Is Wireless Bad for Consensus in Blockchain?

***Abstract*—Wireless technologies now take every part of one's everyday life. As such, it will be no longer a surprise if a blockchain system is composed of wirelessly connected nodes. However, wireless communication is known for its inherent unreliability caused by noise, interference, limited bandwidth, etc. Motivated from this fundamental problem, this paper investigates the impact of wireless communications on the performance of three representative consensus mechanisms–viz., proof of work (PoW), proof of stake (PoS), and proof of coverage (PoC). It features a comprehensive analytical framework that mathematically derives metrics quantifying the scalability and the level of decentralization of the three consensus mechanisms, constituting a key contribution of this work. The paper then proceeds to presenting extensive simulation results as a means to confirm the underpinning theoretical findings. Overall, we emphasize that the framework's holisticity will allow it to be applied to diverse consensus mechanisms.**

***Index Terms*—Consensus, Scalability, Decentralization, Wireless communications**

I. Introduction

A. Motivation

In the dynamic landscape of modern connectivity, the proliferation of wireless technologies has become nothing short of permeating every facet of our daily lives and redefining the way we connect, communicate, and consume information. In fact, over 55% of website traffic comes from mobile devices, and 92.3% of internet users access the internet using a mobile phone [1]. In fact, a wide variety of wireless technologies immerse our daily lives, from Wi-Fi to the fifth-generation (5G) cellular.

This makes a strong case where *increasingly more blockchains will be established on wirelessly connected nodes*. One should notice of the key challenge here: the reliability of wireless communications is generally lower in comparison to wired communications (e.g., ethernet), attributed to various random factors such as noise, interference, and limited bandwidth [3]. One should also note that this lower reliability affects the performance of consensus in blockchain [4,5].

All open public blockchains are based on the idea that they should be able to reach consensus across a distributed network, even when there are conflicts, without putting control in one place [2]. Nonetheless, the additional dynamicity that is brought by the wireless networking has not been thoroughly discussed in the literature as of yet. Therefore, it will be a valuable academic attempt to formally analyze the performance of a blockchain system established on a wireless network.

B. State of the Art

PoW [9] and PoS [10] remain the two most common mechanisms, especially in the cryptocurrencies' context [11]. PoW is a form of cryptographic proof in which one party (the prover) proves to others (the verifiers) that a certain amount of a specific computational effort has been expended [12]. The main purpose of PoW is to deter manipulation of data by establishing large energy and hardware-control requirements to be able to do so, which inevitably leaves it criticized by environmentalists for their energy consumption. Meanwhile, as an effort to avoid such high computational cost that PoW causes, PoS is a type of consensus mechanism for blockchains that are designed to elect validators in proportion to their quantity of holdings in the associated cryptocurrency.

Moreover, there have been proposed a variety of techniques as an effort to improve the performance of blockchains: namely, parallel structure [6], off-chain [7], reinforcement learning [14], and combination of PoW and PoS [8]. Nonetheless, the *blockchain trilemma* sets a limit on this desire: no blockchain can achieve improvements on all three fronts of scalability, security and decentralization at once. In fact, despite the large research and experimental effort, all known approaches turn out to leave tradeoffs [6].

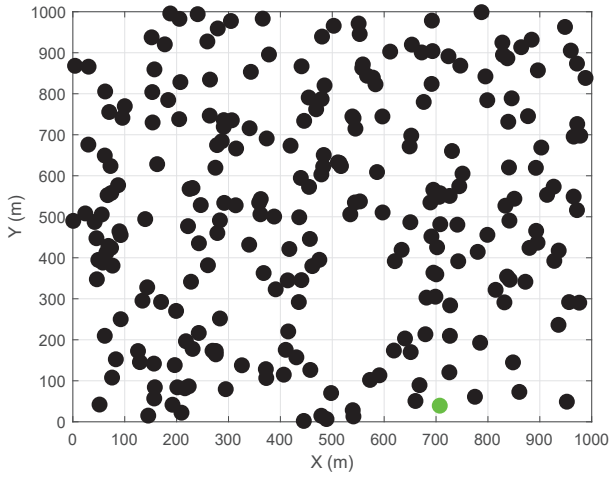
This makes a compelling case the we urgently need a comprehensive framework evaluating the performance of blockchains with *wirelessly connected* nodes with respect to the blockchain trilemma. Many of the current consensus algorithms already consider the possibilities that nodes leave, and new nodes join. However, we claim that more is needed: due to the uncertainty in wireless connections, a blockchain system established on a wireless network will draw a completely different environment than a one with ethernet-connected nodes. Specifically, the existing analytical framework (e.g., universal scalability law [30]) overlooked the impacts of wireless connectivity, which may cause serious imprecision these days with such a high proportion of wireless technologies in any given network.

Furthermore, most of the previous work in the literature of blockchain and distributed systems present experimental results, which limits their applicability to certain practical scenarios with a set of specific parameter settings. Motivated from this limitation, this paper attempts to provide a more general framework that can be applied to any other blockchains for measurement of their consensus performance.

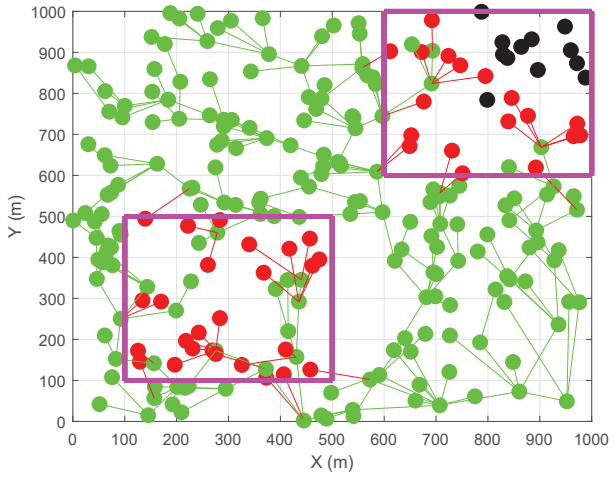
C. Contributions

Addressing the aforementioned shortcoming of the existing literature, this paper aims at precisely evaluating the performance of consensus mechanisms when the nodes are wirelessly connected. The specific technical contributions are summarized as follows:

- It provides an analytical framework for calculation of *scalability* and *decentralization* of a blockchain consensus process;
- It draws *probabilistic analysis* for formulating the impacts of wireless connections on the scalability and decentralization.
- It presents a *comparative study* among PoW, PoS, vs PoC with respect to scalability and decentralization.



(a) Before propagation start



(b) After propagation completion

Fig. 1: Example scenario of gossip protocol among 400 nodes with $p_{fail} = 0.5$ (*Green dots*–Nodes having successfully received the block; *Red dots*–Nodes not having received the block; *Black dots*–Nodes not even having been reached due to disconnection; *Magenta box*–Cluster of node failure)

II. System Model

The key focus of this paper is the nodes being wirelessly connected. As such, the system model is supposed in such a way that the impacts of wireless connection on the blockchain performance can be clearly characterized.

A. Spatial Distribution of Wireless Nodes

A two-dimensional space $\mathbb{R}^2$ is defined as a 1 km-by-1 km square, as illustrated in Fig. 1. The nodes are assumed to be located at fixed positions. We model spatial distribution of the stationary nodes as a Poisson point process (PPP) in $\mathbb{R}^2$ with density λ. Fig. 1 visualizes an example distribution of nodes with $\lambda = 400$. Notice that the λ is the parameter that will be varied according to the type of consensus mechanism. For instance, PoS has a smaller λ as compared to PoW since its consensus is operated only among the validators who are with more than a required amount of stake.

As shall be elaborated in Section II-C, node failures occur in a "*clustered*" manner; the magenta-colored boxes indicate the areas where the failures transpire. It is assumed that the node failures take place within the clusters only; the example presented in Fig. 1 assumes the probability of each node failure to be $p_{fail} = 0.5$. We will vary this p_{fail} as a method to evaluate the performance of the consensus mechanisms–viz., PoW, PoS, and PoC, which will be presented in Section V.

The selection of PPP is justified as follows. Every realization of a finite PPP is a binomial point process (BPP) with the number of realized points [15]. PPP is used to model or abstract a network composed of a possibly infinite number of nodes randomly and independently coexisting in a finite or infinite service area (e.g., large-scale wireless networks such as cellular networks). In contrast, BPP is a more adequate model when the total number of nodes is known, and the service area is finite. Therefore, our choice of PPP is in our pursuit of the *generalization* of the model.

B. Communication among Wireless Nodes

The nodes are connected to each other via wireless communications. Each node holds a 100 m of communication range. This paper assumes a general ad-hoc network in which no central mediator exists, but instead, the nodes are connected to their own neighbors, e.g., device-to-device communications [16]. The rationale is that a centralized network (e.g., cellular, Wi-Fi, etc.) will likely employ a permissioned blockchain (or a private blockchain), whose performance is not exactly what we are interested in measuring.

As a means to accomplish finality of data among nodes, distributed systems often adopt a *gossip protocol* for propagation of information [17]. In a network of size N, these protocols consist of N local, pairwise, and periodical gossiping operations between neighboring nodes.

Fig. 1 shows an example of the gossip protocol operated on a wireless ad-hoc network. The green dot in Fig. 1a indicates the node commencing the flooding, while the black dots are all the other nodes that will potentially receive the block that will be propagated. As shown in Fig. 1b, each node propagates the block to all the neighboring nodes within its communication range. Each connection is supposed to cause a Byzantine fault at the rate of p_{fail}, which is shown as a red line. Such a failed connection yields a failed reception at the receiver node, which is drawn as a red dot.

C. Byzantine Faults and Wireless Connection

Byzantine faults in safetycritical systems are real and occur with failure rates far more frequently than 10^{-9} faults per operational hour [18]. A typical example of a Byzantine fault is a digital signal that is stuck at "1/2," e.g. a voltage that is anywhere between the voltages for a valid logical "0" and a valid logical "1." Such Byzantine faults usually propagate through traditional fault containment zones, thereby invalidating system architectural assumptions.

The problem is that the wireless connection can increase the odds of Byzantine faults. Specifically, we identify *"clustered occurrence" of faults attributed to interference among each*

other [15] as the key characteristic that the wireless connection adds to Byzantine faults, which yields that the nodes located in certain areas face a higher probability of disconnection. We attribute such clustered disconnections to *interference among nodes*. In fact, an ad-hoc wireless network is more susceptible to interference [19], due to its decentralized network structure where no central mediator exists to coordinate scheduling among nodes.

III. Consensus among Wireless Nodes

Taking over from Section II-C, we delve into the impacts of wireless connections on Byzantine faults. As the first step, we define the consensus mechanisms that we analyze in this paper–i.e., PoW, PoS, and PoC–and identify their key characteristics that may be affected by the wireless connections.

Moreover, it is noteworthy that we pay attention to PoC and compare it to PoW and PoS, the two most predominating consensus mechanisms. The rationale is that PoC adopts a significantly different principle from that of PoW and PoS, which thus distinguishes it from the vast majority of recent consensus mechanisms built on PoW and PoS.

A. PoW

It is widely acknowledged that if two computers can agree upon just a single block (i.e., the head block) in a blockchain, one can then iterate through all of the blocks in the blockchain and verify that the rest of it is identical. Therefore, *forming consensus* is just a matter of agreeing upon a single block across "honest" nodes (whose acronym is "Byzantine" nodes [20]) and replicating across all of the honest nodes. *Will this be that simple if the nodes are wirelessly connected?*

Let us take a public blockchain like Bitcoin as an example and identify several key characteristics on which wireless connections may affect. Bitcoin allows its nodes to *join or leave* the network at any time. In other words, nodes should be able to join or leave our system at any time while (i) maintaining consensus; (ii) causing no deadlock in the system; and (iii) being network partition tolerant. This differentiates PoW from traditional consensus algorithms in the sense that it has to tolerate network partitions: while traditional consensus algorithms (e.g., Paxos [21], Raft [22], and practical Byzantine fault tolerance (PBFT) [20]) terminate when a consensus has been reached, PoW keeps iterating continuously, leaving a possibility that it may change at any time what the agreed-upon blockchain is.

The problem here is that if several nodes leave the network, the consensus may never be achieved due to the inability to have a sufficient number of votes to complete the algorithm (i.e., $n > 3f + 1$ for Byzantine fault tolerance (BFT)). Even worse, if several nodes join the network (and if they are Byzantine nodes), then they may be able to manipulate the vote or stop the network from achieving consensus. This is where things get more complicated with the nodes being wirelessly connected: in addition to such Byzantine nodes, even honest nodes may not be able to correctly participate in a consensus when their connections are intermittent due to uncertainties in wireless communications.

Since the Bitcoin PoW adopts a gossip protocol [17], there is a chance that two different nodes have heard about two different blocks that points to the same previous block, which forms a "*fork*" in the blockchain. The way that many PoW protocols resolve such a fork is to add a new block to the longest branch. Even if there is more than one branch with the same length, a new block can choose one randomly, which will immediately form only one longest branch and will grow thereafter via the gossip protocol.

The last characteristic that wireless connections may affect is the possibility of "*livelock*." Suppose two different forks growing separately in a blockchain, which currently shows no sign of going back to a single coherent consensus. The fundamental problem here is that we can add blocks to a chain much faster than we can learn about other nodes adding blocks to the chain. Individual computers are much faster compared to the communication time of our network between the computers, which leaves the only feasible solution to be slowing the system down. Therefore, PoW is designed such that every time a block is added, each node sleeps for a random period of time (also known as a "*timeout*"), only after which the block can be added.

B. PoS

Let us shed light on the key characteristics of PoS on which wireless connections may have impacts. Here we take the Ethereum 2.0 [10] as an example.

Selectivity in consensus participation is the first characteristic that is affected by wireless communications. The reason is that such a smaller number of consensus participants can make the blockchain more vulnerable to faults attributed to uncertainties of wireless connections in addition to already existing Byzantine faults. (This usually is not a case in a PoW-based blockchain.) Let us continue to characterize the Ethereum 2.0's PoS algorithm in further details as an effort to investigate the applicability of wireless connections.

In Ethereum's PoS system, validators commit their capital in the form of Ethereum token (ETH) to a smart contract on the network. These validators are then tasked with validating new blocks and, on occasion, creating and disseminating new blocks themselves [10]. To participate as a validator, an individual must deposit 32 ETHs into the contract and operate three distinct software components: viz., an execution client, a consensus client, and a validator client [24]. Every slot randomly selects one validator as the block proposer, responsible for creating a new block and transmitting it to other nodes in the network. Concurrently, in each slot, a committee of validators is randomly chosen, and their votes are instrumental in determining the validity of the proposed block. The partitioning of the validator set into committees is crucial for effectively managing network load.

Second, due to this unique structure, PoS has a *higher level of reliance on the network layer*. As a means to elaborate, let us take a in-depth look at the procedure of executing a transaction in Ethereum PoS:

1) A user initiates a transaction by *generating and signing it* using their private key. The user specifies the gas amount

they are willing to pay as a tip to incentivize a validator for including the transaction in a block.
2) The transaction is then *submitted to an Ethereum execution client*, where its validity is verified. This verification includes checking if the sender has sufficient ETH for the transaction and if it has been correctly signed with the corresponding key.
3) Upon confirming the transaction's validity, the execution client adds it to its local mempool (a list of pending transactions) and *broadcasts it to other nodes* through the execution layer gossip network. Other nodes, upon receiving the transaction, also add it to their mempool.
4) A node is randomly selected as the *block proposer* for the current slot via the pseudo-random RANDAO process. This proposer is responsible for constructing and broadcasting the next block to be appended to the chain, along with updating the global state.
5) Other nodes receive the new beacon block through the consensus layer gossip network. They forward it to their *execution client*, where transactions are re-executed locally to verify the proposed state change's validity. The *validator client* then attests to the block's validity, confirming that it logically follows the chain with the highest weight of attestations as defined in the fork choice rules. The block is added to the local database in each attesting node.
6) The transaction achieves a state of being "*finalized*" when it becomes part of a chain with a supermajority link" between two checkpoints, indicating agreement from 66% of the total staked ETH on the network regarding two specific checkpoints.

As shown from the above protocol, Ethereum's PoS uses two P2P networks: one for transaction communication among execution clients and another for block information exchange among consensus clients [24]. Transactions are transmitted through the execution layer's P2P network via encrypted communication between authenticated peers. When a validator proposes a block, the transactions are sent to consensus clients, encapsulated into beacon blocks, and disseminated across the P2P network. Such high reliance on networking may lead to performance degradation when the quality of communications fluctuates to a lower level.

Lastly, wireless communication may also affect the *level of "honesty*." There is a protocol that governs how honest validators are selected to propose or validate blocks, process transactions and vote for their view of the head of the chain [23]. In scenarios where several blocks occupy a similar position near the chain's head, a fork-choice mechanism is employed to pick blocks constituting the "heaviest" chain, determined by the count of validators endorsing the blocks, weighted in accordance with their staked ether balance. Elevated uncertainty in wireless communication can lead to inaccuracies in this honesty-assessing protocol, subsequently impacting the overall Byzantine fault tolerance of the network.

C. PoC

The consensus mechanism employed by Helium, known as PoC [13], is a focal point of analysis in this paper. It stands out significantly from the dominant mechanisms, namely PoW and PoS. As elaborated in Section V, PoC is found to amalgamate the benefits of PoW, such as enhanced decentralization due to fluctuations in wireless connectivity, and PoS, offering greater scalability as only specific nodes partake in the consensus process.

Within the PoC framework, miners establish physical hotspots, typically small computers with internet connectivity and antennas for wireless communication with nearby hotspots. These hotspots receive wireless data. During each "challenge," 10 witnesses are randomly selected from all hotspots within the beacon's transmission range. Notably, this random selection of witnesses can change with every challenge, occurring at 30-minute intervals; shorter intervals enhance PoC's decentralization [25], aiming for a level akin to PoW.

The fundamental principle of PoC is to validate that a given hotspot is genuinely operational and serving a real area. The term "consensus" in this context pertains to whether other hotspots can substantiate that the challenged hotspot is genuinely providing service. This verified decision by witnesses constitutes a single block in the Helium network.

This paper focuses on quantifying the impact of uncertainties arising from wireless communications on the performance of PoC consensus. For example, PoC miners facing significant uncertainties may fail to provide accurate input to the consensus, thereby compromising the blockchain's scalability. Conversely, highly variable communication channels are likely to result in the replacement of current miners with others, effectively increasing decentralization [26].

IV. Analysis on Consensus Performance

This section presents the theoretical analysis framework that this paper proposes. The framework features two metrics measuring the performance of a consensus mechanism: normalized number of transactions and Gini coefficient [27] for scalability and decentralization, respectively.

A. Scalability

The literature of distributed systems often rely on the *universal scalability law* [28,29] as the key scalability measure, which is given by

$$\mathsf{S} = \frac{N}{1 + \alpha\,(N-1) + \beta N\,(N-1)} \tag{1}$$

where N denotes the number of nodes in the network; α gives the level of contention; β indicates the delay for achieving coherency across the network. Nonetheless, we acknowledged that this quantity is usually obtained *empirically*, which makes it almost impossible to analytically quantify α and β. For this reason, we decided to move on to find another metric that can accommodate both distributed systems' and wireless communications' characteristics.

It was a challenge to find previous work exactly quantifying (α, β) of S for blockchain. However, we could find an extensive empirical analysis of the parameters for Zookeeper [30], one of the popular distributed systems [31]. In fact, Zookeeper

has appeared as on many of the blockchain systems [32,33]. This makes a case that we adopt the result of S found for the Zookeeper as a *scalability benchmark* for this paper's result, which shall be presented in Section V-B1.

Motivated from the challenge, we define a generalized metric for the scalability in this paper, which is given by

$$\mathsf{R} = \frac{\#\text{ transactions}}{\#\text{ seconds}} = \frac{\mathsf{n}_{\text{tx}}}{\mathsf{T}_c} \tag{2}$$

where T_c denotes the length of time that has been taken to complete a single consensus, which in turn can be written as

$$\begin{aligned}\mathsf{T}_{\text{c}} &= \sum_{x=1}^{X} \mathbb{1}_{x=X} \mathsf{T}\left(x\right) \\ &= \sum_{x=1}^{X} \left(\mathbb{1}_{x=X} \sum_{i,j \in \mathcal{S}(x)} \mathsf{T}_{ij} \right) \\ &= \sum_{x=1}^{X} \left(\mathbb{1}_{x=X} \mathsf{n}_{\mathbb{N}\left[\mathcal{S}(x)\right]} \mathsf{T}_{\text{to}} \right)\end{aligned} \tag{3}$$

Here are details on the parameters of Eq. (3): x denotes each consensus attempt (which is defined as an entire round of having collected verifications from all the participating nodes $\in \mathcal{S}\left(x\right)$); $\mathcal{S}\left(x\right)$ denotes the set of nodes participating in the current consensus attempt x; $X \sim \text{geo}\left(p\right)$ is the number of consensus attempts before the first successful consensus, which is modeled as a *geometric random variable*; $\mathbb{1}_{(\cdot)}$ is an indicator function that gives a 1 when $(\cdot)$ is true, or a 0 when false; n_k gives the number of communications among k nodes participating in the current consensus attempt; T_{to} gives a timeout (in the unit of seconds) for which every node must wait as a means to avoid a network partition among different nodes, which is a uniform random variable between $\left[0, \bar{\mathsf{T}}_{\text{to}}\right]$ where $\bar{\mathsf{T}}_{\text{to}}$ is the maximum value for T_{to}.

We elaborate on T_{ij}, which is defined as the delay equals the sum of the times spent at every node participating in a single networking [34]. The delay can be broken down as follows at a single node:

$$\mathsf{T}_{ij} = \mathsf{T}_{\text{que}} + \mathsf{T}_{\text{prop}} + \mathsf{T}_{\text{to}} \approx \mathsf{T}_{\text{to}} \tag{4}$$

where the subscripts "que," "prop," and "to" indicate delays due to queueing, propagation, and timeout, respectively. It is noteworthy from the reference that we consider no queueing delay as it is less significant than the network delay [34]. We also approximate that $\mathsf{T}_{\text{prop}} \approx 0$ as even wireless signals propagate at the speed of light. Nonetheless, as has been discussed in Section III-A, T_{to} is too significant to ignore. This justifies the approximation $\mathsf{T}_{ij} \approx \mathsf{T}_{\text{to}}$.

In Eq. (3), it is also significant to notice that $\mathsf{n}_{\mathbb{N}\left[\mathcal{S}(x)\right]}$ heavily depends on the type of consensus: i.e., PoW, PoS, and PoC, which is defined as the number of propagations within a consensus attempt via the gossip protocol:

$$\mathsf{n}_{\mathbb{N}\left[\mathcal{S}(x)\right]} = n\big|_{\theta(n)\to 0} \tag{5}$$

where n denotes the number of nodes in the blockchain network and $\theta\left(n\right)$ gives the rate of nodes that are "susceptible" to receive a block (whose acronym is "already received"), which is formulated as [17]

$$\theta\left(n\right) = \mathbb{E}\left[s_{t+1}\left(n\right)\right] = s_t \left(1 - \frac{1}{n}\right)^{n(1-s_t)} \tag{6}$$

where t is each time slot where a single-hop propagation takes place. To elaborate, let s_t and r_t be the proportions of nodes belonging to the "susceptible" and "already received" compartments in the tth time slot. Here, it is straightforward that $s_t = 1 - r_t$. We assume that in the first time slot, i.e., s_0, the master peer has a block to propagate, which is given by $r_0 = 1/n$ and $s_0 = 1 - 1/n$, which means that only one node has obtained the update. Now, assuming that the randomly selected node is chosen independently from other nodes and independently of past decisions, Eq. (6) is formulated as the expectation of $s_t + 1$ as a function of s_t as follows.

In Eq. (5), it is noteworthy that $\mathbb{N}\left[\mathcal{S}_{\text{pos}}\right] << \mathbb{N}\left[\mathcal{S}_{\text{pow}}\right]$ as the validators in PoS are a subset of nodes. We also emphsize that $\mathbb{N}\left[\mathcal{S}_{\text{poc}}\right] >> \mathbb{N}\left[\mathcal{S}_{\text{pow}}\right]$ is also true; however, the identities of witnesses $\in \mathcal{S}_{\text{poc}}$ in a PoC network are subject to change. Although it is not captured in the scalability, this rotation in witnesses' identities will affect the decentralization, which will be detailed in the next subsection.

Now, let us quantify p for the geometric random variable X in Eq. (3), which is defined as the probability of a successful consensus among $\mathbb{N}\left[\mathcal{S}\left(x\right)\right]$ nodes. We highlight that the probability p is a function of $\mathbb{N}\left[\mathcal{S}\left(x\right)\right]$ in every round of consensus attempt, which can be formally written as

$$\begin{aligned}p\left(x\right) &= \mathbb{P}\left[\mathbb{N}\left[\mathcal{S}\left(x\right)\right] > 3f + 1\right] \\ &= 1 - F_{\mathcal{S}}\left(3f + 1\right) \\ &= 1 - \exp\left(-\lambda_{\mathcal{S}}\right) \sum_{j=0}^{\lfloor 3f+1 \rfloor} \frac{\lambda_{\mathcal{S}}^{j}}{j!}\end{aligned} \tag{7}$$

where f denotes the number of Byzantine nodes, which is designed to fluctuate due to wireless connections. It has been proved in the literature [35] that both $\mathbb{N}\left[\mathcal{S}\left(x\right)\right]$ and f followed Poisson distributions with own respective densities, which we denote by $\lambda_{\mathcal{S}}$ and λ_f, respectively.

Here, we propose to model the impacts of wireless channel fluctuation on f as an *increase* in λ_f. This is a plausible assumption in the sense that the number of Byzantine nodes λ_f should increase as more nodes experience communication errors due to the imperfect wireless connections.

Also, notice in Eq. (7) that $F_{\mathcal{S}}$ denotes the cumulative distribution function (CDF) of random variable $\mathbb{N}\left[\mathcal{S}\left(x\right)\right]$, which can be formally written as $F_{\mathcal{S}}\left(s\right) = \mathbb{P}\left[\mathcal{S} \leq s\right]$.

B. Decentralization

Now, we proceed to formulating the decentralization part of our proposed analysis framework.

We propose to adopt *Gini coefficient* (denoted by G) to represent the level of decentralization. To elaborate, we modify G in such a way that the "inequality" indicates the level of participations being concentrated to a fewer number of nodes, which can also be understood as "centralized."

Let us consider an example scenario. Suppose a blockchain that has accomplished 100 consensuses. We divide them up among 10 nodes, giving the first 1, the second 3, etc., so that the kth person is assigned to the distribution function $u(k) = 2k - 1$. Note that as such, in this distribution we are ranking the nodes in *ascending order* according to the number of transactions whose consensus they have taken part in. Also suppose that there can be inequality: one node has partaken in only a single consensus, whereas another node has in 19 consensuses.

Given this distribution function, we can define the associated Lorenz curve, $L(x)$, as the graph of the *cumulative proportion function*. In other words, $L(x)$ is the proportion of consensuses participated by the poorest $100x\%$ of the population. In our example, $L(0.1) = 1/100$, $L(0,2) = 4/100$, and in general, $L(x) = x^2 \forall x = 0, 0.1, 0.2, \cdots, 1$. So, for example, $L(0.5) = 0.25$ which indicates that the bottom 50% of the population has participated in 25% of all the consensuses. The top 10% of the population has taken part in $1 - L(0.9) = 19/100$ or 19% of all the consensuses.

Building on this understanding, one can take a more formal way to write this decentralization analysis. We define the Gini coefficient as half of the relative mean absolute difference, which is equivalent to the definition based on the Lorenz curve [36]. The mean absolute difference is the average absolute difference of all pairs of items of the population, and the relative mean absolute difference is the mean absolute difference divided by the average, $\bar{x}$, to normalize for scale. If x_i is the wealth or income of person i, and there are n persons, then G is given by

$$\mathsf{G} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2\sum_{i=1}^{n}\sum_{j=1}^{n} x_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2n^2\bar{x}} \tag{8}$$

where $\frac{1}{2}$ is there because the $\sum_i \sum_j$ yields two sums for each pair of i and j and thus needs to be divided by 2. Further, $\frac{1}{n^2}$ is to compensate the numerator's scale-up by being summed among $n \times n$ instances. That means, a G is defined to measure the wealth in a *per capita normalized* manner.

It is with the highest importance in this paper to quantify how intermittence due to wireless connection affects the decentralization level of a consensus mechanism. From the discussion provide so far, one can find that if the wireless disconnectivity is *distributed across the network equally*, it should not affect G. This is proved in our results: see Fig. 6 in Section V-B1.

V. Numerical Results

Now, we present the results of R and G, the metrics that this paper proposes to measure the *scalability* and the level of *decentralization* of a consensus mechanism, respectively.

A. Parameters and Setup

We identify several key parameters that we set for the experiments as follows.

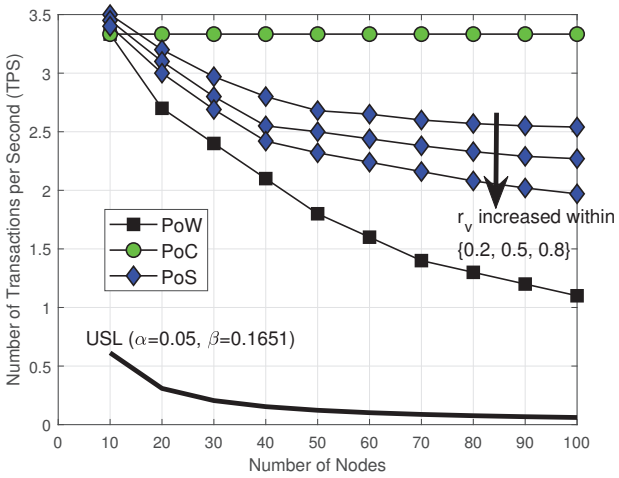


Fig. 2: PoW, PoS, and PoC—Scalability R versus the number of nodes (p_{fail} = 5%, 10 transactions per block)

Number of Nodes: Recall from Section II that we adopt a PPP for the spatial distribution of the wireless nodes. We set the density of the PPP $\lambda = 100$, meaning that the number of nodes distributed in a two-dimensional space $\mathbb{R}^2$ (with the dimension of 1 km by 1 km) follows a Poisson random variable, i.e., Poiss (100). (Revisit Fig. 1 for the layout of our nodes distribution principle.)

Communication Range: Each node is assumed to have 100 m of the communication range: i.e., both transmission and reception of a digital packet.

Number of Transactions per Block: We also assume 10 transaction per block. It is significant to note that this quantity is easily configurable according to various practical scenarios, which forms the key benefit of the analysis framework that this paper proposes.

Number of Iterations: This experiment is generated with several parameters that are highly random: namely, the nodes' positions, the probability of Byzantine failure, the probability of wireless disconnectivity, the rate of validators (for PoS), etc. As such, we have run $1e3$ iterations for every experiment as a means to average out the randomness and hence produce more statistically stable results.

B. Results and Discussion

1) Scalability: We start with presenting the results of R, which has been defined as Eq. (2) in Section IV-A. We recall from the section that this paper adopts S for a Zookeeper network as the scalability benchmark, which is presented as a black solid line in all of Figs. 2 through 5 and is compared to the R's of PoW, PoS, and PoC.

Fig. 2 compares the three consensus mechanisms–i.e., PoW, PoS, and PoC–on the scalability, R. One can easily find that PoC is the most scalable among the three since it is designed to select a fixed number of witnesses, which also explains the "flatness" of its scalability versus the number of nodes. Due to such selectivity, PoC should pose a lower level of decentralization, as shall be presented in subsection V-B2.

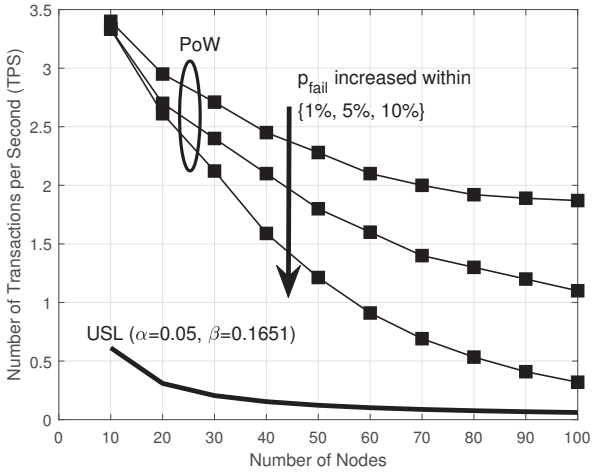


Fig. 3: PoW—Scalability R versus the number of nodes (p_{fail} = $\{1, 5, 10\}\%$, 10 transactions per block)

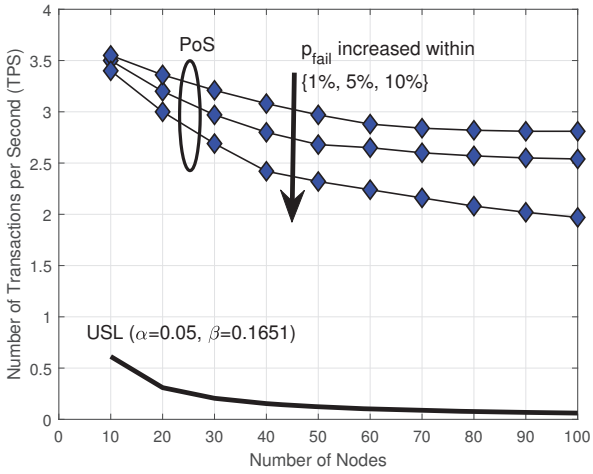


Fig. 4: PoS—Scalability R versus the number of nodes (p_{fail} = $\{1, 5, 10\}\%$, 10 transactions per block)

In Fig. 2, one can find it interesting that the scalability of PoS varies in accordance with parameter r_v, indicating the *rate of validators*. We set the parameter as a Poisson random variable with a justification that there must be a "certain amount" of tokens that the largest number of participants hold. Suppose an Ethereum network. The number of ETHs will vary by holders; but the validators are strictly required to hold more than 32 ETHs. It means that the rate of validators can be formally written as

$$r_v = 1 - \mathbb{P}[Z \geq 32] \tag{9}$$

where $Z \sim (\lambda_{eth})$ denotes a Poisson random variable indicating the number of ETHs held by a holder. The Poisson random variable was chosen to represent the distribution of ETHs across the globe; this modeling refers to a recent study that proved that the distribution of wealth converges to a Poisson distribution [37].

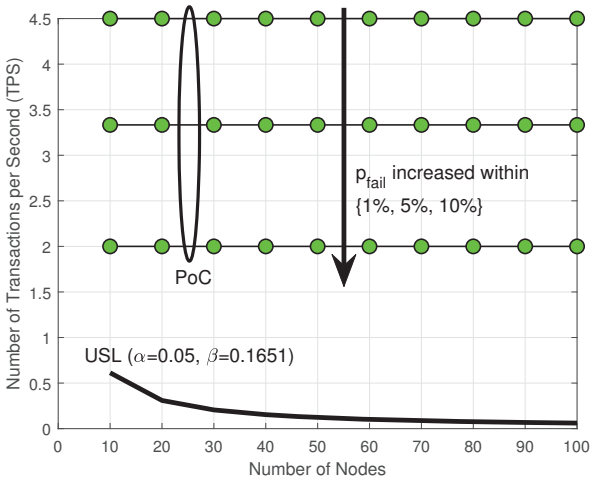


Fig. 5: PoC—Scalability R versus the number of nodes (p_{fail} = $\{1, 5, 10\}\%$, 10 transactions per block)

Through Figs. 3 and 5, we focus on visualizing the scalability R for each of the three consensus mechanisms–i.e., PoW, PoS, and PoC, respectively–with the probability of Byzantine failure at each node (due to the uncertain wireless connection), p_{fail}, varied. One can notice a straightforward but clear *inversely proportional* relationship between p_{fail} and R: a higher p_{fail} yields a lower R. The rationale behind this relationship is as follows. A consensus protocol adopts a gossip protocol to propagate the given block. If achieved among only $n < 3f-1$ nodes, the consensus is regarded to have a Byzantine fault, which calls for another round of consensus attempt. This is where a significant increase in the consensus delay T_c occurs, which negatively affects the scalability R. (See Eqs. (2) and (3) for formulation of the two parameters.)

Fig. 5 reveals a particularly interesting phenomenon in regard to R for PoC. As has already been discussed earlier in this subsection, R remains flat regardless of the number of nodes participating in consensus because PoC employes exactly 10 witnesses for each consensus. However, the same principle applies to PoC as well: a higher p_{fail} degrades R due to the need for multiple block propagation attempts for a single consensus.

2) Decentralization: Fig. 6 indicates that a higher wireless disconnection rate increases the decentralization level. For PoS, the rate of validators among all the nodes is set to $r_v = 0.2$. For PoC, the rate of node replacement in each round of consensus is set to $r_{sfl} = 0.9$, and the number of rounds between a replacement is $\Delta_{sfl} = 0$. To wit, r_{sfl} gives *how many of the current witnesses* will be replaced with other ones; and Δ_{sfl} means *how frequently* the replacement occurs.

Recall that G closer to 0 gives a higher level of decentralization. (This may be a surprise to some readers: the wireless connection actually helps *improve* the decentralization in blockchain!) The reason for this result is straightforward. PoS and PoC allow only "selective" nodes to participate in a consensus. The intermittent wireless connection hinders such participation. Therefore, the level of "inequality" is decreased among nodes participating in consensuses.

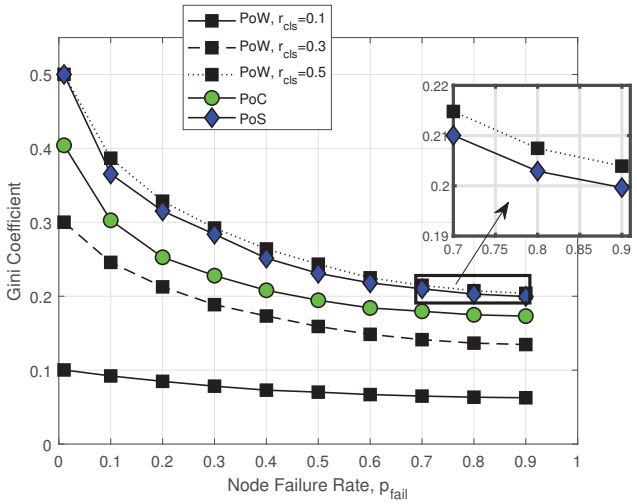


Fig. 6: PoW, PoS, and PoC—Decentralization level G vs. node failure rate (PoW–$r_{cls} = \{0.1, 0.3, 0.5\}$; PoS–$r_v = 0.2$; PoC–$r_{sfl} = 0.9$, $\Delta_{sfl} = 0$)

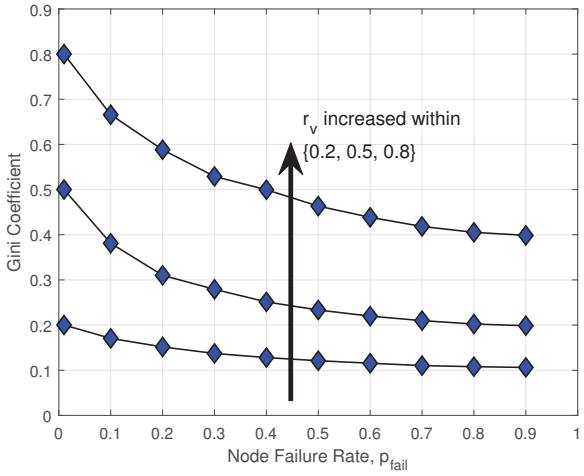


Fig. 7: PoS—Decentralization level G vs. node failure rate ($r_v = \{0.2, 0.5, 0.8\}$)

Notice from Fig. 6 that the rate of cluster (denoted by r_{cls}) was varied with PoW only. The rationale is that PoW is particularly susceptible to such clusteredness of node failure transpiration. In contrast, PoS and PoC will be less impacted since they already adopt "selective" participation of nodes in a consensus anyway; to wit, if failures occur at non-participating nodes, they have almost zero impact on consensus. We elaborate that the rate of cluster differentiating the PoW curves in Fig. 6 is formally written as $r_{cls} = \left(\sum_{i=1}^{\mathbb{N}[\mathcal{S}_{cls}]} \left|\mathbb{R}_{cls,i}\right|\right) / \left(\left|\mathbb{R}^2\right|\right)$ where $\mathbb{R}_{cls,i} \in \mathcal{S}_{cls}$ denotes the ith cluster within the entire region $\mathbb{R}^2$; $|\cdot|$ gives the area of a two-dimensional space.

One should also notice that the three consensus mechanisms pose the order of PoW > PoC > PoS in terms of the decentralization level. This reflects the proportion of chance participating in a consensus: PoS is the most selective according to the number of staked tokens; PoC is less selective as the random wireless connectivity gives a chance of rotating participations in consensus; PoW is almost not selective by its definition.

Fig. 7 is focused on PoS to highlight the impacts of the validator selectivity r_v on the decentralization level G. The result is forthright: a lower r_v (indicating the election of a larger number of validators) yields a lower G (meaning a higher level of decentralization).

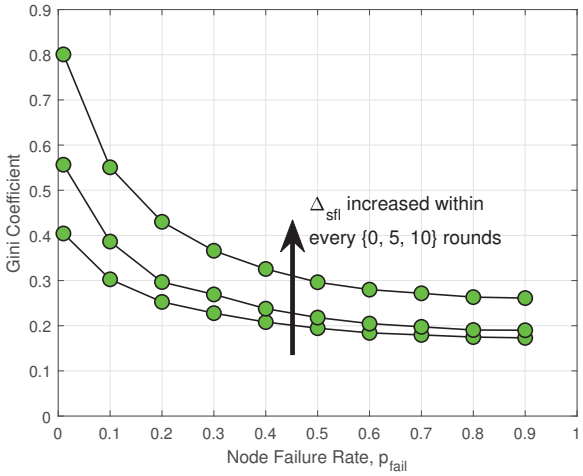


Fig. 8: PoC—Decentralization level G vs. node failure rate ($\Delta_{sfl} = \{0, 5, 10\}$)

Fig. 8 visualizes the impact of wireless disconnectivity on the decentralization level in PoC. The same phenomenon holds: a higher level of disconnectivity improves the decentralization. The particular focus in the figure is variation of Δ_{sfl}, the number of consensus rounds between participating nodes elections: the decentralization is degraded as the re-election of witnesses occur less frequently.

VI. Concluding Remarks

This paper has drawn a theoretical framework assessing the impacts of wireless connectivity on the performance of consensus in a blockchain system. It has formulated the throughput and the Gini coefficient as the metrics for measuring scalability and decentralization of a consensus mechanism, respectively.

The simulation implemented the formulations and revealed that the imperfect connectivity due to wireless deteriorated scalability but improved decentralization. The reason for the scalability degradation is that a higher chance of Byzantine failure leads to a more frequent need for a re-attempt of consensus, which in turn causes a significant delay. Meanwhile, such a higher probability of failure increases the decentralization level because a more frequent failure gives a wider variety of nodes a chance to participate in the consensus.

As future work, we will extend the findings of this paper to analyzing how mobility affects the consensus in blockchain. Interestingly, the literature of wireless communications already acknowledged that higher mobility could increase the chance of successful connections in a wireless ad-hoc network [38].

Moreover, we will investigate impacts of off-chain PoC migration (as part of the recent migration to Solana [39]) on this paper's result. The scalability and decentralization may need to be re-assessed, depending on the characteristics of the oracle that is introduced by the migration, which may necessitate modifications in this paper's analysis framework.

REFERENCES

[1] J. Howarth, "Internet traffic from mobile devices (Nov. 2023)," [Online]. Available: https://explodingtopics.com/blog/mobile-internet-traffic

[2] M. Bedawala, "A deeper dive into consensus mechanisms," [Online]. Available: https://usa.visa.com/solutions/crypto/consensus-mechanisms.html#:~:text=In%20blockchains%2C%20reaching%20consensus%20is,distributed%20nature%20of%20the%20network.

[3] R. Jordan and C. Abdallah, "Wireless communications and networking: An overview," *IEEE Antennas Propag. Mag.*, vol. 44, no. 1, 2002.

[4] X. Zhang, W. Xia, Q. Cui, X. Tao, and R. Liu, "Efficient and trusted data sharing in a sharding-enabled vehicular blockchain," *IEEE Netw.*, Aug. 2022.

[5] S. Okegbile, J. Cai, and A. Alfa, "Performance analysis of blockchain-enabled data-sharing scheme in cloud-edge computing-based IoT networks," *IEEE Internet Things J.*, vol. 9, no. 21. Jun. 2022.

[6] G. Monte, D. Pennino, and M. Pizzonia, "Scaling blockchains without giving up decentralization and security: A solution to the blockchain scalability trilemma," in *Proc. Wksp Cryptocurrencies Blockchains Distributed Syst. 2020*.

[7] S. Reno and M. Haque, "Solving blockchain trilemma using off-chain storage protocol," *IET Inf. Secur.*, vol. 4, Jul. 2023.

[8] Trifecta Blockchain Team, "Trifecta: The blockchain trilemma solved," *White Paper*, Oct. 2019. [Online]. Available: https://pramodv.ece.illinois.edu/pubs/Whitepaper2019-9.pdf

[9] S.Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Oct. 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] Ethereum Foundation, "Proof-of-stake (PoS)," Sep. 2023. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

[11] European Parliament, "Cryptocurrencies and blockchain," PE 619.024, Jul. 2018. [Online]. Available: https://www.europarl.europa.eu/cmsdata/150761/TAX3%20Study%20on%20cryptocurrencies%20and%20blockchain.pdf

[12] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," *LNCS Advances Cryptology*, vol. 740, 1992.

[13] Helium Network, "Proof of coverage," [Online]. Available: https://docs.helium.com/iot/proof-of-coverage/

[14] S. Kim and A. S. Ibrahim, "Byzantine-fault-tolerant consensus via reinforcement learning for permissioned blockchain implemented in a V2X network," *IEEE Trans. Intell. Veh.*, vol. 8, iss. 1, Jan. 2023.

[15] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey," *IEEE Commun. Surveys Tut.*, vol. 15, no. 3, Third Quarter 2013.

[16] 3GPP, "Universal Mobile Telecommunications System (UMTS); LTE; Proximity-based services (ProSe); Stage 2," *ETSI TS 123 303*, V17.1.0, Jul. 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123300_123399/123303/17.01.00_60/ts_123303v170100p.pdf

[17] M. Jelasity, *Gossip*, Springer Berlin Heidelberg, pp. 139–162, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-17348-6_7

[18] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg, "Byzantine fault tolerance, from theory to reality," in *Proc. Int. Conf. Comput. Safety, Reliability, Security 2003*.

[19] P. Cardieri, "Modeling interference in wireless ad hoc networks," *IEEE Commun. Surveys Tut.*, vol. 12, no. 4, Fourth Quarter 2010.

[20] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. USENIX Operating Syst. Design Implementation 1999*.

[21] L. Lamport, "Paxos made simple," *ACM SIGACT News* vol. 32, no. 121, Dec. 2001.

[22] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm (extended version)," [Online]. Available: https://raft.github.io/raft.pdf

[23] Ethereum Foundation, "Consensus mechanisms," Jan. 2023. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/

[24] Ethereum Foundation, "Networking layer," Apr. 2023. [Online]. Available: https://ethereum.org/en/developers/docs/networking-layer/#connecting-clients

[25] A. Haleem, "Helium network (HNT): Decentralizing wireless networks," Oct. 2023. [Online]. Available: https://www.gemini.com/cryptopedia/helium-network-token-map-helium-hotspot-hnt-coin

[26] The Wireless Miner, "An introduction to proof of coverage for blockchains," Feb. 2022. [Online]. Available: https://thewirelessminer.com/2022/02/21/an-introduction-to-proof-of-coverage-for-blockchains/

[27] C. Gini, "On the measure of concentration with special reference to income and statistics," *Colorado College Publication*, General Series no. 208, pp. 73–79.

[28] P. Jogalekar and M. Woodside, "Evaluating the scalability of distributed systems," *IEEE Trans. Parallel Dist. Syst.*, vol. 11, no. 6, Jun. 2000.

[29] N. Lu and X. Shen, "Scaling laws for throughput capacity and delay in wireless networks–A survey," *IEEE Commun. Surveys Tut.*, vol. 16, no. 2, 2014.

[30] M. Bevilacqua-Linn, M. Byron, P. Cline, J. Moore, and S. Muir, "Sirius: Distributing and coordinating application," in *Proc. USENIX Ann. Techn. Conf. 2014*.

[31] P. Hunt, M. Konar, F. Junqueira, and B. Reed, "ZooKeeper: Wait-free coordination for internet-scale systems," in *Proc. USENIX Ann. Techn. Conf. 2010*.

[32] E. Bandara, D. Tosh, P. Foytik, S. Shetty, N. Ranasinghe, and K. Zoysa, "Tikiri–Towards a lightweight blockchain for IoT," *Elsevier Future Generation Comput. Syst.*, vol. 119, 2021.

[33] J. Duan, A. Karve, V. Sreedhar, and S. Zeng, "Service management of blockchain networks," in *Proc. IEEE Int. Conf. Cloud Comput. 2018*.

[34] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks – Part I: The fluid model," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, 2006.

[35] S. Kim and B. J. Kim, "On the Byzantine-fault-tolerant consensus in blockchain built on internet of vehicles," in *Proc. IEEE Int. Conf. Electron. Inf. Commun. (ICEIC) 2022*.

[36] A. Sen, *On Economic Inequality*, Oxford University Press, 1977.

[37] F. Cao and N. Marshall, "From the binomial reshuffling model to Poisson distributin of money," *arXiv:2212.14388v1*, Dec. 2022.

[38] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, Aug. 2002.

[39] Helium Foundation, "HIP-70: Scaling the Helium network," May 2023. [Online]. Available: https://github.com/helium/HIP/blob/main/0070-scaling-helium.md