Janneth Malibago

Automated Monitoring of Internet Service Provider (ISP) Topologies

Master's Thesis MA-2006-11
August 3, 2006 / Summer Term 2006

Tutors: Roel Vandewall, Stefan Lampart (Open Systems AG)
ETH-Tutors: Daniela Brauckhoff, Bernhard Tellenbach
Supervisor: Prof. Dr. Bernhard Plattner

Abstract

Nowadays, manual debugging is the standard solution for solving Internet connectivity problems. This is a time consuming and repetitive task. Based on the observation that many performance problems are correlated with changes in the end-to-end routing path, the goal of this Master's thesis is to implement a tool for monitoring and analyzing route changes.

During this thesis, a prototype for monitoring end-to-end routing paths was developed. The prototype analyzes route changes based on periodically collected traceroute data and generates daily summary tables which provide a visualization of change patterns in Internet end-to-end paths. A distinction is made between IP-level and AS-level routes as well as between significant and insignificant route changes. For example, route changes resulting in a different AS-level path are considered as significant route changes, whereas a route alternation between two routers in the same network is considered as an insignificant change.

In order to verify the functionality of the developed prototype, we deployed it on 21 PlanetLab nodes in Asia, Europe, and the US, and collected 3 months of measurement data. When analyzing this data, we found that for certain end-to-end paths, route changes highly correlate with performance changes. However, it was not possible to find a correlation in general.

Motivated by this observation, we developed a route quality measure that summarizes a route's quality based on the minimal observed RTT and the 0.95 quantile. With the resulting route quality measure, it can be explained why a correlation could be detected for some end-to-end paths and why not for others. Furthermore, we extended this measure to compare the quality of end-to-end paths in general. This allows us to monitor and assess the quality of different Internet Service Providers (ISPs).

Kurzfassung

Heutzutage ist manuelles Debugging die Standard Methode, um Internet-Verbindungsprobleme zu lösen. Dies ist eine zeitraubende und mühsame Aufgabe. Es hat sich gezeigt, dass viele Verbindungsprobleme durch Topologieänderungen in der End-zu-End Verbindung verursacht werden. Das Ziel dieser Master Arbeit ist es daher, ein Tool zu entwickeln, mit dem Topologieänderungen erkannt und analysiert werden können.

In dieser Arbeit wurde ein Prototyp entwickelt, der periodisch Traceroute Daten sammelt und analysiert. Der Tagesverlauf der Routenänderungen von Internetpfaden wird in einer zusammenfassenden Tabelle graphisch dargestellt. Dabei wird sowohl zwischen IP-level und AS-level Routen, als auch zwischen signifikanten und nicht signifikanten Routenänderungen unterschieden. Zum Beispiel werden Routenänderungen, die eine Änderung der AS-level Route zur Folge haben, als signifikant angesehen, wohingegen es eine insignifikante Änderung ist, wenn der End-zu-End Pfad zwischen zwei Routern desselben Netzwerks hin- und her wechselt.

Um die Funktionalität des Prototypen zu verifizieren, wurde er auf 21 PlanetLab Knoten in Asien, Europa und den USA installiert. Eine Analyse der über 3 Monate hinweg gesammelten Daten ergab, dass für gewisse Verbindungen ein grosser Zusammenhang zwischen Topologieänderungen und einer Veränderung in der Verbindungsqualität besteht. Es konnte jedoch kein allgemein gültiger Zusammenhang gefunden werden.

Dieses Resultat war die Motivation dafür, ein Mass zu entwickeln, mit dem die Qualität einer Route bestimmt werden kann, basierend auf der minimal gemessenen Round Trip Time (RTT) und dem 0.95 Quantil. Mit Hilfe dieses Qualitätsmasses konnte schließlich eine Erklärung gefunden werden, warum keine allgemein gültige Aussage über den Zusammenhang zwischen Topologieänderungen und einer Veränderung in der Verbindungsqualität ermittelt werden konnte. Eine Weiterentwicklung dieses Qualitätsmasses ermöglichte es uns ausserdem, die Qualität verschiedener Internet Service Provider (ISP) zu vergleichen.

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Business-critical applications like for example Voice over IP (VOIP) or Virtual Private Network (VPN) services need reliable Internet connections. Reachability problems and/or performance problems are directly reflected in the quality of the VOIP connection, or respectively in the quality of the VPN connection. The reason of such connectivity problems often lies in the Internet itself. VPN connections are tunneled through the public Internet and therefore they are very dependent on the quality of the underlying network. Thus, outages or routing problems of the involved ISPs can result in bad VPN connections. On the other hand, voice communication requires real time speed and thus is very sensitive to high latency/response time.

Nowadays, debugging what caused a connection problem needs manual investigation. This is a time consuming and repetitive task. It normally takes more than one hour work to find the cause for an unstable connection. But all too often manual investigation cannot provide any details about where and why the outage or routing problem occurred. The problem is that Internet services providers (ISPs) consider their routing policies and network topologies as confidential and thus little is known about how packets exactly travel through a particular network. Analyzing publicly available routing tables or using measurement tools like traceroute can only give a limited view of the network. Therefore, pinpointing where and why a problem occurred becomes a daunting task. However, these details can be of great help. In many cases, once an ISP has detailed information about an outage or routing problem the ISP is fairly quick in finding the cause and fixing it. Furthermore such information can help network administrators to compare the performance of local ISPs and to select good upstream providers.

Open Systems AG [51] has made the experience that many performance problems are correlated with changes in the end-to-end routing path. Related work [38] supports this assumption. Thus, in this Master's thesis we want to investigate the correlation of routing changes and performance changes. The idea is to periodically monitor end-to-end routing paths. Measuring only when problems occur often does not give enough information on the root cause. But with sufficient historical data, it will be possible to statistically analyze the quality of certain routing paths in terms of route stability and end-to-end performance. In this way periodical monitoring could greatly improve the debugging procedure as the problem resolution could already be started even before the user is experiencing connection problems. The monitoring system could compare the routing during a connection problem to well-performing routing setups previously seen for a specific route.

1.2 Implementation Environment

The thesis is conducted at Open Systems AG [51] which is based in Zurich, Switzerland. Open Systems AG is a company specialized in Internet Security since 1991 and currently offers "Mission Control" Security Solutions in over 70 countries. The provided services run on hardware which is set up by Open Systems engineers before it is sent to one of the worldwide distributed customer sites. A highly qualified expert team monitors and maintains the IT Security of Open Systems customer sites around-the-clock, 365 days a year. System outages or break-in attempts result in the creation of trouble tickets that are handled by the engineers. The customer gets notified depending on time or type of the incident.

At present Mission Control comprises ten different services one of which is the Mission Control Security Gateway. This service allows different customer sites to securely communicate with each other over large international VPN networks [21]. We call customer sites running this service *VPN sites*. A *VPN site* typically consists of a VPN gateway and about 10 to 200 hosts. Figure 1.1 shows an exemplary network setup with two customer VPN sites.

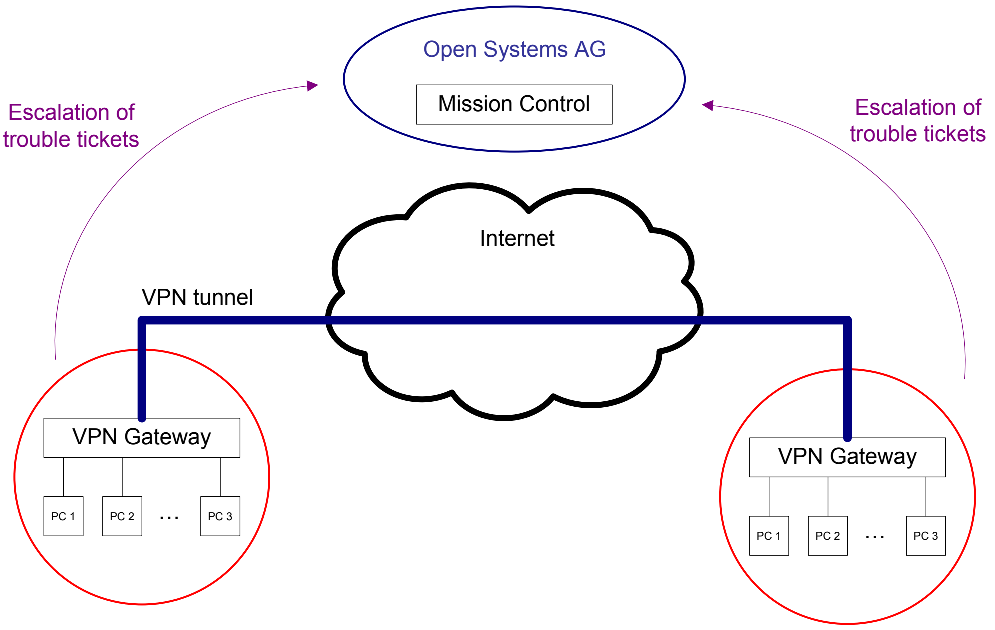


Figure 1.1: Network structure of Open Systems AG with two customer VPN sites

Each VPN gateway can communicate with all other VPN gateways of the same company, it knows the IP addresses of all gateways participating in the company VPN. The communication between the VPN gateways is encrypted whereas the communication within the VPN sites is not. The encryption is achieved using the Encapsulated Security Payload (ESP) [33] protocol. Every packet addressed to an external host must pass through the VPN gateway which encrypts all outgoing packets. Likewise, every packet arriving at the VPN gateway is decrypted and forwarded to the corresponding host. Packets with an unknown IP address or one which is not allowed are rejected, dropped or forwarded to a default route address, depending on the configuration settings.

1.3 Contribution

In this Master's thesis a framework has been implemented for visualizing and analyzing route changes in end-to-end paths. The framework has been used to investigate the correlation of route changes and performance changes. Manual observation of the measurement data could not reveal a correlation in general. Therefore, a measure for determining the quality of a specific route has been developed. With this measure it is possible to explain why a correlation could be detected for some end-to-end paths and why not for others. A similar measure was developed for determining the quality of end-to-end paths. This measure offers a method for comparing the quality of different Internet Service Providers (ISPs).

1.4 Chapter Overview

The thesis is structured as follows:

Background Chapter 2 provides background material on Internet topology and routing policy concepts.

Related Work Chapter 3 gives a short overview on related work.

Problem Discussion Chapter 4 discusses the problem situation and the requirements.

Own Approach Chapter 5 presents our own approach.

Implementation Chapter 6 presents the prototype implemented during this thesis.

Results Chapter 7 presents and evaluates the results.

Conclusion Chapter 8 concludes with a short summary and gives some ideas how the prototype could be improved further.

Chapter 2

Background

This chapter provides background material on Internet topology and routing policy concepts needed to understand the next chapters. First, basic terminology is defined in Sect. 2.1. Then, in Sect. 2.2, we examine the Internet topology. Finally, we give an overview of Internet routing policy concepts in Sect. 2.3.

2.1 Definitions

Before further investigating Internet topology and routing policy concepts, we first define some basic terminology in this section. The intention is to provide the reader with the fundamentals needed to understand the background material given in Sects. 2.2 and 2.3. We assume that the reader is already familiar with basic networking terminology like Internet, IP, router etc.

2.1.1 Topology

In this thesis we understand a *topology* as a graph consisting of a set of nodes and links connecting these nodes. The links may be unidirectional or bidirectional. The former are called *asymmetric* links, the latter are *symmetric* links.

2.1.2 Routing

Routing is the process of selecting a path, i.e. a set of links and nodes, from a source to a destination. In the Internet routing policies determine which paths are selected (refer to Sect. 2.3 for further details on routing policies and routing protocols).

2.1.3 IP Prefix

Throughout this thesis we express an *IP prefix* as a tuple comprising of two components: an IP-address and an IP-mask, e.g. *192.168.0.0/24*. In this example the IP-mask indicates that the prefix is 24 bits long and thus the first 24 bits of the 32-bit IP address are fixed. The remaining 8 bits (32-24=8) can be used for host addressing on that subnet. Thus, the IP prefix *192.168.0.0/24* corresponds to the IP range *192.168.0.0-192.168.0.255*. Note that 192.168.0.0/24 can also be expressed as (192.168.0.0, 255.255.255.0).

2.1.4 Routing Table

Network devices such as routers store a *routing table* which matches destination prefixes to a host IP address. The returned host IP address corresponds to the destination IP if the host is in the same subnet, otherwise it is the IP address of the next router the packet should be forwarded to. The longest prefix match [39] is used in case more than one prefix matches. The prefix 0.0.0.0/0 is the least-specific prefix possible and called the *default route* as it is used when no other entry matches. Note that the core of the Internet is "default-free", i.e. in the core of the Internet the default routes are not used to avoid routing loops due to aggregation [19, page 12].

In 1993 the Classless Inter-Domain Routing (CIDR) [56, 19] was introduced in order to face several serious scaling problems of the growing Internet like address space exhaustion and routing table overflow. CIDR allows the aggregation of routing information for different blocks of addresses into a single routing table entry and thus reduces the size of routing table entries. For example, the networks: 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 and 192.168.3.0/24 can be aggregated to the single prefix 192.168.0.0/22.

2.1.5 Autonomous System (AS)

Let us start with a very brief definition taken from RFC-1930 [26]:

> *"An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy."*

Autonomous Systems can be grouped into three categories [3]:

Multi-homed ASs : A multi-homed AS connects to more than one ISP and thus remains connected to the Internet even if the link to one if its ISPs has an outage.

Stub ASs : A stub AS connects only to a single ISP. Sometimes such an AS may in fact have private peering agreements with other autonomous systems which are not advertised to the global Internet and thus are not visible to other ASs.

Transit ASs : A transit AS provides connections through itself to its neighboring networks.

Every AS is associated with a unique *AS number (ASN)* which is assigned by the Internet Assigned Numbers Authority (IANA). The ASN is also used for the exchange of exterior routing information between neighboring ASs (see Subsect. 2.3.1).
AS numbers are divided into two ranges. The first range, from 1 to 64511, are public AS numbers, which may be used on the Internet. The second range, from 64512 to 65535, are reserved for private use and should not be advertised.

2.1.6 Internet Service Provider (ISP)

Internet service providers (ISPs) [30] offer Internet access and related services to their customers who generally have to pay a monthly access fee. Most of the ISPs are telephone companies. Considering the definition of the last section, ISPs are always transit ASs as they provide transit service to other networks.
An ISP houses a collection of routers in a so called *Point of Presence (POP)*. Usually different ISPs tend to have routers in the same building which is called an *Internet eXchange Point (IXP)*. IXPs allow Internet service providers to exchange traffic with each other. We distinguish three types of peering agreements among ISPs:

- Customer-to-Provider
- Provider-to-Customer
- Peer-to-Peer

The downstream transit customer (usually smaller sized ISPs) pays for the transit service offered by the upstream transit provider. Usually, peering among equally sized ISPs is not charged.

2.1.7 Tier Hierarchy

We refer to the definition of Tier Hierarchy given by [72]:

> *A hierarchical model of the relationships between ISPs. Tier 1 ISPs are large and together hold all the world's Internet routes, and peer with each other to give each other access to all Internet routes. Tier 2 ISPs buy connectivity (upstream transit) to the world Internet routes from one or more tier 1 ISPs, and hence their IP network(s) becomes a sub-set of those tier 1's IP networks. Tier 2 ISPs also peer with each*

> *other to minimize the amount of traffic to and from the tier 1 ISPs from whom they buy upstream transit. Tier 3 ISPs buy upstream transit from Tier 2 ISPs and so on, however the model becomes increasingly vague, since an ISP may buy upstream transit from both a tier 1 ISP and a tier 2 ISP, and may peer with tier 2 and tier 3 ISP's and occasionally a tier 1 ISP, and so on. The term is really only of use to differentiate between tier 1 ISPs who do not need to buy upstream transit due to their peerings with other tier 1 ISPs, and the rest of the ISPs, tier 2 and below.*

2.2 Internet Topology Concepts

As already said in Subsect. 2.1.1 we understand a *topology* as graph consisting of a set of nodes and links connecting these nodes. Related work (e.g. [23], [27], [64], [67] and [44]) has analyzed different types of topologies depending on the level of granularity. In this section we present these topologies. Figure 2.1 illustrates these different topologies.

2.2.1 Physical Topology

The lowest level one can imagine is a topology where each node is a physical device and the links refer to how these devices are physically connected to each other. Though a network administrator could create such a kind of topology of the network he is managing, this kind of low level information (i.e. OSI layer 2) is usually not publicly available.

2.2.2 IP-level Topology

The lowest protocol layer that is exposed to the users is the IP level (i.e. OSI layer 3). Mapping IPs to nodes and IP-level connectivity to links connecting these nodes yields an IP-level topology.
Note that IP addresses correspond to router interfaces and that a router may have multiple interfaces, one for each network attached to it. Thus, in an IP-level topology multiple nodes can correspond to a single router having multiple interfaces and thus multiple IP addresses which are called *router aliases*.

2.2.3 Router-level Topology

Resolving the IP addresses that belong to one single router is crucial for analyzing how packets travel trough a particular network. This process is called *alias resolution* (we present some alias resolution techniques in Subsect. 3.2.2). Aggregating aliases in an IP-level topology to a single node produces a router-level topology in which every node represents a single router.

2.2.4 POP-level Topology

An ISP houses a collection of routers in a so called Point of Presence (POP). By incorporating the geographic location of routers we can create a POP-level topology which is useful for understanding the geographic properties of Internet routing [67].

2.2.5 AS-level Topology

Every router in the Internet is part of an autonomous system (AS). By grouping routers according to which AS they belong we can create an AS-level topology. This kind of topology is useful for studying inter-domain connectivity and routing policies. Every link represents a peering agreement between two neighboring autonomous systems.

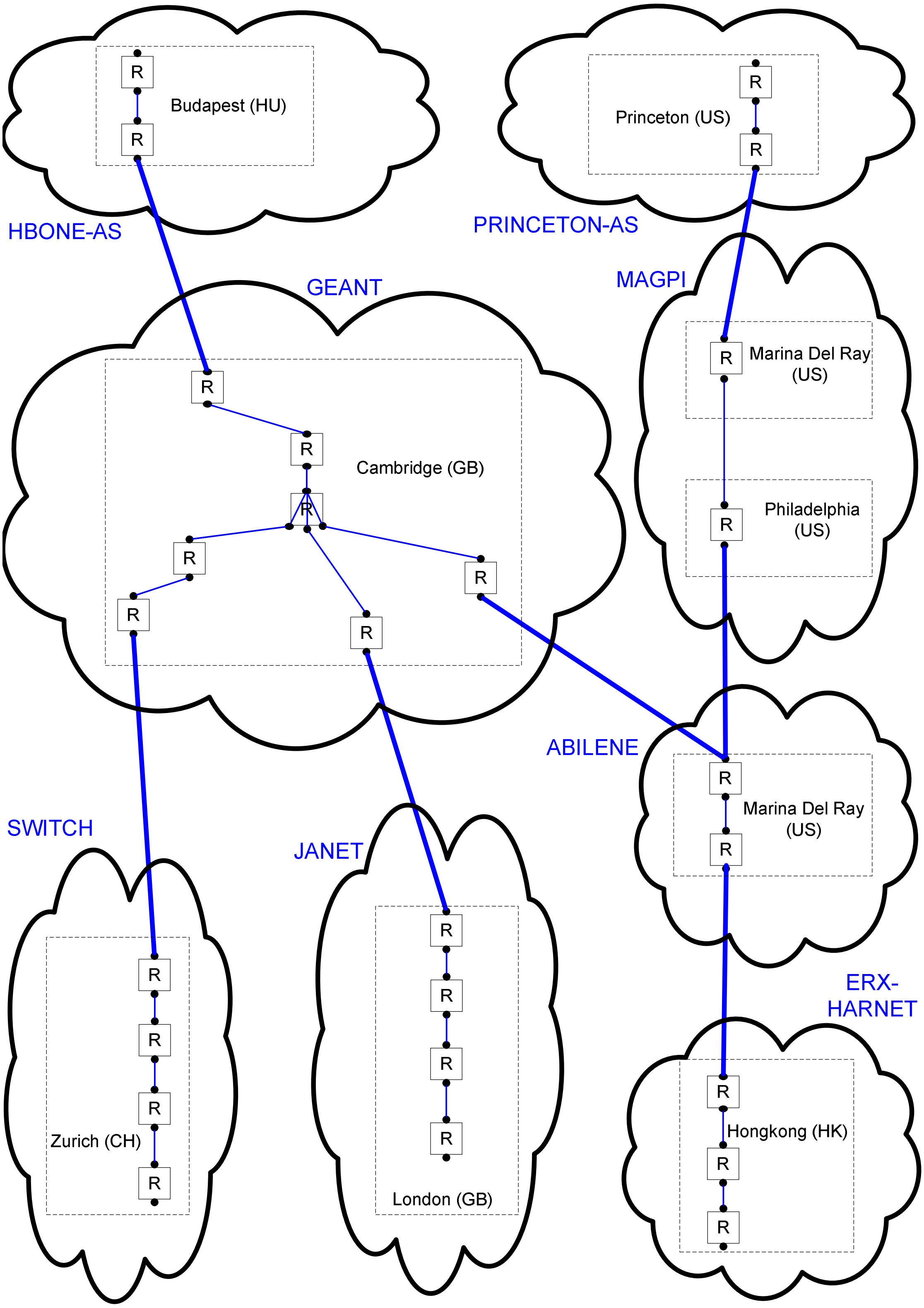


Figure 2.1: An illustration of the four levels of the Internet topology described in the previous subsections. Black dots represent router interfaces, squares labeled with "R" stand for routers and bold clouds for Autonomous Systems. Solid lines correspond to links on the IP-level, and together with the black dots they constitute the IP-level topology. The router-level topology is obtained when all interfaces of a router are grouped in a single identifier. Finally, the AS-level topology is obtained when we look only at ASs and the links between them. Links connecting different Autonomous Systems are bold. The POP-level topology is illustrated by dashed rectangles.

2.3 Internet Routing Concepts

In the Internet, a route is chosen hop-by-hop among the available routes. Routing policies influence these routing decisions. Of course failed links should be detected and avoided when choosing a path. But when multiple paths are available routing policies determine which paths are preferred over others. In this chapter we first introduce the Border Gateway Protocol (BGP) and then present an overview on intra-domain and inter-domain routing policies.

2.3.1 Border Gateway Protocol (BGP)

The Border Gateway Protocol (BGP) [57] is the core routing protocol of the Internet. Routers at the border of a network use external BGP (eBGP) to exchange routes with routers in neighboring Autonomous Systems (ASs). The externally-learned routes are then distributed inside the AS using the internal BGP (iBGP) protocol.

Each router stores a routing table indicating the best AS-level path to reach a destination prefix. A change in the network topology or routing policy causes BGP updates to be sent, either announcements of alternative routes or withdrawals. The time between the change and the time where all routers have chosen their new best routes is called the *convergence time*. During the process of BGP convergence the routers may have inconsistent views of the network.

BGP is implemented as a path vector protocol and thus the routers exchange whole paths. When a router receives a path it adds its Autonomous System number (ASN) to the path before propagating the path to its neighbors. Note that the router needs to prepend itself to the front of the path as route advertisements propagate in opposite direction to the data flow. The last ASN in the AS-path is called the *origin AS* of the destination prefix. For example, when the BGP path for the IP prefix *1.2.3.4/24* looks like *25,3,11,7* then AS7 "originates" the prefix *1.2.3.4/24*.

2.3.2 Intra-domain Routing

As we have seen in the previous subsection BGP is used to determine the AS-level route to reach a destination prefix. Given this information each router in an AS selects the best egress point for forwarding traffic toward that destination prefix. The internal path from the ingress point to the egress point is determined by an Interior Gateway Protocol (IGP), such as OSPF or IS-IS. In general, IGPs forward packets using the shortest path to the destination as is shown in Fig. 2.2.

2.3.3 Inter-domain Routing

ASs connect to each other either at public exchanges or at private peering points. We can distinguish between customer-to-provider, provider-to-customer and peer-to-peer business relationships which are expressed in routing policies. Delivering packets to a provider costs money and thus provider-to-customer or peer-to-peer links are preferred over customer-to-provider links. Usually routes learnt from private peers are not advertised to neighboring ASs. Other aspects like higher performance or reliability may also influence the routing policies.

When packets need to traverse more than one ISP the question arises which ISP will do the most work? An ISP can choose to employ either hot-potato (or early-exit) routing, where it passes packets to the next ISP as soon as possible, or cold-potato (or late-exit routing) routing, where the ISP chooses the peering point closest to destination. Hot-potato routing means less work for the ISPs network whereas cold-potato routing gives the ISP greater control over the end-to-end quality of service experienced by packets. Subramanian et al. [68] have found that most ISPs seem to employ hot-potato routing. However, this is just an hypothesis based on the observed routing behavior. Note that hot-potato routing is a potential cause for routing asymmetry as the early-exit point is the late-exit point in the reverse path [67]. Figure 2.3 illustrates this.

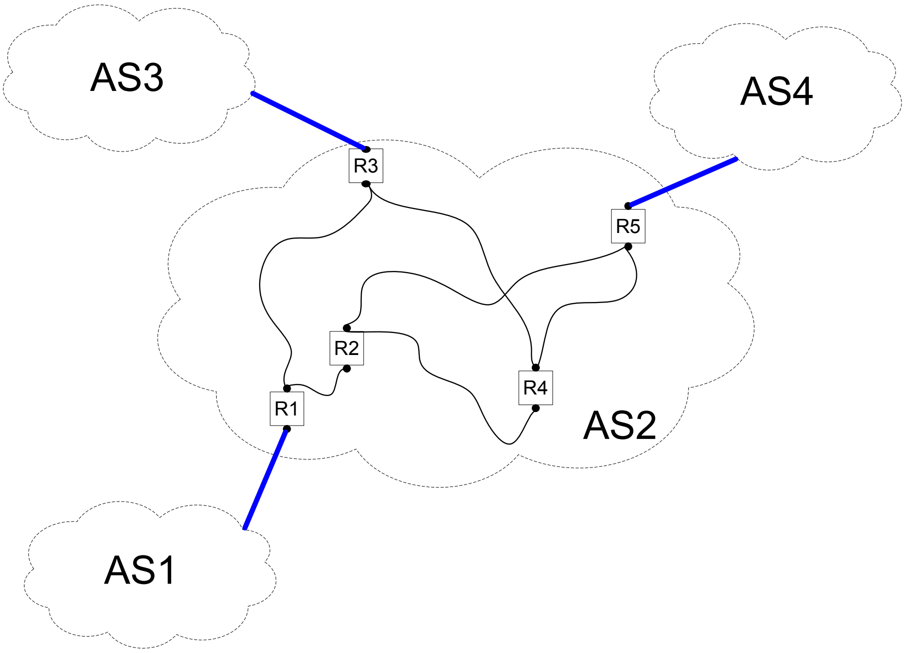


Figure 2.2: An illustration of intra-domain routing. Black dots represent router interfaces, squares labeled with "R" stand for routers and dashed clouds for Autonomous Systems. Solid lines inside AS2 correspond to intra-domain links on the IP-level, links connecting different Autonomous Systems are bold. Inside an Autonomous System packets are usually forwarded along the shortest path to the destination. In this sample network packets coming from AS1 with the destination AS2 are routed using the path R1-R3. Packets with the same source but the destination AS3 are routed using the path R1-R2-R5. In case the link between router R1 and router R3 would fail packets from AS1 to AS3 would take the new path R1-R2-R4-R3. The paths for packets with source A1 and destination A4 would remain the same.

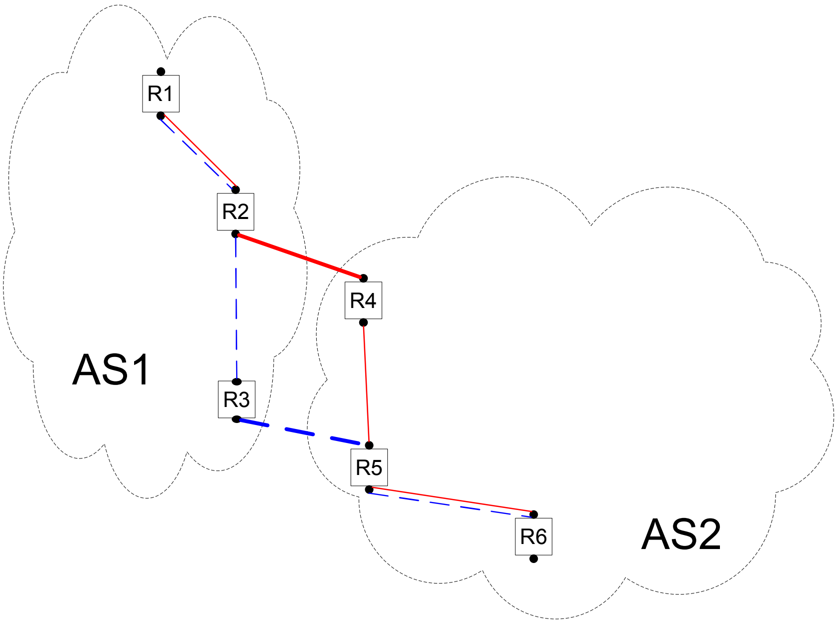


Figure 2.3: An illustration of routing asymmetry caused by hot-potato routing. Black dots represent router interfaces, squares labeled with "R" stand for routers and dashed clouds for Autonomous Systems. In the forward path from R1 to R6 the solid lines depict the hot-potato route and the dashed lines the cold-potato route. In the backward path from R6 to R1 it is the other way round. The solid lines corresponds to cold-potato routing and the dashed lines to hot-potato routing.

Chapter 3

Related Work

In order to compare routing paths and eventually detect routing changes, we first need to find out the underlying topology. In Chapt. 2 we have presented four levels of Internet topologies. All of these topologies have been studied in earlier networking research. Sections 3.1 to 3.4 present related work in the field of Internet topology inference. These projects attempt to create topology maps of the current Internet.
However, creating topology maps of the current Internet is not the goal of this Master's thesis. In Sect. 3.5 we present related work that focuses on the analysis of topology changes in end-to-end routing paths.

3.1 Topology inference at the AS-level

3.1.1 Topology Inference from WHOIS Records

WHOIS [77] is a protocol used to query a distributed database storing contact and registration information of networks. The database is manually maintained and contains information such as the owner of a domain name, an IP address, or an Autonomous System number (ASN). This information can be used to create an AS-level graph by mapping IP addresses to ASNs. Unfortunately, the resulting topology maps are quite inaccurate due to outdated or even wrong WHOIS records since institutions do not necessarily update their WHOIS information [44]. Therefore, WHOIS records alone are not a good source for creating topology maps.

3.1.2 Topology Inference from BGP Data

In Subsect. 2.3.1 we have learnt that the Border Gateway Protocol (BGP) is the core routing protocol of the Internet. Routers which are talking BGP store a routing table indicating the best path to reach a destination prefix. These routing tables can be used for AS-level topology inference as they provide information about the inter-domain connectivity at the AS-level.
In this subsection we first present some resources providing BGP raw data. Then we investigate how BGP routing information can be exploited for topology inference. After that, we present some studies which use the BGP-inferred topologies to investigate Internet topology and routing properties on the AS-level. Finally, we discuss some some limitations of BGP-inferred topologies.

BGP Archives A variety of publicly available Looking Glass [40, 74] and Route Servers [59, 74] provide read-only remote access for the purpose of viewing routing information on a specific router. However, the BGP routing information gathered using these resources typically provides only a limited view of the topology as seen by a single router. The Route Views project [50] from the University of Oregon captures multiple views of the global routing table by collecting and archiving both static BGP snapshots and dynamic message dumps from multiple ISPs. The BGP archive is continually updated. Most ISPs participating on the Route Views Project are located in the USA. A similar project is RIPE's (Réseaux IP Européens) Routing Information Service (RIS) [60]. It started as a RIPE NCC (RIPE Network Coordination Centre) project in

1999 and currently there are over 300 IPv4 and IPv6 peers at 12 data collection points in Europe, Japan and North America. Both BGP archives can only provide a partial view since not all ISPs participate. Nevertheless, various topological studies have benefited from these BGP archives. We cover some of these studies in the following two sections.

However, while BGP data can be easily obtained from public BGP archives, extracting useful information out of these archives is tedious due to the huge amount of raw data. BGP-Inspect [6] is a tool attempting to provide easy access to the information contained in large BGP archives (in form of summaries and statistics) as well as an effective query mechanism for this information.

Topology Inference Using BGP Routing Information BGP is implemented as a path vector protocol and thus each advertised route in a routing table is actually a list of autonomous systems that need to be traversed in order to reach a destination prefix. By combining the routes of several routing tables we can construct a partial view of the inter-domain topology using only passive monitoring of BGP messages.

As mentioned in the previous subsection, BGP archives like Oregon's Route Views project and RIPE's RIS project archive both static BGP snapshots and dynamic message dumps. Therefore, two different types of graphs can be inferred from BGP routing information: one from static snapshots of the BGP routing table and one from dynamic BGP message dumps (updates and withdrawals). Mahadevan et al. [44] found that the graph characteristics for these two topologies are nearly identical in terms of graph metric values like average degree, degree distribution and other graph metrics (see [44]). However, Dimitropoulos et al. [15] showed that BGP dynamics during BGP convergence can reveal additional topological information such as backup paths which usually are not visible in routing tables. They present a new topology inference method which exploits BGP update messages to produce more accurate AS-level maps than can be inferred from BGP routing tables alone. LinkRank [35] and BGPlay [5] are graphical tools for visualizing BGP routing and topology changes.

Studies based on BGP-inferred Topologies The paths chosen along a given Internet topology can be used to infer routing behavior as routing policies influence the path selection process. Several studies used the AS-level topologies inferred from BGP Routing information to investigate Internet topology and routing properties on the AS-level. For example, Gao [20] and Subramanian et al. [69] present algorithms that infer commercial AS relationships from BGP routing tables. Siganos et al. [63] studied AS-level Internet topology using data from Route Views and other BGP repositories. They showed that the Internet topology can be described efficiently with power-laws[1] and that these power-laws are persistent in time.

Note that AS-level topologies are not the only kind of topology which can be inferred from BGP data. Andersen et al. [2] create a cluster graph of IP prefixes by observing temporal correlations between BGP inter-domain routing update messages. Even though the clusters do not refer to the actual topology in terms of AS-level connectivity they reveal some properties like shared routing paths or assignment to the same POP.

Limitations of BGP-inferred Topologies Topology inference from routing messages is a passive approach. However, BGP data does not necessarily reflect the paths packets actually are routed trough because different routers may have different views. Furthermore, we do not know how many nodes and links are missing in the topologies derived from BGP data. BGP export policies may restrict the propagation of certain paths [20]. In addition, backup paths are not visible in the routing tables as these contain only the best paths to a destination prefix. Another limitation inherent to the passive nature of this approach is that we do not know the root cause that actually triggered a BGP update. We can only argue what caused the BGP update, having only a partial view and not knowing the routing policies an ISP applies. This is a complex task given the complex routing dynamics of the Internet. Many types of events can trigger the same routing changes and thus the same sequence of BGP update messages.

[1] Power-laws are expressions of the form $y \propto x^a$ where a is a constant, and x and y are the measures of interest, and $\propto$ stands for proportional to.

3.2 Topology inference at the IP-/Router level

In the last section we have argued that a completely passive approach is insufficient for topology analysis as many types of events can have similar effects. Researchers have thus made use of active measurements which allow for well-controlled experiments. Probes can be injected into the system under study and the observed changes in the system can be used to infer general properties of the system.

In this section we first present the traceroute tool, a popular tool for actively probing IP-level topologies. After giving a short introduction into the Alias Resolution Problem experienced by traceroute-based approaches, we give a short overview of related work using traceroute for topology inference. The related work can be classified by how the source(s) and destination(s) are chosen, whether the router aliases are resolved to get a router-level map or not, and how many vantage points are used. Some approaches additionally use BGP data to convert IP addresses to Autonomous System numbers (ASNs) for inferring and analyzing inter-AS Internet structure. Topology mapping between IP- and AS-level will be discussed in more detail in Sect. 3.3.

3.2.1 The Traceroute Tool

Traceroute is a popular tool for inferring IP-level topology. It allows the user to trace the path packets take from source to destination by taking advantage of the time to live (TTL) value of IP packets[2]. Several UDP packets are sent towards the destination. The first packet has a TTL of 1 which is increased by one in each successive packet until finally the destination is reached. Any router processing an IP packet will decrease the TTL value by one before forwarding it. When the TTL value reaches zero, the router drops the packet and sends its IP address and an "ICMP Time Exceeded" message back to where the packet came from. When the TTL is large enough the packet finally reaches the destination which will respond with its IP address and an "ICMP Destination Unreachable" message. This happens because traceroute addresses every packet to a very high port number (in the range of 32'768 and higher) and typically no one runs UDP services up there. Usually the port number is increased with every packet to keep track of which probe is being replied to. The default starting port is 33434. Because each ICMP response contains the IP address of the router it came from, the sequence of ICMP responses received at the probing host yields the forwarding path IP packets are routed through (Fig. 3.1). This information can help to identify routing problems.

```
traceroute to idwww01.unizh.ch (130.60.68.124), 30 hops max, 38 byte packets
 1  planetlab-gw.net.ic.ac.uk (193.63.75.17)  0.665 ms
 2  core-1-ext-c3550.net.ic.ac.uk (193.61.68.221)  0.493 ms
 3  ext-m7i-1-ge-1-3-0-4008.net.ic.ac.uk (194.82.153.9)  0.906 ms
 4  ic-gsr.lmn.net.uk (194.83.101.1)  0.386 ms
 5  194.83.100.133 (194.83.100.133)  0.515 ms
 6  london-bar1.ja.net (146.97.40.33)  0.524 ms
 7  po10-0.lond-scr.ja.net (146.97.35.5)  0.996 ms
 8  po6-0.lond-scr3.ja.net (146.97.33.30)  1.117 ms
 9  po1-0.gn2-gw1.ja.net (146.97.35.98)  7.357 ms
10  janet.rt1.lon.uk.geant2.net (62.40.124.197)  1.166 ms
11  so-4-0-0.rt1.par.fr.geant2.net (62.40.112.105)  8.434 ms
12  so-7-3-0.rt1.gen.ch.geant2.net (62.40.112.29)  17.394 ms
13  swiCE2-10GE-1-1.switch.ch (62.40.124.22)  19.181 ms
14  swiCE3-10GE-1-4.switch.ch (130.59.36.210)  17.351 ms
15  swiZH2-10GE-1-1.switch.ch (130.59.36.2)  21.455 ms
16  uzrtzi0-vlan811.unizh.ch (192.41.135.11)  21.725 ms
17  uzrtzi10-gig1-1.unizh.ch (130.60.5.2)  21.782 ms
18  uzrtzi21-gig1-1.unizh.ch (130.60.6.22)  21.601 ms
19  idwww01.unizh.ch (130.60.68.124)  21.460 ms
```

Figure 3.1: A sample traceroute output

Traceroute has also been extended with various features since the first implementation by Van

[2] Because of the possibility of transient routing loops, every IP packet includes a time-to-live (TTL) field. Every router decrements the TTL field before forwarding the packet. If the TTL reaches zero, the router sends an "ICMP Time Exceeded" error message back to the source.

Jacobson. For example, there exist a number of graphical traceroute implementations which map the geographical location of routers on a world map [25, 76]. In addition, other tools were implemented on top of traceroute allowing not only to monitor the path packets are routed trough but also properties like link delay or packet loss [32].
Because of its usefulness for network troubleshooting the traceroute tool has become a popular and widely used network debugging tool. Nowadays, a number of public traceroute servers [74] provide online interfaces which allow the execution of traceroute on request.

3.2.2 Router Aliases

The source IP address of the "ICMP time exceeded" messages should by default be the IP address of the router interface which received the traceroute probe [4]. Therefore, the IP addresses discovered by traceroute usually corresponds to the router interface which received the traceroute probe. As a result, the IP addresses seen in the forward path may differ from the ones seen in the reverse path, see Fig. 3.2.

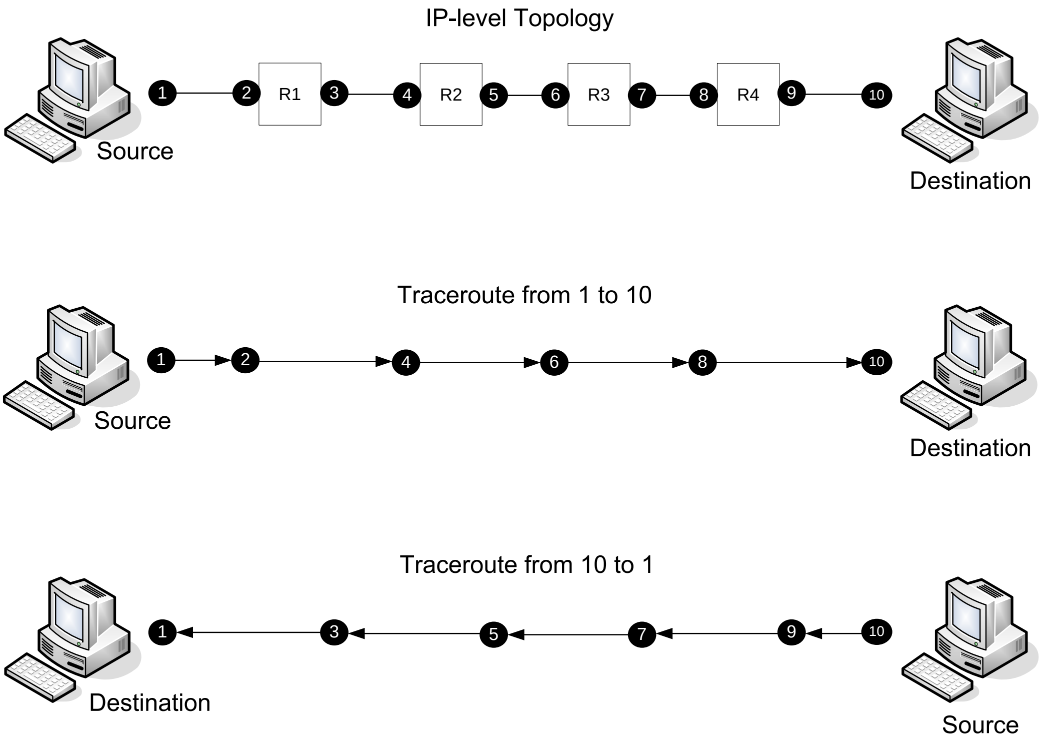
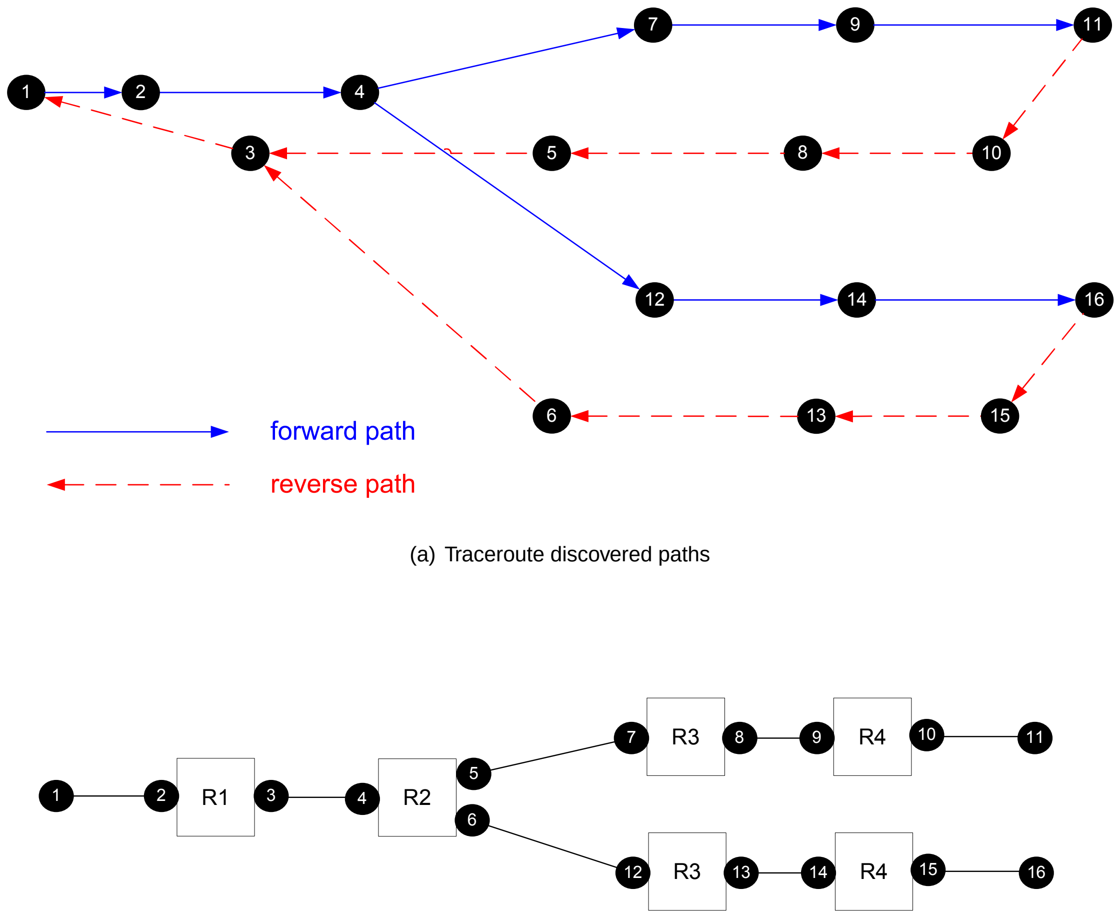


Figure 3.2: Traceroute discovers a router's input interface

For obtaining accurate router-level paths router aliases need to be resolved. Topology maps generated without alias resolving will have more nodes and links than alias resolved maps, see Fig. 3.3. We present some alias resolving techniques in the following subsections.

3.2.3 Mercator

Mercator [23] is a program that uses hop-limited probes to infer a router-level map. Though this approach is very similar to the probing primitives of traceroute, Govindan and Tangmunarunkit [23] intentionally implemented Mercator from scratch. This allowed them to design and test their own probing heuristics.
When using traceroute for topology inference, the topology mapping process needs a list of probing targets as input. Mercator was designed to be simple allowing it to be deployed anywhere, running "from a single, arbitrary location", "using only hop-limited probes". It is therefore restricted as it cannot use external information (e.g. DNS information or BGP data) to derive an initial target list. Instead it uses "informed random address probing" to guess about possible



(a) Traceroute discovered paths

(b) Router-level topology after resolving router aliases

Figure 3.3: Comparing traceroute paths and router-level paths

probing destinations. In addition, it applies "source-routed path probing" from source-route capable routers, thus adding virtual vantage points. Source Routing [29, 65] is a technique which allows the sender of a packet to specify the route that a packet should take. Source-routed path probing can discover routing paths which are not visible when probing from the monitoring host and thus help in creating a more complete topology map (Fig. 3.4). The problem is that most routers disable source routing because of security considerations.

Every node in the map discovered my Mercator represents a router interface. For obtaining an accurate network map, router aliases need to be resolved. The standard [4] requires routers to use the IP address of the outgoing interface as the source address for ICMP messages. This behavior can be taken advantage of by sending a UDP packet to a high-numbered UDP port (like traceroute probes but with at TTL of 255) and comparing the source address of the "UDP Port Unreachable" message. If two router interfaces respond with the same IP, then they in fact belong to the same router.

Mercator uses two refinements to this alias resolution technique which was first proposed by Pansiot and Grad [52]. First, Mercator repeatedly sends alias probes to an interface address as a router may have multiple outgoing interfaces, and it can happen that at a later time the router changes its route to the probing host and eventually also its outgoing interface address. This can reveal some more aliases for the same router. Second, Mercator uses source-routed alias probes as an interface may not be reachable from a particular vantage point, but it may

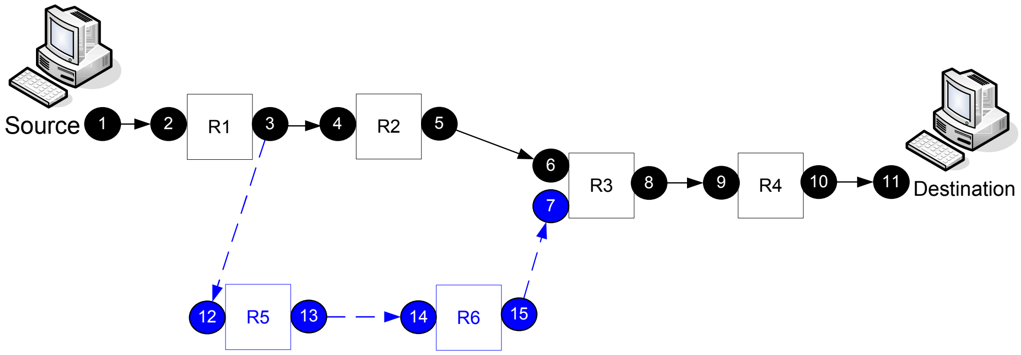


Figure 3.4: Illustration showing that source-routed path probing discovers additional routing paths and interfaces. In this example let us assume that router R1 allows source routing of packets. Solid arrows depict the default path from source to destination. With source-routing via router R5 traceroute additionally discovers the routers R5 and R6 and also another interface of router R3 (alternative path depicted with dashed arrows). Without source-routing traceroute will not discover this path.

be reachable from another vantage point. However, even with these refinements this method is not perfect, and some router aliases may remain unresolved. Furthermore, Mercator can only discover those interfaces which are reachable from its vantage point. The "source-routed path probing" alleviates this limitation, but does not guarantee a complete topology map. Thus, the resulting map may be incomplete, and in parts even incorrect.
It takes several weeks until Mercator can create a map of the Internet. During this time links may have changed, some new nodes may have appeared whereas other nodes may not be reachable anymore. Therefore, given the long time it takes to discover a map of the Internet, and given the fact that Mercator only discovers those links which it actually traversed during the topology mapping process, the resulting map can only be considered as a "time-averaged *routed* topology".

3.2.4 CAIDA's Macroscopic Topology Measurements Project

In 1998 the Cooperative Association for Internet Data Analysis (CAIDA) began its Macroscopic Topology Project [43]. The goal of the project is to collect and analyze Internet-wide topology and round trip time (RTT) data at a representatively large scale. The primary topology measurement tool is skitter [27, 64] which actively probes forward IP paths and round trip times (RTTs). Each skitter monitor continuously sends probe packets to hosts in its target list which is a subset of CAIDA's IPv4 destination list currently comprising more than 970'000 destinations. The selection criterion for the destination list is mainly based on the responsiveness of the probed hosts. A responsive destination provides RTT and non-truncated path information. Because the Internet is highly dynamic the destination lists need to be refreshed every 8 to 12 months to remove hosts which are no longer responsive.
The IP addresses discovered by skitter correspond to router interfaces and not to single routers. CAIDA's iffinder tool uses a similar approach like Pansiot and Grad [52] for resolving aliases discovered by skitter. As already mentioned in Subsect. 3.2.3, this heuristic is not perfect and some router aliases may remain unresolved. Furthermore, without the refinements applied by Mercator, iffinder will obviously resolve less aliases than Mercator would.
Like the topology maps created by Mercator, the topology discovered by skitter represent a time-averaged snapshot of the Internet topology at the IP-level. For a higher-level analysis on the AS-level, the forward IP path information obtained with skitter is correlated with inter-domain BGP routing tables. The observed IP addresses are converted to Autonomous System numbers (ASNs) based on the assumption that the terminating ASN in a routing table entry for a given destination prefix corresponds to the *origin AS*, which is administratively responsible for this prefix. IP addresses that are not advertised by any AS, or the contrary, prefixes that are advertised by several separate ASs (multi-origin ASs, MOASs), may distort the resulting map

and need special attention in the mapping process.

In order to create an AS-level topology graph each AS is laid out according to its geographic location extracted from WHOIS records. Usually the geographic information stored in WHOIS records corresponds to the geographic location of the AS headquarters and not to the geographical location of actual infrastructure. This can cause a wrong placement of an AS in the resulting map.

CAIDA's Macroscopic Topology Measurements Project discovers many more nodes and links than earlier work. Nevertheless, it is far from being complete. Consider that the IPv4 address space comprises millions of possible IPs and CAIDA's IPv4 destination list currently contains only about 970'000 destinations. Furthermore, skitter can only discover Internet connectivity of the current IPv4 address space.

In June 2003, CAIDA started the Macroscopic IPv6 Topology Measurements Project [42] in collaboration with the Waikato Applied Network Dynamics (WAND) research group of the University of Waikato, New Zealand. Like its predecessor skitter, the new tool scamper [62] actively probes the Internet. But unlike skitter, scamper supports not only IPv4 but also IPv6 path probing. In addition, scamper can also discover the maximum transmission unit (MTU) of a given path. The goal of the Macroscopic IPv6 Topology Measurements Project is to gather and analyze macroscopic IPv6 topology data, and to characterize the growth and progress of the IPv6 deployment worldwide.

3.2.5 Rocketfuel

Instead of creating topology maps of the entire Internet, Spring et al. focused on obtaining realistic router-level maps of ISP networks. Their aim was to build accurate ISP maps using as few measurements as possible. This is motivated by the the following observation. Extra traceroute measurements may help in creating a more complete topology map as more nodes and links may be discovered. But extra measurement also means that it takes longer to create a topology map. During this time the network being measured may change due to node or link failures, new nodes may appear, and paths may change. Thus, there is a trade off between the completeness and the correctness of the inferred topology. Focusing on the traceroutes that are likely to provide new information allows for efficient, more complete topology discovery, without sacrifying too much correctness. With the help of their ISP mapping engine Rocketfuel [66, 67] Spring et al. mapped the topology of ten diverse ISPs.

Rocketfuel uses more than 300 public traceroute servers as measurement sources, providing more than 750 vantage points. The destinations to probe for are chosen such that they provide the most valuable information on the ISP being mapped. Spring et al. implemented two different techniques to reduce the required number of traceroutes. First, they applied "Directed Probing" to prune out unnecessary traceroutes. "Directed Probing" uses routing information from the Route Views project [50] to identify the traceroutes that will transit the ISP network on focus. Only these traceroutes are relevant. Second, they applied "Path Reduction" techniques to identify redundant traceroutes, i.e. probes that will take unique paths inside the ISP's network. Using these two techniques the number of required measurements could be reduced by three orders in magnitude compared to a brute-force technique.

To obtain a router-level map, Spring et al. combine several techniques which are implemented in the alias resolution tool Ally. Like Mercator, Ally looks for common source IP addresses, but in addition, it combines this standard alias resolution technique with two pairwise tests. First, it assumes that packets sent consecutively by a single router will have consecutive IP identifiers in the IP header, and thus Ally primarily looks for nearby IP identifiers. Second, some routers generate an "ICMP Destination Unreachable" response only for the first of a number of consecutive probes. In case a response is only received for the first of two probe packets, Ally reorders the two destinations addresses being tested as aliases and probes them again after waiting five seconds. If again only the first probe solicits a response, the rate-limiting test detects a match. This test is not effective by its own because the loss of ICMP responses may happen by chance. Thus, Ally additionally tests possible rate-limited aliases for having IP identifiers which differ by less than 1'000.

Two heuristics are applied to determine the list of likely alias candidates. Ally first groups the destination IP addresses by the TTL of their responses. First, only those pairs which have similar TTLs are tested for being aliases. This approach may miss some aliases but is far more efficient than pairwise testing of all IP addresses discovered by traceroute. Second, to identify most aliases quickly, addresses having similar names are tested first. This heuristic relies on the implicit structure of DNS names.

Spring et al. found that the topology maps created with Rocketfuel contain roughly seven times as many nodes and links than the skitter maps for the corresponding ISP network. Their results also show that Ally finds almost three times more aliases than earlier alias resolution techniques. Furthermore, they found that the aliases discovered by the IP identifier technique is a superset of any technique based on common source addresses. Nevertheless, Ally cannot discover all aliases, mainly because some routers do not respond to alias probes. This a problem common to any alias resolution technique.

In addition to resolving router aliases, Spring et al. also decoded DNS names to find a structure in the resulting ISP map. By analyzing the naming convention of several ISPs they were able to identify which routers actually belong to the ISP being mapped, the role of the router and its geographic location.

3.2.6 Scaling Problems with Multisource Traceroute

The work discussed previously has already suggested that running traceroutes from multiple sources will result in a more complete topology map. But there is another reason which argues for a distributed traceroute approach. Running traceroutes from a few sources to a large number of destinations will create a topology in which the neighborhood near the sources is more explored than nodes which lie further away towards the destination. This leads to a sampling bias in the generated topologies [34, 1]. A distributed traceroute approach would not only discover more nodes and links, but could also compensate the sampling bias.

DIMES [14] is a publicly available monitoring agent, which runs traceroute and PING measurements as a background process, consuming at peak 1KB/s. Its lightweight architecture removes the need for dedicated measurement nodes and allows it to be deployed on any volunteer machine. Currently, over 4’000 users are participating in the measurement community, running more than 8’000 agents in 88 countries.

Donnet et al. [16] argue that the problem of such a distributed measurement structure is the inherent scaling problem. Probe packets from multiple monitors concurrently probing towards the same destination may reassemble to a distributed denial of service (DDoS) attack. To avoid this DIMES traces slowly. The problem with this approach is that the routing changes during the measurement interval may introduce noise into the resulting map. Donnet et al. introduce the Doubletree Algorithm [16] as an efficient cooperative algorithm that allows to perform large-scale topology discovery efficiently and in a network-friendly manner.

The paths discovered by traceroute tend to have a tree-like structure. Looking at the source, the routes seem to have a tree-like structure which is rooted at the source and leading out towards multiple destinations. Similarly, looking at the destinations, the routes seem to come from multiple destination, converging in a tree-like structure rooted at the destination. The Doubletree algorithm tries to guide the topology discovery by taking advantage of this tree-like structure.

"Intra-monitor" redundancy describes the redundancy in the traceroute paths discovered by a single monitor. A large number of traceroute probes in the neighborhood of this monitor can be considered as redundant as they do not find any new interfaces. Similarly, probes from multiple monitors are likely to find nodes which were already discovered by other monitors. This is termed as "inter-monitor" redundancy. The key idea behind the Doubletree algorithm is to reduce the intra-monitor and inter-monitor redundancy as much as possible. This is achieved by first probing each path starting near its midpoint, and then, by applying stopping rules which avoid that already seen interfaces will be probed again.

Probing starts near the midpoint of a path and propagates into two directions: the backward path towards the source, the forward path towards the destination. The "local stop set" consists of all interfaces already seen by a single monitor. When probing the backward path, the monitor stops as soon as it detects an interface which is already in the "local stop set". When probing the forward path, probing stops when an interface is already contained in the "global stop set". For creating the "global stop set" monitors share information regarding the paths that they have already explored. Bloomfilters [7] may be used to reduce the communication overhead from sharing the "global stop set" among the monitors.

Donnet et al. found that compared to the standard traceroute approach, Doubletree was able to reduce the measurement load by approximately 70% while maintaining 90% interface and link coverage. In [17] they present their open-source implementation of Doubletree in a tool called traceroute@home.

3.2.7 Summary

In this section we have described the traceroute tool and we have given a short overview on related work using traceroute for inferring Internet topology and routing properties. We have presented different approaches and the challenges which come with topology inference using traceroute.

A problem we experience with traceroute is the limited view of the monitoring host. Traceroute may not discover backup paths or unused links as it discovers only those paths which are actually taken by the probe packets. For example, if a backup link is never traversed during a traceroute measurement, this link will never be discovered. Furthermore, traceroute paths represent only the forwarding path. Routing asymmetries can only be discovered by tracing the backward path too.

Because traceroute probes from one single source to a number of destinations, traceroute paths are biased toward the location of the monitor. Nodes near the monitoring host are more likely to be detected by one of the traceroutes running from the monitoring host. This results in a nearly complete topology of the neighborhood of the monitoring host, whereas the topology along the path toward the destination gets more sparse the farther away from the monitoring host. A distributed traceroute approach alleviates this problem (see Subsect. 3.2.6) as it can combine the topology information gathered from several monitoring points.

However, the current Internet is too large as that it could completely be probed by any active probing technique. A distributed traceroute approach may result in more complete topology maps but the measurements need to be conducted carefully to avoid looking like a distributed denial of service (DDoS) attack (see Subsect. 3.2.6). In addition, the problem is that active probing produces extra traffic on the network. Therefore, there is a trade-off between the completeness of topology information and the resource consumption. Heuristics for avoiding unnecessary or redundant traceroute probes are essential for large-scale topology discovery (see Subsects. 3.2.5 and 3.2.6).

Note that the description of the traceroute tool in this section does not consider special cases and problems showing up in the traceroute output. We will discuss problems and challenges when using traceroute for topology analysis in more detail in Sect. 5.7.

3.3 Topology mapping between IP- and AS-level

In Sect. 3.2 we have seen that traceroute is a valuable tool for discovering the forwarding path of IP packets, and that these paths can be merged to create an IP-level map of the Internet. Traceroute-inferred topologies can be used to study Internet topology and routing properties. But for debugging routing or performance anomalies it is essential that we can locate the network responsible for the problem. IP-to-AS mapping according to the information stored in Internet routing registries [49] suffers from out-of-date or incomplete registry information. Thus, we need a more sophisticated and accurate IP-to-AS mapping algorithm.

Spring et al. [66] decode DNS names to identify which routers belong to which ISP. For example, the DNS name *att-gw.sea.sprint.net* presumably stands for a gateway of AT&T which is located in Seattle, USA. The problem of this approach is that there is no standard naming convention. Therefore it needs a lot of manual work to find out the naming convention of each ISP. But sometimes it may not be possible to assign a DNS name to an ISP since the ISPs are not obligated to give their routers meaningful names. Some router interfaces may also have misconfigured or obsolete names. In other cases DNS names are not available for a specific router interface. For example, some ISPs in Korea and China do not use DNS names at all [67]. It is very difficult to validate the results of the IP-to-AS mapping as the algorithm is mainly based on interpreting names which do not follow any standard.

CAIDA’s skitter tool [27, 64] uses an approach based on the assumption that the terminating Autonomous System number (ASN) in a BGP routing path corresponds to the origin AS (the Autonomous System which is administratively responsible for this prefix). The same approach is used by the Route Views project [50]. The Route Views project provides the DNS-Zone *asn.routeviews.org* which returns the origin AS for a given IP.[3] However, the AS path advertised via BGP may differ from the AS-level forwarding path due to route aggregation or routing anomalies. Multiple origin ASs (MOASs), route aggregation and unannounced prefixes are another challenge for this approach.

All in all, current approaches have several limitations and the mapping can contain errors. Mao et al. [45] investigated on the root cause of BGP and traceroute AS path mismatches. They found that most mismatches are due to inaccurate IP-to-AS mapping applied to the traceroute paths. For example, the presence of Internet eXchange Points (IXPs), where multiple Autonomous Systems connect to exchange traffic, may cause wrong IP-to-AS mapping as the traceroute paths differ from the announced BGP routes. Multiple ASs belonging to the same organization or unannounced prefixes may also result in wrong IP-to-AS mapping of the traceroute path.

In [46], Mao et al. present a systematic approach for accurate IP-to-AS mapping using dynamic programming and iterative improvement to minimize the number of mismatches. By changing only 2.9% of the initial IP-to-AS mappings the algorithm reduces the fraction of mismatches from 15% to 5%. Refer to [45], [46] and [61] for more details.

3.4 Other approaches for topology inference

3.4.1 IP Measurement Protocol (IPMP)

Current packet probing techniques use existing protocols like ICMP, UDP or TCP to perform active measurements. Luckie et al. [41] argue that the problem with this approach is that these protocols were not designed for measurement. Encapsulating probe packets into these general purpose protocols may bear serious limitations. ICMP, UDP and TCP packets may be subject to protocol-based filtering or priority queuing policies. This may distort delay measurements. For example, some routers filter or entirely block ICMP packets. UDP on the other hand does not implement congestion control and thus may be rate limited to reduce its impact on TCP flows during periods of high UDP usage. In that case, changes in the measured delay do not necessarily correspond to changes in the network load. Finally, the problem with TCP is that the delay measurements conducted with TCP packets will include an overhead introduced by the TCP stack management.

The IP Measurement Protocol (IPMP) [48, 31] is presented as a protocol which is intentionally designed for measurement activities. The protocol is based on an echo request and reply packet exchange. It was designed to be easy to implement and having low packet overhead. Only very few word modifications are needed to create an echo reply packet out of the echo request packet.

The encapsulation of probe packets into a dedicated protocol allows for more flexible filtering and avoids probing packet being blocked by security policies which are meant to block other packets. Furthermore, IPMP supports not only delay measurements but also path measurements in both the forwarding and the reverse path. The IP address of the network interface which received the IPMP packet can be stored in preallocated space. Placing path records in preallocated space allows for efficient packet manipulation. The IPMP echo protocol format has a queue type field which allows router to queue an IPMP packet as it were another protocol. This enables analyzing routing behavior for different protocols.

In short, IPMP enables the measurement of routing paths and packet delay in a single packet exchange between the measurement host and the probed destination. Furthermore it is designed to overcome the limitations of existing probing techniques. In case it will actually be deployed, it would be an efficient alternative to the traceroute approach.

[3] The whois query has the following format: "host -w -t txt REVIP.asn.routeviews.org" where REVIP should be replaced by the reverse IP address (e.g. IP = 1.2.3.4, REVIP = 4.3.2.1). The result of the query is the Autonomous System number of the origin AS of IP (as seen by Route Views)

3.4.2 Network Tomography

Inferential Network Monitoring or Network tomography is a new field which applies statistical inference techniques to determine performance attributes that cannot be observed directly. Based on the observation that one cannot rely on the cooperation of the routers, the problem of network monitoring and inferring network properties becomes a problem similar to signal processing or statistical problems such as medical tomography.
Conventional traceroute approaches for topology inference require the cooperation of network devices. As firewalls are becoming more common traceroute may not be applicable anymore in the future. Network Tomography [9] provides an interesting approach to topology identification that requires only end-to-end measurements and no cooperation from intermediate routers. Of course the knowledge gained by such measurements is limited and thus with network tomography it is only possible to derive "logical topologies". "In the logical topology, each vertex represents a physical network device where traffic branching occurs, that is, where two or more source-destination paths diverge." [9]. The topology identification algorithm is based on a measurement scheme called "sandwich probing" which considers delay differences observed in the end-to-end measurements. Refer to [9] for the technical details.
There is a trade-off between the accuracy of the results and the computational burden of the tomographic approach. Current network tomography algorithms are still computationally too intensive for any network of reasonable scale. However, with the continuing proliferation of firewalls and other security means, Internet tomography will probably gain more importance in the future as active probing techniques may no longer be applicable.

3.5 Detecting and Analyzing Route Changes

As explained in Sect. 4.2, the main goal of this Master's thesis is to detect and analyze routing changes in end-to-end routing paths. While the problem of Internet topology inference has been well studied, we only found a small number of projects which try to analyze routing changes. We present these projects in the following subsections.

3.5.1 Traceanal: A Tool for Analyzing and Representing Traceroutes

In September 2001, SLAC (Stanford Linear Accelerator Center) started its Internet End-to-end Performance Monitoring - Bandwidth to the World (IEPM-BW) project [28]. The IEPM-BW system monitors and evaluates network connectivity and end-to-end performance to many sites that SLAC collaborates with. The goal of the project is to automatically detect major decreases in bandwidth to one of the collaborator sites. During the course of the project the IEPM-BW team has found that many such changes can be associated with route changes. To analyze the correlation of throughput changes and performance changes they developed Traceanal [36, 38], a tool for analyzing and representing traceroutes. SLAC's Traceanal tool monitors routing changes in traceroute paths and generates daily traceroute summary tables which provide a visualization of change patterns in traceroute paths (a detailed description is given in Subsect. 5.4). Logg et al. [38] plotted time series of IEPM-BW measured performance metrics together with significant traceroute changes detected by Traceanal and found that there is a visible correlation.

3.5.2 Analysis of End-to-end Routing Behavior in the Internet

Paxson [53] analyzed 40'000 end-to-end routing paths which were collected by running traceroute measurements between 37 different Internet sites. The measurements were taken during two different measurement periods. The first set of data was collected from November 8 through December 24, 1994. The second set of measurement was conducted from November 3 through December 21, 1995. Paxson analyzed both data sets for routing pathologies like for example routing loops or temporary outages and found that the likelihood of encountering a major routing pathology more than doubled between the end of 1994 and the end of 1995. Paxson also investigated on the routing stability of Internet paths by quantifying first the "prevalence" of Internet routes, and second their "persistence". The prevalence of a route is referred to as the unconditional probability that this route is observed in a traceroute measurement. The persistence of a route on the other hand denotes the likelihood of an observed route to endure before changing. Paxson's results suggest that Internet routes tend to have a dominating path, that is, a route with high prevalence. In addition, two third of the studied paths had quite stable routes, persisting for either days or weeks. Though, Paxson also found that there are significant variations between sites and in average Internet routes might be stable, but there exist a few routing paths which exhibit a high rate of routing changes.

3.5.3 A Framework for Pin-Pointing Routing Changes

Deterioration of round-trip time and/or available bandwidth is often correlated with topology or routing changes in the end-to-end path between two hosts (see Sect. 4.1). Sometimes such changes may even cause the complete loss of connectivity. These problems can only be fixed if we can determine *where* and *why* a change occurred.

Active measurement tools such as traceroute can be used to detect changes in the end-to-end path, but the probing results do not provide enough information on what caused the route to change and where the change originated. Analyzing BGP data is also not sufficient as many types of events can trigger the same routing changes and thus the same sequence of BGP update messages. Teixeira and Rexford [70] showed that many routing changes are not visible in the BGP data. Furthermore, BGP data only provides a partial view of a network and sometimes this may lead to inaccurate conclusions about where and why a routing change occurred. To demonstrate some of these limitations we will give a short summary of two examples taken out of [70].

Example 1: Invisible BGP Changes Due to Partial View Suppose that the routers of the network in Fig. 3.5 apply hot-potato routing. The routers A and C will route packets determined for destination prefix d through AS 2. Routers B and D choose the path trough AS 3. In case the link between router A and AS 2 fails, A and C will have to route their packets through AS 3 to reach destination d. The paths for routers B and D remain unchanged, and thus, a measurement system which is collecting BGP data at router D will not notice the link failure and the resulting route changes. This example shows that a measurement system needs to consider BGP data from all the border routers of a network as otherwise it may miss some changes.

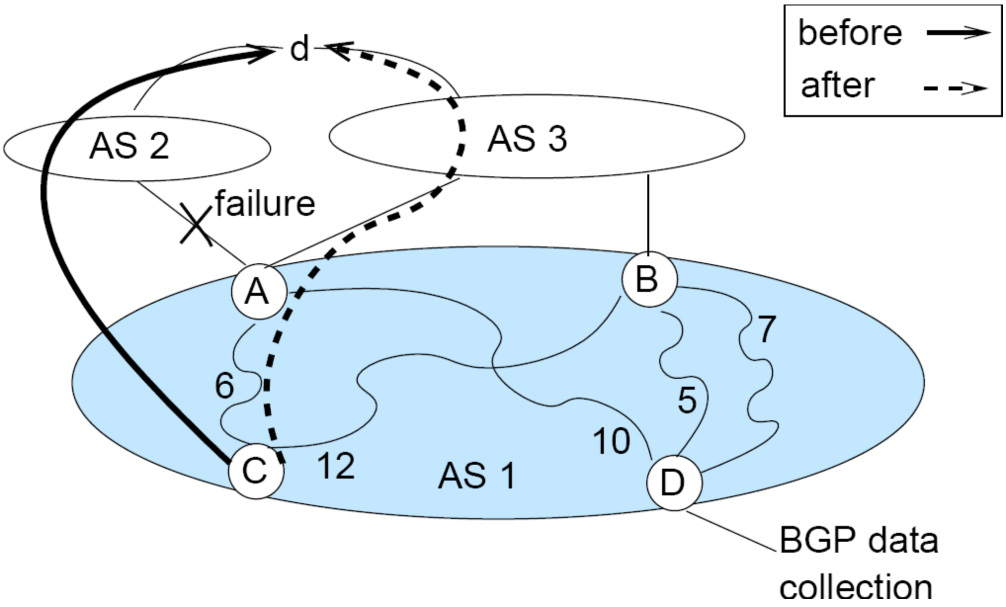


Figure 3.5: BGP changes are not detected at data collection point (This figure is taken out of [70], Figure 1, page 2)

Example 2: Inaccurate Conclusions Due to Partial View Suppose again that the routers of the network in Fig. 3.6 apply hot-potato routing. Router C uses the path trough AS 2 to forward packets towards destination d2. In case the cost of the link between router A and C will increase to 11, router C will switch to the route via AS 3. C's routes for destinations d1 and d2 remain the same. This hot-potato routing change could be misinterpreted by an external observer. An external observer would notice that all paths for prefixes in AS 4 switched from "1 2 3" to "1 3 4". Looking at AS 2 and AS 3 he would notice that the prefixes originated by these Autonomous Systems did not change. Thus, he will come to the conclusion that either AS 4 or the link between AS 2 and AS 4 is faulty, which in this case is not the case. This example shows that routing changes inside an AS may sometimes affect external paths. Such internal changes are not visible to outsiders, only their effects. The route cause for such a routing change can only be determined with the cooperation of all the involved Autonomous Systems.

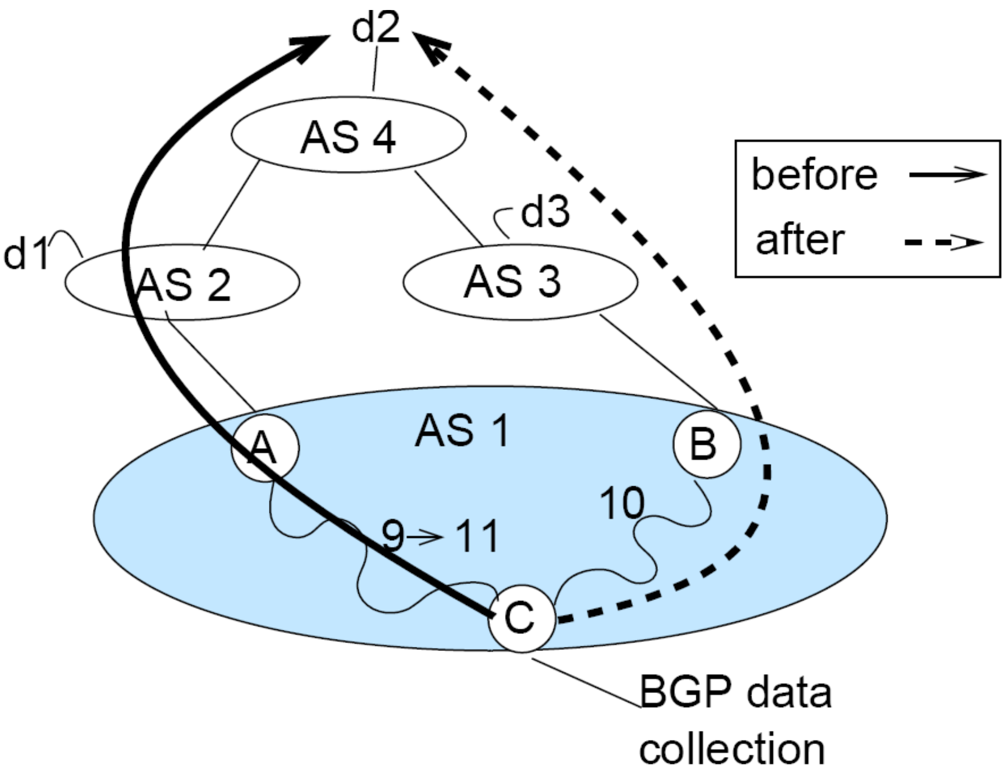


Figure 3.6: Internal routing change affecting only destinations in AS 4 (This figure is taken out of [70], Figure 4, page 3)

Teixeira and Rexford [70] argue that it is crucial that the network directly supports the diagnosis of routing problems. They propose a framework where each AS helps in diagnosing routing problems. Each AS should have an "Omni server" that maintains a network-wide view of the routing state of its AS in a so called "AS-level forwarding table". In case of routing anomalies, the Omni server can be queried if a routing change occurred and when yes, whether the change had a local cause. If the cause was local, then the Omni server can directly respond what caused the routing change. Otherwise, the routing change had an external cause, and the Omni server responds which neighboring AS should be queried for more information. In this way the diagnosis can be propagated to the network which was actually responsible for the routing problems.

This framework would greatly help in diagnosing routing problems. But, like the IP Measurement Protocol, unfortunately it is not yet implemented in the current Internet. However, it gave us a lot insight on the problems encountered when diagnosing routing changes.

3.6 Summary

In this chapter we have seen a number of different approaches for Internet topology inference, and the problems and limitations of each approach. We have seen that measuring the Internet is actually difficult.

In Subsect. 3.4.1 we have presented the IP Measurement Protocol which when deployed would make Internet measurement more efficient and accurate. But because until today we have only limited means for measuring the Internet, it seems best to combine different approaches.

Topology inference using WHOIS records alone is not sufficient as this approach suffers from inaccurate, outdated or even wrong WHOIS records. BGP routing information can be used to infer AS-level topologies. However, these topologies do not necessarily reflect the paths packets are routed trough. Traceroute discovers the paths packets are routed through the Internet, and when combining traceroute results with BGP data and WHOIS records, additional information such as Autonomous System number and geographical location can be derived.

It is important to know the limitations of each approach in order to derive correct conclusions from the measured data.

After discussing the different approaches for topology inference, we presented related work that focuses on the analysis of topology changes in Sect. 3.5. We presented Traceanal, a tool which provides a visualization of change patterns in traceroute paths. Then we gave a short summary of Paxson's report on the analysis of Internet routing behavior. Finally, we presented the (theoretical) measurement framework proposed by Teixeira and Rexford which would help in pin-pointing routing changes.

Chapter 4

Problem Discussion

4.1 Why bother about Topology Changes?

There are various reasons why Internet routes change. On the one hand the Internet is very dynamic and it is very common that node or link failures occur. Hardware defects, or simply the reboot of a router (e. g. for software updates or maintenance work) can result in topology changes which in turn indirectly also result in routing changes. Usually there exist more than one possible path to reach a destination prefix and in case of link or node failures routers need to find a new forwarding path.

In addition to such unintentional routing changes, ISPs can intentionally decide to change their routing policies. In either case the routing changes may result in subsequent path and routing changes, both in the network where the routing change originated, but also in the neighboring networks (see Subsect. 3.5.3).

Routing changes require the routers to update their paths. During the process of path convergence the routers may have inconsistent views of the network. Traffic in transit may get lost due to invalid paths and transient routing loops emerging from this inconsistent view. When finally the paths have converged and all routers have chosen their new best routes, all the traffic will be forwarded using these new paths.

Topology and routing changes in the end-to-end path may cause changes in the performance experienced by the user. This happens when the new path has different properties than the old path. For example the new path may have a larger round-trip time, lower available bandwidth, smaller maximum transmission unit (MTU), or more restrictive packet filtering policies. Misconfigurations in the new path may create routing loops leading to packet loss since the packet remains in the loop until the TTL field expires and the packet gets dropped.

Sometimes paths do not converge. Hardware or software errors, misconfigurations or unreliable connections may result in route flapping [58]. This is when a router repeatedly advertises and withdraws one or more routes during a small time interval, recalculating its best route to a destination prefix over and over again. This is a serious problem. Unstable flapping links lead to an alternately increasing and decreasing delay as well as dropped packets. The need for constant rerouting of packets results in bad link quality.

Open Systems AG [51] has made the experience that many performance problems experienced by the user can be associated with changes in the end-to-end routing path. Logg et al. [38] found that "route changes frequently do not cause throughput changes, but throughput changes often correlate with route changes". According to the results of Logg et al. [38] most (80%) of the route changes do not affect end-to-end performance, but about half of the throughput changes are caused by route changes. This suggests that the first thing to check for when debugging performance problems is whether there was a routing change or not.

4.2 Requirements Analysis

In this Master’s thesis we want to investigate topology changes in end-to-end routing paths and their effects on the performance experienced by the user. For that purpose a prototype should be implemented which regularly monitors the topology of end-to-end paths and detect route changes. The prototype should fulfill the following requirements.

- On the one hand the system should regularly analyze the topologies and detect changes. On the other hand it will be invoked in case of connectivity problems detected by other subsystems running on the monitoring host.
- It should be possible to compare different routing paths and eventually reveal a correlation of performance changes with specific route changes.
- The findings should be summarized and visualized in a report that can be forwarded to the ISP and helps them solve the problem.
- The resource consumption and performance of the prototype should not affect regular operations on the monitoring host or the network.

4.3 Discussion of Related Work

4.3.1 Related Work on Topology Inference

In Chapt. 3 we have presented a number of different approaches for Internet topology inference. The projects that create topology graphs of the current Internet aim to achieve as complete topologies as possible. For that purpose they measure over a long period of time, generating a time-averaged topology graph of the Internet. This is true for all levels of Internet topology.

Whereas for the projects mentioned above it is very important to detect as much nodes and links as possible, we only want to detect the nodes and links (the path) between a given end-to-end connection. This greatly reduces the scale of the measurement and the time needed to complete a topology graph. Indeed, the latter is actually important. The longer it takes to determine the topology of an end-to-end routing path, the more likely are we to miss a topology change. Thus, for our purposes, the up-to-dateness of the measured topologies is a very important quality measure.

Taking all this into account, topology graphs generated from WHOIS records do not fulfill our requirements as they do not provide up-to-date topology information. Also BGP-inferred topologies are not sufficient to analyze route changes because these topologies do not necessarily reflect the paths packets are routed trough. More promising is the traceroute-based approach. Using traceroute we can discover the path packets are actually routed trough the network. Thus, for this Master's thesis, we will focus on a traceroute-based approach for determining the topology of Internet routing paths.

4.3.2 Related Work on Route Change Analysis

The problem of Internet topology inference has been well studied, but during our literature research we only found a small number of projects which try to analyze route changes. [10] and [8] used BGP data to investigate route changes on the AS-level. However, as we have seen in Subsect. 3.5.3 many route changes are not visible in the BGP data. We only found two related projects about route change analysis of end-to-end measurements. For example, [53] investigated on the routing behavior of end-to-end routes using traceroute measurements (see Subsect. 3.5.2). Another example is SLAC's Traceanal tool [36, 38] which provides a simple visualization of route change patterns in end-to-end routes obtained with traceroute (see Subsect. 3.5.1 and Sect. 5.5).

Of all these approaches, Traceanal would best suit our requirements as it already provides a monitoring framework for detecting and visualizing route changes. The prototype implemented during this Master's thesis is based on the Traceanal tool (see Chapter. 6).

4.3.3 Other Related Tools

PingPlotter [54] and Traceping [73] are two tools which try to debug performance problems with the help of traceroute (and ping). They try to locate the source of performance problems by monitoring packet loss over time and analyzing when and where the packet loss originated. This problem is related to the task of this Master's thesis. Though, in this Master's thesis we go a step further. Once a change in the end-to-end performance is detected (e.g. by other subsystems running on the monitoring host) we want to find out whether this performance change was caused by a route change or not. PingPlotter and Traceping do not analyze route changes. Thus, for this Master's thesis we do not consider these two tools, but their approach is tightly correlated with our task.

Chapter 5

Own Approach

In Sects. 3.1, 3.2 and 3.4 we have presented a number of different approaches for topology inference. In Subsect. 4.3.1 we have discussed these approaches in the context of our requirements and finally found that the traceroute approach is the most promising for our purposes. Sect. 3.5 presented related work on the analysis of route changes. In this chapter we describe our own approach in detail. In Sect. 5.7 we will discuss some of the problems and limitations of the chosen approach.

As already mentioned in Subsect. 4.3.2, our own approach is based on SLAC's Traceanal tool [36, 38]. Thanks to the courtesy of SLAC's IEPM-BW [28] team we got a copy of a standalone version of traceanal which can be adjusted to run also with Non-IEPM-BW traceroute data. Our own approach is based on this version of traceanal, but we have improved the whole framework and also implemented a number of new features. We will explain the implementation details of the original and the improved framework in Chapt. 6. In this chapter we will focus on the theoretical background of our own approach.

5.1 Test Environment: PlanetLab

As stated in Subsect. 4.3.1 we have decided to use a traceroute-based approach for topology inference. We have learnt from related work (see Chapt. 3) that monitoring from one single source would only give a limited view. Therefore, we need to collect traceroute data on more than one monitoring host in order to get representative results. In the optimal case, the monitoring hosts are distributed world wide.

As already mentioned in Sect. 1.2, this Master's thesis was conducted at Open Systems AG [51]. At the beginning the idea was to use the VPN gateways as monitoring nodes. But because the services running on the VPN gateways are mission critical it is not a good idea to run a prototype on them which may cause unforeseen side effects[1]. Of course we could have tested the prototype in a local test network but we wanted to test the prototype under real conditions as it is the size and complexity of the current Internet which makes debugging of performance problems difficult. Because of these considerations, we decided to use the PlanetLab nodes [55] instead. This allowed us to test our prototype under real conditions, without the restrictions we would have on the VPN gateways.

PlanetLab [55] is a collection of worldwide distributed machines organized as a consortium of academic, industrial, and government institutions. It was designed to allow researchers to experiment with network applications and services in a worldwide distributed overlay testbed under real-world conditions, and at large scale. PlanetLab currently consists of 689 machines located in over 25 countries all of which are connected to the Internet.

The PlanetLab testbed was very useful as it allowed us to test our prototype in a distributed environment. But we made the experience that managing all the nodes can be quite time consuming. We always needed to check if our measurement script is still running well. We used the PlanetLab monitoring tool CoMon [12] to get an overview of which of our monitoring nodes are

[1] Active measurement techniques like traceroute produce extra traffic and thus may influence other services on the monitoring host and/or the network. Furthermore, Open System's Mission Control Services monitor all traffic on the VPN gateways and our test measurements could cause wrong alerts.

currently active. This helped us in finding the nodes which behaved abnormally. For example, when a PlanetLab node rebooted, sometimes our monitoring script was not restarted although it was configured to do so. The corresponding monitoring node was not listed anymore in the report table of CoMon and thus we knew that it was not running correctly.

Sudden reboots of nodes were not the only problem we had to deal with. During the course of this Master's thesis some of our PlanetLab nodes were not available for a few days. For some of our nodes this occurred quite often and some nodes were even unavailable for weeks. Traceroute measurements running to a destination node which is not available are more intrusive than a successful measurement since all probing packets will time out and traceroute tries probing until the maximum time to live (TTL) value is reached (default is TTL=30). We limited the number of measurements to non available destination nodes in order to reduce the load on the network. But because running less measurements also meant that we will get less topology information we started to probe the default router in case of unavailable destination hosts. With this approach we could keep monitoring the nodes which were unavailable without producing unnecessary network traffic (see Subsect. 5.7.4). After a predefined number of traceroutes (default is 2) to the default router we check if the destination host is online again. If not we keep running traceroutes to the default router, otherwise the traceroutes are run to the destination host again.

5.2 Methodology

5.2.1 Geographic Distribution of the Monitoring Nodes

The Internet consists of a large number of different Internet Service Providers (ISPs), each having its own network setup and routing policy. We have chosen our monitoring nodes to be distributed worldwide so that our paths travel across a large number of different ISPs. This will allow us to investigate different routing behaviors caused by different routing policies.

Figure 5.1 depicts our monitoring nodes on a world map and the table in Fig. 5.2 gives some detail information on each node. At the end of the Master's thesis we had about 1.5 GB of raw traceroute data (109 MB when compressed as ZIP-file) collected from the beginning of May until the end of July 2006.

We organized our monitoring nodes into 3 groups. Each node in one of the three groups is located in a different country. We tried to cover about the same countries in each group. This will allow us to investigate whether we can find a pattern when comparing routing paths between certain countries. But the observed routing behavior may differ if we would have taken another node located in the same country as source or respectively as destination. Thus, we have chosen some nodes as pairs which are located in the same network. For example, *planetlab01.cs.washington.edu* is located in the same network as *planetlab03.cs.washington.edu*. Because they are located in the same network traceroutes from the same monitoring node to any of the two host will result in the same traceroute output. Vice versa, any traceroute from one of the two host to the same destination node will result in the same traceroute result. Thus, this pair of nodes is actually interchangeable. They can be treated as one single node as they are located in the same network. This will allow us to compare for example the end-to-end paths *planetlab01.cs.washington.edu* => *planetlab03.ethz.ch* and *planetlab03.cs.washington.edu* => *planetlab1.csg.unizh.ch* and to investigate whether both routing paths from Washington to Zurich have the same properties.

5.2.2 Measurement Setup

As already mentioned, we organized our monitoring nodes into 3 groups each of which consists of 7 nodes. We chosed this setup because we wanted to run full mesh traceroutes in short time intervals and the traceroutes should run simultaneously. Full mesh traceroutes provide routing information on all possible end-to-end paths between the monitoring nodes. In addition, simultaneous measurements allow for comparisons of forward and backward paths. This would not be possible if the traceroutes were conducted in a time-delayed manner. It does not make sense to compare forward and backward paths which were not measured at the same time as we could not say whether a path change has occurred in the meantime.

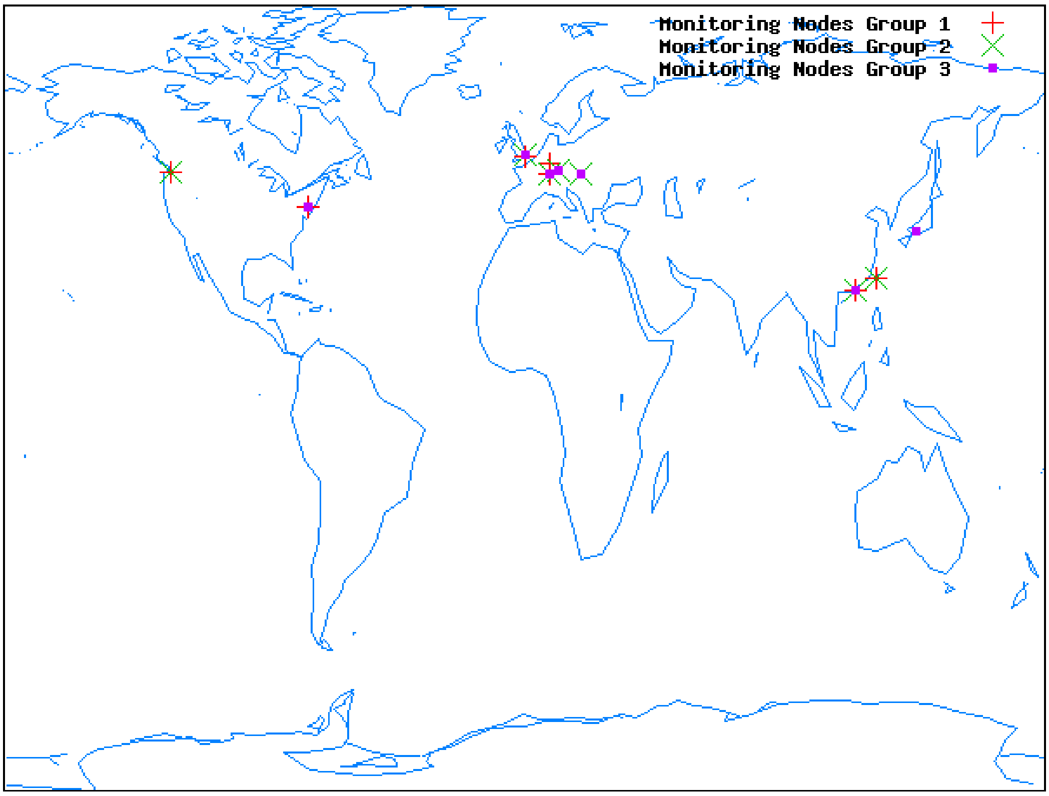

Monitoring Nodes Group 1 +
Monitoring Nodes Group 2 ×
Monitoring Nodes Group 3 ■

Figure 5.1: Our Monitoring Nodes

Running full mesh traceroutes scales in the order of N^2. For N=21 this would mean 420 traceroutes in every cycle. Because our monitoring tool should be as unintrusive as possible we need to keep the average network load at an adequate level. This means that either we reduce the number of destination nodes or we increase the timeout interval before running the next measurements. We preferred to keep the timeout interval small because the larger the timeout interval is the larger is the possibility that we miss a path change. Running full mesh traceroutes in a group of 7 nodes provides us with routing information on 7*6 = 42 end-to-end paths. With three groups this means that we collect routing information of 3*42 = 146 end-to-end paths. We believe that this order of magnitude will provide us with enough raw data for our analysis on path changes and their correlation with performance changes.

5.2.3 Measurement Interval

In order to collect the traceroute measurements, we created a cron job [13] on each of the monitoring nodes which periodically called our measurement script. Our measurement script basically fetches the list of sources and destinations, runs the standard UNIX traceroute for each source-destination pair and stores the result in a per day report file.

We began our traceroute measurement with 7 test nodes running traceroutes in full mesh in a 30 minute interval. A few days later, when we were sure that our measurements do not overburden the monitoring host or the network, we slowly increased the number of monitoring nodes to 21 nodes distributed over 9 countries. In addition, we reduced the timeout interval to 20 minutes. We reduced the timeout interval to 10 minutes after about 3 weeks. This doubled the accuracy of our analysis. On June 30 we reduced the interval for group Nr. 3 to 5 minutes. The other two groups remain running with a timeout interval of 10 minutes. The idea is that we can investigate how much accuracy we gain with an increased measurement interval.

	Hostname	Institution	City, Country	Latitude	Longitude
Group 1	planetlab01.cs.washington.edu	University of Washington	Seattle WA, USA	47.65	-122.31
	planetlab03.ethz.ch	ETH Zuerich	Zurich, Switzerland	47.38	8.53
	planetlab1.iis.sinica.edu.tw	Academia Sinica - Taiwan	Taipeh, Taiwan	25.02	121.37
	planetlab-1.imperial.ac.uk	Imperial College London - ISN	London, UK	51.10	-0.10
	planetlab2.ie.cuhk.edu.hk	Chinese University of Hong Kong	Shatin, Hong Kong	22.30	114.17
	planetlab2.rbg.informatik.tu-darmstadt.de	Darmstadt University of Technology	Darmstadt, Germany	49.53	8.40
	planetlab-5.cs.princeton.edu	Princeton	New Jersey, USA	40.35	-74.65
Group 2	plab1.cs.ust.hk	The Hong Kong University of Science and Technology	Kowloon, Hong Kong	22.30	114.17
	planet1.colbud.hu	Collegium Budapest	Budapest, Hungary	47.43	19.25
	planetlab03.cs.washington.edu	University of Washington	Seattle WA, USA	47.65	-122.31
	planetlab1.csg.unizh.ch	University of Zurich	Zurich, Switzerland	47.23	8.32
	planetlab1.lkn.ei.tum.de	Munich University of Technology	Munich, Germany	48.08	11.34
	planetlab1.nrl.dcs.qmul.ac.uk	Queen Mary, University of London	London UK	51.50	-0.10
	planetlab2.iis.sinica.edu.tw	Academia Sinica - Taiwan	Taipeh, Taiwan	25.02	121.37
Group 3	olive.ics.es.osaka-u.ac.jp[1]	University of Osaka	Osaka, Japan	34.81	135.52
	planet0.jaist.ac.jp[1]	Japan Advanced Institute of Science and Technology (JAIST)	Ishikawa, Japan	39.38	140.05
	planet2.colbud.hu	Collegium Budapest	Budapest, Hungary	47.43	19.25
	planetlab2.csg.unizh.ch	University of Zurich	Zurich, Switzerland	47.23	8.32
	planetlab2.eee.hku.hk	The University of Hong Kong	Hong Kong Island, Hong Kong	22.17	114.09
	planetlab2.lkn.ei.tum.de	Munich University of Technology	Munich, Germany	48.08	11.34
	planetlab2.net-research.org.uk[2]	University College London	London UK	51.50	-0.17
	planetlab2.nrl.dcs.qmul.ac.uk[2]	Queen Mary, University of London	London UK	51.50	-0.10
	planetlab-3.cs.princeton.edu	Princeton	New Jersey, USA	40.35	-74.65

Notes:

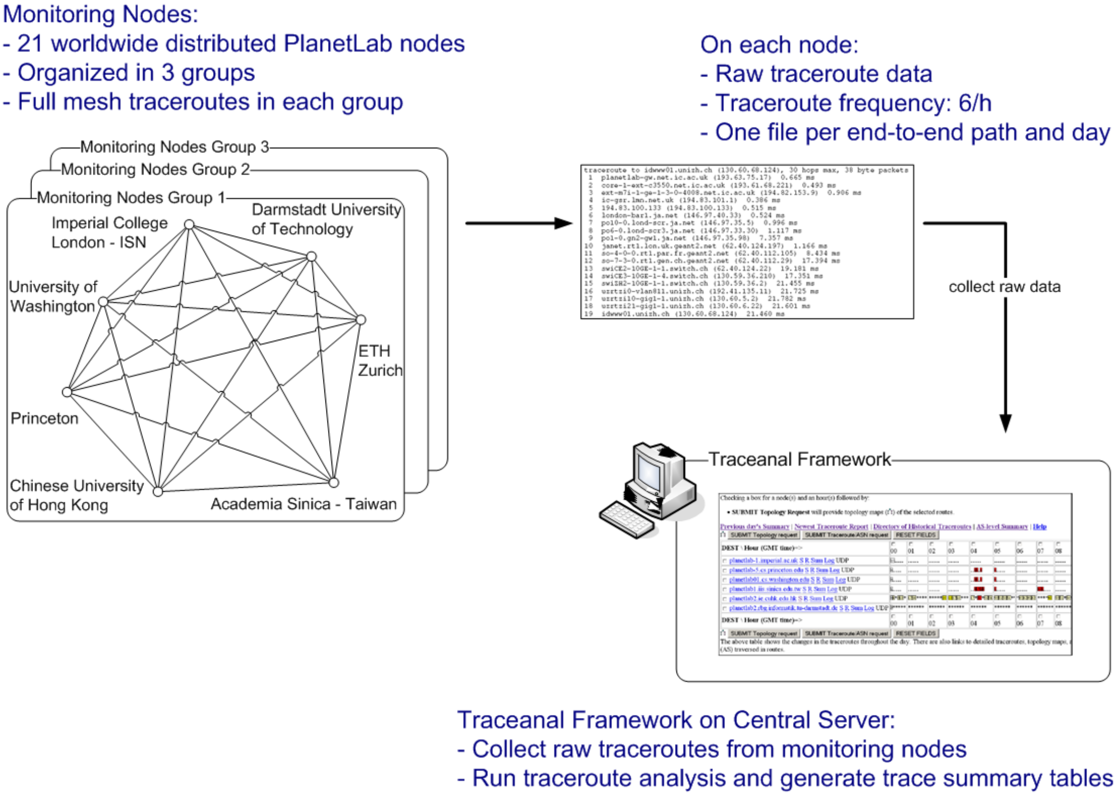
1: planet0.jaist.ac.jp was down too often and thus was replaced by olive.ics.es.osaka-u.ac.jp

2: planetlab2.net-research.org.uk was down too often and thus was replaced by planetlab2.nrl.dcs.qmul.ac.uk

Figure 5.2: Details about our Monitoring Nodes

5.2.4 Centralized Analysis of Traceroute Data

Once the traceroute data has been collected, it needs to be analyzed. To reduce the resource consumption on the nodes all measurement data was transferred to a central server an offline analysis of the data was conducted (see Fig. 5.3). There is no need to run the analysis on the monitoring nodes itself. Of course this introduces an extra overhead because the raw data needs to be transferred from the monitoring nodes to the central node. But this approach has the advantage that the resource consumption on the monitoring nodes is restricted to running traceroutes. Furthermore, traceroute data from different monitoring nodes can be compared and analyzed for correlations.



Monitoring Nodes:
- 21 worldwide distributed PlanetLab nodes
- Organized in 3 groups
- Full mesh traceroutes in each group

On each node:
- Raw traceroute data
- Traceroute frequency: 6/h
- One file per end-to-end path and day

```
traceroute to idwww01.unizh.ch (130.60.68.124), 30 hops max, 38 byte packets
 1  planetlab-gw.net.ic.ac.uk (193.63.75.17)  0.665 ms
 2  core-1-ext-c3550.net.ic.ac.uk (193.61.68.221)  0.493 ms
 3  ext-m7i-1-ge-1-3-0-4008.net.ic.ac.uk (194.82.153.9)  0.906 ms
 4  ic-gsr.lmn.net.uk (194.83.101.1)  0.386 ms
 5  194.83.100.133 (194.83.100.133)  0.515 ms
 6  london-bar1.ja.net (146.97.40.33)  0.524 ms
 7  po10-0.lond-scr.ja.net (146.97.35.5)  0.996 ms
 8  po6-0.lond-scr3.ja.net (146.97.33.30)  1.117 ms
 9  po1-0.gn2-gw1.ja.net (146.97.35.98)  7.357 ms
10  janet.rt1.lon.uk.geant2.net (62.40.124.197)  1.166 ms
11  so-4-0-0.rt1.par.fr.geant2.net (62.40.112.105)  8.434 ms
12  so-7-3-0.rt1.gen.ch.geant2.net (62.40.112.29)  17.394 ms
13  swiCE2-10GE-1-1.switch.ch (62.40.124.22)  19.181 ms
14  swiCE3-10GE-1-4.switch.ch (130.59.36.210)  17.351 ms
15  swiSH2-10GE-1-1.switch.ch (130.59.36.2)  21.455 ms
16  uzrtzi0-vlan811.unizh.ch (192.41.135.11)  21.725 ms
17  uzrtzi10-gig1-1.unizh.ch (130.60.5.2)  21.782 ms
18  uzrtzi21-gig1-1.unizh.ch (130.60.6.22)  21.601 ms
19  idwww01.unizh.ch (130.60.68.124)  21.460 ms
```

Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | AS-level Summary | Help

DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08
□ planetlab-1.imperial.ac.uk S R Sum Log UDP	[illegible]	......	......	......	......	......	......	......	......
□ planetlab-5.cs.princeton.edu S R Sum Log UDP	R....	......	......	......	[illegible]	[illegible]	......	......	......
□ planetlab01.cs.washington.edu S R Sum Log UDP	R....	......	......	......	[illegible]	[illegible]	......	......	......
□ planetlab1.iis.sinica.edu.tw S R Sum Log UDP	R....	......	......	......	[illegible]	[illegible]	......	[illegible]	......
□ planetlab2.ie.cuhk.edu.hk S R Sum Log UDP	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
□ planetlab2.rbg.informatik.tu-darmstadt.de S R Sum Log UDP	******	******	******	******	******	******	******	******	******
DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08

The above table shows the changes in the traceroutes throughout the day. There are also links to detailed traceroutes, topology maps, (AS) traversed in routes.

Traceanal Framework on Central Server:
- Collect raw traceroutes from monitoring nodes
- Run traceroute analysis and generate trace summary tables

Figure 5.3: The Data flow Between the Monitoring Nodes and the Central Server

5.3 Round Trip Time as Performance Metric

Logg et al. [38] used different performance metrics for their analysis on the correlation of changes in end-to-end performance and path changes. They used a number of tools to monitor achievable and available bandwidth, file transfer throughput and ping minimum and average round trip times (RTTs). Because our monitoring system should be as unintrusive as possible we restrict ourselves to the RTT values reported by traceroute as performance metric. Throughout the thesis we refer to end-to-end delays measured with traceroute when we are talking about end-to-end performance. It can be argued that RTT values can be misleading when used as exclusive performance metric. High RTT values can be caused by routers which down-prioritize traceroute probes and thus result in wrong assumptions about the end-to-end performance. However, in this Master's thesis we focus on the analysis of topology changes and leave it to future work to improve the performance measurements.

5.4 Own Approach for Detection of Route Changes

The traceroute tool allows the user to trace the path packets take from source to destination. As has been explained in Subsect. 3.2.1, the sequence of ICMP responses received at the monitoring host contains the IP addresses of the routers on the path from source to destination and thus yields the forwarding path IP packets are routed through.

Traceanal [36] compares two subsequent routes by comparing the IP addresses of the intermediate routers hop by hop. If the sequence of IP addresses reported by the second traceroute measurement differs in one or more hops from the first traceroute result, the path has changed at this/these hop(s) sometime between the execution of the two traceroute queries.

We have chosen the same approach as Traceanal to detect topology changes at the IP-level (besides some minor changes which we describe in Subsect. 5.5.3. The algorithm is very straightforward and not difficult to implement. However there are a number of pitfalls arising from the heterogeneous and very dynamic structure of the Internet. We discuss these problems in Sect. 5.7.

5.5 Own Approach for Analyzing Route Changes

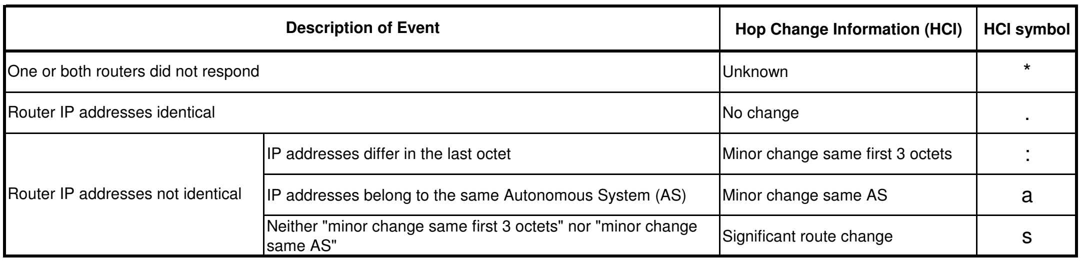
Manual debugging why a specific end-to-end path suddenly exhibits a change in the experienced performance often does not give enough information on the root cause. The basic idea of our approach is that periodical monitoring could greatly improve the debugging procedure. With sufficient historical data, it will be possible to statistically analyze the quality of certain routing paths in terms of route stability and end-to-end performance. The monitoring system could compare the routing setup during a connection problem to the routing paths previously seen for a specific end-to-end connection. In this Master's thesis we primarily use the historical data for the analysis of path changes. However, we believe that the approach of recording historical data is appropriate for any tool which is used for debugging routing problems.

We basically use the Traceanal framework to visualize and analyze historical traceroute data. We describe the chosen approach in Subsects. 5.5.1, 5.5.2 and 5.5.3.

Sometimes the user wants to analyze a path change in more detail. For that purpose we implemented a number of useful features like topology graphs and time series plots which we present in Sect. 5.6.

5.5.1 Visualization of Traceroute Change Patterns

Traceanal aims to provide an "at a glance" visualization of traceroute change patterns. Besides examining whether there was a path change or not, Traceanal also determines the type of path change and visualizes the results in a daily trace summary table. The algorithm consists of two steps. First Traceanal calculates the "Hop Change Information" (HCI) for each hop (see Table in Fig. 5.4), second it analyzes the HCIs for the complete routes and then determines the type of the path change (see Table in Fig. 5.5).



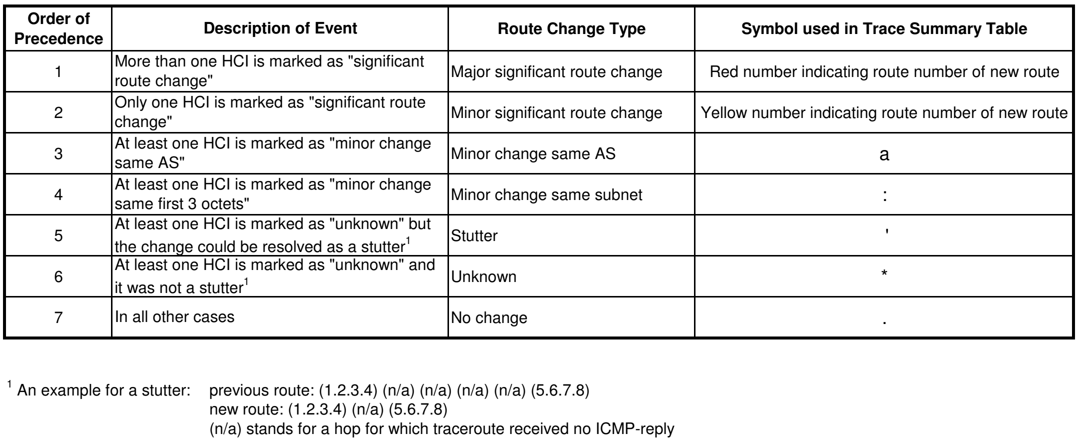
Description of Event		Hop Change Information (HCI)	HCI symbol
One or both routers did not respond		Unknown	*
Router IP addresses identical		No change	.
Router IP addresses not identical	IP addresses differ in the last octet	Minor change same first 3 octets	:
	IP addresses belong to the same Autonomous System (AS)	Minor change same AS	a
	Neither "minor change same first 3 octets" nor "minor change same AS"	Significant route change	s

Figure 5.4: Traceanal's Hop Change Information (HCI) categories

After calculating the HCI for each hop, the type of path change is determined. In the daily trace summary table, minor or no path changes are depicted using single black characters. Significant path changes are depicted using color coded characters indicating the route number of the route encountered after the topology change.

A route number uniquely identifies a sequence of IPs seen in an IP-level path. When the new route differs from the previous route Traceanal looks up whether the route was already seen before. If yes, then it determines the route number, otherwise it creates a new route number for the route.

In the trace summary table, major significant path changes are colored red, minor significant path changes are colored yellow[2]. This kind of coding scheme allows the user to spot significant path changes at a glance (see Fig 5.8). The type of path change is determined according to the order of precedence given in Fig. 5.5.



Order of Precedence	Description of Event	Route Change Type	Symbol used in Trace Summary Table
1	More than one HCI is marked as "significant route change"	Major significant route change	Red number indicating route number of new route
2	Only one HCI is marked as "significant route change"	Minor significant route change	Yellow number indicating route number of new route
3	At least one HCI is marked as "minor change same AS"	Minor change same AS	a
4	At least one HCI is marked as "minor change same first 3 octets"	Minor change same subnet	:
5	At least one HCI is marked as "unknown" but the change could be resolved as a stutter[1]	Stutter	'
6	At least one HCI is marked as "unknown" and it was not a stutter[1]	Unknown	*
7	In all other cases	No change	.

[1] An example for a stutter: previous route: (1.2.3.4) (n/a) (n/a) (n/a) (n/a) (5.6.7.8)
new route: (1.2.3.4) (n/a) (5.6.7.8)
(n/a) stands for a hop for which traceroute received no ICMP-reply

Figure 5.5: Traceanal’s route change categories

5.5.2 Distinction of Significant and Non-significant Route Changes

According to the results of Logg et al. [38] many path changes have no effect on end-to-end performance. However, they have found that there exists a correlation between path changes and performance changes. We began our analysis by investigating whether we can categorize different kinds of path changes. In other terms, is it possible to find one or more patterns which allows the categorization of path changes into two groups: those which have no effect on end-to-end performance and those which result in performance changes?

Examining which type of path changes occurred most frequently Paxson [53] found the pattern of path changes differing in only one single hop. Consecutive measurements returned routes which were equal except for an alternation at a single router. "Furthermore, the names of the router often suggested that they were administratively interchangeable" [53].

When inspecting our measurement data, we made the same observation. The most frequent type of path changes were route alternations at a single hop. As was already observed by Paxson, the names of the router suggest that they are "tightly coupled". Actually, the IPs suggested that they were in the same subnet. For example the end-to-end path having the source node *olive.ics.es.osaka-u.ac.jp* and the destination node *"plantlab2.eee.hku.hk"* exhibits a large number of path changes which are mainly due to an alternation at the fourth hop between the router *"jm20-ext-ge-010.odins.osaka-u.ac.jp"* having IP *"133.1.13.252"* and the router *"jm20-ext-ge-130.odins.osaka-u.ac.jp"* having IP *"IP 133.1.13.244"*. We assume that this is a kind of load balancing. Some ISPs apply topological load balancing in order to force a percentage of traffic onto an alternative link. Load is balanced across multiple paths in order to avoid a router being overburdened, and/or in order to potentially provide higher bandwidth to the destination. Load balancing causes a periodic intentional "path instability" and our path change detection algorithm will report a high rate of topology changes for load balanced routing paths.

[2] Note: In the original version of Traceanal minor significant path changes are colored orange. But traceroutes with an ICMP checksum error are also colored orange. We changed the color of minor significant path changes to yellow in order to avoid ambiguities.

Inspecting the round trip time (RTT) values returned by traceroute, we observed that this kind of route alternation has little consequence on the end-to-end performance as the end-to-end delay remained the same, or respectively when variations occurred, they did not exhibit any correlation to the route alternation. On the other hand, we found a different type of path change which involved multiple Autonomous Systems (ASs). These kinds of path changes usually resulted in a change in end-to-end performance.

It seems that topology changes within a single subnetwork or local to an Autonomous System have little influence on the end-to-end performance, but path changes which involve changes on the AS-level path bear the possibility for performance changes.

Based on this observation we decided to group the path changes we find in our measurement data into different categories. This provides a higher level view which allows us to examine in more detail those path changes which seem to be responsible for performance changes in the first place. We will define such path changes as "significant path changes".

We started with the path change categories distinguished by Traceanal which we have described in Subsect. 5.5.1: "Major significant path change", "Minor significant path change", "Minor change same AS" , "Minor change same subnet", "Stutter", "Unknown" and "No change". Based on our (and Paxson's) observations we expect that the categories "Minor change same AS" , "Minor change same subnet" and "Stutter" will have little consequences on the end-to-end performance. Thus, in our analysis on the correlation of path changes and changes in end-to-end performance we focused on the significant path changes. We assume that Logg et al. [38] had the same considerations when defining their categories of path changes.

5.5.3 Distinction of IP-level and AS-level Route Changes

As described in Subsect. 5.5.1 Traceanal categorizes intra-AS topology changes as a minor change, more precisely as a "Minor change same AS". And actually, in most cases intra-AS topology changes, that is path changes where the involved routers belong to the same Autonomous System (AS), seem to have little influence on the end-to-end performance. Thus, considering them as minor changes seems quite appropriate. In the trace summary table, minor changes are encoded using a single black character. Significant route changes are color coded. This coding scheme enables the user to spot significant changes at a glance. Minor route changes can be easily ignored. This approach is very useful when analyzing route changes. However, choosing the right categories is very important otherwise we may miss some significant route change patterns.

Inspecting our measurement data, we actually found an example where the chosen categories of Traceanal missed some route changes which resulted in a performance change. For example in Fig. 5.6, the intra-AS route changes between 10 a.m. and 12 a.m on May 30 and the route change at 14:45 on May 31 result in significant performance changes.

Due to this observation we changed our route change categories according to the list in Figure 5.7. In short, we considered intra-AS route changes to be significant and worth to be investigated further. A sample trace summary table generated with our categorization scheme is depicted in Fig. 5.8

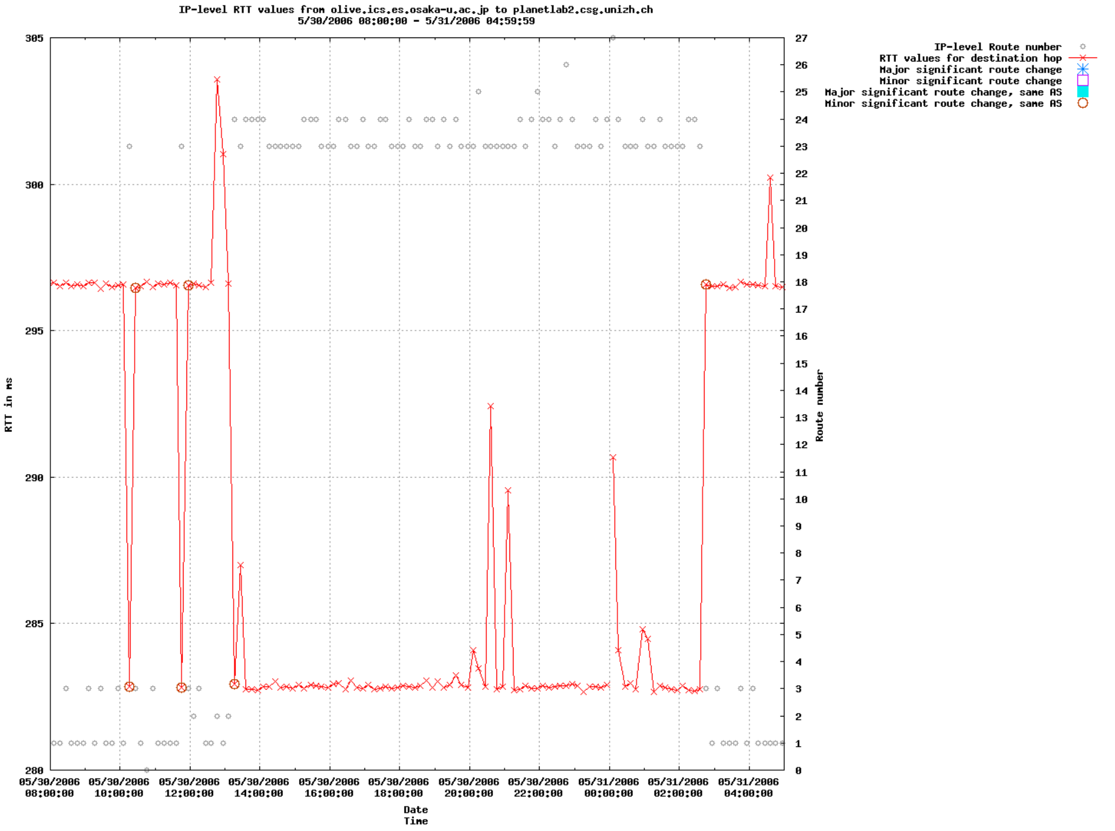
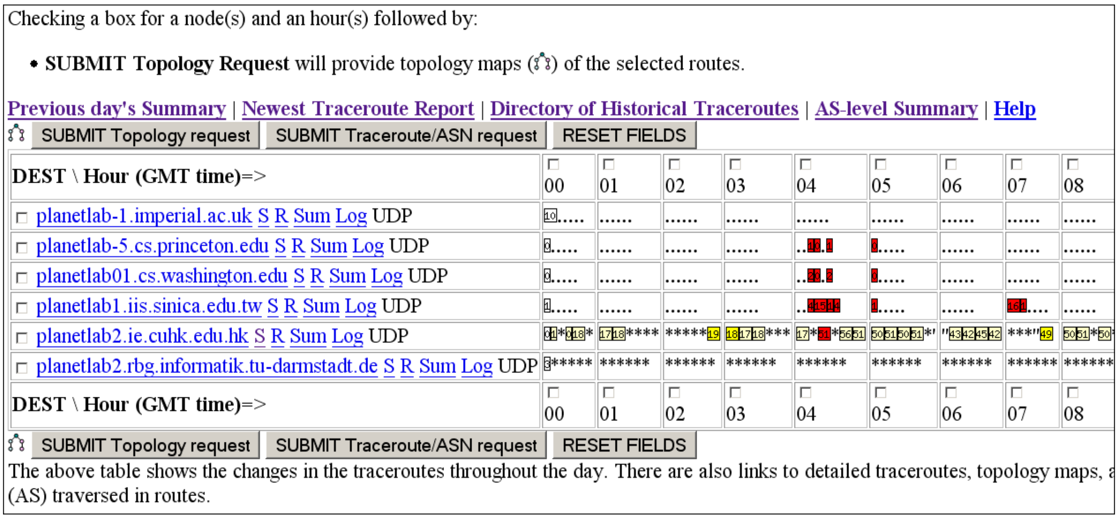


Figure 5.6: Example for intra-AS route changes which result in significant performance changes. The x-axis represents a time line, the left y-axis represents the measured round trip time (RTT) in milliseconds. The brown circles indicate intra-AS route changes. The right y-axis and the grey circles indicate the route number seen for a given measurement.

Order of Precedence	Description of Event	Route Change Type	Symbol used in Trace Summary Table
1	More than one HCI is marked as "significant route change"	Major significant route change	Red number indicating route number of new route
2	Only one HCI is marked as "significant route change"	Minor significant route change	Lightred number indicating route number of new route
3	More than one HCI was marked as "minor change same AS"	Major significant route change same AS	Yellow number indicating route number of new route
4	Only one HCI was marked as "minor change same AS"	Minor significant route change same AS	Lightyellow number indicating route number of new route
5	At least one HCI was marked as "minor change same first 3 octets"	Minor change same subnet	:
6	At least one HCI was marked as "unknown" but the change could be resolved as a stutter[1]	Stutter	'
7	At least one HCI was marked as "unknown" and it was not a stutter[1]	Unknown	*
8	In all other cases	No change	.

[1] An example for a stutter: previous route: (1.2.3.4) (n/a) (n/a) (n/a) (n/a) (5.6.7.8)
new route: (1.2.3.4) (n/a) (5.6.7.8)
(n/a) stands for a hop for which traceroute received no ICMP-reply

Figure 5.7: Our route change categories



Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps ( ) of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | AS-level Summary | Help

DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08
□ planetlab-1.imperial.ac.uk S R Sum Log UDP	10.....	......	......	......	......	......	......	......	......
□ planetlab-5.cs.princeton.edu S R Sum Log UDP	0.....	......	......	......	..10.1	0.....	......	......	......
□ planetlab01.cs.washington.edu S R Sum Log UDP	0.....	......	......	......	..20.2	0.....	......	......	......
□ planetlab1.iis.sinica.edu.tw S R Sum Log UDP	1.....	......	......	......	..41514	1.....	......	161....	......
□ planetlab2.ie.cuhk.edu.hk S R Sum Log UDP	01*018*	1718****	*****19	181718***	17*51*5651	50515051*'	"43424542	***"49	5051*50*
□ planetlab2.rbg.informatik.tu-darmstadt.de S R Sum Log UDP	3*****	******	******	******	******	******	******	******	******
DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08

The above table shows the changes in the traceroutes throughout the day. There are also links to detailed traceroutes, topology maps, a
(AS) traversed in routes.

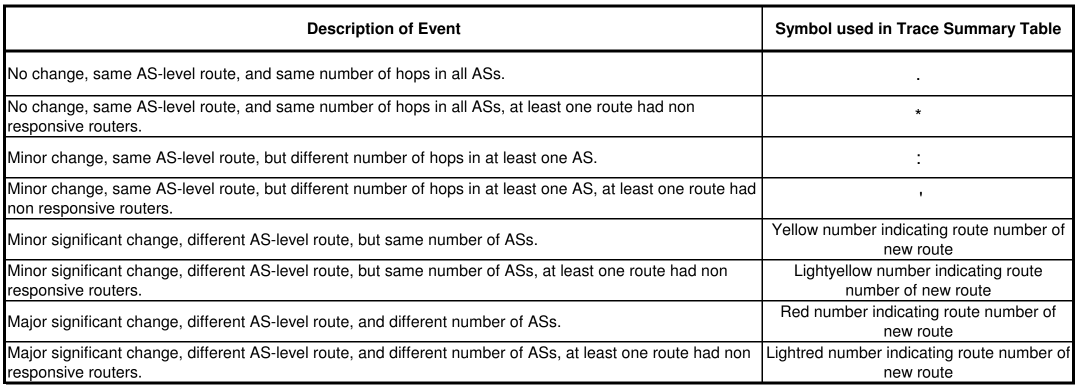
Figure 5.8: Screen shot of part of an IP-level traceroute summary table. Each row represents the traceroute change patterns for a remote destination on a single day. The destination host is indicated in the first column, the other columns represent the hours of the day. Each character encodes a route change as described earlier in this section. For example, "." means that the route did not change, a ":" indicates that the IPs of the router at which the route changed differed only in the last octet. Each route is identified by a unique number which is printed for all significant route changes. Red numbers indicate the route number after a significant change where the corresponding routers do not belong to the same Autonomous System (AS), or respectively yellow numbers indicate the route number after a significant route change where the corresponding routers belong to the same AS. The tabular organization and the color coding scheme makes it easy to detect route changes occurring simultaneously at different routes.

The problem of our categorization scheme is that for routes with a large number of intra-AS changes the trace summary table becomes very clumsy as a route number is printed for each intra-AS route change. This is an undesirable effect. On the one hand for many routes intra-AS changes have no effect on the end-to-end performance, on the other hand there exist routes for which intra-AS route changes cause a performance change and when treating intra-AS changes as non significant we will miss those changes. To solve this problem we implemented a version of Traceanal which analyzes and visualizes AS-level route changes. To distinguish both versions we call the first "Traceanal-IP" and the second we call "Traceanal-ASN".

Traceanal-ASN provides a higher level view of the route changes detected in the traceroute measurements. Like Traceanal-IP it aims to provide an at a glance visualization of traceroute change patterns. The difference is that it only reports route changes if the AS-level path has changed. The AS-level path is determined by mapping IPs to Autonomous System numbers (ASNs). Figure 5.9 illustrates the route change categories we have chosen for Traceanal-ASN and Fig. 5.10 depicts the AS-level trace summary table corresponding to the IP-level trace summary table in Fig. 5.8

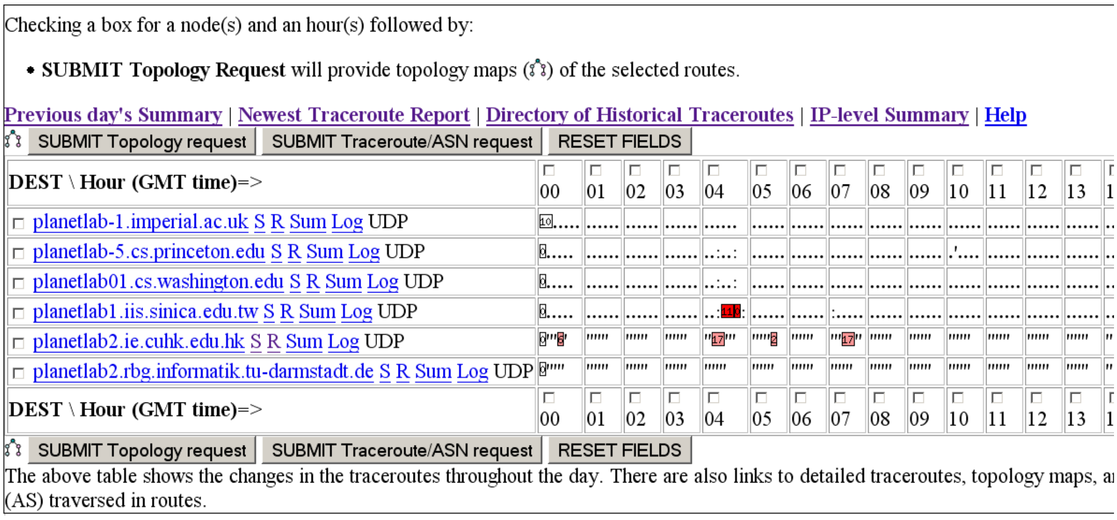
We do not merge the individual hops belonging to the same ASN for determining the AS-level route numbers because we want to investigate whether route changes which only involve a change in the number of hops inside a certain Autonomous System will have any effect on the end-to-end performance. Thus, for example, the AS-level path "1 1 1 2 2 3 4 4" will get a different route number than the AS-level path "1 1 2 2 3 4 4". Furthermore, we distinguish between route changes involving unresponsive routers and those which do not. We think this is important because the route change category cannot be determined exactly for routes which contain unresponsive routers. For example, in Fig. 5.11, at time 0:15 we cannot determine the Autonomous System number (ASN) for hops 3 and 4, either because the corresponding routers did not respond, or their IPs could not be mapped to any ASN (we describe our IP-to-AS mapping algorithm in Sect. 6.4). We do not know which Autonomous Systems this route is crossing. The route at 00:15 could be the same as the route seen at 00:05, or it could be the same as the route seen at 00:35, or it could be any other route.

If we simply ignored all unresponsive routers the route change at 00:15 would be categorized



Description of Event	Symbol used in Trace Summary Table
No change, same AS-level route, and same number of hops in all ASs.	.
No change, same AS-level route, and same number of hops in all ASs, at least one route had non responsive routers.	*
Minor change, same AS-level route, but different number of hops in at least one AS.	:
Minor change, same AS-level route, but different number of hops in at least one AS, at least one route had non responsive routers.	'
Minor significant change, different AS-level route, but same number of ASs.	Yellow number indicating route number of new route
Minor significant change, different AS-level route, but same number of ASs, at least one route had non responsive routers.	Lightyellow number indicating route number of new route
Major significant change, different AS-level route, and different number of ASs.	Red number indicating route number of new route
Major significant change, different AS-level route, and different number of ASs, at least one route had non responsive routers.	Lightred number indicating route number of new route

Figure 5.9: Our AS-level route change categories



Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps () of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | IP-level Summary | Help

DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 1
□ planetlab-1.imperial.ac.uk S R Sum Log UDP	10.....	......	......	......	......	......	......	......	......	......	......	......	......	......	..
□ planetlab-5.cs.princeton.edu S R Sum Log UDP	0.....	......	......	......	..:..:	......	......	......	......	......	.'....	......	......	......	..
□ planetlab01.cs.washington.edu S R Sum Log UDP	0.....	......	......	......	..:..:	......	......	......	......	......	......	......	......	......	..
□ planetlab1.iis.sinica.edu.tw S R Sum Log UDP	0.....	......	......	......	..:110:	......	......	:.....	......	......	......	......	......	......	..
□ planetlab2.ie.cuhk.edu.hk S R Sum Log UDP	0'''6'	''''''	''''''	''''''	''17'''	''''2	''''''	'''17''	''''''	''''''	''''''	''''''	''''''	''''''	''
□ planetlab2.rbg.informatik.tu-darmstadt.de S R Sum Log UDP	0'''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''''''	''
DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 1

The above table shows the changes in the traceroutes throughout the day. There are also links to detailed traceroutes, topology maps, a
(AS) traversed in routes.

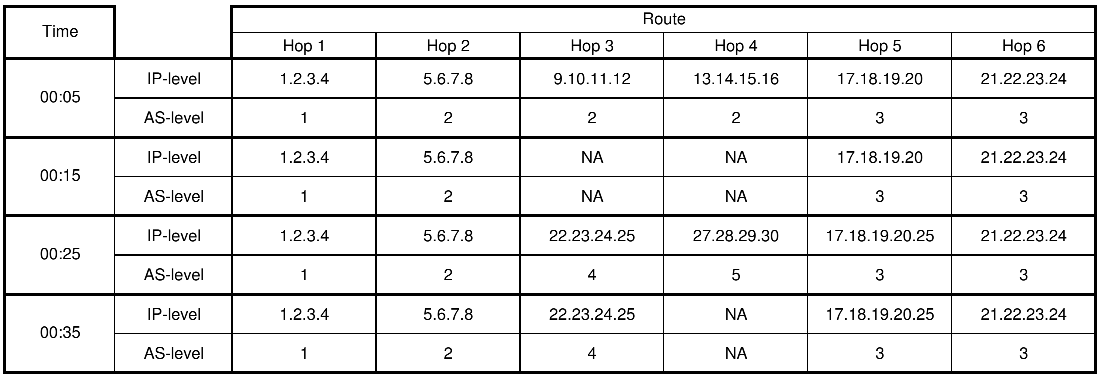
Figure 5.10: Screen shot of part of an AS-level traceroute summary table

.

as a "Minor change" since the AS-level path seems to be the same as before. This is certainly wrong as the route change must be either of type "No change" if the routers at hops 3 and 4 belong to AS 2, otherwise of type "Minor significant change" or "Major significant change" depending on whether the routers at hops 3 and 4 belong to the same AS or not.

In our approach, we have decided to treat the "NA"s as one ASN. For the route change at 00:15 this means that it would be categorized as a "Major significant change". We have no certainty that this is correct and thus we think that its the best to mark route changes involving unresponsive routers as such. When analyzing the correlation of each type of route change this uncertainty can be considered in the interpretation of the results.

Traceanal-ASN does not only provide a better overview for routing paths with a large number of intra-AS route changes, but it can also detect significant route changes which Traceanal-IP would miss. In the example explained before, Traceanal-IP would categorize the route change at 10:15 as a stutter (Fig. 5.7 gives an example of a stutter) even though it is most probably a significant route change as we have explained previously. Another example is the route change at 00:25 in Fig. 5.11. Traceanal-IP would categorize this route change as being of type "Minor change same subnet" because the hop Change Information (HCI) of hop 5 has a higher precedence over the HCIs of hops 3 and 4 which are marked as "Stutter" (the resulting HCIs for all hops are ". . * * : ."). "Minor change same subnet" is only true if the router at hop 3 belongs to AS 4 and the router at hop 4 to AS 3. In all other cases we have a significant route change. Traceanal-IP is not able to see that the route change may actually be a significant change since it only looks at hop differences. By considering the whole AS-level path and not single hop differences only, Traceanal-ASN is able to mark such a route change as one which is possibly a



Time		Route					
		Hop 1	Hop 2	Hop 3	Hop 4	Hop 5	Hop 6
00:05	IP-level	1.2.3.4	5.6.7.8	9.10.11.12	13.14.15.16	17.18.19.20	21.22.23.24
	AS-level	1	2	2	2	3	3
00:15	IP-level	1.2.3.4	5.6.7.8	NA	NA	17.18.19.20	21.22.23.24
	AS-level	1	2	NA	NA	3	3
00:25	IP-level	1.2.3.4	5.6.7.8	22.23.24.25	27.28.29.30	17.18.19.20.25	21.22.23.24
	AS-level	1	2	4	5	3	3
00:35	IP-level	1.2.3.4	5.6.7.8	22.23.24.25	NA	17.18.19.20.25	21.22.23.24
	AS-level	1	2	4	NA	3	3

Figure 5.11: Example of AS-level route changes

significant change.

5.5.4 Distributed Analysis

Traceanal's traceroute summary table depicts the routing paths from a single source to a number of remote destinations. We think that it may be more interesting to compare the routing paths from a number of monitoring hosts to a single destination. When for example a destination host is not reachable, it would be interesting to know whether this problem is only experienced by one source host or whether this destination is not reachable from any of our sources. Thus, we improved the Traceanal framework such that it does not only create traceroute summary tables from one source to a number of destination, but also the other way round, from multiple sources to one single destination.

5.6 Debugging Views

The daily trace summary tables provides an overview of traceroute change patterns. This is useful for learning how a specific route behaved throughout a specific day and to detect significant route changes. However, once a significant route change is detected it may be interesting to take a closer look. Thus, in addition to visualizing the traceroute change patterns, we also implemented some features which allows the user to look at the route changes in more detail. We present these "debugging views" in the following paragraphs.

Traceroute Summaries and Log files Traceanal provides some useful links on the trace summary web page. For example the link "Log" returns the traceroute raw data, "Sum" returns a consolidated representation of the raw data and by clicking on the node name the user can access a tabular, color coded representation of all traceroutes of a particular day.

Topology Graphs There are check boxes before the nodes and above the hour columns on the trace summary table web page (see Fig. 5.10). By checking these boxes and then clicking on the "SUBMIT Topology request" button, topology graphs of the corresponding nodes and hours are generated. The topology graphs are created using Graphviz [24] and depict the IP-level routing paths seen during the selected hours. For each node the host name, the IP and its geographic location is provided. We used MaxMind's open source APIs [47] to map IPs to a geographic location.
The Autonomous Systems (ASs) are also indicated. In an older version of Traceanal AS-level topology graphs were also available, but this was not the case for the version of Traceanal we used in this Master's thesis. This is probably due to the new version of the topology script.

Because for large topologies an AS-level view gives a better overview, we added this feature to the framework. By clicking on the "SUBMIT Traceroute/ASN request" an AS-level topology graph is created.

Gnuplots and Statistics of RTT Values The "S" link on the traceroute summary table opens the Multiple Day's Statistics web page which lists the round trip time (RTT) statistics (min, avg, max, standard deviation and jitter[3]) for the selected dates. The Multiple Day's Statistics web page also provides a functionality for plotting RTT time series with Gnuplot [22]. The time series visualize the end-to-end delay for each traceroute measurement, the route changes detected by Traceanal and the route number of each traceroute.

Route Summary Table and Route Statistics The probably most important views we implemented are the route summary table containing all routes seen for a specific end-to-end path and the route statistics web page which lists the round trip time (RTT) statistics for each route and which also provides a functionality for plotting all the RTT statistics with Gnuplot [22]. We used these plots to analyze the correlation of route changes and end-to-end performance (see Sect. 7.1).
The route summary table is available from the traceroute summary table via the link "R". The route statistics web page is generated by selecting the corresponding link on the route summary table.

5.7 Discussion of Own Approach

In this section we will discuss the challenges and limitations of the chosen approach. Furthermore, we will present some general problems of a traceroute-based approach and discuss the consequences for our algorithm.

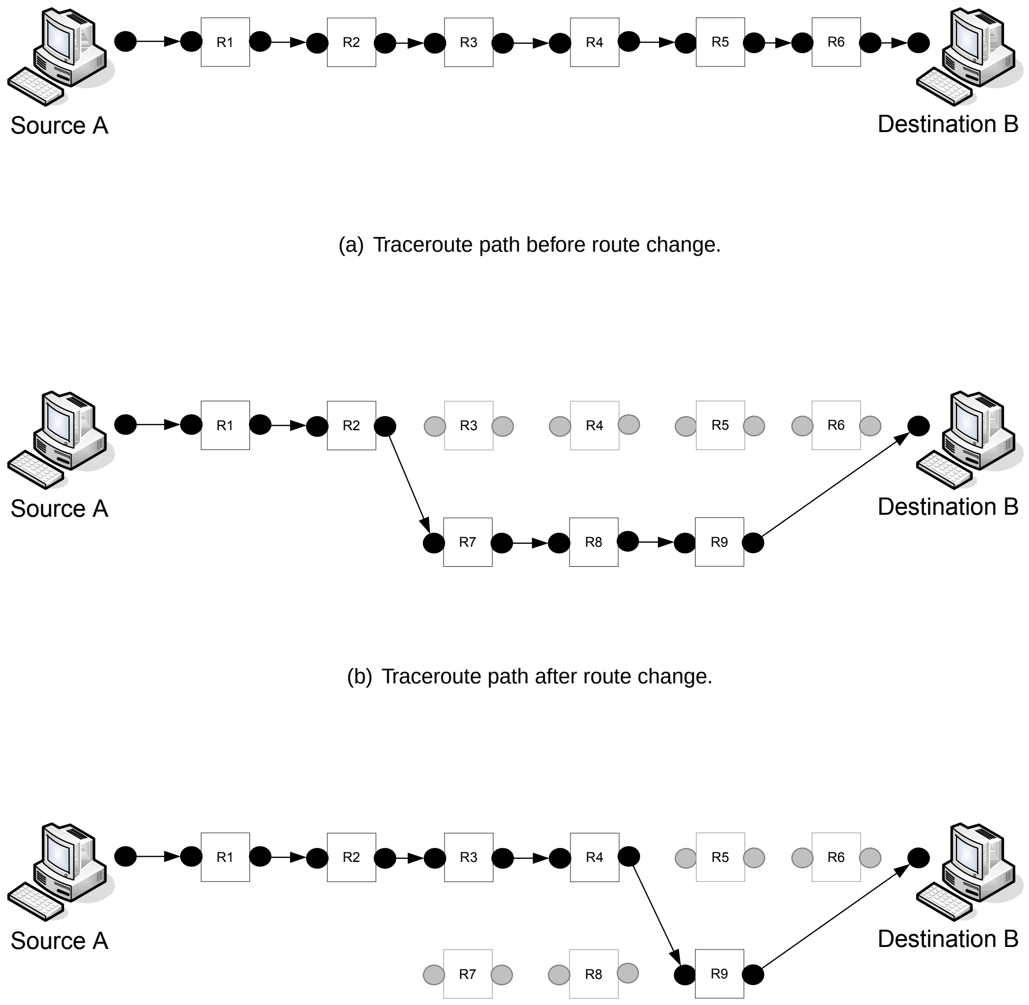
5.7.1 Completeness of our Results

The Internet is highly dynamic and at any time, any node or link may fail and/or routing paths may change. When we define the completeness of the route change detection algorithm (see Sect. 5.4) as the fraction of correctly detected topology changes, then the completeness can be increased by running as many traceroutes as possible. The more measurements we run, the more topology changes we are able to detect. In other words, we may miss a route change if we do not run traceroutes continuously. It can happen that the route changes from path P1 to path P2 at some time t1, and later, at some time t2>t1, back again from path P2 to path P1. When we made our previous traceroute measurement at some time t0<t1, and we run our next measurement not until some time t3>t2, then our algorithm will not discover the route changes from path P1 to path P2 and back again to path P1. Thus, we actually missed two route changes. By missing some route changes we may underestimate the instability of a specific end-to-end path. This problem can be alleviated by increasing the measurement frequency at the cost of generating more monitoring traffic.

5.7.2 Measurements During an Ongoing Routing Change

A severe problem of traceroute measurements is that routing changes during an ongoing measurement may return wrong forwarding paths. For example, assume that the end-to-end path between the source A and the destination B changes from path P1 = A-R1-R2-R3-R4-R5-R6-B to path P2 = A-R1-R2-R7-R8-R9-B at the time t1 (see Fig. 5.12(a) and Fig. 5.12(b)). A traceroute measurement at some time t0 with t0 < t1 will return the path P1 (we ignore preceding topology changes in this example). Vice versa, a traceroute measurement at some time t2 with t2 > t1 will return the path P2 (again, we ignore subsequent topology changes). But what path will we get when running a traceroute at the time t1?

[3] Jitter is the amount of variation in latency/response time. We refer to the definition and implementation given by PingPlotter [54] which is described at http://www.nessoft.com/kb/57



(a) Traceroute path before route change.

(b) Traceroute path after route change.

(c) Sample traceroute path during an ongoing route change.

Figure 5.12: Routing paths derived from traceroute measurements taken before, after and during a route change

It is possible that the first probing packets are forwarded along the old path P1, and, after the routers have updated their paths, the remaining packets are forwarded along the new path P2. For example, a traceroute result may return the path A-R1-R2-R3-R4-R9-B although there is actually no link between router R4 and router R9 (Fig. 5.12(c)). This happens because it takes a while until all routers have updated their routing paths after a route change, leading to an inconsistent view of the network during the path convergence time. Such inconsistencies can result in traceroute results containing wrong links or even routing loops. Such kinds of incorrect paths need to be considered when analyzing raw traceroute data. First, routing paths detected during path convergence may not correspond to real routing paths but are the result of wrong links arising from inconsistencies. Second, when applied to a sequence of traceroutes conducted during an ongoing route change, the route change detection algorithm described in Sect. 5.4 may return multiple topology changes although there was actually only one topology change.

Running more measurements does not solve this problem since a route change can occur at any time. One possible "solution" is to consider only those routes which have been seen more than only once. Paxson [53] for example only considered routing paths which have been seen at least three times. Though, it may happen that we measure the same "wrong" path more than once. . .

5.7.3 Router Aliases

Our route change detection algorithm from Sect. 5.4 reports a topology change each time the IP address for one or more hops differs from the one previously seen for that hop. Because each node in a traceroute-inferred routing path corresponds to a router interface and because a router may have multiple interfaces, the algorithm reports a topology change for a specific hop even when on the router-level there was actually no topology change for that hop. Resolving interfaces to routers generates a router-level path and helps identifying router-distinct paths.

We have presented some alias resolving techniques in Subsects. 3.2.3, 3.2.4 and 3.2.5. However, all these techniques can only discover those interfaces which are reachable from the vantage point of the monitoring host. "Source-routed path probing" alleviates this limitation, but is still not sufficient.

In our example in Fig. 5.13 path P1 and path P2 seem to diverge at router R2. Resolving router aliases however reveals that router R3 and router R7 are actually only router aliases for the same router. The same holds for router R4 and router R8. We can see that the paths P1 and P2 diverge only after router R4(=R8). Thus, on the router-level, there was only one topology change at hop 5, and not as previously thought at hop 3, 4 and 5.

5.7.4 Unresponsive Routers

If for a given hop a router does not respond (the default timeout value of traceroute is 5s) traceroute cannot determine its IP address and thus reports a "*" instead. Non-responsive intermediate routers make it difficult to compare routing paths. The easiest solution is to ignore these hops. The problem is that we cannot know if it is always the same router which does not respond or if the routing path changed and actually we are facing another non responsive router.

If the destination node does not respond to traceroute probes, this can have two reasons. Either the destination node is behind a firewall which blocks the probe packets, or the destination node is not reachable due to network problems or because it is currently not connected to the Internet.

Traceroute increases the time-to-live (TTL) field for every new probe packet until finally it receives a response from the destination. But when the destination is unreachable this will never happen. Traceroute stops after TTL=30 to avoid running forever. Figure 5.14 shows an exemplary traceroute output where the destination is not reachable.

It is meaningless to choose destination nodes which do not respond to traceroute probes when analyzing route changes as the traceroute measurement will always run out of hops.

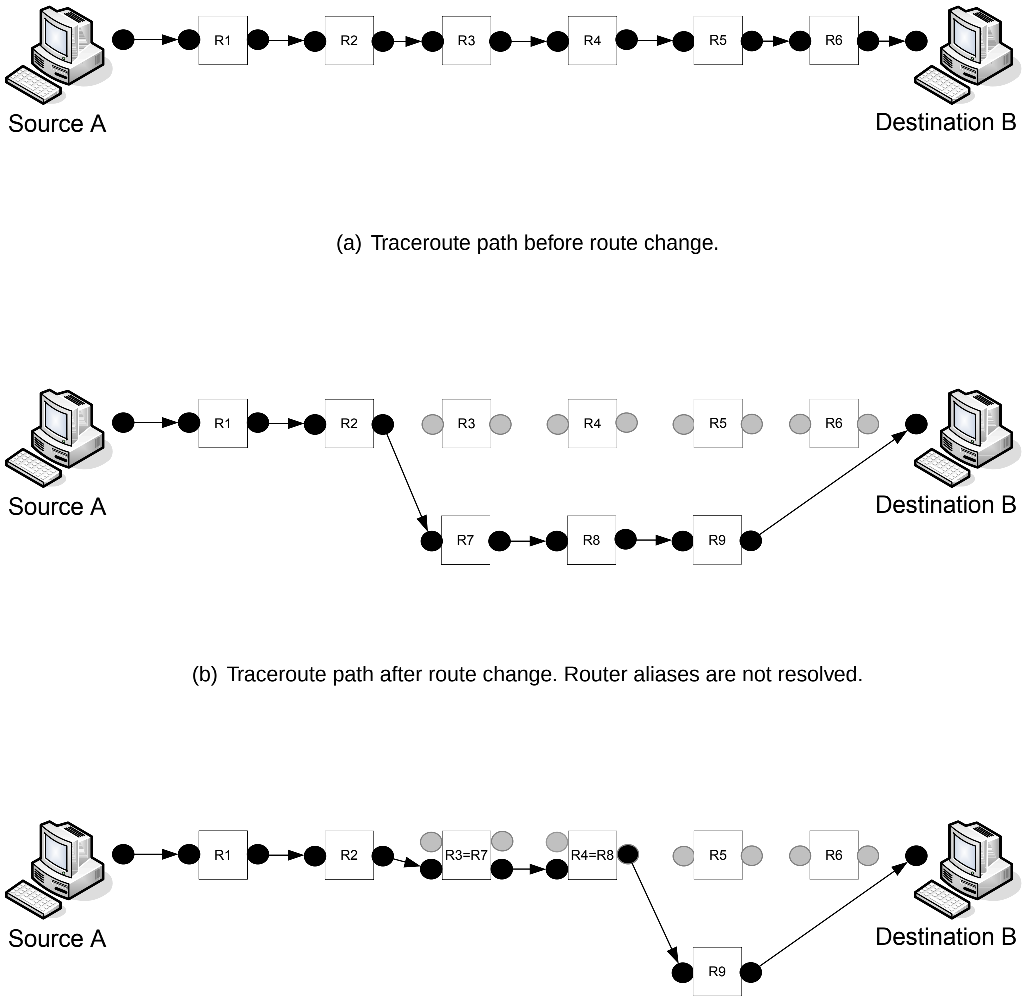
Usually a end host has a single default router and the path from the default router to the end host corresponds to the last hop detected in traceroute and usually is a static route. Thus, when a destination host is not reachable a traceroute to the default router will provide as with the same topology information with the only difference that the last hop is missing. This allows for analyzing route changes even when the destination host is not reachable. We used this approach for some of our destination nodes (see Sect. 5.1).

5.7.5 Unreachable due too many Hops

A traceroute output like the one in Fig. 5.14 usually means that the destination host is not reachable. But sometimes there are other reasons why traceroute ran out of hops.

Very few paths in today's Internet are longer than 30 hops. Thus, traceroute stops probing after TTL=30. But for example CAIDA's skitter tool (see Subsect. 3.2.4) has seen paths with more than 30 hops. When probing such a path the default settings of traceroute needs to be adjusted so it traces more than 30 hops.

Besides this very rare case of paths with more than 30 hops there exists another case where a hop seems to be unreachable only because the probe packets have a too small TTL value. Some routers send the ICMP response using the TTL in the original probing packet. Since at the router the TTL is 0 the response will never arrive at the monitoring host. When this bug



(a) Traceroute path before route change.

(b) Traceroute path after route change. Router aliases are not resolved.

(c) Traceroute path after route change where. Router aliases are resolved to routers.

Figure 5.13: Comparing routing paths with and without resolved router aliases

appears at the destination host the traceroute output can look like the one in Fig. 5.15 which we have taken out of the UNIX manual page for traceroute [75].[4]

Notice that hop 7 to 12 seem to be unresponsive routers. But actually the router at hop 13, the destination, sends the ICMP responses using the TTL of the original probing packets. These ICMP packets will time out on the way back to the monitoring host. Only when the monitoring host sends a probing packet with the TTL at least twice the path length, the ICMP responses will arrive at the monitoring host. In our example this is the case for a TTL=13 as rip.Berkeley.EDU is actually only 7 hops away. An ICMP response with TTL <= 1 is an indication for this bug. Traceroute prints a "!" after the round trip time if the TTL of the ICMP reply is <= 1.

Knowing about such bugs can help to resolve unresponsive intermediate routers when analyzing route changes. Tracerouting an intermediate router with a larger TTL value could reveal its IP address in the case where the router is experiencing that bug. When the bug appears on the destination hop the last few "*" can be ignored as they do not correspond to any (unresponsive) router. Probably the time and effort needed is too large and this procedure is only useful for debugging certain paths. However, this example shows us that we always need to keep in mind

[4] The UNIX manual page for traceroute contains some more traceroute examples for the problems described in this section.

```
traceroute to planetlab03.ethz.ch (192.33.90.197), 30 hops max, 38 byte packets
  1  planetlab-gw.net.ic.ac.uk (193.63.75.17)  0.750 ms
  2  core-1-ext-c3550.net.ic.ac.uk (193.61.68.221)  0.415 ms
  3  ext-m7i-1-ge-1-3-0-4008.net.ic.ac.uk (194.82.153.9)  0.406 ms
  4  ic-gsr.lmn.net.uk (194.83.101.1)  0.362 ms
  5  194.83.100.129 (194.83.100.129)  0.535 ms
  6  london-bar1.ja.net (146.97.40.33)  0.447 ms
  7  po10-0.lond-scr.ja.net (146.97.35.5)  0.773 ms
  8  po6-0.lond-scr3.ja.net (146.97.33.30)  1.027 ms
  9  po1-0.gn2-gw1.ja.net (146.97.35.98)  1.216 ms
 10  janet.rt1.lon.uk.geant2.net (62.40.124.197)  1.102 ms
 11  so-4-0-0.rt1.par.fr.geant2.net (62.40.112.105)  9.859 ms
 12  so-7-3-0.rt1.gen.ch.geant2.net (62.40.112.29)  17.384 ms
 13  swiCE2-10GE-1-1.switch.ch (62.40.124.22)  17.267 ms
 14  swiLS2-10GE-1-3.switch.ch (130.59.37.2)  18.129 ms
 15  swiEZ2-10GE-1-1.switch.ch (130.59.36.206)  21.550 ms
 16  rou-open-net-switch.ethz.ch (192.33.92.161)  22.053 ms
 17  *
 18  *
 19  *
 20  *
 21  *
 22  *
 23  *
 24  *
 25  *
 26  *
 27  *
 28  *
 29  *
 30  *
```

Figure 5.14: Traceroute output with unreachable destination

```
 1  helios.ee.lbl.gov (128.3.112.1)  0 ms  0 ms  0 ms
 2  lilac-dmc.Berkeley.EDU (128.32.216.1)  39 ms  19 ms  39 ms
 3  lilac-dmc.Berkeley.EDU (128.32.216.1)  19 ms  39 ms  19 ms
 4  ccngw-ner-cc.Berkeley.EDU (128.32.136.23)  39 ms  40 ms  19 ms
 5  ccn-nerif35.Berkeley.EDU (128.32.168.35)  39 ms  39 ms  39 ms
 6  csgw.Berkeley.EDU (128.32.133.254)  39 ms  59 ms  39 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  rip.Berkeley.EDU (128.32.131.22)  59 ms !  39 ms !  39 ms !
```

Figure 5.15: Traceroute output with destination unreachable due too many hops (Example taken from UNIX manual page for traceroute)

that an unresponsive router does not necessarily block traceroute probe packets but maybe only sends the ICMP response with a too small TTL value to reach the monitoring host.

5.7.6 Same IP Address for multiple Hops

For some paths the same router IP address is reported for multiple hops. This can have several reasons. In the following example (which is also from the UNIX manual page for traceroute [75]) in Fig. 5.16, hops 2 and 3 have the same IP address. This is due to a bug on the router at hop 2 which forwards packets having a TTL=0. Thus, the IP address of the router at hop 2 is unknown. Traceroute has reported the IP address of the router at hop 3 which received the probe packet with TTL=0 from the buggy router at hop 2.
Logg et al. [37] report another example where a router near Lyon, France replies to multiple TTL settings. The router blocks access to the site network for UDP probes. It first responds with an "ICMP Time Exceeded" to the first probe arriving with a TLL=1 at the router itself. But when the next traceroute probe arrives, it is blocked (cannot be forwarded) and the router responds with an ICMP destination (prohibited) unreachable message.
For our route change detection algorithm the first example is only problematic in that we do not know the IP address of the buggy router. As long as the router always behaves the same (forwards packets with TTL=0) we can detect all route changes correctly. But we do not exactly

```
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 38 byte packet
 1  helios.ee.lbl.gov (128.3.112.1)  19 ms  19 ms  0 ms
 2  lilac-dmc.Berkeley.EDU (128.32.216.1)  39 ms  39 ms  19 ms
 3  lilac-dmc.Berkeley.EDU (128.32.216.1)  39 ms  39 ms  19 ms
 4  ccngw-ner-cc.Berkeley.EDU (128.32.136.23)  39 ms  40 ms  39 ms
 5  ccn-nerif22.Berkeley.EDU (128.32.168.22)  39 ms  39 ms  39 ms
 6  128.32.197.4 (128.32.197.4)  40 ms  59 ms  59 ms
 7  131.119.2.5 (131.119.2.5)  59 ms  59 ms  59 ms
 8  129.140.70.13 (129.140.70.13)  99 ms  99 ms  80 ms
 9  129.140.71.6 (129.140.71.6)  139 ms  239 ms  319 ms
```

Figure 5.16: Traceroute with Same IP Address for multiple Hops (Example taken from UNIX manual page for traceroute)

know which paths traverse the faulty router. Of course we could assume that all paths with duplicate IP addresses for the same hops are traversing this router. However, it could also be another router with the same bug.

The second example is problematic when it occurs at some intermediate router as we are not able to infer the whole path to the destination. When it is the default router of our destination host which blocks the probe packets to the host behind it then this is not problematic as we can just traceroute to the default router as described in Sunsect. 5.7.4.

Chapter 6

Implementation

In this chapter we describe the prototype implemented during this Master's thesis, which is a tool for summarizing and analyzing the performance of end-to-end Internet paths. The prototype is based on the existing Traceanal tool. The Traceanal version developed in SLAC's (Stanford Linear Accelerator Center) Internet End-to-end Performance Monitoring - Bandwidth to the World (IEPM-BW) project [28] uses a MySQL database to store the data and monitoring configuration information. The database is also used for looking up the IP-to-AS mapping information. The Traceanal version provided to us by the IEPM-BW team was a standalone version which can be used without the IEPM-BW database. We used this standalone version as a base for our own version of Traceanal and thus we will refer to it as "the original Traceanal version".

6.1 Data Processing

As mentioned in Subsect. 5.2.4, the raw traceroute data from the monitoring nodes is collected on a central server where an offline analysis of the data is conducted. The Traceanal Framework parses the raw traceroutes, converts the raw data in a format readable by the framework and generates the round trip time (RTT) and Route Numbers files. The RTT files contain the observed RTT values for each traceroute in a consolidated format and the Route Numbers files contain all routes for a specific end-to-end path. Actually this information is redundant. All information in these files is contained in the converted traceroute log files. However, without the consolidated RTT and Route Numbers files the framework would loose a lot of performance as the traceroute data needed to be parsed over and over again. After generating the RTT and Route Numbers files, the Traceanal framework processes all observed end-to-end paths and calculates the corresponding route change patterns. At the end, a trace summary table is generated, first the IP-level trace summary and afterwards the AS-level trace summary. The whole process is illustrated in Fig. 6.1.

6.2 Class Diagram

The original Traceanal version mainly consisted of a number of different Perl and CGI scripts. We have reorganized the whole framework so that we could add an AS-level analysis to the framework without copy-pasting too much of the code. The program logic of the "traceanal.pl" script was reorganized in a Module, "Traceanal.pm". The generation of the trace summary table was subdivided into different subtasks which were assigned to different subroutines: fetchData(), convertData(), processData() and generateTraceSummary().
There already existed a "fetchData.pl" and a "convertData.pl" script but actually fetchData.pl always called convertData.pl and then returned the converted data. We improved fetchData.pl so that it only calls convertData.pl if the requested data was not yet converted, otherwise it directly returns the converted data. The "hopDiff.pl" script was also integrated into Traceanal.pm.
In a second step, we implemented "Traceanal_IP.pm" and "Traceanal_ASN.pm" which both subclass "Traceanal.pm". All IP-level specific code (fetchData(), convertData(), processData(), hopDiff()) was moved to Traceanal_IP.pm and for Traceanal_ASN.pm the corresponding subroutines were implemented. Figure 6.2 illustrates the resulting class diagram.

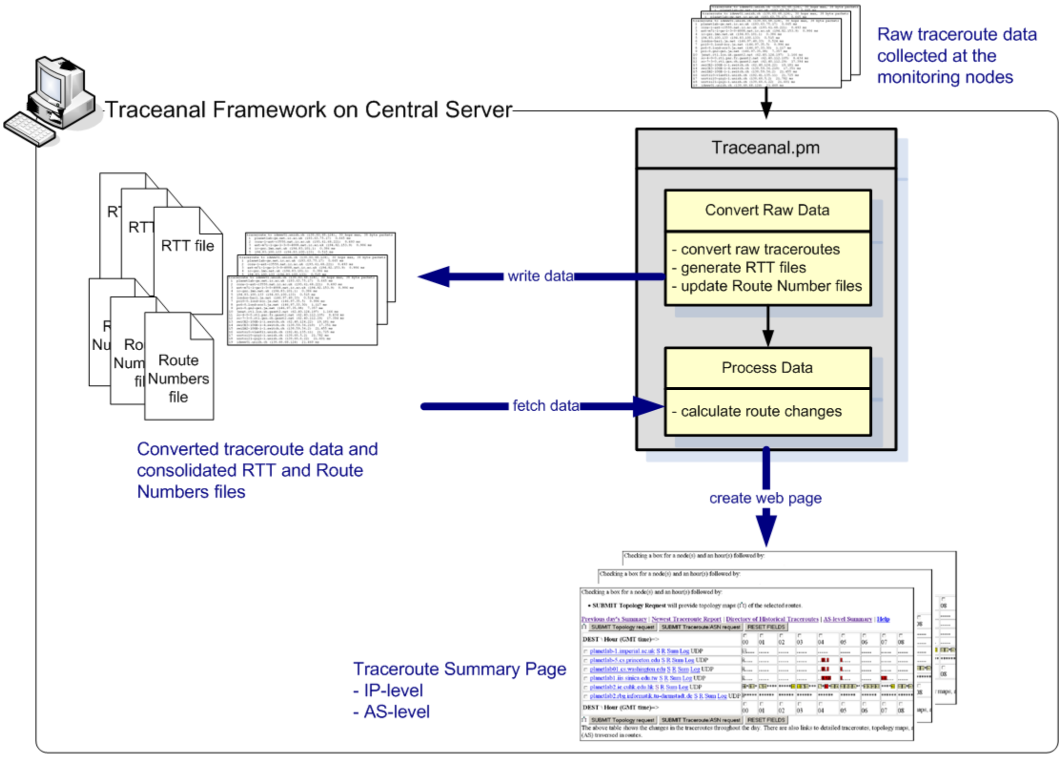

Raw traceroute data collected at the monitoring nodes
Traceanal Framework on Central Server
Traceanal.pm
Convert Raw Data
- convert raw traceroutes
- generate RTT files
- update Route Number files
Process Data
- calculate route changes
write data
fetch data
RTT file
Route Numbers file
Converted traceroute data and consolidated RTT and Route Numbers files
create web page
Traceroute Summary Page
- IP-level
- AS-level

Figure 6.1: The Data Processing

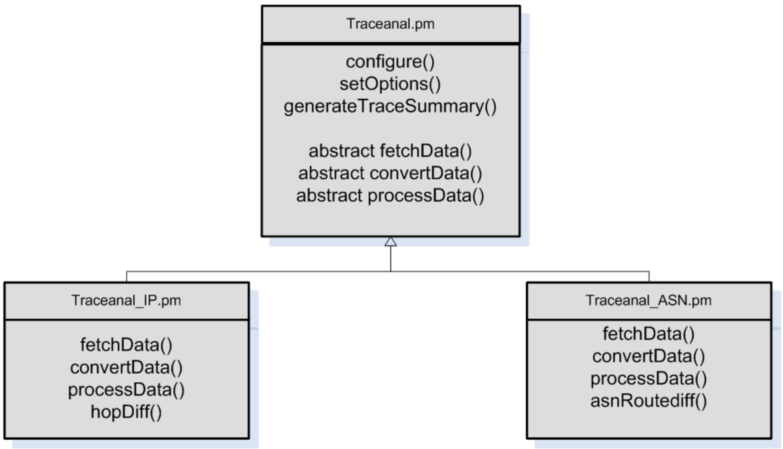
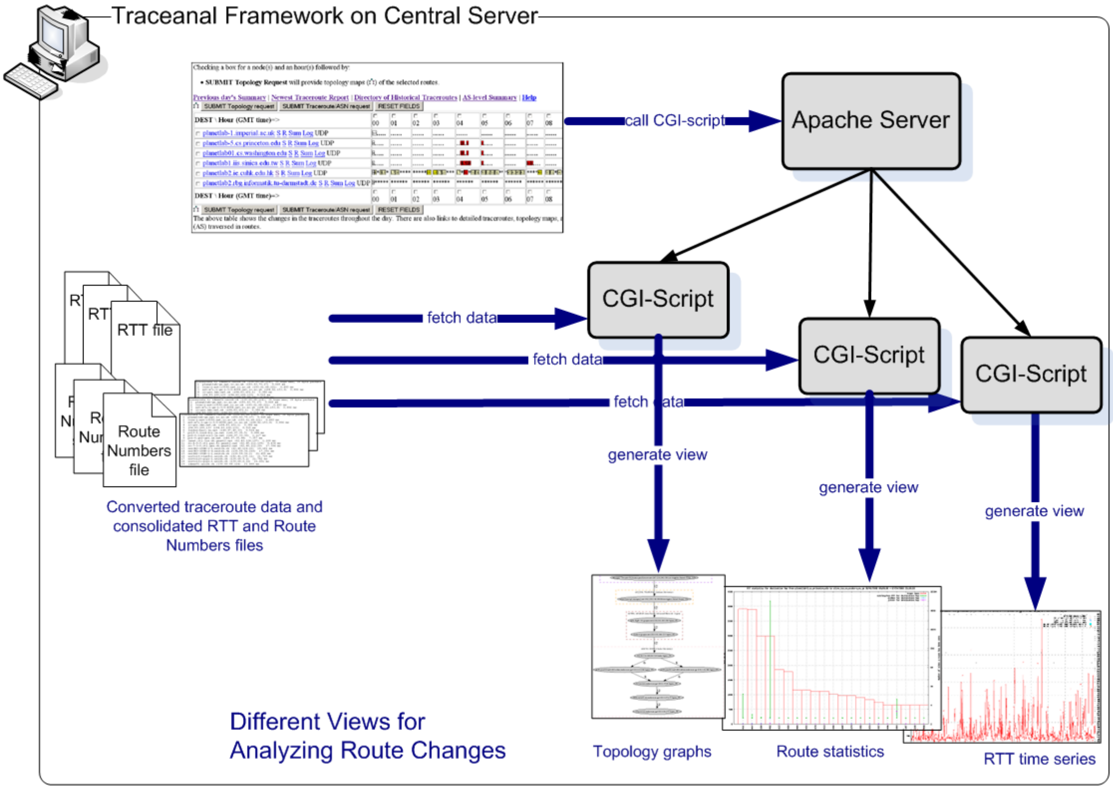


Figure 6.2: Traceanal Class Diagram

The subroutine configure() is used to set all environment variables and setOptions() can be used to change the settings like source, destination, date, etc. without creating a new Traceanal object. This enables the reuse of the same Traceanal object for generating different trace summary tables with generateTraceSummary(). This has the advantage that the caches for the IP-to-AS mapping (see Sect. 6.4) can be reused, resulting in increased performance.

6.3 The Debugging Views

In addition to the trace summary tables, we have implemented a number of debugging views for debugging and analyzing the route changes. The debugging views have been described in Sect. 5.6. All the views are implemented as CGI-scripts and thus can be generated on request. As illustrated in Fig. 6.3, the CGI-scripts use the data stored in the converted traceroute files and the consolidated RTT and Route Numbers files to generate the corresponding statistics and Gnuplots.



Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | AS-level Summary | Help

SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

DEST \ Hour (GMT time)=>	00	01	02	03	04	05	06	07	08
planetlab-1.imperial.ac.uk S R Sum Log UDP									
planetlab-5.cs.princeton.edu S R Sum Log UDP									
planetlab01.cs.washington.edu S R Sum Log UDP									
planetlab1.iis.sinica.edu.tw S R Sum Log UDP									
planetlab2.ie.cuhk.edu.hk S R Sum Log UDP									
planetlab2.rbg.informatik.tu-darmstadt.de S R Sum Log UDP	******	******	******	******	******	******	******	******	******
DEST \ Hour (GMT time)=>	00	01	02	03	04	05	06	07	08

SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

The above table shows the changes in the traceroutes throughout the day. There are also links to detailed traceroutes, topology maps, (AS) traversed in routes.

Figure 6.3: The Different Debugging Views

6.4 IP-to-AS Mapping

The Module *ASNWhoIs.pm* contains the whole logic for mapping IPs to Autonomous System numbers (ASNs). In the original Traceanal version the module uses the DNS-Zone *asn.routeviews.org* of the Route Views project to lookup the origin Autonomous System (AS) for a given IP. Because for some of the IPs in our measurement data Route Views could not determine the origin AS, we started to evaluate a number of different IP-to-AS mapping services and found that the DNS-Zone *whois.cymru.com* provided by Team Cymru [71] can resolve more IPs to origin ASs than Route Views. Thus, we improved the module *ASNWhoIs.pm* such that it first queries *whois.cymru.com*. In case no record is found for a given IP, *ASNWhoIs.pm* will query also *asn.routeviews.org*. And if still it cannot resolve the IP to its origin AS, it also queries *ripe.whois.net*, an IP-to-AS mapping service provided by RIPE's Routing Information Service (RIS)[60]. When non of the three services returns a mapping, *ASNWhoIs.pm* will return a "NA" as Autonomous System number.
To speed up the large number of WHOIS lookups we needed for our approach, the WHOIS lookup performance was improved by introducing two caches: one cache which holds all IP-to-AS mappings and another cache which holds additional information for each Autonomous System such as the Autonomous System number, the name of the AS and a description. Thanks to this improvement the generation of the traceroute summary tables is about 10 times faster than before.

Chapter 7

Discussion of Results and Evaluation

Chapters 5 and 6 described the prototype implemented during this Master's thesis. With the help of this prototype we analyzed topology changes on different end-to-end paths in order to find a correlation between topology changes and performance changes. We actually found some examples which exhibit a correlation between topology changes and performance changes. However, there are also cases for which we could not determine whether there is a correlation or not. Section 7.1 summarizes and discusses the observations made when analyzing the data measured from the beginning of May 2006 until the end of July 2006.

Because we could not find a correlation in general, we looked for an approach which allows us to compare and analyze different routes and end-to-end paths. Section 7.2 describes this approach and the corresponding results which actually give an explanation on the observed or respectively not observed correlation found in the measurement data.

The quality measures described in Sect. 7.2 are not only useful for our analysis on the correlation of route changes and performance changes but enables also the comparison of the Internet connectivity of different ISPs. An example is given in Subsect. 7.2.3.

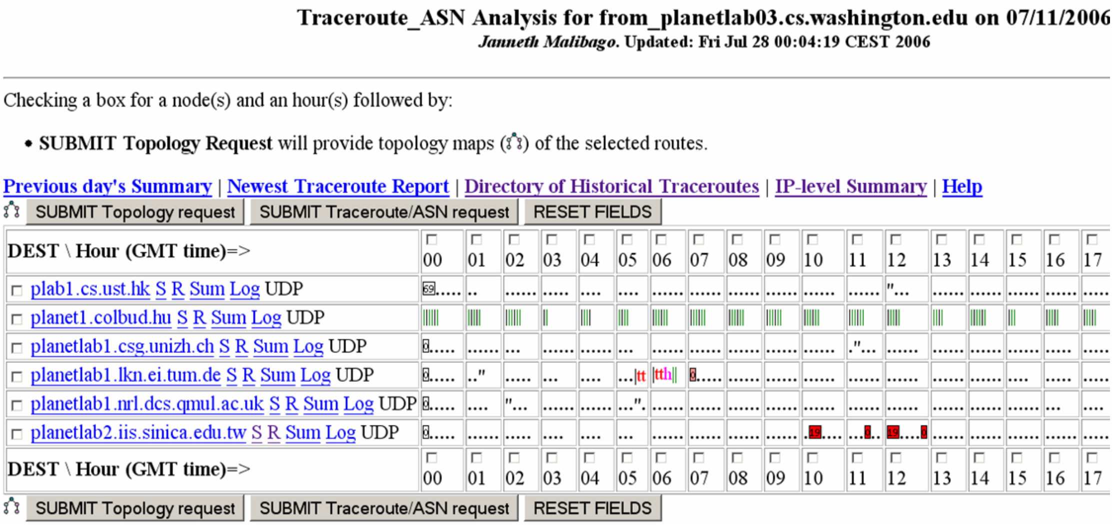
7.1 Correlation of Topology Changes and Performance Changes

7.1.1 Observations

After analyzing the traceroute data measured over the last three months we have found that it is difficult to find a correlation between topology changes and performance changes in general. The difficulty is caused by large variations among the different end-to-end paths. We can coarsely distinguish two types of end-to-end paths, those which are very stable and exhibit almost no route changes and usually only have 1-2 dominant paths, and those which exhibit a very large number of route changes and also a large number of different routes. We describe these two types in the following two paragraphs.

Type 1: Stable end-to-end paths We observed a large number of end-to-end paths which are stable for most of the time. Usually they have only 1-2 dominant routes which remain for hours and even days. Significant topology changes occur very rarely. Once a topology change occurs, it often involves a performance change. We observed a high correlation of topology changes and performance changes for this type of end-to-end paths. The round trip time (RTT) values observed for each route seem to stay within a small range of values. Thus, it is possible to see a correlation between a rise or reduction in the measured RTT and the observed topology changes. Figures 7.1 to 7.3 illustrate an example of a stable end-to-end path.

As can be seen in the last row in Fig. 7.1, there were some significant route changes between 10 a.m and 12 a.m. for the end-to-end path between the monitoring node in Washington and the monitoring node in Taiwan on July 11, 2006. Before 10 o'clock always the same route was reported for this end-to-end path, more precisely, the route with number 0. At 10 o'clock there was a route change from route 0 to route 19. At about 11:30 a.m. the end-to-end path changed back again to route 0 for a short time before it changed again to route 19. Shortly before 13:00 p.m. there was another route change to route 0 which then remained for the rest of the day.



Traceroute_ASN Analysis for from_planetlab03.cs.washington.edu on 07/11/2006
Janneth Malibago. **Updated: Fri Jul 28 00:04:19 CEST 2006**

Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps (⛬) of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | IP-level Summary | Help

⛬ SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16	□ 17
□ plab1.cs.ust.hk S R Sum Log UDP	69.....	..	......	.....	....	....	......	......	......	......	......	......	"...	......	......	......	......	.....
□ planet1.colbud.hu S R Sum Log UDP	|||||	||||||	|||||	||	|||	||||	||||||	|||||	||||||	|||||	||||||	|||||	||||||	|||	||||||	||	|||||	|||||
□ planetlab1.csg.unizh.ch S R Sum Log UDP	0.....	......	...	......	......	...	......	......	......	......	......	."...	......	......	......	......	.....	.....
□ planetlab1.lkn.ei.tum.de S R Sum Log UDP	0.....	.."	.....	...	....	...|tt	|tth||	0.....	......	......	......	......	......	......	......	....	.....	.....
□ planetlab1.nrl.dcs.qmul.ac.uk S R Sum Log UDP	0.....	....	"...	......	......	...".	......	......	......	......	......	......	......	......	......	......	...	....
□ planetlab2.iis.sinica.edu.tw S R Sum Log UDP	0.....	......	.....	....	....	...	......	......	......	......	.19....	...0..	19....0	......	......	......	......	.....
DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16	□ 17

⛬ SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

Figure 7.1: AS-level Trace Summary Table

Inspecting the round trip time (RTT) time series given in Fig. 7.2 it can be seen that the topology changes reported by Traceanal correlate with the changes in the measured round trip time. At 10 o'clock the RTT rises from 100 ms to 250 ms. Exactly to the time where the end-to-end path changes from route 0 to route 19. At about 11:30, the RTT shortly drops to 100 ms, exactly for the same period of time as the end-to-end path changes from route 19 to route 0. The same holds for the last topology change shortly before 13:00 p.m. The RTT level drops from 250 ms to 100ms and, except for some outliers, this level remains for the rest of the day.

The topology graph for route 0 and route 19 which is depicted in Fig. 7.4 gives an explanation for these performance changes. Route 0 runs directly from the USA to Taiwan, whereas route 19 makes a few detours via a few other ISPs in the USA before it finally reaches Taiwan. This explains why route 19 causes much higher round trip times than route 0. The route statistics in Fig. 7.3 support this observation. The average RTT for route 0 is much lower than the average RTT of route 19. The error bars in Fig.7.3 show that most of the routes of the end-to-end path between the monitoring node in Washington and the monitoring node in Taiwan exhibit low variations and thus are very stable. Fig.7.3 also shows that route 0 is the dominant route. Because the routes which occur the most show low variations, topology changes from one route to the other can be correlated with changes in the measured round trip time as the RTT for each route falls into a small range of possible values. Note however that it is probably more difficult (or even not possible) to find a correlation when the end-to-end path changes for example from route 2 to route 10 as these two routes exhibit almost similar round trip times in average.

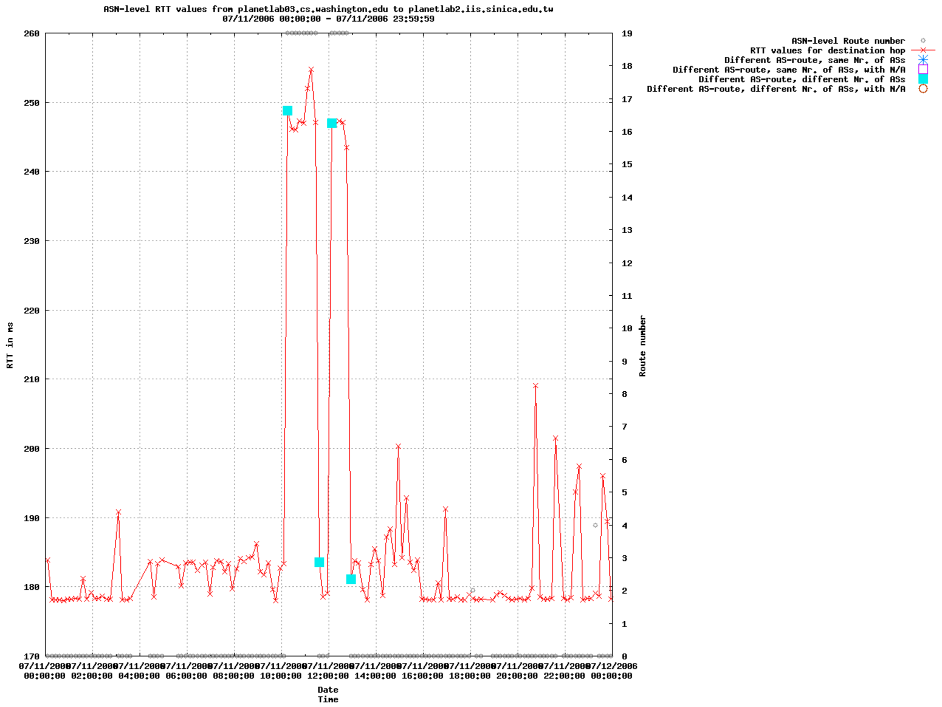


Figure 7.2: AS-level RTT Time Series. The x-axis represents a time line, the left y-axis represents the measured round trip time (RTT) in milliseconds. The blue squares indicates significant route changes of the category "Major significant change, different AS-level route, and different number of ASs" (see Fig. 5.9). The right y-axis and the gray circles indicate the AS-level route number seen for a given measurement.

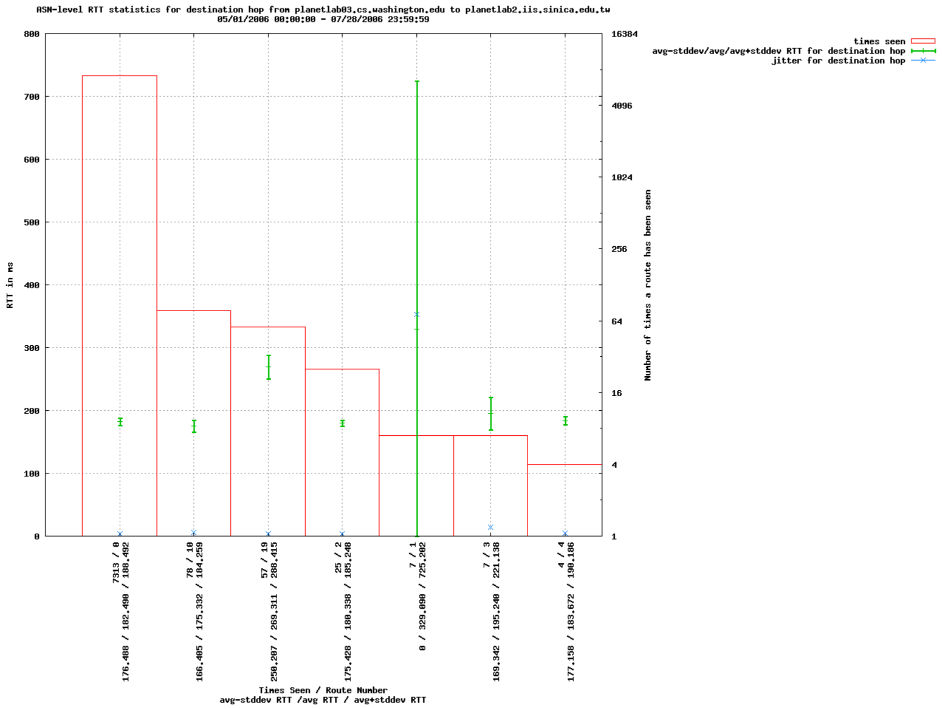


Figure 7.3: AS-level Route Statistics. The x-axis represents all routes seen for a given end-to-end path sorted by their occurrence. The left y-axis and the error bars indicate the average RTT time and the standard deviation measured for the last three months for each route. The right y-axis and the histograms represent the number of times a route has been seen (given in a logarithmic scale).

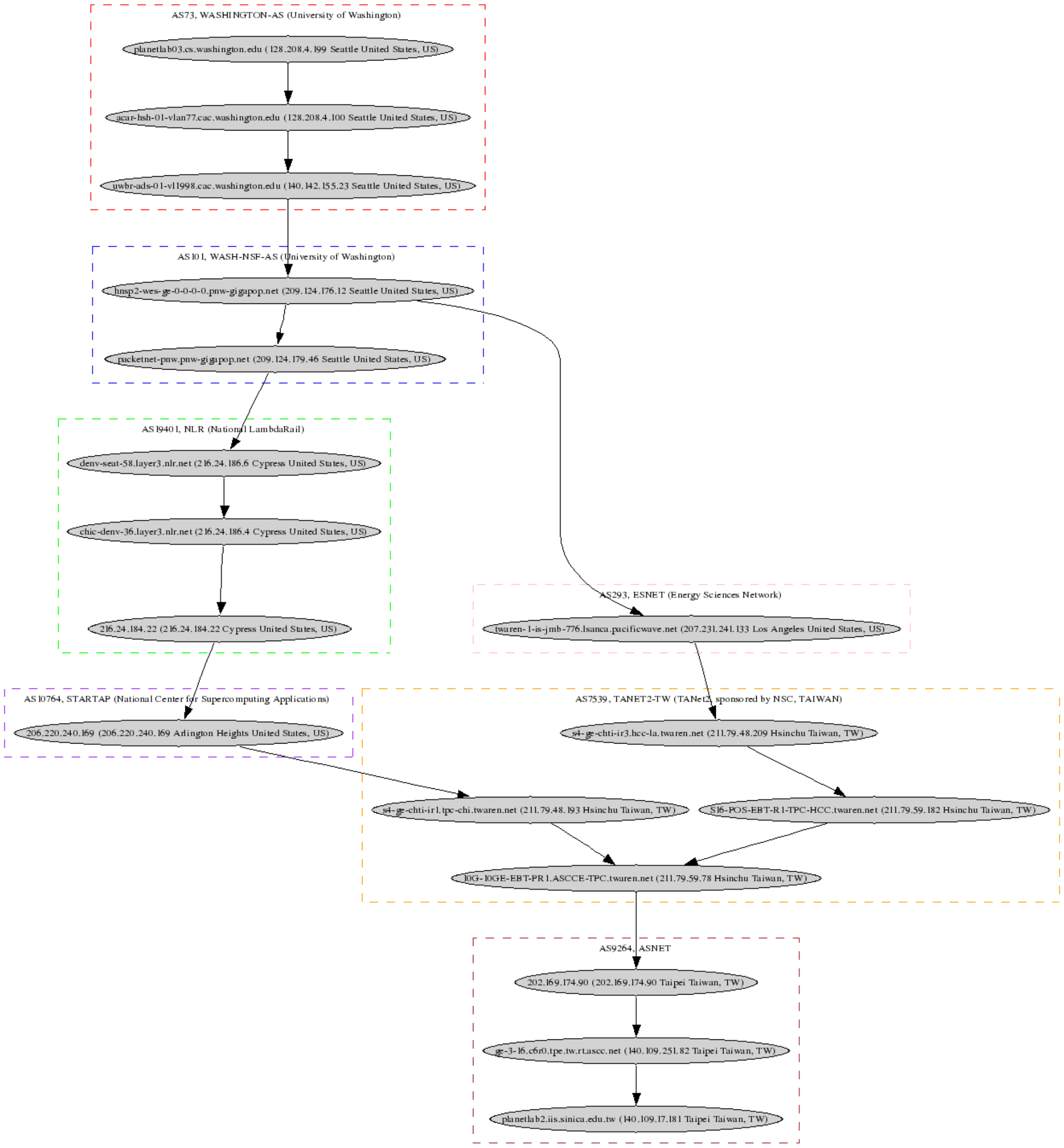
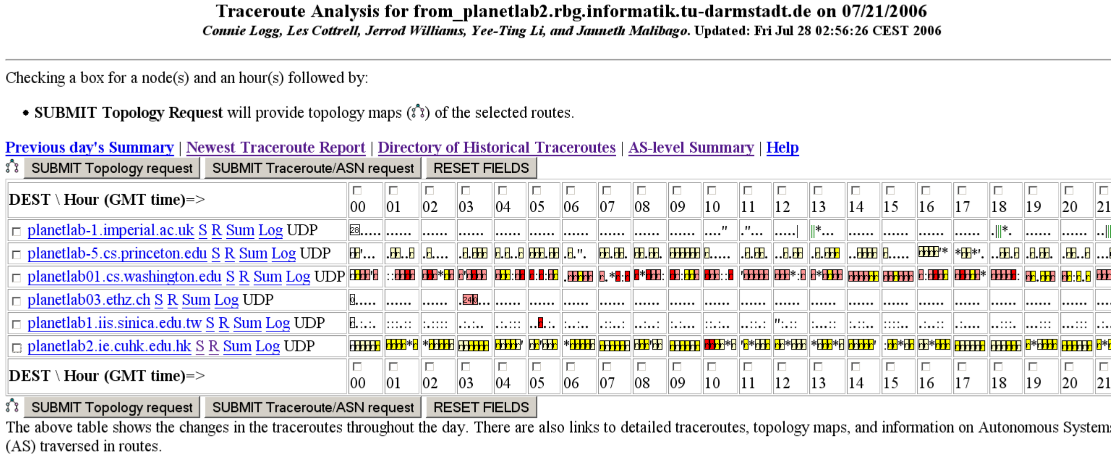

AS73, WASHINGTON-AS (University of Washington)
planetlab03.cs.washington.edu (128.208.4.199 Seattle United States, US)
acar-hsh-01-vlan77.cac.washington.edu (128.208.4.100 Seattle United States, US)
uwbr-ads-01-vl1998.cac.washington.edu (140.142.155.23 Seattle United States, US)
AS101, WASH-NSF-AS (University of Washington)
hnsp2-wes-ge-0-0-0-0.pnw-gigapop.net (209.124.176.12 Seattle United States, US)
packetnet-pnw.pnw-gigapop.net (209.124.179.46 Seattle United States, US)
AS19401, NLR (National LambdaRail)
denv-seat-58.layer3.nlr.net (216.24.186.6 Cypress United States, US)
chic-denv-36.layer3.nlr.net (216.24.186.4 Cypress United States, US)
216.24.184.22 (216.24.184.22 Cypress United States, US)
AS293, ESNET (Energy Sciences Network)
twaren-1-is-jmb-776.lsanca.pacificwave.net (207.231.241.133 Los Angeles United States, US)
AS10764, STARTAP (National Center for Supercomputing Applications)
206.220.240.169 (206.220.240.169 Arlington Heights United States, US)
AS7539, TANET2-TW (TANet2, sponsored by NSC, TAIWAN)
s4-ge-chti-ir3.hcc-la.twaren.net (211.79.48.209 Hsinchu Taiwan, TW)
s4-ge-chti-ir1.tpc-chi.twaren.net (211.79.48.193 Hsinchu Taiwan, TW)
S16-POS-EBT-R1-TPC-HCC.twaren.net (211.79.59.182 Hsinchu Taiwan, TW)
10G-10GE-EBT-PR1.ASCCE-TPC.twaren.net (211.79.59.78 Hsinchu Taiwan, TW)
AS9264, ASNET
202.169.174.90 (202.169.174.90 Taipei Taiwan, TW)
ge-3-16.c6r0.tpe.tw.rt.ascc.net (140.109.251.82 Taipei Taiwan, TW)
planetlab2.iis.sinica.edu.tw (140.109.17.181 Taipei Taiwan, TW)

Figure 7.4: Topology Graph. Each node represents a router seen on the end-to-end path. The dashed boxes surrounding the nodes represent the boundaries of the Autonomous Systems (ASs) the routers belong to.

Type 2: Unstable End-to-end paths Besides the usually stable end-to-end paths we have also observed paths which are very unstable. They exhibit a very large number of topology changes, once per hour and even more. We assume that these kinds of topology changes are caused by bad routing policies which result in instable routing paths. The large number of topology changes in such unstable paths makes it difficult to find a general correlation of topology changes and performance changes. For example, the last row in the IP-level trace summary table in Fig. 7.5 shows multiple significant route changes of the category "Major significant route change same AS" (see Fig. 5.7) for the end-to-end path from Darmstadt to Hong Kong. By inspecting the round trip time (RTT) time series given in Fig. 7.6 it can be seen that the observed round trip time values seem to alternate between two levels, the first level at about 280 ms and the second level at about 320 ms. This indicates that there is a correlation, but due to the large number of topology changes it is difficult to give a profound statement.



Traceroute Analysis for from_planetlab2.rbg.informatik.tu-darmstadt.de on 07/21/2006

Connie Logg, Les Cottrell, Jerrod Williams, Yee-Ting Li, and Janneth Malibago. Updated: Fri Jul 28 02:56:26 CEST 2006

Checking a box for a node(s) and an hour(s) followed by:

- **SUBMIT Topology Request** will provide topology maps () of the selected routes.

Previous day's Summary | Newest Traceroute Report | Directory of Historical Traceroutes | AS-level Summary | Help

SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16	□ 17	□ 18	□ 19	□ 20	□ 21
□ planetlab-1.imperial.ac.uk S R Sum Log UDP	28.....	......	......	......	......	......	......	......	......	......	....″	.″...	.....|	||*...	......	......	......	......	.||*.	......	......	..||
□ planetlab-5.cs.princeton.edu S R Sum Log UDP	▯▯'...	.▯▯..▯	.▯....	.▯.▯▯▯	▯.▯..▯	▯▯▯.▯▯	.▯.″.	▯▯.▯▯.	▯▯.▯▯.	▯▯▯▯▯▯	▯.....	.▯.▯▯.	..▯.▯▯	.▯.▯▯▯	..▯▯▯▯	▯.....	▯▯▯▯'*	*▯▯*'.	..▯▯.▯	.▯.▯▯.	.▯▯..▯	...▯
□ planetlab01.cs.washington.edu S R Sum Log UDP	▯▯▯▯'▯	::▯▯▯▯	▯▯▯*▯▯	▯'▯▯▯▯	▯▯▯:▯▯	▯:▯:▯▯	.▯▯▯▯▯	▯.*▯:▯	▯*▯▯▯:	▯▯:▯▯▯	▯▯▯::▯	'▯▯▯▯▯	▯▯▯*:▯	▯*▯▯▯▯	▯▯▯▯▯▯	▯▯▯▯▯▯	▯:▯▯▯▯	▯▯▯▯▯*	▯▯▯▯▯:	▯▯.▯▯▯	▯▯:▯.▯	▯▯▯
□ planetlab03.ethz.ch S R Sum Log UDP	▯.....	......	......	.240...	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	...
□ planetlab1.iis.sinica.edu.tw S R Sum Log UDP	▯.:.:.	:::.::	:.::::	.:.:..	.:.:::	..▯.:.	:..:..	.::..:	..:..:	:.:...	::.:..	..::.:	″:..::	:...::	:.::..	::::..	.:..::	.:....	..:::.	...:::	...:::	.:..
□ planetlab2.ie.cuhk.edu.hk S R Sum Log UDP	▯▯▯▯▯▯	▯▯▯▯*▯	*▯▯▯▯▯	▯▯▯▯▯▯	▯▯▯▯▯'	▯▯'▯▯▯	*▯▯▯▯▯	▯▯▯▯▯▯	▯▯▯'▯▯	▯▯▯▯▯▯	▯▯▯▯*▯	'▯*▯▯▯	▯▯*▯▯▯	*▯▯▯*▯	▯▯▯▯▯'	:▯▯*▯▯	▯▯*▯▯▯	▯▯▯▯▯▯	▯▯▯▯▯▯	▯*▯▯▯▯	▯▯▯▯▯▯	▯*▯
DEST \ Hour (GMT time)=>	□ 00	□ 01	□ 02	□ 03	□ 04	□ 05	□ 06	□ 07	□ 08	□ 09	□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16	□ 17	□ 18	□ 19	□ 20	□ 21

SUBMIT Topology request | SUBMIT Traceroute/ASN request | RESET FIELDS

The above table shows the changes in the traceroutes throughout the day. There are also links to detailed traceroutes, topology maps, and information on Autonomous System (AS) traversed in routes.

Figure 7.5: IP-level Trace Summary Table

The route statistics in Fig. 7.7 show why it is that difficult to find a correlation. The end-to-end path from Darmstadt to Hong Kong reveals a large number of different routes. Furthermore, a large number of the observed routes have similar round trip times in average. In addition, there are also many routes which show large variations in the RTT. All this makes it nearly impossible to determine whether there is a correlation between route changes and performance changes or whether another problem is responsible for the variations in the observed RTT.

The route statistics on the AS-level given in Fig. 7.8 show that on the AS-level there are much less different routes than on the IP-level. This would facilitate the analysis on the correlation of route changes and performance changes. However, as can be seen in the time series plot in Fig. 7.9, the AS-level analysis is not applicable for this special example as there are no route changes detected at the AS-level. For this example we must conclude with the statement that there is probably a correlation but it is difficult to prove this assumption.

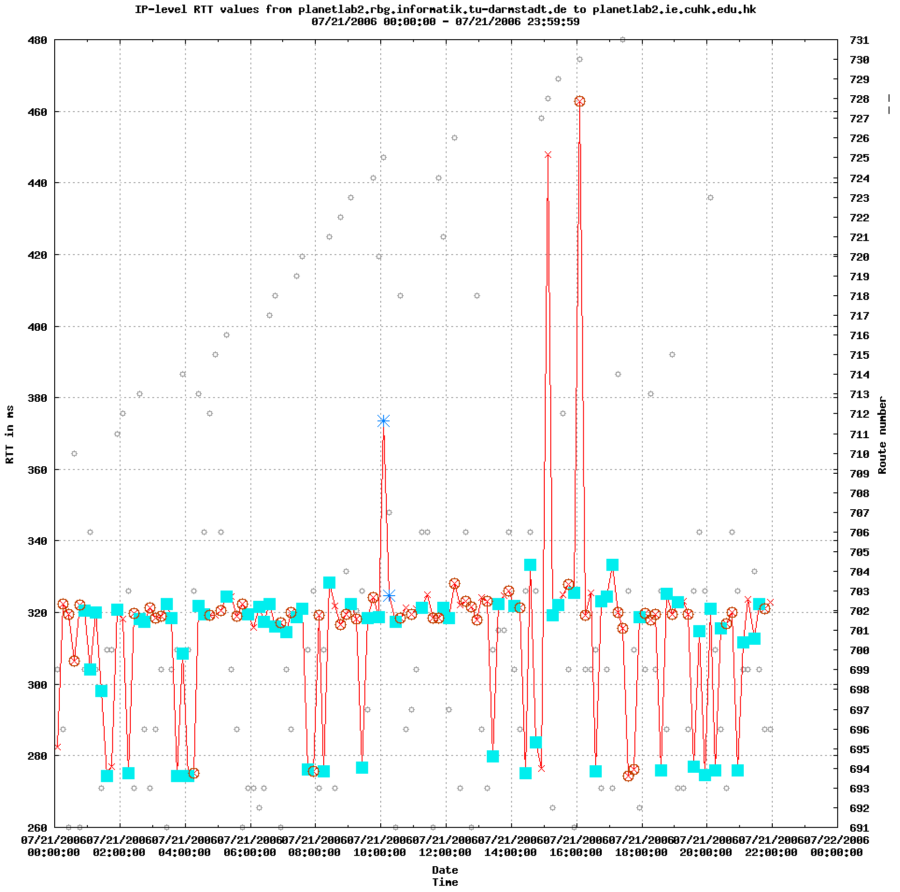


Figure 7.6: IP-level RTT Time Series. The x-axis represents a time line, the left y-axis represents the measured round trip time (RTT) in milliseconds. The right y-axis and the grey circles indicate the IP-level route number seen for a given measurement.

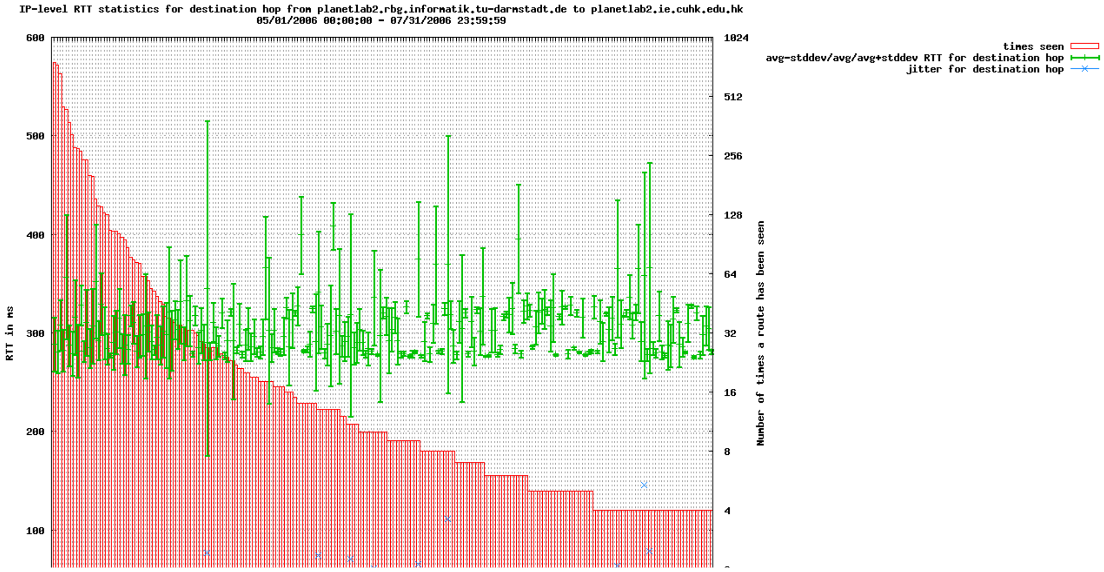


Figure 7.7: IP-level Route Statistics. The x-axis represents all routes seen for a given end-to-end path sorted by their occurrence. The left y-axis and the error bars indicate the average RTT time and the standard deviation measured for the last three months for each route. The right y-axis and the histograms represent the number of times a route has been seen (given in a logarithmic scale).

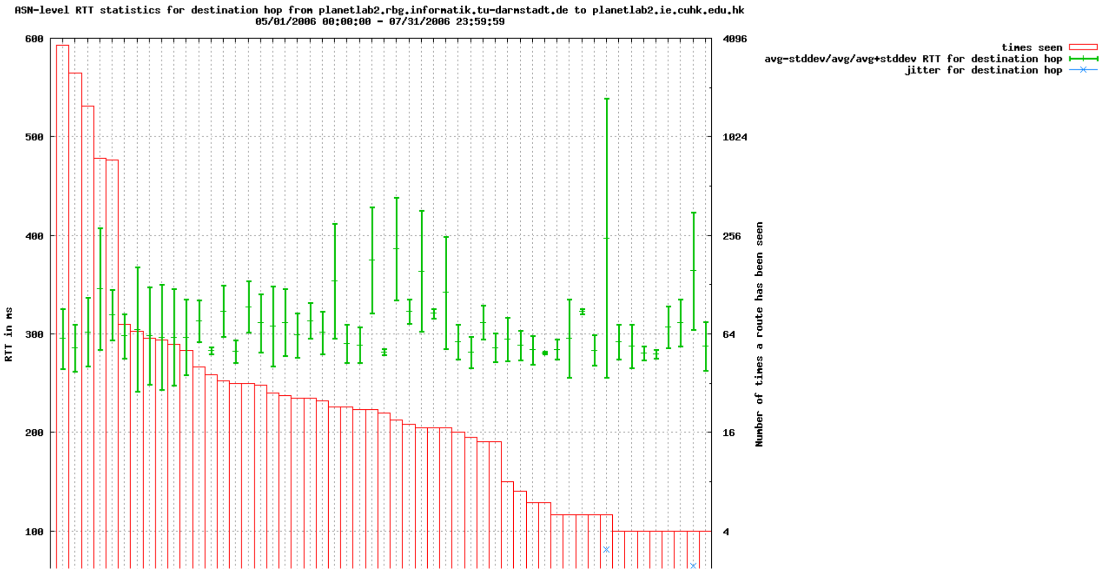


Figure 7.8: AS-level Route Statistics. The x-axis represents all routes seen for a given end-to-end path sorted by their occurrence. The left y-axis and the error bars indicate the average RTT time and the standard deviation measured for the last three months for each route. The right y-axis and the histograms represent the number of times a route has been seen (given in a logarithmic scale).

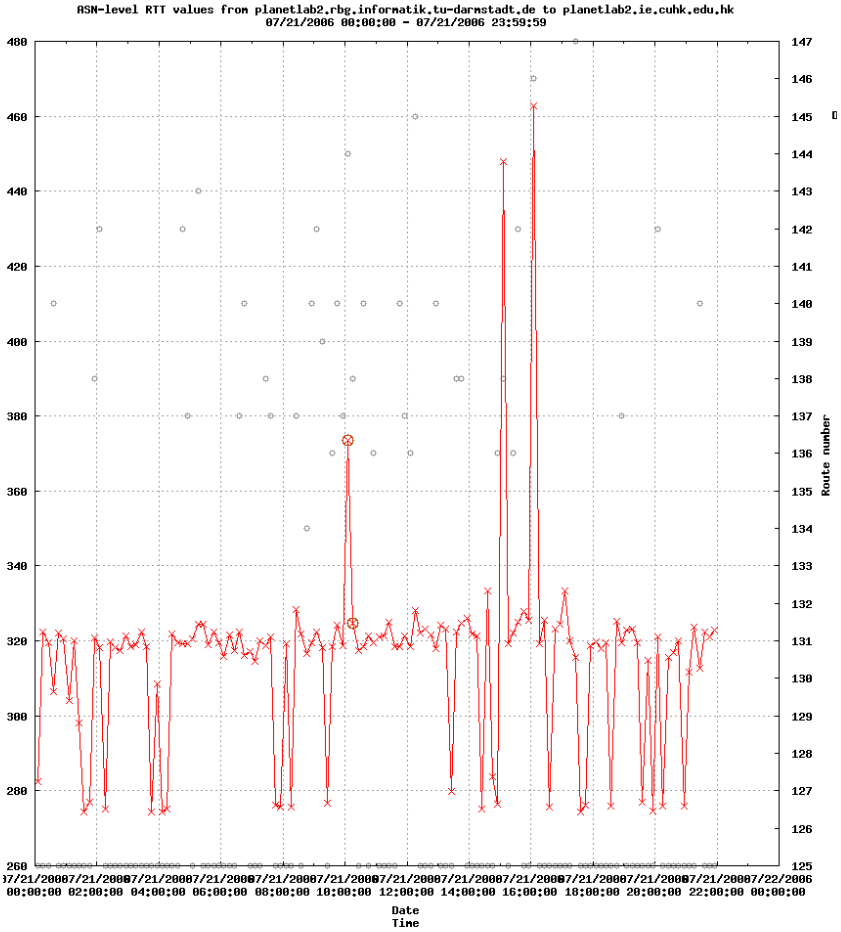


Figure 7.9: AS-level RTT Time Series. The x-axis represents a time line, the left y-axis represents the measured round trip time (RTT) in milliseconds. The right y-axis and the grey circles indicate the IP-level route number seen for a given measurement.

7.1.2 Summary

With the help of the different debugging views implemented during this Master's thesis we have found examples which clearly exhibit a correlation between the route changes and performance changes. On the other hand, we have also found examples for which we could not determine whether there is a correlation or not. The second example given in Subsect. 7.1.1 at least seems to exhibit a correlation even if we could not prove our assumption. But, we have also found examples which did not reveal any correlation at all since the measured round trip times for a single route exhibited too large variations.

Our categorization into two types of end-to-end paths is very coarse. It is very probable that there exist a large number of end-to-end paths neither of type 1 nor of type 2 but something in between. We have chosen these two categories for the sake of simplicity. The coarse categorization scheme does not change the conclusion we derive from our observations. For some end-to-end paths we can state that there is a high correlation of topology changes and route changes. However, there also exist some end-to-end paths for which such a correlation can not be found. Thus, we can not conclude that there exists a correlation of topology changes and performance changes in general.

This result is largely influenced by our own approach. The problem is that although our framework makes it a lot easier to debug and analyze topology changes, the whole procedure still needs manual debugging. Our statement whether there exists a correlation between topology changes or not is solely based on manual observations. We therefore come to the conclusion that it is important to find a more practical way to compare different end-to-end paths with one another than by manually analyzing the properties of different end-to-end paths. We propose such an approach in Sect. 7.2.

7.2 A Quality Measure for Comparing Different End-to-end Paths

7.2.1 Background

In order to compare different end-to-end paths, it is important to find a measure for the quality of one single end-to-end connection. We could for example compare the average round trip times of the routes. The problem of this approach is that outliers can distort the average RTT. Even if we calculated the standard deviation for each route, how can the standard deviation be considered when comparing different average RTT values? In Fig. 7.7 we have seen, that some routes exhibit very large variations.

Another approach is to take the minimum RTT or the maximum RTT as a quality measure. The minimum RTT indicates how good a route can be since the minimum RTT is the round trip time measured during a time where there was no overloading and no queuing problem on the end-to-end path. Another possible quality measure is the maximum RTT which reports the round trip measured during a time where the end-to-end path exhibited overloading or queuing problems. Thus, this value indicates how bad a route can be.

We decided to use the minimum RTT and the 0.95 quantile, because related work [11] showed that that the maximum RTT is not a good quality measure since the outliers are contained in it. High quantiles however (0.95-0.99) are the best representation of the network delay performance. We chosed the 0.95 quantile because of our small number of measurements for some routes. The 0.95 quantile is the value for which 95% of the measured RTT values are smaller or equal. It thus can be imagined as something like the maximum RTT without the outliers.

7.2.2 A Measure For Route Quality

As described in Sect. 7.2.1, the minimum RTT and the 0.95 quantile are good indicators for the quality of a connection. We used these two values to calculate the quality of a route. The resulting equation is depicted in Fig. 7.10.

In the case of an optimal connection, the resulting route quality would be 1, that is the 0.95 quantile equals the minimum RTT. In other words, the worst measured RTT is equal to the best possible RTT. Of course a ratio of 1 is only theoretical as it is very unlikely that we always

$$\text{Route Quality} = \frac{\text{Quantile}_{0.95,R}}{\text{min}_R}$$

$\text{Quantile}_{0.95,R}$ = 0.95 quantile for route R

min_{R} = minimum RTT for route R

Figure 7.10: Route Quality

measure the same RTT value. In practice, the difference of the 0.95 quantile to the minimum RTT is a good indicator for the stability of a connection. If the 0.95 quantile is almost the same as the minimum RTT, that is, the route quality approximately equals 1, then the route is very stable. The worst RTT observed is not much worse than the best possible RTT. Vice versa, the route shows large variations in case the 0.95 quantile is much larger than the minimum RTT. We will illustrate this approach in the examples given in the next two paragraphs.

Note that we assume that the lowest RTT we measured for a specific route, the minimum RTT of this route, is actually the optimal RTT for this route. Since our measurement data covers about 3 months, it is very probable that we achieved to measure the optimal RTT, or respectively a value which is very near to it. One possible improvement to our approach would be to determine the optimal RTT theoretically instead of taking the smallest RTT measured. The theoretical value could be based on an analysis of the geographical distance traversed by the route and the optimal time traffic would need for such a distance to be sent over the Internet.

Example 1: The first example in Subsect. 7.1.1 has shown that topology changes from route 0 to route 19 in the end-to-end path from Washington to Taiwan caused a rise in the measured RTT. In order to compare route 19 and route 0 (and the other routes of this end-to-end path) the 0.95 quantile of the last 3 months was calculated for each route and then plotted against the minimum RTT of this route. Plotting the 0.95 quantile against the minimum RTT instead of only calculating the route quality according to the equation in Fig. 7.10 has the advantage that also the performance of the route is visualized and not only its stability. Figure 7.11 shows the resulting diagram.

The x axis represents the minimum RTT in ms. The y axis the 0.95 the quantile in ms. Each point in the graphics represents one of the possible routes, whereby a small point means that the route was only seen a few times. The larger the point, the more frequent the route has been seen. The closer a point is to the diagonal the more stable is this route, since the diagonal represents the values where the 0.95 quantile equals the minimum RTT. The nearer a point lies to the origin, the smaller is the measured RTT and thus the better is the performance. In short we can say that the distance to the diagonal is a measure for the stability of a route and the distance to the origin is a measure for the performance of a route.

Looking at the routes with number 2, 4 and 10 in Fig. 7.11 it can be observed that they are almost equally stable since their distance to the diagonal is nearly the same. But, routes 2 and 10 are better than route 4 in that they exhibit a smaller round trip time.

For route 19 we can see that first, it is far away from the origin, and second, also far away of the diagonal. Thus, route 19 not only exhibits a larger RTT but in addition it is also very unstable. However, route 19 occurs only rarely, which is suggested by the small point. Route 0 on the other hand occurs very frequently and is located closer to the origin and closer to the diagonal. Thus, it exhibits a smaller RTT than route 19 and is also much more stable. Because all other routes occur much less often, one can say that route 0 is the dominant route between the monitoring nodes in Washington and Taiwan.

It is easy to detect a correlation between changes in the end-to-end performance and route changes from route 0 to route 19 and back again because route 19 has worse performance compared to route 0. This result is not new. We have already found this correlation in Subsect. 7.1.1. But the diagram in Fig. 7.11 helps in understanding better why there is a correlation for this example.

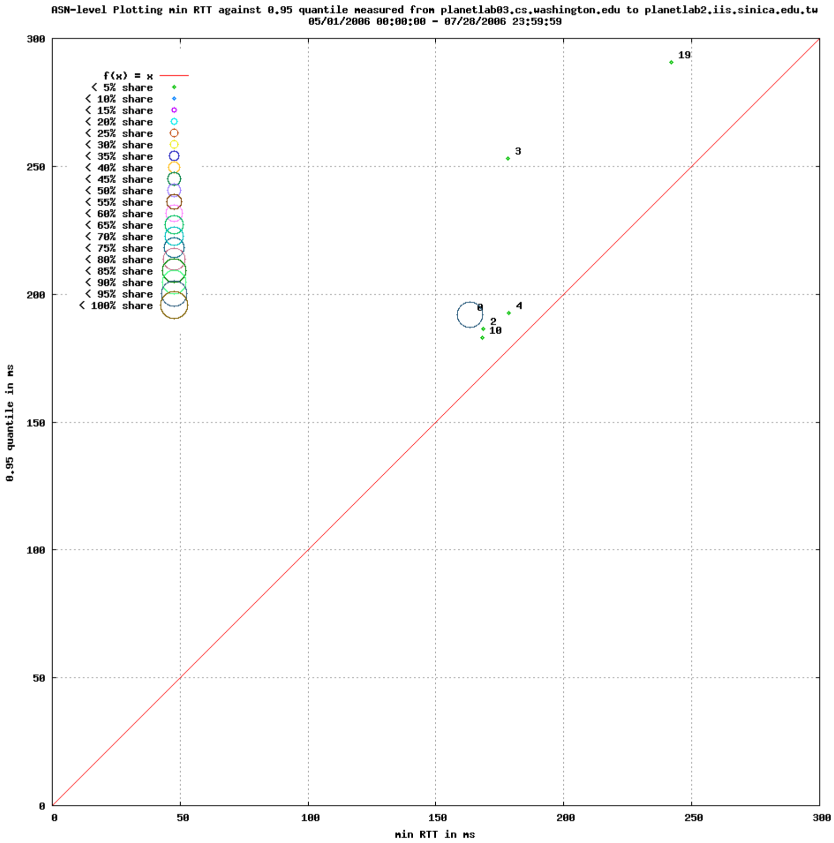


Figure 7.11: Route Qualities

Example 2: In the previous paragraph we have seen that the quality measure introduced in Subsect. 7.2.2 supports the correlation we have found for stable end-to-end paths. Now we will look at an example with an unstable end-to-end path. In Subsect. 7.1.1, we have explained why we were not able to determine whether there exist a correlation between route changes and performance changes in the case of unstable end-to-end paths. In this paragraph we will investigate if our quality measure can solve this question.

The diagram for the second example in Subsect. 7.1.1 is depicted in Fig. 7.12. As we have already seen in Fig. 7.8, the end-to-end path from Darmstadt to Hong Kong exhibits a very large number of different routes and these routes show large variations. Figure 7.12 supports this observation, but in addition, it also gives an explanation why it was so difficult to show a correlation between route changes and performance changes. As can be seen in the figure, many points lie very close to each other, or even on top of each other. This suggests that many routes have very similar properties in terms of performance and stability.

The cloud of points near the diagonal represent stable routes with almost equal performance. No correlation with performance changes can be found for route changes involving only these routes because there is virtually no change in performance. The other type of points which lie farther away from the diagonal represent unstable routes. It can be observed that the minimum RTT for these routes lie in the same range as those of the stable routes. But the large distance to the diagonal indicates that the RTT value for these routes varies largely. These high variations indicate that the end-to-end connection suffers from overloading or queuing problems. Thus, no correlation can be found for route changes involving these kinds of routes as the measured RTT is distorted.

Summary The last two examples illustrated the usage of the route quality measure introduced in Subsect. 7.2.2. With the help of this measure it is possible to distinguish between stable and unstable routes. All in all we have found two explanations why it is not possible to find a correlation between route changes and performance changes in general:

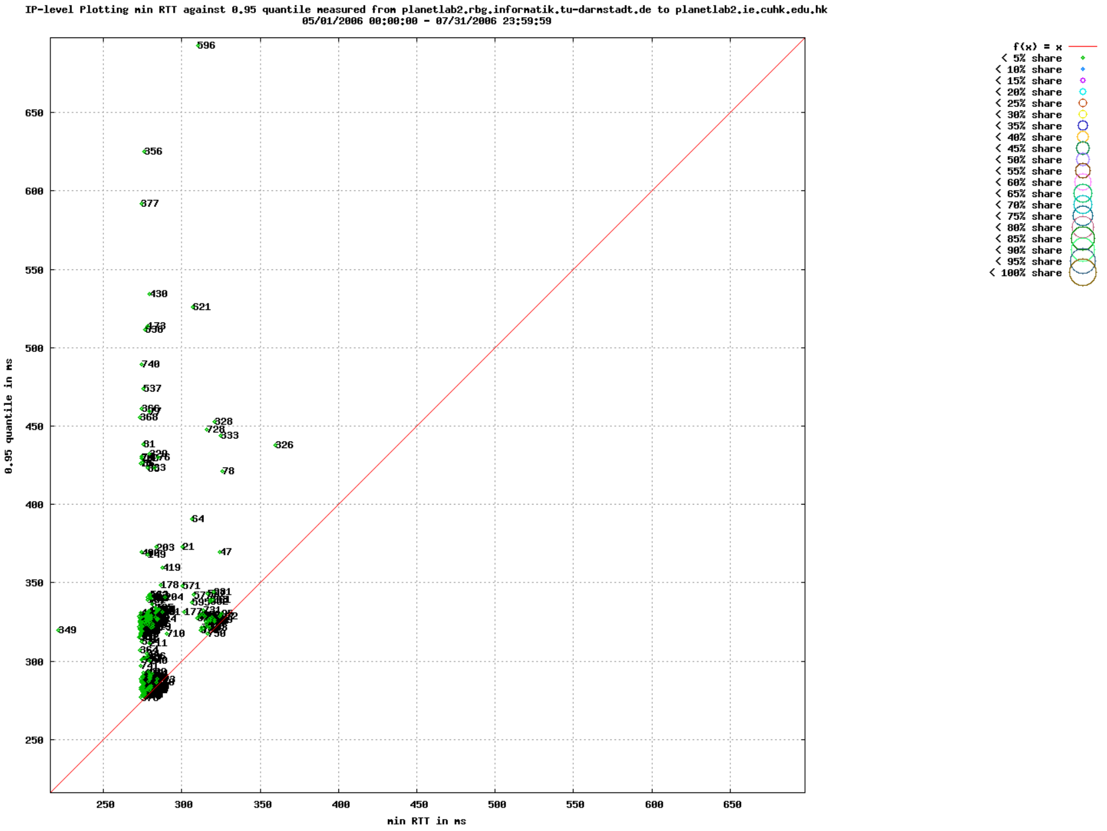


Figure 7.12: Route Qualities

1. The existence of a large number of routes with similar performance makes it difficult to detect a correlation since route changes involving only these routes exhibit no change in performance. Grouping routes with similar properties into a single identifier could alleviate the analysis (see Subsect. 8.2.1).

2. Unstable routes show large variations in the measured RTT values and thus it is very difficult, if not even impossible, to find any correlation between performance changes and route changes involving unstable routes.

The distinction between stable and unstable routes is not based solely on observations as we have done in Subsect. 7.1.1, but based on the ratio of the 0.95 quantile and the minimum RTT. Stable routes have a ratio which approximately equals 1, unstable routes exhibit a ratio which is larger than 1. In the graphics used in the previous examples the ratio was indicated by the distance to the diagonal. We consciously do not state for which range of ratio values a route can be considered as stable or respectively unstable. More research is needed to determine this boundary. We assume that there exists no "strong" boundary though.

7.2.3 A Measure For End-to-end Quality

In Subsect. 7.2.2, we compared different routes of one end-to-end connection. Now we want to compare different end-to-end connections with one another without having to create a route quality diagram for each end-to-end path. The principle remains the same. We use the ratio of the 0.95 quantile and the minimum RTT to distinguish between stable and unstable paths. To get one value for each end-to-end connection the ratio of each route is calculated and then weighted by the probability that the route will be seen. The resulting equation for calculating the quality of an end-to-end path is illustrated in Fig. 7.13

$$\text{End-to-end Quality} = \frac{\sum_R p_R * \text{Quantile}_{0.95,R}}{\sum_R p_R * \text{min}_R}$$

$$p_R = \frac{\text{total number route R has been seen for this end-to-end path}}{\text{total number these end-to-end path has been seen}}$$

$\text{Quantile}_{0.95,R}$ = 0.95 quantile for route R

$\text{min}_{R\,=}$ minimum RTT for route R

Figure 7.13: End-to-end Quality

An end-to-end quality which approximately equals 1 represents a stable end-to-end path. The dominant routes account the most to the end-to-end quality and in case the dominant routes are unstable the resulting end-to-end quality will be unstable too. Thus, an end-to-end path will only be stable if the dominant routes are stable. For the examples given in Subsect. 7.2.2 this means that the end-to-path from Washington to Taiwan is stable and the end-to-end path from Darmstadt to Hong Kong is unstable. This result corresponds to the results of our manual observations. While manual observations can only guess the quality of a given end-to-end path, the measure presented in this subsection provides a more accurate method for the distinction of stable and unstable end-to-end paths as it considers the statistical behavior of each route of the end-to-end path.

Example: Comparing the Connection Quality of Different ISPs The quality measure described in this subsection is not only useful for our analysis on the correlation of route changes and performance changes. The ability to compare different end-to-end paths can be useful to analyze a number of different properties of end-to-end paths. For example we could analyze the Internet connectivity of different ISPs. For that purpose we collected traceroute measurements from 10 different sources located worldwide to 3 different ISPs in Indonesia for 2 days. Then the average 0.95 quantile ($\sum p_R * Quantile_{0.95,R}$) was calculated for each end-to-end path and then plotted against the average minimum RTT ($\sum p_R * min_R$). Again, plotting the 0.95 quantile against the minimum RTT instead of only calculating the end-to-end quality according to the equation given in Fig. 7.13 has the advantage that also the performance of the route is visualized and not only its stability. The resulting diagram is depicted in Fig. 7.14.

The x axis represents again the minimum RTT in ms and the y axis to 0.95 the quantile in ms. The diagonal represents again the values where that 0.95 quantile equals the minimum RTT. In comparison to the graphics in Subsect. 7.2.2, one point now represents no more a route but one end-to-end connection. The individual symbols stand for the different sources and the colors stand for the different destinations. A red triangle represents thus the end-to-end connection of the VPN gateway, which is symbolized with a triangle, to the red ISP. A blue triangle represents the end-to-end connection of the same VPN gateway to the blue ISP. As can be recognized in the graphics, the red and the green ISP approximately have the same end-to-end quality. The blue ISP however is much worse. The blue points are all further away from the origin and also further away of the diagonal. That means that the RTT for the same sources is much larger than it is for the red or green ISP. In addition the end-to-end connections of the blue ISP are more unstable than the connections of the red or green ISP. Thus, with the help of our end-to-end quality measure, we can state that the Internet connectivity of the blue ISP is worse than that of the red or green ISP.

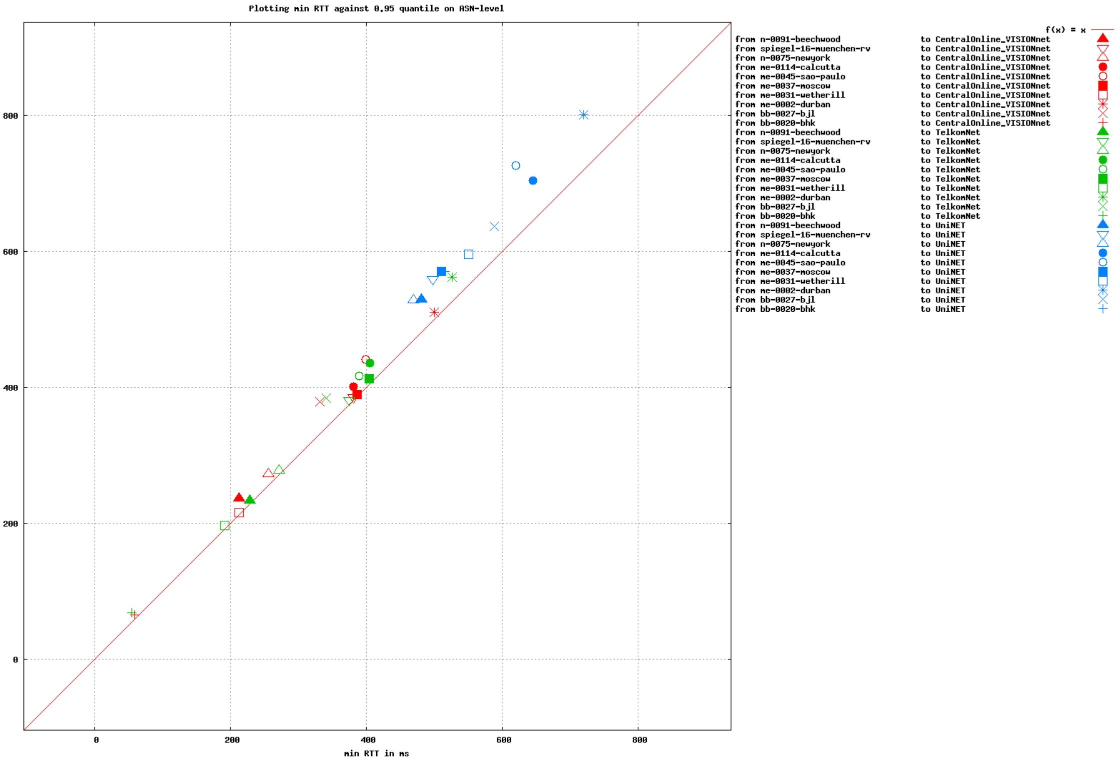


Figure 7.14: End-to-end Path Qualities

7.2.4 Summary

In this section we have presented a measure for determining the quality of a route and also a measure for determining the quality of an end-to-end path. We have applied these measures to different examples and showed that these measures can help to understand why there exists a correlation between route changes and performance changes for certain end-to-end paths and why for other end-to-end paths we can not find such a correlation. We also showed that the end-to-end quality measure can be used to compare the Internet connectivity of different ISPs.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this Master's thesis a framework has been implemented for visualizing and analyzing route changes in end-to-end paths. Trace summary tables provide a daily overview on the route change patterns of each end-to-end path. Additional debugging views like topology graphs and route statistics enable detailed debugging and analysis of each route change. The debugging views are implemented with CGI-scripts and thus can be generated on demand.

The framework has been used to analyze the correlation of route changes and performance changes. Manual observation of the measurement data showed that for certain end-to-end paths, route changes highly correlate with performance changes. However, it was not possible to find a correlation in general.

The aformentioned result was solely based on manual investigations and does not allow a strong statement. Therefore, a measure for determining the quality of a specific route was developed. With this measure it is possible to explain why a correlation could be detected for some end-to-end paths and why not for others.

Besides the analysis of the correlation of route changes and performance changes, an end-to-end quality measure was developed on the basis of the route quality measure. It was shown that the end-to-end quality measure can be used to compare the connection quality of different ISPs.

8.2 Future Work

8.2.1 Enhanced Analysis of the Correlation of Topology Changes and Performance Changes

Revised Distinction of Significant and Non-significant Route Changes In Subsect. 5.5.2 we have described our categorization scheme for distinguishing between significant and non-significant route changes. This distinction was based on the type of the route change: IP-level route change or AS-level route change, same Autonomous System or not, same network or not. This categorization does not consider the performance and the stability of the observed routes. Therefore, we propose to revise the current categorization scheme as for the experienced end-to-end performance the type of the route change is not important. More important is the performance of the current route and its stability. As long as a route change results in a route with similar properties, in terms of performance and stability, the route change should be considered as non significant.

Routes with similar properties can be grouped into a single identifier. For the Traceanal Framework this means that a route number no longer corresponds to a specific sequence of router IPs or respectively Autonomous System numbers, but to a group of routes with equal performance and stability.

The advantage of this new categorization scheme is that we no longer need to look for a correlation of route changes and performance changes. By design, only the route changes which result in a change in performance will be reported. With this approach we can finally implement

a tool which automatically detects and reports problematic route changes, that is, route changes involving high degradations in the end-to-end performance.
A prerequisite of this approach is that we have enough measurement data to provide accurate route statistics. More research needs to be done in order to find a solution for periodic variations in the experienced route quality.

Approximation of Quantiles The calculation of a quantile is very performance intensive as all measurement data needs to be kept in memory. The more data we have, the longer it takes to calculate the quantile. A solution to this problem would be to estimate the quantile with bounded error as proposed in related work [11, 18].

8.2.2 Further Improvements of the Traceanal Framework

Comparison of Forward and Reverse Paths Routes in the Internet may be asymmetric, i.e. the forward path differs from the backward path. In case of route changes it would be interesting to know whether the topology change only affected one direction or both. Such an analysis can help when the performance experienced in one direction is bad, but in the other direction it is not. The Traceanal Framework could analyze both directions of an end-to-end path in case of route changes and report whether the route change was symmetric or asymmetric.
As was illustrated in Subsect. 3.2.2, the router interfaces need to be resolved to routers because otherwise the forward and the reverse path cannot be compared correctly. We have presented some IP-to-AS mapping techniques in Sect. 3.3.

Database The Traceanal version developed at SLAC (Stanford Linear Accelerator Center) uses a MySQL database. Our version of Traceanal stores all data into local files. In future work a database should be introduced. Currently all information needs to be parsed from the traceroute log files. In order to increase performance, some information like for example the route numbers are stored in consolidated files. Introducing a database would not only reduce the amount of redundant data but would also allow to store the IP-to-AS mapping information, resulting in better performance.

Other Metrics The Framework implemented during this Master's thesis uses round trip time (RTT) values as the only performance metric. Considering also other metrics like for example packet loss or bandwidth would give a better estimation of the experienced end-to-end performance.

8.2.3 Analysis of per AS Contribution to Delay

In this Master's thesis we analyzed changes in the end-to-end performance. Future work could focus on the per Autonomous System (AS) contribution to this end-to-end performance. The routing quality of different ASs could be compared by analyzing which ASs are responsible for most of the route changes and which ASs add the most to the end-to-end delay.

8.2.4 Analysis of the Geographic Properties of the End-to-end Paths

Some of the larger ISPs span several cities or even countries. Knowing not only to which network a router belongs but also in which city or country it is located can help to understand various aspects of Internet routing. Subramanian et al. [68]'s analysis of the geographic properties of Internet routes shows that there exists a number of circuitous paths. They observed that the circuitousness of routes often depends on the geographic location of the end host and usually paths traversing a large number of ISPs tend to be more circuitous. An explanation of the latter can be that routing within an ISP's network is much more controlled than routing through a number of ISPs. Routing among ISPs follows business considerations and can influence the routing paths between two different ISPs. For example, if an ISP A has no peering agreement with an ISP B in a city C then this results in an end-to-end path including large detours through another city D where these two ISPs have a peering agreement. Subramanian et al. [68] and also Open Systems AG experienced that some hosts located in countries in Asia have better connectivity to the U.S. than between themselves.

Subramanian et al. [68] could show that the circuitousness of a path does in fact impact the minimum delay of the end-to-end path. And because the end-to-end path is an important performance metric, it is very likely that path changes resulting in circuitous paths will also lead to worse end-to-end performance. Although there is not a perfect correlation of geographic location and performance, an analysis of the geographic properties of an Internet route can give an indication as to which paths are potentially anomalous.

8.2.5 Long-term Analysis of ISP Quality

In Subsect. 7.2.3, we have presented a method for comparing the Internet quality of different ISPs. However, the result of this approach can only be considered as a time averaged snap shot of the observed period. Comparing the observed quality for different days, weeks or months will allow to describe the Internet quality of an ISP in the long term.

Open Systems will follow up on this project. With over 700 sites worldwide it is important for them being able to determine the quality of an ISP, and to select the best possible ISP for their remote locations. Until now this has to be done manually by tracerouting different ISPs from a large number of sites. With the method proposed in this Master's thesis this could be automated to a large extent.

Appendix A

Official Task Description

The following four pages are the official task description for this Master’s thesis.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Master's Thesis
for
Janneth Malibago

Supervisors extern: Roel Vandewall, Stefan Lampart
Supervisors intern: Daniela Brauckhoff, Arno Wagner

Issue Date: 30.01.2006
Submission Date: 30.07.2006

Automated Monitoring of Internet Service Provider (ISP) Topologies

1 Introduction

1.1 Bad VPN connections caused by ISP problems or topology changes

Open Systems AG based in Zurich, Switzerland is a company specialized in Internet Security since 14 years. As part of the Mission Control Services offering, Open Systems AG runs large international VPN networks for large and midsize companies, which allow different sites to securely communicate with each other. Currently Open Systems operates over 600 devices in 70 countries.
VPN connections are tunnelled through the public Internet and therefore they are very dependent on the quality of the underlying networks and topologies of the Internet Service Providers (ISPs). Sometimes it happens that VPN connections between customer sites get slow or even do not work anymore because of outages, problems or topology changes of the involved ISPs. The customer will then complain about the bad VPN connection and Open Systems has to help him to improve the connectivity again. An engineer will manually investigate and will try to find out which router in the Internet causes the problems. Finally a problem report is written which can be forwarded to the ISP by the customer and which helps the ISP to solve the issue.

1.2 Disadvantages of manual debugging of ISP problems

The manual investigation of ISP problems and topology changes is a time consuming and repetitive task. It normally takes more than one hour work to find the cause for an unstable connection and to summarize the findings in a report.
A second disadvantage of manual debugging is that it is triggered by complaints of the customers. A proactive monitoring of the ISP connections and topologies would be much better as the problem resolution could already be started before the customer is complaining.

A third disadvantage is that the manual investigations are done very rarely and only in case of problems. If these investigations could be done regularly and for all sites the resulting statistics could be used to compare the different ISP networks and topologies.
Finally, ISPs often change their peering or routing strategies, sometimes causing poor VPN connections. A pro-active monitoring system could record well-performing routing set-ups and compare them to the routing during a connection problem. This information could aid in finding the routing change that caused the problem in the first place, thus expediting the resolution of the problem by the ISP.

2 The Task

The thesis is conducted at Open Systems AG (http://www.open.ch) in Zurich. The task of this Master's Thesis is to build an ISP topology/routing monitoring and debugging system. On the one hand the system should regularly analyse the topologies and detect changes. On the other hand it will be invoked in case of connectivity problems detected by other subsystems [1] running on the VPN gateway (1). By collecting several measurements and performing network-topology queries at the affected site (on the VPN gateway) and also from remote (from several monitoring centres) it should be possible to automate the problem identification to a large degree (2). The findings should be summarized and visualized in a report that can be forwarded to the ISP and helps them solve the problem (3).
The task of the student is split into four major subtasks that all will be: (i) analysis of known mechanisms to monitor and debug ISP topologies, (ii) specification of an ISP topology monitoring framework, (iii) implementation of a prototype, and (iv) test of and evaluation with the prototype.

2.1 Analysis of known mechanisms to monitor and debug ISP topologies

Existing monitoring and debugging mechanisms/tools that allow to inventory and compare ISP topologies have to be analysed, tested and compared. The most promising mechanisms/tools can be used as inspiration for implementing the monitoring system.

2.2 Specification of ISP topology monitoring framework

Based on the analysis of known ISP monitoring and debugging mechanisms Janneth needs to propose new or improved algorithms to analyse ISP topologies. She needs to write a specification, such that the proposed algorithm can be used as a self-contained monitoring system that allows to merge and visualize the ISP topology views from different monitoring locations. Promising algorithms and ideas that have been encountered in the analysis phase can of course be incorporated.

2.3 Implementation of a prototype

Following the above mentioned specification a prototype should be implemented on Linux (and run possibly also on Solaris). The resource consumption and performance of the prototype must be such that the server can still offer its normal services.

2.4 Testing and evaluating the prototype

The prototype must be thoroughly tested under real network load. Therefore a small test network simulating at least three sites has to be set up. A first objective of the testing phase is to provide a proof of concept, i.e. show that the algorithm is correct and that the whole monitoring system is usable. Besides, and more importantly, the prototype provides the foundation for evaluating the whole thesis. Janneth needs therefore to define evaluation criteria and a methodology how these criteria can be verified with the prototype. The results of the evaluation will possibly, and most probably, trigger a refinement of

[1] One such subsystem is being created as part of the diploma thesis ŚPassive Measurement of Network QualityŠ by Dominique Giger which is currently held at Open Systems.

certain concepts, and improve the implementation. The evaluation will definitely allow to issue recommendations for future work on that topic, and what are steps to consider for an implementation beyond a prototype.

3 Deliverables

The following results are expected:

- Survey of existing ISP topology monitoring/debugging mechanisms and tools A short but precise survey should be written that studies and analyses the different known mechanisms to inventory, monitor and debug ISP topologies.
- Survey of potential topology-related problems. Possible problems that are expected or known to occur in ISP topologies should be found and documented.
- Definition of own approach to the problem In this creative phase the student should find and document a monitoring/debugging framework that allows to analyse and visualize changes in ISP topologies.
- Implementation of a prototype The specified prototype should be implemented.
- Testing of the prototype Tests of the prototype with simulated topologies and topology changes should be made in order to validate the functionality. The efficiency of the prototype has to be measured. Additionally, the prototype should be tested in an actual VPN environment.
- Documentation A concise description of the work conducted in this thesis (task, related work, environment, code functionality, results and outlook). The survey as well as the description of the prototype and the testing results is part of this main documentation. The abstract of the documentation has to be written in both English and German. The original task description is to be put in the appendix of the documentation. One sample of the documentation needs to be delivered at TIK. The whole documentation, as well as the source code, slides of the talk etc., needs to be archived in a printable, respectively executable version on a CDROM, which is to be attached to the printed documentation.

4 Organizational Aspects

4.1 Documentation and presentation

A documentation that states the steps conducted, lessons learnt, major results and an outlook on future work and unsolved problems has to be written. The code should be documented well enough such that it can be extended by another developer within reasonable time. At the end of the thesis, a presentation will have to be given at TIK that states the core tasks and results of this thesis. If important new research results are found, a paper might be written as an extract of the thesis and submitted to a computer network and security conference.

The developed code of the prototype and the implemented algorithms will be released under the terms of GPL2 as open source at the end of the thesis.

4.2 Dates

This Master's thesis starts on January 30th 2006 and is finished on July 30th 2006. It lasts 26 weeks in total. At the end of the second week Janneth has to provide a schedule for the thesis. It will be discussed with the supervisors.

After a month Janneth should provide a draft of the table of contents (ToC) of the thesis. The ToC suggests that the documentation is written in parallel to the progress of the work.

Two intermediate informal presentations for Prof. Plattner and all supervisors will be scheduled 2 months and 4 months into this thesis.

A final presentation at TIK will be scheduled close to the completion date of the thesis. The presentation consists of a 20 minutes talk and reserves 5 minutes for questions. Informal meetings with the supervisors will be announced an organized on demand.

4.3 Supervisors

Roel Vandewall, rv@open.ch, +41 44 455 74 00, Open Systems AG, http://www.open.ch

Stefan Lampart, stl@open.ch, +41 44 455 74 00, Open Systems AG, http://www.open.ch

Daniela Brauckhoff, brauckhoff@tik.ee.ethz.ch, +41 44 632 70 50, ETZ G93

Arno Wagner, wagner@tik.ee.ethz.ch, +41 1 632 70 04, ETZ G64.1

13.01.2006

Bibliography

[1] Dimitris Achlioptas, Aaron Clauset, David Kempe, and Cristopher Moore. On the bias of traceroute sampling; or, power-law degree distributions in regular graphs, 2005.

[2] David G. Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan. Topology inference from BGP routing dynamics. In *Proc. of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[3] Autonomous System. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Autonomous_system_%28Internet%29, March 2006.

[4] F. Baker. Requirements for IP version 4 routers. RFC-1812, IETF, http://www.ietf.org/rfc/rfc1812.txt, June 1995.

[5] BGPlay. University of Rome 3, http://www.dia.uniroma3.it/~compunet/bgplay/.

[6] Dion Blazakis, Jon Oberheide, and Manish Karir. BGP-inspect. http://bgpinspect.merit.edu/.

[7] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors, 1970.

[8] M. Caesar, L. Subramanian, and R. Katz. Towards localizing root causes of BGP dynamics, 2003.

[9] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. *Statistical Science*, 19(3):499–517, August 2004.

[10] Di-Fa Chang, Ramesh Govindan, and John Heidemann. The temporal and topological characteristics of BGP path changes. In *Proceedings of the International Conference on Network Protocols*, pages 190–199, Atlanta, Georga, USA, November 2003. IEEE.

[11] Baek-Young Choi, Sue Moon, Rene Cruz, Zhi-Li Zhang, and Christophe Diot. Practical delay monitoring for ISPs. CoNEXT Conference, October 2005.

[12] CoMon - A Monitoring Infrastructure for PlanetLab. https://www.planet-lab.org/.

[13] Crontab. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Crontab, July 2006.

[14] DIMES. http://www.netdimes.org/.

[15] Xenofontas A. Dimitropoulos, Dmitri V. Krioukov, and George F. Riley. Revisiting Internet AS-level topology discovery. Passive and Active Network Measurement Workshop (PAM), 2005.

[16] Benoit Donnet, Philippe Raoult, Timur Friedman, and Mark Crovella. Efficient algorithms for large-scale topology discovery, June 2005.

[17] Benoit Donnet, Bradley Huffaker, Timur Friedman, and kc claffy. Implementation and deployment of a distributed network topology discovery algorithm, March 2006.

[18] Ulrich Fiedler. Evaluating performance in systems with heavy-tailed input, 2003.

[19] V. Fuller, T. Li, J. Yu, and K. Varadhan. CIDR address strategy. RFC-1519, IETF, http://www.ietf.org/rfc/rfc1519.txt, September 1993.

[20] Lixin Gao. On inferring Autonomous System relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, December 2001.

[21] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. A framework for IP based Virtual Private Networks. RFC-2764, IETF, http://www.ietf.org/rfc/rfc2764.txt, February 2000.

[22] Gnuplot. http://www.gnuplot.info/.

[23] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet map discovery. In *Proc. IEEE INFOCOM*, pages 1371–1380, Tel Aviv, Israel, March 2000.

[24] Graphviz - Graph Visualization Software. http://www.graphviz.org/.

[25] GTrace - A Graphical Traceroute. CAIDA, the Cooperative Association for Internet Data Analysis, http://www.caida.org/tools/visualization/gtrace/.

[26] J. Hawkinson and T. Bates. Guidelines for creation of an Autonomous System (AS). RFC-1930, IETF, http://www.ietf.org/rfc/rfc1930.txt, March 1996.

[27] B. Huffaker, D. Plummer, D. Moore, and k claffy. Topology discovery by active probing. Symposium on Applications and the Internet (SAINT), 2002.

[28] Internet End-to-end Performance Monitoring - Bandwidth to the World (IEPM-BW) project. http://www-iepm.slac.stanford.edu/bw/.

[29] Internet Protocol. RFC-791, IETF, http://www.ietf.org/rfc/rfc791.txt, September 1981.

[30] Internet Service Provider. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Internet_Service_Provider, March 2006.

[31] IP Measurement Protocol (IPMP). NLANR, Measurement and Network Analysis, http://watt.nlanr.net/AMP/IPMP/.

[32] Van Jacobson. Pathchar. ftp://ftp.ee.lbl.gov/pathchar.

[33] S. Kent. IP encapsulating security payload (ESP). RFC-4303, IETF, http://www.ietf.org/rfc/rfc4303.txt, December 2005.

[34] Anukool Lakhina, John W. Byers, Mark Crovella, and Peng Xie. Sampling biases in IP topology measurements, 2003.

[35] LinkRank. http://linkrank.cs.ucla.edu/.

[36] Connie Logg, Les Cottrell, Jerrod Williams, and Yee-Ting Li. Traceanal: a tool for analyzing and representing traceroutes. http://www.slac.stanford.edu/grp/scs/net/talk03/e2ebof-jul04.ppt.

[37] Connie Logg, Les Cottrell, and Jiri Navratil. Experiences in traceroute and available bandwidth change analysis. ACM SIGCOMM Workshops, August 30 and September 3 2004.

[38] Connie Logg, Jiri Navratil, and R. Les Cottrell. Correlating Internet performance changes and route changes to assist in trouble-shooting from an end-user perspective. In *PAM*, pages 289–297, 2004.

[39] Longest Prefix Match. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Longest_prefix_match, March 2006.

[40] Looking Glass Servers. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Looking_Glass_Servers, March 2006.

[41] Matthew J. Luckie, Anthony J. McGregor, and Hans-Werner Braun. Towards improving packet probing techniques, 2001.

[42] Macroscopic IPv6 Topology Measurements Project. CAIDA, the Cooperative Association for Internet Data Analysis, and Waikato Applied Network Dynamics (WAND) research group of the University of Waikato, New Zealand, http://www.caida.org/analysis/topology/macroscopic/IPv6/.

[43] Macroscopic Topology Measurements Project. CAIDA, the Cooperative Association for Internet Data Analysis, http://www.caida.org/analysis/topology/macroscopic/.

[44] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. The Internet AS-level topology: Three data sources and one definitive metric. *ACM SIGCOMM Computer Communications Review (CCR)*, 2006. (to appear).

[45] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy Katz. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, September 2003.

[46] Zhuoqing Morley Mao, David Johnson, Jennifer Rexford, Jia Wang, and Randy Katz. Scalable and accurate identification of AS-level forwarding paths. In *Proc. IEEE INFOCOM*, Hong Kong, China, March 2004.

[47] MaxMind's GeoIP® technology. http://www.maxmind.com/.

[48] A. McGregor and M. Luckie. IP Measurement Protocol (IPMP). http://watt.nlanr.net/AMP/IPMP/draft-mcgregor-ipmp-03.txt, November 2003. Internet Draft.

[49] Merit IRR Services. http://www.irr.net/.

[50] David Meyer. Route Views project. University of Oregon, http://www.routeviews.org/.

[51] Open Systems AG. http://www.open.ch.

[52] Jean-Jaques Pansiot and Dominique Grad. On routes and multicast trees in the Internet, January 1998.

[53] Vern Paxson. End-to-end routing behavior in the Internet. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 26,4 of *ACM SIGCOMM Computer Communication Review*, pages 25–38, New York, August 1996. ACM Press.

[54] PingPlotter. http://www.pingplotter.com/.

[55] PlanetLab - An Open Platform for Developing, Deploying, and Accessing Planetary-scale Services. https://www.planet-lab.org/.

[56] Y. Rekhter and T. Li. CIDR address allocation architecture. RFC-1518, IETF, http://www.ietf.org/rfc/rfc1518.txt, September 1993.

[57] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC-4271, IETF, http://www.ietf.org/rfc/rfc4271.txt, January 2006.

[58] Route Flapping. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Route_flapping, April 2006.

[59] Route Servers. InetDaemon.Com, http://www.inetdaemon.com/tools/route_servers.shtml, March 2006.

[60] Routing Information Service (RIS). RIPE, http://www.ripe.net/projects/ris/index.html.

[61] Scalable and accurate AS-level traceroute tool.
http://public.research.att.com/~jiawang/as_traceroute/.

[62] scamper. http://www.caida.org/tools/measurement/scamper/.

[63] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Powerlaws and the AS-level Internet topology. *IEEE/ACM Transactions on Networking*, 11(4):514–524, August 2003.

[64] skitter. http://www.caida.org/tools/measurement/skitter/.

[65] Source Routing. Internet Security Systems, advICE, database of information security (infosec) and anti-hacker information,
http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Source_Routing/default.htm, June 2006.

[66] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1), February 2004.

[67] Neil Timothy Spring. *Efficient Discovery of Network Topology and Routing Policy in the Internet*. PhD thesis, University of Washington, 2004.

[68] L. Subramanian, V. Padmanabhan, and R. Katz. Geographic properties of Internet routing. USENIX 2002, June 2002.

[69] Lakshminarayanan Subramanian, Sharad Agarwal, Jennifer Rexford, and Randy H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proc. of IEEE INFOCOM*, New York, NY, June 2002.

[70] Renata Teixeira and Jennifer Rexford. A measurement framework for pin-pointing routing changes. ACM SIGCOMM Workshop on Network Troubleshooting, September 2004.

[71] The Team Cymru IP to ASN Lookup Page.
http://www.cymru.com/BGP/asnlookup.html.

[72] Tier Hierarchy. European Internet Exchange Association, Internet and Internet Exchange Definitions, Acronyms and Abbreviations
http://www.euro-ix.net/glossary/index.php#TierHierarchy, March 2006.

[73] Traceping. http://av9.physics.ox.ac.uk:8097/www/traceping_stats.html.

[74] traceroute.org. http://www.traceroute.org/.

[75] UNIX Manual Page for traceroute.

[76] VisualRoute. Visualware, http://www.visualroute.com/.

[77] WHOIS. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/WHOIS, March 2006.