Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Building a Multi-Cryptocurrency Node Explorer

Semester Thesis

Xin Hong

xihongt@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Dr. Lucianna Kiffer
Prof. Dr. Roger Wattenhofer

August 27, 2024

Acknowledgements

I want to express my heartfelt gratitude to my supervisor, Dr. Lucianna Kiffer, for her guidance and supervision throughout this project. Her advice, inspiration, and positive affirmation during our weekly meetings helped me navigate challenges more effectively, deepened my understanding of the subject, and instilled a sense of accomplishment. This project could not have been smoothly completed without her unwavering support and patience.

Abstract

The domain of cryptocurrency networks has developed significantly in recent years, with research exploring various aspects of network dynamics. Previous theses have developed three custom crawlers to discover all nodes and retrieve their routing tables from the Bitcoin-based networks, the Ethereum execution layer, and the Ethereum consensus layer, respectively. However, each crawler targets an individual system, leaving a gap in comprehensive comparisons across these major networks.

This thesis contributes by developing a pipeline that systematizes the crawling and connectivity checking processes of these three crawlers and extracts information from the collected data to perform various analyses. Additionally, a website has been constructed to display up-to-date visualizations derived from these analyses. Our goal is to provide the community with a utility to view and analyze the dynamics of these major cryptocurrency networks.

Contents

Chapter 1

Introduction

1.1 Motivation

Cryptocurrency networks, such as Bitcoin and Ethereum, operate on a peer-to-peer (P2P) model that enables decentralized synchronization of peers' system state views and message dissemination. Numerous research has explored various facets of cryptocurrency P2P networks, including network topology and structure, consensus mechanisms, transaction propagation and latency, and so on. However, prior studies typically concentrate on individual systems, leaving broader comparisons between major networks as an open question.

Previous theses have leveraged corresponding P2P discovery protocols to develop three custom crawlers that identify known nodes and retrieve their routing tables across a range of Bitcoin-based networks, the Ethereum consensus layer, and the Ethereum execution layer, respectively.

In this work, we aim to create a comprehensive pipeline to systemize the crawling process for these networks, extract information on node behavior and network dynamics from collected data. Additionally, we develop a website to display up-to-date information of all these networks, providing a useful tool for the community to understand and analyze cryptocurrency network dynamics.

Chapter 2

Background

In previous theses, students have developed three custom crawlers for exploring all known nodes and their routing tables in different cryptocurrency networks. One crawler is designed for the Bitcoin-based networks and could be adapted to support a range of cryptocurrencies. For this thesis, it supports Bitcoin, Bitcoin Cash, Bitcoin SV, Litecoin and Dogecoin. The other two crawlers are developed for the two layers of the Ethereum network respectively. These distinct layers, including the execution layer and the consensus layer, result from *The Merge* which transitioned the Ethereum network from proof-of-work to proof-of-stake protocol to reduce energy consumption. These crawlers serve as the data sources for the website.

All three crawlers follow a similar workflow. After obtaining the seed nodes from public APIs or pre-discovered sources, the crawler establishes the connection with them and requests their routing tables. By recursively querying the neighbor nodes in these tables until no new nodes are found, the crawler performs a thorough network traversal.

In addition to the crawlers, three relative connectivity checkers are developed. Post-crawl, these checkers monitor the identified nodes by pinging or establishing protocol-level connections with the nodes to track the node activity.

2.1 Bitcoin-based Networks Crawler

The custom Bitcoin-based crawler [1] was developed in Go, leveraging the "btcsuite" [2] and "ltcsuite" [3] packages which provide full node and wire protocol implementations. This crawler is compatible with IPv4 addresses.

Random seed nodes for each currency are sourced via the Blockchair API [4]. The connectivity checker performs TCP pings and version handshakes on all discovered nodes.

The crawling and connectivity results are recorded in JSON files using key-value pairs, where the key is the node's IP address with the port attribute, and the value is a boolean indicating whether the node was active during the daily crawl or responsive during a particular hourly connectivity check. The routing table for each active node is also stored as a list of nodes in a JSON file.

2.2 Ethereum Network Crawlers

The node record formats of these two layers are different, so their crawlers run different verisons of a Kamdemlia-based discovery protocol, with the crawler for the exeuction layer running Discv4 [5] and the crawler for the consensus layer runing Discv5 [6]. All the crawling and checking results of these two crawlers are archived in text format.

2.2.1 Ethereum Execution Layer Crawler

The execution layer uses the Enode URL format [7] to identify nodes, which includes the node ID, IP address, and TCP port. The custom crawler for this layer [8], developed in Go and based on go-ethereum's implementation of Discv4, is designed to collect all Enodes.

Specifically, the crawler employs a pre-computed lookup table for querying nodes in the routing tables to improve efficiency. The connectivity checker sends pings to all unique Enodes discovered and follows up with ICMP pings to each unique IP in each check.

The crawler produces a list of unique Enodes and IP addresses along with the routing tables of each Enode per crawl. And The connectivity checker generates a list of responsive Enodes and IPs for each check.

2.2.2 Ethereum Consensus Layer Crawler

The consensus layer uses the Ethereum Node Records (ENRs) format [9] to identify nodes, which is a signed key-value records that encapsulate more information of a node than Enode. The custom crawler [10] for the consensus layer was developed in Rust, compatible with IPv4 addresses, and instrumental in collecting ENRs of all Beacon nodes running the Discv5 protocol.

The crawler uses predetermined hardcoded ENRs as boot nodes. The connectivity checker sends pings to all ENRs via the Discv5 protocol during each checking process.

The crawler generates a list of discovered ENRs per crawl as well as the routing table for each responding ENR. And The connectivity checker produces a list of active ENRs from each check.

Chapter 3

Data Processing

The three crawlers generate daily archives of all discovered nodes, along with the routing tables of the responding nodes during the crawling process across all networks. The connectivity checkers produce hourly databases of the active nodes. Following the crawl, the data processing phase extracts information from the collected data, conducts various analyses, and generates corresponding plots to illustrate the dynamics of these networks. These plots are then displayed on our custom multi-cryptocurrency node explorer website *http://multi-node-explorer.ethz.ch* [11].

3.1 Data Extraction

The website showcases predominantly common plots for all network, supplemented by several network-specific ones.

3.1.1 Common Analyses for All Crawlers

Continuity Analysis Across Crawl

To analyse the continuity of nodes across crawls, the counts of all discovered nodes as well as the active ones that respond to at least one check from the connectivity checker in each crawl are recorded. Additionally, nodes from each crawl are monitored in the following seven crawls to identify patterns of persistence and overlap.

Temporal analysis of Node Participation

To assess the stability of node participation in the network, the number of nodes responding to the connectivity checker over seven successive days is tracked for each crawl, and the cumulative distribution of node activity, based on the number of days they were active over the past ten days, is computed. Furthermore, to assess the temporal stability of ENRs and Enodes, the responses of a specific set to the connectivity checkers—the active ENRs/Enodes from hour 57 of the most recent 168 hours (the latest 7 days)—are tracked throughout the entire timeline.

Routing Table Analysis

To gauge the frequency of updates to the routing tables, the analysis is performed on common nodes that appeared across the latest ten crawls. All common nodes from the Bitcoin-based data are included, while 5% of common nodes from both layers of the Ethereum network are sampled to manage data processing time, given their significantly larger size of the routing table files which contain many ENRs or Enodes.

The routing table sizes of the common nodes are recorded to analyze their distribution. In order to assess the overall persistence of routing tables across crawls, the average overlap of routing tables from the initial crawl with those in the subsequent seven crawls is computed. Specifically for the Ethereum network, we track the persistence of routing tables for nodes with the highest ENR or Enode turnovers and their longevity in routing tables after being shut down.

Geographical Analysis

The distribution of geographical and organizational information of the discovered nodes is of interest. Information about the country, ASN, and company to which a node belongs is retrieved using a combination of the GeoLite2 database [12] and the IPinfo API [13]. To minimize API calls and improve retrieval efficiency, a local cache file storing the geographical information of recently encountered nodes is used and refreshed regularly.

Mainnet Analysis

Since most enterprise applications utilizing Ethereum technology are built on the public Ethereum Mainnet, data on Mainnet nodes in both the consensus and execution layers of the Ethereum network is of particular interest, and some plots are generated additionally for these Mainnet nodes. Mainnet nodes in the consensus layer are filtered by decoding their ETH2 value in their ENRs and comparing the decoded forkdigest values with Mainnet forkdigest values that were manually searched and collected. However, in the execution layer, it is not possible to determine which network a node is participating in based solely on its Enodes. To estimate the portion of Mainnet nodes, we filter IPs participating in the Mainnet using statistics from a popular Ethereum node explorer *ethernodes.org* [14], providing a lower bound on their number.

3.1.2 Crawler Specific Analysis

Cross-Network IP Analysis

For Bitcoin-based networks, the number of IPs active across various cryptocurrency networks over the past seven days is counted to examine node overlap between different networks. For the Ethereum network, the daily overlap of discovered IPs appearing in both the consensus layer and the execution layer is recorded.

ENR analysis

To assess how a node might change its ENRs for each key-value pair, nodes appeared in at least two different crawls over the past seven days are selected and categorized for each key based on their values into three groups. For each key, nodes with an empty value for a key are grouped as "None", nodes with unchanged values are classified as "Same values", and nodes with varying values are categorized as "Different values". The distributions are then displayed in a stacked bar plot.

We are particularly interested in the ETH2 value from the ENRs, which indicates the network the nodes are running on and their fork choices. The decoded data of testnet, current fork version, and future fork version of the encountered nodes with non-empty ETH2 values over the past ten days are collected and grouped.

3.1.3 Data Processing Implementation

The plots for all the analyses are generated using the Plotly library [15] after extracting the data for those analysis. Plotly is chosen for its expertise in creating interactive, publication-quality visualizations and its extensive range of features for generating various charts and plots. Data intended for long-term display, such as daily counts of discovered nodes, is stored in local databases.

3.2 Website Construction

3.2.1 Website Implementation

The web pages are deployed using the Python framework Dash [16], which is designed for building interactive web applications, particularly those involving data visualization, and integrates seamlessly with Plotly.

The website is currently hosted on a virtual machine using the Nginx web server [16] and is accessible via HTTP at *http://multi-node-explorer.ethz.ch*.

3.2.2 Website Structure

The website consists of four pages. The main page introduces the general workflow of the crawlers and displays the counts of nodes discovered by each of the three crawlers. The remaining pages present plots for data from Bitcoin-based networks, the consensus layer of Ethereum, and the execution layer of the Ethereum network, respectively.

3.3 Pipeline

The execution of crawlers, connectivity checkers, data processing, and plot generation is automated using a crontab setup on a Linux machine at ETH Zurich. The Bitcoin-based networks crawler, Ethereum consensus layer crawler, and Ethereum execution layer crawler run sequentially starting at 1 a.m. each day. Once each crawl is complete, the data processing phase starts to extract data, generates updated plots, and then uploads them to the hosting virtual machine of the website. Raw data files produced by the cralwers and the connectivity checkers, once processed and no longer needed to be re-read, are compressed regularly to save space. All crawling and data processing are completed by 5 a.m. From 5 a.m. to 11 p.m., the three connectivity checkers execute on an hourly basis.

Chapter 4

Results

All the plots generated from the data processing phase could be viewed on the website *http://multi-node-explorer.ethz.ch*. An example of each type of plot is shown here.

4.1 Common Analysis for All Crawlers

4.1.1 Continuity Analysis Across Crawl

Count of Discovered Nodes per Crawl

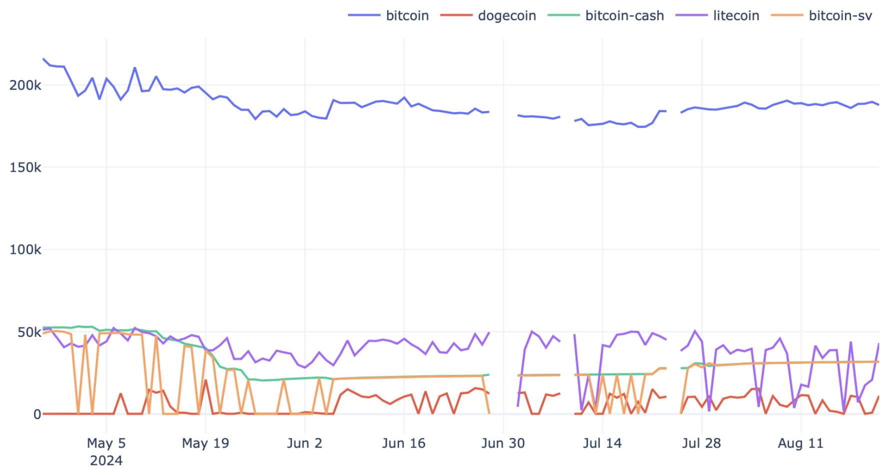


Figure 4.1: Count of Discovered Nodes per Crawl of Bitcoin-based Networks

As shown in Figure 4.1, line charts displaying the count of discovered objects per crawl and active ones from the start date of regular crawler operations are generated. For Bitcoin-based networks, node counts for each currency are plotted; for the Ethereum consensus layer, counts of ENRs, nodes, and specifically those in Mainnet are shown; and for the Ethereum execution layer, counts of Enodes, IPs, and Mainnet IPs are plotted.

Persistence of Discovered Nodes per Crawl Over Time

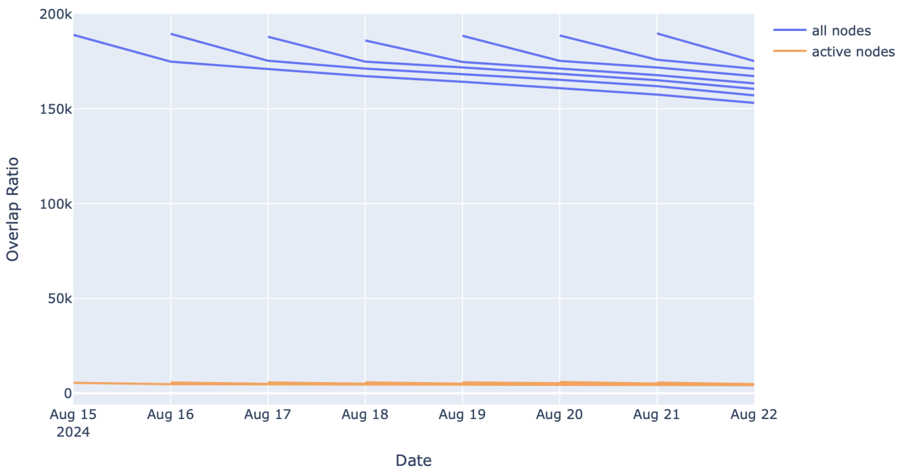


Figure 4.2: Persistence of Discovered Nodes per Crawl of Bitcoin Nodes

As shown in Figure 4.2, line charts illustrating the persistence of all nodes and active nodes across subsequent crawls from the latest seven days are generated. Persistence is measured by the count of nodes from the base crawl that reappear in following crawls. For Bitcoin-based networks, nodes for each currencies are tracked; for the Ethereum consensus layer, both nodes and Mainnet nodes are included; and for the Ethereum execution layer, Enodes, nodes, IPs, and Mainnet IPs are considered.

4.1.2 Temporal Analysis of Node Participation

Node Responsiveness to the Hourly Connectivity Check

Figure 4.3: Bitcoin Nodes Responsiveness to the Hourly Connectivity Check

As shown in Figure 4.3, line charts tracking the number of nodes discovered during the crawling process and the number of active ones that remain responsive to hourly connectivity checks over time are generated.

ENRs / Enodes Responsiveness to the Hourly Pings

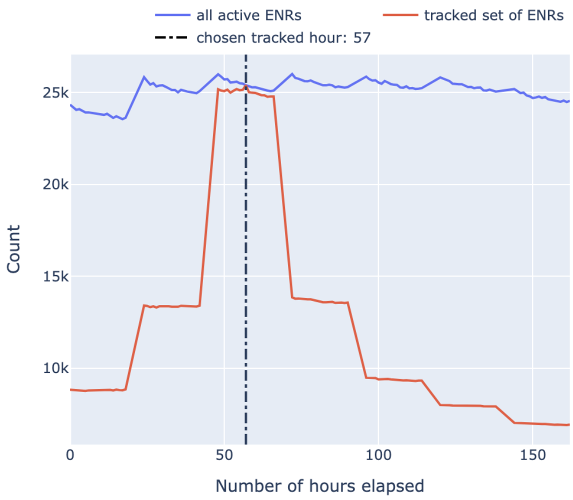


Figure 4.4: Responsiveness of All ENRs to the Hourly Connectivity Check

The temporal stability of ENRs and Enodes is specifically monitored by selecting a specific set, such as the active ENRs / Enodes from hour 57 of the most recent 168 hours (the latest 7 days). Their responsiveness to hourly connectivity checks is then tracked throughout the entire timeline, as shown in Figure 4.4.

Node Responsiveness in Terms of Active Days Count

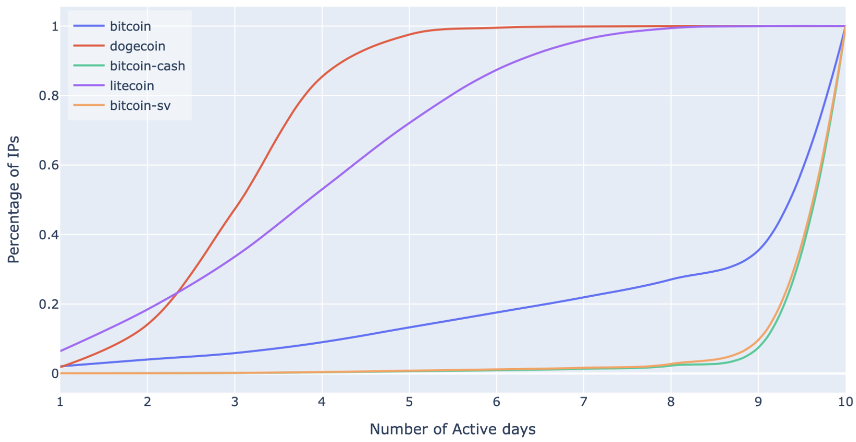


Figure 4.5: Responsiveness of Bitcoin Nodes in Terms of Active Days Count

As shown in Figure 4.5, the cumulative distribution of activity for all Nodes discovered from 7 days ago to 4 days ago, based on the number of days they were active from 10 days ago to 1 day ago are generated.

4.1.3 Routing Table Analysis

The plots related to routing table analysis include data from common nodes identified in the latest ten crawls.

Distribution of Non-empty Routing Table Sizes of Nodes

Histograms displaying the distribution of non-empty routing table sizes for these nodes, with mean and median values provided, are produced as shown in Figure 4.6. For Bitcoin-based networks, the routing table size of a node is calculated as the average of its routing table sizes across all included crawls. For the Ethereum network, the routing table size of a node in a crawl is determined as the average of the routing table sizes of all its ENRs/Enodes, and the overall routing table size is calculated as the average across all included crawls.

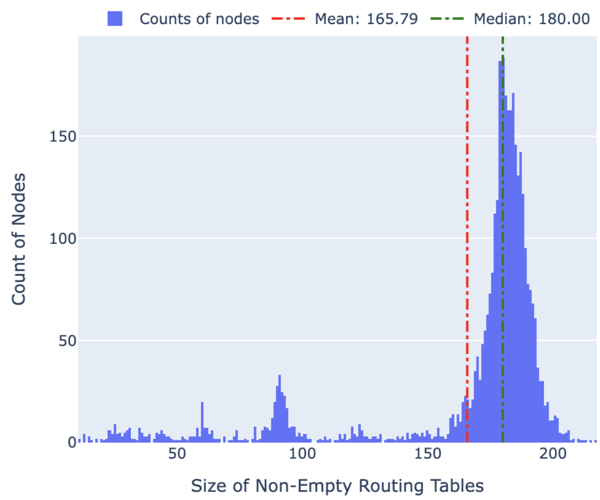


Figure 4.6: Distribution of Non-empty Routing Table Sizes of Ethereum Consensus Layer Nodes

Overlap of Routing Tables with Starting Crawls of Nodes Over Time

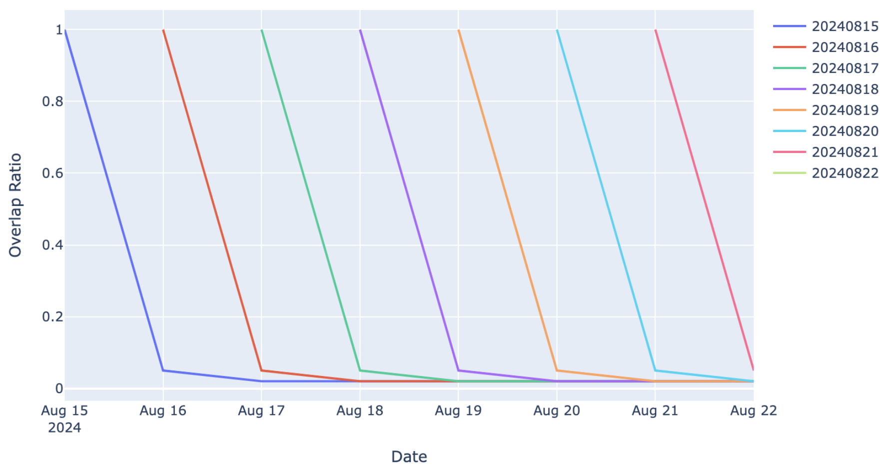


Figure 4.7: Overlap of Routing Tables with Starting Crawls of Bitcoin Nodes Over Time

Line charts displaying the average overlap of routing tables from the initial crawls in subsequent crawls are created as shown in Figure 4.7. The routing table overlap ratio of a node is defined as the quotient of the intersection of its routing tables retrieved from the base and subsequent crawls, divided by its routing table size from the base crawl. Then the routing table overlap ratio between crawls are calculated as the average of the routing table overlap ratios for common nodes in these two crawls.

Persistence in Routing Tables of Nodes with a High ENR/Enode Churn

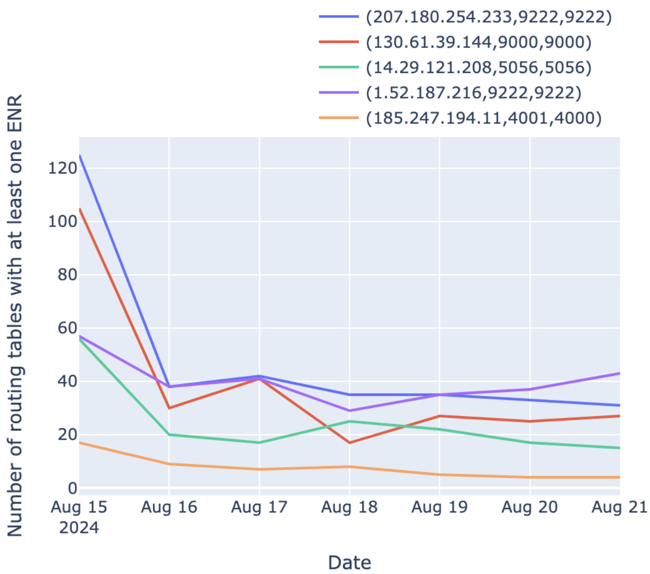


Figure 4.8: Persistence in Routing Tables of Ethereum Execution Layer Nodes wth a High Enode Churn

Line charts, as shown in Figure 4.8, illustrate the persistence of routing tables across subsequent crawls for the top 5 nodes with the highest ENR/Enode turnover. ENR/Enode churn of a node is assessed by the total number of its ENRs/Enodes across all included crawls. The persistence of a node's routing tables is measured by the number of routing tables in each crawl that include at least one ENR/Enode from its tracked set.

Longevity of Ethereum Nodes in Routing Tables after They Have Been Shut Down

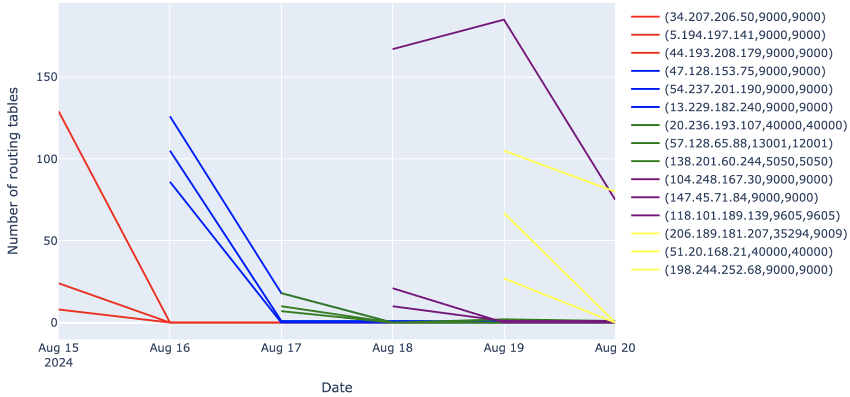
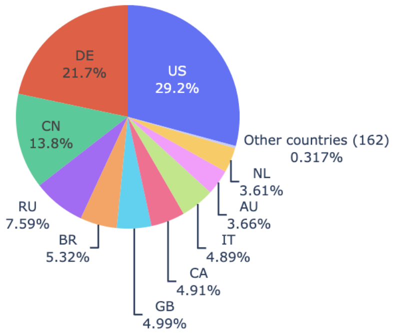


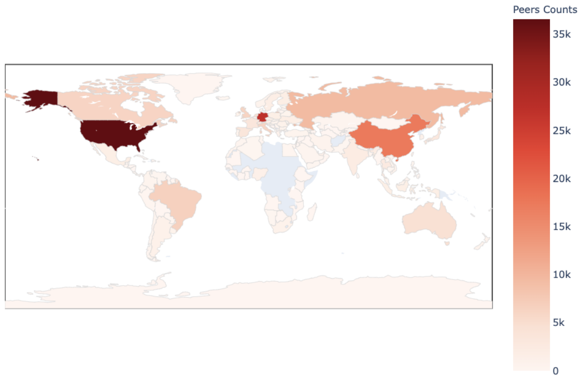
Figure 4.9: Longevity of Ethereum Execution Layer Nodes in Routing Tables after They Have Been Shut Down

The plots in Figure 4.9 illustrate the longevity of the top 3 deactivated nodes with the most ENRs/Enodes from each crawl. A node is deemed deactivated if none of its ENRs/Enodes respond to connectivity checks in subsequent crawls. The node's longevity is evaluated by the number of routing tables in which its ENRs/Enodes continue to appear despite deactivation. For Ethereum consensus layer nodes, all routing tables are checked, whereas for Ethereum execution layer nodes, 5% of routing tables are sampled to manage plot generation time.

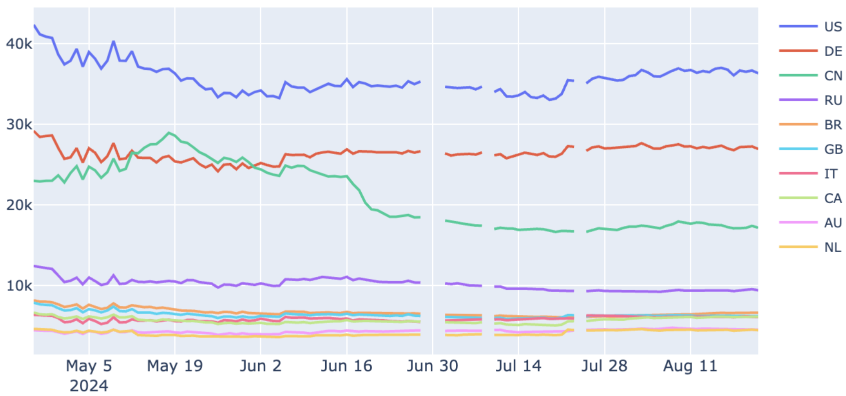
4.1.4 Geographical Analysis



(a)



(b)



(c)

Figure 4.10: Country Information of All Bitcoin Nodes

The country, Autonomous System Number (ASN), and company information for each node were collected and visualized using pie charts, line charts, and maps for geographical and organizational analysis. The pie charts display the distribution of the top 10 countries/ASNs/companies with the most nodes, while the line charts show the changes in the number of nodes discovered for these entities. The map illustrates the overall geographical distribution of the nodes' countries in the most recent crawl. Figure 4.10 provides an example of the plots showing the country information for Bitcoin nodes.

4.2 Crawler Specific Analysis

4.2.1 Cross-Network IP Analysis

Cross-Network IPs in Bitcoin-based Networks

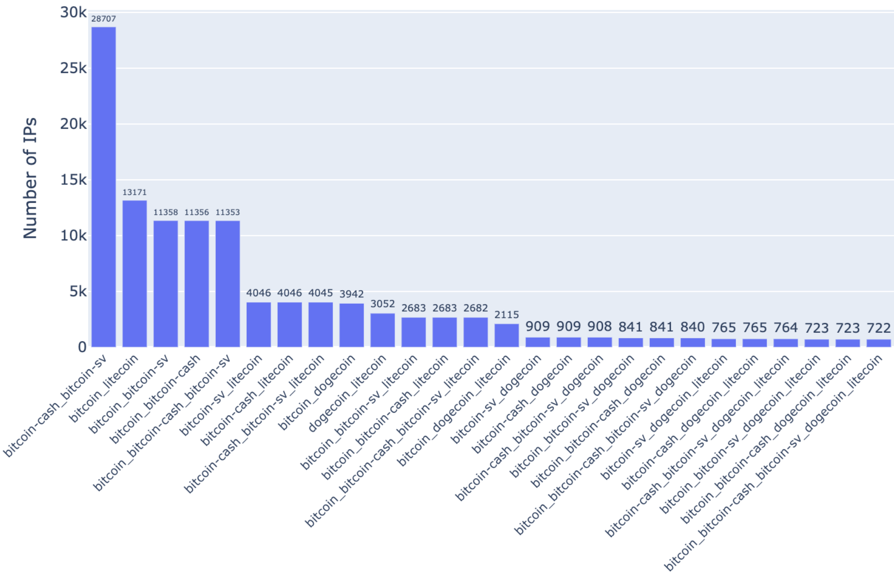


Figure 4.11: Cross-Network IPs among All Nodes in Bitcoin-based Networks

For the Bitcoin-based networks nodes, plots as shown in Figure 4.11 exhibit the counts of IP addresses discovered in multiple cryptocurrency networks over the past 10 days are generated, revealing patterns in multi-network node usage during this period.

Cross-layer IPs in Ethereum Network

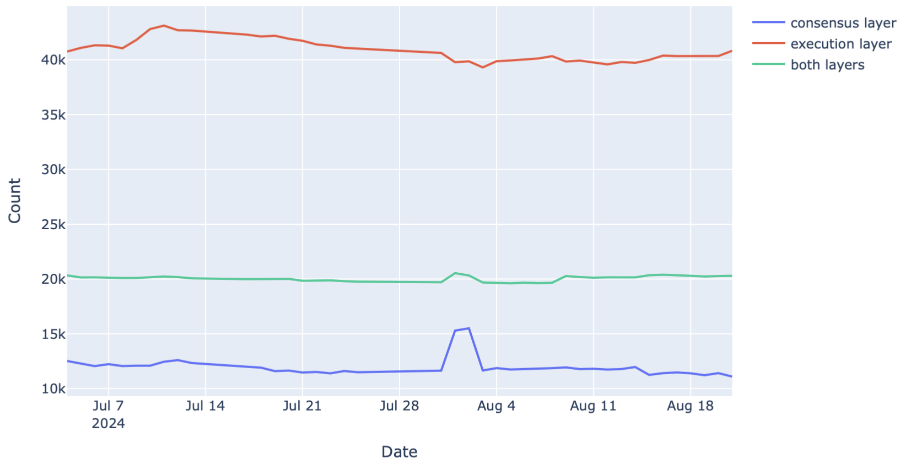


Figure 4.12: Cross-layer IPs of All Nodes in Ethereum Network

For the Ethereum network, the daily overlap of discovered IPs appearing in both the consensus layer and the execution layer are exhibited with line charts as shown in Figure 4.12. As for the meaning of the legend, "consensus layer" and "execution layer" refer to the IPs discovered only in the indicated layers, while "both layer" refer to the IPs appear in these two layers.

4.2.2 ENR Analysis

Distribution of Values Among Nodes for Each Feature in the ENR

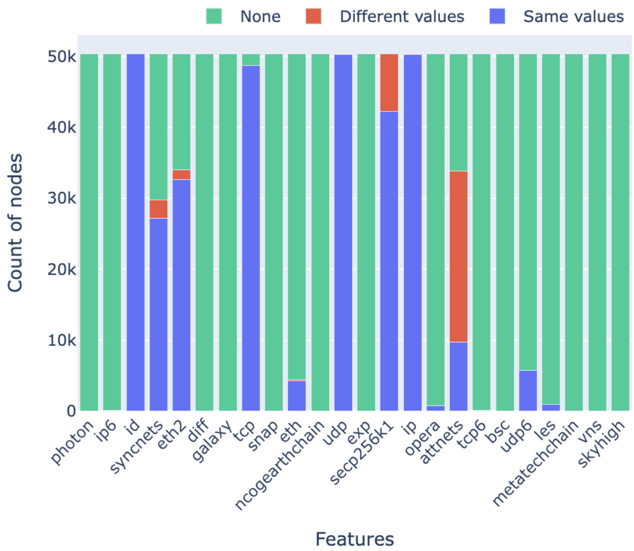
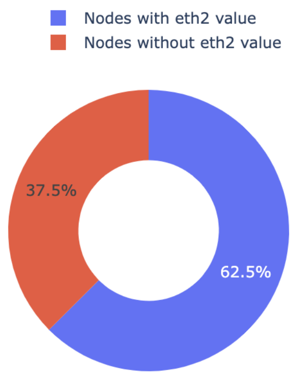


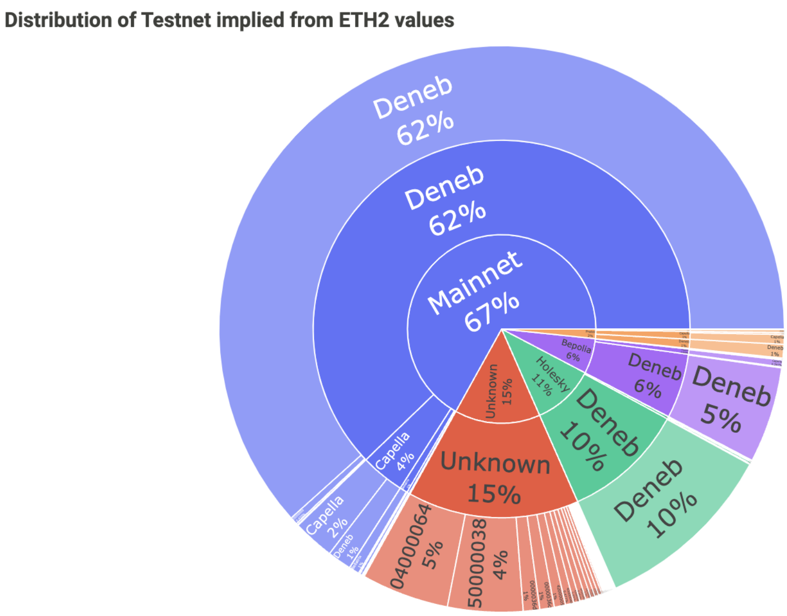
Figure 4.13: Distribution of Values Among All Nodes for Each Feature in the ENR

After decoding the key values across all ENRs from nodes appearing in at least two different crawls among the latest ten crawls, the nodes are categorized based on each key into three groups, as described in Chapter 3. The distribution of values for each ENR feature is displayed in stacked bar charts, as shown in Figure 4.13.

ETH2 Value Distribution



(a) Distribution of Nodes with or without ETH2 values



(b) Mainnet nodes

Figure 4.14: ENR Analysis

The sunburst chart in Figure 4.14 illustrates the distribution of decoded ETH2 values from nodes encountered in the latest ten crawls that have ETH2 values in their ENRs. The inner ring represents the network, the middle ring shows the current fork version, and the outer ring indicates the future fork version.

Bibliography

[1] H.-M. Chang, "Crawling the bitcoin client ecosystem," in *https://pub.tik.ee.ethz.ch/students/2023-HS/SA-2023-25.pdf*, Jan. 2024.

[2] Btcsuite, "btcd: An alternative full node bitcoin implementation written in go (golang)," in *https://github.com/btcsuite/btcd*, 2024.

[3] Ltcsuite, "ltcd/wire: A fork of btcd/wire for litecoin," in *https://github.com/ltcsuite/ltcd/tree/master/wire*, 2024.

[4] blockchair.com, "Blockchair api - fetch data from 17 blockchains," in *https://blockchair.com/api*, 2024.

[5] E. Foundation, "Node discovery protocol version 4," in *https://https://github.com/ethereum/devp2p/blob/master/discv4.md*, 2024.

[6] https://github.com/emhane, "Node discovery protocol version 5," in *https://github.com/ethereum/devp2p/blob/ master/discv5/discv5-theory.md*, 2024.

[7] E. Foundation, "Enode," in *https://ethereum.org/en/developers/docs/networking-layer/network-addresses/*, 2024.

[8] J. Landtwing, "The peer discovery layer of the ethereum network," in *https://pub.tik.ee.ethz.ch/students/2023-HS/SA-2023-14.pdf*, Jan. 2024.

[9] F. Lange, "Eip-778: Ethereum node records (enr),," in *https://eips.ethereum.org/EIPS/eip-778*, 2024.

[10] S. Käser, "Understanding peer-discovery in eth 2.0," in *https://pub.tik.ee.ethz.ch/students/2022-HS/MA-2022-32.pdf*, Aug. 2023.

[11] X. Hong, "multi-cryptocurrency node explorer," in *http://multi-node-explorer.ethz.ch*, 2024.

[12] M. Developer, "Geolite2 geolocation data," in *https://dev.maxmind.com/geoip/geolite2-free-geolocation-data*, 2024.

[13] IPinfo.io, "Ipinfo developer resource," in *https://ipinfo.io*, 2024.

[14] ethernodes.org, "Ethereum mainnet statistics," in *https://ethernodes.org*, 2024.

[15] Plotly, "Plotly library," in *https://github.com/plotly/plotly.py*, 2024.

[16] Dash, "Dash framework," in *https://dash.plotly.com/*, 2024.