Characterising Proxy Usage in the Bitcoin Peer-to-Peer Network

Alexander Mühle
alexander.muehle@hpi.de
Hasso Plattner Institute
Potsdam, Germany

Andreas Grüner
andreas.gruener@hpi.de
Hasso Plattner Institute
Potsdam, Germany

Christoph Meinel
christoph.meinel@hpi.de
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

In the public mind, Bitcoin has often been associated with censorship circumvention and evasion of surveillance measures, specifically in the context of monetary transactions. However, this perceived anonymity is a false sense of security as both on-chain transactions and the underlying message exchange in the peer-to-peer network are attack vectors for deanonymisation and monitoring, as shown in other research. Nonetheless, there has been an increase in Bitcoin usage not only for end-users but also in the context of cybercrime in the form of cryptojacking and ransomware. So there are a number of reasons why proxies might be used in the Bitcoin network, either as a privacy-preserving measure of end-users or as obfuscation in cybercrime.

In this paper, we present a measurement study with the goal of characterising the proxy and VPN usage in the Bitcoin peer-to-peer network. We developed YABA (Yet Another Bitcoin Analyser) to gather network data in a geographically distributed fashion and analyse it. We describe our techniques to infer proxy/VPN usage and load on the peer through different latency measurements and the limitations of our approaches. We utilise port scanning of standard proxy/VPN service ports to compare results. We deployed our infrastructure on three continents (4 workers) and continuously crawled the network, with a total of 26.9 million connection attempts over five days. We conclude the usage of proxies to be minimal, with an estimated 0.4% of peers detected through latency measurements. Similar prevalence was measured through the use of port scans with SOCKS port hitrate at 0.3%, while common VPN ports had hitrates between 0.18% and 0.7%.

ICDCN '20, January 5–8, 2021, Nara, Japan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8933-4…$TBA
https://doi.org/TBA

CCS CONCEPTS

• **Networks** → *Network privacy and anonymity*; **Network measurement**; **Peer-to-peer networks**.

KEYWORDS

Bitcoin, Network Measurement, Proxy Detection

ACM Reference Format:
Alexander Mühle, Andreas Grüner, and Christoph Meinel. 2020. Characterising Proxy Usage in the Bitcoin Peer-to-Peer Network. In *Proceedings of ICDCN '20.* ACM, New York, NY, USA, 10 pages. https://doi.org/TBA

1 INTRODUCTION

Bitcoin [21] as introduced in 2008, has gained broad public attention with its promise of secure peer-to-peer payments without the need for a trusted middleman. Its goal was to be an equivalent of the classic cash transaction, but for the internet age.

Bitcoin seems to be a popular choice for users with motivations of circumventing censorship and content filtering. There have been some widely publicised examples of Bitcoin serving exactly this purpose. Traditional financial institutions such as Visa and PayPal participated in a blackout of Wikileaks in 2012 [23], not processing donations to the site after pressure by various governments. Consequently, Wikileaks started accepting donations in Bitcoin. However, it has also seen application in more nefarious settings such as cryptomining using botnets or drive by in-browser mining [3]. Due to the lack of control over Bitcoin some countries have outright banned the currency or restricted banks from interacting with it [4]. Certainly, it has to be expected that major law enforcement and government agencies are monitoring the Bitcoin network.

Using compromised servers as a proxy is a common practice of cybercriminals to protect themselves against prosecution.

At the same time, Virtual Private Networks (VPN) saw increased popularity in recent years. They have largely been marketed as privacy-preserving technology, giving end-users a simple way to obscure their traffic, in order to circumvent censorship and content filtering by Internet Service Providers (ISPs) as well as governments [17].

1.1 Motivation

There have been numerous studies around the usage of the Bitcoin network. In order to analyse motivation and use-cases researchers have relied on user-surveys as well as analysis of open data such as twitter feeds, forum entries and google search data [9], [18], [2], [13], [24],[16].

These surveys and analyses have confirmed our intuition that Bitcoin's potential for subverting centralised governmental and financial institutions is certainly a reason some users choose to participate in the Bitcoin network [16].

According to Krombholz et al., 18% of users self-reported that they utilise some tools to stay anonymous, amongst them VPNs [18]. VPN usage has, in general, often been motivated by privacy and security concerns for "peace of mind" [9].

Cryptojacking as well has been analysed from a user perspective, tracking usage of cryptocurrency mining scripts in popular web sites [14]. However, the aspect of obfuscation of the usage of Bitcoins by the cybercriminal after a successful attack has not been discussed to our knowledge.

Quantifying the usage of proxies and VPNs in the context of Bitcoin through technical measurements is the gap we recognised and want to tackle in this work. It has to be noted that we analyse users in the network sense (participants in the peer-to-peer network) which can be distinct from the users of the Bitcoin currency itself.

1.2 Contribution

In this work, we will conduct real-life measurements of the Bitcoin network to recognise and quantify proxy and VPN usage. For this purpose, we crawl the network as described in Section 3, gathering information on the participants. This includes data disclosed during handshake procedures, neighbour discovery, as well as latency measurements. There have been a number of projects exploring the Bitcoin network through crawling of participants; however, we have chosen to implement a new architecture with scalable distributed workers as well as storage and analysis of the gathered data using big data tools such as Elasticsearch and Apache Spark.

In Section 4, we describe our methodology for proxy/VPN detection in the Bitcoin network, i.e. how round trip times measured to the Bitcoin peer and the times measured to the visible IP differ. For this, we first explore two different methods for measurement of latency to the visible IP. Additionally, we describe our approach to quantifying the processing time of the nodes by comparing the delay measured inside the Bitcoin protocol and measurements through the Bitcoin TCP port.

Using these results, we show the prevalence and distribution of proxy and VPN users in the network during our measurement period. Port scanning on a number of common proxy/VPN provider ports is used to further argue about the feasibility of the latency based approach as well as to get a sense of the scale of proxy/VPN usage of unreachable servers. We also discuss some further insights we can gather about the characteristics of proxy users in the Bitcoin network, through analysis and correlation of our gathered data with open-source intelligence (OSINT) such as geolocation and autonomous system, reverse domain lookup and the estimated processing time.

In conclusion, we see our main contributions as follows:

- Gathering Bitcoin network data at scale, discovering 204,739 unique nodes in 5 days
- Measuring prevalence of proxy usage in the Bitcoin network through latency characteristics, estimating overall proxy usage at 0.4%
- Measuring processing time characteristics in the Bitcoin network, amounting to 77.5ms as median
- Port scanning in the Bitcoin network, discovering up to 0.7% nodes with common proxy/VPN ports
- Discussion on characteristics of suspected proxy and VPN users

2 RELATED WORK

2.1 Bitcoin

The Bitcoin ecosystem has been analysed from a number of different perspectives. The user experience, adoption, security but also aspects such as anonymity in the peer-to-peer network, the topology of the network and general makeup of it. We present some relevant work of Bitcoin network exploration.

Bitcoin itself has often been analysed from a peer-to-peer network perspective. For this purpose, researchers either modified the reference Bitcoin implementation [1] for extended logging capabilities or wrote specific Bitcoin clients for this purpose.

Donet et al. used a single Bitcoin Sniffer instance to monitor the network for 37 days, focusing on analysing the number and geographic distribution of participants as well as the network stability [7]. Feld and Werner [11] similarly investigated the resilience of the Bitcoin ecosystem, especially by looking at the distribution of peers among autonomous systems.

An essential part of any gossip-based peer-to-peer system is the propagation of new information in the network. This is another topic that a number of projects have used network crawlers and sniffers for. Decker et al. measured the performance of the network for block and transaction propagation by observing multiple participants at the same time and listening to inventory messages [5]. Utilising a now

[1] github.com/bitcoin/

fixed exploit in the address discovery protocol of Bitcoin Miller et al. inferred topology and influential nodes [20]. Using new exploits in the context of orphan transactions and double-spend transactions, respectively, Delgado [6] and Grundmann [12] mapped topologies of the Bitcoin network more recently.

2.2 Proxies

There are a number of commercial offerings targeted at fraud prevention, that provide proxy detection databases. The methodology behind these databases is often times not transparent but likely combines a number of the following.

One more transparent example is whatismyipaddress [2]. They utilise reverse domain lookup to find indicators of datacenter addresses as they are more likely to be proxies specifically when encountered by website providers. Other approaches include user-agent analysis and misconfiguration of the proxy in the HTTP header. Additionally checking against the public Tor node list is an easy way for service providers to exclude some proxy users.

Stepping stone detection has been explored with latency based methods. Zhang and Paxson correlate connections by their latency timings [25]. In order to evade such latency based methods, inserting local jitter and chaff (superfluous packets) is a possible approach. Donoho et al. present stepping stone detection in the face of such evasion tactics by exploiting maximum tolerable delays by a stepping stone user [8]. However, this is for interactive streams, unlike the Bitcoin peer-to-peer network, which is not directly interactive.

2.3 Virtual Private Networks

The landscape of commercial VPN providers, showing the geographic distribution of the businesses and servers, payment methods, technology used as well as privacy concerns have been investigated by Khan et al. [17]. In their analysis of the commercial VPN ecosystem, they however, did not include the users themselves.

In contrast to the focus on providers by Khan et al., Dupuis et al. focused on the usage of VPNs by users as a cybersecurity tool. It showed that VPNs have a rather modest usage rate of only around 18% according to the 2019 survey. In the same study, participants said, they use VPNs mostly because they believe they are effective and offer peace of mind [9]. Yet risky behaviour has only been named as the primary reason for VPN usage by 4.6%. Similarly to user studies in the context of illegal behaviour or anonymity in Bitcoin, the answers for risky behaviour might, however, be influenced by users self-censoring.

They further analysed the usage of circumvention tools in the context of governmental censorship/surveillance, including not only VPNs but also blocking resistant services (i.e. Tor) and simple proxy services (web-page interfaces). To analyse the popularity of VPN services, they relied on online search trends and surveys.

3 DATA COLLECTION

In this Section we will describe our requirements and subsequent implementation of our measurement infrastructure.

3.1 Requirements

The central requirement for our crawler is to run it geographically distributed and to work cooperatively on a single snapshot.

Running sniffers geographically distributed and potentially across different ISPs gives us the possibility to measure latency from different vantage points, averaging out network anomalies and having higher confidence in our results. Additionally, as other projects have noted due to the geographic location of sniffers, the data collection can be biased [3]. This can be due to the initial resolution of seed nodes from DNS servers that perform geographic load balancing and other network factors. There are a number of studies showing the Bitcoin network graph is of non-random structure [10] [19] further biasing results. Through geographic distribution, we can reduce the bias introduced by their location and entry in the network.

One of the limitations with any kind of peer-to-peer network crawler is the inherently progressive nature of network crawls [22]. But in the Bitcoin network, there is also the problem that peer addresses are returned randomly. This makes it harder for us to discover the complete list of known peers of other nodes in a timely fashion and therefore, the completeness of our view on the network. In the reference implementation of Bitcoin [4], a maximum number of 20480 peers is in a node's peer list. Each request returns 23% of a nodes peer list or at most 2500 addresses. Therefore we get the probability of not receiving a certain address expressed as $1 - (1/\frac{20480}{2500})^x$ with x being the number of requests sent.

To combat this, we want to run multiple sniffers in parallel to increase the number of requests sent at a time working on a single snapshot.

3.2 Yet Another Bitcoin Analyser

In order to fulfil the previously described requirements, we developed YABA (Yet Another Bitcoin Analyser). The main motivation behind the development was to provide an up to date open source Bitcoin Sniffer that could be run at scale

[2] whatismyipaddress.com/proxy-check

[3] https://dsn.tm.kit.edu/bitcoin/

[4] github.com/bitcoin

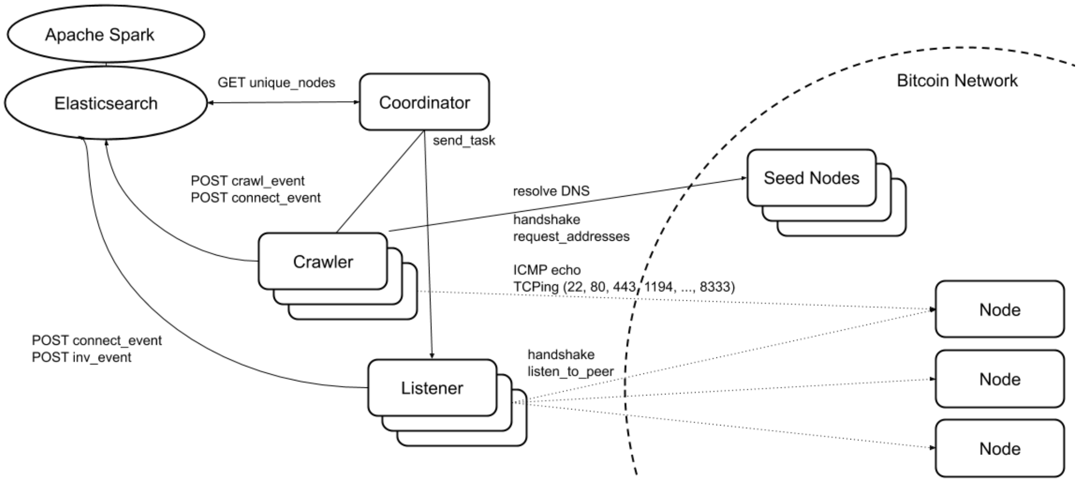

Figure 1: YABA Architecture Overview

with multiple geographically distributed workers. Most previous efforts were single instance systems [5].

Figure 1 shows the architecture of YABA.

YABA consists of a coordinator instance as well as a variable number of crawlers. The coordinator keeps track of a constantly updated unique node list and distributes tasks for crawlers. Crawlers then participate in the neighbour discovery protocol. Initially, each crawler resolves DNS seed nodes to enter the Bitcoin network. This is done at each individual worker instead of centrally at the coordinator. Crawlers then post two different events (*crawl event* and *connect event*) to the centrally located database (Elasticsearch).

Connect events include information gathered through the handshake procedure of the Bitcoin protocol such as user agent, most recent block height known and protocol version. During the handshake, we also measure a number of additional latency information. We utilise the keep-alive feature of the Bitcoin protocol to measure application latency during the ping/pong exchange. We measure ICMP ping to the visible IP and TCPing on some common ports such as ssh, http, https, openVPN and SOCK. During the establishment of the TCP connection used for Bitcoin we also measure latency to the Bitcoin port.

Finally, we record some meta-information such as which worker initiated the connection, at what time and what adress.

Crawl events represent the results of a get address request. Each crawl event includes the discovered neighbour node (ip, port) and which service the node provides in the context of the Bitcoin protocol.

The *crawl events* posted to the Elasticsearch are used by the central coordinator to keep a unique node list and distribute tasks to the individual workers.

In addition to crawlers, there are also listeners that can be used to record announcements for new blocks and transactions.

3.3 Limitations

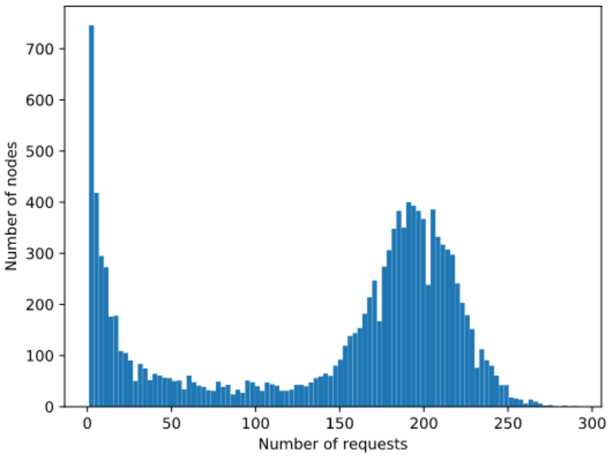

Figure 2: Request distribution for online peers

Figure 2 shows the distribution of the number of requests to the contacted nodes during our five day experiment described further in Section 5. There is a large number of peers that are below an amount of address requests which would give us a complete view of the node's peer database (80 as defined by Biryukov et al. [1]) and are most likely very

[5] https://dsn.tm.kit.edu/bitcoin/
https://github.com/sebicas/bitcoin-sniffer
https://github.com/ayeowch/bitnodes

short-lived. However, for the median, the successful connections with 178 bring the probability of not receiving a certain address from the peer to a negligible amount.

As with any peer-to-peer network, there is also the challenge of churn in the network. According to Imtiaz et al. [15] 97% of network participants actually have intermittent connectivity, further reducing the completeness of our view on the network due to fast fluctuations in the network. We manage to crawl our current view of the network in 1.5h.

Due to the fact that our implementation is using asynchronous programming and runs coroutines for contacting individual remote peers, the time measurements are not guaranteed to be accurate as there could be some inaccuracies with the yielding of other coroutines.

4 METHODOLOGY

In this Section we will introduce our methodology for latency based proxy and VPN detection.

4.1 Approach

In order to recognise proxy usage, we first formulate three types of latencies:

- latency to the visible IP l_v
- latency to the device of Bitcoin peer l_e
- latency to the Bitcoin application l_{btc}

Our approach is based on the assumption that there is a significant difference between the round trip time from our infrastructure to the visible neighbour IP (l_v) and the round trip time to the Bitcoin endpoint (l_e). Similarly, we analyse the latency difference between a measurement to the socket used for Bitcoin (l_e) and a measurement taken inside the Bitcoin protocol (l_{btc}) in order to estimate the processing delay introduced by queues and calculations at the Bitcoin peer.

To further get insights into the makeup of network latency in the Bitcoin network we calculate the following values.

- $\Delta_{proxy} = l_e - l_v$
- $\Delta_{proc} = l_{btc} - l_e$

Δ_{proxy} is the latency difference between the visible IP and the device of the Bitcoin peer. Conversely Δ_{proc} should correspond roughly to the processing time at the node itself as it is the difference between the latency to the device itself and the latency measured inside the Bitcoin protocol. In practice, we select the minimum latencies of each measurement type for each worker in order to minimise jitter. With these minimum latencies, we calculate Δ_{proxy} both TCPing and ICMP based, and Δ_{proc} .

To have a confidence level of our measurements, we utilise the standard deviation of our measurements:

- σ_{proxy}
- σ_{proc}

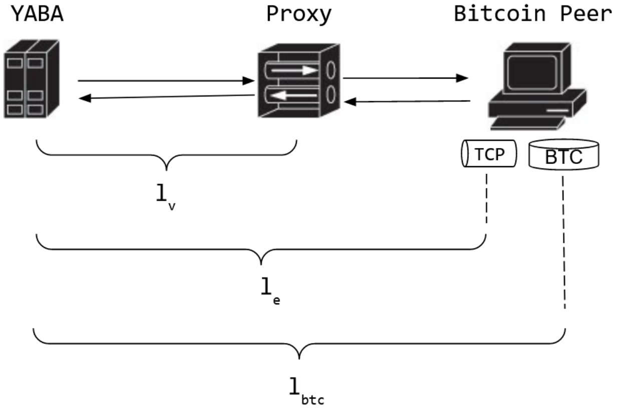

Figure 3: Example for connection with and without proxy usage

σ_{proxy} and σ_{proc} are the standard deviations between the σ_{proxy} and σ_{proc} of each geographically distributed worker. As each worker has a different path to the destination and possibly different peering agreements a low $\sigma_{proxy}/\sigma_{proc}$ should indicate a high confidence that $\Delta_{proxy}/\Delta_{proc}$ are due to proxy usage and not network anomalies.

4.2 Measuring l_v

In order to measure the latency to the visible IP, we explore two different approaches.

ICMP Echo There is a widespread practice of dropping ICMP echo messages, often for security reasons which can limit the effectiveness to measure latency using ICMP. However, in our overall measurement, we found 49% of peers had enabled their ICMP echo. Another consideration, apart from the availability of echo responses, is the potential problem of ICMP low priority during routing, leading to inaccuracies.

TCP Pings In order to increase the coverage of our system and estimate the impact of ICMP low priority routing in the measurements we used TCP pings to common ports such as 22 (ssh), 80 (http), 443 (https/openVPN), 1194 (openVPN) and 1080 (SOCKS). These are potentially not forwarded by a proxy but rather used by the server providing the proxy service instead. In our experiment data, we observed one of these ports to be reachable for 58% of nodes. In total, 31% of nodes had some of the common ports enabled while dropping ICMP echo messages. This was especially pronounced for AWS hosted nodes as the default policy of AWS EC2 is to drop ICMP echo requests. TCP ping to common ports was 0.6ms faster (mean) than ICMP in our experiment data.

Combining Measurements Our total coverage combining ICMP and TCP pings was 83%.

4.3 Measuring l_e

The latency l_e describes the time when the packet reaches the server/computer of the peer running the Bitcoin software. We, therefore, measure it through the TCP connection establishment to the port that is used by Bitcoin at the remote peer. For 96% of peers this is 8333. The second most popular port is 39388 which seems to originate from btcpayservers [6] which are deployed via docker and are mostly not reachable by outgoing connections. Most other non-standard ports seem to be manual changes i.e 8334, 8555, 8999 etc.

4.4 Measuring l_{btc}

The Bitcoin protocol includes keep-alive messages which are ping/pong messages including a nonce. Active peers are obligated to answer ping messages. We measure the latency of a ping/pong exchange to get a value for l_{btc}.

5 EXPERIMENT

5.1 Data Collection

We collected data from 27/5/2020 until 1/6/2020. For this, we deployed the YABA coordinator in Berlin. The crawlers were geographically distributed using a mix of public cloud resources in Ireland, Ohio and Hong Kong as well as resources in Berlin. These regions (EU/NA/Asia) cover the vast majority of participants in the Bitcoin network. The crawlers were connected through IPv4 as well as IPv6 with a total of 26,928,250 connection attempts. During this time we connected to 204,739 unique addresses of which 11,853 were reachable. Of these reachable nodes, 10,012 actually performed a Bitcoin handshake procedure with us. The median number of successful requests per peer was 178.

5.2 Proxy Usage

In Figure 4 the distribution of $\Delta proxy$ for all peers in our measurement can be observed. It has to be noted that these measurements are limited to the 83% of peers that we can actively probe. Finding a threshold for $\Delta proxy$ to designate connections as proxy connections is a hard task, however, the distribution of $\Delta proxy$ gives us an interesting image. Assuming proxies are outliers when observing latency measurements (shown on the right side in Figure 4), we chose the threshold as the upper bound of the $\Delta proxy$ boxplot, resulting in a threshold of 1ms. This chosen threshold was confirmed in further manual analysis and port scanning of SOCKS port 1080 resulting in 0.29% suspected SOCKS users. Although further refinement should be the goal of future work.

Overall there are 5.2% (178) of probed users with a $\Delta proxy$ larger than 1ms. As we have mentioned previously, only relying on $\Delta proxy$ leaves us open to a lot of inaccuracies. We therefore approximate accuracy of $\Delta proxy$ with σ_{proxy}. For this paper, we will further inspect peers with a σ_{proxy} less than 1ms. As is shown in Figure 5 a significant σ_{proxy} is present in most of our measurements, especially so in the ICMP based approach. For our analysis, we ,therefore, rely on the measurement with the lowest σ_{proxy}.

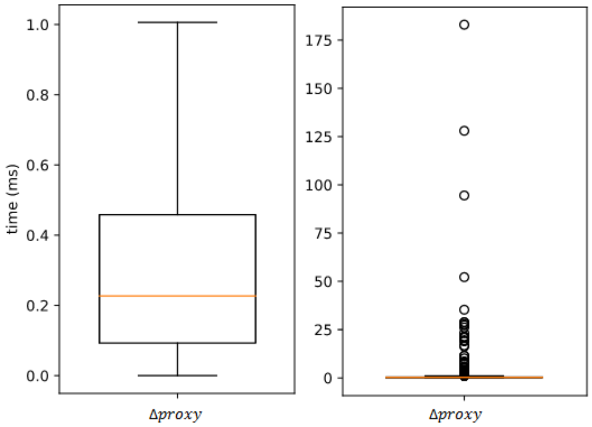

Figure 4: $\Delta proxy$ for TCP and ICMP based measurement approach depending on which has lower $\sigma proxy$

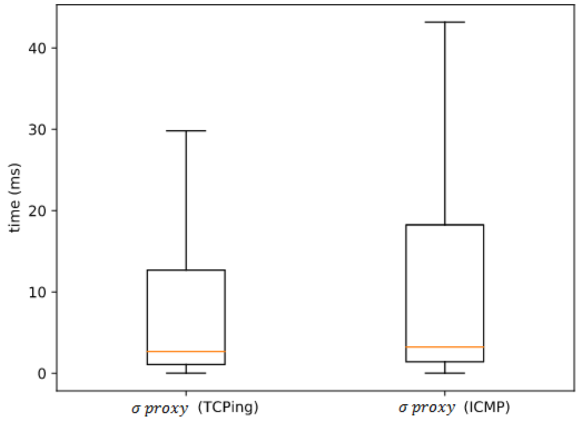

Figure 5: Standard Deviation σ_{proxy} for both TCP and ICMP based methods

With the requirement of $\sigma_{proxy} < 1ms$ as threshold for high confidence detection, we found 16 servers utilising a proxy to participate in the Bitcoin peer-to-peer network. This results in 0.4% proxy usage in the network if our sample with low σ_{proxy} is representative.

As mentioned in Section 4 there are a number of different scenarios how the proxy could be set up in terms of technical aspects (port forwarding etc.) or what they are used for.

[6] https://github.com/btcpayserver

AS	Δ_{proc}	σ_{proc}	$\Delta_{proxy}(TCPing)$	$\sigma_{proxy}(TCPing)$	$\Delta_{proxy}(ICMP)$	$\sigma_{proxy}(ICMP)$
Hetzner	66.54ms	54.29ms	32.03ms	0.86ms	85.2ms	96.0ms
TalkTalk	37.9ms	12.4ms	0.22ms	1.17ms	39.08ms	0.88ms
Hangzhou Alibaba	420.96ms	83.47ms	1.99ms	1.2ms	183.63ms	0.42ms

Table 1: Examples of detected proxies

We will show some examples of these scenarios using our gathered data. In Table 1 the exact measurements can be observed.

One reason we are using both TCP and ICMP is that although there is a good possibility (from manual analysis of the gathered data) that TCP pings are forwarded to the proxy user it sometimes still gives us the desired results. One such example is the peer shown in Table 1 in the Hetzner AS. Here we can observe a high σ_{proxy} when measured through ICMP, yet the TCPing approach gives us very accurate measurements. In the inverse, there is the example of a peer in the TalkTalk AS. Here the TCPing shows a very small Δ_{proxy} leading us to believe that they are indeed forwarded. ICMP on the other hand has a considerable Δ_{proxy}, even with a very good σ_{proxy}, indicating that it is used as proxy. Most of the instances we identified as proxy use, had Δ_{proxy} of 5ms indicating that for these detected users the motivation was not to change jurisdiction but rather general network obfuscation. Two potential examples of using proxy to circumvent regulations are from the AS Hangzhou Alibaba and DigitalOcean (Singapore). Here the Δ_{proxy} of 183ms and 128ms respectively indicate that the user of the proxy is geographically further away with countries such as Vietnam and China having some restrictions on the use of Bitcoin. In order to confirm our results we manually checked against a publicly available proxyblacklist [7] and our results were confirmed for all inspected high certainty detections. In addition to our latency based proxy detection we also used zmap [8] for port scanning of common proxy/VPN ports. For this we selected a range of ports (88, 8008, 443, 992, 1080, 1194, 2460, 3389, 5005, 5555, 1912, 12200, 260000, 981, 12975, 32976, 655) that are used by VPN and proxy providers. In Table 2 we show the number of reachable peers for the most prevalent ports. As some ports such as 80 and most importantly 443 are normally used by http(s) we had to clean the initially 11.7% reachable peers on those ports from the ones using it for http traffic rather than proxy/VPN. The prevalence of SOCKS closely mirrors our results from the latency based approach. But also the usage of VPN seems to be in the same order of magnitude. As expected, typically around 30% of the peers reachable on their proxy/VPN service port were not reachable through Bitcoin, suggesting they are not doing port forwarding on the needed port.

Port	Reachable Peers
443	0.31%
1194	0.7%
1080	0.39%

Table 2: Common VPN port scan

The described usage percentage for either detection method is well below the 18% usage rate according to [17] for general internet usage in 2019. As mentioned by Krombholz et al. although 18% of users take steps to stay anonymous in the Bitcoin network, the majority uses Tor instead of proxies [18]. Most importantly, our active probing approaches rely on the reachability of the peer and its visible IP therefore not capturing the use-case of VPN usage by Bitcoin peers not used as "server". This fact, in combination with the observation of the Δ_{proxy} being often under 30ms for our measurements with high confidence indicates that the discovered proxies are not consumer VPNs.

5.3 Processing Time

Figure 6 shows the processing delay for all observed peers and suspected VPNs. In general, the σ_{proc} was a lot larger than σ_{proxy}, leading us to the conclusion that load on Bitcoin peers can be quite periodic. As median the processing time for all peers was 77.5ms. This is compared to 112ms for suspected VPNs.

In combination with reverse domain lookup, we found that monitoring and seeding provided by large educational institutions such as Karlsruhe Institute of Technology and RWTH Aachen and other organisations have a large processing time due to their high load.

6 DISCUSSION

In this section we will discuss some correlations with open source intelligence.

6.1 Geographic Distribution

The geographic distribution of our high confidence proxy detections is shown in Figure 7. Although the sample size is

[7] ipqualityscore.com

[8] github.com/zmap/zmap

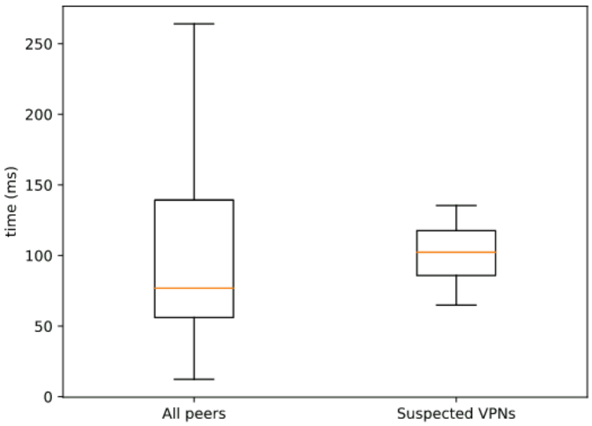

Figure 6: Processing delay of Bitcoin peers

small, the distribution seems to roughly correspond with the normal Bitcoin geographic distribution.

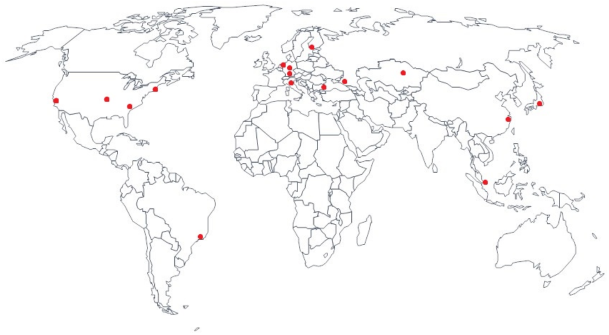

Figure 7: Geographic distribution of suspected proxies

6.2 ASN distribution

While the overall discovered peers heavily tend towards consumer ISPs, it has to be noted that suspected proxies had to be online in order to measure the latency difference or scan ports, as we couldn't reliably employ a passive approach. We see this fact as the main reason the data is biased towards long-lived or always online and, most importantly, reachable servers probably. This is consistent with popularity of providers such as NordVPN that do not support port forwarding [9] compared to AirVPN [10], Windscribe [11] or Mullvad [12] that do.

AS Organisation	Online Peers
Hetzner Online	27.1%
DigitalOcean	25%
OVH SAS	14%
Google	10%
Alibaba	6%
Amazon	3%

Table 3: Distribution of peers in autonomous systems

6.3 Reverse domain lookup

Of the 16 high confidence detections we manually explored, all but one had a domain associated with them at one point. Most of them had a Bitcoin DNS seed pointed at them.

- seed.bitcoin.sipa.be
- dnsseed.bluematt.me
- seed.bitcoin.sprovoost.nl
- seed.bitnodes.io
- dnsseed.emzy.de

This can be explained by the fact that with increased uptime, we are expected to gather more accurate measurements and therefore, the high confidence detections will be biased towards long-lived peers often chosen for seeding.

We ran reverse DNS on all our experiment data and successfully found records for 6.1% of all IPs yet. One interesting observation can be made in that the German "de" top-level domain is very over-represented compared to the overall percentage of German participants. Upon further investigation, this over-representation of the .de TLD is due to the fact that some ISPs provide dynamic DNS to their customers and are responsible for 70% (Deutsche Telekom), 18.5% (Versatel), 4.8% (NetCologne) and 4.4% (Telefonica) of all "de" results.

TLD	Overall
de	52%
com	12%
net	11%
nl	4.2%
ch	3.8 %

7 CONCLUSION

In this paper we present a measurement study of the Bitcoin peer-to-peer network with the goal of gauging the prevalence and motivations of proxy usage in the Bitcoin ecosystem. We developed a system to gather Bitcoin peer-to-peer network data at scale with geographically distributed workers. We described our approach to analyse the collected latency measurements to infer proxy usage and processing time. We conclude that the overall usage of proxies in the network is around 0.4%. This latency based detection approach yielded

[9] https://support.nordvpn.com/FAQ/1047408432/Do-you-offer-port-forwarding.htm
[10] https://airvpn.org/faq/p2p/
[11] https://windscribe.com/features/port-forwarding
[12] https://mullvad.net/en/help/port-forwarding-and-mullvad/

very similar results to a more simple port scanning of common proxy/VPN service ports.

8 FUTURE WORK

There are a number of scenarios which our approaches of active proxy detection are not able to detect. The most glaring is the inability to detect VPN providers which use residential or otherwise for our probing unreachable VPN servers. This means a potentially large use-case of end-users utilising well-advertised consumer VPNs is not covered in this project. Tackling this issue is the goal of our future work.

We want to explore this problem with a two-fold approach. On the one hand, we want to improve latency prediction from geolocation, however, we recognise the inherent inaccuracy of this approach. Hence we propose that another passive approach exploiting measurements of σ_{proxy} is more promising. The hypothesis that we want to investigate in our future work is that a large σ_{proxy} might indicate a consumer VPN shared by a number of participants for outgoing connections. Combined with further exploration of fingerprinting Bitcoin peers this could lead to the detection of such scenarios.

ACKNOWLEDGMENTS

This work has been supported and partially funded by the German Federal Printing Office (Bundesdruckerei).

REFERENCES

[1] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, ACM, 15–29.

[2] Jeremiah Bohr and Masooda Bashir. 2014. Who uses bitcoin? an exploration of the bitcoin community. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust*. IEEE, 94–101.

[3] Domhnall Carlin, Jonah Burgess, Philip O'Kane, and Sakir Sezer. 2019. You could be mine (d): the rise of cryptojacking. *IEEE Security & Privacy* 18, 2 (2019), 16–22.

[4] Global Legal Research Center. 2018. Regulation of Cryptocurrency Around the World. *The Law Library of Congress* (2018).

[5] Christian Decker and Roger Wattenhofer. 2013. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*. IEEE, 1–10.

[6] Sergi Delgado-Segura, Surya Bakshi, Cristina Perez-Sola, James Litton, Andrew Pachulski, Andrew Miller, and Bobby Bhattacharjee. 2019. TxProbe: Discovering Bitcoin's network topology using orphan transactions. In *International Conference on Financial Cryptography and Data Security*. Springer, 550–566.

[7] Joan Antoni Donet Donet, Cristina Pérez-Sola, and Jordi Herrera-Joancomartí. 2014. The bitcoin P2P network. In *International Conference on Financial Cryptography and Data Security*. Springer, 87–102.

[8] David L Donoho, Ana Georgina Flesia, Umesh Shankar, Vern Paxson, Jason Coit, and Stuart Staniford. 2002. Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 17–35.

[9] Marc Dupuis, Tamara Geiger, Marshelle Slayton, and Frances Dewing. 2019. The Use and Non-Use of Cybersecurity Tools Among Consumers: Do They Want Help?. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education*. ACM, 81–86.

[10] Guillermo Escobero Hernández. 2019. *Characterization of the topology of the Bitcoin network*. B.S. thesis. Universidad Carlos III de Madrid.

[11] Sebastian Feld, Mirco Schönfeld, and Martin Werner. 2014. Analyzing the Deployment of Bitcoin's P2P Network under an AS-level Perspective. *Procedia Computer Science* 32 (2014), 1121–1126.

[12] Matthias Grundmann, Till Neudecker, and Hannes Hartenstein. 2018. Exploiting transaction accumulation and double spends for topology inference in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 113–126.

[13] Ivan Hernandez, Masooda Bashir, Gahyun Jeon, and Jeremiah Bohr. 2014. Are Bitcoin Users Less Sociable? An analysis of users' language and social connections on twitter. In *International Conference on Human-Computer Interaction*. Springer, 26–31.

[14] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, and Haixin Duan. 2018. How you get shot in the back: A systematical study about cryptojacking in the real world. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1701–1713.

[15] Muhammad Anas Imtiaz, David Starobinski, Ari Trachtenberg, and Nabeel Younis. 2019. Churn in the Bitcoin Network: Characterization and impact. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 431–439.

[16] Irni Eliana Khairuddin, Corina Sas, Sarah Clinch, and Nigel Davies. 2016. Exploring motivations for bitcoin technology usage. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2872–2878.

[17] Mohammad Taha Khan, Joe DeBlasio, Geoffrey M Voelker, Alex C Snoeren, Chris Kanich, and Narseo Vallina-Rodriguez. 2018. An empirical analysis of the commercial vpn ecosystem. In *Proceedings of the Internet Measurement Conference 2018*. ACM, 443–456.

[18] Katharina Krombholz, Aljosha Judmayer, Matthias Gusenbauer, and Edgar Weippl. 2016. The other side of the coin: User experiences with bitcoin security and privacy. In *International conference on financial cryptography and data security*. Springer, 555–580.

[19] Matthias Lischke and Benjamin Fabian. 2016. Analyzing the bitcoin network: The first four years. *Future Internet* 8, 1 (2016), 7.

[20] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2015. Discovering bitcoin's public topology and influential nodes. *et al* (2015).

[21] Satoshi Nakamoto et al. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[22] Daniel Stutzbach and Reza Rejaie. 2005. Evaluating the accuracy of captured snapshots by peer-to-peer crawlers. In *International Workshop on Passive and Active Network Measurement*. Springer, 353–357.

[23] Wikileaks. 2012. *Twitter.* https://twitter.com/wikileaks/status/234727069282607104

[24] Aaron Yelowitz and Matthew Wilson. 2015. Characteristics of Bitcoin users: an analysis of Google search data. *Applied Economics Letters* 22, 13 (2015), 1030–1036.

[25] Yin Zhang and Vern Paxson. 2000. Detecting stepping stones.. In *USENIX Security Symposium*, Vol. 171. 184.