View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Portsmouth University Research Portal (Pure)

# Proximity Awareness Approach to Enhance Propagation Delay on the Bitcoin Peer-to-Peer Network

Muntadher Fadhil; Gareth Owen; Mo Adda
University of Portsmouth, Buckingham Building, Portsmouth, United Kingdom
{Muntadher.sallal; Gareth.owen; Mo.Adda}@port.ac.uk

***Abstract*— In the Bitcoin system, a peer-to-peer electronic currency system, the delay overhead in transaction verification prevents the Bitcoin from gaining increasing popularity nowadays as it makes the system vulnerable to double spend attacks. This paper introduces a proximity-aware extension to the current Bitcoin protocol, named Bitcoin Clustering Based Ping Time protocol (BCBPT). The ultimate purpose of the proposed protocol, that is based on how the clusters are formulated and the nodes define their membership, is to improve the transaction propagation delay in the Bitcoin network. In BCBPT, the proximity of connectivity in the Bitcoin network is increased by grouping Bitcoin nodes based on ping latencies between nodes. We show, through simulations, that the proximity base ping latency defines better clustering structures that optimize the performance of the transaction propagation delay. The reduction of the communication link cost measured by the information propagation time between nodes is mainly considered as a key reason for this improvement. Bitcoin Clustering Based Ping Time protocol is more effective at reducing the transaction propagation delay compared to the existing clustering protocol (LBC) that we proposed in our previous work.**

***Keywords—Bitcoin; Propagation delay; Clustering Evaluation***

## I. INTRODUCTION

Bitcoin is a digital currency based on cryptography which allows payments between two parties without involving any trusted issuing entity or financial institution. Instead, Bitcoin bases on a peer-to-peer network with peers minting Bitcoin [1]. Bitcoin is currently being integrated across a number of exchange markets and businesses. Bitcoin is considered as a reliable currency which allows global transactions to be processed as fast as local ones. In addition, it offers a public history of all transactions that have ever been processed. It also introduces such new payment strategies, such as micropayment, contract, and escrow transactions [2].

Bitcoin follows a distributed trust mechanism which relies on distributed validation and tracking of transactions. Based on this mechanism, a Bitcoin transaction has to be broadcasted to all nodes within the network to reach a consensus about which transactions are valid. The consensus is recorded in a publicly distributed ledger which is shared by the entire network. This ledger, which is known as block chain, includes all the valid transactions that have ever been processed grouped into blocks. Every block is linked with previous blocks by including the unique hash of the previous block in its header. The first block in the block chain is known as the genesis block and it has no references to previous blocks. Some branches which are a path in the block chain that start from a leaf block to the genesis block are experienced in the block chain [2].

Regarding Nakamoto [1], the key idea of the Bitcoin system is that, it is designed in a way that prevents simultaneous double spending of Bitcoins. Due to this benefit, a trusted third party is not required. Unfortunately, in some cases the Bitcoin system can lose this benefit and spending the same Bitcoin twice would be a potential [4]. The main cause of this issue is the delay overhead in the transaction verification process which is considered as the main challenge that must be faced in this Bitcoin system. Within this process, transactions must be verified by all participants (nodes) in the Bitcoin system in order to achieve an agreement regarding a common transactions history. Alas, achieving this agreement with a high probability is a bit challenging as the Bitcoin network is not well synchronized. Therefore, the replicas of the ledger that every node within the network keeps are inconsistent. This would raise an argument between nodes regarding the transaction history due to the fact that transactions are validated against the public ledger which is inconsistent. This argument causes uncertainty regarding the validity of a given transaction which may cause a situation called block chain fork in which not all nodes agree on the same block chain header. In other words, two blocks are possible to be created simultaneously, each one as a possible addition to the same sub-chain. These two blocks are mutually conflicting even they might be consistent with the history. Consequently, a transaction can appear in two different branches of the block chain, a fact that has been noted by [3]. As a consequence of this conflict, an attacker has an opportunity to perform a double spending attack by which a Bitcoin can be spent twice.

According to [4], this issue can be avoided if transactions are propagated quickly enough through the network as accelerating the transaction propagation would tackle the problem of the agreement on a common transaction history among nodes in the Bitcoin network. This would result in

reducing the probability of performing a successful double spending attack.

Regarding issues that are mentioned above, this paper presents Bitcoin Clustering Based Ping Time protocol (BCBPT), an efficient solution for tackling the problem of the transaction propagation delay. BCBPT aims to increase the proximity of connectivity among nodes in the Bitcoin network based on round-trip ping latencies. Currently in the Bitcoin network, a node connects with nodes regardless of any proximity criteria. In contrast, the core of our solution is to get nodes to gain more information about the proximity of other nodes, thus, enabling them make a better decision on which nodes to connect with. Based on the simulation model that was presented in our previous work [5], BCBPT evaluation results are presented in this work. The evaluation of BCBPT is done based on whether or not the BCBPT protocol is able to speed up the transaction propagation delay. In addition, we perform a comparison based on the transaction propagation delay between the BCBPT protocol and LBC protocol that has been presented in our previous work [6]. However, finding out the best mechanism which is able to speed up the propagation delay in the Bitcoin network is deemed to be the main intended impact of this research.

The paper is organised as follows: in Section II, related work in measuring and analysing Bitcoin information propagation and in modelling approaches to avoid double spending attacks will be outlined. Section III focuses on giving an overview of the Bitcoin system and briefly describing the Bitcoin networking aspects. In addition, we discuss in details the information propagation in the Bitcoin network and analyse the double spending attack which is caused by the transaction propagation delay. Section IV details the proposed clustering protocol (BCBPT) with reference to the clusters generation and clusters maintenance. In Section V, BCBPT's protocol evaluation results regarding speeding up the transaction propagation delay is performed. Furthermore, a comparison between the BCBPT protocol and LBC protocol is provided based on the transaction propagation delay. In Section VI, we conclude the paper and discuss the future work.

## II. Related work

The problem of the delay overhead in information propagation in the Bitcoin field has gained more research attention recently. As this problem is linked to the problem of reaching a consensus in the Bitcoin network, it is classified as a part of the Byzantine Fault Tolerance which is aiming to keep a system working regardless of Byzantine failures. The potential of reaching a Byzantine consensus in a synchronous system regardless of the number of faulty participants has been proved in [7]. Under the umbrella of Bitcoin, it has been discovered that, except for a negligible probability when Byzantine faults make up less than half the network, the Bitcoin protocol can reach a consensus [8]. Decker and Wattenhofer [9] observed that the round trip time of the transactions verification process causes the propagation delay in the network. Therefore, a solution was proposed in their research which then has been adopted by [10]. This solution claims that the information propagation could be pipelined instead of waiting to receive the transaction. In other words, any node can immediately forward an invitation message (INV Message) that includes a list of hashes of available transactions, rather than waiting to receive transactions. However, the transaction verification time still remains inefficient due to the size of the public ledger. Some possible double spending attacks have been discussed in [2]. Their work ignores attacks that can be done by offering a significant hash-rate, which can compute alternative chains on its own. Instead, they focus on attacks on nodes that accept zero-confirmation transactions during fast payments. Several counter-measurements to avoid these attacks have been suggested in the same work.

A prototype system which is applied in vending machines has been proposed in [11] in order to mitigate double spending attacks. This system has performed a fast payment with 0.088% as a probability of double spending attacks by using a server that observes transactions. This server gives an approval regarding the transaction confirmation once a transaction is propagated and reached over 40 nodes. This solution doesn't mitigate double spending attacks significantly as attacker's transaction could still be propagated to the majority of nodes. A model which considers some modifications in the transaction dissemination protocol has been presented in [4].

## III. Bitcoin protocol and information propagation

Bitcoin protocol achieves the distributed validation based on a replicated ledger that is collectively implemented by network voluntaries. This ledger tracks the address balances of all users. An arbitrary number of addresses can be created by each user to send and receive Bitcoins. An ECDSA key pair is used to prove the ownership of Bitcoins associated with that address. Each entry in the public ledger represents a transaction which is a signed data structure that is created by a Bitcoin user who intends to send a specific Bitcoin to one or more destination accounts [3]. Transactions are responsible for claiming some Bitcoins that are associated with the address of the sending party and reassigning them to the address of receiving part/parties. Transactions are represented by a hash of the previous transaction that has been sent before as well as the public key of the future owner. Each transaction includes input and output. For combining or splitting Bitcoins, transactions can handle multiple inputs and outputs. Inputs reference the funds from other previous transactions, whereas outputs indicate the transferred Bitcoins. A transaction output also indicates the new owner of the transferred Bitcoins when it is referenced as an input in a future transaction. Though, the balance of an account is the sum of all values of all unspent outputs owned by that account. The sum of all outputs should be equal or less than the sum of all inputs [4].

Due to the purpose of keeping the chronological order of the valid transactions across nodes, every valid transaction should be included in a block that forms a part of the ledger. Each block is connected with an earlier block through the earlier block's hash which must be included in the block's header. As an exception, only one block, known as the genesis block, cannot reference an earlier block. In order to acknowledge a group of transactions in a block, computational effort must be spent. Therefore, a transaction is added to the ledger once sufficient work is done to acknowledge the block that contains it. This computational effort is provided by special nodes that

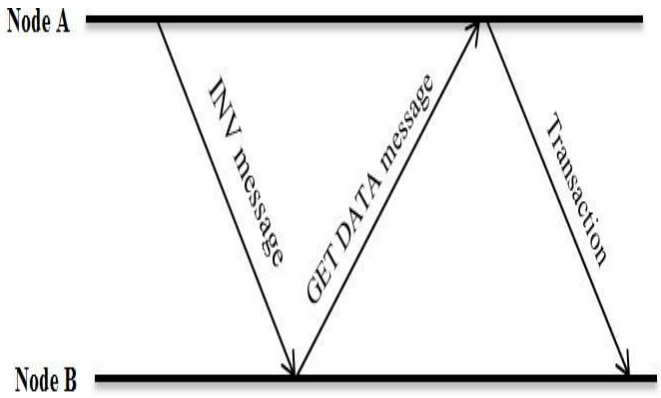


**Fig.1: Transaction propagation protocol between Nodes A and B**

are known as miners [11]. Blocks and transactions are broadcasted in the entire network in order to synchronize the replicas of the public ledger across all nodes. On receiving a new transaction, a peer checks whether the Bitcoin has been previously spent in the block chain, and whether the transaction is correctly formed. Once a transaction has been verified by a peer, as shown in Fig.1, it announces the transaction availability to nodes by propagating an INV(Inventory) message that contains the hash of the transaction. This propagating scenario is followed in order to avoid sending a transaction to a node that already receives it from other nodes. On receiving an INV message, a node would request a transaction if it has not seen before. Requesting a transaction is fulfilled by sending a GETDATA message. Responding to the received GETDATA message, a node sends the transaction's data. Alas, the transaction broadcasting scenario causes a delay in transaction dissemination which, on the other hand, affects the Bitcoin network scalability. This affection is represented by making the public ledger inconsistent. However, inconsistency of the public ledger would encourage attackers to successfully perform double spending attacks [2].

## IV. PROXIMITY BASED PING TIME PROTOCOL: CONCEPT AND IMPLEMENTATION

Based on our previous work [9], we proved that grouping Bitcoin nodes that are geographically close would improve the transaction propagation delay. However, nodes that are geographically close might actually be quite far from each other in the physical internet. This actual, physical internet distance may lead to different results, leading to different conclusions too. Taking that into account, in this work we examine a new protocol that groups the Bitcoin nodes based on ping latencies. Specifically, we introduce a novel clustering protocol that uses the proximity based ping latencies in the neighbour selection in order to incorporate proximity-awareness into the existing Bitcoin protocol. The proposed protocol, which is called Bitcoin Clustering Based Ping Time (BCBPT), aims to convert the Bitcoin network topology from normal randomised neighbour selection to proximity based latency selection. Peers in BCBPT are self-cluster based proximity, thus every peer must know whether other peers are close in the topological term (physical internet) in order to connect to those peers and form a cluster. Therefore, peers within each cluster are highly connected via short link latencies. Giving the visibility into the available information from the outside cluster, each node maintains a few long distance links to the outside cluster. This protocol is implemented in two phases, cluster generation and cluster maintenance. Both phases will be discussed in detail in the following subsections.

### *A. Distance calculation*

As the distributed algorithm principle is followed in our proposed protocol, each node runs the protocol independently by information about the proximity of discovered nodes and local neighbours. In this phase, it is necessary to maintain the clusters of nodes with less ping latencies. By doing this, proximity in the physical internet would be enhanced, while keeping the relay overhead trade-off at an acceptable level. Therefore, each node is responsible for gathering proximity knowledge regarding the discovered nodes. This can be done through calculating the distance in the physical internet between a node and other nodes by the node itself. We define the proximity based on the how far a node is from other nodes in the physical internet. In other words, the proximity between two nodes relies on distance that is measured as the round-trip latency between those nodes. Specifically, when a node discovers new Bitcoin nodes, it calculates the distance between itself and the Bitcoin nodes that it has discovered. The discovered nodes are the nodes that have been supplied by either DNS or the normal Bitcoin network nodes discovery mechanism. Two nodes $N_i$ and $N_j$ are considered close to each other if:

$D_{i,j} < D_{th}$ where $(D_{i,j})$ is the distance between $N_i$ and $N_j$ measured by the round-trip latency, $D_{th}$ is the latency threshold. (1)

We introduce a utility function that could calculate the distance between two nodes in the Bitcoin network measured by latency. This function would dramatically change the behavior of the overlay and help enriching nodes with proximity knowledge. The new utility function is shown in (2):

$$D_{i,j} = MPing / (rate(r)) + 2P + q' \quad (2)$$

Where *i* and *j* are two nodes in the network, *Mping* is the length of the ping message (Bytes). The term rate(r) represents the rate of transmission which is the total amount of data that can be sent from one place to another in a given period of time (around ~100 KB/hour), while *P* refers to the propagation speed which is the amount of time it takes for one particular signal to get from one point to another. *P* is multiplied by 2 because of the roundtrip time. The propagation speed is calculated as:

$$P = D(m)/S \quad (3)$$

The term *D(m)* denotes the distance between two nodes i and j. *D(m)* can be calculated using the geographical distance calculation methodology that has been used in our previous work [9]. *S* is the speed of the signal which is equal to $3*10^8$ *ms* when dealing with Wi-Fi internet, while it is equal to $2/3*3*10^8$ *ms* in terms of copper cable. $q'$ represents the queuing time(average). Queuing time can be calculated as:

$$q' = Mping / r - \lambda * Mping \quad (4)$$

Where $\lambda$ represents the arrival rate (How many pings are arriving to the node *j*).

As distances measurements are subject to network congestion and therefore dynamic, within some variance, multiple messages between pairs of nodes, repeatedly are sent over the time in order to determine variance. In terms of a discovered node close to a node, the node establishes a connection with the discovered node by sending a version message as a handshake. In contrast, these two nodes would have a very little chance to get directly connected and stay in the same cluster if they are so far away from each other. Therefore, clusters in the overlay network become more proximity-aware and nodes make a better decision in terms of limiting the cost of communication. However, to measure the distance between nodes in "ping latency" requires every pair of nodes to interact, which added an extra overhead to the network. This overhead will be evaluated in our future work.

### *B. Cluster creation and maintenance*

In this phase, a protocol for BCBPT clustering maintenance is designed. While joining the network for first time, a node N learns about the available Bitcoin nodes from a list of DNS services. However, the node discovery service should also make a ranking on which node to select and which not as the initial DNS seed service might return sub optimal peers. Therefore, DNS service nodes should recommend available nodes to the node N based on the proximity in the physical geographical location as the geographic distance in the internet is many times a good indication of topologic distance. DNS service follows the geographical distance calculation methodology that has been used in our previous work [6] in order to recommend closest available nodes to node N. The node N calculates the distance to each discovered node in order to get its proximity ordering based on a latency threshold. This ordering would help the node N to be directed to a specific cluster. The role of DNS service stops once the node N joins a cluster. After that, the node N sends a JOIN request destined for the closest node K of the discovered nodes. Once the node N connects to the node K, it receives a list of IPs' of nodes that belong to the same cluster of the node K in order to allow the node N connects to the nodes that belong to K's cluster only. Periodically, Node N discovers other nodes using the normal Bitcoin network nodes discovery mechanism [9]. Then, node N finds out whether the discovered nodes are physically close by following the distance calculation mechanism that has been mentioned above. When the node N wants to leave the network, in this case no further action is required.

## V. EVALUATION METHODOLOGY AND CRITERIA

### *A. Simulation structure*

In order to evaluate the proposed protocol, we simulate our solution on an event based simulator that has been built in [5]. To make our simulations realistic, we maintained all the required conditions to simulate the reality of the Bitcoin network in terms of information propagation. Specifically, different measurements of the most influential parameters that have a direct impact on a client's behavior and information propagation in the real Bitcoin network are attached to the simulator. These measurements that have been presented in our previous work [5],[12], include the number of reachable nodes, link latencies, and nodes' session lengths. As we focus on information propagation in the real Bitcoin network, distributions of link latencies between nodes as well as the number of reachable nodes are attached to the simulator. These distributions were collected by running the developed crawler which was connected to approximately 5000 network peers and observing a total of 20,000 ping/pong messages. Likewise, the nodes' behavior in the real Bitcoin network is simulated by designing joining and leaving events based on the measurements of peers' session length in the real Bitcoin network. These measurements have been gathered in [5] by developing a Bitcoin client that was able to indicate the points in time in which peers left or joined the real Bitcoin network. In [5], the simulator has been validated based on measurements of the transaction propagation delay in the real Bitcoin network. These measurements have been collected by implementing a novel methodology that identifies the time by which a transaction reaches each node in the real network. Compared to the real Bitcoin network propagation delay measurements, validation results revealed that the simulation model approximately behaves as the real Bitcoin network.

### *B. Experiment setup*

The validated simulator that is mentioned above is used to set up the BCBPT evaluation experiment. As we based our evaluation on the transaction propagation delay, the size of the network matters. Though, we start the simulation with the number of nodes that matches the size of the real Bitcoin network which was measured in [5]. Before applying the aforementioned proximity cluster generation algorithm, we assume that the network nodes belong to one cluster. Based on our solution, several proximity based clusters will be generated at certain times based on a chosen ping latency threshold. In particular, BCBPT decides whether two nodes are close to each other if the measured distance based latency is lower than the suggested distance threshold $d_t$= 25ms. Furthermore, the distance between nodes are modelled based on real-world measurements that were collected in [5],[12]. Each node in the overlay is allowed to discover new nodes every 100ms. After getting some proximity based clusters, normal Bitcoin simulator events will be launched. As we based our simulations on measuring how fast a transaction is propagated in the network after applying our clustering approach, we measure the transaction propagation delay using the same methodology which was used in our previous work [5] to measure the transaction propagation delay in the real and simulated Bitcoin network. By doing this, we can evaluate the BCBPT protocol by comparing the measurements of the transaction propagation delay that have been collected in the simulated Bitcoin protocol to the same measurements that have been collected in this experiment. Fig.2 gives a simple diagram of how the simulation experiment works. We implemented a measuring node m which is able to create a valid transaction *Tx* and send 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 *m* to one connected node only. Then node *m* records the latency by which all *m*s' connected nodes would receive the transaction. Suppose the client m has proximity based connections (1,2,3,…., n), m propagates a transaction at time $T$ , and it is received by its connected nodes at different times $(T_1,T_2,T_3,....,T_n)$ as illustrated in Fig.2. The time differences between the first transaction propagation and subsequent receptions of the transaction by connected nodes were calculated ($\Delta t_{m,1}$ ,…., $\Delta t_{m,n}$) according to (5) :

$$\Delta t_{m,n} = T_{\mathrm{n}} - T_{\mathrm{m}} \qquad (5)$$

Where $T_{\mathrm{n}} > T_{\mathrm{n}}$ -1>…….> $T_2$> $T_1$

The time in which the transaction is propagated by our measuring node and reached each node of our measuring node connections was calculated by running the measuring node *m*. In order to get accurate measurements, the latency is determined by an average of approximately 1000 runs as errors such as loss of connection and data corruption are expected to happen while dealing with the network. The distribution of these measured time differences $\Delta t_{m,n}$ represents the exact transaction propagation delay as measurements are indicated when peers receive transactions.

*C. Results and discussions:*

Fig.3 compares the distributions of $\Delta t_{m,n}$ for the simulated BCBPT protocol against the same distributions that have been measured in the simulated Bitcoin protocol and LBC protocol. Results reveal that the BCBPT protocol offers an improvement in propagation delay compared to the Bitcoin protocol as well as LBC protocol. Regarding the comparison between the BCBPT and Bitcoin protocol, the Bitcoin protocol performs variances of delays, which have been collected in our prior work [5], that grow linearly with the number of connected nodes, whereas BCBPT maintains lower variances of delays regardless of the number of connected nodes. This suggests a strong link may exist between these results and connections based on the proximity in the physical internet topology that is maintained in the BCBPT protocol. Precisely, the reduction of the transaction propagation time variances in the BCBPT protocol has to do with the fact that the connections based proximity between nodes implies faster transmissions due to short distance links among nodes.

Turning now to the comparison between BCBPT protocol and locality based protocol LBC. The LBC protocol was proposed in our prior work [6] as a mechanism to improve the transaction propagation delay in the Bitcoin network. The LBC protocol aims to convert the Bitcoin network topology from normal randomised neighbour (connected nodes) selection to location based neighbour selection. Clusters in LBC protocol are formulated by referring an extra function to each node in the Bitcoin network. By this function, each node is responsible for recommending proximity nodes to its neighbours. The proximity is defined based on the physical geographical location. Fig. 3 shows the variances of delay both the BCBPT

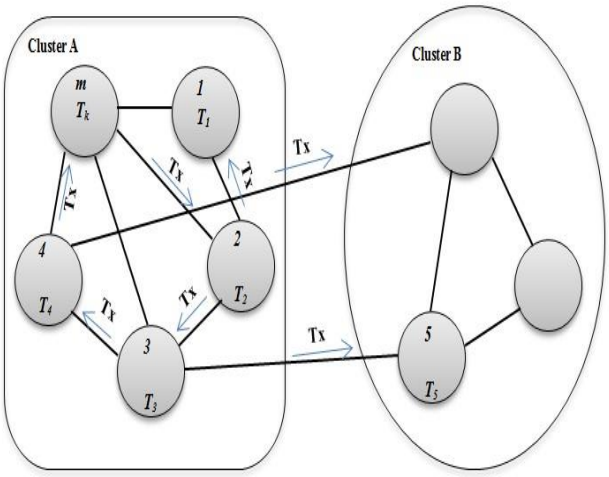


**Fig.2: Illustration of simulation experimental setup**

protocol and LBC protocol. Before proceeding to discuss the difference between both protocols, it is important to mention that these variances of delays in both protocols have been measured using the same methodology. Results show that the BCBPT protocol maintains an improvement in variances of delays over the LBC protocol. The most likely cause of the higher variances of delays in the LBC protocol is that two geographically close nodes may be actually quite far from each other in the physical internet, as in any other P2P network. Therefore, physical distance may lead to better results, leading to a different conclusion that proximity awareness in the physical internet improves the delivery latency with a higher probability due to offering fewer hops as well as shorter links. Furthermore, dynamics of internet routing, as caused by BGP (Boarder Gateway Protocol) peering agreements, can also result in surprising situations that closest differs between geographical and topological terms.

As the proximity based clustering protocol is based on a suggested threshold, it is worth investigating the optimal latency distance threshold that can speed up information propagation. To this purpose, we experiment with BCBPT based on several suggested distance thresholds $d_t$. A comparison among three variances of delays which were measured based on three different suggested thresholds 30ms, 50ms, 100ms, is shown in Fig.4. Results reveal that less distance threshold performs less variance of delays. Judging from that, propagation delay negatively corresponds with the latency threshold, as the total duration of subsequent announcements of the transaction by the remaining nodes increases with a larger latency threshold. 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 physical topology which are offered $d_t$.. There are some security implications that might be raised while selecting peers confined to closest proximity. In particular, it would seem possible for an attacker to more easily launch eclipse attacks by concentrating its bad peers within a small cluster. Though, a good peer from the same area joining

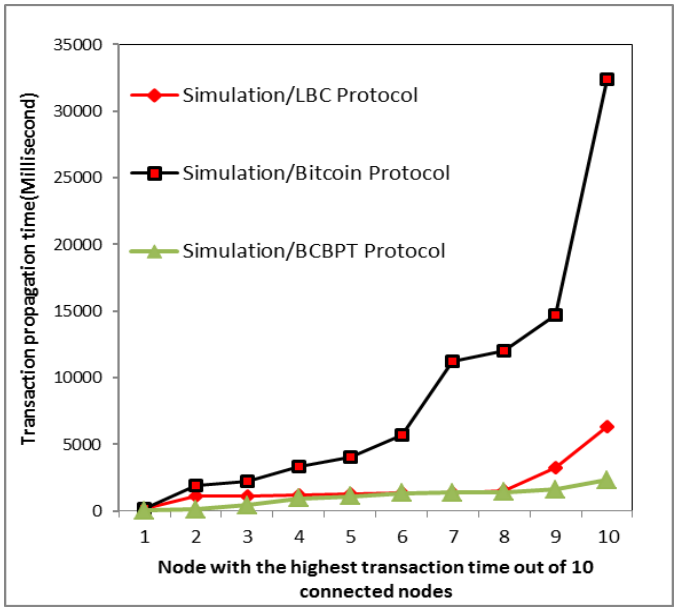


**Fig.3: Comparison of the distribution of $\Delta t_{m,n}$ as measured in the simulated Bitcoin protocol with BCBPT protocol and LBC protocol simulation results.( $d_t$ =25 *ms.)***

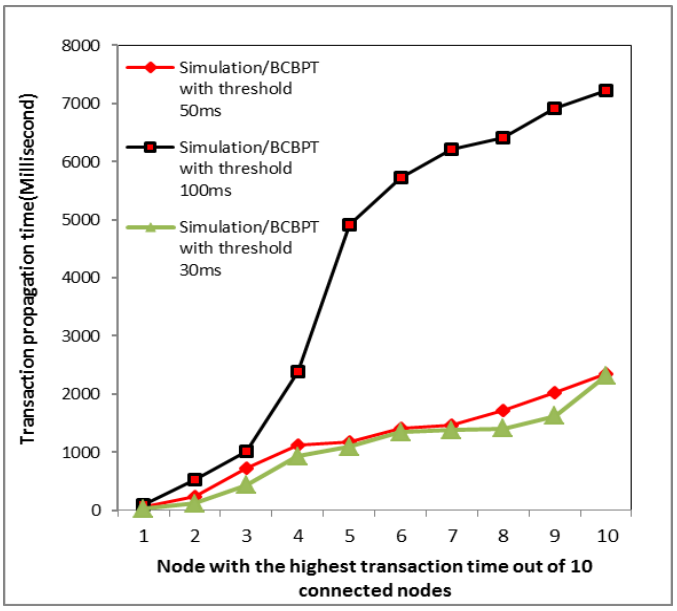


**Fig.4: Comparison of the distribution of $\Delta t_{m,n}$ as measured in the simulated BCBPT protocol with three thresholds ($d_t$ =30*ms, 50ms, 100ms* ).**

The Bitcoin network might have a higher probability of selecting from these bad peers. This would achieve a completely malicious cluster. In our view, an eclipse attack is a bit challenging as the proposed protocol aims to have clusters based on countries. Similarly, partition attacks seem to have a great potential. As undertaking clustering in the Bitcoin network is a fundamentally different proposition to clustering within other classes of network due to the strict requirements on security. Therefore, we plan to evaluate some possible classes of attacks with regards to our proximity protocol. So our future work will include evaluation of partition attacks as well as eclipse attacks.

## VI. Conclusion

In this paper, brief backgrounds of Bitcoin system as well as analysing the information propagation in the real Bitcoin network were presented. In addition, how propagation delay in the Bitcoin network could affect security by offering an opportunity to double spend the same coins; thereby abusing the consistency of the public ledger was discussed in this paper. The BCBPT, a novel clustering protocol that incorporates proximity-awareness into the existing Bitcoin protocol, was presented in this paper. By conducting extensive simulations, BCBPT evaluation results indicate an improvement in the transaction propagation delay over the Bitcoin network protocol. Furthermore, experiments with different distance threshold values have been conducted to identify the distance threshold that would give better improvement in the transaction propagation delay. We discovered that the providing less distance threshold would improve the transaction propagation delay with high proportion. Based on the transaction propagation delay, we compared between the BCBPT protocol and LBC protocol. Comparison results showed that the BCBPT protocol minimises variances of delay

## VII. References


[1] Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. Retrieved from http://www.bitcoin.org/bitcoin.pdf.

[2] 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.

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

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

[6] Fadhil, M., Owenson, G., & Adda, M. (2016).Locality based approach to improve propagation delay on the Bitcoin peer-to-peer network. IFIP/IEEE International Symposium on Integrated Network Management. Lisbon // Portugal.

[7] Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, *27*(2), 228-234.

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

[9] 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.

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

[11] Bamert, T., Decker, C., Elsen, L., Wattenhofer, R., & Welten, S. (2013). Have a snack, pay with Bitcoins. IEEE P2P 2013 Proceedings, 1–5. doi:10.1109/P2P.2013.6688717.

[12] Fadhil, M., Owenson, G., & Adda, M. (2016). Bitcoin network measurements for simulation validation and parametrisation. Internationl networking conference. Frankfurt/ Germany.