IK2200

COMMUNICATION SYSTEMS DESIGN

PROJECT IXP

Identifying Network Disruptions Affecting Colocation Facilities

Enric CARRERA I AGUIAR
<enriccia@kth.se>
Lingfeng CHENG
<lincheng@kth.se>
Mandar JOSHI
<mandarj@kth.se>
Anika BINTEY MANSUR
<abmansur@kth.se>

Shubham BHARGAVA
<shbh@kth.se>
Seba ANNA VARGHESE
<vargh@kth.se>
Sri YULIANTI
<yulianti@kth.se>

Supervisor:
Alexandros MILOLIDAKIS
<miloli@kth.se>

Abstract

Internet Exchange Points (IXP) and Colocation facilities (Colos) have a symbiotic relationship keeping the Internet glued together through a mutual approach to peering, exchanging large amounts of daily data traffic. The Colos hold IXPs, providing services such as shared bandwidth, power and disaster resistant facilities. These facilities, although having large benefits and features, are still not one hundred percent safe from certain circumstances and a problem in any one of the buildings can have significant impact on a large number of networks. In this project, the peering interconnections established over IXP links is monitored using the constrained facility search methodology by analyzing traceroute data obtained from RIPE Atlas measurement platform and peering infrastructure data from PeeringDB and CAIDA. This project will also present a lightweight anomaly detection tool that monitors the signals extracted from the traceroutes to identify any unexpected deviations.

Keywords: Internet Exchange Points, Interconnections, Colocation Facilities, Anomaly Detection

Contents

List of Figures

List of Tables

Acronyms

ASes Autonomous Systems. 1, 4, 5, 38

ASN Autonomous System Network. 3, 34, 38

BGP Border Gateway Protocol. 1, 5

CAIDA Center for Applied Internet Data Analysis. 2, 3, 8, 11, 31, 57

CDN Content Delivery Network. 1

CFS Constrained Facility Search. 33–35, 40

Colos Colocation facilities. i, 1–4, 6, 57

DDoS Distributed Denial-of-Service. 1, 6

ELAN Ethernet Virtual Private LAN. 5

EVPL Ethernet Virtual Private Line. 5

ISP Internet Service Provider. 1, 5

IXP Internet Exchange Points. i, 1–5, 10, 11, 15–17, 34, 38, 40, 54, 62

PoP Point-of-Presence. 5

RTT Round-trip Time. 6, 27–29, 34, 36, 37, 49, 52

TTL Time to Live. 6

1 Introduction

IXPs are now an essential component of the Internet's Autonomous Systems (ASes) level substrate. The rapid growth of various networks that interconnect and exchange data in a reliable, efficient, and cost-effective manner has led to a plethora of research aimed at designing a stronger underlying Internet infrastructure. They are identified as an integral element of this infrastructure and present opportunities for Internet Service Provider (ISP) to extract considerable economic and technical benefits enabling cost savings and creating new revenue opportunities.

There are two important peering infrastructures, namely, IXPs and Colos, that play an important role in facilitating the exchange of Internet traffic. IXPs consist mainly of Layer 2 network infrastructure that aids ISPs and Content Delivery Network (CDN)s to peer and exchange Internet traffic between their ASes. There are hundreds of IXPs in operation around the world and a physical space is needed to house the access points to their infrastructure, which can be provided by the Colos. Colos are buildings that provide a secure space for IXPs and other networks to install their equipment. These facilities provide power and heat protection services, as well as services aimed to improve the hosted equipment runtime. Such services offered can be a power generator, stable uninterrupted power and high bandwidth cables.

Through the equipment hosted in Colos, terabytes of data are exchanged every second, and although the equipment of IXPs in Colos is usually well preserved, problems can arise with the peering traffic between ASes [3]. Therefore, a misadventure in a single IXP, or one of its Colos where its equipment is hosted can cause severe outages over the exchanged traffic of numerous networks. One example could be the power outage that took place on April 9th 2018 at DE-CIX Frankfurt [5], a large Internet exchange with around 800 different peering networks had a ripple effect over large ASes connected directly or indirectly to this IXP. The aim of this project is to be able to monitor such IXP hosting equipment locations and detect potential outages or anomalies that could affect the user traffic exchanged over Colos.

1.1 Problem Definition

Colos although being very reliable, are not immune to failures and may experience outages or disruptions due to power and equipment failure, human errors, natural disasters, Distributed Denial-of-Service (DDoS) attacks or Border Gateway Protocol (BGP) hijacks. These failures can have significant impact on the exchange of daily traffic [3]. Through these public peering links, terabytes of data are exchanged every second. Small networks, that otherwise would not be able to exchange traffic through these links, are also able to directly exchange traffic. As such, IXP links have been characterized as core parts of the Internet infrastructure with Colos providing the ideal locations where the establishment of such peering relations can take place.

1.2 Motivation

The influence of IXPs is vital and plays an instrumental role in the development of local Internet ecosystem. The criticalness of such infrastructure provides a clear motivation to develop important techniques that detect network disruptions. This project aims to a) leverage large scale traceroute measurements that traverse through the potentially affected IXP infrastructure, b) build an anomaly detection tool that monitors the identified public peering links and, c) report cases where the extracted signal deviates from normal.

1.3 Goals

The Goal of this project is to build an anomaly detection tool that monitors the links between the Colos to detect network disruptions. To achieve that, we utilize the RIPE Atlas database [4] which consists of traceroute measurements made at well-known time-intervals from thousands of probing locations distributed around the world. From those traceroute measurements, our **first step** is to identify promising IP addresses of equipment that is likely hosted in Colos. We identify such promising equipment from traceroutes by observing an IXP address. Having identified such equipment, our **second step** is to figure out the Colos that host this equipment (if any). Upon succeeding, our **final step** is to monitor the peering links established over the IXP interface of that equipment with other IXP connected equipment hosted by a different network. The other side of the peering link can either be in the same Colos (intra-facility link) or a different Colos connected over the same IXP infrastructure (inter-facility link). For the purposes of this project, we monitored such links over a period of two months and report examples of the anomalies we discovered in this report document.

1.4 Research Methodology

This project is based on quantitative research methods as it depends on external data that can provide information to answer our research questions. To obtain relevant data, this project leverages various datasets for the traceroute measurements on the Internet wide scale. The simplest way to fetch these data-sets is from the RIPE Atlas [4] daily dumps, which maintains measurements from the last 30 days. Additional peering related information was then extracted from the peering database PeeringDB [6] and the Center for Applied Internet Data Analysis (CAIDA) database [7]. The time frame for collecting the traceroute data will be ideally between two months, as constrained with the duration of this project.

1.5 Benefits, Ethics and Sustainability

Benefits: IXPs can be regarded as the key component of today's Internet infrastructure that contribute to global network efficiency and resilience. They provide various benefits to the networks connected over them. One of the most important benefits can be in reducing latency, transit costs and provide a better user experience by keeping domestic Internet traffic local. This proves beneficial to the academic institutions, content providers, carrier and businesses strengthening the local Internet connectivity. Software based automation is becoming a key topic in context to the operations of IXPs. As more and more IXPs are interested in automating many of their daily functions, our monitoring tool can benefit the IXP operators and their customers in detecting anomalies and simultaneously reduce network disruptions due to power outages or equipment failures.
Ethics: There is a large amount of private data that traverses the IXP network every day. We have kept in mind the privacy of the end users in this project and only focused on publicly available data in the databases. Furthermore, the results of this project would result in identifying colocation facilities based on IP addresses, in other words, creating a mapping between IP addresses to colocation facilities. This is potentially sensitive information that is not publicly available, and could be used for malicious purposes such as cyber-attacks and other security purposes. Some colocation facility operators would potentially not want to be monitored for purposes such as this.
Sustainability: In general, there are no sustainability issues related to this project. The public peering of IXP supports the UN Sustainable Development Goal 12, "Responsible Consumption and Production - By 2030, ensure that people everywhere have the relevant information and awareness for sustainable development and lifestyles in harmony with nature" [8]. With the

sharing economy model referred to as peer-to-peer based sharing of services, underutilised networks can be shared to improve efficiency and sustainability. This helps in reducing the pollutants, emissions and carbon footprints resulting a positive environmental impact [9]. It also adheres with the UN Sustainable Development Goal 13, "Climate Action - Strengthen resilience and adaptive capacity to climate-related hazards and natural disasters in all countries" [10] to reduce the energy and heat produced in the Colos. One such example is the Interxion ST06 data center in collaboration with Stockholm Exergi, is utilizing the heat and energy produced from data-centers to heat up homes in Stockholm [11].

1.6 De-limitations

The project has been narrowed to three databases - RIPE Atlas [4], PeeringDB [6] and CAIDA [7]. As per the RIPE Atlas terminology, this project focuses on the data-set involving "Built-In" measurements as they are publicly available as compared to the "User-defined" measurements which are initiated by the user-community at the cost of credits and also available only for a short period of time. Although the CAIDA database has large scale and up-to-date IXP information (as it combines information from different databases such as PeeringDB, Hurricane Electric and Packet Clearing House) compared to PeeringDB, it missed relevant information such as a mapping between an IXP to Autonomous System Network (ASN), which was relevant to this study. One option to make use of the information available in CAIDA database is to merge this information with that of PeeringDB. However, this merge is not done as part of the project, and only the Routeviews IP Prefix to AS number database [12] from CAIDA will be utilised in this project. This project also focuses on building the anomaly detection tool primarily for monitoring public peering links, solely for IPs in the IPv4 version. Analysing private peering will not be part of this project, as more complex information is required using IP alias resolution [11]. Another delimitation with respect to the traceroutes is an assumption made in this project that routers reply to the traceroutes with the IP of the inbound interface, as this is the common behavior. For cases that deviate from this expected behavior could fail to identify the facility or lead to an incorrect inference of the facility.

1.7 Outline

The rest of the report is organised as follows. First, the necessary background and related work is defined in **Section 2**. Then, the proposed methodology and research process is described in **Section 3**, implementation methods in **Section 4**, detailed analysis in **Section 5** and finally the conclusions and limitations are reported in **Section 6**.

2 Background

In this section, the reader is introduced to the relevant background needed to follow this project. Firstly, key concepts involving IXP, Colos and traceroutes are defined in **Section 2.1, 2.2** and **2.3**, followed by the background of the databases described in **2.4**. Lastly, the related work involving researchers solving a similar problem is described in **2.5**.

2.1 Internet Exchange Point (IXP)

An IXP facilitates public peering connections established among its participant ASes. Its purpose is to provide the underlying infrastructure for participant networks to communicate with low latency, minimum packet loss and high throughput to their customers. This physical infrastructure commonly operates in Layer 2 enabling various connected networks to communicate with one another [3].

The IXP is mainly structured with one or more high-end switches called core switches, depicted as the two centered switches in Figure 1. The customers install access switches in the interconnection facilities. The peering IXP members have the same access switch or back-haul switch to route the traffic to local customers network which is referred as member ASes. The router of an IXP member can be located in different facilities of the same IXP [2]. This overview of how an IXP looks like can be seen in Figure 1.

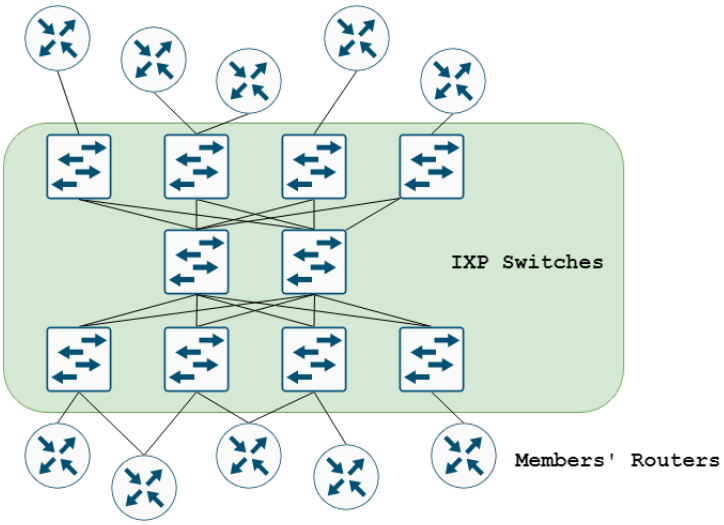


Figure 1: A Typical Internet Exchange Network. Adapted from [1]

.

The backbone infrastructure exchanges traffic with other networks through any of the available peering connections. There are three types of peering: public, private and remote, shown in Figure 2. These are explained in the following subsections.

2.1.1 Private Peering

In private peering, the connection is done in Layer 1 or Layer 2. Peers offer some dedicated capacity to each other to exchange a large volume of traffic, which otherwise will not fit in a shared connection to an IXP [2].

There are two types of private peering:

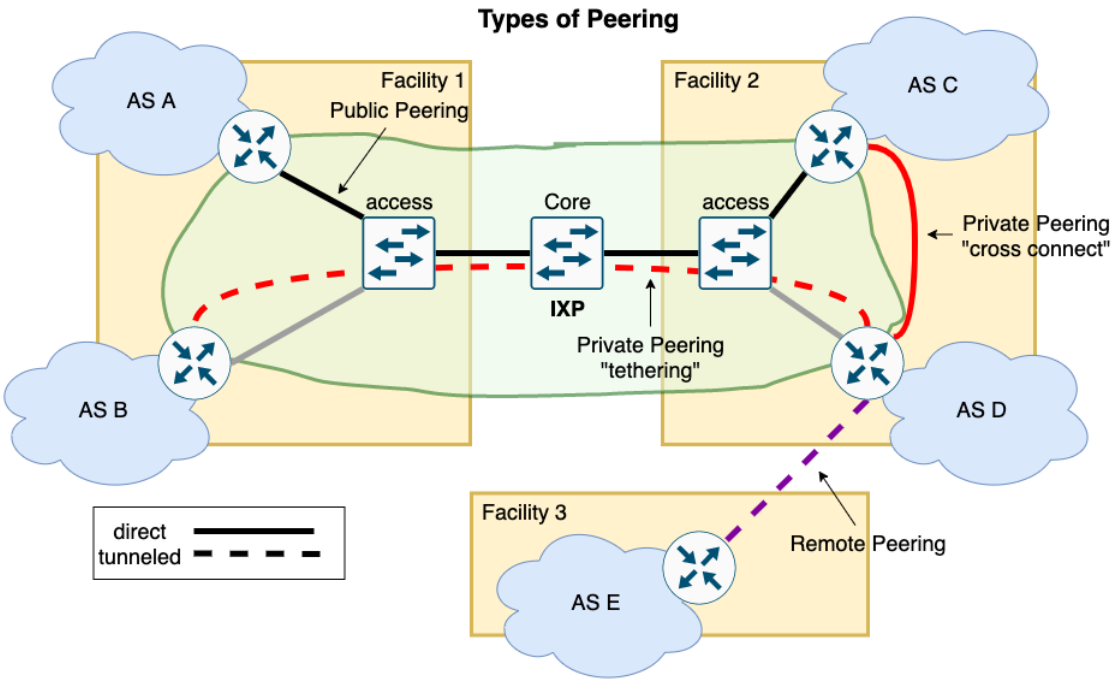


Figure 2: Types of Peering. Adapted from [2].

- **Cross Connect:** The connection is done with a physical hard wire cable which is directly connected within two networks, as depicted in Figure 2 with a solid red line connecting 'AS C' and 'AS D'. This cross-connect can be between networks in the same data center or be in different cities. The requirement solely depends on peers or providers of the network topology [2].
- **Tethering:** Some private peering connections can be established over the public switching fabric called Tethering or IXP metro VLAN, as depicted in Figure 2 with a dotted red line connecting 'AS B' and 'AS C'. With this connection, point to point virtual private line is established through VLAN [2]. Ethernet Virtual Private LAN (ELAN) and Ethernet Virtual Private Line (EVPL) are examples of such type of peering.

2.1.2 Public Peering

In public peering the network connections are done between two IXP members via an IXPs switch fabric. First, to utilise the IXP infrastructure, IXP members need to allocate an IP address from the address space of the IXP. Then, to communicate over the IXP fabric, a BGP connection must be established first between the communicating ASes. The benefit of the public peering is that by leasing one IXP port it is possible to exchange a large amount of traffic between a large fraction of the IXP members [2].

2.1.3 Remote Peering

Remote peerings are connected to the IXP via Ethernet-over-MPLS (Multiprotocol Label Switching), where Layer 2 Ethernet frames are sent transparently over MPLS. The router can be located anywhere in the world, this is depicted in Figure 2 with the purple dotted line. As such, networks utilizing this type of peering do not need to maintain their equipment inside the colocation facility. Instead, this is the responsibility of the one providing the remote peering service. IXPs, collaborate with third party networks that provide remote participants with connectivity to the IXP. Remote peering is done in the Point-of-Presence (PoP) of an ISP in the interconnection facility [2].

2.2 Colocation Facilities

Colos are secure buildings or rooms with certain services available, that allow network administrators to install their equipment. These facilities provide various services such as cooling equipment, fire protection, stable power, backup generators, high bandwidth cables, etc. The large IXPs install access switches in multiple colos in different cities in which they want to operate their network. Peering infrastructure links can be either leased from the colocation facility operator or manually operated by the participating networks [3].

The operator of a colocation facility can manage multiple facilities in large cities since it can consist of large communication hubs. Providing multiple facilities in the same city may allow networks participating in one facility to exchange their traffic to other networks in other facilities [2]. For example, in Figure 2 different networks are shown connected to different facilities communicating with each other, such as AS-A connected to AS-C.

2.2.1 Issues with Colocation Facilities

Though the equipment in Colos is well maintained, various problems can occur [3], for example:

- Power and equipment failure: Outages can happen due to the weather or fire breaks which leads to power and equipment failure.
- Human errors: These can happen during any maintenance window.
- Natural disasters: Natural disaster can lead to power failure and disturb the systems.
- DDoS attacks: DDoS attacks occur when multiple machines are used together to attack a single target, in which the targeted host/network is flooded with network traffic to the point where it is unable to respond [13].
- BGP hijacks: This is an application-layer DDoS attack which can break the peering infrastructure as it allows an attacker to use a network prefix as their own. [14].

2.3 Traceroutes

Traceroute is a network diagnostic tool which purpose is to trace the paths that ICMP data packets take from their source to a destination address. To do so, the Time to Live (TTL) header of the packet is modified to obtain certain information from the different nodes in the path to the destination. The TTL is the distance (in nodes, also called hops) that the packet can travel before it is dropped ($TTL = 0$) by the last reached node, which in such case will respond with a ICMP Error message that contains the IP address of the interface that received that packet as well as the time it took the packet to reach the node and get back to the source called Round-trip Time (RTT).

Starting from $TTL = 1$, for each node, the TTL is increased by one to control how far the packet can travel, to be able to get the information of each node in the path from the source to the destination. An example of this is shown in Figure 3. In the Figure, between the source and the destination there are three nodes to be mapped, meaning that four hops are needed. Listed above each ICMP message arrow in purple, there is the TTL used (going from 1 - for the first node, to 4 - for the destination). These measurements for each hop are repeated several times to minimize the likelihood of occurrence of traceroute errors involving the RTT (commonly repeated three times, but can be repeated more times).

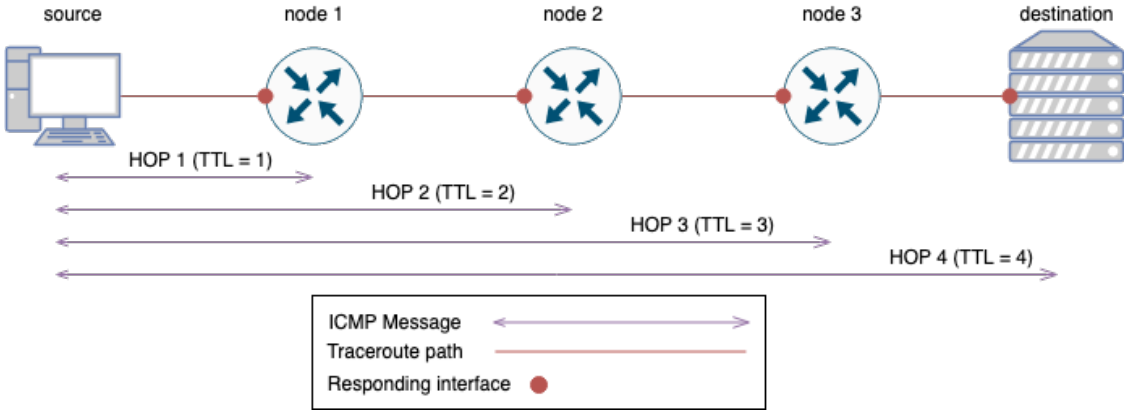


Figure 3: Traceroute Workflow

2.3.1 Paris Traceroutes

A version of traceroute called Paris traceroute is used in the context of this project because of its ability to obtain the real topology of a network. Paris traceroutes avoid common load balancing issues introduced by routers that load balance packets (see Figure 4 below), working around these load balancing nodes to deliver a reliable path that is necessary for the project implementation. To do so, they control the contents of the packet header, allowing for a more precise measurement of the routes taken by the packet [15]. It is important to note that Paris traceroutes do not solve all problems of load balancing nodes, they only partially mitigate the problem. Looking at Figure 4, the expected paths that a traceroute should deliver are:

- **{Source, Node1, Node2, Node3, Node5, Destination}**
- **{Source, Node1, Node4, Node5, Destination}**

However, if Node 1 starts applying load balancing, the following route could be the result of the traceroute (assuming "Round Robin" load balancing in this example). This is due to the functionality of traceroutes.

- If the first node after Node 1 is Node 4: **{Source, Node1, Node4, Node3, Destination}**

These type of results give false information about the topology of the network and affect the validity of traceroutes. However, Paris traceroutes mitigate this problem and provide reliable paths.

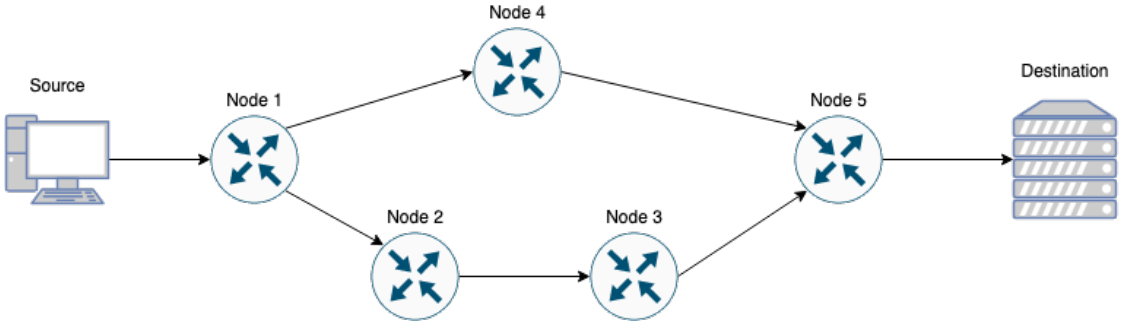


Figure 4: Load Balancing Explanation

2.4 Databases

In order to complete the goals set out by this project, several different databases need to be utilised in order to fetch peering information regarding colocation facilities and traceroutes. The databases are described as follows.

2.4.1 RIPE Atlas Platform:

Firstly, this project uses the RIPE Atlas [4] which is a data collection system from the RIPE NCC (Réseaux IP Européens Network Coordination Centre), a regional Internet registry. RIPE Atlas has around eleven thousand probes distributed around the world across 170 countries to collect data, and provides maps, data tools and visualisations based on the results from the probes. The locations of these are shown in Figure 24. These probes are maintained by RIPE Atlas "hosts", and any individual can apply to become a host, provided they are in a location which RIPE deems suitable, and the hosts simply have to connect the probe to their network. The probes are attached to Ethernet ports on routers via USB power and a network cable, they collect measurements and relay the data to the RIPE NCC. The probes can collect the following types of measurements: ping, traceroute, SSL/TLS, DNS, NTP and HTTP. The measurements that this project will be interested in are the traceroute measurements.

2.4.2 PeeringDB:

PeeringDB [6] was initially set up to facilitate peering between networks and peering coordinators, but recently has developed to keep up with the fast pace and diversity in which the internet is growing. This development has made this database not only for its original purpose, but now also includes all types of data about interconnections for networks, clouds, services and enterprises, as well as interconnection facilities that are developing at the edge of the Internet. Examples of datasets hosted by PeeringDB include a list of colocation facilities (datacenters), Internet exchanges, networks, network points of contact and organisations.

2.4.3 CAIDA:

CAIDA [7] conducts network research and has a large data infrastructure that collects and contrasts several types of data from other different databases such as PeeringDB, Hurricane Electric (HE) and Packet Clearing House (PCH) and makes it available for the community. For this project the Routeviews IP Prefix to AS number dataset [12] of CAIDA will be utilised. It provides a mapping between an IP subnet and the ASN it is attached to. More detail about how this dataset is used is provided in Section 3.2. There's also datasets with IXP information but are not used due to some conditions that will be explained in section 3.2.

2.5 Related Work

In the paper "Detecting Network Disruptions At Colocation Facilities" [3] the authors have succeeded in monitoring delay and routing patterns between colocation facilities which this paper is aiming to emulate. They have evaluated their methodology using RIPE Atlas traceroute data and managed to identify an IXP outage, a DDoS attack and a power failure inside a colocation facility. The authors also manage to identify anomalies and outages which were common on a city wide scale that affected up to eight colocation facilities. A similar study has also been performed by Giotsas et al. in the paper "Detecting Peering Infrastructure Outages in the Wild" [16]. By observing BGP community attributes that appended in BGP routing updates, the authors were

able to identify locations which experienced network outages. Their methodology also allowed for real time tracking of the reaction of the networks to those outages, and the authors discovered four times as many outages than those which were publicly reported under a period of five years.

The methodology that this project will use was proposed by authors Giotsas et al. in their paper "Mapping Peering Interconnections to a Facility" [2]. Their contribution is named *constrained facility search*, which extracts the physical interconnection location of a particular colocation facility out of a set of facilities. The authors used data from PeeringDB combined with data from IXP websites for associated interconnection facilities. Using traceroute measurements, the authors succeeded in identifying specific colocation facilities, and verified their data with operators of those locations privately. The methodology will be described in more detail in Section 3.4.

Moreover, related work by Augustin et al. [15] about a new type of traceroute named the *Paris traceroute* is mentioned and explained in Section 2.3. The authors created this type of traceroute due to problems caused by load balancing routers in networks when using traceroutes. Other related work with traceroutes includes using the RIPE Atlas traceroute measurements to monitor delay and forwarding anomalies [17] as well as improving the IP-to-AS mapping by analysing traceroute probes, reverse DNS lookups and BGP configurations [18].

Following the previous section on the comparison of databases, previous work by Klöti et al. [19] have looked into comparing public IXP datasets. The authors performed a cross comparison of three IXP datasets: PeeringDB, Euro-IX and PCH. They found that the databases differed in their approach of data collection, making the databases either AS-centric or IXP-centric. The authors identified that the union of the three databases included roughly 40% more IXPs and 66% more IXP participants compared to PeeringDB alone.

Related work has also been conducted on the nature of IXPs. Papers from Chatzis et al. [20] and Ager et al. [1] have investigated large IXPs and have observed that IXPs contain a rich peering fabric and the important role that they play in the ecosystem of the Internet.

3 Methodology

This section discusses the methodology used. The overview of the project is presented in Section 3.1, and data collection methods are discussed in 3.2. Then the more detailed methodology is presented in Sections 3.3, 3.4 and 3.5. Finally, Section 3.6 discusses data validation.

The aim of this project is to detect network disruptions at colocation facilities. To achieve this goal, there is a need to monitor the behaviour of traffic through colocation facilities under a period of time, and compare this data with the past observations regularly (in this case, every hour). In order to successfully map and monitor traffic to specific colocation facilities, data from several different datasets needs to be combined and used together (Section 3.3). First, the IXP is detected as part of the traceroute hop, then the colocation facility is identified (Section 3.4). Finally, network traffic is monitored for anomalies using deviations in delay and forwarding patterns (Section 3.5).

3.1 Methodology Overview

The first stage of the project begins with a literature study on the background of IXPs and colocation facilities. Alongside this literature study, a closer look into the different databases is taken as this is the source of the data used in this project. The second stage consists of implementing a set of scripts to filter the information provided by the databases and obtain the data which is useful for the purposes of this project. Based on the goals, the tasks of the project are well defined before beginning the implementation. These are listed below, also referred to as 'Phases' throughout the report.

- **Phase 1** focused on developing automated scripts that fetch and analyze traceroutes, as well as extracting relevant peering information from the related databases.
- **Phase 2** begins after all the relevant information is fetched from the traceroute measurements and relevant databases in Phase 1. The main focus of Phase 2 is to identify the near-end and far-end colocation facilities of the hops extracted from the traceroutes.
- **Phase 3** begins once the mapping between hops and facilities is obtained as part of Phase 2 results. Phase 3 aims to develop a tool to monitor these facility links. This tool is used to detect any network disruptions at colocation facilities.

The overview of the methodology is shown in Figure 5. In the overview, the different components that make up Phase 1 are shown to make up Phase 2, and Phase 3. The blue fields represent the different datasets that data is fetched from. The green fields represent scripts that are written to extract and parse this data. The yellow field is used to represent a network map that is created using the data. These fields are explained in more detail in Section 4.

The research process for this project involves working with a large number of datasets which makes the project follow a quantitative approach. The research method used is an experimental one, as from these datasets a large number of variables are extracted (such as different sets of colocation facilities). The methodology for this project then involved finding various relationships between these variables. It is important to note that the project uses data from real world internet measurements for testing purposes. Simulations are not conducted, and the results can be replicated using the same type of measurements and datasets in the future.

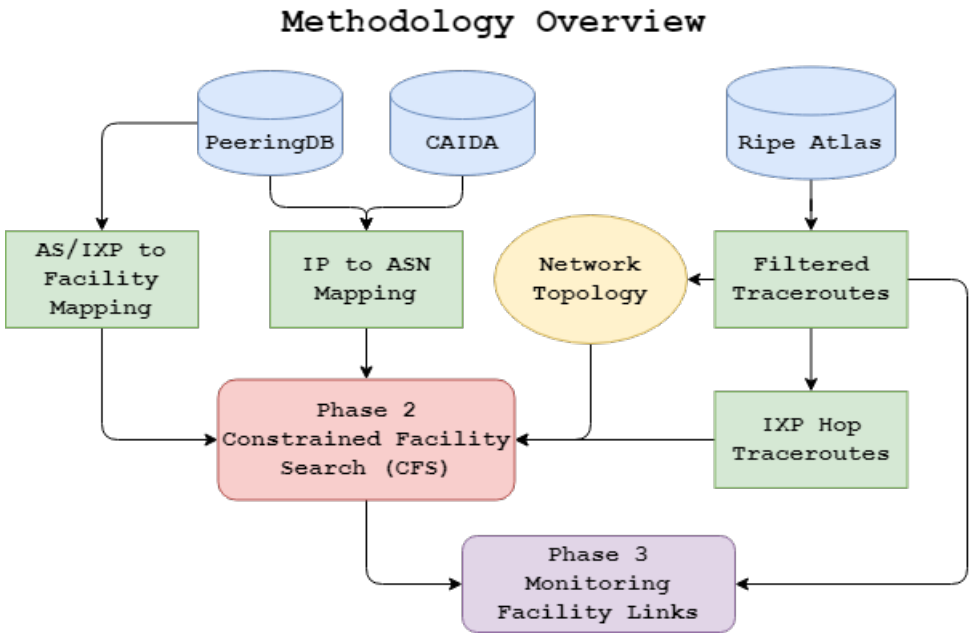


Figure 5: An Overview of the Phases

The hypothesis for this project is the following:

- By scanning millions of traceroutes stored in a large database, it will be possible to build an anomaly detection tool, thereby being able to detect network disruptions in real time as they are happening in different colocation facilities in the world. If strong disruptions are happening, it may be possible to confirm these by looking at local news sites at the location of the disruption.

In order to fulfill this hypothesis, a deductive research approach is used in which statistics are extracted from the datasets and analysed.

3.2 Data Collection

To retrieve the data needed for achieving the goal of monitoring facility links, data from three main sources needs to be pooled together: PeeringDB, RIPE Atlas and CAIDA. Data from these sources make it possible to carry out this goal as there is no singular dataset available which contains a direct mapping between colocation facilities and IP addresses.

RIPE Atlas is used for its built-in traceroute datasets, which are updated hourly. PeeringDB contains information regarding IXP related information, and lastly CAIDA is used for extra mappings which were not possible obtain with the datasets from PeeringDB.

There was an option to use CAIDA for IXP information alongside PeeringDB because its information is obtained from different databases and that gives it a larger scale and amount of information. However, some specific information needed was not provided by CAIDA, and that meant a merge between CAIDA and PeeringDB had to be done. This was outside the scope of this project, therefore only PeeringDB was used for IXP information.

3.2.1 IXP Information

All the information related to IXPs and colocation facilities is obtained from PeeringDB, as it is possible to query their API to access different datasets. In these various different datasets, useful

data can be extracted and stored using JSON formatting.

To obtain the information needed for the identification of colocation facilities, four different datasets were used: *ixlan*, *netixlan*, *ixfac* and *netfac*. As mentioned previously, this is because there is no singular mapping identifying colocation facilities by IP addresses. Due to this constrain, it is needed to use these four datasets to probabilistically identify the colocation facility. These datasets are described below, and the specific fields used in this project are shown.

ixlan: This dataset contains a list of values, each corresponding to a different IXP. In these datasets there is all types of information about an IXP, for example their ID, their name, the AS networks that are connected to it, and much more information as can be seen in Listing 1. From all the data that provides, the values of interest are the IXP ID and the prefixes, these are highlighted in Listing 1. The IXP ID is used to be able to identify each IXP and access its information in a fast way. The prefixes can be IPv4 or IPv6. The prefixes considered for this project are the IPv4 ones as only IPv4 traceroutes are analyzed.

```
{
    "id": 0,
    "ix_id": 0,
    "ix": "string",
    "name": "string",
    "descr": "string",
    "mtu": 0,
    "dot1q_support": true,
    "rs_asn": 0,
    "arp_sponge": "string",
    "net_set": [],
    "ixpfx_set": [
        {
        "id": 0,
        "protocol": "IPv4",
        "prefix": "string",
        "in_dfz": true,
        "created": "2019-08-24T14:15:22Z",
        "updated": "2019-08-24T14:15:22Z",
        "status": "string"
        }
    ],
    "ixf_ixp_member_list_url": "http://example.com",
    "ixf_ixp_member_list_url_visible": "Private",
    "created": "2019-08-24T14:15:22Z",
    "updated": "2019-08-24T14:15:22Z",
    "status": "string"
}
```

Listing 1: ixlan Response Example

netixlan: This dataset contains the information of an ASN inside a determined IXP, which means there can be multiple datasets with the same ASN but associated with different IXPs, an example of this datasets is shown in Listing 2. This is one of the reasons why it is challenging to identify a colocation facility. From this response the useful data for the project is the ASN number, used

to identify an ASN each time it appears. Furthermore, the IPv4 address linked to that ASN inside a certain IXP is also useful. Both of these values, highlighted in Listing 2, are used to create a mapping of IP address to ASN which is used later to find the ASN of IXP IPs.

```
{
    "id": 0,
    "net_id": 0,
    "net": "string",
    "ix_id": "string",
    "name": "string",
    "ixlan_id": "string",
    "ixlan": "string",
    "notes": "string",
    "speed": 0,
    "asn": 0,
    "ipaddr4": "string",
    "ipaddr6": "string",
    "is_rs_peer": true,
    "operational": true,
    "created": "2019-08-24T14:15:22Z",
    "updated": "2019-08-24T14:15:22Z",
    "status": "string"
}
```

Listing 2: netixlan Response Example

ixfac: This dataset provides a list of facilities inside an IXP. This way a IXP to Facility mapping is added to the datasets of each IXP that are created to be used in later steps of the project. In the response from PeeringDB every connection IXP to FAC is a different set of values shown in Listing 3, but once imported is appended to create a list of facilities which makes later steps easier.

```
{
    "id": 0,
    "ix_id": 0,
    "ix": "string",
    "fac_id": 0,
    "fac": "string",
    "created": "2019-08-24T14:15:22Z",
    "updated": "2019-08-24T14:15:22Z",
    "status": "string"
}
```

Listing 3: ixfac Response Example

netfac: This dataset contains the ASN and a facility it is connected to, enabling to create a ASN to Facility mapping which is used to infer the colocation facilities involved in Phase 2. An example is shown in Listing 4.

```
{
    "id": 0,
    "name": "string",
    "city": "string",
```

```
        "country": "string",
        "net_id": 0,
        "net": "string",
        "fac_id": 0,
        "fac": "string",
        "local_asn": 0,
        "created": "2019-08-24T14:15:22Z",
        "updated": "2019-08-24T14:15:22Z",
        "status": "string"
    }
```

Listing 4: netfac Response Example

3.2.2 Traceroute Information

Traceroutes are one of the most important pieces of information for the project as they are what allows for tracking and monitoring traffic and the paths that different packets take through the internet. For this project the database used to obtain those datasets is RIPE Atlas, as mentioned in Section 2.4. This database contains built-in traceroute measurements which are performed every 30 minutes from all RIPE Atlas probes towards DNS root servers. These traceroutes are built-in to the probes directly and executed automatically, but can also be user-defined. The data files are obtained through a daily dump that is updated every hour and has a backup of files for up to one month in the past.

From the RIPE Atlas database, the measurement ID, probe ID and the timestamp of the traceroute is also saved. The measurement ID and probe ID is saved because of how RIPE Atlas conducts user-defined and built-in measurements. RIPE Atlas measurements with a measurement ID lower than 1,000,000 are built-in measurements. More specifically, a traceroute with measurement ID 5XXX is an IPv4 traceroute [4]. The timestamp can be used to confirm when exactly the traceroute was performed.

3.2.3 Non-IXP IP to ASN mapping

This is a database which provides a mapping between an IP subnet and its ASN. This mapping is important as it is used to identify to which ASN a certain non-IXP IP is attached to. From there a list of colocation facilities can be extracted.

It is not possible to obtain this mapping from PeeringDB as the IPs of an ASN are not in any dataset. Therefore, the dataset used here is the "Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4" [12], which is updated daily. This dataset contains 3 values: the IP prefix, the length of the prefix and the ASN number linked to that prefix, an example of this is shown in Listing 5. Even though it is updated daily, most of the values are overwritten because there is not many changes. For the purposes of this project, the file from the 5th of October was chosen because it was the latest update in the beginning of the implementation.

```
    {
        "1.0.240.0/21": "23969",
        "1.0.248.0/21": "23969",
        "1.1.1.0/24": "13335",
        "1.1.8.0/24": "4134",
        "1.1.64.0/19": "2519",
```

```
        "1.1.96.0/24": "2519",
    }
```

Listing 5: Non-IXP IP to ASN Mapping

3.3 Phase 1: Fetching Peering Infrastructure Data

The goal of this Phase is to gather all the data needed for this project from the databases, and filter traceroutes in such a way that only traceroutes traversing an IXP are recorded. The reason the IXP is recorded is because colocation facilities are where IXPs interconnect, and as a result of that, the colocation facilities involved can be identified in later stages of the methodology.

First, the built-in traceroutes from the RIPE Atlas database are filtered by their IP version and Paris id. Hops that include an invalid value (denoted with a *) are also removed. Once these traceroutes are filtered, data extracted from the PeeringDB API is queried to check for traceroutes which pass through an IXP. If an IXP hop is found in a traceroute, the previous hop and the next hop is recorded along with their average RTT values. The previous hop is used to identify the access router where the network is connected to access the IXP, and the next hop is used for the constrained facility search in Phase 2. The RTT value is used for the link monitoring in Phase 3.

An example can be seen in Figure 6, in which the traceroute passes through an IXP in which IP A, the IXP IP and IP B are denoted. Referring back to Figure 2, it can be concluded that the traceroute traversed a public peering link since an IP belonging to an IXP appeared in the path.

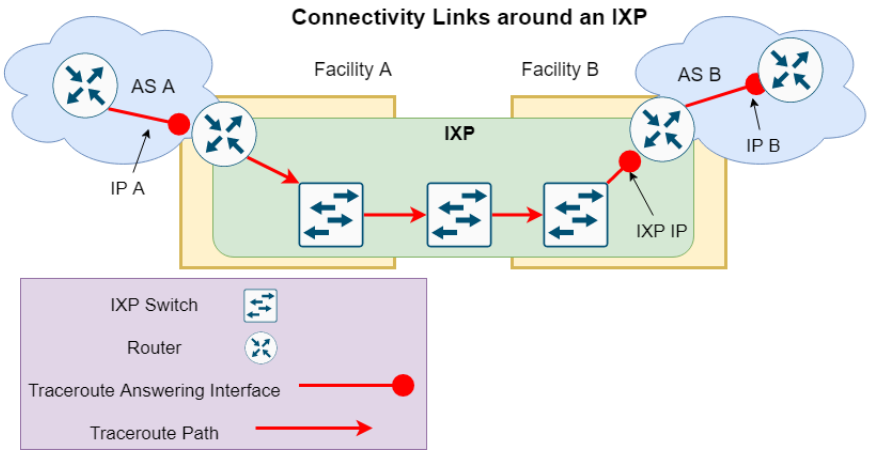


Figure 6: Traceroute Links. Adapted from [3].

An example of the result of this phase is given in Listing 6. In this listing, the previous hop, IXP hop and the next hop IP addresses can be observed. These fields are accompanied with the PeeringDB IXP id and the RTT values of the previous and the IXP hop, as well as the RIPE Atlas measurement ID, probe ID and timestamp can be seen.

```
    {
        {"previous_hop":"89.216.5.95","ixp_hop":"195.66.224.238"
           ,"next_hop":"5.56.17.13","ixp_id":"18","rtts":[18.0596
           7,42.07367],"msm_id":16423082,"prb_id":22467,"
           timestamp":1603756908}
    }
```

Listing 6: Example Result of Phase 1

3.4 Phase 2: Constrained Facility Search

The goal of this phase is to identify the colocation facilities involved in the traceroutes. These facilities are termed as near-end facility and far-end facility, according to the order a facility appears in the traceroute probes. The near-end facility is the one that appears before the traceroute traverses an IXP while the far-end is the one that appears after the traceroute has traversed the IXP. In Figure 6, the near-end facility is Facility A and the far-end facility is Facility B.

Based on the results obtained from Phase 1, the Constrained Facility Search (CFS) [2] is used with some modifications. The approach used to constrain the facility is the Rule-based Constrained Facility Search introduced in [3]. The first rule in [3] refers to 'User Yielded Information' is not used in this methodology, and the other two rules referring to 'Facilities of the alias ASNs' and 'Facilities of next hop (IXP)' are modified for this project. These modifications are described in this section, corresponding to Step 3 and Step 4. The 'User Yielded Information' consists of identifying additional IP interfaces that are already known as belonging to the colocation facility router and the remaining two rules consist of alias recognition that are outside the scope of this project. This limitation is described in Section 1.6.

The CFS uses several sets of facilities to get their intersections and ends up with a successful convergence, meaning that only one facility is obtained from the intersection. For identifying the near-end facility, we use the previous hop and the IXP hop. In Figure 6, the previous hop and the IXP hop are IP A and IXP IP respectively. Using previous hop to identify the near-end facility is important because it provides the IP used by the ASN in the "outer interface" of the access router to the IXP. Hence, every time that IP appears, it can be concluded that the traceroute traverses the same facility where that IXP is associated to. Similar to the near-end facility, for identifying the far-end facility, initially only the IXP hop is used, this decision comes from the fact that the next hop cannot be guaranteed to be associated with the same facility where the IXP is located in. Therefore, similar to the near-end facility, the ASN information to start the CFS is obtained from the IXP hop for the far-end facility, using a useful mapping between an AS number and an IXP IP obtained from PeeringDB. The other reason not to use the next hop to identify the far-end facility is because the IXP information is more reliable than translating the next hop IP to an AS number as the PeeringDB entries are updated frequently by IXPs in order to attract new customers [16, 20].

Based on the above approach, to get the neighbour IPs in further steps of the methodology, it is decided to use the previous hop and the IXP hop. This decision ensures that both hops exist in the facility which will be inferred.

Identification of near-end colocation facility using the above methodology is carried out in four different steps and each of these steps along with examples is described below. Input to each step is the colocation facilities from the previous step. Each step upon receiving this input tries to constrain the facilities and forwards them to the next step. Once a step is successful in identifying a single facility, then it does not forward again. There can be also cases where this method fails to identify a single facility.

- **Step 1** IXP information is queried to fetch the facilities associated with that IXP. If the IXP has installed its access switches only in one facility, then the output of this step is a single facility. This means that the near-end colocation facility has been successfully identified. If the output is not a single facility, then Step 2 is performed. Multiple facilities can be obtained if the IXP in question is a large IXP and therefore has a presence in multiple facilities in a city or even a country.

As an example, let the facility set extracted from IXP hop $F_{(IXP)}$ has three facilities with ids 1, 3 and 7, such that,

$$F_{(IXP)} = \{1,3,7\}$$

Since the facility set has multiple facilities, this information is forwarded to the next step.

- **Step 2** The input to this step is the colocation facilities from Step 1. In this step, the facility set of the AS associated with the previous hop is compared with input facility set and the common facilities among these two are taken. If there is only a single facility in common, then this is considered as the near-end facility. If there are more than one common facility, then Step 3 is performed.

  For example, let the facility set extracted from previous hop IP A denoted by $F_{(IPA)}$ has three facilities with ids 1, 2 and 3, such that,

  $$F_{(IPA)} = \{1,2,3\}$$

  Common facilities between $F_{(IXP)}$ and $F_{(IPA)}$ is given by $F_{(Z)}$,

  $$F_{(Z)} = F_{(IXP)} \cap F_{(IPA)} = \{1,3,7\} \cap \{1,2,3\} = \{1,3\}$$

  In this example, the output facility still in this step has more than a single facility. Hence in this case, Step 3 is performed.

- **Step 3** In this step, output of Step 2 is compared with the facility sets of all IXPs which appear as a neighbour of previous hop IP A.

  As an example, let IXP' and IXP" be the two facilities that have previous hop as IP A. This information is found by looking at other traceroutes.

  $$F_{(IXP')} = \{1,2,3,7,9\} \rightarrow F_{(IXP')} \cap F_Z = \{1,3\}$$

  $$F_{(IXP'')} = \{3,7,9\} \rightarrow F_{(IXP'')} \cap F_Z = \{3\}$$

  Here, $F_{(IXP')}$ and $F_{(IXP'')}$ are the facility sets extracted from IXP' and IXP" respectively. There are two common facilities between facility set of IXP' and facility set from Step 2, hence this cannot be considered. But there is only a single facility with facility id 3 common to facility set of IXP" and facility set from Step 2. Thus, facilty 3 is identified as the near-end facility.

- **Step 4** This step is followed as a last step to constrain facilities, if it is not possible to identify a single facility from Step 3. Here, the output from Step 3 is compared independently with the facility sets of all of the non-IXP neighbours of IP A. Among the independent intersections, the most recurring facility that accounts for at least 75% of the those intersections is considered as the constrained facility.

For the far-end colocation facility identification, a very similar methodology is applied. Only Step 2 has a difference in this case, here instead of using the AS of the previous hop, the AS associated with the IXP IP is used.

The Threshold Value: To find a suitable threshold value for Step 4 in the CFS, an experiment is carried out using different values, starting from 45%, up to 100%. Figure 7 illustrates the relation between the threshold value and the number of traceroutes with successfully inferred facilities in

step 4. The horizontal axis corresponds to the threshold values ranging from 45% to 100% and the vertical axis corresponds to the total interfaces. Two curves that represent the near-end CFS and the far-end CFS are plotted with blue and orange colors respectively.

The results in Figure 7 show that for the near-end CFS, the number of successful convergences is reduced gradually at a regular pace as the threshold is increasing. Meanwhile for the far end CFS, the trend seems to be stable from the starting point to around 72-73%. Looking at the results, by fixing the threshold at 45%, more interfaces or IPs mapped to the near-end facilities can be inferred before it drops gradually, but to have the most reliable results, the higher the threshold the better because it means that the requirements are more strict. Meanwhile for the far-end CFS, by choosing the threshold value at 75% yields a satisfactory number of IP to facility mappings, because before that there's almost no difference in the results. Overall, the threshold chosen for both near-end and far-end should not significantly compromise the reliability of the results, and for this the 75% value is used , as it means that most of the far-end mappings are kept while still maintaining a good amount of near-end mappings.

In the figure, there is a significant difference between the near-end and the far-end in terms of total succesful inferences. For the near-end, the number is much bigger due to the fact that in the last step of CFS, which is the one that's being analyzed here, the neighbours of the previous hop that are not IXPs, can be connected via cross-connect. This means those neighbours will probably be in the same facility or close to each other, so that the facility is more provable to appear in the neighbours facility sets. Meanwhile, in the far-end CFS, the location of the neighbours of the IXP hop cannot be assumed as the next hop after the access router nor in the facility involved. For example, the remote peering in Figure 2 that shows that the neighbour of the IXP hop would be in facility 3 instead of facility 2. Therefore, it is more difficult to have a successful convergence with the facility sets of the neighbours in the far-end case.

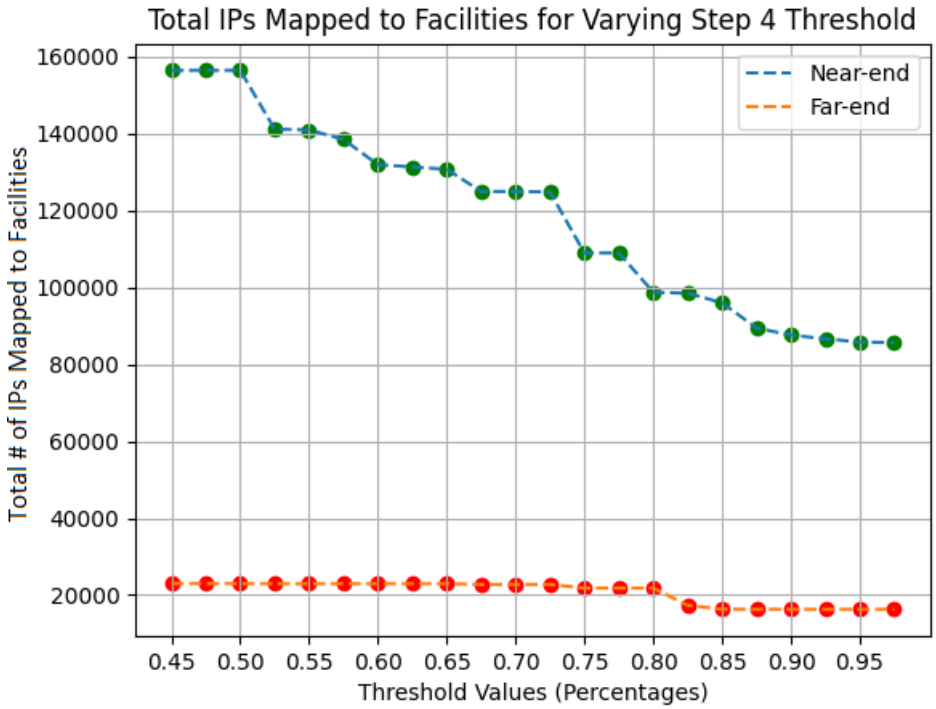


Figure 7: Varying Threshold Values

3.5 Phase 3: Facility Link Monitoring for Network Disruptions

As mentioned in the previous phases, in Phase 1 and 2 the objective is to fetch and analyze traceroutes, and as a result, a mapping of IP Addresses to colocation facilities is obtained. In this stage of the project, the goal is to monitor disruptions occurring in the links between colocation facilities. Using the mapping of IP addresses to colocation facilities, two measurements are observed every hour to decide whether the link is working properly or is experiencing problems. These two measurements are the number of traceroute packets that traverse the link between the near-end and far-end colocation facility, and the RTT delay, the latter being the time that the traceroute packet takes to traverse the link.

The methodology is divided in the two separate measurements, in both cases, a reference or expected behaviour is computed using previous data to the date that is being observed, and then compared to check if the values lie inside a proper behaviour or not. The expected pattern is very stable, and when anything happens usually large peaks or measurement deviations are expected. The following Subsections 3.5.1 and 3.5.2 explain each of these methodologies in detail.

3.5.1 Forwarding Model

Beginning with the forwarding model, to further understand how it works, first an example of the monitored measurements will be described. Figure 8 shows all the links between facilities observed in one hour of data. These links, are traversed by multiple traceroutes during this time period, and usually because the traceroutes go from the same probes to the same DNS Root servers, the paths should be the same if nothing happens, therefore, the observed usage for every hour should be almost the same.

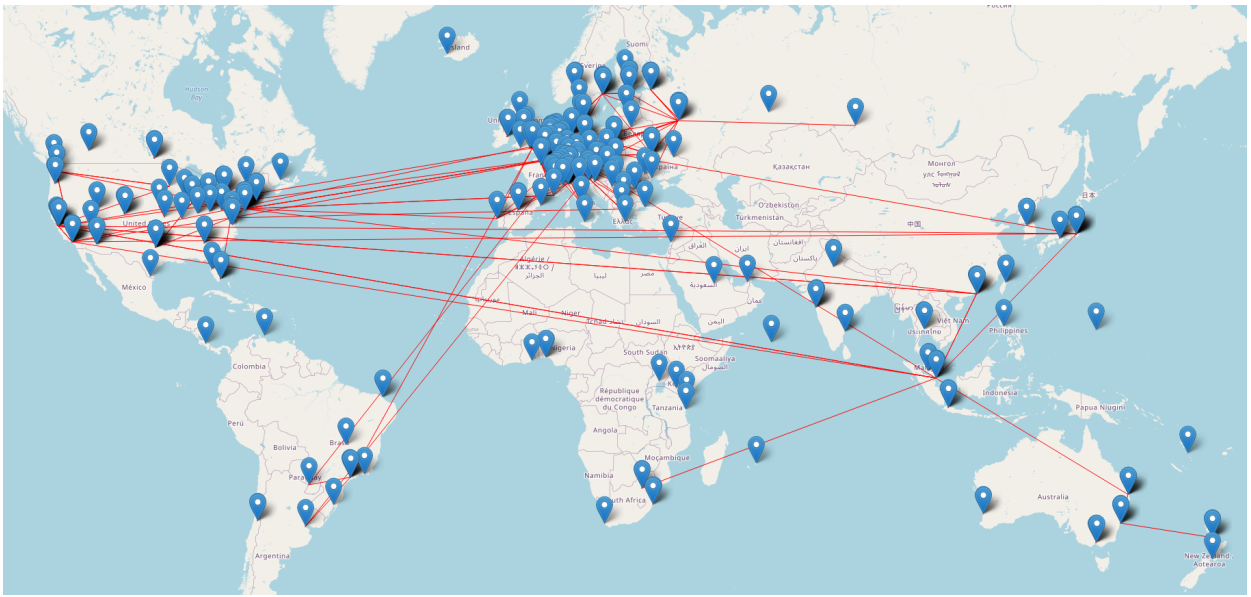


Figure 8: Map of the links between facilities observed during 1 hour of data

After Phase 1 and 2 the observed usage for each link can be easily obtained, but that will not provide any information on how the links evolve altogether, to do so, these values are put together in sets called Forwarding Models, where all links have a common near-end facility , for example:

```
1     "A": {
2         "B": 40,
3         "C": 150,
```

```
        "D": 478
    }
```

Listing 7: Forwarding model example

In the example shown in Listing 7 the forwarding model corresponds to facility A, had links with facilities B, C and D, and they were traversed 40, 150 and 478 times. This way the routing and forwarding pattern of facility A can be observed and evaluated altogether. These models then are monitored and evaluated every hour to track their evolution and detect unexpected patterns, the methods to do so are explained in more detail in the sections below.

The links that will be monitored are expected to be in the Europe and North America regions as seen in Figure 8, because is where most DNS Root servers and probes are situated which means that more traceroutes are going to traverse those links, making it easier for them to be identified in the Constraint Facility Search.

Forwarding Model Reference:

The first step to figure out when anomalies occur is to calculate reference values. Reference values for the forwarding model use the previous nine hours of data and reflect the behaviour of the links between facilities in the near past, making this way, a prediction on what the expected values for the next observation are. They are calculated by using the median of the forwarding models, to do this, the median is computed for each link individually with the usage recorded for every hour during the aforementioned period of time. The references are then compared to the observed model to compute the deviation between them and have a sense of how much the link usage has changed in regard to the pattern recorded the previous nine hours, this is shown in the next section where the link monitoring and alarm detection are explained.

Links that consistently appear are the ones of interest as a certain amount of stability is desired, which is why the methodology only takes into account links which have been utilised at least 5 times during each hour. Moreover, only links which have been utilised by at least 4 unique RIPE Atlas probes are considered to ensure that the traceroutes that account for those values are not the consequence of a probe malfunction, which for example could have reported the same measurement multiple times.

The value of nine hours was chosen due to the fact that a large majority of alarms (roughly 90%) reported in the related work lasted five hours or lower [3], which can be observed as large peaks or drops in the figures presented in the results section, for example Figure 22 or Figure 34. Using this nine hour period means that the alarms that last between 1 and 4 hours, which are most of them, are not going to alter the median value of the reference, this is because there will be more non anomalous values.

Moreover, other events can also trigger alarms, the ones that occur the most are routing changes. These are, as the name implies, a rerouting of some traffic from one link to another that is distinguished by a drop or an increase of the normal usage of the link as can be seen in the examples of Section 5.3.3. These events are not expected to occur by observing the past, which means that at first are going to trigger alarms because they are not an expected pattern, but as soon as they surpass the 5 hour mark with a stable pattern, then they are treated as routing changes. The reason why elongating the time period of the reference is so larger alarms, like in Figure 30, do not affect the median is not beneficial, as the longer the time period, the longer it is going to take to stop reporting alarms when a routing change happens. This is discussed more in Section 5.3.

Forwarding Link Monitoring and Alarm Detection: After the reference values have been

calculated, a new hour of data is loaded in, the 'observed values'. To compare the observed values with the reference values, the following steps are taken, depending on whether or not there are multiple links for the same near-end facility. To see the whole decision flowchart of this methodology, go to Figure 9.

If there are multiple links for the same near-end facility:

- The first step is to perform the Chi-Squared statistical test to compare the observed and the reference (expected) models. For this test, first the null hypothesis and the alternate hypothesis have to be defined:
  - **Null hypothesis:** This hypothesis accounts for the assumed behaviour, which in this case is that the observed and the expected forwarding models are consistent with each other.
  - **Alternate hypothesis:** This one is the hypothesis which is going to be assumed in case the null hypothesis is rejected. In this case is that the observed and expected forwarding models are not consistent.

  Once these are stated, the test consists in assuming that the null hypothesis is true, after that, the likelihood of the observed values when the null hypothesis is true is calculated, the name of this parameter is "p-value". If the sample relationship is extremely unlikely, then the null hypothesis is rejected and an alarm is expected, otherwise, the hypothesis is retained and it is considered as consistent, i.e, there are no alarms raised.

  To evaluate if the samples are consistent, a threshold is fixed for the p-value called significance, represented as α (alpha). In this case the threshold is set to 0.01 and as mentioned, this is the chance of sample when the null hypothesis is true. The obtained value will need to be higher than 0.01 to retain the null hypothesis, otherwise, it is rejected.
- If the first step raises a warning for an alarm, which means that the null hypothesis has been rejected, all the links that have the same near-end are evaluated individually to see which ones are responsible for the alarm. To do so, a simple function is created which conducts this evaluation by subtracting the observed value from the reference value, and dividing this by the reference: $r = \frac{(obs-ref)}{ref} \in [-1, \infty)$, this formula will provide a way to quantify the deviation in a single link without taking into account the others.
- If this evaluation results in a value lower than -0.2 ($r < -0.2$) or higher than 0.2 ($r > 0.2$), an alarm is reported. these values have been chosen by carrying on several experiments to observe if all the anomalies that were considered important were reported.

When an alarm is reported, the Mean-Squared Error (MSE) value for the individual link is saved, along with the MSE of the forwarding model (all the links combined), for further steps in the alarm classification. The forwarding model MSE is useful also to rank the alarms by magnitude, this calculation is shown in the equation below where n is the number of unique links for one near-end facility, F_i is the observed usage of the i link, and the $\bar{F_i}$ is the expected value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(F_i - \bar{F_i})^2$$

If there is only one link connected to a near-end facility, a similar methodology is followed, without calculating the Chi-Squared value. This is because the Chi-Squared test requires at least two links in order to calculate it.

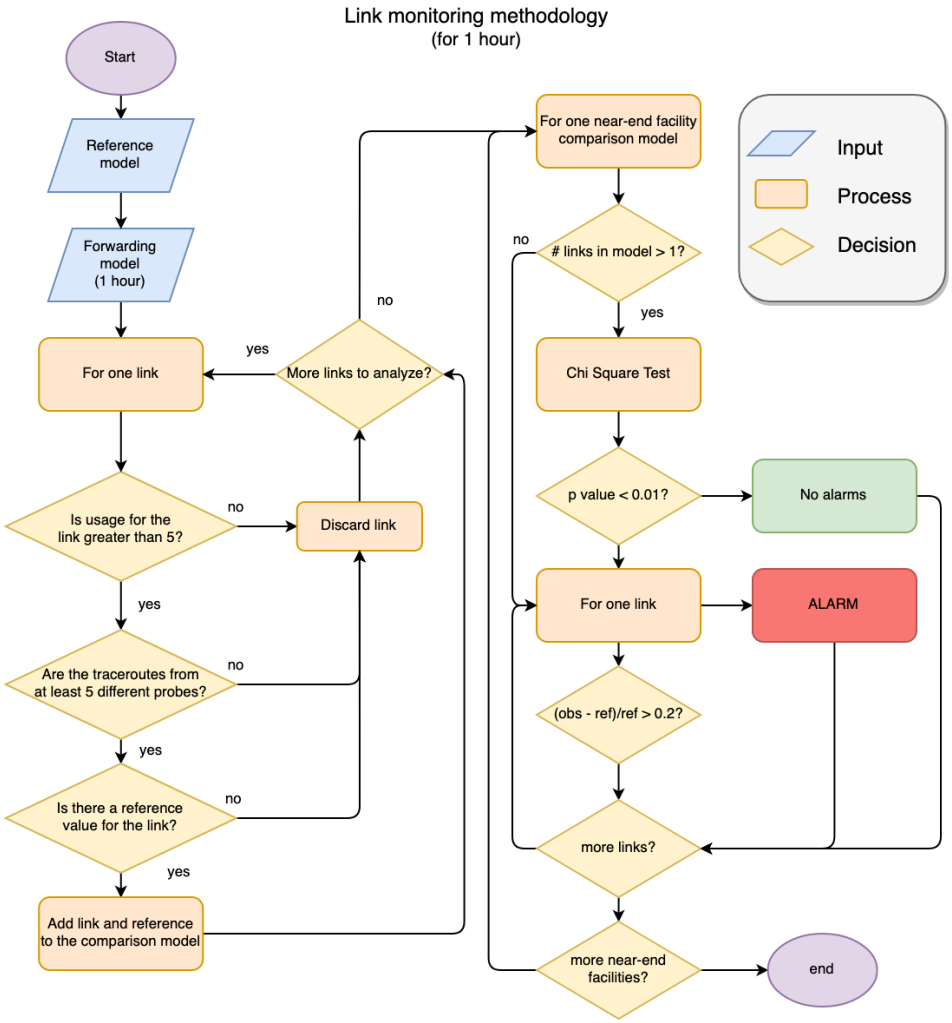


Figure 9: Overview of Link monitoring methodology

Example Forwarding Model module alarm detection

Initially after loading the forwarding model this is the information used for the link monitoring:

```
    "1": {
        "2":{
            "comp": [90, 60],
            "probes": [24423, 23534, 23544, 23424, 23434, ...]
        },
        "3":{
            "comp": [1250, 750],
            "probes": [34534, 354533, 35345, 34534, 34521 ...]
        },
        "4":{
            "comp": [0,30],
            "probes": [234243, 234243]
        }
    }
```

Listing 8: Phase 3 comparison model

In the listing 8, the first key corresponds to the near-end facility, and the next tier of keys corresponds to the far-end facilities, which values are lists that contain the reference value first, and the observed usage of the link in the next position. If the first value is 0 that means that the reference was not calculated due to lack of information and the link is not used for that hours monitoring.

The first step for the monitoring as shown in Figure 9 is to check if the link is used more than 5 times (all of them pass), and if the probes involved are greater than 4, which only the (1,2) and (1,3) links pass. In the third links case, it would also be discarded due to the fact that has no reference.

Once that filter is done, two lists are created for the Chi-squared test, the first one will contain the expected values (references), and the second one has the observed values, and they will look like this:

$$Expected = [90, 1250]$$

$$Observed = [60, 750]$$

$$\textbf{Chi-squared test} \rightarrow p - value = 5.107 * 10^{-8} < 0.01$$

Because the p-value is lower than the significance level fixed to 0.01, for this near-end facility, an alarm warning is raised, now each link is analyzed individually to observe the different deviations.

$$Link(1 \rightarrow 2) : deviation = \frac{750 - 1250}{1250} = -0.4 \rightarrow |-0.4| > 0.2 \rightarrow ALARM$$

$$Link(1 \rightarrow 3) : deviation = \frac{60 - 90}{90} = -0.333 \rightarrow |-0.333| > 0.2 \rightarrow ALARM$$

Alarm Classification:

After a list of alarms is noted for each hour, a classification of alarms is required. This is due to the fact that some links might have an unstable normal pattern (constantly having major changes regarding the usage) and thus report a lot of false alarms, or the normal pattern usage is very low, so it is more sensible to changes that apparently are small for other links.

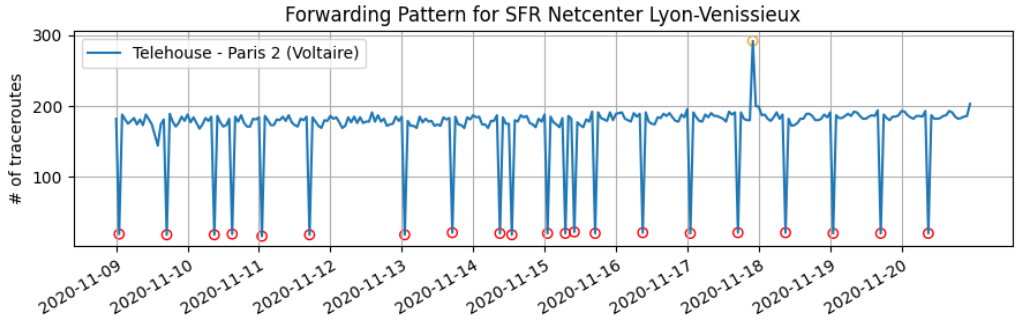
The classification is distributed in two different tiers of alarms, "reds" are the true alarms considered important or highly deviated from the reference, and "yellows" are the lower tier or slightly deviated alarms as well as the possible false alarms caused mainly by a low normal usage, that as mentioned above, are much more sensible to small deviations that might not indicate a true alarm. A summary of the alarm classification methodology is presented in Figure 13.

The first step of the classification is to calculate the reference values of alarms observed across the span of one week, using the median value alongside a confidence interval. This period of time has been chosen because the number of alarms is much lower than the hourly samples and allow a precise calculation of this confidence interval, the link requires at least ten triggered alarms over the past week, if this is not accomplished, the reference is not calculated meaning that the only check that can be done is if the deviation is very high, or low. The confidence interval used is based on the Wilson score, which will be explained in further detail in Section 3.5.2. Because one full week of data is needed to do this calculation, during the first week of the observation time span, there is no classification, this will be clearly seen in the following graphs.

Once the references are calculated, for the next steps, the methodology is based off three cases that have been observed during the development of this phase, in the figures presented below, the

evolution of the usage of the link is shown with a blue line, representing the number of traceroutes that traverse the link every hour, and the red and yellow circles, are the alarms triggered during that period of time. On top of the figures there's also the names of the facilities forming the link.

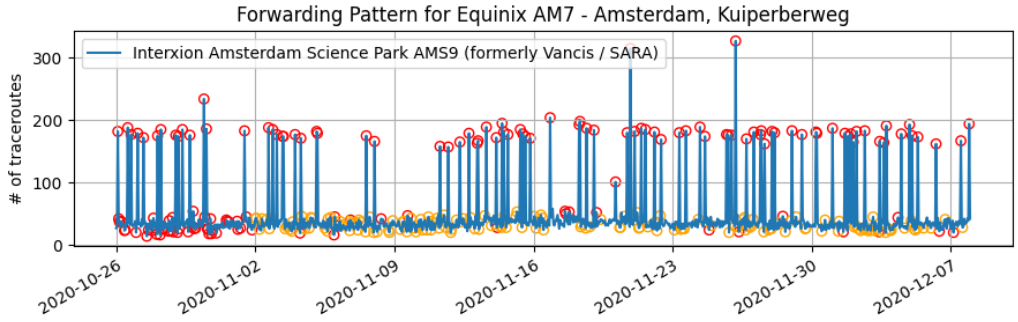
- **Case 1:** In this first situation, the normal pattern is high enough in traceroutes/hour that the small variations that occur do not get reported as alarms like you will see in case 2 and 3. As presented in Figure 10a, the alarms reported are all considerably important deviations, but some smaller anomalies can happen that are not necessarily as important to report as the others, i.e the one that appears around the 18th of November marked in yellow.



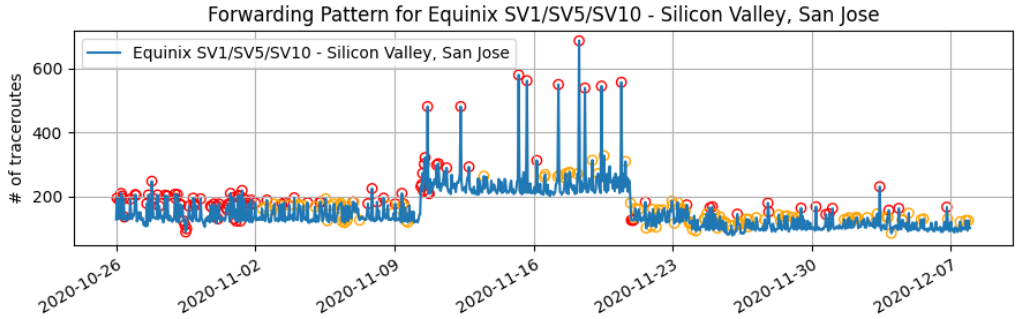
(a) Link from SFR Netcenter Lyon to Telehouse - Paris 2(Voltaire)

Figure 10: Alarm pattern case 1 examples

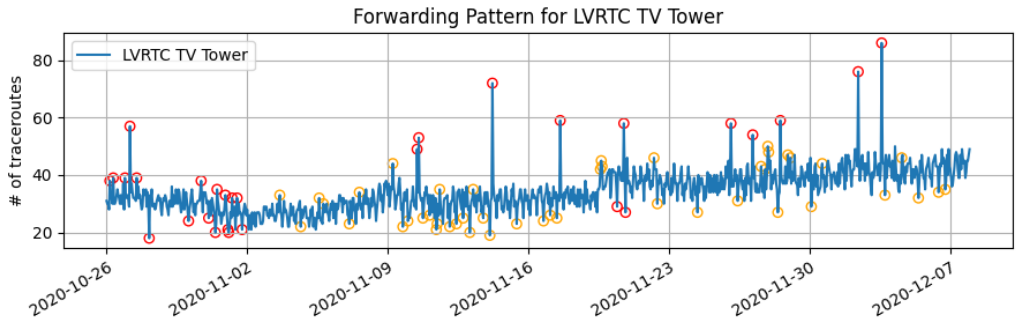
- **Case 2:** The alarms reported are mostly false due to the effect of having very low usage in the normal pattern, i.e Figure 11a shows most of the alarms very close to the normal pattern that is close to 50 traceroutes/hour, but the alarms that actually have more importance when it comes to deviation appear at around 200 traceroutes/hour for some hours only. The same thing happens in Figure 11b where most alarms reported have a deviation of approximately 50 traceroutes compared to the normal pattern, but between the 9th nad the 23rd of November, much larger deviations are observed considered of higher importance. In this situations, the reference will represent the evolution of the low deviation alarms or "false" alarms.



(a) Link from Equinix AM7 Amsterdam to Interxion Amsterdam Science Park AMS9



(b) Intralink in Equinix SV1/SV5/SV10 - Silicon Valley, San Jose



(c) Intralink LVRTC TV Tower

Figure 11: Alarm pattern case 2 examples

- **Case 3:** This last case, regards the situation when multiple links with different levels of usage are observed for the same near-end facility. An example is presented in Figure 12 where two different links can be observed. Taking a look at the one close to 0, due to no major alarms reported, only by looking at that one link the alarms would be considered red, because if the reference value is at about 40 traceroutes/hour, then a deviation of about 20-30 traceroutes which is completely normal in most links, would report an alarm because comparatively is a 50 to 75 % increase, but when put in perspective with the other links, these alarms do not represent such a big deviation for the forwarding model and need to be reported as yellow.

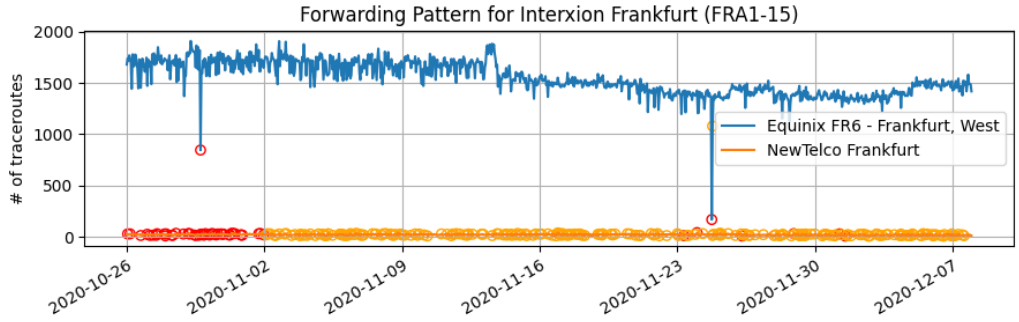


Figure 12: Alarm pattern case 3 example

The first 2 cases, depend on the reference computing, because the methodology can only be used if the reference has been previously calculated. If the first condition is met, the new alarms MSE values (only considering that link) are compared with the reference interval, and there's three different outcomes. The first is when the MSE is lower than the reference interval, then it's considered first case, meaning that the alarm will be "yellow" because the reference is considered to report the level of the high deviation alarms, as can be seen in Figure 10a. On the other hand, if the observation is higher than the interval, it's considered the second case, which means that the alarm will be "red" because the reference is probably reporting the false alarm pattern, as can be seen in Figures 11.

Finally, if none of this cases is granted, it means that the observation is either inside the bounds of the interval or doesn't have a reference. For these cases, the procedure is to use the comparison value from the alarm detection in the previous section, and if the absolute of the comparison is above 0.4 $\{\frac{(obs-ref)}{ref} < |0.35|\}$ the alarm will be considered "red", otherwise, "yellow".

Once the first classification is done, to deal with situations corresponding to case 3, the two MSE values calculated for each alarm are used, these being the forwarding model MSE and the link MSE, then, a calculation is carried to obtain the contribution from each link to the forwarding model deviation:

$$contribution = \frac{link\_mse}{fw\_model\_mse}$$

If the contribution is less than 0.3, the alarm of the less used link is considered "lower tier". Here's an example of this process:

In Figures 11 and 12 from the 26th of October to the 2nd of November no classification is done as explained previously, and can be compared with the results when this methodology is applied the rest of the time, most of the alarms that were reported as red at the beginning without very large deviations, later are reported as yellow fro the most part.

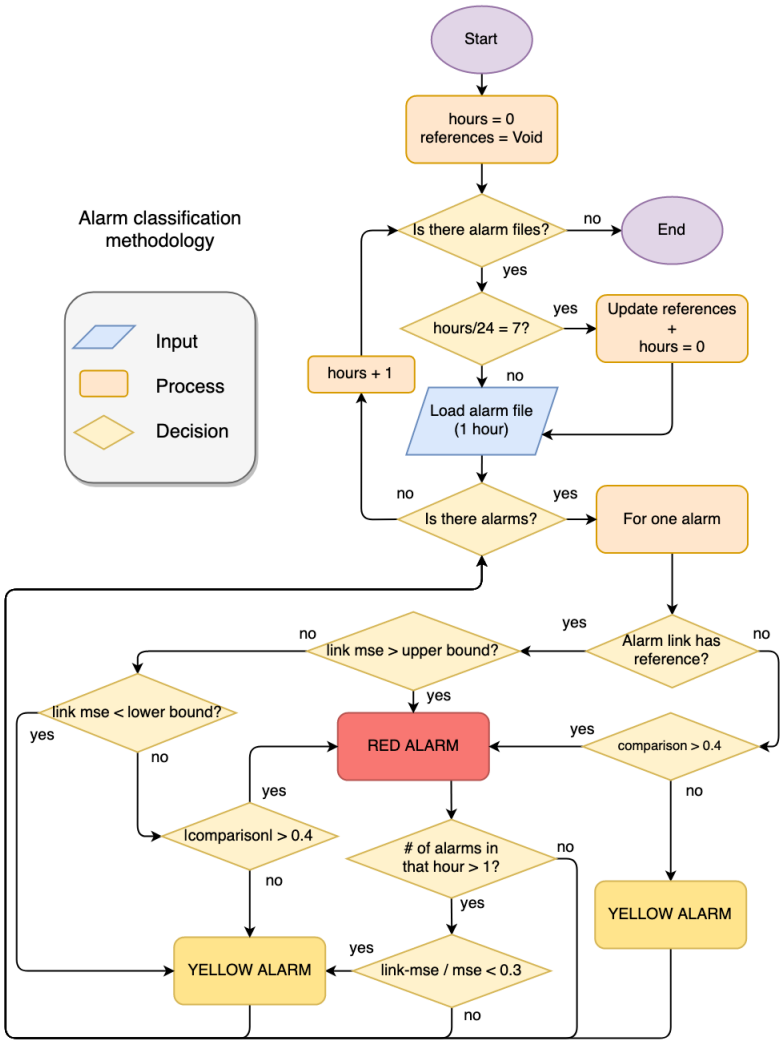


Figure 13: Overview of Alarm classification methodology

3.5.2 RTT Delay Monitoring

Continuing from the forwarding model, once a facility pair has been identified, it is possible to calculate the RTT value for that link. This is the time it takes for a traceroute to cross that link between two facilities. This value is called as differential RTT, and is denoted by RTT_{diff}. For instance, RTT_{diff} for a link between near-end facility 'Facility A' and far-end facility 'Facility B' as shown in Figure 6 is calculated in the following manner:

$$RTT_{diff} = RTT_{FacilityB} - RTT_{FacilityA}$$

The RTT values used here are extracted from the original traceroute measurements, and a simple subtraction is done. The reason why absolute RTT values are not used here is because of the fact that this would allow for the delay results to have high fluctuations [17].

RTT delay monitoring method consists of two steps. The initial step involves calculation of initial reference values for differential RTTs and the computed reference values are used to build a normal reference pattern for differential RTTs. Once the the initial reference values are computed, next step is to detect anomalies by monitoring the differential RTTs between links and comparing these

monitored values against the computed initial reference values. These steps are explained in detail below.

Initial reference computation: In this step, normal reference value for each link is computed and maintained on an hourly basis and this value can be used as a base value upon which an RTT delay change detection can be performed. A sliding window of 24 hours is used to compute the initial reference. For example, initial reference value used to detect an RTT delay anomaly at a particular hour x is the median value for RTT values over links across 24 hours of traceroute data prior to x. In order for the data to be as stable as possible, only links which are traversed more than five times per hour are selected. Apart from the median value, confidence intervals are also calculated to account for the uncertainty in the median [17]. The confidence intervals are calculated using the Wilson score as the data is distribution free and a binomial calculation, and the Wilson score has good performance a small sample of data [17]. The Wilson score is defined by the following equation, in which n represents the number of samples, p is the probability of success, and z is set to be 1.96, representing a 95% confidence interval.

$$w = \frac{1}{1+\frac{1}{n}z^2}\left(p+\frac{1}{2n}z^2 \pm z\sqrt{\frac{1}{n}p(1-p)+\frac{1}{4n^2}z^2}\right)$$

This results in two values, w_l and w_u, which when multiplied by the number of samples, n, gives the lower and upper bound of the confidence interval: $l = n * w_l$ and $u = n * w_u$ [17].

For instance, there are n differential RTTs computed for a link, say, ${RTT_{diff}}^1$, ${RTT_{diff}}^2$, ..., ${RTT_{diff}}^n$, and these values are in ascending order, i.e., ${RTT_{diff}}^1 \leq {RTT_{diff}}^2 \leq ... \leq {RTT_{diff}}^n$. Here, the lower and upper bound of the confidence interval is given by ${RTT_{diff}}^l$ and ${RTT_{diff}}^u$ respectively.

The sliding window of 24 hours is divided into two splits such that the second split contains those hours that are closer to the measurement hour. The normal reference median value, $mRTT_{ref}$ is calculated such that the second split is given a higher weightage compared to the first as shown below:

$$mRTT_{ref} = 0.1 * mRTT_{s1} + 0.9 * mRTT_{s2}$$

Here $mRTT_{s1}$ is the median of differential RTTs for the first split and $mRTT_{s2}$ is the same for second split. The second split is given more weightage compared to the first one.

In order to fix a ratio in which 24 hours have to be divided into two splits as discussed above, an empirical evaluation of number of alarms obtained using different split ratio is performed. For this evaluation, a cdf plot is plotted to compare the number of alarms raised across the links monitored for more than a month. Figure 14 shows the cdf plot for links monitored for a time duration from 26/10/2020 till 01/12/2020. Total number of links monitored in this period was 251. 181 links out of the 251 monitored links do not raise an alarm , which means those links are quite stable. For this plot, the remaining 70 links are considered. X-axis is the number of alarms in powers of 2 and Y-axis is the cumulative distribution function for percentage of links that raise alarm during the monitoring period. As seen in Figure 14, a colored curve represents a particular split ratio used. Different splits used in this evaluation are 12/12 (green curve), 18/6 (red curve), 21/3 (blue curve) and 24/0 (yellow curve). Here, a split of x/y means first split consists of first x hours from the sliding window of 24 hours and second split consists of last y hours.

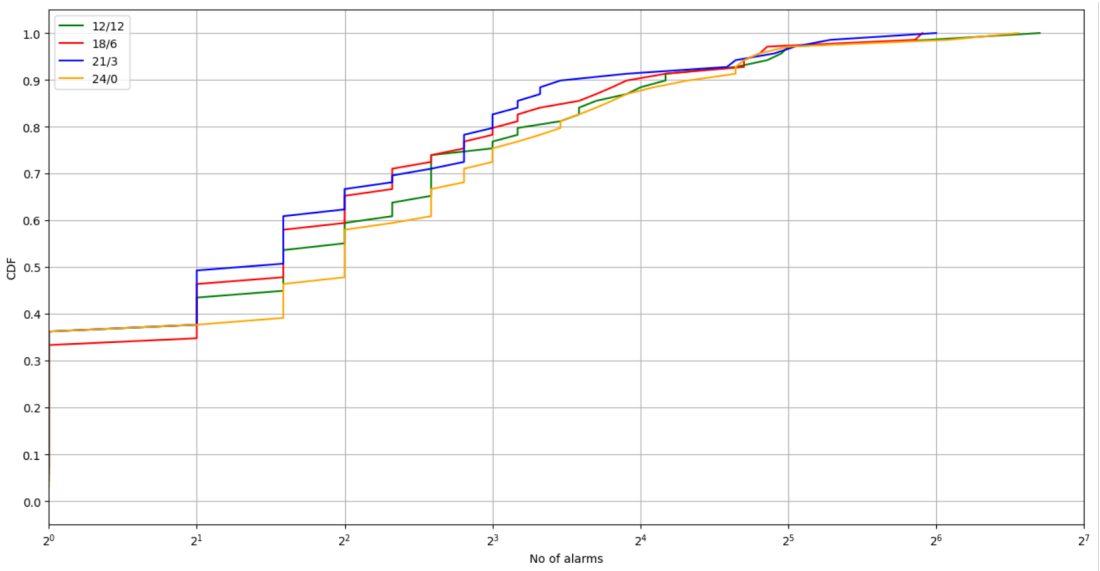


Figure 14: CDF plot for number of alarms across links monitored for one month duration

From Figure 14, it can be seen that for the split ratio 24/0, about 37% of the links raise at most 2 alarms and for the split ratio 21/3, about 50% of the links raise at most 2 alarms (refer to the values on these curves corresponding to X-axis value equal to 2^1). Similarly, the values are 42% for 12/12 split and 45% for 18/6. This observation indicates that the split ratio 21/3 gives a more stable percentage of links compared to other split ratios. On observing the tail portion of the curves, the maximum number of alarms raised by the links monitored can be inferred. With the split ratios of 21/3 and 18/6, the maximum number of alarms raised is reduced by a factor of 1.5 with respect to the other split ratios. Thus based on the cdf plot, 21/3 split is chosen as the optimum split ratio in terms of minimum number of alarms raised.

Anomaly detection based on RTT delay monitoring: This step involves monitoring of links on differential RTT values. At first, the median of RTT_{diff} and confidence intervals are measured for the current hour in which links are monitored. These values are compared against the corresponding initial reference values of median RTT_{diff} and confidence intervals.

An anomaly is reported when the gap between confidence intervals of the monitored hour and the confidence intervals of the initial reference values is more than a preset threshold value which is different for different links. This threshold value is calculated as the difference between the median of upper bound RTT values across the sliding window of 24 hours obtained from the confidence interval with that of the median of lower bound RTT values. For this reason, these values are updated on an hourly basis. This approach to dynamically calculate threshold values for different links is performed as an improvement over the method that uses a static 1ms threshold value as described in the related work[17]. Dynamic threshold values seems to work properly in multiple cases, especially when the two facilities are located in different cities and a single threshold value would not be good to account for the RTT delays in such cases. By using this approach, the detected anomalies can be significantly reduced compared to the static method since not all anomalies trigger the alarm.

3.6 Data Validation and Reliability

In order to validate the data and the results achieved, manual checks with PeeringDB and IXP websites were required. In the early stages of the project, in order to validate the scripts for

traceroutes and IXP information, a small set of values were chosen to examine and these were cross referenced with the information on the PeeringDB website [6]. Generally, information about IXPs present on PeeringDB are kept up to date for reasons mentioned in Section 3.4. Regarding the reliability of the results achieved, a comparison has been made to the other studies which have used a similar methodology, such as [3].

Regarding the validity of our anomaly detection tool, the methods to verify this involve using websites that utilise different methods to gauge anomalies occurring in the infrastructure of the Internet. Examples of such websites are Fing [21], ThousandEyes [22], Internet Health Report [23], and Internet Disruption Report [24]. Without knowing exact anomalies which have occurred in the past to analyse and report, a manual check is needed with these websites to figure out if an anomaly reported by our methodology is also seen in other anomaly detection services.

4 Implementation

This section discusses the implementation of the three phases of the project. Firstly, an overview of the whole implementation describing the programming environment, different libraries/packages used in this project and specifications of VM environment is provided in Section 4.1. Section 4.2 and Section 4.3 discusses the implementation for Phase 1 and Phase 2 respectively, and Section 4.4 discusses the implementation of the final anomaly detection tool.

4.1 Overview

The project implementation is carried out in three phases: Phase 1, Phase 2 and Phase 3 as listed in Section 3.1. Phase 1 deals with fetching of peering infrastructure data. Phase 2 is about identifying the near-end and far-end facilities using Constrained Facility Search. Phase 3 is associated with facility link monitoring to detect network disruptions. More detailed description of each phase is given in Figure 15. This figure is an extended version of Figure 5, detailing the implementation and execution of each phase.

4.1.1 The Programming Environment

The project involved developing several scripts to extract/fetch relevant data from traceroutes and different databases such as PeeringDB and CAIDA. These scripts were written in Python. Python was chosen at it offers a wide range of libraries that consist of packages which are useful for data analysis, statistical analysis and visualisation. Another reason to choose Python over other programming languages was because of the type of data that was analysed as part of this project. Most of the data that were fetched for analysis was of json format, and Python facilitates working with json data.

4.1.2 Libraries/Packages used

An explanation of the libraries and the reasons that we used each one in this project is provided below:

tqdm [25]: The tqdm package provides a progress indication for iterable objects in python. By decorating an iterator, a progress bar is dynamically printed which keeps track of the iteration steps until the iterators completion.

This package is used to monitor the number of iterations per second when it has to extract item by item from a parsed json string as well as tracking the time that takes to run those iterations.

ujson [26]: This module is used to convert python data objects and save them in json data format. This is an ultra fast json encoder and decoder compared to the built-in 'json' package offered by Python.

RIPE Atlas traceroute dump available for one hour has a size of 17 gigabytes. Processing of json data from such large files needs to be done in the quickest way possible. The ujson module which has a much higher calls/second (performance gain) compared to json module, helps to achieve this.

networkx [27]: networkx is a Python package available that can be used for creation, manipulation, analysis of the structure and functions of large and complex networks.

In this project, network x is used to create a map of the peer data exchange made between colocation facilities.

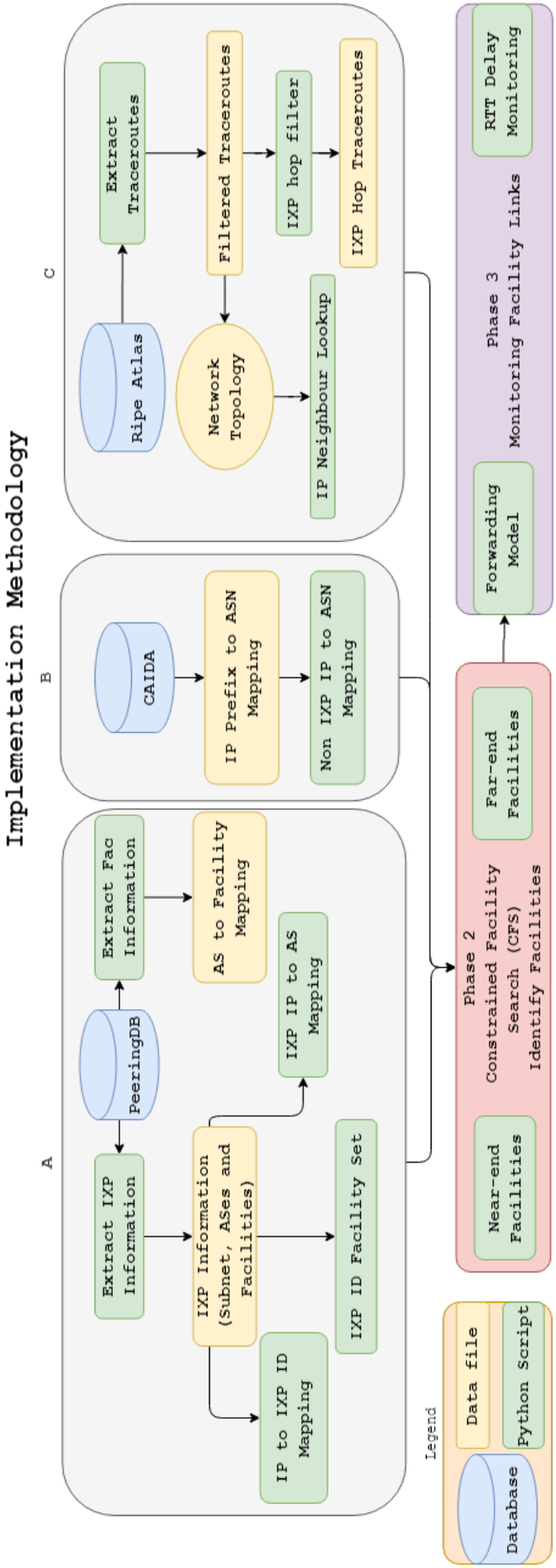


Figure 15: An Overview of the Implementation

matplotlib [28]: matplotlib is a library to create visualizations in Python. These visualizations can be static, animated and interactive.

This is the library we use to plot all relevant information to our data in this project.

pytricia [29]: This is a library that stores prefixes in a Patricia tree and allows for a fast IP address lookup. This library was mainly used to facilitate IP to IP subnet mapping in several scripts developed as part of the project.

4.1.3 Technical details of VM environment

The execution environment for the project was mainly Linux. The technical specifications of the VM used are listed in Table 1 below, with the important factors being RAM and storage as large files were processed:

Table 1: VM Specifications

Architecture	x86_64
Number of CPU cores	4
Processor type	Intel Core Processor
System memory (RAM)	32GB
Linux version	4.15.0-117-generic
Storage	100GB

4.2 Phase 1 Implementation

As noted in the overview, Phase 1 was divided into three blocks. These three blocks are explained in this section. The systematic implementation of the project can be seen in Figure 15.

4.2.1 Block A: PeeringDB Data Extraction

Block A implements a set of scripts that extract and use PeeringDB information. Beginning with extracting this information, two different scripts are used: one to extract IXP information and the other to extract facility information.

IXP and Facility Information: The IXP information script uses the API from PeeringDB to extract the relevant information such as IP subnet, ASes and facilities associated with the IXPs. This information is stored in an output file with all the IXPs and their corresponding useful peering data. Once this information has been extracted, an additional set of scripts are created, which are mainly used for Phase 2 to identify the near-end and the far-end facilities using Constrained Facility Search (CFS). Descriptions of these scripts are as follows:

IP to IXP ID mapping: An IP to IXP ID mapping was created as a script which consisted of receiving as input a set of IP replies, and it returns whether any of these IP replies belongs to an IXP subnet. If the IP replies belong to an IXP subnet, the script should return the IXP id. To implement this script, the pytricia library was used to provide a fast prefix lookup.

IXP IP to AS mapping: The main purpose of this script is to map which ASN does an IXP IP belong to. The script takes a set of IXP IPs as input and returns IPs and their ASs.

IXP ID Facility Set: This script receives a set of IXP IDs and returns the list of facilities in which the IXP is involved.

Also, a mapping between ASs and facilities is created using the PeeringDB API 'netfac' as described in Section 3.2 to use later in Phase 2.

4.2.2 Block B: CAIDA Data Extraction

Block B provides a lookup for non-IXP IPs to AS numbers. CAIDA dataset is used for this purpose as PeeringDB provides only the mapping between IXP IPs to AS numbers, and not for non-IXP IPs.

Non IXP IP to AS mapping: The objective of this script is to do a IP lookup with the pytricia library, using the prefix to ASN mapping to find the corresponding ASN of the input IP value. This ASN information is used later in Step 2 of Phase 2 methodology to find the facilities associated to that ASN.

4.2.3 Block C: Traceroute Parsing and Filtering

Block C implements parsing and filtering for traceroute data extracted from the RIPE Atlas database.

Extract Traceroutes: This script extracts relevant traceroute information from a RIPE Atlas traceroute dump. The relevant data are the hops, the RTTs of each hop in a traceroute, the measurement ids and the probe ids of the traceroute. Furthermore, as each hop has a varying amount of distinct measurements of the RTT (the RTT measurements can range from two to five, with three being the most common), this script calculates the average RTT of these values. This RTT value is used for RTT delay based monitoring of links in Phase 3.

IXP hop filter: This script takes as input the results from the previous script. Here, the traceroutes that have an IXP IP hop are filtered from the input and the output is the IXP hop, with its previous and next hops as well as its corresponding IXP ID. The output is then written in another file similar to the input. This script is needed to infer the previous hop of the IXP hop to identify the near-end facilities in Phase 2.

IP Neighbour Lookup: The objective of this script is to gather topological information from traceroutes. Given an IP hop as the input, it returns all the next hops (neighbors). This script is used to identify the facilities later in Step 3 and Step 4 of Phase 2 methodology.

4.3 Phase 2 Implementation

The implementation for Phase 2 is illustrated by the red block in Figure 15. This block is implemented using two main scripts. Both scripts apply the CFS methodology described in Section 3.4, one applies the near-end methodology and the other one the far-end version.

Near-end Facilities: This script receives as input the traceroutes of one hour with a visible IXP hop, obtained from the scripts written in Phase 1, and executes the near-end methodology with each traceroute. Once the whole file is analysed, it provides a list with a mapping of the 'previous hop' to facility, as well as some basic statistics such as the success rate for each step described in the methodology. It also gives the number of unique facilities found as well as the average number of total IPs per facility.

Far-end Facilities: Following the same structure as the near-end script, this one also provides an IP to facility mapping of the facilities successfully inferred during the CFS using the far-end methodology.

Once the near-end and the far-end facilities are identified, this information is stored in a dictionary that contains all the link usages for a particular hour for all the traceroutes, along with other information of importance for the monitoring phase like the actual rtt values for each rtt-diff, the unique probes involved in that links usage and the measurement ID of each trecaroute (not unique). The structure of the resulting dictionary looks like below:

```
{
    "(FAC-near, FAC-far)": {
        "rtts": [(RTT_IXP - RTT_IPA), ...],
        "actual_rtts": [(IP_A RTT, IXP_hop RTT), ...],
        "probes": [unique probes involved],
        "msm_id": [measurement ID of each traceroute]
    }
    ....
}
```

Listing 9: Phase 2 link usage result example

4.4 Phase 3 Implementation

As written in the Methodology section, the implementation of Phase 3 consists of two sections, the forwarding model and the rtt delay monitoring. These will be discussed in the following sections, with examples provided.

4.4.1 Forwarding Model Monitoring

Starting with the Forwarding Model, the implementation scripts are divided in two groups, the first scripts are used to trigger the alarms for anomalies and the second scripts classify the alarms to differentiate the most significant ones.

4.4.1.1 Reference Calculation

There are two different references, one is used for the alarm triggering and the other is used for the alarm classification, in both cases only the links that are consistent in their respective time periods are taken into consideration, otherwise they are discarded for that reference file.

4.4.1.1.1 Link monitoring references

This reference considers the link usage values for the 9 hours prior to the slot to analyze, to do so, all the files containing the values for those hours are loaded one by one, and once all the usage values for the links are in their respective lists, the median of these is carried to obtain the final reference values that are returned as an output.

4.4.1.1.2 Classification references

For this reference, the time period considered as explained in the methodology section is a week, in this case, instead of only executing a median of the values observed, the confidence intervals for

the median are also calculated in the same way as for the rtt references, that's why to ensure the values are more precise, only the links with at least 10 alarms have a reference. Another difference with the triggering references is that these are calculated weekly and used for the alarms triggered in the next full week.

Alarm triggering (Link monitoring): These process is done by loading the observed values and the references for that time slot, once those are loaded into a comparison dictionary, the steps explained in the methodology are done to decide if there's alarms or not, and once that is done, a file with a list containing all the alarms is returned. In this file all the alarms are considered important ("red").

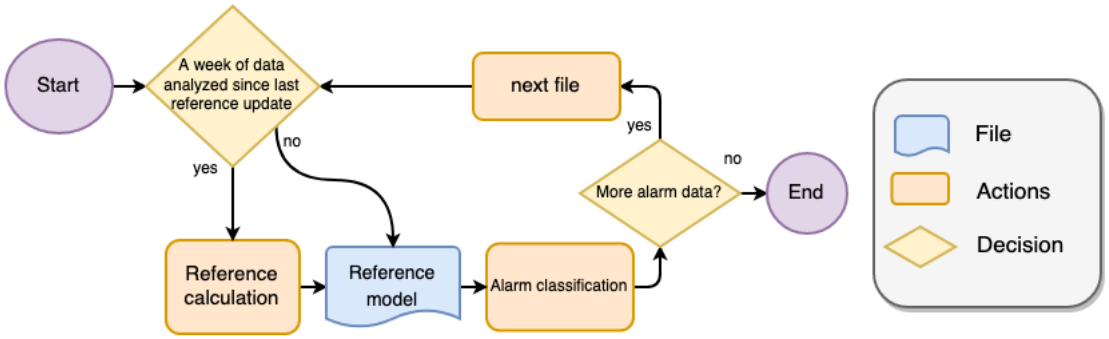


Figure 16: Overview of the alarm classification implementation

Alarm classification: Once the forwarding model link monitoring is done and all the alarms are raised, for each hour a classification is done, according to the previous methodology explained, to carry on this process, at least a week of data is needed, that's why for the first week of the data pool this is not done as it can be seen in Figure 16. Once this is done for every alarm file, the resulting classified alarms are saved into a file with two different lists (one for "yellow alarms" and one for "red alarms").

4.4.2 RTT Delay Monitoring

RTT delay monitoring implementation is divided into two main scripts. The first script is to compute the initial reference values for a link on an hourly basis and the second script is to monitor the link and to detect anomalies based on RTT delays.

Initial Reference Computation: The script takes as input the start date and end date for which monitoring has to be performed. It starts from computing the initial reference values for every hour in this period using the sliding window method described in 3.5.2. Only those links which appear consistently in every hour are considered here.

The end result of this script is the confidence interval and the median differential RTT value that can be used as the initial reference values for every link monitored in each hour. This result is stored as a dictionary with the links as its keys and corresponding initial reference computed as its values. This dictionary is used as the normal reference to monitor links and detect anomalies for a specific given hour.

Link Monitoring: The next script is used to monitor a particular link for RTT delays within a given hour. The goal here is to detect any anomalies or disruptions that occur in the link at that hour. Similar to initial reference computation, the monitoring starts from computing the

confidence interval and the median of differential RTT value for that link at that particular hour. Then, the comparison between the current intervals and the normal reference intervals will trigger an alarm if the difference between these values exceeds the preset threshold as described in 3.5.2.

The end result of this script is the confidence interval and the median differential RTT value for each link monitored. This result is also stored as a dictionary on an hourly basis similar to the initial reference case. Apart from these values, the links that trigger an alarm at each hour is also stored. For those hours where no links raise an alarm, it can be possibly because the links are quite stable during these hours.

Along with the differential RTT values for anomaly detection, actual RTT values for the near-end and the far-end facility for a link is also extracted from the output of constrained facility search in Phase 2. These values can be useful later when the graphs are plotted, so as to mark the actual RTT values that trigger an alarm.

5 Results and Analysis

In this section the results are presented and analysed. The results for this project can be divided up into three parts, each corresponding to the Phases implemented. Section 5.1 presents the results for Phase 1, while Sections 5.2 and 5.3 presents results for Phases 2 and 3 respectively.

5.1 Data Collection Phase Results

This section contains a comparison of the original statistics noted by the databases involved in the implementation, with the actual results extracted once Phase 1 is concluded.

PeeringDB: Table 2 shows the statistics extracted from the results of the data related to facilities, IXPs and networks extracted from PeeringDB. The values in the left column are global database statistics presented in the main web page of PeeringDB accessed on the 20th of October 2020. [6]. These statistics are compared to the ones in the right column, which represent the statistics that were actually usable and which were extracted for Phase 1.

The first row values are the number of IXPs, and as seen, almost all of the original values are extracted **(95.5%)**. Following that, the networks or ASes value is a merge of two different network sets, the first network set is extracted from the IXP information dataset and contained 11 002 Networks, the second one uses the ASN to Facility mapping, resulting in 8 286 Networks. These sets were merged to make a list of unique networks, which accounts for **94.7%** of the PeeringDB value. For the facilities the same method is used to obtain the final value accounting for **76.1%** of the total facilities mapped in PeeringDB, in this case in the IXP dataset there are 1 198 Facilities and 2 910 in the ASN to Facility mapping. The last two values correspond to the total number of ASN to IXP connections and the total number of networks installed in Facilities, losing 2.5% on the first one and 1% on the last one.

Table 2: PeeringDB Statistics

	Original Values	**Results**
Exchanges	822	785
Networks	20 358	19 288
Facilities	3 902	2 982
Connections to Exchanges	35 134	34 253
Networks in facilities	32 065	31 763

Looking at Table 2, a clear loss of information can be observed between the values stored in PeeringDB and the values extracted, this is due to the fact that PeeringDB is a user-maintained database and some of the values are not completely up-to-date or in some cases simply lack information. For example, the number of exchanges is affected by the lack of information by some IXPs, which have no information regarding IPv4 deployment, therefore are not considered useful for the scope of this project.

As mentioned in Section 3.4, the most reliable information is data related to IXPs, and this can be clearly observed as the number of facilities obtained when adding up all the facility sets of each IXP and each ASN is much lower than the other values. This results in the fact that not all IXPs and ASes installed in facilities are reported in PeeringDB.

Moreover, connections to exchanges have a very slight loss due to the fact that this is compared to the values that are reported in the database. The same thing happens with the last one, only in this case its evaluating the number of networks that have peerings in facilities.

RIPE Atlas Platform: The information in Table 3 represents the size of hourly traceroute dumps from the RIPE Atlas platform in which different Phase 1 scripts were applied. To obtain these results, three randomly selected different dump files are contemplated, each one contains the results for the built-in measurements of **one hour**. Looking at the original values, these represent the amount of different traceroutes a hourly dump contained. Each file was then filtered for only IPv4 traceroutes that had complete RTT information. The size of the resulting file can be seen in the second column. In the third column, the size of the file after searching for IXP IP addresses can be seen.

As it can be seen for the 14 of September file, in the first filter only **54.2%** of the original set of traceroutes remains, and from those, **28.8%** have a IXP hop, corresponding to **15.5%** of the total traceroutes inside the original file.

Table 3: RIPE Atlas Traceroute File Evolution

	Original file	**Filtered**	**With IXP hop**
Traceroutes (2020-09-14:11:00)	6 215 939	3 358 224	966 498
Traceroutes (2020-10-14:11:00)	6 361 752	3 424 601	1 006 298
Traceroutes (2020-10-17:06:00)	6 359 743	3 405 786	1 001 236

The traceroute files that are available as hourly dumps on RIPE Atlas platform have excessive information which is not relevant for the project implementation. Since the project focuses only on monitoring Paris traceroutes which have hop IPs that are of IPv4 version, it was necessary to filter these original files to extract only the relevant traceroutes. Also, the information of interest from the traceroutes is only the information about hops. This filtering of relevant information helped in obtaining files of significantly reduced sizes that were easier to work with.

Table 3 lists down the names of different traceroute files chosen for testing purposes, along with the statistics after each filtering rounds or steps. The first two traceroute files are the datasets that are a month apart and the third one is an hourly dump taken at an hour different from the first two. These traceroute files were chosen with a purpose to analyse the variations in traffic in a clear manner.

Looking at Table 3, the original file which had a file size of 17GB had about 6 million traceroutes. The first round of filtering discarded all the non IPv4 as well as non Paris traceroutes, and extracted the information about hops from the remaining traceroutes. This filtering took about 8 minutes on the VM and resulted in a new file marked as filtered traceroutes in the table, which had about half the number of entries compared to the original file. These filtered traceroutes were examined even further to extract traceroutes with an IXP hop inside them. The result of this filtering can also be seen in Table 3 under the title "With IXP hop". The resulting file is roughly 100 megabytes with 1 million entries. Also, there were not any significant differences in the values obtained from different traceroute files.

CAIDA

Similarly, from the Routeviews dataset on CAIDA about IP prefix to AS mappings, the following

result is the number that appears in the file from 2020-10-05 at 12:00. The file contained 882,446 IP Prefix to AS mappings.

5.2 Constraint Facility Search Results

This section presents the results and the statistics obtained from Phase 2 of the project, i.e, the implementation of the CFS methodology and identifying colocation facilities (Section 3.4).

After fixing the threshold value to 75% (as discussed in Section 3.4), these results were obtained using the same three files shown in Section 5.1 to compare them and asses the reliability of the algorithm used. These three files are chosen because they provided representative cases and the results for other files were similar to the ones shown below. Tables 4 and 5 show the success rates for each step of the algorithm used to infer the facilities. From the near-end facility search success rates listed in Table 4, the overall success rate of the algorithm used in Phase 2 ranges between 40% and 42%. The result is slightly lower for far-end facility search as listed in Table 5, and the value is around 40%.

Table 4: Near-end Facility Search success rates

Traceroute File	Step 1	Step 2	Step 3	Step 4	Total
2020-09-14:11:00	2.15%	29.27%	0.09%	10.49%	42%
2020-10-14:11:00	1.98%	29.12%	0.11%	11.18%	42.39%
2020-10-17:06:00	1.96%	28.69%	0.10%	9.96%	40.71%

Table 5: Far-end Facility Search success rate

Traceroute File	Step 1	Step 2	Step 3	Step 4	Total
2020-09-14:11:00	2.15%	34.33%	0.35%	3.96%	40.3%
2020-10-14:11:00	1.98%	34.35%	0.58%	3.47%	40.38%
2020-10-17:06:00	1.96%	34.31%	0.48%	3.49%	40.24%

Table 6 and Table 7 provide a detailed statistics on the unique near-end and far-end facilities identified respectively, the average number of IPs mapped to these facilities and the maximum number of IPs mapped to a facility in each case. One interesting thing to note from these statistics is that in both near-end and far-end case, the maximum number of IPs are mapped to the same facility with facility id 58 for all the three traceroute files used so far in the project. The information available in PeeringDB about facility id 58 reveals that this id corresponds to Interxion Frankfurt (FRA1-15) colocation facility which is connected to hundreds of networks and 15 IXPs.

Table 6: Near-end Facility Search statistics

Traceroute File	Unique Facilities	Avg. IPs / Facility	Max IPs mapped to Facility
2020-09-14:11:00	368	15.285	373 (Facility 58)
2020-10-14:11:00	376	15.168	373 (Facility 58)
2020-10-17:06:00	373	15.287	370 (Facility 58)

Table 7: Far-end Facility Search statistics

Traceroute File	Unique Facilities	Avg. IPs / Facility	Max IPs mapped to Facility
2020-09-14:11:00	328	7.17	121 (Facility 58)
2020-10-14:11:00	342	6.97	123 (Facility 58)
2020-10-17:06:00	341	7.04	121 (Facility 58)

To finish the results of Phase 2, a map is created showing the locations of the constrained facilities, and the links identified between them. This map is shown in Figure 8.

5.3 Facility link Monitoring Results

The results for the facility link monitoring are presented in this section. Three different sections are presented. The first section discussed the statistics of the alarm detection implementation, showing the duration of both forwarding and RTT alarms and the top ranking alarms. The second section presents some interesting patterns between observed facilities. Lastly, the third section presents patterns that signify routing changes in colocation facilities.

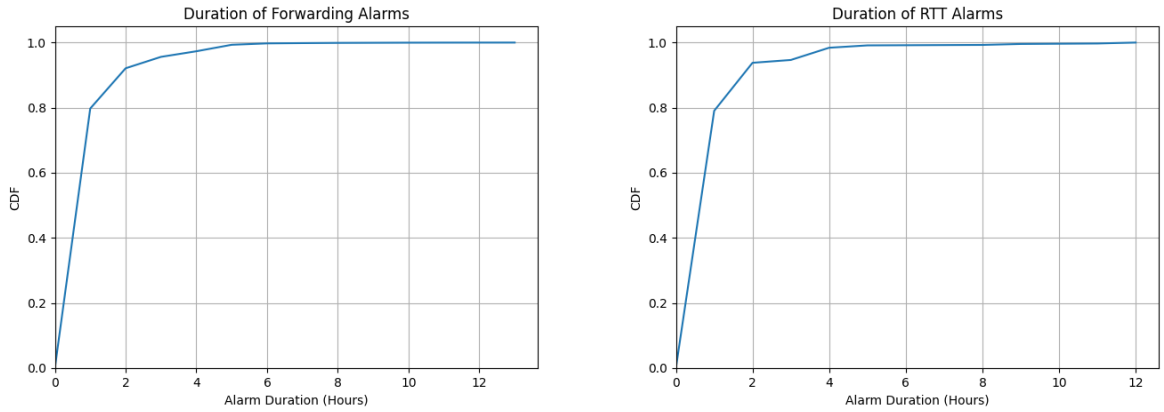
5.3.1 Anomaly Detection Results

Within the time period monitored of two months, from 26th of October to the 26th of December 2020, 241 links were reported anomalous at least once out of 272 total links. In total, the RTT module reported 959 alarms accounting for 692 anomalous patterns. Meanwhile, the forwarding model module reported 8025 "red alarms", and 9595 "yellow alarms" that account for 5875 anomalous patterns overall.

Alarm Durations

The alarm duration statistic distribution of the aforementioned reported alarms is shown in Figure 17a for forwarding alarms and Figure 17b for delay alarms.

Figures 17a and 17b show the cumulative distribution function, of the observed alarm duration distribution, which indicates the probability of the duration being lower or equal to a certain value. As expected in this kind of function, when approaching higher values, the probability will increase until it reaches 100%. In this case, for the forwarding model alarms, only the ones recorded as red alarms are taken into account.



(a) Forwarding Alarm Duration

(b) RTT_{Diff} Alarm Duration

Figure 17: The duration of alarms of detection links

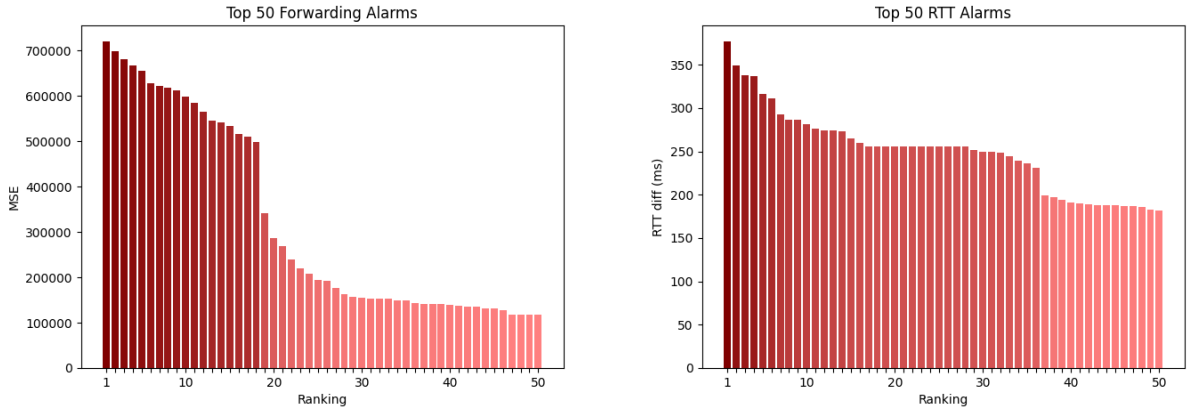
Looking at both figures, the tendencies are clearly similar, and the only major difference is that for one hour or less, the RTT delay alarms have a higher probability, 82.5% approximately, while the forwarding model alarms are at 75%. That can be partly because there is many more forwarding alarms than RTT delay alarms, as well as the RTT diff being usually much more stable than the forwarding models. Meanwhile, when it comes to larger duration alarms, the probabilities stay more or less the same, except for the fact that in the forwarding models a few larger alarms are reported (up to 14h versus 12h for the RTT).

Although these figures reflect the duration of the reported alarms during the observation time, these might not be completely accurate due to some limitations of the methodology used to do the detection. The alarm detection of the forwarding model is vulnerable in situations like the one shown in Figure 30 where the consecutive alarms have the same tendency i.e between 14th and 17th of November in the aforementioned figure. For these cases, the reported alarms will not last more than five hours, because from this point, the median is computed using more anomalous values than normal values (5 or more are anomalous and 4 or less will be normal), meaning that the reference will start to reflect the evolution of a new normal pattern (routing change). This leads to inconsistencies, because if the pattern returns to the previous normality, the change will again be reported as anomalous when in fact it is not.

Using a longer period of time to calculate the reference would allow larger duration alarms but at the same time would be worse in cases where an actual routing change happened, because the change would be reported as anomalous for a longer period of time. Finally, the decision of using the previous nine hours as a reference was based on the fact that these type of anomalies are not as common (Figure 17a), so shorter and bigger anomalies were prioritized when fixing the values for the methodology. This is why all the alarms that report more than 5 hours are caused by links that have constant changes in their pattern like the one shown in Figure **??** and are much more uncommon, hence the lower percentage for these larger duration alarms.

Ranking Forwarding and RTT Alarms

Figures 18a and 18b show the top 50 most important alarms of the link monitoring, for both the forwarding model and RTT module. These are the biggest alarms recorded during the two month period of observation, to rank them, different values have been taken into account for each one of them.



(a) Forwarding Alarms Ranking

(b) RTT_{Diff} Alarm Ranking

Figure 18: Top 50 Alarms

For the forwarding model, the metric used to rank the alarms is the forwarding model MSE, that as mentioned in previous sections, considers all the links involved with the same near-end facility. This way the ranking is not about individual link alarms, but about the alarms that affected most the set of links connected to a near-end facility as a whole. Observing in Figure 18a, there are several alarms that stand out and lay inside the top 20, after that the magnitude decreases considerably. From the top 20 alarms, all of them except one are reported in the intralink in facility "Equinix DC1-DC15 - Ashburn", including the top 5 alarms. This link is discussed in detail in the following

section. Meanwhile, the only alarm from a different link is the top 6 and was recorded in the interlink from "Digital Realty NYC (111 8th Ave)" to "165 Halsey Meet-Me Room" on the 5th of December with a MSE of 627463.

Meanwhile, for the RTT delay alarm ranking, the RTT difference is used to compare the alarms. In this case the top alarms do not stand out as much as for the forwarding model, but four that have higher magnitude than the rest. All top four alarms were recorded in links directed to "Equinix AM1/AM2 - Amsterdam", the first comes from "Interxion Amsterdam Science Park AMS9", the second and third ones come from "Equinix AM7 - Amsterdam" and the fourth comes from "ITENOS Frankfurt (FRA1)". As shown in Figures 19, 20 and 21, the values alarmed in these cases are due to very unstable links that always have a very large RTT difference and makes it difficult to know if these alarms are true anomalous patterns. On the other side, an alarm reported in a very stable link is the top eleven, from "Serverius DC1" and "NIKHEF Amsterdam" shown in Figure 23 which shows a very clear anomaly.

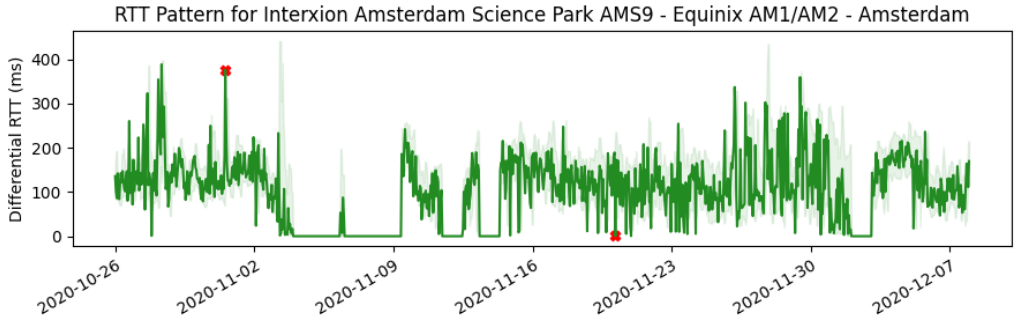


Figure 19: Facility AMS9-AM1/AM2 RTT Pattern

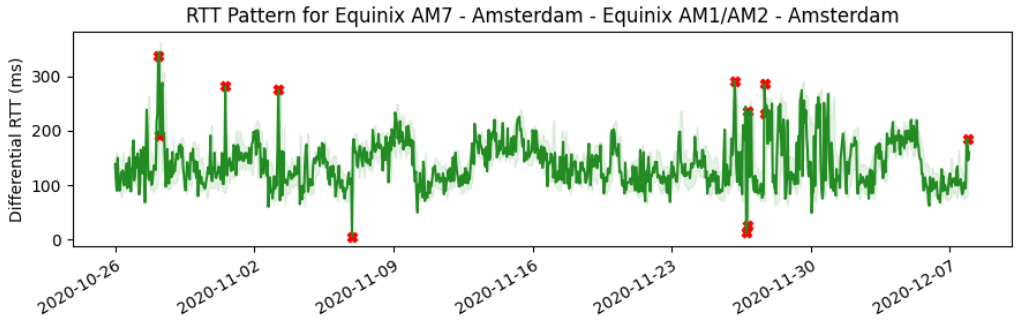


Figure 20: Facility AM7-AM1/AM2 RTT Pattern

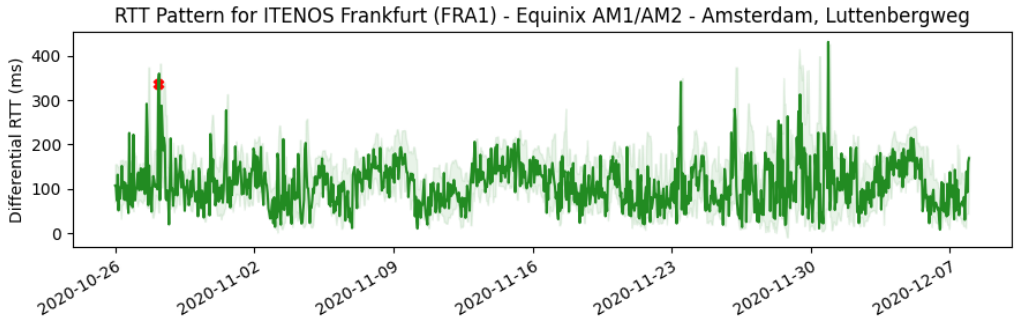


Figure 21: Facility FRA1-AM1/AM2 RTT Pattern

Top Forwarding Alarm

Several of the top alarms reported using the forwarding model correspond to the intra-link (the near-end facility is also the far-end) in facility Equinix DC1-DC15 - Ashburn. In Figure 22 these reported alarms can be seen to the right side of the image just after the 14th of December, around that time there is a big dip that lasts for several days, causing that when the pattern goes back to normal, a series of false alarms are also reported as the anomalous pattern. This happened because the anomalous pattern lasted a long time, and it was considered to be the new normal when it was not. The drop this alarms report accounts for about 900 traceroutes approximately, which explains why the MSE of the forwarding model is so high. Another cause of this large MSE is the fact that only one link is observed for this facility, so no other links will be included in the mean calculation, which in some cases can lower the resulting MSE. For example:

One link MSE for large alarm with observed value of 200 traceroutes and expected value of 50 traceroutes:

$$MSE = (200 - 50)^2 = 22.500$$

Now if another link was observed and had an observation of 300 and expected value of 295:

$$MSE = \frac{1}{2}[(200 - 50)^2 + (300 - 295)^2] = 11.262,5$$

Apart from these, other major alarms are reported in this link during the observation period, the largest of them being the one observed the 20th of November that lasted around 2 hours and was of similar scale as the top alarms. Another interesting observation is that the three drops observed have similar time spacing, which could be attributed to a routine maintenance break, although this is just a hypothesis that cannot be verified.

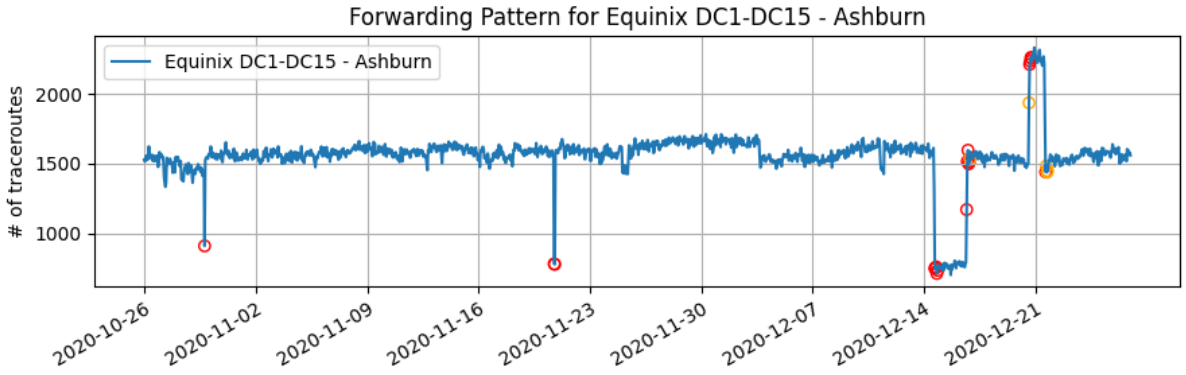


Figure 22: Facility Equinix DC1-DC15 Top Forwarding Alarm

Top RTT Alarm

The alarm with the highest differential RTT corresponds to facility "Interxion Amsterdam Science Park AMS9" that connects to facility "Equinix AM1/AM2 - Amsterdam, Luttenbergweg". Both of the facilities are located in Amsterdam, Netherlands. Meanwhile, one of the clearest anomalies is observed according to the pattern that can be seen in Figure 23. It affected the link between "Serverius DC1" and "NIKHEF Amsterdam", both in Netherlands. By looking to Figure 23, it clearly shows the anomaly detected when the differential RTT at the end of November 12th is hugely increasing up to 250 ms before it drops back till the beginning of November 13th. It can be seen that the usual delay is very close to 0 ms as these two facilities are both located in the same city, Amsterdam. In comparison, this huge spike triggers the alarms which can be shown by the scatter dots on the top of the spike.

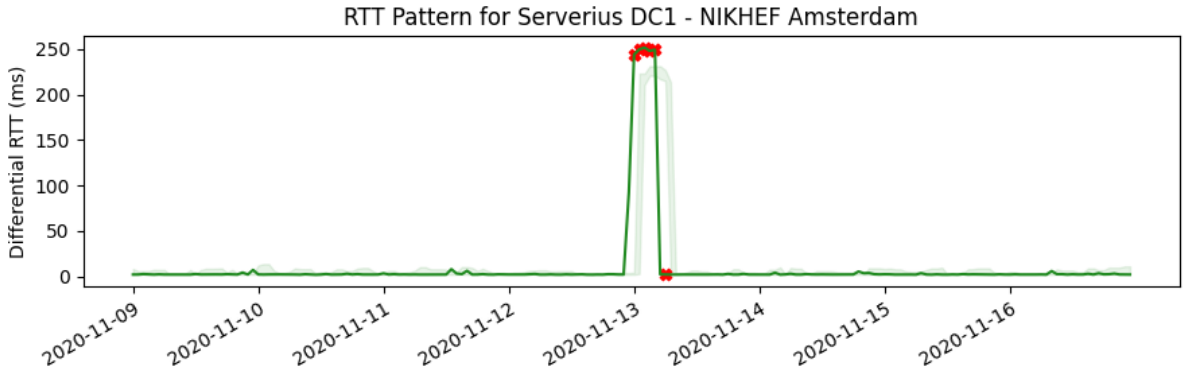


Figure 23: Example of RTT Anomaly

Locations of Alarms

In Tables 8 and 9 the results show the locations of the observed alarms in colocation facilities. For the forwarding alarms, 8 out of 10 cities where alarms were observed are located within Europe, while the other two are in Russia and North America. A similar pattern can be seen for the RTT alarms, in which 6 cities are located in Europe and 4 cities in North America.

Table 8: RTT Alarm Top cities

Cities	Frankfurt	Paris	Amsterdam	Berlin	Hamburg
Alarms	*192*	*175*	*113*	*105*	*82*
Cities	Lyon	Stockholm	London	Moscow	Newark
Alarms	*66*	*47*	*44*	*33*	*26*

Table 9: FW Alarm Top cities

Cities	Frankfurt	Amsterdam	London	Zurich	San Jose
Alarms	*1607*	*1304*	*879*	*493*	*455*
Cities	Newark	Hamburg	Los Angeles	Stockholm	New York
Alarms	*436*	*416*	*311*	*261*	*257*

To put the results into perspective, in Table 10 the cities with most colocation facilities are presented, from these it is clear that the distribution of facilities throughout the world is mostly concentrated in central Europe and North America, hence the results that can be observed in Tables 8 and 9 that show the top-10 cities where most alarms were recorded. The location information of the facilities is obtained from one of the datasets that PeeringDB has available, that enabled an easy location lookup for all the alarms and facilities observed. For the Forwarding Model module, all alarms are into account, both red alarms and yellow alarms.

Table 10: Top cities with most facilities

Cities	Singapore	London	Amsterdam	Paris	Los Angeles
Facs	*42*	*37*	*36*	*31*	*30*
Cities	Frankfurt	Moscow	Ashburn	Santa Clara	New York
Facs	*29*	*29*	*28*	*28*	*27*

This distribution is also observed in Figure 24 which shows the locations of RIPE Atlas probes around the world. As these are the probes conducting the measurements towards root DNS servers, the observation of colocation facilities is higher in Europe and North America.

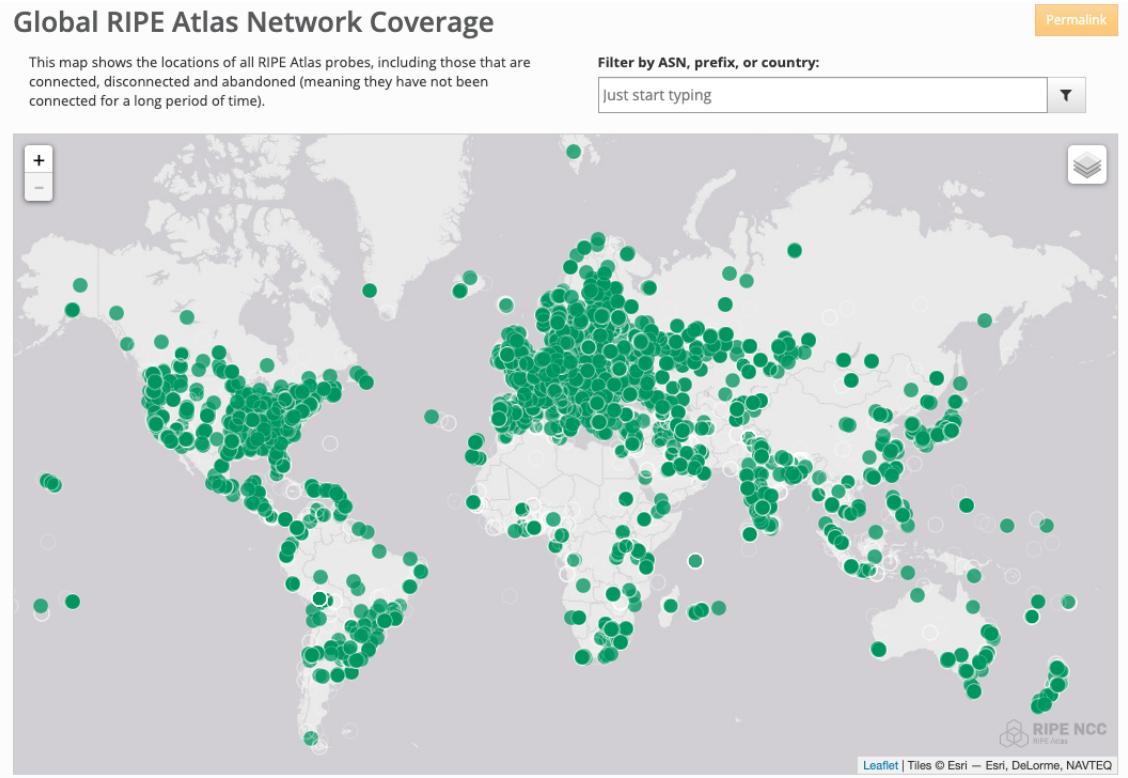


Figure 24: Locations of RIPE Atlas Probes [4]

5.3.2 Interesting observations

In this section, some interesting observations will be presented, beginning with examples of stable and unstable forwarding patterns. This is followed by presenting a possible outage in the links to a colocation facility located in Amsterdam in which the RTT pattern was affected in multiple links. A facility in Frankfurt is also presented in which the forwarding pattern was affected in links from

multiple near-end facilities. Lastly, instances of a forwarding and rtt change of pattern is shown in facilities in Amsterdam, London and New York.

Stable Pattern

Figure 25 shows an example of a stable forwarding and RTT pattern. An average of 1400 traceroutes utilise the link on an hourly basis. Similarly, the RTT pattern is unchanging with a difference of at most 0.1ms. A stable forwarding pattern in this case is one in which the number of traceroutes remains stable over a large period of time. This has the effect of making the reference calculation easier, which in turn makes the alarm detection more reliable. If an alarm were to happen in this link, it would be clear to see how it affected the normal pattern. A stable RTT pattern leads to the same conclusion, especially as this is a link within a singular colocation facility, it is expected to see the RTT very low. Similar to a having a stable forwarding pattern, calculating the references for this RTT pattern is more reliable, and alarms that occur would be easier to make out.

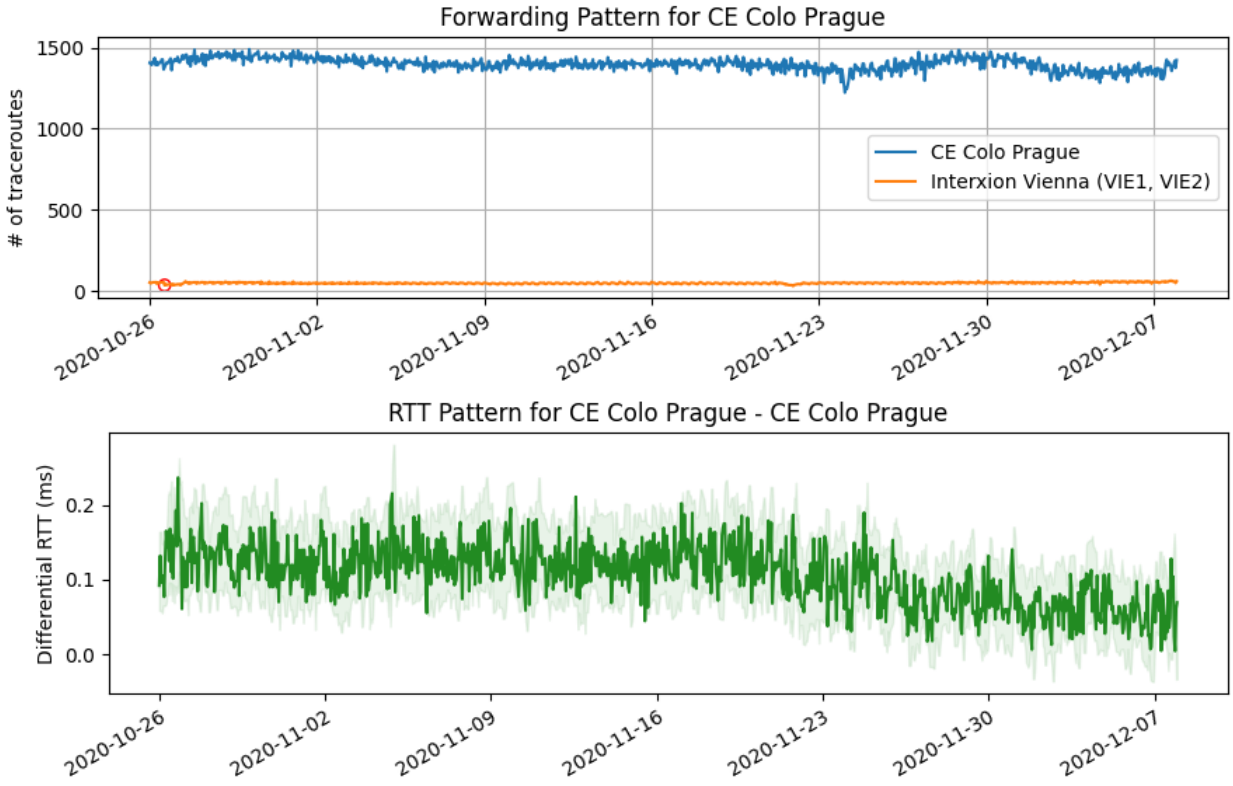


Figure 25: Stable Forwarding Model and RTT Pattern

Unstable Pattern

Figure 26 shows an example of an unstable forwarding pattern situated in the intra-link of colocation facility Interxion Frankfurt (FRA1-15). The number of traceroutes varies heavily, ranging from 500 up to 2000 on an hourly basis. This unstable pattern affects the reference calculation immensely, which in turn can affect the alarm detection. This can be seen in this figure, in which there are many alarms reported. A potential cause of this issue could be load balancing in the router, which causes the fluctuation of the number of traceroutes observed. Load balancing can occur when the access router in question has multiple links to send its packets to, which could explain the observation of one hour of measurements having roughly 500 traceroutes to another hour with 2000.

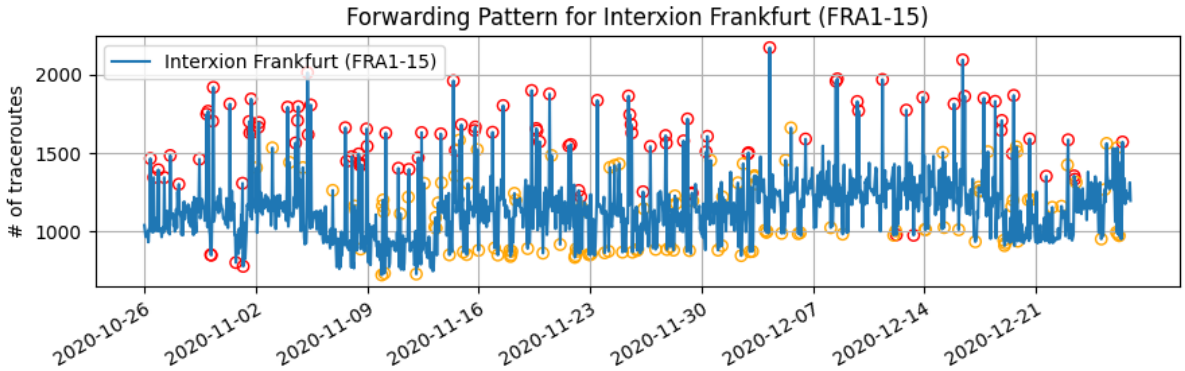


Figure 26: Unstable Forwarding Pattern

Links to NIKHEF Amsterdam

An interesting observation for anomalies detected with RTT delay monitoring method is presented in Figure 27 below. This figure shows three RTT patterns, and far-end facility for all these patterns is the facility NIKHEF in Amsterdam. The first subfigure is the RTT pattern of the link between the NIKHEF facility to itself. Second subfigure is of the link between Equinix AM7-Amsterdam, Kuiperberweg and the NIKHEF facility, whereas the third one is between Serverius DC1 facility in Netherlands and the NIKHEF facility. RTT pattern presented in Figure 27 is for the duration of a week from 09/11/2020 to 16/11/2020. The green colored plot on these subfigures represents the differential RTTs of the link across the monitoring period. Anomalies are marked using red colored cross-marks as seen in the figure. It is interesting to observe the anomaly (represented by the spike in differential RTT values) that occurred during the late hours on 12/11/2020 which lasted till the early hours on 13/11/2020, in all the three subfigures.

From Figure 27, it can be observed that anomalies detected for all the three links shows a huge variation in differential RTT values compared to the normal values. Since the far-end facility for all three links is the same, it can be assumed that the cause for these anomalies reported could be the possible consequence of a disruption that happened at the NIKHEF facility at that time. It is also important to note that this disruption affected almost all the links which are connected to this facility.

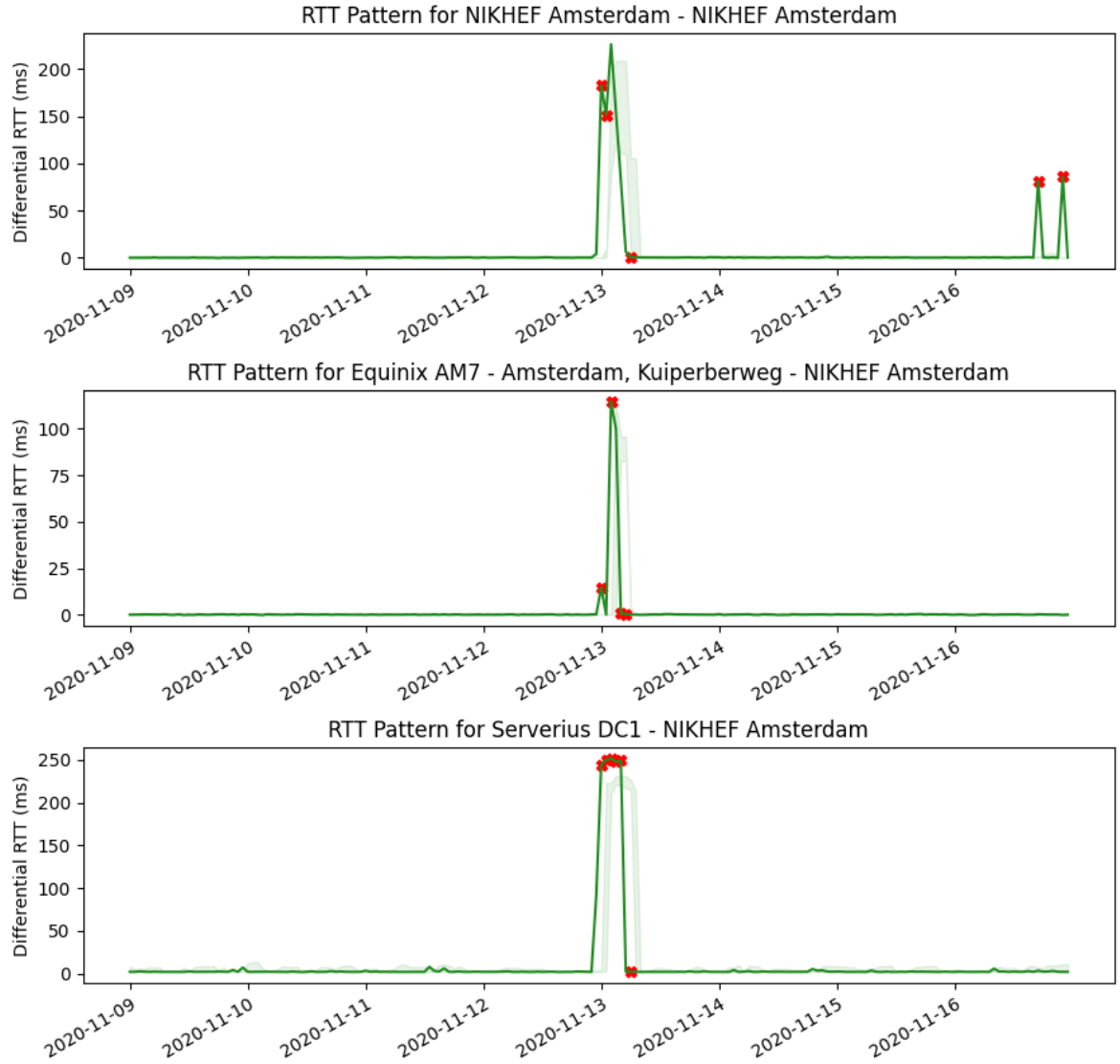


Figure 27: NIKHEF Amsterdam RTT Alarms

Links to Equinix FR6 - Frankfurt West

On the 29th October and 24th November a clear drop in the forwarding pattern was observed in several links towards the Equinix FR6 Colo located in West Frankfurt. In Figure 28, multiple links towards facility Equinix FR6 - Frankfurt West were disrupted on both the 29th of October and 24th of November. It can be theorised that these disruptions could be part of a monthly maintenance cycle. Furthermore, in this case, this facility is the far end-facility which is being impacted. It is interesting to observe that this facility is never identified as a near-end facility. This means that traceroutes never go in the other direction, all connections observed are towards this facility. It could be therefore hypothesised that this facility is located near or close to the destination of the traceroutes, i.e, near the destination of a root DNS server.

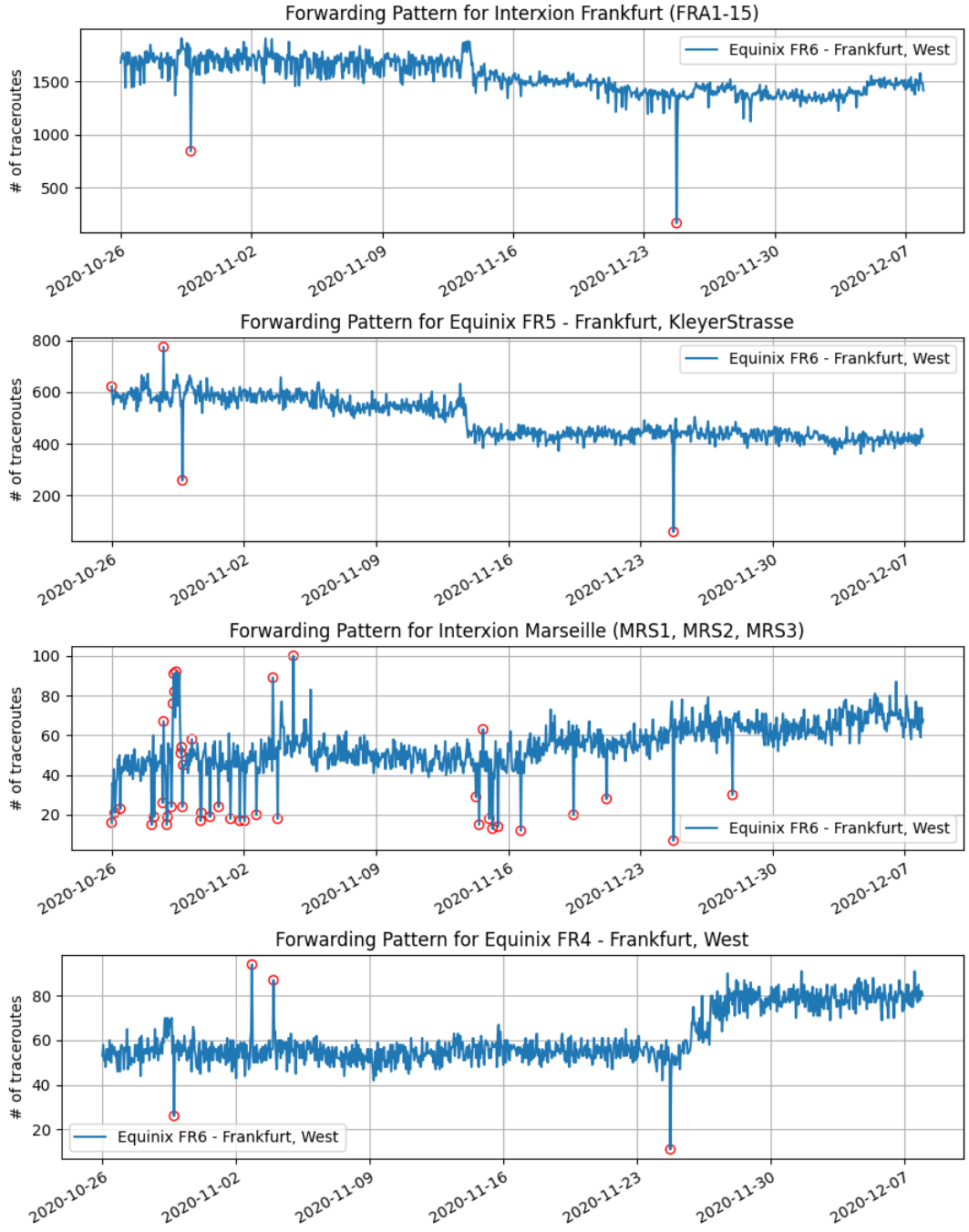


Figure 28: Links to Facility Equinix FR6 - Forwarding Alarms

RTT and Forwarding Model Affected (NIKHEF Amsterdam)

Figure 29 illustrates the comparison between forwarding and differential RTT pattern on facility "NIKHEF Amsterdam" that connects to itself. Within half of month period of monitoring, at least three groups of spikes are shown from both sub figures.

The interesting point to see in Figure 29 is the similar spikes occurred on the forwarding and RTT at the beginning of November 3rd and at the end of November 16th. Both groups of spikes trigger the alarms which indicate something has occurred in that link that affected to the both patterns. This could be firstly triggered by either the forwarding anomaly that affected to the RTT or in reverse. If there were too many packets routed between the link in this facility, thus congesting

the link, this could have potentially caused the RTT delays that were observed.

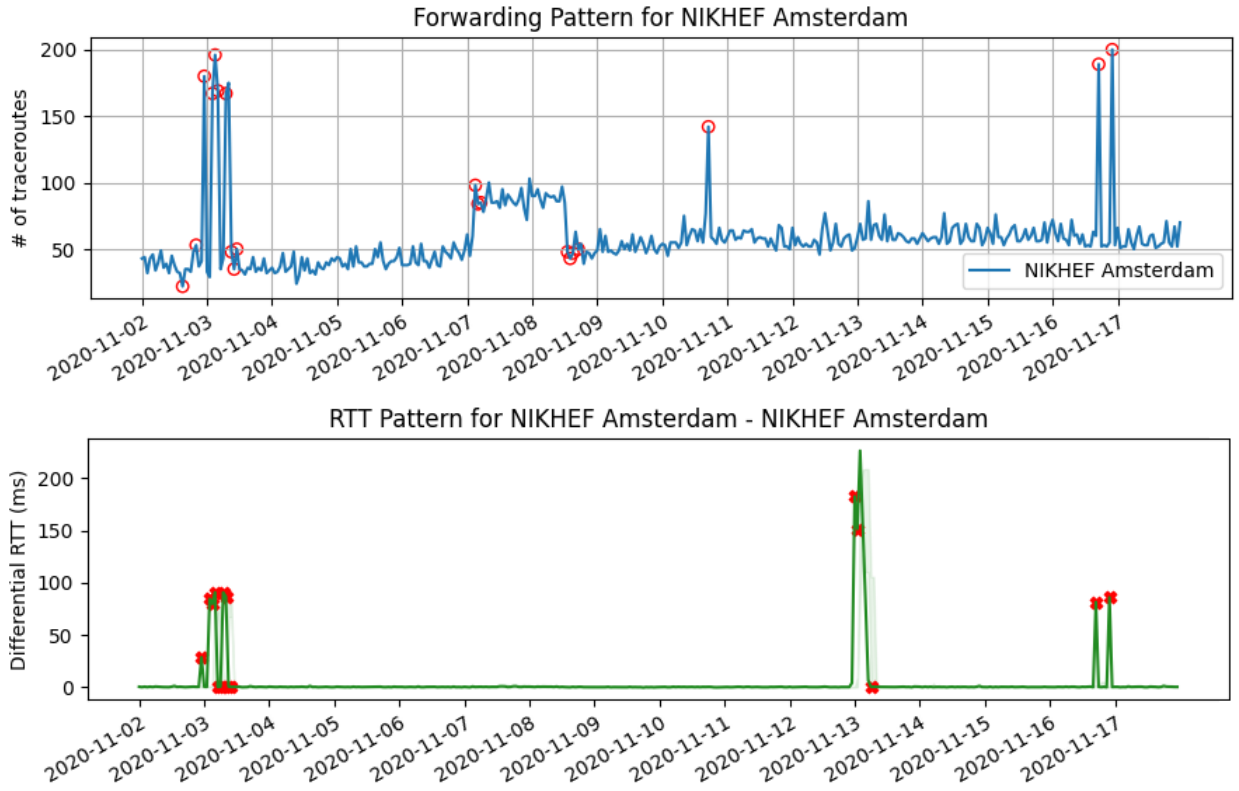


Figure 29: RTT and Forwarding Alarm on Facility NIKHEF Amsterdam

RTT and Forwarding pattern between Digital Realty LON (Cloud House) and Telehouse - London (Docklands East)

Figure 30 illustrates the RTT and forwarding pattern for the link between Digital Realty LON (Cloud House) facility and Telehouse - London (Docklands East) facility. The monitored period used to plot this figure is from 31/10/2020 to 19/11/2020. The first subfigure shows the forwarding pattern whereas the second subfigure shows the RTT pattern. An interesting observation on comparing these subfigures is that there is a spike in differential RTT values for the exact time periods when the number of traceroutes show a dip. However, there are not many alarms reported.

From Figure 30, for the the first dip seen in forwarding pattern which started during the late hours of 01/11/2020 and extended for almost a day, we see a corresponding increase in differential RTT values across the same period. The same pattern can be observed even for the second dip in forwarding pattern which started during the early hours of 14/11/2020 and extended for almost three days. However these deviations were quite small and did not contribute to many alarms. But it is quite interesting to observe the correlation between forwarding and RTT patterns in this case.

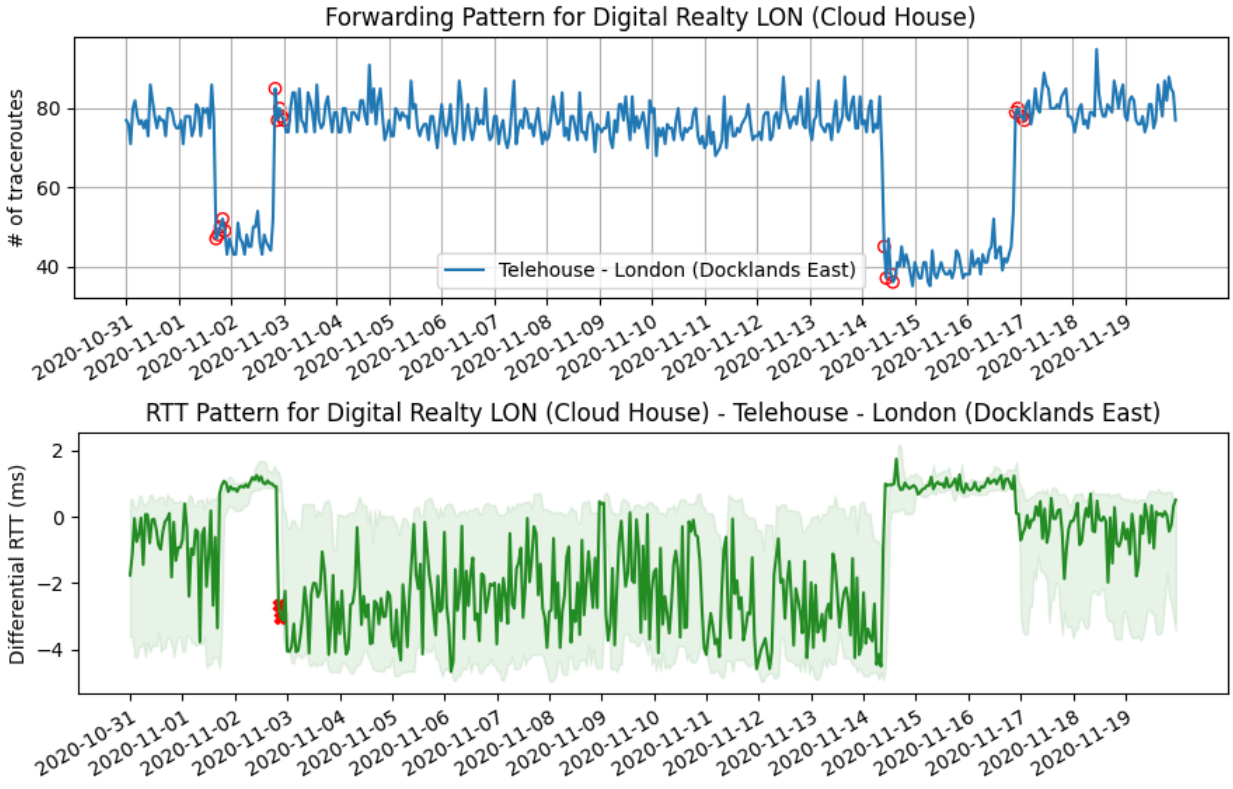


Figure 30: RTT and Forwarding pattern between Digital Realty LON (Cloud House) and Telehouse - London (Docklands East)

Link from Digital Realty NYC (111 8th Ave) to 165 Halsey Meet-Me Room

Figure 31 shows the forwarding patterns of the Digital Reality NYC (111 8th Ave) facility, in which the interesting link is the one connected to 165 Halsey Meet-Me Room. The blue line shows some drops in the forwarding pattern across the period of one month. The more interesting result comes during the end of the monitoring period in which there is a large drop in the forwarding pattern on the 5th of December contributing to one of the top forwarding alarms, and there is a increase in the RTT as seen in Figure 32. The link is not utilised after this drop.

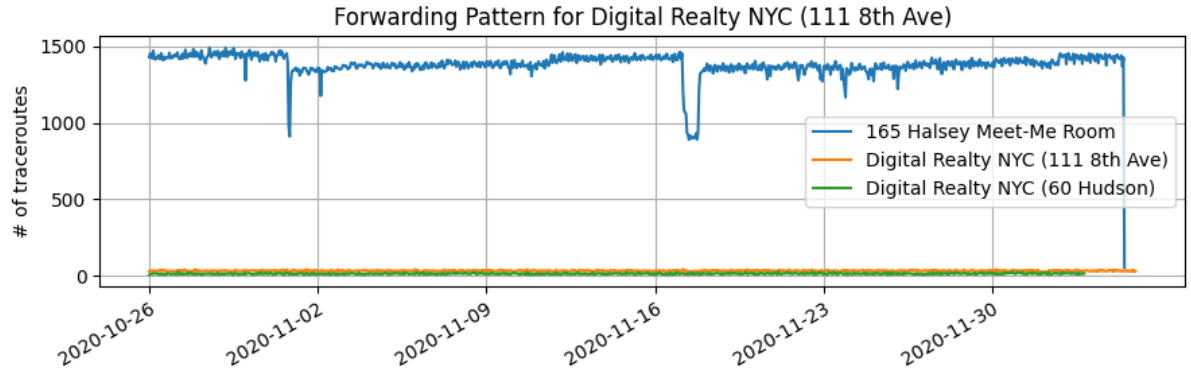


Figure 31: Facility Digital Reality NYC Forwarding Pattern

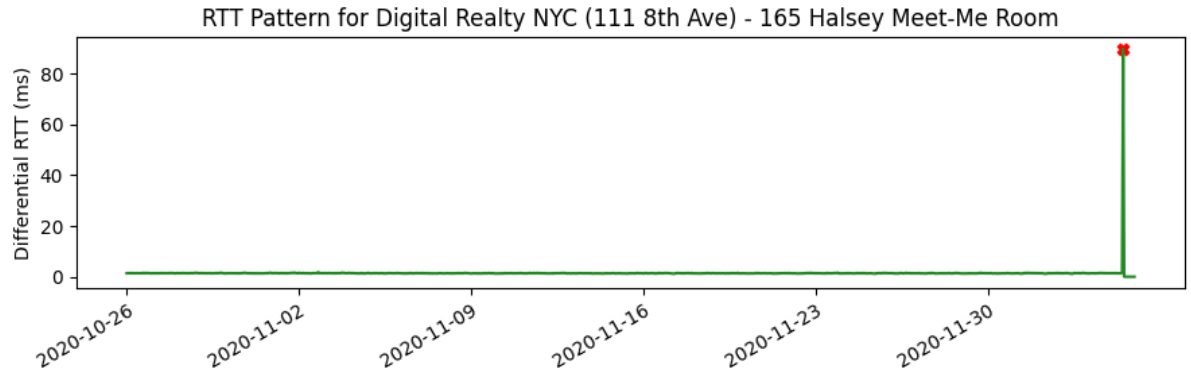


Figure 32: Facility Digital Reality NYC RTT Pattern

Matching Increases and Decreases in Forwarding Model

Figure 31 shows the forwarding patterns of the Westin Building Seattle facility, connected to itself and Equinix SE2/SE3. This is the pattern for traceroute data from the 26th of October until the 14th of November. In Figure 31 it can be seen that the forwarding pattern for a link can coincide with the forwarding pattern for another link. During certain days, increases in one pattern are matched with decreases in the other pattern. For example, this can be seen clearly on the 1st, 3rd and 7th of November. An explanation for this would be that packets simply utilise the other links available in the facility if their usual link is not available at the time.

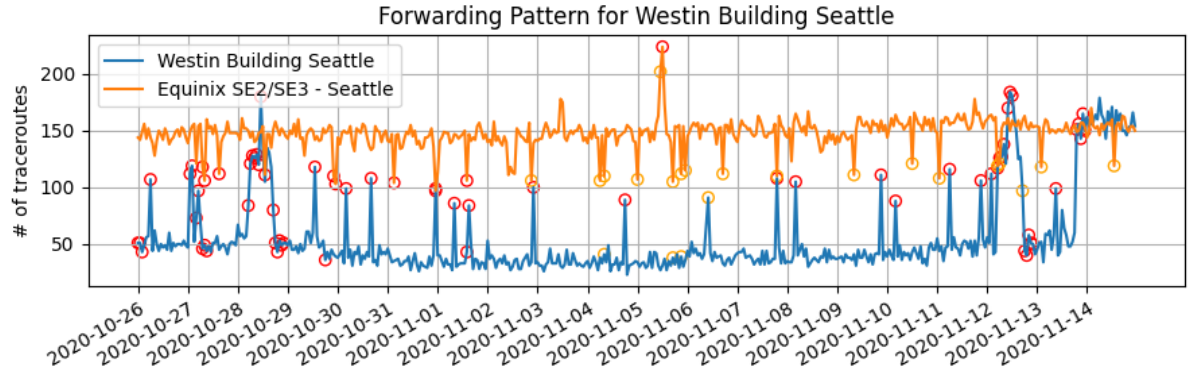


Figure 33: Facility Westin Building Seattle Forwarding Pattern

5.3.3 Routing changes

During the observed period, various links showed changes in the normal patterns of the forwarding model. These changes can be attributed to a possible routing change or configuration change within the equipment of the colocation facility. Some of the clearest routing changes were shown in facilities 5 (Equinix SV8 - Silicon Valley, Palo Alto), 62 (Equinix AM7 - Amsterdam, Kuiperberweg) and 71 (Westin Building Seattle). These routing changes are shown in Figures 34, 35 and 36. These figures show a forwarding pattern that is believed to be a routing change. This is because a stable pattern is observed for a significant duration of time, after which there is a sudden increase or decrease in the number of traceroutes observed. This new pattern is then stable for a similar, if not longer duration of time. These routing changes could have occurred due to an outage in the facility or a predetermined change with the IXP networks residing within the colocation facility.

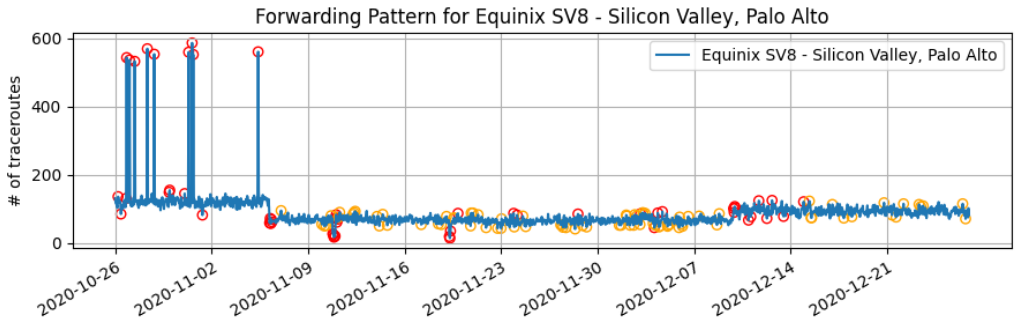


Figure 34: Facility Equinix SV8 Routing Change

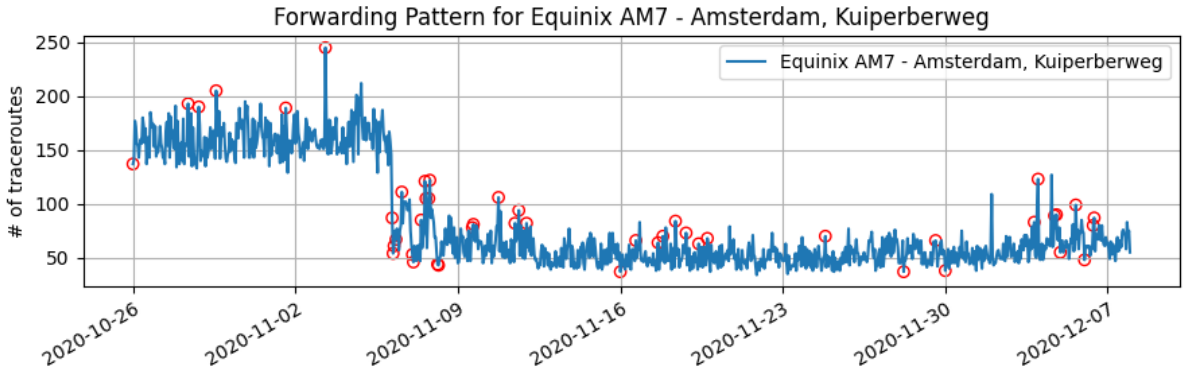


Figure 35: Facility AM7 Routing Change

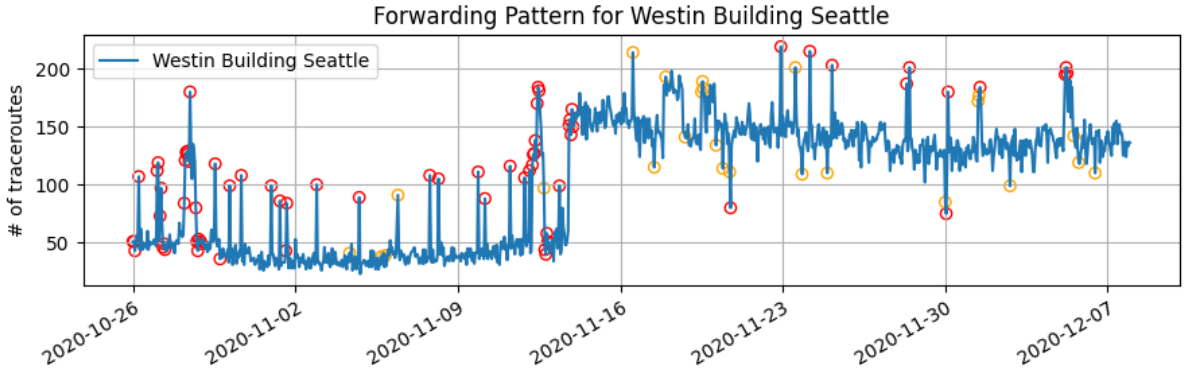


Figure 36: Facility Westin Building Seattle Routing Change

6 Conclusions

The conclusion of this project is stated in Section 6.1, followed by the limitations in Section 6.2. Finally, future work is stated in 6.3, and the individual contributions of the group is listed in Section 6.4.

6.1 Discussion and Conclusion

The goal of this project was to monitor the links between colocation facilities and report any observed anomalies, using publicly available traceroute information. The implementation consisted of three separate phases which allowed for combining different datasets to identify and constrain colocation facilities, as well as create a monitoring tool that allowed for detecting certain anomalous patterns in the traffic between colocation facilities. Several examples of anomalous patterns were shown and discussed.

The implementation works with traceroute data taken from the RIPE Atlas platform, and a decision was made to use the built-in measurements as they offered stability in the data extracted. In the beginning of the project, both built-in and user-defined measurements were used, which led to more traceroutes observed. This led to a certain instability in the data collected because there was no way to know what were the characteristics of the user defined measurements. Therefore, using the built-in measurements was necessary as they have known characteristics, which are that every probe executes every 30 min a traceroute measurement to DNS root servers. After the data was restricted to built-in measurements, fewer links were possible to observe as the number of measurements had decreased but the values were much more stable and less changing for the most part. This is discussed further in the limitations. The results obtained in this project are fully reproducible on any future traceroute data from the RIPE Atlas platform provided that the format of the dataset does not change.

Furthermore, the reproducibility aspect also applies to the data obtained from the PeeringDB and CAIDA databases. Obtaining this data was necessary to produce the results of this project, and if the format is not changed, more results are possible to obtain. A discussion can be made on how often the extracted datasets need to be updated. Looking on the front page of PeeringDB, Internet exchanges and colocation facilities can be subject to change on a daily basis, while networks (ASNs) can potentially change on an hourly basis. Based on this, it can be reasonable to update the datasets used in the constrained facility search on a weekly basis.

The number of links observed and where these are located geographically were presented in Section 5.1. It can be seen that there were more links observed in cities with a higher presence of both probes and colocation facilities where large networks and Internet exchanges interconnect. This is tied in with the fact discussed above in which the built-in measurements of the RIPE Atlas platform follow certain patterns and pass through certain central facilities more often than they pass through other remote facilities. For some intra- and inter-links connecting facilities, over thousands of traceroutes were seen on an hourly basis. On the other hand, certain links only registered fewer than 50 traceroutes per hour. This discrepancy affects both the alarm calculation and thereby increases the difficulty of monitoring the link for anomalies.

A result of a low link utilisation between facilities impacts the forwarding model as well. Due to the methodology removing the number of traceroutes if that number is below five, or if the number of unique RIPE Atlas probes is below four, this can result in it being not possible to observe the forwarding pattern. For example, if a link is stable with 100 traceroutes over a period of 1 month, and if during one hour the number of traceroutes drops to 3, it will not be possible to observe that

pattern for more than a period of one month.

This project has shown that it is possible to monitor the links between colocation facilities and thereby observe potential anomalies in the infrastructure on which the Internet is based on. There are some limitations to the work done which is discussed in the next section.

6.2 Limitations

After thorough evaluation, the following limitations of this project were identified, and are listed below.

1. **Real-time monitoring:** The anomaly detection tool created utilises traceroute data from the RIPE Atlas built-in measurements. These traceroute measurements are published on the RIPE Atlas database the next day after they are conducted. For example, the traceroute data from the 8th of December for the full day would be uploaded in the morning on the 9th of December. Due to this feature, it is not possible to build a live monitoring tool as the data is only available for the previous day.
2. **Performance of Scripts that Extract Data:** As mentioned in the previous limitation, data for one day of measurements is available online on the next day. The scripts that have been implemented in this project then analyse this data, which takes a certain amount of time that will be discussed here. In total, downloading (bzip2 file), extracting data and running scripts from Phase 1 and Phase 2 takes roughly 12 hours for one day of traceroute measurements. This is a further limitation when implementing a live monitoring service based on the VM that was used for the implementation.
3. **Identified Colocation Facilities:** The Colos identified are based on the CFS methodology and how the build-in measurements of the RIPE Atlas platform function. Based on this, links with low utilisation will not be identified, which leads to fewer Colos that can be monitored. This is one of the limitations of the project, in which not every colocation facility in the world can be monitored with this methodology.
4. **Limitations in the Forwarding Pattern:** Following the discussion about not observing forwarding patterns if the number of traceroutes are low, there is a limitation here in which it is not possible to observe patterns if there is a "gap" in the pattern, i.e, there is missing information in the recorded number of traceroutes.

6.3 Future work

Following the limitations of the project, the next step would be to create a live monitoring service. This would mean conducting or getting access to hourly traceroute information which is updated in real-time.

The project implementation currently relies only on the datasets provided by PeeringDB to retrieve information about IXPs. There are other datasets like CAIDA that combines information from different databases and finding out a way to merge the data from all these databases can help to improve the accuracy of the available data, for example the results of the CFS method. This can be taken up as an improvement in the future.

The alarms that were reported used needed a reference value computation to compare the observed values. This is something that was done without any machine learning or linear regression models. Analysing data with these more advanced models could contribute to an improvement of the

precision in the alarm detection, as it would be easier in differentiating a normal pattern with an abnormal pattern.

Regarding the interface of the project, the code involves running graphs for the period that is being monitored. As a map (Figure 8) was created as part of the dataset, it could be possible to work on the user interface of the results, and it would be possible to build an interactive map consisting of the identified facilities, along with associating the traceroute figures to each link that is visible on the map.

6.4 Individual Contributions

The different contributions of each members of the group are listed below. Communication with the group and the TA was held on Slack, and the internal group communication was handled through WhatsApp. The weekly reports and the division of work between team members was held through meetings on Zoom and using a shared Google Drive folder.

- Enric Carrera I Aguiar
  - Contributions in the report: Implementation(Phase 1,2 and 3), Methodology(Phase 1 and 2, 3 only for the Forwarding module), Analysis, Conclusions, Background.
  - Contributions with the code: All the scripts with the exception of the last versions of the RTT module.
- Lingfeng Cheng
  - Contributions in the report:
  - Contributions with the code:
- Mandar Joshi
  - Contributions in the report: Background (Related work), Methodology, Implementation (Phase 1, 2 and Forwarding Model in Phase 3), Results and Analysis, Conclusions.
  - Contributions with the code: All the scripts with the exception of the last versions of the RTT module.
- Anika Bintey Mansur
  - Contributions in the report: Background:Internet Exchange point, Colocation facilities, Facility link Monitoring results: Anomaly detections results
  - Contributions with the code: Phase 1 scripts: IXP and AS mapping, IXP and non IXP mapping, Phase 3 scripts: RTT delay monitoring
- Shubham Bhargava
  - Contributions in the report: Abstract, Introduction, Background(IXP, Colocation Facilities, Traceroutes, Databases), Limitations
  - Contributions with the code: Non-IXP IP info script, Non-IXP IP to ASN mapping scripts, IXP to ASN mapping script, RTT delay monitoring and Forwarding model scripts
- Seba Anna Varghese

  - Contributions in the report: Methodology and Implementation: Phase 2, Phase 3 (RTT Delay Monitoring), Analysis.
  - Contributions with the code: Phase 1: Extract Traceroutes, IXP hop filter, Phase 2: Far-end facility identification, Phase 3: RTT delay monitoring scripts.
- Sri Yulianti
  - Contributions in the report: Background : Colocation Facilities, Methodology and Implementation : Phase 2, Phase 3 (RTT Delay Monitoring), Analysis
  - Contributions with the code: Pre-implementation : Extract Traceroutes script, Phase 1: IXP hop filter script, Phase 2 : Script to identify Far-end Facilities, Phase 3 : RTT delay monitoring scripts.

References

[1] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a large european IXP," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, ser. SIGCOMM '12. Association for Computing Machinery. doi: 10.1145/2342356.2342393. ISBN 978-1-4503-1419-0 pp. 163–174. [Online]. Available: https://doi.org/10.1145/2342356.2342393

[2] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and k. claffy, "Mapping peering interconnections to a facility," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '15. Association for Computing Machinery. doi: 10.1145/2716281.2836122. ISBN 978-1-4503-3412-9 pp. 1–13. [Online]. Available: https://doi.org/10.1145/2716281.2836122

[3] A. Milolidakis, R. Fontugne, and X. Dimitropoulos, "Detecting network disruptions at colocation facilities," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. doi: 10.1109/INFOCOM.2019.8737615 pp. 2161–2169, ISSN: 2641-9874.

[4] RIPE atlas. [Online]. Available: https://www.ripe.net/analyse/internet-measurements/atlas

[5] Understanding internet exchanges via the de-cix outage. [Online]. Available: https://blog.thousandeyes.com/network-monitoring-de-cix-outage/

[6] PeeringDB. [Online]. Available: https://peeringdb.com/

[7] CAIDA: Center for applied internet data analysis. [Online]. Available: https://www.caida.org/home/index.xml

[8] Responsible consumption and production. [Online]. Available: https://www.un.org/development/desa/disabilities/envision2030-goal12.html

[9] Z. Mi and D. Coffman, "The sharing economy promotes sustainable societies," vol. 10, no. 1, p. 1214. doi: 10.1038/s41467-019-09260-4 Number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-019-09260-4

[10] Climate action. [Online]. Available: https://www.un.org/development/desa/disabilities/envision2030-goal13.html

[11] Sto6 - the optimal data center for sustainable and secure digitization. [Online]. Available: https://www.interxion.com/se/vara-datacenter/europa/stockholm-campus/sto6?gclid=CjwKCAjwrKr8BRB_EiwA7eFapuNFivMfO0ZGnToieShp_VjEtAGZT9lOIByYJXA2hdf79wi3y6_sgRoC4xQQAvD_BwE

[12] Routeviews prefix to AS mappings dataset (pfx2as) for IPv4 and IPv6. [Online]. Available: https://www.caida.org/data/routing/routeviews-prefix2as.xml

[13] Understanding denial-of-service attacks | CISA. [Online]. Available: https://us-cert.cisa.gov/ncas/tips/ST04-015

[14] What is a BGP hijacking? [Online]. Available: https://www.netscout.com/what-is-ddos/bgp-hijacking

[15] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, ser. IMC '06. Association

for Computing Machinery. doi: 10.1145/1177080.1177100. ISBN 978-1-59593-561-8 pp. 153–158. [Online]. Available: https://doi.org/10.1145/1177080.1177100

[16] V. Giotsas, C. Dietzel, G. Smaragdakis, A. Feldmann, A. Berger, and E. Aben, "Detecting peering infrastructure outages in the wild," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. Association for Computing Machinery. doi: 10.1145/3098822.3098855. ISBN 978-1-4503-4653-5 pp. 446–459. [Online]. Available: https://doi.org/10.1145/3098822.3098855

[17] R. Fontugne, C. Pelsser, E. Aben, and R. Bush, "Pinpointing delay and forwarding anomalies using large-scale traceroute measurements," in *Proceedings of the 2017 Internet Measurement Conference*, ser. IMC '17. Association for Computing Machinery. doi: 10.1145/3131365.3131384. ISBN 978-1-4503-5118-8 pp. 15–28. [Online]. Available: https://doi.org/10.1145/3131365.3131384

[18] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. Association for Computing Machinery. doi: 10.1145/863955.863996. ISBN 978-1-58113-735-4 pp. 365–378. [Online]. Available: https://doi.org/10.1145/863955.863996

[19] R. Klöti, B. Ager, V. Kotronis, G. Nomikos, and X. Dimitropoulos, "A comparative look into public IXP datasets," vol. 46, no. 1, pp. 21–29. doi: 10.1145/2875951.2875955. [Online]. Available: https://dl.acm.org/doi/10.1145/2875951.2875955

[20] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger, "There is more to IXPs than meets the eye," vol. 43, no. 5, pp. 19–28. doi: 10.1145/2541468.2541473. [Online]. Available: https://doi.org/10.1145/2541468.2541473

[21] Internet outages worldwide: info about outage, service down or problems. [Online]. Available: https://app.fing.com/internet/outages

[22] Internet outages map. [Online]. Available: https://www.thousandeyes.com/outages/

[23] Internet health report. [Online]. Available: https://ihr.iijlab.net/ihr/en-us/

[24] Internet disruption report. [Online]. Available: https://internetdisruption.report/

[25] "tqdm/tqdm," original-date: 2015-06-03T13:13:14Z. [Online]. Available: https://github.com/tqdm/tqdm

[26] "ultrajson/ultrajson," original-date: 2011-02-27T20:00:51Z. [Online]. Available: https://github.com/ultrajson/ultrajson

[27] "networkx/networkx," original-date: 2010-09-06T00:53:44Z. [Online]. Available: https://github.com/networkx/networkx

[28] "matplotlib/matplotlib," original-date: 2011-02-19T03:17:12Z. [Online]. Available: https://github.com/matplotlib/matplotlib

[29] J. Sommers, "jsommers/pytricia," original-date: 2012-08-14T19:52:18Z. [Online]. Available: https://github.com/jsommers/pytricia

[30] Matthias Wichtlhuber. Advanced blackholing at internet exchange points. [Online]. Available: https://blog.apnic.net/2019/07/26/advanced-blackholing-at-internet-exchange-points/

[31] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[32] "Remote peering," Oct. 2020, page Version ID: 982911697. [Online]. Available: https://www.franceix.net/en/solutions/remote-peering/

[33] "netaddr/netaddr," original-date: 2010-05-24T23:14:56Z. [Online]. Available: https://github.com/netaddr/netaddr

[34] D. Freedman, B. Foust, B. Greene, B. Maddison, A. Robachevsky, J. Snijders, and S. Steffann, "Mutually agreed norms for routing security (manrs) implementation guide," 2019.

[35] How to secure routing in the ixp route servers infrastructure. [Online]. Available: https://www.manrs.org/2020/08/how-to-secure-routing-in-the-ixp-route-servers-infrastructure/

[36] Manrs IXP programme. [Online]. Available: https://www.manrs.org/ixps/

A Appendix

All code is available on the GitHub repository for this project: https://gits-15.sys.kth.se/enriccia/IK2200HT201-IXP