universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Integrating Geolocation and AS Affiliation for Scalable Internet Graph Reduction and Visualization

verfasst von | submitted by

Christoph Walser BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, | Vienna, 2025

Studienkennzahl lt. Studienblatt | degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Univ.-Prof. DI Dr.techn. Johanna Ullrich BSc

Danksagung

Ich möchte mich bei allen bedanken, die mich auf meinem Weg zur Fertigstellung dieser Masterarbeit begleitet und unterstützt haben.

Ganz besonders möchte ich meiner Betreuerin, Univ.-Prof. DI Dr.techn. Johanna Ullrich BSc, für die wertvolle fachliche Unterstützung und die konstruktiven Anregungen während der gesamten Forschungs- und Schreibphase danken. Ebenso danke ich meinem Co-Betreuer, DI Markus Maier BSc, der mir durch hilfreiche Diskussionen und technische Unterstützung während dieser Zeit zur Seite stand.

Ein herzliches Dankeschön gilt auch meinen Kommilitoninnen und Kommilitonen für den inspirierenden Austausch sowie die gemeinsame Zeit, die mein Studium bereichert hat.

Contents

1 Introduction **9**

2 Background **11**
2.1 Term Explanations . . . 11
2.2 Literature Review . . . 13
2.2.1 Preparation of the Base Data . . . 14
2.2.2 Modeling the Internet as a Graph . . . 17
2.2.3 Reducing the Complexity . . . 18
2.2.4 Evaluation of the Graph using Metrics . . . 19
2.2.4.1 Topological Structure Metrics . . . 19
2.2.4.2 Centrality Metrics . . . 19
2.2.4.3 Connectivity and Robustness . . . 20
2.2.4.4 Spectral Metrics . . . 20
2.2.4.5 Advanced Metrics for Internet-Specific Analysis . . . 20
2.2.5 Literature Review Synthesis and Positioning . . . 21

3 Methodology **23**
3.1 Data Sourcing . . . 23
3.2 Basic Creation of the Graph . . . 24
3.3 Integration of Metadata . . . 26
3.4 Reduction . . . 27
3.5 Representation as a Graphic Map . . . 30
3.5.1 Exporting Data for Visualization . . . 32
3.5.2 Visualization . . . 34
3.6 Graph Metrics . . . 37
3.6.0.1 Degree Centrality . . . 38
3.6.0.2 Closeness Centrality . . . 38
3.6.0.3 Betweenness Centrality . . . 38
3.6.0.4 Katz Centrality (abandoned) . . . 39

4 Data Analysis **41**
4.1 Evaluation of Performance . . . 41
4.1.1 Assignment of Metadata . . . 41
4.1.2 Reduction Efficiency . . . 42
4.1.3 Runtime Observations . . . 43
4.2 Evaluation of IPv4 Metrics . . . 44
4.3 Evaluation of IPv6 Metrics . . . 48

4.4 Analysis of Significant IPv4 nodes 49
4.4.1 Betweenness Centrality 49
4.4.2 Closeness Centrality 52
4.4.3 Degree Centrality 55
4.5 Analysis of Significant IPv6 nodes 56

5 Discussion **59**
5.1 Performance and Hardware Requirements 59
5.2 Computational Bottlenecks in Metric Calculation 59
5.3 Pipeline Bottlenecks and Parallelization Trade-offs 60
5.4 Metadata Integration and Lookup Performance 60
5.5 Debugging and Testing Challenges 61
5.6 Graph Reduction Side Effects: Island Nodes 61
5.7 Visualization Trade-offs 62
5.8 Limitations of the Underlying Data Structure 62
5.9 Summary and Outlook 63

6 Conclusions and Future work **65**

Zusammenfassung

Das Internet ist ein komplexes, dezentrales Kommunikationssystem. Es besteht aus Tausenden von einzelnen Netzwerken, die verschiedene Einheiten miteinander verbinden. Diese Arbeit befasst sich mit der Herausforderung, wie man die großräumige Struktur des Internets untersuchen kann, indem man Geolokalisierungsdaten und die Zugehörigkeit zu autonomen Systemen (AS) einbezieht, um die Topologiegraphen des Internets zu vereinfachen. In dieser Thesis werden Methoden zur Reduzierung von Internetgraphen durch die Integration von Geolokalisierungs- und AS-Zugehörigkeitsdaten vorgeschlagen. Knoten werden auf der Grundlage gemeinsamer Standorte oder der Zugehörigkeit zu autonomen Systemen (AS) zusammengeführt. Durch Edge Pruning (Kantenreduktion) und clusterorientierte Graph-Einbettungen wird sichergestellt, dass wichtige strukturelle Eigenschaften erhalten bleiben. Die effiziente Berechnung wichtiger struktureller Metriken wie Konnektivität, Zentralität und hierarchische Struktur wird durch diese kombinierte Strategie der räumlichen und organisatorischen Reduzierung unter Beibehaltung wesentlicher topologischer Eigenschaften unterstützt. Die Arbeit umfasst auch eine umfassende Strukturanalyse der kleineren Graphen und zeigt, wie wichtige Internetregionen und große AS-Knoten über geografische Grenzen hinweg verbunden bleiben. Darüber hinaus wird eine grafische Karte der vorverarbeiteten Daten erstellt. Diese liefert neue Einblicke in die räumlichen und organisatorischen Strukturen des Internet-Backbones. Dieser Ansatz ermöglicht eine skalierbare Kartierung des Internets und verbessert unser Verständnis der Beziehungen zwischen Geografie, AS-Struktur und globaler Netzwerkkonnektivität. Die Arbeit baut auf früheren Ansätzen auf und ergänzt sie durch einen flexiblen Rahmen für die Konstruktion, Reduzierung und Untersuchung von Graphen.

Abstract

The internet is a complex, decentralized communication system. It is composed of thousands of individual networks that connect different entities. This work addresses the challenge of how to explore the large-scale structure of the internet by incorporating geolocation data and autonomous system (AS) membership to simplify the topology graphs of the internet. This work proposes methods for reducing internet graphs by integrating geolocation and AS affiliation data. Nodes are merged based on shared locations or autonomous system (AS) membership. Edge pruning and cluster-aware embeddings ensure that key structural properties are preserved. The efficient calculation of important structural metrics such as connectivity, centrality, and hierarchical structure is supported by this combined strategy of spatial and organizational reduction while preserving essential topological properties. The thesis also includes a comprehensive structural analysis of the smaller graphs and shows how important internet regions and major AS nodes remain connected across geographical boundaries. Furthermore, a graphical map of the preprocessed data is created. This provides new insights into the spatial and organizational structures of the internet backbone. This approach allows the internet to be mapped on a large scale and improves our understanding of the relationships between geography, AS structure, and global network connectivity. We build on previous work and supplement it with a flexible framework for graph construction, reduction, and exploration.

1 Introduction

The global internet is an ever-expanding and highly intricate system. It comprises billions of interconnected nodes. These range from individual hosts to complex subnets. This vast infrastructure is organized primarily through relationships between autonomous systems (AS). Each AS is a unique routing domain, managed by independent operators. A distinct segment of the IP address space is overseen by each AS, and local routing strategies are used. Providing important insights into the large-scale structure and operational dynamics of the internet is possible through analysis of its structure at a fundamental level [34, 5, 14].

The enormous size and complexity of the internet mean that graph theory is a powerful tool for modeling its topology. Nodes can be used to represent different levels of granularity, such as autonomous systems or routers, while edges show their connections to each other. However, a significant challenge is posed by the size of these graphs. They are difficult to analyze and visualize effectively and require scalable methods for reducing graphs while preserving essential structural properties. Analysis of internet graphs is, for example, important for network design, maintenance, and security. It helps to detect weak points in the network, optimize routing, and plan expansions or redesigns for better scalability and resilience [80, 76].

One promising approach to creating these scalable representations is to integrate geolocation data and AS affiliation information. Geolocation assigns approximate physical locations to nodes, providing spatial context in diagram layouts, while node affiliation groups nodes based on shared organizational or geographical criteria, enabling meaningful aggregation. Combining these dimensions allows the diagram to be simplified in a domain-specific way. This can include grouping nodes into geographic regions or organizational clusters. The result is a reduction in visual clutter while retaining important connection patterns [27, 62, 79].

This thesis explores novel methods of combining geolocation and autonomous system (AS) affiliation data to produce scalable, reduced representations of the internet graph, which are optimized for visualization, and analysis. Building on existing datasets from sources such as CAIDA and IPinfo, which provide comprehensive mappings of AS relationships alongside geospatial attributes, it employs techniques such as neighborhood pruning and edge pruning to strike a balance between reduction and integrity [9, 36, 41, 75].

Through the integration of these dimensions, we aim to enhance the spatial understanding of the internet's topology, reveal important geographic and organizational structures. Ultimately, the main goal is to enable dynamic exploration and advanced analysis of large-scale

internet graphs while preserving essential connectivity and performance characteristics. By building on previous work, we are completing a framework that can cover all aspects, from scanning to graph construction and reduction to representation.

The remainder of this thesis is structured as follows:

- Chapter 2 surveys related work in geospatial and AS-level network processing, visualization and reduction, including current methods and their limitations.
- Chapter 3 details the methodology for integrating geolocation and AS affiliation in scalable internet graph reduction.
- Chapter 4 contains an in-depth analysis of the obtained data and the efficiency of the used approach.
- Chapter 5 discusses the encountered problems and limitations of the employed strategy.
- Chapter 6 concludes with the results and discusses possible future research directions.

2 Background

2.1 Term Explanations

Notation	Description
G	A Graph
G'	A reduced Graph
V	Set of graph nodes
V'	Reduced set of graph nodes
E	Set of graph edges
E'	Reduced set of graph edges
N	Number of nodes
N'	Reduced number of nodes
ASN	Autonomous system number, assigned unique identification number of autonomous systems
CC2	Abbreviation for country code with 2 digits described in the ISO 3166 international standard.
ASN graph	Reduced graph using autonomous system number metadata
CC2 graph	Reduced graph using country code metadata

Table 2.1: Notations used in this paper.

Internet Protocol with IPv4 and IPv6
IPv4 and IPv6 are two unique versions of the internet Protocol, which is responsible for identifying and forwarding data between hosts on the internet. IPv4, the first and widely used version, has a 32-bit address space and supports around 4.3 billion unique addresses [69, 1]. However, due to the rapid increase in devices connected to the internet, IPv4 addresses are becoming increasingly scarce and will probably finally be exhausted within the next few years. IPv6, which was introduced as the next-generation protocol, addresses this limitation with a 128-bit address space that offers a larger number of unique addresses (approx. $3.4{\times}10^{38}$) [1, 33]. In addition to the address space, IPv6 also offers further improvements in the areas of security, mobility, and autoconfiguration [69, 33]. Although IPv4 and IPv6 are not directly compatible, there are various transition mechanisms that have been developed to enable coexistence during the gradual migration process [29].

Network Graph
Network graphs in the internet context are usually undirected graphs defined as G(V, E). They are a mathematical structure which is built out of a set of vertices (V) and a set of edges (E) each. In the context of internet networks, the vertices represent internet devices, usually routers, and the edges represent the connection paths between these devices [77]. This form of graph structure allows for effective modeling of the topology of internet networks, where the bidirectional nature of connections between routers is captured by the undirected edges [43]. The undirected graph structure follows this form of representation:

- Vertices (V): Each vertex represents a router in the network. The routers are communications devices which forward data between different parts of the entire network [70].
- Edges (E): Each edge represents a communication link between two routers. These links can be physical connections (e.g., fiber optic cables, Ethernet) or logical connections established over physical infrastructure (e.g., VPN over Ethernet) [43].

This graph-based form of modeling internet networks is ideally suited for network administrators and researchers allowing them to analyze various aspects of network topology, such as connectivity, routing efficiency, and potential bottlenecks [23]. It provides a powerful framework for the analysis and optimization of the complex structure of large-scale internet infrastructures [77, 57].

Autonomous Systems (AS)
Autonomous systems (AS) in networks are entities of related internet protocol routing prefixes which are under the control of a single administrative entity. These entities are commonly internet service providers, Tier-1 providers, backbone operators, or fairly large organizations [37]. The autonomous systems represent the basic building blocks of the internet's routing infrastructure and operate independently of each other. They are connected to each other and create the global network structure when combined [72]. Each block is identified by a unique number, which is called Autonomous System Number (ASN). They use the Border Gateway Protocol (BGP) in order to exchange routing information among each other [64]. With the continuous growth of the internet, the Autonomous Systems fulfill an important role in managing routing policies, and maintaining the hierarchical structure of the internet remains critical for scalable and efficient global connectivity [64, 72].

Metadata (Graph)
Metadata is descriptive information related to graph elements (i.e., nodes or relationships). It describes properties of the graph's nodes or edges and provides essential context and structure about the data contained within a graph. The addition of such data helps to improve representation and usability. In our work, we are using metadata in the form of geolocation and autonomous system affiliation related to the nodes in the processed graph.

WHOIS (IP)
IP WHOIS is a protocol which provides information about the ownership and owner details of a single IP address or a block of IP addresses. It enables data to be retrieved from databases belonging to various internet registry operator organizations. Among the provided data is usually the owner or organization which the IP is assigned to, as well as contact details (name, email, phone number) and a physical address. Additionally, WHOIS data can also contain further details about the IP address range, status, and geographical information.

2.2 Literature Review

In this literature review the goal is to methodically examine the existing body of research in this area. The chosen approach involves a structured examination of the relevant literature, following a logical sequence of processing steps as shown in figure 2.1, in order to ensure comprehensive coverage and critical analysis. First, the key topics and concepts in this area are identified in order to create a basis for our investigation. We then categorize the literature according to methodological approaches. This methodological process not only provides a deeper understanding of the topic but also reveals possible research directions. Through this step-by-step analysis, we aim to present a coherent synthesis of the current state of knowledge and make a meaningful contribution to the academic discourse on analyzing the internet as a graph.

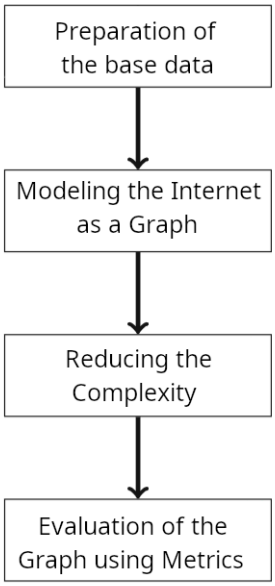


Figure 2.1: Core processing steps of internet graph analysis.

2.2.1 Preparation of the Base Data

When preparing the basic data for the internet graph, there are in general two different approaches possible. The first one is to simply use already prepared datasets, which are provided by several different organizations like CAIDA, iPlane, M-Lab, and RIPE NCC [10].

The CAIDA Archipelago (Ark) network scans use a globally distributed infrastructure of 287 monitors in 66 countries to perform large-scale internet topology measurements. Using the Scamper tool, Ark monitors perform daily traceroute scans to each routed IPv4 /24 prefix and IPv6 prefix and collect path data via internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), or User Datagram Protocol (UDP) probes. These scans primarily provide macroscopic insights into the internet infrastructure and reveal interdomain links, AS relationships, and routing behavior. The IPv4 dataset targets $\sim$10 million /24 networks with 100 packets per second (pps), while IPv6 scans cover all announced prefixes every 48 hours. The output data includes hop IPs, RTTs, and ICMP responses, with statistical outputs providing AS-level topology mapping and alias resolution. Ark's architecture supports custom measurement applications via Python-based NiteOwl primitives, allowing researchers to develop tools for performance analysis (e.g., latency, loss) or DNS/HTTPS queries [6, 7, 8].

iPlane network scans use a distributed infrastructure of PlanetLab nodes and public traceroute servers to systematically map the topology of the internet and predict path performance. Using traceroute and ICMP probes, iPlane creates an ”atlas” of Autonomous Systems (AS) and Border Gateway Protocol (BGP) atoms - prefixes that share identical AS paths to identify routing policies and hierarchical clusters of interfaces within Points-of-Presence (PoPs) [42]. This clustering reduces the measurement load by grouping geographically or functionally related nodes, which enables efficient path partitioning for performance predictions (e.g., latency,
bandwidth) [42]. Probes are strategically assigned via a breadth-first search (BFS) algorithm to the vantage points closest to the target links to balance the load while capturing dynamic attributes such as loss rates and capacity [42]. The system minimizes intrusiveness by coordinating low-intensity scans (2-5 Kbps per node) and leveraging opportunistic data from peer-to-peer swarms to infer edge network characteristics [42]. By integrating topology mapping with performance metrics, iPlane supports applications such as informed peer selection in overlays, which show improved download times compared to random or coordinate-based approaches [42].

M-Lab performs customer-initiated network scans across a global infrastructure of 750 servers in over 70 major cities, hosted at ISP connection points and cloud networks [60, 50]. These scans primarily perform server-side measurements for tests such as the Network Diagnostic Tool (NDT), which measures TCP transport capacity (RFC 3148) by measuring download and upload speeds over WebSocket via TLS [24, 60]. The platform captures TCP metadata, including packet headers, latency, and traceroute data from scamper probes, while enforcing open-source software and data policies [24, 65].

Unlike active probing systems, M-Lab focuses on a passive approach by relying on user-driven diagnostics, with measurements triggered by client connections, providing insights into real-world performance issues (e.g, ISP bottlenecks, regional outages) [24, 65]. The generated data is regularly archived in Google BigQuery in order to support research on net neutrality, censorship, and infrastructure reliability [60, 51]. The platform uses Kubernetes-managed servers and Docker containers in order to ensure the isolation of resources, while partnerships with transit providers improve the transparency of inter-domain routing [24, 50].

The RIPE Network Coordination Centre (NCC) performs network scans and measurements using tools such as RIPE Atlas, a global platform with thousands of probes that perform traceroute scans, DNS queries, and latency tests to monitor internet connectivity and routing in real time [67]. These types of scans provide insights into IPv6 adoption, BGP dynamics, and Internet Exchange Point traffic paths, with data archived live in services such as RIPEstat and Routing Information Service to identify AS relationships, routing anomalies, and DNS performance [67, 20].

The measurement.network project, which is supported by the RIPE NCC Community Fund, enables research-scale scans such as analyzing traffic in dark networks and resolving the entire DNS tree, while mitigating the risk of unintended impact on the network [20, 21]. In addition, the RIPE database documents routing policies and IP resource allocations and enables scientific studies on topology and operational trends [68]. Security tests, such as penetration tests from Radically Open Security, validate the integrity of these systems and ensure encrypted communication and secure data processing during the scans [66].

Table 2.2: Comparison of internet topology datasets: CAIDA, iPlane, M-Lab, RIPE NCC

Dataset	Data Source	Topology Level	Coverage	Data Format	Update Frequency
CAIDA ITDK	Traceroutes via Archipelago (Ark) & RIPE Atlas	Router-level	Global IPv4 & IPv6 Internet topology	Nodes and Links files, AS assignment, Geo locations, DNS names	Updated every 6-12 months
iPlane	Active probing, traceroutes	Router & IP-level	Focus on Internet core	Graph data with router and IP level	Periodic, less frequent updates
M-Lab	Network measurement tests and active probes	IP-level	Large-scale internet measurements globally	Various raw measurement files and aggregated stats	Ongoing continuous data collection
RIPE NCC Atlas	Active probes (traceroute, ping, DNS)	IP-level & AS-level	Global Internet with a focus on latency and reachability	JSON, CSV, and TSV formats via API	Continuous real-time measurements

The second possible approach is to create a new dataset by performing individual traceroute based probing which brings the additional advantage of being able to use an individualized scan setup which, for example, enables the granularity of the network area scans to be fine-tuned [44]. Yelling at Random Routers Progressively (Yarrp) is a fast, stateless network scanning tool designed for active topology discovery on the internet. In contrast to

traditional traceroute methods, it uses randomized probing over IP addresses and TTLs to quickly map router interfaces while minimizing network load [3]. The key features are:

- Its stateless operation reconstitutes path data from asynchronous ICMP responses, enabling probing rates of 100k packets per second (e.g. mapping all IPv4 /24 prefixes in 1 hour from a single vantage point) [3].
- Randomized Probing: Allows IP x TTL space to spread the load and avoid triggering rate limits or alarms [3].
- Allows the export of scanned data in various standardized formats (e.g., warts) for compatibility with analytics tools such as CAIDA's Ark platform [3].
- It can capture topology snapshots for the investigation of transient network dynamics, IPv6 mappings, and cross-protocol comparisons [3].

Yarrp is open source and optimized for research purposes which require large-scale internet scanning.

ZMap is a high-speed stateless network scanner designed for internet-wide surveys, capable of scanning the entire IPv4 address space on a single port in less than 45 minutes from a single machine with a gigabit connection [16]. Originally developed by researchers at the University of Michigan, it uses a modular architecture and pseudo-random IP probing to minimize network load in order to achieve near-theoretical top speeds [16].

- Its stateless operation uses cyclic multiplicative groups in order to generate probes without connection state, which enables a theoretical usage of 97% of the maximum possible Gigabit Ethernet throughput [16].
- Modular design: Supports TCP SYN, ICMP, DNS, and UDP probes, with extensible output handlers for integration with tools such as ZGrab (TLS/HTTP handshakes) [17].
- The used application design allows for scanning speeds of up to 1 Gigabit per second under optimal conditions. This property makes it an invaluable tool for network measurements and security research [17].

ZMap is open source, cross-platform and emphasizes "good internet citizenship" through randomized probing, rate limiting, and avoidance of sensitive targets (e.g., RFC 1918 addresses) [16]. While its speed comes at the expense of retransmissions ($\sim$2% of hosts are missed), its scalability makes it essential for large-scale network analysis. Although it only sends one sample per host by default, it achieves $\sim$98% coverage, with minimal impact on security. ZMap is a cornerstone of network security research and is cited in over 500 academic papers. It enables large-scale internet studies in real time [17]. The application's design balances speed, efficiency, and ethical scanning practices, making it indispensable for both offensive and defensive security analysis [16].

2.2.2 Modeling the Internet as a Graph

The internet can be represented as a graph on various levels of granularity. The focus of the representation can in general be the Autonomous System (AS) level [78, 57, 23, 31], which is the roughest possible granularity. The medium level is the router-based representation [44, 13]. The finest level is the end-user-centric internet device-level granularity [45]. Network graphs are commonly represented as a set of vertices and, where vertices can be AS [78], router-level devices [44, 10] or end-user devices [45], and the edges represent the relationships between them.

Processing traceroute-based scans into a graph involves several key steps. First, traceroute tools are run from one or multiple source nodes to a target destination, collecting the sequence of intermediate IP addresses (hops) traversed by each probe. Each occurring unique IP address is then mapped as a node (vertex) in the graph, while each consecutive pair of hops in the traceroute output forms a directed edge, representing the path taken by packets through the network [73, 28]. When multiple traceroute scans are performed from different sources or at different times, their paths are merged, resulting in a comprehensive network topology graph that reveals alternative routes and dynamic changes in routing [73, 28].

For further processing opportunities, it is also possible to additionally add various kinds of metadata to the graph. Geolocation data is a commonly used solution for supporting the analysis of network graph data [57, 58, 13, 45]. In [57] the authors investigate the impact of geographical distance on network performance metrics and understand how intra-AS routing decisions relate to physical locations within the internet infrastructure. [58] focuses on developing a method to create a Point of Presence (PoP) level topology map of the internet, which captures the backbone structure and interconnections of Autonomous Systems (AS) with high accuracy by using multiple geolocation sources. In [13] a novel IP geolocation technique based on combining propagating IP location information through traceroute scans with IP interpolation is presented, which aims to resolve incomplete IP database coverage. In [45] a device graph is built using geographic information in the form of IP-colocation inferred through shared IP addresses or physical proximity.

Furthermore, various aspects of additional ASN information (i.e., which ASN a router or device belongs to) often play a crucial role in augmenting the analysis of network graph data, like for example, the hierarchy of ASN [78, 56, 58]. In [56] an enhanced method for constructing the AS-level topology of the internet by integrating data from multiple sources, including RouteViews, RIPE RIS, route servers, and routing registries is developed. [58] utilizes autonomous system information to organize and cluster IP addresses into meaningful PoP nodes and to provide a detailed understanding of the internet's backbone structure beyond traditional AS-level views. In [78] the authors aim to improve AS topology graphs by accumulating topological information from routing updates over time by gathering data from various sources, including BGP routing tables, route servers, looking glasses, and routing registries. Regardless of the chosen data source and representation form, reading information from the resulting internet graph is a challenge due to the sheer size of the IPv4 and

IPv6 network address ranges.

2.2.3 Reducing the Complexity

With the continuous evolution of the internet [23], the constantly growing size and complexity of graph datasets pose tremendous challenges in computation and analysis [30]. Graph Reduction is an operation that compresses large graph datasets without losing their essential properties and principal information. As indicated in figure 2.2, graph reduction is an operation that refers to the process of reducing a large graph dataset to a smaller one and making sure that critical information is maintained. The reduction is achieved using algorithms that work on the original graph dataset to generate a smaller version which retains critical features [30]. We now provide a brief introduction to the key techniques involved.

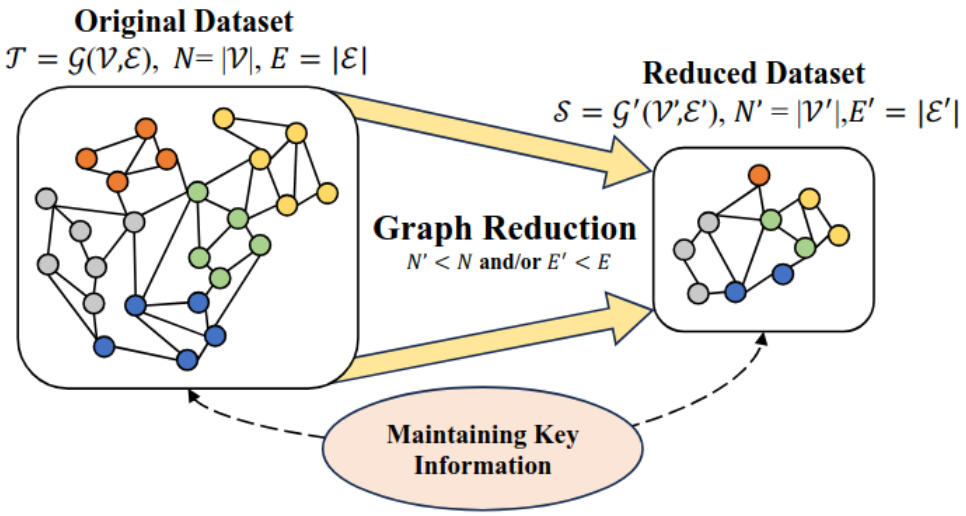


Figure 2.2: The goal of graph reduction is to identify a reduced (smaller) graph dataset that retains certain information from the original graph dataset. Source: [30]

Graph sparsification is a technique where significant edges or nodes are chosen based on some criteria to create a smaller graph. It is concerned with maintaining certain graph properties like spectrum and centrality [30, 11]. Graph coarsening is a process that shrinks nodes and edges to get a smaller representation of the graph. This can assist in reducing complexity without losing the overall structure [30, 39].

Condensation of the graph is used to simplify the graph by combining nodes or edges, which can be used to simplify the graph without losing vital information [30, 18]. Edge pruning is another highly effective method for the analysis of the controllability and observability of complex systems [52]. If properly applied, this is a powerful technique of network simplification with relatively low loss of connectivity [81]. Clustering and hierarchical aggregation of the network can be combined in order to comprehend complex connectivity patterns. This is done by grouping nodes having similar properties, which facilitates comprehension

and control of the network. It is an effective method of decomposing complex networks into controllable units, such that related items are linked and coordinated effectively [19]. Graph reduction usually results in various benefits. It makes graph algorithms scalable, fast, and efficient. Additionally, reduced graphs tend to be easier to visualize and understand, which simplifies human perception. Another advantage lies in privacy preservation because the transformation of the original structure or node attributes complicates the recovery of sensitive information. [30].

2.2.4 Evaluation of the Graph using Metrics

For the analysis of the graph properties, a variety of possible network-related graph metrics can be used. These commonly belong to the categories of node adjacency, clustering, connectivity, distance, throughput, or spectral (i.e., hybrid) metrics [59]. The exact variation of the available metrics categories to use requires careful analysis based on the given requirements, the accuracy-cost trade-off, and the limitations of the considered metrics [59]. Common focus topics for research in this area are:

- Topological characteristics of the internet graph [23, 31, 78]
- Connectivity patterns at the Autonomous System (AS) level, along with the discernment of core and peripheral AS [56, 23]
- The geographical distribution of internet infrastructure and its consequential effects on network performance [57, 58]

2.2.4.1 Topological Structure Metrics

Among the topological structure metrics, the average path length, which is the mean number of hops between node pairs, is crucial for the evaluation of routing efficiency [40]. The network diameter metric is defined as the longest shortest path in the graph, i.e., it quantifies the maximum number of edges (or ”hops”) required to connect the two farthest nodes and serves to identify the worst-case communication latency [40]. Another important measure is the node degree with its variation in/out-degree for directed graphs. It is the number of links per node and can be used for identifying hubs (e.g., core routers) and leaf nodes [32].

2.2.4.2 Centrality Metrics

The betweenness centrality measures how often a node lies on the shortest path between pairs of nodes. It is important for identifying critical nodes (i.e., Bottlenecks) in networks [26]. Closeness centrality expresses how central a node is in a network by measuring its average shortest-path distance to all other nodes. Nodes with high closeness centrality can efficiently interact with or spread information across the network due to their proximity to others [59, 32, 26]. The eigenvector centrality measures the influence of a node in a network both through its direct connections and the importance of its neighbors (e.g., Tier-1

Autonomous Systems) [59, 26]. Katz centrality is a measure used in network analysis to determine the relative importance of a node within a graph. It is particularly useful for identifying influential nodes in various types of networks, such as social networks, biological networks, and information networks. It quantifies the influence of a node by considering not only its direct connections but also the connections of its neighbors and assigns a score to each node based on the number of paths leading to it, weighted by the length of those paths [26].

2.2.4.3 Connectivity and Robustness

One of the key metrics within this category is the algebraic connectivity, which reflects the overall network resilience. It is a global measure defined as the second-smallest eigenvalue (λ_2) of its Laplacian matrix L. It describes the global connectivity of the graph, with higher values indicating greater robustness against partitioning (i.e., disconnection) [32, 40, 59]. The vertex/edge connectivity is the minimum number of vertices or edges whose removal disconnects the graph. As such, it quantifies the fault tolerance of a network [40, 32]. With the clustering coefficient, we have another important metric in this category that quantifies fault tolerance. It has both a local version that measures the density of connections among a node's neighbors, indicating how close the neighborhood is to forming a complete graph (clique), and a global version, which is usually the average local coefficient [32, 59].

2.2.4.4 Spectral Metrics

For the spectral metrics category, the graph spectrum reveals global network properties like expansion and congestion. It is the set of eigenvalues of matrices associated with a graph, typically the adjacency matrix or Laplacian matrix [32, 25]. Effective resistance is a measure that quantifies the resistance of a network to disconnection and failure. It is derived from the Laplacian matrix of the network, which captures the connectivity between nodes [59, 74].

2.2.4.5 Advanced Metrics for Internet-Specific Analysis

Modularity is a significant metric used in network analysis, particularly for identifying community structures within large graphs. Modularity measures the quality of a division of a network into communities. It quantifies how well a network can be divided into groups where connections are denser within groups than between them [59, 54]. Ollivier-Ricci curvature is a concept that extends the classical notion of Ricci curvature to general metric spaces using optimal transportation theory. Ollivier-Ricci curvature is defined for metric spaces equipped with a probability measure. It quantifies how much the geometry of a space deviates from being flat. The Ollivier-Ricci curvature can effectively detect community structures in networks and is used for detecting network backbone and congestion [55].

2.2.5 Literature Review Synthesis and Positioning

A strong foundation of resources, datasets, and techniques for measuring internet topology and modeling graphs is provided by the studied literature. Numerous methods for gathering data at various scales and network layers are offered by measurement platforms such as CAIDA, iPlane, M-Lab, RIPE Atlas, Yarrp, and ZMap. Though they differ in terms of granularity, focus, and data freshness, these tools allow for broad internet mapping.

As a result, modeling internet topology can expose its structural complexity at several levels, ranging from autonomous system graphs to device-level representations. Although problems with data consistency and completeness still exist, this complexity can be resolved by integrating metadata like geolocation or AS affiliation. Although the majority of current algorithms prioritize topological properties without closely integrating domain-specific geographic and organizational information, graph reduction techniques such as sparsification, coarsening, edge trimming, and clustering offer promising ways to address this complexity.

Additionally, a variety of graph aspects, from sophisticated spectral characteristics to connectedness and centrality, are captured by commonly used assessment metrics. Nevertheless, there is still a limited amount of research on the use of these metrics in internet-specific situations, especially when it comes to combined geographical and organizational graph reduction.

By creating a scalable, integrated internet graph reduction methodology that works in tandem with high-quality geolocation data and AS ownership information obtained from CAIDA traceroute scans enhanced with IPinfo and WHOIS data, this thesis fills these gaps. This approach goes beyond current general-purpose reduction techniques by facilitating dynamic aggregation and structural preservation based on spatial and organizational domain knowledge. It offers fresh perspectives on how geography, AS structure, and internet connectivity interact, which makes it easier to visualize and analyze large-scale internet graphs with greater clarity and precision.

3 Methodology

The methodology section describes the process of systematically merging geolocation data and autonomous systems (AS) affiliation data in order to make it possible to reduce data and scale visualizations of the internet graphs. The design of the techniques has two primary goals in mind. They apply to large network topologies and improve interpretability by incorporating spatial and organizational information. This section explains the reasons for the choice of methods and describes the processes for data collection, graph processing, reduction algorithms, and visualization strategies. The methods described in this chapter in their entirety establish a reproducible framework which directly supports the research goals to improve scalability and insights in the visualization of internet graphs.

In this research, the AI-powered tool Perplexity [61] was utilized to assist in identifying relevant sources and for partial text refinement. All factual information and references were independently verified and cited from their original academic sources.

3.1 Data Sourcing

In this work we are using a dataset created through scans using the traceroute scanning tool Yarrp, which is specifically optimized for high-performance scanning of broad network ranges while preventing network congestion or overloading [3]. The scanning setup and parameters were defined in the thesis by Maier, which focused on investigating router misconfigurations in the IPv4/IPv6 Internet [44]. The machine that performs these scans is located in Vienna and has been regularly performing new traceroute scans of the IPv4 and IPv6 internet since then. A side effect of this is that the collected data reflects the global internet structure from the perspective of Vienna. We are utilizing a scan of the internet landscape that was produced in March 2024. This is due to the fact that this was the latest scan available when work began, and we have suitable ASN and geolocation sources available for this point in time.

The scanning parameters are set up so that for IPv4 scans, one random traceroute per /24 network prefix is collected. For IPv6, one random traceroute per /48 network prefix is collected. For IPv4 a full scan is finished within a single day and produces about 9 GB of raw scanning data, capturing 12.57 million nodes. A scan of IPv6 takes about three weeks and produces roughly 2.5 TB of raw scanning data capturing 5.7 billion nodes. The amount of data produced by the IPv6 scan poses a challenge for further processing and is discussed in detail in chapter 5. The necessary efficiency in performance is also the primary reason why

the previous work was implemented as Rust code instead of Python [22]. We have therefore continued to use Rust for extending and implementing the core tasks within the graph.

3.2 Basic Creation of the Graph

The basic process of creating a graph, on which we build, was developed in the thesis by Fodor, where an analysis of the internet as a graph was conducted [22]. Because we are reusing and further extending the same basic concepts, we will now briefly discuss the process of graph creation.

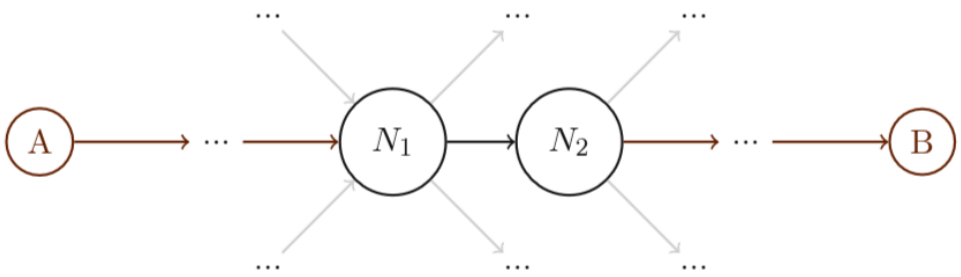
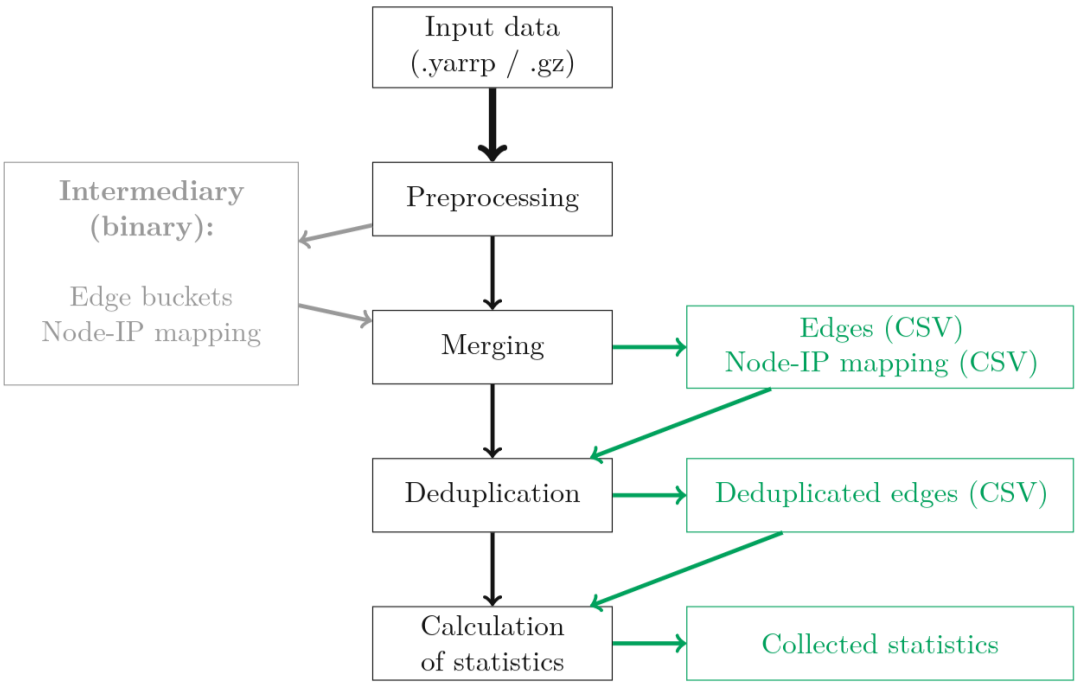
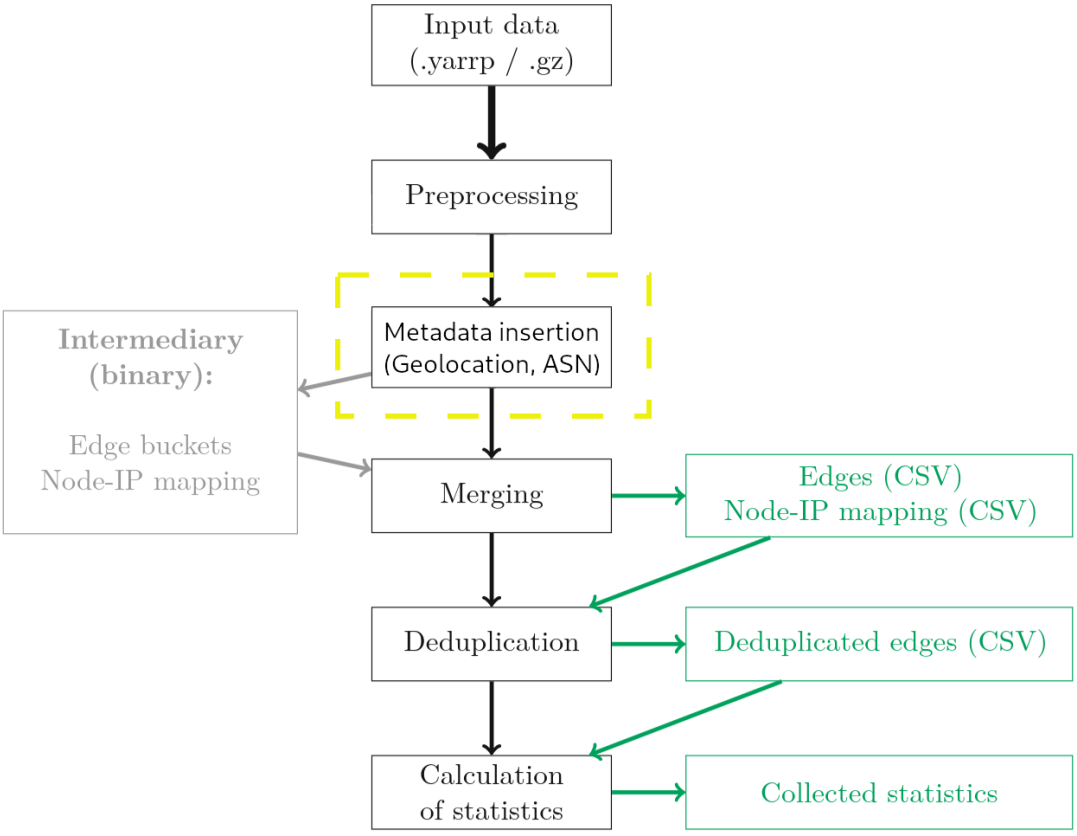


Figure 3.1: Visual representation of merged traceroute paths through the graph. Source: [22]

The Yarrp tool generates several separate files as scan output. These contain individual, ungrouped traceroute scans in pseudo-random scan order [3]. As a consequence, these individual traceroute datasets need to be merged together into full paths during the creation of the graph as depicted in figure 3.1. This process happens in the preprocessing/merging step as shown in figure 3.2a, during which each IP is assigned a unique 128-bit integer and then stored in 256 temporary buckets, including the hop IP, hop TTL, and target IP for each dataset. We need to consider a special case with regard to the hop IP. Unfortunately, not every node responds to the traceroute request, which means we occasionally have gaps in the paths. We know that these gaps exist as stations along the path taken, but we have no further data, such as the IP address. This is solved in the graph by inserting pseudo IDs with a negative sign as placeholders [22]. This approach was originally taken to preserve the original structure of the scans. Since we cannot connect any metadata for these nodes due to the missing IP address, we will not consider them further when reducing the graph. For IPv4 datasets this is done by choosing the target bucket by XOR of the second and fourth byte. For IPv6, this is done through XOR of the last byte from the public segment with the last byte of the private segment [22]. This preprocessing step using buckets helps to reduce the amount of RAM required. After each input file has been processed, the corresponding buckets are stored on the disk. During the merging phase, all buckets are read back in, but they are now pre-sorted and can be processed bucket by bucket to form complete traceroute paths [22].



(a) Original graph creation processing pipeline.



(b) Modified graph processing pipeline with additional metadata insertion (new section marked with dotted box).

Figure 3.2: Comparison of the two graph processing pipelines. Source: [22]

3.3 Integration of Metadata

In order to link the IPv4 and IPv6 nodes with geolocation and ASN metadata, we needed a suitable source for each to be able to match them. For ASN data we are using the Routeviews Prefix to AS mapping Dataset (pfx2as) for IPv4 and IPv6, which is provided by CAIDA [9]. The provided files have the following structure:

- IP prefix
- Prefix length
- AS number

The process of matching the IP nodes of our graph with the corresponding AS number itself is straightforward. Essentially, for each node in our graph, a lookup is performed using the associated IP address. The data provided by the file is then used to identify the matching IP prefix range, from which the AS number is determined.

Worth mentioning though is that early tests, especially with IPv6, have shown that due to the large number of lookup requests, an efficient implementation of the search logic is crucial for performance reasons. After several experimental tests, a binary search algorithm is finally used. It requires the search data to be presorted, which is a small price to pay for the performance gain. The algorithm works by halving the possible result space with each step of the search and has an average runtime of $O(log\, n)$ [53].

For matching the nodes with a geolocation (latitude, longitude), we are using an IP to geolocation database provided by IPinfo as MMDB database format [36]. The database contains the following information:

Field Name	Example	Data Type	Description
network	71.50.174.48/28	TEXT	CIDR or single address of the IP block
city	Spring Lake	TEXT	City of the IP address
region	North Carolina	TEXT	Region/State of the IP address
region_code	NC	TEXT	Region code in two-letter format (ISO 3166)
country	United States	TEXT	Name of the country of the IP address
country_code	US	TEXT	ISO 3166 country code of the IP address
continent	North America	TEXT	Name of the continent
continent_code	NA	TEXT	Continent name code in two-letter format
latitude	35.16794	FLOAT	Latitude value of the IP address
longitude	-78.97281	FLOAT	Longitude value of the IP address
timezone	America/New_York	TEXT	Local timezone of the IP address location
postal_code	28390	TEXT	Postal code or zip code of the IP address

Table 3.1: Overview and description of the fields contained within the IPinfo database.

The geolocation data is assigned through matching the node IP with the contained network range. The lookup procedure itself is provided through the database driver and turned out to be sufficiently performant for our purposes during initial tests. We actually read the following fields from the information provided for our processing stages:

- city
- region
- country_code
- postal_code
- latitude
- longitude

The metadata is inserted into the graph between the preprocessing and merging phases. The new pipeline with the additional processing step is now shown in figure 3.2b. Basically, the metadata is inserted here and then passed through the merging and deduplication phase. In the reduction phase described in section 3.4, the additional data is finally actively used to reduce the size of the graph. The included ISO 3166 country code with 2 digits (alpha-2) is used as the key for this reduction. For simplicity, we will refer to this dataset as ”CC2” from now on.

3.4 Reduction

In this section we will take a look at the core graph reduction process using the previously integrated ASN and geolocation metadata itself. This additional new stage is located at the end of the processing pipeline 3.3. In this step, the nodes and edges of the graph are filtered and completely rewritten. The approach which we are using here can essentially be classified as a form of graph coarsening or condensation. To be specific, we apply node-centric pruning based on local structure equivalence or redundancy. The criteria used for the two types of graphs we handle are ASN and geolocation (i.e., the included country code).

For autonomous systems, we check each node for its neighboring nodes. Whenever a node has neighbors that are all assigned to exactly the same autonomous system number, the node in question and the associated edges are removed from the graph. A formal representation of this process can be seen in figure 3.4. In this example, based on the process described, two nodes with ASN 10, 56, and 31 as well as the associated edges are removed.

We take essentially the same approach with country codes. For each node, we again compare the country code with the country codes of its neighbors. If all neighbors have the same country code, the node and its edges are removed from the graph. As shown in the example depicted in figure 3.5 two nodes with country code ”AT”, ”DE”, and ”CH” as well as the associated edges are removed.

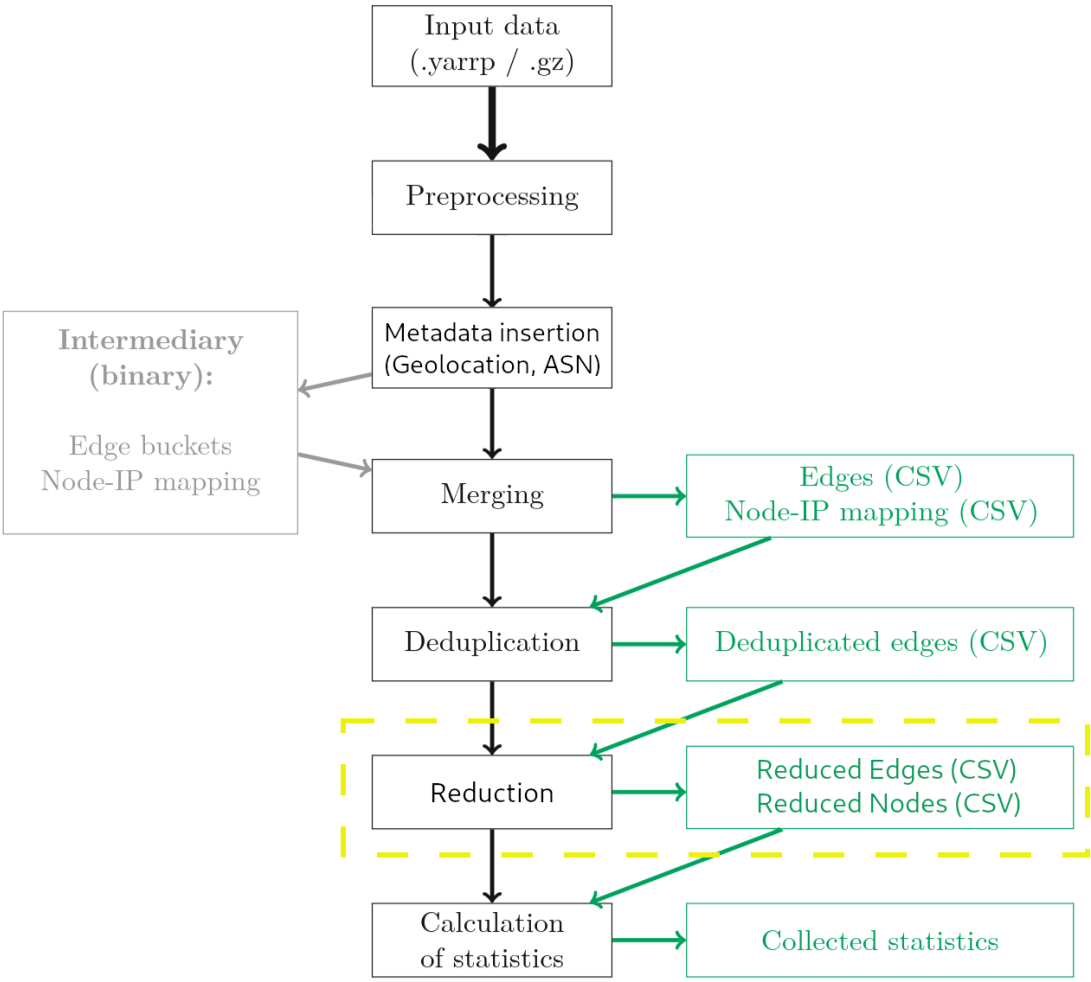


Figure 3.3: Modified graph processing pipeline with additional reduction step (new section marked with dotted box). Source: [22]

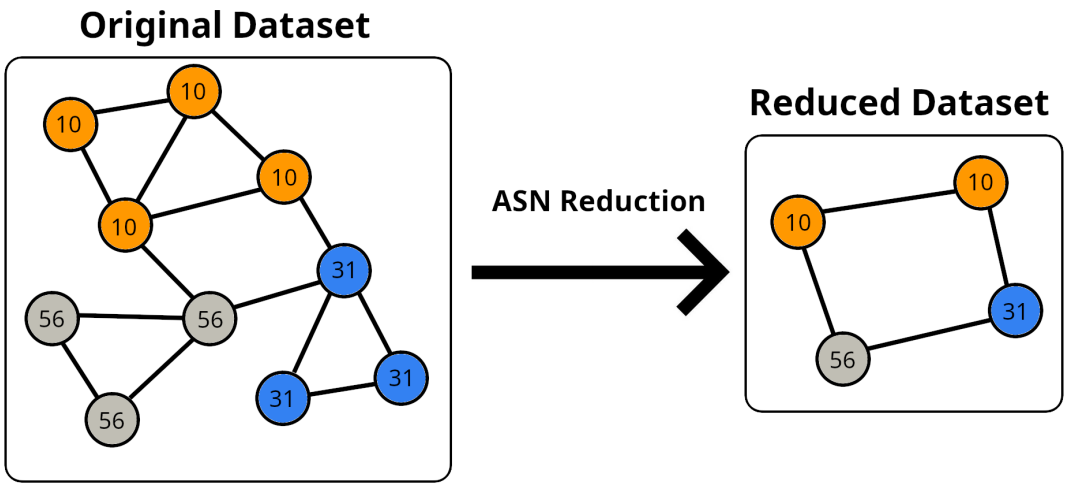


Figure 3.4: ASN reduction process, nodes that only have neighbors with identical AS numbers and the corresponding edges are removed.

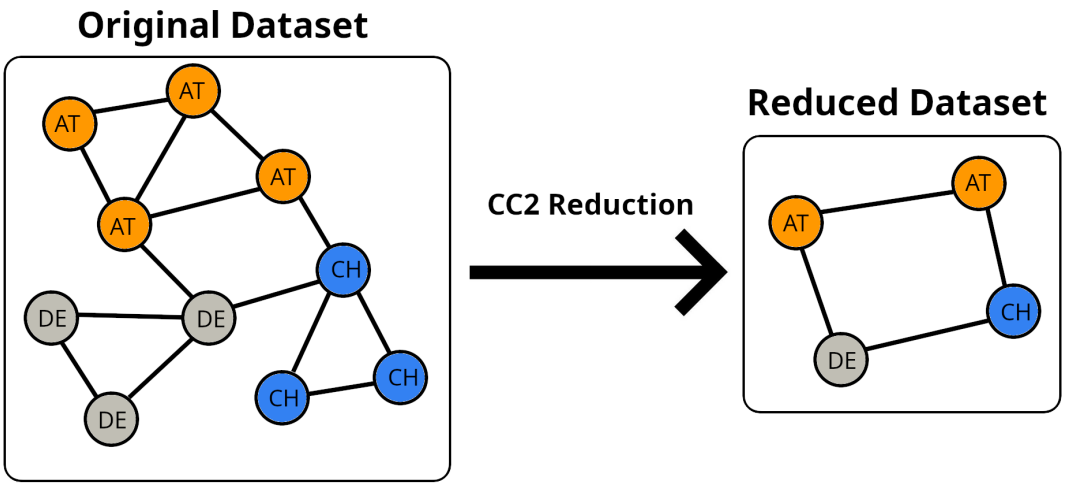


Figure 3.5: CC2 reduction process, nodes that only have neighbors with identical country codes and the corresponding edges are removed.

3.5 Representation as a Graphic Map

This chapter describes the process of creating a visual map from the reduced raw graph data of the network. At the beginning, we will start with a brief overview of the technology stack which is used for the implementation:

Table 3.2: Overview of the technology stack used for the creation of the graphic map.

Technology	Description
OpenStreetMap (OSM)	The OpenStreetMap project provides a free, editable map of the entire world, which is created and updated by volunteers all over the world [12].
OpenMapTiles	OpenMapTiles by MapTiler is an open-source project that provides tools, schemas, and ready-to-use packages for generating and using vector tiles from open geospatial data, primarily OpenStreetMap [71].
GeoJSON	A format for encoding geographic data structures in JSON. Widely used for representing spatial features and geometries [35].
Vector tiles	Vector tiles are packets of geographic data divided into predefined, roughly square-shaped tiles for efficient transfer over the web. Unlike traditional raster tiles, which are pre-rendered map images, vector tiles contain raw vector data (like points, lines, and polygons) clipped to the tile boundaries. This allows client applications to render maps dynamically, enabling greater flexibility such as customizable styles, smooth zooming, rotation, and interactive features. Vector tiles reduce data transfer volume compared to raster tiles and shift rendering load to the client side, improving performance and enabling high-quality, scalable maps [71].
Docker	Docker is a platform that packages applications and their dependencies into portable containers, ensuring they run consistently across different environments. It isolates software without the overhead of virtual machines, simplifying development, deployment, and scaling [15].
Tileserver GL	Vector tile server for hosting vector tiles and providing map styles and tiles over HTTP [48].
MapLibre GL JS	Web mapping library for rendering interactive vector tile maps in browsers using WebGL. A community-driven fork of Mapbox GL JS (v1) [47].

Continued on next page

Table 3.2 – *Continued from previous page*

Technology	**Description**
tippecanoe	Command-line tool by Mapbox used to generate optimized vector tile sets from large-scale GeoJSON or other geospatial data [46].
MaxMind Database (MMDB)	The MaxMind DB (MMDB) format is a compact, highly optimized binary database format designed for mapping IPv4 and IPv6 addresses to associated data records efficiently, often used in IP geolocation databases. It uses a search tree structure to quickly find the data record that corresponds to an IP or network range. The database includes metadata describing the record structure and supports localization of data fields. The format is flexible, allowing custom data schemas in each record, and is optimized for fast IP lookup performance in production environments [49].
Rust programming language	Rust is a modern system programming language focused on safety, speed, and concurrency. The integrated ownership system ensures memory safety without a garbage collector, preventing bugs like data races. It provides powerful abstractions and encourages writing reliable, efficient software suitable for a wide range of applications [38].
Python programming language	Python is a type of computer programming language. It was created by the Python Software Foundation, and is known for being a high-level language. Python code is easy to read and simple, so developers can express ideas in fewer lines of code than in many other languages. It supports multiple programming styles, including procedural, object-oriented, and functional programming. Python provides a large standard library and a wide range of third-party packages. As a result, it is frequently used in the fields of web development, data analysis, automation, scientific computing and artificial intelligence. [63].

The backend and frontend parts of the map server are started by using Docker. This applies to Tileserver GL and MapLibre GL JS as listed in table 3.2. Docker was chosen because it enables simple, efficient, and platform-independent execution of the map server. The map material is sourced from OpenStreetMaps, as this is a widely used and well-maintained source of freely available map material.

3.5.1 Exporting Data for Visualization

This section deals with exporting the relevant data from the graph for display as a geographical map. For the export of the data, we are using the GeoJSON format as listed in table 3.3. It shows a full example dataset with the structure we use when exporting data, each of which contains a node and an edge:

Below is a brief explanation of the structure contained within the file. ”FeatureCollection” is the main node, which contains all node and edge datasets as ”Feature” data blocks. A node corresponds to a ”Feature” of type “Point” and the associated ”properties” section contains its properties. An edge corresponds to a ”Feature” of type “LineString”, where the coordinates specify the position and length of the edge on the map. The associated ”properties” section then specifies which two nodes are connected by it.

The code that converts the graph data into a map overlay is not part of the core framework and is written as a separate toolkit in Python. At the beginning, we read in the components of the graph. The relevant components for this section are as follows:

- Reduced nodes (CSV)
- Reduced edges (CSV)
- Geolocation data (CSV)
- Metrics data (CSV, optional)

First, we read in the respective CSV files and merge them in sequence. A special feature we have incorporated here is the optional inclusion of metrics. Building on the static reduction of data in the graph as described in 3.4, we supplement this section by introducing a dynamic option to reduce the amount of data during the construction of the map. We have the degree, betweenness, and closeness centrality to choose from in the implemented scope.

In order to dynamically adjust the granularity of the data displayed to the situation-specific requirements and the performance of the map server, we have also integrated degree centrality and have decided to filter out all nodes with a value less than 5 by default, as these are not particularly significant in the global context. However, the metric used can be easily replaced, allowing the resulting representation to express different emphases.

The data exported for display is also structured in modules so that it can be exported as a filtered list by selecting countries (i.e., country codes). The collection of nodes is then filtered using the selected countries, and only those that are valid according to the selection are retained. Subsequently, the edges are also filtered, whereby both the origin and the destination must be contained in the remaining nodes. This also allows partial display of map material without having to host the entire world.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": 1078063,
      "geometry": {
        "type": "Point",
        "coordinates": [
          13.48856,
          48.21123
        ]
      },
      "properties": {
        "id": 1078063,
        "ip": "46.75.125.13",
        "as_number": 8447,
        "country": "AT"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            13.48856,
            48.21123
          ],
          [
            16.37208,
            48.20849
          ]
        ]
      },
      "properties": {
        "from": 1078063,
        "to": 21
      }
    }
  ]
}
```


Table 3.3: Example of a node and edge dataset exported in the GeoJSON format.

Depending on the version of the graph used (i.e., country code or ASN reduced) for processing, colors are then assigned to each node based on unique country codes or ASN. At this point, we will now also address a specific issue with the display of overlapping nodes. When assigning geolocation data, it often happens that several nodes are assigned to exactly the same geospatial location (i.e., longitude and latitude). To avoid this, we now merge all edges that are also contained in the dataset as overlapping duplicates. The overlapping edges themselves are displayed as a selectable list via an overlay in the card server frontend design, as shown in figure 3.6.

Finally, the data from the GeoJSON file is converted into vector tile format using the tippecanoe tool as shown in table 3.2 so that it can be read by the map server. Below is the reference command which we used to convert the data:

```
tippecanoe -o tileserver-gl/mbtiles/network_at.mbtiles -Z5 -z14
--no-feature-limit --layer=filtered_network_colored input.geojson
```

Here is a brief explanation of the used parameters:

Parameter	Value	Description
-o	tileserver-gl/mbtiles/network_at.mbtiles	Output file path for the mbtiles
-Z	5	Minimum zoom level to generate tiles (zoom from level 5)
-z	14	Maximum zoom level to generate tiles (zoom up to level 14)
–no-feature-limit	(flag)	Do not limit the number of features per tile (disable feature limit) so that no features get dropped
–layer	filtered_network_colored	Name the output layer ”filtered_network_colored”
input.geojson		Input GeoJSON file containing features

Table 3.4: Tippecanoe command parameters

3.5.2 Visualization

As further explained in chapter 3.5.1 the graph data is exported to a format which can be used for geospatial representation by using the standardized and widely used GeoJSON format. For the visualization itself, we are using map material provided by the OpenStreetMap project as a basis.

For our initial experiments, we started with public hosted OpenStreetMap instances such as uMap [4]. These instances allow users to create individualized maps by uploading a custom, self-created map layer, which can then be displayed as overlay of the basic OSM maps. However, the disadvantage of this approach is that all uploaded layer data is rendered via the web browser on the client side, which already proved to be a performance issue even

during the early experimental phase, given the amount of data to be processed here. This, combined with the fact that the visual representation of ready-made instances can be influenced little or not at all, led us to switch to working with our own OpenStreetMap server instance.

After some research, the combination of using TileServer GL on the server side and MapLibre GL JS as the frontend proved to be the powerful and adaptable solution which we needed. In conjunction with the underlying vector tile technology, which dynamically renders the individual components of the map, this provides us with the desired adaptability and scalability with good performance. A combined backend and frontend server-side setup was created using Docker and can be run easily.

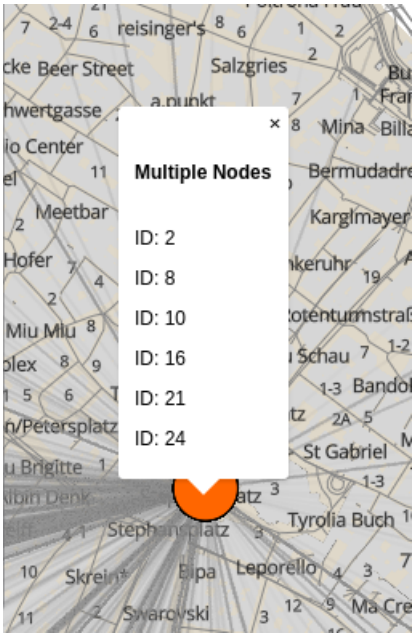


Figure 3.6: Multiple nodes selection overlay example.

The visual representation of the network layer over the map material also presents a number of additional challenges. Despite the reduction, we are still working with a significant number of nodes and edges. In order to display these in a reasonably clear manner, we had to develop our own frontend layout for the representation. In the standard zoom level of the map, not all details are displayed in order to conserve resources. Figure 3.7 shows an example of the default zoom map visuals of Austria.

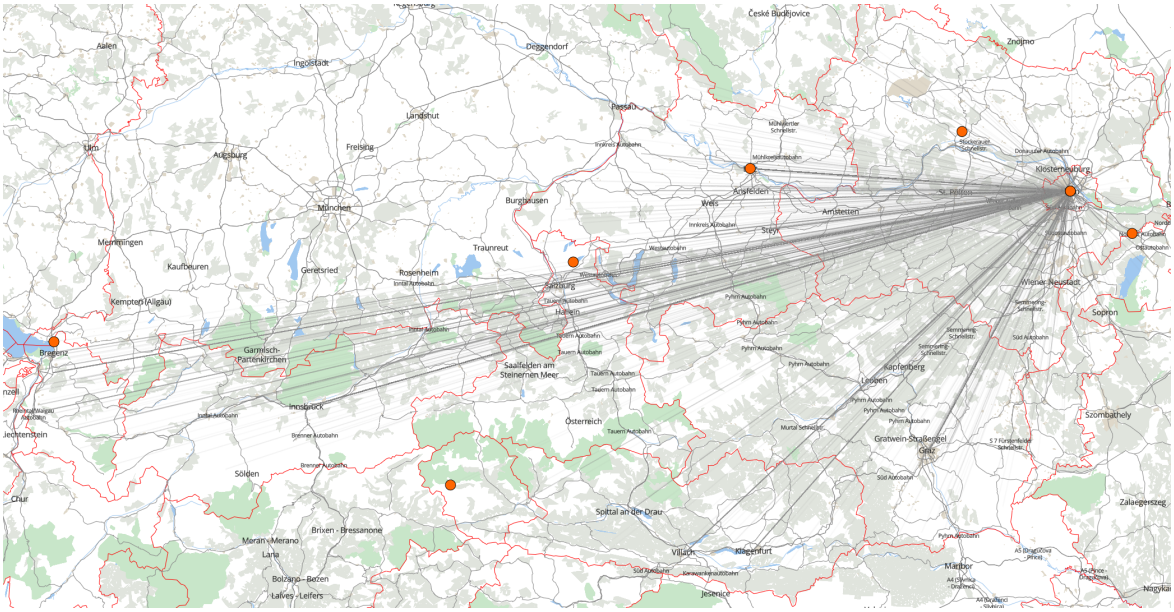


Figure 3.7: Default zoom map visuals example of Austria.

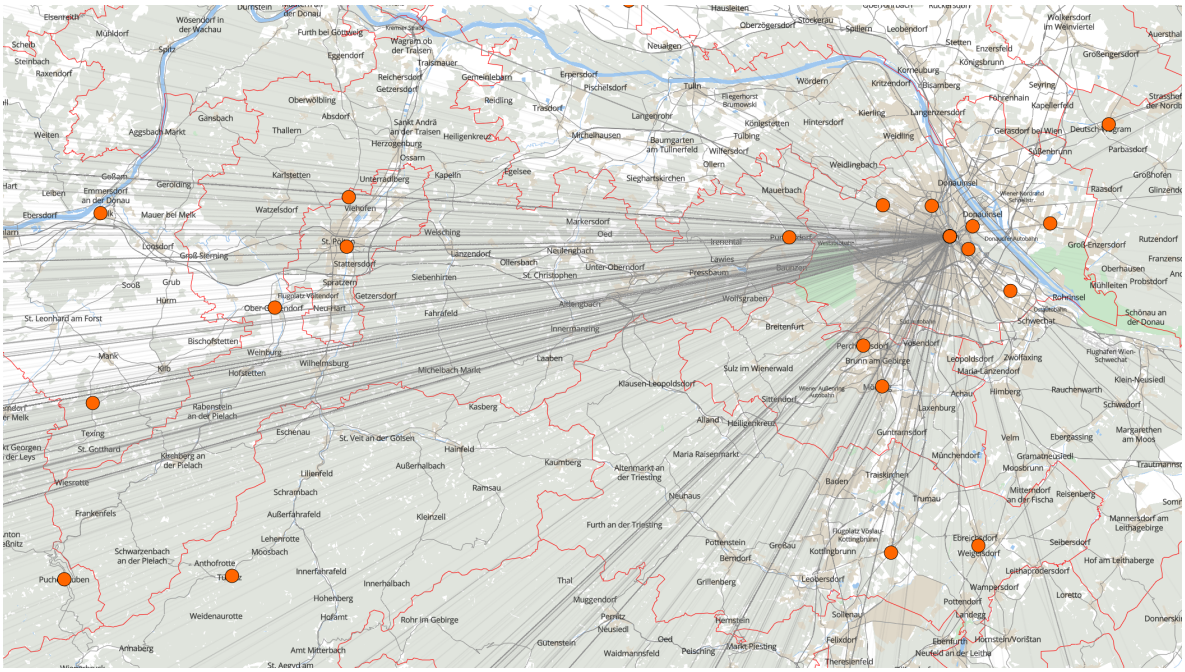


Figure 3.8: Further zoomed in map visuals example of Austria.

However, the edges between the nodes are displayed in gray even if not all nodes are shown. This was a deliberate decision so that the connection patterns can be recognized at every zoom level. In figure 3.8 it is easy to recognize that at a higher zoom level additional nodes are now displayed.

The nodes are positioned on the map using the assigned geographic coordinates. The challenge here is that it often happens that several nodes are placed overlapping each other at the same position on the map. As can be seen in figure 3.6 in order to handle these multi-nodes, a separate overlay was implemented in the design, which displays the overlapping nodes as a list and also allows individual selection.

It was also a difficult decision how best to handle the edges. Hiding them completely resulted in the loss of many details. However, fully displaying all of them in full contrast led to a chaotic display on the map during the testing phase. As a compromise, it was decided to display them but in greyedout form with low contrast settings as default. In this view, as soon as a node is actively selected by clicking on it, an overlay with information about the selected node is displayed, and all associated edges are interactively marked in red as can be seen in figure 3.9.

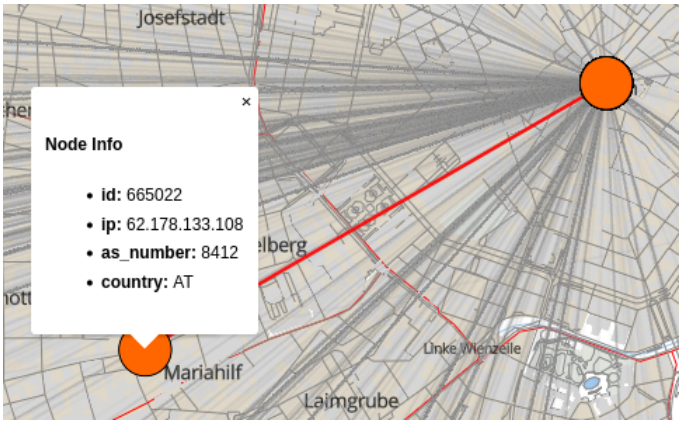


Figure 3.9: Node information overlay example.

The entire design was optimized and tested to ensure a compact and clear presentation of the relevant information.

3.6 Graph Metrics

In selecting appropriate graph metrics for analyzing and reducing the internet graph in this study, the focus is placed on centrality measures. In particular our actively used choices include degree centrality, betweenness centrality, and closeness centrality. How to implement these metrics is generally well documented, so we will only give a brief overview of the actual implementation details. Instead, our focus lies on explaining the general idea behind the reasons for the choice and also how their secondary use case can influence the dynamic generation of the visual map layer data.

3.6.0.1 Degree Centrality

Degree centrality is an important measure which evaluates the importance of a node based on the number of direct connections it has towards other nodes in the network. In an undirected network, it simply counts the number of edges incident to a node, indicating its immediate reach or activity level. In directed networks, degree centrality can be distinguished into in-degree (number of incoming edges) and out-degree (number of outgoing edges), reflecting a node's influence or susceptibility within the network [26, 59].

This measure is straightforward to compute and provides a quick estimate of a node's prominence based on its direct connections. High degree centrality suggests that a node is highly connected, potentially serving as an influential or highly active element within the network, which is relevant for understanding the network's structure and dynamics [26]. The implementation of the degree centrality is adapted from the previous work by Fodor [22] for our use.

3.6.0.2 Closeness Centrality

Closeness centrality measures how close a node is to all other nodes in the network, based on the shortest path distances. It calculates the reciprocal of the sum of the shortest distances from a node to every other node, thereby identifying nodes that can quickly reach or influence others. Nodes with high closeness centrality are considered to be well-positioned within the network, acting as central points that facilitate efficient communication or dissemination of information [26, 59, 32].

This measure is particularly useful for identifying nodes that are strategically located to minimize the number of steps required to reach the rest of the network [26]. The implementation uses the classic approach using Breadth-First Search (BFS) in order to find the shortest distances from the source node to every reachable node. It is a simple approach, but it has been greatly accelerated by the use of parallelization.

3.6.0.3 Betweenness Centrality

Betweenness centrality measures how often a node appears on the shortest paths between pairs of nodes in a network. It essentially quantifies the role of a node as a bridge or broker within the network. Nodes with high betweenness centrality have significant influence over the flow of information or resources because they connect disparate parts of the network. The measure can be calculated for both unweighted and weighted networks, considering either shortest paths or random-walk-based paths. It highlights nodes that control communication between different groups, which can be crucial for network stability, spreading processes, or identifying key influencers or bottlenecks [26, 32].

The implementation, which we are using here, is based on shortest paths, which is not trivial to compute, especially for huge graphs. We have adapted the implementation from the pre-

vious work by Fodor, which uses the Brandes algorithm and can be efficiently parallelized for our use [22].

3.6.0.4 Katz Centrality (abandoned)

Another metric which we planned to consider is the katz centrality. It offers a quantitative approach for evaluating node significance within a network. Unlike simpler metrics, it does not just account for direct connections, it also accounts for indirect relationships by analyzing all the possible paths leading to a node. Of course, it also weights these paths - the longer the path, the less influence it has on the final score. As a result, the nodes with a high katz centrality are not just well-connected themselves, they are also linked to other important nodes, making this metric a powerful tool for identifying key players in complex networks [26].

In fact, this metric was already fully implemented, but unfortunately it is not always guaranteed to converge or produce a useful result [2], and we had serious issues with our graph during the testing phase. Although a considerable amount of time was invested in attempts to solve the problem, no stable function was achieved within a reasonable time frame. Ultimately it was decided to abandon the katz centrality for this reason.

The combination of these metrics provides a solid basis for the evaluation of node importance as well as connectivity, redundancy, and communication efficiency of network structures. We use these metrics for several purposes in this work. In chapter 4, we use the results to compare and discuss the filtered country code and ASN datasets. The alternative use case is described in chapter 3.5.1. This utilizes the further potential of the metrics to enable a dynamic scope of visual representation. However, only one of the available metrics is actively used in the current scope. By changing the metric used, the central focus of the displayed map can also be changed.

4 Data Analysis

In order to determine how many nodes were represented in the chosen data source and to assess the efficacy of our reduction technique for each dataset, the analysis part starts by looking at the coverage rate of metadata related to the nodes in the graph. Then, using the chosen metrics, we present a summary of the processing time comparisons between IPv4 and IPv6 datasets. After that, we use the implemented Metrics to examine the differences between the country code and ASN datasets. Lastly, using measures of betweenness, closeness, and degree centrality, we examine and compare the key nodes in each dataset.

4.1 Evaluation of Performance

In this section of the thesis, we will examine key aspects of overall performance. This includes both the efficiency of metadata allocation through the use of available sources and a general comparison of the observed runtimes for IPv4 and IPv6.

4.1.1 Assignment of Metadata

Table 4.1 shows a comparison of the IPv4 allocation quotas for the country code and the autonomous system number for each of the two primary datasets, ASN and CC2. As we can see, the coverage of the geolocation (CC2 dataset) provided by IPinfo is almost perfect. In terms of numbers, out of approximately 12.6 million hosts, only 84 were not assigned a geolocation, which is statistically insignificant in percentage terms and therefore represents complete coverage.

The coverage of the pfx2as (prefix to AS) provided by CAIDA for the assignment of autonomous system numbers in the ASN dataset is not quite as perfect, but with a coverage rate of 98.67%, it can still be considered excellent. In absolute terms, out of a total of around 12.6 million hosts, 167,240 were not assigned an ASN.

Dataset	Nodes	Skipped	Coverage (%)
ASN	12568197	167240	98.67
CC2	12568197	84	100

Table 4.1: IPv4 metadata assignment quota

For IPv6, an overview of the values can be found in Table 4.2. When assigning geolocation data from the IPinfo data source, we once again see an excellent result. Of the approximately 5.6 billion IPv6 addresses, only 3,443 were not assigned a geolocation. Calculated as a percentage, this is also not statistically relevant and can certainly be described as perfect.

Dataset	Nodes	Skipped	Coverage (%)
ASN	5609422740	440701776	92.14
CC2	5609422740	3443	100

Table 4.2: IPv6 metadata assignment quota

However, we can observe here that the coverage of autonomous system numbers for IPv6 nodes by CAIDA's pfx2as data is not as good as for IPv4. Of the approximately 5.6 billion IPv6 nodes 440,701,776, or around 441 million, were not assigned a value. Statistically speaking, this means coverage of 92.14 %. Although this can still be considered good, it is noticeably worse than with IPv4.

We also note here that the metadata is assigned before the paths are deduplicated in the graph, as shown in figure 3.3. This also explains the difference in the amount of nodes between tables 4.1, 4.2 where the total number of nodes in the graph is specified and 4.3, 4.4. However, the number of nodes after deduplication of the paths in the graph is used for the reduction, as we also use this data as a basis for the tables listed here.

4.1.2 Reduction Efficiency

In this section of the paper, we will take a closer look at the efficiency of the strategy used to reduce the data in the graph using the associated metadata. The exact details of the reduction have already been described in detail in chapter 3.4. In table 4.3, an overview of the statistical data related to the metadata-based reduction of our two IPv4 datasets can be found. For the ASN dataset we were able to reduce the amount of nodes by 21.65 % and the edges by 40.84 %. As for the CC2 dataset, here the reduction process was able to reduce the amount of nodes by 29.09 % and the edges by 43.94 %.

Dataset	Nodes	N. Reduced	Reduction (%)	Edges	E. Reduced	Reduction (%)
ASN	1951263	1528755	21.65	7143556	4225991	40.84
CC2	1951263	1383676	29.09	7143556	4005235	43.93

Table 4.3: IPv4 reduction strategy efficiency

In table 4.4, the overview of statistical data for the metadata based reduction of our two IPv6 datasets can be seen. For the CC2 dataset we achieved a 47.9 % reduction for the number of nodes and the edges by 72.76 %. It appears that we have a particularly high concentration

of connected nodes with identical metadata in the CC2 dataset. As for the ASN dataset we have managed to achieve a reduction of 5.06 % for the amount of nodes and 24.72 % for the amount of edges.

As shown in Table 4.2, the IPv6 ASN dataset has the worst metadata coverage of our collection. It is therefore not particularly surprising and was to be expected that this shows an impact on the efficiency of our reduction strategy. This result now confirms our assumption and also shows how the quality of the data source is an important factor to consider.

Dataset	Nodes	N. Reduced	Reduction (%)	Edges	E. Reduced	Reduction (%)
ASN	230816217	219132918	5.06	454604865	342244960	24.72
CC2	230816217	120264902	47.9	454604865	123848334	72.76

Table 4.4: IPv6 reduction strategy efficiency

All in all, we have shown that the chosen strategy was able to significantly reduce the number of datasets for both nodes and edges. The differences with IPv4 are minor, but surprisingly, the reduction seems to have been particularly efficient with the combination of IPv6 and CC2.

4.1.3 Runtime Observations

At the end of this section, we will now take a closer look at the situation regarding the general processing pipeline runtimes, with a particular focus on the calculation of metrics for IPv4 and IPv6 datasets. We have singled out this specific area because it usually accounts for by far the largest share of the total processing time, especially with increasing data volumes. As a reference, we had 80 (virtual) cores in use on a server equipped with 900 GB of RAM and 500 GB of SSD swap space in use.

On average, a processing run through the basic pipeline as shown in figure 3.3, including metadata matching, but without metrics calculation, takes about 1 hour with IPv4, regardless of the dataset. With the IPv6 datasets, however, a basic run already takes about 4 days due to the substantially higher amount of data.

Metrics Type	Runtime IPv4 (Hours)	Approx. Runtime IPv6 (Hours)
Degree Centrality	1	95
Betweenness Centrality	112	2.000
Closeness Centrality	53	950

Table 4.5: Comparative overview of metrics runtime IPv4 / Ipv6

Table 4.5 provides an overview of the average processing time required by our reference system to calculate the metrics for IPv4 and IPv6. The listed reference durations were benchmarked using the reduced node amounts listed in tables 4.3 and 4.4.

The basis for estimating the durations was our worst-case ASN IPv4/IPv6 dataset, as this contains the largest amount of data. These are average measured values just to be taken as a reference in order to get an impression of the situation and therefore not related to a specific dataset. The time listed for degree centrality was actually measured in real time since we have actual reference values of all the datasets for this metric. The time specified for betweenness centrality IPv6 was extrapolated based on the progress achieved after a one-month test run. The time specified for closeness centrality was extrapolated based on the progress achieved in a reference test run lasting several days.

Starting with degree centrality, the processing times are still quite reasonable, with 1 hour for processing an IPv4 dataset. An IPv6 dataset takes 95 hours to process which is about four days. The next level is closeness centrality, which already requires 53 hours for an IPv4 dataset. For IPv6, it takes 950 hours per dataset, which is to about 40 days. The situation is worst in terms of betweenness centrality, where an IPv4 dataset already requires 112 hours, which is almost 5 days. An IPv6 dataset here then requires an astronomical 2,000 hours to complete a run, which corresponds to around 3 months. During the runs, all available server resources were used. The code is already highly optimized and parallelized. Further discussion of performance and the associated challenges can be found in chapter 5.

4.2 Evaluation of IPv4 Metrics

Figure 4.1 shows the distribution of betweenness centrality for the reduced CC2 and ASN datasets. We can observe that there are noticeable differences in the resulting structure between our two reduced datasets, CC2 and ASN. The two curves show discontinuities in the middle of the value range between 10^{-10} and 10^{-6}. In this range, CC2 values are generally lower than ASN.

The two jumps in the curves of the datasets suggest the presence of clusters or communities with similar results. Compared to ASN, the lower betweenness values of the CC2 data indicate a decentralized structure with fewer dominant intermediary nodes. The observed differences reflect the effects of the applied reduction strategies.

The CC2 and ASN datasets' cumulative distribution of closeness centrality values is shown in figure 4.2. Both distributions exhibit significant deviations from both the norm and from one another. The curve in the ASN dataset indicates that many nodes exhibit high connectivity to the rest of the network, rising sharply between 0.2 and 0.3 before flattening gradually.

The presence of particularly scattered nodes or substructures is indicated by the CC2 dataset, which grows gradually at first but then exhibits two rapid increases between 0.5 and 0.7. The ASN network displays a centralized and effective structure for information flow, whereas the CC2 graph exhibits a scattered topology, based on these contrasts.

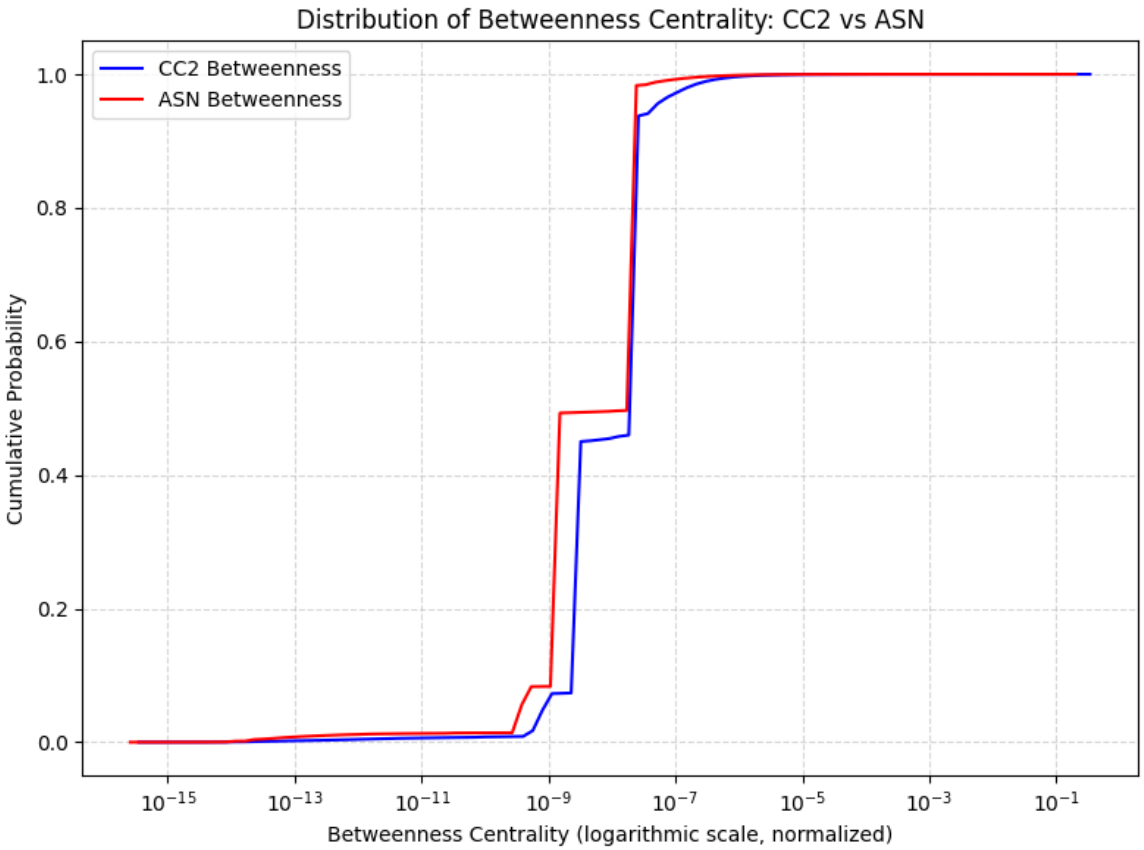


Figure 4.1: Distribution of IPv4 betweenness centrality values for ASN and CC2 datasets.

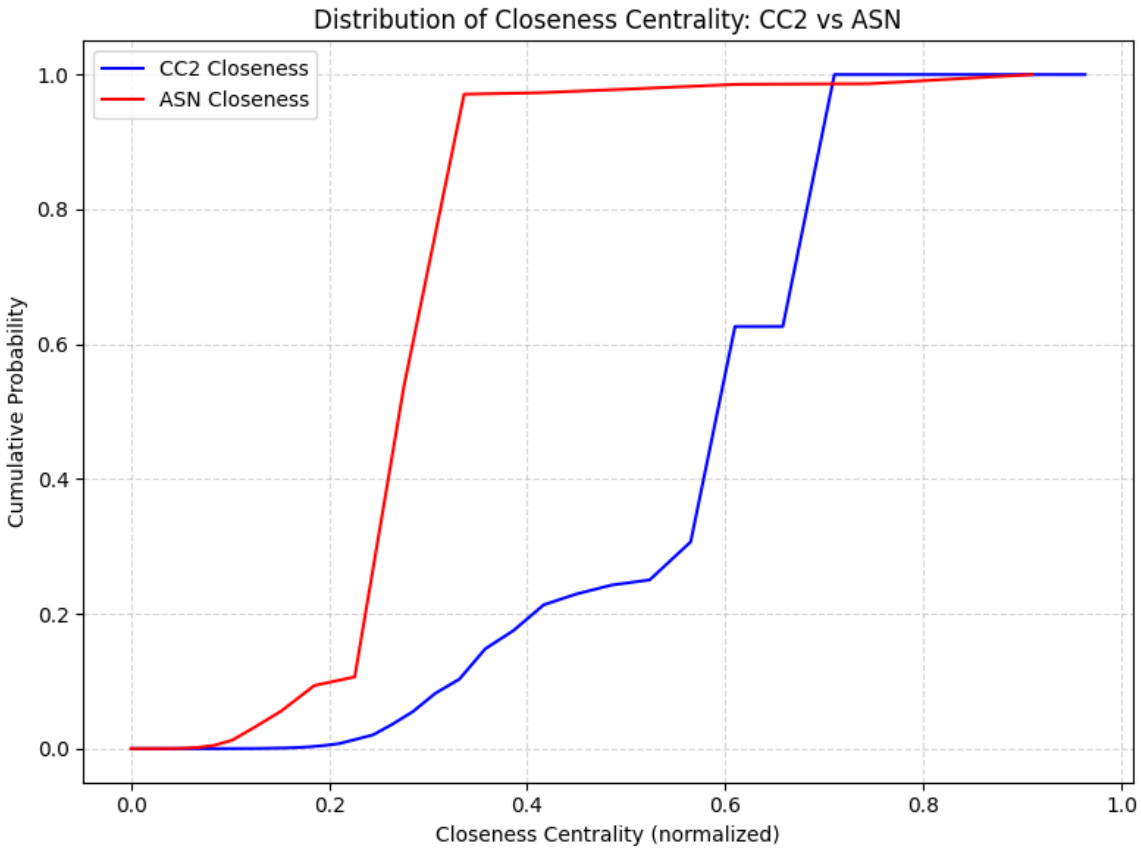


Figure 4.2: Distribution of IPv4 closeness centrality values for ASN and CC2 datasets.

The cumulative distribution of degree centrality values for the ASN and CC2 datasets is displayed in figure 4.3. The two curves highlight two distinct structural patterns. The cumulative probability for the ASN dataset remains at zero until $10^{-6.5}$, then sharply increases from 0.25 at $10^{-6.5}$ to almost 0.98 at $10^{-6.25}$, before rapidly reaching saturation. Nearly all ASN nodes lie within a narrow range of degree centrality values, as indicated by this sharp concentration. With minimal indication of dominant hubs and the majority of nodes occupying comparable topological roles, such a distribution points to a homogeneous and uniform connectivity structure.

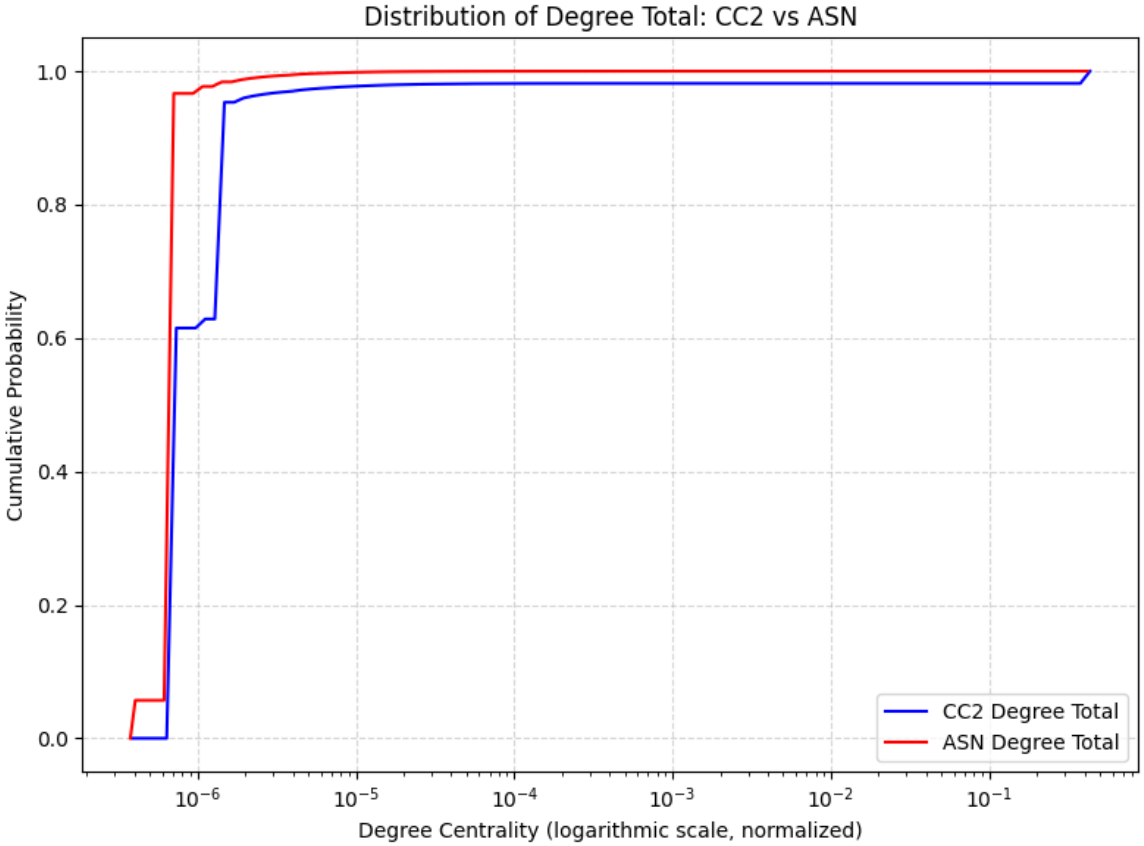


Figure 4.3: Distribution of IPv4 degree centrality values for ASN and CC2 datasets.

The CC2 dataset, on the other hand, shows a slower progression. CC2 reaches 0.65 at $10^{-6.25}$ and increases to 0.90 at 10^{-6}; it only gets close to saturation after this. With a smaller collection of highly connected hubs and a mixture of several distant low-degree nodes, this slower, multi-stage expansion suggests increased degree value heterogeneity. A modular or hierarchical topology is reflected in such a structure.

Together, these findings show that the CC2 reduction approach facilitates region-based visualization and analysis by highlighting spatial hierarchies and creating networks with rec-

ognizable hub regions. The organizational distribution of internet connectivity at the AS level is mirrored by the flat, decentralized structure that results from the ASN decrease. This complementary pattern implies that each reduction technique emphasizes different structural elements: ASN stresses diffused organizational linkages, whereas CC2 reveals geographic hubs and hierarchy.

4.3 Evaluation of IPv6 Metrics

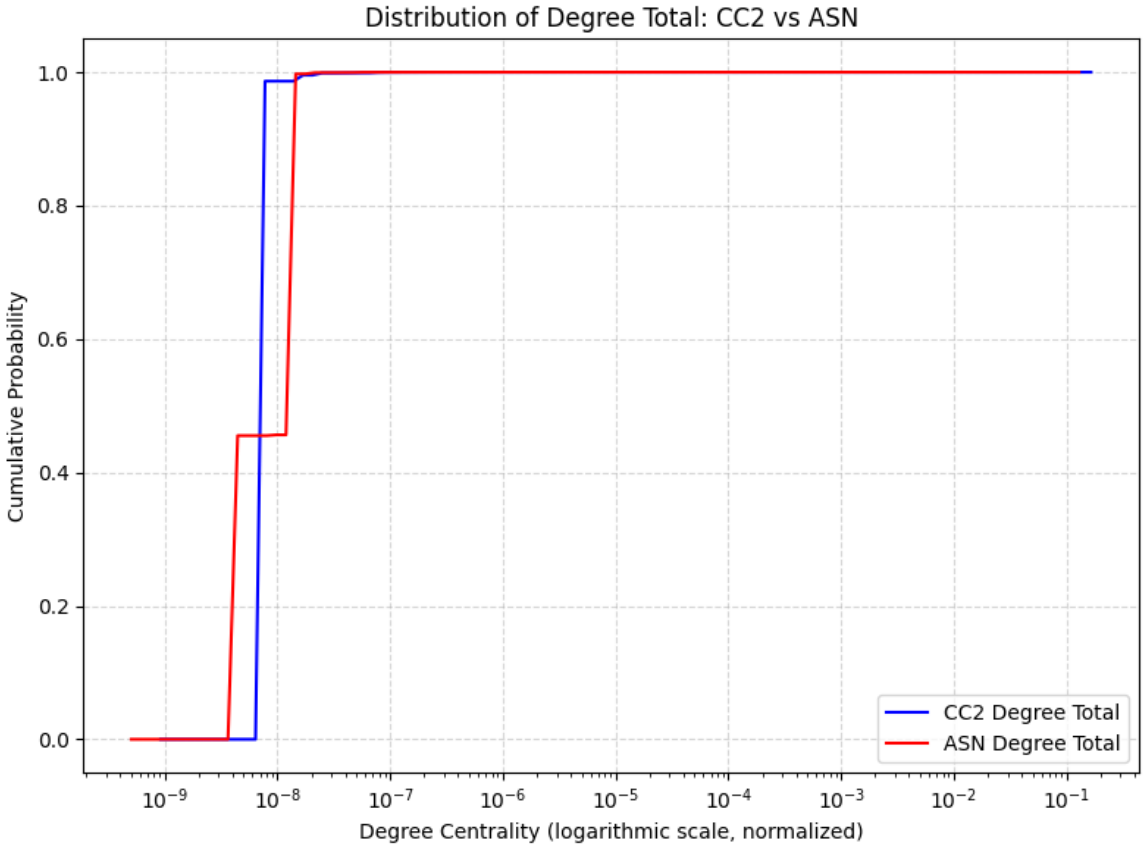


Figure 4.4: Distribution of IPv6 degree centrality values for ASN and CC2 datasets.

Only degree centrality was assessed for the IPv6 datasets due to computational limitations. Within the time and computational budget constraints, excessively resource-intensive metrics such as betweenness centrality and closeness were not calculated (for details, see table 4.5). However, degree centrality continues to offer important information about the IPv6 topology.

The cumulative distribution function (CDF) of degree centrality values for the IPv6 datasets based on CC2 and ASN is displayed in figure 4.4. The distribution of the ASN dataset shows

a high initial climb, a flat area at 10^{-8}, and then another sharp increase. Nodes do not lie within this centrality range, as indicated by the flat interval, which points to structural discontinuities in node connectedness. The CC2 dataset, on the other hand, shows a steady increase with only a slight plateau about 10^{-8}.

In comparison to IPv4, the degree centrality distributions for IPv6 exhibit a compact and steep form overall, with limited dispersion (figure 4.3). This suggests that highly centralized hubs are absent from IPv6 networks. Practically speaking, this points to a less hierarchical infrastructure than IPv4, representing an internet whose connectivity is dispersed widely between nodes instead of being concentrated in a few powerful hubs. One plausible reason is that IPv6, which is still in the deployment stage, has not yet developed to the level of centralization and aggregation already present in the fully established IPv4 environment.

4.4 Analysis of Significant IPv4 nodes

In this part of the work we focus on identifying and interpreting the key nodes within the datasets. We examine the nodes with the highest scores across the three metrics betweenness, closeness, and degree centrality and compare them with each other.

At this point, we want to emphasize once again that the scan server, which our data originates from, is located in Vienna. The collected data therefore reflects the structure of the internet as observed from the perspective of Vienna.

4.4.1 Betweenness Centrality

Table 4.6 shows the 25 nodes of the CC2 dataset with the highest betweenness centrality values. Unfortunately, the CAIDA pfx2as database that we used for assigning AS numbers did not provide AS numbers for all nodes, so we decided to supplement the missing ASN values where possible by using WHOIS in the table. The data clearly shows that AS1764 dominates the list while also representing many nodes with high values. This highlights the critical role of AS1764 as a backbone or transit network in the region.

Numerous nodes with high scores are concentrated in Austria (AT), particularly those belonging to AS1764. Other countries, like, for example, Germany (DE), the Netherlands (NL), China (CN) and the USA (US) also show presence but with fewer nodes and, in general lower betweenness values. This regional concentration is to be expected since our scanning server is located in Vienna, introducing a dataset bias. Even with this bias, these nodes play an important role in international data traffic, despite showing lower betweenness values in our results. We can also see a sharp drop from the highest betweenness node (0.3973) and the second (0.1906) as well as the subsequent nodes in the list. The top nodes probably act as important hubs in the network, which highlights their importance for routing and information flow.

Node ID	Betweenness	IP	AS number	AS name	Country
24	0.397303	92.60.1.233	1764[1]	NEXTLAYER-AS	AT
2	0.190610	212.133.7.161	3356	Level 3 Parent, LLC	AT
28	0.171612	92.60.1.231	1764[1]	NEXTLAYER-AS	AT
10	0.017938	92.60.0.231	1764[1]	NEXTLAYER-AS	AT
16	0.015655	92.60.0.233	1764[1]	NEXTLAYER-AS	AT
84164	0.012146	92.60.1.22	1764[1]	NEXTLAYER-AS	AT
21	0.009494	92.60.1.118	1764[1]	NEXTLAYER-AS	AT
1869	0.008715	80.249.214.255	1200[1]	Amsterdam Internet Exchange	NL
123699	0.007621	195.72.81.194	3257	GTT Communications Inc.	DE
222917	0.006820	141.136.106.30	3257	GTT Communications Inc.	CA
7041	0.006362	219.158.3.241	4837	CHINA UNICOM Backbone	CN
53	0.004899	80.81.195.33	-	-	DE
166	0.004598	92.60.2.201	1764[1]	NEXTLAYER-AS	DE
45	0.004101	92.60.2.197	1764[1]	NEXTLAYER-AS	DE
3629	0.003872	80.249.211.184	1200[1]	Amsterdam Internet Exchange	NL
68887	0.003148	92.60.0.140	1764[1]	NEXTLAYER-AS	AT
147	0.003108	80.81.194.117	-	-	DE
5525	0.002919	64.71.165.22	6939	Hurricane Electric LLC	US
35	0.002153	154.54.36.253	174	Cogent Communications	DE
2705913	0.001320	213.226.72.21	41114	Blue Telecom	FR
1187200	0.001319	178.137.95.252	15895	Kyivstar PJSC	UA
334	0.001293	77.67.76.53	3257	GTT Communications Inc.	AT
277301	0.000892	202.97.39.182	4134	CHINANET-BACKBONE	CN
8	0.000798	62.115.41.224	1299	Arelion (Tier-1 ISP)	AT
24779	0.000666	4.69.140.134	3356	Level 3 Parent, LLC	US

Table 4.6: CC2 IPv4 betweenness centrality statistics

[1] ASN sourced by WHOIS query as it was not covered in the CAIDA pfx2as dataset.

Node ID	Betweenness	IP	AS number	AS name	Country
24	0.235035	92.60.1.233	1764[1]	NEXTLAYER-AS	AT
28	0.134430	92.60.1.231	1764[1]	NEXTLAYER-AS	AT
1869	0.016100	80.249.214.255	1200[1]	Amsterdam Internet Exchange	NL
10	0.009899	92.60.0.231	1764[1]	NEXTLAYER-AS	AT
16	0.009594	92.60.0.233	1764[1]	NEXTLAYER-AS	AT
84164	0.009133	92.60.1.22	1764[1]	NEXTLAYER-AS	AT
123699	0.007754	195.72.81.194	3257	GTT Communications Inc.	DE
3629	0.007154	80.249.211.184	1200[1]	Amsterdam Internet Exchange	NL
21	0.007091	92.60.1.118	1764[1]	NEXTLAYER-AS	AT
222917	0.006824	141.136.106.30	3257	GTT Communications Inc.	CA
2	0.006794	212.133.7.161	3356	Level 3 Parent, LLC	AT
147	0.005606	80.81.194.117	-	-	DE
166	0.004339	92.60.2.201	1764[1]	NEXTLAYER-AS	DE
45	0.004152	92.60.2.197	1764[1]	NEXTLAYER-AS	DE
53	0.002865	80.81.195.33	-	-	DE
68887	0.002521	92.60.0.140	1764[1]	NEXTLAYER-AS	AT
374727	0.001999	193.203.0.16	1120[1]	ACOnet	AT
1187200	0.001777	178.137.95.252	15895	Kyivstar PJSC	UA
2705913	0.001777	213.226.72.21	41114	Blue Telecom	FR
24779	0.001074	4.69.140.134	3356	Level 3 Parent, LLC	US
3162	0.000982	4.68.38.90	3356	Level 3 Parent, LLC	AT
8	0.000629	62.115.41.224	1299	Arelion (Tier-1 ISP)	AT
3136	0.000536	130.117.50.42	174	Cogent Communications	FR
231489	0.000449	80.249.211.63	1200[1]	Amsterdam Internet Exchange	NL
6582	0.000448	80.249.211.171	1200[1]	Amsterdam Internet Exchange	NL

Table 4.7: ASN IPv4 betweenness centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

In Table 4.7, we see a comparison of the 25 highest-ranked nodes from the ASN dataset as determined by the betweenness centrality values. We can see that AS1764 still dominates the list of nodes with the highest scores, but the betweenness centrality values are generally lower compared to the CC2 dataset. The highest score in the ASN reduced set (0.2350) is significantly lower than the highest value (0.3973) in the CC2 dataset. It can also be observed that the dispersion of values between the results is significantly lower in this dataset. It seems that reduction by AS affiliation reduces peak centralities and the distribution of centrality.

This reduction effect occurs because grouping nodes by autonomous system affiliation combines many individual IP nodes into fewer representative units. As a consequence, the significance of the scores shifts from individual nodes to complete AS groups. This is also evident from the fact that, compared to the CC2 dataset, the significance of individual nodes with extreme values has been significantly reduced. This demonstrates and confirms the approach that ASN-based reduction focuses on organization-based structures, while CC2 reduction emphasizes geographically clustered high centrality nodes.

4.4.2 Closeness Centrality

Similar to betweenness centrality, we see in table 4.8 that AS1764 also occupies the top positions in the ranking for closeness centrality of the CC2 dataset. This further emphasizes the role of AS1764 as a transit network that is primarily important for Austria locally. The observed accumulation of regionally located nodes with high scores in Austria can also be explained here by the location of the scan server in Vienna.

In contrast to betweenness centrality, which primarily identifies nodes functioning as bottlenecks or bridges between communities, closeness centrality highlights nodes that are capable of disseminating information with high efficiency across the network as a result of their short average distance to other nodes. The majority of the top-rated nodes is located in Austria here as well. There are, however, also nodes from other countries showing up, such as the United States (US), Brazil (BR), and Colombia (CO). The appearance of these nodes highlights the international cross-border influence of autonomous systems such as AS3356 and AS3549. This shows that some AS apparently play an important role in intercontinental routing for certain regions.

These results are significant because they indicate that centrality complements betweenness in highlighting efficient nodes for information dissemination, in addition to structural bottlenecks. The combination of these two measures truly captures some of the important characteristics in a network.

Table 4.9 contains the list of the top 25 nodes ranked by their closeness centrality scores for the ASN dataset. The nodes of AS1764 also dominate the top of the list here. However, it is striking that, compared to the CC2 dataset, a higher geographical distribution of nodes

Node ID	Closeness	IP	AS number	AS name	Country
24	0.336320	92.60.1.233	1764[1]	NEXTLAYER-AS	AT
10	0.331399	92.60.0.231	1764[1]	NEXTLAYER-AS	AT
16	0.330760	92.60.0.233	1764[1]	NEXTLAYER-AS	AT
28	0.327850	92.60.1.231	1764[1]	NEXTLAYER-AS	AT
166	0.323443	92.60.2.201	1764[1]	NEXTLAYER-AS	DE
2	0.322783	212.133.7.161	3356	Level 3 Parent, LLC	AT
45	0.322555	92.60.2.197	1764[1]	NEXTLAYER-AS	DE
123699	0.322528	195.72.81.194	3257	GTT Communications Inc.	DE
222917	0.318680	141.136.106.30	3257	GTT Communications Inc.	CA
19	0.315564	154.54.59.182	174	Cogent Communications	DE
8	0.315151	62.115.41.224	1299	Arelion (Tier-1 ISP)	AT
199	0.313707	92.60.3.74	1764[1]	NEXTLAYER-AS	AT
58	0.313482	92.60.3.72	1764[1]	NEXTLAYER-AS	AT
334	0.312814	77.67.76.53	3257	GTT Communications Inc.	AT
1128	0.312585	176.52.248.241	12956	TELXIUS Cable	FR
264193	0.312010	213.19.223.50	3356	Level 3 Parent, LLC	NL
210801	0.312006	200.189.212.194	3549	Level 3 Parent, LLC	BR
1684468	0.311973	4.59.250.98	3356	Level 3 Parent, LLC	US
572364	0.311973	4.68.94.50	3356	Level 3 Parent, LLC	US
355012	0.311957	4.7.158.118	3356	Level 3 Parent, LLC	US
626157	0.311956	4.15.45.186	3356	Level 3 Parent, LLC	US
104318	0.311955	212.187.202.154	3356	Level 3 Parent, LLC	DE
1159533	0.311953	200.189.207.194	3549	Level 3 Parent, LLC	CO
8944116	0.311953	4.15.196.146	3356	Level 3 Parent, LLC	US
4597117	0.311953	4.31.132.138	3356	Level 3 Parent, LLC	US

Table 4.8: CC2 IPv4 closeness centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

by country and by ASN can be observed. This shows that this type of analysis emphasizes cross-border structures.

The general range of scores is noticeably higher for the ASN dataset, being between 0.36 and 0.39 compared to the CC2 dataset with a range of 0.31 to 0.33. This can be explained by the fact that, due to the ASN-based reduction, several nodes are merged within identical AS. This approach reduces the distance within the merged components, thereby increasing the closeness scores. There are also some specific examples of nodes within the dataset that support this, like those belonging to AS3257, which have clearly moved up in the ranking.

Both datasets show a major influence from the AT scan server, but the ASN dataset regulates the local influence with clarity, while nodes from other countries receive stronger emphasis. This shows once again that ASN-based reduction can efficiently highlight larger connection patterns in network structures.

Node ID	Closeness	IP	AS number	AS name	Country
24	0.388124	92.60.1.233	1764[1]	NEXTLAYER-AS	AT
10	0.382940	92.60.0.231	1764[1]	NEXTLAYER-AS	AT
16	0.379964	92.60.0.233	1764[1]	NEXTLAYER-AS	AT
123699	0.377316	195.72.81.194	3257	GTT Communications Inc.	DE
28	0.376888	92.60.1.231	1764[1]	NEXTLAYER-AS	AT
222917	0.374513	141.136.106.30	3257	GTT Communications Inc.	CA
166	0.370425	92.60.2.201	1764[1]	NEXTLAYER-AS	DE
45	0.369858	92.60.2.197	1764[1]	NEXTLAYER-AS	DE
199	0.362820	92.60.3.74	1764[1]	NEXTLAYER-AS	AT
58	0.362635	92.60.3.72	1764[1]	NEXTLAYER-AS	AT
1187200	0.362548	178.137.95.252	15895	Kyivstar PJSC	UA
2705913	0.362524	213.226.72.21	41114	Blue Telecom	FR
2	0.362377	212.133.7.161	3356	Level 3 Parent, LLC	AT
1128	0.362174	176.52.248.241	12956	TELXIUS Cable	FR
177	0.362171	92.60.3.31	1764[1]	NEXTLAYER-AS	AT
77926	0.361567	202.97.53.73	4134	CHINANET-BACKBONE	CN
204527	0.361323	87.226.133.28	12389	ROSTELECOM-AS	RU
210801	0.361186	200.189.212.194	3549	Level 3 Parent, LLC	BR
100030	0.361134	106.38.244.114	23724	IDC, China Telecom Corp.	CN
1159533	0.360990	200.189.207.194	3549	Level 3 Parent, LLC	CO
1454711	0.360990	8.8.246.93	989	ANAXA3-ASN	US
7743026	0.360990	8.19.178.202	26143	GSP4040 Greystone Power	US
277301	0.360774	202.97.39.182	4134	CHINANET-BACKBONE	CN
1636166	0.360619	124.74.210.98	4812	China Telecom (Group)	CN
527725	0.360228	100.41.23.91	701	UUNET Verizon Business	US

Table 4.9: ASN IPv4 closeness centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

4.4.3 Degree Centrality

Table 4.10 contains the top 25 degree centrality values for the CC2 dataset. In this dataset, we see a particularly clear dominance of nodes belonging to AS1764 from Austria. This is followed by a sharp drop in values for other AS. The observed contrast between the degree scores further hints toward AS1764 taking a central role in our topology.

The dominance of values from Austria clearly indicates a regional cluster of highly connected nodes. From this, we can once again recognize the bias caused by the scan server located in Vienna. Degree centrality observes direct connectivity opposed to betweenness and closeness, which rely on shortest paths instead. Its role is to contribute to identifying local hubs and connectors in the network within the selected combination of metrics.

The observed nodes with high degree centrality are not necessarily bottlenecks or able to efficiently spread information when their neighbors are poorly connected. There are quite a few nodes from the United States (US) and Germany (DE) listed in the top 25 and even though they themselves do not have high scores, this still indicates transit or gateway nodes.

Node ID	Degree	IP	AS number	AS name	Country
24	0.227646	92.60.1.233	1764^{1}	NEXTLAYER-AS	AT
10	0.192521	92.60.0.231	1764^{1}	NEXTLAYER-AS	AT
16	0.178920	92.60.0.233	1764^{1}	NEXTLAYER-AS	AT
28	0.150209	92.60.1.231	1764^{1}	NEXTLAYER-AS	AT
21	0.059918	92.60.1.118	1764^{1}	NEXTLAYER-AS	AT
84164	0.049320	92.60.1.22	1764^{1}	NEXTLAYER-AS	AT
68887	0.016633	92.60.0.140	1764^{1}	NEXTLAYER-AS	AT
1347	0.005532	62.159.61.126	3320	Deutsche Telekom AG	AT
2	0.003822	212.133.7.161	3356	Level 3 Parent, LLC	AT
166	0.001496	92.60.2.201	1764^{1}	NEXTLAYER-AS	DE
1644	0.001471	213.192.184.77	6667	EUNET-FINLAND	FI
691	0.001154	62.115.123.125	1299	Arelion (Tier-1 ISP)	US
45	0.001071	92.60.2.197	1764^{1}	NEXTLAYER-AS	DE
270940	0.001071	195.8.30.246	8657	MEO S.A.	PT
5659	0.001008	80.81.193.22	-	-	DE
852	0.000983	62.115.123.123	1299	Arelion (Tier-1 ISP)	US
35764	0.000775	80.81.195.188	-	-	DE
14758	0.000756	4.68.75.102	3356	Level 3 Parent, LLC	US
14706	0.000750	62.115.145.99	1299	Arelion (Tier-1 ISP)	US
57517	0.000748	80.81.192.229	-	-	DE
517327	0.000740	92.60.1.102	1764^{1}	NEXTLAYER-AS	AT
2329	0.000700	62.115.138.190	1299	Arelion (Tier-1 ISP)	US
147	0.000662	80.81.194.117	-	-	DE
2084	0.000641	80.81.194.250	-	-	DE
2299	0.000610	80.81.196.112	-	-	DE

Table 4.10: CC2 IPv4 degree centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

In table 4.11 we can observe that the situation concerning the dominance of Austrian nodes belonging to AS1764 appears similar in this case. When comparing the two datasets, CC2 and ASN, there is hardly any difference in the order of the top-ranked nodes. This in turn confirms that the geographically observed network hubs actually appear to be organizational network hubs as well. The steep drop of scores following after the first group of top-ranked Austrian nodes can be recognized here as well. The important role of AS1764 within the local network structure appears to remain stable in both perspectives (geography and AS affiliation). The type of stability we observed here confirms the central structural importance of nodes and emphasizes their significance for the resilience of the network.

Node ID	Degree	IP	AS number	AS name	Country
24	0.212759	92.60.1.233	1764^{1}	NEXTLAYER-AS	AT
10	0.179933	92.60.0.231	1764^{1}	NEXTLAYER-AS	AT
16	0.167221	92.60.0.233	1764^{1}	NEXTLAYER-AS	AT
28	0.140389	92.60.1.231	1764^{1}	NEXTLAYER-AS	AT
21	0.056005	92.60.1.118	1764^{1}	NEXTLAYER-AS	AT
84164	0.046100	92.60.1.22	1764^{1}	NEXTLAYER-AS	AT
68887	0.015550	92.60.0.140	1764^{1}	NEXTLAYER-AS	AT
166	0.001404	92.60.2.201	1764^{1}	NEXTLAYER-AS	DE
1644	0.001374	213.192.184.77	6667	EUNET-FINLAND	FI
691	0.001015	62.115.123.125	1299	Arelion (Tier-1 ISP)	US
45	0.001007	92.60.2.197	1764^{1}	NEXTLAYER-AS	DE
270940	0.001000	195.8.30.246	8657	MEO S.A.	PT
5659	0.000947	80.81.193.22	-	-	DE
19416	0.000880	195.8.30.238	8657	MEO S.A.	PT
852	0.000871	62.115.123.123	1299	Arelion (Tier-1 ISP)	US
76131	0.000759	195.8.30.250	8657	MEO S.A.	PT
35764	0.000724	80.81.195.188	-	-	DE
14758	0.000705	4.68.75.102	3356	Level 3 Parent, LLC	US
57517	0.000704	80.81.192.229	-	-	DE
14706	0.000698	62.115.145.99	1299	Arelion (Tier-1 ISP)	US
517327	0.000692	92.60.1.102	1764^{1}	NEXTLAYER-AS	AT
36370	0.000654	195.8.30.242	8657	MEO S.A.	PT
147	0.000619	80.81.194.117	-	-	DE
2329	0.000614	62.115.138.190	1299	Arelion (Tier-1 ISP)	US
2084	0.000601	80.81.194.250	-	-	DE

Table 4.11: ASN IPv4 degree centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

4.5 Analysis of Significant IPv6 nodes

The distribution of the highest IPv6 degree centrality values for the CC2 network is highly skewed, as shown in Table 4.12. The score of the top-ranked node, belonging to AS3257 and located in Germany, far exceeds all others. This discrepancy indicates the existence of

nodes that support the global IPv6 infrastructure and serve as important transit points for regional and international data traffic.

European and East Asian autonomous systems are particularly well represented among the top-ranked nodes, which include several countries from around the world. There are a number of nodes that we were unable to assign to an AS based on the available data sources. This serves as a reminder of the challenges we still face in global IPv6 measurement. The diversity of the top central nodes at the international and AS level shows that although IPv6 adoption is increasing, its backbone still heavily depends on a select group of networks.

Node ID	Degree	IP	AS number	AS name	Country
9	0.180272	2001:668:0:3::9000:181	3257	GTT Communications Inc.	DE
8505951	0.043528	2001:668:0:3:ffff:1:0:1c32	3257	GTT Communications Inc.	DE
680	0.024655	2001:67c:1874:1::39	8772	NLineLLC	CZ
9279	0.018228	2a03:21c0:0:106::2	41722	MIRAN-AS	RU
16911563	0.014179	2001:19f0:fc00::a42:36	-	-	US
343	0.013423	2001:1af8:8000::251	60781	LEASEWEB-NL-AMS-01	NL
2322	0.013400	2001:1af8:8000::250	60781	LEASEWEB-NL-AMS-01	NL
16909336	0.013062	2a12:3200:2000:2000::9	25198	ZetServers	DE
16915593	0.010440	2a02:6ea0:1:3::67	-	-	BR
1756	0.010093	2001:7f8:1::a504:3641:1	6777[1]	AMS-IX-RS	NL
4174582	0.008725	2a04:ac00::7:a	56534	Comfortel Ltd.	RU
4180498	0.006473	2001:19f0:fc00::a43:82	-	-	KR
6554	0.006242	2001:19f0:fc00::a44:38e	-	-	KR
5669	0.006240	2001:19f0:fc00::a44:37e	-	-	KR
5213	0.005910	2001:19f0:fc00::a49:28a	-	-	KR
24910	0.005900	2001:19f0:fc00::a49:292	-	-	KR
4189040	0.005428	2a02:6ea0:1:3::89	-	-	BR
4187251	0.005424	2a02:6ea0:1:3::93	-	-	BR
4171033	0.004895	2602:fd50:0:ffff::243	835	GOCODEIT-EDGE	CA
38144572	0.004353	2001:7f8:1::a506:1323:1	6777[1]	AMS-IX-RS	NL
4184287	0.004353	2a00:1768::3:1	43350	NForce Entertainment B.V.	NL
38150455	0.004352	2404:a800:2a00::506	9498	Bharti Airtel Ltd.	IN
46655877	0.004347	2001:19f0:fc00::a40:9ae	-	-	KR
8234	0.004345	2001:19f0:fc00::a40:97e	-	-	KR
7656	0.004343	2001:1890:c07:ae03::1168:d82a	7018	ATT-INTERNET4	US

Table 4.12: CC2 IPv6 degree centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

The ASN and CC2 IPv6 degree centrality tables show some structural insights into the IPv6 topology. The global importance of organizations such as, for example, Hurricane Electric (AS6939) and Deutsche Telekom AG (AS3257) is emphasized when the data is filtered by AS affiliation. These units have a high degree of centrality in the metrics and help to enable global reachability. In contrast to that, reduction by using country codes reveals larger groups of nodes with degree centrality values distributed over various different countries instead.

The classic core-periphery structure is indicated by the sharp decline in degree after the first few nodes in both tables. Larger global AS are emphasized clearly in the ASN dataset, while the CC2 data highlights a distributed, geolocated backbone. Analyzing global IPv6 topology is challenging due to unassigned AS numbers and reliance on external data sources

with limited coverage. The results allow for two important observations. IPv6 is becoming increasingly widespread, however it is controlled by a few influential organizations and regions. This concentration presents both unique opportunities for coordinated development and risks related to potential vulnerabilities during this key stage of IPv6 integration into the internet.

Node ID	Degree	IP	AS number	AS name	Country
2760	0.141022	2001:470:15b::1	6939	Hurricane Electric LLC	US
9	0.099012	2001:668:0:3::9000:181	3257	GTT Communications Inc.	DE
9279	0.030957	2a03:21c0:0:106::2	41722	MIRAN-AS	RU
8505951	0.026279	2001:668:0:3:ffff:1:0:1c32	3257	GTT Communications Inc.	DE
8475	0.019064	2a02:2518:0:31:0:4:4812:2	28917	Fiord-AS	RU
680	0.015738	2001:67c:1874:1::39	8772	NLineLLC	CZ
10484	0.011840	2001:978:2:a1::b:2	174	Cogent Communications	TR
8506710	0.010448	2a00:13c0:0:100::2	29076	CityTelecom LLC	RU
4174582	0.010007	2a04:ac00::7:a	56534	Comfortel Ltd.	RU
1643	0.009104	951:0:0:30::3	-	-	TR
4171920	0.007966	2a13:3d80:0:12::2	213220	Norlys Digital A/S	RU
16911563	0.007782	2001:19f0:fc00::a42:36	-	-	US
16909336	0.007168	2a12:3200:2000:2000::9	25198	ZetServers	DE
4170889	0.007167	2a02:6ea0:de0a::2	60068	Datacamp Limited	CA
4182167	0.006568	2a02:6ea0:1:1::81	-	-	GB
4173154	0.006567	2a02:6ea0:1:1::79	-	-	GB
16915904	0.005736	2a02:6ea0:1:3::71	-	-	US
16915593	0.005730	2a02:6ea0:1:3::67	-	-	BR
2382	0.005650	2001:978:2:a1::b:3	174	Cogent Communications	TR
1756	0.005539	2001:7f8:1::a504:3641:1	6777[1]	AMS-IX-RS	NL
4186955	0.005022	2001:7f8:13::a500:1136:1	-	-	NL
4188782	0.004984	2001:7f8:13::a500:1136:2	-	-	NL
4169826	0.004877	2001:678:c5c::29	50509	Transroute Telecom Ltd	RU
4171944	0.004783	2a12:b200:1:2:1::12	49581	Ferdinand Zink Trading	DE
10663	0.004778	2001:470:1:eab::2	6939	Hurricane Electric LLC	US

Table 4.13: ASN IPv6 degree centrality statistics

[1] ASN was supplemented by a WHOIS query, as it was not covered in the CAIDA pfx2as dataset.

5 Discussion

This chapter discusses the key challenges and limitations in implementing the proposed framework for reducing internet graphs. These range from computational and hardware limitations to methodological compromises and structural side effects of graph processing. For each point, the cause, effects, and possible improvements are considered.

5.1 Performance and Hardware Requirements

One of the central challenges in this work is overall the performance and hardware requirements for data processing. As discussed in the sections 3.1 and 4.1.2, we are working with extensive amounts of data, particularly with IPv6. We have employed a well-equipped system consisting of the following hardware:

- 80 virtual cores
- 900 GB RAM
- 500 GB SSD swap space

Despite these substantial resources, data processing remained a bottleneck. Even after applying graph reduction strategies, the IPv6 datasets still contained hundreds of millions of nodes and edges, making analysis highly resource-intensive.

5.2 Computational Bottlenecks in Metric Calculation

The employed static reduction strategy also clearly showed its intended effect of reducing data, as further discussed in section 4.1.2. With Rust, we are using a highly efficient programming language, and the code has already been as heavily optimized as it seemed possible. However, even dealing with the reduced amounts of data has still remained a resource-intensive task so far in some aspects.

The calculation of metrics remains a tedious task, with betweenness centrality being the computationally heaviest in terms of performance. Its computation would have taken between two and three months to compute even while using the full available resources. In our case, however, the analysis was constrained by the hardware resources available. Therefore, it was not possible within the scope of this work to calculate the metrics for the

IPv6 datasets after the basic work on the framework had been completed. Calculating all implemented metrics for IPv6 would have taken several months in itself, as can be seen in Table 4.5, and was therefore not possible. As already mentioned, the code has already been extensively optimized, and the metrics are calculated in parallel. The only remaining solution is to scale computation by allocating additional hardware resources (i.e., CPU cores) to increase calculation speed.

5.3 Pipeline Bottlenecks and Parallelization Trade-offs

In the basic processing pipeline, we tend to struggle with a different bottleneck. The large amount of output data (2.5 TB) from the IPv6 source data of the original internet scan that needs to be processed forces us to make compromises here. The entire pipeline is necessarily executed by a single CPU core or thread. This is due to the fact that the individual traceroute scans must be merged as described in section 3.2. As further discussed in this part, the processing requires that the temporary buckets, where paths are stored for merging, be held in memory. This holds true for every thread which is processing the data. Even using only a single thread, a peak memory usage (i.e., RAM and swap combined) of roughly 1.3 TB was observed. Thus, the processing pipeline faces a trade-off between foregoing parallelization and a massive increase in peak memory consumption that needs to be managed. Handling this solely on the software side is a particularly difficult task, so implementing effective parallelization here results in a massive increase of memory requirements. Thus, pipeline processing creates a tradeoff between parallelization and memory requirements that is not easily solved at the software level.

5.4 Metadata Integration and Lookup Performance

Due to the basic graph structure on which we based our work, unfortunately, the metadata had to be added during initial graph creation. This initially had a significant impact on performance and made it necessary to optimize the lookup process. For the geolocation data, the performance of the MaxMind DB used as the data source was sufficient. However, for the ASN data from CAIDA, which we had initially loaded into a binary search tree structure, the lookup for IPv6 addresses was quite slow. For a full IPv6 dataset, the basic processing pipeline runtime increased by about four weeks. Surprisingly, that was not the case for IPv4 datasets, though, since they only took about an hour to finish with ASN metadata assignment. Normally, a binary search tree is a particularly efficient structure, but it seemed as if the high number of IPv6 addresses and their significantly greater length with 128-bit length compared to IPv4 with 32-bit addresses led to significantly poorer performance.

However, this is only a guess, as we were unable to identify the exact cause despite conducting comparative benchmarks with other lookup methods. Ultimately, we opted to go with a special adaption of the binary search algorithm, which fixed the runtime issues for the

AS lookup procedure and brought the required time for processing a full IPv6 ASN dataset back down to roughly four days instead of four weeks.

5.5 Debugging and Testing Challenges

Generally speaking, debugging new processes and problems that have been identified has been and continues to be a major challenge. We already had a relatively complex process here with the original processing pipeline, and adding metadata and reduction did not improve the situation. The multi-stage processing, as now shown in figure 3.3 in the current status, is a closely linked process. It is possible to run the steps individually using cached data from previous runs. However, this option is highly error-prone and requires great caution when used. A major challenge here was that many of the sub-processes required a certain minimum amount of data in order to be tested meaningfully. This often made it difficult to construct small datasets for testing purposes. Since the data from the scan had to be combined into complete paths, it was also quite difficult to test with only parts of the data.

5.6 Graph Reduction Side Effects: Island Nodes

An additional challenge encountered was that there were some islands created in the reduced graphs. This means that some of the nodes were separated from the main graph by the reduction strategy used and the associated removal of edges. This problem only came to light during the final evaluation of the collected data. It is highly probable that this is an unintended side effect caused by the data reduction algorithm. However, the reduction itself is already a relatively complex process within the graph structure. Basically, for each node to be removed, it is necessary to additionally check whether it represents the last connection between two larger blocks of the entire graph. This makes it possible to avoid this side effect. However, a simple solution to this problem was not possible in the short term and can be implemented as part of a future continuation of the work.

Once the problem had been identified, we conducted a detailed analysis of its effects. The results are shown in table 5.1. In the CC2 dataset, this led to an unintended disconnection of a total of 2.32 % of the nodes in the reduced graph. In numbers, this corresponds to 30,893 of 1,383,676 nodes. For the ASN dataset, the figure is 5.32 %, which corresponds to 81,357 of a total of 1,528,755 nodes.

Dataset	Nodes Total	Nodes Disconnected	Quota
ASN	1528755	81357	5.32 %
CC2	1383676	30893	2.32 %

Table 5.1: Analysis of island nodes IPv4 datasets.

The situation with the IPv6 datasets shows a slightly different picture. In table 5.2 we can see that for the ASN dataset we have 38,721 island nodes of a total of 219,132,918 nodes. This amounts to a total share of 0.02 %. The CC2 dataset contains 10,725 island nodes of 120,264,902 in total. This corresponds to a share of 0.01 %.

If we now draw a conclusion, we can see that IPv4 datasets as a whole show a higher percentage of island formation. At first glance, this seems somewhat surprising, because overall the IPv6 dataset is much larger. Therefore, one can intuitively assume that these have at least an equal or even higher proportion of island nodes. However, we have already established in section 4.3 that the infrastructure hierarchy in IPv6 is significantly flatter in comparison. This is a plausible explanation for why we actually found fewer islands in the analysis compared to IPv6. Since we have many nodes with few edges in the direct comparison, the chance of unintentionally producing islands when removing them is significantly lower.

Dataset	Nodes Total	Nodes Disconnected	Quota
ASN	219132918	38721	0.02 %
CC2	120264902	10725	0.01 %

Table 5.2: Analysis of island nodes IPv6 datasets.

5.7 Visualization Trade-offs

When visualizing the data, we had to make a fundamental decision. Map servers use two fundamentally different approaches, which have a corresponding impact on the necessary hardware and performance. The classic approach is rasterized maps. With this approach, pre-rendered map tiles are provided on the server side. These are usually in formats such as PNG and JPEG. This means that the server requires powerful hardware (i.e., CPU and memory). The advantage here is that the requirements for the client PCs are minimal.

This actually argues in favor of using pre-rendered rasterized map tiles. However, we use a considerable amount of dynamic logic in the form of overlays for multiple overlapping nodes, for example. Features like this are basically only possible if the display is rendered on the client side. This feature, which is quite important for our purposes, and the dynamic zoom levels it enables, clearly tipped the scales in favor of the vertex tiles using client-side rendering we now use. This choice of client rendering improves scalability for exploratory analysis but also comes at the cost of heavier client requirements.

5.8 Limitations of the Underlying Data Structure

Another issue that has repeatedly caused us problems is that the underlying data structure originating from the work of Fodor, on which we have built upon, uses a directed graph [22].

Although this is not unusual for certain analyses in this context, it has repeatedly caused problems when applied to our purposes, as both directions of existing edges between nodes were needed in the course of our data processing because undirected edges accurately representation of internet connectivity. However, in the underlying graph, these are only stored one-sided. In order to be able to jump back from node B to node A when two nodes, A and B, are connected, complex adjustments had to be made in many places. This was just one example of the necessary adaptations, but we repeatedly encountered problems arising from the use of a classic directed graph, since the available framework did not allow the use of an undirected graph.

However, it was not possible to change this. Building on the finished graph from the previous work saved us a lot of basic work, but the existing processes also restricted us in many aspects when designing our additional work steps.

5.9 Summary and Outlook

In summary, this chapter outlined the key challenges encountered when implementing the proposed framework. These challenges range from computational and memory bottlenecks to methodological side effects in graph processing. Although the reduction approach was effective in theory, its application on the graph data was limited by hardware requirements, pipeline trade-offs, and the limitations of the underlying data structure. These observations show that the main challenges are found less in the reduction concept itself. They are focused on practical constraints imposed by data volume and processing infrastructure.

The findings presented here, when considered together, provide a realistic view of the current framework's strengths and weaknesses. The importance of balancing algorithmic design, available computational resources, and data representation choices when working with internet-scale graphs is also shown. The knowledge gained from these challenges forms the basis for general conclusions and possible future research directions, which are discussed in the last chapter.

6 Conclusions and Future work

The goal of this thesis was to find improved ways of making the enormous complexity of the internet easier to understand. The internet, as a graph, contains billions of nodes and edges, and while this scale is impressive, it makes meaningful analysis or visualization almost impossible without simplification. The central idea developed here was to combine geolocation data with autonomous system (AS) affiliation in order to reduce the internet graph while preserving its key structural properties.

The work resulted in a flexible framework for graph reduction that is able to merge nodes into larger groups, based on either their physical location or their organizational affiliation. Techniques such as edge pruning ensured that even though the graphs became smaller, the key connectivity patterns and relationships were still visible. In practice, this allowed us to retain important internet backbone structures while reducing much of the visual clutter that makes raw graphs difficult to interpret.

Beyond the methodology, this thesis carried out a detailed structural analysis of the reduced graphs. We showed that the important regions and central AS nodes remain connected even after reduction, and the resulting visualizations provide a new perspective on how geography and organizational boundaries shape the internet. These contributions demonstrate that meaningful reduction is possible when domain-specific knowledge, in this case, location and AS membership, is carefully integrated into the process.

Large-scale internet data, especially IPv6, proved extremely challenging from a computational perspective. Metric calculations such as betweenness centrality were simply too time-consuming to be useful, even on powerful hardware. Another obstacle was the data pipeline itself. Parallelization was limited by the enormous memory requirements needed to merge traceroute records into complete paths. We also observed unexpected side effects, such as the formation of ‘island nodes,’ where reduction inadvertently caused parts of the graph to become disconnected. Despite the small percentages involved, these results demonstrated the practical fragility of large-scale graph processing.

At first, technical issues like metadata lookup for IPv6 addresses caused portions of the processing pipeline to operate at unfeasible speeds. The complexity and interdependencies of the data turned debugging the multi-stage procedure into a challenging task. These drawbacks show that in addition to sophisticated algorithms, internet-scale analysis necessitates rigorous resource management and design trade-off considerations.

There are a number of ways that this work can be expanded in the future. Scalability remains the central issue. The existing obstacles in metric calculation and memory utilization can be addressed by distributing the pipeline among clusters or creating streaming techniques that can process data gradually. This would enable us to process in near real-time and handle entire IPv6 graphs with high efficiency.

The improvement of the reduction strategies themselves is a key step. In the current state, the aggregation of similar nodes can occasionally cause unintended disconnections within the graph. Further extending the used strategies to increase environmental awareness helps to prevent the creation of island nodes.

Additionally, graphs can be enhanced with additional kinds of metadata. Structural graphs become multifaceted tools when AS business relationships, traffic quantities, and latency or performance data are added. Future developments in visualization might include real-time dashboards, adaptive detail, or semantic overlays, transforming condensed internet graphs into useful tools for observation and decision-making.

Moving toward incremental or live graph analysis is, last but not least, one of the greatest opportunities. Large scans might be processed offline, but future systems could continuously integrate fresh data, reduce it in real time, and display updated visualizations. This approach shows promise in turning internet graph reduction a useful tool for network operators and organizations looking to plan for expansion, security, and resilience in addition to researchers.

In the end, this thesis shows both the promise and the challenges of trying to make sense of the internet at scale. We learned that meaningful simplification is not about discarding information, but about designing reductions that respect the internet's underlying structure—its geography, its organizational boundaries, and its backbone of critical connections. At the same time, we also learned that technical details, such as how data is stored, processed, and optimized, can make the difference between success and infeasibility.

While much work remains, the framework created here is a start in the right direction toward bridging the gap between unprocessed internet measurement data and maps of worldwide connectivity that can be used and understood. It can play a significant role in how we comprehend, keep an eye on, and fortify the internet in the years to come with additional development.

List of Figures

2.1 Core processing steps of internet graph analysis. 13
2.2 The goal of graph reduction is to identify a reduced (smaller) graph dataset that retains certain information from the original graph dataset. Source: [30] 18

3.1 Visual representation of merged traceroute paths through the graph. Source: [22] 24
3.2 Comparison of the two graph processing pipelines. Source: [22] 25
3.3 Modified graph processing pipeline with additional reduction step (new section marked with dotted box). Source: [22] 28
3.4 ASN reduction process, nodes that only have neighbors with identical AS numbers and the corresponding edges are removed. 29
3.5 CC2 reduction process, nodes that only have neighbors with identical country codes and the corresponding edges are removed. 29
3.6 Multiple nodes selection overlay example. 35
3.7 Default zoom map visuals example of Austria. 36
3.8 Further zoomed in map visuals example of Austria. 36
3.9 Node information overlay example. 37

4.1 Distribution of IPv4 betweenness centrality values for ASN and CC2 datasets. 45
4.2 Distribution of IPv4 closeness centrality values for ASN and CC2 datasets. . 46
4.3 Distribution of IPv4 degree centrality values for ASN and CC2 datasets. . . . 47
4.4 Distribution of IPv6 degree centrality values for ASN and CC2 datasets. . . . 48

List of Tables

2.1 Notations used in this paper. 11
2.2 Comparison of internet topology datasets: CAIDA, iPlane, M-Lab, RIPE NCC 15

3.1 Overview and description of the fields contained within the IPinfo database. 26
3.2 Overview of the technology stack used for the creation of the graphic map. 30
3.3 Example of a node and edge dataset exported in the GeoJSON format. . . . 33
3.4 Tippecanoe command parameters 34

4.1 IPv4 metadata assignment quota 41
4.2 IPv6 metadata assignment quota 42
4.3 IPv4 reduction strategy efficiency 42
4.4 IPv6 reduction strategy efficiency 43
4.5 Comparative overview of metrics runtime IPv4 / Ipv6 43
4.6 CC2 IPv4 betweenness centrality statistics 50
4.7 ASN IPv4 betweenness centrality statistics 51
4.8 CC2 IPv4 closeness centrality statistics 53
4.9 ASN IPv4 closeness centrality statistics 54
4.10 CC2 IPv4 degree centrality statistics 55
4.11 ASN IPv4 degree centrality statistics 56
4.12 CC2 IPv6 degree centrality statistics 57
4.13 ASN IPv6 degree centrality statistics 58

5.1 Analysis of island nodes IPv4 datasets. 61
5.2 Analysis of island nodes IPv6 datasets. 62

Bibliography

[1] T.S Aluko, O.J Olusanya, O.E Oloyede, and A.F Ebisin. Comparative analysis between internet protocol version 4 & 6 (ipv4 and ipv6). *International Journal of Scientific & Engineering Research*, 5(8):494–497, 2014. URL https://www.ijser.org/researchpaper/COMPARATIVE-ANALYSIS-OF-INTERNET-PROTOCOL-VERSION.pdf.

[2] Silvia Bartolucci, Fabio Caccioli, Francesco Caravelli, and Pierpaolo Vivo. Distribution of centrality measures on undirected random networks via the cavity method. *Proceedings of the National Academy of Sciences*, 121(40):e2403682121, 2024. doi: 10.1073/pnas.2403682121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2403682121.

[3] Robert Beverly. Yarrp'ing the internet: Randomized high-speed active topology discovery. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, page 413–420, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345262. doi: 10.1145/2987443.2987479. URL https://doi.org/10.1145/2987443.2987479.

[4] Yohan Boniface and contributors. umap: Create maps with openstreetmap layers, 2025. URL https://umap-project.org. Open-source project licensed under GNU AGPL v3.

[5] CAIDA. As rank: Autonomous system ranking and relationships. https://asrank.caida.org/, 2023. Accessed 2025-10-05.

[6] CAIDA. Archipelago (ark), 2025. URL https://www.caida.org/projects/ark/. Accessed: 2025-04-07.

[7] CAIDA. Ipv4 address space prefix probing dataset, 2025. URL https://catalog.caida.org/dataset/ark_ipv4_prefix_probing. Accessed: 2025-04-07.

[8] CAIDA. Ipv6 traceroute dataset, 2025. URL https://catalog.caida.org/dataset/ark_ipv6_traceroute. Accessed: 2025-04-07.

[9] CAIDA: Center for Applied Internet Data Analysis. Routeviews prefix to as mappings dataset (pfx2as) for ipv4 and ipv6, 2019. URL https://www.caida.org/catalog/datasets/routeviews-prefix2as/. Accessed: 2025-07-21.

[10] M. Abdullah Canbaz, Jay Thom, and Mehmet Hadi Gunes. Comparative analysis of internet topology data sets. In *2017 IEEE Conference on Computer Communications*

Workshops (INFOCOM WKSHPS), pages 635–640, 2017. doi: 10.1109/INFCOMW.2017.8116451.

[11] Yuhan Chen, Haojie Ye, Sanketh Vedula, Alex Bronstein, Ronald Dreslinski, Trevor Mudge, and Nishil Talati. Demystifying graph sparsification algorithms in graph properties preservation. *Proc. VLDB Endow.*, 17(3):427–440, November 2023. ISSN 2150-8097. doi: 10.14778/3632093.3632106. URL https://doi.org/10.14778/3632093.3632106.

[12] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org, 2017. URL https://www.openstreetmap.org. Accessed: 2025-04-07.

[13] Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. Ip geolocation using traceroute location propagation and ip range location interpolation. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 332–338, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383134. doi: 10.1145/3442442.3451888. URL https://doi.org/10.1145/3442442.3451888.

[14] Akhil Dhamdhere and kc Claffy. Internet topology evolution and the rise of hypergiants. In *ACM SIGCOMM Computer Communication Review*, volume 48, pages 65–76, 2018. doi: 10.1145/3230543.3230577.

[15] Docker, Inc. Docker documentation. https://docs.docker.com/, 2025. Accessed: 2025-07-14.

[16] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. Zmap: Fast internet-wide scanning and its security applications. In *Proceedings of the 22nd USENIX conference on Security*, pages 605–620. USENIX Association, 2013. URL https://zmap.io/paper.pdf.

[17] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. Ten years of zmap. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 139–148, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705922. doi: 10.1145/3646547.3689012. URL https://doi.org/10.1145/3646547.3689012.

[18] Junfeng Fang, Xinglin Li, Yongduo Sui, Yuan Gao, Guibin Zhang, Kun Wang, Xiang Wang, and Xiangnan He. Exgc: Bridging efficiency and explainability in graph condensation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 721–732, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645551. URL https://doi.org/10.1145/3589334.3645551.

[19] Felipe Schreiber Fernandes, Daniel Ratton Figueiredo, and Maximilien Dreveton. Clustering networks with node and edge attributes using bregman divergence. *Anais do XXXVII Concurso de Teses e Dissertações (CTD 2024)*, 2024. URL https://api.semanticscholar.org/CorpusID:271349328.

[20] Tobias Fiebig. Navigating network measurements: A review of results and services, April 2024. URL https://labs.ripe.net/author/tfiebig/navigating-network-measurements-a-review-of-results-and-services/. Accessed: 2025-04-07.

[21] Tobias Fiebig. Navigating network measurements: Research risks and reliability, April 2024. URL https://labs.ripe.net/author/tfiebig/navigating-network-measurements-research-risks-and-reliability/. Accessed: 2025-04-07.

[22] Thomas Bogdan Fodor. Analysing the internet as a graph. Diploma thesis, Technische Universität Wien, Vienna, Austria, 2023. URL https://repositum.tuwien.at/handle/20.500.12708/154357. Published at reposiTUm (TU Wien's institutional repository).

[23] Agostino Funel. The graph structure of the internet at the autonomous systems level during 10 years. *Journal of Computer and Communications*, 07:17–32, 01 2019. doi: 10.4236/jcc.2019.78003.

[24] Phillipa Gill, Christophe Diot, Lai Yi Ohlsen, Matt Mathis, and Stephen Soltesz. M-lab: user initiated internet data for the research community. *SIGCOMM Comput. Commun. Rev.*, 52(1):34–37, March 2022. ISSN 0146-4833. doi: 10.1145/3523230.3523236. URL https://doi.org/10.1145/3523230.3523236.

[25] Christos Gkantsidis, Milena Mihail, and Ellen W. Zegura. Spectral analysis of the internet topology. In *IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 3, pages 1727–1736 vol.3. IEEE, 2003. URL https://sites.cc.gatech.edu/fac/Ellen.Zegura/papers/SpectralAnalysis03Infocom.pdf.

[26] Sergio Gómez. *Centrality in Networks: Finding the Most Important Nodes*, pages 401–433. Springer International Publishing, Cham, 2019. ISBN 978-3-030-06222-4. doi: 10.1007/978-3-030-06222-4_8. URL https://doi.org/10.1007/978-3-030-06222-4_8.

[27] Mehmet H. Gunes and Kaya Sarac. Correlation of as topology and geographical properties of the internet. *Computer Networks*, 162:106858, 2019. doi: 10.1016/j.comnet.2019.106858.

[28] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier. Network topologies: inference, modeling, and generation. *IEEE Communications Surveys & Tutorials*, 10(2):48–69, 2008. doi: 10.1109/COMST.2008.4564479.

[29] Zunainah Hamid, Sharipah Daud, Intan Shafinaz Abd. Razak, and Nurzurawani Abd. Razak. A comparative study between ipv4 and ipv6. *ANP Journal of Social Science and Humanities*, 2(1):68–72, Feb. 2021. doi: 10.53797/anpjssh.v2i1.9.2021. URL https://journalarsvot.com/index.php/anp-jssh/article/view/48.

[30] Mohammad Hashemi, Shengbo Gong, Juntong Ni, Wenqi Fan, B. Aditya Prakash, and Wei Jin. A comprehensive survey on graph reduction: sparsification, coarsening, and condensation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/891. URL https://doi.org/10.24963/ijcai.2024/891.

[31] Yihua He, Georgos Siganos, and Michalis Faloutsos. *Internet Topology*, pages 4930–4947. Springer New York, New York, NY, 2009. ISBN 978-0-387-30440-3. doi: 10.1007/978-0-387-30440-3_293. URL https://doi.org/10.1007/978-0-387-30440-3_293.

[32] Javier Martín Hernández and Piet Van Mieghem. Metric list. Technical report, Delft University of Technology, Faculty of Electrical Engineering, Mathematics, and Computer Science, Nov 2011. URL https://www.nas.ewi.tudelft.nl/people/Piet/papers/TUDreport20111111_MetricList.pdf.

[33] Md Turab Hossain, Jesmin Binti, and Rayhan Uddin. A review paper on ipv4 and ipv6: A comprehensive survey. *American Journal of Computer Science and Technology*, 07: 170–175, 10 2024. doi: 10.11648/j.ajcst.20240704.14.

[34] Geoff Huston. The architecture of the internet: Autonomous systems and routing. *The Internet Protocol Journal*, 25(2):1–15, 2022. URL https://www.internetsociety.org/tutorials/architecture-internet/.

[35] Internet Engineering Task Force (IETF). Geojson specification, 2016. URL https://geojson.org/. RFC 7946.

[36] IPinfo. Ip to geolocation database developer resource, 2025. URL https://ipinfo.io/developers/ip-to-geolocation-database. Accessed: 2025-07-21.

[37] Mayur Jadhav and N. Pokale. Autonomous system dealings in the internet. In *Computer Networks and Information Technologies Second International Conference on Advances in Communication, Network, and Computing, CNC 2011, Bangalore, India, March 10-11, 2011. Proceedings*, pages 331–336, 03 2011. ISBN 978-3-642-19541-9. doi: 10.1007/978-3-642-19542-6_61.

[38] Steve Klabnik and Carol Nichols. *The Rust Programming Language*. No Starch Press, 2019. URL https://doc.rust-lang.org/book/.

[39] Manoj Kumar, Anurag Sharma, Shashwat Saxena, and Sandeep Kumar. Featured graph coarsening with similarity guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[40] David S. Lee and Jeffrey L. Kalb. Network topology analysis. Unlimited Release SAND2008-0069, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Jan 2008. URL https://digital.library.unt.edu/ark:/67531/metadc845229/m2/1/high_res_d/1028919.pdf.

[41] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009. doi: 10.1080/15427951.2009.10129127.

[42] Harsha Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane: an information plane for distributed services. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*, OSDI '06, page 26, USA, 2006. USENIX Association.

[43] Damien Magoni. Network Topology Analysis and Internet Modelling with Nem. *International Journal of Computers and Applications*, 27(4):252–259, 2005. doi: 10.2316/Journal.202.2005.4.202-1540. URL https://hal.science/hal-00344484.

[44] Markus Maier and Johanna Ullrich. In the loop: A measurement study of persistent routing loops on the ipv4/ipv6 internet. *Computer Networks*, 221:109500, 2023. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2022.109500. URL https://www.sciencedirect.com/science/article/pii/S1389128622005345.

[45] Matthew Malloy, Paul Barford, Enis Ceyhun Alp, Jonathan Koller, and Adria Jewell. Internet device graphs. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1913–1921, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098114. URL https://doi.org/10.1145/3097983.3098114.

[46] Mapbox. Tippecanoe: Build vector tilesets from large collections of geojson features. https://github.com/mapbox/tippecanoe, 2024. Accessed: 2025-04-07.

[47] *MapLibre GL JS Documentation*. MapLibre Project, 2024. URL https://maplibre.org/maplibre-gl-js/docs/. Accessed: 2025-04-07.

[48] MapTiler.com. *TileServer GL Documentation*, 2023. URL https://tileserver.readthedocs.io. Accessed: 2025-04-07, version 1.0.

[49] MaxMind Inc. Maxmind db file format specification. https://maxmind.github.io/MaxMind-DB/, 2025. URL https://maxmind.github.io/MaxMind-DB/. Version 2.0 specification.

[50] Measurement Lab. Contributing to m-lab's infrastructure is easier than ever, April 2024. URL https://www.measurementlab.net/blog/contributing-mlab-infrastructure/. Accessed: 2025-04-07.

[51] Measurement Lab. M-lab data, 2024. URL https://www.measurementlab.net/data/. Accessed: 2025-04-07.

[52] Simachew Mengiste, Ad Aertsen, and Arvind Kumar. Effect of edge pruning on structural controllability and observability of complex networks. *Scientific reports*, 5:18145, 12 2015. doi: 10.1038/srep18145.

[53] Shiri Morshtein, Ran Ettinger, and Shmuel Tyszberowicz. Verifying time complexity of binary search using dafny. *Electronic Proceedings in Theoretical Computer Science*, 338:68–81, 08 2021. doi: 10.4204/EPTCS.338.9.

[54] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. doi: 10.1073/pnas.0601602103. URL https://www.pnas.org/doi/abs/10.1073/pnas.0601602103.

[55] Chien-Chun Ni, Yu-Yao Lin, Feng Luo, and Jie Gao. Community detection on networks with ricci flow. *Scientific Reports*, 9(1):9984, July 2019. doi: 10.1038/s41598-019-46380-9. URL https://doi.org/10.1038/s41598-019-46380-9.

[56] Abdullah Yasin Nur and Mehmet Engin Tozal. Identifying critical autonomous systems in the internet. *The Journal of Supercomputing*, 74(10):4965–4985, oct 2018. ISSN 0920-8542. doi: 10.1007/s11227-018-2336-3. URL https://doi.org/10.1007/s11227-018-2336-3.

[57] Abdullah Yasin Nur and Mehmet Engin Tozal. Geography and routing in the internet. *ACM Trans. Spatial Algorithms Syst.*, 4(4), September 2018. ISSN 2374-0353. doi: 10.1145/3239162. URL https://doi.org/10.1145/3239162.

[58] Abdullah Yasin Nur and Mehmet Engin Tozal. Inferring internet topology at point of presence level. *2024 IEEE International Systems Conference (SysCon)*, pages 1–7, 2024. doi: 10.1109/SysCon61195.2024.10553435. URL https://api.semanticscholar.org/CorpusID:270567829.

[59] Milena Oehlers and Benjamin Fabian. Graph metrics for network robustness, a survey. *Mathematics*, 9(8), 2021. ISSN 2227-7390. doi: 10.3390/math9080895. URL https://www.mdpi.com/2227-7390/9/8/895.

[60] Lai Yi Ohlsen. Measure your network openly with m-lab, March 2024. URL https://ripe88.ripe.net/presentations/34-2024-03_-RIPE-88-Measure-Your-Network-Openly-with-M-Lab_v2.pdf. Accessed: 2025-04-07, presented at RIPE 88.

[61] Perplexity AI. Perplexity ai: Ai-powered research assistant tool. https://www.perplexity.ai, 2025. Accessed: 2025-10-01.

[62] Ilia Poese, Stefan Uhlig, Mohamed Ali Kaafar, Bruno Donnet, and Bué Gueye. Ip geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2):53–56, 2011. doi: 10.1145/1956197.1956207.

[63] *Python Programming Language*. Python Software Foundation, 2023. URL https://www.python.org/. Accessed: 2025-07-14, version 3.x.

[64] Nematullo Rahmatov, Faisal Saeed, and Anand Paul. Analysis of the vulnerability estimation and neighbor value prediction in autonomous systems. *Scientific Reports*, 12(1):9457, 6 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-13613-3. URL https://doi.org/10.1038/s41598-022-13613-3.

[65] Ben Julian Riegel, Simon Bauer, and Johannes Zirngibl. Measurement lab: Analyzing internet speed tests and the influence of transport layer security. Technical Report NET-2020-11-1, Technical University of Munich, 2020. URL https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2020-11-1/NET-2020-11-1_17.pdf.

[66] RIPE NCC. Ripe ncc response to the penetration test report from radically open security, July 2021. URL https://www.ripe.net/media/documents/RIPE_NCC_Response_to_the_Penetration_Test_Report_from_Radically_Open_Security.pdf. Accessed: 2025-04-07.

[67] RIPE NCC. Analyse, 2025. URL https://www.ripe.net/analyse/. Accessed: 2025-04-07.

[68] RIPE NCC. Ripe database, 2025. URL https://www.ripe.net/manage-ips-and-asns/db/. Accessed: 2025-04-07.

[69] Atena Shiranzaei and Rafiqul Zaman Khan. A comparative study on ipv4 and ipv6. *International Journal of Advanced Information Science and Technology (IJAIST)*, 4(1):9–16, 2015. doi: 10.15693/ijaist/2015.v4i1.9-16.

[70] Keith Stouffer, Michael Pease, CheeYee Tang, Timothy Zimmerman, Victoria Pillitteri, Suzanne Lightman, Adam Hahn, Stephanie Saravia, Aslam Sherule, and Michael Thompson. Guide to operational technology (ot) security, September 2023. URL https://doi.org/10.6028/NIST.SP.800-82r3.

[71] MapTiler team. Maptiler - global map service provider and open source software developer, 2025. URL https://www.maptiler.com/. Accessed: 2025-04-07, includes OpenMapTiles and other open-source mapping tools.

[72] Mehmet Engin Tozal. The internet: A system of interconnected autonomous systems. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–8, 2016. doi: 10.1109/SYSCON.2016.7490628.

[73] Fabien Viger, Alain Barrat, Luca Dall'Asta, Cun-Hui Zhang, and Eric Kolaczyk. Network inference from traceroute measurements: Internet topology 'species'. *Computing Research Repository - CORR*, 10 2005.

[74] Xiangrong Wang, Evangelos Pournaras, Robert E. Kooij, and Piet Van Mieghem. Improving robustness of complex networks via the effective graph resistance. *The European Physical Journal B*, 87(9):221, September 2014. doi: 10.1140/epjb/e2014-50276-0. URL https://doi.org/10.1140/epjb/e2014-50276-0.

[75] Kun Xu, Ding Luo, and Yifan Chen. Graph reduction techniques for scalable network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):566–576, 2020. doi: 10.1109/TVCG.2019.2934533.

[76] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. doi: 10.1007/s10115-013-0703-9.

[77] Soon-Hyung Yook, Hawoong Jeong, and Albert-László Barabási. Modeling the internet's large-scale topology. *Proceedings of the National Academy of Sciences*, 99(21):13382–13386, 2002. doi: 10.1073/pnas.172501399. URL `https://www.pnas.org/doi/abs/10.1073/pnas.172501399`.

[78] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, January 2005. ISSN 0146-4833. doi: 10.1145/1052812.1052825. URL `https://doi.org/10.1145/1052812.1052825`.

[79] Jie Zhang, Sheng He, and Meng Zhou. A review of network visualization methods for big graphs. *IEEE Access*, 9:128815–128828, 2021. doi: 10.1109/ACCESS.2021.3118649.

[80] Zhen Zhong, Aisha Elmokashfi, and Christos Dovrolis. Characterizing the internet as-level topology. *IEEE/ACM Transactions on Networking*, 30(2):713–727, 2022. doi: 10.1109/TNET.2022.3145747.

[81] Fang Zhou, Sebastien Malher, and Hannu Toivonen. Network simplification with minimal loss of connectivity. In *2010 IEEE International Conference on Data Mining*, pages 659–668, 2010. doi: 10.1109/ICDM.2010.133.