Relayer-enabled Sharding Blockchain for Satellite Internet with High Concurrency

Hong Tang[1], Junqin Huang[1], Kai Liu[1], Linghe Kong[1], Guihai Chen[1], Shuangxi Cao[2], Chuyan Niu[34], Yufeng Wei[2]
[1]*Shanghai Jiao Tong University, China*
[2]*China Satellite Network Innovation Co., Ltd., China*
[3]*Shanghai Satellite Network Research Institute Co., Ltd., China*
[4]*Shanghai Key Laboratory of Satellite Network, China*
Email: {tanghong, junqin.huang, kai_liu, linghe.kong, chen-gh}@sjtu.edu.cn, {caoshx, niuchy, weiyf}@chinasatnet.com.cn

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437472

***Abstract*—Satellite internet (Sat-Net) enables high-speed connectivity with low latency and extensive coverage. However, it faces challenges related to data security and reliable networking. Blockchain technology, with its decentralized and tamper-proof nature, offers a promising solution to these challenges. However, applying blockchain to the Sat-Net topology is difficult due to its highly dynamic structure, which results in restricted communication periods between satellites and ground stations.. To address these issues and adapt to the Sat-Net topology, we propose a solution called RelSharding, which utilizes relayer satellites to relieve concurrent transmission pressure of ground stations by collecting and transmitting data transactions from client satellites. Through caching block headers and Merkle trees in relayer satellites, RelSharding can also reduce authentication latency between satellites utilizing simplified payment verification (SPV). Conducted on the modified SimBlock framework, our experiments indicate that RelSharding can enhance throughput up to 45x and decrease latency by 292x compared to existing blockchain solutions.**

***Index Terms*—Blockchain, satellite internet, sharding, concurrency, scalability.**

I. Introduction

Satellite internet, also known as Sat-Net, has become an essential technology that is revolutionizing global communication. This advanced system of internet connectivity relies on satellites orbiting the Earth to provide high-speed, reliable, and extensive access to the World Wide Web. Its significance lies in its ability to connect remote and underserved regions, overcoming the limitations of terrestrial infrastructure. For example, Starlink has provided Sat-Net access coverage to more than 53 countries, and plans to provide global mobile phone services after 2023 [1]. Sat-Net plays a critical role in promoting digital inclusivity and sustainable development worldwide by providing equal opportunities for access to information, education, healthcare, and economic growth.

Satellite systems offer many advantages for providing various services, but they also face several challenges. One major challenge is ensuring the security of these systems, as they are vulnerable to various security threats, including cyber-attacks, which can result in system paralysis or data breaches. In particular, inter-satellite links (ISLs) are more vulnerable than terrestrial links [2], making satellite systems susceptible to these kinds of attacks. The vulnerability of satellite systems to cyber-attacks is a pressing matter that needs to be addressed to ensure the reliability and security of satellite-based communication.

Blockchain technology has the potential to address various issues in the Sat-Net, as its decentralized and distributed nature can enhance the security of the system. Researchers have proposed several blockchain-based solutions to improve the efficiency and security of Sat-Net. For example, Fu et al. [3] proposed a resource management system based on blockchain to ensure throughput fairness and data security in ground-to-satellite links (GSLs). Liu et al. [4] designed a collaborative credential management strategy based on blockchain for use in space-air-ground integrated vehicular networks. Additionally, blockchain technology can be used to design secure communication protocols. Bao et al. [2] implemented two communication protocols using a consortium blockchain: Fulgor, a secure communication protocol between users and satellites, and Rayo, a privacy-preserving protocol for user-to-user communication.

Despite the potential benefits of blockchain technology in Sat-Net, the solutions mentioned above may not be sufficient to address the scalability problem as the number of satellites and ground stations (GS) increases. Sharding is a promising solution to scale blockchain [5]. It scales blockchain by splitting it into multiple partitions called shards. Each shard processes a portion of the network's transactions and maintains its own independent state. Note that a transaction in blockchain refers to the process of transferring value or data between participants within the network. Researchers have proposed the use of sharding techniques to improve blockchain scalability in Sat-Net. For example, Wang et al. [6] proposed a blockchain sharding scheme for satellite-based Internet of Things (S-IoT). However, this work mainly focuses on the efficiency of shard consensus, while ignoring the limitation of the actual GSL bandwidth, which lacks practical significance.

To make blockchain sharding more suitable for Sat-Net, we need to address two challenges. Firstly, as the number of satellites in the sharding-based blockchain grows, the pressure

* Linghe Kong is the corresponding author.

979-8-3503-1090-0/23/$31.00 © 2023 IEEE

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.

faced by each GS to process satellite requests concurrently also increases. While building more GSs is an intuitive solution, it can be costly. Therefore, designing a more efficient sharding protocol for concurrency is crucial. Secondly, a satellite cannot directly access GSs for the majority of its orbit period. During this period, the satellite must either relays its requests through ISLs, or uses a store-and-forward technique, where a satellite stores the data until a direct connection with a GS is established. Both methods cause high latency and affect the sharding-based blockchain performance. Addressing these challenges is essential to ensure the successful implementation of blockchain sharding in the Sat-Net.

To address these challenges, we propose a relayer-enabled sharding approach, namely RelSharding. A new role called the "relayer" is introduced to reduce the concurrency connections and the latency of data authentication. The relayer is a satellite that receives, packs, and transmits data transactions from satellites to ground stations. In the Sat-Net, only relayers can transmit data transactions through GSLs, reducing the number of concurrent GSL connections and the probability of collision. The relayers also provide fast data authentication for client satellites (call them clients for short) in space, which largely reduces the authentication latency between clients. By using relayers to manage the flow of data, RelSharding aims to improve the efficiency and performance of blockchain sharding in the Sat-Net. Our contributions are summarized as follows:

- We propose a scaling blockchain framework for Sat-Net that utilizes sharding. Taking into consideration the practical demands of Sat-Net, we introduce relayers to the blockchain system to improve throughput and reduce data authentication latency.
- To address security concerns related to the introduction of relayers, we propose a committee comprised of relayers. Within this committee, relayers must come to a consensus on proposed data transactions, thereby mitigating the effects of malicious relayers.
- We implement a simulation system for the proposed blockchain and conducted experiments to evaluate the scalability, overhead, and latency. Our experimental results indicate that RelSharding significantly enhances performance with increasing throughput up to 45x and reducing latency by 292x in comparison to existing solutions.

II. Related Work

Several recent studies [2]–[4] have concentrated on creating secure Sat-Net frameworks that incorporate blockchain technology. Although blockchain serves as the foundation for these frameworks, the size expansion of Sat-Net may result in scalability issues. To overcome this, researchers have proposed using a sharding approach to scale the IoT based Sat-Net [6]. Despite these advancements, two key challenges remain unaddressed: high concurrency and low latency.

High concurrency. GSs will face more pressure on concurrently processing requests through GSL, which is caused by the growth of satellites and the sharding deployment. Numerous works have sought to optimize the usage of the concurrency capability of satellite systems through routing algorithm design. Dong et al. [7] proposed a load balancing routing algorithm that combines congestion avoidance mechanisms and expands available paths to enhance the performance of Low Earth Orbit (LEO) networks. Lu et al. [8] introduced a distributed traffic balance routing protocol for LEO networks that addresses topological dynamics using a virtual topology model. Zhou et al. [9] presented an adaptive routing strategy for Sat-Net of Things based on an improved double Q-learning technique. By treating satellites and GSs as intelligent agents and optimizing various factors, the strategy outperforms existing methods in dynamic environments, improving delivery rate, average delay, and overhead ratio.

Low latency. Existing solutions include advanced routing algorithms and optimizing satellite constellations. Markovitz et al. [10] discussed LEO satellite constellations, such as SpaceX and OneWeb, focusing on the unique challenges of routing traffic and service planning due to their dynamic topology. The goal is to guarantee service metrics/QoS (bandwidth and latency) while managing satellite handovers in these rapidly evolving networks. Korcak et al. [11] proposed a link-layer handover algorithm, the Virtual Node Handover Algorithm (VN-HO), for earth-fixed LEO satellite systems using a Virtual Node approach. They also suggested increasing satellites per orbit for soft handover, resulting in a faster and smoother process with a minor cost increase. Roth et al. [12] introduced a geographical routing scheme for LEO constellations. In a system-level simulator, this approach decouples MAC and IP addresses for flexibility and considers satellite and terminal mobility.

The aforementioned solutions achieved good performance in their scenarios. However, they are not tested in sharding-based blockchain networks. In this paper, we propose a solution that combines both Sat-Net and blockchain. It optimizes network congestion by reducing the number of concurrent transmission connections and minimizes latency by shortening the path. Besides, it is orthogonal to these existing solutions, because it does not change the routing algorithm.

III. RelSharding Overview

A. Overview

Fig. 1 illustrates the identification of four distinct system roles in the sharding blockchain based Sat-Net: clients, relayers, GSs, and control centers (CCs). These nodes possesses a blockchain account and is assigned a shard ID or several shard IDs that represent the shards they belong to.

The division of a blockchain into shards is a technique known as sharding, which can improve scalability, and performance by allowing each shard to process a subset of data transactions. This means that nodes validate and store only partial data, which increases resource efficiency and enables higher transaction throughput. Blockchain participants are assigned specific shard IDs and are therefore responsible for processing transactions within their designated shard.

GSs and CCs serve as static infrastructures within Sat-Net, possessing superior computing power that enables them

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.

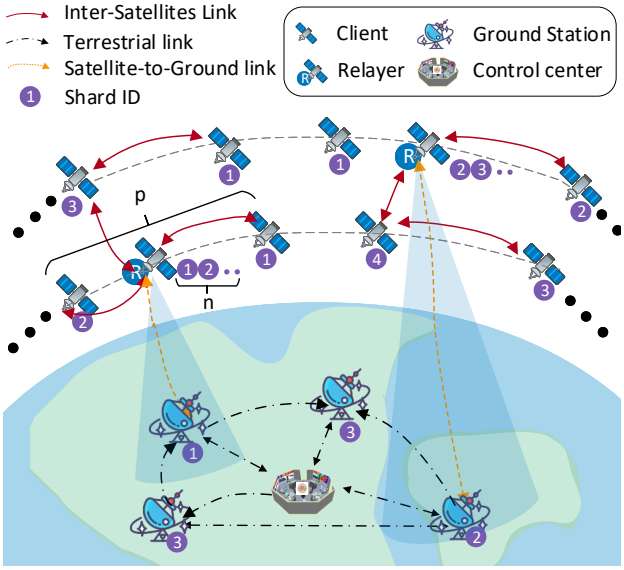

Fig. 1: Overview of the RelSharding.

to operate as full blockchain nodes. GSs function as ground gateways, responsible for engaging with satellites. Their tasks include collecting data transactions from satellites, processing and including them into blocks, synchronizing blockchain data with other GSs, sending transaction receipts to satellites, and synchronizing block headers and Merkle trees with relayers. Every GS belongs to one shard. CCs act as full blockchain nodes and manage network configuration, as depicted in Fig. 1. CCs do not belong to any shards and store all blockchain data.

Satellites with high mobility and limited computing resources serve as either clients or relayers. Clients only propose data transactions, and each client is assigned to a specific shard where its data resides. Relayers, on the other hand, are lightweight nodes that collect data transactions and provide authentication services for clients, while also transmitting transactions to GS nodes. They do not store the full blockchain data or participate in consensus, but instead authenticate transactions by caching block headers and Merkle trees. With the Simplified Payment Verification (SPV) technology, relayers can provide efficient transaction authentication for clients.

To improve performance, relayers store multiple shards so that they can relay and authenticate data transactions from multiple shards, as shown in Fig. 1. The satellites are arranged into orbital shells, maintaining a fixed relative position within each orbit. For every set of p satellites, one satellite is designated as a relayer, while the remaining satellites serve as clients. The relayers store the block headers and the Merkle tree of p shards, represented by the shard IDs.

B. Attack Model

The introduction of relayers in RelSharding has the consequence of increasing the attack surface. As a result, there is a possibility that some relayers could be compromised, which could lead to various types of attacks, including Denial of Service (DoS) attacks, transaction delays, and fraud attacks.

- **DoS attacks** occur when a malicious relayer either refuses to provide services to specific client satellites or disrupts their communication with the network, negatively affecting these satellites' regular data transactions.
- **Transaction delay attacks** arise from malicious relayers delaying the broadcast of data transactions, subsequently prolonging the transactions' confirmation time.
- **Fraud attacks** encompass malicious relayers forging data transaction receipts, causing satellites to mistakenly assume their data transactions have been confirmed in the blockchain and ultimately leading to fraudulent activities.

IV. RelSharding Design

A. Hybrid Sharding Initialization

In a sharding-based blockchain, every node has its shard ID. The shard ID distribution strategy should meet load balancing, decentralization, and scalability. In RelSharding, we design a hybrid sharding strategy by combining a hashed-based strategy and a geographical location-based one.

In the initialization phase, shard IDs are assigned to clients, relayers, and GSs. Other nodes can derive a relayer's shard IDs from its public key K_{pub}. Firstly, they calculate the hash value of K_{pub} with a hash function $H_m(\cdot)$ computed with an m-bit key, as shown in Eq. (1)

$$T_{pub} = H_m(K_{pub}). \tag{1}$$

Then, they divide T_{pub} into n equal-sized groups bitwise, represented as $\{g_1, g_2, \ldots, g_n\}$. Each group has $\lfloor m/n \rfloor$ bits. An example is shown in Eq. (2)

$$T_{pub} = 0\text{x}\underbrace{\text{AA...AA}}_{g_1}\underbrace{\text{AA...AA}}_{g_2}\ldots\underbrace{\text{AA...AA}}_{g_n}. \tag{2}$$

Finally, nodes perform modulo N operations and add the result by 1 on all groups to obtain shard IDs $\{S_1, S_2, \ldots, S_n\}$ from 1 to n, as shown in Eq. (3)

$$S_i = (g_i \bmod N) + 1, \quad 1 \le i \le n, \quad i \in \mathbb{N}. \tag{3}$$

The number of a relayer's shard IDs is less than or equal to n, as the shard IDs may be duplicated.

A client's shard ID is assigned following a similar pattern to a relayer's. The difference is that when calculating the client's shard ID, there is no step of splitting its hash value T into n groups. The shard IDs distribution of clients and relayers is based on a hashed-based strategy, which meets the requirement of load balancing, decentralization, and scalability.

Different from satellites, the number of GSs is smaller than them. For example, in the Starlink project, there are currently 147 operational GSs [13]. A hashed-based strategy may cause an uneven distribution of shard IDs.Therefore, the GSs' shard IDs are assigned according to their geographical locations and ensure that there are different shard IDs in every region and the number of each shard ID is close. Due to the small number of GSs and their high cost, manually assigning IDs has a negligible impact on scalability.

After assigning the shard IDs, the relayers synchronize block headers and Merkle trees from the GSs maintaining the

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.

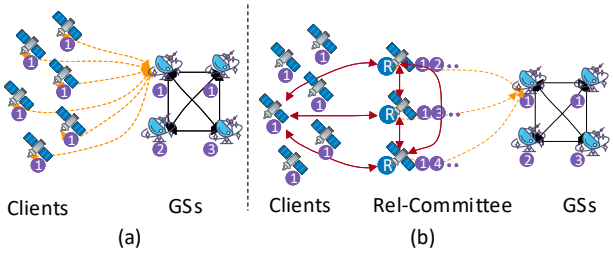

Fig. 2: Data flow comparison of transaction transmission. (a) w/o Rel-Committee; (b) with Rel-Committee.

targeted shard data. The relayers operate within their respective orbits, acquiring data transactions from GSs that include the targeted shard IDs. In the event of a match between shard IDs, the relayer initiates a synchronization request.

B. Relayer-enabled Data Transaction Transmission

A data transaction, initially proposed by a client, is ultimately processed by GSs. As the number of satellites expands, the volume of transactions correspondingly escalates, thereby exerting increased pressure on GSs. Traditionally, as illustrated in Fig. 2(a), a client proposes a data transaction to the blockchain following a routing algorithm to send it to the GSs. However, as the number of clients increases, the resource race in GSLs intensifies, because more clients attempt to submit transactions to GSs simultaneously. Consequently, designing an high concurrency scheme is of paramount importance.

In RelSharding, as depicted in Fig. 2(b), we introduce a relayer committee (Rel-Committee) to relay data transactions to GSs, aiming to reduce concurrent connections and resource race in GSLs. The process of data transaction submission is illustrated in Fig. 3 and includes three steps: 1) a client proposes a data transaction to a Rel-Committee; 2) the Rel-Committee processes, signs, and categorizes the data transaction; 3) the committee members transmit the data transactions to GSs.

Data transaction proposal. A client generates a data transaction containing source shard ID S^s and target shard ID S^t, source shard block number, digital signature, body, committee member list, and other custom fields. The source shard block number is used to extract the timestamp of the current block in the source shard. The body comprises serialized bytes representing the data transaction content. The client signs the data transaction with its private key to generate a digital signature. The client randomly selects Rel-Committee members from the active relayers which contain both the S^s and S^t. If $S^s = S^t$, it indicates that this transaction is not a cross-shard transaction. Note that explicitly specifying the relayers containing S^s and S^t can simplify the routing problem. If we do not tie the relayers and GSs with specific shard IDs, the more complicated routing mechanism is required for relayers to choose the GSs.

Data transaction pre-processing. In a sharding-based blockchain, it is important to ensure the atomicity of cross-shard data transactions. In RelSharding, we design a two-phase commitment (2PC) process based on Rel-Committee, as shown in Fig. 3 and Fig. 2(b). Assume that there is a data

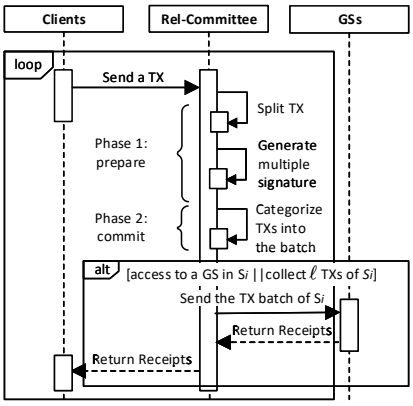

Fig. 3: Workflow of relayer-enabled transaction submission.

transaction TX. The first phase is the prepare phase, where the Rel-Committee members split the TX and then generate multiple signature (Multisig) among them. It is important to reach a consensus among relayers about the processing of a TX. The second phase is the commit phase, where the relayers categorize TX into batches.

In the first phase, relayers receive TX and extract its source shard ID S^s and target shard ID S^t. If $S^s \neq S^t$, TX is a cross-shard data transaction, and the relayers split it into multiple steps. For instance, TX transfers α units of data asset from client C_a in shard A to client C_b in shard B. TX is split into reducing C_a's balance by α as TX_1 and increasing C_b's balance by α as TX_2. After processing TX, committee members synchronize with each other to create a Multisig requiring at least k out of O members to sign the TX_1 and TX_2.

In the second phase, relayers add TX to the body of TX_1, and add TX_1 to the data transaction batch related to shard A. TX_2 is processed in the same way. Note that TX is included in the body for the convenience of recovering TX's receipt.

If TX is not a cross-shard data transaction, it does not require a division. The Rel-Committee members generate a Multisig on TX and add it to the data transaction batch related to TX's target shard.

Data transaction transmission. A relayer starts transmitting a data transaction batch in two cases, as shown in Fig. 3. The first case occurs when a relayer accumulates ℓ data transactions for one shard, where ℓ represents the maximum number of data transactions within a single batch. The relayer then transmits the data transaction batch through the ISL until finding a relayer with direct access to a target GS and enough bandwidth to transmit the data transaction batch. The second case arises when a relayer traverses past a GS. If the GS's shard ID matches a data transaction batch, the relayer transmits this batch.

Data transaction processing on GSs. GSs receive data transaction batches from relayers. Initially, GSs unpack it, extract the target shard ID of the unpacked data transactions, and check if the IDs match the GSs. Subsequently, GSs check whether the data transactions have been included in blocks or not. If not, GSs verify every data transaction's signature

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.

and Multisig, and then process it in a manner consistent with traditional blockchains. Finally, data transaction receipts are returned to the respective relayers.

C. Low-latency Data Transaction Authentication

In the Sat-Net, clients from different shards may exchange data assets. Traditionally, clients have to send an authentication request to GSs for proving the existence of assets, resulting in high latency. In RelSharding, with relayers, the latency can be significantly reduced.

Assume that in shard A, a client C_a wants to make an authentication about data transaction TX, which is initiated sent by the client C_b in shard B. C_a computes the hash value $T = H(TX)$. and send it to a relayer R_a containing block headers and Merkle trees of shards A and B. According to T, R_a sends Merkle proof $P = [P_1, P_2, \ldots, P_n]$, where P_1 is the leaf node, and P_n is the child node of the Merkle root γ. T and P_1 are sibling nodes. The hash value of their combination is calculated according to the given formula. If T_i is in the left of P_i, then

$$T_i = \begin{cases} T, & i = 0, \\ H(T_{i-1}||P_i), & 1 \leq i \leq n. \end{cases} \tag{4}$$

If P_i is in the left of T, then

$$T_i = \begin{cases} T, & i = 0, \\ H(P_i||T_{i-1}), & 1 \leq i \leq n. \end{cases} \tag{5}$$

Finally, compare T_n and γ. If they are the same, it indicates the data is on chain. In this way, the existence of data can be proved without going through GSs, thus greatly reducing the verification latency.

V. Evaluation and Analysis

We implemented the RelSharding protocol using SimBlock [14], a blockchain network simulator. SimBlock enables straightforward node behavior modification and investigation of their effects on blockchains, allowing for practical experimentation on Sat-Net and the impact of relay networks on transactions propagation time. We modified the code to support a sharding protocol and further implemented the RelSharding based on it.

We evaluated the RelSharding protocol in terms of its throughput, overhead introduced by the Rel-Committee, and the improvement in latency reduction when authenticating data. Since the capabilities of GSLs are greatly smaller than ISLs [15], we assume that the GSLs are fully occupied. In other words, the concurrency capacity of the blockchain network for each GS reaches its upper limit. The number of satellites is 1,000, with one relayer of every 5 satellites.

A. RelSharding Concurrency Evaluation

The concurrency capacity is exemplified by throughput, while scalability can be assessed by varying the quantity of nodes within the system. In the throughput evaluation, we consider two benchmarks. The first benchmark involves a blockchain utilizing Proof of Stake [16] (PoS) as the consensus algorithm, which is widely used in both public and consortium

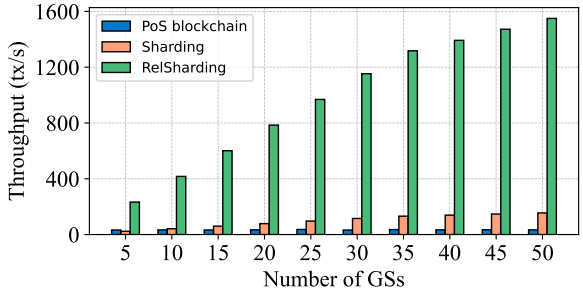

Fig. 4: Throughput comparison between the PoS blockchain, Sharding, and RelSharding.

blockchains. We assume that every GS stakes the same token. In other words, they have the same probability to propose blocks. The second benchmark applies a straightforward sharding approach to the blockchain of the first benchmark, with GSs uniformly distributed and the number of consensus nodes in each shard remaining constant. The consensus algorithm employed in every shard is also PoS. And we apply the RelSharding protocol based on the second benchmark.

The number of GSs ranges from 5 to 50, while the number of nodes in each shard is 5. We set the number of relayers in a committee to 1, aiming to show its maximum concurrency capability. A data transaction batch in relayers contains 10 data transactions.

The experimental result is shown in Fig. 4. As the number of GSs increase, the throughput of PoS blockchain does not scale, while the throughput of Sharding and RelSharding scale. It is because sharding-based blockchain divides the GSs into multiple shards to process data transactions in parallel. Comparing Sharding and RelSharding, the latter one has higher throughput. The reason is that the data transactions are delegated to relayers, thereby reducing the concurrent connections and resource race in GSLs. Besides, a relayer sends more data transactions (in batch) once time than a client. When the number of GSs is 50, the throughput of RelSharding is about 45x larger than PoS blockchain.

B. Rel-Committee Security Overhead Evaluation

The primary performance loss of RelSharding arises from the duplicated transmissions from relayer committee members. We find that when the number of committee members exceeds one, each GS will receive duplicate data transactions, which decreases the effective utilization of GSLs. Thus, we evaluated the impact of the committee size in this section. The configuration consists of 50 GSs and 10 shards. The number of relayers in a committee ranges from 1 to 6.

The experimental results are presented in Fig. 5(a). We observe a decrease in throughput as the number of committee members increases. The throughput is inversely proportional to the number of committee members. With one committee member, the throughput is approximately 1549.67 tx/s, while with six members, it decreases to 258.28 tx/s, roughly one-sixth of the initial throughput. The reason is that when the number of committee members is larger, there are more duplicated transactions in GSLs. The system's security is positively

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.

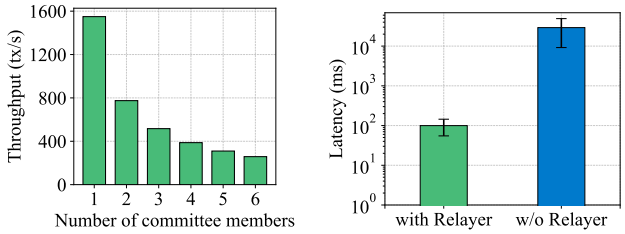

Fig. 5: (a) Throughput of RelSharding with different numbers of committee members. (b) Data authentication latency.

correlated with the number of committee members. It is a trade-off between performance and security, thus selecting an appropriate number based on the specific scenario is crucial.

C. Data Authentication Latency Evaluation

Relayers, who store block headers and Merkle trees, allow client satellites to authenticate data integrity via ISLs. Without relayers, clients must send data authentication requests through both ISLs and GSLs to GSs. A comparison of authentication latency for these two cases is shown in Fig. 5(b).

In a Sat-Net of 1,000 satellites, one relayer is assigned for five satellites. Experiment is conducted under the worst-case assumption, where each relayer stores Merkle tree and block headers for one shard. From Fig. 5(b), we can observe that without a relayer, the average latency is 29202.056 ms, while with a relayer, the average latency is 99.645 ms, resulting in $\sim$292x reduction in latency. These results demonstrate that the integration of relayers can considerably reduce latency, as it significantly shortens the communication distance.

D. Security Analysis

The Rel-Committee design enhances system security by mitigating various attacks, when assuming the majority of relayers are honest. The risk of *DoS attacks* is significantly reduced due to service redundancy provided by multiple relayers. Without a Rel-Committee, a client sends its data transaction to an active relayer at random. If this relayer suffers from a DoS attack, the client's data transaction may fail. However, when a client send its data transaction to multiple relayers in a committee, even if some are affected by DoS attacks, others can process the data transaction normally. *Transaction delay attacks* and *fraud attacks* can also be mitigated in a similar way. Transactions are considered valid only if signed by a sufficient number of committee members. If some relayers attempt to delay a data transaction or initiate a fraud attack, the signatures of other relayers remain valid, ensuring that the data transaction is included in the blockchain or preventing invalid data transactions from being included.

VI. Conclusion

In this paper, we introduce RelSharding, a relayer-enabled sharding blockchain for Sat-Net, designed to address scalability, concurrency, and latency challenges as the number of satellites grows. Since the RelSharding protocol offloads data transaction transmission from client satellites to these relayers, it largely reduces concurrent connections in GSLs and data authentication latency. Evaluations and analysis demonstrate the enhanced throughput and shorten latency provided by RelSharding, with manageable and tunable security overhead. In the future, we further explore the optimal proportion of relayers in satellites and the size of Rel-Committee to balance the performance and security.

Acknowledgment

Supported by Shanghai Key Laboratory of Satellite Network, the open project of Satellite Internet Key Laboratory in 2022 (Project 6: Research on Distributed Trusted Communication Networking Technology of Blockchain for Satellite Internet). This work was supported in part by NSFC grant 62141220, 61972253, U1908212.

References

[1] Wikipedia. Starlink. https://en.wikipedia.org/wiki/Starlink, 2023. Accessed: 2023-05-04.
[2] Zijian Bao, Min Luo, Huaqun Wang, Kim-Kwang Raymond Choo, and Debiao He. Blockchain-based secure communication for space information networks. *IEEE Network*, 35(4):50–57, 2021.
[3] Shu Fu, Jie Gao, and Lian Zhao. Integrated resource management for terrestrial-satellite systems. *IEEE Transactions on Vehicular Technology*, 69(3):3256–3266, 2020.
[4] Dongxiao Liu, Huaqing Wu, Cheng Huang, Jianbing Ni, and Xuemin Shen. Blockchain-based credential management for anonymous authentication in SAGVN. *IEEE Journal on Selected Areas in Communications*, 40(10):3104–3116, 2022.
[5] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *ACM CCS*, pages 931–948, 2018.
[6] Bingzheng Wang, Jian Jiao, Shaohua Wu, Rongxing Lu, and Qinyu Zhang. Age-critical and secure blockchain sharding scheme for satellite-based internet of things. *IEEE Transactions on Wireless Communications*, 21(11):9432–9446, 2022.
[7] Chaoying Dong, Xin Xu, Aijun Liu, and Xiaohu Liang. Load balancing routing algorithm based on extended link states in LEO constellation network. *China Communications*, 19(2):247–260, 2022.
[8] Yong Lu, Fuchun Sun, Youjian Zhao, Hongbo Li, and Heyu Liu. Distributed traffic balancing routing for LEO satellite networks. *International Journal of Computer Network and Information Security*, 6(1):19–25, 2014.
[9] Jian Zhou, Xiaotian Gong, Lijuan Sun, Yong Xie, and Xiaoyong Yan. Adaptive routing strategy based on improved double Q-learning for satellite internet of things. *Security and Communication Networks*, 2021:1–11, 2021.
[10] Oren Markovitz and Michael Segal. Advanced routing algorithms for low orbit satellite constellations. In *IEEE ICC*, pages 1–6, 2021.
[11] Ömer Korçak and Fatih Alagöz. Link-layer handover in earth-fixed LEO satellite systems. In *IEEE ICC*, pages 1–5, 2009.
[12] Manuel Roth, Hartmut Brandt, and Hermann Bischl. Implementation of a geographical routing scheme for low earth orbiting satellite constellations using intersatellite links. *International Journal of Satellite Communications and Networking*, 39(1):92–107, 2021.
[13] Starlink. Starlink ground station locations: An overview. https://starlinkinsider.com/starlink-gateway-locations/, 2023. Accessed: 2023-04-14.
[14] Yusuke Aoki, Kai Otsuki, Takeshi Kaneko, Ryohei Banno, and Kazuyuki Shudo. Simblock: A blockchain network simulator. In *IEEE INFOCOM WKSHPS*, pages 325–329, 2019.
[15] Weisen Liu, Qian Wu, Zeqi Lai, Hewu Li, Yuanjie Li, and Jun Liu. Enabling ubiquitous and efficient data delivery by LEO satellites and ground station networks. In *IEEE GLOBECOM*, pages 687–692, 2022.
[16] Peter Gaži, Aggelos Kiayias, and Dionysis Zindros. Proof-of-stake sidechains. In *IEEE S&P*, pages 139–156, 2019.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 01,2024 at 16:30:57 UTC from IEEE Xplore. Restrictions apply.