COM 5335 : NETWORK SECURITY

RESEARCH PAPER STUDY FINAL PROJECT

Delay and Communication Tradeoffs for BlockchainSystems With Lightweight IoT Clients

Pietro Danzi , Student Member, IEEE, Anders E. Kalør , Student Member, IEEE,
Cˇ edomir Stefanovic´ , Senior Member, IEEE, and Petar Popovski , Fellow, IEEE

STUDENT ID :	108064535
NAME :	Wen-Yuan Chen (Chris)
PROFESSOR :	Scott Huang
SUPERVISOR :	Thattapon Surasak

I. INTRODUCTION

This following Table 1. will provide a brief explanation about this paper.

Q5	Questions	Explanation
1	**What is the research / telling question(s) ?**	Common IoT devices have the constraint on computing, low-power and memory. So the authors will discuss how to reduce the cost of communication.
2	**What are the key concepts ?**	**(1)** Blockchain **(2)** Internet of Thing (IoT)
3	**What methods are used ?**	This paper proposed a new system model (Aggregation Protocol) to aggregate the information in order to reduce the communication cost.
4	**What answers are presented ?**	The obtained experiment result from Ethereum show that the new scheme will sacrifice communication delay time to exchange a lower communication cost.
5	**What are the contributions of this paper ?**	**(1)** Provide an aggregation scheme that implemented at Blockchain Nodes. This scheme can reduce the device duty cycle and the amount of propagation data. **(2)** The authors extend their previous model in [1]. The result is for a lightweight client model which is rich and easy to be analyzed. They also show the potential application that increase the IoT devices privacy and keep the lower communication cost. **(3)** They research the cost of propagation for updated data and also provide the experiment results that obtained from Ethereum.

Table 1. A brief explanation about this paper.

□ What is blockchain ?

It’s a kind of database but it’s decentralized rather than centralized database. Which is mean the data is not held by a central server or central service whereas held by every node in the blockchain network. Hence, all blockchain nodes can stay on the top of data.

□ Why we need blockchain technology ?

It’s aiming to support the implementation of decentralized applications (DAPPs). And these kind of applications will reduce the requirement of a central server to provide service and oversee the interactions between clients. Which is mean we don’t need to via a third party to complete a transaction whereas complete by ourselves. Furthermore, blockchain also has very strong security because all of transactions need to be validated by the nodes in the blockchain network according to the consensus algorithm.

□ IoT devices with blockchain

IoT devices normally locate on the edge of blockchain network and via wireless base station connect to Blockchain Nodes, see Fig 1. In the application of wireless sensor network, devices are simple data sources and normally no one can oversee. Therefore, they’re easy to be attacked, but this problem can be solved by blockchain network.

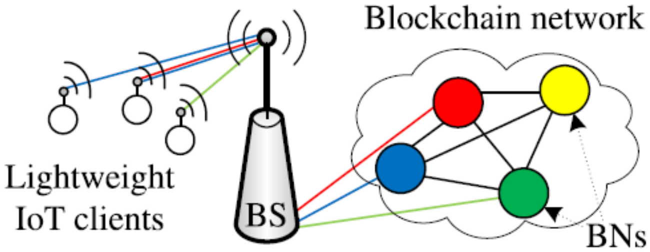


Fig. 1. Communication architecture for the interaction of IoT devices with a set of BNs via wireless links provided by a BS.

□ Blockchain protocol

The blockchain database is replicated at multiple nodes that are interconnected by a communication network. Every time a node appends a new block to its copy of the blockchain, it transmits the block to the rest of the other nodes in the network, to keep the blockchain database replications consistent.

□ Block data structure for Ethereum

A block is consisted of a block header and a body, you can reference Fig 1. The block header size is fixed, and the remainder parts of the block is composed of the actual transactions and have variable size. When the amount of transaction is high, the variable parts will almost account for the whole block. In this paper, the authors mainly consider the transaction tree and state tree.

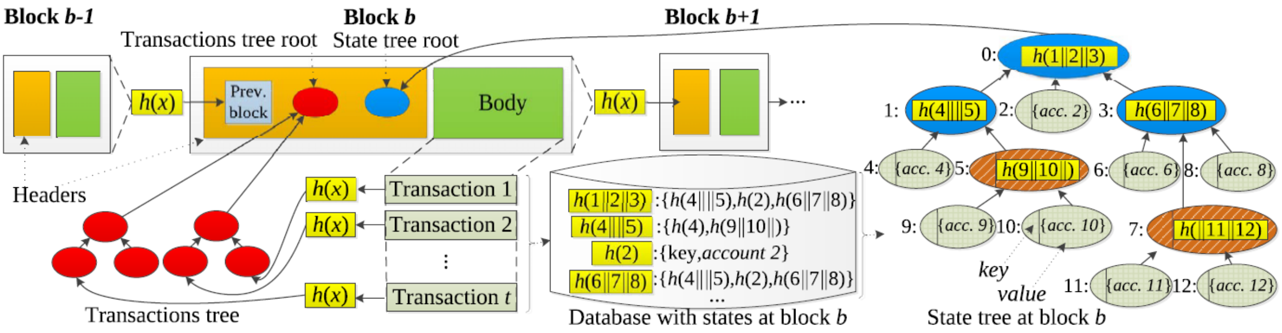


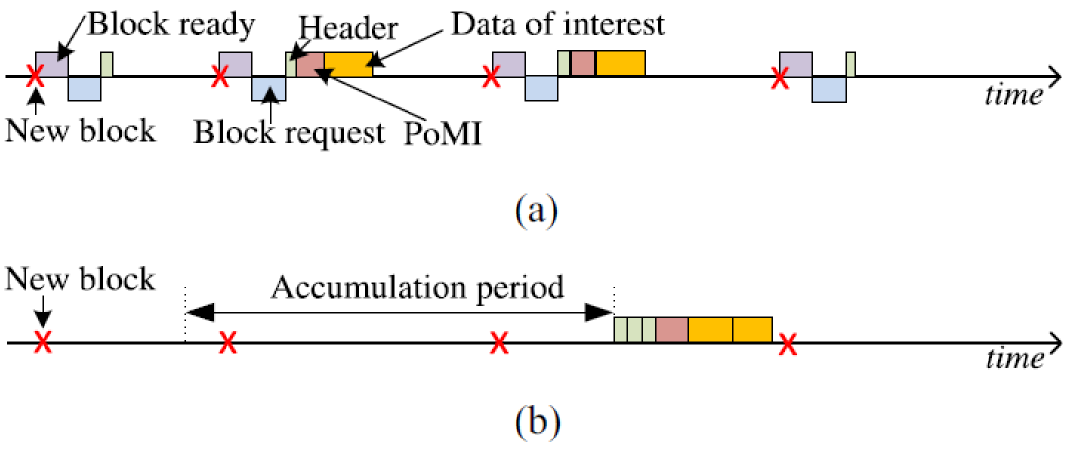
Fig. 2. Example structure of a blockchain. $h(x)$ is the hash value of node x and $||$ is the concatenation operation. The t transactions included in block b apply modifications to the accounts' states, which are stored in a database. The state tree depicted on the righthand side, ternary in this example, is build from this database, and its root included in the block header. Branch nodes are colored in blue and extension nodes in brown. Leaf nodes (there are eight of them in the example) are composed by key and value, and colored in green.

□ PoI via Merkle-Patricia Trees

Ethereum uses Merkle-Patricia Trees to provide Proof of Inclusion (PoI). Merkle-Patricia Trees have three types of nodes such as leaf, extension and branch nodes, you can reference right side of Fig 1. In Blockchain, string is the hash value of account or the address of transaction, the data structure that we wanna retrieve is account or transaction.

□ Synchronization protocols

Blockchain Node (BN) through synchronization protocols to update the content, modification of blockchain database observed by Blockchain Node. In general case, clients receive block headers from Blockchain Node when someone modify Blockchain Node. We call this scheme as lightweight protocol. Since it can reduce the amount of propagation in downlink and the amount of local processing. In this paper the authors use the IoT device with both low-memory and communication functionality but it still supports the lightweight protocol. Fig 3. show the exchanged information between client and Blockchain Node during three block period. The red cross means the time that generate a new block.



(a)

(b)

Fig. 3. Information exchanged between a client and a BN using a lightweight protocol during four block periods, (a) without aggregation scheme and (b) with aggregation. Downlink/uplink messages are depicted above/below the time arrow. "Data of interest" includes the accounts' data and relative PoMIs.

□ Problems from lightweight IoT devices

IoT devices need to retrieve the validation information stored in blockchain. But access the whole blockchain and handle each transaction needs lots of memory and computation, it’s impossible for IoT devices because of the memory, computation and power constraint. Hence, the authors came up with a scheme to decrease communication cost.

The following Table 2. is the common term of this paper.

Abbreviation	**Description**
BN	Blockchain node
BS	Base station
dAPP	Decentralized application
PoI	Proof of inclusion
PoMI	Proof of multiple inclusions
RPL	Recursive Length Prefix
SPV	Simplified Payment Verification

II. METHOD

The scheme proposed in this paper aggregates the information to reduce communication costs, as shown in Fig 3(b). The information can be accumulated at the BN and then periodically released at the subsequent aggregation point according to the tolerable delay of client. The method BN follows is to send a certificate through the state tree to trigger the replacement of the client's local copy. This allows to send only the latest version of the account that was modified multiple times during the accumulation process and merge the PoIs of the accounts modified in different blocks in a single PoMI.

□ Blockchain Network and IoT Device

We consider a blockchain network where new blocks are generated at (network-wide) rate λ at exponentially distributed intervals. A single IoT device is connected to a group of N BNs via BS and wireless links, see Fig 1. We consider a blockchain network where new blocks are generated at (network-wide) rate λ at exponentially distributed intervals. A single IoT device is connected to a group of N BNs via BS and wireless links, see Fig 1. The blockchain traffic on the wireless link is generated through two different processes: (1) the transmission of transactions, including the blockchain from the device to the BN, and (2) the exchange of messages as part of the synchronization protocol. However, we note that: (1) only involves the transmission of transaction metadata with a deterministic size (mainly the signature of the device). Therefore, the rest of this article will not deal with this process. Regarding the process and (2), the device subscribes to the block headers of all generated blocks and the status update of a set of accounts A, which is a subset of the existing accounts. The general account with index $j \in N$ is independently updated in blocks with probability P_j. We consider a situation where the device is not interested in the complete status history of the observed accounts, but only in their latest status. In other words, it is only necessary to inform the device about the latest status of the observed account and receive PoMI that proves that a specific account status is included in the blockchain.

□ Aggregation Protocol

The block header and the updated observation account are aggregated at the BN selected by the device and sent to the device every T seconds. The value of T depends on the information delay allowed by the application from modifying the account to delivering the update to the device. After successful reception, the IoT device will acknowledge the packet. We assume that the device selects the sequence of aggregated BNs in different aggregation cycles as part of the initial network association process, such as through a seed sequence. Therefore, the implementation of the protocol requires only downlink messages, because all the information required by the BN is sent by the IoT device during the initialization phase. If no transmission is in progress, the device is assumed to be in power saving mode.

□ Wireless Link

Assume that the wireless downlink from the BS to the IoT device is a Rayleigh-fading channel, which has a constant channel gain during the entire transmission process, and an independent channel gain during the transmission process. Due to the power constraints of the IoT device, we assume that the BS has no information about the channel and hen performs no power or rate adaptation. As a result, a transmission may fail with probability

$$\mathrm{p_{out}} = 1 - e^{-\frac{2^{\frac{R}{W}}-1}{\gamma}}$$

Where γ is the average received signal-to-noise ratio (SNR), R is the transmission rate in bits/s, W is the bandwidth of the channel in HZ. The downlink packet is retransmitted until it has been received successfully by the device.

In contrast to the downlink transmissions, we assume that the transmission of the acknowledgment packet in the uplink happens instantaneously and is always received reliably, thanks to power control, performed at the IoT device side, based on the received transmission.

□ Frame Structure

The Fig 4. Show the downlink frame is consisted of F bits. And it is divided into a fixed number of header bits H representing a standard communication protocol overhead and a variable number of effective payload bits D corresponding to the blockchain information, that is, $F = H + D$. Its duration is related to the transmission rate R

$$T_w = \frac{kF}{R} \qquad [s]$$

Where $k \geq 1$ is the number of transmissions, including retransmissions due to outage.

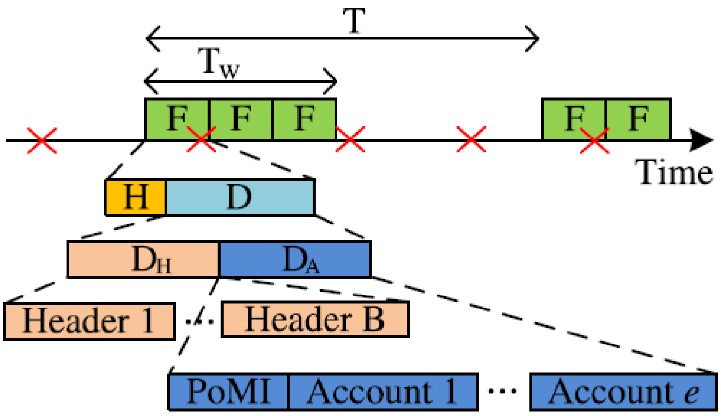


Fig. 4. Periodic release of information. Red crosses represent block generations. In the first release, there are two retransmissions of the frame (F), due to failure, in the second release, only one retransmission.

If the transmission of a frame takes longer than the transmission period, ie $T_w > T$, it is suspended and considered as a failure due to retransmission. In this paper, the authors consider the channel and block generation parameters. These parameters provide a small probability that the frame cannot be received in the current transmission cycle, so this phenomenon can be ignored. The D payload bits are divided into D_H bits for block headers and D_A bits for account updates, i.e., $D = D_H + D_A$. D_H and D_A are random as they depend on the number of generated blocks and account updates.

III. RESULT

In order to verify their model and show the performance of the aggregation solution, they modified the Ethereum protocol in the python module PyEthereum. The parameter settings are shown in Table 3. The system includes a randomly generated blockchain. This is obtained by generating an account containing random information of size l_A bits and inserting it into a newly initialized blockchain database.

Blockchain					
λ	$0.1\ s^{-1}$	**H**	1200	$\mathbf{l_H}$	$4046\ bit$
$\mathbf{l_a}$	$320\ kb$	$\mathbf{l_s}$	$256\ bit$	**L**	16
Communication channel					
R	250 kbit/s	**W**	180 kHz	**γ**	30 dB

Table 3. System parameters

□ Statistical Characterization of Accounts Updates

Account update statistics play a fundamental role in the design and evaluation of blockchain protocols. Fig 5. shows the update frequency of the 10^4 latest accounts, indexed in descending order of their update frequency. We model the relative frequency of account j updates based on a broken power law

$$\mathrm{p_j} = \begin{cases} \alpha_1 j^{\alpha_2} & ,if\ \ j \le \alpha_3 \\ \alpha_3^{\alpha_2-\alpha_4}\alpha_1 j^{\alpha_4}, otherwise \end{cases}$$

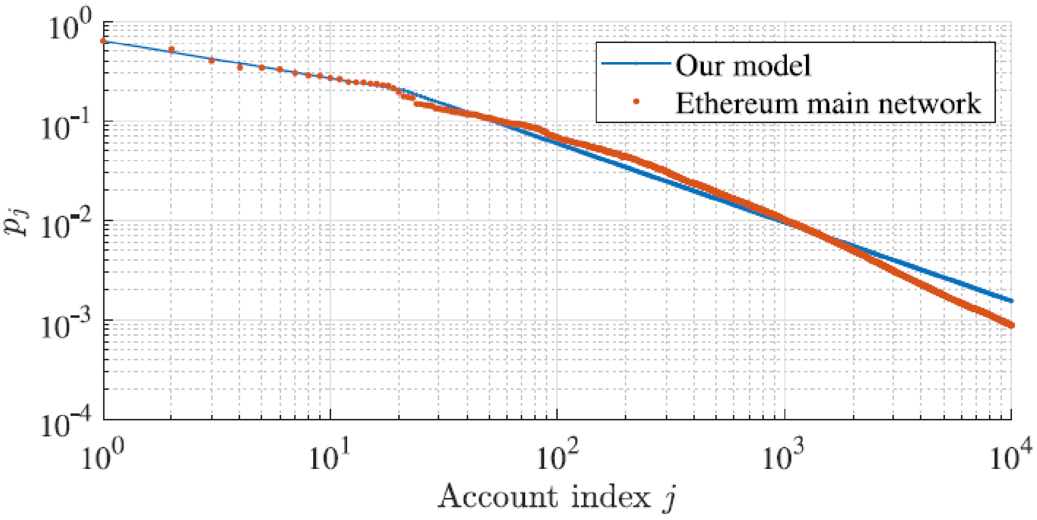


Fig. 5. Relative frequency of updates for the most active accounts, in log-log scale.

In addition to obtaining account update probabilities, we also checked the accuracy of modeling the number of blocks between two account updates as a hypothetical geometric distribution in the model. We compare the CDF of account j with the CDF of the geometric distribution with parameter $\mathrm{p_j}$. Fig 7. shows the results obtained from some dataset accounts. The result is that this assumption applies only to frequently updated accounts, so this should be considered in future work.

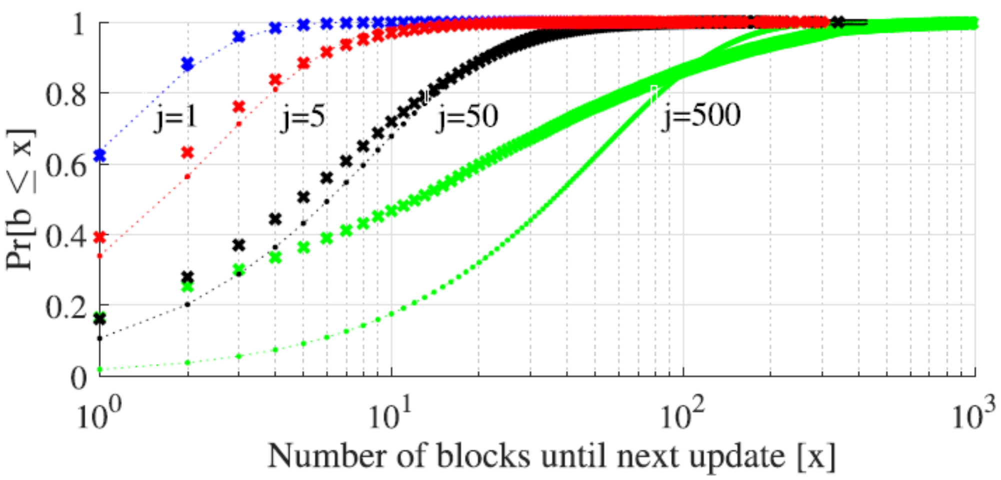
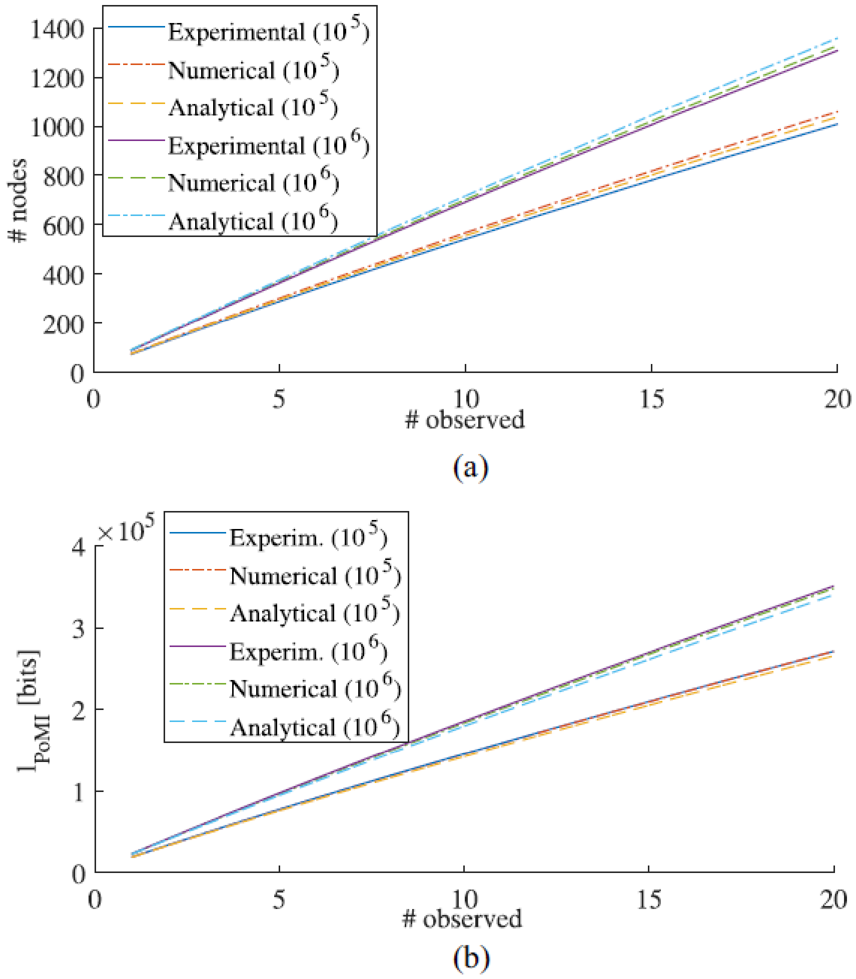


Fig. 7. Comparison of empirical CDF of accounts (represented with crosses), with index *j*, with the CDF of geometrical distribution (represented with dots).

□ Validation of Merkle-Patricia Proofs Length

Because the analysis of Merkle-Patricia proof length is based on the assumption that the tree is perfectly balanced, the analysis results can be compared with the numerical results obtained from the perfectly balanced tree and the measurements obtained from Merkle to verify the validity of the analysis. The Patricia tree implementation in PyEthereum is usually unbalanced, you can see Fig 8.



(a)

(b)

Fig. 8. Comparison of analytical approximation, numerical, and experimental results for the (a) number of nodes needed in the PoMI and (b) its length.

The experimental setup also allows to characterize the distribution of the number of nodes in PoMI, you can see Fig 9, where we show the results of a blockchain with η = 6 levels, fully populated, and thus containing $L^6 = 16^6$ accounts. L is the maximum number of children of the node. The relative position of accounts in the tree significantly affects the length of their PoMI, and therefore the communication cost of transmitting them.

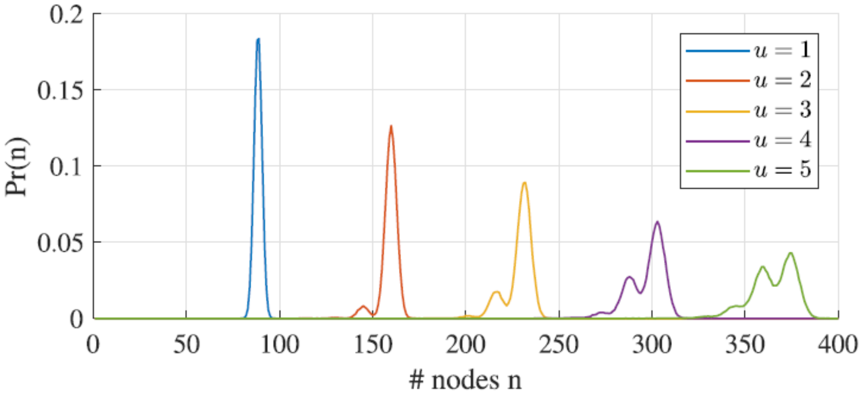


Fig. 9. Probability distribution of the number of nodes in a PoMI, for different number of observed events u, obtained via experiment.

□ Performance of the Aggregation Protocol

The author considers a scenario where the device is connected to the BN via a parameterized communication link as shown in Table 3. If the device is only interested in observing accounts that are updated occasionally, the aggregation protocol can only reduce the communication overhead (frame header). Therefore, we focus our evaluations on cases where the device observes an active account. Account j is considered active if it is updated at least once every T seconds and the probability is

$$1 - \left(1 - p_j\right)^{\lceil T/T_{\mathrm{B}} \rceil} \geq P_A$$

In other parts, we set P_A to 0.9. By requesting updates for more active accounts than actual interest accounts, the device can increase its privacy, but at the cost of downloading unnecessary information.

➢ Duty Cycle Tradeoffs

Fig 10. illustrates the complementary CDF of the duration T_w of the transmission for two identified sets of observation accounts: $A_1 = \{1, 2\}$. The number of accounts observed and their statistics clearly play a significant role in forming the CDF, as the size of the account data structure is much larger than the size of the block header.

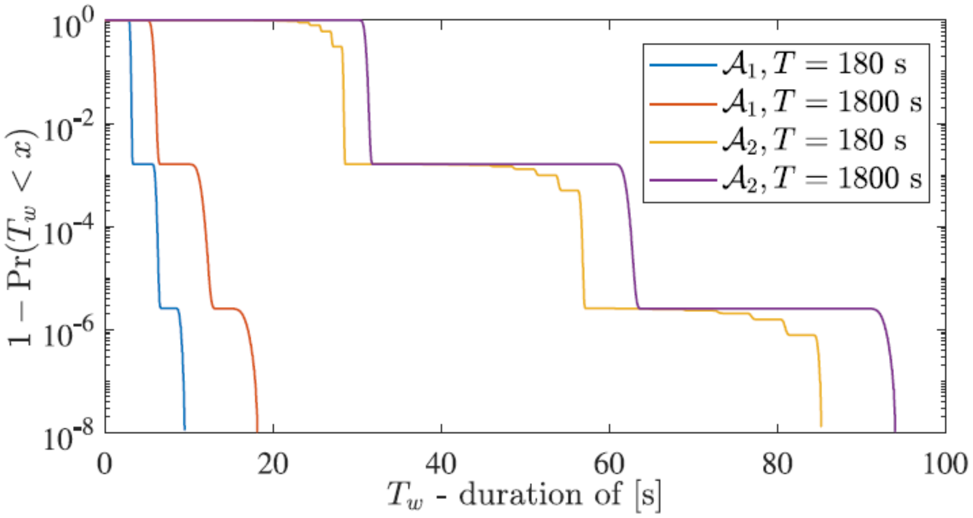
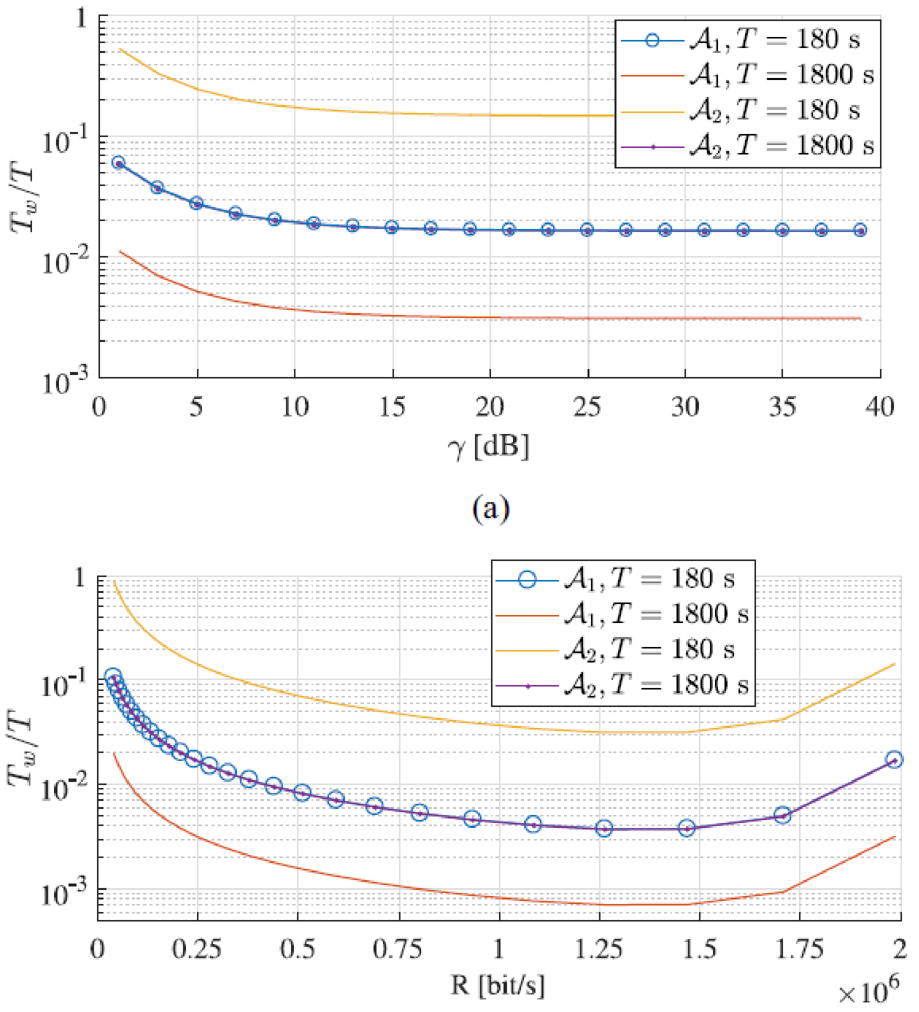


Fig. 10. Complementary CDF of T_w for two different intervals.

Fig 11. illustrates the study of the equipment duty cycle, covering some basic tradeoffs. Fig 11. (a) shows that due to reducing the number of retransmissions, the duty cycle decreases when the channel quality (SNR) increases, and because the SNR is unlikely to occur, the duty cycle reaches saturation for high SNR values. Fig 11. (b) illustrates the duty cycle as a function of the wireless link transmission rate. At low rates, the duty cycle of the equipment increases dramatically.



(a)

(b)

Fig. 11. Duty cycle of the device (a) for different values of SNR and (b) for different rates.

- **Communication Cost**

We construct a collection A_3 with $|A_3| = 20$ accounts. This is done randomly from the active account during T = 1800 seconds. It should be noted that A_2 is a possible implementation of A_3. Fig 12. shows the amount of information downloaded during the 24 hours execution for different values of T and different implementations of the account in A_3. Their effect will be a proportional increase in all quantities. The figure shows that most of the communication costs are due to the size of the account data structure, which is an order of magnitude higher than the size of PoMI, and two orders of magnitude higher than the size of block headers and communication protocol headers.

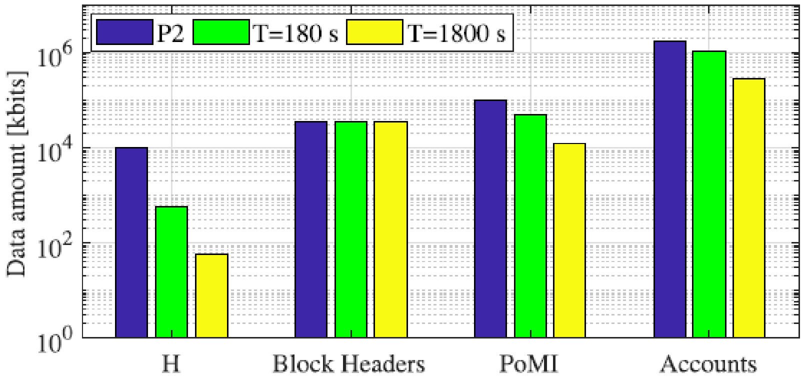


Fig. 12. Amount of information downloaded for $\mathcal{A}_3$, during 24 h, for protocol P2 and for protocol with aggregation with different values of T.

The aggregation gain Γ of several values of the aggregation period T is shown in Fig. 13 and compared with the system simulation. The figure shows a good match between simulation and analysis, and even for small T-values, the benefits are quite impressive. When T → ∞, the observed account will almost certainly be updated over a period of time and will be downloaded on the next transfer. This results in a linear increase in gain as T is large.

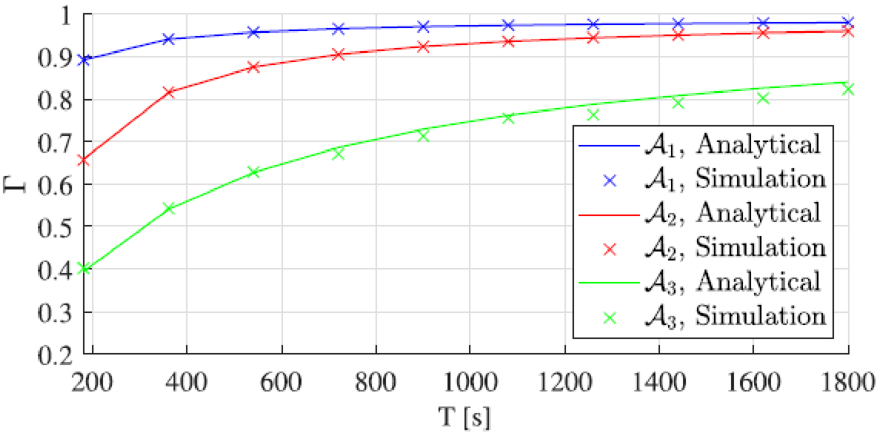


Fig. 13. Gain of the aggregation protocol, analytical expression, and simulation.

IV. CONCLUSION

In this paper, the authors research and analyze the communication cost when blockchain information be sent to a lightweight client. And they also proposed a brand new aggregation scheme to obtain a lower communication cost but it will sacrifice the information delay time. The analysis of this scenario shows the probability distribution of the data structures exchanged over the wireless link and their impact on the total downlink budget. Finally, the results obtained illustrate that if the statistics of account updates and channel status are known, the lightweight client can build a list of events of interest to provide a predictable average communication cost. This sample application illustrates how to apply our findings to improve the privacy of IoT devices. The guidelines presented in this article can be used to design more advanced blockchain lightweight protocols.

V. REFERENCE

[1] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, “Analysis of the communication traffic for blockchain synchronization of IoT devices,” in Proc. IEEE Int. Conf. Commun. (ICC), 2018, pp. 1–7.