Performance Analysis of Blockchain Systems with Wireless Mobile Miners

Gilsoo Lee, Jihong Park, Walid Saad, and Mehdi Bennis

***Abstract*—In this paper, a novel framework that uses wireless mobile miners (MMs) for computation purposes in a blockchain system is proposed. To operate blockchains over such a wireless mobile network with minimum forking events, it is imperative to maintain low-latency wireless communications between MMs and communication nodes (CNs) that store the blockchain ledgers. To analyze the sensitivity of the system to latency, the probability of occurrence of a forking event is theoretically derived. Also, in mobile blockchain using MMs, minimizing energy consumption required for networking and computation is essential to extend the operation time of MMs. Hence, the average energy consumption of an MM is derived as a function of the system parameters such as the number of MMs and power consumed by the computing, transmission, and mobility processes of the MMs. Simulation results verify the analytical derivations and show that using a larger number of MMs can reduce the energy consumption by up to 95% compared to a blockchain system with a single MM.**

I. Introduction

Blockchains will be an integral part of the emerging Internet of Things (IoT) system [1]. Blockchain applications can securely store data without a central trusted authority by leveraging distributed consensus mechanisms. Recently, blockchains have been adopted in a number of wireless and IoT domains [2]–[6]. For instance, the authors in [2] investigate the performance of a blockchain system operating in a vehicular network in which each vehicle mining node only has a limited time period to exchange blockchain information. The authors in [3] develop a mobile blockchain application that executes a mining process on a mobile device platform. The work in [4] proposes a firmware update scheme for the autonomous vehicles by using a blockchain to ensure the authenticity and integrity of software updates. Also, the authors in [5] develop a false-report attack detection scheme in a vehicular network by exploiting the moving vehicles to compute a consensus mechanism of a blockchain for the sake of data authentication. Moreover, the work in [6] introduces edge computing used to maintain a blockchain for operation of mobile applications.

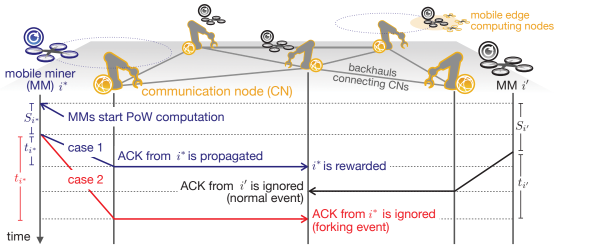

Fig. 1: System model and timing diagram of normal and forking events.

This prior art on mobile blockchains [2]–[6] generally assumes that the miners store the ledger while updating their ledger by communicating with each other. However, a device cannot perform networking functions such as updating their ledger if the miners are unable to maintain a stable network connectivity due to the randomness of the wireless channel and the dynamic networking environment (e.g., as experienced in a vehicular network). Also, a device that does not have sufficient computing capabilities cannot perform mining functions. Therefore, it is more effective to consider a novel blockchain architecture in which mining and networking functions are migrated to two different types of nodes, respectively. Moreover, the prior works in [2]–[6] do not account for the impact of the transmission latency at the wireless link of mobile devices on the blockchain performance. Furthermore, the existing literature mostly relies on isolated experimental results focused on simplistic use cases. In contrast, a rigorous and generalized performance analysis of mobile blockchains in a wireless environment is needed to show how the parameters of a wireless blockchain system can affect its performance metrics such as forking (i.e., a situation when the node who is eligible to verify a block fails to be the first to transmit a verification message to other nodes) and device energy consumption. Consequently, unlike the existing literature [2]–[6] which investigates the use cases of mobile blockchains that store the ledger at the miners, our goal is to design and analyze a novel blockchain system using wireless mobile miners (MMs) such as drones, vehicles, or computationally capable mobile nodes to process the mining computation while the ledgers are stored at the communication nodes (CNs) connected to MMs. For instance, our architecture can be used to operate a blockchain over an existing wireless communication system that includes a network consisting of low computing capability devices. In our architecture, the existing network nodes are reused as CNs while state-of-the-art MMs are newly deployed to process computational tasks. Therefore, the proposed architecture can facilitate different applications in manufacturing systems of smart factories [4], [7], software security of intelligent transportation systems [2], [5], and automation systems in smart building and cities [8].

The main contribution of this paper is a novel, mobile blockchain system architecture and the performance analysis of the proposed system. In our architecture, each MM is connected to a CN via a wireless link, and the computing result of an MM is transmitted to other MMs through the backhaul network that interconnects the CNs. In such an architecture, forking events can occur when an MM propagates its computing result to other MMs, since the transmission latency between an MM and its associated CN can be large

This work was supported in part by NSF Grant CNS-1814477, in part by Nokia Foundation under a personal grant, in part by the Academy of Finland under Grant 294128, in part by the 6Genesis Flagship under Grant 318927, in part by the Academy of Finland through the MISSION Project under Grant 319759, and in part by the Artificial Intelligence for Mobile Wireless Systems (AIMS) project at the University of Oulu.

G. Lee and W. Saad are with Wireless@VT, Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA USA (e-mail: {gilsoolee, walids}@vt.edu).

J. Park is with School of Information Technology, Deakin University, Geelong, Australia (e-mail: jihong.park@deakin.edu.au).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, Oulu, Finland (e-mail: mehdi.bennis@oulu.fi).

due to the wireless and mobile nature of the system. To this end, we derive a tractable expression for the probability of occurrence of a forking event, as a function of the wireless network parameters such as the number of MMs and the MM power consumed by the MMs for computing, transmission, and mobility. We use the derived metric to find the average energy consumption required for an MM by calculating the number of trials in processing a blockchain consensus algorithm. Our analytical result shows that the delay required for movement and the possibly high latency resulting from a wireless link can incur a forking event. Simulation results corroborate the analytical derivations and show that the energy consumption for blockchain computation can be reduced by using a lower transmission power and decreasing movement of each MM.

II. Mobile Blockchain Architecture

Consider a blockchain network that consists of a set $\mathcal{I}$ of I MMs and CNs as shown in Fig. 1. CNs essentially represent fixed wireless network infrastructure such as base stations associated with MMs. MMs can be computationally capable mobile devices such as industrial drones or ground vehicles gathering transaction data from other ground devices (e.g., see [2] for additional motivation on the use of such mobile mining nodes). Mining computation is executed by each MM that can be an independent computing node [3]. Also, to process computation, a mobile edge computing can be adopted [9], [10]. By deploying a mobile edge computing system locally as shown in Fig. 1, the computation can be managed by the head node of this local mobile edge computing cluster. We mainly focus on drone-type MMs due to their ability to flexibly travel and move in nearly unconstrained locations [11]. However, our model can accommodate any other type of MMs. In Fig. 1, MMs in remote locations are unable to directly communicate each other. Therefore, each MM or the cluster head is associated with a different CN that is connected to a backhaul network.

In the considered system, the ledger is located at the CNs while the MMs are used for computing. MMs process computational tasks required to run a consensus mechanism. For instance, a consensus can be guaranteed by using a proof of work (PoW). When transaction records are stored as blocks at the CNs, those blocks must be validated by PoW schemes so as to guarantee that the transaction in the block is original. Then, the CNs can delegate the PoW computation needed for this validation to the wireless MMs. Once each MM completes its PoW computation, an acknowledgment (ACK) message is sent from the MM to the associated CN, called the *source CN*. The source CN propagates the reception of the ACK message to other CNs though the backhaul links among the CNs. We assume high-bandwidth, fiber backhaul links between CNs and, hence, the message propagation latency in backhaul network will be negligible. By using the considered architecture, mobile blockchain can be applied to different applications as follows.

A. System Model: Computation, Mobility, and Transmission

In a blockchain system, the reception order of the ACK messages from multiple MMs to the backhaul network should be identical to the order of completion of the PoW computation, as shown in Case 1 of Fig. 1. If the ACK message that was sent earliest arrives to the CNs interconnected by the backhaul network later than other ACK messages, it will lead to a so-called *forking event*, as shown in Case 2 of Fig. 1. The MM that completes the PoW with the shortest delay will receive a unit of reward. However, when a forking event occurs, the MMs can no longer discern which MM completed the current PoW computation with the shortest delay. Thus, *the probability of occurrence of a forking event* is an important metric in a blockchain system that we will derive in Section III. This metric will also allow us to analyze the average number of PoW computations and average MM energy consumption required to earn a reward by processing blockchain computation at an MM.

In Fig. 1, the computing latency s_i and transmission latency t_i of an MM $i \in \mathcal{I}$ are the realization of random variables S_i and T_i, respectively. We assume that S_i and T_i, for all MMs $\forall i \in \mathcal{I}$, will follow identical probability distributions. This is reasonable for the case in which all independent MMs are set to use the same computing resource and wireless parameters such as a transmit power. For notational simplicity, we use S and T to denote S_i and T_i, respectively. Therefore, we will derive the probability distributions of S and T.

According to the PoW, all MMs start their PoW computation at the same time and keep executing the PoW computation until one of the MMs completes the computational task by finding the desired hash value [3]. When an MM executes the computational task for the PoW of the current block, the time period needed to finish this PoW computation will be an exponential random variable S whose distribution is $f_S(s) = \lambda_c e^{-\lambda_c s}$ where $\lambda_\text{c} = \lambda_0 P_\text{c}$ refers to the computing speed of an MM with P_c being the power consumption for computation of an MM and λ_0 being a constant scaling factor.

Once an MM finishes its PoW computation for the current block, the ACK message must be delivered to the associated CN, and the ACK is sent to other CNs so that all other MMs can stop their current PoW computation. Therefore, we derive the transmission latency when an MM transmits the ACK message to the associated CN through a wireless link. Under a Rayleigh fading channel, the small-scaling fading gain between an MM and the CN is a random variable H with distribution $f_H(h) = \exp(-h)$ where the statistical average gain of the Rayleigh fading is unity. We assume that MMs move along a circular trajectory around the associated CN and, thus, they have a constant path loss g. Then, the signal-to-noise-ratio (SNR) of any MM at its associated CN is the realization for the random variable given by:

$$\Gamma_0 = gHP_\text{tx}/\sigma_n^2, \tag{1}$$

where P_tx is the transmit power of an MM, and σ_n^2 is the noise power. Since H is the only random variable in Γ_0, the distribution of the random variable Γ_0 will be $f_{\Gamma_0}(\gamma) = k_0 e^{-k_0\gamma}$ where $k_0 = \sigma_n^2/(gP_\text{tx})$.

An MM transmits the ACK if the channel gain is higher than a threshold γ_0 that can be seen as the minimum SNR required to decode the transmitted data at the receiver. Hence, achieving an SNR higher than γ_0 is necessary to transmit the ACK data from an MM to the CN. In particular, each MM observes the

SNR at any given location, and if the SNR is lower than γ_0, the MM moves to another location to obtain a better SNR. Hence, each MM will dynamically seek a location that yields an SNR higher than γ_0. The number of new location that an MM needs to visit can be given by $n+1, n \in \mathbb{Z}^{\geq 0}$. At a given location, the probability that a certain MM achieves an SNR higher than γ_0 is $p_s = \Pr(\Gamma_0 \geq \gamma_0) = 1 - F_{\Gamma_0}(\gamma_0) = e^{-k_0\gamma_0}$ where $F_{\Gamma_0}(\gamma)$ is the cumulative probability distribution of random variable Γ_0. Hence, the number of movements is n that is a realization of a random variable N modeled by using a geometric distribution with the probability mass function:

$$f_N(n) = (1-p_s)^n p_s. \tag{2}$$

In order to change the small-scale fading gain by moving from one location to another, an MM needs to move by a distance of $\lambda/2$ where λ is the wavelength of the carrier frequency. The time period needed to move by a distance of $\lambda/2$ is given by $t_m = (\lambda/2)/v$ where v is the speed of the MM. The power consumed[1] to move an MM is P_m. Therefore, the movement latency of the MM during n movements becomes $T_m = t_m N$.

After finishing N movements[2], the probability density function of SNR Γ, will be $f_\Gamma(\gamma) = g(\gamma)/(1 - F_\Gamma(\gamma_0))$ where $g(\gamma) = f_{\Gamma_0}(\gamma) = k_0 e^{-k_0\gamma}$, if $\gamma_0 < \gamma$, and $g(\gamma) = 0$, otherwise. Therefore, the probability distribution of Γ is rewritten as

$$f_\Gamma(\gamma) = \begin{cases} k_0 e^{-k_0(\gamma-\gamma_0)}, & \text{if } \gamma_0 < \gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The data rate of the MM is $R = B\log_2(1+\Gamma)$ where B is the bandwidth. The wireless transmission latency of the MM in the uplink will then be $T_u = K/R$ where K is the size of the ACK message. Then, the probability density function of T_u becomes $-f_\Gamma(v(t))\frac{d}{dt}v(t)$ where $v(t) = 2^{\frac{K}{Bt}} - 1$, and, therefore, we have

$$f_{T_u}(t) = \begin{cases} k_0 e^{-k_0\left(2^{\frac{K}{Bt}}-1-\gamma_0\right)} \frac{K\ln 2}{Bt^2} 2^{\frac{K}{Bt}}, & \text{if } 0 < t < \bar{t}, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $\bar{t} = K/(B\log_2(1+\gamma_0))$ is the largest wireless transmission latency in the uplink since the SNR is higher than γ_0. Thus, the total transmission latency including both movement and wireless transmission latencies becomes $T = T_m + T_u$.

By using the random variables S, T_m, and T_u, the energy consumption of an MM in a single round of the PoW computation becomes a random variable given by:

$$E = P_\text{c}S + P_m T_\text{m} + P_\text{tx}T_u. \tag{5}$$

Next, we analyze the performance of the proposed system by deriving the probability of no forking and the average energy consumption of an MM.

III. Average Energy Consumption Analysis

We now analyze the performance of the proposed mobile blockchain system in terms of the occurrence of a forking event and the MM energy consumption. The occurrence of a forking event increases the energy consumption required to complete the current block's PoW due to the increment of the total computing and transmission latency used for the PoW recomputation. Therefore, the probability of having no forking event (called *no-forking probability* hereinafter), that we derive next, is essential to derive the average MM energy consumption until a reward is earned by an MM.

A. No-forking Probability

For the PoW computation of a block, the MM that is the first to finish its PoW is indexed by i^*, i.e., $i^* = \text{argmin}_{i\in\mathcal{I}} s_i$. Therefore, when $s_{i^*} < s_{i'}, \forall i' \in \mathcal{I}\setminus\{i^*\}$, the ACK of MM i^* should arrive to the source CN so that the ACK information is propagated to all CNs via the backhaul network before any ACKs from other MMs arrive, i.e., $s_{i^*} + t_{i^*} < s_{i'} + t_{i'}$ as shown in Case 1 of Fig. 1. However, the order of arrival of ACK messages from multiple MMs to the source CN can be different, i.e., $s_{i^*} + t_{i^*} > s_{i'} + t_{i'}$ as shown in Case 2 of Fig. 1. The change in the order of arrival happens when the transmission latency of MM i' is shorter than that of MM i^*, i.e., $t_{i^*} > t_{i'}$, due to the different mobility patterns and the wireless transmission latency. Therefore, the ACK message of MM i' can arrive at the source CN earlier than the ACK message of MM i^*, thus resulting in a forking event. We next derive the no-forking probability by calculating the probability that the ACK from MM i^* arrives earlier than any ACK from any other MM $i' \in \mathcal{I}\setminus\{i^*\}$.

Theorem 1. *The no-forking probability is given by*

$$p_n = e^{-\lambda_c(I-1)}\int_0^{\bar{t}}\int_0^\infty \left(\int_0^{t_{i^*}} e^{-\lambda_c(t_{i^*}-t)} f_T(t)dt + \int_{t_{i^*}}^{\bar{t}} f_T(t)dt\right)^{I-1} f_S(s_{i^*})ds_{i^*} f_T(t_{i^*})dt_{i^*}. \tag{6}$$

Proof. See Appendix A. □

The no-forking probability p_n is derived as a tractable function of the wireless parameters such as the number of MMs, the MMs' transmission power, and the computing speed. If a forking event occurs, additional energy is needed for processing computation for the next block, and, hence, we analyze the average MM energy consumption next.

B. Average Energy Consumption

Since a forking event incurs an additional round of computation, the number of PoW computations needed to earn a reward follows a geometric distribution with mean of $1/p_n$. Also, MM i^* in each PoW computation will consume the average energy $\mathbb{E}[E] = P_\text{c}\mathbb{E}[S_{i^*}] + P_\text{tx}\mathbb{E}[T_u] + P_\text{m}\mathbb{E}[T_m]$. Our goal is to derive the average energy consumption of MM i^* in each PoW computation round until a block's PoW computation is completed without forking, i.e., $(1/p_n)\mathbb{E}[E]$.

We outline how to derive $\mathbb{E}[E]$ by finding the average latency of the computation, movement, and wireless transmission, i.e., $\mathbb{E}[S_{i^*}]$, $\mathbb{E}[T_u]$, and $\mathbb{E}[T_m]$, respectively. Since the shortest computing latency among all MMs is S_{i^*}, the complementary cumulative probability distribution (CCDF) of S_{i^*} is given by

$$\begin{aligned} \Pr(S_{i^*} > z) &= \Pr\left(\min_{i\in\mathcal{I}}(S_i) > z\right) \\ &= \prod_{i=1}^{I} \Pr(S_i > z) \\ &= (1 - \Pr(S \leq z))^I. \end{aligned}$$

[1] The power consumption needed to move a drone-type MM will be $P_\text{m} = P_H(v) + P_I(v)$ [11]. The closed-form equations of $P_H(v)$ and $P_I(v)$, are given in [11, Equations (57)-(59)].

[2] If the velocity is set to zero in our model, each MM can be seen as a fixed mining node connected to the CN via a wireless link.

Therefore, the average computational latency of MM i^* is derived as

$$\begin{aligned}\mathbb{E}[S_{i^*}] &= \int_0^\infty (1-\Pr(S\le z))^I\, dz \\ &= \int_0^\infty e^{-\lambda_c I z} dz \\ &= 1/(\lambda_c I). \end{aligned} \tag{7}$$

The average latency due to mobility is given by $\mathbb{E}[T_\mathrm{m}] = t_m(e^{(\gamma_0\sigma_n^2)/(gP_\mathrm{tx})}-1)$ since the average of N is $(1-p_s)/p_s = e^{(\gamma_0\sigma_n^2)/(gP_\mathrm{tx})}-1$. Also, the average wireless transmission latency can be calculated by using the CCDF of the probability of T_u given by

$$\Pr(T_u > z) \;=\; \begin{cases} 1-e^{-k_0(2^{\frac{K}{Bz}}-1-\gamma_0)}, & \text{if } 0\le z\le \bar{t}, \\ 0, & \text{otherwise.}\end{cases}$$

The average of the transmission latency is:

$$\mathbb{E}[T_u] = \int_0^{\bar{t}} 1-e^{-k_0(2^{\frac{K}{Bz}}-1-\gamma_0)} dz. \tag{8}$$

Hence, by combining $\mathbb{E}[S_{i^*}]$, $\mathbb{E}[T_u]$, and $\mathbb{E}[T_m]$, we can have a tractable expression of $(1/p_n)\mathbb{E}[E]$. By extending our results, other consensus mechanism can be studied. Particularly, if proof of stake (PoS) is used, the computing latency can be negligible. To model this case, the computing latency s_i can be set to a small constant value, and the rest of the analysis remains the same. By doing so, a new probability distribution of the computing latency can be directly applied to perform the same analysis outlined in Theorem 1.

IV. Simulation Results

For our simulations, we consider that an MM is associated with a CN at a distance of 50 m, and the path loss gain g is calculated by using a free space model due to air-to-air communications. The power spectral density of the noise is -174 dBm/Hz, and the bandwidth is 180 kHz. To model the computational speed of an MM, the power consumption for computation is set to 8 W and the scaling factor λ_0 is 0.04 where computing latency scale is set to be comparable to communication latency.

Figs. 2 and 3 show the no-forking probability and the average energy consumption per MM when the ACK message size is set to $K=1500$ bytes. First of all, Figs. 2 and 3 show that simulation and analysis results are matched. In Fig. 2, a forking event can occur with a high probability as the number of the MMs increases. This is due to the fact that, as more MM join in the PoW computing, the blockchain network is more likely to use an MM having a lower ACK reception period than that of MM i^*. In addition, for a comparison purposes, one considers an ideal system with no wireless latency. In such a case, the gap between the probability of one, i.e., no forking event in the ideal case, and no forking probability in Fig. 2 shows the performance gap between optimal and our practical approach. Fig. 3 shows that the average energy consumption to complete the PoW computation decreases as the number of MMs increases. This is due to the fact that deploying more MMs reduces the average computation time, i.e., $\mathbb{E}[S_{i^*}]$, thus decreasing the energy consumption. For example, using 20 MMs can reduce the energy consumption by up to 95% compared to using 1 MM in a blockchain system.

Fig. 4 shows the energy consumption required to complete the PoW computation when $K=125$ kbytes. The energy consumption of an MM increases with the SNR threshold γ_0 and the transmission power P_tx. This is because the energy consumption for mobility increases with γ_0. Also, the total energy consumption can increase with the transmission power because a high transmission power increases the forking probability, thus increasing the number of repeated PoW computations. Moreover, in Fig. 4, the simulation results show that a time-varying wireless channel gain may lead to SNR outage events with a non-negligible probability, resulting in additional energy consumption.

V. Mobile Blockchain Applications

By using the considered blockchain architecture, the conventional systems having low computational capability can be modified to adopt and operate a blockchain-based application. Particularly, conventional manufacturing systems can be converted into a smart factory using a blockchain to maintain the system up-to-date [4] and run machine learning-based factory quality control software [7]. In a conventional factory, a static legacy control system attached to manufacturing hardware is connected to a network, but it does not have enough computing capability. Therefore, the mobile nodes can be deployed as MMs to assist the computation. Also, by deploying edge computing overlaid on the top of a conventional manufacturing system, MMs can provide sufficient computing power to the actuator's controller in the manufacturing system, while the existing network built in the manufacturing systems can be reused to exchange the messages with other MMs at remote locations. Moreover, in a smart factory or a smart grid [8], industrial IoT devices owned by multiple operators will need to call for consensus for a purpose of system setup or security enhancement. To this end, wireless surveillance drones and warehouse robots can be used as MMs, while CNs are wired-networked static robotic assemblers and machines. Furthermore, our framework is applicable to other scenarios by modifying the mobility and energy consumption models. For example, in a vehicular network [5], MMs are vehicles, and CNs are infrastructure nodes such as road-side units and traffic lights. In the vehicular case, our analysis can be easily extended by adopting a proper mobility model. Finally, in smart building and home automation, MMs are surveillance unmanned robots, laptops, and user's smart devices, and the CNs are installed access points [12]. As shown in these examples, smart devices can be deployed to improve the computing capabilities at the network edge [13]. By exploiting smart devices, a blockchain application can be implemented while reusing the existing infrastructure, and the capital expenditure required to deploy a blockchain system can be reduced. Thus, our framework enables conventional systems to be economically upgraded to use a blockchain by integrating the communication capabilities of CNs and the high computation power of MMs.

VI. Conclusion

In this paper, we have proposed a novel framework using MMs to operate a blockchain over a wireless mobile network. We have derived the no-forking probability and the average MM energy consumption required to earn a reward. Simulation

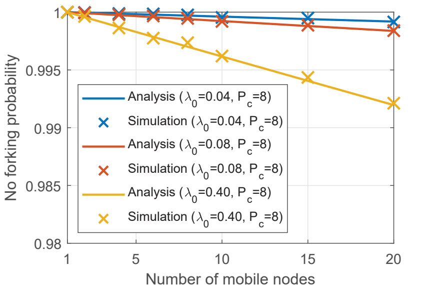

Fig. 2: Probability of a no forking event in one PoW computation.

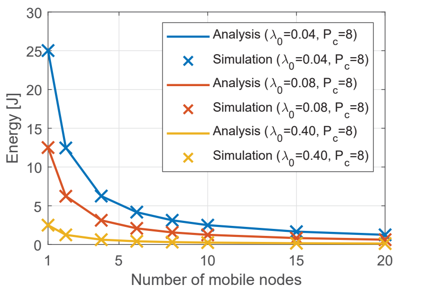

Fig. 3: Average energy consumption of an MM required to compute the PoW of a block.

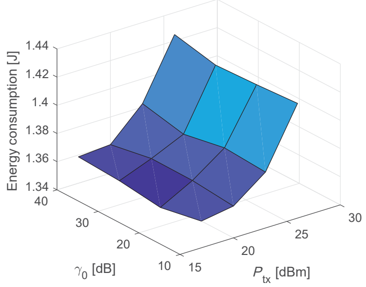

Fig. 4: Average energy consumption of an MM for different values of P_{tx} and γ_0.

results have shown that the wireless transmission power, SNR threshold, and the number of MMs significantly impact the energy consumption of the MMs. Our analytical results serve as a basis for future work that can account for a forking probability and energy consumption of a blockchain system operating in a wireless network. Also, our results can be further used to derive a closed-form expression by adopting asymptotic analysis. Future work can also include real-world experiments that can look at other practical aspects affecting the blockchain performance such as the decoding latency, synchronization, generalized mobility model, trajectory dynamics control, the impact of forking recovery process, and the uplink transmission from the CN to MM. Furthermore, our approach can be extended to study a joint problem of resource allocation and task scheduling among the computing nodes in a local edge computing network.

Appendix A
Proof of Theorem 1

Proof. As a first step, suppose that the shortest computing latency is s_{i^*}, and that MM i^* has a transmission latency t_{i^*}. Given s_{i^*} and t_{i^*}, the probability that all MMs other than MM i^*, do not incur a forking event becomes:

$$\Pr\Bigg(\bigcap_{i' \in \mathcal{I} \setminus \{i^*\}} s_{i^*} + t_{i^*} < S_{i'} + T_{i'} | S_{i^*} = s_{i^*}, T_{i^*} = t_{i^*}, s_{i^*} < S_{i'} \Bigg)$$

$$\overset{(a)}{=} \prod_{i' \in \mathcal{I} \setminus \{i^*\}} \Pr\Bigg(s_{i^*} + t_{i^*} < S_{i'} + T_{i'} | S_{i^*} = s_{i^*}, T_{i^*} = t_{i^*}, s_{i^*} < S_{i'} \Bigg)$$

$$\overset{(b)}{=} \Bigg(\Pr\left(s_{i^*} + t_{i^*} < S + T | S_{i^*} = s_{i^*}, T_{i^*} = t_{i^*}, s_{i^*} < S \right) \Bigg)^{I-1}$$

$$\overset{(c)}{=} \Bigg(\int_0^{\bar{t}} \int_{\max(s_{i^*}, s_{i^*} + t_{i^*} - t)}^{\infty} f_S(s | s_{i^*} < S) ds f_T(t) dt \Bigg)^{I-1}$$

$$= \Bigg(e^{-\lambda_c (I-1)} \int_0^{t_{i^*}} e^{-\lambda_c (t_{i^*} - t)} f_T(t) dt + \int_{t_{i^*}}^{\bar{t}} f_T(t) dt \Bigg)^{I-1} . \quad (9)$$

The equality (a) in (9) holds since the MMs independently process the computation and transmit the ACK message. Also, equality (b) in (9) holds since all MMs have identical distributions for S_i and T_i. From the equality (c) in (9), the computing latency of an MM is non-negative and is greater than s_{i^*}, i.e., $s = \max(0, s_{i^*})$. To avoid a forking event, the value of the computing latency has to be greater than $s_{i^*} + t_{i^*} - t$, i.e., $s = \max(0, s_{i^*} + t_{i^*} - t)$. Therefore, the range of the computing latency is $[\max(s_{i^*}, s_{i^*} + t_{i^*} - t), \infty]$. Moreover, if s_{i^*} and t_{i^*} are given, the PoW computing latency of all MMs other than MM i^* must be greater than s_{i^*}. Therefore, the computing latency of MM i' becomes the conditional probability distribution given by:

$$f_S(s | s_{i^*} < S) = \frac{\lambda_c e^{-\lambda_c s}}{1 - F_S(s_{i^*})} = \frac{\lambda_c e^{-\lambda_c s}}{e^{-\lambda_c s_{i^*}}}. \quad (10)$$

Thus, given s_{i^*} and t_{i^*}, the no-forking probability yields (9).

Next, the values of s_{i^*} and t_{i^*} in (9) follow the probability distributions $f_S(s_{i^*})$ and $f_T(t_{i^*})$, respectively. By integrating (9) multiplied with $f_S(s_{i^*})$ and $f_T(t_{i^*})$ over the intervals of s_{i^*} and t_{i^*}, the no-forking probability is derived as p_n in (6). □

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network, to appear*, 2020.

[2] S. Kim, "Impacts of mobility on performance of blockchain in vanet," *IEEE Access*, vol. 7, May 2019.

[3] K. Suankaewmanee, D. T. Hoang, D. Niyato, S. Sawadsitang, P. Wang, and Z. Han, "Performance analysis and application of mobile blockchain," in *Proc. Int. Conf. on Comput., Networking and Commun. (ICNC)*, Maui, HI, USA, Mar. 2018, pp. 642–646.

[4] M. Baza, M. Nabil, N. Lasla, K. Fidan, M. Mahmoud, and M. Abdallah, "Blockchain-based firmware update scheme tailored for autonomous vehicles," in *Proc. IEEE Wireless Commun. and Net. Conf.*, Marrakech, Morocco, Apr. 2019.

[5] M. Baza, M. Nabil, N. Bewermeier, K. Fidan, M. Mahmoud, and M. Abdallah, "Detecting sybil attacks using proofs of work and location in vanets," *arXiv preprint arXiv:1904.05845*, 2019.

[6] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," vol. 56, no. 8, pp. 33–39, Aug. 2018.

[7] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, Jun. 2019.

[8] M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, and M. Rahman, "Blockchain-based charging coordination mechanism for smart grid energy storage units," *Proc. IEEE International Conference on Blockchain (Blockchain 2019)*, 2019.

[9] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing," *IEEE Trans. on Systems, Man, and Cybernetics: Sys.*, vol. 50, no. 1, pp. 43–57, Jan. 2020.

[10] Y. Jiao, P. Wang, D. Niyato, and Z. Xiong, "Social welfare maximization auction in edge computing resource allocation for mobile blockchain," in *Proc. IEEE Int. Conf. on Commun.*, Kansas City, MO, USA, May 2018.

[11] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov 2017.

[12] G. Lee, W. Saad, and M. Bennis, "Online optimization techniques for effective fog computing under uncertainty," *MMTC Communications-Frontiers*, vol. 12, no. 4, pp. 19–23, Jul. 2017.

[13] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint:1502.01815*, 2015.