Estimation of Data Propagation Time on the Bitcoin Network

Reiki Kanda
Tokyo Institute of Technology
Tokyo, Japan
kanda.r.aa@m.titech.ac.jp

Kazuyuki Shudo
Tokyo Institute of Technology
Tokyo, Japan
shudo@is.titech.ac.jp

ABSTRACT

The goal of this research is to estimate the data propagation time on the Bitcoin network. Using network coordinates, we estimate the communication latency between computers. Such latency estimation contributes future optimization of data propagation. In this research, we report an experiment on computing the network coordinates. In the current Bitcoin network, it is very difficult to acquire internode delay because the network topology is not available. In this study, we calculate the delay based on our topology estimation and describe the effectiveness of the network coordinates using various topology estimation parameters.

CCS CONCEPTS

• **Networks** → Peer-to-peer networks;

KEYWORDS

Bitcoin, network coordinates, Vivaldi, Pharos

ACM Reference Format:
Reiki Kanda and Kazuyuki Shudo. 2019. Estimation of Data Propagation Time, on the Bitcoin Network. In *Asian Internet Engineering Conference (AINTEC '19), August 7–9, 2019, Phuket, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3340422.3343640

1 INTRODUCTION

Bitcoin is currently the most popular cryptocurrency; however, cryptocurrency networks face the problem of low transaction throughput. This is because the blockchain which is a public ledger that records all the Bitcoin transactions, takes some time to make the transactions secure [10]. The bitcoin transaction volumes have increased recently. Therefore, more time is required to settle a transaction, and the number of transactions that can be processed per unit time for a user has decreased. Conversely, if the duration and the number of transactions processed per user can be improved, more people will be able to use Bitcoin and blockchain comfortably.

To improve the transaction throughput, it is necessary to shorten the block generation interval [15]. When each node generates a block, it sends the blocks to relay around itself. If another node generates a block before the block has reached all the nodes, a phenomenon called a fork, which needs to be avoided. In order to raise the transaction throughput while suppressing the occurrence probability of the fork, it is necessary to increase the block propagation speed. By increasing the speed, it is possible to shorten the block generation interval while avoiding the occurrence of a fork. Therefore, it is necessary to know the internode delay to shorten the propagation delay. If we can measure the internode delay, it will be possible to devise an efficient block relay method. Currently, a block in a Bitcoin is propagated to all the nodes by flooding on a random netwrok, and there is room for improvement. If the block relay efficiency is increased, it will improve the transaction throughput and delay problems associated with Bitcoin and other similar blockchain systems.

In a Bitcoin network, it is difficult to measure the internode delay because the topology is unknown. However, by taking the difference of the block arrival time, it is possible to partially obtain the internode delay. Therefore, we attempt to compute network coordinates from the partially obtained internode delays. By using the network coordinates, it is possible to estimate the internode delays that are not measurable. We compute a map that approximates the delay relation between all the nodes by placing coordinates on the Euclidean space with the distance relations based on the partial node delay. The network coordinates is a map that regards the internode delay as a Euclidean distance, and use well-known configuration methods, such as Vivaldi [2] and Pharos [14]. In this research, we applied Vivladi and Pharos and computed network coordinates for delay estimation in a Bitcoin network.

In the next section, we introduce the related research. Next, we describe the purpose of propagation delay estimation in Section 3 and our proposed method in Section 4. Section 5 gives the details of the experimental method, and Section 6 evaluates the experiment. In Section 7, we provide a conclusion and future work.

2 RELATED WORK

In this section, we refert to two studies related to propagation delay and a research that presented an algorithm for constructing the topology of the Bitcoin network. Decker et

AINTEC '19, August 7–9, 2019, Phuket, Thailand

ACM ISBN 978-1-4503-6849-0/19/08…$15.00
https://doi.org/10.1145/3340422.3343640

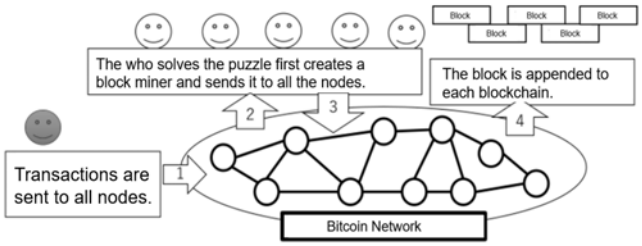

Figure 1: Functioning of Bitcoin nodes.

al. collected timestamp when the block was propagated, and when the median and mean values were 6.5s and 12.6s, respectively, for a new block to propagate from one node to another. This indicates that the block propagated to 95% of the whole node in 40s. However, although this research investigated the propagation delay on a global basis, it did not investigate the delay between nodes. We estimate the delay between the nodes and perform block propagation delay estimation in detail. Moreover, in the above paper [3], to improve the block propagation speed, it is proposed to make the block verification stage more efficient and to increase the number of connections. If we improve the accuracy of internode propagation delay estimation in this research, it can be illustrated as another method to improve the propagation speed [5] [6].

Next, Coinscope [9] was investigated as the topology of the Bitcoin network. They used an algorithm called AddressProbe to track how each node was connected. However, we can not apply the algorithm for the current Bitcoin network because of changes in the Bitcoin Core version. Therefore, in this research, we did not obtain an actual topology, and we decided to estimate a topology. Coinscope was searching for an influential node in the Bitcoin network by examining the topology. If the network coordinates in this research are high in accuracy, it is useful to search for influential nodes besides nodes for efficient propagation.

3 PURPOSE OF PROPAGATION DELAY ESTIMATION

In this section, we first outline the Bitcoin and describe the problems related to the speed of transaction confirmation. Then, we provide an overview of the methods used to overcome these problems. Finally, we explain why it is necessary to estimate the propagation delay.

3.1 Overview of the Bitcoin system

In this section, we outline the Bitcoin system with reference to Fig. 1. Bitcoin consists of a transaction settler who generates the transactions, miners who produce blocks, and a P2P network called a Bitcoin network that propagates blocks and transactions. The Bitcoin transaction proofs are saved in a Bitcoin network in such a way that anyone can confirm these transactions subsequently. It is not possible to falsify the blockchain ledger or change it retrospectively;therefore, the transactions can be confirmed, and we can identify the owner of the current Bitcoin from anyone on the network.

When a Bitcoin transaction is made, the transaction is propagated to the Bitcoin network and stored in the transaction pool of each node. This unconfirmed transaction group waits for processing by the Proof-of-Work system, which is a concensus process that prevents data tampering. The Proof-of-Work process involves the generation of a block that contains a transaction based on certain rules;it is performed by a miner. When multiple miners try to generate the same block, other nodes verify that the generated block is the correct block at the time of transfer.Therefore the block having the incorrect generation is not diffused; only the correct block that was generated earlier is diffused.

The generated block is propagated to the entire Bitcoin network and added to the blockchain possessed by each node.

This blockchain is a history of transactions that occur in the Bitcoin network, and it is a very large transaction ledger. In addition, when generating a block, the information obtained from the block immediately before this blockchain is also used, and the continuity of the ledger is also guaranteed.

3.2 Problems related to the speed of transaction confirmation

The Bitcoin system faces the following two problems related to the speed of transaction confirmations;

- It has a long block confirmation interval.
- It has a low transaction throughput.

3.2.1 Long block confirmation interval. The block transaction confirmation interval is the time between the generations of two adjacent blocks. When a transaction is placed in a new block, the transaction is said to have been confirmed. The block interval is currently designed to be appropriately 10 min. If it takes more time to confirm, then it means that it is taking longer to trust a certain settlement. This is not appropriate for people to use Bitcoin daily as currency.

A 10-min confirmation interval is set to prevent the blockchain fork. This fork is caused by two blocks arriving almost simultaneously on a node when the block propagates over a network. The occurrence of blockchains that are not single chains means that they have separate ledgers between nodes. This becomes problematic because the transaction records cannot be made consistent between the nodes.

When a blockchain is updated in a node for a short time, the node has a high occurrence of blockchain forks. Therefore, a blockchain fork can be prevented by not generating blocks frequently. To reduce the incidence of occurrence of blockchain forks, it is necessary to reduce the block propagation delay [3]. Therefore, the efficiency of block propagation makes it possible to shorten the confirmation interval.

3.2.2 Low transaction throughput. This shows that the transaction throughput of Bitcoin is very low compared with the

transaction throughput of Visa [13] and Paypal [12], which have more than 8000 and 230 transactions, respectively, per second. The transaction throughput is determined by dividing the amount of data that can be included in a block by the block confirmation interval. Therefore, there are two possible strategies for increasing the transaction throughput. One strategy is to increase the amount of data that can be included in a block, and the second strategy is to shorten the confirmation interval. In the first case, the propagation time is prolonged in proportion to the increase in the amount of data; therefore it does not improve the processing performance. However, there are studies in which transactions have been compressed to substantially increase the amount of data [1].

3.3 Importance of propagation delay estimation

To solve the above two problems, it is necessary to improve the block propagation efficiency. Knowing the internode delay of the Bitcoin network helps to improve the efficiency of block propagation. However, it is difficult to measure all internode delays because the number of Bitcoin nodes is currently over 10,000 [7], and the nodes are connected in a P2P network. Therefore, we consider the partial delay measurement of internode delay to estimate the overall internode delay. In the next section, we will explain the methodology in detail.

4 ESTIMATION METHOD

In this report, we use network coordinates to estimate internode delays. Based on the internode delays that are partially measured, we perform the mapping by assigning coordinates on the Euclidean space. By doing this, we aim to compute a map that approximates the delay between all the nodes. We use Vivaldi [2] and Pharos [14] as a map computation method.

Vivaldi is based on the principle of the spring. The natural length of the spring corresponds to the network distance in the actual measurement results. Then, by stretching the spring between the nodes and repeating the operation to release the tension, the energy of the entire spring is minimized. The energy E of the entire spring corresponds to the sum of squared errors of the measured delay and the coordinate distance as follows.

$$E = \sum_{i,j} (l_{ij} - ||p_i - p_j||)^2 \quad (1)$$

Here, i, j is the latency measured between the nodes l_{ij} to be measured latency between nodes l_{ij} and p_i to coordinates of node i In other words, loosening the spring means finding an appropriate node mapping. Algorithm 1 shows the algorithm of the central part of Vivaldi. Pharos, like Vivaldi, uses the spring principle. The difference of Pharos from Vivaldi is that each node has local and global coordinates. Pharos groups nodes that are close together in the network distance

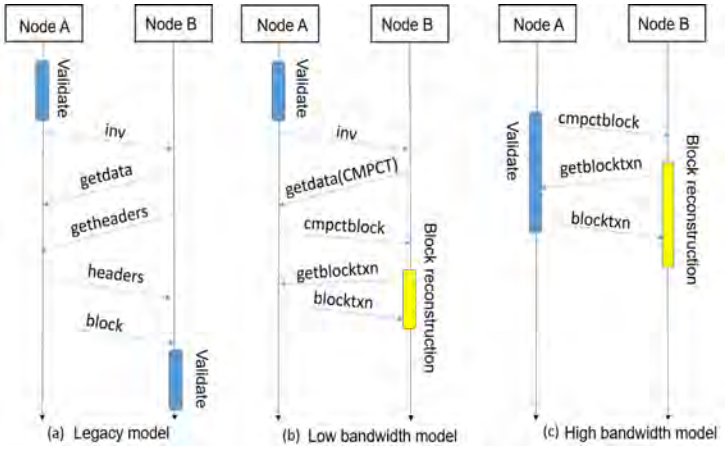

Figure 2: Block relay protocol.

into a set of nodes called a cluster. An algorithm similar to Vivaldi is executed for local coordinates of nodes belonging to the same cluster, a similar algorithm is executed for global coordinates between the nodes belonging to other clusters.

Algorithm 1 Vivaldi's core algorithm.

```
Input : L_ij:measured node i.j delay
        x_i:coordinates of node i
Output : x:coordinates of node after processing Vivaldi
N : Node set
u(x) : Unit vector of x
while error(L, x) > torelance do
  for all i ∈ N do
    F = 0
    for all j ∈ N do
      e = L_ij − ||x_i − x_j||
      F = F + e × u(x_i − x_j)
      x_i = x_i + t × F
    end for
  end for
end while
```

5 ESTIMATION EXPERIMENTS

To estimate the block propagation delay, we first describe a method to partially acquire the internode delay. Next, we estimate the topology of the Bitcoin network, and then execute Vivaldi and Pharos on it. Since it is very difficult to obtain the actual topology; therefore we will estimate the topology by using the transaction propagation delay here. Although the block propagation delay includes the transfer time and the validation time, it also measures the duration of the validation time.

5.1 Block relay protocol

It is necessary to explain the subsequent internode delay acquisition method, therefore we explain here Bitcoin's block

transfer protocol. Each block is propagated throughout a network using flooding from one node as a starting point. At the time of this flooding, timestamps at the time of block arrival on each nodes are issued at each node. If it is combined with the neighbor node list, we can determine the delay between the two points. The procedure for block propagation is illustrated in Fig. 2. In August 2016 Bitcoin Core 0.13.0 was released, a scheme called compact block relay was introduced for efficiently block propagation. The compact block relay was roughly divided into two relay systems-a low bandwidth mode and a high bandwidth mode. In low bandwidth mode, we use the inv message but do not use it in high bandwidth mode. Each user decides which bandwidth mode to use. In Figs.2 (a) and (b), when the block validation is completed, an inv message is sent if it already has the same block before relaying it. A node that receives an inv message sends a getdata message and receives a block. For model (b), all the block data is not received there is only partial reception of the necessary parts. It is not possible to obtain an accurate timestamp indicating the block arrival time on a peer to peer network. To acquire internode delay, we use the timestamp of the block propagation provided by Bitnodes [7]. The Bitnodes issues timestamps when they obtain the inv message before receiving the block. This inv message is the inv message of the model (a) of Fig. 2. Bitnodes provides the timestamp when the inv message was acquired and the IP address that sent the message. However, only one timestamp list with the lowest delay and the highest 1000 nodes can be obtained at one flooding. By retrieving the timestamp of inv messages related to a number of block flooding from Bitnodes, we could obtain the timestamp of the inv message going all over the Bitcoin network.

5.2 Acquiring internode delay

Next, we will show what the difference $t_B - t_A$ of the timestamps between these two points specifically represents. We define the following;

t_A The time point when the inv message was received from node A

T_{inv}**(A → B)**
The time interval when node A sent an inv message to node B

T_{inv}**(A → Bitnodes)**
The time interval when node A sent an inv message to node Bitnodes

$t_B - t_A$
$= T_{inv}(\text{A} \to \text{B}) - T_{inv}(\text{A} \to \text{Bitnodes})$
$+ T_{inv}(\text{B} \to \text{Bitnodes})$
+ (Duration between the exchange A and B) + (validation time)
Since $T_{inv}(\text{A} \to \text{B}) \simeq T_{inv}(\text{A} \to \text{Bitnodes})$,
$t_B - t_A$
$\simeq$ (Duration between the exchange A and B)
+ (validation time) + $T_{inv}(\text{B} \to \text{Bitnodes})$

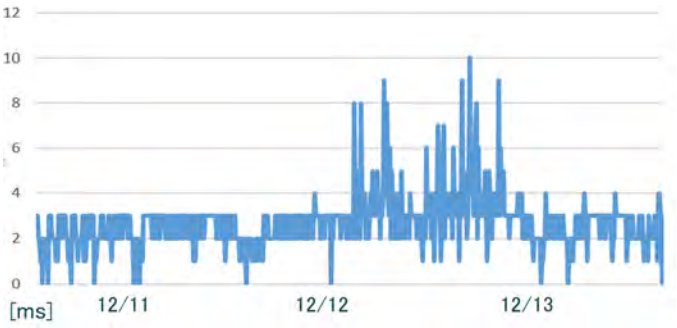

Figure 3: Validation time.

The internode delay can be obtained while correlating the timestamp difference with the estimated topology. However, even with the known neighbors, it is difficult to determine whether the blocks have been exchanged exactly between the nodes only as per the timestamp table. Therefore, a node group of the first hop from the starting node and a node group of the second hop are formed. When there are neighbors in both groups, the difference between the timestamps is defined as an internode delay. The first hop node group is the first to the nth node list, and the second hop node group is the $(n+1)$th to the n^2th second hop node group. Beyond these node groups, the n value is the number of hops. There may be times when multiple internode delays can be acquired. In this case, we made the median value as representative value of the internode delay after eliminating the extreme value.

5.3 Topology estimation

When a transaction is issued, it propagates from the source node. The faster a node receives a propagated transaction, the more likely the node is the connecting to the source node. Therefore, the source node of the transaction can be acquired by Blockcypher [11], and the propagation timestamp is acquired by Bitnodes. We can obtain1000 IP addresses in the order in which the transactions are propagated, in response to the input of the transaction hash value. This hash value was obtained from Blockchain.info [8]. According to the Coinscope, the degree of Bitcoin network account for more than 10 degrees [9]. Therefore, in this experiment, we form topologies with degrees of 10 and 20. The candidates for the neighbors are;

1. The nodes that received the transactions several times in a short period.
2. The nodes that received the transactions earlier.

We formed a topology with degrees of 10 and 20 with the priorities 1 and 2, respectively.

5.4 Validation time

To measure the validation time, `CheckBlock()` function of the Bitcoin-Core was called, and the time was measured until the end. The `CheckBlock()` function is responsible for

validation when the block propagates. The experimental environment used an instance of t2.large of Amazon EC2. We used std :: chrono of C ++ as a timer. Figure.3 shows the block validation time over the 3 days of December 2017. Many of them are around 2 msec, and it was understood to be a maximum length of appropriately 10 msec.The Bitcoin network propagates 50% of the whole in appropriately 1000 msec [7]. Therefore, the propagation delay is dominated by the transfer time rather than the validation time.

5.5 Estimation of the block propagation delay

Block propagation delay is estimated based on a network coordinate. To evaluate the effectiveness of the network coordinate system, we introduce a block propagation delay estimation method that does not use the network coordinate system. The experimental data used have 716 blocks in a block height ranging from 543,580 to 545,099. This is a block of appropriately 5 days, and the topology of the Bitcoin network is treated as if it does not change significantly [9]. Also, if there are a number of .onion addresses above the propagation timestamps, these addresses will not be acquired as internode delay. This is because of the communication delay over the Tor network, which makes it difficult to reflect the geographical position relations, and it is also difficult to compute a highly accurate network coordinate.

5.5.1 Vivaldi and Pharos. We applied Vivaldi and Pharos to compute network coordinates with two dimensions and height. In addition, Pharos is capable to compute a 7D network coordinate. This was because we applied the dimensions in the report of Pharos and Vivaldi. Vivaldi and Pharos search for coordinates that reflect the actual delay, but the initial arrangement used random numbers between 10 and 100. Pharos first forms clusters and then applies Vivaldi to the formed clusters. The IP addresses can be acquired by Bitnodes; therefore, the region can be identified from the IP addresses. In addition, we divided the region into six states and made it an initial cluster of Pharos. We used Vivaldi and Pharos as the training data for internode delay based on an estimated topology; we also computed network coordinates, and estimated block propagation delay.

5.5.2 Estimation of block propagation delay without network coordinates. We describe a method of estimating the block propagation delay without using network coordinates. We calculate the sample mean from the training data and use it as an estimate of internode delay. Therefore, this estimate takes the same value for any internode delays.

6 EVALUATION

We divided the test data and the training data, evaluated the accuracy using the k-cross validation method, and used the mean squared error for error evaluation as

$$MSE = \frac{(Observed\ value - Estimated\ value)^2}{(the\ number\ of\ internode\ delay)}$$

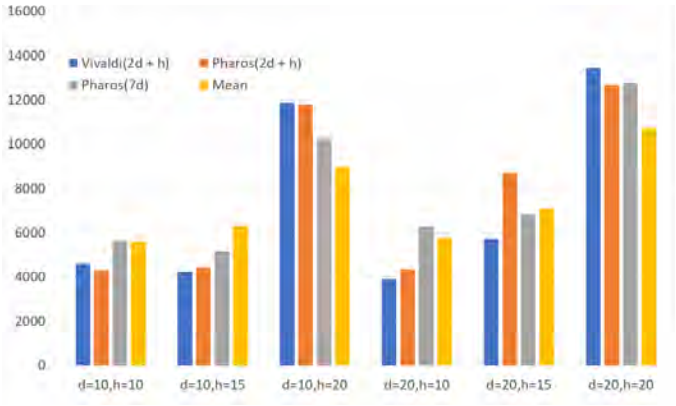

Figure 4: Mean squared error result.

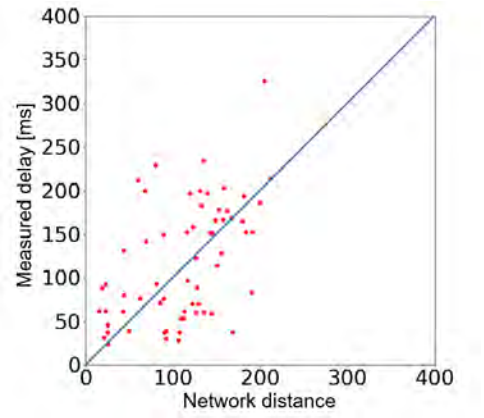

Figure 5: Evaluation of network coordinates estimated with the setting (degree, the number of hop) = (10, 15) on Vivaldi in 2D in addition to height coordinates.

The mean in Fig. 4 is the mean squared error when the mean value of the training data was the estimated value. From this point, we evaluate each method according to the Fig. 4. We can see that the estimation method that is smaller than the error of mean(i.e., when the the estimation accuracy tends to be high) is the network coordinates in which the height coordinates is combined with the two dimensions of Vivaldi. It is also clear that the estimation accuracy of the network coordinates is higher than the mean when the setting of the internode delay acquisition method is (degree, the number of hops) = (10, 15). The correspondence between the estimated values and true values need to be examined in greater detail. Figures 5 and 6 show the correspondence between the network distance and the measured delay. The network distance is the Euclidean distance on the network coordinate system, which is an estimated value. The accuracy of the network coordinates is directly proportional to the number of points that lie close to the central straight line, the more accurate network coordinates is computed. Figure

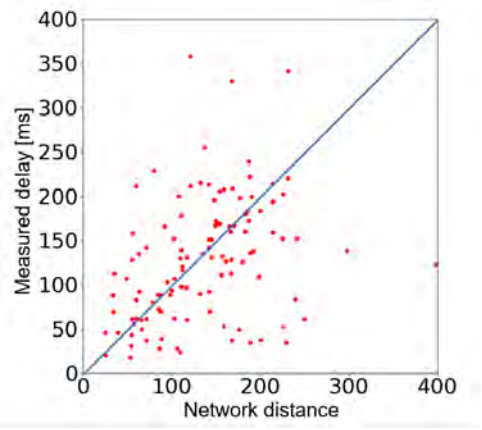

Figure 6: Evaluation of network coordinates estimated with the setting (degree, the number of hop) = (10, 20) on Vivaldi in 2D in addition to height coordinates.

5 is an evaluation diagram of the network coordinates that was generated based on the internode delay obtained by acquiring Vivaldi in 2D in addition to the height coordinates with (degree, the number of hops) = (10, 15). Figure 6 is an evaluation diagram of the network coordinates generated based on the internode delay obtained by acquiring Vivaldi in 2D in addition to height coordinates with (degree, the number of hops) = (10, 20).It is clear from Fig. 6 that many points are located at a large distance from the straight line. This corresponds to the fact that in Fig. 4, the mean squared error is large when the setting of Fig. 6 is used.

7 CONCLUSION AND FUTURE WORK

Using the network coordinates is effective under certain conditions for estimating the internode delay of Bitcoin network. We found that the effectiveness of the network coordinate system changes substantially depending on how the topology and the block relay movement are assumed. In this experiment, we showed that when (degree, the number of hops) = (10, 15) is set, the method of the network coordinates system is more accurate than the estimation using a simple average value. The network coordinate system is estimated by imitating the actual geographical node arrangement in the Euclidean space. In the estimation that uses a simple average value, the estimation accuracy was worse because that part was not considered. However, depending on the tentative topology and the setting of the number of hops, the estimation that uses a simple average value would be more accurate. This is because it is difficult to estimate the topology of the Bitcoin network, and it is also difficult to measure the delay between nodes. It is essential to estimate the topology more accurately to increase the estimation accuracy. In the future, we will use methods such as TxProbe [4], that have been proposed.

In this experiment, several network coordinates and algorithms were used, but it was not clear which network coordinates reflects the propagation delay most. Vivaldi and Pharos have different results depending on their parameter settings and the initial positions; therefore, we need to find better parameters and settings. In our future studies, we will compute a network coordinate under the best conditions and discuss which network coordinates would be suitable for block propagation delay estimation.

ACKNOWLEDGMENTS

This work was supported by SECOM Science and Technology Foundation.

REFERENCES

[1] BitcoinCore website Segregated Witness Benefits. Accessed: 2019. https://bitcoincore.org/en/2016/01/26/segwit-benefits/
[2] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. 2004. *Vivaldi: A Decentralized Network Coordinate System.* New York, NY, USA. https://doi.org/10.1145/1015467.1015471
[3] C. Decker and R. Wattenhofer. 2013. *Information propagation in the Bitcoin network.*
[4] Sergi Delgado-Segura, Surya Bakshi, Cristina PÃ©rez-SolÃ , James Litton, Andrew Pachulski, Andrew Miller, and Bobby Bhattacharjee. 2019. *TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions.* https://arxiv.org/pdf/1812.00942.pdf.
[5] Falcon. Accessed: 2019. Falcon - A Fast Bitcoin Backbone. https://www.falcon-net.org/
[6] FIBRE. Accessed: 2019. FIBRE Fast Internet Bitcoin Relay Engine. http://Bitcoinfibre.org/public-network.html
[7] Addy Yeow website BITNODES. Accessed: 2019. https://bitnodes.21.co/
[8] BLOCKCHAIN LUXEMBOURG S.A. Accessed: 2019. https://www.blockchain.com/ja/explorer.
[9] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2015. *Discovering Bitcoin's Network Topology and Influential notes.* https://cs.umd.edu/projects/coinscope/coinscope.pdf.
[10] Satoshi Nakamoto. Accessed: 2019. *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://bitcoin.org/bitcoin.pdf
[11] website Blockcypher. Accessed: 2019. https://live.blockcypher.com/
[12] website PayPal. Accessed: 2019. https://investor.paypal-corp.com//releasedetail.cfm?ReleaseID=1009339.
[13] website Visa. Accessed: 2019. http://www.visa-asia.com/ap/jp/general/abtvisa/security.shtml.
[14] Xiaohui Shi Beixing Deng Yang Chen, Yongqiang Xiong. 2007. *Pharos: A Decentralized and Hierarchical Network Coordinate System for Internet Distance Prediction.*
[15] Aviv Zohar Yonatan Sompolinsky. 2015. *Secure High-Rate Transaction Processing in Bitcoin.*