Received February 2, 2019, accepted February 26, 2019, date of publication March 6, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903271

A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain

YINING HU[1,2], **AHSAN MANZOOR**[3,4], **PARINYA EKPARINYA**[5], **MADHUSANKA LIYANAGE**[3,6], **(Member, IEEE)**, **KANCHANA THILAKARATHNA**[5], **GUILLAUME JOURJON**[5], **AND ARUNA SENEVIRATNE**[2]

[1]Data61-CSIRO, Eveleigh, NSW 2015, Australia
[2]University of New South Wales, Kensington, NSW 2033, Australia
[3]University of Oulu, 90014 Oulu, Finland
[4]Rovio Entertainment, 02150 Espoo, Finland
[5]The University of Sydney, Camperdown, NSW 2006, Australia
[6]University College Dublin, Dublin D4, Ireland

Corresponding author: Yining Hu (yining.hu@data61.csiro.au)

This work is performed under the framework of RESPONSE 5G (Grant No: 789658) project European Union and 6Genesis Flagship (Grant no. 318927) project funded by the Academy of Finland.

ABSTRACT Digital banking as an essential service can be hard to access in remote, rural regions where the network connectivity is unavailable or intermittent. The payment operators like Visa and Mastercard often face difficulties reaching these remote, rural areas. Although micro-banking has been made possible by short message service or unstructured supplementary service data messages in some places, their security flaws and session-based nature prevent them from wider adoption. Global-level cryptocurrencies enable low-cost, secure, and pervasive money transferring among distributed peers, but are still limited in their ability to reach people in remote communities. We propose a blockchain-based digital payment scheme that can deliver reliable services on top of unreliable networks in remote regions. We focus on a scenario where a community-run base station provides reliable local network connectivity while intermittently connects to the broader Internet. We take advantage of the distributed verification guarantees of the *Blockchain* technology for financial transaction verification and leverage smart contracts for secure service management. In the proposed system, payment operators deploy multiple proxy nodes that are intermittently connected to the remote communities where the local blockchain networks, such as Ethereum are composed of miners, vendors, and regular users. Through probabilistic modeling, we devise design parameters for the blockchain network to realize robust operation over the top of the unreliable network. Furthermore, we show that the transaction processing time will not be significantly impacted due to the network unreliability through extensive emulations on a private Ethereum network. Finally, we demonstrate the practical feasibility of the proposed system by developing Near Field Communication (NFC)-enabled payment gateways on Raspberry-Pis, a mobile wallet application and mining nodes on off-the-shelf computers.

INDEX TERMS Blockchain, delay-tolerant network, digital banking, remote regions.

I. INTRODUCTION

Internet access has evolved to be one of basic needs of humans primarily due to the variety of over-the-top services delivered through Internet. In fact, by June 2017, 51% of global population had access to basic Internet [1] and nearly one billion users to high-speed fixed broadband Internet [2]. Nevertheless, there are still more than four billion people, mostly in developing regions, who do not have access to or can only intermittently connect to the broader Internet [3]. This has led to the development of distributed networking techniques, such as ad-hoc networks and delay-tolerant networks that utilize the opportunistic connectivity to provide some level of service in the past [4], [5]. Another proposed solution for remote regions is Community-Run Base Stations (CBSs) for local-area connectivity,

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Perara.

for example, Nokia Kuha base stations [6] and Telstra small cells [7] that enable connectivity to the broader Internet via Satellite connectivity. However, both these approaches do not guarantee the connectivity at all time.

In spite of these network irregularities, the ubiquitous availability of connectivity is taken for granted in the development and management of many services such as finance, entertainment, or health-care. As a result, customers of digital services and businesses in rural areas often face operational challenges associated with network connectivity, mainly due to the lack of robustness in digital service design and management. For example, none of the popular online payment schemes, including credit cards, will work well in these connectivity-restricted environments. Since all of these services rely on real-time verification and processing for security purposes. In fact, Tapscott and Tapscott [8] estimated that nearly two billion people worldwide still do not have access to basic banking services. Despite the paramount importance of online banking for economic growth, little has been done to deliver banking services to connectivity-restricted environments, due to the inability to meet the security requirements locally without connecting to the centrally controlled databases [9], [10].

Cryptocurrencies have received significant attention in recent years and are known for their ability to distributedly verify transactions [11], [12]. The technology behind, *Blockchain*, is achieved by hard-coded software programs and enables peer democracy to settle transactions. However, like most other pervasive services of today, cryptocurrencies also require continuous network connectivity to constantly exchange large volumes of data among the collaborators of the service. For instance, a typical Bitcoin client uses around 200 MB of data per hour [13] while the Ethereum blockchain size surpassed Bitcoin in June 2017 [14]. Moreover, partitioning of the network due to interruptions increases the chances of malicious activities and hinder the security guarantees of the blockchain [15]–[18]. As a result, cryptocurrency-based solutions in its current form do not support to provide online banking services for connectivity restricted environments.

In this paper, we take advantage of distributed verification guarantees of *Blockchain* technology to design a novel digital payment system that can be deployed over the top of a network with a very low quality of service guarantees. Our focus is at remote rural villages with community-run base stations such as Nokia Kuha [6] that are connected to the public Internet via unreliable satellite links. We propose an Ethereum-blockchain-based local transaction verification scheme for online payment operators such as VISA and MasterCard and also leverage smart contracts for service management.

The paper makes the following contributions:

- The design of a low-cost and secure digital payment scheme based on a private Ethereum blockchain.
- The use of smart contracts for payment service management such as user account initiation, interactions with credit operator and management of rewards for blockchain miners.
- Probabilistic modelling of the proposed system to devise robust design parameters for ensuring reliable functionality under regular as well as extreme operating conditions. Furthermore, we show that transaction processing is highly related to block creation, and with a sufficient block size, transaction arrival rate does not affect transaction processing time.
- Through extensive emulations on a private Ethereum platform, we show that network unreliability would not slow down transaction processing due to Proof-of-Work (PoW) difficulty adjustment and also the average block generation time is relatively stable even with unreliable availability of miners.
- Demonstrates practical viability of the proposed system through a prototype implementation of NFC (Near Field Communication) enabled payment gateways on Raspberry Pis and a mobile wallet on an off-the-shelf mobile devices.

The remainder of the paper is organized as follows: Section II provides background information on CBS, micro-payment systems. Section III describes the system architecture and Section IV presents models of the system design, followed by Section V that validates of our models and evaluates our solution over a local blockchain deployment. Section VI demonstrates a prototype implementation with multiple test results. Section VII discusses the related work. Section VIII highlights insights on future directions and finally Section IX concludes the paper. Additional key technical details of Ethereum blockchain are provided in Appendix.

II. BACKGROUND

In this section, we introduce the concept of CBSs and existing solutions to deliver digital banking to remote regions.

A. COMMUNITY-RUN BASE STATIONS (CBS)

Despite the advances in mobile technology, almost all countries still have disconnected rural areas where the mobile coverage is not available. Most mobile operators are reluctant to provide service to these areas due to various reasons such as limited roadway connectivity, high maintenance cost, lack of security for infrastructure, inability of secure return of investment.

To provide connectivity for such areas, operators are now using CBSs that can be operated by anyone in the rural area. A CBS requires a limited backhaul connectivity, which for example can be achieved via a satellite link. CBSs are being deployed in various countries and regions such as Europe [19], South America [20], and Australia [7]. Nokia is already building and selling the Kuha Mobile Network [6] around the world.

B. DIGITAL BANKING IN REMOTE REGIONS

To reduce the service cost and enable payments that involve very small sums of money, a number of micro-payment schemes were proposed. Some of them leverage SMS or USSD of the cellular networks, for instance, the Bank for Agriculture and Agricultural Co-operatives (BAAC) in Thailand [9] and the M-Pesa Service in Kenya [10]. However, SMS messages are easily spoofable and hence require additional user verifications for security, and USSD could be affected by session time-outs. Other micro-banking systems are the early-generation token payment platforms secured by time-lock puzzles, e.g. PayWord and MicroMint [21]. IBM also proposed a Micro-iKP protocol for frequent micropayments [22] using "coupons" sent between players and verified by the bank. This scheme however, increases the communication cost.

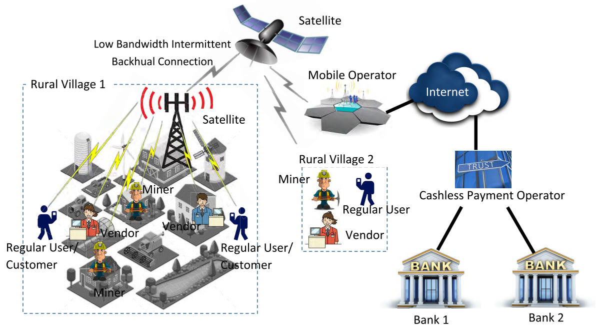
III. SYSTEM ARCHITECTURE

The primary objective of the proposed architecture is to enable cheap, cashless payments for remote villages that have intermittent Internet connectivity. We consider the scenario where local network operators provide wireless coverages within remote communities, using technologies similar to Nokia Kuha [6]. The backhaul connection to/from outside of the communities is a low-bandwidth connection, e.g., a satellite link, that does not provide robust service guarantees.

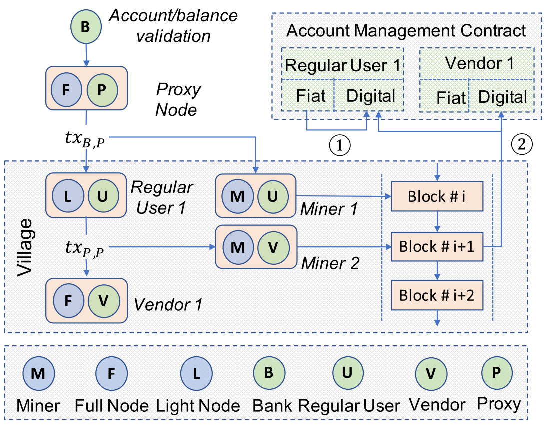
We denote a set of payment operators as $O = \{o_1, o_2, o_3, \ldots\}$, of which each o_i controls a set of proxy nodes, $P_i = \{p_{i1}, p_{i2}, p_{i3}, \ldots\}$. The proxy nodes each connects a corresponding village in $V_i = \{v_{i1}, v_{i1}, v_{i1}, \ldots\}$ via the backhaul connection. As the village networks are relatively isolated, we propose to deploy blockchains for transaction processing in that they can be operated by individuals without any centralized authority. Our scenario also fits in the conditions of a public-permissioned blockchain according to the guidelines provided by Wust *et al.* [23]. For local blockchain deployment, we propose to use Ethereum [24] for its fast, decentralized consensus and verification guarantees. The village blockchains are independent from each other and operate on different Tokens.

As illustrated in FIGURE 1a, the proposed framework enables *Regular Users* or *Customers* to perform cashless financial transactions with *Vendors* in the village irrespective of the connectivity to the proxy. In addition, some villagers can participate in the decentralized verification and become *Miners*. Regular users can join the service by simply installing a mobile application on their personal devices, which operates a blockchain light node as shown in Figure 1b. Vendors and miners will be required to have additional hardware to run full nodes and mining nodes respectively. The proxies are full nodes themselves to avoid creating forks when rejoining the village blockchain. The proxy nodes continue to operate regardless of the condition of backhaul connection. There is no restriction on users joining the system. The account management and mining reward distribution are operated by proxies. Instead of the default *Ether* generation scheme, we propose to use a *Token*-based local currency to avoid inflation in the local economy.

The integration of *Smart Contracts* for admission control and the Token based service management is unique as it automatically imposes restrictions on users entering the



(a)



(b)

FIGURE 1. Overview of the proposed system. (a) System architecture. (b) Transaction flows.

system and makes it easier for the payment operators to manage user accounts, even when local blockchain network is disconnected from the proxy node.

A. TRANSACTION FLOW

FIGURE 1b illustrates transaction flows of the proposed framework. There are two types of transactions associated with regular users: i) regular transactions ($tx_{P,P}$) between regular users and vendors, and ii) currency exchange transactions ($tx_{B,P}$) for converting real money (fiat currency) to Tokens. To perform a regular transaction, every regular user first needs to load their digital wallet (associated with the mobile app) with Tokens. As shown in FIGURE 1b, once validated by a bank, the proxy issues a transaction $tx_{B,P}$, which is picked up by Miner 1 and included in Block #i. Once $tx_{B,P}$ is confirmed, the code inside the *Management Contract* is triggered and Regular User 1's Token account is updated accordingly as shown in Step ①. Next, Regular User 1 sends Vendor 1 a transaction $tx_{P,P}$ to obtain the service, which is collected by Miner 2 and confirmed in Block #i+1. Each regular transaction is treated equally, processed and authorized by the blockchain miners irrespective of the proxy node's presence. During the proxy's next synchronization, once it reaches Block #i+1, the smart contract is executed locally and the Token accounts of both Regular User 1 and Vendor 1 are updated as shown in Step ②. The proxy also creates two accounts for itself, just like other blockchain nodes.

If an outside party without Tokens wants to make transactions with people in the village, it first submits a request to the proxy. Upon validation, the proxy node converts the payment to the Token currency locally and redirects it to the target receiver. When Regular User A in Village A initiates a transaction with Vendor B in a Village B, Proxy A in Village A receives the request and notifies Proxy B in Village B, who settles the transaction in Village B and confirms with Proxy A. If Regular User A moves to Village B, she has to exchange new Tokens for spending in Village B as a unique Token is designated to each individual village.

B. SMART CONTRACT BASED SERVICE MANAGEMENT

To avoid creating forks during disconnection, the payment operators deploy passive full nodes (Appendix A) as proxies. To track and manage the system operation, the payment operators create a smart contract, i.e. the *Management Contract*, to record account types, user balances in both fiat currency and Token currency, as well as distributing mining rewards. The proxy gets notified via the smart contract when a new user joins the system or when miners find blocks, and accesses the latest version of the ledger when it is connected. States in the smart contract are constantly updated regardless of the network condition between the village and the proxy. When connection is established, the proxy synchronizes with other blockchain nodes, updates user balances and processes currency exchange requests by issuing *currency-exchange transactions* ($tx_{B,P}$ in FIGURE 1b).

The *user* struct maps user type and balance to the address of the registering user. (cf. Algorithm 1) Users register themselves to the system by calling the *registerUser* method, which notifies the proxy node. Whenever the *registerUser* method is called, an event containing the user address and the *user* struct is sent along with the notification. A user calls the *sendToken* function for internal transfers of Token within the village and user balance is tracked via the *Balance* function call. The *createToken* function call initializes the contract with initial supply Tokens to the contract creator, i.e. the proxy. During synchronization, for each new block, the *blockReward* function transfers reward to the miner accounts. Token conversion is performed via *convertToken*.

Algorithm 1 Management Contract

```
mapping (address => user) userList
Init: struct {
    string type
    uint256 balance
} user
Event registerUser(Requester, user);
Function registerUser(Requester, userType):
    users[Requester].type = userType
    users[Requester].balance = 0;

Function sendToken(Sender,Receiver, Amount):
    if balanceOf[Sender] > Amount then
        balanceOf[Receiver] += Amount
        balanceOf[Sender] -= Amount
        return TRUE
    else
        return FALSE
    end

Function Balance(Requester):
    return balanceOf[Requester]

Function createToken(InitialSupply):
    balanceOf[Owner] = InitialSupply

Function convertToken(Requester, Amount):
    if balanceOf[Requester] > Amount then
        balanceOf[Requester] += Amount
        balanceOf[Owner] -= Amount
        return TRUE
    else
        return FALSE
    end

Function blockReward():
    balanceOf[Blockgenerator] += 1
```

IV. SYSTEM MODELLING AND DESIGN

We first probabilistically model transaction processing of the local blockchain system and the synchronization delay of the proxy node. We also model the overall deployment and operational costs of one such system and use the models to design system properties to enable an efficient and reliable local area payments under the given network constraints.

A. ASSUMPTIONS

We make the following assumptions without loss of generality:

1) All individual mining nodes have equal and stable computational power.
2) Block size is sufficient to include all transactions for immediate processing as the transaction rate within a village can be considerably lower compared to public Ethereum blockchain.
3) Zero transaction fees and no rewards for mining stale blocks as in the proposed system rewarding is controlled by the payment operator using Tokens.
4) Network bandwidth available within the village is sufficient for all blockchain related data traffic.

B. MODELLING TRANSACTION PROCESSING

Regular transaction arrival r_{t_i}, where t_i is the transaction initiation time, follows a Poisson distribution as observed in [25]. We denote the average arrival rate as λ_t transactions per second (tps). Block generation time T has been shown to be exponentially distributed with probability density function $G(T) = \lambda e^{-\lambda T}$, where $\lambda = \frac{1}{E[T]}$ [26]. In Ethereum, the expected block time $E[T] \approx 12s$. Since we assume block size to be large enough (cf. assumption 2), there are no pending transactions as all new transactions arriving in the current mining session are included in the next block. Therefore, transaction processing time $t_p = T - t_i$, where $t_i, t_p \in (0, T]$. We further analyze the distribution of block time and transaction processing time under different arrival rates in Section V-B. Transaction throughput can also be calculated as $\frac{\sum_{\forall t_i \leq T} r_{t_i}}{T}$. With transaction size being s_t bits, a block needs to accommodate $s_b = s_t \sum_{\forall t_i \leq T} r_{t_i}$ bits of regular transactions, averaging $\bar{s_b} = \lambda_t s_t E[T]$ bits over multiple sessions.

There are *currency-exchange transactions* with the proxy full node when it is connected. We model currency-exchange transaction arrival e_{t_i} similar to regular transactions with average arrival rate λ_e and size s_e. With these transactions, a block size is $s_b' = s_t \sum_{\forall t_i \leq T} r_{t_i} + s_e \sum_{\forall t_i \leq T} e_{t_i}$, with the average being $\bar{s_b'} = (\lambda_t s_t + \lambda_e s_e) E[T]$.

C. MODELLING THE SYSTEM COST

We derive cost models from the payment operators' point of view for mining rewards and system operation using the metrics in [27]. We omit transaction validation and storage costs in our calculations as they are only a small fraction of mining equipment cost and mining itself [27].

1) REWARDING COST

Although the block reward is an incentive for miners, it is an extra cost for the payment operators. We use R to denote the block reward, meaning that a payment operator has to spend R Tokens for each valid block. We calculate the rewarding cost as $C_R = nR$, where n is number of blocks.

2) NETWORK RESOURCES

Since we assume ideal network conditions (cf. assumption 4) locally, we only model the network resource usage of the backhaul network. When connected, the proxy full node synchronizes past transactions and processes money exchange requests. We define one service period as $T_S = T_C + T_U$, where T_C and T_U stand for the durations of connection and disconnection respectively.

We denote the backhaul network bandwidth requirement as BW and cost as C_{BW} in Token per bit. Then, the expected network connectivity cost would be $C_N = C_{BW} T_C BW$ during one service period.

3) SYSTEM COST

To sum up, the overall system cost to the payment operator is the sum of all the aforementioned costs. Since cost components are defined with different time units, we calculate the overall cost within time period T_0, which equals x_b blocks and x_s service periods as:

$$C_{All} = C_R + C_N x_s = R x_b + C_{BW} T_C BW x_s. \tag{1}$$

D. MODELLING THE PROXY NODE SYNCHRONIZATION

Regular transactions arrive when the proxies are connected via the back-haul link and when disconnected, while currency-exchange transactions only occur when the proxies are disconnected. We here assume the connection and disconnection periods to be much longer than block time. We use n_U ($\sum_{i=1}^{n_U} T_i = T_U$) and n_C ($\sum_{i=1}^{n_C} T_i = T_C$) to represent number of blocks created when connected and disconnected respectively. Transactions committed during one disconnected period can add up to a total size of s_U. Substituting the block size derived in Part IV-B, we obtain,

$$\begin{aligned} s_U &= \sum_{j=1}^{n_U} s_{b_j} = \sum_{j=1}^{n_U} s_t (\sum_{\forall t_i \leq T_j} r_{t_i}) \\ &= s_t \sum_{j=1}^{n_U} \sum_{\forall t_i \leq T_j} r_{t_i} = s_t \sum_{\forall t_i \leq T_U} r_{t_i} \approx \lambda_t s_t T_U \end{aligned} \tag{2}$$

We do not consider network delay or communication overhead when the proxy node establishes connections with the village network. We use t as the time passed since the beginning of the connection and denote data remaining to be synchronized as s_r,

$$\begin{aligned} s_r &= s_U + (s_t \sum_{\forall t_i \leq t} r_{t_i} + s_e \sum_{\forall t_i \leq t} e_{t_i}) - BWt \\ &\approx s_U + (\lambda_t s_t + \lambda_e s_e)t - BWt \end{aligned} \tag{3}$$

When new transactions arrive, synchronization only finishes when the connection closes. When BW is big enough, there exists a time t_0 ($t_0 \leq T_C$), from which data remaining to be synchronized reaches 0, i.e., $s_r = s_U + (\lambda_t s_t + \lambda_e s_e)t_0 - BWt_0 = 0$. Hence,

$$t_0 = \frac{s_U}{BW - (\lambda_t s_t + \lambda_e s_e)}. \tag{4}$$

To find the range of BW that satisfies Equation 4, we let $t_0 \leq T_C$, i.e., $\frac{s_U}{BW - (\lambda_t s_t + \lambda_e s_e)} \leq T_C$, from which we obtain

$$BW \geq (\lambda_t s_t + \lambda_e s_e) + s_U / T_C. \tag{5}$$

When BW is small, i.e., $BW < (\lambda_t s_t + \lambda_e s_e) + s_U/T_C$, there will always be data left when connection closes. Over time new transactions accumulate and the proxy node cannot access the latest version of the ledger.

E. MINING NETWORK DESIGN

We now find ranges of multiple mining network parameters including number of miners and their connectivity to ensure reliability and security.

1) MINER OUTAGES

Miners are incentivized to work, but they may join or leave the network spontaneously (churn) [28]. We denote the total number of online miners as l_{on}, $l_{on} \leq l_m$. The probability of one miner going offline in each time slot is $p_1, p_2, p_3, \cdots, p_{l_m}$. Hence the probability that $X = k$ miners are offline in the same time slot can be represented by a Poisson Binomial Distribution:

$$P_r(X = k) = \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j), \tag{6}$$

where the set F_k contains all subsets of k integers that can be selected from $\{1, 2, 3, \cdots, l_m\}$, and A and A^c are complementary. If all miners have the same chance of going offline, i.e. $p_1 = p_2 = p_3 = \cdots = p_{l_m} = p_d$, the expectation of l_{on} is:

$$E[l_{on}] = l_m - E[X] = l_m(1 - p_d), \tag{7}$$

or equivalently,

$$l_m = \frac{E[X]}{p_d} = \frac{E[l_{on}]}{1 - p_d}. \tag{8}$$

We demonstrate how network churn affects the blockchain system performance in Section V-C.

2) MINING REWARD

Payment operators determine the mining reward R according to the status of local economy and its budget. It becomes harder for each individual miner to find a valid block when there are more competitors (cf. assumption 1). As the expected reward reduces, mining becomes less profitable. We calculate the minimum mining reward needed to keep miners incentivized. We denote the mining cost as η per hash, similar to [29]. With individual hash rate being h, each miner's operational cost per second is ηh. The expected hash cost of the entire network per second C_H when all miner are online (worst case) is $C_H = l_m \eta h$. Since miners are equal, their expected revenue per mining round is $R' = R/l_m$. We do not consider rewards provided for stale blocks (cf. assumption 3). We derive the expected profit per mined block for each individual as:

$$\Pi = R' - \frac{C_H T}{l_m} = \frac{R}{l_m} - \eta h T. \tag{9}$$

Clearly, for mining to be profitable, Π should be greater than 0, i.e., R should satisfy:

$$R > l_m \eta h T. \tag{10}$$

3) MINIMAL CONNECTIVITY REQUIREMENT

A key factor of blockchains is the synchronization across the network, i.e. every node should obtain a copy of the ledger. Individual miners may want to reduce their data usage by connecting to fewer other nodes, but for security all miners are encouraged to connect to as many other peers as possible. We here provide an intuitive picture on the minimal direct connections required for efficient information propagation across the network. For simplicity, we assume that on average every miner maintains connections with l_c other nodes, $0 < l_c \leq l_m$. Once Miner A receives a transaction or a block, she sends it to all her directly connected nodes. The same procedure repeats until the message reaches every node in the network. We define information propagation from one network node to another as a hop. Restricting maximum number of hops to k, we obtain:

$$\begin{aligned} &1 + l_c + l_c(l_c - 1) + \cdots + l_c(l_c - 1)^{k-1} \\ &= 1 + l_c \sum_{i=0}^{k-1} (l_c - 1)^i \geq l_m. \end{aligned} \tag{11}$$

We solve Equation 11 under given k and l_m. We do not use l_{on} as eventually all miners will store the blockchain. We obtain γ, fraction of nodes that one should directly connect to as:

$$\gamma \geq \frac{min\,(l_c)}{l_m}. \tag{12}$$

F. SUMMARY

Although the payment operators' intention is to have fewer mining nodes to reduce the equipment cost, there should be sufficient miners to keep the fairness of the system. The payment operators should also consider real-world churn rate and network resource costs to determine the mining reward. Furthermore, all villagers, especially miners, are recommended to increase their connectivity for better synchronization even though this leads to higher bandwidth usage.

V. EVALUATION

In this section, we first validate our transaction processing model by comparing it to a real deployment of a private Ethereum blockchain and make predictions about the system behavior based on the validated model. We also dimension the local blockchain network design to satisfy the minimum average connection required for each mining node. Finally, we evaluate the blockchain performance under different network conditions on the experimental deployment.

A. EXPERIMENTAL ENVIRONMENT

We deployed a private Ethereum blockchain on multiple virtual machines running Ubuntu Linux v16.04 in an Openstack environment.[1] Each virtual machine is given 1 virtual CPU core, 2 GB of memory, and 10 GB of persistent storage to meet the minimum hardware requirement for running Ethereum. Elastic Search[2] was used to store block-related information. The network behavior was monitored using the Python Web3 Library.[3]

All virtual machines are linked together in a low-latency local network that can be customized on demand. Nodes are connected via a single Gigabit Ethernet switch and form a star topology similar to the cellular network shown in FIGURE 1a. With this setup, a communication round-trip time between any two nodes is less than 1 ms on average.

[1]https://www.openstack.org
[2]https://www.elastic.co
[3]https://web3py.readthedocs.io/en/latest/index.html

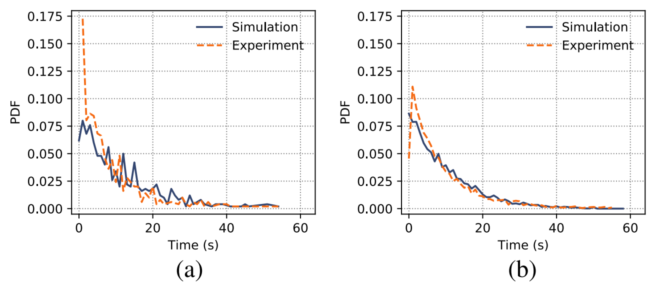


FIGURE 2. Block generation and transaction processing. (a) Block time. (b) Transaction processing time.

We employ Linux traffic control to introduce delays to emulate the high latencies in mobile networks and configure the Linux kernel firewall with Iptables[4] to emulate churns.

We used Geth v1.6.4 [5] for all of our empirical evaluations. Before the actual experimentation, we let the system stabilize for a few hours to obtain appropriate parameters of the genesis block in later runs. Based on our observations, we set the initial *nonce value*, *gas limit* and *difficulty* to 0×42, 0×08000000, 0×400000 respectively to make the system stabilize quickly. To form a fully connected P2P blockchain network, we disabled the auto-discovery feature supported by Geth and configured the overlay network manually. Finally, if not mentioned otherwise, all experiments involved 10 mining nodes and 10 light nodes.

B. MODEL VALIDATION AND NUMERICAL SENSITIVITY ANALYSIS

1) MODEL VALIDATION

To validate our model described in Section IV-B, we emulated a transaction processing scenario by issuing transactions at 1 tps (overall rate) from multiple clients to randomly selected addresses. We sort block timestamps in ascending order and calculate block generation time as the difference between two adjacent timestamps. To obtain transaction processing time, we record the timestamp upon initiation of a transaction and extract the inclusion timestamp from the transaction receipt. FIGURE 2a and FIGURE 2b present the simulation and emulation results of 500 consecutive blocks. Both illustrate a strong agreement between modelling and experimental measurements. Additionally, an inter-comparison between FIGURE 2a and FIGURE 2b confirms that transaction processing and block generation are highly correlated.

2) IMPACT OF TRANSACTION ARRIVAL RATE

We then intended to empirically study the impact of transaction rate on transaction processing and benchmark the local blockchain performance. However, when increasing the arrival rate, we hit the transaction sending limit before the processing limit due to implementation flaws in the Ethereum version we used. Indeed, with more than 3 tps, the transaction generator stopped working due to connection errors as the algorithm assumes that some nodes are flooding the system with too many requests. Despite that the transaction generator worked smoothly with lower arrival rates, in the end only 4246 out of 17265 transactions were mined at 2 tps. We hence decided to analytically study the system behavior under multiple transaction rates.

[4]https://help.ubuntu.com/community/IptablesHowTo
[5]https://github.com/ethereum/go-ethereum

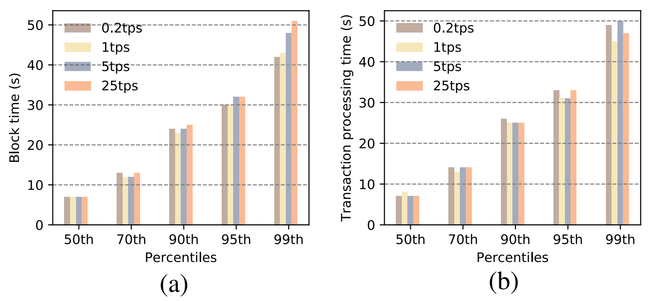


FIGURE 3. Simulation with multiple transaction rates. (a) Block time. (b) Transaction processing time.

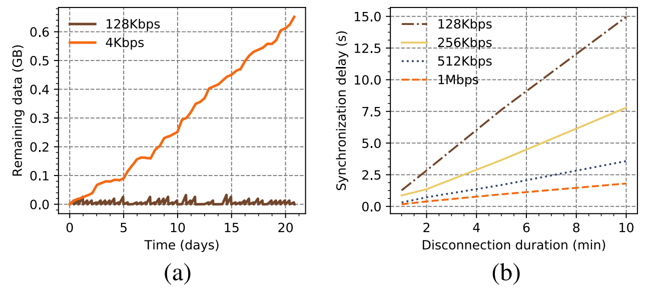


FIGURE 4. Simulated synchronization results. (a) Data remaining. (b) Synchronization delay.

Using the validated model in Part V-B.1, we further generated transactions at various rates including 0.2 tps, 1 tps, 5 tps, and 25 tps in our simulation. FIGURE 3a and FIGURE 3b compare the 50th, 70th, 90th, 95th and 99th percentiles of block time and transaction processing time under all the simulated arrival rates. These results confirm that with a sufficiently large block size, block creation time and transaction processing time are not affected by the arrival rate.

3) PROXY NODE SYNCHRONIZATION

We simulated 50 consecutive services sessions with the disconnection and connection in each service session being $T_U = 9$hrs and $T_C = 1$hr respectively. We estimated the transaction size to be 200 bytes from the real-life data collected by https://www.blockchain.comBlockchain.Info, and an average rate of 2 tps and 1 tps for regular and currency-exchange transactions respectively. Figure 4a illustrates data remaining to be synchronized by the proxy node over these 50 consecutive service sessions. We use bandwidths of 4 Kbps and 128 Kbps to show the two scenarios discussed in Part IV-D. Indeed, under a large synchronization bandwidth, the proxy node can always access all transactions by the end of a service session. When the bandwidth reduces,

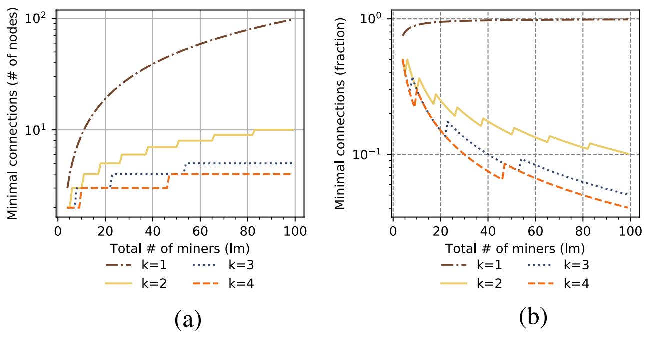


FIGURE 5. Minimal connection requirement. (a) # of connections. (b) Fraction of the network.

new transactions accumulate, making it impossible for the proxy to catch up.

We then analyzed the synchronization when transactions only arrive during disconnection. We did not limit the connection duration and measure the time taken for a full synchronization averaged over 10 service sessions, with T_U ranging from 1 min to 10 min and T_C unlimited. A distribution of the synchronization times is shown in Figure 4b. The result with off-the-shelf devices is presented in Part VI-B.1.

4) MINIMAL CONNECTION REQUIREMENT

Based on the analysis presented in Part IV-E.3, we obtain the minimal connection requirement when varying number of hops k from 1 to 4 in (Condition 11) with a total number of miners l_m increasing from 4 to 100. FIGURE 5 displays the results in the number of nodes and fraction of the network. Minimum l_c that satisfies Condition 11 may remain unchanged for a range of l_m, which results in steps or spikes on the curves. As shown in FIGURE 5a, when $k = 2$, with a total of 100 miners, each individual needs to maintain at least 10 connections. When $k = 3$, the minimum number of connections required reduces to 4. Generally, if only one hop is allowed for information propagation, all miners need to connect with all other miners. While if more hops are allowed, miners can reduce the number of their connections and save the bandwidth usage. This indicates the system's ability to scale without miners exhausting their network resources.

C. EFFECT OF NETWORK DISTURBANCES

To further evaluate the performance of the local blockchain network, we investigated its behavior under disturbances such as *Network Delays* and *Network Churns*. Since it was impossible to obtain the whole picture from transaction processing (cf. Part V-B.2), we have considered block generation time to be an alternative evaluation metric (cf. Part V-B.1).

1) THE (NON)IMPACT OF NETWORK DELAYS

In order to understand the stability of the proposed system when inter-node delay increases, we introduced various delays of 0, 10ms, 50ms, 100ms, 500ms, and 1000ms per

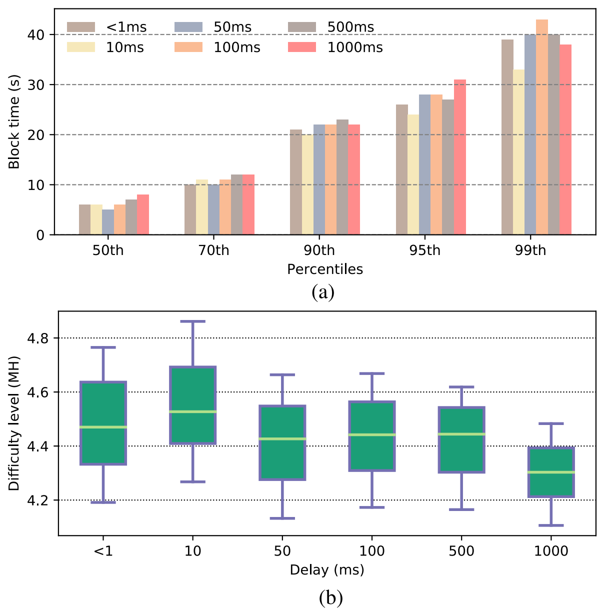


FIGURE 6. System behavior with network delays. (a) Block time percentiles. (b) PoW difficulty levels.

connectivity channel.[6] FIGURE 6a shows the 50th, 70th, 90th, 95th, 99th percentiles of block time, from which it is possible to observe that the block time remains stable even with delays reaching 1 second. We ascribe this to PoW difficulty adjustment where network delay causes time difference between two adjacent blocks to increase, and as a result the internal algorithm reduces difficulty level to achieve a shorter block time. FIGURE 6b illustrates block difficulty under multiple network delays. A decreasing trend can be observed in the difficulty level when delay increases. This behavior helps to maintain the transaction processing speed, but is undesirable because with a fixed total network hash rate, a lower difficulty level leads to more stale blocks and inconsistencies.

2) NETWORK CHURNS

a: TRANSIENT RESPONSE

One of the main concerns of P2P networks is the churn rate of nodes. We emulated a scenario in which some miners go offline at time t_0 and observed variation in block creation time. As shown in FIGURE 7, when some miners go offline, block time experiences a significant increase followed by a "resolving period" during which PoW difficulty is adjusted in response to changes in network hash rate. To find the end point of the "resolving period", we select the timestamp from which the absolute variation is less than 5% for the next 3 consecutive points. Duration of "resolving period" is measured as 361s, 781s, 577s, 527s, 969s respectively. Overall, it increases with the number of offline nodes. While the network recovers, we also observed "backwards mining",

[6]Today's mobile network delays can hardly go beyond 1 second: https://hpbn.co/mobile-networks/

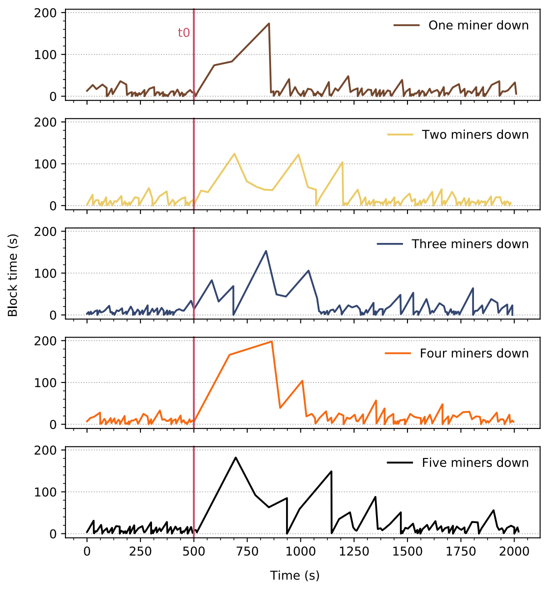


FIGURE 7. Transient response to mining nodes going offline.

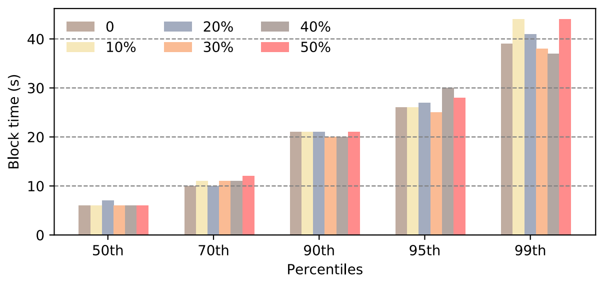


FIGURE 8. System steady-state behavior with network churns.

where newer blocks sometimes have smaller block number than older ones. The disagreement is resolved with the stabilization of the network.

b: CHANGES OVER TIME

We then tested system performance when each miner has a probability of 10%, 20%, 30%, 40% and 50% to go offline. We ran each set of experiment for approximately 2 hours and changed miner status (online or offline) every 20 minutes. FIGURE 8 shows the 50th, 70th, 90th, 95th, 99th percentiles of block generation time, from which we can see block time does not vary much across all churn rates. In other words, number of miners do not affect speed of block generation or transaction processing.

3) SUMMARY

Despite being unable to test the processing limitation due to the shot comings of the Ethereum implementation and obtaining all transactional data experimentally, we managed to study the effect of network disturbances. Difference between empty blocks and blocks with transactions depends on block size, hence the additional transmission delays which are expected to have similar impacts to network delays. Results show that delays and churns may cause block generation time to change especially in the transient state. However, due to the difficulty adjustment they do not necessarily slow down block generation in the long run. Overall, the main observed drawback of the local blockchain is that disturbances may cause more temporary inconsistencies.

TABLE 1. Devices and their capabilities.

	Miner	Full node	Light node
Device	Dell Latitude 6430u	Raspberry Pi 3	Google Pixel XL
# of nodes	3	4	2
OS	Ubuntu 17.04	Raspbian Jessie	Android v8.0.0
RAM	4GB	1GB	4GB
Geth	v1.6.5	v1.6.5	v1.6.7
# of mining threads	4	-	-

VI. IMPLEMENTATION

We demonstrate the feasibility of the system a prototype implementation containing a private Ethereum blockchain and an intermittently connected proxy node was implemented. We tested the synchronization delay of the proxy node when varying the bandwidth and disconnection duration. We also measured the usage of data, CPU, and battery of the mobile wallet app using DDMS [7] and Battery Historian.[8]

A. SETUP

Figure 9 illustrates the setup with one full node, 2 shops, and 3 regular users. Each shop runs a mining node and in addition, Shop 1 owns two full nodes and Shop 2 owns one full node. Regular User 3 operates a miner while the other two regular users are light mobile clients. We summarize the device types, their capabilities and Geth versions in Table 1.

We used a D-Link DSR-250N Wi-Fi router connected to the Oulu public WAN (PanOULU) [30] to emulate the community base station. We used an IEEE 802.11n Wi-Fi network to emulate the local network and interconnect the above. The proxy node's backhaul bandwidth via the router's WAN port. The auto-discovery protocol of Geth was used on all nodes.

We installed an application developed in Python 2.7.12 on the proxy node to update the account information. It is implemented on a Raspberry Pi 3, which also acts as an Ethereum full node. This application synchronizes with the blockchain using the Python-JSON RPC library.[9] It keeps track of the connectivity and update the account balances in SQLite database. We have also developed a smart contract in Solidity v0.4.12 for token creation, conversion and transfer (cf. Section III).

Each payment gateway is composed of one Adafruit PN532 NFC module and a Raspberry Pi 3 attached to a touchscreen, as shown in Figure 10. After entering the amount, the vendor can select from multiple payment options. The application also synchronizes with the blockchain for

[7]https://developer.android.com/studio/profile/ddms.html
[8]https://developer.android.com/studio/profile/battery-historian.html
[9]https://github.com/ConsenSys/ethjsonrpc

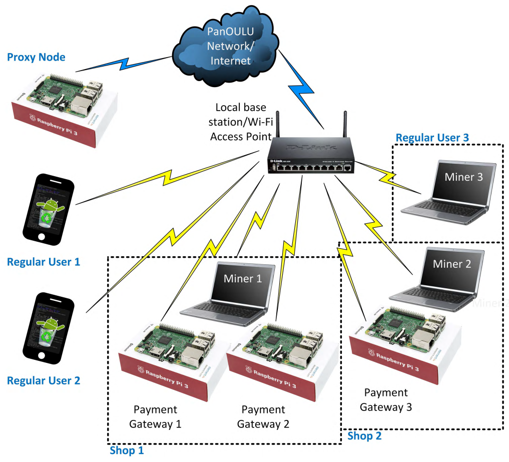


FIGURE 9. Prototype implementation.

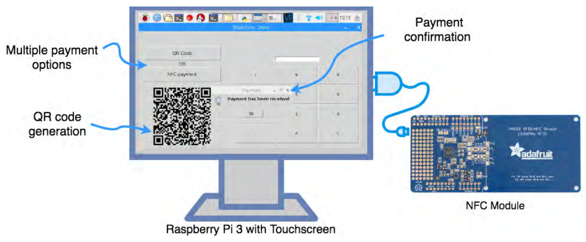


FIGURE 10. Payment gateway.

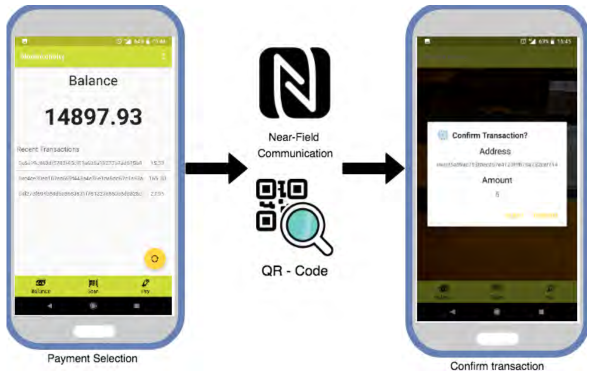


FIGURE 11. Mobile wallet application.

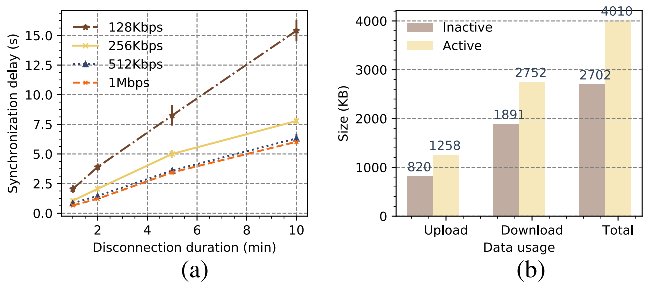


FIGURE 12. Test results on prototype implementation. (a) Synchronization delay. (b) Mobile wallet data usage.

confirmation of token transfer. Once the payment has been confirmed in the user wallet and a block is created, a new transaction is recorded on the blockchain. At the same time the transaction receives one block confirmation and the payment gateway notifies the vendor.

For regular users, we developed the mobile wallet app (cf. Figure 11) in Android Studio 3.0. The mobile client joins the blockchain as a light node, which submits and retrieves transactions directly to/from the blockchain. Both NFC and QR code based payment modes are embedded in the app. After the user selects a payment method, the app asks the user to confirm the transaction information. Once confirmed, the app signs the transaction and submits it to the Ethereum blockchain. The app also shows the account balance, recent transaction history and can store multiple Ethereum accounts.

B. MEASUREMENTS

1) PROXY NODE SYNCHRONIZATION

We emulated a situation where transactions are only committed during disconnection, as described in Part V-B.3. FIGURE 12a illustrates the average synchronization delay and variation over 10 runs of each scenario. Compared to the simulation results in Figure 4, the synchronization time does not increase proportionally with the disconnection duration, especially for bandwidths higher than 512 Kbps. This is because TCP communication speed cannot reach the maximum bandwidth at the beginning of the connection. When disconnection lasts for only a few minutes, data to be downloaded by the proxy node is small and synchronization finishes before TCP communication reaches its maximum bandwidth. As the disconnection duration increases, the amount of data to be downloaded also increases, TCP communication can reach a higher and more stable bandwidth during the synchronization, which results in the decrease in the slope of the plots.

2) MOBILE WALLET PERFORMANCE

We then measured data, CPU and battery usage of the mobile wallet in its idle state (i.e.*Inactive* mode) and when it sends one transaction per minute (i.e. *Active* mode) for a total of one hour. Note that without sending any transactions, the mobile light client continuously synchronizes with the network and downloads block headers. As shown in FIGURE 12b, our wallet app uses only 2.71 MB in total in its idle state, and an additional 21.65 KB to perform one transaction per minute. The overall CPU usage increased from 51.9s to 127.7s within the one-hour period. When it comes to power consumption, the wallet used 3.2 mAh or 0.04% of the mobile phone's total battery usage in its idle state and 19.8 mAh or 0.11% when sending transactions. These results confirm our wallet app requires low bandwidth, CPU processing and power for its operations compared to available cryptocurrency clients [13].

VII. RELATED WORK

We categorize related work into i) Side-Chain and Off-Chain Solutions, ii) Pervasive, Delay-Tolerant Networks, and iii) Blockchain Cost Models.

A. SIDE-CHAIN AND OFF-CHAIN SOLUTIONS

A number of side-chain and off-chain solutions have been developed to solve the on-chain scalability problem. The main idea is to shift part of the transactions away from the main chain. As side-chain and off-chain solutions can work independently from the main chain, they have the potential to be applied to intermittently connected environments. However, although there are implementations like Ethereum Plasma [31] and Bitcoin Lightning Network [32], the available platforms are still at their early stage and are yet to be deployed widely. For example, side-chains like Plasma increase complexity when it comes to data availability and block withholding attacks. While off-chain solutions like Bitcoin Lightning lacks flexibility during channel opening and closing. In addition, as an off-chain channel only involves two parties, in comparison to a public blockchain, complete trust is hard to achieve between only the two.

B. PERVASIVE, DELAY-TOLERANT NETWORKS IN REMOTE REGIONS

Delay-tolerant networks can be a potential solution to deliver Internet services to developing regions. Blattman *et al.* [33] found delay-tolerant networks are sufficient for digital services to meet the needs of most rural communities. Pentland *et al.* [5] later developed DakNet that uses a pervasive mobile coverage to enable asynchronous digital services in India and northern Cambodia. DakNet is remarkably low-cost and well-received by local users as it is more accessible than a centralized, community telephone. The success of DakNet proves that decentralization is an effective way to deliver service to remote regions.

Taking use of a CBS, together with an intermittent connection with the outside to form a delay-tolerant network, we are able to deploy a blockchain system to process inter-village transactions and keep track of the transactions from outside. However, delays can greatly affect the security of blockchains whose operation relies on continuous network connectivity when forks happen. To avoid forming disagreements during the disconnection, we only operate a full node as the proxy.

C. BLOCKCHAIN COST MODELS

Croman *et al.* [27] did a reality check of Bitcoin and analyzed the cost to confirm transactions. Rimba *et al.* [34] later proposed cost models using parameters such as *gas* and *gasPrice* to compare Ethereum with cloud services for business process execution. We stick to findings in [27] to obtain deployment and operation costs as the gas-related parameters can be adjusted by the bank.

VIII. DISCUSSION

Although the ultimate control belongs to the payment operator, the reliable operation of the system is achieved via decentralization. Compared to traditional, centralized solutions, our approach is robust against node churn and can effectively avoid single-point-of-failure with lower deployment cost. However, some aspects of the system design can be further improved.

A. BLOCKCHAIN INEFFICIENCY

PoW is a heavy consensus algorithm, and its inefficiency becomes more significant in public cryptocurrencies. It was not problem in our prototype implementation. As blockchain technology evolves, lightweight consensus protocols are being explored [35]–[38]. If it was to become a problem, these new protocols can be easily incorporated with the proposed system.

B. SECURITY

In this paper, we have demonstrated the case where there are no intended, malicious behaviors. Below we present a brief discussion on security based on the current literature.

1) TEMPORARY INCONSISTENCY (NOT A REAL THREAT)

Even though *blockchain forks* or *stale blocks* are an important indicator of inconsistency [26], if players behaves honestly, all conflicts can be resolved eventually. Hence, stale blocks themselves are not a direct threat to network security. However, they increase the chance of *double-spend* attacks that are usually caused by disagreements within the network [39].

Countermeasure: As suggested by Karame *et al.* [40], *double-spend* attacks can be mitigated by applying a "listening period" of a few seconds on the recipient side. In the proposed system, this can be integrated into the mobile wallet design.

2) NETWORK PARTITIONING (A REAL THREAT)

An attacker who knows about other nodes and their connections can perform routing attacks to partition the network, such as the eclipse attack performed by [16], and the balance attack [17].

Countermeasure: Two potential countermeasures can be deployed, one centered around the proxy and another around participating node themselves.

In the first case, we can enable the proxy to deploy a smart contract that tracks node connections and detects network partitioning. When the proxy is connected, it accesses results generated by this smart contract. If subgraphs are detected, the proxy can take action, e.g. to force nodes in the subgraph to restart and refresh their connections as suggested by Apostolaki *et al.* [15]. Furthermore, full nodes and light nodes can also help to increase connectivity, and mitigate network attacks. In the second one, participating nodes themselves keep track of their own connections and changes in the network topology. As for a network partitioning attack to succeed, significant changes in the observable topology are needed [17], if a node observes that it lost a large number of its peers within a small timeframe, transactions can be frozen until the overlay network stabilizes again.

C. THE DELAYED ACTIONS

Limited by the intermittent connectivity, in the current system, a user can only exchange tokens with the proxy when a connection is available. In addition, although network data is collected and analyzed in real time by miners, proxy control does not happen immediately.

IX. CONCLUSION

We proposed a blockchain-based payment scheme for intermittently connected regions. As a result, we have achieved this through a novel use of smart contracts and a token-based admission control, account management, and mining rewards distribution. Through mathematical models we analyzed system dynamics and the costs for the setup and operation. We then validated the proposed system through prototype implementation on a private Ethereum testbed and demonstrated the practicality of system design using off-the-shelf laptops and mobile devices. Finally, through a comprehensive study of the local blockchain operation, we showed the system stability under network disturbances and demonstrated the whole system operates well in resource-constrained, dynamic, intermittently connected environments.

In the future, we will address security vulnerabilities further and look into solutions that enable seamless operation under network irregularities. In addition, we will explore the potential of using the proposed management framework in other domains.

APPENDIX
ETHEREUM BASICS

We base our system design on the Ethereum platform [24], as it has a short block-generation time that makes it easy to generate large volume of block data for analysis. A summary of key terminology that is required to understand the Ethereum mechanisms is presented below.

A. NODES AND THEIR CONNECTIVITY

A blockchain consists of a P2P communication overlay network. In the case of Ethereum, each network node continuously attempts to connect to other nodes until they have peers. By default, Ethereum nodes use a gossip protocol to find out about other nodes. Each node maintains connections to a few peers either discovered during startup through the P2P protocol, or manually added using their public IP addresses.

Once the overlay network is created, nodes on a blockchain may act as full nodes, miners or light nodes. All nodes contribute to the network connectivity and information (i.e., transactions and blocks) propagation. Miners and full nodes verify all transactions based on their signatures and update any state changes, and keeps a complete transaction record. In addition, miners settle transactions via a process called mining as explained in Part IX-C. Light nodes operate in the Simplified Payment Verification (SPV) mode and only download block headers and verify and store transactions that are related to them. Table 2 summarizes node types and their capabilities in a typical blockchain network.

B. TRANSACTION, BLOCK AND SMART CONTRACT

The public Ethereum blockchain works as a global state machine, where the state is represented by peer interactions or transactions. For simplicity and efficiency, transactions are grouped together to be settled and immutably recorded in blocks. A user needs an Externally Controlled Account (EOA) to send and receive transactions. An EOA is linked to an ether balance and is controlled by the user's private key.

TABLE 2. Summary of node capabilities.

Node Type	Mining	Verification	Storage
Miner	Yes	Yes	Yes
Full Node	No	Yes	Yes
Light Node	No	Partial	Partial

Ethereum also uses smart contracts to automatically form agreements among different entities. Smart contracts are identified by contract accounts, which are similar to EOAs but are associated with "code" that can be triggered by transactions or messages (calls) received from other EOAs or contracts. Smart contracts also have storage capabilities. The states recorded in a smart contract are updated upon a valid transaction and the messages it carries.

Transactions can be sent between two EOAs, two contract accounts, or an EOA and a contract account. In the public Ethereum network, miners may also collect transaction fees to make a profit [24]. Transaction priority, i.e. how likely and how soon a transaction is to be picked up by miners, depends mostly on its value, age and the transaction fee attached. State of a blockchain is computed after every block [41], and the code in smart contracts is executed by all nodes when synchronization happens (cf. Part IX-C).

C. THE BLOCKCHAIN

Nakamoto in his original paper proposed a PoW scheme to confirm transactions and select representation in majority decision making [42]. PoW is also adopted by the current version of Ethereum as described below.

1) POW MINING AND DIFFICULTY CONTROL

PoW mining is essentially the process of repeatedly calculating a hash value with an incrementing nonce until the hash is smaller than a target. Hashrate is the speed at which a miner computes hashes. Miners compete against each other in solving PoW puzzles and broadcast blocks to the rest of the network upon block creation for validation and synchronization. In addition to mining, miners also listen for new transactions and blocks discovered by others to prepare the next block.

Difficulty is a measure of how difficult it is to solve a PoW puzzle. Difficulty adjustment over PoW puzzles leads to a stable block creation rate. In Ethereum the calculation is based on the block number, timestamp of the current block, and timestamp & difficulty of its parent block. For Ethereum Homestead,[10] the adjustment is always a multiple of `parent_diff // 2048`, with parameter $a \in [-99, 1]$. We denote `block_timestamp - parent_timestamp` as δt, and summarize the adjust-

[10]http://ethdocs.org/en/latest/introduction/the-homestead-release.html

TABLE 3. PoW difficulty adjustments.

Timestamp difference	Adjustment
$\delta t < 10s$	$a = 1$
$10s <= \delta t < 20s$	Difficulty unchanged, $a = 0$
$20s <= \delta t < 1000s$	Adjust downwards proportional to δt, $a \in [-99, -1]$
$\delta t >= 1000s$	$a = -99$

ments in Table 3. Difficulty adjustment is insignificant under a high network hashrate.

2) CONSENSUS AND CHAIN GROWTH

Transmission delays may lead to stale blocks, or forks [26] as multiple blocks could be created at the same time, and received by different nodes across the network. A block is attached to the chain once created, with its own block header pointing to the previous block header, and all the way back to the genesis block. All peers work out these inconsistencies by selecting the longest chain. Ethereum determines the longest chain based on the total difficulty of all the blocks.[11]

3) NODE SYNCHRONIZATION

Network delays and node churns are common in blockchain systems. If one node restarts after going off-chain, it enquires its peers to obtain the latest version of the ledger. States stored in a particular block can only be accessed if the node synchronization has reached it.

D. COIN SUPPLY

To compensate the resources consumed in mining, a fixed amount of new tokens, or mining reward, is generated upon the creation of blocks and paid to the winners. This rewarding scheme is necessary as it makes the blockchain more secure [44], but the continuous creation of coins may result in inflation.

REFERENCES

[1] (2018). *Global Internet Usage*. Accessed: Apr. 11, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Global_Internet_usage

[2] (2018). *Number of Fixed Broadband Internet Subscriptions Worldwide From 2005 to 2017 (in Millions)*. Accessed: Apr. 11, 2018. [Online]. Available: https://www.statista.com/statistics/268673/number-of-broadband-internet-subscriptions/

[3] (2018). *4 Billion People Still don't Have Internet Access*. Accessed: Apr. 11, 2018. [Online]. Available: https://www.weforum.org/agenda/2016/05/4-billion-people-still-don-t-have-internet-access-here-s-how-to-connect-them/

[4] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proc. IEEE Int. Workshop Sensor Netw. Protocols Appl.*, May 2003, pp. 30–41.

[5] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.

[6] (2018). *KUHA Mobile Network*. Accessed: Oct. 31, 2018. [Online]. Available: https://www.kuha.io

[7] (2018). *Telstra Media Release*. Accessed: Mar. 28, 2018. [Online]. Available: https://www.telstra.com.au/aboutus/media/media-releases/Telstra-trials-small-cell-technology-on-TasNetworks-infrastructure-to-fill-mobile-black-spots

[11] Ethereum does not implement the full Greedy Heaviest-Observed Sub-Tree (GHOST) protocol proposed by Sompolinsky and Zohar [43], instead, it uses a longest-chain rule with rewards for stale blocks [39].

[8] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Baltimore, MD, USA: Penguin, 2016.

[9] D. Fitchett, *Bank for Agriculture and Agricultural Cooperatives (BAAC), Thailand (Case Study)*. Washington, DC, USA: Consultative Group to Assist the Poorest (CGAP) Working Group on Savings Mobilization, 1999

[10] I. Mas and D. Radcliffe. (2010). *Mobile Payments go Viral: M-Pesa in Kenya*. [Online]. Available: http://siteresources.worldbank.org/AFRICAEXT/Resources/258643-1271798012256/M-PESA_Kenya.pdf

[11] T. J. MacDonald, D. W. Allen, and J. Potts, "Blockchains and the boundaries of self-organized economies: Predictions for the future of banking," in *Banking Beyond Banks and Money*. Cham, Switzerland: Springer, 2016, pp. 279–296.

[12] A. Mackenzie, "The fintech revolution," *London Bus. School Rev.*, vol. 26, no. 3, pp. 50–53, 2015.

[13] (2017). *Requirements and Warnings—Bitcoin Core*. Accessed: Sep. 21, 2017. [Online]. Available: https://bitcoin.org/en/bitcoin-core/features/requirements

[14] (2017). *Ethereum's Blockchain Size Surpasses Bitcoin's by 40 Percent*. Accessed: Mar. 19, 2018. [Online]. Available: http://www.altcointoday.com/ethereums-blockchain-size-surpasses-bitcoins-by-40/

[15] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 375–392.

[16] K. Wüst and A. Gervais, "Ethereum eclipse attacks," ETH Zürich, Zürich, Switzerland, 2016.

[17] C. Natoli and V. Gramoli. (2016). "The balance attack against proof-of-work blockchains: The R3 testbed as an example." [Online]. Available: https://arxiv.org/abs/1612.09426

[18] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," IACR Cryptol. ePrint Arch., 2018, p. 236.

[19] (2018). *Nokia Kuha: Community-Run Small Cells*. Accessed: Mar. 28, 2018. [Online]. Available: http://smallcells.3g4g.co.uk/2017/06/nokia-kuha-community-run-small-cells.html

[20] (2018). *Where Cellular Networks Don't Exist, People Are Building Their Own*. Accessed: Mar. 28, 2018. [Online]. Available: https://www.wired.com/2015/01/diy-cellular-phone-networks-mexico/

[21] R. L. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes," in *Proc. Int. Workshop Secur. Protocols*. Berlin, Germany: Springer, Apr. 1996, pp. 69–87.

[22] R. Hauser, M. Steiner, and M. Waidner, "Micro-payments based on iKP," IBM TJ Watson Research Center, IBM, New York, NY, USA, 1996.

[23] K. Wüst and A. Gervais, "Do you need a Blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54.

[24] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[25] S. Kasahara and J. Kawahara. (2016). "Priority mechanism of bitcoin and its effect on transaction-confirmation process." [Online]. Available: https://arxiv.org/abs/1604.00103

[26] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE P2P*, Sep. 2013, pp. 1–10.

[27] K. Croman *et al.*, "On scaling decentralized blockchains," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2016, pp. 106–125.

[28] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proc. USENIX Secur. Symp.*, 2015, pp. 129–144.

[29] P. R. Rizun, "A transaction fee market exists without a block size limit," in *Proc. Block Size Limit Debate Working Paper*, 2015, pp. 1–16.

[30] T. Ojala, J. Orajárvi, K. Puhakka, I. Heikkinen, and J. Heikka, "panOULU: Triple helix driven municipal wireless network providing open and free Internet access," in *Proc. 5th Int. Conf. Communities Technol.*, 2011, pp. 118–127.

[31] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," White Paper, 2017, pp. 1–47.

[32] J. Poon and T. Dryja. (2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[33] C. Blattman, R. Jensen, and R. Roman, "Assessing the need and potential of community networking for developing countries: A case study from India," Harvard Center Int. Develop., Cambridge, MA, USA, 2002. [Online]. Available: http://edevelopment.media.mit.edu/SARI/papers/CommunityNetworking.pdf

[34] P. Rimba, A. B. Tran, I. Weber, M. Staples, A. Ponomarev, and X. Xu, "Comparing blockchain and cloud services for business process execution," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 257–260.

[35] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," Self-Published Paper, Aug. 2012.

[36] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. NSDI*, 2016, pp. 45–59.

[37] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui. (2017). "LocalCoin: An ad-hoc payment scheme for areas with high connectivity." [Online]. Available: https://arxiv.org/abs/1708.08086

[38] T. Crain, V. Gramoli, M. Larrea, and M. Raynal. (2017). "(Leader/randomization/signature)-free byzantine consensus for consortium blockchains." [Online]. Available: https://arxiv.org/abs/1702.03068

[39] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 3–16.

[40] G. Karame, E. Androulaki, and S. Capkun, "Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin," IACR Cryptol. ePrint Arch., Tech. Rep. 2012/248, 2012.

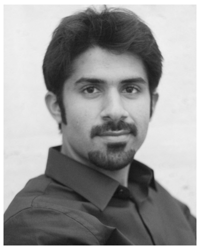
[41] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2015, pp. 104–121.

[42] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[43] Y. Sompolinsky and A. Zohar, "Accelerating bitcoin's transaction processing. Fast money grows on trees, not chains," IACR Cryptol. ePrint Arch., Tech. Rep. 881, 2013.

[44] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 154–167.

YINING HU received the B.Eng. degree (Hons.) in electrical engineering from The University of Sydney, Australia, in 2016. She is currently pursuing the Ph.D. degree with the University of New South Wales, affiliated to Data61-CSIRO, Australia. Her current research interests include enabling the seamless use of blockchain in environments with intermittent connectivity, and Bitcoin transaction analysis.



AHSAN MANZOOR received the B.Sc. degree in computer software engineering from the Ghulam Ishaq Khan Institute, Pakistan, in 2014, and the M.Sc. degree from the University of Oulu, Finland, in 2017, where he is currently pursuing the Ph.D. degree. He started working as a Research Assistant with the Centre for Wireless Communications, University of Oulu. He is also a Blockchain Research Developer with Rovio Entertainment, Espoo, Finland. His research interests include Blockchain, the Internet of Things, and ubiquitous computing.

PARINYA EKPARINYA received the B.Eng. degree in computer engineering from the King Mongkut's Institute of Technology Ladkrabang, in 2009, and the Master of Information Technology degree from The University of Sydney, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include software-defined networking and network function virtualization. He is currently a recipient of the Royal Thai Government Scholarship to pursue the Ph.D. degree with the School of Information Technology, The University of Sydney.

MADHUSANKA LIYANAGE (M'16) received the Ph.D. degree in communication engineering from the University of Oulu, Oulu, Finland.

From 2011 to 2012, he was a Research Scientist with the I3S Laboratory and Inria, Sophia Antipolis, France. Also, he was a Visiting Research Fellow with the University of Oxford, Data61-CSIRO, Infolabs21, Lancaster University, and the University of New South Wales, from 2016 to 2018. He is currently an Adjunct Professor with the University of Oulu. His research interests include SDN, the IoT, blockchain, and mobile and virtual network security. He is also a co-author of over 50 publications, including two edited books with Wiley. He is a member of the ICT. He is currently a Marie Curie Fellow of the School of Computer Science, University College Dublin, Ireland. He is also a Management Committee Member of the EU COST Action IC1301, IC1303, CA15107, CA15127, CA16116, and CA 16226 projects. He has served as a Technical Program Committee Member of the EAI M3Apps 2016, 5GU 2017, EUCNC 2017, EUCNC 2018, MASS 2018, 5G-WF 2018, MCWN 2018, IEEE WCNC 2019, and 5G-Wf 2019 conferences, and the Technical Program Co-Chair in SecureEdge Workshop at the IEEE CIT 2017, the MEC-IoT Workshop at the 5GWF 2018, and Blockchain in IoT workshop at Globecom 2018 conferences. He has also served as the Session Chair in a number of other conferences, including the IEEE WCNC, the EAI CROWNCOM, the EAI 5GU, the IEEE CIT, the IEEE PIMRC, the EAI BODYNET, and the IEEE 5GWF. He is also the Demo Chair of the IEEE WCNC 2019.

KANCHANA THILAKARATHNA received the Ph.D. degree in electrical engineering and telecommunications from the University of New South Wales, Australia. He is currently a Lecturer in distributed computing, and a member of the Centre for Distributed and High-Performance Computing, School of Information Technologies, The University of Sydney. His research interests include network and device function virtualization, the Internet of Things, cybersecurity and privacy, mobile networks, and augmented/mixed reality.

GUILLAUME JOURJON is currently a Senior Scientist with Data61-CSIRO, Sydney, Australia. Prior to his position at Data61, he was a Senior Researcher with NICTA, Australia. His research interests revolve around network measurements, fraud detection, and new network architectures. His current projects include large scale measurement architecture and performance evaluation; network economics; software-defined networking; multithreaded high-speed packet processing for network-bound applications; and cyber-security.

ARUNA SENEVIRATNE is currently a Foundation Professor of telecommunications with the University of New South Wales, Australia, where he holds the Mahanakorn Chair of telecommunications. He has also worked at a number of other Universities in Australia, UK and France, and industrial organizations, including Muirhead, Standard Telecommunication Labs, Avaya Labs, and Telecom Australia (Telstra). In addition, he has held visiting appointments at INRIA, France, and has been awarded a number of fellowships, including one at the British Telecom and one at the Telecom Australia Research Labs. His current research interests are in physical analytics: technologies that enable applications to interact intelligently and securely with their environment in real time. Most recently, his team has been working on using these technologies in behavioral biometrics, optimizing the performance of wearables, and the IoT system verification.