On the Peer Degree Distribution of the Bitcoin P2P Network

Matthias Grundmann, Max Baumstark, Hannes Hartenstein
Institute of Information Security and Dependability (KASTEL)
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

***Abstract*—A recent spam wave of IP addresses in the Bitcoin P2P network allowed us to estimate the degree distribution of reachable peers. The resulting distribution indicates that about half of the reachable peers run with Bitcoin Core's default setting of a maximum of 125 concurrent connections and nearly all connection slots are taken. We validate this result empirically. We use our observations of the spam wave to group IP addresses that belong to the same peer. By doing this grouping, we improve on previous measurements of the number of reachable peers and show that simply counting IP addresses overestimates the number of reachable peers by 15 %. We revalidate previous work by using our observations to estimate the number of unreachable peers.**

I. Introduction

To join the Bitcoin [1] P2P network, new peers need to find peers that are already part of the network. Each peer has one or multiple addresses that can be used to find and initiate connections to the peer. Bitcoin uses a decentralized approach to disseminate addresses to peers: A peer announces its own addresses by sending ADDR messages to its neighbors and, based on certain conditions, the neighbors forward the addresses to other peers. In July and August 2021, a huge wave of addresses was flooded in the Bitcoin P2P network that caused an increase in the number of addresses distributed per day from 40,000 to about 6,000,000 unique addresses per day [2]. These spam addresses did not belong to actual peers and were sent by an unknown party. While we do not know the purpose of sending the spam addresses, we look at the effects that the spamming had and what information about the topology of the Bitcoin P2P network can be learned by observing the effects. We estimate the degree (number of neighbors) of reachable peers. While previous work has shown that the peer degree distribution of other cryptocurrencies' P2P networks resembles a power law distribution [3]–[5], our observations indicate that the peer degree distribution of the Bitcoin P2P network is different and about half of the peers have a degree of around 125. Because most peers run Bitcoin Core [6] and 125 is the default maximum for connections in Bitcoin Core [7], this finding means that many peers do not have slots available for new incoming connections. We run an experiment to validate this observation and find that more than 50 % of all reachable peers do not accept additional incoming connections or are close to their connection limit. We also show that the majority of peers being hosted in the networks of cloud providers have around 125 connections while the networks of ISPs include peers that tend to have fewer neighbors. Further, we find sets of addresses that belong to the same reachable peers. This mapping shows that estimating the number of reachable peers by counting reachable addresses overestimates their number by about 15 %. Finally, we discuss a coarse-grained model for the allocation of connection slots in the Bitcoin P2P network. Based on this model, we estimate that there are about 32,300 unreachable peers in the network which aligns with estimations from previous work [8]–[10].

Related Work. While different methods to learn about the topology of the Bitcoin P2P network have been proposed, most of them were impractical or too costly to be run in the real Bitcoin P2P network. A notable exception is AddressProbe [11] that exploited an information leak in the handling of addresses to infer connections between reachable peers. Miller et al. used AddressProbe to infer the topology of the P2P network's subgraph that contains only reachable peers and calculated the resulting peer degree distribution [11]. The degree distribution showed that the majority of reachable peers had a degree between eight and twelve which differs strongly from our results because our results also include connections between reachable and unreachable peers. Other methods to infer parts of the topology [3], [12]–[14] were too expensive to be run in the Bitcoin P2P network. Topology inference approaches have also been proposed for other cryptocurrencies' P2P networks [4], [5], [15], [16] and the peer degree distributions of the P2P networks of the Bitcoin testnet [3], Monero [4], and Ethereum [5] have been analyzed. The Bitcoin transaction network, sometimes simply referred to as the 'Bitcoin network', is the graph defined by the transactions in the Bitcoin blockchain. The topology of this network has been analyzed previously [17]–[22] but the transaction network is completely different from the Bitcoin P2P network which is the focus of this work.

II. Observations and Monitoring Setup

In July 2021, user piotr_n reported in the BitcoinTalk Forum [23] that spam addresses were distributed in the Bitcoin P2P network. piotr_n found that the behavior of the spamming peers is to connect to reachable peers, send them 500 ADDR messages with ten spam addresses each, and then disconnect. We observed the behavior described by piotr_n at a reachable

This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

978-1-6654-9538-7/22/$31.00 ©2022 IEEE

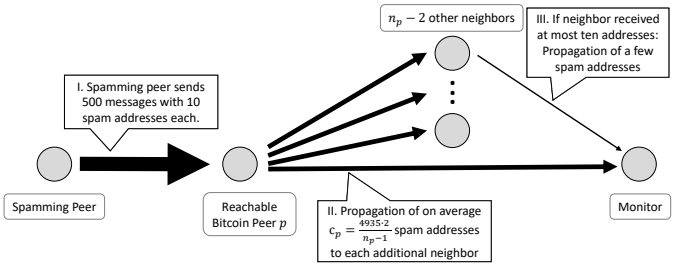

Fig. 1. Overview of the peer degree estimation. A reachable peer p is connected to a spamming peer, our monitor and $n_p - 2$ other peers. The spamming peer sends $500 \cdot 10$ addresses to the reachable peer (I). The peer propagates the addresses to all neighbors except the spamming peer (II). From the number of propagated addresses, the monitor can estimate the number of neighbors.

peer: During July and August 2021, about 400 times one of this peer's neighbors sent within a few seconds a batch of 5,000 unique IPv4 addresses. Over the observed time, the spam originated from 243 different IP addresses. All spam addresses in a batch had the same associated timestamp which was set to a value up to nine minutes into the future. We analyzed the distribution of the received spam addresses and found that they were distributed uniformly over the IPv4 address space and included IP addresses from reserved IPv4 address blocks like 127.0.0.0/8. We take this finding as evidence that the spam addresses were randomly chosen.

Our monitoring setup consists of three monitor nodes that connect to all reachable peers but do not accept incoming connections. Two of those monitor nodes are located in the network of our university (AS 34878) and a third monitor node is located in a different location (AS 680). All monitor peers log received ADDR messages and connections to other peers that are opened or closed.

III. Estimating the Degree of Reachable Peers

A reachable peer that receives the spam addresses from a spamming peer and is connected to one of our monitor nodes, propagates an approximately equal part of the addresses to each of the reachable peer's neighbors. From the number of spam addresses propagated to our monitor, we can estimate the number of neighbors of the reachable peer:

Bitcoin Core, the Bitcoin reference client that is used by most peers [6], accepts addresses with an associated timestamp of up to ten minutes into the future and propagates addresses until their associated timestamp is older than ten minutes. Additionally, an address is only propagated if it was received in an ADDR message with at most ten entries. Because both conditions are met when a peer receives the spam addresses, a peer that runs Bitcoin Core considers these addresses for propagation to its neighbors. However, Bitcoin Core forwards only routable addresses and about 1.3 % of the IPv4 address space are considered as unroutable [24]. Therefore, on average $4,935$ addresses of the 5000 received addresses are forwarded. Because each routable address is forwarded to two peers but not the peer that the address was received from, a peer p with n_p neighbors forwards each address to two out of $n_p - 1$ neighbors and sends on average $c_p = 4,935 \cdot \frac{2}{n_p - 1}$ addresses to each neighbor. Consequently, our monitor nodes receive on average c_p addresses from each peer that receives 5,000 spam addresses and we can estimate the number of neighbors of each reachable peer based on these observations (see Fig. 1). While the main idea of this estimation approach has been proposed in 2014 by Biryukov et al. [12, Section 10.1], to the best of our knowledge results of this method applied to the Bitcoin P2P network have so far not been published.

A. Estimation and Validation

Our monitor nodes are connected to each reachable peer and receive the propagated spam addresses (see Fig. 1, II). However, our monitor nodes also receive spam addresses that are not directly forwarded from a spamming peer (Fig. 1, III). To filter out these indirectly forwarded messages, (1) we analyze only ADDR messages received at the monitor that contain at least four entries, (2) we select only those addresses that have a timestamp that is three to ten minutes into the future from the point when the ADDR message was received and (3) we analyze only addresses if $c_{p,t}$, the number of addresses we received with the same timestamp t from peer p, is greater than ten. For each batch of spam address messages with size $c_{p,t}$, we calculate $n_{p,t} = 1 + 4,935 \cdot 2/c_{p,t}$ as an intermediate estimate for the number of neighbors of peer p. As the intermediate estimates contain outliers, we calculate the estimate n_p for the number of neighbors of peer p by determining the median of all intermediate estimates $n_{p,t}$ during the time window of one day. The length of this time window is chosen as a trade-off between a short time window during which the number of a peer's neighbors remains constant and a longer time window during which we collected more observations to receive a more precise estimate.

To validate the estimation approach, we logged at three reachable validation peers the number of neighbors and compared the logs to our estimation. As ground truth we take for each peer the peer's average connection count per day. We compute the mean deviation of each estimate from this ground truth in percent and average the absolute percentage points. This calculation leads to an average deviation of 4.1 % which means that the estimation is reasonably reliable.

B. Resulting Degree Distribution

One important topological characteristic of a network is the distribution of peer degrees which can be estimated based on our observations. Using the method described above, we receive one estimate per peer per day. To calculate the distribution of peer degrees across the Bitcoin P2P network, we calculated a histogram (see Fig. 2) of all estimates during the observation time period. The distribution shows that the majority of reachable peers have an estimated degree of around 125, which is the default maximum number of connections in Bitcoin Core. These results suggest that about 50 % of the reachable peers use this default configuration and all of their connection slots are filled. The distribution of estimated peer degrees has a long tail of a few peers that have more than

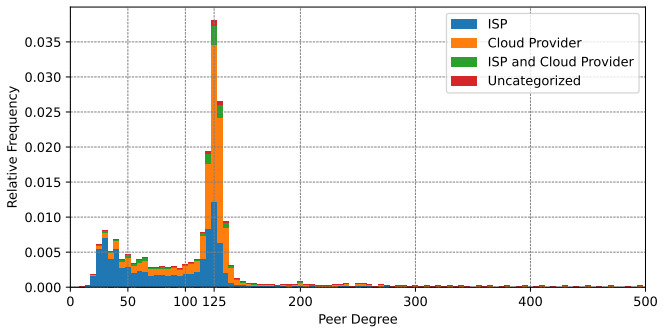

Fig. 2. Normalized histogram (with bin width of 5) of the peer degree of all reachable peers estimated from our observations of the spam wave during July and August 2021. The peak at 125 represents $5 \cdot 3.8\,\% = 19\,\%$ of all peers. The colors indicate our categorization of a peer's autonomous system.

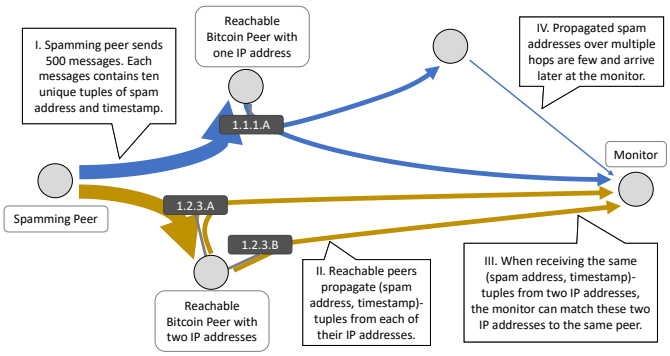

Fig. 3. A peer with multiple reachable IP addresses is connected to the monitor with each IP address. By observing the propagated spam, the monitor can find IP addresses that belong to the same peer.

140 connections. We suppose that there are reachable peers with even more connections, however, we can only estimate the degree of peers with up to 1,000 connections.[1]

We looked up the autonomous system (AS) of each peer's address using Team Cymru's IP to ASN Mapping Service [25] and categorized each AS into the four categories 'ISP' (Internet Service Provider), 'Cloud Provider', 'Both', and 'Uncategorized'. We manually classified ASes that contain a large percentage of peers and retrieved the category for the remaining ASes from the ASdb [26] database. Figure 2 shows the distribution of peer degrees separated by the category of a peer's AS. While the median of estimated degrees for peers hosted at cloud providers is 125, the median of estimated degrees for peers located in networks by ISPs is 97. One reason might be that peers running in data centers accumulate more incoming connections because they are less often restarted and their addresses are better distributed in the network because they change their address less often than other peers.

C. Measurement of Available Slots for Incoming Connections

We validate the observation that many reachable peers do not have slots for incoming connections available using the following experiment. A reachable peer running Bitcoin Core always accepts a new incoming connection but, if the new connection fills the last remaining connection slot, a connection is evicted. The evicted connection might be the connection that was just accepted but it might also be a previously existing connection.

In our experiment, we run a test peer that walks through a list of all reachable peers and opens a TCP connection to each peer. If a connection was established, the test peer waits for three seconds and checks if the connection is still open. If it is, the test peer opens four additional TCP connections to this peer, waits for three seconds and checks whether all five connections are still open.

[1]This restriction is due to the condition that $c_{p,t}$ must be greater than ten to distinguish between addresses that were directly forwarded after being received from a spamming peer and addresses that were received from another peer. With the knowledge of which spam addresses are sent to which peer, this restriction is not necessary and higher degrees can also be estimated.

We ran the experiment in November 2021 from three test peers located in two different ASes. To create the list of reachable peers, we collected all addresses that we had received in unsolicited ADDR messages at one of our monitors on the day before. Our test peers were able to connect to on average 9,461 peers of which 4,493 (47 %) accepted all five incoming connections. On average, 2,360 (25 %) accepted the first connection but not all five connections and 2,608 (28 %) evicted already the first connection. We conclude that for 28 % of the reachable peers the slots for incoming connections are all taken while 25 % of the reachable peers are close to their capacity. Only 47 % of the reachable peers seem to freely accept incoming connections. This result confirms our interpretation of the peer degree distribution and shows that slots for incoming connections are a limited resource.

IV. Finding Peers with Multiple IP Addresses

When counting the peers of the Bitcoin P2P network, it is a typical assumption that each IP address belongs to exactly one peer (cf. [8], [10], [27]–[31]). This assumption is not fulfilled for the Bitcoin P2P network but, to the best of our knowledge, there is no general practical way to find IP addresses that belong to the same peer. However, our observation of the spam wave allowed us to obtain such a grouping of IP addresses.

As far as we observed, the 5,000 spam addresses that are sent to one peer with one timestamp are not sent to another peer with the same timestamp. Thus, a batch of spam addresses with the same timestamp mark a specific peer and we can use these markers to find multiple IP addresses that belong to the same peer (see Fig. 3). A spammed reachable peer with multiple IP addresses forwards the spam addresses on all of its IP addresses (Fig. 3, II). If tuples of spam address and timestamp are received by the monitor from two different IP addresses, we can match these IP addresses to the same peer (III). False positives can occur if spam addresses are propagated over multiple hops (IV). To filter out these indirectly received spam addresses, we ignore spam addresses that are received in an ADDR message that contains ten or fewer addresses and we ignore spam addresses that have a timestamp less than five minutes into the future. Further, we

only match two IP addresses to the same peer if there are more than five tuples of spam address and timestamp that were received by the monitor from both IP addresses.

We run this analysis on data collected by our monitor nodes and obtain sets of IP addresses that belong to the same peers. After merging all intersecting sets of IP addresses that belong to the same peer, we obtain a mapping of 3,614 IP addresses to 1,449 peers. While there seems to be one peer having 286 IPv6 addresses of the same /118 subnet, the majority of peers (89 %) have only two addresses. Most of these pairs of addresses are an IPv4 and IPv6 address, however, there are some pairs that are both IPv4 or IPv6 addresses. We validate the method using three of our peers that are using an IPv4 and IPv6 address and find that their IP addresses were correctly matched.

In August 2021, our monitor nodes were connected on average to 8,647 reachable IP addresses per day of which on average 2,220 IP addresses belonged to 1,091 of the 1,449 reachable peers with multiple IP addresses. Hence, our monitor nodes were connected on average to 7,518 unique reachable peers per day which shows that estimating the number of reachable peers by counting reachable IP addresses overestimates the number of reachable peers by $\frac{8{,}647-7{,}518}{7{,}518} \approx 15\,\%$.

V. Sum of Peer Degrees and Relation to the Number of (Unreachable) Peers

The slots offered by reachable peers for connections are a limited resource. Now, we discuss a coarse-grained model for the allocation of this resource. If the Bitcoin P2P network consisted only of reachable peers that opened ten outgoing connections, the number of incoming and outgoing connections in the network would be equal and we could infer the total number of peers by dividing the number of all incoming connections by ten. However, in the real network there is also an unknown number of unreachable peers and not every peer creates exactly ten outgoing connections. For instance, there are peers (as our monitor nodes) that create outgoing connections to all reachable peers. We call these peers 'super peers'. Further, there are peers that open connections to many but not all of the reachable peers. We model these by assuming 'semi-super peers' that open connections to half of the reachable peers. Together with reachable and unreachable peers, this results in an allocation as depicted in Fig. 4.

In the following, we give a rough estimation of how this allocation might look like in the Bitcoin P2P network. We have estimated above that the number of reachable peers is about 7,518 peers. Assuming that each reachable (non-super) peer opens ten connections, we estimate that there are $7,518 \cdot 10$ outgoing connections of reachable peers which implies also $7,518 \cdot 10$ incoming connections from reachable peers. From our measurements at three reachable peers in October and November 2021, we assume that there are 18 super peers that are connected to all reachable peers in the network and about 26 semi-super peers. The super peers take up $18 \cdot 7,518$ connection slots in the network and the semi-super peers take up $26 \cdot 7,518/2$ connection slots. To estimate how many connections exist in the network in total,

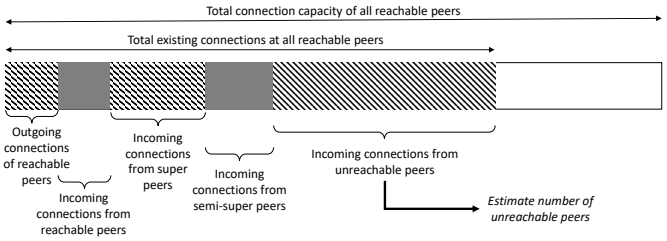

Fig. 4. Usage of connection slots of all reachable peers. We estimate the number of outgoing connections of reachable peers and the number of incoming connections from reachable peers, super peers, and semi-super peers. From the inferred number of incoming connections from unreachable peers, we estimate the number of unreachable peers.

we calculate the sum of all estimated peer degrees of peers that have an estimated degree not higher than 130. We use this cut-off of the default maximum of 125 and an error margin of 4 % and ignore peers with a higher degree because for peers with a higher degree we know that they are not using the default configuration and we do not know how many of their connections are outgoing or incoming connections. Using our estimated peer degree distribution, we obtain an estimate of $700,541$ filled connection slots. Subtracting the number of incoming connections that we ascribe to reachable peers, super peers, and semi-super peers, we receive a remaining number of $317,123$ connection slots that are probably filled by unreachable peers.

To estimate the number of unreachable peers from the number of connections of unreachable peers, we need to know the number of outgoing connections of unreachable peers. We determine the distribution of clients used by unreachable peers by calculating the distribution of user agents that are announced to our reachable peers. Based on their distribution and the default number of outgoing connections created by each client, we calculate that unreachable peers open on average 9.8 outgoing connections. This result leads to an estimated number of $32,300$ unreachable peers. This estimation lies in the broad range of previous estimates and, thus, reconfirms previous work [8]–[10].

VI. Conclusion

Based on the observation of a spam wave of addresses in July and August 2021, we have determined the peer degree distribution of the Bitcoin P2P network. As the openness of the P2P network depends on reachable peers accepting incoming connections, it is a notable result that about half of the reachable peers are close to their connection limit.

A similar spam wave is not possible anymore in the Bitcoin P2P network because, right before the spam wave started, a change that reduces the impact of such spam by rate-limiting ADDR propagation was implemented [32] and has been released with Bitcoin Core 22.0 [33] in September 2021. However, the revealed degree distribution and the improved estimate of the number of reachable peers can be helpful for refining models of the Bitcoin P2P network, enhancing Bitcoin's protocol, and measuring other P2P networks.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Tech. Rep., 2008.

[2] DSN. (2021) Bitcoin Network Monitoring. [Online]. Available: https://dsn.kastel.kit.edu/bitcoin/

[3] S. Delgado-Segura, S. Bakshi, C. Pérez-Solà, J. Litton, A. Pachulski, A. Miller, and B. Bhattacharjee, "TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, I. Goldberg and T. Moore, Eds. Cham: Springer International Publishing, 2019, pp. 550–566.

[4] T. Cao, J. Yu, J. Decouchant, X. Luo, and P. Verissimo, "Exploring the Monero Peer-to-Peer Network," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, J. Bonneau and N. Heninger, Eds. Cham: Springer International Publishing, 2020, pp. 578–594.

[5] T. Wang, C. Zhao, Q. Yang, S. Zhang, and S. C. Liew, "Ethna: Analyzing the Underlying Peer-to-Peer Network of Ethereum Blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2131–2146, Jul. 2021.

[6] A. Yeow. (2021) Bitnodes. [Online]. Available: https://bitnodes.io/

[7] Bitcoin-Developers. (2021) net.h. [Online]. Available: https://github.com/bitcoin/bitcoin/blob/v22.0/src/net.h#L72

[8] T. Neudecker, "Security and Anonymity Aspects of the Network Layer of Permissionless Blockchains," Ph.D. dissertation, 2019. [Online]. Available: https://publikationen.bibliothek.kit.edu/1000089033

[9] Luke-Jr. (2021) Bitcoin node count history. [Online]. Available: https://luke.dashjr.org/programs/bitcoin/files/charts/historical.html

[10] M. Grundmann, H. Amberg, M. Baumstark, and H. Hartenstein, "What Peer Announcements Tell Us About the Size of the Bitcoin P2P Network," in *Financial Cryptography and Data Security*, 2022.

[11] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee, "Discovering Bitcoin's Public Topology and Influential Nodes," 2015.

[12] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of Clients in Bitcoin P2P Network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 15–29.

[13] T. Neudecker, P. Andelfinger, and H. Hartenstein, "Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network," in *Proceedings of the 13th IEEE International Conference on Advanced and Trusted Computing*, Jul. 2016, pp. 358–367.

[14] M. Grundmann, T. Neudecker, and H. Hartenstein, "Exploiting Transaction Accumulation and Double Spends for Topology Inference in Bitcoin," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, Eds., vol. 10958. Springer Berlin Heidelberg, 2019, pp. 113–126.

[15] E. Daniel, E. Rohrer, and F. Tschorsch, "Map-Z: Exposing the Zcash Network in Times of Transition," *arXiv:1907.09755 [cs]*, Jul. 2019. [Online]. Available: http://arxiv.org/abs/1907.09755

[16] K. Li, Y. Tang, J. Chen, Y. Wang, and X. Liu, "TopoShot: Uncovering Ethereum's Network Topology Leveraging Replacement Transactions," in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 302–319.

[17] D. Ron and A. Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, A.-R. Sadeghi, Ed. Berlin, Heidelberg: Springer, 2013, pp. 6–24.

[18] F. Reid and M. Harrigan, "An Analysis of Anonymity in the Bitcoin System," in *Security and Privacy in Social Networks*, Y. Altshuler, Y. Elovici, A. B. Cremers, N. Aharony, and A. Pentland, Eds. New York, NY: Springer, 2013, pp. 197–223.

[19] M. Lischke and B. Fabian, "Analyzing the Bitcoin Network: The First Four Years," *Future Internet*, vol. 8, no. 1, p. 7, Mar. 2016.

[20] E. Filtz, A. Polleres, R. Karl, and B. Haslhofer, "Evolution of the Bitcoin Address Graph," in *Data Science – Analytics and Applications*, P. Haber, T. Lampoltshammer, and M. Mayr, Eds. Wiesbaden: Springer Fachmedien, 2017, pp. 77–82.

[21] D. Di Francesco Maesa, A. Marino, and L. Ricci, "Data-driven analysis of Bitcoin properties: exploiting the users graph," *International Journal of Data Science and Analytics*, vol. 6, no. 1, pp. 63–80, Aug. 2018.

[22] B. Tao, H.-N. Dai, J. Wu, I. W.-H. Ho, Z. Zheng, and C. F. Cheang, "Complex Network Analysis of the Bitcoin Transaction Network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2021.

[23] Various. (2021) BitcoinTalk Forum. [Online]. Available: https://bitcointalk.org/index.php?topic=5348856.msg57469495

[24] Bitcoin-Developers. (2021) netaddress.cpp. [Online]. Available: https://github.com/bitcoin/bitcoin/blob/54460704/src/netaddress.cpp#L490

[25] T. Cymru. (2021) IP to ASN Mapping Service. [Online]. Available: https://team-cymru.com/community-services/ip-asn-mapping/

[26] M. Ziv, L. Izhikevich, K. Ruth, K. Izhikevich, and Z. Durumeric, "ASdb: a system for classifying owners of autonomous systems," in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 703–719.

[27] J. A. Donet Donet, C. Pérez-Solà, and J. Herrera-Joancomartí, "The Bitcoin P2P Network," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, R. Böhme, M. Brenner, T. Moore, and M. Smith, Eds. Berlin, Heidelberg: Springer, 2014, pp. 87–102.

[28] M. Fadhil, G. Owenson, and M. Adda, "A Bitcoin Model for Evaluation of Clustering to Improve Propagation Delay in Bitcoin Network," in *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, Aug. 2016, pp. 468–475.

[29] V. Deshpande, H. Badis, and L. George, "BTCmap: Mapping Bitcoin Peer-to-Peer Network Topology," in *2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, Sep. 2018, pp. 1–6.

[30] G. Pappalardo, T. Di Matteo, G. Caldarelli, and T. Aste, "Blockchain inefficiency in the Bitcoin peers network," *EPJ Data Science*, vol. 7, no. 1, p. 30, Sep. 2018.

[31] S. Park, S. Im, Y. Seol, and J. Paek, "Nodes in the Bitcoin Network: Comparative Measurement Study and Survey," *IEEE Access*, vol. 7, pp. 57 009–57 022, 2019.

[32] P. Wuille. (2021) Rate limit the processing of rumoured addresses. [Online]. Available: https://github.com/bitcoin/bitcoin/pull/22387

[33] Bitcoin-Project. (2021) Version History. [Online]. Available: https://bitcoin.org/en/version-history