SAPIENZA
UNIVERSITÀ DI ROMA

BGP Data Analysis: Exploring Solutions for Autonomous Systems Relationships Inference

Faculty of Information Engineering, Informatics, and Statistics
Master's Degree in Cybersecurity

Alessio Mobilia
ID number 1896932

Advisor
Prof. Francesca Cuomo

Co-Advisor
Dr. Flavio Luciani

Academic Year 2022/2023

BGP Data Analysis: Exploring Solutions for Autonomous Systems Relationships Inference

Master's Thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: info@alessiomobilia.com

Talk is cheap.
Show me the code.

Linus Torvalds

Abstract

The vast and intricate web of the Internet is a marvel of modern technology, connecting us to people and information from all corners of the globe, allowing the exchange of confidential information, sensitive data, and pictures of adorable cats. This connectivity comes with the need for security, reliability, and speed - crucial components that require a deep understanding of the relationships between the multiple Autonomous Systems (AS) that make up the Internet. Unfortunately, these relationships are often private agreements, making their analysis a challenging task that requires inference. To tackle this issue, I have studied three inference protocols in detail - ASRank, Problink, and Toposcope - and developed a comprehensive framework to create datasets for evaluating relationship inference algorithms. With this framework, we can better understand the complex relationships between ASes and let's work towards creating a safer and more reliable Internet for everyone.

Acknowledgments

I am grateful to all the people who helped me achieve this accomplishment.

First and foremost, I would like to thank Professor Cuomo for allowing me to delve into computer networking and enhance my knowledge in this field. I also extend my gratitude to Pietro and Stefano for assisting me in developing this thesis.

A special mention goes to Flavio Luciani, the CTO of Namex, who gave me a practical understanding of BGP networking and wrote the book that set me on the path to this thesis.

I want to thank my family for their inexhaustible support in pursuing my dreams, even if it meant being away from them. Thanks to my mother, who introduced me to the wonders of computer science at a young age, and my father, who taught me to be curious and take things apart to understand how they work.

I also want to thank Jessica for being there for me, far from home, and listening to my complaints. I express my gratitude to my sister Emy, my almost-brother Marco, my colleague and friend Andrea, my grandparents, aunts, all my friends, and all the people who helped and loved me.

Lastly, thank you, the reader, for taking the time to read my work.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

The entire internet infrastructure relies on a protocol called BGP, which is quite old. All the information we send over the internet, including personal and private data, is forwarded using this protocol. It's essential to work towards making BGP more secure and efficient.

In order to achieve our objective, we need to gather more valuable information about the topology of the internet. With this information, we can detect issues such as route leaks, route hijacking, and misconfiguration.
This knowledge also helps us improve the configuration of BGP routers to facilitate faster interconnections and reduce costs for internet and content providers.

More accurate data about internet topology and the relationships between Autonomous Systems (ASes) will undoubtedly enhance research efforts in the network field.

1.2 Research Objectives

This thesis aims to evaluate the effectiveness of various Autonomous System (AS) relationships inference algorithms and identify the conditions under which they perform optimally.
Furthermore, this study aims to develop a technique for generating validation datasets that can facilitate the analysis and validation of these algorithms. The absence of a standardized approach to validating AS relationships inference outcomes poses a significant challenge, given that AS private agreements typically govern relationships. Nonetheless, this thesis endeavours to overcome this challenge and provide valuable insights into the performance of AS relationships inference algorithms.

1.3 Thesis Organization

The following is an outline of the thesis:

Chapter 2 provides an introduction and an overview of internet infrastructure, including how they are connected. The chapter highlights different types of relationships identified by researchers and why it is essential to infer them accurately.

Chapter 3 gives an overview of the BGP protocol, including how it works and its basic concepts. It also explains how this protocol concretely brings to practice the relationships between ASes.

Chapter 4 discusses BGP's security problems by showing actual examples. It also outlines the security measures used to secure this protocol.

Chapter 5 covers the source of information available for measuring and analyzing internet infrastructures, along with some valuable tools.

Chapter 6 briefly explains the inference algorithms used in the analysis.

Chapter 7 is the core of the thesis, discussing the environment in which the analysis was conducted, how algorithms were executed, how data was organized, and how it was visualized with an interactive dashboard. The chapter also describes the framework developed to generate the validation dataset, the comparison, and the validation results.

Chapter 8 shares thoughts about possible future work based on the analysis.

Chapter 9 concludes the thesis by summarizing the work done.

Chapter 2

Internet Infrastructure Overview

The Internet, often called a "network of networks", comprises many hosts connected through links and routers. There are thousands of Administrative Areas on the Internet, each with a single or several autonomous systems (ASes) within them. At the time of writing, in August 2023, there are about 74796 ASes detected [21] in Routing System.

In order to exchange information, autonomous systems use a protocol known as BGP, an Exterior Gateway Protocol that will be discussed further in the upcoming chapter. Instead, within an Autonomous System, information is routed through an Internal Gateway Protocol of the AS's choosing, such as IGRP, OSPF, or RIP.

2.1 Autonomous Systems

In computer networking, an autonomous system (AS) refers to a sizable network, or collection of networks, governed by a cohesive routing policy. It's worth highlight that every device or computer that connects to the Internet is invariably linked to an AS.

ASes collaborate using the Border Gateway Protocol (BGP) to share routing information and guarantee global accessibility. The links between ASes are affected by business agreements that dictate traffic exchange's economic and technical aspects. Therefore, BGP policies mirror the business connections between neighbouring ASes. In general, AS pairs can have either customer-provider or peer-to-peer relationships. Providers offer Internet connectivity services to their clients, while peers provide connectivity between their respective customers.

Every Autonomous System (AS) is identified on the Internet by a unique AS number (ASN). These ASNs can be 16-bit numbers between 1 and 65534 or 32-bit numbers between 131072 and 4294967294. However, not all ASNs are assigned, and not all the ones that are assigned are used.
The allocation of Autonomous System Numbers (ASNs) to Local Internet Registries (LIRs) and end-user organisations is carried out by Regional Internet Registries (RIRs). The RIRs receive these numbers in blocks from the Internet Assigned

Numbers Authority (IANA) and are subsequently distributed to LIRs and end-user organisations. This process ensures the efficient and effective management of Internet resources.

Each AS has its own set of routers and routing policies, which allow it to exchange traffic with remote hosts. Routers within an AS have extensive knowledge of the topology within their own AS but only limited information about the reachability of other ASes.

ASes connect through dedicated point-to-point links or public Internet exchange points like Namex [67] or MIX [64]. IXPs usually have a shared medium that connects routers from various ASes, but physical connectivity at the IXP doesn't mean that every AS exchanges traffic with each other.

Internet Service Providers (ISPs) are commonly distinguished into tiers, and Geoff Huston's classification [45] is widely recognised.

The category consists of three primary tiers:

Tier 1 ASes are the backbone of the Internet and are often called backbone Internet providers. A Tier 1 autonomous system exchanges Internet traffic with other Tier 1 providers non-commercial via private settlement-free peering interconnections. The collective group of these autonomous systems is known as the Default Free Zone (DFZ).

Tier 2 Internet Service Providers (ISPs) utilise a combination of paid transit via Tier 1 ISPs and peering with other Tier 2 ISPs to deliver Internet traffic to end customers through Tier 1 providers. These providers are typically regional or national in scope.

Tier 3 ISPs purchase Internet transit exclusively and focus on delivering Internet access to end customers. The focus of Tier 3 providers is on local business and consumer market conditions.

Autonomous Systems (AS) can be distinguished into two primary categories: single-homed and multi-homed. Single-homed AS is defined as an AS with only one connection to a different AS, which can be either single or redundant. Multi-homed AS, conversely, are characterised by having one or multiple connections to several ASes. [58, 102]

2.2 Types of Relationships in Internet Connectivity

Researchers have been studying the issue of deducing the various types of relationships between ASes using publicly available BGP routing data for more than 20 years. These relationship inferences have many applications and research areas, including detecting network congestion, identifying malicious ASes, and deploying incentive-compatible BGP security.

ASes establish relationships based on contracts that outline pricing and traffic exchange agreements between domains.

There are two main types of relationships between Autonomous Systems (AS): customer-provider (c2p) and settlement-free peering (p2p). In a c2p relationship, the customer AS pays the provider AS for the ability to connect to and from the rest of the Internet [94].
In contrast, a p2p relationship involves two networks agreeing to exchange traffic without any associated fee, as long as it is destined for prefixes they or their customers own.

AS relationships can be complex and hybrid, falling between c2p and p2p. This can happen when two ASes have multiple contractual agreements, with one for each geographical region where an interconnection exists. Sibling relationships also exist between different ASes owned by the same organisation, allowing them to exchange traffic without cost or routing restrictions [48].

We can depict the AS relationship as an annotated graph based on Gao's work [35]. This graph is partially directed and comprises nodes that signify ASes, with edges classified as provider-to-customer, customer-to-provider, peer-to-peer, and sibling-to-sibling. Only edges that link providers and customers are directed. If we move from a provider to a customer through an edge, we refer to it as a provider-to-customer edge. If we move from a customer to a provider through an edge, we refer to it as a customer-to-provider edge. When two ASes have a peering relationship, we call the edge linking them a peer-to-peer edge. Lastly, the edge between two ASes with a sibling relationship is a sibling-to-sibling edge.
Formally we indicate the AS graph as $G = (V, E$where The vertex V consists of ASes, and the edge E consists of pairs of ASes that exchange traffic with each other.

These are the rules that govern BGP export policies for AS relationships:

When exchanging routing information with a provider , an AS can export its own and customer's routes, but usually not its provider or peer routes.

When exchanging routing information with a customer , an AS can export its own, customer's, and provider's or peer's routes.

When exchanging routing information with a peer , an AS can export its own and customer's routes, but usually not its provider or peer routes.

When exchanging routing information with a sibling , an AS can export its own, customer's, and provider's or peer's routes.

It is essential to note that most BGP paths are valley-free in AS relationships. This means a path will consist of zero or more c2p links, followed by zero or one peering link, and then zero or more p2c links. In other words, the traffic between two peers or providers should not pass through a client. This assumption is based on the economic incentives determining traffic exchange between ASes. It is not in an AS's best interest to intentionally advertise routes learned from a peer or provider

to another peer or provider because this "free transit" increases infrastructure costs without providing any financial benefit [48].

Gao and colleagues [35, 106, 94, 28, 27], noted that providers tend to have a greater node degree than customers, whereas peers typically have similar degrees. Node degree, as precedently defined, refers to the number of neighbouring autonomous systems (AS) that an AS is directly connected to, regardless of whether those neighbours are providers, peers, or customers.
Formally, we can define the degree of an AS u as $D(u) = |v|(u, v) \in E|$
Unfortunately, Willinger et al. [104] found that node degree is significantly affected by the limited placement of vantage points adjacent to peer-to-peer AS links, which results in only a subset of the complete Internet topology being available in the topological data.

2.3 Importance of Discovering Accurate AS's Relationships

Understanding Autonomous System (AS) relationships is crucial as they represent the business agreements between competing entities that dictate how traffic is routed between ASes. These agreements can lead to longer AS paths and less efficient routing on the Internet.
It is vital to accurately model AS relationships for realistic simulation studies of Internet protocols. Neglecting to consider AS relationships can result in different simulation outcomes compared to simulations that do consider these relationships based on known relationships between Internet Service Providers (ISPs) [28].

Relationship inferences have a variety of applications in different fields of research. Some of these include identifying network congestion [26], detecting malicious Autonomous Systems (ASes) [51, 23], deploying security mechanisms for BGP [22], safeguarding the anonymity of data [72], optimising video streaming [31], understanding the effects of public policy proposals on Internet governance [55] and selecting BGP routing paths [25] .

ASwatch [51] can be an example of an application of AS Relationship inference to identify malicious ASes by monitoring their control plane behaviour. It is important to accurately model AS relationships to ensure the success of such systems.
By using its knowledge of business relationships with other ASes, an AS can detect potential route leaks from its customer ASes by verifying if incoming route announcements are valley-free [63].

Moreover, the AS relationships can be leveraged for collaborative fault detection, such as identifying spamming. Heaberlen et al. [40] also utilised the trust and business relationships between ASes to identify routing anomalies.
A more accurate inferring of AS relationships can, undoubtedly, improve the results of all this field of applications improving internet security end academic research.

Chapter 3

BGP Protocol Overview

In 1989, during an Internet Engineering Task Force (IETF) conference, Cisco's Kirk Lougheed and IBM's Yakov Rekhter brainstormed a new routing protocol. They scribbled their ideas on napkins, and thus, the Border Gateway Protocol (BGP), also known as the "Two Napkin Protocol," was born [99].

BGP remains an essential component of the internet, which has grown from 80 thousand hosts in 1989 to over one billion hosts today. Over time, BGP's capabilities have expanded to include carrying routes for Multicast, IPv6, VPNs, and other data. In short, BGP plays a critical role in enabling the internet to transmit vast amounts of data quickly and efficiently .

When two Internet Service Providers (ISPs) connect, they use BGP to exchange routing information. The ISPs worldwide collectively exchange the Internet's routing table through BGP. Enterprises can also use BGP to exchange routing information with one or more ISPs, enabling their routers to learn Internet routes.

The BGP version currently in use is number 4, which offers a range of notable features. Firstly, it functions as a Path Vector routing protocol, which operates similarly to a Distance Vector protocol that measures distances in hops. BGP routers, instead, measure it in numbers of Autonomous Systems (ASes).

Additionally, It supports CIDR (Classless Inter-Domain Routing) to determine the best paths through a complex selection process based on various metrics.

It allows for the creation of routing policies in both incoming and outgoing traffic due to different metrics, and the exchange of routing information is reliable and accomplished through TCP connections.

Finally, BGP only updates in response to triggered events, making it an efficient protocol.

It is worth noting that BGP transcends traditional routing protocols, as it functions as a routing policy enforcement protocol.

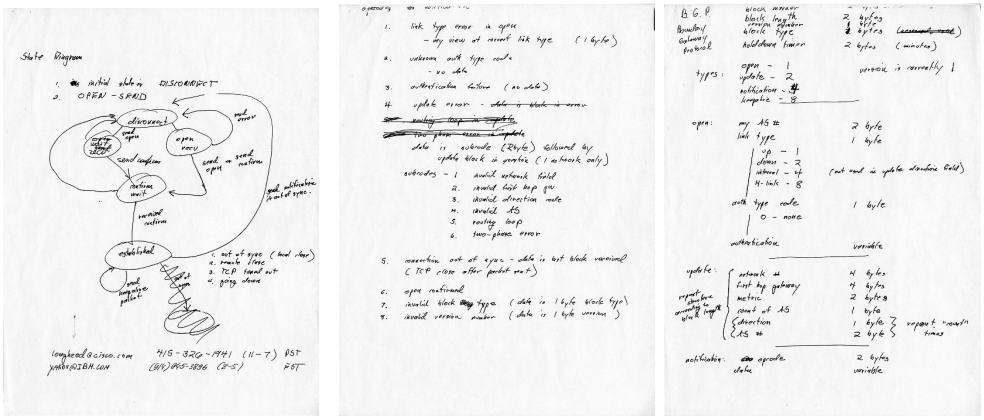


Figure 3.1. The Two Napkin Protocol from Cisco Archive [18]

Its designers emphasised scalability in exchanging large quantities of IP prefixes over other aspects, such as convergence speed or load balancing.
This approach has proven successful, as BGP can now manage the exchange of routing information for hundreds of thousands of IP prefixes across routers operating in large IP networks.

The information in this chapter is mainly based on CCNP Route 300-101 [102], BGP: From Theory to Practice [58], and BGP: Building Reliable Networks with the Border Gateway Protocol [9].

3.1 BGP Basics and Concepts

BGP uses TCP on port 179 when communicating with its neighbours, which is uncommon among routing protocols. Most routing protocols either run on top of IP or use UDP, allowing broadcasts or multicasts to find neighbouring routers. However, BGP does not require this neighbor-discovery feature. Using TCP, BGP avoids the need for many transport protocol functionality, such as fragmentation, sequencing, and data retransmission.

As previously stated, BGP functions as a Path Vector protocol. BGP announcements comprise an attribute named *AS_PATH*, an ordered list of AS (Autonomous System) traversed by the announcement. The path vector indicates the AS traversed to reach a prefix.

A router may receive multiple announcements for the same prefix but only choose one path to propagate to other routers via BGP. The metrics configured by the operators determine the best path.

BGP offers a routing policy that can be tailored to the specific needs of network operators. The policy can be customised through two main tools: announce filtering and BGP attribute manipulation. With announce filtering, a router can choose which announcements to accept and distribute, using either inbound or outbound

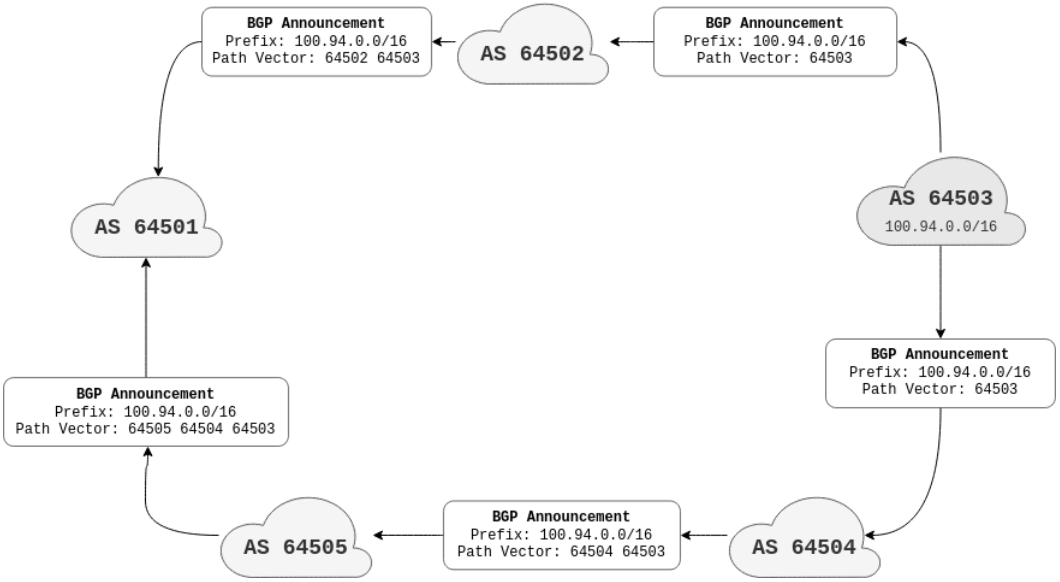


Figure 3.2. Example on how BGP announcement propagates, and ho AS_PATH is used. In this case, the AS 6453 announce his addresses

filtering. Routing policies are highly effective, as they enable the selection of primary and backup AS and load balancing. Although implementing these policies can be complex, Cisco and Juniper provide helpful tools to simplify the process.

3.1.1 BGP Session and Messages

When BGP neighbours establish a TCP session, they exchange BGP information through "messages." Each message starts with a header, followed by the message's content.

Marker	Lenght	Type	Message contents
16 bytes	2 bytes	1 byte	0 - 4077 bytes

The *marker* typically contains all 1s and ensures the sender and receiver are synchronised. If the receiver detects an unexpected value in the *marker* field, it sends back an error indication and closes the connection because something must have gone wrong. The *length* field displays the length of the BGP message, which can be as short as 19 bytes (just a header with no message) or as long as 4,096 bytes. The *type* specifies the message's purpose: *open* (1), *update* (2), *notification* (3), or *keepalive* (4).

Open Messages

Both sides send an *open message* immediately after the established TCP session. The open message conveys essential information about the BGP speaker's configuration and abilities.

Version	My AS	Hold Time	Identifier	Parlen	Optional Parameters
1 byte	2 bytes	2 bytes	4 bytes	1 byte	0-255 bytes

In BGP communication, the first field indicates the version, usually 4.
The following field shows the sender's AS number. The *hold time* specifies the maximum number of seconds of idle time allowed before the session is terminated due to a timeout. The routers take the lower hold time of both open messages, considering the minimum hold time is three seconds. A value of zero implies that the session would not time out.
The *identifier* field contains one of the BGP speaker's IP addresses, which should be the same for all BGP sessions. The optional parameter length field *parlen* indicates the absence or length of an optional parameter field. If there are optional parameters, they are preceded by a one-byte parameter type and a one-byte parameter length. The *optional parameters* field negotiates authentication and extended capabilities, such as multiprotocol extensions and route refresh.

If the open message contents are satisfactory, the router responds with a keepalive message. It sends a copy of the BGP routing table (depending on the configured policies for this peer) using update messages.
After completion, the router only sends periodic keepalive messages and incremental updates if there are any changes in the routing table.

Update Messages

The update message can list withdrawn and new routes, but it can include one or both options.

UR length	Withdrawn routes	PA length	Path attributes	NLRI
2 bytes	Variable	2 bytes	Variable	Variable

The *UR length* field stands for unfeasible routes and specifies the length of the *withdrawn routes* field. A value of zero means that this field is absent. Similarly, the *path attributes length* field and the *path attributes* field function similarly. As previously announced, the *withdrawn routes* field lists all routes no longer reachable. When attributes change, there is no need to withdraw a route explicitly. An *update message* with the new attributes and matching *NLRI* (Network Layer Reachability Information) is sufficient.

Each withdrawn route includes a length field that indicates the prefix length (in bits) and sufficient bytes to hold the prefix. The path attributes begin with a byte containing attribute flags and a second byte indicating the attribute type. The attribute flags consist of four bits, each with a specific purpose. The first bit, optional, determines whether the attribute is well-known or optional. If the bit is set to 0, then the attribute is well-known, meaning all BGP routers must recognise it. If set to 1, the attribute is optional.

The second bit, *transitive*, determines whether the attribute is transitive or well-known. If the bit is set to 0, then the attribute is nontransitive. If set to 1, the attribute is transitive or well-known.

The third bit, the *partial bit*, determines whether the information in the optional transitive attribute is partial. If the bit is set to 0, the attribute is complete, nontransitive, or well-known. If set to 1, the attribute is partial, meaning the information in the optional transitive attribute may not have been processed as desired at all previous hops.

The fourth bit, the *extended length* bit, determines the length of the attribute length field. If the bit is set to 0, the attribute length field is one byte. If set to 1, the attribute length field is two bytes.

The remaining bits of the attribute flags byte are not used. The values and interpretation of the path attribute field are determined by the path attribute type, which includes *Origin*, *AS path*, *Next hop*, *Multi Exit Discriminator*, and *Local Preference*, each with its type code.

When a router receives multiple routes from different BGP neighbours, it will utilise path attributes to determine the most optimal route to a destination network. This is explained in more detail later. The optional, transitive, and well-known permutations allow new path attributes to be added to BGP so that existing implementations can handle them without knowing their specific meaning.

Notification and Keepalive Messages

A *notification message* will be sent out if a fatal error occurs, and the TCP connection will be terminated. This message includes a one-byte error code and a one-byte error subcode and may contain optional data.
Keepalive messages are transmitted when no other activity is detected on the connection to prevent the hold timer from expiring. These messages only contain the BGP header with the type field set to 4 and no additional data.

3.1.2 BGP States

The BGP RFC outlines various states a session may be in and a state transition diagram known as the BGP "finite state machine". The state of the BGP session

determines the behaviour of a router. The MIB module for BGP has an SNMP trap message, which can be sent when a session transitions from a "higher" state to a "lower" state.

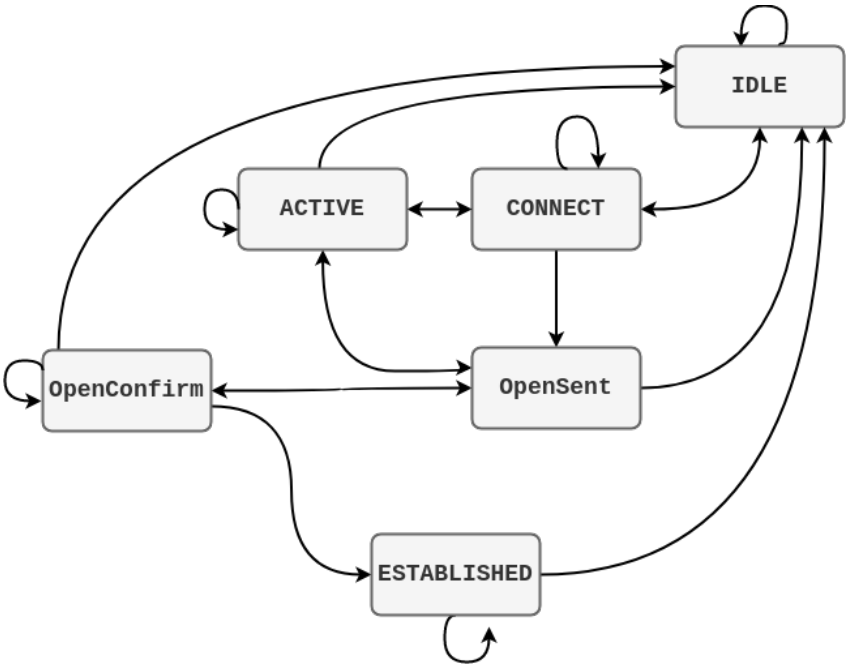


Figure 3.3. The BGP state machine

These states are as follows:

Idle: The router is not attempting to establish a BGP session. If a neighbour were to try to create a session, the TCP connection would be refused. The router waits for a "start" event, such as the user enabling BGP, adding a neighbour, or when an interface comes up.

Connect: The router waits for its own TCP session establishment attempt to complete and listens for incoming TCP sessions.

Active: BGP is waiting for a TCP session.

OpenSent: The open message has been sent, but the neighbour still needs to receive an open message.

OpenConfirm: The open message from the neighbour has been received, but the initial keepalive message that completes the BGP session setup phase has yet to be received.

Established: The initial keepalive message has been received, and the session is now ready to transmit update, keepalive, and notification messages.

3.1.3 Propagation of BGP Routes

When a BGP router receives a new route through a BGP update message, it first checks all incoming filters defined for the BGP session. If one of the filters does not allow the route, it is ignored. If the route is allowed, it is inserted into the BGP table and compared to other routes in the BGP table with the same destination prefix (NLRI).
The BGP route-selection algorithm is then executed, and if the new route is considered the best route, it replaces the old best route in the routing table. The old best route is then removed and revoked in BGP updates to all neighbours who have received a copy. Finally, if the filters configured for the neighbour allow it, the new best route is propagated to BGP neighbours in external ASes.
Routers Propagate the new best route to BGP neighbours in the local AS if that route was not received from another BGP neighbour in the local AS (There is usually no filtering between BGP neighbours in the same AS).

3.1.4 How BGP Selects Routes

To withstand network outages, BGP networks typically connect to multiple networks. This enables access to numerous destinations through two or more networks. Consequently, BGP requires selecting the optimal route from the available routes offered by different neighbours. To do route selection, BGP speakers communicate various attributes to each other, which may or may not have an impact on the process.
The most critical of these attributes include:

Local Preference The Local Preference is a value that belongs to an AS and is shared through intra-AS BGP sessions. BGP will always prioritise the route with the highest Local Preference. By default, all routes on Cisco routers have a Local Preference of 100.

AS path The AS path is a list of all the AS numbers between the local router and the source of the route. For non-local routes, it includes the source AS number but not the local AS number. The path serves several purposes. Firstly, it prevents routing loops by having the router ignore any routes it receives from a neighbouring AS that contains its own AS number. Secondly, the path allows the router to make policy decisions based on the presence of certain ASes in the path. Lastly, routes with a shorter AS path are preferred over routes with a longer AS path.

Next Hop The next hop attribute refers to the IP address of the router within the remote AS that will receive packets for the current route.

MED The Multi Exit Discriminator (MED) is a feature in BGP that helps neighbouring ASes determine which connection is preferred when there are multiple connections. In earlier versions of BGP, this feature was referred to as "Inter-AS Metric," and it may still appear as "metric" in some instances. Although the MED is typically used as a tie-breaker in the route selection process, it is possible to configure routers to compare MEDs between routes from different

ASes. The preferred route is the one with the lowest Multi Exit Discriminator metric.

Origin The Origin attribute indicates where the BGP announcement comes from, whether it's from an IGP, the EGP protocol, or other incomplete means. Despite being mandatory, this attribute doesn't serve any practical function.

Community Routes can consist of multiple communities, each represented by a 32-bit value. These values are usually expressed in the "AS number: value" format, such as "701:120", where 701 represents the AS number and 120 represents a specific value within that AS. Although communities do not directly affect the route selection process, they may activate customised actions determined by the user.

Certain attributes function differently when transmitted over BGP sessions within the same AS (internal BGP or *iBGP*) compared to BGP sessions with routers in other ASes (external BGP or *eBGP*). The Local Preference attribute is exclusively communicated over *iBGP*, while the next hop and MED attributes remain unaltered when transmitted over *iBGP*.

The route-selection algorithm

Router vendors have chosen not to implement a user-defined route evaluation procedure as outlined in the RFC. Instead, they utilise a decision-making algorithm that involves several steps, taking into account one or more variables. These variables can be modified by the user.

The general BGP route-selection algorithm can be summarised as follows:

- Apply user-configured policies to adjust variables.
- Select the route with the highest Local Preference.
- Select the route with the shortest AS path.
- Select the route with the lowest MED metric if the routes were received from the same AS or if the router is always configured to compare MEDs.
- Apply the remaining tie-breaking rules.

While Cisco routers include additional steps in the route selection process, they typically do not alter the algorithm's outcome.

All routers within an AS don't need to agree on the best route selection. Two border routers commonly believe their link to an external AS is the best route for a specific destination. In the event of a tie, each router selects its exterior route as the best. Packets arriving at Router A will be routed via ISP 1, while packets arriving at Router B will be routed via ISP 2. This is not problematic and can even

be desirable for large geographic ASes.

In the event of a tie, the following tie-breaking rules are applied, assuming no policies are in place to favour one route over another:

- Select the route with the lowest MED metric from the routes received from a single AS.
- Select the route with the lowest cost or metric for the next hop address in the interior routing protocol.
- Select the route advertised by the external BGP peer with the lowest IP address as its BGP identifier.
- Select the route the internal BGP peer advertised with the lowest IP address as its BGP identifier.

The tie-breaking process commences with all possible valid routes considered equally preferable by the regular BGP route-selection process. Each tie-breaking rule selects the best route. If only one route remains, that route is selected. The next rule is applied exclusively when two or more routes are considered "best" by the rule.

3.1.5 Other BGP capabilities

BGP can be utilised for more than just IPv4 encoding. It enables the encoding of routing information for different address families through two optional nontransitive path attributes: MP_*REACH_NLRI* and *MP_UNREACH_NLRI*. This feature allows BGP-4 to support various address families without causing any compatibility issues with older implementations.

Additionally, it is possible to implement Multicast with BGP. Multicast is a method of sending packets to multiple destination hosts, but only to those interested in them. This is done through special multicast or "group" addresses. Multicasts are well-understood and widely used, particularly at the LAN level.

There are several protocols that address necessary functions for multicast routing, including Multiprotocol Extensions for BGP (*MBGP* or *MP-BGP*). These protocols distribute alternative reachability information for multicast purposes, making it possible to have separate infrastructures for unicast and multicast.

MBGP has another important use, which is to transfer Virtual Private Network (VPN) reachability information through Multiprotocol Label Switching (*MPLS*) backbones. MPLS is a method that enables switching over multiple lower-layer infrastructures. MPLS borrows from Ethernet VLANs and ATMs, where each packet receives one or more labels before it enters the MPLS backbone. When the packet resides in the backbone and has labels, it is forwarded according to its first label, ignoring the network-layer destination address. Each MPLS switch replaces the

existing label with a new one, similar to how ATM changes VPI/VCI information in the cell header at each hop. MPLS doesn't come with its own layer two protocol, but it can be used with existing datalink-layer protocols and can take advantage of existing labels, such as Ethernet VLAN tags or ATM VCIs.

The original goal of MPLS and its predecessors was to improve packet-forwarding performance, which isn't any more necessary with the actual hardware. Nonetheless, MPLS has another crucial advantage: it conceals the network layer while retaining the internetworking function. Hence, it's still possible to use different underlying datalink-layer protocols. This feature enables MPLS to transfer other protocols and different instances of the same protocol over a network that isn't end-to-end switched at the datalink layer.

3.2 BGP Routing Policies and AS's Relationships

As explained in the previous chapter, there are two main types of relationships between Autonomous systems - Peering and Transit.

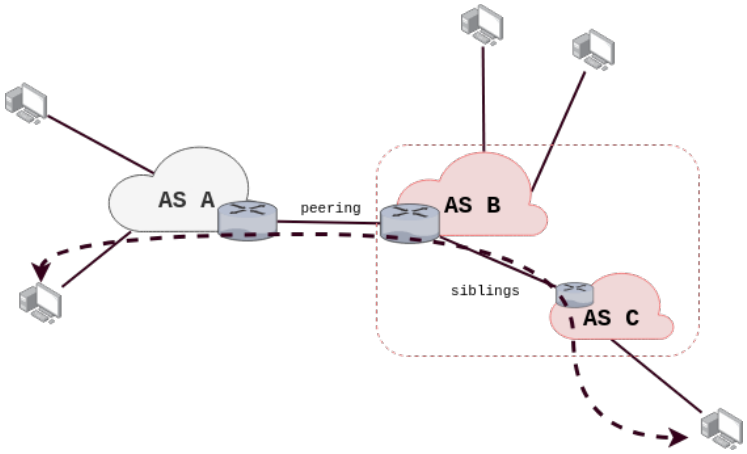


Figure 3.4. BGP peering relationship

When peers exchange traffic between customers, the AS exports only customer routes to a peer and the peer's routes only to its customers. On the other hand, the customer needs to be reachable from everyone, so the provider tells all its neighbours how to reach the customer. The customer does not want to provide transit service, so they do not let their providers route through it.

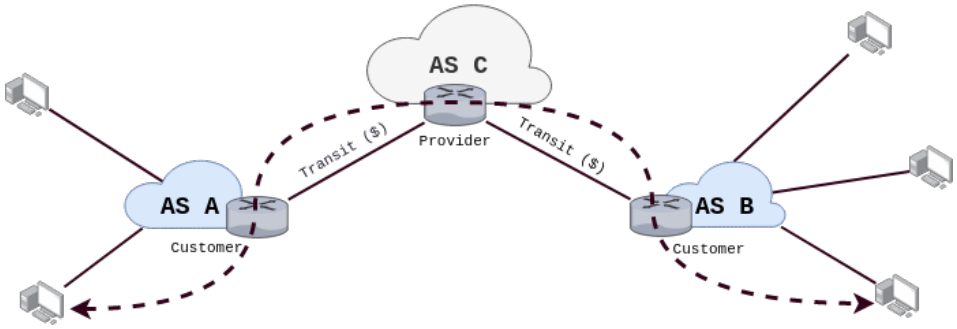


Figure 3.5. BGP Transit relationship

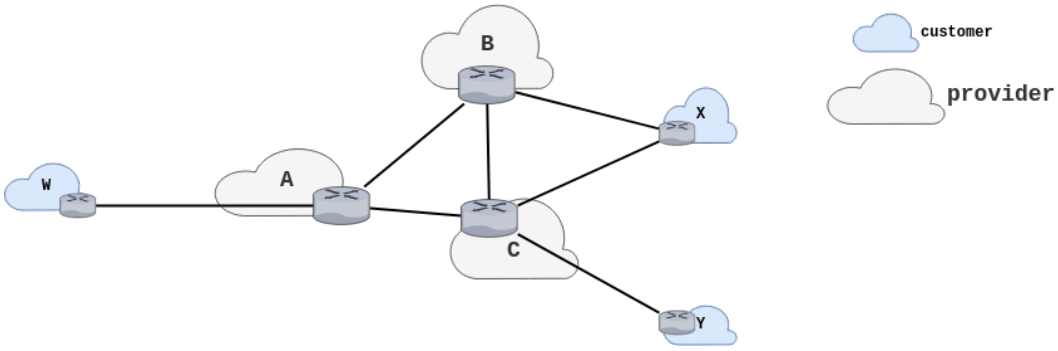


Figure 3.6. BGP routing example

Let me provide an example to explain more clearly (Fig. 3.6). Imagine that there are three network providers called A, B, and C, and three customers named X, W, and Y. X is connected to two networks and does not want to allow traffic from B to C, so X will not share a path to C with B and vice versa.

A advertises the path AW to B, who in turn advertises the path BAW to X. B does not share the path BAW with C. B won't earn any revenue for routing CBAW because neither W nor C are B's customers. B wants to compel C to route to W through A and only wants to route traffic to and from its own customers.

3.3 BGP Routing Information Base (RIB)

When using the Border Gateway Protocol (BGP), speakers receive updates from peers, analyze the data, and selectively broadcast routes to other peers based on policies. BGP uses a database called the BGP Routing Information Base (RIB) to do this.
It is essential to understand the role of RIB because it is the primary input for the inference algorithm used in this work.

The BGP Routing Information Base has three components:

Adj-RIBs-In stores unprocessed routing information learned from BGP updates received from peers. Routes in Adj-RIBs-In are considered feasible routes.

Loc-RIB contains routes the BGP speaker selects by applying the decision process to the routes in Adj-RIBs-In. These routes populate the routing table (RIB) along with routes discovered by other routing protocols.

Adj-RIBs-Out contains the routes the BGP speaker advertises to its peers in BGP updates. The outgoing routing policies determine what routes are added to Adj-RIBs-Out.

The BGP decision process selects routes by applying incoming routing policies to Adj-RIBs-In routes and adding the selected or modified routes to the Loc-RIB. The decision process is carried out in three phases:

Phase 1 calculates the preference degree for each feasible route in the Adj-RIBs-In. It is invoked whenever a router receives a BGP update from a neighbouring AS containing a new, changed, or withdrawn route. A non-negative integer is derived for each route, indicating its preference degree.

Phase 2 Chooses the best route out of all the available routes to a destination and installs it in the Loc-RIB. This phase is invoked only after phase 1 has been completed. Loops are detected in phase 2 by examining the AS_PATH. Any routes with the local AS number in the AS_PATH are dropped.

Phase 3 Adds the appropriate routes to the Adj-RIBs-Out for advertisement to peers. This phase is invoked after the Loc-RIB has changed and only after phase 2 has been completed. If it is to be performed, route aggregation happens during this phase.

The database explained here ([78]) is distinct from the routing table, as it exclusively serves the BGP process and not packet forwarding.
The Local-RIB routes that meet the criteria outlined by the local BGP speaker's vendor implementation, and routing protocol preferences are the only ones installed in the routing table.
It is worth noting that this approach to BGP implementation is not mandatory, and vendors can choose their methodologies.

Chapter 4

BGP security problems

The initial implementation of BGP was designed for a limited, trusted network. Still, with the growth of the internet and the financial motivations behind it, security has become a critical concern.

Despite all the efforts done in the last years, BGP is still an insecure protocol. Recently, the US government warned that insecurity could cause serious damage: "BGP vulnerabilities put US-person data and communications (including government communications) at risk of theft, espionage, and sabotage by foreign adversaries, both directly and through third parties. These are not hypothetical concerns" (the US Department of Justice (DoJ) and the US Department of Defense (DoD) [30]

BGP attacks can lead to router overloading, causing instability and connectivity issues. Adversaries can hijack prefixes or eavesdrop on traffic, leading to the creation of backholes. Blackholing is sometimes used to enforce private and non-allocated IP ranges. However, malicious blackholing involves false route advertisements that aim to attract traffic to a specific router before dropping it [73].
Not less important, Intercepted traffic can be subjected to man-in-the-middle attacks.

4.1 Common BGP Security Vulnerabilities

BGP faces three fundamental vulnerabilities that pose serious threats. First, outsiders can physically attack the BGP infrastructure [34].
Second, neither BGP nor its underlying protocols have mechanisms to prevent tampering with protocol data by outsiders. Since BGP messages are transmitted over a TCP session, measures used to secure TCP connections (such as cryptography) can also be used to secure BGP [103, 107, 24, 100].

Third, even if we can eliminate intentional corruption of control messages by outsiders by hardening the TCP protocol and physical links, BGP does not guarantee that legitimate participants won't use protocol data maliciously or distribute fake data injected into the routing information. I will explore this third group of vulnerabilities [63].

4.1.1 Data falsification attacks

Data falsification attacks are a threat to the integrity of routing data in the Border Gateway Protocol (BGP). These attacks involve a malicious Autonomous System (AS) injecting corrupted routing data into BGP messages. Several attack vectors are possible, including prefix hijack, sub-prefix hijack (also known as deaggregation attack), AS path forgery, interception attack, replay/suppression attack, and collusion attack.

Prefix hijack occurs when an AS falsely claims to originate a prefix not delegated to it. This causes a multiple origin AS (MOAS) conflict to be observed by other ASes. In April 2020, AS12389 (Rostelecom) hijacked 8,800 prefixes, affecting several providers, including Amazon and Akamai [90].

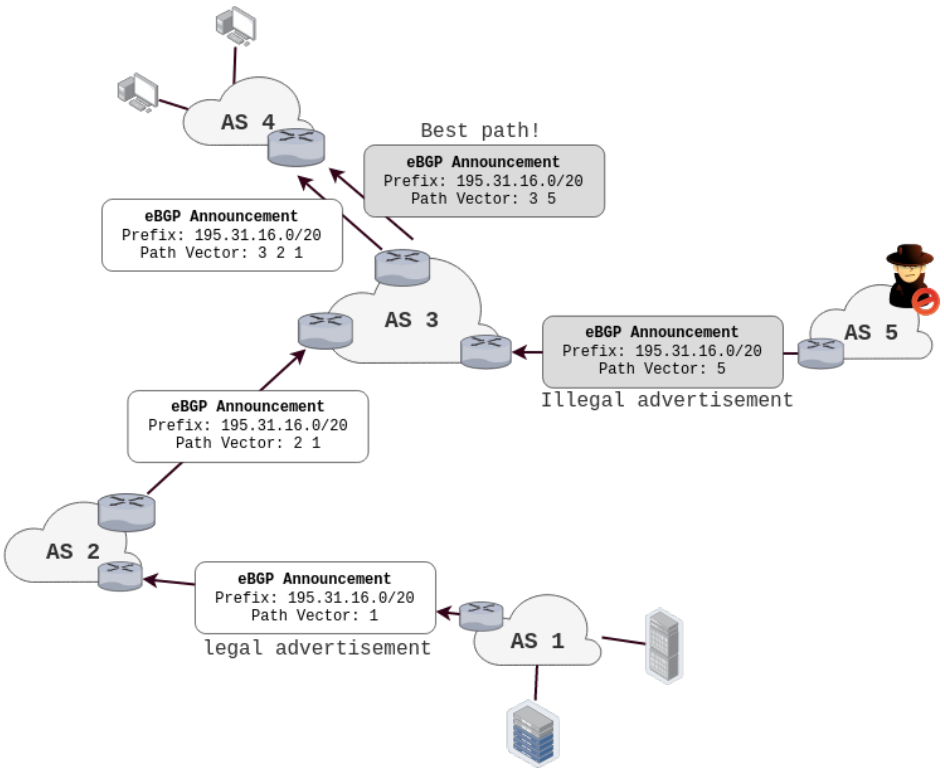


Figure 4.1. Example of Prefix Hijacking

Subprefix hijack is another way the attacker can prevent a MOAS conflict. The attacker advertises a subnetwork of an existing prefix that does not belong to them. If no other ASes originate this prefix, most ASes adopt the route due to the longest prefix match rule. Prefix hijacking caused the incident with YouTube in 2008 [71].

Another incident happened in April 2018, when an attacker stole cryptocurrencies using a BGP leak announcing prefixes of DNS resolvers [10].

AS path forgery is an attack where the attacker arbitrarily tampers with the AS path in UPDATE messages. Instead of forging the origin AS, the attacker modifies the AS path to avoid a MOAS conflict and causes a one-hop prefix hijack. The attacker may also announce a fake link to a subprefix of the victim AS, known as a one-hop subprefix hijack. ASes may intentionally modify the AS path in BGP messages to advertise more attractive routes at the control plane but still use another sequence of ASes at the data plane to forward the traffic. This is known as traffic attraction attack [37].

Interception attack is an improved version of (one-hop) (sub) prefix hijacks. The attacker has a valid route to the victim AS and can redirect traffic through it and forward it back to the real destination without disturbing the connectivity. In 2017, a Russian AS conducted several prefix and sub-prefix hijacks of IP blocks belonging to well-known and high-traffic Internet organisations and rerouted the attracted traffic back to legitimate destinations [38].

Replay/suppression attack is when a malicious AS replays or suppresses withdrawal for a previously announced route. Although there is no documented real event of this attack, every registered Internet outage may potentially be caused by it.

Collusion attack is another type of attack where two colluding non-neighbouring ASes create a virtual tunnel between each other and build a BGP session through it. This allows them to generate forged routes without causing suspicious routing conflicts [54].

4.1.2 Protocol manipulation attacks

Protocol manipulation attacks involve a malicious Autonomous System (AS) attempting to manipulate the properties of the routing protocol itself. This can be achieved through several attack vectors, such as MED modification and exploiting RFD/MRAI timers.

MED modification efers to the ability of a malicious AS to tamper with the multi-exit discriminator (MED) values of routes. Since MED, like other BGP attributes, is not protected, a malicious AS can manipulate the values to influence the decisions of other ASes.

Exploiting RFD/MRAI timers involves a malicious AS artificially withdrawing and re-announcing a route. ASes using the RFD timer may consider the route unstable and ban it, while ASes employing the MRAI timer may delay their distribution of the corresponding UPDATE messages, making the route seem unreachable for some ASes [92].

Additionally, flaws in BGP implementation can be exploited by manipulating BGP protocol messages, causing routers to behave unpredictably and leading to denial of service [76, 75, 74].

4.1.3 Data misuse attacks

An Autonomous System (AS) can use accurate routing data in a malicious manner, resulting in several possible attack vectors. These include:

Denial of Service (DoS) attacks create heavy congestion on routers or links that carry Border Gateway Protocol (BGP) messages. This leads to congestion-induced BGP session failures. When the BGP sessions are restored, routers must exchange full routing tables, increasing their load and introducing significant convergence delays. Another type of DoS attack is to create continuous withdrawals and re-advertisements of target routes to a victim AS, causing deliberate link flapping. Other ASes tag these routes as unstable and start suppressing their further propagation.

Route leak attacks occur when an AS propagates routes to ASes that are not intended to receive them under the terms of negotiated business agreements. For example, a customer AS may leak a route received from one provider to another, even though it contradicts the valley-free export rules. On August 26, 2017, Google accidentally leaked routes learned from its peers to some of its providers, and as a result, Google became a transit AS [39]. This caused slowness in the internet and even complete connectivity disruption for many users.

4.2 Existing Security Measures

The BGP protocol offers a wide range of metrics and functionalities, which can be strengths and weaknesses. As we have seen, the BGP protocol lacks security measures, as highlighted in RFC 4275 [41].

Over the years, several security measures have been developed to prevent attacks, but many of them are not mandatory and are up to the discretion of Autonomous Systems. Basic countermeasures commonly applied by AS, such as prefix filtering, can serve both economic and security purposes.

Given that BGP runs on TCP, it means that a router can be attacked by a remote location. One simple countermeasure is configuring the router to accept only BGP requests with TTL=1, which is supposed to be received only by a neighbour. However, this can be easily bypassed by setting the TTL according to the number of nodes between the victim router and the attacker [11]. To address this issue, AS can implement an Access Control List (ACL) to block IPv4/6 addresses not from a neighbour.

The RFC 2385 defines a TCP extension that enhances security for BGP. It introduces a TCP option that carries an MD5 digest in a TCP segment, which acts as a signature for that segment, incorporating information known only to the connection endpoints. This option reduces the risk of certain security attacks on BGP [43].

Fortunately, Internet Exchange Points (IXs) play a crucial role in helping Autonomous Systems achieve security by implementing various security mechanisms.

4.2.1 BGP protocol extensions

To enhance the security of BGP and prevent attacks, several BGP extensions have been developed, including S-BGP, psBGP, soBGP [8], and BGPsec as defined in RFC 8205 [53, 52].

BGPsec introduces path and origin validation mechanisms to strengthen BGP's byzantine robustness. Byzantine robustness refers to the ability of hosts to receive the same message sent by the original host, decide on the message's contents within a finite time period, and have their decisions match those of other hosts in the network, even in the presence of malicious or faulty behaviour.

Path validation allows routers to validate the path information contained in UPDATE messages by verifying whether the announced path matches the actual path packets will take. On the other hand, origin validation verifies whether the announcing AS owns the prefix contained in the UPDATE message. To execute these validations, an additional infrastructure is required to hold information about AS numbers and prefix owners. One possible implementation of such infrastructure is the Resource Public Key Infrastructure [77].

4.2.2 Internet Routing Registry (IRR)

The Internet Routing Registry (IRR) is a widely used path origin control that serves as a distributed routing database for configuring and debugging Internet routing and addressing.
However, the IRR has a weak security model that has been a known issue for a long time and contains records that are either incorrect or missing. The lack of cryptographic signing of records and the presence of multiple suppliers of IRR data further compound the issue.
To enhance the security of the infrastructure, the IRR should be used in combination with a more secure system, such as the Resource Public Key Infrastructure (RPKI)[47].

4.2.3 Resource Public Key Infrastructure (RPKI)

Resource Public Key Infrastructure (RPKI) is a cryptographic method for creating digital signatures associating a route with an originating AS number. The five

Regional Internet Registries (RIRs) - AFRINIC, APNIC, ARIN, LACNIC, and RIPE - provide their members with the ability to sign a Route Origin Authorization (ROA) record, which contains an IP/ASN pair. Once a route is signed, anyone can use the data to filter routing or monitor it since ROAs are public [84, 33].

Since the ROA is a digitally signed object, it provides a way to verify that an IP address block holder has authorised an AS (Autonomous System) to originate routes to one or more prefixes within the address block. The RPKI system provides an attestation method for BGP routing.

The Internet Routing Registry (IRR) and RPKI use third-party entities to hold the database information. However, with RPKI, the same entity that allocated or assigned a numeric resource (such as an IP address or ASN) also holds the Certificate Authority (CA) used to validate the ROA records. In the RPKI world, CAs are called TAs or Trust Anchors.

While each RIR supports RPKI for its members, the toolset for successfully operating a network with RPKI-enabled route filtering is still very limited. In addition, not all networks are currently participating in both signing and verifying IP routes via RPKI. Therefore, the global Internet is not yet entirely secure.

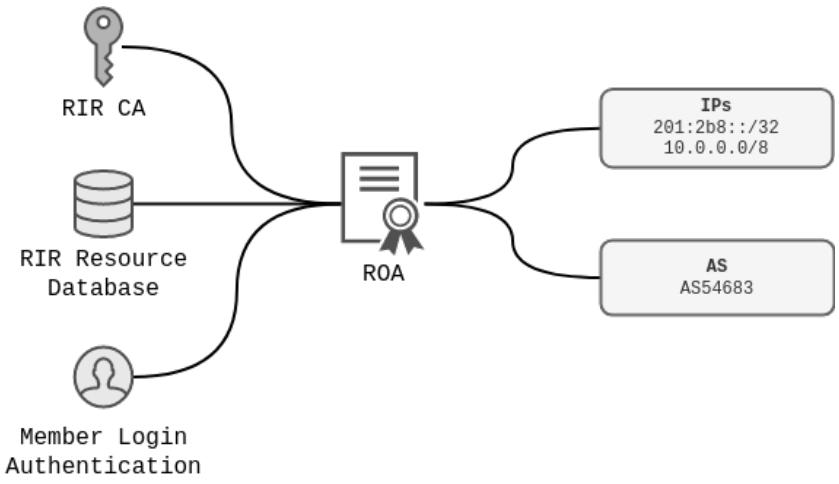


Figure 4.2. Graphic representation of a ROA record

Although RPKI prevents some types of attacks and misconfigurations, it still has significant flaws that allow for attacks such as wormhole attacks. These vulnerabilities indicate that the protocol is still a work in progress. One of the main challenges that RPKI faces is deployment, which is hindered by prevalent issues.

While RPKI provides additional security, it comes at a cost. The process of creating signatures and validating them takes more time and requires additional space. Additionally, collecting and managing the necessary data is difficult since no single authority is responsible for managing AS numbers and prefixes. [77].

4.2.4 Remote Triggered Black Hole filtering (RTBH)

Remote Triggered Black Hole filtering (RTBH) is a widely used tool to mitigate inter-domain Distributed Denial-of-Service (DDoS) attacks. The primary use case for RTBH filtering is to prevent volumetric DDoS attacks, where attack traffic is dropped before reaching the intended destination, thereby limiting the damage to the network infrastructure under attack.

Internet Exchange Points (IXPs) are particularly well-suited for this type of prevention mechanism as they offer a central point where numerous Autonomous Systems (ASes) converge to exchange inter-domain traffic. The RTBH signals from the victim are propagated across the IXP by the route servers, and all IXP members receive and accept the signals. The members then forward all traffic destined to the victim prefix to the blackhole, which results in both attack traffic and legitimate traffic being dropped, leading to collateral damage.

However, any BGP peer that does not accept a blackhole route from the route server will continue forwarding the traffic intended to be filtered. The acceptance of blackhole routes is beyond the control of the triggering AS and is subject to the local BGP policies of the receiving peer.

One major drawback of RTBH is that it is a coarse-granular traffic filtering tool. While it drops DDoS traffic early in the network, it completes the attack, making the victim unreachable. We could extend the filter rules to reduce collateral damage and keep the victim reachable during the attack. Fine-grained blacklisting is possible to blacklist attack traffic, but whitelisting legitimate traffic is difficult and highly variable.

Most volumetric attacks mitigated using RTBH utilise three attack vectors: NTP and DNS. Blocking these attack vectors, each identified by the default source port of the misused application, can effectively block the attack and prevent collateral damage [68].

4.2.5 Mutually Agreed Norms for Routing Security (MANRS)

The Mutually Agreed Norms for Routing Security (MANRS)[61] is a global initiative the Internet Society supports.
It aims to collaborate with operators, enterprises, and policymakers to implement necessary fixes to reduce the most common routing threats. MANRS consists of four simple but concrete steps significantly improve Internet security and reliability.
The first two operational improvements eliminate common routing issues and attacks, while the second two procedural steps bridge universal adoption and decrease the chances of future incidents.

4.2.6 ASPA

ASPA objects are digitally signed objects that allow an Autonomous System (AS) identifier holder to authorize one or more upstream providers. The Resource Public Key Infrastructure (RPKI) uses ASPAs to attest that a Customer AS holder (CAS) has authorized a Set of Provider ASes (SPAS) to propagate the Customer's IPv4/IPv6 announcements onward, for instance, to the Provider's upstream providers or peers.

When validated, the content of an ASPA can be used to detect and mitigate route leaks. The mechanism also helps block invalid routes that ROAs objects indicate as valid, thereby increasing BGP security [91].

The use of ASPA in RPKI is currently defined in three Internet Engineering Task Force (IETF) drafts [5, 93, 6]. However, these drafts have no formal status until they gain strong community support to be published as Request for Comment (RFC)—Internet standard—a currently ongoing process in the IETF.

4.2.7 Other Mitigations

Over the years, various mitigation and detection techniques have been developed to address BGP problems. One such example is ARTEMIS (Automatic and Real-Time dEtection and MItigation System) [86], which is a self-operated and unified approach for detecting and mitigating BGP issues through control-plane monitoring. This defence approach is based on accurate and fast detection operated by the AS.

Another example is ASIRIA [7], a route leak detection and protection mechanism. It comprises three main components: (i) the ASIRIA Inference Algorithm, which is used to infer relationships between ASes based on available information in the IRR; (ii) the ASIRIA Route Validation, which is run by ASIRIA-enabled routers to perform real-time detection of invalid routes using the inferred AS relationship information, and (iii) the ASIRIA Alarm System, which detects leakage events.

4.3 State of Internet Security

The deployment of RPKI by major transit providers, known as Tier 1, such as Cogent, GTT, Hurricane Electric, NTT, and Telia, has made many downstream networks more secure without needing to deploy any validation software.

Looking at the evolution of the successful tests per ASN, we notice a significant increase over the recent months.

RPKI is growing, and we would like to express our gratitude to the hundreds of network engineers worldwide who are contributing to making Internet routing more secure by deploying RPKI.

Currently, 25% of routes are signed, and 20% of the Internet is doing origin validation, but these percentages are growing daily[98].

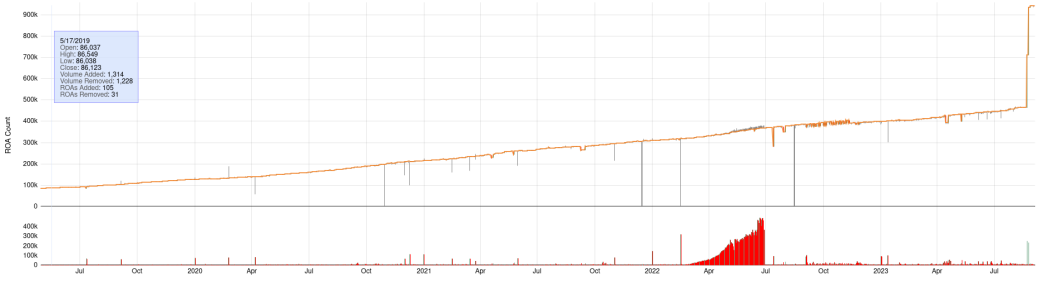


Figure 4.3. Historical data of numbers of ROA records by Cloudflare [85]

Despite the countermeasures taken, the internet remains vulnerable to security threats. Incidents related to BGP continue to occur, as reported by [2] and [1]. It is crucial to keep working towards securing BGP since it is a critical protocol that the entire internet runs on.

Chapter 5

Internet Measurement and Data Collection

Gathering data and measuring internet activity is crucial for understanding and managing the global network. This enables organisations and service providers to optimise their infrastructure for an improved user experience, with reduced latency, high availability, and optimised resource allocation.

Recognising internet traffic patterns and detecting anomalies is paramount for mitigating threats like DDoS attacks, botnets, and network intrusions. Collecting data helps identify unusual behaviour and potential vulnerabilities so that cybersecurity measures can be implemented.

Internet measurement data is also priceless for researchers studying the Internet's architecture, performance, and evolving trends. Such data drives innovation and the development of new technologies and standards.

Policymakers and regulatory bodies use internet usage and performance data to inform decisions related to net neutrality, privacy, and competition. Accurate data is essential for effective Internet governance.

Moreover, network administrators rely on data collection to diagnose and resolve network issues quickly. Monitoring tools can help pinpoint the root cause of any problems and minimise downtime, ensuring a seamless user experience.

5.1 CAIDA

The Cooperative Association for Internet Data Analysis (CAIDA) [19] has the primary goal of providing data, tools, and analysis to support the engineering and policy concerns of the global internet. CAIDA collects internet data through various methods and projects, including the Ark infrastructure [3].

The Ark infrastructure consists of distributed measurement nodes strategically

placed across the internet to continuously collect data on internet topology, routing, and performance.

CAIDA uses both active and passive measurements to collect data. Passive data collection involves monitoring network traffic at various vantage points, allowing CAIDA to gather data on the internet backbone and interdomain traffic patterns without actively probing networks. On the other hand, active data collection involves sending test packets such as ICMP and UDP probes to various internet destinations and analysing the responses to measure network reachability, latency, and other performance metrics.

CAIDA also collects data related to the Border Gateway Protocol (BGP) to understand how routing information propagates through the internet. This data helps researchers and network operators monitor and analyse changes in routing tables. In addition, CAIDA is involved in projects aimed at mapping the physical infrastructure of the internet by monitoring router-level and topological information. This way, CAIDA creates maps and visualisations of the internet's structure.

CAIDA's ASRANK [4] is a ranking of Autonomous Systems (AS) and organisations (Orgs) that approximately map to Internet Service Providers. The ranking is derived from topological data collected by CAIDA's Archipelago Measurement Infrastructure and Border Gateway Protocol (BGP) routing data collected by the Route Views Project and RIPE NCC.

Furthermore, CAIDA developed bgpstream [16], an open-source software framework for live and historical BGP data analysis that supports scientific research, operational monitoring, and post-event analysis.

5.2 RouteViewProject

The RouteViews project [83] at Oregon University was originally created to provide real-time access to BGP information about global routing systems from the perspectives of various backbones and locations around the internet. While other tools like Looking Glass Collections provide similar services, they typically offer only a limited view of the routing system or do not provide real-time data.

Although the RouteViews project was initially intended to help operators determine how the global routing system viewed their prefixes and AS space, the collected data has been used for various purposes. For instance, NLANR utilised the data to visualise AS path and study IPv4 address space usage. Others have used it to map IP addresses to origin AS for different topological studies. CAIDA has also used the data to generate the geographic locations of hosts in conjunction with the NetGeo database, a functionality supported by CoralReef and the Skitter project.

5.3 RIPE NCC

RIPE NCC (RIPENetworkCoordination) is a non-profit organisation that supports the Internet's infrastructure through technical coordination within its service region. They play a significant role in Europe by providing tools such as RIPE Atlas and RIPE RIS to measure and analyse internet data [80, 82, 81].
In addition, they encourage collaboration among network operators and researchers. The organisation's primary activity is to act as the Regional Internet Registry (RIR), providing global Internet resources and related services, including IPv4, IPv6, and AS number resources.

5.4 The Peering DB

PeeringDB [79] is an accessible and user-maintained database of networks that serves as the go-to location for interconnection data. This public tool facilitates the global interconnection of networks at Internet Exchange Points (IXPs), data centres, and other interconnection facilities, making it the first stop in making informed interconnection decisions.

The database is a non-profit, community-driven initiative that is run and promoted by volunteers and is aimed at promoting the growth and development of the Internet.

Initially, PeeringDB was set up to facilitate peering between networks and peering coordinators. However, in recent years, its vision has evolved to keep up with the speed and diverse manner in which the internet is growing. The database now includes all types of interconnection data for networks, clouds, services, and enterprises, as well as interconnection facilities that are developing at the edge of the Internet.

5.5 Looking Glass

A "Looking Glass" is a tool used for network diagnostics that enables users to obtain information about the network status and routing of an Autonomous System (AS) or network from a remote location. It provides a view into the BGP (Border Gateway Protocol) routing tables and network configuration of the AS, facilitating various network analysis tasks without requiring direct access to the AS's infrastructure.

Looking Glass servers are typically offered and maintained by network operators, ISPs, and organisations with a substantial internet presence. They can be made publicly available or restricted to a specific group of users, empowering them to gain insights into the functioning of the AS and the state of the internet routing system. These tools are highly valuable for network administrators, researchers, and anyone curious about comprehending and resolving issues related to internet routing and connectivity [56, 60, 96, 97, 57].

5.6 Other Tools

Besides the previously mentioned projects and tools, there exist numerous other internet measurement and data collection tools that researchers, network operators, and other interested parties can utilise. These tools provide valuable insights into the internet's performance and structure, which can aid in optimising network infrastructure and informing policymaking.

For instance, bgp.tools [12] and bgpview [17] are web-based platforms that offer a variety of BGP-related utilities and analysis tools. These platforms enable users to search for AS numbers, IP prefixes and BGP communities, and view details such as origin AS, path, and next hop. Users can also perform route and AS path analysis, visualise BGP updates and announcements, and access historical data.

There are several other tools and platforms available for collecting and analysing Internet measurement data. For example, BGPmon [15] allows you to evaluate the routing health of your network, providing you with information to determine the stability of your networks and potential risks to your data.

Chapter 6

AS relationships inference

The topic of AS relationship inference was first introduced by Gao's pioneering work [35].
The study classified an AS link into two main relationship types: provider-to-customer (p2c) and peer-to-peer (p2p). In a p2c (and c2p) relationship, the customer AS pays the provider AS to obtain global reachability for the traffic transmitted between them. In a p2p relationship, no payment is involved as the two parties transmit traffic between their networks and their customers' networks.
Another relationship that can be taken into consideration is the s2s relationship, where the same organisation owns the two ASes, and they transmit traffic between their providers, peers, and siblings for free.

Over the years, several inference algorithms have been proposed to achieve higher accuracy [32, 4, 50, 49, 14, 101, 105, 94, 35]. However, these algorithms face uncertainty regarding relationship inference due to issues such as incomplete coverage of the AS-level topology and unreliable heuristics. I aim to provide a way to compare these algorithms and identify which heuristic produces the most accurate result. [32]

As I mentioned previously in Chapter 2, Relationship inferences have a wide range of applications across various research fields. Such as identifying network congestion [26], detecting malicious Autonomous Systems (ASes) [51, 23], deploying security mechanisms for BGP [22], protecting the anonymity of data [72], optimizing video streaming [31], and understanding the effects of public policy proposals on Internet governance [55].

6.1 Challenges and limitations

Inferring relationships between Autonomous Systems (ASes) is a complex task as their relationships can be more intricate than what can be easily described. While some ASes may peer for certain IPs, they may have C2P agreements for others. The algorithms used to determine these relationships may oversimplify the situation, leading to inaccuracies.

Furthermore, most inference algorithms assume that internet routing is valley-

free, which is not entirely accurate. Valley paths are not solely caused by BGP errors, but rather result from complex business relationships and intentional policies of Autonomous Systems (ASes) that employ unique and unconventional models [36].

It's difficult to establish a truly accurate internet view as a comprehensive view of all possible ASes on the internet is unfeasible. Data can only be gathered from a limited subset of vantage points, which may not be sufficient to see all ASes and their connections.

Common inference algorithms rely on routing data provided by Route-Views and RIPE RIS, which are collected from various vantage points (VPs) across the world. These VPs can be categorized as either full or partial. Full VPs report a significant number of routes to Route Collectors (RCs), while partial VPs report only a small number of routes due to intentional export limitations. The routes reported by each VP represent its internet observations, which an inference algorithm extracts AS links and relationships from. However, even a full VP observes only 1.12% of point-to-point (p2p) links and 71.87% of point-to-customer (p2c) links compared to the total reported by all VPs. This vast disparity between p2p and p2c observations is mainly due to the underlying AS relationships and the no-valley principle, where learned routes from a p2p neighbor or a provider are only propagated to customers [50].

The observation bias among each VP is noticeable, which can result in a collective observation bias that heavily relies on the set of VPs that an inference algorithm utilizes. Additionally, the absence of observations on AS links can deceive inference algorithms that rely heavily on factors such as total degree, transit degree, or AS neighbors. Bias stems not only from the limited number of VPs but also from the distribution of VPs in the network [50, 89].

VPs will inevitably miss some AS links, including p2p ones. Therefore, other methods such as mining IXP peering [42] and BGP community data [106], data plane probing, and even route manipulation can be used to uncover hidden links [50].

Moreover, unexpected events like route leaks or link failures can cause noises that can lead to inaccurate reporting of routes that do not depict intended AS relationships. While some of these noisy routes can be eliminated using heuristics, the others, along with normal routes, are indiscriminately treated in an inference algorithm. To prevent such issues, the algorithm employs a filtering mechanism.

6.2 Algorithms compared

It is not possible to compare all available inference algorithms, so I have selected a subset that uses different approaches to creating heuristics. Specifically, I have compared three algorithms: ASRank, Problink and Toposcope.

6.2.1 ASRank

The AS-Rank algorithm is used to infer relationships between ASes and is created and used by CAIDA [59, 48].
It assumes that there is a group of major transit providers at the top of the hierarchy, that most customers buy transit services to be globally accessible, and that there are no cycles of p2c links. The algorithm has 11 steps to determine the relationship type between each link, which could be customer-provider (c2p), provider-customer (p2c), or peer-to-peer (p2p).

It is worth noting some important features of the AS-Rank algorithm. Firstly, the transit degree attribute plays a crucial role in determining the relationship labels in AS-Rank. Transit degree is the number of ASes that appear on either side of an AS in adjacent links of BGP paths. However, it does not include neighbours that do not transit traffic. Transit connectivity is easily observed by Route Collectors [42] and provides a more reliable metric to describe the prominence of an AS than node degree.
Secondly, AS-Rank follows a specific order while considering ASes and links. In some cases (step 5), it uses transit degree information, while in others (step 7), it does not.

When inferring relationships, this algorithm only covers c2p, p2c, and p2p relationships, and does not cover sibling relationships.

The steps ASRank takes:

1) Remove or clean up paths that have artifacts.

2) Arrange Autonomous System (AS) networks based on the transit degree and node degree, in descending order.

3) Identify the clique at the top of the AS topology.

4) Discard any paths that have been poisoned.

5) Determine the customer-to-provider (c2p) relationships using the ranking generated in step 2, starting from the top.

6) Determine c2p relationships for ASes that are not announcing provider routes, based on the Vantage Points (VPs) inferred to be announcing their routes.

7) Identify c2p relationships for ASes where the provider has a smaller transit degree than the customer.

8) Determine customers for ASes that have no providers.

9) Identify c2p relationships between stubs and clique ASes.

10) Determine c2p relationships where the adjacent links have no inferred relationship.

11) Identify the remaining links as point-to-point (p2p) relationships.

6.2.2 ProbLink

ProbLink [49, 48] is an algorithm that utilizes link attributes with stochastic information value. It takes into account all information about links and paths that traverse them, providing a framework for integrating conflicting information. ProbLink doesn't prescribe a specific order in which ASes and links are considered but rather continually updates the link type inferences until it reaches a fixed point in terms of the underlying stochastic distributions.

The algorithm starts with an initial classification of links based on the inference result of ASRank, so each link has deterministic relationship probabilities at the beginning. If ASRank labels L as a $p2p$ link, we convert it to $P(L = p2p) = 1.0$, $P(L = p2c) = 0.0$, $P(L = c2p) = 0.0$ and provide that as the input to our algorithm. Note that ProbLink is essentially a meta-inference algorithm that can be bootstrapped by outcomes of any algorithm.

ProbLink computes the conditional probability distribution for each feature based on observed data and the initial set of link relationship types. In each iteration, it updates the probabilities of each link's types ($P(L = p2p)$, $P(L = p2c)$, $P(L = c2p)$) by running the probabilistic algorithm. It then recomputes the distributions of features using the updated probability values of each link. This process is repeated until convergence, i.e., the percentage of links that change labels between each iteration drops below a small threshold.

The features that are taken into consideration when calculating the probability of each relationship are:

Triplet feature The triplet feature considers link triplets that appear in paths and attributes probabilistic values to the relationships of the first and last links, given the relationship of the middle link. For instance, suppose there are three consecutive links "L1 - L - L2" in a BGP path, where L1, L, and L2 are three links (AS pairs). This sequence "L1 - L - L2" is referred to as a link triplet. The main aim of the triplet feature is to probabilistically model valley-freeness.

Non-path feature The non-path feature is a measure of the probability of how many adjacent p2p or p2c links a particular link has, but none of these links appear before it on any of the paths. It is designed to capture the property that a link is unlikely to be a p2c link if it has many adjacent p2p/p2c links and none of them appear as a previous link on any of the paths containing the link. The non-path feature is also similar to the triplet feature, as it models valley-freeness in a probabilistic way.

Distance to clique feature The distance-to-clique feature is a way to identify that high-tier Autonomous Systems (ASes) are typically closer in terms of AS hops to other ASes within the same clique than low-tier ASes. Additionally, ASes within the same tier are more likely to be peers, whereas high-tier ASes tend to act as providers to low-tier ASes. This feature can be used to capture these observations.

Vantage point feature The number of VPs observing a link can indicate the type of the link. The vantage point feature determines the likelihood of a specific number of VPs with at least one path crossing a particular link, depending on its link type. This feature incorporates the following assumption: p2c links are more likely to be observed by multiple VPs compared to p2p and c2p links.

Co-located IXP and co-located private peering facility feature The information about co-located Internet Exchange Points (IXPs) and peering facilities is obtained from PeeringDB. This information is based on the understanding that when two Autonomous Systems (ASes) share multiple IXPs or facilities, they are more likely to be peering with each other.

6.2.3 TopoScope

TopoScope [50, 108] is a reliable and versatile method for inferring AS relationships. It operates in four steps, as outlined below.

1. The entire dataset is divided into smaller groups, and an existing empirical inference algorithm (such as AS-Rank) is applied to each group. A set of consensus links is then identified by using voting, where a majority of groups must agree on the inferred relationship.

2. Within the consensus links, a few trusted links are identified using an empirical rule based on the observability of p2p or p2c links.

3. To infer fuzzy links, A Bayesian Network is trained and optimised with Expectation Maximization [66]. Fuzzy links are those that have not been identified as trusted links in step 2.

4. A decision tree is trained for detour couples and is used to identify potential detour couples and their corresponding hidden links. The relationships on these links are then determined using xgboost [20].

6.2.4 A different algorithm: BGP2Vec

Another algorithm, worth mentioning is BGP2Vec [87, 88, 95], it is an innovative deep-learning approach that utilises only BGP announcements. These vectors can be used to solve important classification problems such as AS business types, AS Types of Relationships (ToR), and even IP hijack detection.
Similar to natural language processing (NLP) models, the embedding represents the latent characteristics of the ASN and its interactions on the Internet. In this approach, the AS number (ASN) is first mapped to an embedded vector using a shallow neural network.

During the training procedure, the network is fed with the ASN pairs, where the input is a one-hot vector representing the input ASN and the training outputs are also one-hot vectors representing the output ASNs (the context ASNs).
Gradient descent learning is then applied to adjust the network weights to maximise

the log probability of any context word given the input word.

For each task, an Artificial Neural Network (ANN) is activated that receives the vectors from the previous stage. In this context it executed only to infer AS Types of Relationships.

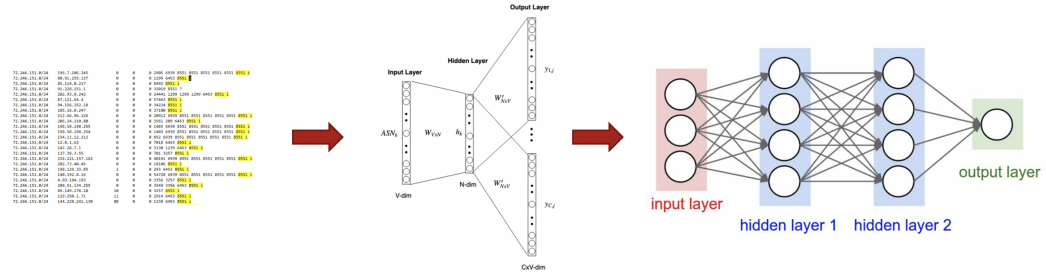


Figure 6.1. BGP2Vec [13]

I decided to exclude one algorithm from the comparison due to its neural network nature. The algorithm requires training on inferred data since there are no secure relationship datasets available. Using the algorithm without the proper training will result in a concatenation of errors that would decrease the output accuracy. However, this algorithm has the advantage of performing faster than the others. Nonetheless, I have developed a way to create a subset of more reliable relationships that can be used to train the neural network in the future.

Chapter 7

AS relationships inference algorithm comparison

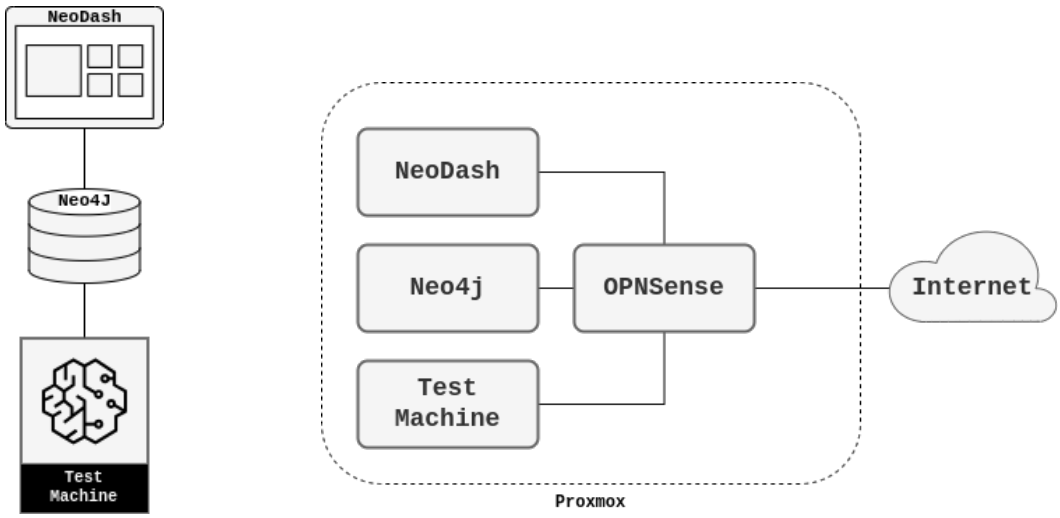
7.1 Environment

The machine used for testing boasts an Intel(R) Xeon(R) CPU E3-1245 V2, 16 GB of RAM, and approximately 60 GB of swap memory (essential for running problink and toposcope). These tests are conducted within an LXC container on Proxmox, which runs Debian 12. Access to the machines is closely regulated through an OPNsense VM, also located within Proxmox. In order to ensure more efficient comparisons, a separate container has been installed and configured with Neo4J DB to store and handle all inference results. Additionally, an Apache server has been set up in another LXC container, complete with the NeoDash Dashboard, to allow for more convenient data visualisation.

Opting for a graph database appears to be the most logical decision as the relationship between the data points inherently lends itself to a graph structure. Utilising this type of database enables the swift presentation of copious amounts of information and simplifies the process of querying intricate relationships. Conversely, a conventional SQL database would necessitate numerous joins and intricate queries, even for more straightforward tasks.

Proxmox has been selected as the operating system to efficiently manage the infrastructure. Its integration of the KVM hypervisor and Linux Containers (LXC), along with software-defined storage and networking functionality, provides an all-in-one platform. This empowers effortless management of virtual machines, containers and networks, and also offers backup functionality with minimal performance loss.

NeoDash allows for easy visualisation of Neo4j data, creating graphs and generating statistics. The dashboard connects seamlessly to the Neo4j database through the Bolt interface.



(a) Logical view

(b) Physical view

Figure 7.1. Test Envirorment

7.2 Data acquisition

To ensure greater coherence in the algorithms, I have used BGP data in the form of Routing Information Base (RIB) from Routeviews and RIPE NCC, which were obtained using pybgpstream [16]. I have considered the entire first day of each month between 01/05/2023 and 01/10/2023. The data download process took nearly three days for each day of data.

```
6423|13335
1798|13335
6423|6939|7545|2764|38803
1798|174|7545|2764|38803
6423|1299|2519
1798|2497|2519
6423|1299|2516|7670|18144
1798|2497|7670|18144
6423|6939|38040|23969
```

Listing 7.1. An example of the rib path input used for inference algorithms

To gather information about the IXP used by certain algorithms, I have downloaded the peering db history data from CAIDA datasets for each relevant day [46]. Figure 7.2 can give a glance of the type of data that can be found in the dataset.

To identify siblings, toposcope relies on the Caida AS to Organization Mapping Dataset [62]. This dataset is not frequently updated, so we consider the closest available mapping file to each day.

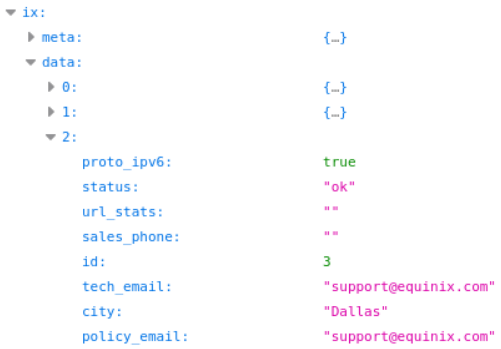


Figure 7.2. PeeringDB extract from CAIDA dataset

```
# format:aut|changed|aut_name|org_id|opaque_id|source
1|20180220|LVLT-1|LPL-141-ARIN|e5e...83c_ARIN|ARIN
2|20120621|UDEL-DCN|UNIVER-19-ARIN|c3a...101_ARIN|ARIN
[...]
279|20180220|LVLT-279|LPL-141-ARIN|e5e...83c_ARIN|ARIN
280|20120302|NTTA-280|NTTAM-1-ARIN|9f1...438_ARIN|ARIN
281|20180220|LVLT-281|LPL-141-ARIN|e5e...883c_ARIN|ARIN
```

Listing 7.2. An excerpt from the Caida AS to Organization Mapping Dataset file

7.3 Inference Algorithm Execution

The algorithms are executed independently, and the rib data is parsed as indicated in the algorithm's respective paper.

To generate input for Problink, I remove duplicated ASes resulting from BGP path prepending, filter out paths with AS loops, and sanitise BGP paths by removing reserved ASes. These artefacts of route poisoning can impact relationship inference [49].

The Toposcope's preprocessing differs slightly. To ensure accurate relationship inference, I filter out mistakes and noise, remove duplicated AS paths with different prefixes, and exclude AS paths with loops or reserved AS numbers [50]. As a result, the Toposcope's input is slightly larger compared to Problink's.

For each day, the Routing Information Base is downloaded, sanitised, and processed through asrank, problink, and toposcope. The resulting data is then parsed and inserted into the neo4j database.

The data format returned by asrank, toposcope, and problink is as follows:

```
<provider-as>|<customer-as>|-1
<peer-as>|<peer-as>|0
<sibling-as>|<sibling-as>|1
```

However, asrank does not return information on siblings. It should be noted that toposcope has the ability to infer hidden links, but they are not taken into consideration in this comparison.

This is a bash code showing how the data have been calculated:

```
!/bin/bash
# IFTTT Webhooks URL
IFTTT_WEBHOOKS_URL="https://maker.ifttt.com/trigger/<ifttt-server>/json/with/key/<ifttt-key>" #modify according to your needs (not mandatory)

# Function to execute a command and check its exit status
execute_command() {
    local cmd="$1"
    echo "Starting: '$cmd' ." >> run.log
    notify_ifttt "Starting: '$cmd' ."
    $cmd
    if [ $? -eq 0 ]; then
        echo "Command '$cmd' executed successfully." >> run.log
        notify_ifttt "Command '$cmd' executed successfully."
    else
        echo "Command '$cmd' failed." >> run.log
        notify_ifttt "Command '$cmd' failed."
        exit 1
    fi
}

# Function to send a notification to IFTTT
notify_ifttt() {
    sanitized_string=$(echo "$1" | sed -e 's/[^a-zA-Z0-9_:]/_/g' -e "s/['\"]//g")
    local message="{\"message\":\"$sanitized_string\"}"

curl --silent -o /dev/null -X POST -H "Content-Type: application/json" -d $message $IFTTT_WEBHOOKS_URL
}

#move problink file outside folder
if [ ! -s file_list_problink.txt ]; then
    ls ProbLink/ > file_list_problink.txt
    execute_command "mv ProbLink/* ./"
```

```
33      fi
34
35      #ASRANK
36
37      # execute ASRANK
38      execute_command "perl asrank.pl rib.txt " >
           asrank_result_native.txt
39
40      #execute Upload asrank
41      execute_command "python3 DataUpload.py -f
           asrank_result_native.txt -a asrank -d 2023-10-01 " &>
           upload_asrank.log
42
43
44      # PROBLINK
45
46      # parser for problink
47      execute_command "python3 bgp_path_parser.py
           peeringdb_2_dump_2023_10_01.json"  &> parser_problink.
           log
48
49      # execute ASRANK for problink
50      execute_command "perl asrank.pl sanitized_rib.txt " >
           asrank_result.txt
51
52      # execute PROBLINK
53      execute_command "python3 problink.py -p
           peeringdb_2_dump_2023_10_01.json -a 20231001.as-
           org2info.txt " &> problink.log
54
55      #execute Upload problink
56      execute_command "python3 DataUpload.py -f problink_result.
           txt -a problink -d 2023-10-01 " &> upload_problink.log
57
58      #move problink file inside folder
59      while IFS= read -r file; do mv "./$file" ProbLink/; done <
            file_list_problink.txt
60      rm file_list_problink.txt
61
62      #move TopoScope file outside folder
63      #move problink file outside folder
64      if [ ! -s file_list_toposcope.txt ]; then
65          ls TopoScope/ > file_list_toposcope.txt
66          execute_command "mv TopoScope/* ./"
67      fi
68
69
70      # TOPOSCOPE
71
72      #execute parser for toposcope
73      execute_command "python uniquePath.py -i rib.txt -p
           peeringdb_2_dump_2023_10_01.json "  &> parser_toposcope
           .log
```

```
74
75      #execute ASRANK for toposcope
76      execute_command "perl asrank.pl aspaths.txt " > asrel.txt
77
78      #execute TOPOSCOPE
79      execute_command "python toposcope.py -o 20231001.as-
           org2info.txt -p peeringdb_2_dump_2023_10_01.json -d tmp
           /"
80
81      #move TopoScope file inside folder
82      while IFS= read -r file; do mv "./$file" TopoScope/; done
           < file_list_toposcope.txt
83      rm file_list_toposcope.txt
84
85      #execute Upload toposcope
86      execute_command "python3 DataUpload.py -f asrel_toposcope.
           txt -a toposcope -d 2023-10-01 " &> upload_toposcope.
           log
87
88      # All commands have executed successfully
89      echo "All commands have completed successfully."
90
91      # Exit the script with a success status
92      exit 0
```

Listing 7.3. bash script for the tests execution

consider that notify_ifttt is just used to send notifications about the execution status through ifttt.

7.4 Data Storage

The inference Data has been collected inside a Neo4j DB [69] for easy access and performance purposes. In fact, the data are easily represented by relationships between nodes.

The Autonomous Systems (ASes) are represented as nodes labelled as AutonomousSystems. Each AS node is connected to a Time node through a relationship called INFERRED_ON.

The Time node contains information about the time and the inferred data. The AS nodes connected to the same Time node can be connected through relationships known as GIVE_TRANSIT_TO, PEER_OF, and SIBLING_OF. The GIVE_TRANSIT_TO relationship represents the customer-to-provider and provider-to-customer relationships depending on the direction. The PEER_OF and SIBLING_OF relationships represent peering relationships and sibling relationships, respectively, with no directional meaning.

Each relationship stores the property 'algorithm', which indicates the algorithm that inferred the relative relationship. Meanwhile, the AutonomousSystem node

store information about the AS itself, like the as name and Autonomous System number.

The relationships contain more attributes useful for analysis, such as "equal to past" with boolean values and "comparison" with values shown in the table below:

Value	Size	Same decision
0	3	yes
1	2	yes
2	1	-
3	3	no
4	2	no

The "value" refers to the comparison value, while the "size" indicates the number of relationships between the same ASes. There are three possibilities for the size, one for each algorithm. "Same decision" indicates whether the relationships between these ASes are equal across all algorithms.

The evaluation datasets are stored on the Neo4j database as inference results. However, the AS nodes are not connected to a Time node. Instead, they are linked to a dataset node containing the dataset's name and the date it is about.

The script I have written to upload the data on Neo4j uses the BGPView API [17] to retrieve the AS name when uploading data to the DB.

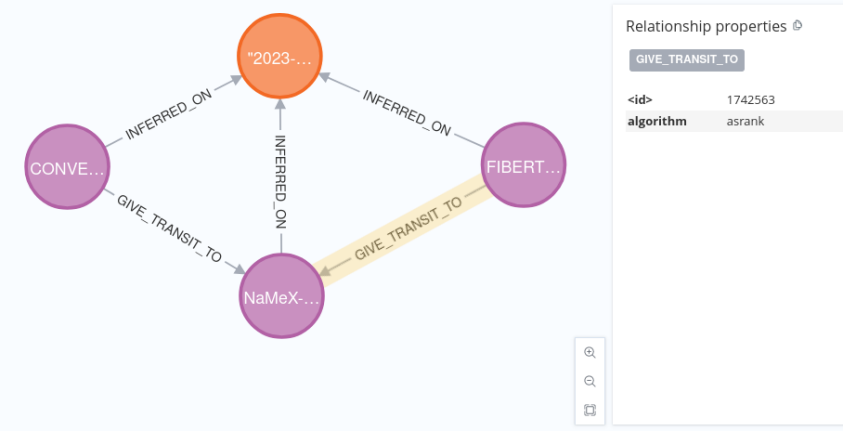


Figure 7.3. Example of relationship in Neo4j DB

Moreover, Settings for the Neodash dashboard are stored in the database under the node labelled _Neodash_Dashboard.

The database dump and the scripts used are avaliable on github [65].

7.5 Data Visualization

I have implemented a dashboard using NeoDash [70] for easy navigation among data.

The dashboard is served through an Apache web server and connected directly to the database through bolt over SSL.

The dashboard allows users to search and filter data by name, number, algorithm, and inference date and it returns the data in the form of graphs and tables 7.4.

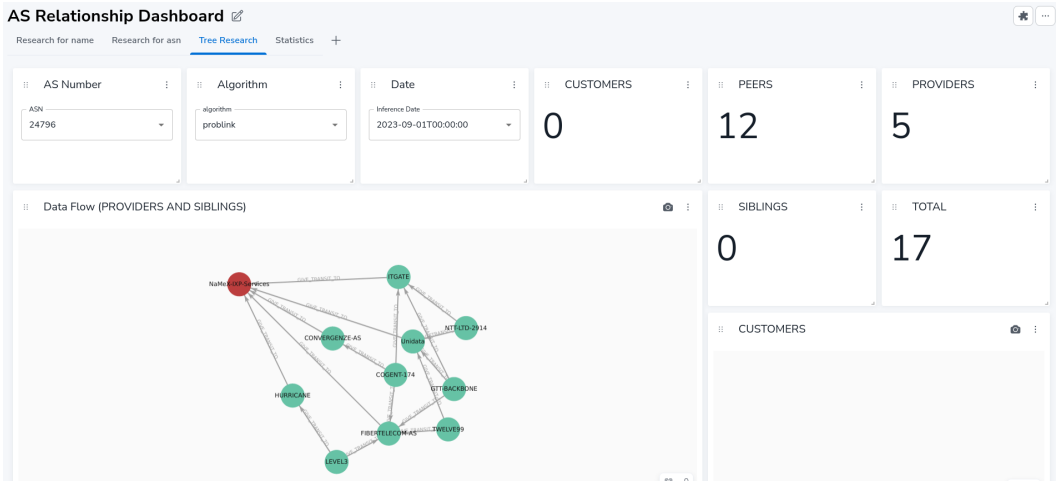


Figure 7.4. Dashboard

It allow to see easily the flow of information aggregating transit relationships and siblings in a single view of multiple levels of relationships 7.5.

I have developed a method to visualize data results and compare graphs without uploading files to neo4j. However, it is less flexible than a dashboard connected to a database [65].

All the neodash settings are saved in the Neo4j DB and can be retrieved from the DB dump [65].

7.6 Evaluation Dataset

The relationships between autonomous systems are private, making it challenging to validate inference algorithms.

To address this, my first attempt was to find an autonomous system that published AS relations, so I decided to create a dataset from the information provided by Hurricane's BGP toolkit [44].
I used a spider crawl to extract the data and generated a dataset of approximately 306925 relationships, sufficient to validate inference algorithms [65].
However, despite observing that relationships between autonomous systems appear

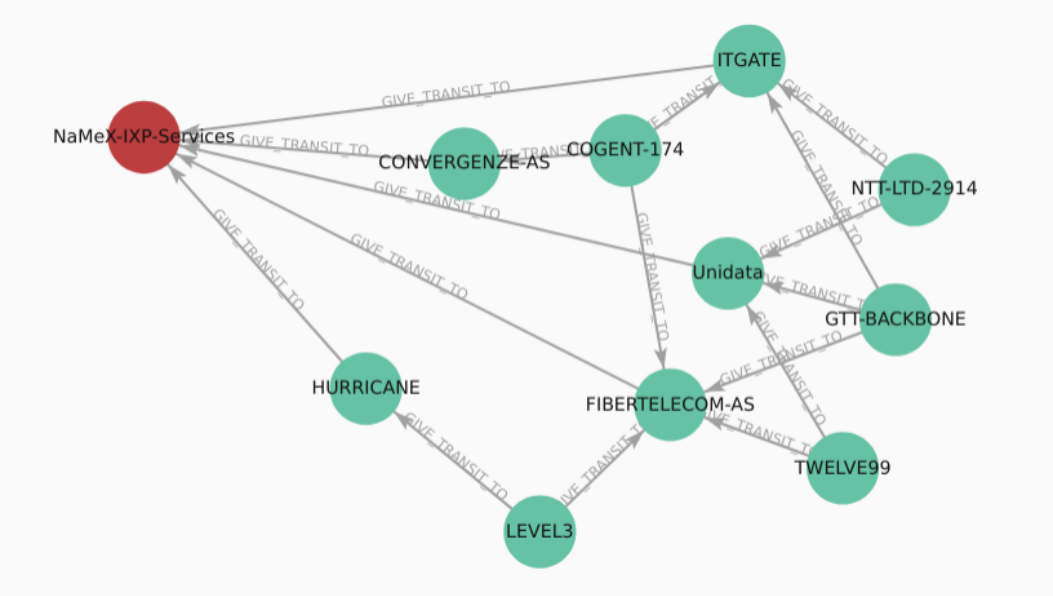


Figure 7.5. Flow of information in the dashboard

to remain stable over time, I noticed some changes in the information displayed on the Hurricane website. This led me to question the validity of the dataset obtained.

Another issue was that the dataset did not precisely match the time of the inference performed, and this slight variation could potentially lead to incorrect validation results.
Additionally, the reproducibility of the dataset was a concern as it is impossible to generate a dataset for past periods.

So, I developed a way to generate datasets I needed using a different approach.

7.6.1 A Dataset Creation Framework

I wanted to create a way to generate more accurate datasets for specific periods of time. So, I built an extensible framework to generate the datasets.

The concept behind the framework is to utilize the BGP community attribute that is included with routing information in order to enhance the filtering of BGP routes.
Some ASes share their community policies on IRR or their webpages, which allows other ASes that wish to communicate with them to properly configure their BGP routers.

Certain ASes employ BGP communities to filter messages to specific regions, label a path as a black hole, differentiate between providers, customers, and peers,

and distinguish between primary and backup routes for a client [29].

It is important to note that there is no motivation for an AS to provide inaccurate community information, as other ASes rely on this information to interpret the values received from the AS.

BGP Community attributes are transitive, which means that their attribute values can be passed through different ASes along a path. Each AS can attach its own Community values without having to remove the Communities previously applied by other ASes. As a result, the BGP Community attribute can contain an array of values defined by different ASes. However, some community paths are used only within the AS and are removed before passing the path to other ASes.

```
prefix type communities
-------------------------------------------------------
3356:123  - Customer route
3356:666  - Peer route
-------------------------------------------------------
```

Listing 7.4. An extract of the IRR for AS3356 (Level3)

```
CUSTOMER COMMUNITIES:
MTS Allstream customers may choose to affect our
local preference from their routers by setting their
routes with the following BGP communities:
Community  Definition Description
---------  ---------- -----------
(default)  Local preference = 100  customer
15290:100  Local preference = 100  customer primary
15290:90   Local preference = 90  customer back-up
15290:30   Local preference = 30  customer fall-back
If a multi-homed customer sets the community to
15290:30,  MTS Allstream will prefer an announcement
from a peer over directly connected customer
---------
```

Listing 7.5. An extract of the IRR for AS15290 (MTS Allstream)

In the RIB 7.6, we can observe that AS3356 (Level3) and AS174 (Cogent) have a peering relationship. This is because the community attribute shows the community 3356:666, which is listed as a peer path in the Lumen3 IRR as we see in 7.4. Moreover, we can infer that the peering relationship is from the USA due to the presence of the communities 3356:575 and 3356:3.

```
Prefix: 82.167.16.0/20, AS_PATH: 28624 3356 174 43766,
   Community: {'3356:575', '3356:901', '28624:3356', '3356:3',
    '3356:86', '3356:666', '3356:2022'}
```

Listing 7.6. A BGP RIB from a route view collector in 1 oct 2023

JSON Representation for filtering rules

In order to establish a dataset creation process that can be scaled up and reproduced, I have created rules for filtering paths and defining relationships. To achieve this, I have utilized the JSON format to create a file that is easy to manage and transfer. It's important to format the files in a specific way:

```
[
    [ASN,[community regex pattern,...], relative position,
       relationship, [community regex pattern to exclude,...],
    [...]
]
```

Listing 7.7. Format rapresentation for filter rules

The first element is the autonomous system number (ASN) of the AS the rule is regarding.

The second element is an array of regex. To match the rule, all regex must be found in at least one correspondence with the community attribute of the list.

The third element represents the relative position of the target AS in the path with respect to the previously indicated ASN. The value can be -1 (left) or 1 (right).

The fourth element defines the relationship as in the following scheme:

```
AS|<customer-as>|-1
AS|<peer-as>|0
AS|<sibling-as>|1
AS|<provider-as>|2
```

The last element is an array of regex. The RIB relationship isn't added if all regex in the array are found in at least one correspondence in the community attribute list.

To give an example, let's consider the plate number AS16030 which is associated with FIBRACAT Telecom.

This AS defines several BGP attributes, including:

- 16030:103 to send the path to all customers
- 16030:1020 to not advertise the path to peers
- 16030:1010 to not advertise all transits
- and 16030:1030 to not advertise to customers

We can use this information to create a rule that identifies all ASes following AS16030 in the path as customers if the BGP community indicates that the path

should be sent to all customers and not advertised to peers or transit.
Moreover, we can exclude paths that match the BGP community attribute indicating not to advertise to customers:

```
[16030, ["16030:103", "16030:1020", "16030:1010"], 1, -1,
    ["16030:1030"]]
```

7.6.2 The execution

The Python script DatasetDownloader.py [65] is designed to download RIBs from various Vantage Points of Route View and Ripe NCC using the pyBGPstream library.

For each RIB, the code checks the rules specified in the JSON file. If a rule is matched, a line is added to the output file following the syntax consistent with asrank, toposcope, and problink:

```
<provider-as>|<customer-as>|-1
<peer-as>|<peer-as>|0
<sibling-as>|<sibling-as>|1
```

The script can be runned easily passing as arguments the start date, the duration in seconds, and the path to the file json with the rules:

```
python3 DatasetDownloader.py -s 10/01/2023 -d 86400 -f
    patterns.json &> out.txt &
```

The output dataset will be saved in the file 'dataset.txt'

7.6.3 The Dataset

I have created multiple datasets [65], one for the first day of every month from July to October 2023, containing around 9900 relationships each.
These datasets include information on about 8000 different Autonomous Systems.

The JSON file used to generate the datased can be found in the appendix A and on github [65].

After uploading them onto the Neo4j database, I checked some of their properties. In Figure 7.6, we observe that the dataset contains many relationships involving Tier 1 Autonomous Systems (ASes).
The dataset includes only 17 ASes in tier 1, yet more than 8000 relationships involve at least one of them.
However, despite this imbalance, there are over 9000 relationships that do not include Tier 1 ASes. This should provide a reasonable validation dataset for the inference algorithms.

For the analysis, the following ASes have been considered in tier 1: 174, 209, 286, 701, 1239, 1299, 2828, 2914, 3257, 3320, 3356, 4436, 5511, 6453, 6461, 6762, 7018, 12956, 3549.

In Graph 7.7a, we have a representation of the dataset from October 10th, 2023, where each bubble represents an AS in the dataset, and the bubble size indicates the number of relationships in the dataset. Looking at the graph, we can see a few autonomous systems with many relationships, and the majority have a lot of transit or peering relationships, but not both.

If we remove the outliers, like AS3356, which is the most represented with 6461 relationships and AS209, with 819 relationships, along with others, we obtain Graph 7.7b. This graph shows that most ASes have only a few relationships, and these lesser-represented ASes are more balanced between peering and transit relationships.

Tier 1 ASes usually have more relationships, even if not indicated in the JSON file. All the big-sized ASes in the dataset are in the pattern file, but only some of the ASes in the pattern file are well-represented in the dataset. Some of them have only a few relationships.

Graph 7.8a shows a difference in the presence of relationships in the validation dataset due to how the dataset was generated.
Meanwhile, Graph 7.8b shows that there are more transit relationships than peering relationships in this particular dataset. This is because the JSON file has more rules regarding transit relationships. However, these factors do not significantly affect the validation process.

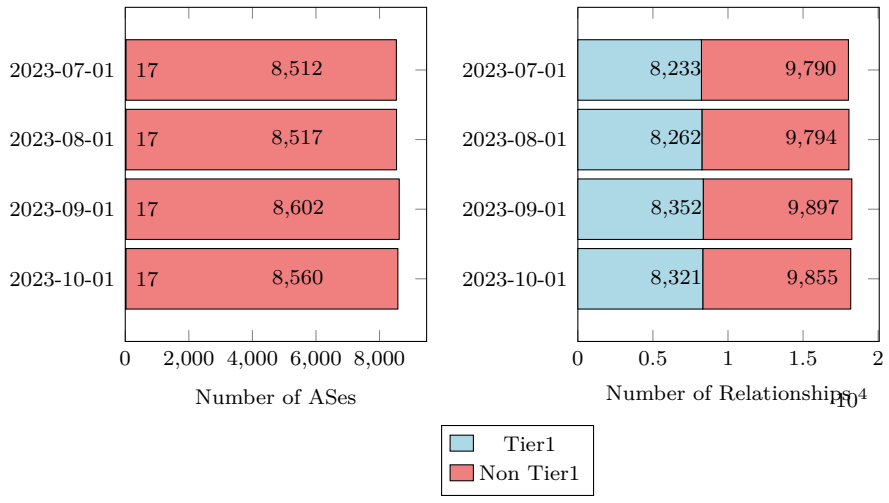
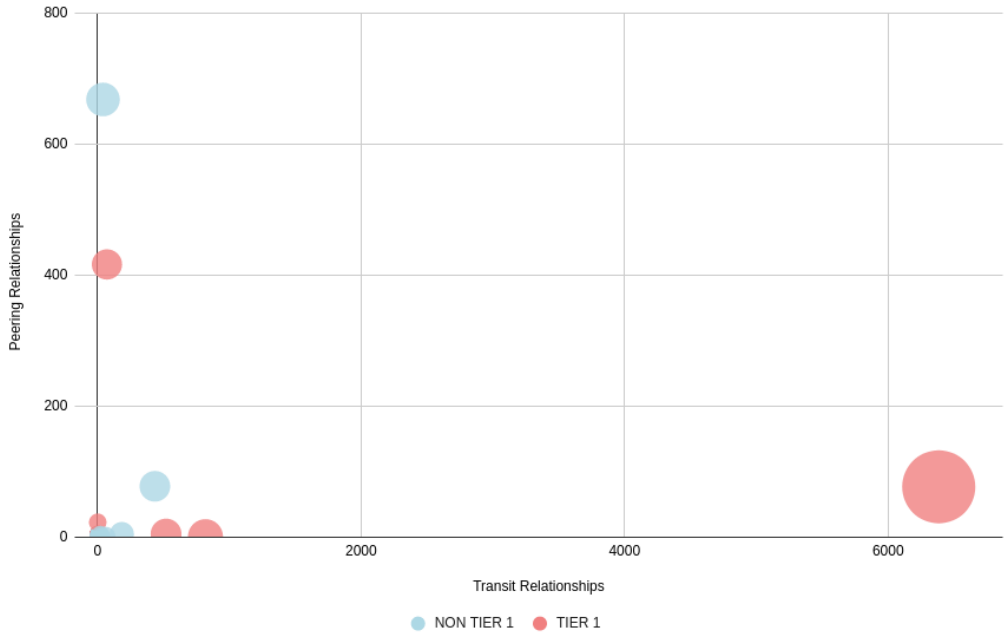
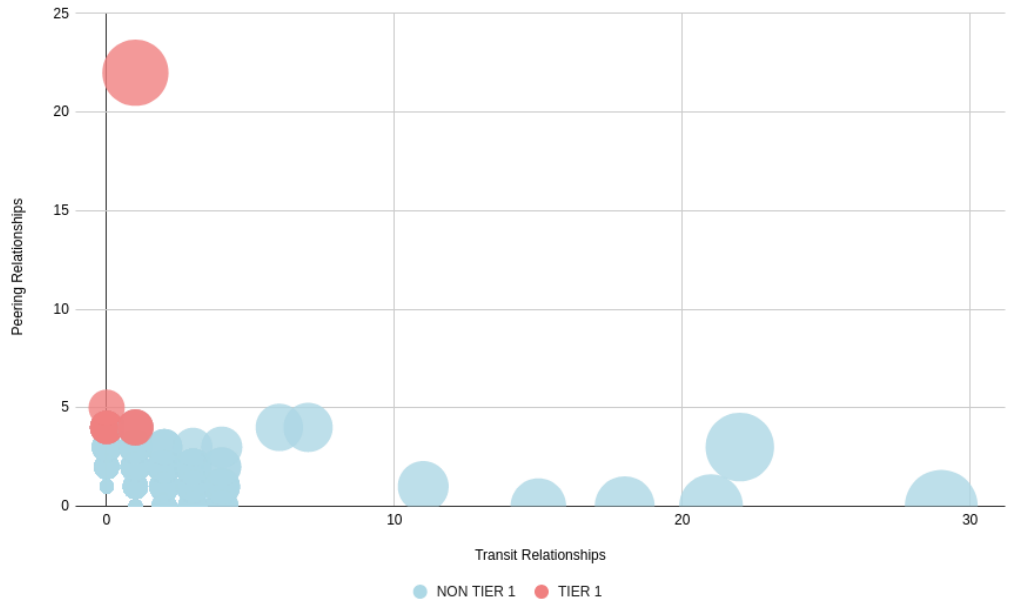


Figure 7.6. Tier 1 presence in the evaluation datasets

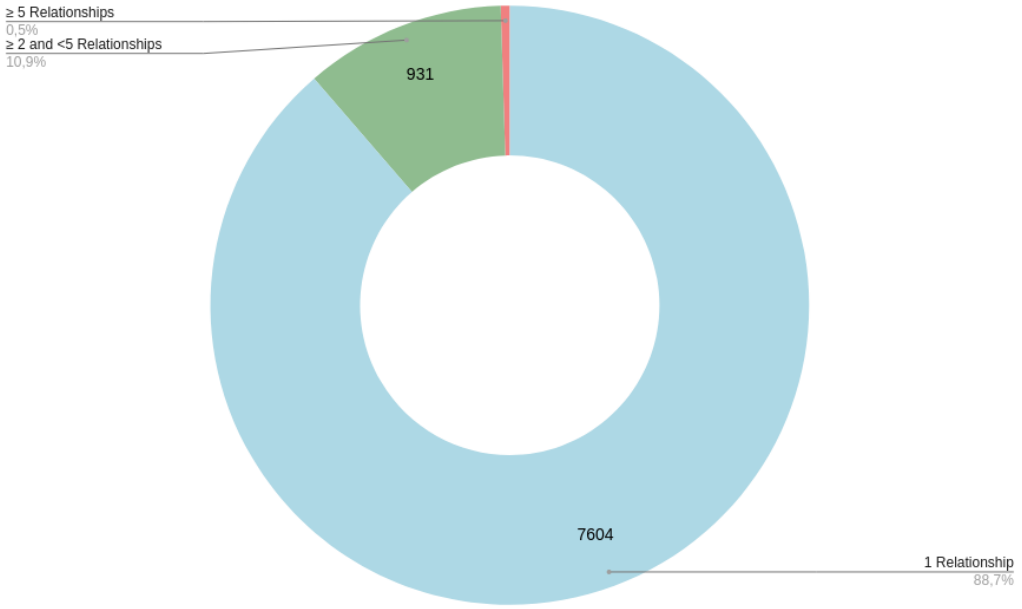


(a) Rapresentation of the evaluation dataset on 1 october 2023.

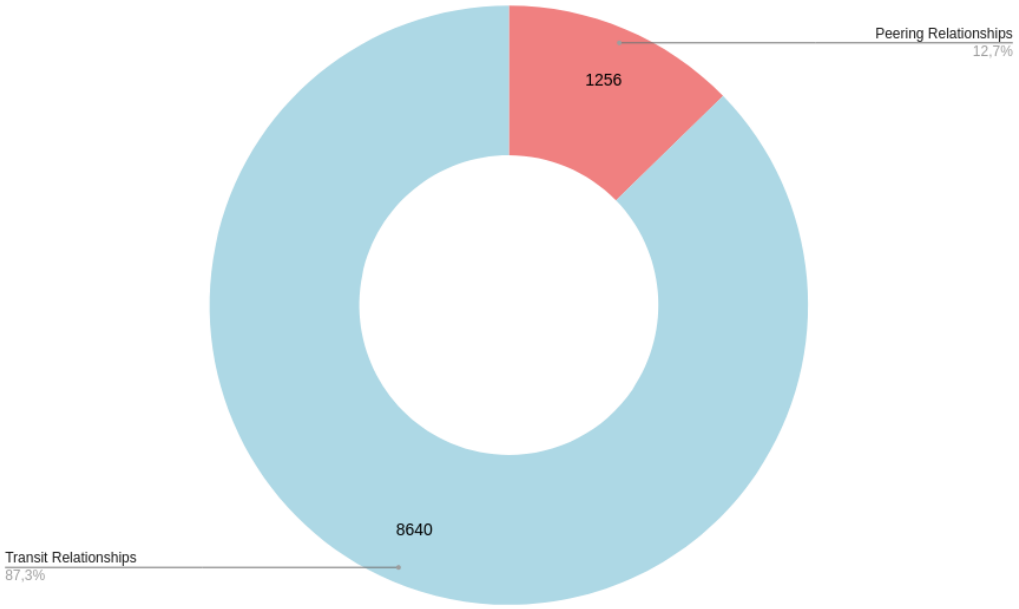


(b) Rapresentation of the evaluation dataset on 1 october 2023 without the outliers.

Figure 7.7. Rapresentation of the evaluation dataset on 1 october 2023.



(a) Number of ASes in the Dataset respect to the number of relationships



(b) Number of Relationships in the Dataset

Figure 7.8. ASes and Relationship in the Dataset of 1 October 2023

7.7 Comparison

The relationships between ASes remain consistent over time, as evidenced by the inference results. This stability in the inter-AS relationships is a notable observation, as it suggests reliability and predictability in the network.
Figures 7.10 and 7.9 shows that the number of ASes and the relationships between them remain stable between months.

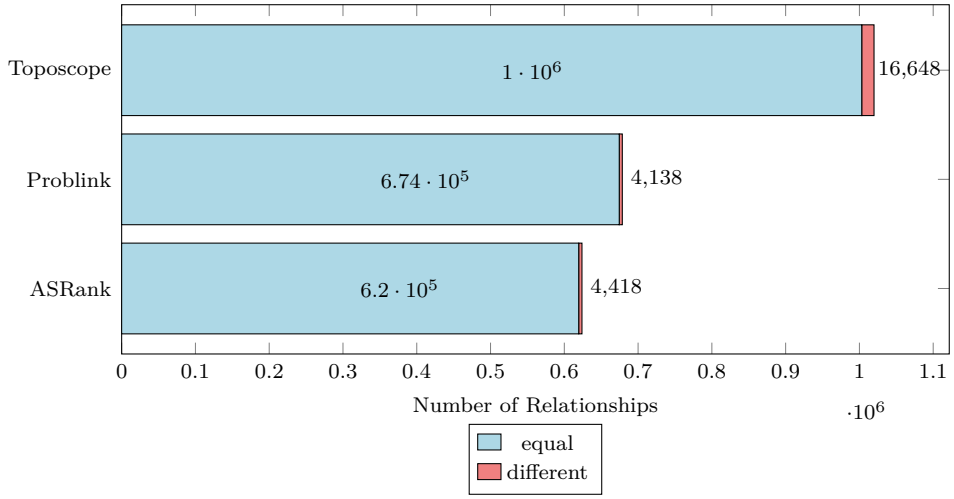


Figure 7.9. Difference between relationships inferred on 1 October 2023 and 1 Septeber 2023

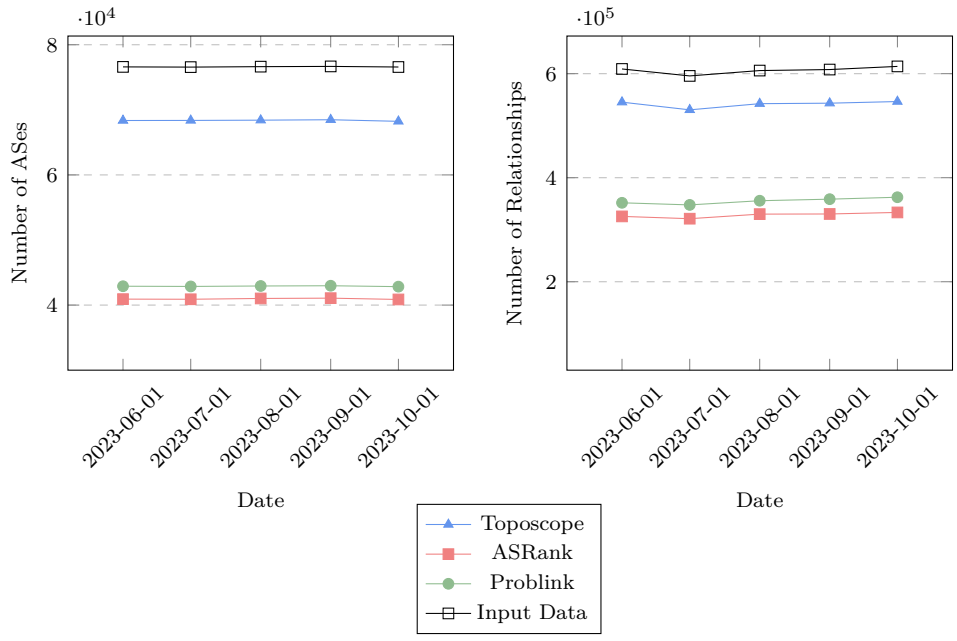


Figure 7.10. Number of ASes and relationships inferred over time

After conducting an initial analysis, it is evident that the Toposcope algorithm

can identify a significantly greater number of relationships than Problink. As shown in Figure 7.11, the input rib files contain 613975 AS couples, and Toposcope has identified relationships between 546413 of these couples while ignoring only 67562 couples that could be attributed to noise, misconfiguration, or route leaks. Substantially, ASRank only considers 54% of relationships, Problink considers 59%, and Toposcope considers 89%.

This difference can also be seen in graphs 7.12. A substantial number of the ASRank labels are confirmed by both Toposcope and Problink, with only a few relationships identified by ASRank not being considered by both Problink and Toposcope.

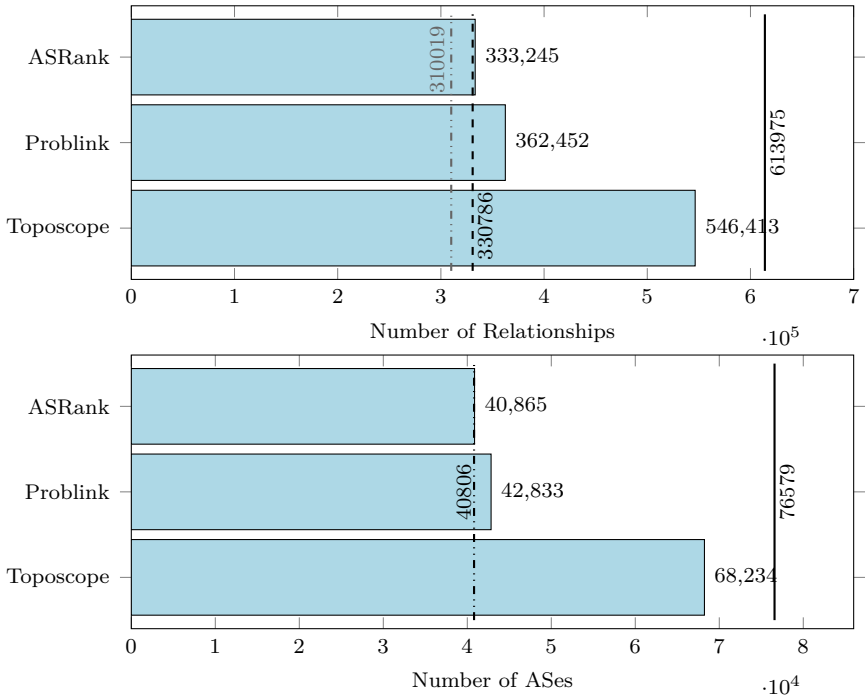


Figure 7.11. Comparison of algorithms for inferences made on 1st October 2023. The dashed-dotted line represents the relationships labelled similarly by all three algorithms. The dashed line represents all the relationships between the same ASes found by all three algorithms. The straight line indicates the number of AS couples in the original RIB file used as input.

The graph 7.12, highlights that the sibling's relationships inferred by Problink are identical to those inferred by Toposcope, as expected. This is because both Problink and Toposcope use data for organization mapping from the same source, CAIDA. However, Toposcope implies more relationships as it considers more Autonomous Systems, possibly due to its less aggressive filtering of input data.

Although time efficiency is not the primary goal of these algorithms, it can be useful to compare their execution times. Based on the data in graph 7.13, we can observe that ASRank takes an average of 32 minutes to execute, which is impressive compared to the 6.12 hours for Toposcope and 12.65 hours for Problink. It is

important to note that both Toposcope and Problink rely on the results of ASRank as input, so the actual time includes data sanitization, ASRank execution, and effective execution time (the time showed in the graph).
The execution time of Toposcope and Problink is undoubtedly affected by excessive RAM and swap memory usage.

I decided to verify the accuracy of the algorithm's results on transit relationships. The algorithms are designed to indicate only one relationship between two ASes. However, in a real environment, there can be multiple relationships between the same ASes, and the validation dataset may not contain all possible existent relationships. If a relationship is present in the dataset, it exists for sure. However, we cannot confirm its existence if it's not in the dataset.
Therefore, I labelled the results as 'true' if they had the same transit relationships as the ones in the database. On the other hand, I labelled them as 'false' if the relationships in the dataset were considered as transit, but in the inference result, they were labelled as peering or as transit relationship with the wrong direction.

I didn't consider the relationships labelled as siblings by Problink and Toposcope because it would have required validating the CAIDA organization mapping dataset, which was used to infer these relationships, but that's not the focus of this thesis. For a fair validation, these relationships are not even considered in ASRank.

In Figure 7.14, we can observe that all the algorithms perform well in identifying

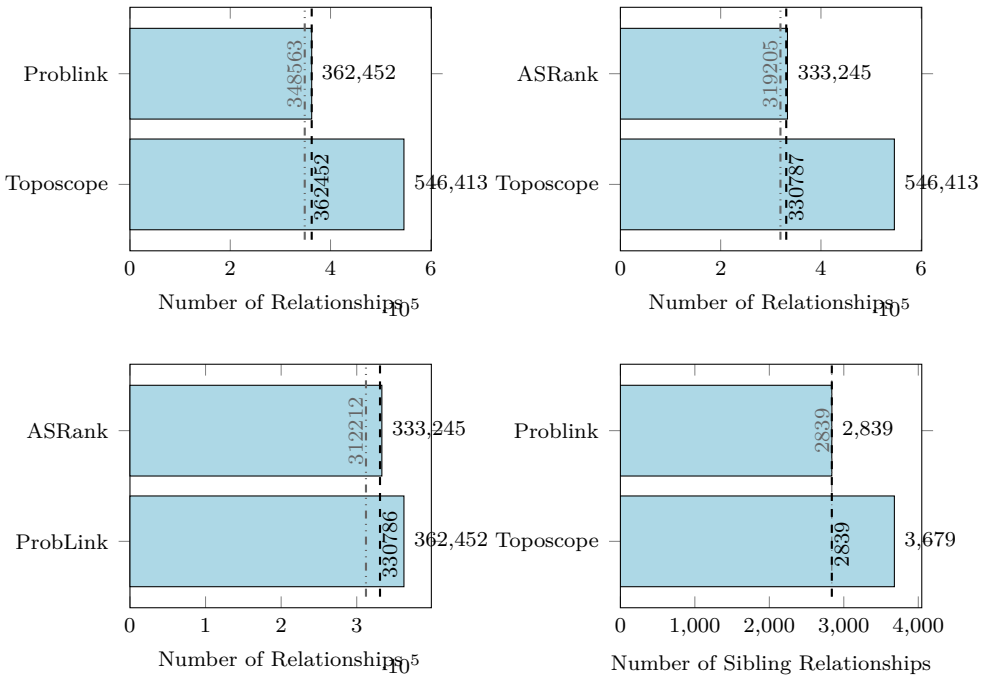


Figure 7.12. Algorithm Comparison for Inferences Made on 1st October 2023. The red dash-dotted lines indicate relationships that are equally labeled by both algorithms, while the blue dashed line represents the relationships between the same ASes.

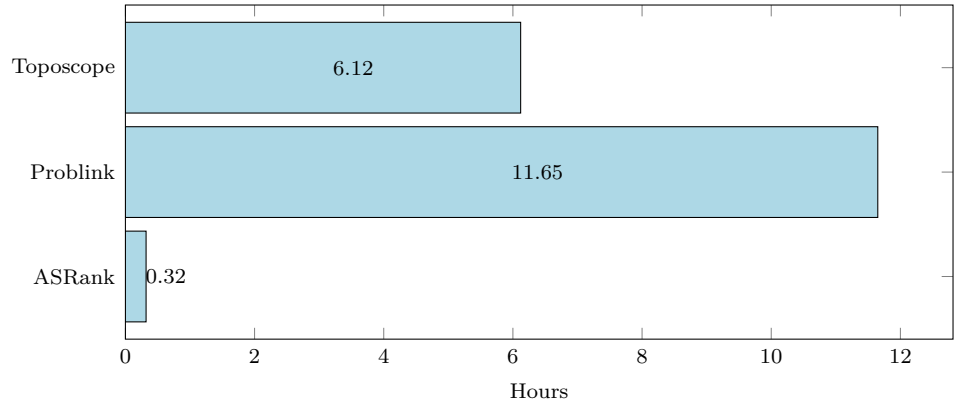


Figure 7.13. Average time of execution of the inference algorithms

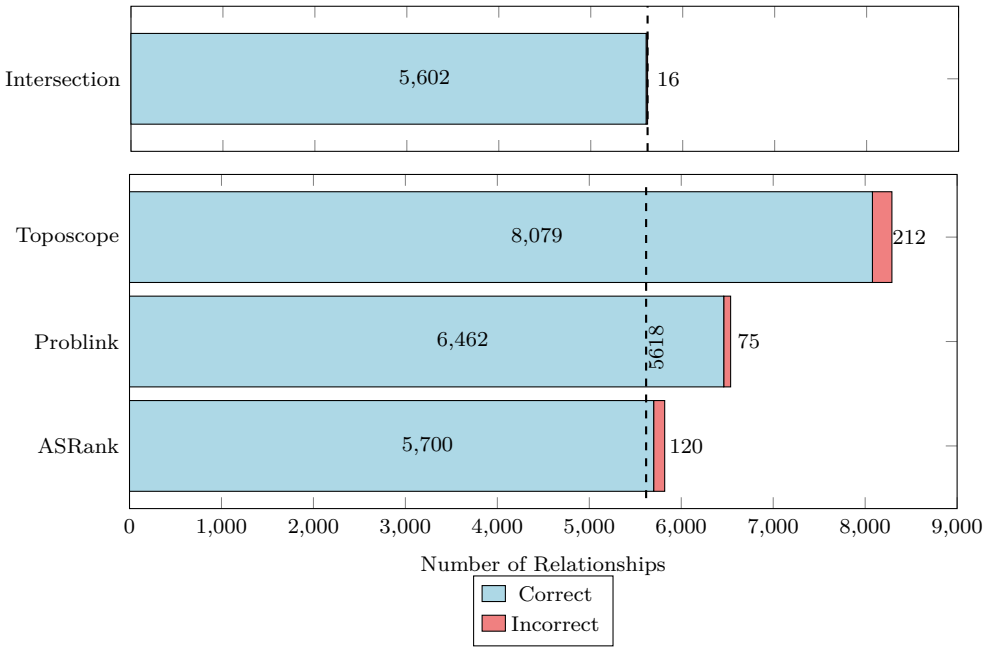


Figure 7.14. Comparation verified labels on Transit relationships

transit relationships. However, the intersection graph yields the most accurate results, with a 99.71% success rate. This is exceptional, particularly when compared to ASRank, which counts slightly fewer relationships.
Overall, Problink had the highest accuracy rate (98.85%), followed by ASRank (97.93%) and Toposcope (97.44%).

If we compare all three algorithm on the same subset of relatioships we can confirm that Problink is more accurate (98.79%) but it is followed by Toposcope (98.79%) and ASRank at the last (97.93%)

In previous observations (Figure 7.10), we noticed that relationships remain constant over time. However, as depicted in Figure 7.16, we can observe a significant

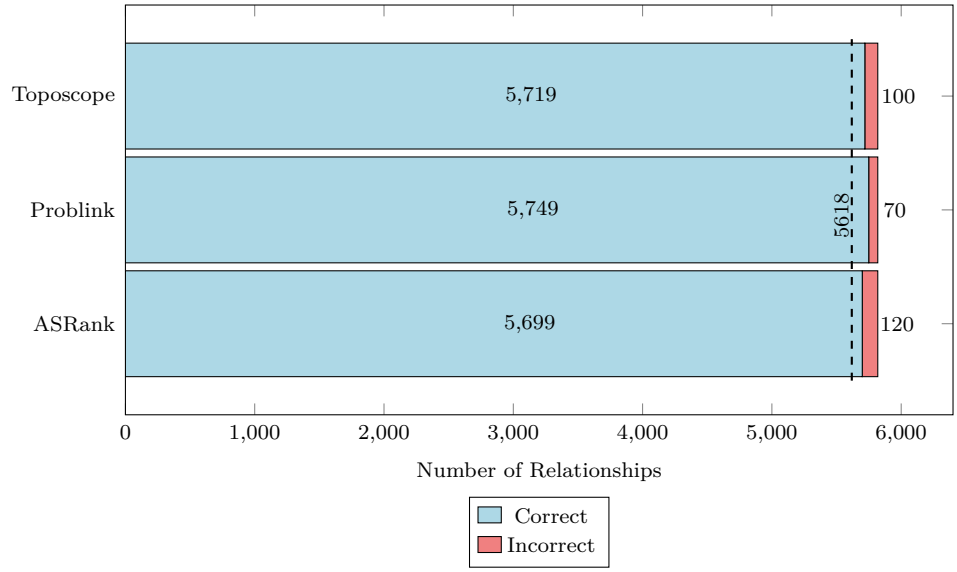


Figure 7.15. Comparation verified labels on Transit relationships on the same subset

difference in accuracy between relationships that resemble those seen in the past and those that differ from them.
The relationships labelled in the same way as in the past are much more accurate in all algorithms.

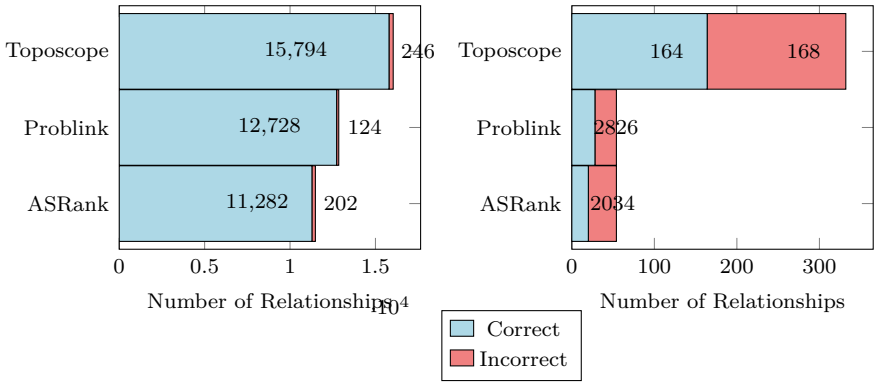


Figure 7.16. Difference between 01 October 2023 and 01 Septeber 2023 respect to the validation results. On the left the relationships equal to the past. On the right the relationships different from the past

Graph 7.17, shows that although there are only 19 tier 1 Autonomous Systems (ASes), they are involved in a significant portion of the relationships.
Moving on to Figure 7.18, we can observe that relationships involving tier 1 ASes are more accurate than those that do not involve them. However, it's worth noting that we have more samples involving tier 1 ASes in the validation dataset.

Toposcope outperforms all other methods in inferring non-tier 1 relationships (96.78%), but performs the worst in inferring tier 1 AS relationships (97.51%). On

the other hand, Problink performs the best in inferring tier 1 AS relationships (99.37%), but it is the second when inferring relationships with non-tier 1 ASes (92.4%). ASRank is positioned in the middle when inferring tier 1 AS relationships with 98.53% accuracy and it is the worst in inferring non-tier 1 AS relationships having 92.42% accuracy

Despite the excellent results achieved by Problink, it is important to note that this algorithm is the only one that failed to identify the direction of some transit links. Specifically, it mislabeled three links involving tier 1 and large-sized ASes, which were present in the validation dataset. The links are AS20940-AS209, AS11996-AS209, and AS13335-AS3491.

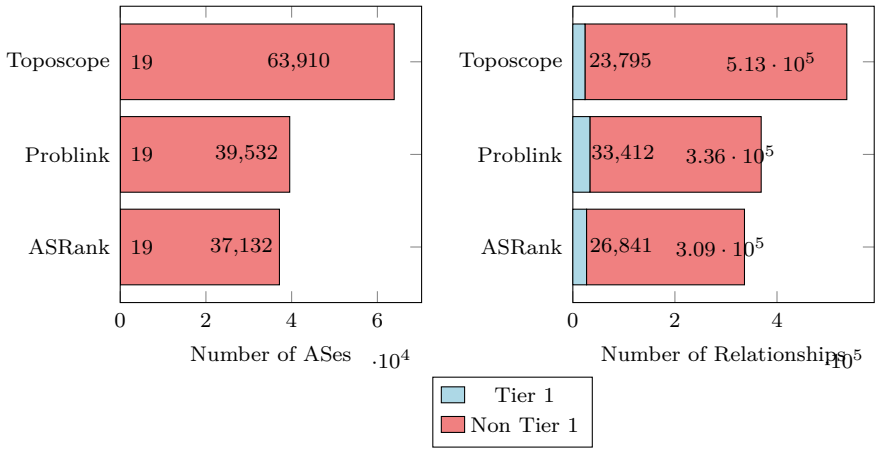


Figure 7.17. Tier 1 presence in the Algorithms on 01 October 2023

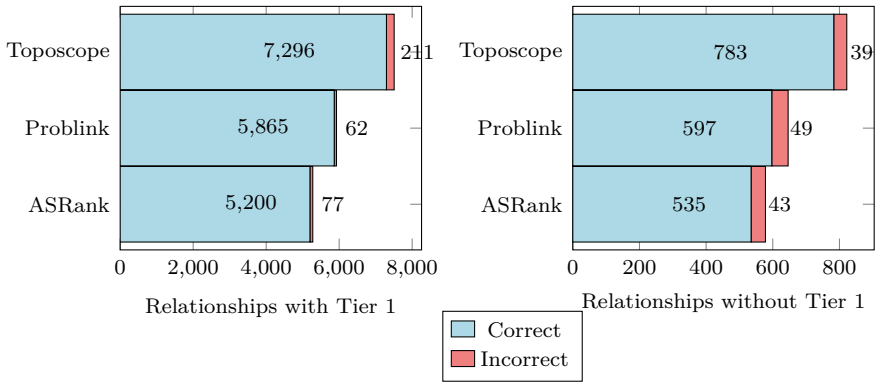


Figure 7.18. Tier 1 transit validation results per algorithm

In Figure 7.19, we can observe the outcomes for relationships that are not marked by ASRank, but are taken into account by both Problink and Toposcope. In most cases, both algorithms agree on the label, resulting in a success rate of 99.68%. However, when they disagree, Problink correctly labels the relationship 95.55% of the time.

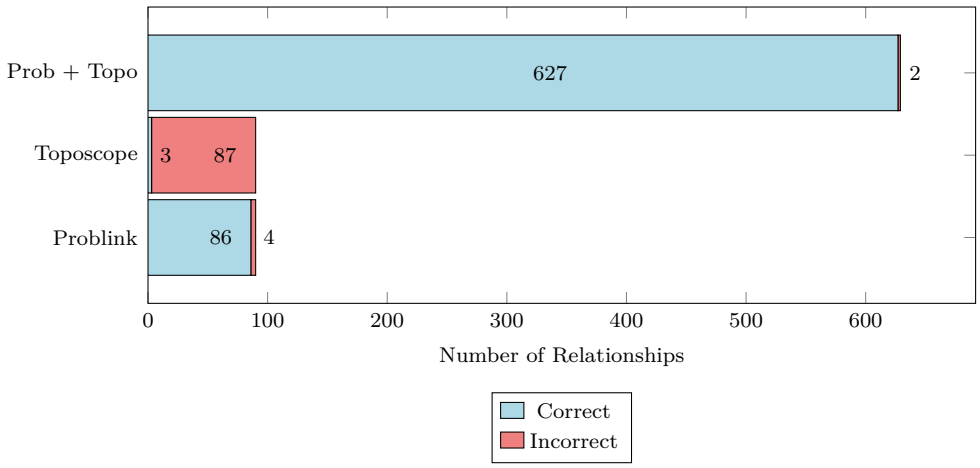


Figure 7.19. Comparison between transit validation results where the relationships are labeled equally by Problink and Toposcope, and where they are labeled differently. Only relationships not labeled by ASRank are considered.

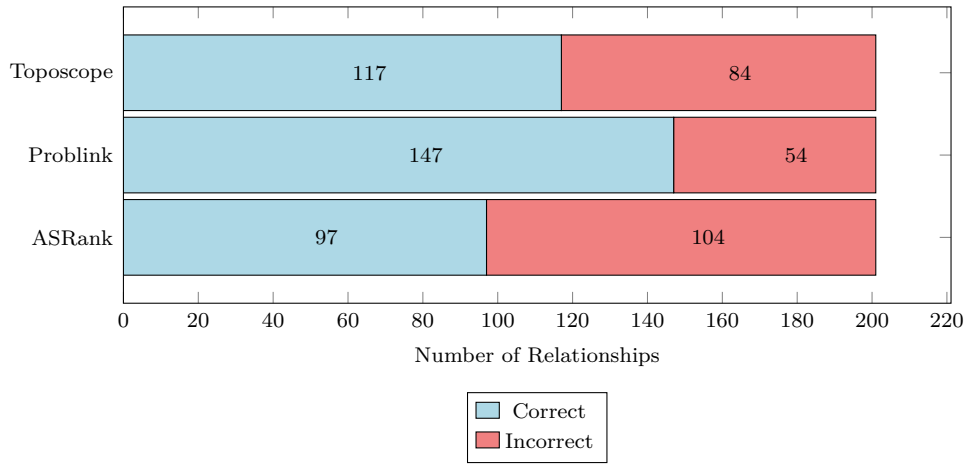


Figure 7.20. Comparation between validation results where the relationships are labeled differently by at least one algorithm

Figure 7.20, shows the case where all three algorithms label the same relationship, but at least one gives a different result. Even in this scenario, Problink performs better (73.13%) than Toposcope (58.2%) and ASRank (48.25%).

It is crucial to evaluate cases where only one algorithm labels a relationship (Figure 7.21).
In such cases, Problink does not consider relationships more than Toposcope does. However, ASRank considers 2457 relationships that are not considered by other algorithms. Unfortunately, there isn't enough data in the validation dataset to evaluate these relationships. Out of the 2457 relationships, only one is in the validation dataset, confirming the ASRank result.
Toposcope infers many more relationships than the other algorithms, and in this portion of data with an accuracy rate of 98.63%.

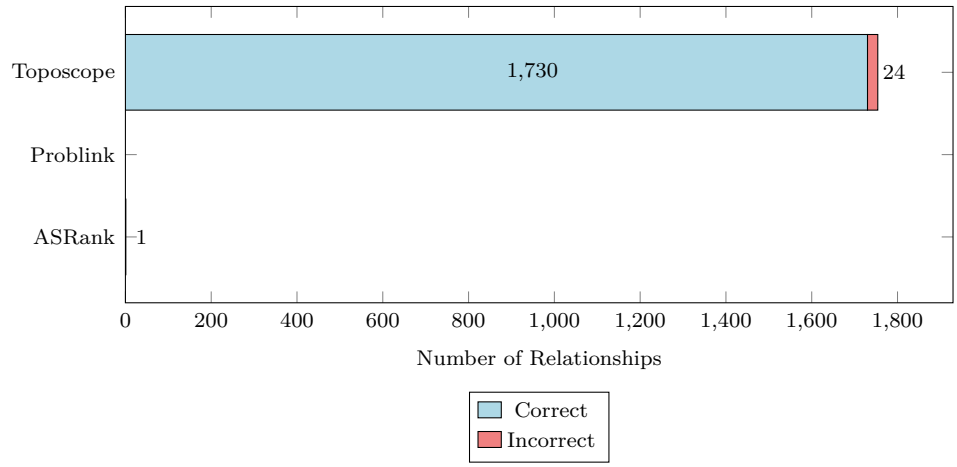


Figure 7.21. Comparation between validation results where only one algorithm labeled a relationships

Chapter 8

Future Directions and Recommendations

The framework for generating datasets, the analysis of inference algorithms, and the handling and visualizing relationships between Autonomous Systems can open up a lot of possibilities for future work.
The dataset creation framework can be improved by adding an automated way to extract and parse BGP community rules from IRR and other sources.
Neural networks and natural language processing approaches could be a good start to handle this non-homogeneous data that must be interpreted correctly.

Expanding the ruleset could benefit the dataset creation in many ways.
The existing script allows multiple JSON files, so creating different JSON files with other purposes could be helpful.

The datasets that can be generated have the potential to serve a multitude of purposes and can be leveraged for various applications and fields.
Using the dataset as a training set for relationship inferences based on a machine learning approach, such as BGP2Vec, could be interesting.

Another wise usage of this dataset creation framework could be integrated into an inference algorithm that calculates probabilities from this set of relationships.
The results can be even more accurate with more trustworthy sources of information.

In general, a mixed approach to relationship inference could be helpful. The community attribute can be seen as a fundamental part of the probability calculus, in addition to paths.

Furthermore, the analysis results can be combined with ASRank, Toposcope, and Problink to draw new inferences and assign a confidence level to each label.

There is room for further exploration of the data. Anyone who wishes to delve deeper may do so, as I have left all of my findings open for future reference [65].

Chapter 9

Conclusions

I have devised a technique to create validation datasets that can be used to test inference algorithms in real-world scenarios, complete with noise and complex agreements between Autonomous Systems (ASes).

I have analyzed three inference algorithms - ASRank, Problink, and Toposcope - and have made some observations that could be helpful for future studies.

My findings indicate that Problink performs better than both Toposcope and ASRank regarding overall accuracy and has a shorter execution time than Toposcope. However, Toposcope is better than the other two algorithms at inferring more relationships, especially with non-tier 1 ASes.

Moreover, the intersection of the algorithms produces the best accuracy, and relationships labelled by the same algorithm as in the past are more accurate.

These insights can benefit researchers working on inference algorithms and help them achieve better results.

Appendix A

JSON pattern file

This is the json file that describe the rules used to generate the datased for the inference data evaluation.

```
[
    [1273, ["1273:1\\d{4}"], 1, -1, []],
    [577, ["577:140"], 1, -1, []],
    [577, ["577:130"], 1, -1, []],
    [209, ["209:209"], 1, -1, []],
    [57187, ["57187:1000"], 1, 2, []],
    [57187, ["57187:2500"], 1, 0, []],
    [57187, ["57187:3000"], 1, -1, []],
    [58511, ["65001:0"], 1, 0, []],
    [58511, ["58511:200"], 1, 0, []],
    [58511, ["65004:0", "65002:0"], 1, 2, []],
    [49544, ["65004:\\d{1}6\\d{3}"], 1, 2, []],
    [49544, ["65004:\\d{1}5\\d{3}"], 1, 2, []],
    [49544, ["65004:\\d{1}4\\d{3}"], 1, 0, []],
    [49544, ["65004:\\d{1}2\\d{3}"], 1, -1, []],
    [3491, ["3491:9001"], 1, -1, []],
    [3491, ["3491:9002"], 1, 0, []],
    [6762, ["6762:40"], 1, -1, []],
    [3356, ["3356:123"], 1, -1, []],
    [3356, ["3356:666"], 1, 0, []],
    [15290, ["15290:444"], 1, -1, []],
    [174, ["174:21000"], 1, 0, []],
    [174, ["174:21100"], 1, 0, []],
    [12779, ["12779:65000"], 1, -1, []],
    [12779, ["12779:65097"], 1, 0, []],
    [12779, ["12779:65098"], 1, 2, []],
    [6461, ["6461:5997"], 1, 0, []],
    [6461, ["6461:5998"], 1, -1, []],
    [8374, ["8374:310\\d{1}"], 1, 0, []],
    [8374, ["8374:320\\d{1}"], 1, 0, []],
    [8374, ["8374:330\\d{1}"], 1, 0, []],
    [8374, ["8374:340\\d{1}"], 1, 0, []],
    [8374, ["8374:350\\d{1}"], 1, 0, []],
    [5400, ["5400:\\d{1}300"], 1, 0, []],
    [19092, ["19092:100"], 1, -1, []],
```

```
36      [19092, ["19092:320"], 1, 0, []],
37      [19092, ["19092:310"], 1, 0, []],
38      [12732, ["12732:9001"], 1, -1, []],
39      [8001, ["8001:30\\d{2}[1-3]"], 1, 0, []],
40      [8001, ["8001:30\\d{2}"], 1, 0, []],
41      [8001, ["8001:40\\d{2}"], 1, -1, []],
42      [8001, ["8001:40\\d{2}[1-3]"], 1, -1, []],
43      [8001, ["8001:200\\d{1}"], 1, 0, []],
44      [8001, ["8001:200\\d{1}[1-3]"], 1, 0, []],
45      [8928, ["8928:65291"], 1, 0, []],
46      [29838, ["29838:10"], 1, -1, []],
47      [29838, ["29838:50"], 1, -1, []],
48      [29838, ["29838:100"], 1, -1, []],
49      [6667, ["6667:90\\d{1}"], 1, -1, []],
50      [6667, ["6667:3000"], 1, -1, []],
51      [6667, ["6667:3001"], 1, 0, []],
52      [6667, ["6667:3003"], 1, 0, []],
53      [11164, ["11164:52200"], 1, 0, []],
54      [11164, ["11164:52200"], 1, 0, []],
55      [2683, ["2683:1"], 1, -1, []],
56      [2683, ["2683:2"], 1, 0, []],
57      [2683, ["2683:3"], 1, 2, []],
58      [3561, ["3561:1\\d{4}"], 1, -1, []],
59      [3561, ["3561:2\\d{4}"], 1, 0, []],
60      [22773, ["22773:10100"], 1, -1, []],
61      [22773, ["22773:10100"], 1, -1, []],
62      [5580, ["5580:\\d{1}[156]\\d{3}"], 1, -1, []],
63      [5580, ["5580:\\d{1}[43]\\d{3}"], 1, 0, []],
64      [5580, ["5580:\\d{1}3\\d{3}"], 1, 0, []],
65      [7922, ["7922:888"], 1, -1, []],
66      [9026, ["9026:3000"], 1, -1, ["9026:1000"]],
67      [9026, ["9026:3001"], 1, -1, []],
68      [9026, ["9026:5000"], 1, 0, []],
69      [9026, ["9026:0002"], 1, -1, []],
70      [9026, ["9026:1000"], 1, 2, ["9026:3000"]],
71      [16030, ["16030:1010"], 1, 0, []],
72      [16030, ["16030:103", "16030:1020", "16030:1010"], 1, -1,
           ["16030:1030"]],
73      [16030, ["16030:103\\d{1}", "16030:1020", "16030:1010"],
           1, -1, ["16030:1030"]],
74      [5400, ["5400:\\d{1}300"], 1, 0, []],
75      [7132, ["7132:8888"], 1, -1, []],
76      [8308, ["8308:60000"], 1, -1, []],
77      [8308, ["8308:60100"], 1, -1, []],
78      [8308, ["8308:60050"], 1, 0, []],
79      [7474, ["7474:100"], 1, -1, []],
80      [7474, ["7474:80"], 1, -1, []],
81      [7474, ["7474:90"], 1, -1, []],
82      [7474, ["7474:70"], 1, -1, []],
83      [2764, ["2764:65408"], 1, -1, []],
84      [15756, ["15756:2009","15756:3009"], 1, -1, []],
85      [6663, ["6663:135"], 1, -1, []],
86      [6663, ["6663:125"], 1, -1, []],
```

```
    [6663, ["6663:40100", "6663:40200"], 1, -1, []],
    [8897, ["8897:64900"], 1, 0, []],
    [8897, ["8897:64850 "], 1, 0, []],
    [8897, ["8897:65000 "], 1, -1, []],
    [8897, ["8897:64901"], 1, 0, []],
    [45177, ["45177:39999"], 1, -1, []],
    [8218, ["8218:1010"], 1, -1, []]
]
```

Appendix B

Cypher queries

I provide a series of queries to comprehensively understand the data management techniques employed in the neo4j database.

The following query returns relationships that are equal for different algorithms. The same result can be obtained by querying for relationships with the attribute comparison = 0 when available.

```
MATCH (t:Time)-[]-(a:AutonomousSystem)-[r1]-(b:AutonomousSystem)
WHERE ID(a) < ID(b) AND r1.algorithm IN [‘toposcope’, ‘asrank’, ‘problink’] AND
    toString(t.inference_date) = ‘2023-10-01T00:00:00’
WITH a, b, COLLECT(r1) AS relationships, date(t.inference_date) AS date, t
WHERE
    SIZE(relationships) = 3 AND
    ALL(rel IN relationships WHERE rel.algorithm IN [‘toposcope’, ‘asrank’,
    ‘problink’]) AND
    ALL(rel IN relationships WHERE TYPE(rel) = TYPE(relationships[0])) AND
    ALL(rel IN relationships WHERE TYPE(rel) <> ‘GIVE_TRANSIT_TO’ OR (startNode(rel) = startNode(relationships[0])))
    WITH date, a, b, relationships,
    CASE WHEN TYPE(relationships[0]) = ‘PEER_OF’ THEN 0 WHEN TYPE(relationships[0]) = ‘GIVE_TRANSIT_TO’ THEN -1 ELSE null END AS relation,
    CASE WHEN TYPE(relationships[0]) = ‘PEER_OF’ THEN a.asn WHEN TYPE(relationships[0]) = ‘GIVE_TRANSIT_TO’ AND startNode(relationships[2]).asn = a.asn THEN a.asn WHEN TYPE(relationships[0]) = ‘GIVE_TRANSIT_TO’ AND startNode(relationships[2]).asn = b.asn THEN b.asn ELSE null END AS as1,
    CASE WHEN TYPE(relationships[0]) = ‘PEER_OF’ THEN b.asn WHEN TYPE(relationships[0]) = ‘GIVE_TRANSIT_TO’ AND startNode(relationships[2]).asn = a.asn THEN b
```

```
            .asn WHEN TYPE(relationships[0]) = 'GIVE_TRANSIT_TO
            ' AND startNode(relationships[2]).asn = b.asn THEN
            a.asn ELSE null END AS as2
    RETURN as1, as2, relation;
```

This query will retrieve the number of transit relationships verified using the community dataset:

```
MATCH (d:Dataset{name:'community',inference_date:
    1696118400})-[]-(a:AutonomousSystem)-[r:GIVE_TRANSIT_TO
    ]-(b:AutonomousSystem)
WHERE ID(a) < ID(b)
WITH a,b, r
MATCH (t:Time) WHERE toString(t.inference_date) =
    '2023-10-01T00:00:00'
WITH a,b, r, t
MATCH (t)-[]-(a2:AutonomousSystem{asn:a.asn})-[r2]-(b2:
    AutonomousSystem{asn:b.asn})
SET r2.cverified_transit = (CASE WHEN TYPE(r2)=TYPE(r) AND
     startNode(r).asn = startNode(r2).asn THEN true ELSE
    false END)
RETURN COUNT(r2)
```

This query will return the number of ASes and Relationships for each evaluation dataset in the database:

```
MATCH(d:Dataset{name:'community'})-[]-(a:AutonomousSystem)
    -[r]-(b:AutonomousSystem)
WHERE
    a.asn in ['174', '209', '286', '701', '1239', '1299',
        '2828', '2914', '3257', '3320', '3356', '4436',
        '5511', '6453', '6461', '6762', '7018', '12956',
        '3549'] OR
    b.asn in ['174', '209', '286', '701', '1239', '1299',
        '2828', '2914', '3257', '3320', '3356', '4436',
        '5511', '6453', '6461', '6762', '7018', '12956',
        '3549']
RETURN count(DISTINCT a) as ases, count(DISTINCT r) as rel
    , d.inference_date
```

Bibliography

[1] *A Brief History of the Internet's Biggest BGP Incidents*. June 2023. URL: https://www.kentik.com/blog/a-brief-history-of-the-internets-biggest-bgp-incidents/ (visited on 09/03/2023).

[2] *A Tale of Two BGP Leaks*. Aug. 2023. URL: https://www.kentik.com/blog/a-tale-of-two-bgp-leaks/ (visited on 09/03/2023).

[3] *Archipelago (Ark) Measurement Infrastructure*. Dec. 2006. URL: https://www.caida.org/projects/ark/ (visited on 11/08/2023).

[4] *AS Rank: A Ranking of the Largest Autonomous Systems (AS) in the Internet*. URL: https://asrank.caida.org/ (visited on 11/08/2023).

[5] Alexander Azimov et al. *A Profile for Autonomous System Provider Authorization*. Internet-Draft draft-ietf-sidrops-aspa-profile-16. Internet Engineering Task Force / Internet Engineering Task Force, July 2023. 14 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-profile/16/.

[6] Alexander Azimov et al. *BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects*. Internet-Draft draft-ietf-sidrops-aspa-verification-16. Internet Engineering Task Force / Internet Engineering Task Force, Aug. 2023. 23 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/16/.

[7] Marcelo Bagnulo et al. "Practicable Route Leak Detection and Protection with ASIRIA". In: *Computer Networks* 211 (July 2022), p. 108966. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2022.108966. URL: https://www.sciencedirect.com/science/article/pii/S1389128622001402 (visited on 07/19/2023).

[8] Sara Bakkali, Hafssa Benaboud, and Mouad Ben Mamoun. "Security Problems in BGP: An Overview". In: *2013 National Security Days (JNS3)*. Apr. 2013, pp. 1–5. DOI: 10.1109/JNS3.2013.6595458.

[9] Iljitsch van Beijnum. *BGP: Building Reliable Networks with the Border Gateway Protocol*. "O'Reilly Media, Inc.", Sept. 2002. ISBN: 978-1-4493-9082-2.

[10] *BGP Leaks and Cryptocurrencies*. Apr. 2018. URL: http://blog.cloudflare.com/bgp-leaks-and-crypto-currencies/ (visited on 02/23/2023).

[11] *BGP Support for TTL Security Check*. URL: https://www.cisco.com/c/en/us/td/docs/ios/12_2sx/feature/guide/fsxebtsh.html (visited on 09/05/2023).

[12] *BGP.Tools*. URL: https://bgp.tools/ (visited on 11/08/2023).

[13] *BGP2VEC Project Page*. URL: https://talshapira.github.io/portfolio/bgp2vec/ (visited on 09/14/2023).

[14] *BGP2Vec: Unveiling the Latent Characteristics of Autonomous Systems | IEEE Journals & Magazine | IEEE Xplore*. URL: https://ieeexplore.ieee.org/document/9761992 (visited on 10/09/2023).

[15] *BGPmon | BGPmon*. URL: https://bgpmon.net/ (visited on 11/08/2023).

[16] *BGPStream*. URL: https://bgpstream.caida.org/ (visited on 11/08/2023).

[17] *BGPView - BGP Toolkit and BGP ASN Routing Lookup Tool*. URL: https://bgpview.io/ (visited on 11/08/2023).

[18] *Border Gateway Protocol Napkin*. URL: http://ciscoarchive.lunaimaging.com/luna/servlet/detail/CHMC~4~4~265~943:Border-Gateway-Protocol-Napkin (visited on 12/21/2023).

[19] *CAIDA*. URL: https://www.caida.org/ (visited on 11/08/2023).

[20] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Aug. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. arXiv: 1603.02754 [cs]. URL: http://arxiv.org/abs/1603.02754 (visited on 09/13/2023).

[21] *CIDR Report*. URL: https://www.cidr-report.org/as2.0/ (visited on 08/05/2023).

[22] Avichai Cohen et al. "Jumpstarting BGP Security with Path-End Validation". In: *Proceedings of the 2016 ACM SIGCOMM Conference*. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 342–355. ISBN: 978-1-4503-4193-6. DOI: 10.1145/2934872.2934883. URL: https://dl.acm.org/doi/10.1145/2934872.2934883 (visited on 08/06/2023).

[23] Giovanni Comarela, Evimaria Terzi, and Mark Crovella. "Detecting Unusually-Routed ASes: Methods and Applications". In: Nov. 2016, pp. 445–459. DOI: 10.1145/2987443.2987478.

[24] Sean Convery. *An Attack Tree for the Border Gateway Protocol*. Internet-Draft draft-ietf-rpsec-bgpattack-00. Internet Engineering Task Force / Internet Engineering Task Force, Apr. 2004. 23 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-rpsec-bgpattack/00/.

[25] Wenping Deng et al. "Shedding Light on the Use of AS Relationships for Path Inference". In: *Journal of Communications and Networks* 14.3 (2012), pp. 336–345. DOI: 10.1109/JCN.2012.6253094.

[26] Amogh Dhamdhere et al. "Inferring Persistent Interdomain Congestion". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '18. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–15. ISBN: 978-1-4503-5567-4. DOI: 10.1145/3230543.3230549. URL: https://dl.acm.org/doi/10.1145/3230543.3230549 (visited on 08/06/2023).

[27] G. Di Battista, M. Patrignani, and M. Pizzonia. "Computing the Types of the Relationships between Autonomous Systems". In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. Vol. 1. Mar. 2003, 156–165 vol.1. DOI: 10.1109/INFCOM.2003.1208668.

[28] Xenofontas Dimitropoulos et al. "AS Relationships: Inference and Validation". In: *ACM SIGCOMM Computer Communication Review* 37.1 (Jan. 2007), pp. 29–40. ISSN: 0146-4833. DOI: 10.1145/1198255.1198259. arXiv: cs/0604017. URL: http://arxiv.org/abs/cs/0604017 (visited on 08/06/2023).

[29] Benoit Donnet and Olivier Bonaventure. "On BGP Communities". In: *SIGCOMM Comput. Commun. Rev.* 38.2 (Mar. 2008), pp. 55–59. ISSN: 0146-4833. DOI: 10.1145/1355734.1355743. URL: https://doi.org/10.1145/1355734.1355743.

[30] *ECFS - Filing Details*. URL: https://www.fcc.gov/ecfs/search/search-filings/filing/1091496862125 (visited on 09/02/2023).

[31] Qilin Fan et al. "Video Delivery Networks: Challenges, Solutions and Future Directions". In: *Computers & Electrical Engineering* 66 (Feb. 2018), pp. 332–341. ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2017.04.011. URL: https://www.sciencedirect.com/science/article/pii/S0045790617308972 (visited on 08/06/2023).

[32] Guoyao Feng, Srinivasan Seshan, and Peter Steenkiste. *PARI: A Probabilistic Approach to AS Relationships Inference*. May 2019. DOI: 10.48550/arXiv.1905.02386. arXiv: 1905.02386 [cs]. URL: http://arxiv.org/abs/1905.02386 (visited on 09/14/2023).

[33] *For a Safer Internet - Instructions to Use RPKI*. Mar. 2020. URL: https://labs.ripe.net/author/flavio_luciani_1/for-a-safer-internet-instructions-to-use-rpki/ (visited on 09/05/2023).

[34] Sylvain Frey et al. "It Bends But Would It Break? Topological Analysis of BGP Infrastructures in Europe". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. Mar. 2016, pp. 423–438. DOI: 10.1109/EuroSP.2016.39.

[35] Lixin Gao. "On Inferring Autonomous System Relationships in the Internet". In: *IEEE/ACM Transactions on Networking* 9.6 (Dec. 2001), pp. 733–745. ISSN: 1558-2566. DOI: 10.1109/90.974527.

[36] Vasileios Giotsas and Shi Zhou. "Valley-Free Violation in Internet Routing — Analysis Based on BGP Community Data". In: *2012 IEEE International Conference on Communications (ICC)*. 2012, pp. 1193–1197. DOI: 10.1109/ICC.2012.6363987.

[37] Sharon Goldberg et al. "Rationality and Traffic Attraction: Incentives for Honest Path Announcements in Bgp". In: *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*. SIGCOMM '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 267–278. ISBN: 978-1-60558-175-0. DOI: 10.1145/1402958.1402989. URL: https://doi.org/10.1145/1402958.1402989.

[38] Dan Goodin. *"Suspicious" Event Routes Traffic for Big-Name Sites through Russia*. Dec. 2017. URL: https://arstechnica.com/information-technology/2017/12/suspicious-event-routes-traffic-for-big-name-sites-through-russia/ (visited on 09/03/2023).

[39] *Google Leaked BGP Prefixes Knocked Japan off the Internet*. Aug. 2017. URL: https://www.internetsociety.org/blog/2017/08/google-leaked-prefixes-knocked-japan-off-internet/ (visited on 09/03/2023).

[40] A. Haeberlen et al. "Netreview: Detecting When Interdomain Routing Goes Wrong". In: *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2009*. 2009, pp. 437–452. ISBN: 978-1-931971-67-6.

[41] David Hares and Susan Hares. *BGP-4 MIB Implementation Survey*. Request for Comments RFC 4275. Internet Engineering Task Force, Jan. 2006. DOI: 10.17487/RFC4275. URL: https://datatracker.ietf.org/doc/rfc4275 (visited on 09/05/2023).

[42] Yihua He et al. "Lord of the Links: A Framework for Discovering Missing Links in the Internet Topology". In: *IEEE/ACM Transactions on Networking* 17.2 (Apr. 2009), pp. 391–404. ISSN: 1558-2566. DOI: 10.1109/TNET.2008.926512.

[43] Andy Heffernan. *Protection of BGP Sessions via the TCP MD5 Signature Option*. Request for Comments RFC 2385. Internet Engineering Task Force, Aug. 1998. DOI: 10.17487/RFC2385. URL: https://datatracker.ietf.org/doc/rfc2385 (visited on 09/04/2023).

[44] *Hurricane Electric BGP Toolkit*. URL: https://bgp.he.net/ (visited on 12/11/2023).

[45] G. Huston. "Interconnection, Peering and Settlements". In: 2003. URL: https://www.semanticscholar.org/paper/Interconnection%2C-Peering-and-Settlements-Huston/44906c7a4f6cd8f168e23cb93354cdc09a43c96e (visited on 08/07/2023).

[46] *Index of /Datasets/Peeringdb*. URL: https://publicdata.caida.org/datasets/peeringdb/ (visited on 11/09/2023).

[47] *IRR Overview*. URL: https://www.irr.net/docs/overview.html (visited on 02/23/2023).

[48] Yuchen Jin. *ProbLink*. Feb. 2023. URL: https://github.com/YuchenJin/ProbLink (visited on 07/21/2023).

[49] Yuchen Jin et al. "Stable and Practical {AS} Relationship Inference with {ProbLink}". In: *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 2019, pp. 581–598. ISBN: 978-1-931971-49-2. URL: https://www.usenix.org/conference/nsdi19/presentation/jin (visited on 07/21/2023).

[50] Zitong Jin et al. "TopoScope: Recover AS Relationships From Fragmentary Observations". In: *Proceedings of the ACM Internet Measurement Conference*. IMC '20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 266–280. ISBN: 978-1-4503-8138-3. DOI: 10.1145/3419394.3423627. URL: https://dl.acm.org/doi/10.1145/3419394.3423627 (visited on 07/19/2023).

[51] Maria Konte, Roberto Perdisci, and Nick Feamster. "ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes". In: *ACM SIGCOMM Computer Communication Review* 45.4 (Sept. 2015), pp. 625–638. ISSN: 0146-4833. DOI: 10.1145/2829988.2787494. URL: https://dl.acm.org/doi/10.1145/2829988.2787494 (visited on 08/05/2023).

[52] Matt Lepinski and Kotikalapudi Sriram. *BGPsec Protocol Specification*. RFC 8205. Sept. 2017. DOI: 10.17487/RFC8205. URL: https://www.rfc-editor.org/info/rfc8205.

[53] Matt Lepinski and Sean Turner. *An Overview of BGPsec*. Internet Draft draft-ietf-sidr-bgpsec-overview-08. Internet Engineering Task Force, June 2016. URL: https://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-overview-08 (visited on 09/03/2023).

[54] Qi Li et al. "Invalidating Idealized BGP Security Proposals and Countermeasures". In: *IEEE Transactions on Dependable and Secure Computing* 12.3 (2014), pp. 298–311.

[55] Aemen Hassaan Lodhi. "The Economics of Internet Peering Interconnections". In: (Nov. 2014). URL: http://hdl.handle.net/1853/53092 (visited on 08/06/2023).

[56] *Looking Glass - Hurricane Electric (AS6939)*. URL: https://lg.he.net/ (visited on 11/08/2023).

[57] *Looking GlassCogent*. URL: https://www.cogentco.com/en/looking-glass (visited on 11/08/2023).

[58] Flavio Luciani, Antonio Prado, and Tiziano Tofoni. *BGP, Dalla Teoria Alla Pratica*. seconda edizione. June 2022. ISBN: 978-88-905806-9-7.

[59] Matthew Luckie et al. "AS Relationships, Customer Cones, and Validation". In: *Proceedings of the 2013 Conference on Internet Measurement Conference*. IMC '13. New York, NY, USA: Association for Computing Machinery, Oct. 2013, pp. 243–256. ISBN: 978-1-4503-1953-9. DOI: 10.1145/2504730.2504735. URL: https://dl.acm.org/doi/10.1145/2504730.2504735 (visited on 07/21/2023).

[60] *Lumen's Looking Glass*. URL: https://lookingglass.centurylink.com/ (visited on 11/08/2023).

[61] *MANRS*. URL: https://www.manrs.org/ (visited on 09/03/2023).

[62] *Mapping Autonomous Systems to Organizations: CAIDA's Inference Methodology - CAIDA*. URL: https://www.caida.org/archive/as2org/ (visited on 11/09/2023).

[63] Asya Mitseva, Andriy Panchenko, and Thomas Engel. "The State of Affairs in BGP Security: A Survey of Attacks and Defenses". In: *Computer Communications* 124 (June 2018), pp. 45–60. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2018.04.013. URL: https://www.sciencedirect.com/science/article/pii/S014036641731068X (visited on 08/06/2023).

[64] *MIX – Milan Internet eXchange – Italy's Leading Interconnection Platform.* URL: https://mix-it.net/en/ (visited on 08/06/2023).

[65] Alessio Mobilia. *AS-ToR-Inference*. Nov. 2023. URL: https://github.com/AlessioMobilia/AS-ToR-Inference (visited on 11/16/2023).

[66] T.K. Moon. "The Expectation-Maximization Algorithm". In: *IEEE Signal Processing Magazine* 13.6 (Nov. 1996), pp. 47–60. ISSN: 1558-0792. DOI: 10.1109/79.543975.

[67] *Namex*. June 2023. URL: https://www.namex.it/ (visited on 08/06/2023).

[68] Marcin Nawrocki et al. "Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs". In: *Proceedings of the Internet Measurement Conference*. IMC '19. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 435–448. ISBN: 978-1-4503-6948-0. DOI: 10.1145/3355369.3355593. URL: https://dl.acm.org/doi/10.1145/3355369.3355593 (visited on 09/03/2023).

[69] *Neo4j Graph Database & Analytics – The Leader in Graph Databases*. URL: https://neo4j.com/ (visited on 11/14/2023).

[70] *NeoDash - Dashboard Builder for Neo4j - Neo4j Labs*. URL: https://neo4j.com/labs/neodash/ (visited on 11/14/2023).

[71] Publication date: 17 Mar 2008- NEWS, RIS, and Internet Governance. *YouTube Hijacking: A RIPE NCC RIS Case Study*. URL: https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study (visited on 09/03/2023).

[72] Rishab Nithyanand et al. *Measuring and Mitigating AS-level Adversaries against Tor*. Dec. 2015. DOI: 10.48550/arXiv.1505.05173. arXiv: 1505.05173 [cs]. URL: http://arxiv.org/abs/1505.05173 (visited on 08/06/2023).

[73] Ola Nordström and Constantinos Dovrolis. "Beware of BGP Attacks". In: *ACM SIGCOMM Computer Communication Review* 34.2 (Apr. 2004), pp. 1–8. ISSN: 0146-4833. DOI: 10.1145/997150.997152. URL: https://doi.org/10.1145/997150.997152 (visited on 09/02/2023).

[74] *NVD - Cve-2022-40302*. URL: https://nvd.nist.gov/vuln/detail/cve-2022-40302 (visited on 09/02/2023).

[75] *NVD - Cve-2022-40318*. URL: https://nvd.nist.gov/vuln/detail/cve-2022-40318 (visited on 09/02/2023).

[76] *NVD - Cve-2022-43681*. URL: https://nvd.nist.gov/vuln/detail/cve-2022-43681 (visited on 09/02/2023).

[77] Jan Oesterle, Holger Kinkelin, and Filip Rezabek. "Challenges with BGPSec". In: *Network (Bristol, England)* 5 (2021).

[78] *Operation of BGP > Introduction to BGP | Cisco Press*. URL: https://www.ciscopress.com/articles/article.asp?p=2738462&seqNum=2 (visited on 12/26/2023).

[79] *PeeringDB*. URL: https://www.peeringdb.com/ (visited on 11/08/2023).

[80] *RIPE Atlas - RIPE Network Coordination Centre*. URL: https://atlas.ripe.net/ (visited on 11/08/2023).

[81] *RIPEstat UI*. URL: https://stat.ripe.net/app/ (visited on 11/08/2023).

[82] *RIS Live*. URL: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-live (visited on 11/08/2023).

[83] *RouteViews – University of Oregon RouteViews Project*. Oct. 2023. URL: https://www.routeviews.org/routeviews/ (visited on 11/08/2023).

[84] *RPKI - The Required Cryptographic Upgrade to BGP Routing*. Sept. 2018. URL: http://blog.cloudflare.com/rpki/ (visited on 09/03/2023).

[85] *RPKI Portal*. URL: https://rpki.cloudflare.com/?view=statistics (visited on 09/05/2023).

[86] Pavlos Sermpezis et al. *ARTEMIS: Neutralizing BGP Hijacking within a Minute*. June 2018. DOI: 10.48550/arXiv.1801.01085. arXiv: 1801.01085 [cs]. URL: http://arxiv.org/abs/1801.01085 (visited on 09/03/2023).

[87] Tal Shapira and Yuval Shavitt. "BGP2Vec: Unveiling the Latent Characteristics of Autonomous Systems". In: *IEEE Transactions on Network and Service Management* 19.4 (Dec. 2022), pp. 4516–4530. ISSN: 1932-4537. DOI: 10.1109/TNSM.2022.3169638.

[88] Tal Shapira and Yuval Shavitt. "Unveiling the Type of Relationship Between Autonomous Systems Using Deep Learning". In: *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. Apr. 2020, pp. 1–6. DOI: 10.1109/NOMS47738.2020.9110358.

[89] Yuval Shavitt and Udi Weinsberg. "Quantifying the Importance of Vantage Point Distribution in Internet Topology Mapping (Extended Version)". In: *IEEE Journal on Selected Areas in Communications* 29.9 (2011), pp. 1837–1847. DOI: 10.1109/JSAC.2011.111008.

[90] Aftab Siddiqui. *Not Just Another BGP Hijack*. Apr. 2020. URL: https://www.manrs.org/2020/04/not-just-another-bgp-hijack/ (visited on 09/03/2023).

[91] Aftab Siddiqui. *Unpacking the First Route Leak Prevented by ASPA*. Feb. 2023. URL: https://www.manrs.org/2023/02/unpacking-the-first-route-leak-prevented-by-aspa/ (visited on 10/24/2023).

[92] Yang Song, Arun Venkataramani, and Lixin Gao. "Identifying and Addressing Reachability and Policy Attacks in "Secure" BGP". In: *IEEE/ACM Transactions on Networking* 24.5 (Oct. 2016), pp. 2969–2982. ISSN: 1558-2566. DOI: 10.1109/TNET.2015.2503642.

[93] Kotikalapudi Sriram and Alexander Azimov. *Methods for Detection and Mitigation of BGP Route Leaks*. Internet-Draft draft-ietf-grow-route-leak-detection-mitigation-09. Internet Engineering Task Force / Internet Engineering Task Force, July 2023. 10 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-grow-route-leak-detection-mitigation/09/.

[94] L. Subramanian et al. "Characterizing the Internet Hierarchy from Multiple Vantage Points". In: *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. June 2002, 618–627 vol.2. DOI: 10.1109/INFCOM.2002.1019307.

[95] talshapira. *BGP2VEC*. July 2023. URL: https://github.com/talshapira/BGP2Vec (visited on 07/21/2023).

[96] *Tata Communications AS6453 IPv4 and IPv6 Looking Glass*. URL: https://lg.as6453.net/bin/lg.cgi (visited on 11/08/2023).

[97] *Telia Looking Glass*. URL: https://lg.telia.net/ (visited on 11/08/2023).

[98] *The Internet Is Getting Safer: Fall 2020 RPKI Update*. Nov. 2020. URL: http://blog.cloudflare.com/rpki-2020-fall-update/ (visited on 09/05/2023).

[99] *The Two-Napkin Protocol*. Mar. 2015. URL: https://computerhistory.org/blog/the-two-napkin-protocol/ (visited on 12/21/2023).

[100] Dr. Joseph D. Touch. *Defending TCP against Spoofing Attacks*. RFC 4953. July 2007. DOI: 10.17487/RFC4953. URL: https://www.rfc-editor.org/info/rfc4953.

[101] *UNARI | Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. URL: https://dl.acm.org/doi/10.1145/3359989.3365420 (visited on 07/19/2023).

[102] Kevin Wallace and Wendell Odom. *CCNP Routing and Switching Route 300-101 Official Cert Guide*. 1° edizione. Indianapolis, Ind: Cisco Systems, 2014. ISBN: 978-1-58720-559-0.

[103] Paul Watson. "Slipping in the Window: TCP Reset Attacks". In: *Presentation at* (2004).

[104] W. Willinger and M. Roughan. "Internet Topology Research Redux". In: 2013. URL: https://api.semanticscholar.org/CorpusID:12685125 (visited on 08/05/2023).

[105] Tianhao Wu et al. "RouteInfer: Inferring Interdomain Paths by Capturing ISP Routing Behavior Diversity and Generality". In: *Passive and Active Measurement*. Ed. by Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 216–244. ISBN: 978-3-030-98785-5. DOI: 10.1007/978-3-030-98785-5_10.

[106] Jianhong Xia and Lixin Gao. "On the Evaluation of AS Relationship Inferences [Internet Reachability/Traffic Flow Applications]". In: *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.* Vol. 3. Nov. 2004, 1373–1377 Vol.3. DOI: 10.1109/GLOCOM.2004.1378209.

[107] Ying Zhang, Zhuoqing Morley Mao, and Jia Wang. "Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing." In: *NDSS*. Citeseer, 2007.

[108] Zitong-Jin. *Toposcope*. Feb. 2023. URL: `https://github.com/Zitong-Jin/TopoScope` (visited on 07/21/2023).