Blockchain-based IoT Consensus Mechanism System

Yue Wu¹.².*, Liangtu Song¹.², Lei Liu¹.² and Linli Zhou¹.²

¹Institute of Intelligent Machines, and Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, Anhui, PR China
²University of Science and Technology of China, Hefei 230026, Anhui, PR China

*Corresponding author: wyw533k@mail.ustc.edu.cn

Abstract. With the continuous development of the Internet of Things (IoT), the application of blockchain technology to the Internet of Things has become a trend of the times. As we all know, the core of the blockchain lies in the consensus mechanism. In order to apply the blockchain to the Internet of Things, and for subsequent research needs, this research proposes a lightweight, high-throughput consensus mechanism system. Using scalable technology, any number of nodes and gateways can be set up. The experimental results show that the system throughput increases synchronously with the increase of the test load; in addition, when the test load is 800tps, the system throughput reaches the maximum value, which is close to 600tps. When the test load exceeds 800tps, the actual system throughput begins to drop, and about 90% of transactions have a delay time of 5000ms. This blockchain consensus system can be used for the simulation of lightweight IoT systems.

1. Introduction

During the long-term development and evolution of the Internet of Things, it has encountered the following five industry pain points: device security, personal privacy, rigid architecture, communication compatibility, and multi-agent collaboration [1].

In terms of device security, Mirai’s Botnets of Things (Botnets of Things) was rated as one of the top ten breakthrough technologies in 2017 by MIT Technology Review [2][3]. According to statistics, the Mirai botnet has accumulated more than 2 million cameras and other IoT devices [4], The DDoS attack initiated by it paralyzed the American domain name resolution service provider Dyn, and many popular websites such as Twitter and Paypal were inaccessible at that time [5][6]. Later, there were botnets that enslaved IoT devices and allowed them to mine Bitcoin [7], and there were also larger and more active http81 botnets.

In terms of personal privacy, the main reason is that the centralized management structure cannot prove self-innocence, and the relevant time when personal privacy data is leaked occurs from time to time [8][9]. Recently, People's Daily Online reported that 266 cameras in Chengdu were broadcast live on the Internet as a case [10].

The research on this topic is Blockchain-based IoT consensus mechanism system. The main contents include the research on cryptographic algorithms and mechanisms suitable for lightweight IoT devices to access the blockchain. We attempt to avoid excessive communication overhead [11], and we introduce machine learning methods to identify or remove outliers in sensor data before data are uploaded to the chain [12].
2. System Design

As shown in Fig. 1, the overall design of the system is shown in the figure below.

![Figure 1. System overall design](image)

2.1. Node API

1) BlockchainNode, GatewayNode, SensorNode, WorkloadNode all inherit from TimerNodeClass, by overriding the on_start method and on_event method to achieve specific logic.

2) on_start (sim): When the simulation engine starts, the on_start () method of all nodes will be called, so the node initialization operation can be performed in this method.

3) on_event (event, sim): When the target object of the event processed by the simulation engine is the current node, the on_event method of the current node will be called.

4) The sim parameter of the above method represents the simulation engine object, that is, the Simulation object provided by biot_sim.

2.2. Simulation engine API

5) The Simulation class of biot_sim corresponds to the simulation engine, which provides the following APIs:

6) run(duration=30000): Start the simulation engine, the default simulation duration is 30 seconds

7) schedule(event): Add a specified event to the event queue of the simulation engine

8) broadcast (source, message, time): Broadcast the specified message to all nodes in the network starting from the specified time, where source represents the source node.

2.3. Event API

The simulation is event-driven and is defined by the Event class of biot_sim. The main fields are:

9) time: trigger time, unit: milliseconds

10) subject: Event target node

11) type: Event type, currently supports two types: EventType. TIMER and EventType. MESSAGE, which respectively represent timing events and communication message events
12) payload: The payload of the event. When the event type is EventType.MESSAGE, payload represents a message object. When the event type is EventType.TIMER, the payload is None.

The message object is defined by the simulation application layer and can be any valid Python type. The pojo() method provided by biot_sim is usually used to convert a dictionary into a simple object as a message.

2.4. Blockchain storage API

The Blockchain class of biot_sim implements memory blockchain storage, and its main API is defined as follows:

1) create_next_block(time): Create the next block, time represents the block timestamp, in milliseconds
2) commit_block(block): upload the specified block locally
3) get_height(): Returns the current height of the blockchain
4) cache_tx(): Add the received transaction to the memory transaction pool
5) clear_block_txs(block): Clear the transactions in the specified block from the memory transaction pool
6) print_chain(): Display the current chain composition on the terminal, the hash of each block only displays the first 6 characters.

The overall flow of the algorithm is shown in Figure 2.

![Figure 2. Algorithm process](image)

As shown in Fig.2, the system has two types of transactions: sensor-aware transactions that are submitted to gateways, and normal transactions that occur on blockchain nodes or upper-level applications. Considering the performance difference between IoT terminal devices and node or application hosts, we design different verification mechanisms for both transactions:

Transaction aware: From sensors to gateways, symmetric encryption algorithm is used for transaction verification.

Ordinary transactions: Asymmetric encryption algorithm is used for transaction verification.

Gateways use machine learning models to detect the abnormality in the sensor perception transaction, attaches the detection result and re-encapsulates it, and broadcasts it to the blockchain network. After receiving and verifying the broadcast transaction, blockchain nodes temporarily store it in the pending transaction pool and waits for block confirmation. We divide the continuous time into consensus rounds at equal intervals, and blockchain nodes package pending transactions, produce blocks, and broadcast to the blockchain network in rounds. To improve throughput, in a round, certain nodes are randomly
selected as dominant nodes, and only the selected dominant nodes can produce blocks. Given that multiple leading nodes may be selected in a round, nodes may receive blocks of the same height broadcast from multiple leading nodes. We use the longest chain rule to solve the blockchainfork conflict.

3. Consensus Mechanism

3.1. Key agreement protocol
As shown in Fig.3, after sensors are started, the Diffie-Hellman algorithm is used for key negotiation with the blockchain nodes:

![Figure 3. Sensor and node key agreement protocol](image)

1) Sensors and blockchain nodes share prime number $p$ and base $g$.
2) Sensors first select private key $a$, and then send public key $A$’s $a$ to blockchain nodes

\[ A = g^a \mod p \]  \hspace{1cm} (1)

3) Nodes first select $a$ private key $b$, and then send $b$’s public key $B$ to sensor nodes

\[ B = g^b \mod p \]  \hspace{1cm} (2)

4) Sensors calculate the shared key:

\[ K_{sensor} = B^a \mod p \]  \hspace{1cm} (3)

5) Nodes calculate the shared key:

\[ K_{bc-node} = A^b \mod p \]  \hspace{1cm} (4)

6) Considering that $K_{sensor} = K_{bc-node}$, the shared key can be used by sensors and blockchain nodes to generate HMAC signatures for sensor-aware transactions.

3.2. Block node selection
To avoid the computational overhead in PoW consensus and the communication overhead in BFT consensus, this study adopts Verifiable Random Function (VRF) to realize the offline fast election of block nodes. VRF is a cryptographic hash of the public key version. Only the private key holder can calculate the hash, but any participant who knows the public key can verify the correctness of the hash. The algorithm flow is summarized as follows:

1) The private key holder uses the private key SK and the public input data alpha to calculate the hash $\beta$ and evidence $\pi$: 
The verifier uses the hash provider’s public key $PK$, evidence $pi$, and input data $alpha$ to recalculate hash $beta$. If it matches the provider, the hash is correct:

$$beta = vrf\_proof\_2hash(p_i)$$

(6)

2) The verifier uses the hash provider’s public key $PK$, evidence $pi$, and input data $alpha$ to recalculate hash $beta$. If it matches the provider, the hash is correct:

$$beta = vrf\_verify(PK, alpha, pi)$$

(7)

In this study, the consensus algorithm uses a round consensus mechanism, which divides the time into rounds of fixed length. In each round, VRF is used to determine whether the current node of the current round is selected as the block generation node, and if so, the block is broadcasted. VRF requires all parties involved to hold a key pair, and the block selection algorithm flow of each round is as follows:

1) Calculate the shared information $alpha$ of the current round, where $t$ represents the current time and $T$ represents the duration of the round:

$$alpha = \frac{t}{T}$$

(8)

2) The node uses its private key and shared information $alpha$ to calculate the hash beta and evidence $pi$:

$$pi = vrf\_prove(SK, alpha)$$

(9)

$$beta = vrf\_proof\_2hash(p_i)$$

(10)

Assuming that the probability of winning a node in an election is $p$, and the selection experiment for each node $n$ times is repeated, the probability $X_k$ of finally selecting $k$ nodes conforms to the binomial distribution, namely,

$$X_k \sim b(n, p)$$

(11)

Calculate the corresponding $k$ when $X_k$ is the following value, where $hashlen$ represents the bit length of hash $beta$:

$$p_{node} = \frac{beta}{2^{hashlen}}$$

(12)

If $k > 0$, then the node is selected.

3) The selected block node broadcasts the block, with its public key and evidence.

4) After receiving the block, the other nodes calculate the alpha value of the current round, use the public key $PK$ of the block-producing node in the block and the evidence $pi$ to calculate the hash, and perform Step 3 above. After verifying that the node is indeed a block-producing node, change the area. The block is also added to the local chain.
4. Results
Transaction delay time represents the time it takes for transactions to be submitted to the block confirmation. From the data sent by sensors to the final transactions entering blocks, the total delay time can be divided into two sections.

\[ \text{Latency} = L_1 + L_2 \] (13)

where \( L_1 \) represents the time from sensor submission to gateway detection completion, and \( L_2 \) represents the time from gateway submission transaction to block packaging. The histogram of transaction delay statistics is shown in Fig.4.

Figure 4. Transaction delay analysis

The average proportion of the overall delay from the sensor to the gateway confirmation stage is approximately 10%. Fig.5 displays a comparison chart of the two-stage delay:

Figure 5. Comparison of transaction delay between two stages

The transaction delay in the first phase remains basically unchanged, whereas that in the second phase exhibits an obvious periodic characteristic. The reason is that the data submission of sensors is periodic; thus, the transaction submission of gateways also remains basically periodic. When the moment in which
gateways submit transactions is close to the next block generation round, only a short transaction delay can be confirmed; otherwise, a large transaction delay time is required. The throughput experimental results are shown in Fig.6.

![Figure 6. Throughput analysis](image)

Within a certain range (test load <800 tps), the system throughput synchronously increases as the test load increases, and when the test load is 800 tps, the system throughput reaches the maximum which is close to 600 tps. When the test load exceeds 800tps, the actual system throughput begins to decrease, indicating that the simulation system has been overloaded.

**Acknowledgments**
This study was supported by Key Research and Development Plan of Anhui Province(201904a06020056).

**References**
[1] Yue Wu, Liangtu Song, Lei Liu, Jincheng Li, Xuefei Li, Linli Zhou, "Consensus Mechanism of IoT Based on Blockchain Technology", Shock and Vibration, vol. 2020, Article ID 8846429, 9 pages, 2020. https://doi.org/10.1155/2020/8846429
[2] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," Telecommunications policy, vol. 41, no. 10, pp. 1027-1038, 2017.
[3] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and Iot integration: A systematic survey," Sensors, vol. 18, no. 8, p. 2575, 2018.
[4] S. T. Mehedi, A. A. M. Shamim, and M. B. A. Miah, "Blockchain-based security management of IoT infrastructure with Ethereum transactions," Iran Journal of Computer Science, vol. 2, no. 3, pp. 189-195, 2019.
[5] J. Wu and N. K. Tran, "Application of blockchain technology in sustainable energy systems: An overview," Sustainability, vol. 10, no. 9, p. 3067, 2018.
[6] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using blockchain and trusted execution environment," in 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018: IEEE, pp. 15-22.
[7] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), 2017: IEEE, pp. 173-178.
[8] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and privacy," arXiv preprint arXiv:1712.02969, 2017.

[9] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1184-1195, 2018.

[10] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, vol. 82, pp. 395-411, 2018.

[11] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," Ieee Access, vol. 4, pp. 2292-2303, 2016.

[12] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in 2018 17th international symposium infoteh-jahorina (infoteh), 2018: IEEE, pp. 1-6.