in an arbitrary manner, named *Byzantine General Problem* [Lamport et al. 1982]. BFT is a replica-based approach that exploits communications between replicas to achieve an agreement on request without restrictions or presumptions.

The Practical Byzantine Fault Tolerance (PBFT) [Castro et al. 1999] is based on the BFT mechanism. It improves system performance by reducing the complexity of *Classic Byzantine General Problem* to polynomial level and enabling it to function correctly in asynchronous systems [Castro et al. 1999]. PBFT network should consist of at minimum $3f + 1$ nodes in order to tolerate f defective nodes. In a blockchain system, it requires $2f + 1$ nodes to gain the consensus of the block of transactions.

HQ replication [Cowling et al. 2006] is a protocol that optimized the PBFT. It reduces the amount of information disseminated, making PBFT more efficient. HQ replication does not fundamentally change the structure of PBFT, mainly in the case where the client does not become involved in a competition when sending a request. It classifies the operations as *read* and *writes*. Requests from clients are normally *write* operations. In process of *write*, it is divided into two phases: Phase 1, the client first sends write request to a node, collecting the status information of the server at the same time. If $2f + 1$ nodes are in the same state and agree to execute the client's request, the client initiates the second phase, phase 2, causing the node to execute the client request; otherwise, it means the request encounters the competition. In this case, nodes will execute PBFT to reach a consensus, then the leader node will finally determine the execution order of the request.

Specular Byzantine Fault Tolerance protocol (SBFT) is also a vital part of BFT protocols. A typical SBFT protocol is introduced in Zyzzyva [Kotla et al. 2007], which significantly improves PBFT efficiency. The main idea is that nodes in a Byzantine system are in a normal state most of the time. Thus, clients need not wait for each request to be executed after consistency is reached. After receiving client's request, a node will not perform the costly pairwise interaction as in the PBFT, but instead directly executes the client's request. Nodes are only supposed to gain consistency when an error occurs. In this case, the response time to a request will be greatly accelerated.

Robust Byzantine Fault Tolerance protocol (RBFT) [Corp 2019] is actually a series of protocols that strengthen the robustness of traditional PBFT. In other BFT protocols mentioned above, researchers try to enhance the efficiency of the BFT system under normal circumstances. Therefore, when a Byzantine fault is encountered, the performance of such protocols decreases dramatically. It is even difficult to ensure the liveness of the system as a whole. Through improvement of the PBFT algorithm, RBFT ensures that the system performance roughly the same in the best or worst case conditions, thus providing a truly practical BFT.

However, the BFT series consensus protocol still have some problems. For example, most BFT-based blockchain systems are permissioned blockchain so that people cannot dynamically join or exit a blockchain system. Accordingly, it is difficult to deploy a permissionless BFT blockchain system.

4.1.2 Proof-of-Somethings (PoX) Series. We use Proof-of-Somethings to represent a series of consensus protocols that achieve consensus by demanding the devotion of computational resources from every node in the entire network. Compared with the BFT protocol and its variations, PoX algorithms adopt specific probabilistic mechanisms that do not require full information about the node operations in the system. It is based on the assumption that there are always more benign nodes than malicious nodes in the network. This means that compared with malicious nodes, benign nodes always have more resources. Because PoX algorithms need not require all the information from participant nodes, systems using PoX set no limits for nodes to join or exit.

Proof-of-work (PoW) was proposed by Nakamoto *et al.* in 2008 [Nakamoto 2008]. It was initially designed to reduce jam emails and then employed by the Bitcoin. In Bitcoin, the PoW consensus protocol requires nodes to solve a computationally complicated math puzzle. The mathematical problem serves as a means to verify the validity of blocks on the network before accepting it.

Solving this puzzle requires a large amount of computational power for every node. It acts as a mandatory regulation for every node to obey, thus building trust among all nodes. The blocks chained using PoW also become extremely resistant to tampering, because of the extremely high computational cost it requires. However, a blockchain system running PoW may also be attacked in the mining process where an attakcer can get more profit by launching mining attacks and bribery attacks [Gao et al. 2019]. A refined version of PoW is introduced in [Dwork and Naor 1992], which proposes a probabilistic solution to the Byzantine General Problem. It also ensures the stability of blockchain systems in environments with a large number of untrusted nodes.

Proof-of-Stake (PoS) [Kiayias et al. 2017] relies on a hypothesis that users holding more currency are more incentived to ensure the reliability of the system, and is less likely to play as malicious nodes. For pure PoS [Azima 2018], users holding more coins over a long period have a higher possibility of being selected by the system to generate the next block. In DPoS [Larimer 2014], people will elect several super nodes according to the amount of currency of each node. News blocks will be generated by the selected super nodes. This approach decrease the decentralization of the blockchain system so that it requires new nodes trust these super nodes.

Proof-of-Capacity (PoC) [Dziembowski et al. 2015], also called Proof-of-Space (PoSpace), exploits the idle space in local computers' disk for mining. PoC is similar to the PoW consensus protocol, except that instead of devoting computation resource, space in disks is utilized. In PoC, all the nodes store the solutions for a series of computation puzzels, which are hard to find but easy to verify. If a node has a calculation method on its disk, which is the fastest solution corresponding to the puzzle in the latest block, this node will have the right to issue a new block and get the mining rewards. One well-known application of the PoC consensus protocol is InterPlanetary File System (IPFS) [IPFS 2019], a peer-to-peer file system based on the distributed hash table. IPFS provides an incentive system with the built-in cryptocurrency FileCoin, which can be mined by PoC consensus protocol. In the IPFS, all the files are given a unique fingerprint, which is a cryptographic hash. Each network node only stores the content that it is interested in. A user, who wants to find a file, needs to find the nodes that are storing the required files.

The Proof-of-Authority (PoA) [Naumoff 2016] consensus protocol was proposed to relieve the requirement of computational power or amount of coins. Instead, POA introduces a new concept called *authority*. Different from PoW and PoS, only nodes with authority are allowed to generate new blocks in systems using PoA and thus ensuring the security of the whole chain. It is more flexible with less computational overhead compared with other PoX when applied to private systems.

Proof-of-Importance (PoI) [NEM 2018] is a consensus mechanism developed to detemine which network participants are eligible to add a block to the blockchain. Each account will be evaluated and given a importance mark. The importance depends on the account's transaction amount, transaction parterner number and vested stake. According to the importance score, you can not only get the corresponding reward, but also do not need to use any computing resources of your computer, even when you are offline.

Proof-of-Burn (PoB) [Frankenfield 2018] is a consensus protocol that miners should show proof that they have burned some coins, which is sent these coins to some unspendable addresses. The amount of these destroyed coins determines the probability of a miners to issue a new block. This consensus protocol does not need massive useless computation so that it is environmental.

4.1.3 DAG Series. Directed Acyclic Graph (DAG) [Benčić and Žarko 2018] was proposed [Lerner 2015] to improve the parallelism of traditional single-chain blockchain system. DAG employs the directed graph data structure to connect blocks, and its related consensus mechanism is distinct from others. Each transaction will be linked with the previous two transaction records. In this case, the legality of the current transaction can be proven by referencing previous transactions. In this way, the legality of all transactions can be confirmed.

Compared with other PoX consensus protocols, DAG only cares about countable linked transactions. In addition, it is more than simple PoX consensus protocol. Considering the mining process, DAG employs the PoW to mine a new block. However, DAG blocks are not chained in a linked list but a tree. Each node only checks the legitimacy of transactions it is associated with. This mechanism enhances the efficiency and availability with a sacrifice in partition tolerance. Byteball [Churyumov 2016] introduces a decentralized system that adopts DAG as a consensus mechanism. In this system, everyone is allowed to add new data into the database by paying a fee proportional to the data size. Next nodes will confirm the previous data by including their hash into current data. Moreover, the previous node will receive confirmations from the next nodes.

4.1.4 Ripple Series. There are two kinds of nodes in the Ripple network [Pilkington 2016]: server node and client node. A Ripple server maintains a Unique Node List (UNL) for the consensus process, then votes on the authenticity of transactions. If the consensus agreements are more than 80%, then all transactions that recognized legal are included into the blockchain. Then the ledger is closed as a new last-closed ledger. In contrast to those PBFS nodes that need to query all nodes in the network, a server in Ripple only queries a minimum percentage of 80% of its UNL.

4.2 Comparison between Consensus Mechanisms

We summarize the characteristics of some widely-deployed consensus mechanisms [Baliga 2017; Castro et al. 1999; Chalaemwongwan and Kurutach 2018; Cowling et al. 2006; Dziembowski et al. 2015; Frankenfield 2018; King and Nadal 2012; Kozak 2018; Lerner 2015; NEM 2018; Salimitari and Chatterjee 2018; Zheng et al. 2017] in Table 4, where we compare the blockchain type, throughput, system scalability, transaction finality, tolerated power of adversary, advantage, disadvantage, vulnerability, and example of use. From the table, we can see that the consensus mechanism of PoS, DAG and Ripple have high transaction speed and system scalability. PoS has the best tolerated power of adversary. However, the tolerated power of DAG will be higher in practice, because there are coordinator nodes in the system.

In a small scale network, BFT consensus mechanism can provide high throughput and security. The performance of BFT drops as the number of nodes grows, because of the additional communication cost between the new and existing nodes. Additionally, BFT has advantages of extensibility that it can combine with various kinds of improvement algorithms to meet specific needs.

PoX mechanism is suitable for the public chain because the requirement of computational power or currency is useful in preventing denial of service attacks, service abuse and making the chain more secure and reliable. However, PoX also suffers from inefficiency and high computational overhead.

DAG is a new trend in blockchain systems, because it can achieve a high throughput of BFT-like consensus under the premise of obtaining the flexibility of the PoX series algorithm. It combines the advantages of BFT and PoX such as high efficiency, low computational overhead, and high throughput. However, the application of DAG is not mature. More research is required to substantiate its practical value.

Different consensus protocols have different advantages, disadvantages, performance, and security levels. In this way, different consensus protocols are applicable to different scenarios. Consensus protocol such as PoW has high security levels and so can be deployed in public blockchain systems. In addition, consensus protocol such as PBFT, which can achieve a high throughput, can only be deployed in the permissioned blockchain systems since the limitation of its participant nodes. Therefore, we provide a thorough comparison between different consensus protocols.

Table 4. Comparison between consensus protocols

Name	Type	Throughput	Scalability	Finality	Adversary Tolerance	Advantage	Disadvantage	Vulnerability	Application
BFT	Permissioned	High	Low	Deterministic	33.3% Replicas	Low Transaction Cost Instant Block Finality	Communication Overhead Centralization	33% Attack	Tendermint
PBFT	Permissioned	High	Low	Deterministic	33.3% Faulty Replicas	High Throughput Instant Block Finality	Communication Overhead Centralization	33% Attack	Hyperledger
PoW	Permissionless	Low	Low	Probabilistic	50% Computing Power	Free to Join Adaptive Consensus	Low Throughput Waste Energy High Fork Rate	Selfish Mining	Bitcoin
PoS	Permissionless	Low	Low	Probabilistic	50% Stake	Energy Efficient Rolling Committee	Communication Overhead Matthew Effect	Long Range Attack	Peercoin
PoC	Permissionless	Low	Low	Probabilistic	50% Space	Energy Efficient	Waste Disk Space	Selfish Mining	IPFS
PoA	Permissionless	Low	High	Probabilistic	50% of Online Stake	Energy Efficient	Trust Requirement Limited Application Scenarios	Single Point Failure	Decred
PoI	Permissionless	Low	Low	Probabilistic	50% Stake	Less Chance of Hoarding	Trust Requirement	Single Point Failure	NEM
PoB	Permissionless	Low	Low	Probabilistic	50% Coins	Long-Term Incentive	Low Confirmation Latency	Denial-of-Speding Attack	XCP
DAG	Permissioned	High	High	Probabilistic	33.3% Computing Power	High Throughput	Communication Overhead	Sybil Attack	IOTA
Ripple	Permissionless	High	High	Deterministic	20% Faulty Nodes	Energy Efficient Fast Block Finality	Trust Requirement	Single Point Failure	Ripple

The BFT series consensus protocols can only be deployed in the permissioned blockchain system so that it cannot be directly applied to the public chain. However, the BFT series consensus protocols can achieve a high throughput compared to some other public chain consensus protocols such as the Proof-of-Work (PoW) protocol.

The PoX series consensus protocols can be applied to the public chain. However, it cannot meet the commercial usage performance demand. Moreover, most PoX series consensus protocols are probabilistic so that the fork problem exists. However, a deterministic consensus protocol can alleviate the blockchain system from the fork problem.

As to the performance, even with the same consensus protocol, slightly changing some parameters may result in different performance. Accordingly, we talk about the performance of some specific consensus protocols. The classical Nakamoto consensus could provide a 7 tx/s processing rate. In Ethereum, the speed is 15 tx/s. As to the PoS protocol, Ouroboros system's throughput can reach 256 tx/s while Snow-White can reach 110 tx/s. Moreover, in some hybrid consensus protocols, Algorand can provide a 90 tx/s transaction processing rate. The Hyperledger can reach 110 tx/s speed.

Considering the security levels of these protocols, most protocols do not provide a formal security analysis of their protocols. This is because most of the agreements are just engineering implementations rather than theoretical constructs. Specifically, Ouroboros provides a formal security analysis and proves their protocol is secure and can reach a δ-Nash Equilibrium.

These consensus mechanisms also have some limitations. For example, the usage of PoW, DAG, and PoC is limited by the total amount of computing power/space in the system, which means that a system without enough computation/storage resource is vulnerable to 51% attack. As to the BFT and PBFT, they can provide a high-throughput service for a few nodes. However, when the number of nodes increases, it is hard to maintain the transaction processing rate because of the heavy communication overhead. For PoS consensus protocol, there are situations where rich people are getting richer, and poor people are getting poorer. This is because rich people have a greater chance of releasing a new block, which in turn will bring them more benefits. Therefore, the PoS consensus protocol needs a sustainable incentive mechanism, which can alleviate the PoS protocol from centralization.

4.3 Consensus Characteristic in IoT Blockchain

In Table 2, the majority of IoT-blockchain application use PoW as their consensus mechanism. However, in considering the compatibility between the limitations of IoT devices and consensus mechanisms, PoW and the above common consensus mechanisms, which are the backbone for reaching the global consensus, may not be perfect. Because the IoT devices have limited computing capability, energy, and storage capacity, at the same time, the proof of work in the decentralized network must be done to achieve validity and consistency. A suitable consensus mechanism for IoT blockchain should achieve a balance between energy consumption, system performance, and security. We summarize the characteristics of IoT blockchain as high energy efficiency and lightweight consensus mechanism.

4.3.1 High Energy Efficiency. IoT blockchain needs to execute a consensus process efficiently. Considering the limitations in storage and computing capability of IoT devices, light clients or no miners are adopted for IoT blockchain. IOTA adopts DAG consensus: Each node in the network who wants to generate transactions must actively participate in consensus process by approving two previous transactions. Moreover, there are no miners in IOTA, which is more energy efficient.

Ambrosus [Ambrosus 2019] uses IoT sensors to collect client supply chain data, which is then authenticated by Proof-of-Authority (PoA) consensus protocol. This protocol confers authentication capabilities to the most trusted and high-integrity master nodes.

Helium [Haleem et al. 2018] adopts Honey Badger BFT which offers a perfect balance between compromise and results. First, there are no timing assumptions: Honey Badger BFT assumes that messages in a network eventually get delivered, which is usually an issue in a network. Second, there is censorship resistance, which means that players cannot pick and choose whom to support so that they can always win. In mining terms, it implies that miners cannot look into transactions before they are published. Third, it is a permission-less consensus, so that anyone can join the network.

4.3.2 Lightweight Mechanism. Atonomi [Atonomi 2019] partially adopts Ethereum network, even though its transaction processing might become costly or speed-prohibitive. Adopting Ethereum as its network means that Atonomi can use the well-deployed features of Ethereum immediately. To maintain a healthy ecosystem, Atonomi can evaluate migration or simultaneous use of alternative blockchain technologies for these transactions.

IoT Chain (ITC) [Database 2018] combines several technologies, such as Verifiable Virtual Routing Forwarding (VRF), Practical Byzantine Fault Tolerance (PBFT), Directed Acyclic Graph (DAG), and Simplified Payment Verification (SPV), to build a light operating system based on the blockchain. The data in IoT Chain are stored in different node. To provide a high transaction processing rate, ITC employs a hybrid consensus mechanism, PBFT and DAG. Specifically, the PBFT works for the main chain and DAG works for the side chains.

Waltonchain [WaltonChain 2018] aims to provide a blockchain-based complete data exchange and absolute information transparency for IoT. It adopts a parent-child chain architecture. Its consensus mechanism is called WPoC (Waltonchain Proof of Contribution), which combines PoW, PoS, and PoL (Proof-of-Labor). The parent chain uses the combination of PoW and PoS while the data exchange between parent and child chains are processed by master node using the PoL.

Blockcloud [Ming et al. 2018] proposes a service-centric blockchain architecture to empower IoT. It also designs a new consensus protocol called proof-of-service to provide decentralized trust and security service. Proof-of-service is a hybrid consensus protocol, which combines the advantages of both permissioned- and permissionless- protocol. It relies on the permissionless protocol to achieve a Byzantine agreement and uses the permissioned protocol to improve the performance.

5 TRAFFIC MODEL OF IOT BLOCKCHAIN

In this section, we summarize traffic models of existing distributed systems, and provide our analysis and traffic models of IoT blockchains.

5.1 Current Traffic Models

Traffic modeling work has been carried out from the Internet [Wu et al. 2018; Zhang et al. 2015, 2012, 2013] to IoT networks [Al-Turjman 2018; Al-Turjman et al. 2017; Masek et al. 2016]. In the existing work of blockchain traffic models, the majority are based on peer-to-peer (P2P) traffic models since the message exchange, message validation, and transaction process of blockchain are all handled by peer nodes as in a P2P network. We will first summarize the existing work on P2P traffic modeling and then illustrate the related work of blockchain traffic modeling.

5.1.1 P2P Network Modeling. A peer-to-peer (P2P) network consists of connected devices that share resources between each others without centralized management. BitTorrent is a popular application based on the P2P network. Qiu *et al.* [Qiu and Srikant 2004] introduce a fluid model to analyze the performance of the BitTorrent system. The modeling results give insight into how network performance is affected by different parameters. Kazaa is another famous P2P application. Gummadi *et al.* [Gummadi et al. 2003] analyze Kazaa P2P file-sharing traffic in order to investigate the nature of file-sharing workloads. Hefeeda *et al.* [Hefeeda and Saleh 2008] carry out an eight-months traffic measurement to study the cache-related P2P traffic characteristics. The results show that the traffic pattern of a P2P network follows a Mandelbrot-Zipf distribution. The distribution shows the probability $p(i)$ of a user accessing a resource at a popular order i out of N accessible resources is:

$$p(i) = \frac{K}{(i+q)^{\alpha}}, \tag{1}$$

where $K = 1/(\sum_{i=1}^{N} 1/(i+q)^{\alpha})$, with plateau factor q and skewness factor α. q defines the degree of smoothing of the distribution head. The smoothed distribution head represents the most desired resources. These objects have characteristic of immutable and large file size, and they are suitable for storing in the cache. When $q = 0$, the probability will follow a Zipf-like distribution with the skewness factor α.

In addition, Li *et al.* [Li et al. 2011b] establish a P2P network traffic recognition model based on the multi-dimensional support vector machine (MSVM). The model uses the MSVM as a classifier to identify P2P traffic and Non-P2P traffic, and conduct feature extraction, data processing, and MSVM training.

5.1.2 Blockchain Related Traffic Modeling. With the enormous growth in the field of blockchain technology, there are different communication models. They have different traffic models and vary in terms of algorithm, processing power, time and storage approach. We categorize them into three types: Gossip protocol, Kademlia algorithm and Direct Acyclic Graph (DAG).

Gossip protocol

Gossip is a widely used protocol designed for P2P communications in distributed systems including a large portion of current blockchain systems. In the gossip protocol, every IoT device always chooses one or several random targets in each round before sending messages (namely gossips). Eventually, all IoT devices receive the gossip twice in a round of data transmission. This ensures data integrity even when the message sent is corrupted (by comparing the two received messages). Therefore, the gossip protocol can serve as a robust and reliable message delivery mechanism for general IoT systems.

However, the performance of gossip protocol suffers from the large message overhead in the communication process. Fraigniaud *et al.* [Fraigniaud and Giakkoupis 2010] infer that when message size is ρ, the time complexity of gossip protocol is $O(logn)$ and the message bit complexity of it is $\Omega(n(\rho + log(logn))$. Jenkins *et al.* [Jenkins et al. 2001] show that gossip overhead rises considerably as the number of nodes in the network grows. For the p^{th} node N_p with *infectivity* $I(N_p)$ and *susceptibility* $S(N_p)$ in a subgroup g, the expected number of messages sent from N_p to the subgroup g in a round is:

$$\sum_{N_p \in g} I(N_p)(\sum_{N_q} S(N_p)). \tag{2}$$

And the expected numbers of messages received by N_p from the subgroup g in a round is

$$\sum_{N_p \in g} S(N_p)(\sum_{N_q} I(N_p)). \tag{3}$$

Kademlia algorithm In Kademlia, every node is assigned a random and unique ID before joining the network. Each ID consists of only either 1 or 0 and has a fixed length. Kademlia uses IDs to calculates the logical distances between any two IoT devices by applying exclusive or (XOR) operation on node IDs. Every node owns a random neighbor list, in which the IDs of nodes are grouped into K-buckets based on the common prefixes. The K-buckets only stores the active nodes or records. Compared with other methods (such as gossip), the approach used by Kademlia largely reduces the computational overhead. For instance, when one node wishes to communicate with other nodes, the message will only be relayed no more than n times to reach the target node in a network with 2^n nodes. Maymounkov *et al.* [Maymounkov and Mazieres 2002] summarize that Kademlia only requires to exchange information with $O(logn)$ nodes during a search in a network of n nodes. Cai *et al.* [Cai and Devroye 2013] give a mathematical proof to support this conclusion. Consider a Kademlia network of n nodes with IDs from $x_1, ..., x_n$ and all IDs have the same length d. For a node $x \in x_1, ..., x_n$, let $D_i(x)$ be the set of nodes that share the same prefix with x with prefix length $d - i$. The i^{th} bucket of this node contains k IDs randomly selected from $D_i(x)$. In other words, filling $i - th$ bucket of node x can be considered to be adding pointers from the leaf x to k leaves chosen randomly from $D_i(x)$. Let T_{xy} be the number of attempts needed by Kademlia to search from leaf x to target ID y. Assume that $x_1, ..., x_n$ are chosen deterministically from $\{0, 1\}^d$. It can be proved that

$$\sup_{x_1,\ldots,x_n} \sup_{x \in \{x_1,\ldots,x_n\}} \sup_{y \in \{0,1\}^d} \mathbb{E}[T_{xy}] \leq (1 + o(1)) \frac{\log n}{H_k}. \tag{4}$$

In Eq.4, H_k is the k^{th} Harmonic Number. Since $H_k/\log k \longrightarrow \infty$, when $E[T_{xy}]$ is bounded by $log_k n$. This justifies that in Kademlia only $O(logn)$ nodes are needed to contact with to search for a target node.

The experiment on the performance of Kademlia is shown in [Brunner and Biersack 2006]. The result shows that there is 60% probability that the Kademlia system can search and find the target data within 15 seconds after a user publishes new data. Also, there is 100% probability that the system can obtain the target data within 45 seconds. Another experiment evaluation of Kademlia is shown in [Ou et al. 2010]. The result shows the Kademlia system is resilient against churns and data corruption. Also, the paper concludes that the Kademlia system's power consumption is low with decent efficiency.

Direct Acyclic Graph (DAG) Directed Acyclic Graph (DAG) is a technique used in distributed ledger technology. On a DAG network, the transactions are verified on a P2P basis. Each node needs to verify the transactions of at least two previous nodes in order to perform a transaction. And the ordered sequence of the verification events is called *DirectedGraph*.

Compared with the gossip protocol, DAG has no ring structure and it is unidirectional. It means the performance of resource locating in DAG is guaranteed with a deterministic searching path.

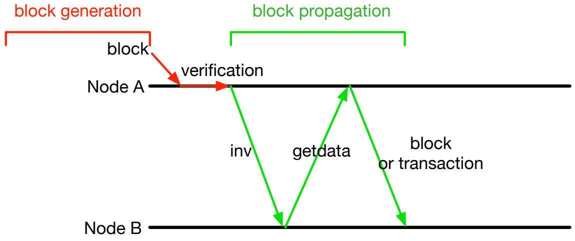


Fig. 6. Block generation and block propagation in blockchain.

DAG achieves higher search efficiency and computation compared with other communication protocols that contain ring structure or non-directional structures.

DAG is capable of connecting thousands of IoT devices in an IoT system. Each device in the system will be assigned a unique label for rapid positioning and communication. Combinatorial enumeration [Robinson 1977] illustrates the DAG labeling issue. The numbers on n labeled nodes, for $n = 0, 1, 2, 3, \ldots$ from the On-Line Encyclopedia of Integer Sequences are $1, 1, 3, 25, 543, 29281, 3781503, 1138779265, \ldots$

Tangle [Tangle IOTA 2018] is an implementation of DAG. In Tangle, IoT devices are connected with one another with a specified order and direction. When a device wishes to contact another device, it can send a message directly to the target host. All devices in the network will be simultaneously updated. Tangle achieve a much higher overall throughput by allowing different branches of the DAG to merge. Popov *et al.* [Popov 2016] analyze the mathematical foundations of IoTA, Tangle and DAG. The paper infers that the total number of unapproved transactions in Tangle L in a certain period of time can be calculated as

$$L_0 = 2\lambda h. \tag{5}$$

In the equation, λ is the expected event rate of Poisson distribution, and h is the average time that a device needs to issue a transaction.

Benčić *et al.* [Benčić and Žarko 2018] compare DAG with traditional blockchain approaches. They show that DAG has higher efficiency and scalability than traditional blockchain approaches.

5.2 Proposed Traffic Models

5.2.1 Blockchain Traffic Models. The traffic of blockchain can be categorized mainly into two sections, block generation, and block propagation. Fig. 6 shows the message flow of block generation and block propagation with the block generation process taking place before Node A receives a block. After Node A receives a block, it verifies the block and sends it to neighbors. The block propagation process takes place in turn.

In the block generation, based on Proof-of-Work [Decker and Wattenhofer 2013], a node needs to calculate a value called nonce to publish a block. The nonce is a hash value included in the block header. The node needs to try a large number of times to find a valid nonce. If the current target of the nonce is 400 billion, a node will have 400 billion/2^{256} chances to solve the nonce and publish a block. Blocks are generated around the world through the global computing power of nodes. The difficulty of finding a valid nonce is dynamically adjusted according to the latest block generation interval to keep the block generation interval remains relatively unchanged. On average, in bitcoin network, a block is produced in every 10 minutes [Beccuti et al. 2017]. In Ethereum, the average block produce time is 15 seconds. The difficulty of the mathematical problem increases over time as the global computing power of miners increase. The expected time between blocks remains 10 minutes. It shows that the probability of generating a block satisfies the Poisson distribution. Let λ_b be the success rate of generating a block, t_c be a certain time interval, and N be the generated blocks. The probability of generating N blocks in time t_c follows a Poisson distribution:

$$P_g = \int_0^\infty \frac{(\lambda_b t_c)^N}{N!} e^{-\lambda_b t_c} dt_c. \tag{6}$$

In block propagation, blocks and transactions are continuously propagated in the blockchain network for updates and synchronization. To avoid message duplication, an *inv* message is sent first [Decker and Wattenhofer 2013]. If a node receives an *inv* message that contains a not received block hash or transaction hash in its inventory, it will respond with a *getdata* message. The sender of *inv* message will transfer the block or transaction to the node after getting the *getdata* message. On average, a block contains 500 transactions and the size of it is 1MB in Bitcoin [Bowden et al. 2018]. The total size of the *inv* and the *getdata* message is 120B on average. It is negligible when compared to the block size. The blockchain transaction rate can be found on the official blockchain website of Bitcoin. It shows there are 2.73 Bitcoin transactions added to the mempool per second. Let A_g be the average arrival rate of blocks, L_b be the average size of blocks, R_b be the transmission rate of a block, and T_p be the time period. The traffic intensity of block propagation on a single node can be defined as:

$$\frac{A_g L_b}{R_b} T_p. \tag{7}$$

In addition, we also model the traffic distribution of propagating a new block to the entire blockchain network. In different blockchain implementations, the traffic distribution differs with variances in the communication protocols. In this paper, we mainly discuss the traffic distribution of the propagation process in Bitcoins or Bitcoin-like blockchain systems. The gossip protocol is considered when modeling propagation traffic.

Assume in the initial state of blockchain network with $x + y$ nodes, a new message is generated and acknowledged by y nodes. After that, the message should be propagated to the remaining x nodes. A node will only receive the same message once. When a node receives a new message, it will actively propagate the message to b neighbors. We assume every package can be sent and received without error. And we assume the number of nodes remains stable for the entire process. Initially, we have $y = 1$ and $x = n$. So the total number of nodes can also be presented as $1 + n$.

During the process of a message being broadcast, x will decrease while the message y will increase. Based on the gossip protocol, the decreasing rate of x should be $\frac{dx}{dT} = -\beta xy$. β is the contact rate of all the (x, y) pair combination, which can be calculated by $\beta = b/n$. While $x + y = n + 1$, base on the Epidemic Information Dissemination [Eugster et al. 2004] we can further obtain:

$$y = \frac{(n+1)}{1 + ne^{-\beta(n+1)T}}. \tag{8}$$

We define a constant c as $c = \frac{2n}{b(n+1)}$. When $T = c * log(n)$, the number of nodes that have received the new message can be estimated as $y \approx (n+1) - \frac{1}{n^{cb-2}}$. Since $\frac{1}{n^{cb-2}}$ is relatively small, we can assume that the propagation process ends at the time $T = clog(n)$. Since every node is broadcasting the same message to its peers, the sizes of the received messages and sent messages are the same as S_{msg}. During the propagation period, we calculate the traffic due to broadcasting blocks or transactions, since the control messages are trivial compared to the total traffic. The traffic at time T is corresponding to the number of received nodes. It can be calculated as:

$$TRFC(T) = \frac{(n+1)S_{msg}}{1 + ne^{-\beta(n+1)T}}. \tag{9}$$

We simulate the infected nodes over time for the gossip protocol in Fig. 7. The simulation shows a network with N nodes, and each node has h neighbors in its neighbor list. Each node will infect b neighbors randomly in a gossip round. The experiment is conducted with inputs with $b = 5$, $h = 10$ and $N = 2000, 5000, 10000$. In Fig. 7, the horizontal axis shows the number of gossip rounds. We can see that after seven gossip rounds, the first infected node can update 2000 nodes in the network. The traffic can be calculated by the number of infected nodes times the average size of a packet in a specific period of time.

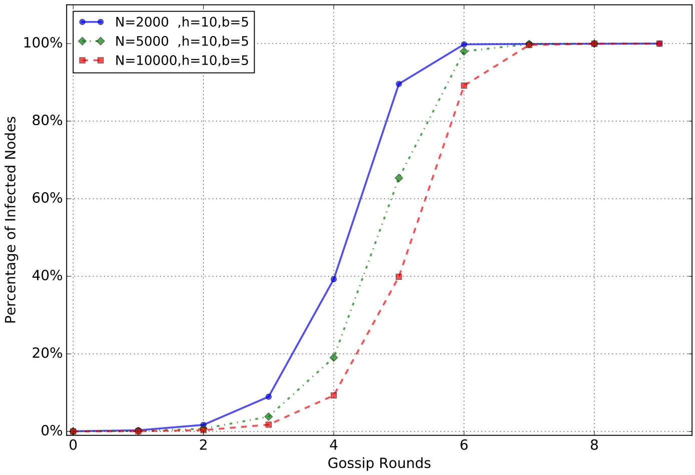


Fig. 7. Percentage of infected nodes versus gossip rounds.

5.2.2 IoT-Blockchain Traffic Model. We divide the traffic model of IoT blockchain into two categories. They are different in the deployments of IoT devices, blockchain nodes, and their connections. This refers to the physical layer, blockchain layer, and network layer of the 5-layer architecture for IoT blockchain we proposed in section 3.4.

IoT Devices in the Blockchain Network Structure

In this structure, the IoT devices act as normal nodes in the blockchain network, as shown in Fig. 8. In the blockchain network, the IoT device nodes are considered as light nodes. They don't need to store a copy of the full blockchain and conduct the mining work.

This structure achieves maximum decentralization, since there is no single central node to manage or monitor the nodes. Overall performance will not be adversely affected even if some IoT nodes are corrupted. Also, there are pre-configured high-performance nodes are acting as the backbone of the blockchain network. Work that requires heavy computation or massive data storage will be assigned as the backbone nodes. Therefore, malfunctioning of the IoT device nodes will not affect blockchain performance.

The rate of generating a block of traffic in this structure will follow Eq. 6. The traffic of nodes in this structure can be calculated by Eq. 9.

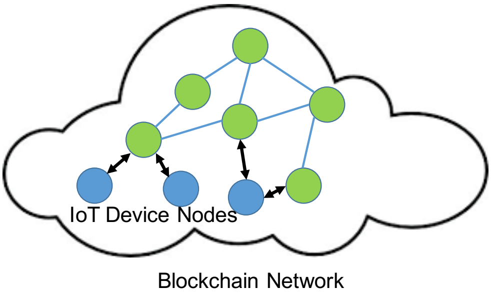


Fig. 8. Structure of IoT devices in a blockchain network.

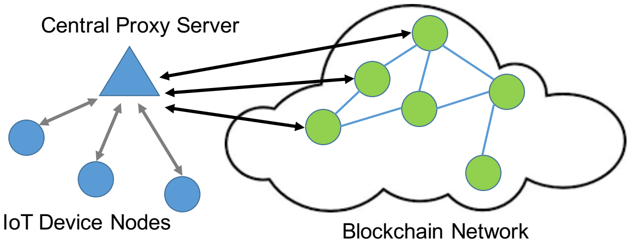


Fig. 9. Structure of IoT devices outside of a blockchain network.

IoT Devices Outside of the Blockchain Network Structure

In this structure, the blockchain network is considered as a separated network. A central base station is placed to interact with the IoT device nodes directly. The requests sent by IoT devices nodes will be processed by the base station and delivered to the blockchain network. When the blockchain network confirmed the transaction, the IoT device will be notified. This structure is illustrated in Fig. 9.

The benefit of this structure comes from its convenient configuration and maintenance. Since the IoT device nodes are not directly connected to the blockchain network, the proxy server can act as a traffic regulator. It becomes easier to control traffic and provide efficient transactions by caching and reducing the possibility of data deluge. Compared to the structure of IoT devices in a blockchain network, this structure allows the IoT system to utilize the functions of blockchain without maintenance costs and configuration overhead of blockchain. The traffic model of the IoT network depends on the types of IoT services. For the blockchain network, the traffic will follow Eq. 6 and Eq. 9.

6 CONCLUSIONS

Blockchain technology has been extensively applied in diverse services. The Internet of Things (IoT) is getting ubiquitous in modern life. Blockchain has provided a practical solution to address many of the limitations of traditional IoT applications. The integration of the IoT and blockchain creates a verifiable and secure network. This paper presents a comprehensive overview of IoT blockchains, including typical architectures, communication protocols, applications, and traffic models. This paper provides a good summary for researchers or practitioners who are interested in understanding the concepts, functions and traffic models of IoT blockchain. Furthermore, we make comparisons between current consensus algorithms and communication protocols and show their strengths and shortcomings when applied in IoT blockchains. Finally, we analyze the current traffic models of the P2P system and blockchain system. We also give traffic models of IoT blockchain systems.

Considering that the number of IoT-blockchain applications grows each year, we believe that new consensus mechanisms for enhancing the performance of IoT devices in the blockchain network will be good directions in the coming year. Also, the solutions for solving scalability, processing power, or storage problem of IoT device in the blockchain network is an interesting topic. Another interesting future research direction is IoT-blockchain traffic model analysis. However, it may require a large scale of simulation, data collection, and data analysis.

Apart from this, how to ensure security when integrating IoT with blockchain is still a challenging topic because the IoT components do not have enough resources to perform the encryption algorithms in the blockchain. Also, IoT devices are resource restricted to validate transitions and maintain the correctness of the blockchain. Moreover, the access control of keeping the system security while allowing numerous IoT components to join a private network is hard to achieve.

Additionally, regulation and development policies on the IoT-blockchain applications are still missing. There is no instruction or rules to follow. Comprehensive regulation is needed to clarify the rules and guide the integration of IoT systems and blockchain technology.

7 ACKNOWLEDGMENTS

This work was partially supported by the HK PolyU grant P0011609 and HK GRF PolyU 152124/19E.

REFERENCES

Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2347–2376.

Fadi Al-Turjman. 2018. Information-centric framework for the Internet of Things (IoT): Traffic modeling & optimization. *Future Generation Computer Systems* 80 (2018), 63–75.

Fadi Al-Turjman, Enver Ever, and Hadi Zahmatkesh. 2017. Green Femtocells in the IoT Era: Traffic Modeling and Challenges–An Overview. *IEEE Network* 31, 6 (2017), 48–55.

Olivier Alphand, Michele Amoretti, Timothy Claeys, Simone Dall'Asta, Andrzej Duda, Gianluigi Ferrari, Franck Rousseau, Bernard Tourancheau, Luca Veltri, and Francesco Zanichelli. 2018. IoTChain: A blockchain security architecture for the Internet of Things. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. IEEE, Barcelona, Spain, 1–6.

Ambrosus 2019. Ambrosus - Enabling Sensors to Talk to Blockchain. https://ambrosus.com/.

Jeffrey G Andrews, Stefano Buzzi, Wan Choi, Stephen V Hanly, Angel Lozano, Anthony CK Soong, and Jianzhong Charlie Zhang. 2014. What will 5G be? *IEEE Journal on selected areas in communications* 32, 6 (2014), 1065–1082.

Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*. ACM, New York, NY, 30.

Andreas M Antonopoulos. 2014. *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", California, CA.

Giuseppe Ateniese, Michael T Chiaramonte, David Treat, Bernardo Magri, and Daniele Venturi. 2018. Hybrid blockchain. US Patent 9,959,065.

Hany F Atlam, Ahmed Alenezi, Madini O Alassafi, and Gary Wills. 2018. Blockchain with internet of things: Benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications* 10, 6 (2018), 40–48.

Atonomi 2019. Atonomi - Bringing Trust and Security to IoT. Retrieved March 20, 2019 from https://atonomi.io/

Attores 2019. Attores - Smart Contracts as a Service | Blockchain Singapore. Retrieved March 20, 2019 from https://attores.com/

Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Computer networks* 54, 15 (2010), 2787–2805.

H Azima. 2018. NXT White Paper; New Transducer Ltd. *Huntingdon PE18 6ED Harris, N* (2018).

Arshdeep Bahga and Vijay K Madisetti. 2016. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications* 9, 10 (2016), 533.

Arati Baliga. 2017. Understanding blockchain consensus models. In *Persistent.* Persistent Systems Ltd., India.

Juan Beccuti, Christian Jaag, et al. 2017. *The bitcoin mining game: On the optimality of honesty in proof-of-work consensus mechanism.* Technical Report. Swiss Economics.

Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. 2013. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In *Annual Cryptology Conference.* Springer, Berlin, Heidelberg, 90–108.

Federico Matteo Benčić and Ivana Podnar Žarko. 2018. Distributed ledger technology: Blockchain compared to directed acyclic graph. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS).* IEEE, Vienna, Austria, 1569–1570.

Dan Bieler. 2018. Blockchain and the Internet of Things: the IoT Blockchain Opportunity and Challenge. Retrieved February 27, 2019 from https://www.i-scoop.eu/blockchain-distributed-ledger-technology/blockchain-iot/

Carsten Bormann, Mehmet Ersue, and Ari Keranen. 2014. Terminology for constrained-node networks. *Internet Engineering Task Force (IETF): Fremont, CA, USA* (2014), 2070–1721.

Rory Bowden, Holger Paul Keeler, Anthony E Krzesinski, and Peter G Taylor. 2018. Block arrivals in the Bitcoin blockchain. *arXiv preprint arXiv:1801.07447* (2018).

Richard Gendal Brown, James Carlyle, Ian Grigg, and Mike Hearn. 2016. Corda: an introduction. *R3 CEV, August* 1 (2016), 15.

René Brunner and E Biersack. 2006. A performance evaluation of the Kad-protocol. *Institut Eurecom, France* (2006).

Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* (2014).

Christian Cachin. 2016. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Vol. 310. IBM Research, Zurich.

Xing Shi Cai and Luc Devroye. 2013. A probabilistic analysis of Kademlia networks. In *International Symposium on Algorithms and Computation.* Springer, Berlin, Heidelberg, 711–721.

Miguel Castro, Barbara Liskov, et al. 1999. Practical Byzantine fault tolerance. In *OSDI*, Vol. 99. ACM, New Orleans, USA, 173–186.

Nutthakorn Chalaemwongwan and Werasak Kurutach. 2018. State of the art and challenges facing consensus protocols on blockchain. In *2018 International Conference on Information Networking (ICOIN).* IEEE, Chiang Mai, Thailand, 957–962.

Moumena A Chaqfeh and Nader Mohamed. 2012. Challenges in middleware solutions for the internet of things. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on.* IEEE, Denver, CO, USA, 21–26.

Cello Chen. 2018. AlipayHK and GCash Launch Cross-Border Remittance Service Powered by Alipay's Blockchain Technology. Retrieved February 20, 2019 from https://www.businesswire.com/news/home/20180625005561/en/AlipayHK-GCash-Launch-Cross-Border-Remittance-Service-Powered

Deyan Chen and Hong Zhao. 2012. Data security and privacy protection issues in cloud computing. In *2012 International Conference on Computer Science and Electronics Engineering*, Vol. 1. IEEE, Hangzhou, China, 647–651.

Konstantinos Christidis and Michael Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things. *Ieee Access* 4 (2016), 2292–2303.

Anton Churyumov. 2016. Byteball: A Decentralized System for Storage and Transfer of Value. https://byteball.org/Byteball.pdf.

Cisco IoT 2019. Internet of Things Cisco IoT is the bridge to business outcomes. Retrieved March 20, 2019 from https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. 2016. Blockchain for the Internet of Things: A systematic literature review. In *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of.* IEEE, Agadir, Morocco, 1–6.

Hyperchain Corp. 2019. hyperchain white paper. https://hyperchain.readthedocs.io/zhCN/latest/consensus.html.

James Cowling, Daniel Myers, Barbara Liskov, Rodrigo Rodrigues, and Liuba Shrira. 2006. HQ replication: A hybrid quorum protocol for Byzantine fault tolerance. In *Proceedings of the 7th symposium on Operating systems design and implementation.* USENIX Association, Berkeley, CA, 177–190.

Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. 2016. Blockchain technology: Beyond bitcoin. *Applied Innovation* 2, 6-10 (2016), 71.

Li Da Xu, Wu He, and Shancang Li. 2014. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics* 10, 4 (2014), 2233–2243.

Whitepaper Database. 2018. ITC White Paper Complete Version. https://whitepaperdatabase.com/iot-chain-itc-whitepaper/. (2018).

Christian Decker and Roger Wattenhofer. 2013. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on.* IEEE, Trento, Italy, 1–10.

DokChain 2019. DokChain | PokitDok. Retrieved March 12, 2019 from https://pokitdok.com/dokchain/

Ali Dorri, Salil S Kanhere, and Raja Jurdak. 2016. Blockchain in internet of things: challenges and solutions. *arXiv preprint arXiv:1608.05187* (2016).

Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017a. Blockchain for IoT security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, Kona, HI, USA, 618–623.

Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017b. LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy. *arXiv preprint arXiv:1712.02969* (2017).

Kevin Driscoll, Brendan Hall, Håkan Sivencrona, and Phil Zumsteg. 2003. Byzantine fault tolerance, from theory to reality. In *International Conference on Computer Safety, Reliability, and Security*. Springer, Berlin, Heidelberg, 235–248.

Cynthia Dwork and Moni Naor. 1992. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, 139–147.

Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of Space. In *Advances in Cryptology – CRYPTO 2015*. Springer, Berlin, Heidelberg, 585–605.

Ariel Ekblaw, Asaph Azaria, John D Halamka, and Andrew Lippman. 2016. A Case Study for Blockchain in Healthcare:'MedRec' prototype for electronic health records and medical research data. In *Proceedings of IEEE open & big data conference*, Vol. 13. IEEE, WA, USA, 13.

ElectriCChain 2018. ElectriCChain The Solar Energy Blockchain Project for Climate Change and Beyond. Retrieved March 20, 2019 from https://www.electricchain.org/

Patrick T Eugster, Rachid Guerraoui, A-M Kermarrec, and Laurent Massoulié. 2004. Epidemic information dissemination in distributed systems. *Computer* 37, 5 (2004), 60–67.

Factom 2019. Factom | A blockchain innovations company. Retrieved March 20, 2019 from https://www.factom.com/

Filament. 2018. Filament's Industrial Internet of Things Blockchain Solution Wins 2018 IoT Innovator Award. https://globenewswire.com/news-release/2018/09/26/1576581/0/en/Filament-s-Industrial-Internet-of-Things-Blockchain-Solution-Wins-2018-IoT-Innovator-Award.html.

Pierre Fraigniaud and George Giakkoupis. 2010. On the bit communication complexity of randomized rumor spreading. In *Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures*. ACM, New York, NY, 134–143.

Jake Frankenfield. 2018. Proof of Burn. https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.

S. Gao, Z. Li, Z. Peng, and B. Xiao. 2019. Power Adjusting and Bribery Racing: Novel Mining Attacks in the Bitcoin System. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM, New York, NY, 833–850.

Arthur Gervais, Ghassan O Karame, Vedran Capkun, and Srdjan Capkun. 2014. Is bitcoin a decentralized currency? *IEEE security & privacy* 12, 3 (2014), 54–60.

Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.

Krishna P Gummadi, Richard J Dunn, Stefan Saroiu, Steven D Gribble, Henry M Levy, and John Zahorjan. 2003. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. *ACM SIGOPS Operating Systems Review* 37, 5 (2003), 314–329.

Amir Haleem, Andrew Allen, Andrew Thompson, Marc Nijdam, and Rahul Garg. 2018. A Decentralized Machine Network. (2018).

Mohamed Hefeeda and Osama Saleh. 2008. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Transactions on networking* 16, 6 (2008), 1447–1460.

Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang. 2019. Resource Allocation and Consensus on Edge Blockchain in Pervasive Edge Computing Environments. In *2019 IEEE 39Th international conference on distributed computing systems*. IEEE.

HUAWEI IOT 2019. Enterprise IoT Leading IoT, Driving Industry Digital Transformation. Retrieved March 20, 2019 from https://e.huawei.com/en/solutions/technical/iot

Steve Huckle, Rituparna Bhattacharya, Martin White, and Natalia Beloff. 2016. Internet of things, blockchain and shared economy applications. *Procedia computer science* 98 (2016), 461–466.

Seyoung Huh, Sangrae Cho, and Soohyung Kim. 2017. Managing IoT devices using blockchain platform. In *Advanced Communication Technology (ICACT), 2017 19th International Conference on*. IEEE, Bongpyeong, South Korea, 464–467.

IOTA 2019. The Next Generation of Distributed Ledger Technology | IOTA. Retrieved March 12, 2019 from https://www.iota.org/

IPFS 2019. IPFS powers the Distributed Web. https://ipfs.io/.

Reuben Jackson. 2018. Why IoT Needs the Blockchain, and Blockchain needs IoT.

JD. 2018. The JD. http://ledger.jd.com/.

JDChain 2019. JD enterprise blockchain service. Retrieved March 12, 2019 from http://blockchain.jd.com/blockchain_store/pc/index.html#/BlockChainTrace

Kate Jenkins, Ken Hopkinson, and Ken Birman. 2001. A gossip protocol for subgroup multicast. In *Distributed Computing Systems Workshop, 2001 International Conference on*. IEEE, Mesa, AZ, USA, USA, 25–30.

Jiaojiao Jiang, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou. 2016. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys & Tutorials* 19, 1 (2016), 465–481.

Zura Kakushadze and Ronald P Russo Jr. 2018. Blockchain: Data Malls, Coin Economies and Keyless Payments. (2018).

Jiawen Kang, Rong Yu, Xumin Huang, Sabita Maharjan, Yan Zhang, and Ekram Hossain. 2017. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics* 13, 6 (2017), 3154–3164.

Minhaj Ahmad Khan and Khaled Salah. 2018. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems* 82 (2018), 395–411.

Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. 2012. Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE, Islamabad, India, 257–260.

Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, Cham, 357–388.

Sunny King and Scott Nadal. 2012. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 19 (2012).

Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. 2007. Zyzzyva: speculative byzantine fault tolerance. *ACM SIGOPS Operating Systems Review* 41, 6 (2007), 45–58.

Tim Kozak. 2018. Consensus Protocols That Meet Different Business Demands. https://blockchain.intellectsoft.net/blog/consensus-protocols-that-meet-different-business-demands/.

Nir Kshetri. 2017. Can blockchain strengthen the internet of things? *IT Professional* 19, 4 (2017), 68–72.

Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4, 3 (1982), 382–401.

Daniel Larimer. 2014. Delegated proof-of-stake (dpos). *Bitshare whitepaper* (2014).

In Lee and Kyoochun Lee. 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons* 58, 4 (2015), 431–440.

LeewayHertz 2019. Blockchain Development for Startups and Enterprises | USA | UAE. Retrieved March 12, 2019 from https://www.leewayhertz.com/

Sergio Demian Lerner. 2015. DagCoin: a cryptocurrency without blocks. https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf.

Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen, and Xiaodong Lin. 2011a. Smart community: an internet of things application. *IEEE Communications Magazine* 49, 11 (2011).

Xiaoyong Li, Feng Zhou, and Xudong Yang. 2011b. A multi-dimensional trust evaluation model for large-scale P2P computing. *J. Parallel and Distrib. Comput.* 71, 6 (2011), 837–847.

Zhetao Li, Jiawen Kang, Rong Yu, Dongdong Ye, Qingyong Deng, and Yan Zhang. 2017. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE transactions on industrial informatics* 14, 8 (2017), 3690–3700.

Iuon-Chang Lin and Tzu-Chun Liao. 2017. A Survey of Blockchain Security Issues and Challenges. *IJ Network Security* 19, 5 (2017), 653–659.

Erik Linask. 2018. Blockchain Momentum Continues, Will Reach $12 Billion in 2020. In *The Blockchain Domain*.

Bin Liu, Xiao Liang Yu, Shiping Chen, Xiwei Xu, and Liming Zhu. 2017. Blockchain based data integrity service framework for IoT data. In *Web Services (ICWS), 2017 IEEE International Conference on*. IEEE, Honolulu, HI, USA, 468–475.

Liu Liu, Olivier De Vel, Qing-Long Han, Jun Zhang, and Yang Xiang. 2018. Detecting and preventing cyber insider threats: a survey. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1397–1417.

LO3 2018. LO3 Energy The Future of Energy. Retrieved March 12, 2019 from https://lo3energy.com/

Pavel Masek, Jan Masek, Petr Frantik, Radek Fujdiak, Aleksandr Ometov, Jiri Hosek, Sergey Andreev, Petr Mlynek, and Jiri Misurec. 2016. A harmonized perspective on transportation management in smart cities: The novel IoT-driven environment for road traffic modeling. *Sensors* 16, 11 (2016), 1872.

Petar Maymounkov and David Mazieres. 2002. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*. Springer, Berlin, Heidelberg, 53–65.

Dejan S Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. 2002. Peer-to-peer computing.

Zhongxing Ming, Shu Yang, Qi Li, Dan Wang, Mingwei Xu, Ke Xu, and Laizhong Cui. 2018. Blockcloud: A Blockchain-based Service-centric Network Stack. https://www.block-cloud.io/blockcloudtechnicalwhitepaper.pdf. (2018).

Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. 2012. Internet of things: Vision, applications and research challenges. *Ad hoc networks* 10, 7 (2012), 1497–1516.

M.Padma, N.KasiViswanath, and T.Swathi. 2019. Blockchain for IoT Application: Challenges and Issues. *International Journal of Recent Technology and Engineering* 7 (2019), 40–48.

MuleSoft IoT 2019. Solutions for IoT Extend connectivity from your enterprise and the cloud to devices at the edge of your network. Retrieved March 20, 2019 from https://www.mulesoft.com/integration-solutions/api/iot

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

Alicia Naumoff. 2016. Why Blockchain Needs 'Proof of Authority' Instead of 'Proof of Stake'. https://cointelegraph.com/news/why-blockchain-needs-proof-of-authority-instead-of-proof-of-stake.

NEM. 2018. NEM whitepaper. https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf.

Oracle IoT 2019. Accelerate Your Business with the Power of the Internet of Things. Retrieved March 20, 2019 from https://www.oracle.com/solutions/internet-of-things/

Zhonghong Ou, Erkki Harjula, Otso Kassinen, and Mika Ylianttila. 2010. Performance evaluation of a Kademlia-based communication-oriented P2P system under churn. *Computer Networks* 54, 5 (2010), 689–705.

Alfonso Panarello, Nachiket Tapas, Giovanni Merlino, Francesco Longo, and Antonio Puliafito. 2018. Blockchain and IoT Integration: A Systematic Survey. *Sensors* (2018), 2575.

Zhe Peng, Haotian Wu, Bin Xiao, and Songtao Guo. 2019. VQL: Providing Query Efficiency and Data Authenticity in Blockchain Systems. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, Macao, 1–6.

Marc Pilkington. 2016. 11 Blockchain technology: principles and applications. *Research handbook on digital transformations* (2016), 225.

Suporn Pongnumkul, Chaiyaphum Siripanpornchana, and Suttipong Thajchayapong. 2017. Performance analysis of private blockchain platforms in varying workloads. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, Vancouver, BC, Canada, 1–6.

Serguei Popov. 2016. The tangle. *cit. on* (2016), 131.

Giulio Prisco. 2016. Slock. it to Introduce Smart Locks Linked to Smart Ethereum Contracts, Decentralize the Sharing Economy. *Bitcoin Magazine. Nov-2015 [Online]. Available: https://bitcoinmagazine.com/articles/slock-it-to-introduce-smart-locks-linked-to-smart-ethereum-contracts-decentralize-the-sharing-economy-1446746719.* (2016).

Dongyu Qiu and Rayadurgam Srikant. 2004. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. 34, 4 (2004), 367–378.

Gowri Sankar Ramachandran and Bhaskar Krishnamachari. 2018. Blockchain for the IoT: Opportunities and challenges. *arXiv preprint arXiv:1805.02818* (2018).

Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. 2018. On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems* 88 (2018), 173–190.

Robert W Robinson. 1977. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*. Springer, 28–43.

Gokhan Sagirlar, Barbara Carminati, Elena Ferrari, John D Sheehan, and Emanuele Ragnoli. 2018. Hybrid-IoT: Hybrid Blockchain Architecture for Internet of Things-PoW Sub-blockchains. *arXiv preprint arXiv:1804.03903* (2018).

Omar Said and Mehedi Masud. 2013. Towards internet of things: Survey and future vision. *International Journal of Computer Networks* 5, 1 (2013), 1–17.

Mehrdad Salimitari and Mainak Chatterjee. 2018. An Overview of Blockchain and Consensus Protocols for IoT Networks. *arXiv preprint arXiv:1809.05613* (2018).

Mayra Samaniego and Ralph Deters. 2016. Blockchain as a Service for IoT. 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), 2016 IEEE International Conference on*. IEEE, Chengdu, China, 433–436.

Rüdiger Schollmeier. 2001. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*. IEEE, Linkoping, Sweden, Sweden, 101–102.

Ludwig Seitz, Goeran Selander, Erik Wahlstroem, Samuel Erdtman, and Hannes Tschofenig. 2017. Authentication and authorization for constrained environments (ace). *Internet Engineering Task Force, Internet-Draft draft-ietf-aceoauth-authz-07* (2017).

Pradip Kumar Sharma, Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park. 2017. Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. *IEEE Communications Magazine* 55, 9 (2017), 78–85.

Janusz J Sikorski, Joy Haughton, and Markus Kraft. 2017. Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Applied Energy* 195 (2017), 234–246.

Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. 2017. A survey on LPWA technology: LoRa and NB-IoT. *Ict Express* 3, 1 (2017), 14–21.

sloct.it 2018. slock.it A Blockchain Company. Retrieved March 20, 2019 from https://slock.it/

Biljana L Risteska Stojkoska and Kire V Trivodaliev. 2017. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production* 140 (2017), 1454–1464.

Nan Sun, Jun Zhang, Paul Rimba, Shang Gao, Leo Yu Zhang, and Yang Xiang. 2018. Data-driven cybersecurity incident prediction: A survey. *IEEE Communications Surveys & Tutorials* 21, 2 (2018), 1744–1772.

Girish Suryanarayana and Richard N Taylor. 2004. A survey of trust management and resource discovery technologies in peer-to-peer applications. (2004).

Melanie Swan. 2015. *Blockchain: Blueprint for a new economy.* "O'Reilly Media, Inc.".

Lu Tan and Neng Wang. 2010. Future internet: The internet of things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, Vol. 5. IEEE, Chengdu, China, V5–376.

Tangle IOTA 2018. Meet the Tangle. https://www.iota.org/research/meet-the-tangle.

Teachracers 2019. Blockchain Development Company | Blockchain Services and Solutions. Retrieved March 20, 2019 from https://www.techracers.com/

TraceRx 2019. Tracerx : Global Blockchain Supply Chain for Drugs. Retrieved March 10, 2019 from https://www.leewayhertz.com/project/tracerx/

UniquID 2018. UniquID Incorporation Blockchain Identity Access Management. Retrieved March 20, 2019 from https://uniquid.com/

Mališa Vučinić, Bernard Tourancheau, Franck Rousseau, Andrzej Duda, Laurent Damon, and Roberto Guizzetti. 2015. OSCAR: Object security architecture for the Internet of Things. *Ad Hoc Networks* 32 (2015), 3–16.

Marko Vukolić. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security.* Springer, Cham, 112–125.

WaltonChain. 2018. WaltonChain White Paper V2.0. (2018).

Sheng Wen, Mohammad Sayad Haghighi, Chao Chen, Yang Xiang, Wanlei Zhou, and Weijia Jia. 2014. A sword with two edges: Propagation studies on both positive and negative information in online social networks. *IEEE Trans. Comput.* 64, 3 (2014), 640–653.

Andrew Whitmore, Anurag Agarwal, and Li Da Xu. 2015. The Internet of Things-A survey of topics and trends. *Information Systems Frontiers* 17, 2 (2015), 261–274.

Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014), 1–32.

Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du. 2010. Research on the architecture of Internet of Things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, Vol. 5. IEEE, Chengdu, China, V5–484.

Tingmin Wu, Sheng Wen, Yang Xiang, and Wanlei Zhou. 2018. Twitter spam detection: Survey of new approaches and comparative study. *Computers & Security* 76 (2018), 265–284.

Xage 2018. Home page of Xage Security. Retrieved March 20, 2019 from https://xage.com/

Jinhong Xie and Steven M Shugan. 2001. Electronic tickets, smart cards, and online prepayments: When and how to advance sell. *Marketing Science* 20, 3 (2001), 219–243.

Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang, and Wenji Liu. 2011. Study and application on the architecture and key technologies for IOT. In *Multimedia Technology (ICMT), 2011 International Conference on.* IEEE, Hangzhou, China, 747–751.

Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things journal* 1, 1 (2014), 22–32.

Jun Zhang, Xiao Chen, Yang Xiang, Wanlei Zhou, and Jie Wu. 2015. Robust network traffic classification. *IEEE/ACM Transactions on Networking (TON)* 23, 4 (2015), 1257–1270.

Jun Zhang, Yang Xiang, Yu Wang, Wanlei Zhou, Yong Xiang, and Yong Guan. 2012. Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed systems* 24, 1 (2012), 104–117.

Jun Zhang, Yang Xiang, Yu Wang, Wanlei Zhou, Yong Xiang, and Yong Guan. 2013. Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed Systems* 24, 1 (2013), 104–117.

Yu Zhang and Jiangtao Wen. 2017. The IoT electric business model: Using blockchain technology for the internet of things. *Peer-to-Peer Networking and Applications* 10, 4 (2017), 983–994.

Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An overview of blockchain technology: Architecture, consensus, and future trends. In *Big Data (BigData Congress), 2017 IEEE International Congress on.* IEEE, Milan, ITALY, 557–564.

Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. 2018. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14, 4 (2018), 352–375.