Title
Optional Subtitle

B. van IJzendoorn

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

TITLE

OPTIONAL SUBTITLE

by

B. van IJzendoorn

in partial fulfillment of the requirements for the degree of

Master of Science
in Applied Physics

at the Delft University of Technology,
to be defended publicly on Tuesday January 1, 2013 at 10:00 AM.

Supervisor:	Prof. dr. ir. A. Einstein	
Thesis committee:	Prof. dr. C. F. Xavier,	TU Delft
	Dr. E. L. Brown,	TU Delft
	Ir. M. Scott,	Acme Corporation

This thesis is confidential and cannot be made public until December 31, 2013.

TU Delft Delft University of Technology

PREFACE

Preface...

B. van IJzendoorn
Delft, January 2013

Contents

1

Problem Description

The goal of this thesis is to create a decentralized exchange market where users can trade in private with each other.

A decentralized market has been implemented in Tribler by Olsthoorn (2016) that does not guarantee the privacy of traders. Traders can exchange BitCoin against Multichain coin in a decentralized system. Ensuring the privacy of traders in an exchange is important because otherwise traders can play games and abuse the trade information of other parties for their own benefit. Sensitive trading information becomes public to other users and the trading position of a trader can potentially be derived at two points. At first, there is a decentralized matching engine where bid and ask offers are broadcasted to all other traders to make a match. [1] Secondly, the trading position of a trader might be exposed because the BitCoin wallet does not ensure privacy. The payment transactions are recorded in a decentralized public ledger from which much information can be deduced. An alternative to the BitCoin wallet is the Zerocash wallet which uses a changed version of the blockchain that ensures the privacy of transactions with zero knoweldge proofs and onion routing. [2] However, this is not an option because users should be allowed to pay with the BitCoin wallet and with other wallets from for instance traditional banks like ABN AMRO or ING.

Privacy of peers has been implemented in Tribler with anonymous P2P file sharing. A protocol is created inspired by TOR anonymous routing. The protocol first creates a circuit with specific messages that also exchange a session key between nodes in the circuit using Diffie Hellman key exchange. The messages in circuit creation are encrypted using elliptic curve assymetric encryption. The public keys of all the nodes required for the assymetric encryption are maintained and distributed by dispersy. The data messages of the file sharing are encrypted with AES using the session keys between every node. [3] [4]

A new TOR inspired protocol will be created with a secure matching engine and payment system that ensures privacy. The matching engine should make it impossible for traders to deduce the party that places bid and ask offers in the market. Also, the payment system should be made in such a way that the counterparty in a trade is not known to the trader. By not knowing with who you are trading no sensitive information is leaked and the privacy of all traders is ensured. The payments should be done indirectly via other peers and not directly with the counter-party.

1.1. DDoS problem

The availability of the system can be easily undermined by attacking the system with a distributed denial of service attack (DDoS). Adversaries can create a large number of peers and let these peers do offers without accepting trade proposals. The peers of normal users cannot handle the amount of offers from peers controlled by adversaries and the system will become unavailable to normal users. A solution would be to let peers pay a tiny amount for sending offers. The amount to be payed should be low enough to make the usage of the system cheap to make the system available to a large number of users and the amount should be high enough such that the amount to be payed becomes significant to prevent a DDoS attack.

1.2. LATENCY COMMUNITY

1.2.1. THE IMPORTANCE OF LOW LATENCY

In the past 30 years, trading has become faster. The time it takes to process a trade has gone from minutes to seconds to milliseconds. "Low Latency" would be under 10 milliseconds and "Ultra-Low Latency" as under one millisecond . It is estimated that 50% of trades in the U.S. are done in high frequency trading with an "Ultra-low latency". Thus, low latency is a major differentiation factor for exchange firms. Some firms state that a 1 millisecond advantage can save an exchange firm 100 million U.S. dollars. [5] An individual trader has numerous advantages when using trading in a system with low latency: [6]

1. Better decision making: A trader makes trading decisions based on the information the trader has from the market. Other traders send the prices and quantities they offer as orders to other traders. Let's say these traders maintain these orders in an order-book. If these orders arrive later, the individual trader is limited in it's trading decision making.

2. Competitive advantage towards other traders: When an individual trader can trade relatively faster than another trader due to low latency it has a competitive advantage. Let's say a price differentiation takes place, a price suddenly becomes lower. A trader with a relatively lower latency can act on it earlier than it's competitors and take advantage of the lower price before a price correction takes place.

3. Lower latency traders are served with a higher priority. Offering a lower price gives a trader always a higher priority as other traders would buy a product with a lower price faster. However, when the price is the same. The offer that arrives first is served. A trader with a high latency needs to lower its price in order to get a higher priority. If the high latency trader does not lower its price it is simply not served. Also, offers at the same price level with a higher priority have less adverse selection. [7] [8]

Moallemi and Saglam (2013) estimate the latency cost based on cross-sectional data on volatilities and bid-offer spreads in the U.S. between 1995-2005 from the dataset of Ait Sahalia and Yu (2009). The results can be seen in 1.1. The median latency cost approximately increased threefold in the 1995-2005 time period. To obtain the latency cost estimation the data set is used in a model that under simplifications calculates the latency cost. The model assumes an individual trader with a fixed latency of 500ms. As time increases, the cost for this latency also increases. As can be seen later on, the Tribler market has latencies around 150 ms. The assumption of a trader with 500ms is realistic in the Tribler context. For details of the model we refer to the paper of Moallemi and Saglam (2013). [5]

1.2.2. LATENCY IN TRIBLER

The latency of Tribler applications appears to be around 150ms normally. There are however outliers of latencies of 10 seconds. The normal latency response of 150ms is high for a exchange market but explained by the distributed nature of the Tribler market. Other exchange markets that are considered low latency have latencies around 10 ms. The outlier latencies of 10 seconds are unacceptable in the market application. These super high latencies result almost directly in the problems described by Cespa and Foucault, 2009. 1) Competitive advantage for other traders 2) Bad decision making from traders due to incomplete information and 3) Low priority serving because another trader gets served earlier due to the first come first served principle. [6]

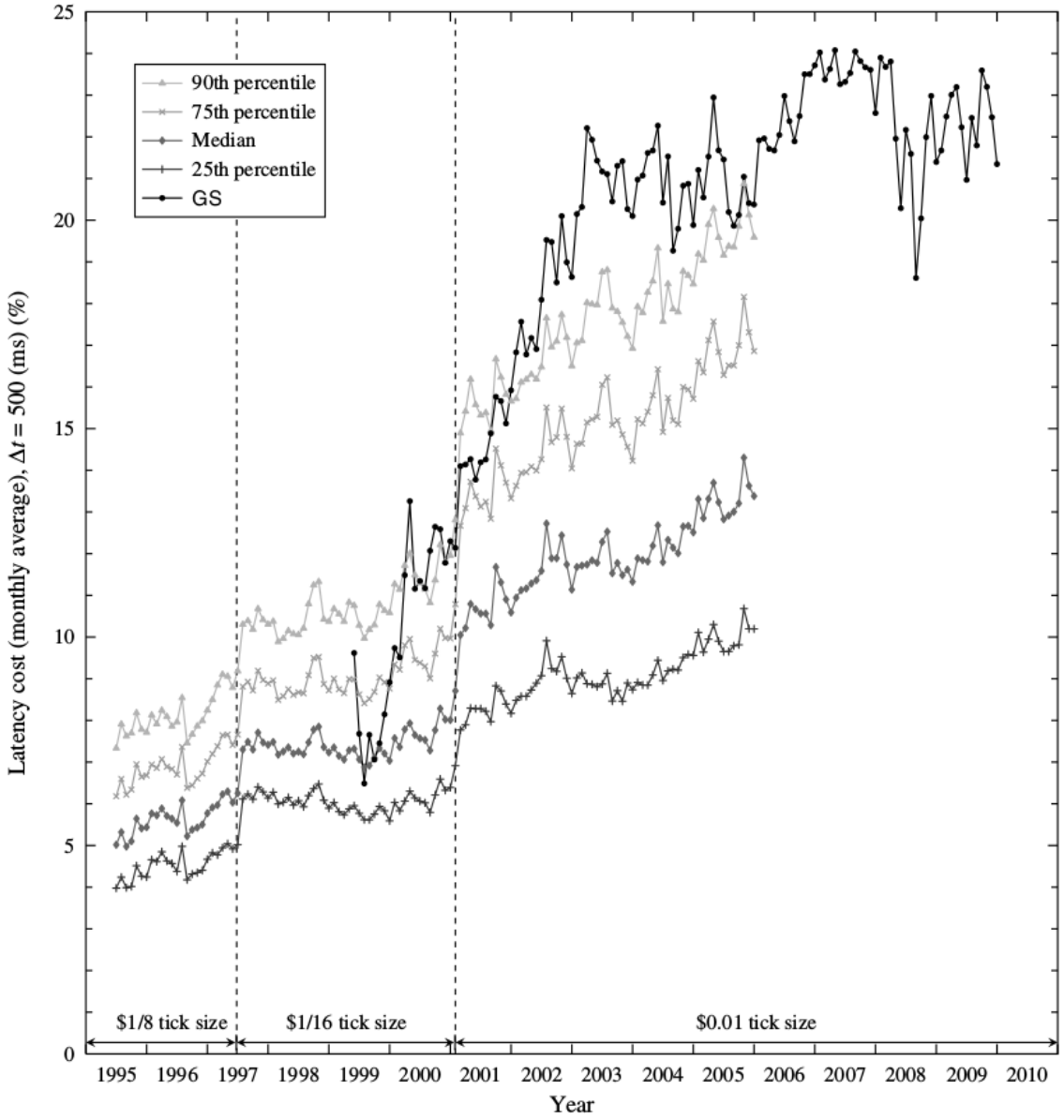

Figure 1.1: A hypothetical investor with a fixed latency of 500 ms is assumed. The latency costs are computed from the data set of Ait Sahalia and Yu (2009) The latency cost for GS is also reported, beginning from its IPO. The dashed lines correspond to dates where the NYSE tick size was reduced. The latency cost had a consistent increasing trend over the 1995-2005 period. The median latency cost approximately increased threefold by reaching roughly 14% from 5%.

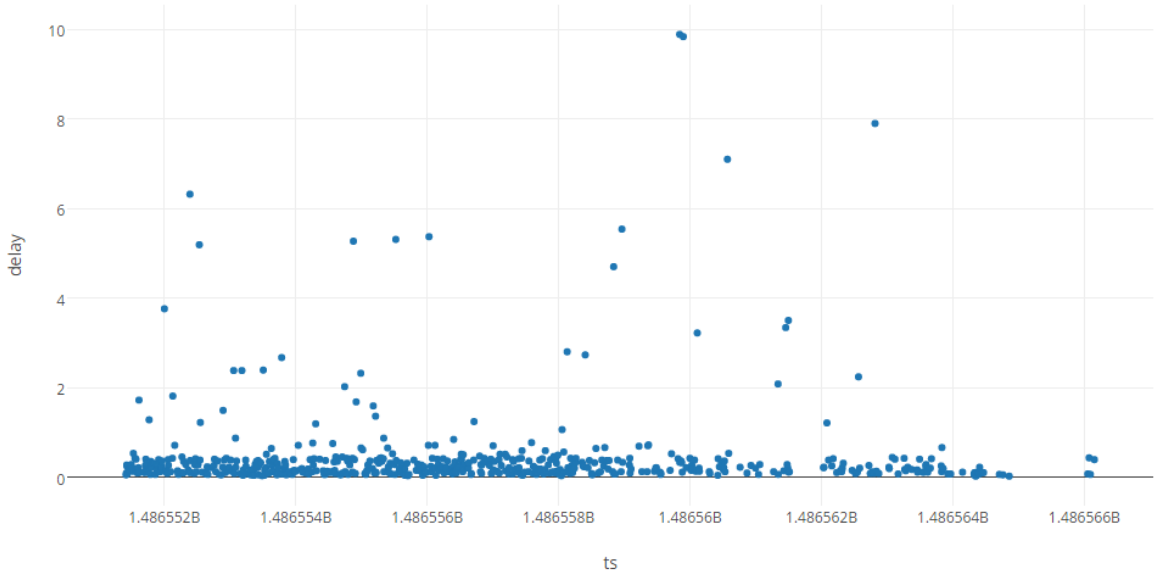

Figure 1.2: Delays while waiting for similarity responses for 4 hours in a real world Tribler application. [9]

2

PRIVACY ENHANCING TECHNOLOGIES

2.1. CHAUM MIXES

Chaum , D.L. first published about anonymization techniques in 1981 now known as *mix networks.* [10] The purpose of mix networks is to unlink the sender and receiver of messages. A mix is a node in the network with its own public/private key pair. Messages are send towards mixes encrypted with the public key of the mix. The mix hides the correspondence between incoming and outgoing message. To achieve this the mix does three things:

1. Incoming messages are batched together and send in one batch.
2. The mix strips of the encryption layer of incoming messages with its private key and forwards messages to another mix or to the final destination node of the messages.
3. The order of the messages is permuted.

A mix network is a series of mixes connected together. More mixes in the network make the unlikability property stronger but result in a higher latency.

The identity of the next recipient in the network is encrypted together with the message to let the mix know to which node it has to send the next batch.

$E_{MIX}(message, A) \xrightarrow{MIX} message, A$

Thus the encryption for a mix network of three layers looks the following.

$E_{MIX_1}(E_{MIX_2}(E_{MIX_3}(message, A), MIX_3), MIX_2), MIX_1) \xrightarrow{MIX_1} E_{MIX_2}(E_{MIX_3}(message, A), MIX_3), MIX_2), MIX_1$

$E_{MIX_2}(E_{MIX_3}(message, A), MIX_3), MIX_2) \xrightarrow{MIX_2} E_{MIX_3}(message, A), MIX_3), MIX_2$

$E_{MIX_3}(message, A) \xrightarrow{MIX_3} message, A$

Because messages are batched together mix networks require that a (threshold) mix has to wait until N messages are arrived to forward a new batch of messages. This gives a high latency to the system. In a timed mix the mix forwards every t seconds. If a limited number of messages arrive in the time interval the mix loses its unlinkability property. For instance, if one message arrives in the time interval it can be easily linked to the only outgoing interval. To solve this problem dummy messages with no meaning can be send into the network. Dummy messages also lower the latency and make the unlinkability property in a threshold stronger.

In a *Trickle attack* the adversary can slow down messages that are send into the mix to ensure only one message is send into a timed mix every t seconds. The *Flooding attack* injects $N-1$ messages in a threshold mix and then distinguishes its own injected message from other messages.[11] [12]

2.2. TOR ONION ROUTING

TOR onion routing is a method developed by Dingledine, R. *et al* that like mix networks also aims to provide anonymity for users but operates at a lower latency compared to mix networks. The onion routers are real

time mix networks. Messages are not batched together but passed on nearly in real-time. This makes TOR onion routing vulnerable to the global passive attack where peers sniff all the network traffic and can then link sender and receiver to each other. When only parts of the network can be sniffed, TOR onion routing still provides anonymity.

Clients create a path through the network where each node only knows its predecessor and successor node in the path. The end node connects with the recipient of the messages. Session keys are negotiated between each pair of successive nodes in the path to ensure "Perfect forward secrecy" With "Perfect forward secrecy" a hostile node cannot record traffic and decrypt it later at another compromised node in the network.

3

System Design

3.1. Latency Community

The latency community has 4 distinct packets. Ping Pong Request Latencies Response Latencies

3.2. Enhancing privacy

3.2.1. Privacy of traders in matching engine

A matching has to be found by broadcasting the price and quantity details towards other peers. Peers gossip the information towards each other. In this broadcasting process a path between two traders is made via other peers in the peer to peer network. The path creates a tunnel like in the design of the TOR protocol and chaum mixes. The path is used in all future communication between the two traders to ensure privacy. A session key is shared using Diffie Hellman key exchange between the two peers in the tunnel to ensure privacy against the 3 peers that facilitate the tunnel. The session key is shared using Diffie Hellman key exchange. [10] [13]

The first step in the matching process is the broadcast of a bid or ask towards other peers in the network as shown in Figure 3.1. The price and quantity (qtt) details of the bid or ask are first encrypted with the private key of the sending peer to let the receiving peer make sure the match is coming from the sending peer. A second layer of encryption is added with the public key of the receiving peer to ensure that only the receiving peer can read the information of the match. The match is three times forwarded towards other peers to make the tunnel with three peers into it. The time to live (ttl) field maintains how many times the match is forwarded. Also the first part of the Diffie Hellman key exchange A and a unique random number n_i to distinguish between peers to which the match is forwarded is calculated and send with the broadcast. The peer saves the peer to which the match is forwarded in the tuple (m_{id}, n_i) where m_{id} is the match id. For example in Figure 3.1 P1 would save P2 in the tuple (m_{id}, n_i). This information is later used to distinguish between multiple matches made with one broadcast. Also the m_{id} is saved tell from which peer a broadcast was coming. For example $P2$ would let m_{id} correspond to $P1$ because the match with m_{id} was coming from $P1$.

When after three hops a match is found the second step starts and the matching peer sends a proposed trade back towards the broadcasting peer via the tunnel. The second part and the session key of the Diffie Hellman key exchange is calculated. Because multiple matches can be made there will be multiple unique session keys. The proposed trade is encrypted with the session key K and is send back into the tunnel together with the second part B of the Diffie Hellman key exchange. The broadcasting peer receives the second part B of the Diffie Hellman key and calculates the session key K to decrypt the proposed trade. The communication to accept a trade, decline a trade or propose a counter-trade between the two trading peers at the end of the tunnel is from this point in time done with the session key that both ends know.

To distinguish between multiple matches in the same broadcast the tuple (m_{id}, n_i) was saved that tells to which peer the broadcast was send. A path identifier (m_{id}, n_i, n_j, n_k) is created on the way back from matched peer to the broadcast peer and can be used to distinguish between paths on the way forward from the broadcast peer. Thus (m_{id}, n_i) tells the first peer who is the next peer in the path. (m_{id}, n_j) tells the second peer the next peer in the path and (m_{id}, n_k) tells the third peer the last peer in the path. The m_{id} is used by a peer to go back toward the broadcasting peer. An overview of the second step is given in figure 3.2

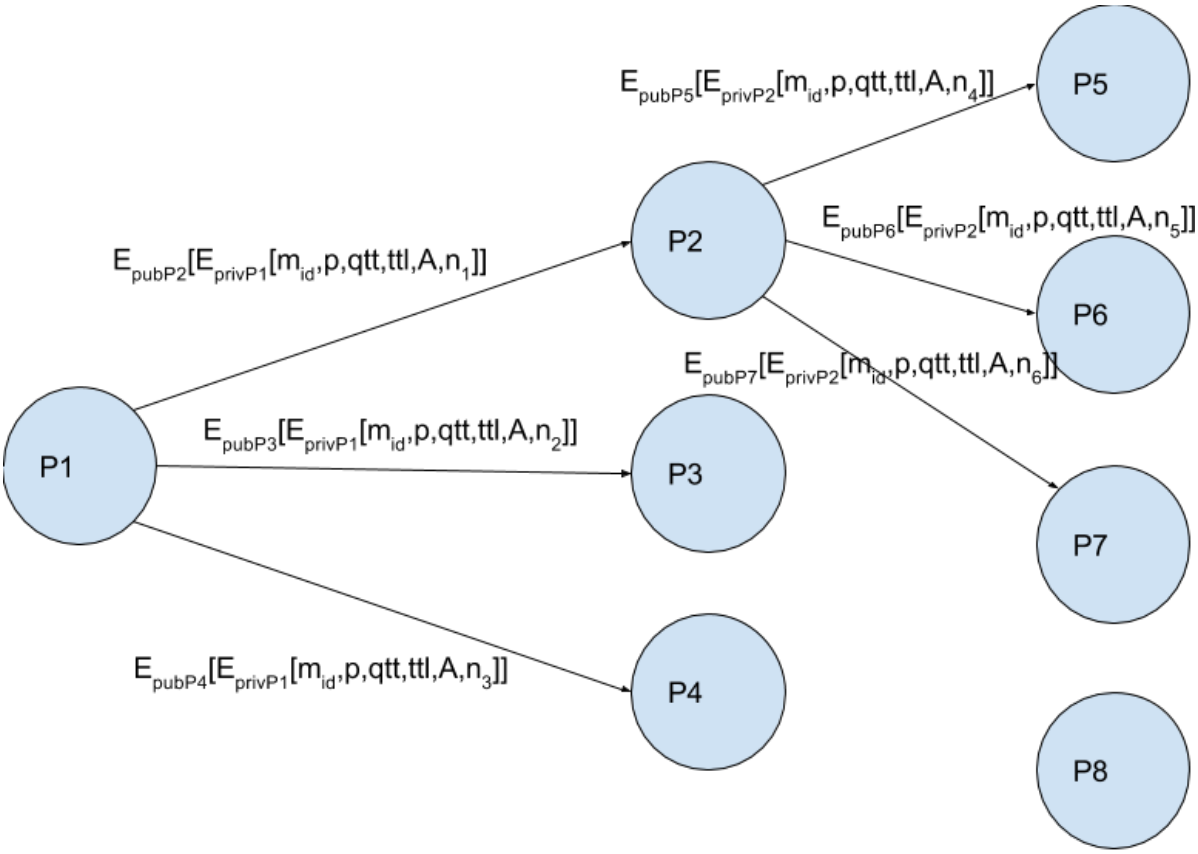

Figure 3.1: Broadcast of bid or ask match request towards other peers.

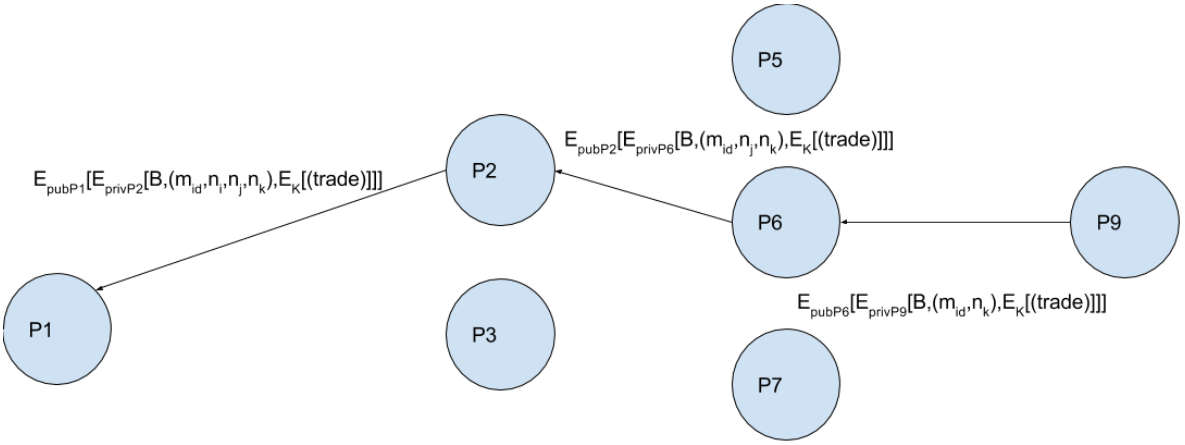

Figure 3.2: Match send back towards broadcaster. The path identifier is created upon hopping back

GNP is often more accurate on a small number of hosts. Calculating GNP with a large number of nodes takes a lot of computation time. Therefore a small number of nodes is required. The only requirement for the nodes is to remove outliers. This should be feasible with a small number of nodes.

3.2.2. REQUIREMENTS

- Proxies in trading to ensure privacy.
- No Trader id in first offer.
- Encrypted trader id in proposed trade.
- Reputation system to prevent DDoS attacks.
- Reputation reward upon contributing in a proxy trade.
- Only trade with low latency peers to prevent DDoS attacks.

3.2.3. OPTIONAL REQUIREMENTS

- Multiple relays in proxies
- Use proxies in negotiating proposed trade.

3.2.4. TESTS

- DDoS test with a large number of peers doing trade offers to the system.

4

EXPERIMENTS

The following figures show that applying the GNP algorithm has scalability problems.

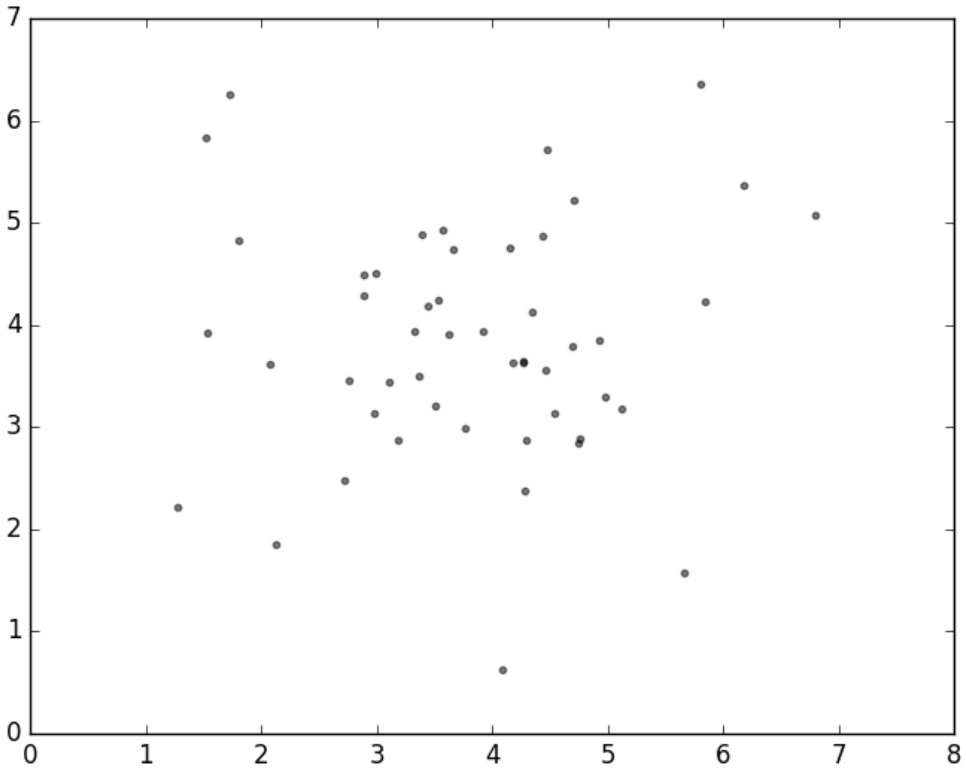

Figure 4.1: Plot of landmark coordinates after applying the GNP algorithm to the king dataset with 50 nodes.

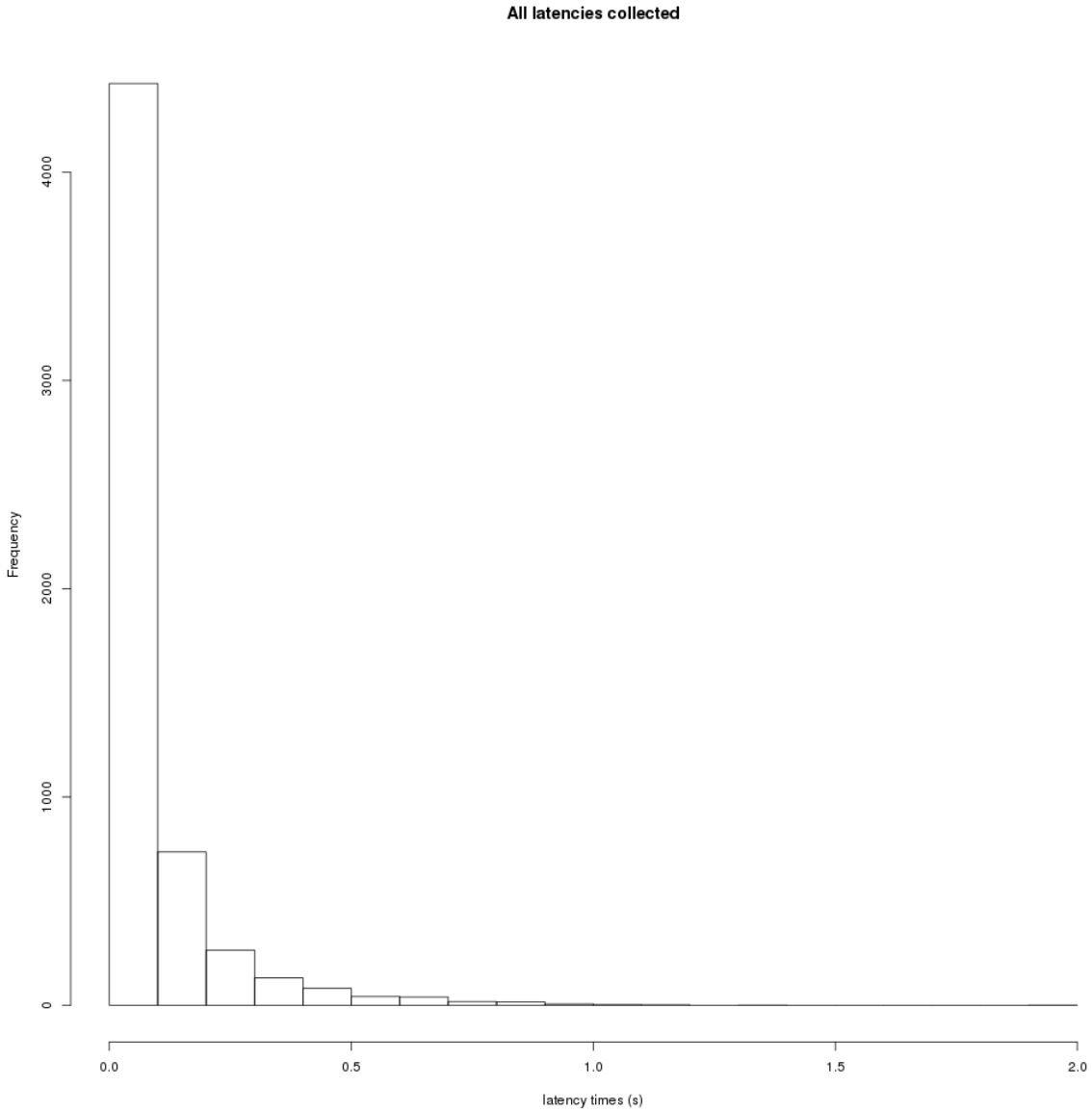

Figure 4.2: Histogram of all collected latencies after application of the decentralized GNP algorithm described by Szymaniak et al, 2004 [14] in a Tribler environment with 75 nodes. The system appears to function normal. The algorithm does not add extra latency to the system.

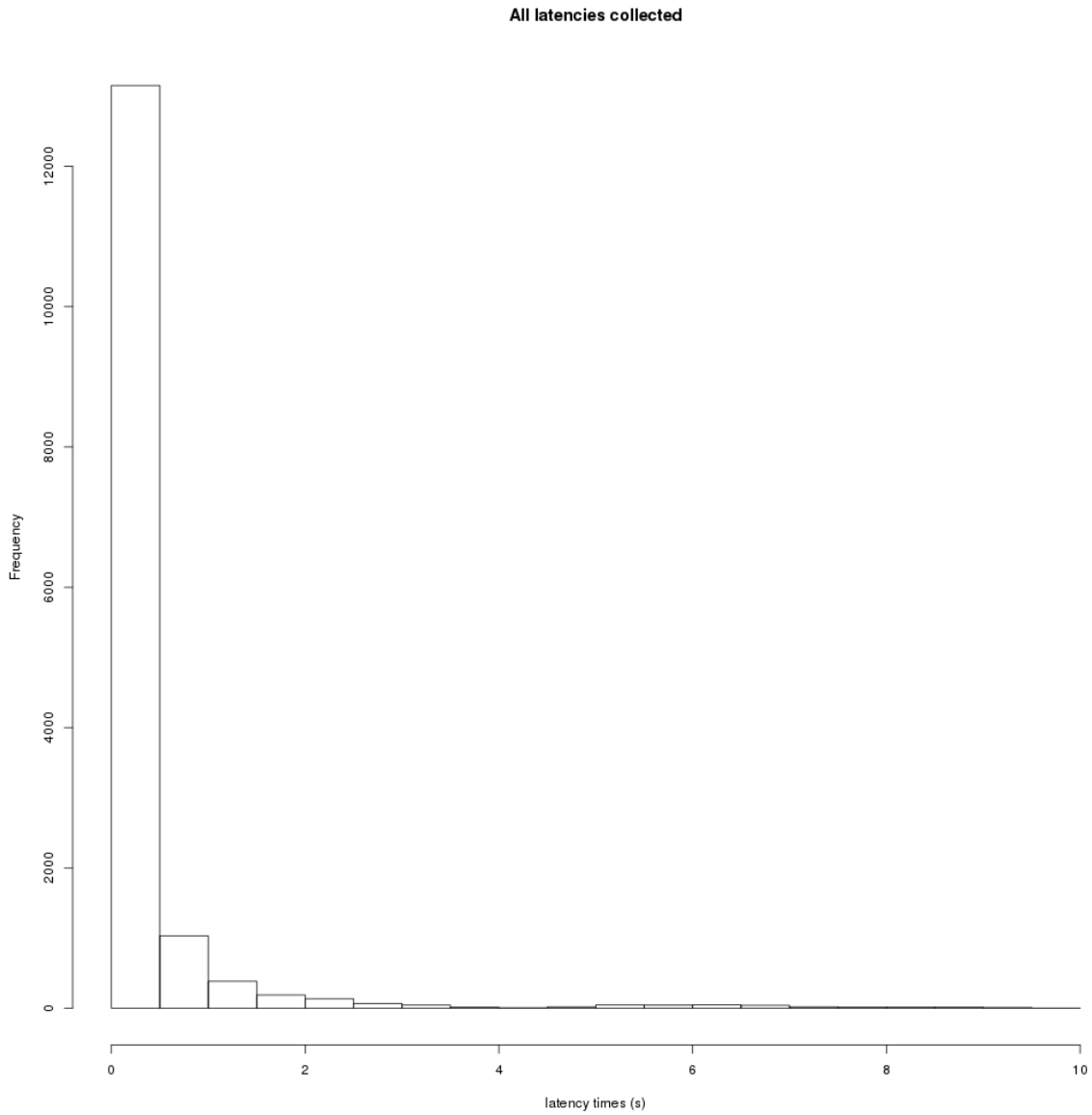

Figure 4.3: Histogram of all collected latencies after application of the decentralized GNP algorithm described by Szymaniak et al, 2004 [14] in a Tribler environment with 100 nodes. The system has some latencies higher than 1 second. The algorithm appears make the system slower.

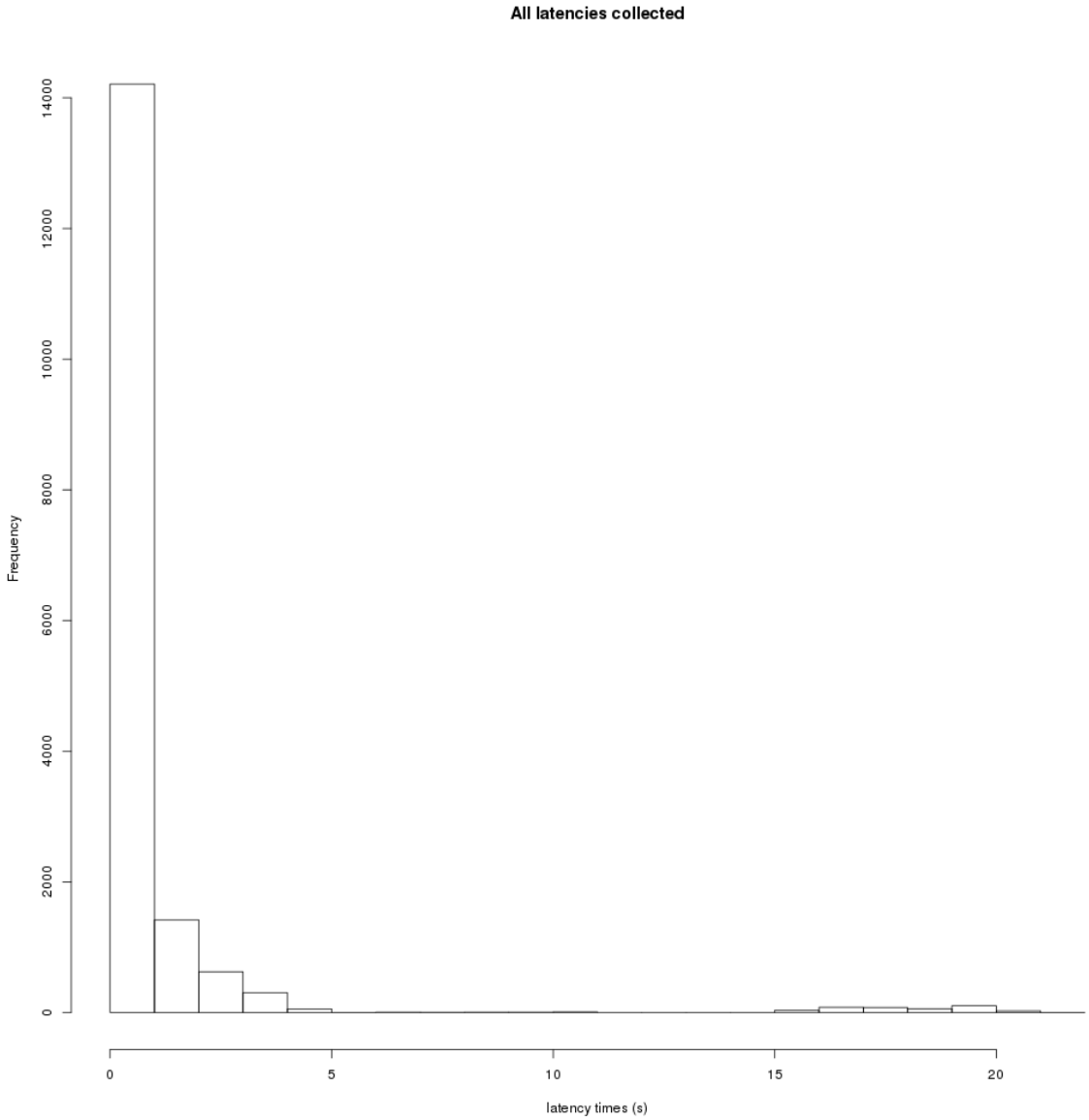

Figure 4.4: Histogram of all collected latencies after application of the decentralized GNP algorithm described by Szymaniak et al, 2004 [14] in a Tribler environment with 75 nodes. The system has a lot of latencies higher than 1 second. The algorithm appears to disrupt the system because of the long calculations.

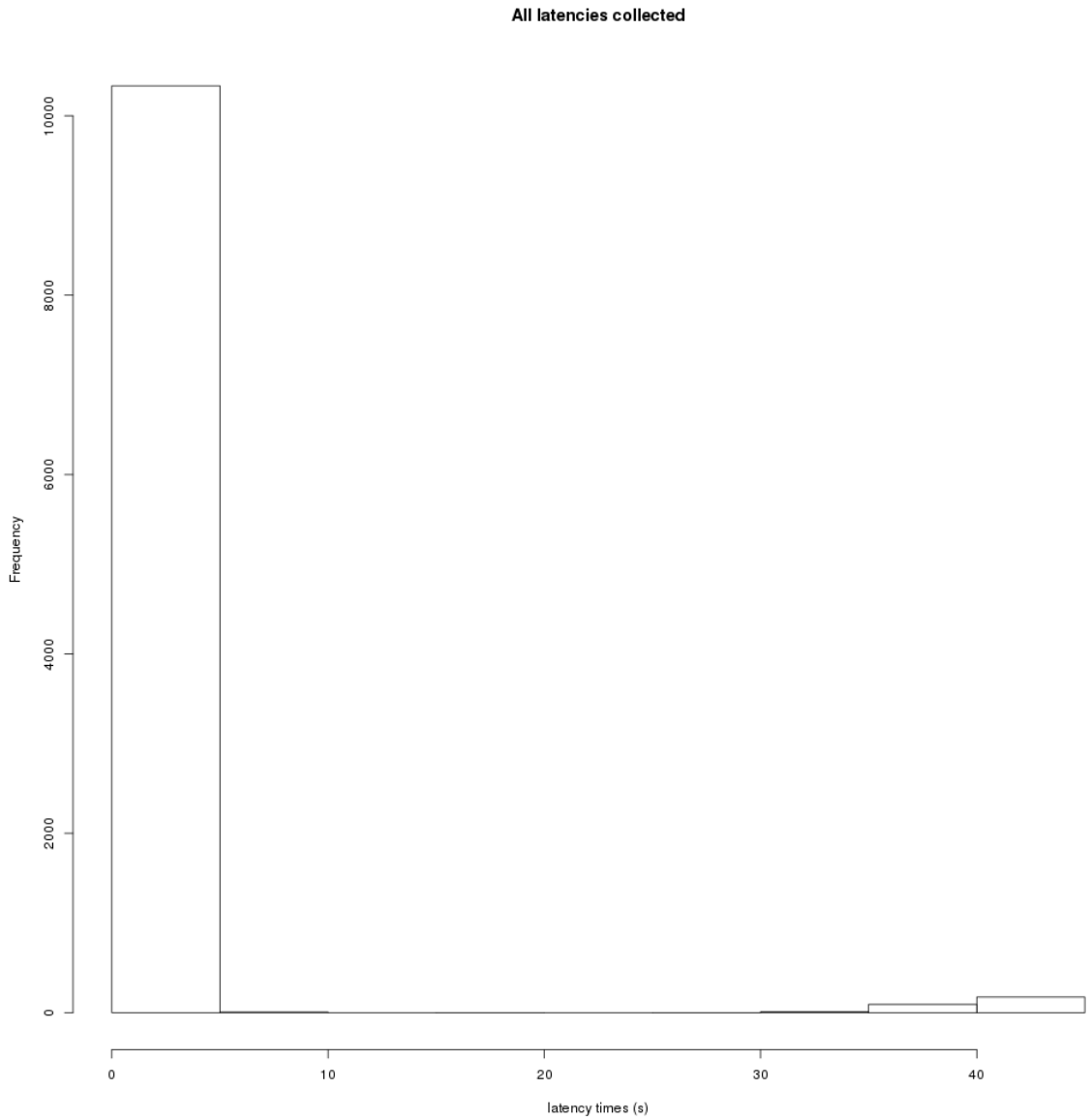

Figure 4.5: Histogram of all collected latencies after application of the decentralized GNP algorithm described by Szymaniak et al, 2004 [14] in a Tribler environment with 100 nodes. The system has a lot of latencies higher than 1 second. The algorithm appears to disrupt the system because of the long calculations.

BIBLIOGRAPHY

[1] Olsthoorn, M.J.G.,Winter, J., *Decentral market,* (2016).

[2] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, *Zerocash: Decentralized anonymous payments from bitcoin,* in *Security and Privacy (SP), 2014 IEEE Symposium on* (IEEE, 2014) pp. 459–474.

[3] Rutger Plak, *Anonymous internet,* (2014).

[4] Risto J.H. Tanaskoski, *Anonymous hd video streaming,* (2014).

[5] C. C. Moallemi and M. Sağlam, *Or forum—the cost of latency in high-frequency trading,* Operations Research **61**, 1070 (2013).

[6] G. Cespa and T. Foucault, *Insiders-outsiders, transparency and the value of the ticker,* (2009).

[7] L. R. Glosten, *Is the electronic open limit order book inevitable?* The Journal of Finance **49**, 1127 (1994).

[8] P. Sandås, *Adverse selection and competitive market making: Empirical evidence from a limit order market,* The review of financial studies **14**, 705 (2001).

[9] .

[10] D. L. Chaum, *Untraceable electronic mail, return addresses, and digital pseudonyms,* Communications of the ACM **24**, 84 (1981).

[11] A. Peter, *Anonymous communication,* (2016).

[12] A. Serjantov, R. Dingledine, and P. Syverson, *From a trickle to a flood: Active attacks on several mix types,* in *International Workshop on Information Hiding* (Springer, 2002) pp. 36–52.

[13] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The second-generation onion router*, Tech. Rep. (DTIC Document, 2004).

[14] M. Szymaniak, G. Pierre, and M. van Steen, *Scalable cooperative latency estimation,* in *Parallel and Distributed Systems, 2004. ICPADS 2004. Proceedings. Tenth International Conference on* (IEEE, 2004) pp. 367–376.