Towards Better Understanding of Bitcoin Unreachable Peers

Liang Wang, Ivan Pustogarov
UW-Madison, Cornell Tech
liangw@cs.wisc.edu, ivan.pustogarov@cornell.edu

arXiv:1709.06837v1 [cs.NI] 20 Sep 2017

ABSTRACT

The bitcoin peer-to-peer network has drawn significant attention from researchers, but so far has mostly focused on publicly visible portions of the network, i.e., publicly reachable peers. This mostly ignores the hidden parts of the network: unreachable Bitcoin peers behind NATs and firewalls. In this paper, we characterize Bitcoin peers that might be behind NATs or firewalls from different perspectives. Using a special-purpose measurement tool we conduct a large scale measurement study of the Bitcoin network, and discover several previously unreported usage patterns: a small number of peers are involved in the propagation of 89% of all bitcoin transactions, public cloud services are being used for Bitcoin network probing and crawling, a large amount of transactions are generated from only two mobile applications. We also empirically evaluate a method that uses timing information to re-identify the peer that created a transaction against unreachable peers. We find this method very accurate for peers that use the latest version of the Bitcoin Core client.

1. INTRODUCTION

Bitcoin [13] is a cryptocurrency and a peer-to-peer network. The explosion in its popularity is fueled in large part by its decentralized nature, low transaction fees, and ease in participating. A whole Bitcoin ecosystem, the core of which is the Bitcoin network, exists today. Understanding its properties and usage patterns of its participants can give insights into improving the Bitcoin network, as well as the ecosystem.

Known research on the Bitcoin network has mostly focused on the publicly visible part of the network, i.e., publicly reachable peers, and ignored the hidden part – clients behind NATs and firewalls that do not allow inbound connections [3, 6, 7, 12, 14]. Nonetheless, gathering statistics on such peers is equally, if not even more important, since the number of such peers is estimated to be to order of magnitude larger than the number of reachable peers. In this paper we conduct a large scale measurement to collect different statistics and analyze the usage patterns of unreachable Bitcoin peers that might hide behind NAT or firewalls. To facilitate our measurement, we designed and implemented an open-source measurement tool that call we *bcclient*, which can serve as a Bitcoin node but with extended features to aid in collecting specific connection and transaction information. We conduct a measurement study using bcclient using 102 Bitcoin nodes, which were distributed over 14 geographical regions spanning all continents, in the Bitcoin main network for seven days, and collected information on about 3 M connections and 2.5 M unique transactions. These connections are generated from about 190 K IPv4 IPs, and we used bcclient to discover that 87% of these IPs can be associated with unreachable peers. Our further analysis on collected data suggests:

- Bitcoin unreachable peers appear to be centralized in terms of Internet routing and transaction propagation: 16% of these peers are hosted in only 5 Autonomous Systems (which constitues only 0.07% of all observed ASes), and 50 unreachable peers are involved in propagation of 43% of transactions.
- About 80% of unreachable peers are associated with two mobile Bitcoin applications, which contribute to 61% of unique transactions and 20% of connections.
- A large fraction of connections came from IPs in public clouds; a considerable amount of these connections might be used for Bitcoin network probing and crawling.

Inspired by our measurement results, we further exercised bcclient and an experimental framework to empirically evaluate a variation of a known method that uses timing information to re-identify the peer that created a transaction against unreachable peers. This method can be useful to further enrich the collected dataset with valuable statistics, such as number transaction generated by each country. In our experiments we were able to re-identify every second transaction generated by us with the latest at the time of writing version of Bitcoin Core software (v0.14.1). We observed negligible false positive rates. We have made the source code for bcclient publicly available to facilitate future research.

Roadmap: We start by providing the necessary background to understand Bitcoin peer-to-peer network and message propagation rules in Section 2. In Section 3, we present our custom-built Bitcoin software, describe our large scale experiments, and present statistics on Bitcoin clients. In Section 4, we empirically evaluate our peer triage method. Section 5 concludes the paper.

2. BACKGROUND

Bitcoin P2P network. Bitcoin is a digital decentralized currency that relies on cryptography and a peer-to-peer network for double-spending prevention instead of a trusted

third party. Bitcoin users can pay each other by creating cryptographically signed *transactions* and broadcasting them in the Bitcoin peer-to-peer network. In order to facilitate transaction propagation the network implements a simple gossip protocol: every peer that received a message forwards it to its neighbors.

The core of the Bitcoin network is a set of about 7000 servers with public IP addresses operated by volunteers and companies. The network is open and anybody can run a Bitcoin server and contribute with bandwidth and computational resources. Bitcoin users (often behind firewalls and/or NAT's) that do not allow inbound connection access the network, send transactions, and learn about transactions of others through these servers. The list of servers is publicly available and the default behavior for a server is to accept up to 117 inbound connections on TCP port 8333. The default behaviour for a Bitcoin client is to maintain connections to 8 different servers. After Bitcoin peers establish a TCP connection, they complete a handshake protocol by exchanging `VERSION` messages, containing among other fields the software version/name.

Transaction forwarding. A common transaction consists of the sender's and recipient's Bitcoin addresses, the sender's public key and the signature. This makes the transaction pseudonymous as there is not any identifying information besides a randomly looking public key/address of the user who generated it. One possible way to identify the sender though is to monitor the network traffic, and to look at the IP address from where the transaction was first sent. In order to make such traffic analysis harder and improve users' privacy, the Bitcoin protocol defines special rules when forwarding a transaction. The goal of these rules is to make a transaction generated by a user indistinguishable from transactions generated by others.

First, according to the Bitcoin reference implementation whenever a Bitcoin peer (either a client or a server) receives a transaction from one of its neighbors, it broadcasts it further to the rest of its neighbors. In this way the peer's own transactions get mixed with the ones it relays.

Second, when a peer generates or receives a new transaction it does not relay it immediately to its neighbors. Instead it chooses and assigns a random exponentially distributed delay for each of its neighbors; the actual transmissions occur when the corresponding delays expire. This impedes timing analysis for an attacker. The mean of exponential distribution is different for inbound (5 seconds) and outbound (2.5 seconds) connections. The random exponentially distributed delays were introduced in the most recent (at the time of writing) version of Bitcoin reference implementation and previous research [8, 10] did not take this modification into account.

Bitcoin Testnet. In order to facilitate testing of new features, the Bitcoin community runs a small separate testing network called *Testnet* with an independent Blockchain. At the time of submission Testnet consisted of about 250 nodes with public IP addresses. By analogy we call the actual Bitcoin network *Mainnet*.

3. MEASUREMENTS

3.1 Methodology and dataset

Most of the existing research on the Bitcoin P2P network focuses on its backbone: the servers with public IP addresses. Such research is facilitated by the fact that one can readily connect to the servers and start collecting the data. In this paper we go one step deeper and try to collect data on the less visible (but nonetheless large) part of the Bitcoin ecosystem: clients behind NATs and firewalls that do not accept incoming connections. We refer to them as *unreachable peers*. During our measurement we injected more than 100 Bitcoin servers with public IPs distributed over the globe into the network and were collecting data on incoming connections. We were mostly interested in the following:

- The number of incoming connections over time and number of unique IP addresses. We use these figures to estimate the number of Bitcoin clients behind NATs and firewalls.
- Location and Autonomous Systems of the discovered IP addresses. This information can be used to better understand which countries use Bitcoin the most and whether Bitcoin clients are centralized in terms of Internet routing.
- Type of software used by Bitcoin clients. This can give us some insights on whether Bitcoin clients use smartphones to access the network, and whether clients access Bitcoin through Tor.
- Volume and properties of transactions generated by clients which can be used to infer Bitcoin clients usage patterns.

Bcclient. In order to facilitate our measurements we developed a special-purpose measurement tool, called *bcclient*, using libbitcoin library [4]. In a nutshell, bcclient is a customized, lightweight Bitcoin software that can either run as Bitcoin full node or client, while offering extended functionality such as (1) recording specified events of users' interest with precise timestamp, (2) establishing several parallel connections to a given Bitcoin node, (3) running as a full node without downloading any part of Blockchain, and more. Bcclient was optimized for memory and storage usage, and for processing huge volumes of data, which make it suitable for conducting large-scale and long-term measurement experiments in the main Bitcoin network. Bcclient also includes an analysis engine that contains a set of tools for analyzing data. The code of bcclient was released publicly to facilitate future research[1].

Data collection. For our data collection we set up a total of 102 AWS EC2 servers running in diverse networks around the world, as shown in Figure 1 [11]. More specifically, we set up servers in each of the availability zones in each AWS EC2 region (that correspond to different geographical locations). On each of the EC2 servers we were running an instance of bcclient. We were logging detailed information on

[1]https://github.com/ivanpustogarov/bcclient

Figure 1: Number of measurement nodes in different locations.

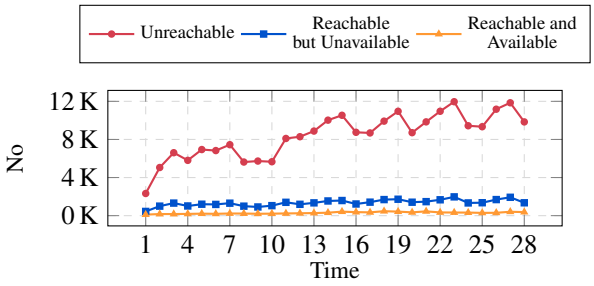

Figure 2: Change over time in different types of IPs that our nodes have seen. Each point on X-axis represents a 6-hour period.

inbound connections from IPv4 addresses such as connection IP, timing, transaction information, etc. We periodically collected logs from all bcclient instances, and extracted and deduplicated source IP addresses of inbound connections.

During our study we were interested primarily in clients behind NATs and firewalls, and thus for each extracted IP address we checked if we were able to establish a reverse connection back to the client. We sent a TCP SYN probe to its port 8333 (Bitcoin default), tried to complete the Bitcoin handshake protocol if the port was open, and recorded the connection result, i,e., success or fail.

We have run the measurement three times in March and May 2017. In this paper, we use the 60 G of data collected from the latest 168-hour measurement from May 10, 2017 to May 17, 2017.

Ethical considerations. To avoid adding burden on the peers being scanned, we only tried to establish a reverse connection to an IP once in every 6 hours, and would close a connection immediately after the connection was established successfully. We used a special version string in bcclient so operators of Bitcoin public peers could find and contact us if they did not want to be probed.

Overview of dataset. During our experiments we observed a total of 2,956,515 inbound connections from 189,204 unique source IP addresses; for 99.6% of these connections, we had the timing information: both their opening times and closing times in our logs. We say such connections are *completed*

Our measurement nodes have received a total of 2,490,042 unique transactions from 4,516 IPs. A transaction might be relayed to our nodes multiple times by different peers; we refer to each transaction receive event as *propagation*. We saw 83,878,269 propagations.

3.2 Characterizing Bitcoin clients

Reachable vs. unreachable. We observed that most of the incoming connection come from peers that do not accept incoming connections (i.e. unreachable peers).

We extracted a total of 189,204 unique source IP addresses (IPv4) from the completed connections and categorized them into groups based on whether they responded to our probes.

Unreachable IPs. We say an IP address is unreachable if it fails to respond to any of our TCP probes during the measurement. We found that 86.8% of the collected IPs are unreachable. We mainly focus on unreachable IPs in our analysis, but also report on statistics about other types of IPs for comparison purposes in some cases.

Reachable IPs. We say that an IP address is reachable if we were able to successfully establish a TCP connection at port 8333 back to such IP address. We found only 13.2% of IPs to be reachable under this definition. We further split reachable peers into those that successfully completed the Bitcoin handshake protocol (we call them *available*) and those that did not (we call them *unavailable*). We found only 1,587 (0.8%) IPs successfully completed the handshake. We further checked them against the IPs of public Bitcoin nodes provided by [5], and found that 1,073 of available IPs belong to the set of public Bitcoin servers. Table 3 further shows the breakdown of number of connections, transactions, and propagations from different type of IP addresses.

Remark. It is possible for peers to use a non-default port in which case we would mark them as unreachable.

Estimating number of clients. Figure 2 shows how the number of reachable and unreachable IPs changed over time. At the beginning, the number of unreachable peers gradually increases; this is due to the fact our Bitcoin servers were new to the network and Bitcoin clients were learning their IP addresses over time. After 78 hours the number of unreachable peers become relatively stable with the average about 10K peers over 6 hours periods. This can be extrapolated to estimate the total number of Bitcoin peers behind NATs and firewalls. Given that an unreachable client establishes 3.5 parallel connections to the network on the average (see below) and that there were 5,540 Bitcoin servers with IPv4 address on the average during our experiment, we estimate the number of unreachable clients to be at least 155,000 at any given 6-hours time interval.

Remark. Note that this number assumes that there is one-to-one correspondence between an IP address and a client. Different clients however can share the same NAT address, and the real number of unreachable clients might be bigger. An unreachable or unavailable IP could be used as a *frontend* IP, which is for the gateway of a network, or the frontend of a cluster of peers.

Client version. We extracted a total of 534 unique client version strings from collected connections, and found that the most popular three Bitcoin client software across all IPs are BitcoinWallet, Breadwallet, and Bitcore, which are associated with 59.9%, 20.0%, and 6.9% of all IPs respectively. Unreachable and unavailable IPs are mostly associated with version strings “Bitcoin Wallet” or “breadwal-

Type	# IP	# Conn
0	164,198 (86.8)	936,845 (31.8)
1	23,419 (12.4)	1,095,445 (37.2)
2	1,587 (0.8)	911,173 (31.0)
Total	189,204 (100)	2,943,463 (100)
Type	**# Prop**	**# TX**
0	40,071,909 (47.8)	2,409,087
1	8,987,300 (10.7)	8,987,300
2	34,819,060 (41.5)	2,354,397
Total	83,878,269 (100)	2,490,042

Table 3: An overview of dataset. "Conn","TX" and "Prop" stand for connection, unique transaction ID, and propagation, respectively. Type 0, 1, 2 correspond to unreachable, unavailable, available IPs, respectively. Number in parenthesis is the percentage of the total number in the last row.

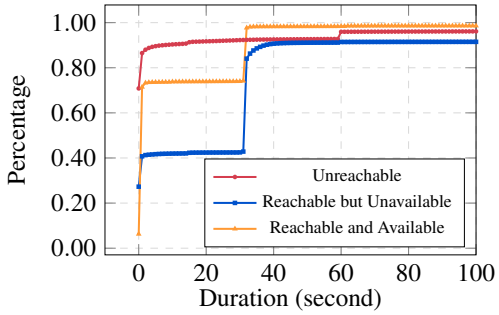

Figure 4: CDF of durations of connections. A connection with "0" duration means its duration is less than one second. X axis is truncated at 100.

let". We looked at the their source code and found that BitcoinWallet establishes 3 parallel connections to the network; Breadwallet establishes either 4 or 6 parallel connections depending on the amount of RAM on the device.

Available IPs are mostly associated with version strings "Satoshi" or "BitcoinUnlimited". We also found 17,144 unreachable IPs (19,813 of all IPs) have sent empty version strings. Note that we saw 13% of IPs were associated with multiple version strings which suggests that these are different users behind the same NAT address.

We found many users with outdated versions of these apps: only 48.3% of BitcoinWallet and 79.4% of Breadwallet clients are using the latest releases. Surprisingly, 98.9% of Bitcore clients uses outdated versions of client software.

Centralization of Bitcoin clients. We checked geolocation (i.e., country) and Autonomous Systems (ASes) of all the collected IPs using a tool called pyasn [1]. We observed that Bitcoin peers are highly centralized in terms of Internet routing [2] and transaction propagation.

(1) *Peers location.* The observed connections coming from 7,070 distinct ASes that span all countries. The country with most IPs is United States, which has 24.2% of collected IPs, followed by Germany (7.3%), Canada (5.5%), Russia (4.9%), and United Kingdom (4.7%).

(2) *Autonomous systems.* We found that just 5 ASes (which constitues only 0.07% of all observed ASes) host as much as 16.0% of the unreachable peers; these top 5 ASes are associated with T-Mobile, Comcast, Verizon, ATT, and Rogers which suggests that associated peers might run on mobile devices. Furthemore 100 ASes (which constitues only 1.4% of all observed ASes) host as much as 60.1% of oberved IPs.

We also analyzed how active each AS was. We found that 63.5% of completed connections were generated from only 10 ASes. Furthermore we found that near 99% of the ASes are only associated with a small number of connections – each of these ASes generated less than 0.1% of completed connections. For connections from unreachable IPs the result is similar: 33% of them are from 10 ASes.

(3) *Transaction propagation.* We observed that a small number of IPs are associated with most of the transaction propagations. More specifically, 100 IPs are associated with 89% of all the propagations; and 50 unreachable IPs are involved in 43% of all the propagations. These IPs could become the "bottleneck" of Bitcoin networks, i.e., taking down one of these IPs can affect propagation of a considerably large number of transactions. Inspecting the top ten IPs in terms of relayed transactions manually, five of them are public Bitcoin nodes; however, the other IPs seems not be frontends of services or gateways of networks. This leaves why they involved in so many transactions an open question and we are in process of inspecting it.

Remark. Centralization in Bitcoin peers might cause security and privacy issues [2]. For example, most of unreachable peers are associated to two mobile Bitcoin applications, which means that security of a large number of transaction is in hands of just two companies.

3.3 Understanding usage patterns.

Connection duration. We found that a large number of incoming connections were short-lived connections; we also discovered peers performing crawling and probing for Bitcoin network by establishing connections with very small duration.

We first examined the duration of completed connections for all IPs; the median was only 1.3 seconds; 93.9% of connections were less than 60 seconds; and 34.6% of the connections had duration less than 1 second. Only about 0.5% of completed connections lasted for more than 1 hour and we found a large fraction of them came from a service called BitcoinRussia. We compared durations for different types of connections. As shown in Figure 4, connections from unreachable IPs have higher percentage of short-live connections, e.g., the median connection duration from such IPs is 0.5 seconds and more than 70% of the connection durations are less than 1 second.

We examined connections with very small duration: we call a connection *ephemeral* if its duration is less than 0.5 second. A large portion (48.3%) of connections from unreachable IPs are ephemeral. These connections account for 59.4% of all ephemeral connections (760,948 in total). Unavailable IPs contribute 34.7% of all ephemeral connections. We checked the client version strings in the ephemeral connections from unreachable IPs and found 49% of them are "breadwallet" or "Bitcoin Wallet", which indicates these connections might issued by two mobile Bitcoin applications: breadwallet and Bitcoin Wallet. We will show more

Type	Service	Client version	# conn	%
1	BitcoinRussia	bt-russia.ru:0.0.1f	338,399	11.5
2	Hetzner	bitnodes.21.co:0.1	222,594	7.6
1	Linnode	8btc.com:1.0	194,285	6.5
1	DigitalOcean	Snoopy:0.2.1	191,228	6.5
0	Google	bitcoin-seeder:0.01	53,843	1.8
1	DigitalOcean	bitcoin-seeder:0.01	53,592	1.8
2	Amazon	bitcoin-seeder:0.01	52,491	1.8
2	Amazon	bitcoin-seeder:0.01	52,312	1.8
0	Amazon	bitcoin-seeder:0.01	52,267	1.8
2	Amazon	bitcoin-seeder:0.01	52,078	1.8

Table 5: Top 10 IPs based on number of associated connections per IP. Type 0, 1, 2 correspond to unreachable, unavailable, available IPs, respectively. Percentages are the fraction of the total number of completed connections.

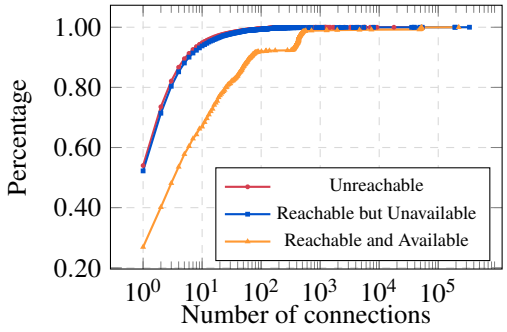

Figure 6: CDF of the total number connections from a given IP.

analysis results on these two applications in later section.

For the ephemeral connections from unavailable IPs, 188,536 out of 264,313 (or 71%) connections, are associated with two client version strings: "Snoopy" or "bitcoin-seeder". Note that Snoopy or bitcoin-seeder are tools for performing crawling and probing for Bitcoin network. The client version strings suggest all of these clients might be used for performing Bitcoin network crawling or probing. More surprisingly, 96% of these suspicious probing connections from unavailable IPs, which is 6.1% of all completed connections, came from 4 IPs in two public clouds: three IPs are in DigitalOcean and one IP in AWS EC2. Actually, almost all the probing connections originated from clouds were from only one unavailable IP in DigitalOcean, and some Bitcoin developers already suspected that this IP was for conducting unknown attacks against Bitcoin network.

Connection frequency. Next, we examine how many connections each unreachable peer made during the measurement period. We found about 54% of unreachable IPs only made one connection, 95% made less than 10, and 99% made less than 50, while unavailable IPs have a very similar long-tailed distribution (see Figure 6). In contrast, available IPs tend to make more connections on average: only 37% made one connection, while about 9% of them made more than 100 connections and 2% made more than 500 connections. Some unreachable or unavailable IPs have made a large number of connections, as shown in Table 5. As we can see from Table 5, seven out of ten IPs belong to public clouds.

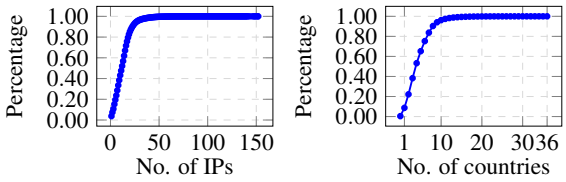

Figure 7: CDF of number of IPs **(left)** and number of countries **(right)** that involve in the propagation of a given transactions.

Transaction characteristics. As shown in Figure 7, a propagation could be relayed by as many as 211 IPs. The IPs that are involved with a given propagation could be distributed across ASes and countries. For example, 1% of propagations are associated with IPs from more than 14 countries. We further found propagations could be relayed by different types of peers: only 5.3% of the propagations are relayed by IPs of the same type, while 75.4% are relayed by a mix of unreachable, unavailable, and available IPs.

Another observation is that propagations are usually sent over long-live connections, e.g, 99.9% of the propagations were sent over the connections the durations of which were longer than 100 seconds. We grouped propagations based on source IP types and found the observations still hold true for different groups of propagations. Only 64 transactions were sent over ephemeral connections.

Characterizing different types of clients In this section, we focus on different types of peers: (1) *mobile* peers, which are the peers whose version strings contain "breadwallet" or "Bitcoin Wallet", (2) *probe* peers, which are whose version strings contain "Snoopy" or "bitcoin-seeder", and (3) *tor* peers, which are the peers that also serve as Tor exit nodes or relay nodes. They represent different types of clients.

As shown in Table 8, mobile peers are associated with 131,610 of unreachable IPs, which is 80% of unreachable IPs. These IPs contributed to about 20% of all connections, most of which were short-lived connections (less than 0.5 seconds). We saw only 1.2% of propagations sent from these IPs; however, these propagations correspond to 61% of unique transaction IDs. On the average, five transactions were sent over a connection. One possible explanation for this phenomena is that these peers were configured not to forwarding transactions.

We observed a small number of IPs that we also identified as either Tor exit or Tor relay nodes. Similar to mobile peers, they are involved in less than 1% of propagations but have sent over 41% of transactions. Compared to other types of peers, such tor peers have relatively longer connection durations: the median duration of tor peers, probe peers, and mobile peers are 2.1, 1.2, and 0.4, respectively

No probe peers from available IPs have sent any transactions but generated a large number of connections.

Limitation and discussion. We didn't verify if a transaction is valid or invalid. Doing so requires a parser for Bitcoin blockchain information, and we are working on such parser as part of bcclient analysis engine.

	Mobile			Probe			Tor		
	Unreachable	Unavailable	Available	Unreachable	Unavailable	Available	Unreachable	Unavailable	Available
#IP	131,610 (69.6)	16,862 (8.9)	210 (0.11)	12 (0.01)	38 (0.02)	490 (0.26)	171 (0.09)	295 (0.16)	14 (0.01)
#Conn	506,921 (17)	67,356 (2.3)	2,165 (0.07)	97,627 (3.3)	323,885 (11.0)	567,069 (19.3)	2,069 (0.07)	5,819 (0.2)	137
#EphConn	314,394 (10.7)	39,724 (1.4)	1,422 (0.05)	195 (0.01)	188,536 (6.4)	1,011 (0.03)	66	300 (0.01)	71
#Prop	1,008,567 (1.2)	497,848 (0.6)	386,557 (0.5)	41,009 (0.05)	154,393 (0.18)	0	219,261 (0.26)	533,865 (0.64)	46,753 (0.06)
#TX	1,518,206 (61.0)	626,736 (25.2)	679,264 (27.3)	41,283 (1.7)	168,589 (6.7)	0	255,699 (10.3)	717,077 (28.9)	61,112 (2.5)

Table 8: A breakdown of IP, connections and transactions for different types of peers. “IP”, “Conn”, “EphCOnn”, “Prop”, and “TX” stand for connection, unique IP addresses, connections, ephemeral connections, propagations, and unique transaction ID respectively. Number in parenthesis is the percentage of the total number of all IPs/connections/propagations /unique transactions IDs, depending on statistics type.

4. ON FURTHER ENRICHING THE DATASET

In the previous section, we observed that a single transaction can be forwarded by as many as 151 different IP addresses (excluding IPs of publicly known full Bitcoin nodes) from a wide range of countries which makes extracting per-country transaction origin statistics problematic: it would require one to distinguish between the peer that actually created the transaction (*originator*) from the rest (i.e. peers that merely relayed it, we call such peers *relays*). This problem is well-suited for timing analysis used in [15] for inferring Bitcoin network topology of publicly reachable peers. We are working on implementing the timing analysis approach for unreachable clients as a part of the analysis engine of bc-client. As a preliminary step, we want to understand how efficient this method would be for unreachable peers, and design experiments to evaluate it. We find this method to be quite efficient and reliable.

To better explain the problem, consider a simple example in Figure 9 with two peers C and C', that connect to the Bitcoin network as regular unreachable clients; C is connected to one of our *monitoring* nodes A (which is a public Bitcoin peer but doesn't connect to any other peers). Assume C and C' generate transactions tx and tx' both of which will be delivered to A through C. Node A wants to distinguish transaction tx (for which C is true originator) from tx' (for which C is a relay). To do this we introduce additional *listener* node L that is connected and listens to the publicly known set of Bitcoin servers and use the following criteria to distinguish between these two cases: *If node A receives tx before L, we conclude that C is the originator for tx, and a relay otherwise*.

Experiments in the Testnet. We first evaluate true-positive rate of this method i.e. the chance of successfully identifying C as the originator of tx, by carrying out experiments in the Bitcoin Testnet. For all the experiments, C was running Bitcoin core software version v0.14.1. In total we conducted 21 different experiments in which we sent 8400 transactions from node C. We moved nodes A and C across 14 different geographical regions (see Section 3) and used transactions of different sizes (i.e., the number of receiver addresses in the transaction, chosen from 5, 10, 20, 50, and 100) in order to estimate how these factors affect the efficacy of our timing analysis. We found that the true-positive rates varied from experiment to experiment (from 20% to 60%), but on the average was consistent across different regions and different transaction sizes. The resulting true positives rates for each of the 14 geographical regions are shown in Figure 10 with the average 45.43% over all regions.

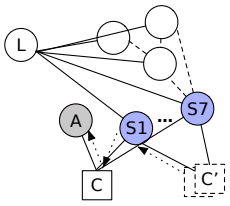

Figure 9: Distinguishing transaction true originator

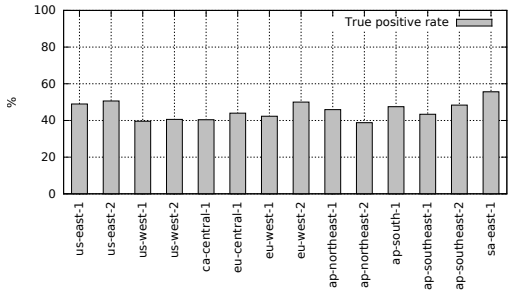

Figure 10: True positive rate for different regions.

To estimate the false-positive rate, we set up client C and monitoring node A in the same availability zone in `us-east-1` region. We count the number of transactions forwarded by node C that appeared at A before L. In order to estimate how number of concurrent connections affects the false-positive rate, we make listener node L establish different number concurrent connections (1, 2, 5, 10, 15, 20) to every peer in the Bitcoin network. For each setting, we record 5,000 transactions relayed by C, and examine how many of them are false positives. We only discovered a total of 12 false positives out of 30,000 transactions (which amounts for just 0.04%).

To better understand this result we further look at the transaction latency (i.e., the time between C creates a transaction and node A receives it) under different transaction sizes and region combinations, and find that in the majority of cases client delay (see Section 2) dominates network latency. For instance, even though C and A are in the same EC2 zone, the longest transaction latency can be as long as 8.5 seconds.

Experiments in the Mainnet. We now estimate true-positive and false-positive rates in the Bitcoin Mainnet. We first

place client node *C* monitoring node *A* in the same region (us-east-1) and carry out two experiments in which we vary client's entry nodes; we send 20 transactions from client *C* in each of the experiments. The resulting true positive rate is 65% and 40%. Next we place client *C* and node *A* in different regions (us-east-1 and ap-northeast-1) and make client *C* send 24 transactions. The true positive rate for this case is 41.6%. These true-positive rates are very close to the success rate we obtained earlier in the Testnet. We finally estimate the false-positive rate. Out of 25000 transactions only 20 arrived first at node *A*, which amounts to 0.08% only. This proves that our method is a very reliable way to identify transaction origins.

Discussion. Kaminsky was the first who described this general idea in his BlackHat presentation [9] in the context of client deanonymization, and Koshy et al. [10] was among the first to try to practically evaluate it. While he was able to deanonymize a number transactions that exposed anomalous behavior (e.g. transactions relayed only once or transaction that were relayed multiple times by the same IP), he concludes that assigning a transaction's ownership to its first relayer is ineffective. Neudecker et al. [15] evaluate this approach in the mainnet but only consider public reachable peers. Compared to their results, we actually achieve much better performance. We believe the discrepancy with our results is due to that we considered unreachable peers and levereaged the recent modifications to Bitcoin core software.

5. CONCLUSION

In this paper, we developed an experimental framework which we used to collect and analyze various statistics on Bitcoin unreachable peers, a part of the Bitcoin network that despite its size did not receive much attention from the research community but which is crucial to enhance the scientific understanding of the Bitcoin system.

In our study we deployed monitoring nodes spread over the globe and collected more than two million connections and transactions. We find several previously unreported and surprising results. First, we find that Bitcoin unreachable peers appear to be centralized: a large number of peer are hosted in few Autonomous Systems. Second, the version messages that we received from the clients as well their ISP's strongly suggest that the overwhelming majority of Bitcoin users access Bitcoin through a mobile application. Moreover, we found that those peers are associated to two mobile Bitcoin applications, which means that security of a large number of transaction is in hands of just two companies. Third, we discovered a large fraction of connections came from public clouds; these might be used by researcher and other interested parties to monitor and crawl the network. We further experimentally evaluated a timing analysis-based technique for triage the first relay that forwards a given transaction and our results suggest the technique can achieve near zero false-positive rates against unreachable Bitcoin peers.

We believe that our work sheds more light into a more hidden part of the Bitcoin network and is a valuable step towards better understanding the Bitcoin ecosystem as a whole.

6. REFERENCES

[1] P. I. address to Autonomous System Number lookup module. https://github.com/hadiasghari/pyasn, 2014.

[2] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies.

[3] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking bitcoin: Large-scale network attacks on cryptocurrencies. *CoRR*, abs/1605.07524, 2016.

[4] Bitcoin Cross-Platform C++ Development Toolkit. https://github.com/libbitcoin/libbitcoin, 2017.

[5] G. B. N. D. Bitnodes. https://bitnodes.21.co/, 2017.

[6] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.

[7] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun. Tampering with the delivery of blocks and transactions in bitcoin. Cryptology ePrint Archive, Report 2015/578, 2015. http://eprint.iacr.org/2015/578.

[8] P. L. Juhász, J. Stéger, D. Kondor, and G. Vattay. A Bayesian Approach to Identify Bitcoin Users. *ArXiv e-prints*, Dec. 2016.

[9] D. Kaminsky. http://www.slideshare.net/dakami/black-ops-of-tcpip-2011-black-hat-usa-2011, 2011.

[10] P. Koshy, D. Koshy, and P. McDaniel. An analysis of anonymity in bitcoin using P2P network traffic. In *Proceedings of Financial Cryptography and Data Security (FC'14)*. Springer, 2014.

[11] H. Liu. Amazon data center size. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html, 2012.

[12] A. Miller, J. Litton, A. Pachulski, N. Gupta, D. Levin, N. Spring, and B. Bhattacharjee. Discovering bitcoins public topology and influential nodes. 2015.

[13] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009. http://www.bitcoin.org/bitcoin.pdf.

[14] T. Neudecker, P. Andelfinger, and H. Hartenstein. A simulation model for analysis of attacks on the bitcoin peer-to-peer network. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1327–1332, May 2015.

[15] T. Neudecker, P. Andelfinger, and H. Hartenstein. Timing analysis for inferring the topology of the bitcoin peer-to-peer network. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*, pages 358–367. IEEE, 2016.