Short Paper: What Peer Announcements Tell Us About the Size of the Bitcoin P2P Network

Matthias Grundmann, Hedwig Amberg, Max Baumstark, and Hannes Hartenstein

Institute of Information Security and Dependability (KASTEL),
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{matthias.grundmann,hannes.hartenstein}@kit.edu

Abstract. Bitcoin is based on a P2P network of which only a few quantities are publicly known. While the number of peers that disseminate transactions and blocks is relevant for the robustness of the network, only the number of reachable peers is so far being measured. However, there exists an unknown number of unreachable peers in the network, that is, peers that do not accept incoming connections but typically also disseminate transactions and blocks. We propose the Passive Announcement Listening (PAL) method that gives an estimate of the number of unreachable peers by observing peer announcements in ADDR messages. We use the PAL method to analyze data from a long-term measurement of the Bitcoin P2P network from 2015 to 2021. The PAL estimate shows that since 2018 the number of unreachable peers is at least three times higher than the number of reachable peers. A first empirical validation indicates that the approach finds about 78 % of the unreachable peers that disseminate transactions and blocks and, thus, we estimate their current number to be in the range of 27,000 to 35,000. We also report on a spam wave of ADDR messages that shows that peer announcements 'leak' even more information than the size of the network.

1 Introduction

Bitcoin [16] is based on a peer-to-peer (P2P) network that is used to disseminate transactions and blocks of the blockchain. For reasons of robustness, the P2P network should disseminate blocks quickly and transactions efficiently. As the number of peers in the network influences the dissemination of transactions and blocks, the number of peers needs to be known to understand the P2P network and to build realistic models used for the development and evaluation of protocol mechanisms. By design, the Bitcoin protocol does not implement a method to collect such quantities about the P2P network and, thus, these quantities can only be estimated or inferred from observations. In 2014 and 2015, some methods to infer the topology of the P2P network based on Bitcoin Core's handling of peer announcements were discussed [1,12,15]. However, these methods showed a high complexity or were impeded by subsequent updates in the implementation of Bitcoin Core. In this paper, we present a novel approach

based on observations of peer announcements to estimate the number of peers that disseminate transactions and blocks.

To form the P2P network, each peer creates outgoing connections to other peers. Not every peer, however, is able or willing to accept incoming connections, either because a peer is behind a NAT or a firewall or because of a deliberate policy choice. Thus, peers can be categorized into reachable peers that accept incoming connections and unreachable peers that do not accept incoming connections [4]. Categorizing peers into reachable and unreachable peers is not trivial because reachability depends on the vantage point. A first approach might be to define a peer as unreachable if all other peers cannot initiate a connection to that peer. However, a peer might accept incoming connections from one group of peers but refuse incoming connections from other peers. Thus, one could use the following definition: A peer p is called unreachable if the majority of other peers cannot initiate a connection to p. While this definition clarifies the set of unreachable peers, one cannot practically measure it. Thus, we will follow a 'relativistic' approach for our measurements by categorizing peers based on our given vantage point. In [8], we further discuss the challenges of defining unreachability.

The number of unreachable peers that disseminate transactions and blocks (*disseminating peers*) is relevant for the robustness of the P2P network on the one hand because unreachable disseminating peers support dissemination just as reachable peers but are harder to attack precisely because they are unreachable, and on the other hand because anomalies in the number of unreachable peers can indicate attacks on the P2P network. Some projects [22,6] continuously measure the number of reachable peers. However, unreachable peers are harder to detect because one cannot connect to them. One way to get an estimate of the number of unreachable peers is to observe a fraction of unreachable peers and extrapolate the whole number of unreachable peers, e.g., by running a reachable peer that accepts connections from unreachable peers (see [21]). Another way is to observe effects that are caused by unreachable peers and infer their number from these observations.

In this paper, we follow the latter approach of 'observing effects' and present the Passive Announcement Listening (PAL) method to estimate the number of unreachable peers. This approach relies on observing peer announcements that are propagated by peers in the network. The PAL method uses a passive monitor node that connects to all reachable peers and waits for unsolicited ADDR messages. The rationale behind the PAL method is that if the monitor receives an address in an unsolicited ADDR message, one can conclude – based on how Bitcoin Core propagates peer announcements – that less than ten minutes ago there was a peer at this address. Because peers regularly announce their address, collecting all unsolicitedly sent addresses during one day gives an estimate of the set of peers having existed during this day. By filtering out reachable peers, we obtain an estimate of the set of unreachable peers.

Previous work has estimated the number of unreachable peers to be around 16,000 peers [18], 54,000 peers [17], 90,000 peers [1], and 155,000 peers [21]. The wide range of estimations comes not only from different measuring times

and methods but also from the fact that the number of unreachable peers at a certain point in time differs from the number of unreachable peers measured over a time interval. In this work, we consider the problem of estimating the number of unreachable peers during time intervals. Using a model for churn (see [11]), this number can be used to estimate how many unreachable peers existed at a given point in time.

We will give an overview of related work in Section 2. In Section 3, background on the peer behavior of the most common Bitcoin implementation is provided. Then, in Section 4, we present the PAL method and the results of applying the method to data collected from the Bitcoin P2P network. As there is no ground truth available, we validate our approach in Section 5 by verifying our assumptions and by comparing the results of our approach to an observation of a fraction of unreachable peers. In Section 6 we describe how a recent spam wave of ADDR messages helped to estimate the number of neighbors of reachable peers and to find peers with multiple addresses. We conclude in Section 7.

2 Related Work

The number of *reachable* peers has been analyzed by previous research [5,19] and is continuously measured by different projects [22,6]. These projects share the basic approach of recursively searching the network for reachable peers. As an example, we explain the approach of Bitnodes [22] which is similar to that of Donet et al. [5] and Park et al. [19]: The software starts with an initial set of peers, connects to each peer and requests addresses from each peer using a GETADDR message. This message is replied to by an ADDR message that contains up to 1,000 entries from the sending peer's database of which some addresses might be outdated and not belong to a peer anymore. On receiving the ADDR message as a reply, the software tries to connect to each of the addresses in the reply and, for each successfully opened connection, addresses are requested over this new connection. The set of peers that a connection has been established to is regarded as the set of reachable peers. In case a connection to an address cannot be established, it is unknown whether there is an unreachable peer at this address or the address is outdated and there is no peer at this address. Consequently, this approach is not capable of measuring the number of unreachable peers.

Only few attempts have been made to estimate the number of *unreachable* peers. In May 2017, Wang and Pustogarov [21] ran 102 reachable peers as probes for seven days and logged all incoming connections and associated information. For each peer that connected to one of the probes, they tested whether it was reachable by trying to open a connection to that peer's address. They observed on average about 10,000 unique unreachable addresses in a six-hour interval and estimate without a detailed explanation that there were at least 155,000 unreachable peers in each six-hour interval. Bitcoin developer Luke-Jr runs a website [14] that lists about 54,000 unreachable peers and 6,000 reachable peers at the time of writing (December 2021). The methodology behind the website is not publicly documented, but, in the absence of other reference points, we also

compare our measurements to the numbers obtained from this website. The role of unreachable peers in the Bitcoin P2P network has only been studied to a very limited degree. Franzoni and Daza [7] recently showed how the robustness and efficiency of the P2P network can be improved by giving unreachable peers a special role in the dissemination of transactions.

3 Background on Bitcoin Peers

We refer to an implementation of a client for the Bitcoin protocol as Bitcoin software. We define a *peer* as a running instance of a Bitcoin software that is connected to at least one other running instance of a Bitcoin software. We expect most peers to be connected to multiple peers in order to reduce chances of being eclipsed [10]. A Bitcoin P2P network consists of peers that are directly or indirectly connected to each other. In this paper, we consider only peers in the Bitcoin P2P network that is referred to as the "Bitcoin mainnet" [3].

Peers are identified by their addresses. A peer can have multiple addresses (in the most common case an IPv4 address and an IPv6 address) and multiple peers can share an address (e.g., an IPv4 address because they are behind the same NAT). We will make the simplifying assumption that each peer has exactly one address. If we simply use the term address, then it refers to any type of address being used in the Bitcoin protocol, e.g., IPv4, IPv6, or Tor address (see [13]).

In the following, we describe the protocol for peers in the Bitcoin P2P network [2] and the behavior of Bitcoin Core, the software that is run by the majority of peers [22]. Peers need to know the addresses of other peers to be able to connect to them. To this end, addresses are exchanged between peers using ADDR messages that contain between one and 1,000 entries. Each entry consists of an address, a port, a timestamp, and service flags. The service flags describe the services offered and extensions implemented by the peer running at the address. A peer unsolicitedly sends a *self announcement* of its address to a connected peer once a connection has been established and then on average every 24 hours. The self announcement contains the announcing peer's service flags and the timestamp of the self announcement is set to the time of sending. If the announced address is routable, i.e. not from an IP address range that is reserved for private use, and the service flags contained in the self announcement include certain required flags (the NODE_WITNESS flag and the NODE_NETWORK or NODE_NETWORK_LIMITED flag), then the address is propagated in the network together with the associated timestamp and service flags until the timestamp is older than ten minutes. In Bitcoin Core, the sending of ADDR messages per connection is limited to two messages per minute (on average), and addresses received in multiple incoming ADDR messages might be batched in one outgoing ADDR message. If an incoming ADDR message contains ten or fewer entries, Bitcoin Core considers the addresses in the ADDR message as a batch of self announcements originally sent unsolicitedly and, therefore, for propagation. In the remainder of this paper, we only consider such unsolicited ADDR messages that contain up to ten entries.

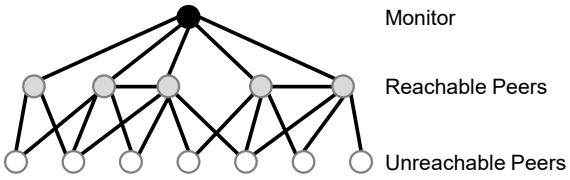

Fig. 1. Setup overview. The monitor node that collects the data for the PAL method is connected to all reachable peers but not to unreachable peers.

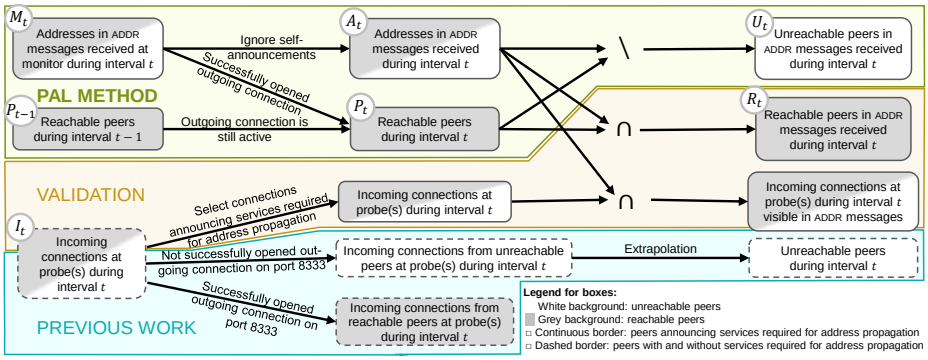

Fig. 2. Data flow of the PAL method, validation, and previous work [21]. The sets M_t and I_t are collected during measurements and the arrows show filters and operations to derive more specific sets during the analysis. The border of each box indicates whether the respective set contains only peers that set the flags required for address propagation or also those peers that do not set these flags. The background colors indicate whether the respective sets contain reachable and/or unreachable peers.

4 PAL Method and Results

In this section, we present the PAL method's setup for data collection, the methodology for analyzing the data and the resulting findings.

Data Collection. The monitor node connects to all known reachable peers in the network (see Fig. 1) and does not send any ADDR messages. The only messages the monitor sends are VERSION messages during connection establishment and GETADDR messages. The solicited ADDR messages that are received in reply to GETADDR messages are ignored for the PAL method but are used to learn about reachable peers. The monitor tries to connect to each received address (rate-limited per address to once every six hours). The monitor logs all received ADDR messages, VERSION messages and the time when a connection to another peer is established or closed.

Data Analysis. We analyze the logs created by the monitor to learn the number of peers in the network. This process is depicted in the upper part of Fig. 2.

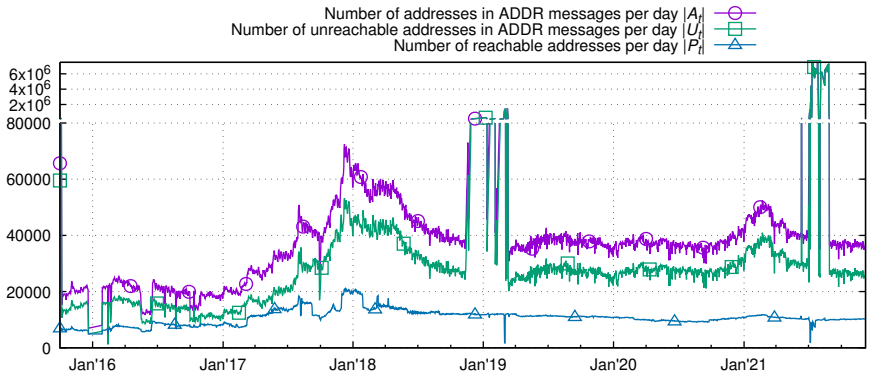

Fig. 3. Number of addresses observed in ADDR messages compared to reachable addresses per day. Note that the upper part uses a different scale than the lower part.

For each day t, we collect all unsolicited addresses that were received by the monitor (M_t in Fig. 2). We define the set A_t by ignoring the self announcements of (reachable) peers, i.e. entries of an ADDR message that equal the address of the sender of this ADDR message. The set A_t includes addresses of reachable and unreachable peers that were announced on day t. To determine the set P_t of all addresses that the monitor node was connected to on day t, we collect all addresses that the monitor already was connected to at the beginning of day t or a connection was established and a VERSION message received during day t. We consider this set P_t as the set of all reachable peers at day t. Our estimate of the set of unreachable peers U_t for day t is $U_t = A_t \setminus P_t$.

Limitations. The PAL method cannot distinguish whether an unreachable peer existed only for a short moment on a day or the whole day. Also, the addresses and associated information in ADDR messages are not authenticated. Therefore, the approach can be disturbed by flooding the network with bogus addresses.

Measurements. We applied the method to data collected from 2015 to 2021 by a monitor node hosted in the network of KIT (AS 34878). Figure 3 shows $|A_t|$, the number of addresses received in ADDR messages for each day t and the number $|U_t|$ of addresses that were unreachable. In each set, an address is counted only once if it is received multiple times during t. On the majority of days in the observation range, between 20,000 and 60,000 addresses were received in ADDR messages. Most noticeably, the plot shows a high number of addresses at the end of 2018 and recently in July 2021 which we will discuss later. The remaining plot shows that the number of addresses varied over the years and had local maxima in December 2017 (72,000 addresses) and in February 2021 (51,000 addresses). The number of unreachable peers $|U_t|$ is on average about 73 % of the number of all addresses $|A_t|$. In December 2021, the number of unreachable peers $|U_t|$ equals about 27,000 peers. A comparison with the number of reachable peers $|P_t|$

shows that since 2018 the number of unreachable peers in ADDR messages was about three times the number of reachable peers and had a similar development.

The peak at the end of the year 2018 seems like many unreachable peers joined the network within a few days. An alternative explanation would be that bogus addresses were distributed that do not actually belong to peers. We examined the addresses that were received only during this time and did not find any irregularities with regard to their distribution in the IP address space, autonomous system, or country of autonomous system. However, for the highest peak in March 2019, we found that this peak was caused by many IP addresses from the same /8 subnet. As IP addresses from this subnet were only very rarely observed before and after March 2019, we assume that this effect was caused by unknown actions of one party that flooded the network with these IP addresses. Examples of such actions might be the explanations we find for the recent peak in July and August 2021 that we discuss in Section 6.

5 Validation

Reachable Peers. Validating the PAL method is difficult because we do not have a reliable ground-truth to compare our results to. However, while the goal of the PAL method is to find unreachable peers, it can also be used to find reachable peers. As we know the set of reachable peers quite accurately, we can validate whether reachable peers can be found in ADDR messages during each day. Putting this into the context of Fig. 2, this means that, if the PAL method works perfectly, we expect that set R_t equals set P_t. We evaluate this for the data collected during the year 2020 and find that on average 95.4 % of the addresses of reachable peers on a day were received in an ADDR message on the same day (excluding self-announcements). Increasing the length of the interval t from one day to five days increases the share of observed reachable peers to 96.1 % while with an interval length of one hour only on average 84.9 % of the addresses of reachable peers were received in an ADDR message in the same hour. This indicates that reachable peers are consistently found by the PAL method and that the interval length of one day is a reasonable trade-off.

Unreachable Peer. To validate our assumption that an unreachable peer is being found by the PAL method, we permanently ran an unreachable peer from December 2020 to June 2021. The monitor received the unreachable peer's address on 200 of 212 days which means that on each day the probability for the peer to be detected was 94 %.

Second Monitor. For further validation with another vantage point, we have run a second monitor node since 2019. The second monitor node is set up as described above for the first monitor node but runs in a different location and a different autonomous system. If the measurement method is reproducible, the addresses received by the two monitor nodes should largely overlap. We analyzed the addresses received by both monitors since 2019 and find that 96 % of the

addresses overlap. This indicates that the measurement is reproducible and that the view of our monitor node is not subjective to the specific instance of the monitor.

Validation with Incoming Connections. The approach of Wang and Pustogarov [21] is to run many reachable peers and wait for unreachable peers to connect to them. This approach can only find a fraction of unreachable peers and it is unclear how to reliably extrapolate from this fraction to the whole network. However, the approach can collect reliable information about the observed fraction of unreachable peers because they are directly connected. For further validation, we use a similar approach and run two additional peers p_{I} and p_{R} that accept incoming connections. We treat the fraction of unreachable peers seen by p_{I} and p_{R} as a ground-truth and calculate the precision and recall of the PAL method against this ground-truth. We find that the PAL method detects unreachable disseminating peers with a precision of 83 % and a recall of 78 %. This means that 83 % of peers that are detected by the PAL method and have connected to p_{I} or p_{R} actually disseminated transactions and blocks to p_{I} or p_{R}. False positives that decrease the precision are, for example, addresses of Tor exit nodes that are shared by multiple peers of which some announce the exit node's address associated with the services required for address propagation while others use the same exit node but do not disseminate transactions and blocks. A recall of 78 % means that 78 % of peers that connected to p_{I} or p_{R} and disseminated transactions and blocks were detected by the PAL method. Explanations for unreachable peers being undetected by the PAL method could be that unreachable peers do not exist long enough in the network to be detected by the PAL method, that peers run software with a different behavior, or that they announce an address that is different from the address they use for connections.

Comparison to Previous Measurements. There is no ground truth that we could compare the PAL method's results to but we can compare it to previous estimations and measurements. Neudecker et al. [18] simulated the Bitcoin P2P network in 2016 and estimated from the simulated propagation behavior that the P2P network had about 16,000 unreachable disseminating peers. The PAL method calculates about 14,000 unreachable peers per day averaged over the year 2016. As the results of Neudecker et al. are for one point in time and the PAL method estimates the number of unreachable peers during one day, we would rather expect that the PAL method would find *more* unreachable peers. The lower number of unreachable peers detected by the PAL method might be caused by peers not announcing services required for address propagation.

A measurement of unreachable peers was conducted by Wang and Pustogarov in 2017 [21] (see Section 2). Based on their observation of a fraction of unreachable peers, they estimated at least 155,000 unreachable peers to be active in each six-hour interval. They report that 93.9 % of all connections lasted shorter than one minute and 80 % of unreachable peers were mobile peers. We assume that these peers either did not announce their IP addresses or that they did not provide services required for address propagation. In this case, they would

be invisible to the PAL method which explains why the estimate by Wang and Pustogarov is higher than the results obtained through the PAL method.

The measurement by Luke-Jr [14] gives an estimate of the number of reachable and unreachable peers over a time span similar to our measurements. The number of unreachable peers in the data from Luke-Jr is higher compared to the estimation using the PAL method. This is probably accounted for again by the fact that not all addresses of unreachable peers are propagated.

6 Observation of ADDR Spam in July and August 2021

The number of unique addresses in ADDR messages increased significantly in July 2021 (see Fig. 3) from about 40,000 unique addresses per day to about 6,000,000 unique addresses per day. This increase was caused by an unknown party sending many spam addresses into the Bitcoin P2P network. Observations of the propagation of the spam addresses show that more than the number of unreachable peers can be learned from observing peer announcements (see [9]): We analyzed the behavior of the spamming peers and found that our observations of the propagated spam addresses could be used to estimate the node degree (number of neighbors) of reachable peers based on an idea that dates back to 2014 [1, Section 10.1]. Further, we found that the observed propagation of spam addresses could be used to map multiple addresses to the same reachable peers when the same spam addresses are forwarded to our monitors from different IP addresses. In August 2021, our monitor nodes were connected to 8,800 reachable addresses per day on average. From the obtained mapping from addresses to actual peers, we infer that the monitor nodes were connected to only 7,650 peers per day on average. This shows that estimating the number of reachable peers by counting reachable addresses overestimates their number by 13 %.

7 Conclusion

Unreachable peers contribute to the Bitcoin P2P network by disseminating blocks and transactions, but are inherently hard to detect and count. We have presented the PAL method that analyzes peer announcements to estimate the number of unreachable peers. Our observed number of unreachable peers in December 2021 is about 27,000 peers which, as indicated by our validation, might correspond to about 78 % of all disseminating peers. We estimate by extrapolation that there could actually be 27,000 to 35,000 unreachable peers which corresponds to three to four times the number of reachable peers. In contrast to the costly approach of running many reachable peers to find unreachable peers, the PAL method is deployable as a continuously running project. We will continue to monitor and publish the updated data and results [6].

Acknowledgments

The authors would like to thank the anonymous reviewers and Till Neudecker for their feedback. The authors acknowledge support by the State of Baden-Württemberg through bwHPC. This work was supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

1. Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of Clients in Bitcoin P2P Network. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 15–29. CCS '14, Association for Computing Machinery, New York, NY, USA (Nov 2014). https://doi.org/10.1145/2660267.2660379
2. Bitcoin-Developers: Bitcoin Reference (2019), https://developer.bitcoin.org/reference/index.html
3. Bitcoin-Developers: Bitcoin Glossary (2020), https://developer.bitcoin.org/glossary.html
4. Delgado-Segura, S., Pérez-Solà, C., Herrera-Joancomartí, J., Navarro-Arribas, G., Borrell, J.: Cryptocurrency Networks: A New P2P Paradigm. Mobile Information Systems **2018** (Mar 2018). https://doi.org/10.1155/2018/2159082
5. Donet Donet, J.A., Pérez-Solà, C., Herrera-Joancomartí, J.: The Bitcoin P2P Network. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) Financial Cryptography and Data Security. pp. 87–102. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44774-1_7
6. DSN: Bitcoin Network Monitoring (2021), https://dsn.kastel.kit.edu/bitcoin/
7. Franzoni, F., Daza, V.: Improving Bitcoin Transaction Propagation by Leveraging Unreachable Nodes. In: 2020 IEEE International Conference on Blockchain (Blockchain). pp. 196–203 (Nov 2020). https://doi.org/10.1109/Blockchain50366.2020.00031
8. Grundmann, M., Amberg, H., Hartenstein, H.: On the Estimation of the Number of Unreachable Peers in the Bitcoin P2P Network by Observation of Peer Announcements. arXiv:2102.12774 [cs] (Feb 2021), http://arxiv.org/abs/2102.12774
9. Grundmann, M., Baumstark, M.: Estimating the Node Degree of Public Peers and Detecting Sybil Peers Based on Address Messages in the Bitcoin P2P Network. arXiv:2108.00815 [cs] (Aug 2021), http://arxiv.org/abs/2108.00815
10. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on Bitcoin's peer-to-peer network. In: Proceedings of the 24th USENIX Conference on Security Symposium. pp. 129–144. SEC'15, USENIX Association, USA (Aug 2015)
11. Imtiaz, M.A., Starobinski, D., Trachtenberg, A., Younis, N.: Churn in the Bitcoin Network. IEEE Transactions on Network and Service Management (2021). https://doi.org/10.1109/TNSM.2021.3050428
12. Jonas Nick: Guessing Bitcoin's P2P Connections (2015), https://jonasnick.github.io/blog/2015/03/06/guessing-bitcoins-p2p-connections/
13. van der Laan, W.J.: Bip 155: addrv2 message (2019), https://github.com/bitcoin/bips/blob/master/bip-0155.mediawiki
14. Luke-Jr: Bitcoin Node Count History (2021), https://luke.dashjr.org/programs/bitcoin/files/charts/historical.html

15. Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., Bhattacharjee, B.: Discovering Bitcoin's Public Topology and Influential Nodes (2015)
16. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. Tech. rep. (2008)
17. Naumenko, G., Maxwell, G., Wuille, P., Fedorova, A., Beschastnikh, I.: Erlay: Efficient Transaction Relay for Bitcoin. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security - CCS '19. pp. 817–831. ACM Press, London, United Kingdom (2019). https://doi.org/10.1145/3319535.3354237
18. Neudecker, T., Andelfinger, P., Hartenstein, H.: 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. pp. 358–367 (Jul 2016). https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0070
19. Park, S., Im, S., Seol, Y., Paek, J.: Nodes in the Bitcoin Network: Comparative Measurement Study and Survey. IEEE Access **7**, 57009–57022 (2019). https://doi.org/10.1109/ACCESS.2019.2914098
20. Tange, O.: GNU Parallel 20200522 ('Kraftwerk') (May 2020), https://doi.org/10.5281/zenodo.3841377
21. Wang, L., Pustogarov, I.: Towards Better Understanding of Bitcoin Unreachable Peers. arXiv:1709.06837 [cs] (Sep 2017), http://arxiv.org/abs/1709.06837
22. Yeow, A.: Bitnodes (2021), https://bitnodes.io