Witness-based Approach for Scaling Distributed Ledgers to Massive IoT Scenarios

Nguyen, Lam Duc; Leyva-Mayorga, Israel; Popovski, Petar

Published in:
2020 IEEE 6th World Forum on Internet of Things (WF-IoT)

DOI (link to publication from Publisher):
10.1109/WF-IoT48130.2020.9221269

Publication date:
2020

Document Version
Accepted author manuscript, peer reviewed version

Witness-based Approach for Scaling Distributed Ledgers to Massive IoT Scenarios

Duc-Lam Nguyen, Israel Leyva-Mayorga, and Petar Popovski
Connectivity Section, Department of Electronic Systems, Aalborg University
Aalborg, Denmark
Email: {ndl, ilm, petarp}@es.aau.dk

***Abstract*—Distributed Ledger Technologies (DLTs) are playing a major role in building security and trust in Internet of Things (IoT) systems. However, IoT deployments with a large number of devices, such as in environment monitoring applications, generate and send massive amounts of data. This would generate vast number of transactions that must be processed within the distributed ledger. In this work, we first demonstrate that the Proof of Work (PoW) blockchain fails to scale in a sizable IoT connectivity infrastructure. To solve this problem, we present a lightweight distributed ledger scheme to integrate PoW blockchain into IoT. In our scheme, we classify transactions into two types: 1) global transactions, which must be processed by global blockchain nodes and 2) local transactions, which can be processed locally by entities called *witnesses*. Performance evaluation demonstrates that our proposed scheme improves the scalability of integrated blockchain and IoT monitoring systems by processing a fraction of the transactions, inversely proportional to the number of witnesses, locally. Hence, reducing the number of global transactions.**

***Index Terms*—Distributed Ledgers, Blockchain, IoT, Witness, Environment Monitoring, scalability.**

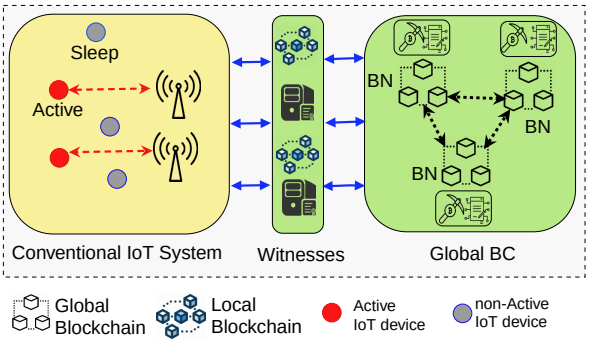


Fig. 1: Overview of our Blockchain-enabled IoT system *wiBlock*. The IoT nodes generate and send the transactions to the base stations, which in turn send them to the witnesses. These decide which transactions must be sent to the GB and process the rest.

I. Introduction

Distributed Ledger Technologies (DLTs) provide high levels of security, accountability, tractability, and privacy to the transmitted data [1]. This is achieved by enabling key functionalities, such as transparency, distributed operation, and immutability [2]. The benefits of DLTs are particularly appealing for Internet of Things (IoT) applications, where large amounts of data are generated and the devices can only implement weak security mechanisms [3].

The trust provided by DLTs is greatly valuable in IoT monitoring applications with a large number of devices. As an example, consider an urban IoT application that monitors the air quality and gas emissions. The data generated by this application is critical, so it must be protected, tractable, immutable, and transparent. Nevertheless, in a traditional monitoring system, the inter-organization sharing the data may be untrusted, complex, unreliable, and non-transparent. Besides, the current IoT-based monitoring systems are centralized, which leads to a single point of failure, where data can be lost or modified [4].

The problems described above may be solved by integrating Blockchain into IoT applications. However, Blockchain architectures were not designed to handle a large number of transactions, which would be generated by naively integrating Blockchain into IoT. Specifically, IoT deployments usually present a star topology, in which the devices communicate directly to the base stations (BS), which then redirects the gathered data to the destination [5] (e.g., from Narrowband IoT (NB-IoT) or LoRa deployments to a cloud server), as shown in the left part of Fig. 1. In the most Blockchain and IoT integration, this same architecture would be used, and the BS would be in charge of communicating with the Blockchain [6]. Thus, every packet generated by the IoT devices would represent a transaction, which can easily overload the Blockchain.

Three main challenges must be overcome to achieve an efficient integration of Blockchain into IoT. First, DLTs use diverse resource-intensive gating functions, for example, *Proof-of-Work (PoW)* and *Proof-of-Stake (PoS)*, while IoT devices are resource-constrained. As a consequence, the processing time of these functions in IoT devices would be restrictive. Second, the widely-used Blockchain arrangement cannot handle the massive transactions generated by IoT devices. For example, Bitcoin network produces 1 MB blocks, roughly once every 10 minutes, with an average size of transaction around 500 bytes, which give 7 transactions per second (tps). In comparison, Visa system performs 2000 tps on average, and an average daily peak of 4000 tps, with a maximum capacity of 56000 tps. Third, the power saving mechanisms of the IoT devices can cause problems during knowledge dissemination and synchronization. For instance, an update may be severely

delayed or even fail to arrive if a device is in sleep mode.

In this paper, we present a witness-based Blockchain system called *wiBlock*, especially designed to integrate Blockchain into resource-constrained IoT applications. It is aimed to solve three of the main problems of traditional IoT monitoring systems, namely trust, scalability, and cost. This is achieved by: 1) enabling the use of DLTs to store IoT data, 2) limiting the number of transactions that must be processed at the Global Blockchain (GB), and 3) eliminating the need for complex computations and supporting sleep-awake mechanisms at the IoT devices, respectively.

The architecture of *wiBlock* is illustrated in Fig. 1, where the IoT devices interact exclusively with the *witness* system, which then may process the transactions locally or communicate directly with the GB. The transactions that must be processed by the GB are called *global transactions*, whereas the transactions that can be verified locally at the *witness system* are called *local transactions*. In order to see the need for this differentiation, consider a pollution monitoring system, in which a number of sensors in a given local area are associated to the same witness. Then a local transaction can be used to send local sensing data from a device associated with the same witness. For instance, the alarm sensor periodically requests gas sensor which collects the concentration of pollutants e.g., SO2, CO2, NO to detect the abnormal condition in air. In order to see the need for a global transaction, note that sensors may wish to store their sensing data to external storage system e.g, IPFS [7] or control a thermostat sensor, which is located in a different area and associated with a different witness to adapt temperature. In this case there is a need to communicate via different heterogeneous networks and record the transaction results to the GB via global transactions. Thus, the witness system reduces the number of transactions that need to be processed by the *GB* and the latency of transaction verification. Furthermore, *wiBlock* allows each IoT device to communicate with several witnesses. This avoids having a single point of failure (i.e., bridge) between the IoT device and the GB, which in turn greatly increases the reliability of the IoT application. For example, Blockchain witness models have been found to be beneficial for Cloud Service Level Agreement [8].

The contributions of this work are as follows:

1) We investigate the possibilities of naively integrating Blockchain directly into resource-constrained IoT systems. We identify some of the major problems that arise in this setup, which illustrate that Blockchain technology is not directly applicable to massive IoT.
2) We propose a new IoT-friendly distributed ledger system named *wiBlock*. It aims to solve the scalability issues of Blockchain in massive IoT environment by defining two types of transactions: global and local.
3) We thoroughly compare the performance *wiBlock* with that of a naive Blockchain and IoT integrated architecture. Our results show that our proposed system enhances the scalability of the GB network.

The remainder of this paper is organized as follows. In Section II, we present the system model, followed by the design of our novel *wiBlock* system in Section III. We present the analysis and performance evaluation of *wiBlock* in Section IV and Section V, respectively. Finally, we conclude the paper in Section VI.

II. System model

We consider an IoT application with k devices. These are deployed uniformly at random in a squared area of interest $A \in \mathbb{R}^2$. The IoT devices generate transactions with the data collected from the environment according to a Poisson process with rate λ.

In the most simple Blockchain and IoT integrated architecture, the transactions are sent to the BS, which then redirects them to the GB. In *wiBlock*, the transactions are sent to the *witness system* instead. This is a set of v witnesses, which have the capacity to verify transactions locally and to communicate with the GB. The time needed for a witness to perform these operations determine its capacity and depend on numerous factors. However, it is out of the scope of this paper to derive their precise values. Transactions are grouped into blocks of size b. Therefore, a new block is created when b new transactions are received at a server.

Witnesses may be either physical or logical entities, hence, their organization is flexible. For simplicity, throughout this paper we assume one witness is deployed at each BS and use these terms interchangeably. The BSs are distributed randomly within A. We denote the set of IoT devices and witnesses as $\mathcal{D} = \{1, 2, \ldots, k\}$ and $\mathcal{W} = \{1, 2, \ldots, v\}$, respectively.

The IoT devices and witnesses communicate through wireless links under a standard path loss model and large-scale (slow) fading. Thus, a transaction is transmitted successfully from IoT device i to a witness $w \in \mathcal{W}$ with probability $p_s(i, w)$. The IoT device i selects the witness w according to a predefined strategy. If the transmission fails, i attempts the transmission to a different witness. This process is repeated until the transaction is confirmed or until a given number of attempts is reached without success.

We consider a simple shadowing propagation model for the communication between IoT devices and witnesses where, for a given transmission power P_t and carrier frequency f, the received power at a distance d is

$$P_r(d) = 10 \log_{10} \left(\frac{P_t G_t G_r \, c^2}{(4\pi f)^2 d^\beta} \right) + N(0, \sigma_{\text{dB}}) \text{ dB} \quad (1)$$

where G_t and G_r are the transmitter and receiver antenna gains, respectively, $c = 3 \cdot 10^8$ m/s is the speed of light, $N(0, \sigma_{\text{dB}})$ is a zero-mean Gaussian random variable (RV) with standard deviation σ_{dB} dB, and β is the path loss exponent.

From there, the outage probability at a given distance and receiver sensitivity γ is

$$p_{\text{out}}(d) = 1 - Q\left(\frac{1}{\sigma_{\text{dB}}} 10 \log_{10} \left(\frac{\gamma (4\pi f)^2 d^\beta}{P_t G_t G_r c^2} \right) \right) \quad (2)$$

and $p_s(i, w) = 1 - p_{\text{out}}(d(i, w))$. Throughout this paper, we assume that the wireless resources are sufficient to support the communication between the IoT devices and the witness

system and do not go into the details of the access protocols. Therefore, collisions caused by simultaneous transmissions from the IoT devices to a witness w can be avoided or resolved if the links toward w are not in outage. Finally, no errors occur in the communication between the witness system and the GB.

III. *WiBlock* Design

This section presents the detailed description of *wiBlock* architectural elements and operation.

A. Witness-based Blockchain System

As illustrated in Fig. 1, the witness-based Blockchain System consists of three main components: the GB, the witness system, and the physical IoT devices. The first action performed by the IoT devices after deployment is authentication. For this, each device $i \in \mathcal{D}$ performs a key exchange procedure with a witness $w \in \mathcal{W}$ to gain the necessary permissions and build secure channels to perform transactions. After authentication, the tuple (i, w) is added by w to the shared registry of the witness system $\mathcal{R}$. For this, w shares the authentication information of i (i.e., credentials) with the rest of the witnesses. By keeping a shared registry, device i can communicate with any witness, even though is registered with w. After authentication, IoT devices collect the data and sign it by using a `SecretKey` $s_{\text{key}}(i)$ that is unique for each i as *Sign(data, $s_{key}(i)$, timestamp τ)*. Next, the transaction is created and transmitted to a *witness* w. Note that this latter witness may be different to the one which i is registered with. Local transactions, denoted as L_l, are exchanged exclusively between w and all the IoT devices registered with it $\{i \in \mathcal{D} : (i, w) \in \mathcal{R}\}$, for which the managers implement a consensus procedure. On the other hand, Global transactions, denoted as L_G, must be sent from a *witness* w to the GB when $(i, w) \notin \mathcal{R}$. These two types of transactions are further described in the following.

1) Local Transactions: These transactions are transmitted from IoT device i to the witness w, whose key management component confirms that $(i, w) \in \mathcal{R}$. Then, this same component checks whether the `PublicKey` $p_{\text{key}}(i)$ of i has been associated with any block in the *local ledger*. If $p_{\text{key}}(i)$ has not been associated with any block, the *witness* w generates a new block for the given i. Then, w arranges the transactions in order, updates the *local ledger*, and a notification feedback message is transmitted to the devices.

2) Global Transactions: These transactions are transmitted from IoT device i to the witness w, whose key management component confirms that $(i, w) \notin \mathcal{R}$. Then, this same component will clarify which *witness* i is registered with. If $\exists w' \in \mathcal{W}$ s.t. $(i, w') \in \mathcal{R}$, the transaction is forwarded to the GB. In case the GB has a block associated with given device i, the transaction will be validated based on the corresponding signature *Sign(data, $s_{key}(i)$, timestamp τ)* and, if the signature is valid, the transaction is appended to the block and transmitted back to the *witness* w'. Note that this type of transactions will be frequently generated when the IoT devices are mobile. For example, cargo, supply chain, and car subsystem monitoring.

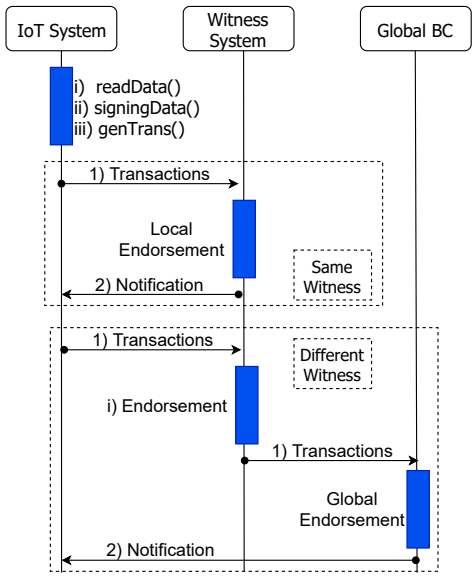


Fig. 2: Transaction flow in *wiBlock*, from generation to confirmation.

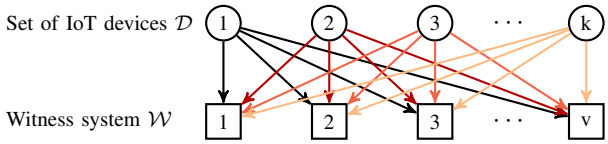


Fig. 3: In *wiBlock*, each IoT device has a list of eligible witnesses. Transactions generated by the IoT devices are sent to a witness in this list, according to the witness selection strategy.

B. Witness Selection

Numerous witness selection strategies can be implemented at the IoT devices and each one may offer different benefits. However, the focus of the present work is to evaluate the benefits of the witness-based architecture, rather than to identify an optimal witness selection strategy. Therefore, we consider the following a heuristic witness selection strategies and evaluate the performance of the witness system. As illustrated in Fig. 3, IoT devices select one of the v available witnesses with probability $1/v$ and transmit the transaction. Then, if the link between IoT device i and witness w is not in outage, the transaction is confirmed. Otherwise, i selects a new witness uniformly at random from $\mathcal{W} \backslash w$ and transmits the transaction. This process is repeated until the transaction is confirmed or until a given number of attempts $l \leq v$ is reached without success. This is the simplest strategy and assumes the IoT devices have no information about the state of the wireless channel toward each witness separately.

IV. Analysis

A. Queuing model of the witness system

We consider a queuing model for witness-based Blockchain network as described in Fig. 4. The witnesses and Blockchain

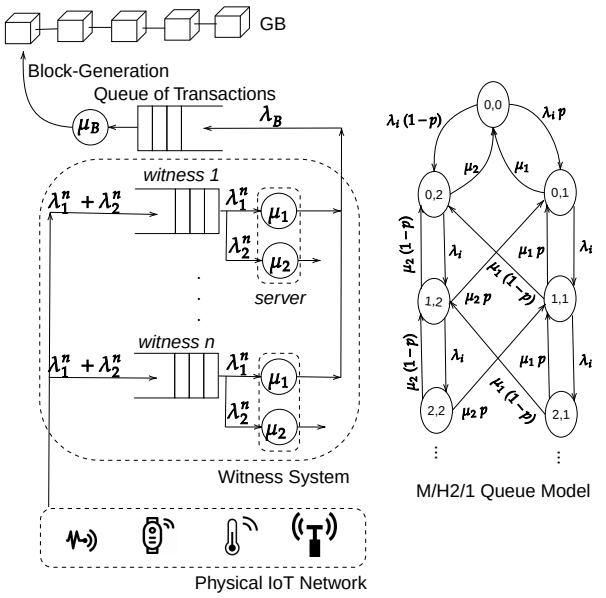


Fig. 4: Witness-based Blockchain queuing model described in Section IV.

are modelled as queuing nodes to capture the number of transactions that must be i) processed locally by witnesses and ii) forwarded to the GB to be processed. We assume that transactions are generated by the IoT devices following a Poisson process. Hence, we denote $\lambda(i)$ as the transaction generation rate at IoT device i.

Let $p(i,w)$ be the probability that i chooses witness w and $p_s(i,w)$ be the probability that the link between i and w is not in outage. Building on this, the average transaction arrival rate at the witness w is

$$\lambda_w = \sum_{i=1}^{k} p(i,w) p_s(i,w) \lambda(i). \tag{3}$$

Hence, the transaction arrival rate of different witnesses depends on the density and location of the deployed IoT devices and witnesses, but also on the witness selection criteria.

The probability $p(i,w)$ depends on the witness selection strategy. For the strategy 1, random selection, let $A(w,u)$ be the matrix of permutations of u elements taken from $\{1-p_s(i,w')\}_{w' \in \mathcal{W} \setminus w}$ with ${}_{(v-1)}P_u$ rows and u columns. The element in row x and column $y \leq u$ of $A(w,u)$ is denoted $a_{xy}(w,u)$. From there, we can calculate $p(i,w)$ as:

$$p(i,w) = \frac{1}{v} + \frac{1}{v!} \sum_{u=1}^{l-1} (v-u-1)! \sum_{x=1}^{{}^{(v-1)}P_u} \prod_{y=1}^{u} a_{xy}(w,u) \tag{4}$$

As mentioned above, generated transactions are either *global* L_G or *local* L_l. We define p as the probability that a transaction sent to a witness is *Global*. Hence, $1-p$ is the probability that a transaction is *local*. Please observe that the value of p only depends on the number of witnesses v and is $p := \Pr[(i,w) \notin \mathcal{R}] = (v-1)/v$.

The transaction processing time is assumed to follow an exponential distribution with service rates μ_1 and μ_2 for *global* and *local* transactions, respectively, and transactions are served according to a first-come first-served (FCFS) policy. Building on this, we model the operation of each witness as an M/H2/1 queue, which means that transactions arrive at the witness w at a rate λ_w and the service time is represented by a two-phase hyper-exponential distribution. With probability p, the first transaction in the queue receives service at rate μ_1, while with probability $1-p$, it receives service rate at rate μ_2. That is, the type of transaction is defined at the beginning of service.

The state of each witness is represented by a pair (m,n), in which m is the total number of transactions in the witness and $n \in \{1,2\}$ is the current service phase, which depends on the type of transaction being served. The stationary distribution of this queue in the witness w can be obtained by Neuts' Matrix Geometric Method [9]. We denote the stationary probability vector as:

$$\boldsymbol{\tau}^{(w)} = \left[\tau_0^{(w)}, \tau_1^{(w)}, \tau_2^{(w)}, \ldots, \tau_k^{(w)}, \ldots\right], \tag{5}$$

where $\tau_m^{(w)}$ is the steady-state probability of m transactions in the witness w. Alternatively, the mean service rate is

$$\mu = \left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)^{-1}, \tag{6}$$

and the offered load to w is $\rho_w = \lambda_w/\mu$. From there, the we calculate the variance of the service time

$$\sigma_w^2 = 2\left(\frac{p}{\mu_1^2} + \frac{1-p}{\mu_2^2}\right) - \frac{1}{\mu^2} \tag{7}$$

and the coefficient of variation $C_w^2 = \mu^2 \sigma_w^2$. Then, the average number of transactions in the queue of w is

$$L(w) = \sum_{m=0}^{\infty} m\tau_m^{(j)} = \rho_w + \left(\frac{1+C_w^2}{2}\right) \frac{\rho_w^2}{1-\rho_w}. \tag{8}$$

Then, the number of *local transactions* and *Global transactions* handled by w are, respectively,

$$L_g(w) = pL(w) \tag{9}$$

and

$$L_l(w) = (1-p)L(w) = L(w) - L_g(w). \tag{10}$$

B. Global Blockchain (GB) System

We model the GB as a modified $M/G^B/1$ queue as in [10]. Let L_g and T_g be the RVs that define the number of transactions in the Blockchain queue and the confirmation time. We are interested in finding their mean values. For this, we define b to be the maximum number of transactions in a block (i.e., the maximum block size). Hence, transactions are grouped into blocks and a new block is created when there are b transactions in the Blockchain server.

Given that p is the probability that a transaction sent to a witness is processed at the GB, the transaction arrival rate at the GB from the v witnesses in the IoT deployment is

$$\lambda_B = \sum_{w=1}^{v} \lambda_w p. \tag{11}$$

We denote U as of the block generation time (i.e., the time it takes to generate a block) at the GB. Then,

We define U to be the continuous RV of the processing (i.e., service) time of a block at the GB. Hence, the system is stable and a limiting probability exists if and only if $\lambda_B E[U] < b$.

The cumulative distribution function (CDF) and the probability density function (pdf) of U are denoted $G(x)$ and $g(x)$, respectively. We use these to calculate the hazard rate of U as

$$\theta(x) = \frac{g(x)}{1 - G(x)}. \tag{12}$$

Next, we define $L_g^s(t)$ as the number of transactions in server at time t, $L_g^q(t)$ as the number of transaction in the queue at time t, and $X(t)$ as the elapsed service time of the current transaction at t. From [11], we define

$$P_{m,n}(x,t)dx = \Pr\left[L_g^s(t) = m, L_g^q(t) = n, \right. \\ \left. x < X(t) \leq x + dx\right] \tag{13}$$

to be the joint probability that, at time $t \geq 0$, there are $m \in \{0,1,2,...,b\}$ and $n \in \{0,1,2,...,x\}$ transactions in server and queue, respectively, and the elapsed service time lies between x and $x + dx$. Next, we denote $P_{m,n}(x) = \lim_{t\to\infty} P_{m,n}(x,t)$ and consider the two following cases. In the first one we have $\frac{d}{dx}P_{m,n}(x) = -\left[\lambda_B + \theta(x)\right]P_{m,n}(x) + \lambda_B P_{m,n-1}(x)$, for $0 \leq m \leq b$ and $n \geq 1$, which shows that the number of transactions in the server and the queue does not change during a small interval. In the second one we have $\frac{d}{dx}P_{m,0}(x) = -\left[\lambda_B + \theta(x)\right]P_{m,0}(x)$, for $0 \leq m \leq b$, which occurs when a transaction arrives at the system with 0 transactions in the queue. For the purposes of our study, it is sufficient to calculate the mean confirmation time as

$$E[T_g] = \Bigg[\lambda_B^2 E[U^2] - b(b-1) - 2\left(b - \lambda_B E[U^2]\right) \\ + \sum_{n=0}^{b-1} \alpha_n \Big(\lambda_B E[U^2](b-n) + 2bE[U](b-n) \\ + E[U]\left(b^2 - b - n^2 + n\right)\Big)\Bigg] \frac{1}{2\lambda_B(b - \lambda_B E[U])}, \tag{14}$$

where $\alpha_n = \sum_{m=0}^{b} \int_0^\infty P_{m,n}(x)\theta(x)dx$. The interested reader is referred to [10] for the fully detailed Blockchain queuing model.

V. Performance Evaluation

In this section, we use the queuing models described in Section IV to evaluate the performance of *wiBlock* in terms of scalability. For this, we obtain the maximum transaction generation rate, along with the mean confirmation time and ledger size for both, the local and global Blockchain. We use the performance of a naive Blockchain and IoT integrated architecture, where the IoT devices communicate directly to the GB, as a benchmark. The mean results regarding the connectivity of the IoT devices with the witness system are obtained by a large number of Monte Carlo simulations and then used as an input to the queuing models.

TABLE I: Parameter settings for the performance evaluation.

Parameter	Symbol	Value
Area of deployment	A	100×100 m^2
Number of IoT devices	k	500
Number of witnesses	v	$\{2, 3, ..., 10\}$
Carrier frequency	f	914 MHz
Transmission power	P_t	0.28183815 W
Antenna gains	G_t, G_r	1
Receiver sensitivity	γ	$3.652 \cdot 10^{-10}$ W
Standard deviation of shadow fading	σ_{dB}	6 dB
Path loss exponent	β	3
Block size	b	1000 transactions

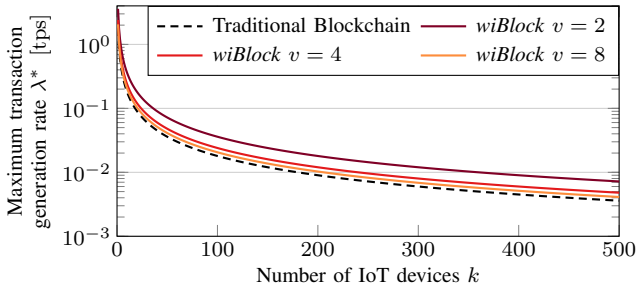


Fig. 5: Maximum transaction generation rate per IoT device λ^* for traditional Blockchain IoT and *wiBlock* with random witness selection.

In our analysis, each device generates transactions at a rate $\lambda(i) = \lambda$ for all $i \in \mathcal{D}$. The block generation time U is exponentially distributed with parameter $\mu_B = 1.8 \cdot 10^{-3}$ blocks per second. So we have $g(x) = \mu_B \exp(-\mu_B x)$, $E[U] = 1/\mu_B$, and $E\left[U^2\right] = 1/\mu_B^2$. Furthermore, we define the default block size to be $b = 1000$ transactions. The rest of relevant parameters are listed in Table I; these values are used unless otherwise stated.

For the selected parameter settings, the GB system is stable when $\lambda_B^* = b/E[U] = 1.8$. Building on this, from (11) we have that $\lambda < b/(kE[U]) = 1.8/k$ must hold for the GB to be stable in a traditional Blockchain architecture with k identical IoT devices. Conversely, for *wiBlock* with random witness selection, we have that only a fraction $p = (v-1)/v$ of the transactions must be sent to the GB. Hence, assuming no wireless channel errors occur and all the generated transactions are sent to a witness (i.e., $\sum_{w=1}^{v} \lambda_w = k\lambda$), the maximum load per IoT device that *wiBlock* can handle is

$$\lambda < \frac{bv}{kE[U](v-1)} = \frac{1.8v}{k(v-1)} = \lambda^*(v), \tag{15}$$

as shown in Fig. 5 for $v = \{2, 4, 8\}$.

Hence, from the GB perspective, *wiBlock* allows to deploy $1/p = v/(v-1)$ times more IoT devices than the naively integrated approach, as illustrated by Fig. 6. Note that the greatest gains in the scalability are obtained when v is small, however, other factors such as the area coverage and process-

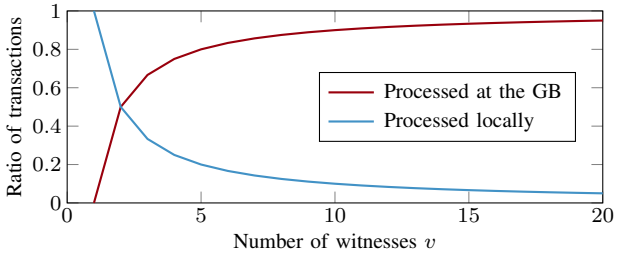
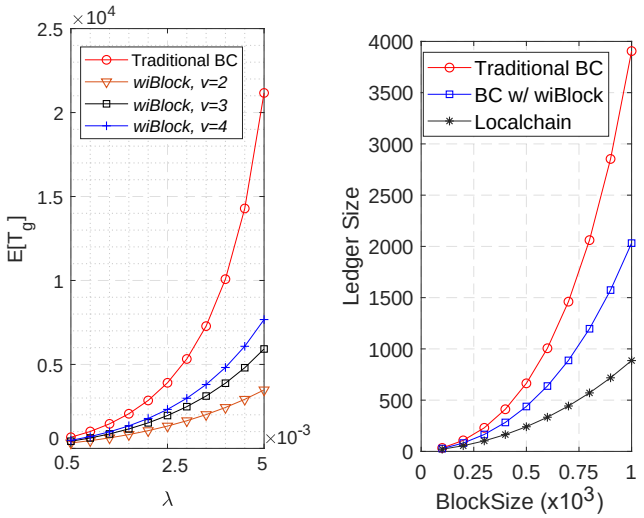


Fig. 6: Ratio of transactions processed at the GB and at the witness system as a function of the number of witnesses v.



(a) (b)

Fig. 7: (a) Mean transaction confirmation time $E[T_g]$ as a function of the transaction generation rate at the IoT devices λ and (b) ledger size at the GB and at each witness for $v = 2$ as a function of the block size b for traditional Blockchain IoT and *wiBlock*.

ing capacity of the witness system must be taken onto account to select adequate values of v.

Next, we evaluate the mean transaction confirmation time at the GB $E[T_g]$. Note that, in case a single witness is deployed in the system, all the transactions generated by the IoT devices will be considered as local transactions and processed locally. This can overload the witness, depending on its capabilities. In particular, the witness is stable if and only if the load offered to the witness is $\lambda_1 < \mu_2$. Furthermore, deploying a single witness does not provide the necessary wireless coverage. That is, the more witnesses are deployed, the higher the probability of being able to communicate to, at least, one of them. Hence, we consider the cases where at least two witnesses are deployed, as shown in Fig. 7a for $v = \{2, 3, 4\}$.

As Fig. 7a shows, the witness system reduces the number of transactions sent to the GB and, as a consequence, greatly reduces the transaction confirmation time. Besides, the ledger size is considerably reduced, depending on the number of witnesses. This can be seen in Fig. 7b for $v = 2$, where the ledger size of the GB is half of that with the traditional Blockchain and IoT integration, and the local ledger size at each witness is $1/v^2 = 1/4$ of it.

VI. Conclusion

In this paper, we presented and evaluated the performance of a novel witness-based Blockchain system for IoT applications. As a starting point, we described the benefits of integrating Blockchain into IoT and the main challenges that must be overcome to achieve this integration. Building on these, we designed *wiBlock*, an IoT-friendly distributed system that incorporates a witness system to address scalability issues of Blockchain. The scalability gains provided by *wiBlock* are achieved by processing some of the transactions generated by the IoT devices locally, at the witness system. Our results show that the witness system greatly reduces the number of transactions transmitted to the Blockchain network and the transaction confirmation time. Future work includes the design of witness selection algorithms and implement *wiBlock* in real testbed to further exploit the benefits provided by the witness system.

VII. Acknowledgement

This work has been in part supported by the European Research Council (Horizon 2020 ERC Consolidator Grant Nr.648382 WILLOW).

References

[1] P. Danzi, A. E. Kalor, R. B. Sorensen, A. K. Hagelskjær, L. D. Nguyen, C. Stefanovic, and P. Popovski, "Communication aspects of the integration of wireless iot devices with distributed ledger technology," *IEEE Network*, vol. 34, no. 1, pp. 47–53, 2020.

[2] M. Swan, *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc, 2015.

[3] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.

[4] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, Jan. 2017.

[5] T. Maksymyuk, S. Dumych, M. Brych, D. Satria, and M. Jo, "An IoT based monitoring framework for software defined 5G mobile networks," in *Proc. of the 11th International Conference on Ubiquitous Information Management and Communication*, 2017, pp. 105:1–105:4.

[6] P. Danzi, A. Kalør, Č. Stefanović, and P. Popovski, "Delay and communication tradeoffs for blockchain systems with lightweight IoT clients," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, Apr. 2019.

[7] M. S. Ali, K. Dolui, and F. Antonelli, "Iot data privacy via blockchains and ipfs," in *Proceedings of the Seventh International Conference on the Internet of Things*. ACM, 2017, p. 14.

[8] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2019, pp. 1567–1575.

[9] W. Stewart, *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton University Press, 2009.

[10] Y. Kawase and S. Kasahar, "Transaction-confirmation time for bitcoin: A queueing analytical approach to blockchain mechanism," in *Proc. International Conference on Queueing Theory and Network Applications*, 2017, pp. 75–88.

[11] M. Chaudhry and J. Templeton, "The queuing system M/G^B/l and its ramifications," *European Journal of Operational Research*, vol. 6, no. 1, pp. 56–60, 1981.