# A Location-based and Hierarchical Framework for Fast Consensus in Blockchain Networks

Hao Guo*, Wanxin Li†, Mark Nejad†
*School of Software, Northwestern Polytechnical University, Taicang Campus, China
† Department of Civil and Environmental Engineering, University of Delaware, U.S.A.
haoguo@nwpu.edu.cn & {wanxinli,nejad}@udel.edu

2021 4th International Conference on Hot Information-Centric Networking (HotICN) | 978-1-6654-0549-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/HOTICN53262.2021.9680858

***Abstract*—Blockchain-based IoT systems can manage IoT devices and achieve a high level of data integrity, security, and provenance. However, incorporating the existing consensus protocols in many IoT systems limits scalability and leads to high computational cost and network latency. We propose a hierarchical and location-aware consensus protocol for IoT-blockchain applications inspired by the original Raft protocol to address these limitations. The proposed consensus protocol generates the consensus candidate groups based on nodes' individual reputation and distance information to elect the leader in each sub-layer blockchain and uses our threshold signature scheme to reach global consensus. Experimental results show that the proposed consensus protocol is scalable for large IoT applications and significantly reduces the communication cost, network latency, and agreement time by more than 50% compared with the Raft protocol for consensus processing.**

***Index Terms*—Blockchain, Consensus Protocol, Geographic Information, Hierarchical Architecture, Hyperledger Ursa, Internet of Things.**

## I. Introduction

Modern IoT networks are often large-scale, dynamically located, and globally distributed. By 2025, IoT devices such as smart home appliances, smartphones, and other types of smart sensors will increase to more than 75 billion [1]. Many IoT networks require massive data communication and need to manage unreliable and failure messages, among others, automatically. Consensus protocols promise to achieve overall system reliability in the presence of inconsistent and failure messages by coordinating processes to reach agreements. State machine replication (SMR) is a fundamental method for system availability and fault tolerance in the distributed systems. For example, Autopilot [2] builds fault-tolerant replicas in Microsoft's data centers worldwide using consensus protocols. Google File System utilizes the Chubby [3] lock service to reach the consensus for the replication of different files. However, there are significant challenges in the current IoT applications, including data integrity, resource-intensive consensus mechanisms, high latency, and limited scalability [4].

Blockchain is a distributed ledger that records transactions among multiple participants in a verifiable manner. Blockchain can reduce the costs involved in verifying transactions as a distributed ledger by removing the need for a trusted third-party operating as a centralized authority. Since the introduction of Bitcoin [5], blockchain applications have expanded beyond cryptocurrencies and financial-related fields. The smart contract's invention [6], [7] leads to developing more varied applications, such as blockchain-based intelligent transportation systems (e.g., [8]–[11]) and smart health (e.g., [12], [13]). However, blockchains with a complex application layer and smart contracts can incur significant computation for transaction execution.

Many IoT-blockchain applications, such as for managing the electric power grid [14], can benefit from a hierarchical architecture to reduce the consensus process and data communication time. A hierarchical multi-layer blockchain network can communicate within its sub-layers and achieve the consensus in a more efficient way [15]. In this paper, we propose a novel consensus protocol for blockchain-based IoT applications, which is inspired by the original Raft protocol [16]. By incorporating the geographic information of the IoT devices, our propose consensus protocol boosts the blockchain performance and makes the system more dynamic and immune to the Sybil attacks [17].

In each sub-layer blockchain network, our consensus protocol engages a few local nodes with a low consensus latency, making the blockchain-based IoT system more efficient. Our scheme generates candidate groups based on nodes' reputation and distance information to elect the leaders. Our design solution partitions the blockchain network into a hierarchical structure of sub-blockchains with a threshold signature scheme to reach the global consensus to achieve higher system scalability.

Specifically, we design the novel consensus protocol, which constructs sub-layers *local consensus* based on IoT devices' reputation and geographic information to elect the leader, and construct a hierarchical architecture by utilizing a threshold signature scheme to reach *global consensus* among the multiple layers. We evaluate the new consensus protocol's performance and compare it with the classical Raft protocol. Moreover, we experiment with a threshold signature scheme for both signing and verification time using the Hyperledger Ursa cryptography library [18]. Compared to the original Raft protocol, our proposed scheme reach consensus significantly faster with lower network overhead.

The rest of the paper is organized as follows. We discuss the related work in Section II. In Section III, we describe the system architecture. In addition, we present the threshold signature scheme and location-based hierarchical raft protocol.

In Section IV, we conducted experiments and evaluations based on our proposed new consensus protocol. In Section V, we conclude the paper and point out future research directions.

## II. Related Work

Yu et al. [19] proposed a hierarchical edge-cloud blockchain architecture named LayerChain. They described a layered structure to save the blockchain transaction data in multiple distributed clouds and edge nodes. Moreover, to mitigate lengthy delays during block propagation, they developed a tree-based clustering algorithm where blocks are propagated via different clusters with a compressed tree depth. Chuang et al. [20] proposed a hierarchical blockchain-based data service platform in MEC environments. This system provided an adaptive PoW consensus scheme that dynamically changed the hash puzzle's difficulty and enhanced resource-constrained IoT devices. Yang et al. [21] proposed a hierarchical trust networking architecture to implement JointCloud (HTJC). By developing the credit bonus-penalty strategy (CBPS), HTJC can address the trust issue and provide participants with a secure and trusted trade environment.

Lao et al. [22] proposed G-PBFT (Geographic-PBFT), a location-based and scalable consensus mechanism for IoT-blockchain applications. In their design, G-PBFT utilized the era switch mechanism to maintain the dynamics in the IoT devices. The experiment results showed that G-PBFT reduced the network overhead and consensus time significantly. Li et al. [15] proposed a scalable multi-layer PBFT consensus protocol for blockchain. The proposed double-layer PBFT scheme reduces the communication complexity significantly. They also analyzed the security threshold based on the faulty probability determined and the faulty number determined models. An et al. [23] proposed a decentralized privacy-preserving model based on the twice verifications process and consensuses of the blockchain system. They introduced a twice consensus mechanism, ensuring that data can be traced and prevented from being impersonated and denied.

## III. System Architecture

The underlying location-centric characteristics inherent in many IoT applications necessitates location-aware design solutions. This section describes our proposed system architecture for location-aware consensus in IoT networks. In our approach, the global network is divided into sub-blockchains based on regional information. These sub-blockchains are connected in a hierarchical structure forming the more extensive global blockchain network. The system provides a multi-chain and multi-level structure, as shown in Fig. 1. We first define the following entities that take part in the proposed architecture.

- Blockchain: Blockchain serves as the coordinator for IoT devices and manages data sharing and access activities. It forms a dynamic hierarchical structure based on the IoT device's geographic information, including top, middle, and leaf layers.
- IoT Nodes: The IoT nodes participate in the consensus process. There are three types of IoT nodes: *Follower*, *Candidate*, and *Leader* nodes. They can switch the role seamlessly between these statuses. All *Candidate* nodes together form the consensus nodes group, which elect the *Leader* node.
- Client: A client node only requests new transactions to append data to the ledger, and they do not participate in the consensus procedure. For example, a healthcare system's smart devices can host client nodes to request electronic health record updates.
- Threshold Signature Scheme: The threshold signature scheme is proposed to achieve consensus among hierarchical blockchain layers in the architecture. We will explain the detailed construction in the following subsections.

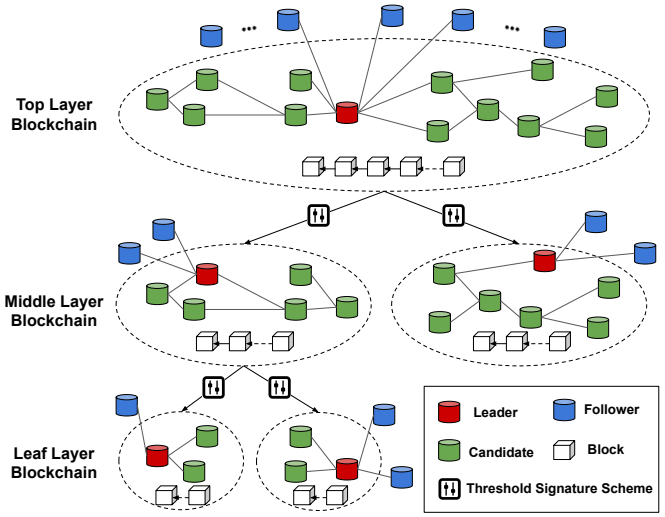


Fig. 1: Location-based Hierarchical Blockchain Architecture for IoT Applications.

As shown in Fig. 2, the proposed location-based hierarchical raft protocol has three participant entities: *Follower*, *Candidate*, and *Leader*. These interconvertible nodes can change their status to other roles. The *Candidate* and *Leader* nodes participate in the consensus process. The *Leader* nodes maintain the integrity and confidentiality of the blockchain system and broadcast the newly generated transactions to the *Follower* nodes. By contrast, the *Follower* nodes will only start new election processes and form *Candidate* nodes groups based on their geographic information. Transactions are determined among *Candidate* and *Leader* nodes to reduce communication overhead. If any message is failed, a node will resend the message again after the timeout period. The role of a node in our proposed scheme is not fixed; a *Follower* node can become the *Candidate* node, and *Candidate* node can become the *Leader* node. On the other hand, if the location of a *Leader* node has been changed or it conducts malicious action, it can be detected by the voting process by the *Candidate* nodes.

The proposed architecture is designed in a way that is not affected by the size of the IoT network. Rather than all nodes participating in the consensus procedure, nodes execute

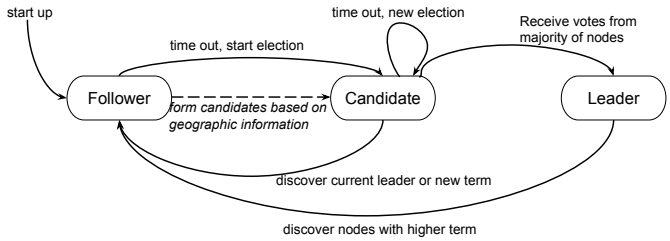


Fig. 2: Location-based Hierarchical Raft Protocol.

*local consensus* within each sub-layer blockchain. Each sub-layer blockchain leader engages in the hierarchical consensus by utilizing the threshold signature scheme to reach *global consensus*.

In the remainder of this section, we first present the threshold signature scheme to reach the consensus between multiple blockchain layers in the hierarchical architecture. Next, we describe the location-aware hierarchical raft protocol with detailed constructions.

### A. Threshold Signature Scheme

We proposed the threshold signature scheme for trust-based hierarchical coordination and operation between two blockchain layers. For example, assume that a lower layer blockchain network contains $n$ consensus nodes and the threshold is set as $t$-out-of-$n$ between the lower layer and the upper layer blockchain network. The upper layer blockchain, acting as the verifier network, will trust this lower layer blockchain only if at least $t$ consensus nodes' signatures are verified as legitimate. In addition, utilizing bilinear pairing-based cryptography, the signatures can be verified without disclosing any sensitive information.

**Algorithm 1:** Key Generation

**Input :** For each consensus node $i$
**Output:** Verifier key $v_i$

The permission issuer selects a random $a_i \in \mathbb{Z}_p$ for consensus node $i$ ;
The permission issuer computes the verifier key as $v_i = g^{a_i} \in G$ ;
The permission issuer returns $v_i$ ;

**Algorithm 2:** Threshold Signature Generation

**Input :** Each consensus node's MAC address $m_i$
**Output:** One-time signature $\delta_i$

The system computes a hash digest $h_i$ based on MAC address $m_i$ via [24], as $h_i = H(m_i)$ ;
The system generates the one-time signature $\delta_i = {h_i}^{a_i} \in G$ ;
The system sends $\delta_i$ to upper layer network ;

We describe the procedure of the proposed threshold signature scheme in Algorithms 1, 2, and 3. The algorithm contains three main functions: Algorithm 1 shows the key generation function by the system administrator; Algorithm 2 describes the threshold signature generation by lower layer blockchain nodes; Algorithm 3 presents threshold signature verification process by upper layer blockchain nodes, and a threshold $(t,\ n)$ is considered in verifying the lower layer blockchain network. The detailed construction of the proposed threshold signature scheme is shown as below:

**Algorithm 3:** Threshold Signature Verification

**Input :** One-time signature $\delta_i$, hashed MAC address $h_i$, verifier key $v_i$
**Output:** Identity verification result $r$

k = 0;
**for** *each one-time signature* $\delta_i$ **do**
  **if** $e(\delta_i, g) == e(h_i, v_i)$ **then**
    $r_i = True$ ;
    $k = k + 1$;
  **else**
    $r_i = False$ ;
  **end**
**end**
the system checks **if** $k \geq t$ **then**
  $r = True$ ;
**else**
  $r = False$ ;
**end**
The system returns $r$ ;

*Initial Setup*: The system has the bilinear pairing function $e$: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, the secure hash function $H : M \rightarrow \mathbb{G}_1$, and the $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, p, h)$ represents the public parameters.

*Key Generation*: A trusted authority generates signing-verifying key pairs for all consensus nodes in this step. The key generation function selects a random integer $a_i$ as the signing key and computes $g^{a_i}$ as the verifying key for the consensus node $i$.

*Signing*: Each consensus node $i$ computes its hashed identity information $m_i$ as $h_i = H(m_i)$, where $H$ is a hash function such as SHA256 algorithm [24]. Then, this consensus node generates the one-time signature $\delta_i = {h_i}^{a_i}$ and sends it to the upper layer blockchain network.

*Verifying*: Given the one-time signature $\delta_i$ and the verifying key $v_i$, the upper layer blockchain network can verify that $e(\delta_i, g) = e(h_i, v_i)$. This holds because $e({h_i}^{a_i}, g) = e(h_i, g^{a_i}) = e(h_i, g)^{a_i}$ due to the *Bilinearity*. Based on the threshold condition, the validity of the lower layer blockchain network $r$ is verified only if at least $t$-out-of-$n$ consensus nodes' one-time signatures $r_i$ $(1 \leq i \leq n)$ are verified, we write $(r_1, r_2, ..., r_n) \xrightarrow{(t,n)} r$.

$(t, n)$ threshold signature scheme has been applied based on the BLS signature scheme, where $1 \leq t \leq n$. For instance, If consensus nodes generated three different signatures related to the identity and location information. The upper layer verifier

node, which verifies the generated signature $\delta_i$ for consensus node $i$, will follow the threshold of *2 out of 3* to authenticate the identity and location information. Also, the verifier node can apply the threshold signature scheme dynamically, such as the *1 out of 3* rule, to provide more flexibility on the trust between the upper and lower layer blockchain network.

Compare to the original *Raft* protocol, our proposed new consensus algorithm could tolerate the crash and byzantine fault. Classical *Raft* protocol requires that all participant nodes are honest and conduct truthful action. By applying the *t-of-n* threshold scheme, the verifier node could check and verify the generated signatures independently with fault-tolerance property and protecting the privacy of candidate nodes.

### B. Location-based Hierarchical Raft Protocol

We describe the process of the location-based hierarchical raft protocol in Algorithm 4. The algorithm contains six primary phases: lines 2-7 shows the startup of election by a follower node; lines 8-11 describes the nearby node $n_i$ ($\forall\ n_i$; $n_i \in N$) sends its candidate group formation (CGF) score (including both reputation and geographical information) to the follower node; lines 12-15 presents the follower node $F$ sorts the $CGF_i$ score $\forall i \in \mathcal{N}$, and chooses the top M nodes to form the candidate group $C$ and broadcasts the candidate group $C$ information, lines 16-18 indicates the confirmation information of candidate group, lines 19-22 shows the confirmation of the elected leader, and finally lines 23-26 presents the current leader switches its role to the follower.

Each IoT device must periodically submit its location, reputation, and timestamp information in the new consensus protocol. We utilize the Crypto-Spatial Coordinates (CSC) mechanism to connect geographical information of IoT devices [22]. With CSC, IoT devices could have access to their historical location information. After the qualified IoT node is elected as the *Leader* node, it will start to validate and generate a new block and manage blockchain new transaction based on our proposed consensus protocol. If there is a missing block or forking issue caused by the *Leader* node, *Leader* node will be removed from its *Leader* status.

To become the qualified *candidate* nodes, the *follower* nodes need to satisfy the geographic location requirements. Therefore, the new consensus protocol will check the geographic information of *follower* nodes periodically. It will determine if the *follower* nodes are within a particular geographic area and whether the node changes its location over a while. If a node's geographic location information has been changed significantly over the past period *t*, it will be removed from the *candidate* group.

**Algorithm 4:** Proposed Consensus Protocol

1: **OUTPUT:** The $Leader$ of consensus Protocol
2: **START UP** *Follower node $F$ begins the Election.*
3: $F$ sends $FormGroup$ request to all nearby nodes $N$;
4: $F$ waits for messages in a time period T;
5: $F$ gets $CGF$ scores from all nearby nodes $N$;
▷ include reputation and geographical score for each node $n_i \in N$
**if** *no $answer$ from $N$ within time $T$* **then**
6: $F$ restarts $Election$ procedure;
7: **END START UP**
8: **UPON EVENT** *Nearby nodes $N$ receive the FormGroup message:*
9: $N$ calculate $CGF_i$ score for each $n_i$;
10: each $n_i$ sends $CGF_i$ score to $F$;
11: **END UPON EVENT**
12: **UPON EVENT** *Follower node $F$ receives the $CGF$ score from all nearby nodes $N$:*
13: $F$ sorts all $CGF_i$ scores and chooses the top $M$ nodes;
14: $F$ broadcasts candidates group $C$ to all followers $F$;
15: **END UPON EVENT**
16: **UPON EVENT** *Nearby nodes $N$ receive the Follower message:*
17: $N$ accept the nodes $M$ in candidate group $C$;
18: **END UPON EVENT**
19: **UPON EVENT** *Candidates $C$ reach timeout and starts new $Election$:*
**if** *$C_i$ receives $majority$ votes* **then**
20: $C_i$ becomes the $Leader$ and broadcasts the $Leader$ message to $Followers$ and $Candidates$;
**else**
21: $C_i$ waits for $Leader$ message from other candidates $C$;
22: **END UPON EVENT**
23: **UPON EVENT** *Leader $L_i$ discovers nodes with higher term:*
24: $L_i$ accepts other node $L_j$ as the new $Leader$;
25: $L_i$ switches its role to $Follower$;
26: **END UPON EVENT**

## IV. Experiments and Evaluations

We developed the IoT blockchain system prototype with the new consensus protocol and compared our proposed protocol with the classical Raft protocol.

### A. Communication Cost

Our proposed consensus protocol could reduce the communication cost significantly when the number of IoT devices is large. We set the sub-layer blockchain's *Candidate* nodes to be 20 at maximum and evaluate the communication cost for a single blockchain transaction. As shown in Fig. 3, communication cost in the classical Raft protocol keeps increasing when the number of nodes increases. Moreover, the larger is the number of nodes participating in the system, the more significant is the increase in communication cost. However, for our proposed consensus protocol, the communication cost reaches the upper bound of about 470kb since the sub-layer blockchain has the maximum capability for *Candidate* nodes.

Similar to the network latency analysis, the classical Raft protocol cannot work well when candidate nodes are greater

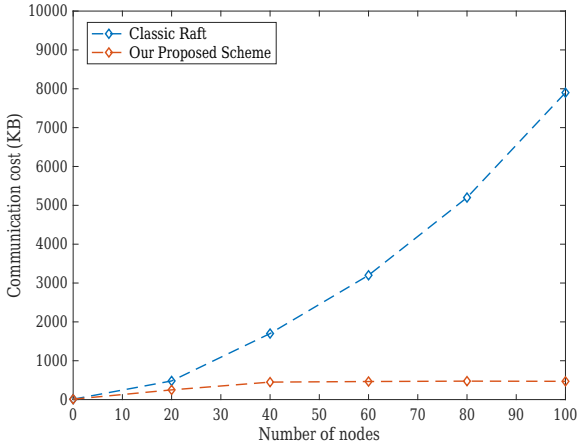


Fig. 3: Our proposed scheme communication costs vs. classical Raft communication costs.

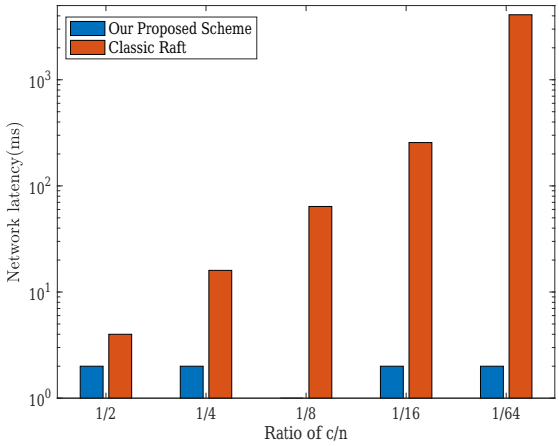


Fig. 5: Sensitivity analysis with different ratios of candidate nodes and total IoT nodes.

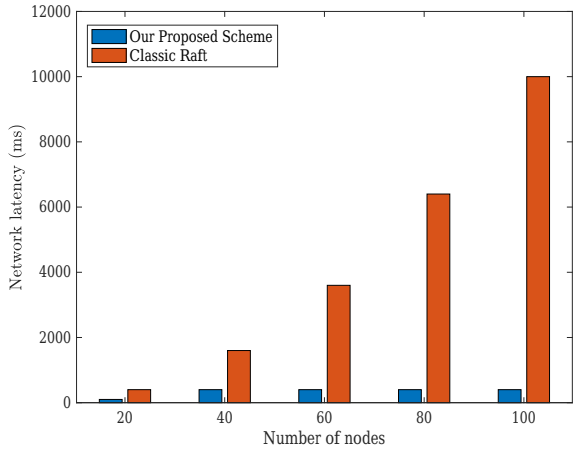


Fig. 4: Network latency in classical Raft vs. our propose scheme.

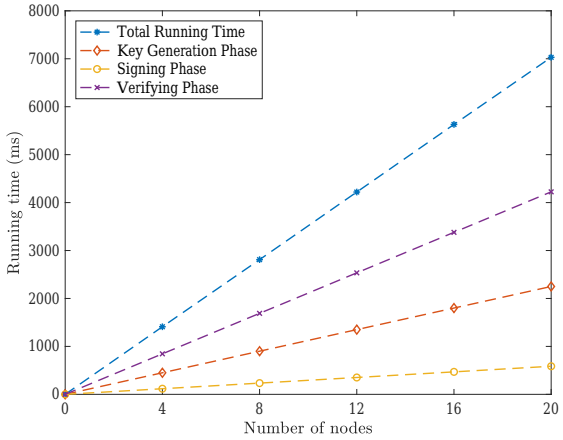


Fig. 6: Threshold signature scheme running time vs. number of consensus nodes.

than 100. The blue dashed line representing the classical Raft protocol has the 8000kb communication cost when nodes are 100. In contrast, our proposed new consensus protocol can reduce the communication cost to 5.07% as observed from Fig. 3.

### *B. Network Latency*

In this subsection, we compare the network latency results of our consensus protocol and classical Raft protocol. We pick one *Follower* node randomly per sub-layer blockchain. As shown in Fig. 4, our consensus protocol showed significant latency improvements over the classical Raft by increasing the number of nodes (e.g., *2.1x* network latency decrease for 20 nodes). Note that the maximum number of candidates group is 20 here. Compared to the classical Raft, our proposed approach selects a smaller number of *Candidate* nodes based on the geographic information. When nodes increase from 20 to 100, network latency increases exponentially in the classical Raft protocol.

By contrast, the new consensus protocol performs a better performance in terms of the network latency result. All qualified nodes can join the consensus committee when the number of nodes is smaller than the maximal threshold of candidate groups (i.e., 20). Consequently, when the number of candidate nodes grows from 1 to 20, the network latency increases, similar to the process in classical Raft protocol. However, once the number of candidate nodes reaches the threshold, no more new nodes can join the candidate group, and the network latency will not increase anymore.

### *C. Sensitivity Analysis*

In this subsection, we conduct a sensitivity analysis to evaluate the effects of changing the *c/n* ratio on the performance comparison between our proposed consensus scheme and classical Raft protocols. As stated in the previous section, our protocol significantly reduces the network latency and communication cost, especially when the ratio of *c/n* is greater.

To evaluate the reduction of network latency, we showed multiple groups of experiments with different ratios of *Can-*

*didate Nodes* to total IoT nodes. As shown in Fig. 5, our proposed consensus protocol showed significant performance improvements over the classical Raft by increasing the ratio of *c/n*. For instance, when the ratio of *c/n* grows from *1/2* to the *1/64*, the network latency of classical Raft protocol will grow exponentially in comparison with our protocol. In *1/64* case, the classical Raft protocol's network latency is beyond $10^3$ while the new consensus protocol's performance remains at $10^0 to\ 10^1$ level.

### D. Threshold Signature Scheme

Our threshold signature scheme is developed on the Hyperledger Ursa. We first instantiated three consensus nodes to form a lower-layer blockchain network instance. Its unique MAC address can identify each node. Then, we follow Algorithms 1, 2, and 3 in the construction of key generation, signing and verifying processes. In this example, three signatures are signed by the consensus nodes from the lower layer blockchain network and verified by the upper layer blockchain network. For each signature, the average time for key pair generation, signing, and verifying are 113 ms, 30 ms, and 215 ms, respectively.

We measure the performance of the threshold signature scheme by varying the number of consensus nodes from 4 to 8, 12, 16, and 20 in the lower layer blockchain network. As shown in Fig. 6, the total running time of each phase increases linearly with the increase in the number of consensus nodes in the lower layer blockchain network. Besides, the verifying phase takes additional time than the signing phase because the former requires the computation of pairings based on bilinearlity. In addition, our threshold signature scheme can offer constant running time for signing and verifying phases when varying the length of the identity information.

## V. Conclusion

Consensus algorithms are the defining mechanism behind the security and performance of blockchain networks. This paper proposed a hierarchical and location-aware consensus protocol for IoT-Blockchain applications. Compared to the original Raft protocol, our proposed consensus protocol is scalable by design and reaches consensus faster with lower network overhead and less communication cost. We prototype the experiments by utilizing the Hyperledger Ursa cryptography library to evaluate the threshold signature scheme. The results indicate that the architecture is scalable and suitable for large-scale IoT networks. For future work, we plan to investigate the cross-chain consensus among multiple blockchain systems.

## Acknowledgment

This work is partially supported by the Fundamental Research Funds for the Central Universities under the Grant G2021KY05101.

## References

[1] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.
[2] M. Isard, "Autopilot: automatic data center management," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 2, pp. 60–67, 2007.
[3] M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Proceedings of the 7th symposium on Operating systems design and implementation*, 2006, pp. 335–350.
[4] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-based blockchain authorization for iot," *IEEE Transactions on Network and Service Management*, 2021.
[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[6] Ethereum, "Blockchain Application Platform," https://ethereum.org.
[7] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
[8] W. Li, H. Guo, M. Nejad, and C.-C. Shen, "Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach," *IEEE Access*, vol. 8, pp. 181 733–181 743, 2020.
[9] W. Li, C. Meese, H. Guo, and M. Nejad, "Blockchain-enabled identity verification for safe ridesharing leveraging zero-knowledge proof," *in IEEE International Conference on Hot Information-Centric Networking (HotICN)*, 2020.
[10] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "Proof-of-event recording system for autonomous vehicles: A blockchain-based solution," *IEEE Access*, vol. 8, pp. 182 776–182 786, 2020.
[11] W. Li, C. Meese, Z. Zijia, H. Guo, and M. Nejad, "Location-aware verification for autonomous truck platooning based on blockchain and zero-knowledge proof," *in IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021.
[12] H. Guo, W. Li, E. Meamari, C.-C. Shen, and M. Nejad, "Attribute-based multi-signature and encryption for EHR management: A blockchain-based solution," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020.
[13] H. Guo, W. Li, M. Nejad, and C. Shen, "Access control for electronic health records with hybrid blockchain-edge architecture," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 44–51.
[14] W. Lu, Z. Ren, J. Xu, and S. Chen, "Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid," *IEEE Transactions on Network and Service Management*, 2021.
[15] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, 2020.
[16] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, 2014, pp. 305–319.
[17] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.
[18] Hyperledger Ursa, https://www.hyperledger.org/projects/ursa.
[19] Y. Yu, S. Liu, P. Yeoh, B. Vucetic, and Y. Li, "Layerchain: A hierarchical edge-cloud blockchain for large-scale low-delay iiot applications," *IEEE Transactions on Industrial Informatics*, 2020.
[20] I.-H. Chuang, S.-H. Chiang, W.-C. Chao, S.-H. Huang, B.-L. Zeng, and Y.-H. Kuo, "A hierarchical blockchain-based data service platform in mec environments," in *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology*, 2020, pp. 95–99.
[21] H. Yang, J. Yuan, H. Yao, Q. Yao, A. Yu, and J. Zhang, "Blockchain-based hierarchical trust networking for jointcloud," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1667–1677, 2019.
[22] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-pbft: a location-based and scalable consensus protocol for iot-blockchain applications," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2020, pp. 664–673.
[23] J. An, H. Yang, X. Gui, W. Zhang, R. Gui, and J. Kang, "Tcns: node selection with privacy protection in crowdsensing based on twice consensuses of blockchain," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 1255–1267, 2019.
[24] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978, no. 1, 2018, p. 012116.