Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.021202
Article

Tech Science Press

Domain Name Service Mechanism Based on Master-Slave Chain

Siyuan Liu[1], Shaoyong Guo[1,*], Ziwei Hu[2], Xin Xu[3], Wei Bai[2], Ningzhe Xing[4], Xuesong Qiu[1] and Siwen Xu[5]

[1]Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]Global Energy Interconnection Research Institute Co., Ltd., Beijing, 102209, China
[3]State Grid Chongqing Electric Power Co. Electric Power Research Institute, Chongqing, 401123, China
[4]State Grid Jibei Information & Tzelecommunication Company, Beijing, 100053, China
[5]Université Paul Sabatier-Toulouse 3[5], Toulouse, 31000, France
*Corresponding Author: Shaoyong Guo. Email: syguo@bupt.edu.cn

Received: 25 June 2021; Accepted: 26 July 2021

Abstract: Although the current Domain Name System (DNS) has been able to satisfy the use of network services, there are still many challenges in the future development of the Internet. The centralized management of traditional domain name management systems has many risks, and cannot defend against Distributed Denial of Service (DDoS) attacks and single points of failure. As a decentralized tool, blockchain provides innovative ideas for the improvement of domain name management systems. Starting from the existing network resolution system and combining the application of cross-chain communication in DNS, this paper proposes a domain name resolution service architecture model based on the master-slave chain, adopts a Multi-Sig Notary Schemes to achieve cross-chain communication, and proposes based on the master-slave chain consensus algorithm (MSBFT) of the PBFT algorithm. In a single blockchain, this paper uses the VRF algorithm to select the master node. Finally, the simulation experiment designed in this paper tests the throughput and analysis delay of the analytical model, compares the analytical delay of the main chain and slave chain, and analyzes the advantages of the model.

Keywords: DNS; blockchain; cross-chain; master-slave chain

1 Introduction

The Domain Name System (DNS) is the most core part of the Internet. Its main function is to complete the conversion service of domain names to IP addresses for network users all over the world. Although the current DNS system has been able to meet the needs of the network, there are still some potential problems in the future Internet. Domain name is managed in a tree-shaped centralized system. The structure and service of DNS still have a high degree of centralization. This centralized management mode brings potential hidden dangers and risks to the stable operation of the system [1]. Moreover, the DNS adopts a centralized management model, which gives some countries an advantage in the management and supervision of the

DNS system. With the rapid development of the Internet, traditional DNS are facing great challenges in performance and security [2].

Blockchain technology provides a new solution for the DNS system. Blockchain has the characteristics of decentralization, transparency, immutability and traceability. It can solve the problem of insecurity of data stored on a single node in a centralized system and the high cost of single node performance maintenance. The data stored in the blockchain system can be traced back [3]. Every complete node in the network has a complete backup of transactions with timestamps. The transactions in the network must achieve consensus, so the data of the blockchain system is cannot be tampered with. As a special private chain, the consortium chain is only open to specific organizations. The participating nodes in the consortium chain first need to register for permission. The alliance chain is limited to the participation of specific members on the chain. According to the consensus mechanism of the alliance chain, the blockchain has the authority to query the data on the chain and the right to participate in the blockchain [4].

The cross-chain technology of the blockchain makes the blockchain expandable and realizes data communication and asset transfer between different blockchains. The current mainstream cross-chain technologies mainly include: Sidechains/Relays, Hash-locking, Distributed Private Key Control, and Notary Schemes [5]. Sidechains/relays uses SPV (Simplified Payment Verification) proof, relay chain and others to realize the credible intercommunication between different blockchains. Hash-locking uses Hash Lock and Time Lock to complete the exchange of cross-chain assets. Distributed private key control uses distributed key generation algorithms to enable assets to be kept in the cross-chain process. The notary schemes are to elect one or a group of trusted nodes as notaries to verify specific events on a specific chain and provide proof to another chain [6]. When the relationship between two chains in a cross-chain is not a peer-to-peer relationship, the notary schemes can be used. The notary schemes are divided into Centralized Notary Schemes and Multi-Sig Notary Schemes. Moreover, Multi-sig Notary Schemes can avoid Single Point of Failure (SPOF) [7] of DNS system.

Blockchain technology can solve the problem of the high degree of centralization of DNS, and the consortium chain architecture based on the master-slave chain can solve the limitation and efficiency of the single chain operation in the consortium chain [8]. In order to face the challenges of future network systems to the existing DNS system, the security of the domain name resolution system is enhanced. Therefore, this article proposes a new DNS model based on the master-slave chain.

2 Related Work

2.1 Overview of Application of Blockchain in DNS

Namecoin [9] is the first distributed DNS system based on blockchain technology. Its characteristic is that it does not require network censorship and can guarantee the free release of information. The purpose of its design is to realize the registration, management and resolution of domain names as a public chain that can replace the DNS system. Namecoin adheres to the idea of decentralization, but the security of the blockchain is low, because the network can accommodate fewer nodes and is vulnerable to attacks.

Blockstack [10] overcomes the vulnerable characteristics of Namecoin, separates the control layer and data storage layer in the system, uses the underlying blockchain to process domain name data, and the domain name information is stored in the system's database. New functions are designed on the basis of the blockchain to realize basic functions such as domain name registration, domain name update and ownership transfer. Blockstack's use of public chains is not conducive to data supervision in the DNS system and on-chain member authority management, and it is more difficult to replace the existing DNS system.

Li et al. [11] proposed a new type of multi-modal domain name management technology framework, which integrates the voting management of changeable co-management and the mutual access of identity domain names and mutual tunnel functions under multi-modal networks. The paper adopts the Consortium Chain PoV consensus mechanism to realize the basic functions of DNS system, such as domain name access and domain name registration and resolution. However, the number of nodes that can be accommodated in the network is limited.

2.2 Overview of Application of Cross-chain Technology in DNS

DNSTSM proposed a multi-channel algorithm for DNS domain name caching resources, and designed a method to calculate the credibility of DNS resolution nodes to evaluate malicious nodes to avoid affecting the system and reduce the service delay of the system [12]. However, DNSTSM did not solve the problem of low service speed caused by multiple nodes in the alliance chain.

Duan et al. [13] proposed a domain name resolution system model based on slave chains, provided a complete DNS system structure, designed a new domain name storage method and realized a complete solution for domain name application, modification and resolution services. However, DNSLedger uses a local database method, which cannot guarantee the uniqueness and legality of the domain name.

In short, the above scheme is the innovation and application of blockchain technology in the domain name resolution scenario, and proposes new ideas and innovations to solve the decentralization problem of DNS, but they do not solve the centralization of DNS domain name resolution Existing problems. This article is based on the domain name resolution service mechanism of the master-slave chain, which can satisfy multiple participants in the DNS system to jointly manage DNS system.

3 Domain Name Service Mode Based on Master-Slave Chain

3.1 Overall Framework

The traditional DNS is a hierarchical tree system [14]. The top layer of the system is the Root DNS Server, and the second layer is the Top-level DNS Server, including gTLDs (generic top-level domains) and ccTLDs(country code top-level domains). The next level is the second-level domain name and third-level domain name server corresponding to each top-level domain name. Its architecture is shown in Fig. 1.

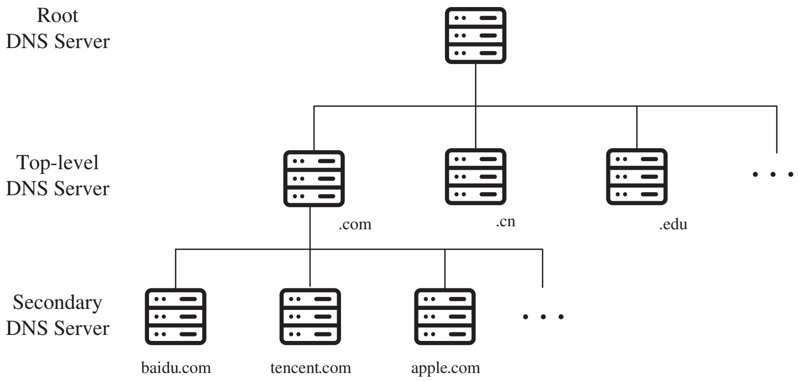

Figure 1: Traditional DNS architecture

This article proposes a domain name resolution service mechanism based on the master-slave chain. There is no centralized domain name manager node in the system, but a peer-to-peer network formed by the participants in the system to maintain the domain name data in the network, using consensus The mechanism serves as the domain name data storage specification for system participants, so that the domain name data in the system remains consistent.

The DNS architecture model based on the master-slave chain is shown in Fig. 2. The system consists of at least two chains. The DNS architecture is mainly divided into two layers. The first layer is the main chain of the alliance composed of top-level domain name servers. There is only one master chain. The function of the master chain is the same as that of the root server in the original DNS system; The second layer is the slave chain constructed by the top-level domain name based on the actual needs of the region. In each chain, the nodes on the chain participate in the accounting of information and transactions in their respective domains. Therefore, different consensus algorithms can be used in different slave chains.

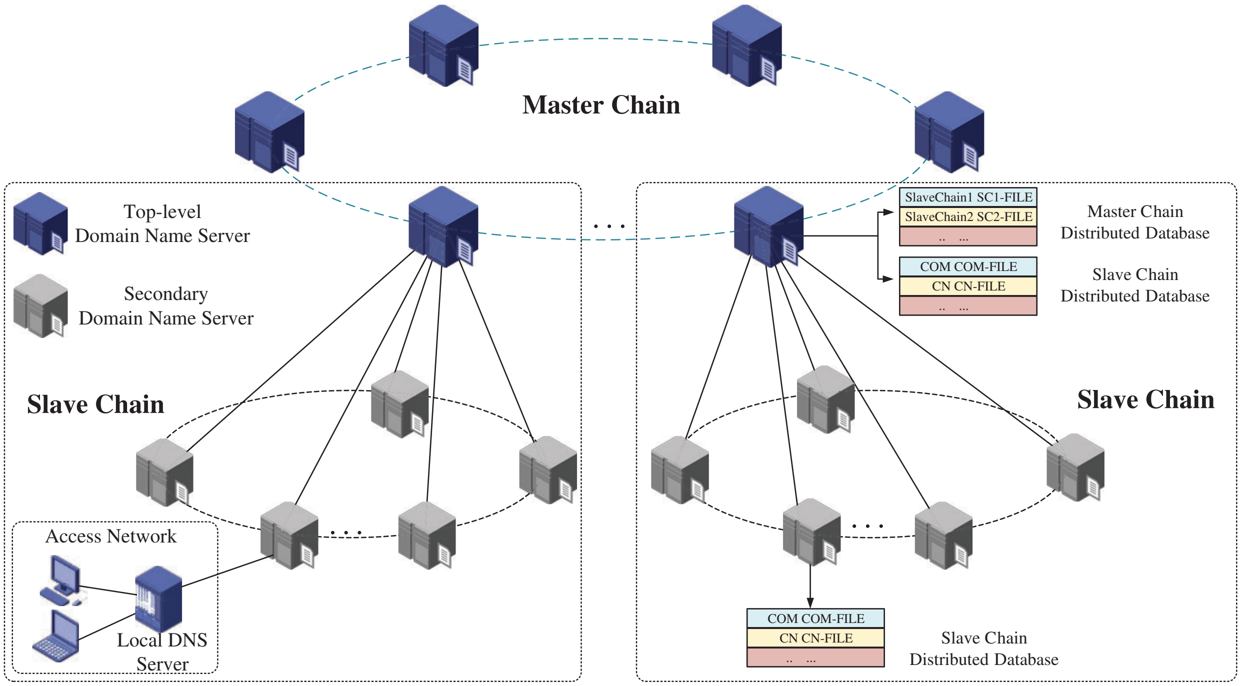

Figure 2: DNS architecture model based on the master-slave chain

This article divides network nodes into three types: **Master Chain Node**, **Gateway Node** and **Slave Chain node**. Fig. 3 shows the role of nodes in the network. The nodes in the main chain are composed of top-level domain name servers, and the slave chain nodes are composed of secondary domain name servers except for gateway nodes.

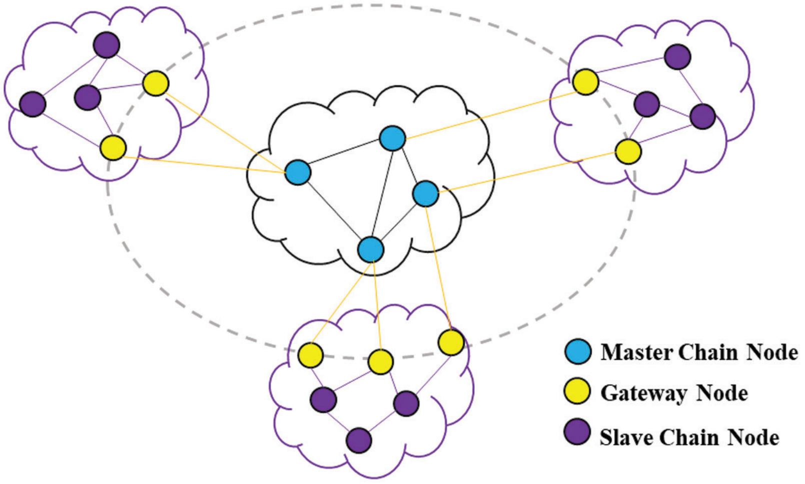

Figure 3: The role of nodes in the network

Master Chain Node: As a member node on the master chain. Master chain node completes the accounting and confirmation of the domain name transaction information of the master chain.

Gateway Node: As a trusted entity in the network, Gateway Node belongs to the master chain and the slave chain at the same time, participates in the transaction in the slave chain as a notary node, and submits the transaction to the master chain after signing.

Slave Chain node: As a member node in the slave chain, it participates in the accounting and confirmation of domain name transactions in the slave chain.

In the cross-chain communication of the system, this article adopts the Multi-sig Notary Schemes [15], which can complete the communication between the master chain and the slave chain and avoid the single point of failure caused by a single notary. The top-level domain name server in the master chain serves as the gateway node of the network, and the domain name transaction information completed in the slave chain is submitted to the master chain after being signed by the gateway center. The master chain has multiple slave chain domain name transactions at the same time. The master chain reviews and confirms the transactions in the slave chain. In slave chain, there are generally multiple gateway nodes. In a slave chain, there are generally multiple gateway nodes. The system fulfills the function of the DNS system through this multi-party co-management method.

3.2 Consensus Mechanism

3.2.1 Master Node Selection Based on VRF Algorithm

The three kinds of nodes mentioned above need to select the master node, and each node extracts the master node by executing the VRF algorithm [16]. The full name of the VRF algorithm is Verifiable Random Functions. The VRF algorithm has three characteristics: verifiable, random, and function group. The VRF algorithm mainly contains three functions: G, F, V algorithm:

G is a probabilistic generating function:

$$(PK, SK) = G(x) \tag{1}$$

x is the safety factor of the input, and output is *PK* (public key) and the *SK* (private key);

$F = (F_1, F_2)$ is a deterministic algorithm:

$$value = F_1(s, SK), proof = F_2(s, SK) \tag{2}$$

value is the random number calculated, *proof* is the proof of the random number, and *s* is the seed parameter input in the function;

V is a probabilistic verification function:

$$YES/NO = V(PK, s, value, proof) \tag{3}$$

It can verify the authenticity of *value* through *PK*, *s* and *proof*.

The master node selection process based on the VRF algorithm designed in this paper is as follows:

1. Each node in obtains a random value through the master node extraction algorithm:

$$(proof, value) = VRF\_Value(s, SK) \tag{4}$$

s is the block height; *SK* is the node's private key; *proof* is the proof of the random value, *value* is the random value output by the VRF.

2. After the node draws the lottery, it broadcasts its own lottery result and public key. The broadcast content is:

$$(proof, value, PK) \tag{5}$$

3. After each node receives the broadcast information of other nodes, it verifies through the verification function:

$$YES/NO = VRF\_Verify(PK, s, proof, value) \tag{6}$$

sorts and compares the random values, elects the node with the largest random value as the master node, and broadcasts the node public key.

4. After the master node receives the confirmation information of 2/3 of the total number of nodes, it determines itself as the master node.

3.2.2 Master-Slave Chain Consensus Mechanism Based on PBFT Algorithm

This article proposes a consensus mechanism based on the slave chain. The whole process includes four processes: transaction request, slave chain consensus, master chain consensus, and transaction response [17]. The flow chart of the consensus algorithm is shown in Fig. 4.

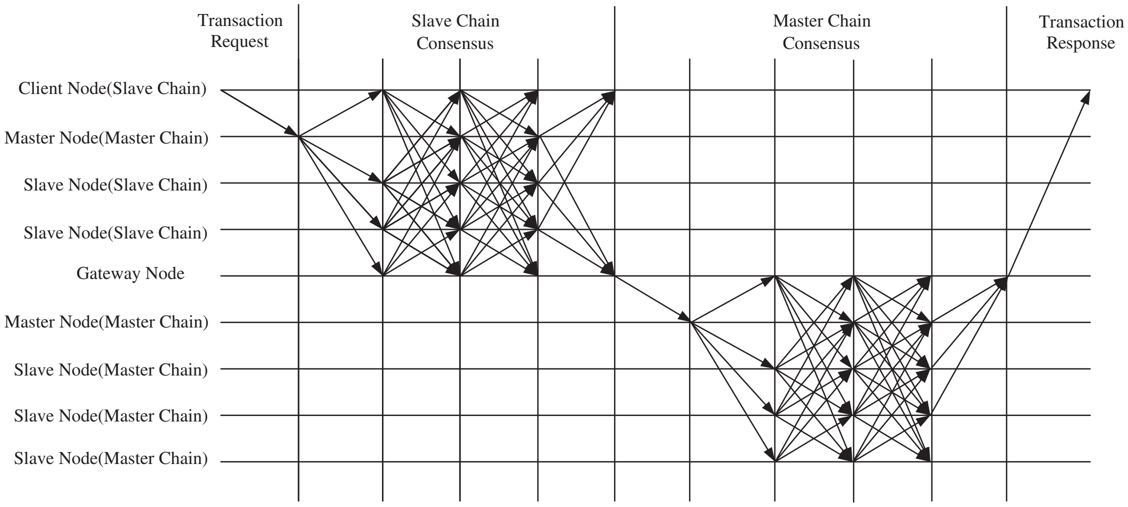

Figure 4: Master-slave chain consensus mechanism based on PBFT algorithm

1. Transaction request

The transaction request is completed in the slave chain, and the client sends the transaction request to the master node of the slave chain.

2. Slave chain consensus

Pre-preparation stage: After the master node receives the client's request transaction, it verifies the transaction. Illegal transactions will be discarded, and legitimate transactions will be signed and broadcast. Preparation stage: The slave node verifies the broadcast of the master node. Confirmation stage: The client needs to get double confirmation from the master node and the gateway node. After the master node receives the verification information from the slave node, it confirms and responds to the client; after the gateway node receives the verification from the slave node, it signs the transaction and sends the transaction Submit the master chain and request transactions in the master chain.

3. Master chain consensus

Pre-preparation stage: Similar to the slave chain, the gateway node submits the transaction to the master node, and the master node verifies the legality of the transaction and broadcasts the legal transaction. Preparation stage: The slave node verifies the transaction after receiving the broadcast from the master node, and broadcasts it in the master chain after verification. Confirmation stage: The master node saves the transaction after receiving the verification broadcast from the slave node.

4. Transaction response

Master chain response: After receiving the confirmation message, the master node responds to the gateway node with a successful transaction message. Slave chain response: After the gateway node receives the confirmation message from the master chain node, the gateway node responds to the client. If the main chain transaction fails, re-initiate the failed request and complete the cancellation of the slave chain transaction. After the client receives the confirmation message from the gateway node, the transaction is completed.

3.3 DNSMSC Operation Process

3.3.1 Domain Name Registration

Domain name registration is initiated in the slave chain. Since domain names are divided according to regions, the same domain name cannot be registered on multiple slave chains through the access of the master node to the gateway node. The domain name registration process is as follows:

1. Identity registration: The client or enterprise first applies for domain name registration. In the network, a domain name resource that can be accessed by this query needs to apply for registration with the master node in the slave chain. After the master node and gateway node in the slave chain confirm, the domain name can be accessed normally in the network.

2. Blockchain node authentication: After receiving the domain name registration request from the client, the nodes in the alliance chain review its content and confirm the registration of the domain name through a consensus algorithm. Return to the registered client after confirmation.

3. Identity confirmation: The registration node confirms the registration information and completes the registration.

3.3.2 Domain Name Update

After applying for a domain name, the domain registrar can modify the registered domain name information. The process of domain name renewal is the process of domain name transaction. The ownership and status of the domain name are changed. The specific process is: the user initiates a domain name modification request to the master node in the slave chain. After receiving the domain name modification request, the master node queries and checks the domain name information from the distributed database. After completing the transaction on the chain through the consensus algorithm, after the domain name information is updated, respond to the client.

3.3.3 Domain Name Cancellation

When the registered domain name no longer provides services, the domain name can be cancelled. In the slave chain, the client makes a domain name cancellation application to the master node. After verification by other nodes, it is similar to the domain name transaction update, and the domain name registration status is changed to complete the domain name cancellation.

3.3.4 Domain Name Resolution

The domain name resolution process requires the client to initiate data requests from the master chain and the slave chain. The nodes on the chain complete the resolution process by querying transactions on the chain. A complete domain name resolution process is shown in Fig. 5.

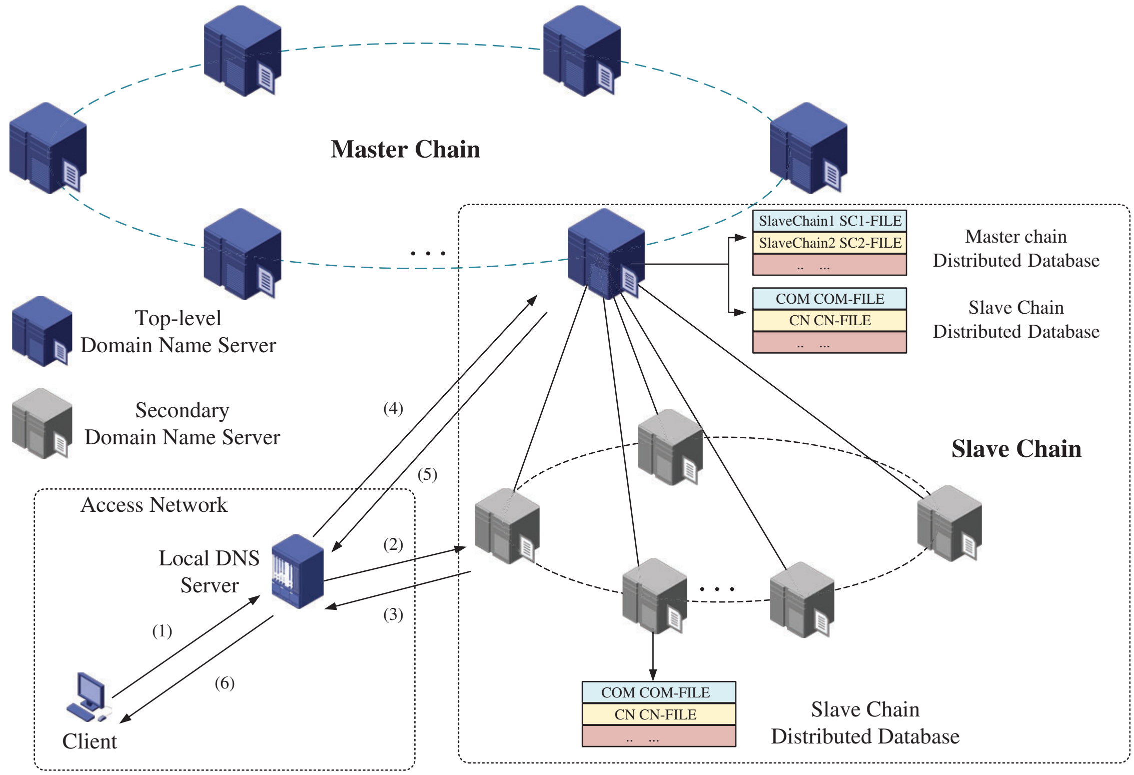

Figure 5: A complete resolution process of DNSMSC

A complete domain name resolution process can be completed in six steps:

1. The client initiates a DNS query request. First, the client queries the local DNS cache and hosts file. If the query result is stored in the cache area, the parsing query ends. Otherwise, the client sends a query request to the local DNS server.

2. After receiving the client's request, the local DNS server queries the local cache and the local server. If the record is found, the query result is returned to the client, and the query ends. Otherwise, the local domain name server sends a request to the slave chain. The local DNS service can be divided into different slave chains according to geographic location or node status. When the client queries, the default slave chain node is preferred.

3. After receiving the query request, the slave chain node searches the data index on the chain. If there is corresponding domain name information, initiate a database query request from the chain node to the distributed database, and send the queried destination IP address to the local domain name server. The local DNS server sends to the client, and the query ends. If the destination IP address is not found, the slave chain node will send feedback information to the local DNS server.

4. The local DNS server receives the feedback information and sends a query request to the master chain node.

5. After receiving the query request, the main chain node indexes the query on the blockchain data to verify the legality of the domain name information. If the domain name information is illegal, the main chain node returns an error message. If the domain name is legal, the main chain node initiates a data query to the distributed database and sends the query result to the local DNS server.

6. The local DNS server sends the query result to the client and saves it in the local cache. A complete domain name resolution is completed.

4 Simulation Evaluation

This paper uses Hyperledger Fabric to realize the simulation of the system with multiple channels, test the concurrent throughput of the system and compare the query delay between the master chain and the slave chain. The first part introduces the simulation hardware configuration and basic software environment. The second part is the concurrent test and delay test of the network.

4.1 Fabric Blockchain Platform Construction

This article installs a virtual machine in the host, and deploys Hyperledger in the virtual machine [18]. The simulation runs on Fabric v1.2, and through the local login execution test, the master chain and slave chain of the multi-channel simulation system are created. The required software environment is shown in Tab. 1.

Table 1: Software environment

Software environment	Installation version
OS	Ubuntu16.04.6 LTS
Docker	18.09.7
Docker-compose	V1.8.0
Golang	Go1.14.6
Hyperledger Fabric	V1.2.0

4.2 Performance Testing

In the concurrent test, this article compares the system throughput of concurrent queries. The total number of transaction is 1000, the step is 5, and the system throughput changes are shown in Fig. 6. When concurrency level is less than 50, the throughput gradually increases. When 50 users access at the same time, the throughput reaches the maximum, the maximum is 106.72Tps/s. When the concurrency level is greater than 50, the throughput gradually decreases. In the throughput test and latency test, the concurrency level is set to 50 in this article.

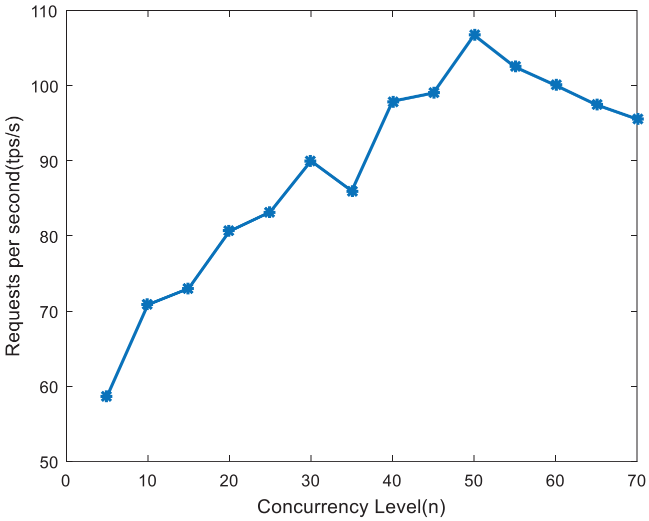

Figure 6: Concurrency test

When the number of concurrent level is 50, the total number of requests gradually increases, and the step is 100. We tested the system's domain name query and added transaction throughput and average delay as shown in Figs. 7 and 8.

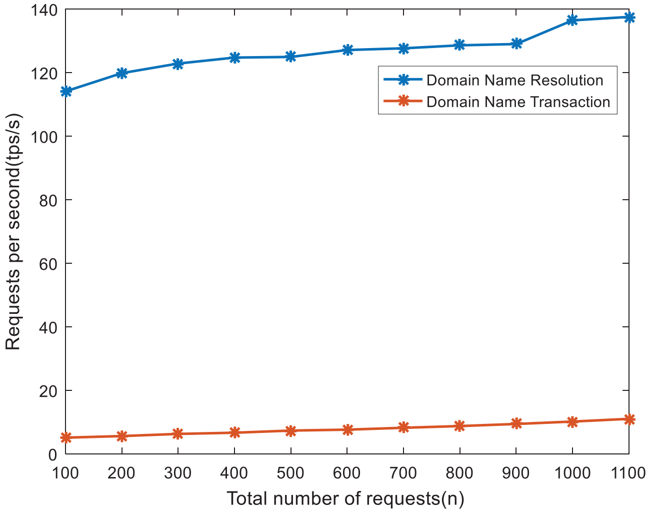

Figure 7: Throughput test

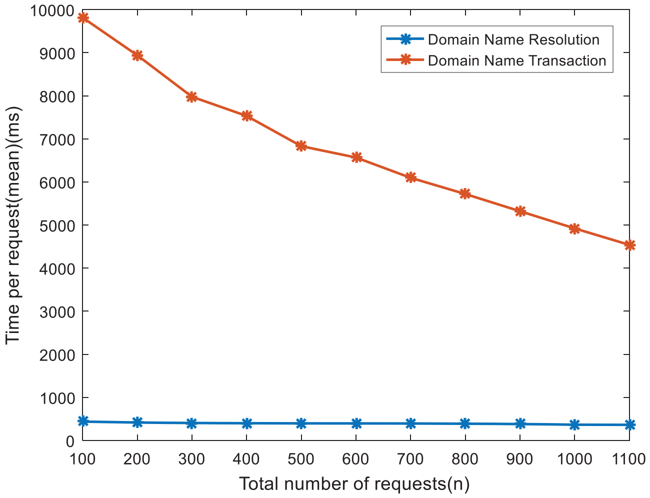

Figure 8: Delay test

From the Figs. 7 and 8, it can be found that the domain name query throughput is much larger than the transaction addition, and the average delay of the transaction addition is much longer than the domain name query. The reason is that the domain name query process does not require the signature and confirmation of blockchain nodes. As the number of transactions increases, the throughput of domain name query and adding calls gradually increases. When the number of transactions is 1000, the query throughput basically reaches the maximum. The domain name query delay and transaction addition delay gradually increase with the increase in the number of transactions. The reason is that the throughput of the blockchain increases and the average equal delay of a single node gradually decreases.

This article compares the single query latency of domain name query on the master chain and slave chain. The latency of 100 visits is shown in Fig. 9. The query latency of direct access to the master chain is higher than that of only the slave chain. The average access latency of the slave chain is 18.72 ms, and the average access latency of the master chain: 30.66 ms. By comparison, this article finds that the DNS service mechanism through the slave chain can reduce the resolution delay.

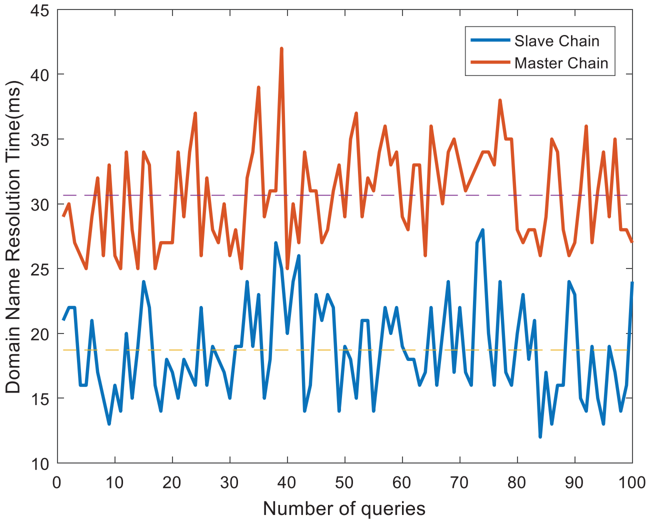

Figure 9: Comparison of DNS resolution delay

5 Summary

The future network system will be equal, open, and co-managed by multiple parties. The existing highly centralized Internet system will be replaced. This paper proposes a DNS domain name resolution service mechanism based on the master-slave chain, and designs a domain name management model based on the master-slave chain. This article uploads transaction information from the chain to the main chain through Multi-Sig Notary Schemes. This model reduces the number of nodes in a single blockchain, enables the entire blockchain to accommodate more nodes, and reduces the impact of the increase in the number of nodes in the alliance chain on transaction efficiency. On the basis of realizing the domain name of the consortium chain, this article increases the speed of domain name resolution and solves the single point of failure problem in traditional DNS system. However, if there is only one gateway node in the slave chain, it may cause a single point of failure. Based on the architecture of the main and slave chains, this article also designs a consensus mechanism MSBFT to ensure that domain name information can be stored on both the main chain and the slave chain. The slave chain method is used to facilitate the management of network nodes and reduce the delay of domain name resolution.

Funding Statement: This work has been supported by State Grid Corporation of China science and technology project “Research on Reliable Transmission Technology in Electric Internet of Things Based on IPv6” (5700-202058178A-0-0-00).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Hu, S. Yin, S. Su, X. Jia, Q. Xiang *et al.,* “Blockzone: A decentralized and trustworthy data plane for DNS,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1531–1557, 2020.

[2] W. Liu, Y. Zhang, W. Zhang, L. Liu, H. Zhang *et al.,* "Self-certificating root: A root zone security enhancement mechanism for DNS," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 521–536, 2020.

[3] J. Liu, X. Sun and K. Song, "A food traceability framework based on permissioned blockchain," *Journal of Cyber Security*, vol. 2, no. 2, pp. 107–113, 2020.

[4] J. Wang, W. Chen, L. Wang, Y. Ren and R. S. Sherratt, "Blockchain-based data storage mechanism for industrial internet of things," *Intelligent Automation & Soft Computing*, vol. 26, no. 5, pp. 1157–1172, 2020.

[5] Y. Yan, Y. Dai, Z. Zhou, W. Jiang and S. Guo, "Edge computing-based tasks offloading and block caching for mobile blockchain," *Computers, Materials & Continua*, vol. 62, no. 2, pp. 905–915, 2020.

[6] J. Liu, B. Li, L. Chen, M. Hou, F. Xiang *et al.,* "A data storage method based on blockchain for decentralization DNS," in *2018 IEEE Third Int. Conf. on Data Science in Cyberspace (DSC)*, Guangzhou, China, pp. 189–196, 2018.

[7] A. Singh, K. Click and R. Parizi, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *Journal of Network and Computer Applications*, vol. 149, no. 7, pp. 102471, 2020.

[8] L. Kan, Y. Wei, A. Hafiz Muhammad, W. Siyuan, L. C. Gao *et al.,* "A multiple blockchains architecture on inter-blockchain communication," in *2018 IEEE Int. Conf. on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, pp. 139–145, 2018.

[9] A. Loibl, "Namecoin," *namecoin. info,* 2014. [Online]. Available: https://www.namecoin.org.html.

[10] M. Ali, J. Nelson, R. Shea and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *USENIX Annual Technical Conf. (USENIX ATC 16)*, Denver, CO, USA, 2016.

[11] H. Li, J. Wu and K. Xing, "Prototype and testing report of a multi-identifier system for reconfigurable network architecture under co-governing (in Chinese)," *SCIENTIA SINICA Informationis*, vol. 49, no. 9, pp. 1186–1204, 2019.

[12] Z. Yu, D. Xue, J. Fan and C. Guo, "DNSTSM: DNS cache resources trusted sharing model based on consortium blockchain," *IEEE Access*, vol. 8, no. 1, pp. 13640–13650, 2020.

[13] X. Duan, Z. Yan, G. Geng and B. Yan, "DNSLedger: Decentralized and distributed name resolution for ubiquitous IoT," in *2018 IEEE Int. Conf. on Consumer Electronics (ICCE)*, Las Vegas, NV, pp. 1–3, 2018.

[14] B. Benshoof, A. Rosen, A. G. Bourgeois and R. W. Harrison, "Distributed decentralized domain name service," in *2016 IEEE Int. Parallel and Distributed Processing Sym. Workshops (IPDPSW)*, Chicago, IL, pp. 1279–1287, 2016.

[15] H. Jin, X. Dai and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *2018 IEEE 38th Int. Conf. on Distributed Computing Systems (ICDCS)*, Vienna, pp. 1203–1211, 2018.

[16] Zhiguo Qu, Yiming Huang and Min Zheng, "A novel coherence-based quantum steganalysis protocol," *Quantum Information Processing*, vol. 19, no. 362, pp. 1–19, 2020.

[17] G. He, W. Su and S. Gao, "TD-Root: A trustworthy decentralized DNS root management architecture based on permissioned blockchain," *Future Generation Computer Systems*, vol. 102, no. 5, pp. 912–924, 2020.

[18] Zhiguo Qu, Siyi Chen and Xiaojun Wang, "A secure controlled quantum Image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 380, pp. 1–25, 2020.