Implementation of Micropayment System Using IoT Devices

Ryota Nakada, Kien Nguyen* and Hiroo Sekiya

Graduate School of Engineering, Chiba University
1-33 Yayoi-cho, Inage-ku, Chiba-shi, Chiba 263-8522, Japan
E-mail: r.nakada@chiba-u.jp, nguyen@chiba-u.jp, sekiya@faculty.chiba-u.jp

Abstract

In the IoT, a small amount of payment (i.e., micropayment) enables the trading of sensor data collected by IoT devices. The IOTA cryptocurrency, which achieves high-speed transactions without transaction fees, shows the potential to realize such micropayment. In this work, we introduce and implement an IOTA-based micropayment system for IoT devices (i.e., Raspberry Pi), assuming an air quality monitoring application. We then evaluate the latency performance and power consumption.

1. Introduction

With the development of IoT systems in many applications, there are several situations in which customers wish to acquire sensor data from IoT devices. The sensor data collected by IoT devices are not shared for free. It is assumed that a monetary transaction will be required to share the sensor data across an organization. The application will use a micropayment system for trading sensor data and money. The micropayment represents a small electronic payment, such as several yen or tens of yen in the standard currency [1]. IOTA is expected to be used in such a micropayment system as a currency to trade for sensor data.

IOTA does not use the chain structure as do other cryptocurrencies. Instead, it uses a directed acyclic graph (DAG) data structure, which forms a network of transactions called Tangle [2]. IOTA does not require a miner for the approval work. It enables high-speed transactions with no requirement of transaction fees. However, the technologies in IOTA are not yet matured. Moreover, the development of a micropayment system on IoT devices is insufficient. Therefore, it is necessary to investigate and evaluate these topics in various practical examples.

In previous work, there is a Bitcoin micropayment system on IoT devices [3]. In [4], the authors introduced an IOTA micropayment system. However, they implement the system on normal PCs and without real data consideration. In this paper, we introduce a micropayment system based on IOTA for IoT devices, considering an air-quality monitoring application. We first implement the micropayment system using Raspberry Pi devices. We then evaluate the latency performance and power consumption. The evaluation results show that the latency consumed by one micropayment is about 3 min in our system. Most of the latency is caused by the hash calculation to approve IOTA’s new transaction. The power consumption for approving IOTA’s new transaction is about 112 J on average.

2. Micropayment with IOTA

2.1 IOTA

IOTA is an open-source distributed ledger technology and cryptocurrency forming the Tangle, which enables high-speed, feeless transactions. First, IOTA has a data structure different from other blockchain technologies, as it employs not blocks, but transactions. Therefore, IOTA only needs to approve a transaction; this makes high-speed transaction possible. Second, IOTA uses a new type of proof of works (PoW) mechanism that allows a transaction to join the Tangle by two or more previous transactions. Hence, IOTA does not require the miner.

The approval mechanism of IOTA includes three periods: Tip selection, PoW, and Broadcast. IOTA’s tip selection is the task of referring to two previous transactions before joining the Tangle. This algorithm processes a transaction while preventing double-spending and conflict with other transactions [2]. The difficulty of this algorithm changes depending on the Depth parameter. A larger Depth parameter increases the tip selection time and the computational power [2]. The PoW is the hash calculation for approving the transaction. The difficulty of PoW changes depending on the parameter called the minimum weight magnitude (MWM). The larger the MWM value is, the more difficult the hash calculation becomes. Broadcast is the process of sending the information of a new transaction to the entire Tangle. IOTA transaction contains data strings called trytes. In IOTA, data is represented in as a balanced ternary consisting of 1, 0, or -1. These values are called trits, and three trits are equal to...
one tryte. IOTA trytes have 27 possible values, consisting of A–Z and the number 9 [5]. IOTA transaction has several parameters such as an address, nonce, and value. The address is IOTA’s personal account. The nonce is the trytes that certify that PoW was done. The value represents the amount of IOTA tokens. The transaction, which has a 0 value, is called a zero-token transaction. In IOTA, a group of transactions is named Bundle, which may consist of zero-value transactions. Bundle can contain either messages, signatures, or both.

2.2 IOTA on IoT device

The IOTA client has several implementations in different programming languages, such as C, Python, and JavaScript, among which, PyOTA is a Python package for IOTA clients with extension functions. In this work, we use PyOTA to implement our micropayment system. CCurl is an implementation of the Curl library [6]. As above which is IOTA’s unique hash function, in C language.

2.3 Micropayment system with air-quality monitoring

Figure 1 shows the architecture of micropayment system with air quality monitoring. The micropayment system has three channels: PAYMENT CHANNEL, RECORD CHANNEL, and DATA CHANNEL. In PAYMENT CHANNEL, the seller and buyer exchange IOTA for the transactions. RECORD CHANNEL is used to store the transaction invoice and micropayment details in the IOTA Tangle. DATA CHANNEL is for the data transmission, which uses TCP and encryption. In addition, we connect a Nova PM sensor to the seller to monitor and collect PM2.5 and PM10 data. PM2.5 is a particle with an aerodynamic diameter of 2.5 \( \mu m \) or less, and PM10 is a particle with an aerodynamic diameter of 10 \( \mu m \) or less. We measured the PM data throughout the day and save every one minute.

In Fig. 2, we present detailed processes of micropayment between a seller and a buyer, similar to the one in [4]. In this case, the buyer has IOTA and wants to buy the seller’s data. Different from [4], we implement and tailor the seller and buyer on Raspberry Pi. First, the buyer establishes a TCP connection to the seller. The buyer then sends a HELLO message to initialize the payment process. Next, the seller sends a MENU message as a response to the HELLO message. The MENU message provides the set of data types, prices, and amounts of data. The MENU message is encrypted with a signature. The buyer sends an ORDER message that includes the amounts of data. The buyer also sends its public key. Next, the seller establishes a SESSION KEY encrypted by the buyer’s public key. Then, the seller sends sensor data to the buyer of the amount set in the ORDER message. This data is encrypted by the SESSION KEY. The buyer uses the SESSION_KEY to decrypt the data sent by the seller. After sending data, the buyer sends IOTA as a reward, and the seller stores the transaction invoice in IOTA. Finally, the buyer terminates the TCP connection.

3. Evaluation

3.1 Experiment environment

In this evaluation, we consider the micropayment system
in an air-quality application using Raspberry Pi 4 Model B. Table 1 shows the operating system and configurations used in the experiment. Initially, we install PyOTA version 2.1.0 [7] and ccurl-interface version 1.1.0 [8] in the devices. The former is an IOTA client, while the latter is used to interact with CCurl using the Python language. Second, we select the IOTA public nodes that connect our IoT nodes to the IOTA tangle. We randomly choose three reachable IOTA nodes in different geographic regions. The detailed information of three public nodes is shown in Table 2. We set the MWM value to 14, which is the default value. The Depth value is 2.

### 3.2 Evaluation results

#### 3.2.1 Micropayment in air-quality application

In this experiment, we exchange PM data of the last 20 minutes and 0 IOTA. We observe the micropayment operations. After a successful micropayment, the system will issue a transaction invoice. Figure 3 shows the invoice in one payment. In the figure, we can see that the transaction value is 0 and the weight magnitude is 14 (similar to our settings). In addition, we can confirm that Address, Bundle, and Nonce are represented by IOTA trytes. Moreover, we can see random strings at the bottom of Fig. 3. This is sensor data converted to IOTA trytes. Figure 4 shows sensor data reconverted to normal text from IOTA trytes stored in IOTA. Therefore, we can conclude that the micropayment has been completed in our system with IOTA.

#### 3.2.2 Latency evaluation

In this experiment, we will measure the latency consumed by different processes when issuing a new transaction and repeat it 100 times [9]. We measure the latency divided into three terms, Tip selection, PoW, and Broadcast, by setting time-stamp, and we also calculate their total time. We then collect and calculate the maximum, minimum, and average values. The results are shown in Fig. 5.

Tip selection in Fig. 5 is the latency of the tip selection process, which is the time from selecting the two previous transactions to Tangle’s approval. When an IoT device issues a new transaction, IOTA generates a Bundle address and sends the address to an IOTA public node. Then, the public node performs a Tip selection algorithm and selects the transaction to approve from Bundle. After that, the public node sends tips information to the IoT device. This latency is not dependent on the IoT device. It however, depends on the Depth value, which represents the difficulty of the Tip selection algorithm, and the number of transactions on the public node. In our evaluations, the resulting latencies are about 2 to 3 s in all public nodes.

PoW in Fig. 5 indicates the PoW time, which is the time of hash calculation to confirm the transaction selected by the Tip selection algorithm. The PoW operation strongly depends on the CPU power; hence, it is different from Tip selection and Broadcast latencies. In this environment, the average time was 45 s. Moreover, there is a large difference between the maximum value and the minimum value caused by the non-deterministic nature of PoW operation.

Broadcast in Fig. 5 shows the broadcast latency, which is the time required to finish broadcasting the transaction information. This latency depends on the number of transactions.
in the public node. Moreover, Total in Fig. 5 shows the total latency, which includes all three latencies mentioned above. We can see that the values of Tip selection and Broadcast time values are small. Hence, the PoW latency causes a large effect on the total time values.

We let the IOTA client on IoT devices connect to the IOTA network three times in the micropayment system. Additionally, a small amount of time is required for other work, such as exchanging messages. Therefore, it can be said that micropayment is completed in about 150 s for any public node on our system.

3.2.3 Power consumption evaluation

In this evaluation, we measure the power consumption during the hash calculation (PoW) (during the approval period of IOTA’s new transaction) using AVHzY CT-2 USB Power Meter [10]. The experiment has been repeated ten times. AVHzY CT-2 USB Power Meter can measure the current and voltage values on Raspberry Pi during the hash calculation. Therefore, we track the values and calculate the power consumption by multiplying the current and voltage. Moreover, we also measure the PoW time. The results are shown in Fig. 6, in which the left and right y-axes show the power consumption and PoW time, respectively. In all ten experiments, the average power consumption is about 112 J. Moreover, the difference between the maximum value and the minimum value is caused by the unstable PoW values.

4. Conclusion

We introduced an IOTA-based micropayment system for IoT devices, considering an air-quality monitoring application. We have implemented and confirmed its operations on the Raspberry Pi devices. We have also evaluated the latency performance and power consumption of the processes during the issuance of IOTA’s new transactions. The latency is about 150 s on average; meanwhile, the average power consumption of the hash calculation is approximately 112 J.

In the future, we will examine the reduction of the hash calculation time using a proxy server or additional hardware for IOTA (i.e., CryptoCore for high-speed hash calculation [11]). Moreover, we will incorporate real IOTA values into the system.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 19K20251 and 20H04174.

References

[1] F. Rezaeibagha, Y. Mu and J. Cho: Efficient micropayment of cryptocurrency from blockchains, Comput. J. Vol. 62, No. 4, pp. 507-517, 2019.
[2] S. Popov: The Tangle, https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf
[3] A. Xu, M. Li, X. Huang, N. Xue, J. Zhang and Q. Sheng: A blockchain based micro payment system for smart devices, Proc. Conference on Emerging Topics in Interactive Systems, 2016.
[4] R. Radhakrishnan and B. Krishnamachari: Streaming data payment protocol (sdpp) for the internet of things, Proc. IEEE iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData, pp. 1679-1684, 2018.
[5] Iota docs: https://docs.iota.org
[6] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza and T. Dryja: Cryptanalysis of curl-p and other attacks on the iota cryptocurrency, IACR Trans. Symmetric Cryptology, Vol. 202, No. 3, pp. 367-391, 2020.
[7] iota.py: https://github.com/iotaledger/iota.py
[8] ccurl.interface.py: https://github.com/iotaledger/ccurl.interface.py
[9] X. Chen, K. Nguyen and H. Sekiya: An experimental study on performance of private blockchain in IoT applications, Peer-to-Peer Netw. Appl, 2021.
[10] AVHzY Website: https://www.avhzy.com
[11] Cryptocore: https://docs.iota.org/docs/iot/0.1/cryptocore/introduction/get-started