Security and Performance Evaluation of Master Node Protocol in the Bitcoin Peer-to-Peer Network

Muntadher Sallal
Nottingham Trent University
School of Computing and Technology
E-mail: muntadher.sallal@ntu.ac.uk

Gareth Owenson
University of Portsmouth
School of Computing
E-mail: Gareth.Owenson@port.ac.uk

Mo Adda
University of Portsmouth
School of Computing
E-mail: Mo.Adda@port.ac.uk

***Abstract*—This paper proposes a proximity-aware extensions to the current Bitcoin protocol, named Master Node Based Clustering (MNBC). The ultimate purpose of the proposed protocol is to evaluate the security and performance of grouping nodes based on physical proximity. In MNBC protocol, physical internet connectivity increases as well as the number of hops between nodes decreases through assigning nodes to be responsible for propagating based on physical internet proximity.**

***Keywords*—Bitcoin network, Propagation delay, Clustering**

1. Introduction

As the Bitcoin network topology is not proximity defined, c onnecting t o o ther p eers i s m aintained randomly without considering any proximity criteria. In other words, long-distance links are not taken into consideration when the Bitcoin physical network topology is built. This increases non-compulsory hops that the information passes through. In addition, as it is mentioned in [1], the sheer distance between the origin of a transaction or block and other nodes is deemed as the most significant problem in the Bitcoin network. As a result, transaction verification process i s slower [2]. Hence, the potential of double spending attacks, that are more difficult to discover in a slow network, increases due to the conflict between nodes regarding the transactions history. Uncertainty regarding the validity of a given transaction causes the blockchain forks where a transaction can appear in two different branches of the blockchain [3].

Therefore, the propagation delay between nodes in the Bitcoin network is critical even though the probability of reaching an agreement about transaction history is high [4]. Aiming at evaluating security and performance of proximity based clustering in the Bitcoin network thereby reduce the possibility of double spending attacks, this paper proposes and implements a clustering protocol, named as Master Node Based Clustering (MNBC). MNBC protocol relies on several nodes, known as master nodes, to achieve fully connected clusters based on the physical Internet proximity and random peers selection. MNBC is implemented in a distributed manner where all nodes contribute in achieving a proximity based network layout. This prevents any node having control over the network as there is no node that would have a full knowledge of the entire network topology.

The rest of the paper is organized as follows. In Section 2, related work in measuring and analysing Bitcoin information propagation and in modelling approaches to avoid double spending attacks will be outlined. Section 3 details the proposed clustering protocols MNBC with reference to the clusters generation and clusters maintenance. In Section 4, performance evaluation of the proposed clustering protocols regarding speeding up the transaction propagation delay are performed. Section 5 evaluates the potential of partition attacks in the proposed protocol. We conclude the work in Section 6.

2. Related Work

Measurements of the probability of double spending attacks based on measurements in the real Bitcoin network have been provided in [1] through developing an analytical model of the Bitcoin system. A model that considers some modifications in the transaction dissemination protocol in the Bitcoin network has been presented in [5]. The core idea of this model is that a node can add a received transaction to its pool and forwarded it to other nodes if the received transaction has not been seen before. Otherwise, the node directly forwards the transaction to other neighbours without adding it to its pool. This scenario allows a fake transaction to be received by the node that issues the original transaction. Though, the initiator node would immediately detect the attempted of a double spending attack when the fake transaction is received. The most serious disadvantage of this method is that a large volume of nonessential traffic would flood the network which results in inefficient performance of the Bitcoin network.

As introduced in [2], faster information propagation can be achieved by pipelining information dissemination in order to minimize the round-trip times between nodes and their neighbours. This solution claims that incoming INV message which includes a list of hashes of the available transactions, can be immediately forwarded instead of waiting to receive transactions. Therefore, nodes can ask for a transaction even though it has not arrived yet. On receiving the transaction, it will be forwarded immediately for nodes

that have asked for it, considering that a GETDATA message has already been received from those nodes. By doing this, the idle time in which nodes are normally waiting for the GETDATA message to arrive, would be utilized. However, we believe that this theory might reduce the propagation delay with a very low rate as the transaction still needs to pass through random and unlocalized connections to visit most of the Bitcoin network nodes. Another change has been proposed in the same study that is closely related to what we are presenting in this paper. This change increases the geographical connectivity in Bitcoin network through several coordinator nodes, distributed strategically around the globe. These nodes are able to search and suggest Bitcoin network nodes to each other. The main downside of this solution is that it is relatively centralized.

3. Master Node Based Clustering Protocol: Concept and Implementation

Master Node Based Clustering protocol (MNBC) extends the BCBSN protocol that was proposed in our previous work [6], with the aim of addressing security and performance limitations of BCBSN protocol. As it is mentioned in [6], the BCBSN protocol aims to generate a set of geographically diverse clusters in the Bitcoin network by exploiting super peers technology. Within each cluster, the BCBSN protocol assigns one node to be a super peer that is responsible for maintaining the cluster and broadcasting information in the Bitcoin network. In the BCBSN protocol, clusters are fully connected via super peers only. Due to this, the information flow between clusters in the BCBSN protocol is only fulfilled through super peers. Furthermore, super peers in the BCBSN protocol group peers based on their geographical location in order to increase the proximity of connectivity in the network. However, long-link distance might be applied between any two peers even though they are in the same geographical location. The node selection in BCBSN protocol is not random, instead, the node is forced to connected to the list of nodes that supplied by the super peer that the node connects to. From a security point of view, the level of security awareness in the BCBSN protocol can be improved if more nodes between clusters are maintained as well as random selections of peers which is important in the Bitcoin network is preserved. This improves the network resistance against the partitioning attack as well as eclipse attack.

The limitations of BCBSN protocol mentioned above have motivated the development of a Master Node Based Clustering(MNBC) which relies on several nodes, known as master nodes, to achieve fully connected clusters based on the physical Internet proximity and random peers selection. The idea of the MNBC protocol is inspired by the Master node technology that was originally adopted in [7]. However, master nodes in Darkcoin were responsible only for propagating the network information to the majority of nodes without taking into account whether or not those nodes are close.

3.1. Master Node Selection

Master node role requires gaining a score which is calculated based on how much each node burns bitcoins and how long a node has been online. The main advantage of this approach is that, impersonation of a master node by a malicious node would be challenging. Therefore, this score helps in electing master nodes that are better suited for that role. To To incentivize nodes to compete towards winning the master node's role, as it has proven in [8], a reward is given for a master node when it propagates a valid transaction and behaves honestly. When a particular node achieves the best score over other nodes in the network, as it is illustrated in Algorithm 1, the node is elected as a master node.

Algorithm 1: Master node score calculation algorithm

```
   Let M as: Master nodes set in the network
   Let z as : Best master node score to achieve
 1 while M ≠ 0 do
 2     for master node in M do
 3         n ← masternode.CalculateScore()
 4         if n > z then
 5             z = n
 6             winning − node ← masternode
 7             Exit()
 8         end
 9     end
10 end
```

When a particular peer wants to occupy the role of master nodes, the peer invites other peers that connect to it by propagating two types of messages a $masterINV$ and an $AcceptINV$. Consider a node M decides to be a master node and a peer P receives a $masterINV$ from M. On receiving of the $masterINV$ message, as illustrated in Algorithm 2, the node P accepts $M's$ invitation if it finds the node M to be closer in the physical internet and has a bigger weight than the master node that P is connected to. The node P decides whether or not M is close in the physical internet by calculating the internet distance based on ping latencies, following the same methodology that has been adopted in [9] to measure the physical internet distance between peers. The node P accepts $M's$ invitation by sending $AcceptINV$. The node P should keep forwarding the $masterINV$ to all its connected nodes which in turn will propagate the $masterINV$ further.

3.2. Cluster Maintenance

In order to increase the network resistance to eclipse attack or partition attack, peer selection in MNBC reserves the idea of random selections of peers which is important in the Bitcoin network. Specifically, peers in MNBC protocol select other peers based on a combination of factors of physical proximity (link latency) and random

Algorithm 2: On receiving masterINV do

Let M **as :** nearest master node() with Bigger weight
Let mp **as:** current master node
if $mp \neq M$ **then**
  $mp = M$
  $connectTo\ (mp)$
  $Forward\ (masterINV)$
else
  $Do\ Nothing$
  $Forward\ (masterINV)$
end

selection. Let $R\{n_0,n_1,....,n_{i-1}\}$ be a set of peers in the Bitcoin network, where i is the number of total peers. Let $M\{mp_0, mp_1, ..., mp_{j-1}\}$ be a set of master nodes, where j is the number of master nodes and $M \subseteq R$, Let $mp_l\{mp_l, b_0, b_1, ..., b_{k-1}\}$, $(l = 0, 1, ..., j-1)$ and k is the number of peers in the cluster, mp_l be a set of peers in the lth cluster. Therefore, we have $mp_l \subseteq R$ and $R = mp_0 \cup mp_1 \cup ... \cup mp_{j-1}$. When a node z wants to join the Bitcoin network, it first learns about the available master nodes by contacting an arbitrary node T which already have been learnt from DNS service. The node T responds with a list of the master nodes it knows about in the network. The node z selects a master node mp_i such that $\forall mp_j \in M, distance(z, mp_i) \leq distance(z, mp_j)$. Then, the node z sends a $JoiningRequest$ message to the selected master node. Note that the distance is also calculated based on the link latency, following the same methodology that has been adopted in our previous work [9].

After that, the node z learns about the available Bitcoin nodes from a list of DNS services, where a list of random nodes is supplied. The node z calculates the distance to each node in the supplied list in order to get its proximity ordering based on a link latency threshold. This ordering would help the node z to be directed to a specific cluster. After that, the node z tries to connect to a node k which is the closest node in the nodes list that is supplied by the DNS service. However, the role of the DNS service stops once the node z connects to the node k. Periodically, the node z discovers other nodes in the network using the Bitcoin network nodes discovery mechanism [5], where the discovered nodes are the nodes that is supplied by either DNS or the normal Bitcoin network nodes discovery mechanism. Then, the node z finds out whether the discovered nodes are physically close by following the physical distance calculation algorithm mentioned in our previous work [9]. When the node z wants to leave the network, it sends a disconnect message to its master node, which requires no reply. Then, the node mp_i should update its nodes list automatically.

As mentioned before, clusters are fully connected by their edge nodes and master nodes. Therefore, edge nodes will be selected between every pair of clusters. Specifically, let $S = s_1, s_2, ..., s_m$ and $R = r_1, r_2....., r_n$ represent two clusters, and let $[s_b, r_b]$ denote their border nodes, where $s_b \in S$ and $r_b \in R$, then for all other pairs of clusters (such that $s_i \neq s_b$, $r_j \neq r_b$, $s_i \in S$, $r_j \in R$), *distance*$(s_i, r_j) \geqslant$ *distance*(s_b, r_b). Note that *distance*(x, y) represents the physical internet distance between the two nodes x and y in the network.

4. Performance Evaluation

The information propagation delay is considered as the main performance metric in the evaluation of the proposed protocol. We simulate our solution on an event based simulator that has been built in [6]. we designed an event-based simulation model that is based on Bitcoin protocol specification and measurements of real Bitcoin network. Integration of the Bitcoin protocol based on Bitcoin client behavior(*bitcoind*) as well as measurement of the conditions in the real Bitcoin network are modelled in order to make the simulator behave as closely to the real Bitcoin network as possible. Measurements of link latencies between peers in the real Bitcoin network, which were collected in our previous work [6], are fixed in the simulation model as the information propagation delays are the main aspect that we concentrate on which requires an accurate estimate of link latencies between Bitcoin network peers. The expensive cryptographic operations are abstracted in the proposed model, aiming to allow full scale experiments of the Bitcoin network.

4.1. Experiments setup

After getting some proximity based clusters in every simulation scenario, normal Bitcoin simulator events will be launched. Within the proposed protocol, we implemented a measuring node c which is able to create a valid transaction Tx and send it to one node of its connected nodes, and then it tracks the transaction in order to record the time by which each node of its connections announces the transaction. In other words, the transaction is propagated from node c to one connected node only. Then node c records the latency by which all cs' connected nodes would receive the transaction. Suppose the client c has proximity based connections (1,2,3,...., n), c propagates a transaction at time T, and it is received by its connected nodes at different times $(T_1, T_2, T_3, ..., T_n)$. The time differences between the first transaction propagation and subsequent receptions of the transaction by connected nodes were calculated $(\Delta t_{c,1}, ..., \Delta t_{c,n})$ according to equation(1):

$$\Delta t_{c,n} = T_n - T_c \tag{1}$$

Where $T_n > T_{n-1} >, ..., T_2, T_1$. However, the latency is determined by an average of approximately 1000 runs in order to increase the accuracy of the collected latencies which might be affected by several factors such as data corruption and loss of connection.

The simulations were performed on a 2,6 GHz machine equipped with 64 GB of main memory; simulation of 10 hours at real-world scale of around 7,000 peers required 29 hours of computation time and 22 GB of memory.

4.2. Results and discussions

The simulation results show that the proposed protocol offers an improvement in propagation delay compared to the Bitcoin protocol. Fig.1 compares the distributions of $\Delta t_{c,n}$ for the simulated Bitcoin protocol against the same distributions that have been measured in the simulated proposed protocol MNBC. In the figure, the number of connected nodes represents the sequence of the random nodes that the measuring node connects with at each run. Regarding the comparison between the MNBC and Bitcoin protocol, the Bitcoin protocol performs variances of delays, which have been collected in our prior work [6], that grow linearly with the number of connected nodes, whereas MNBC maintains lower variances of delays regardless of the number of connected nodes. The reduction of the transaction propagation time variances in the proposed protocol has to do with the fact that the Bitcoin network layout in which nodes connect to other nodes without taking advantage of any proximity correlations results in a long communication link cost measured by the distance between nodes. Consequently, the average delay to get transactions delivered is also increased which, on the other hand, would affect the consistency of the public ledger. On the other hand, maintaining clusters, which are fully connected via master nodes and edge nodes, based on physical internet proximity implies faster transaction propagation in the MNBC protocol. In fact, contrary to what previously thought in this area, we found that reconstructing the Bitcoin network layout on proximity bases implies faster transmissions.

Turning now to the comparison between MNBC protocol and BCBSN Protocol. As shown in Fig.1, both proposed protocols show relatively same variances of delays over nodes 1,2,3,4,5 and 6. From node 7, variances of delays in BCBSN protocol started climbing steadily and reached

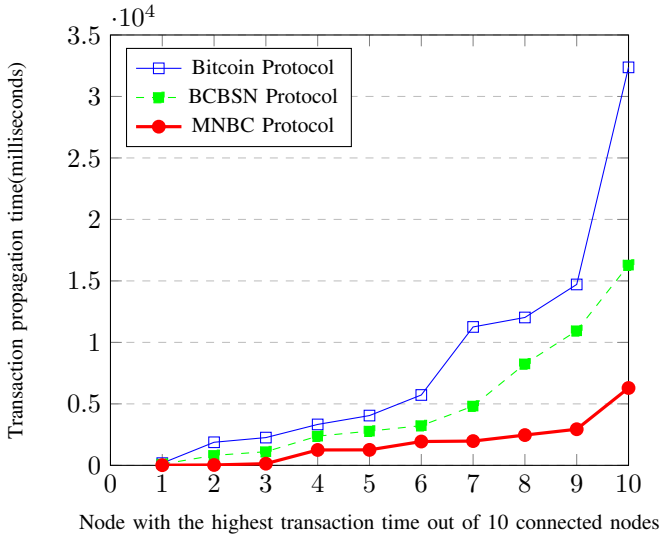

Figure 1: Comparison of the distribution of $\Delta t_{c,n}$ measured in the simulated Bitcoin protocol with MNBC protocol and BCBSN Protocol simulation results.(d_t in MNBC=25ms)

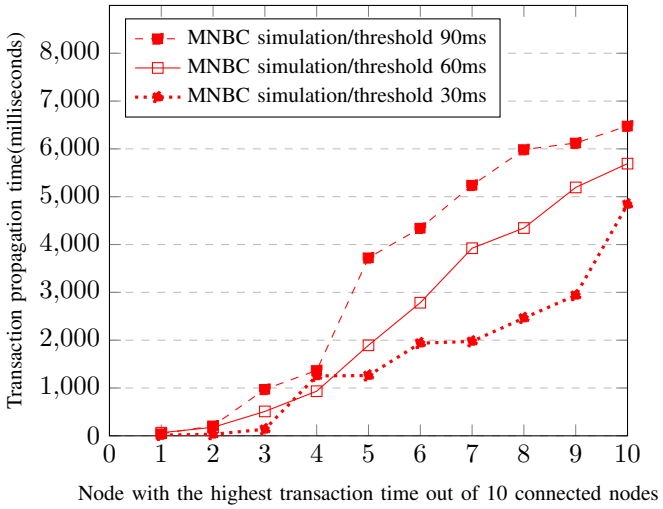

Figure 2: Comparison of the distribution of $\Delta t_{c,n}$ as measured in the simulated BCBPT protocol with three thresholds (d_t =30ms, 60ms, 90ms)

a peak over node 10 recording transaction propagation delay nearly 18000ms.

On the other hand, the variances of delays were totally improved in the MNBC protocol over the Bitcoin and BCBSN protocol, especially at nodes 8,9, and 10. The most likely cause of the higher variances of delays in the BCBSN protocol is the fact that the information flow between clusters in BCBSN protocol can only be maintained through supers peers. This causes lack of transformation channels between clusters which results in inefficient information distribution over the network. The lack of connections between clusters in BCBSN protocol has been tackled in MNBC protocol by considering the edge nodes technology which adds an extra connection channels between cluster. Therefore, faster information propagation has been achieved in MNBC compared to BCBSN.

In MNBC, the comparison among three variances of delays was done based on three different latency suggested thresholds 30 ms, 60 ms, and 90 ms. Results that are shown in Fig.2 reveal that the less latency distance threshold in MNBC performs less variance of delays. The key reason of variances of delays have been declined when the threshold value is reduced is that the number of nodes at each cluster is minimised due to the limited coverage of the physical topology which is offered by d_t.

5. Security Analysis

Eclipse attack: The proposed protocol ensures that performing eclipse attack is challenging as the selection of peers is carried out using combination of physical proximity and random selection. However, increasing the number of outgoing connection at each node would further improve the resistance of the network against the eclipse attack.

Partition attack: We assume that the botnet is able to also perform DDoS attacks on a limited number of peers in

the Bitcoin peer-to-peer network. According to our attack model, the attack will be performed within three phases. The first phase starts when several malicious nodes which belong to an attacker join the peer-to peer Bitcoin network and connect to many honest nodes. To ensure that attacker nodes connect to as many honest nodes as possible, only IP addresses of attacker nodes are announced by other attacker nodes. Once the attacker guarantees that satisfied number of connections to honest nodes were maintained and the connectivity graph is thinned out, a proximate snapshot of the network graph layout will be given by launching the second scenario of the attack. This scenario can be achieved through a probabilistic method which has been introduced in [10]. By this method, the Bitcoin network topology can be learnt with a reasonable probability through indicating whether or not two peers in the network are connected by sending marker addresses and observing the flow of these addresses. By doing so, the attacker will be able to indicate the minimum vertex cut of the network. Minimum vertex cut is defined as minimum honest peers that removing them causes splitting the graph into at least two partitions [11]. When the attacker selects peers for minimum vertex cut, denial-of-service (DDOS) attack will be performed on the selected peers. However, our partition attack evaluation will be based on minimum vertex cut, as a metric to indicate the cost of performing partition attacks. This metric has been used in [12] to evaluate partition attacks in the Bitcoin network protocol.

5.1. Experiment setup

We developed four experiment scenarios with differnt network sizes (2000,4000,6000, and 8000). The size of the attacking botnet was chosen to match the number of honest peers in each scenarios. The first phase of the attack starts when the network topology is restructured according to each protocol of the proposed protocols. Specifically, several attacker nodes join the network and start establishing connections to many honest nodes. As we based our partition attack evaluation on *minimum vertex cut* as a cost metric, *minimum vertex cut* of the network topology is determined at regular intervals using *metis* graph partition toolkits [13]. $Metis$ algorithm can achieve a balance partitioning that minimizes either the communication volume or number of edge cut.

5.2. Results and discussions

Fig.3 shows the results of three simulated attacks on a model of the real Bitcoin network, MNBC, and BCBSN protocol. Each attack was launched based on different network sizes (2000,4000,6000, and 8000). The configuration of the desired imbalance factor was done in a way that the largest partitions does not include more than 60% of all nodes.

In the small scenarios with number of nodes (2,000 and 4,000), the number of honest peers in the minimum vertex cut in all protocols after launching the partition attack stayed below 500 which reveals that all protocols are relatively

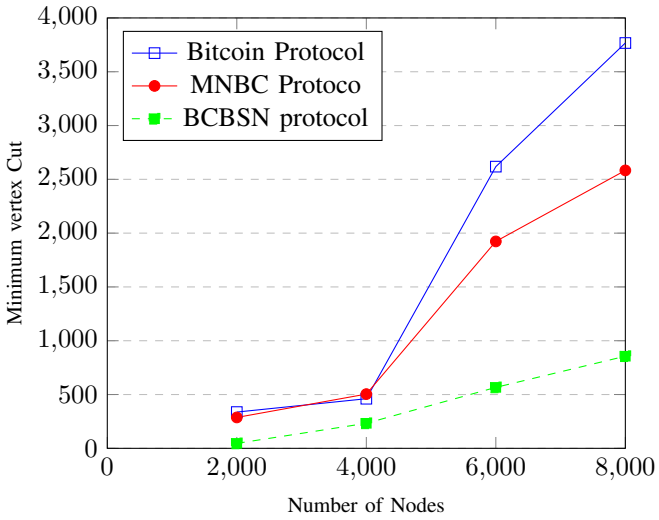

Figure 3: Number of honest peers on the minimum vertex cut

similar in terms of resistance against partition attacks. While in the large scenarios with 6,000 and 8,000 peers, the level of resistance against partition attacks increased in all protocols as the number of nodes increases. The highest level of resistance is experienced in the Bitcoin protocol, while the lowest level is appeared in BCBSN protocol. Precisely, the minimum vertex cut in the Bitcoin protocol increased from around 500 to 3,800 with the scenario of 8,000 peers resulting a notable gap in the minimum vertex cut between the Bitcoin protocol and other protocols. Whereas, MNBC protocol shows a higher resistance against partition attacks over the BCBSN protocol, where the number of honest nodes in the minimum vertex cut goes above 2,500 in the scenario of 8,000 nodes. BCBSN protocol is considered as the worst protocol of the proposed protocols in terms of the ease of performing partition attacks as it showed the lowest minimum vertex cut in both large and small scenarios. Although the MNBC and BCBSN protocols show less minimum vertex cut compared to the Bitcoin protocol, the number of honest nodes required to cut in the proposed protocols is still high which needs significant resources to be considered. As expected, clusters in the MNBC that are fully connected via master nodes and edge nodes reflect less number of honest nodes in minimum vertex cut. While, clusters in BCBSN that are connected via super peers result in number of nodes in the area of minimum vertex cut going down.

Fig.4 shows the results of the simulated partition attacks on a model of the real Bitcoin network, MNBC, and BCBSN protocol, captured within different session lengths. Within 24 hours of attack, the number of nodes in the minimum vertex cut declined in the simulated real Bitcoin network as well as the MNBC and BCBSN protocol as follows: the minimum vertex cut declined from around 3,700 to 1,500 in the real Bitcoin network. The same scenario happened in the MNBC where the minimum vertex cut decreased

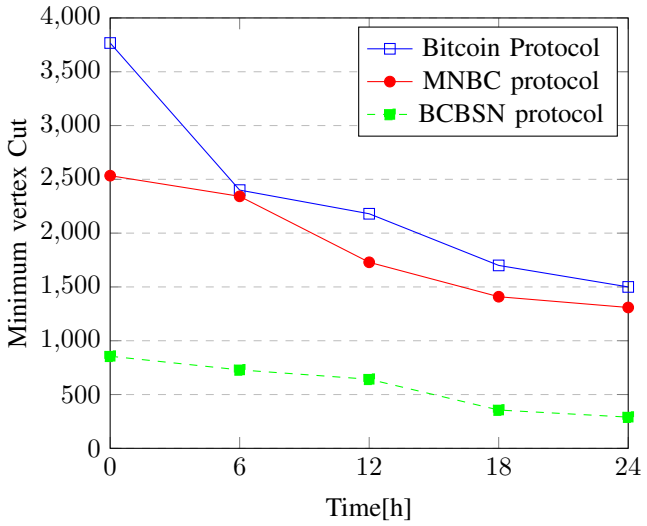

Figure 4: Number of non-attacker peers on the minimum vertex cut during an attack with 7,000 honest peers parametrized as in the real-world network, attacker's session length $S_A = 6h$

from around 2500 to 1150. Similarly, minimum vertex cut dropped down from 850 to 290 in the BCBSN protocol. It can also be seen that the simulated real Bitcoin network still performs better than the MNBC and BCBSN protocol in terms of the resistance to partition attacks. However, it can be concluded from the obtained results that more patience from the attackers with a higher number of peers, the better chances of success in splitting the network.

The simulation were performed with 7,000 honest peers parametrized as in the real-world network, attacker's session length $S_A = 6h$.

6. Conclusion

By conducting extensive simulations, MNBC evaluation results indicate an improvement in the transaction propagation delay over the Bitcoin network protocol. However, MNBC maintains lower variance of delays over the BCBSN protocol. Furthermore, evaluation of partitioning attacks in the Bitcoin network as well as the MNBC and BCBSN protocol was presented in this paper. Results revealed that the Bitcoin network is more resistant against attackers than the proposed protocols. However, attackers still need more resources to split the network in the proposed protocols especially with a higher number of nodes.

References

[1] Decker, C., & Wattenhofer, R. (2013, September). Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference* on (pp. 1-10). IEEE.

[2] Stathakopoulou, C.(2015).A faster Bitcoin network. Tech. rep., ETH, Zurich,. SemesterThesis, supervised by Decker.C and Wattenhofer.R.

[3] Sompolinsky, Y., & Zohar, A. (2013). Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains. IACR Cryptology ePrint Archive, 2013, 881.

[4] Miller, A., LaViola Jr, J.J.: Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin (2014).

[5] Karame, G. O., Androulaki, E., Roeschlin, M., Gervais, A., & Èapkun, S. (2015). Misbehavior in Bitcoin: A Study of Double-Spending and Accountability. In *ACM Transactions on Information and System Security (TISSEC)*, 18(1), 2.

[6] Fadhil, M., Owenson, G., & Adda, M. (2016). A Bitcoin model for evaluation of clustering to improve the transaction propagation delay in Bitcoin network. *In 19th IEEE International Conference on Computational Science and Engineering*. Paris.

[7] Duffield, E., Schinzel, H., Gutierrez, F. (2014). Transaction locking and masternode consensus: A mechanism for mitigating double spending attacks.

[8] Babaioff, M., Dobzinski, S., Oren, S., & Zohar, A. (2012, June). On bitcoin and red balloons. In Proceedings of the 13th ACM conference on electronic commerce (pp. 56-73). ACM.

[9] Sallal, M., Owenson, G., Adda, M. (2017, July). Proximity awareness approach to enhance propagation delay on the Bitcoin peer-to-peer network. In 37th IEEE International Conference on Distributed Computing Systems. IEEE.

[10] Biryukov, A., Khovratovich, D., & Pustogarov, I.(2014). Deanonymisation of clients in Bitcoin P2P network. arXiv Preprint arXiv:1405.7418.

[11] Ugurlu, O., Berberler, M. E., Kızılates, G., Kurt, M. (2012, September). New algorithm for finding minimum vertex cut set. In Problems of Cybernetics and Informatics (PCI), 2012 IV International Conference (pp. 1-4). IEEE.

[12] Neudecker, T., Andelfinger, P., & Hartenstein, H. (2015). A simulation model for analysis of attacks on the Bitcoin peer-to-peernetwork. In Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on (pp. 1327-1332). IEEE.

[13] Karypis, G., & Kumar, V. (2016). Metis - Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0. 1995.

[14] Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015). Eclipse Attacks on Bitcoin's Peer-to-Peer Network. *In: 24th USENIX Security Symposium(USENIX Security 15)*, Washington, D.C., USENIX Association.

[15] Karame, G. O., Androulaki, E., & Capkun, S. Double-spending fast payments in bitcoin.In *The 2012 ACM conference on Computer and communications security*, pages 906–917.ACM, 2012.