Journal of Information Processing Vol.31 387–391 (June 2023)

[DOI: 10.2197/ipsjjip.31.387]

Technical Note

A Study of Using Plumtree Algorithm in Blockchain Networks

Yusuke Kitagawa[1,a)] Kazuyuki Shudo[2,†1] Osamu Mizuno[1] Ryohei Banno[1]

Received: November 29, 2022, Accepted: March 7, 2023

Abstract: Blockchain is gaining attention as a technology to support cryptocurrency. However, one of the problems in using blockchain is the consumption of communication resources. It is caused by that the information received by a node can be delivered to the same node more than once through different neighbors. In this paper, we propose a method to reduce duplicate messages in a blockchain network by applying Plumtree algorithm. Through simulation experiments, we confirmed that the number of messages can be reduced by more than 70% when block propagation is simulated up to 50 blocks.

Keywords: blockchain, SimBlock, plumtree, peer-to-peer network

1. Introduction

The distributed ledger technology represented by blockchain is a system in which participants share records and keep being difficult to tamper with them. It is known for its applications in virtual currencies and financial services such as Bitcoin [1]. The Ministry of Economy, Trade and Industry (METI) estimates the Japanese market size for blockchain to be approximately 67 trillion yen [2]. In a typical blockchain, such as Bitcoin, a large number of nodes randomly interconnect to form a Peer-to-Peer (P2P) network. Each node broadcasts information to all other nodes in the P2P network. It is conducted in a flooding manner, i.e., a node that receives the information forwards it to the neighbors. As a result, information that has already been received may be received multiple times through different neighbor nodes, resulting in excessive consumption of communication resources.

In this study, we aim to solve the problem of communication resource consumption in the Bitcoin network by using the Plumtree algorithm [3], which is known as an efficient broadcast method.

This paper is an extended version of our previous work [4] presented in IEICE ICETC 2021. The differences include the calculation of the message breakdown and the evaluation by simulation of a case in which the number of blocks is set to 50 and a leaf node generates a block. Additional experiments and discussion of related works are also included.

2. Related Work

There are several existing studies aiming to reduce network traffic in blockchain networks.

Kan et al. [5] have proposed a method to introduce a tree-based structure in blockchain networks. To obtain redundancy, each node joins a node group that composes a delivery tree. This method has the advantage of tree-based routing, but redundant communication still occurs since each message is shared among group members in a flooding manner inside the group.

Jin et al. [6] have proposed a method to use erasure coding in block propagation. This method enables that each node does not need to send the complete information of a block. It can reduce the required bandwidth for block propagation, but it still needs to exchange inventory (inv) and getdata messages.

3. Broadcast in Bitcoin Network

Figure 1 shows the sequence of broadcasting a block in Bitcoin networks. A Bitcoin node first sends an “inv” message to neighbor nodes when it receives a block. The node that receives the message checks if there are any missing blocks. If there is a missing block, it sends a “getdata” message to the sender node of the inv message and asks it to send the missing block. This mechanism eliminates duplicate delivery of a block. Although duplicate delivery of a block can be avoided, there still be a problem of duplicate delivery of inv messages. Since an inv message is propagated in a flooding manner, a node could receive it multiple times via different neighbor nodes.

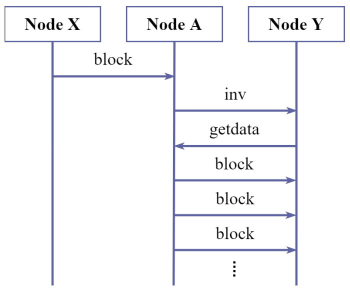


Fig. 1 Broadcasting a block in Bitcoin network.

1 Kogakuin University, Hachioji, Tokyo 192–0015, Japan
2 Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
†1 Presently with Kyoto University
a) cm21016@g.kogakuin.jp

4. Proposed Method

As mentioned in Section 3, there is a problem that a message that has already been received is delivered multiple times via different nodes. In this study, we solve the problem of the excessive communication resource consumption by using the function of reducing redundant message transmission in Plumtree based on the Bitcoin network.

4.1 Plumtree

Plumtree is a broadcast method that combines Tree-based and Gossip-based approaches to reduce redundancy while maintaining high message reliability. In this method, each node forwards messages basically in a Tree-based manner. It also uses a Gossip-based approach to properly handle network failures.

A Gossip-based approach is that all nodes contribute equally to message propagation. When a node broadcasts a message, it randomly selects nodes to send the message. Each node that receives a message for the first time repeats this process. This property allows the system to respond flexibly to network failures or an increase or decrease in the number of nodes. Forwarding messages in a Gossip-based approach is classified into the following mechanisms, as shown in **Fig. 2**.

- Eager push : When a message is received, it is immediately sent to neighboring nodes.
- Lazy push : When a node receives a message, it sends the message identifier to its neighbors. If the node that received the identifier has not received the message, it makes a Pull request.

Plumtree uses both Eager push and Lazy push mechanisms. That is, it uses Eager push in Tree-based routing for reducing redundancy, and Lazy push in Gossip-based routing for maintaining reliability.

4.2 Applying Plumtree to Bitcoin Network

Initially, a block is broadcasted by using the standard procedure of Bitcoin as shown in Fig. 1.

During the initial broadcast process, a spanning tree is constructed according to the Plumtree algorithm. In subsequent broadcasts, the constructed tree is used. As shown in **Fig. 3**, the processes of exchanging inv and getdata messages among nodes are omitted, and blocks are sent directly to the child nodes of the tree. For handling node churn and network failures, the Gossip-based repairing process of Plumtree is used.

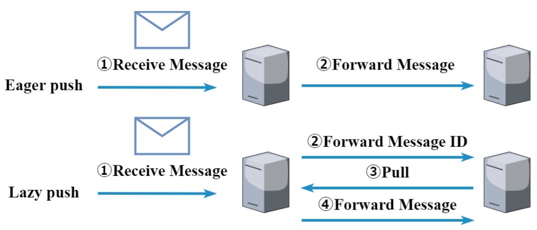


Fig. 2 Gossip-based routing.

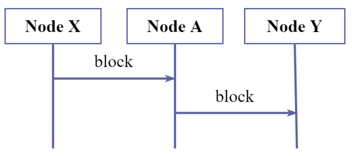


Fig. 3 Broadcasting a block in proposed method.

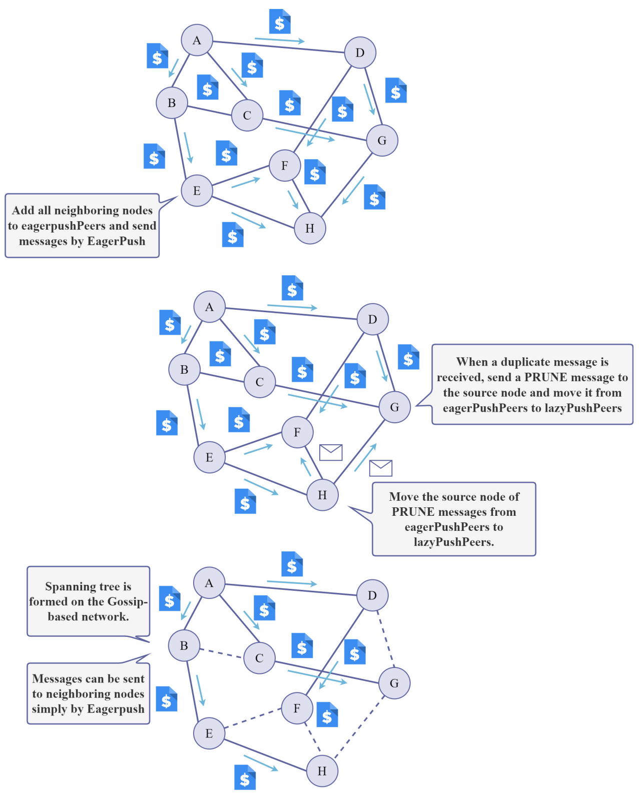
4.2.1 Spanning Tree Construction

The proposed method applies Plumtree to the Bitcoin network. Initially, blocks are broadcast using standard Bitcoin procedures, as shown in Fig. 1. Based on the routing information used in the broadcast, each node is assigned adjacency information in the spanning tree. The information possessed by each node is shown below.

- eagerPushPeers : Neighboring node set in spanning tree
- lazyPushPeers : Set of adjacent nodes other than eagerPushPeers in spanning tree

In the first broadcast, a spanning tree is constructed according to the Plumtree algorithm. In subsequent broadcasts, the constructed tree is used. As shown in Fig. 3, the exchange of inv and getdata messages between nodes is omitted, and blocks are sent directly to the child nodes of the tree. For node switching and network failures, Plumtree's Gossip-based repair process is used.

The following **Fig. 4** illustrates the operation of Plmutree. All nodes that have information on eagerPushPeers send messages by eager push. If a message is received for the first time, the source node is added to the eagerPushPeers. If the message has already been received, the source node is moved from eagerPushPeers to lazyPushPeers. It then sends a PRUNE message to the source node. The node that receives the PRUNE message moves the source node of the PRUNE message from eagerPushPeers to lazyPushPeers. In this way, the PRUNE message is used to optimize the route. When the first broadcast is completed, a spanning tree is formed. In a stable network, a broadcast can only be made



A
B
C
D
E
F
G
H
Add all neighboring nodes to eagerpushPeers and send messages by EagerPush
When a duplicate message is received, send a PRUNE message to the source node and move it from eagerPushPeers to lazyPushPeers
Move the source node of PRUNE messages from eagerPushPeers to lazyPushPeers.
Spanning tree is formed on the Gossip-based network.
Messages can be sent to neighboring nodes simply by Eagerpush

Fig. 4 Spanning tree construction by Plumtree.

by an eager push on the spanning tree. If a node C leaves and can no longer send messages, it does not know whether it has received new messages, so it queries the lazyPushPeers in that node for new messages with lazy pushes. Thus, the Plumtree algorithm is used to construct the spanning tree.

4.2.2 Tree Repair

In addition to eager push, each node performs lazy push. When a failure occurs, Eager push cannot reach all nodes. Lazy push is responsible for delivering messages to unreceived nodes and repairing the tree in such cases. Lazy push periodically sends IHAVE messages to each neighboring node in lazyPushPeers. The IHAVE message contains a message ID, and the amount of communication can be reduced by increasing the interval between periodic transmissions. Depending on the interval of the periodic transmission, multiple IHAVE messages are sent together in a single communication. When a node receives an IHAVE message, it waits for a certain period of time as a "missing" message if it has not yet received the message by Eager push. If the message is received by Eager push before the waiting time elapses, tree repair is not performed. If the message is not received by Eager push before the waiting time elapses, the following process is performed.

5. Evaluation

This section describes the experimental environment and the result of the evaluation. The experimental parameters of SimBlock [7] used in this experiment are listed in **Table 1**. A comparison was made between the Bitcoin transfer method and the method used when Plumtree was applied. To investigate the impact on the network, we used the 50% block propagation time as an evaluation metric. This variable has been used in research [8] and is an appropriate metric to study the impact. No increase or decrease of nodes was assumed, Compact Block Relay (CBR) was not used, and block propagation was simulated up to 50 blocks. The default values of SimBlock were used for the rest of the simulations. The number of trials was 10.

5.1 Number of Messages

We evaluated the number of messages when Plumtree is applied to a blockchain. **Figure 5** shows a comparison of the number of messages between the proposed and conventional methods. The vertical axis is the total number of messages received by all nodes, and the horizontal axis is the number of blocks. The number of messages in the proposed method is the number of inv messages only at the time of the first broadcast. The second time, the number of messages increases because if the message received by eager push is one that has already been received, a PRUNE message is sent to the source node to update the routing information. For the third and subsequent messages, the spanning tree created by Plumtree is used, so there is no need to send and receive inv messages. As a result, the number of messages is smaller than with the conventional method. In a stable network, the number of messages can be reduced by reducing the frequency of IHAVE messages.

Table 1 SimBlock parameters.

Parameter	Value
NUM_OF_NODES	100
BLOCK_SIZE	535 Kbyte
NUM_OF_RATE	100%
BLOCK_OF_RATE	10
CBR_USAGE_RATE	0
CHURN_NODE_RATE	0

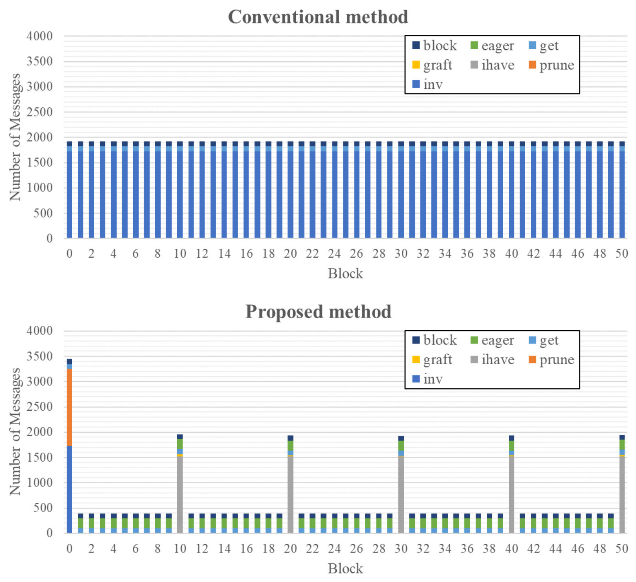


Fig. 5 Number of messages.

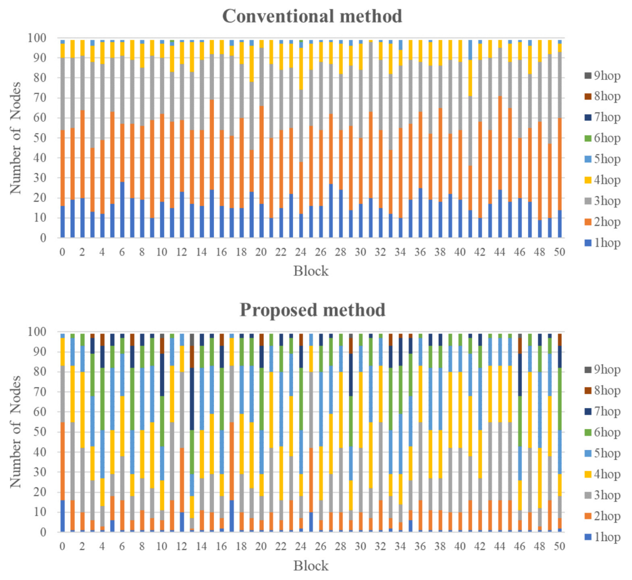


Fig. 6 Number of hops.

5.2 Number of Hops

The number of hops per block was measured. **Figure 6** shows the number of hops for the proposed method and the conventional method. The vertical axis indicates the number of nodes, and the horizontal axis indicates the number of blocks. The conventional method sends inv messages to its neighbors for each block to share messages, so it can send messages to nodes as close as 1 or 2 hops. On the other hand, the proposed method constructs a route for the node sending the first block, so if subsequent blocks are sent by other nodes far from the root of the constructed spanning tree, they are forwarded using a path that is not the shortest,

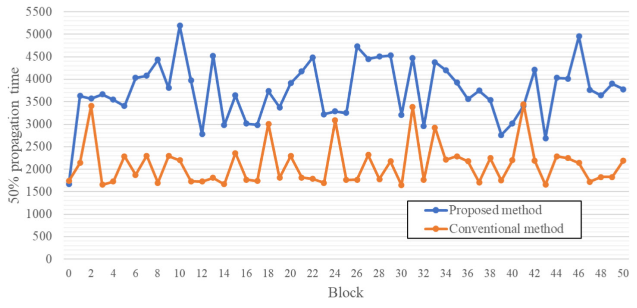


Fig. 7 50% block propagation time.

resulting in an increase in the number of hops.

5.3 Block Propagation Time

Because the number of hops increases with the Plumtree, we checked whether the block propagation time is affected. **Figure 7** shows the 50% block propagation time for the entire network. It can be seen that the conventional method transmits blocks with relatively stable block propagation time because it communicates with its own neighbors. The proposed method constructs a route according to the node that sends the first block, so the propagation time of the first block is similar to that of the conventional method, but as the number of hops increases, the propagation time of the block increases.

6. Conclusion

In this paper, we have proposed a method of applying Plumtree algorithm to blockchain networks and evaluated it by simulation. As a result, we found that the number of messages can be reduced compared to existing methods. Future work includes the improvement of the construction of spanning trees in order to reduce the number of hops.

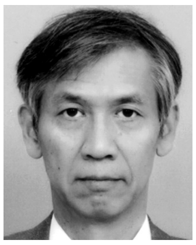
Acknowledgments This work was supported in part by the JGC-S Scholarship Foundation. A part of this work was supported by JSPS KAKENHI Grant Number JP21H04872.

References

[1] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008), available from ⟨https://bitcoin.org/bitcoin.pdf⟩ (accessed 2022-11-10).
[2] Ministry of Economy, Trade and Industry (METI): FY2015 Survey Report on the Infrastructure Development for the Informatization and Servitization of Japan's Economy and Society (2015), available from ⟨https://www.meti.go.jp/main/infographic/pdf/block_c.pdf⟩ (accessed 2022-11-10).
[3] Leitão, J., Pereira, J. and Rodrigues, L.: Epidemic Broadcast Trees, *2007 26th IEEE International Symposium on Reliable Distributed Systems* (*SRDS 2007*), pp.301–310 (online), DOI: 10.1109/SRDS.2007.27 (2007).
[4] Kitagawa, Y., Shudo, K., Mizuno, O. and Banno, R.: Verification of Applying Plumtree Algorithm for Blockchain Networks, IEICE International Conference on Emerging Technologies for Communications (ICETC) (2021).
[5] Kan, J., Zou, L., Liu, B. and Huang, X.: Boost Blockchain Broadcast Propagation with Tree Routing, *Smart Blockchain*, Qiu, M. (Ed.), Cham, Springer International Publishing, pp.77–85 (online), DOI: 10.1007/978-3-030-05764-0_8 (2018).
[6] Jin, M., Chen, X. and Lin, S.-J.: Reducing the Bandwidth of Block Propagation in Bitcoin Network With Erasure Coding, *IEEE Access*, Vol.7, pp.175606–175613 (online), DOI: 10.1109/ACCESS.2019.2957496 (2019).
[7] Aoki, Y., Otsuki, K., Kaneko, T., Banno, R. and Shudo, K.: SimBlock: A Blockchain Network Simulator, *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops* (*INFOCOM WKSHPS*), pp.325–329 (online), DOI: 10.1109/INFCOMW.2019.8845253 (2019).
[8] Otsuki, K., Shudo, K. and Banno, R.: Effects of Relay Networks in Bitcoin, *IEICE Technical Report*, NS2019-192, Vol.119, No.460, pp.89–94 (2020).

Yusuke Kitagawa received his B.S. degree in Information Communication Engineering from Kogakuin University, Tokyo, Japan, in 2021. He is currently a master's course student at the Graduate School of Engineering, Kogakuin University. His research interest is on blockchain networks. He is a student member of IEICE.

Kazuyuki Shudo received his B.E. degree in 1996, his M.E. degree in 1998, and his Ph.D. degree in 2001 all in computer science from Waseda University. He worked as a Research Associate at the same university from 1998 to 2001. He later served as a Research Scientist at National Institute of Advanced Industrial Science and Technology. In 2006, he joined Utagoe Inc. as a Director, Chief Technology Officer. From December 2008, he served as an Associate Professor at Tokyo Institute of Technology. Since April 2022, he currently serves as a Professor at Kyoto University. His research interests include distributed systems and language systems. He is a member of IEEE, ACM, IPSJ and IEICE.



Osamu Mizuno received B.S. and M.S. degrees in Applied Electronics Engineering from Tokyo Institute of Technology in 1983 and 1985, respectively. He received a Ph.D. in Global Information Telecommunication from Waseda University, Tokyo in 2008. From 1985–2009, he worked in research and development of network service systems for the Nippon Telegraph and Telephone Corporation. He joined Kogakuin University, Tokyo, in 2009. His research interests include IoT service systems and ID distribution. He is a member of IEEE, IEICE, IPSJ and IEEJ.

Ryohei Banno received his B.E. and M.I.S.T. from Hokkaido University in 2010 and 2012, respectively, and his Ph.D. in Science from Tokyo Institute of Technology in 2018. From 2012 to 2018, he was a researcher at NTT Network Innovation Laboratories. From 2018 to 2020, he was a researcher at Tokyo Institute of Technology. He joined Kogakuin University as an Assistant Professor in 2020. Since 2022, he has been an Associate Professor at Kogakuin University. His research interests include distributed systems and IoT systems. He is a member of ACM, IEEE, IEICE, and IPSJ.