University of Würzburg
Institute of Computer Science
Research Report Series

Measurement-based Topology and Performance Investigations of D-A-CH Research Networks

Dominik Klein[1], Kurt Tutschku[2], Thomas Zinner[1]

Report No. 473 August 2010

[1] University of Würzburg
Institute of Computer Science
Chair of Communication Networks
Am Hubland, D-97074 Würzburg, Germany
{dominik.klein,zinner}@informatik.uni-wuerzburg.de

[2] University of Vienna
Faculty of Computer Science
Institute of Distributed and Multimedia Systems
Chair of Future Communication
Universitätsstrasse 10, 1090 Wien, Austria
kurt.tutschku@univie.ac.at

Measurement-based Topology and Performance Investigations of D-A-CH Research Networks

Dominik Klein, Thomas Zinner
University of Würzburg
Institute of Computer Science
Chair of Communication Networks
Am Hubland, D-97074 Würzburg,
Germany
{dominik.klein,zinner}@informatik.
uni-wuerzburg.de

Kurt Tutschku
University of Vienna
Faculty of Computer Science
Institute of Distributed and Multimedia
Systems
Chair of Future Communication
Universitätsstrasse 10, 1090 Wien, Austria
kurt.tutschku@univie.ac.at

1 Introduction

The Internet comprises several types of networks which are usually administered by different authorities. These networks might be academic networks which are administered by the corresponding university as well as public networks which are administered by the corresponding service provider. The different networks are connected at certain point of presences and all networks together form the heterogeneous structure of the Internet. Due to this composition only little knowledge is publicly available about the detailed structure within and between these networks although this information might be of special interest regarding fault toleration and re-routing in case of a link failure. Additionally, detailed information about the underlying network structure allows for an efficient utilization of upstream links in case of a multi-homed network. This scenario also requires further information about different metrics which constitute a rating for the different paths within the topology. Such a metric is for example the bottleneck bandwidth for a path between two endpoints. The contribution of this technical report is to summarize current technologies for mapping Internet topologies as well as tools for measuring the mentioned path metric and to present an implementation of one approach in order to map the topology between about 30 universities in Germany, Austria and Switzerland. The remainder of this report is structure as follows: Chapter 2 presents related work addressing the topic of inferring network topologies in an incorporative environment. Chapter 3 introduces the measurement setup which was used in this work to map the Internet topology between the universities of Germany, Switzerland and Austria. Chapter 4 then presents our results as well as a graphical representation of the inferred topology and a statistical analysis of the bottleneck bandwidth measurements. The last Chapter summarizes the former chapters and gives an outlook for future work using these findings as a starting point for research.

2 Mechanisms for Topology Measurements - Evaluation and Selection

This chapter introduces related work which addresses the topic of network topology mapping. Our approach to map the D-A-CH topology places certain requirements towards the mapping architecture. This section discusses several approaches for generating a network topology map. After that we evaluated these approaches with regard to their usability for mapping the D-A-CH topology.

2.1 Tools

This section provides short descriptions of the different tools along with a short evaluation whether the introduced tool is applicable for our project.

2.1.1 DIMES: Let the Internet Measure Itself

DIMES [1] is an active distributed architecture for measuring the Internet topology on PoP level granularity. The idea behind DIMES is to provide a small agent which can be downloaded by Internet users. The agents perform the actual measurement from the local computer of the Internet user to the distributed DIMES architecture. The measurements comprise ping and traceroute either with UDP or ICMP. DIMES collects the measurements from different sources and infers the topology map. In order to associate the IP addresses to ASs, DIMES performs the concept of the longest prefix match. This is normally used by BGP routers to forward packets. The crux is to match the source IP addresses of the collected measurements against the IP prefixes advertised by ASs in the global BGP. The advertising AS for a specific prefix is usually the upstream provider for a user whose IP address falls into this prefix. DIMES has access to various vantage points in different kinds of networks due to the federated design but it is not possible to map a specific network or region with a certain resolution. DIMES would require Internet users willing to install the agent and this is not possible for all networks. Thus DIMES is not sufficient to map the topology between the D-A-CH universities.

2.1.2 Hynetd: Hybrid Network Discovery

Hynetd [2] is a hybrid measurement approach. It uses active and passive measurement techniques in order to map the network topology. The active measurement part is a backtrace algorithm which executes ICMP traces in a reverse direction to exploit the existence of common middle nodes on different paths. The passive part tests the SNMP capability for each node responding to the ICMP echo request. Nodes are stored either in the SNMP list if the SNMP test was successful or in the ICMP list if not. IP addresses in the ICMP list need to be tested regarding possible aliases. An alias is a set of at least two IP addresses belonging to the same router. It is important to resolve possible alias in order to get an exact map of the topology. Hynetd uses an alias resolution technique which executes ICMP pings with record route option. Additionally hynetd may send UDP packets to non-existent ports as further optimization. These packets induce ICMP error messages which can be also used to resolve aliases. Hynetd is available as a client for local execution and thus it is not possible to use it in an uncooperative environment where one has no local access on nodes located in the target network. The use case for hynetd is to map the own local network. It is not possible to map the networks of specific ASs like the topology between the D-A-CH universities.

2.1.3 An Efficient Approach Towards IP Network Topology Discovery for Large Multi-Subnet Networks

The approach described in [3] uses SNMP agents enabled in routers, switches and network printers to map the subnetwork topology. It constitutes a passive approach to measure the network topology. The algorithm can be executed on any host in the target network. The first step is to query the gateway to get the IP to MAC address mappings from the ARP cache of the gateway. This information is used to infer the IP range of the subnetwork. The next step is to send ping probes to all addresses in the IP range in order to find all active hosts within the subnetwork. Once all active hosts were discovered, the algorithm tests their type by querying the different nodes and stores them in the appropriate node group. The last step is to infer the links between the different nodes by querying the forwarding tables in switches and the routing tables in routers via SNMP. The MAC to port mapping information is then used along with the MAC addresses of the earlier found nodes to infer the links between the router/switches and the discovered nodes. The algorithm is a proper instrument to discover the topology of the own local subnetwork but is not capable of discovering the topology of an uncooperative network and thus the algorithm can not be applied to map the D-A-CH topology.

2.1.4 Mercator: Heuristics for Internet map discovery

Mercator [4] is an active tool for mapping the Internet core. It sends UDP packets with successively increasing TTL value to randomly chosen IP addresses. Mercator uses a random informed address probing heuristics in order to find addressable prefixes as targets for the UDP traces. Mercator itself is a client which is executed on a single machine and there is no constraint on the location of this machine. Mercator uses source routing in order to perform traces from different vantage points. Each time Mercator discovers a new router Rn, the source routing capability is tested by sending a source routed UDP packet via the newly discovered router Rn to another already known router Rk. The newly discovered router Rn is marked as source route capable if Mercator receives a reply from router Rk. This procedure allows Mercator to perform traces from different origins without the need to have a distributed architecture. The final step in mapping the topology is to resolve IP addresses belonging to the same router. Mercator therefore sends UDP packets to non-existent ports and uses the ICMP error message to recognize aliases. Mercator can be used to map entire ASs but it is not applicable to map stub networks like the university networks.

2.1.5 Nec: Network Cartographer

The idea of the architecture described in [5] is to use traceroute servers as distributed vantage points for the measurements. It is an active approach and the traces are performed from all available traceroute servers towards the target IP address in order to get the path information. The intention of nec is to map the entire core of the Internet rather than to map a specific AS. Nec uses BGP dump data available from [6] in order to find promising target IP addresses. Once the traces were executed and the data was stored locally, nec post-processes the acquired data in order to resolve IP addresses belonging to the same router. Finally the path and node information is used to construct the network topology. Nec seems to be a promising approach for measuring the D-A-CH topology but relies on publicly available traceroute servers whose configuration can not be changed and thus only traces can be executed from these vantage points. Nec can be used simply for mapping but the approach is not flexible enough to include additional measurements like the bottleneck bandwidth.

2.1.6 Rocketfuel Approach

The Rocketfuel approach relies on the same technique like nec [5] but uses its own measurement facility called Scriptroute instead of publicly available traceroute servers. This is a major advantage because it is possible to modify the vantage points and run own scripts for various active measurements. The Rocketfuel approach along with the Scriptroute measurement facility thus constitute a flexible implementation with an own distributed measurement architecture.

2.2 Evaluation Criteria

There are various approaches to map a network topology, and not all are reasonable for our specific objective. A first classification to identify usable approaches can be done according to the following taxonomy:

Active Algorithms which use some sort of active ICMP measurements like traces or pings.

Passive Algorithms which use the SMTP protocol in order to gather information from network nodes running a SMTP agent.

Hybrid Algorithms which combine active and passive measurement techniques in order to improve the overall quality of the network maps and to speed up the measurement.

Changeable Architectures which allow execution of custom measurements like for example the bottleneck bandwidth.

Since we want to map the connections between stub autonomous systems (ASs), only the first class, active measurements, is possible. Passive and also hybrid measurements assume some sort of cooperation of the mapped network and this assumption is not met here. Another important requirement is that it must be possible to have access to different vantage points in order to map specific networks from different directions without the need to run an own distributed measurement facility. Additionally, it must be possible to modify the configuration of the vantage points to execute custom measurements in order to measure for example the bottleneck bandwidth. These constraints are used to rate the architectures described in the former section and to select the most suitable one for this project.

2.3 Evaluation and Selection

This section gives a short summary about the evaluation of the different approaches and motivates the selection of the Rocketfuel approach. Table 1 sums up the evaluation provided in the former section:

Tool	**Active**	**Passive**	**Hybrid**	**Changeable**
DIMES	✓			
Hynetd			✓	
Efficient Approach		✓		
Mercator	✓			
Nec	✓			
Rocketfuel	✓			✓

Table 1: Evaluation Summary

From Table 1 we see that Rocketfuel is the only tool which performs active measurements and also offers the possibility to specify custom measurements. This circumstance motivated our decision to use the Rocketfuel approach along with the Scriptroute facility for this project. A detailed overview of our measurement setup is provided in the next chapter.

3 Measurement Setup and Objectives

IP-level based topology maps are usually not publicly available, although this information is of special interest for researchers or network administrators. Researchers could use these topologies to get a better understanding of the Internet structure for developing new protocols and mechanisms. Examples are new re-routing schemes which avoid broken links and dynamically change to backup links. Such protecting mechanisms are used for intra-domain routing, but due to poor knowledge of neighboring networks their usage is limited for inter-domain routing. Another example would be to utilize several uplinks of a stub network in order to increase the overall bandwidth. This scenario additionally requires knowledge about available or bottleneck bandwidth and thus, this information is also of special interest beneath the topology itself. This circumstance motivated our idea to implement an approach for mapping the topology connecting the universities

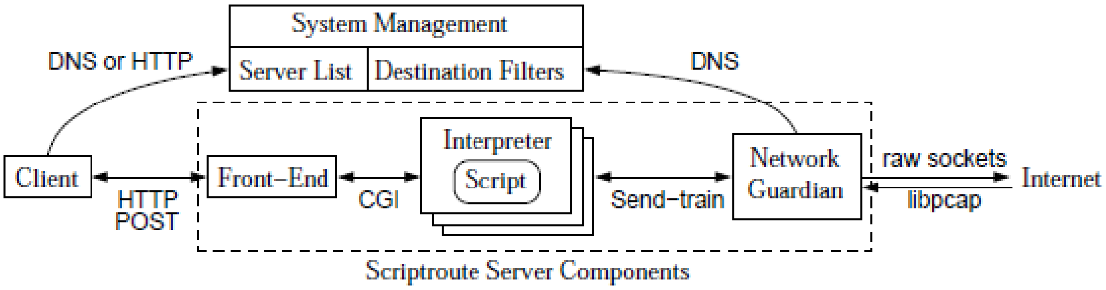

Figure 1: Scriptroute Architecture taken from [7]

in Germany, Austria and Switzerland. Additionally, we applied a bandwidth measurement tool to measure the according bottleneck bandwidth for the paths in the inferred topology map. The remainder of this section introduces the overall measurement setup which comprises the modified Rocketfuel approach, the Scriptroute architecture and the SProbe tool for measuring the bottleneck bandwidth. Both the Rocketfuel approach and the SProbe tool are implemented in Scriptroute and thus is the basis for mapping the Internet topology.

3.1 Scriptroute: A Public Internet Measurement Facility

Scriptroute [7] is a distributed platform for executing different kinds of active measurements in the Internet. The underlying architecture consists of three basic nodes, and a sketch of this system can be seen in Figure 1. The following list introduces the different nodes along with a short description.

System Management Maintains a list of currently active Scriptroute servers in a dynamic DNS-database. Additionally, destination filters are used to prohibit measurement traffic towards certain destinations.

Client Performs a measurement by sending a script to the Scriptroute server.

Scriptroute Server A node which can be used to execute the measurement scripts. The server itself consists of three additional components, the Front-End, the Interpreter, and the Network Guardian.

In order to perform a measurement with Scriptroute, several steps are required and the interaction between the different components during a measurement is explained by means of Figure 1. The client first queries the system management for all available servers by using either DNS or HTTP. Once the client has chosen a set of servers, it sends a script by using a HTTP Post to the set of servers. The front-end offers an interface for inserting measurement scripts and limits the amount of running scripts. Upon accepting a script, the front-end passes the script to the Scriptroute interpreter. The Scriptroute interpreter is a Ruby interpreter and runs in a resource-limited sandbox. Thus, the system can not be congested by too many requests and different measurements do not

IP Prefix	AS Path
1.2.3.0/24	10 1 8 5
	14 7 11 5
	9 2 5
3.5.0.0/16	4 2 13
	2 13

Table 2: BGP Table Excerpt

influence each other. The measurement script which is running inside the interpreter uses the send-train API for sending probes and getting responses. The network guardian interacts with the interpreter and controls the outgoing probe packets. The guardian matches the destination address of each probe packet with the destination filters and blocks packets if necessary. Once the probe packets pass the test, they are sent through a raw socket to the network.

3.2 RocketFuel Approach

The Rocketfuel [8] idea provides an approach for mapping the network topology on IP router granularity. It introduces different steps which constitute the technique to infer the network topology. The basic principle behind this approach is to perform traces from different vantage points towards specific destinations in such a way that the traces traverse the network of interest. A naive approach would be to perform traces from every available vantage point to every available destination. Obviously, this method brings the most detailed and thorough snapshot of the current topology but is impossible due to the huge number of nodes. In order to reduce the number of required measurements, Rocketfuel uses BGP data to only chose those traces which are likely to traverse the target network. This procedure is called directed probing, which is explained in detail in the following.

3.2.1 Direct Probing

A BGP table contains for each destination IP prefix a set of AS paths that can be used to reach that destination. Each AS path is represented as a list of AS numbers. Table 2 contains an excerpt of such a BGP table. The required traces can be classified into different groups and only those belonging to one of the following three groups are executed.

Dependent Prefix Prefixes advertised by the target network or one of its single-homed customers are called dependent prefixes. All traces to these dependent prefixes from any vantage point traverse the target network because there are no other paths available in BGP. All AS paths for these prefixes contain the AS number of the target network being mapped. In our example 3.5.0.0/16 is a dependent prefix of AS 2 (cf. Figure 2).

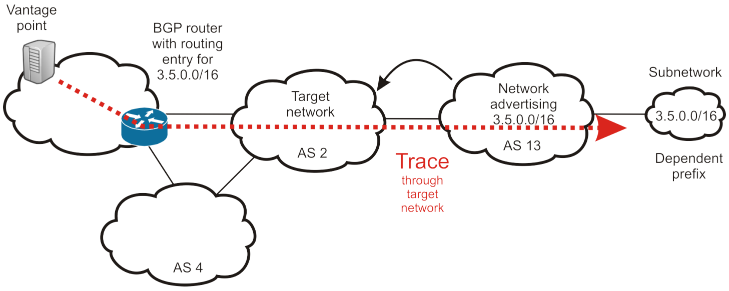
Vantage point
BGP router with routing entry for 3.5.0.0/16
Target network
AS 2
Network advertising 3.5.0.0/16
AS 13
Subnetwork
3.5.0.0/16
Dependent prefix
Trace through target network
AS 4

Figure 2: Dependent Prefix

Insiders A vantage point located inside a dependent prefix is called an insider. Traces from insiders to any of the globally reachable prefixes in BGP should transit the target network (cf. Figure 3).

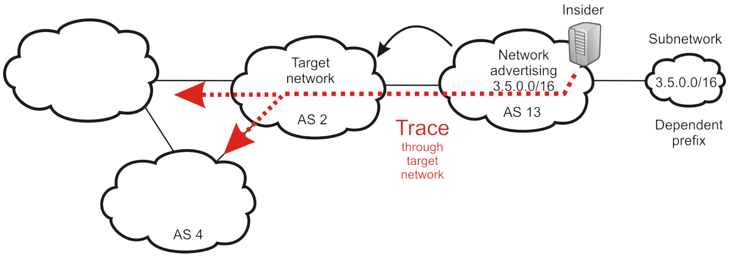

Figure 3: Insider Vantage Point

Up/down traces Traces that are not matched by either of the above rules but which are likely to transit the target network based on some AS-path in the BGP table, are up/down traces. "‘Up/down"’ means that a starting point of a trace is in an "‘upstream"’ AS and the destination is in a "‘downstream"’ AS. In the above BGP example, a trace from a server in AS 9 to 1.2.3.0/24 is an up/down trace (cf. Figure 4).

Rocketfuel now skips traces that do not match any of those three criteria but because of incomplete information in BGP tables, dynamic routing changes, and multiple possible paths, two kinds of errors can occur with directed probing. First, it is possible that traceroutes that do not traverse the target network have been executed. These traceroutes should not have been taken. And second, it is also possible that executed traces that would have transited the target network were skipped because of limited or wrong BGP data. This can cause a loss in completeness and thus care has to be taken when omitting certain traces.

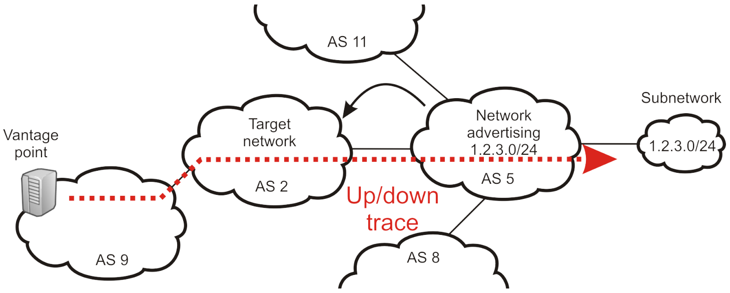

Figure 4: UP/Down Trace

3.2.2 Path Reduction

Some of the traces chosen by directed probing will likely take the same paths inside the target network and thus add only redundant information and can be skipped for further optimization. Traces that enter and exit the network at the same points for example also take the same path through the network and are redundant. These overlapping paths can be identified and skipped by path reduction. There are three techniques that can help to reduce the number of required traces.

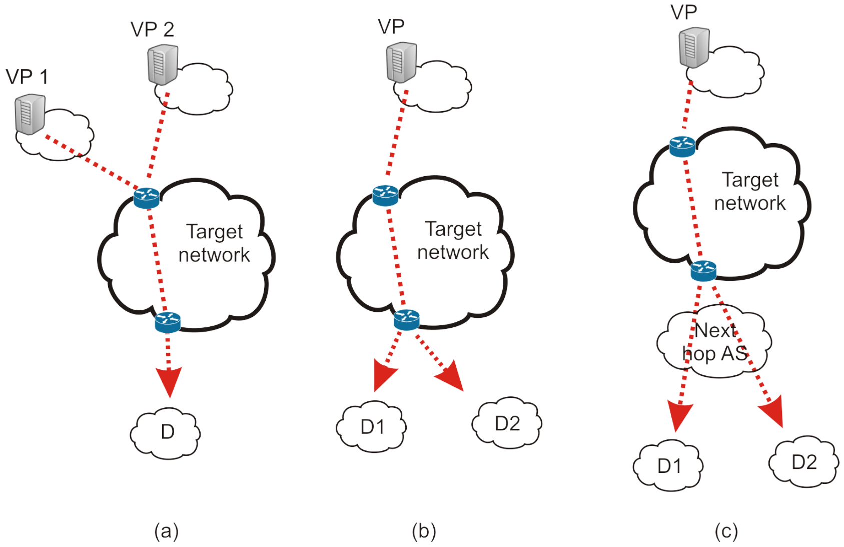

Figure 5: Path Reduction Techniques

Ingress Reduction The path of a packet through a network is usually destination-specific. Traces from two different vantage points may point to the same destination. When two vantage points share an ingress, it is very likely that they take the same path through the target network. An example for this behavior is given in the abstract network topology in Figure 5a. A trace from VP1 to the destination would give us the same information as a trace from VP2 to the same destination. So only one trace is sufficient and the other one is redundant and can be skipped.

Egress Reduction Traces from the same ingress to any prefix behind the same egress router uses the same path through the target network. Figure 5b illustrates such a behavior. Again only one trace is sufficient and the other one is redundant and can be skipped.

Next-Hop AS Reduction The path through a network usually depends on the next-hop AS and not on the specific destination prefix. As illustrated in Figure 5c, it is very likely that only one trace from ingress router to the next-hop AS is valuable. Next-hop AS and egress reduction are very similar because both predict where a trace will leave the target network.

The above mentioned techniques help to chose and to reduce the number of required traces. However, we need to infer the topology from the gathered information. Due to the fact that IP addresses identify only the interface and not the whole node, it is possible that different IP addresses of different traces belong to the same router. These IP addresses are called aliases. Rocketfuel proposes a method called alias resolution which effectively groups IP addresses of the same router and combines them to one node in the topology. This method is detailed in the next subsection.

3.2.3 Alias Resolution

The crux behind the alias resolution is to send probe packets to possible aliases and to compare the returned reply packets from different IP addresses. The resolution algorithm exploits the normal behavior of TCP and sends packets to non-existent TCP ports at possible alias addresses. Following normal TCP implementation, a router replies to those packets with an ICMP port unreachable message including an IP identifier. This identifier is incrementally increased by the correspondent router upon each sending ICMP packet. Thus, the algorithm recognizes an alias if the identifier in the response ICMP packets from different IP addresses lies in the same range.

3.2.4 DNS Information

Once the traces have been analyzed regarding possible aliases it is important to map the different inferred routers to the right network. Therefore, the DNS information is used to determine which router belongs to which target network. The DNS names give a better characterization than the IP address space for the following three reasons.

1. Routers of non-BGP speaking neighbors may be numbered from its provider networks address space itself. In this case, the DNS names help to locate the edge of the provider and the customer network because the neighboring domain routers are typically not named in the provider's domain. A path can include several routers from att.net followed by some routers in example.com without leaving AT&T's

address space. Some provider use a special naming convention that helps to locate the network edge. For example, Sprint names customer routers slneighborname.sprintlink.net, which is different from Sprint's internal naming convention.

2. Edge links between two networks could be numbered from either address space. Again, DNS names help to identify the network edge correctly if they are assigned correctly.

3. DNS names can be effective in pruning out cable modems, DSL, and dialup modem pools belonging to the same organization as the provider, and hence numbered from the same IP address space. The Rocketfuel approach also uses the information embedded in the DNS names to identify the role of each router as well as its location. As mentioned above, most provider have a naming convention for their routers. For example, s1-bb11-nyc-3-0.sprintlink.net is a Sprint backbone (bb11) router in New York City (nyc), and p4-0-0-0.r01-miamfl01.us.bb.verio.net is a Verio backbone (bb) router in Miami, Florida (miamfl01).

But some routers have no (useful) DNS names or location information is missing in their names. The Rocketfuel approach then infers the location of such routers from the location of their neighbors.
The RocketFuel approach combines these different mechanisms and executes them step-by-step to infer the network topology. The different steps are implemented as scripts for the Scriptroute [7] architecture and these scripts were adapted and implemented to map the topology of Germany, Austria and Switzerland (cf. Section 4).

3.3 SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments

SProbe [9] is used for measuring the bottleneck bandwidth in an uncooperative environment. That means the tool has only access to the local machine and is not able to control the remote end of the measurement. SProbe uses the packet pair technique which takes the time gap between consecutive probe packets to calculate the bottleneck bandwidth. Therefore, the initial probe packets need to be large compared to the reply packets, so that only the initial packets experience a queuing delay inside the bottleneck. SProbe can be used to measure the bottleneck bandwidth in upstream and downstream direction. Both approaches are explained in the following.
In order to measure the bottleneck bandwidth in the downstream direction, a large packet pair needs to traverse the path from the local host to the remote host. It is practically impossible to know the time dispersion of the packet pair arriving at the remote host without cooperative communication partners. Under the assumption that the response packets do not queue at the bottleneck bandwidth on the reverse path, the time dispersion can be used as an approximation to the initial large packet pair time dispersion (cf. Figure 6). SProbe therefore exploits the functionality of the TCP protocol. A train of six synchronize (SYN) packets is sent to an inactive port at the remote machine. The remote host answers with six reset (RST) packets (for each SYN packet a RST packet).

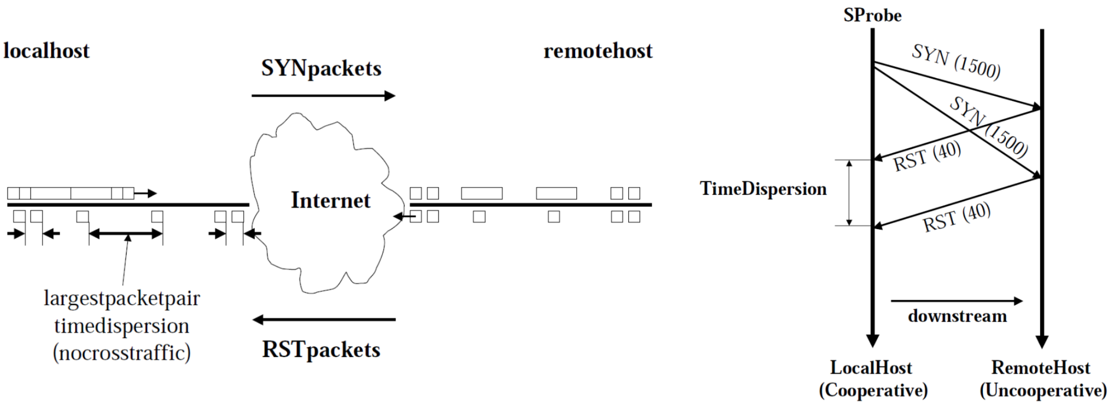

Figure 6: Downstream Bottleneck Measurement taken from [9]

A normal TCP SYN packet has a size of 40 byte and no payload data. SProbe appends a payload of 1460 bytes to the third and fourth SYN packet. The answer RST packet has only a size of 40 bytes. So it is very unlikely that the RST packets are queued at a bottleneck link on the reverse path. The time dispersion between the received two RST packets, corresponding to the third and fourth SYN packet, is measured and used to calculate the bottleneck bandwidth (cf. Figure 6). The smaller the time dispersion, the higher the downstream bottleneck bandwidth. The measurement results need to be checked regarding their correctness since cross traffic would significantly reduce the bottleneck bandwidth and SProbe therefore applies two different heuristics to check the received SYN packets:

- The shuffle heuristics test compares the sequence numbers of the received RST packets with the numbers of the initial SYN packets and if they do not match, the packets were reordered on the path and the measurement is discarded.
- The consistent arrival times heuristic test is applied after the RST packets were tested for their right order. The time dispersion of the two consecutive RST packets in the middle of the train should be larger than the dispersion of any of the smaller 40-byte packet pairs. Violation of this constraint indicates cross traffic during the probing and again results in discarding of the measurement.

Measuring the bottleneck bandwidth in the upstream direction requires some cooperation of the remote node. SProbe uses an active TCP connection to estimate the bottleneck bandwidth by sending a HTTP GET request to a web server (cf Figure 7). SProbe advertises a MSS of 1500 bytes and assumes that the server thus uses an initial TCP congestion window size of at least two times the advertised MSS. According to standard TCP behavior, the server sends as much packets as its congestion window allows. This results in at least two response packets, each with 1500 bytes from the server to the SProbe client. This packet pair can then be used to calculate the bottleneck bandwidth

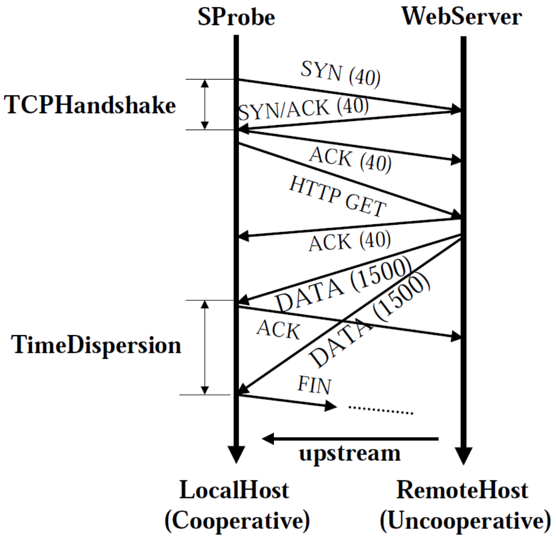

Figure 7: Upstream Bottleneck Measurement taken from [9]

by taking the time dispersion between both packets. If the initial congestion window size on server side is only one MSS, the server sends one packet with 1500 bytes. In this case, SProbe times out after waiting for a second packet and acknowledges the received packet. This induces an increase in the congestion window at the server and eventually the server sends two packets. This mechanism enables SProbe to produce an estimation of the bottleneck bandwidth very fast. SProbe is available as a separate tool or as a script for the Scriptroute architecture.

4 Measurement Implementation in Detail

4.1 Mapping Topology

We applied the measurement technique based on the Rocketfuel approach introduced in Section 3.2 in the Scriptroute architecture to map the topologies of about 30 universities in Austria, Germany and Switzerland. This section describes the implementation in detail along with some problems we faced during the measurement. Scriptroute is strongly related to Rocketfuel and comes with the original scripts described there. These scripts were adopted to map the topology between the investigated universities. Our approach differs from Rocketfuel because we are mapping stub networks and the links between them. The original Rocketfuel approach is used to map highly connected provider networks. Thus, some techniques are not required for our attempt like, e.g. direct probing and path reductions. The measurement comprises the execution of five different scripts which constitute our implementation of the Rocketfuel approach. We installed the Scriptroute API on a local machine to run the different scripts which are explained in the following.

pl-edges.rb This script uses the IP addresses of the universities as destination and all available Scriptroute servers as source address. Most universities block incoming ICMP packets and thus, we used the addresses of PlanetLab servers inside university networks. These servers are constraint to be open to the outside and ICMP packets destined to those servers are accepted. Traces are conducted from every source to every destination and a record is saved in an output file.

pl-names.rb This script looks up the DNS name and AS number for each recorded IP address. Then the IP address is replaced by the DNS-name, if possible. In case no DNS name could be found, the IP address is used as representation instead. The DNS names along with AS numbers are stored in an output file.

pl-aliases.rb This script discovers pairwise aliases by applying the alias resolution algorithm described in Subsection 3.2.3. It uses the output of the former two scripts.

pl-showaliases.rb This script simply groups the aliases discovered by the showaliases.rb script.

pl-finish.rb This script creates the two output files nodes-planetlab and links-planetlab which contain the IP addresses and all discovered links respectively.

4.2 Measuring Bottleneck Bandwidth

We used the bottleneck bandwidth metric, measured with SProbe, to rate the different paths in the inferred topology. The basic principle of this mechanism was introduced in Section 3.3. In a first attempt, we tried to use the SProbe script for Scriptroute but we were not able to conduct a valid measurement. So we used PlanetLab [10] instead. The approach was to install the Scriptroute client on PlanetLab nodes located within the different university networks and then perform the measurement from each available PlanetLab server. We executed the measurement during several timeslots distributed over four days and each timeslot comprises at least ten measurements. This procedure was necessary to one the one hand avoid wrong results because of a congested link and to verify the mechanism used by SProbe.

5 Measurement Results

This chapter presents the results from our experiment along with an outlook regarding future work.

5.1 Topology Maps

The output of our experiments are lists of nodes and pairs of nodes representing IP routers and IP links respectively. Such a representation of the topology is hard to understand. Thus, Google Maps was used to visualize the network topology between the investigated universities. Therefore, it was also necessary to infer the coordinates for each IP address. This was done by using the IPlocator tool available at [11]. For

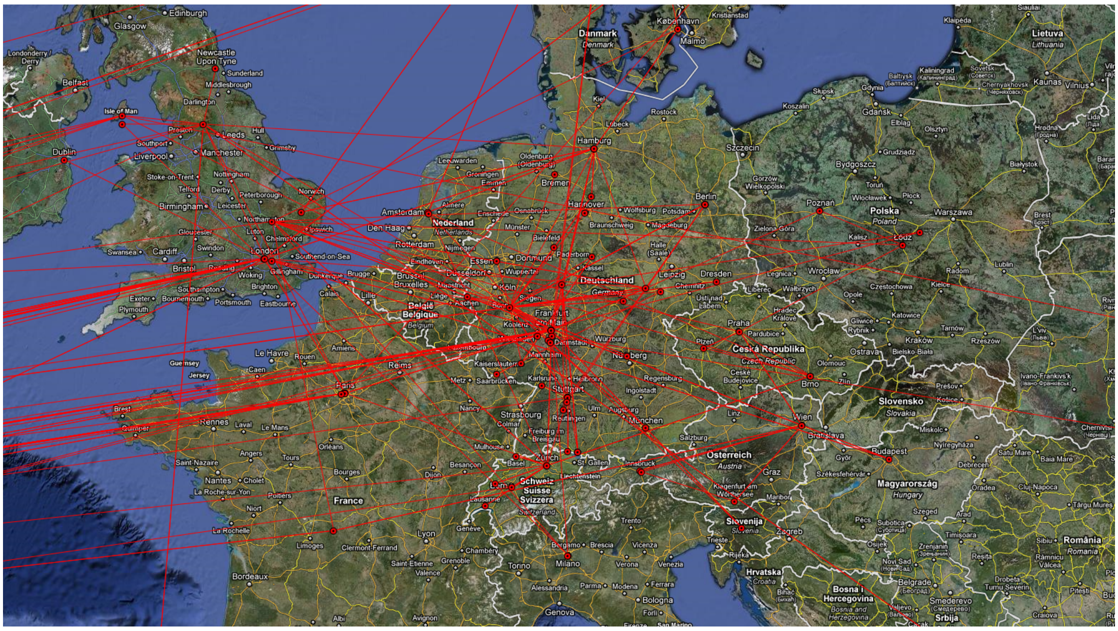

Figure 8: D-A-CH Topology

presentation purposes, we discarded all nodes which were more than one hop away from the target network. An example for the topology can be seen in Figure 8. The figure shows all inferred nodes and links which are at most one AS hop away from the D-A-CH topology. One of our main objective was to discover alternative paths between two endpoints. As an example, we highlighted the existing paths between Berlin and Zurich, see Figure 9. It can be seen that, despite the more or less direct path from Zurich to Berlin via Stuttgart, alternative paths via London and Cologne exist. These paths could be used to improve resilience or maybe the capacity of a data transfer.

A possible use case regarding the existence and knowledge of alternative paths is to test new multipath mechanisms in a experimental platform like PlanetLab [10]. An example is multipath transport where the existence of multiple paths is used to increase the overall throughput. Normally, it is not possible to exploit several paths due to the BGP decision process. BGP provides only one best path and alternative paths are only chosen in case the primary path fails. Our topology can be used to set up a multipath overlay architecture within PlanetLab. This overlay could be used to test whether their is an increased throughput for multipath applications. One possibility is to plan and install different paths via tunneling between PlanetLab servers located in the different university networks. The tunnels constitute an overlay where we can implement an own routing protocol providing multipath routes. This approach also requires knowledge regarding the performance of outgoing and incoming links at the different university networks. Therefore, we combined the inferred topology with the results of the bottleneck bandwidth measurements presented in the next section.

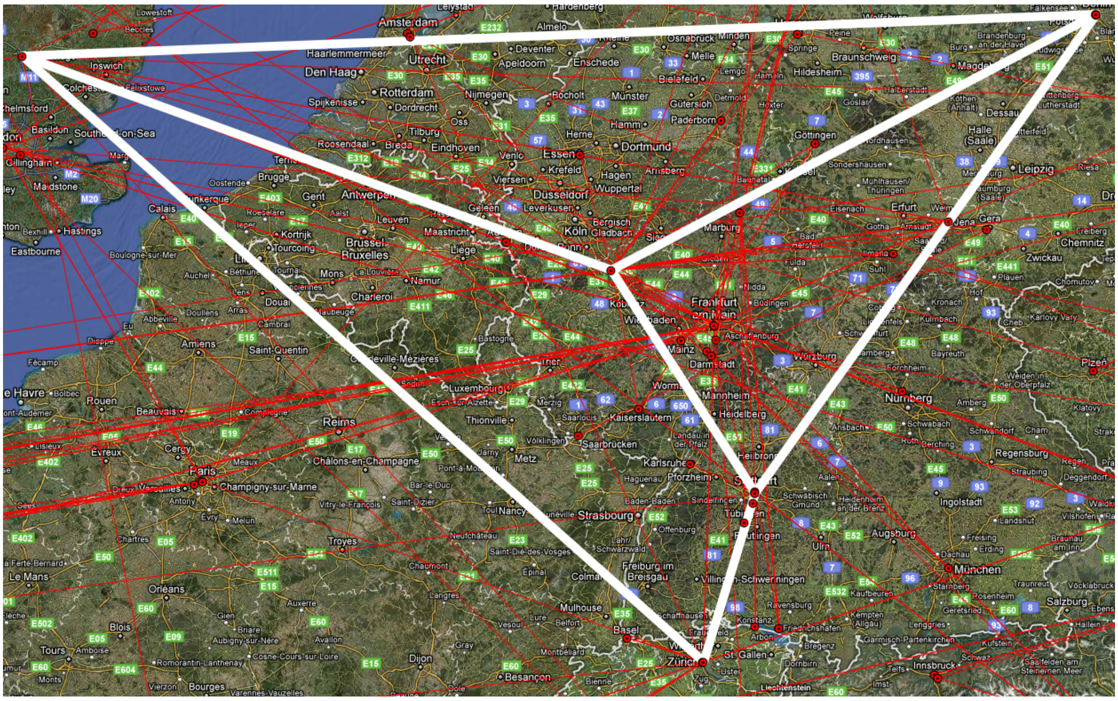

Figure 9: Alternative paths between Zurich and Berlin

5.2 Bottleneck Bandwidth

We measured the bottleneck bandwidth between PlanetLab servers located in the different university networks during several timeslots per day. We repeated the measurements on several days to increase the accuracy and to avoid errors. Possible errors are, for instance overloaded PlanetLab nodes or other experiments generating cross traffic and thus influencing the bottleneck bandwidth measurements. The underlying problem with the PlanetLab architecture is that there is actually no resource isolation between different experiments running on the same node. Also traffic sent from experiments running in other slices on the same node influences the measurement and thus a highly loaded node is inappropriate for our measurements. Further, local policies may also infect the available bandwidth of the measured link.

The gathered measurement results were analyzed regarding different aspects and, as one example, we present the maximum available bottleneck bandwidth per path between two universities. One could also chose the minimum available bottleneck bandwidth as metric. However, both metrics could be used in combination to rate the overall performance of a path between two universities.

We took CAIDA:GeoPlot [12] to visualize the rating of the different paths. An example for this can be seen in Figure 10. It shows the path rating for ETH Zurich. For presentation purposes, we divided the path quality into different categories regarding the bottleneck bandwidth and colorized the links in Figure 10 accordingly. The caption

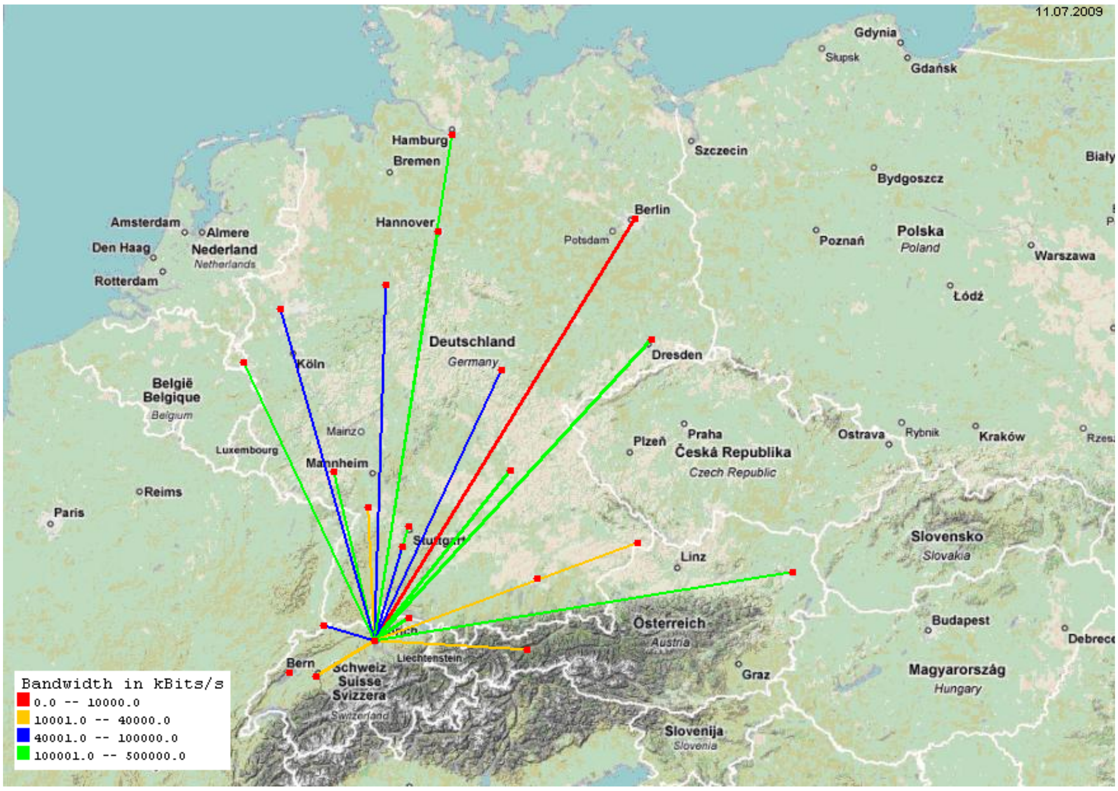

Figure 10: Bottleneck Bandwidth for ETH Zuerich

explaining our taxonomy can be seen in the bottom left part of the figure.
Our results give a first impression about possible problems regarding the measurement of metrics like the bottleneck bandwidth. As future work, it might be interesting to test other measurement tools than SProbe to verify whether the underlying mechanism is reasonable. On the other hand, we could test alternate paths between universities connected by a path with poor quality. As an example, consider the path between Zurich and Berlin which offers a maximum bottleneck bandwidth of about 10 Mbit/s (cf. Figure 10). A possible experiment could utilize an overlay created by means of our topology map and rate alternative paths between poorly connected universities. This information might be interesting when designing future experiments within PlanetLab.

6 Summary

This technical report gave a short overview of current available tools for mapping network topologies. We described the different tools along with a short evaluation and motivated our decision to chose the Rocketfuel approach. We applied a modified version of the Rocketfuel approach along with the Scriptroute measurement facility to map the topology between universities in Germany, Austria and Switzerland. We were able to identify alternative paths between the different universities. This paths could be utilized for example to improve resilience or the capacity of a data transfer.

In addition, we measured the bottleneck bandwidth for the different links with SProbe in order to rate the connections between the different universities. We have seen that the more or less direct path is not always the bast path regarding available bandwidth. The discovered alternative paths could be used in this case to increase the available bandwidth.

Further, the gathered topology along with the measured bottleneck bandwidth provides a valuable view of the inter-domain connections between the different university networks and may be used for example to plan future experiments between the mapped universities.

Acknowledgments

The authors would like to thank the bachelor students Christina Ochsenhofer, Silvia Fürst, Doris Braunöder, Evelyn Hinterramskogler and Evelyn Imrek for their contribution regarding this project.

References

[1] Y. Shavitt and E. Shir, "DIMES: Let the Internet Measure Itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 71–74, Oct 2005.

[2] A. Botta, W. de Donato, A. Pescape, and G. Ventre, "Discovering Topologies at Router Level: Part II," in *Globecom*, (Washington, D.C., USA), Nov. 2007.

[3] F. Nazir, M. Jameel, T. H. Tarar, H. A. Burki, H. F. Ahmad, A. Ali, and H. Suguri, "An Efficient Approach Towards IP Network Topology Discovery for Large Multi-Subnet Networks," in *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, (Washington, DC, USA), pp. 989–993, IEEE Computer Society, 2006.

[4] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, (Tel Aviv, Israel), Mar. 2000.

[5] D. Magoni and M. Hoerdt, "Internet core topology mapping and analysis," *Computer Communications*, vol. 28, pp. 494–506, Sept. 2004.

[6] D. Meyer, "RouteViews Project." http://www.routeviews.org.

[7] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A Public Internet Measurement Facility," in *USENIX Symposium on Internet Systems and Technologies (USITS)*, 2003.

[8] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," in *ACM SIGCOMM*, (Pittsburgh, Pennsylvania, USA), Aug 2002.

[9] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments," in *IEEE INFOCOM*, 2002.

[10] PlanetLab, "An open platform for developing, deploying, and accessing planetary-scale services." http://www.planet-lab.org, 2009.

[11] IPInfoDB, "IP Locator." http://ipinfodb.com/index.php, 2009.

[12] CAIDA, "GeoPlot." http://www.caida.org/tools/visualization/geoplot, 2009.