An NDN-Enabled Data Transfer Strategy for Blockchain Network Layer

Dian Yi1, Ru Huo2,3,4, Shuo Wang1,3 and Tao Huang1,3

1 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2 Information Department, Beijing University of Technology, Beijing 100124, China
3 Purple Mountain Laboratories, Nanjing 211111, China.
4 Author to whom any correspondence should be addressed.
Email: dian_yi@outlook.com; huoru@bjut.edu.cn; {shuowang, htao}@bupt.edu.cn

Abstract. The performance bottleneck problem of blockchain has become a core obstacle for its role in more important areas, especially low throughput and long transaction waiting time. Many researches have been conducted to improve the efficiency of blockchain by simplifying the consensus process or redesigning the chain structure at a higher level, but these approaches are subject to various limitations such as the type of blockchain and even pose some security problems. Meanwhile, the blockchain network layer based on the traditional TCP/IP system suffers from problems such as solidified transmission mechanisms and massive redundancy of traffic. Named-Data Networking (NDN), as a new network architecture, can well improve the efficiency of data transmission and information interaction by reshaping the network layer of blockchain. This paper proposes a blockchain data transmission structure using NDN, which will establish the separated data distribution channels among blockchain nodes for block data and transaction data while achieving active data pushing. We design a new table structure for NDN routers to implement path-specific routing and forwarding and exploit the multipath forwarding feature of NDN for fast transmission. Simulation results show that this method has better performance than both existing TCP/IP approaches and typical NDN usage.

Keywords. Named-data networking; blockchain technology; routing and forwarding; data transfer; scalability and security.

1. Introduction
Blockchain technology has gained increasing attention and adoption around the world and is now bringing a revolution in many businesses such as finance, healthcare, supply chain, insurance, registry, and the Internet of Things [1]. Unlike traditional networks or systems that require an authority to dominate the entire system, blockchain enables consensus trust in a distributed manner and collectively maintains tamper-evident data [2]. Blockchain, as a disruptive technology with profound potential, is becoming an effective value communication tool. As the number of users of blockchain systems increases, various mainstream blockchain platforms are experiencing scalability problems, which greatly limit the further widespread use of blockchain [3], especially the poor user experience caused by low transaction throughput and long waiting time for transaction confirmation [4].

Many studies have proposed a large number of different solutions in different directions to improve the performance of blockchains. Due to the decoupling characteristics among the layers in the hierarchical blockchain architecture, the research on blockchain empowered by the network layer is
gradually gaining attention, and the high-speed transmission in the network layer can provide universal acceleration services for the blockchain. In the existing TCP/IP system, blockchain data broadcasting requires nodes to organize and maintain multiple P2Ps at the application layer, which brings computational cost and processing delay, and also limits the potential of the network layer. Unlike traditional IP networks, the content-based driven approach of named data networks (NDNs) enables fast retrieval and delivery of content [5]. The NDN network has excellent features of request aggregation and data caching, which improve the efficiency of content delivery, provide better acceleration of blockchain data and reduce the traffic load on the network.

NDN-based blockchain technology is now becoming popular [6]. In [7], the authors proposed BlockNDN, a scheme to build blockchains on the NDN networks, which aims to improve the single point of failure and the problems that are not conducive to multicast in the blockchain under the IP system, and reduce propagation delay and data transmission overhead. By designing naming rules, they combine NDN names of interest with blockchain data hashes to specify the required data content, and rely on the routing and caching of the NDN network to respond. In addition, the authors in [8] proposed a distributed peer-to-peer network layer based on NDN for blockchain data transfer, which reduces block propagation latency, minimizes network overhead, reduces forking opportunities, and maintains network stability through hierarchical publishing and subscribing. The authors in [9] addressed the problem of a large amount of redundant traffic due to a mismatch between traffic and the underlying network topology in P2P upper layer networks by proposing DIBN, naming categories to decentralize traffic. They established an any-to-all category propagation structure and implemented bidirectional appending strategies.

Although these studies have explored the initial integration of NDNs with blockchain and achieved some results, they still have some problems. Firstly, these approaches focus more on the design of namespaces, while the process of implementation is relatively complex and has limited applicability to different blockchain node communication services. Secondly, some approaches use the selection of volunteer or seed nodes to perform additional functions in a way that increases centralization to some extent, which is contrary to the decentralized design principles of blockchain. Most importantly, relying solely on the NDN's packets of interest for routing and forwarding, as blockchain nodes generate blocks or transactions randomly, it is not clear to the network how to find access to this resource until the data is generated, which on the one hand requires waiting for the routing state to be updated synchronously generating latency, and on the other hand duplication of transmission paths make link utilization inefficient and traffic is too topologically concentrated in parts of the network.

Based on these existing researches, this paper focuses on the design of routing and forwarding of NDN networks to better support blockchain business, hoping to investigate how to use NDN networks to accelerate the data propagation process of blockchain while overcoming the above-mentioned problems as much as possible. This paper first analyzes the necessity and importance of using NDN network as a blockchain network layer. By designing the communication mechanism of the blockchain network layer and the routing and forwarding method of NDN network, this paper proposes a practical NDN-based blockchain data transmission scheme to realize one-to-many data transmission of block and transaction data among blockchain nodes by establishing a specific data transmission structure. This paper also gives a scheme for the establishment, dynamic maintenance and cancellation of the distribution structure, and designs a new table structure and processing algorithm on the NDN router. This mechanism is able to overcome the conflict between the blockchain's communication model based on the content owner's active push and the NDN network's communication model based on the content demander's active pull. It is also able to take advantage of the multipath forwarding capability of the NDN network to reduce latency and further improve data transmission efficiency.

This paper is organized as follows. Section 2 provides background knowledge on NDN and blockchain technology and why NDN is used as a network layer for blockchain. The NDN-enabled data transfer strategy for the blockchain network layer is presented in Section 3, where the namespace design and the NDN routing and forwarding algorithm to support the data transfer structure are given. In
Section 4, we present and discuss the simulation results. In Section 5, we summarize the work done in this paper and propose some subsequent improvements.

2. Overview of NDN and Blockchain

Named data network is a future information-centric network architecture that follows the hourglass model of IP networks, but uses data names as a fine waist to enable data naming-based routing and forwarding by naming data instead of hosts [10]. The communication under NDN networks is driven by the requesting party, and its data messages are divided into two categories: interest packets and data packets. Interest packets identify user expectations by containing content names, and data packets use names to identify the data of the load. Routing nodes accomplish forwarding through three important data structures, as shown in figure 1: Forwarding Information Table (FIB), Pending Interest Table (PIT) and Content Cache (CS), which are responsible for determining the next hop based on name, recording unresponsive interest packets and saving node cache contents, respectively [11].

Blockchain is a data structure with decentralized, irreversible and tamper-evident properties [12], which relies on peer-to-peer (P2P) networks for data transmission, where each node can be both a provider and a user of data, and where the latest current transactions and block data are obtained and verified through direct data transmission between nodes [13]. Multicast is the most frequently used means of communication in the blockchain network layer, where blocks and transactions are often generated by a single node and need to be broadcast to a group of nodes. The traditional blockchain network layer is built on TCP/IP [14], where two nodes must obtain each other's IP addresses before communicating and connect after multiple handshakes before subsequent data packets can be forwarded. For multicast, a node that wants to transmit data to all N nodes in the network needs to establish multiple P2P connections that encapsulate N data packets to be sent to each of the N nodes. The block and transaction data are often the same repetitive content, but must be transmitted multiple times because each request is independent of each other. These duplicate transmissions are not only redundant, but also bring more risk of occurrence of packet loss retransmission, which brings greater impact on the efficiency and performance of the blockchain system, and also greatly increases the traffic load of the network.

An NDN-based blockchain network layer will bring many benefits. First, NDN is an information-centric model [15], which can separate content location from name and improve the efficiency of data acquisition. Using the aggregation feature of NDN, a single copy of data can be sent directly to all recipients by replication at the desired router node. The data may fail to be sent or may be requested again to get a hit in the router. These features can well support various blockchain technology applications and ensure better real-time performance of these services from the communication aspect. Using NDN as the network layer of blockchain can alleviate the large communication overhead generated by end-to-end connections when blocks and transactions are synchronized across the network under IP networks, reduce the redundant transmission of the same content, and reduce the pressure on network traffic.

Moreover, the design idea of NDN is in line with the higher requirement of anonymity in blockchain by broadcasting the content of the transaction while hiding the information of the counterparty, making the source of the transaction untraceable in the real world. Even though the blockchain has a sophisticated cryptographic design, its anonymity and untraceability are not completely impenetrable, and the side-channel attack is one of the attack schemes. Whether it is a time-domain side-channel attack or a network traffic-based side-channel attack, the key thing is to obtain the address of the counterparty by intercepting data packets, and the NDN network will make such attacks history and further improve the security of the blockchain.

3. NDN-Based Blockchain Data Transmission Method

In order to support the data transmission of blockchain business, we design the Inter-Blockchain-Node Data Transmission Structure (IBN-DTS) for its communication features. This structure carries fast forwarding of transactions and block data by establishing one or more specific links between blockchain
nodes. In order to make the transport structure work, we must plan and design the namespace in the NDN network and adjust the routing and forwarding policies of the NDN routers. IBN-DTS can be seen as a tree structure consisting of several blockchain nodes, NDN routing nodes and network link connections between nodes, as shown in figure 2. When a blockchain node needs to broadcast data, it only needs to send one copy of the packet with a specific naming to the edge router connected to the node. The packets will be forwarded along the IBN-DTS and replicated in NDN routers if needed, thus completing the one-to-many data distribution.

Between the distributed nodes of the blockchain system, we establish two types of IBN-DTS, the main IBN-DTS and the auxiliary IBN-DTS. The main IBN-DTS is unique in the cluster and will associate all nodes within the blockchain and provide each node with one-to-all data transmission services respectively. This structure is responsible for communication services with low real-time requirements but high data service volume, such as block data broadcasting and synchronization. Auxiliary IBN-DTS is built for each blockchain node, providing it with one-to-group data transmission services that meet the needs of that node. Different nodes will have different auxiliary IBN-DTS structures. It is responsible for communication services with high real-time requirements and frequent communication but small packets, such as online or offline messages of this node, broadcasting of transaction data and exchange of address lists, etc.

3.1. Namespace Design
The establishment, cancellation, and other operations of the IBN-DTS structure are controlled by a series of signals that are transmitted using interest packets. These interest packets use the same namespace as the data packets that carry transaction and block data. The basic format of the naming used by IBN-DTS is as follows.

/IBN–DTS/Blockchain_name/Type/Method/Sponsor_ID

The first item “IBN-DTS” is fixed as the specific name of this method to distinguish ordinary traffic from blockchain service traffic, and the router identifies and processes the corresponding traffic according to this naming. The Blockchain_name variable identifies the blockchain to which the IBN-DTS belongs and consists of the blockchain name and a string of random characters to ensure the isolation between different blockchain services. The Type variable identifies the type of the IBN-DTS and takes the values main and minor, corresponding to the main IBN-DTS and auxiliary IBN-DTS types, respectively. The Method variable identifies the operation method of the IBN-DTS and has three values: use indicates the use of structure for data transfer, which is mainly placed in packets to identify transactions or block data; construct is used in the establishment process, and clear is used for clearing. Sponsor_ID identifies the creator of the IBN-DTS and consists of the ID number of the initiating node. The port group identifies the associated ports of the IBN-DTS in the router, and its role is to forward packets to all ports in the port group except the incoming port when a packet with the corresponding matching name is received through a port.

3.2. NDN Router Processing Algorithm
Blockchain nodes send interest packets and data packets using specific naming, and NDN router nodes respond according to their contents. For this purpose, we add Correlation Face table (CFT) to NDN to record the status of each IBN-DTS in real time and guide the forwarding path of data packets. The CFT of each router records the matching name and port group information. The basic format of the CFT is shown in table 1.
Table 1. Comparison table of basic format of CFT.

| Name | Port Group | Survival Time |
|------|------------|---------------|
| / IBN-DTS / Blockchain_name / Type / Method / Sponsor_ID | < Face ID(1), Face ID(2), Face ID(3) …> | <TTL>(s) |
| Example:/ IBN-DTS / Finance Chain_AiEktxYu{xjKnQbz / Main / use / mKB2TvFCZqv5WStj | < face 3 , face 6 , face 7 > | 3600 |
| ……. | ……. | ……. |

When the router nodes in the NDN network receive data, each node runs the same processing algorithm, as shown in figure 3. Interest packets and data packets are processed separately according to packet types, with interest packets used for signaling transmission and data packets used to carry the communication data of the blockchain. For interest packets, we first perform name detection to distinguish between IBN-DTS services and other services. For other service interest packages, follow the NDN default processing logic. For IBN-DTS signaling, it is judged that the structure is established or cancelled, and if the structure is established, the corresponding CFT table entry is added according to the naming and forwarded according to the FIB, and if the structure is cancelled, the CFT table entry is cleared, and at the same time, after operating the table entry, forwarding is performed according to the table entry. For packets, the same distinction is made between service types, other services are forwarded according to PIT and copied to CS for content caching according to the caching policy selection. For blockchain data distribution carried by IBN-DTS, it is forwarded to each of the remaining ports in the port group except the incoming port by querying the CFT.

Figure 1. The packet forwarding model of the NDN.

Figure 2. IBN-DTS architecture diagram.
3.3. IBN-DTS Transmission Structure Establishment

3.3.1. Main IBN-DTS Structure. The main IBN-DTS implements the function of connecting all blockchain nodes within a cluster. When a node needs to synchronize a block it combines its own arithmetic power and network state to consider the establishment, and the node is responsible for the management of the global unique structure. At the time of establishment, the initiating nodes are sorted according to the arithmetic power level of the blockchain nodes and connected one by one. The initiating node will send a message to the next ready-to-connect node informing it of the IBN-DTS node list (the current node list contains only the initiating node). The destination node tests the connection link with each node one by one according to the node list, including delay, jitter and packet loss rate, to evaluate the uniform Qos of the multi-hop link, and completes the creation of the optimal link by selecting the node with the best performance and sending an IBN-DTS establishment request. The signaling is forwarded according to the NDN default routing rules, and during the forwarding process each router records the name, incoming port and outgoing port into the associated port table of that router. After the initiating node updates the IBN-DTS node list (the list contains blockchain nodes and routing nodes), it repeats the above process by selecting the next node according to the ranking until all blockchain nodes have joined the IBN-DTS.

We use a completely randomized network as an example. As shown in figure 4, the NDN network consists of eight router nodes from R1 to R8 and six blockchain nodes from BN1 to BN6. All blockchain nodes have been formed on the underlying NDN network and joined the same blockchain network. In order to describe the improvement of the network layer, we divide all links in the network into two categories: edge links and core links, where edge links refer to the links between blockchain nodes and routers, and core links Refers to the link between routers.

When a node in BN1-BN6 needs to send data to other nodes (including but not limited to block data), first check whether the main IBN-DTS exists, and if it does not exist and the conditions are appropriate, it will be constructed. Assuming that it is initiated by BN1, the order of computing power is: BN6>BN3>BN2>BN4>BN5. The whole establishment process will be initiated by BN2 to BN6 respectively, and is divided into five steps. BN1 will notify in order of arithmetic power, BN6 will first initiate the establishment request, and the interest packet will make R7, R8 and R1 create CFT table entries, and the three routers will contain the two ports connected by the red line in the corresponding CFT table additions. At this point, BN1 updates the node list to R1, R8 and R7, and sends this list to BN3, which detects the network status (latency) of each of the three routers and selects the router with the best link status (e.g. R1) to send the establishment request, which will cause R2 and R2 to create CFT table entries containing the port numbers connected by their orange lines, allowing R1 to add new ports to the existing CFT table entries. Finally, BN1 updates the node list to R1, R2, R3, R7, R8, and so on, finalizing the entire structure according to steps 1 to 5 in the figure. When any blockchain node
needs to send block data again, it only needs to order data packets according to the established IBN-DTS name and send them to the router connected to the node. Finally, each blockchain node will receive the data through the main IBN-DTS.

3.3.2. Auxiliary IBN-DTS Structure. The auxiliary IBN-DTS is decided by each blockchain node and can be built when it needs to broadcast a transaction, and if it chooses to use IBN-DTS, it sends a specifically named interest packet to all node addresses in the node list to complete the build. This interest packet route router makes path selection and forwards according to its FIB table, while establishing the corresponding CFT table entry by detecting the naming of the interest packet. Again using the previous network model as an example, suppose BN1 tries to establish multicast to groups of BN2, BN3, BN4 nodes, the whole construction process is shown in figure 5. Since the auxiliary IBN-DTS structure is built exclusively for the originating node to broadcast data, there will be originating nodes that use the packets of interest to update the router CFT table entries and not other nodes. In addition, the auxiliary IBN-DTS carries a one-to-many data stream that is unidirectional. (These two points are different from the establishment process of the main IBN-DTS structure.)

BN1 sends specifically named interest packets to BN2, BN3 and BN4 respectively for auxiliary IBN-DTS structure building, forwarding them according to default routing rules, forming three sub-paths to their respective nodes. The first path will create a CFT table entry on R1, which will contain the incoming and outgoing ports of the interest packets (i.e., the ports connected by the green path). The second path will cause R2 and R3 to create CFT table entries that will contain the two ports connected by the blue path, and cause R1 to update the CFT table entries to include the new outgoing ports. The third path is similar. The final result is the auxiliary IBN-DTS structure connected by the solid black line segment. When BN1 needs to broadcast data, it orders packets according to this established IBN-DTS’s name and sends a copy to R1, the router to which BN1 is connected. The router that receives the data will forward it according to the structure according to the CFT, causing each BN in the node list to receive the data.

3.4. Dynamic Maintenance and Cancellation of the Structure
Both the main IBN-DTS and the auxiliary IBN-DTS are maintained by the initiating node (e.g., BN1). The main IBN-DTS has a certain lifetime after it is established, and the timer is refreshed in the router each time the structure is used, and the CFT table entry will be cleared when the timer expires. The structure can also be actively logged out by the maintenance node. The master IBN-DTS will be checked for link status by the initiating node. When the inter-router link is disconnected, the involved router nodes try to report the affected main IBN-DTS structure to the initiating node to facilitate the re-establishment of the structure (the initiating node can be reselected at this point). In case of link failure between the edge router and the blockchain node, port state maintenance is performed by the edge router, which will wait for a period of time for secondary detection and revoke the port in its CFT table entry.
if it is still offline. If this node joins again subsequently according to the new node identity. When a new node joins, it will obtain the structure information through other nodes that join the structure and initiate a join request, which is the same process as the auxiliary IBN-DTS establishment process in a round. When a node leaves on its own initiative, it can revoke the CFT record to the edge router or wait for its expiration.

The life cycle of the auxiliary IBN-DTS is determined by the blockchain node that establishes it and can use specific named packets of interest to control the edge router clear CFT table entries. When a new node joins, the blockchain node that establishes it refreshes (adds) the auxiliary IBN-DTS by updating its node list to include the newly joined node in the transport structure. The auxiliary IBN-DTS has a strong binding relationship with the node list of the initiating node. When a link fails or a node goes offline, the locally maintained node list is automatically adjusted or the structure is re-established. For accidental offline nodes, the edge router also has a secondary detection mechanism.

4. Simulation Results and Analysis
We conducted simulations in a network environment with eight router nodes and six blockchain nodes as described in the previous section. In this paper, a simple bitcoin blockchain system is simulated using python for numerical simulation. Blockchain nodes randomly and continuously generate transaction data and synchronize it across the network. We selected two metrics, block synchronization time and block synchronization traffic, to compare the TCP/IP approach, the typical NDN network transmission approach (BlockNDN), and the IBN-DTS approach described in this paper, respectively. The TCP/IP approach sends data from the initiating node to the blockchain nodes in the node list, and the blockchain nodes receive it and continue forwarding it to their own node list; the blockchain nodes that receive the same data a second time will not forward it again. The BlockNDN approach leaves the data at the route node, and the rest of the nodes get the data from the nearest router cache according to a random polling cycle, which requires the use of interest packets and packets of data to complete. Considering that the computation time that blockchain nodes need to recheck the hash value after receiving the data is much smaller than the data transmission time in the network, we approximate that the time for the block to complete synchronization is the time for the block data to be transmitted in the network. The main experimental parameters are shown in table 2.

First, we compared the three network communication methods in terms of the total time required for full network synchronization of blocks and the average time required for a single block to achieve full network synchronization. We counted the blockchain system in milliseconds from the start of the first block generation to the fiftieth block, as shown in figure 6. To see a clearer comparison of the results, we have zoomed in on the process from the generation of the first block to the twentieth block. It can be seen that the total time to achieve network-wide synchronization increases steadily as the number of blocks grows. The NDN network generally outperforms the traditional TCP/IP network under the same network conditions, reflecting the fact that NDN networks can provide some performance improvements to blockchain systems. At the same time, the method described in this paper outperforms the typical transmission method of existing blockchains under NDN networks, a feature that is not obvious when the number of outgoing blocks is small, mainly due to the time-consuming process of establishing the IBN-DTS structure, which brings about a performance degradation. After repeated experiments, the method described in this paper outperforms the performance of the TCP/IP approach in approximately 3-6 block cycles and the NDN approach in 8-12 block cycles, and as the number of blocks generated increases, the gap in the time required for blocks to achieve network-wide synchronization between this method and the other two approaches is gradually widening.
Table 2. Experimental parameters table.

| Experimental parameters                              | Specific values                                      |
|------------------------------------------------------|------------------------------------------------------|
| Edge link propagation delay                          | Randomly generated within 0-1ms                      |
| Core link propagation delay                          | Randomly generated within 1-3ms                      |
| Routing algorithm                                    | Shortest path                                        |
| Hop-by-hop processing delay and queuing delay        | $\mu=0$, $\delta=10\text{ms}$ normal distribution    |
| Block generation                                     | Random node generation                               |
| Arithmetic order                                     | Random ordering                                      |
| Number of node lists maintained by blockchain nodes  | 2                                                    |
| Polling period for block / transaction fetching      | 100ms                                                |

Through our experiments, we also obtained the average time taken for a single block to achieve network-wide synchronization, as shown in figure 7, and zoomed in on the process from the twentieth block to the fiftieth block generated in the figure. Due to the randomness of block generation at nodes and network fluctuations, the synchronization elapsed time of a single block exhibits a large volatility. However, it can still be found that the method shown in this paper outperforms the NDN method, and the TCP/IP approach has the worst performance, and this trend gradually stabilizes as the number of blocks increases. The time cost consumed by this method to synchronize individual blocks is higher in the initial stage, which is mainly due to the time required for the structure building process.

![Figure 6](image-url). Block network synchronization time.

![Figure 7](image-url). Average synchronization time overhead of a single block.

In addition, data synchronization in blockchain not only affects its own performance, but also puts traffic pressure on network devices. In this regard, we compare the traffic metrics in MB. The experiment assumes a block size of 1 MB. We first simulate the average traffic generated by the synchronization process of a single block across the network, as shown in figure 8. We generated a block at a random node in the blockchain network and counted the size of traffic consumed by the block when synchronization was achieved at six nodes across the network. The experiment was conducted 100 times to get the average value. It can be clearly seen that NDN is able to reduce the blockchain network traffic by about more than half. The method described in this paper also has a slight performance improvement over the typical NDN network approach during the synchronization process.

Secondly, we simulated the blockchain system from the generation of the first block to the fiftieth block and compared the average traffic generated by the synchronization process across the network in different ways as the number of blocks synchronized varied. We generated blocks at random nodes in the blockchain network and counted the number of blocks generated and the total amount of traffic
consumed by the network-wide synchronization, as shown in figure 9. The TCP/IP approach generates a large amount of traffic for data synchronization between blockchain nodes, and as the number of blocks increases, more traffic is consumed, while NDN can reduce the traffic pressure well and this approach can bring further performance improvements.

5. Conclusion
This paper focuses on the fast data transmission strategy of blockchain enabled by NDN network. In order to support blockchain business, this paper establishes a data transmission structure for blockchain system based on NDN network and designs the process of structure establishment, dynamic maintenance and cancellation, and proposes the operational processing algorithm of NDN router to support this data transmission structure. Simulation results demonstrate that the IBN-DTS method proposed in this paper has excellent performance in accelerating data broadcast and reducing the traffic pressure on network devices for block synchronization in blockchain networks, reducing data synchronization time from TCP/IP by about one-fifth and reducing redundant traffic in the synchronization process by about half, compared with the way blockchain data transmission is performed in typical NDN networks. The proposed approach in this paper also provides further performance improvements. With the gradual expansion of network size and complexity of network topology, the way of establishing IBN-DTS globally will become fragile and difficult to maintain; therefore, in future work, we consider grouping blockchain nodes according to network locations and organizing IBN-DTS for communication among and within clusters separately, so as to improve the mechanism described in this paper in larger scale and more complex network topologies, and explore the use of caching and other means to further extend the performance bottleneck of the blockchain.

Acknowledgments
This work is supported by The National Key Research and Development Program of Guangdong Province (No.2018B010113001) and the Huawei collaboration project (20201124006).

References
[1] Sanka A I, Irfan M, Huang I, et al. 2021 A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research Computer Communications. 169 (10).
[2] Namasudra S, Deka G C, Johri P, et al. 2020 The revolution of blockchain: State-of-the-art and research challenges Archives of Computational Methods in Engineering 2020 (2).
[3] Zhou Q, Huang H and Zheng Z 2020 Solutions to scalability of blockchain: A survey IEEE Access 2020 (99).
[4] Zheng P, Zheng Z, Luo X, et al. 2018 A detailed and real-time performance monitoring framework for blockchain systems 2018 IEEE/ACM 40th ICSE-SEIP pp 134-143.
[5] Zhang L, Afanasyev A, Burke J, et al. 2014 Named data networking ACM SIGCOMM Computer Communication Review 44 (3) 66-73.
[6] Asaf K, Rehman R A and Kim B S 2020 Blockchain technology in named data networks: A detailed survey Journal of Network and Computer Applications 171.
[7] Tong J, Xiang Z, Liu Y, et al. 2017 BlockNDN: A bitcoin blockchain decentralized system over named data networking Ninth International Conference on Ubiquitous & Future Networks.
[8] Sedky G and Mougy A E 2018 BCXP: Blockchain-centric network layer for efficient transaction and block exchange over named data networking 2018 IEEE 43rd LCN.
[9] Li R and Asaeda H 2020 DIBN: A decentralized information-centric blockchain network GLOBECOM 2019.
[10] Feng M, Ma H, Hu Y, et al. 2020 Research on the influence of different network topologies on cache performance for NDN ICAIS 2020 pp 365-375.
[11] Ariefianto W T and Syambas N R 2017 Routing in NDN network: A survey and future perspectives 2017 11th International Conference on TSSA.
[12] Nakamoto S 2012 Bitcoin: A Peer-to-Peer Electronic Cash System https://itcoin.org/itcoin.pdf (accessed on 11 February 2012).
[13] Wang H, Zheng Z, Xie S, et al. 2018 Blockchain challenges and opportunities: A survey International Journal of Web and Grid Services 14 (4) 352-375.
[14] Liu J, Huo R, Li C, et al. 2018 Information transmission mechanism of Blockchain technology based on named-data networking Tongxin Xuebao/Journal on Communications (39) 24-33.
[15] Wang L, Lehman V, Hoque A, et al. 2018 A secure link state routing protocol for NDN IEEE Access 2018 (99) 1-1.