



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Automatic Rating of VPN Links

Master Thesis (MA-2011-17)

October 2011 - March 2012

Guido Hungerbühler

hguido@ee.ethz.ch

Department of Computer Engineering and Networks Laboratory (TIK), ETH Zurich
Conducted at Open Systems AG

Tutor: David Schweikert
ETH-Tutor: David Gugelmann, Dr. Martin Burkhart
Supervisor: Prof. Dr. Bernhard Plattner



Abstract

Understanding VPN tunnel performance is crucial in helping to improve the quality of globally distributed networks. If we know the performance of every individual tunnel, we are able to spot problems and pin-point bottlenecks in the network. We present a novel way on analyzing and visualizing the long-term performance of VPN tunnels. By using geographical clustering of VPN endpoints, we found that tunnels which connect similar regions also show performance characteristics alike. This allows defining performance baselines with respect to specific regions. Furthermore, it enables the detection of individual connections that constantly perform below standard. The proposed method takes advantage of globally spread networks with multiple links between distinct regions. We have developed a ready-to-use prototype which rates VPN tunnels and visualizes problems in the network. The prototype has successfully been used at Open Systems, a company that operates over 7000 VPN tunnels world-wide.

Keywords

rating, VPN, network performance

Acknowledgments

I would like to thank all the people that supported me during my Master's Thesis.

A special thank goes to David Schweikert from Open Systems who provided during the entire thesis very valuable inputs, critical remarks and great support. Also I would like to thank David Gugelmann and Martin Burkhart from ETH for supervising my thesis and for the many creative meetings we had. My Master's Thesis gave me a very good insight to the field of network performance measurement and to what it means to manage large networks.

Furthermore, I am grateful to Prof. Dr. Bernhard Plattner for the opportunity to do this thesis in the Communication Systems Group, supervising this thesis and giving important remarks in the meetings.

Finally, I would like to thank Stefan Lampart for making it possible to do my Master's Thesis at Open Systems and to all the people at Open Systems for sharing their ideas with me and all their support.

Guido Hungerbühler

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Task	2
1.3. Overview	2
2. Related Work	3
2.1. Performance Measurements	3
2.2. Service Quality	4
2.3. Internet Measurement	4
2.4. VPN Performance	5
3. Data Analysis	7
3.1. Network Characteristics	7
3.1.1. Topology	7
3.1.2. Technology	9
3.1.3. Available Data	10
3.2. Global View	20
3.2.1. Basics	20
3.2.2. RTT Locality	21
3.2.3. Loss Locality	23
3.3. Clustering	24
3.3.1. Algorithms	25
3.3.2. Cluster Number	26
3.3.3. Adapted Clustering	29
3.4. Clustered Endpoints	34
3.4.1. Tunnel Analysis	34

3.4.2. Cluster Rating	35
3.5. Stability and Availability	38
4. Rating Methods	39
4.1. Rating Idea	39
4.2. Performance Reference	40
4.2.1. Data of Interest	40
4.3. Comparison of Tunnel Performance	45
4.3.1. Proximity Metrics for Tunnels	46
4.3.2. Reference Selection	50
4.3.3. Short Tunnels	52
4.3.4. Coverage	52
4.4. Accuracy	52
4.5. ISP Selection	53
4.6. Performance Estimation	56
5. Implementation	57
5.1. Setup	57
5.2. Preparation	58
5.2.1. Topology Database	58
5.2.2. Database Structure	58
5.2.3. Precalculation	59
5.2.4. Import	60
5.3. VPN Performance Rating	60
5.3.1. Data Retrieval	60
5.3.2. Collect Data	61
5.3.3. Report Generation	62
5.4. Performance Report	62
5.4.1. Information	62
5.4.2. Rating Algorithms	63
5.4.3. Result	69
6. Evaluation	71
6.1. Rating Evaluation	71
6.2. Known Provider Issues	72
6.2.1. Case 1 - Zurich to Shanghai	72
6.2.2. Case 2 - Zurich to Guernsey	73
6.3. Known RTT Issue	75
6.3.1. Zurich to London	75
7. Conclusion	79
7.1. Summary	79
7.2. Conclusions	80

7.3. Future Work	80
7.3.1. Evaluation	80
7.3.2. Technology	80
7.3.3. Integration to Network Monitoring	81
7.3.4. Additional Metrics	81
7.3.5. Internet Line Rating	81
A. Appendix	83
A.1. Terms	84
A.2. Spline Interpolation	85
A.3. XML Data Export	86
A.4. File Structure	87
B. Sample Performance Report	89

CHAPTER 1

Introduction

1.1. Motivation

The Internet makes it possible to connect arbitrary hosts and people among each other, to share data and exchange information world-wide. It connects almost every part of the world to the World Wide Web. The Internet is also often used to connect different branch offices among a company. This allows access to data and business applications from all over the world. But using the Internet for business critical information exchange implies risks too, such as security issues and dependency on good performing communication networks. To assure secure communication between different sites of a company, Virtual Private Network (VPN) technology can be used. But badly performing communication networks may directly affect the business continuity. To have the possibility to overcome this problem, we need to spot it first.

As a Managed Security Service Provider, Open Systems operates large Virtual Private Networks (VPNs) covering over 170 countries¹. With the goal to offer outstanding quality to the customers and improve the network performance, Open Systems is interested in knowing how their VPN tunnels perform. The automatic rating and detection of performance issues would enhance the way a network can be monitored and maintained. In case of bad performance, the problem could be analyzed and countermeasures could be taken.

Many of today's network performance monitoring systems merely assume that a network is most of the time performing good and only abnormal events are reported and thereafter alerts generated. The system may also use prerequisite thresholds as performance limits. But if we want to rate the long-term performance of a VPN tunnel this is not sufficient. We want to rate

¹Status: February 2012

it in general to be able to inform a customer about performance issues and possible solutions, e.g., to change the Internet service provider (ISP) if it is a provider related issue or to inform the customer about what services can be used under what constraints. An automatic rating system for VPN tunnels is crucial to detect insufficient performance of links.

1.2. Task

We assume that the performance of communication channels is strongly related to the locations of the two endpoints. This means that we are not able to estimate how good a tunnel should perform without taking these locality effects into account. Therefore we are interested in a reference value indicating how good VPN tunnels between two different regions can perform in general. A single tunnel can then be rated by using this local performance estimate. In the first part of this thesis we analyze the available data and evaluate how the location of the endpoints can be used to get location dependent performance estimates.

So far there is no rating possibility for the general performance of a tunnel. A lot of measurements can be done which indicate how the performance of a link changes over time. Hence anomalies can be detected and reported. But we are interested in a rating system that allows to detect bad performing links even if they perform very constantly. Especially if the traffic of the VPN tunnel is routed very inefficiently, there will be no anomaly in the measurements of a single link.

We propose a different approach to rate a single tunnel. In addition to only looking at a single tunnel's metrics and rate it accordingly, we will also take the measurements of tunnels into account that are geographically close. This allows to rate the performance baseline of a tunnel and not only the instantaneous changes. Normally the VPNs are strongly separated among the companies but by breaking this separation for the performance measurements we want to gain more information about the performance between specific regions. The goal is to get a performance reference for each tunnel which allows to rate its long-term performance.

Since the rating has to be possible in a productive environment, the methods we use should not cause excessive traffic. In addition we want it to be frugal concerning measurement data. The result of this thesis will be a prototype which allows to rate an arbitrary tunnel.

1.3. Overview

First we will present some related work that already has been done at Open Systems or by other researchers dealing with this or similar topics (Chapter 2). In Chapter 3 we will analyze the RTT and loss data and present some insights into the characteristics of the different metrics used to indicate network performance in virtual private networks. Based on the gained information a rating system is introduced in Chapter 4 which will use global knowledge to rate a single tunnel. The implementation of the prototype is described in Chapter 5. Afterwards we evaluate our implemented prototype against some known performance issues in Chapter 6.

CHAPTER 2

Related Work

Network performance is an important topic for all ICT based services. The measuring and the rating of network performance is a wide research area. Many research projects and papers are related to this thesis. Also at Open Systems some previous studies already dealt with performance measuring and rating of networks or focused on the detection of network problems. Zimmerli [30] developed an approach to rate the quality of Autonomous Systems and Stich [26] developed a distributed monitoring system which allows passive measurement of the round trip time (RTT).

This thesis is mostly based on the work of Wagner [28]. He analyzed different metrics which are quality indicators for the links and studied different detection approaches to detect anomalies. We will use his results as a basis for our approach.

In this chapter we will present a small selection of scientific work related to this thesis.

2.1. Performance Measurements

Although the measurement data we can use in this work is restricted to round trip time (RTT), loss rate and load information, it is important to know what is being measured, in what way and how different metrics are linked to each other. The IP Performance Metrics (IPPM) working group of the IETF proposes a framework for network measurements [22] and provides documents for fundamental metrics like RTT or packet loss ¹.

The work by Paxson [23] and Tang et al. [27] demonstrate well the dynamics of the packets in the Internet and were important for the understanding of their behaviour in networks. They

¹IPPM working group <http://datatracker.ietf.org/wg/ippm/charter/> (last visit: Oct. 2011)

also came up with a good model for the RTT. Tang et al. proposed an estimation model for the minimum End-to-End delay. The paper of Sommers et al. [25] gave a deeper insight to the characteristics of packet loss.

2.2. Service Quality

In addition to the direct measuring of RTT and packet loss, it is important to know how they affect the total performance of a link. Padhye et al. [19] and He et al. [9] introduced a method to model TCP throughput as a function of the loss rate and RTT. Padhye et al. also suggested a way to predict the throughput with history-based approach. So we will be able to not only stick to delay and loss rating but are also able to link it to the throughput.

Not all services are only depending on the pure throughput of a link. In addition they are depending in a complex way on delay and loss characteristics. Especially for Voice over IP (VoIP) many studies were made to investigate how the quality depends on the different metrics [29, 17, 5].

Another well-known problem to rate different services is to find out whether the measured quality is good or bad for an individual user. This is called the Quality of Experience (QoE). Especially in the field of IP telephony it is hard to get good estimates about the quality of a call and even harder to get a value for the QoE. Researchers developed the Mean Opinion Score (MOS) which should indicate the quality of a call by asking the user's opinion about it. The MOS is standardized in the ITU recommendation P.800 [12]. The E-model [10] on the other hand is based on metrics that can be retrieved from actual link characteristics. The importance of this model is that its result can be directly linked to the MOS which allows to estimate the QoE [5].

2.3. Internet Measurement

Since the VPNs of Open Systems are globally distributed, our work is also related to the big research projects that measure the Internet performance in general. Such as the ongoing *RIPE Atlas*² project where volunteers install a probe and participate in probably the biggest Internet measurement network of all time. Another project is *skitter*³, where CAIDA collected traffic for about 10 years or its successor the *Archipelago Measurement Infrastructure*⁴. Other projects which should be mentioned are *Surveyor*⁵ from the Internet Society (ISOC) or *PingER*⁶ from the National Accelerator Laboratory (SLAC). SLAC in general provides a lot of information about Internet performance on their website⁷. All these projects monitor world-wide Internet connectivity and performance. A downside to these projects is either that only one site monitors a lot of remote sites (PingER) or that the nodes are not well distributed around the globe. The

²RIPE Atlas <http://atlas.ripe.net> (last visited Oct. 2011)

³skitter <http://caida.org/tools/measurement/skitter> (last visited Oct. 2011)

⁴Archipelago <http://www.caida.org/projects/ark> (last visited Oct. 2011)

⁵Surveyor www.isoc.org/inet99/proceedins/4h/4h_2.htm (last visited Oct. 2011)

⁶PingER <http://www-iepm.slac.stanford.edu/pinger/> (last visited Oct. 2011)

⁷SLAC <http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html> (last visited Oct. 2011)

Archipelago project for example does not have one single node in India or Russia (status Oct. 2011).

We are also interested in estimating the performance based on the locality of the endpoints. Gummadi et al. tried to estimate the delay to an arbitrary host based on the delay to DNS servers that are nearby the hosts. They use DNS reverse look-up techniques to find the nearby DNS servers [8]. Some projects also try to do the opposite and estimate the locality based on the delays. Well-known examples are *Practical Internet Coordinates* [3], *Global Network Positioning* [18] or the pioneering work of Francis et al. *IDmaps* [7].

2.4. VPN Performance

Measuring the performance of VPN links can be seen as a similar task to measuring the performance of normal links as described above but with the understanding that the additional security and tunneling introduce some overhead. In general the article of Ferguson and Huston [6] gave a good insight to what VPN exactly is and Parrott et al. [21] described in his work how the Quality of Service (QoS) of VPN networks can be measured and differs depending on different packet sizes or encryption algorithms. Palmieri [20] also claims that the bottleneck of VPN is its scalability if not operated over an MPLS (Multi-Protocol Label Switching) network.

To the best of my knowledge there exist no papers that directly focus on rating the performance of VPN tunnels based on their endpoints' location. The thesis by Wagner [28] describes metrics and methods to detect badly performing tunnels. It is based on the assumption that tunnels are generally performing well and only changes or abnormalities outside simple thresholds are detected. The location of the endpoints was not taken into account in his work. We will continue his work by taking the location of the endpoints into account and restricting to metrics considered as valuable for the performance rating.

The measurements are performed in a productive network. The different VPN endpoints are distributed in over 170 countries which makes the analysis very interesting. It allows to get a global view on network performance. In addition a productive environment restricts us, not to influence the quality of the customers' network with additional traffic.

CHAPTER 3

Data Analysis

In this chapter we analyze the network characteristics of different VPN tunnels. We get a general insight about the performance measurements and the performance itself. A VPN tunnel is always a connection between two endpoints. Throughout this thesis we talk about endpoints and hosts and we always refer to VPN endpoints and never to the end-user's host or a server.

3.1. Network Characteristics

We analyze the global VPNs operated by Open Systems. A VPN tunnel connects always two sites of a company.

3.1.1. Topology

The analyzed VPNs contain over 1300 hosts in 170 different countries. They are connected with more than 7300 VPN tunnels. These networks allow us to get a global view on network performance. Figure 3.1 shows the global VPNs of all Open Systems customers.

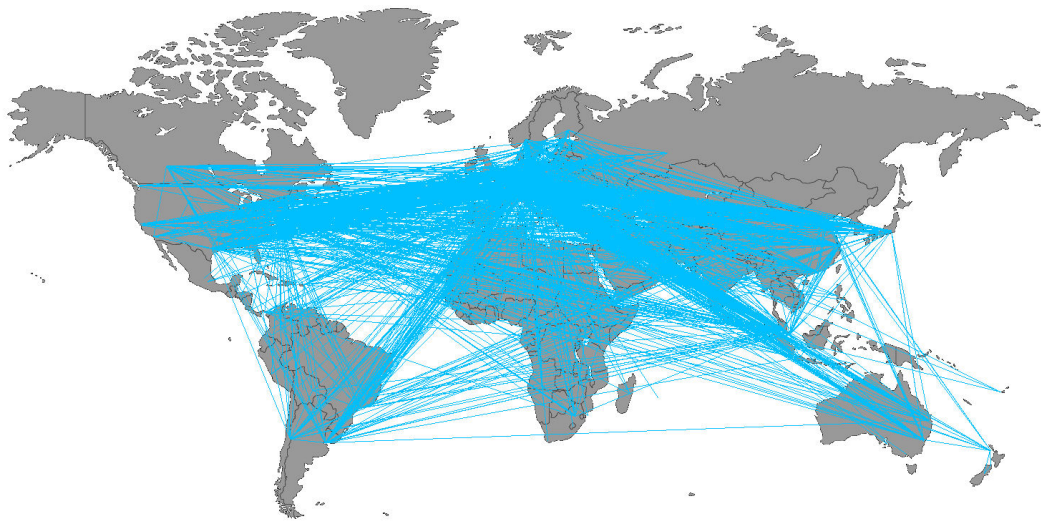


Figure 3.1.: All connections between the 1300 endpoints

Often companies have their headquarters in one location and all other sites are connected to it. This leads to the fact that we have many star-like network topologies. In Figure 3.2 a network of a customer is shown. We see that certain nodes have many neighbours and some have a connection only to the headquarters. Full-mesh networks are rare and exist only for a few customers.

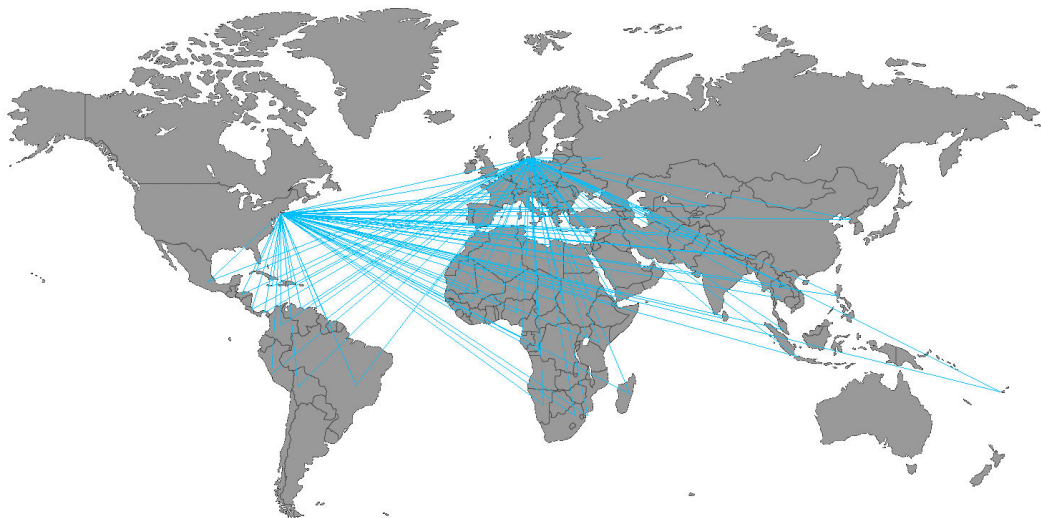


Figure 3.2.: Network of a customer

3.1.2. Technology

The connections of this global networks are established using different carriers such as land-cables, sea-cables or satellite. These carriers have also different characteristics. Optical fibre cables are known to have a very small bit-error rate and are approximated to have a propagation speed of about 0.6 times the speed of light [13]. Satellite links have a very different performance baseline and evaluations have to be done with respect to these performance differences.

Since it is not known whether a certain tunnel's path contains a satellite link we use an approximation to detect satellite hops. We use the fact that the distance to the satellite and back is a lot longer than the same distance between the two endpoints on earth. A geo-stationary satellite is at an altitude of 35'000km above ground ¹. A signal needs at least 250ms to travel to the satellite and back. For a RTT measurement this distance is even doubled which results in 500ms minimum time for a RTT signal using geo-stationary satellites. If this delay of 500ms happens at one particular hop we assume that the link contains a satellite connection. A general classification of tunnels in satellite and non-satellite links is not possible because satellite links are often used as a back-up solution if a wired connection fails. Nevertheless there are locations which are only reachable using satellite technology. Figure 3.3 shows all tunnels that are established using satellite links at the time of analysis.

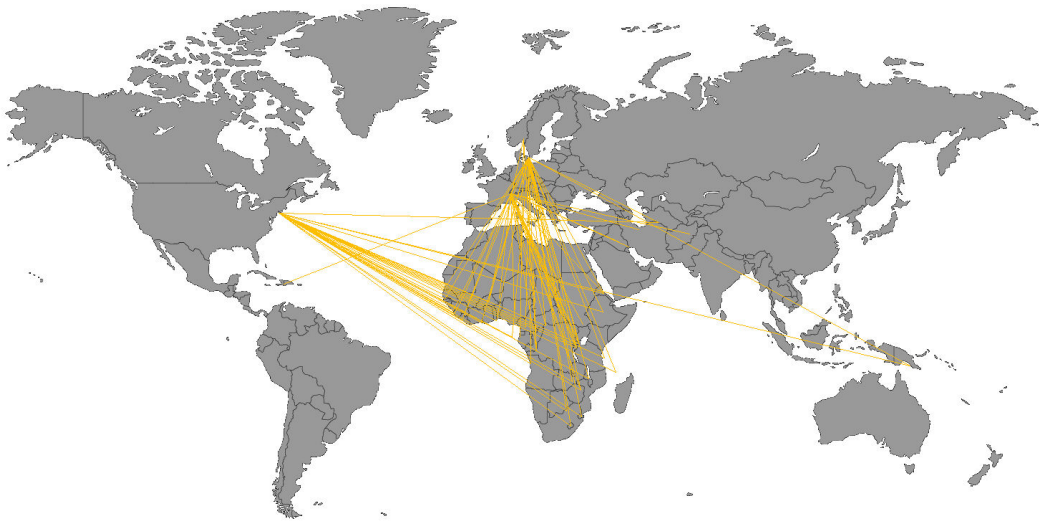


Figure 3.3.: Endpoints connected via satellite

¹Geo-Stat. Satellite http://en.wikipedia.org/wiki/Geosynchronous_satellite (last visited Nov. 2011)

3.1.3. Available Data

The thesis is done by using data from a productive environment. The advantage is that we have traffic which represents real-world conditions. But it also restricts us to a solution which is feasible for a productive environment and it is not allowed to generate excessive probe traffic or bandwidth measurements. Open Systems already collects performance data and we simply use this existing data for our analysis. The performance data we use is limited to:

- round trip time
- packet loss rate
- tunnel load on endpoints (incoming / outgoing)
- location of endpoints

The round trip time and packet loss information are based on ICMP probes. Since the probe traffic is sent encapsulated in a VPN tunnel it only differs from normal traffic in its size. We assume that it shows similar RTT and loss characteristics like the normal traffic and is routed with the same priority and path through the Internet.

Every minute a daemon sends 5 pings. By using RRDtool² the measured average RTT and the packet loss rate are stored in a round robin database (RRD). A RRD stores data that is recently added with a high resolution in time. The older the measurement data the lower the resolution because the used RRD aggregates a time-span of high resolution data to a single value using pre-defined aggregation functions. The used functions are 'maximum' and 'average'. E.g. regarding the average function every 30 minutes the average of all 5-minute RTT values is stored.

The RRDtool is configured to have a 5 min resolution for about the last 2 days. The more we go back in time, the smaller the resolution:

- 2 days: 5 min averages
- 14 days: 30 min averages
- 2 months: 2 hour averages
- 2 years: 24 hour averages

The loss rate is aggregated in the same way. The relatively small resolution of 1 minute between the probes and the fact that we only measure RTT imply that some metrics like jitter can not be retrieved.

Another restriction affects the path. Since we are analyzing VPN links we generally have no information about the path of the packets. The best approximation of the path is a traceroute which is not routed through the tunnel. Therefore it is not guaranteed that the probe packets take the same route like the ESP packets.

We will present some insights into the data which is used in the following sections.

²RRDtool: <http://oss.oetiker.ch/rrdtool/> (last visited Nov. 2011)

Round Trip Time

The three main factors that influence the RTT are the distance between the endpoints, the amount of hops in between and the time each router needs to forward a packet including the time a packet is queued in a buffer. The RTT is roughly given by³:

$$RTT = 2 \cdot (distance / (0.6 \cdot c) + hops \cdot delay)$$

Where c is the speed of light⁴ and $0.6c$ the speed of light in a fibre optic cable. Unfortunately we do not know the number of hops between two endpoints unless we continuously use the traceroute tool to determine the number of hops. Also the delay for processing can only be estimated roughly [13]. The formula above would be the most precise by using the real length of the cable between the two endpoint. An often-used approximation is to merely use the linear distance between two endpoints. Therefore the formula changes in our setup to:

$$RTT = 2 \cdot (distance / (0.6 \cdot c) + \varepsilon)$$

The variable ε represents the unknown additional delay for each connection that can not be calculated simply. It contains all unknown effect of additional cable length, processing time and queuing. In addition the ε is not a constant. The additional delay which is mainly caused by queuing is depending on the congestion of all the intermediate networks (AS) which are part of the path. Therefore the RTT can be split into the propagation delay and queuing delay. The propagation delay can be approximated with the minimum RTT whereas all additional delay is caused by queuing. The processing time for a packet is relatively small in comparison to the propagation delay and the queuing delay and is neglected.

$$RTT = RTT_{min} + RTT_{queuing}$$

Figure 3.4 shows the minimum RTT for different linear distances between the endpoints compared to the formula above. For every tunnel the minimum RTT of an entire month⁵ was used. The black line indicates the RTT formula above neglecting the ε . Figure 3.5 shows a boxplot of the RTT for different distances. The black bar represents the median whereas the blue box represents the inter-quartile range (IQR).

$$IQR = 75^{th} \text{ percentile} - 25^{th} \text{ percentile}$$

We directly see that the main factor for the RTT is the linear distance and that the correlation between RTT and the distance is given. On the other hand we see also that the ε can not be neglected if we want to get a good estimate of a connection's RTT.

³PingER SLAC <http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html#metrics> (last visited: Nov. 2011)

⁴Speed of light: $c = 300'000\text{km/s}$

⁵The Data are from January 2012

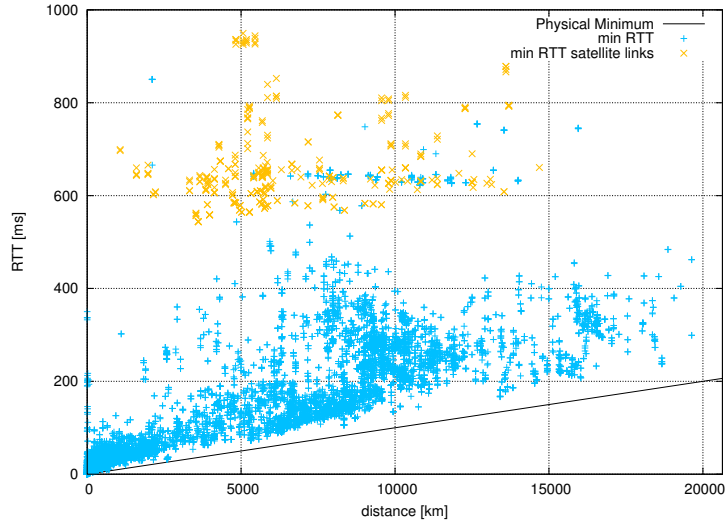


Figure 3.4.: RTT measurements compared to the formula for different distances

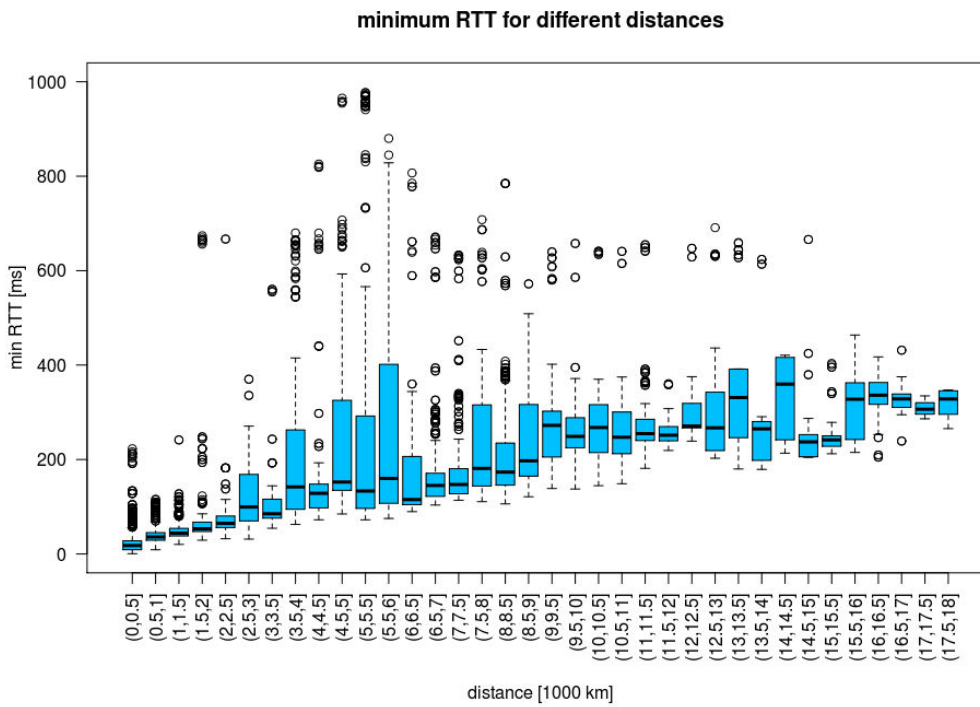


Figure 3.5.: Distribution of RTT measurements for different distances

The ITU-T recommends a model [11] where the distance is measured more accurately by using an approximation of the real path length. They calculate the length by summing up all the linear distances d between the intermediate gateways and then multiplying them with a routing factor

rf which is dependent on the distance between these gateways. The routing factor represents an approximation for our ε .

$$d_i = \begin{cases} 1500km & \text{if } 1000km \leq d \leq 1200km \\ d & \text{else} \end{cases}$$

$$rf_i = \begin{cases} 1.5 & \text{if } d_i < 1000km \\ 1 & \text{if } 1000km \leq d_i \leq 1200km \\ 1.25 & \text{if } d_i > 1200km \end{cases}$$

The RTT can then be estimated by summing up all intermediate distances d_i between the gateways multiplied by the routing factor rf_i . The estimated RTT is the time the light needs to travel this total distance in a cable.

$$RTT_{\text{estimated}} = \frac{1}{0.6 \cdot c} \sum_{i \in \text{path}} rf_i \cdot d_i$$

Nevertheless, this model is not valid for tunnels containing satellite links.

To gather the location of the intermediate gateways on a path, we could use the *traceroute* tool and query the location of each gateway's IP using geo-location IP database services. But with this approach we would introduce an additional unknown source of error. The results would directly depend on the precision of IP geo-location databases. Poese et al. [24] analyzed the reliability of IP geo-location databases and came to the conclusion that the service is unusable for general-purpose geo-location services. An additional inaccuracy is introduced because we are querying IPs of the providers' backbone where the distance in IP-space can be small while the geographical distance is in the range of hundreds of kilometers. This is a problem as the geo-location IP databases assign a location to a certain IP block.

For our thesis we want to find characteristic RTTs for connections between different places in the world. We are interested in the differences in performance in different locations. To show that different regions perform differently we need to compare links of different length with each other. Since the RTT between two European countries could hardly be compared with long-haul intercontinental links, it would be an option to use the metric RTT per distance [ms/km]. The problem is that this metric is only appropriate for connections starting at a certain length. Figure 3.6 shows all RTTs per distance for all tunnels and shows that for small distances the RTT per distance has a huge variance. It would still not be possible to compare short-distance links with long-distance ones and it would be hard to define a proper boundary between short and long distances to handle them differently.

If we assume that on every hop there is a certain queuing and processing delay, the delay per distance is much higher for short-distance links. Even short-distance links have generally about 10 hops at least even if the linear distance is only a few kilometers. The additional delay caused by the routing seems to be close to the two endpoints. This means that it needs many hops to

reach the geographically close Internet backbone gateway but only a few hops in the backbone to make the distances.

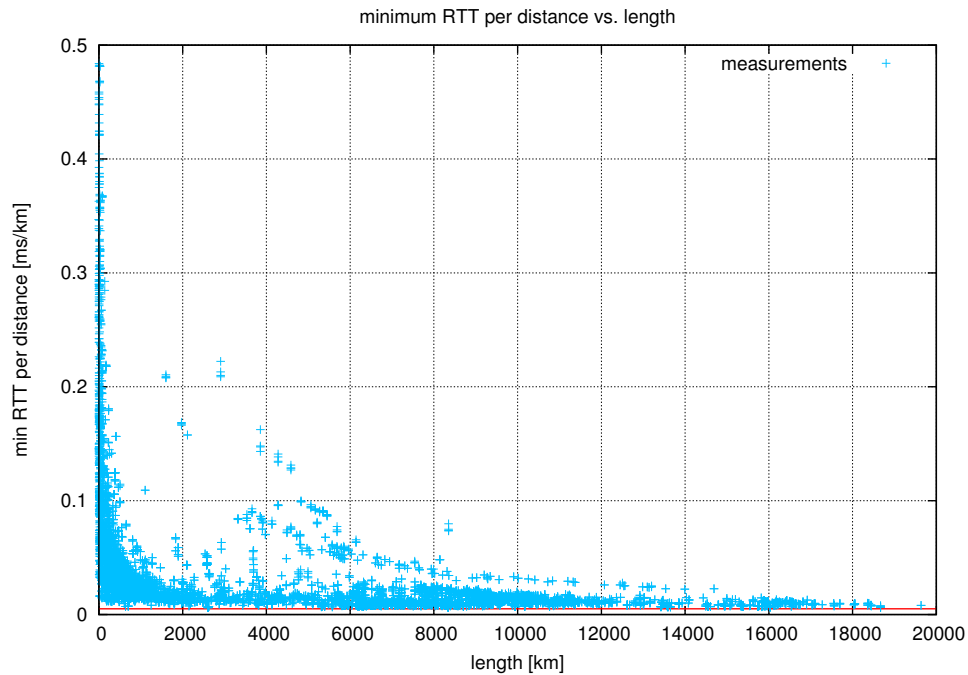


Figure 3.6.: RTT per distance for different distances between two endpoints

To still be able to compare links of different lengths among each other we try to find an approximation for the typical RTT in function of the distance. It should be based on the measurements we have, such that we can show the differences between the measurements and the approximation. To achieve this approximation we use the median RTT values of all links and their distances and calculate the median of these values for distance ranges of 500km. The median of the RTTs allows to rule out outliers. For these median values we calculate a linear spline interpolation. The calculation is done in Matlab and the script can be found in the appendix A.2. We evaluate different number of segments and the approximation for 1 up to 6 segments is shown in Figure 3.7.

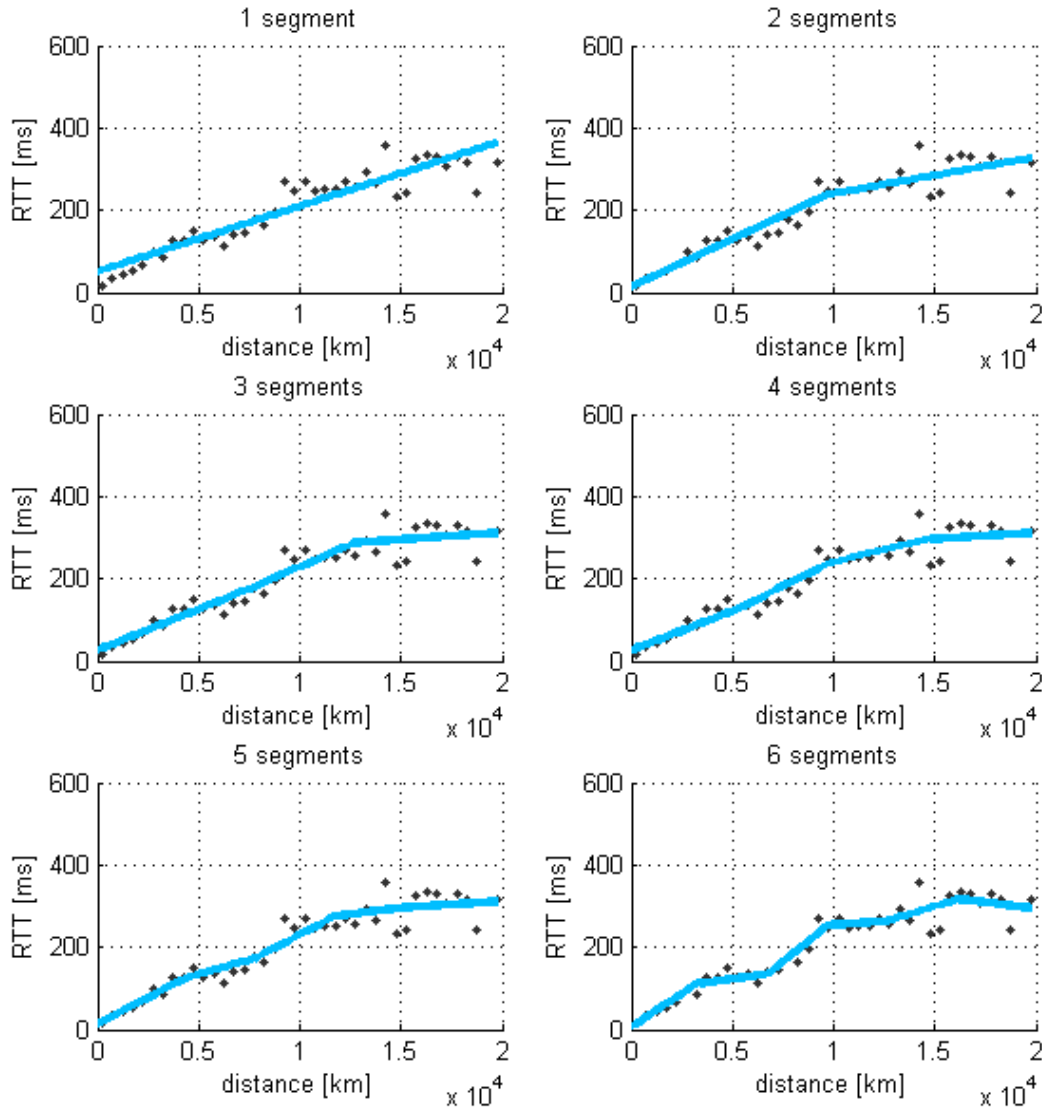


Figure 3.7.: RTT approximation with different linear segments

We analyzed the Root Mean Square Error (RMSE) for the different approximations and decided to take 4 segments each representing the RTT in its range. The resulting formula for the RTT in ms is plotted in Figure 3.8 and is defined as:

$$RTT_{\text{approx}} = \begin{cases} 0.020 * d + 25.4 & \text{if } 0\text{km} \leq d < 5250\text{km} \\ 0.025 * d - 5.5 & \text{if } 5250\text{km} \leq d < 9750\text{km} \\ 0.012 * d + 123.3 & \text{if } 9750\text{km} \leq d < 14750\text{km} \\ 0.003 * d + 251.4 & \text{if } 14750\text{km} \leq d \end{cases}$$

We use a linear fitting model because the minimum physical RTT is a linear function depending on the distance. This model will be used in section 3.2.2.

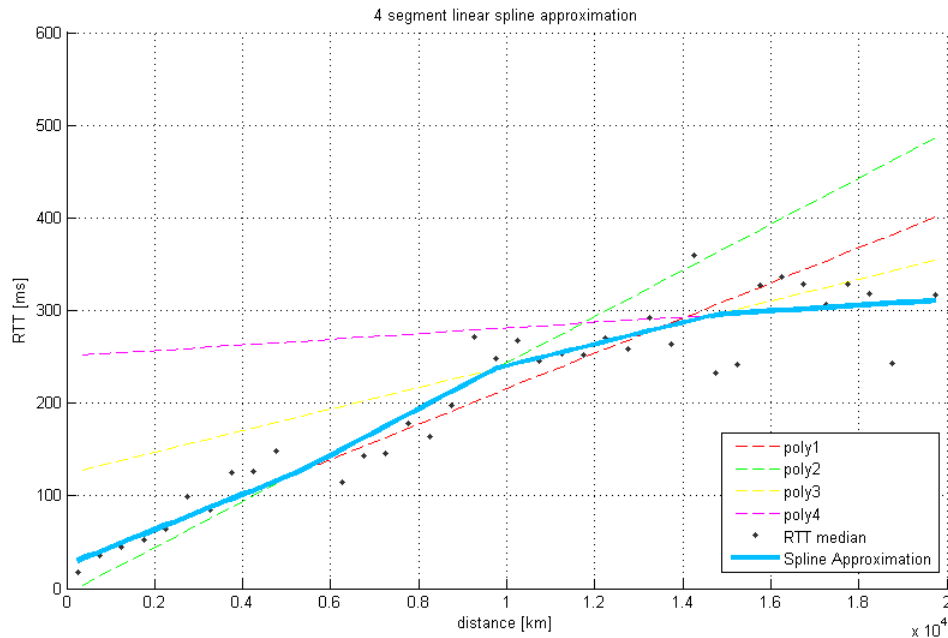


Figure 3.8.: Approximation for the minimum RTT

Loss Rate

The loss rate is another metric for quality. It indicates how many of the packets are lost. A lost packet means that we did not get a reply to our ping request because either the request or the reply was lost. There are different reasons for packet loss. It can be influenced directly by the endpoint itself if, for example, the load is too high or a firewall is blocking the request or the reply. A packet loss can also appear at every intermediate hop along the path from the sender to the receiver. A typical reason for packet loss is the congestion of a network. A packet loss caused by bit errors in a cable is very rare whereas for satellite connections bit errors are a lot more common.

Since we send 25 pings every 5 minutes and then store the loss rate that appeared for these probes, we have a maximum resolution of 0.04. We are aware of the fact that this measurement does not allow detailed interpretation for the loss behaviour of a tunnel. In particular it is not

possible to analyze the relation between loss and RTT. For example in the case of congestion the RTT should arise first because packets spend more time in the queue and packet loss should follow if the queues are full. Although this fine-grained analysis is not possible the average loss is still a good indication on the link's long-term behaviour.

Our analysis shows that the loss rate is not correlated to the distance between the two endpoints. We calculated the Pearson⁶ and Spearman⁷ correlation coefficient. Both would give an indication to a correlation if the values are close to 1. For the average loss rate the Pearson coefficient is 0 and the Spearman coefficient 0.3⁸.

In general we state that there is no formula that describes how loss can be estimated based on the distance between endpoints. Packet loss can appear in every device and on every line with a certain probability. The root-cause for packet loss is congestion in the network which causes dropping of packets. Figure 3.9 shows the distribution of the average packet loss for satellite and non-satellite links. We see that for non-satellite links the average packet loss rate is below 1% for almost 90% of the links whereas for satellite links packet loss appears more often. Many tunnels almost never suffer from packet loss.

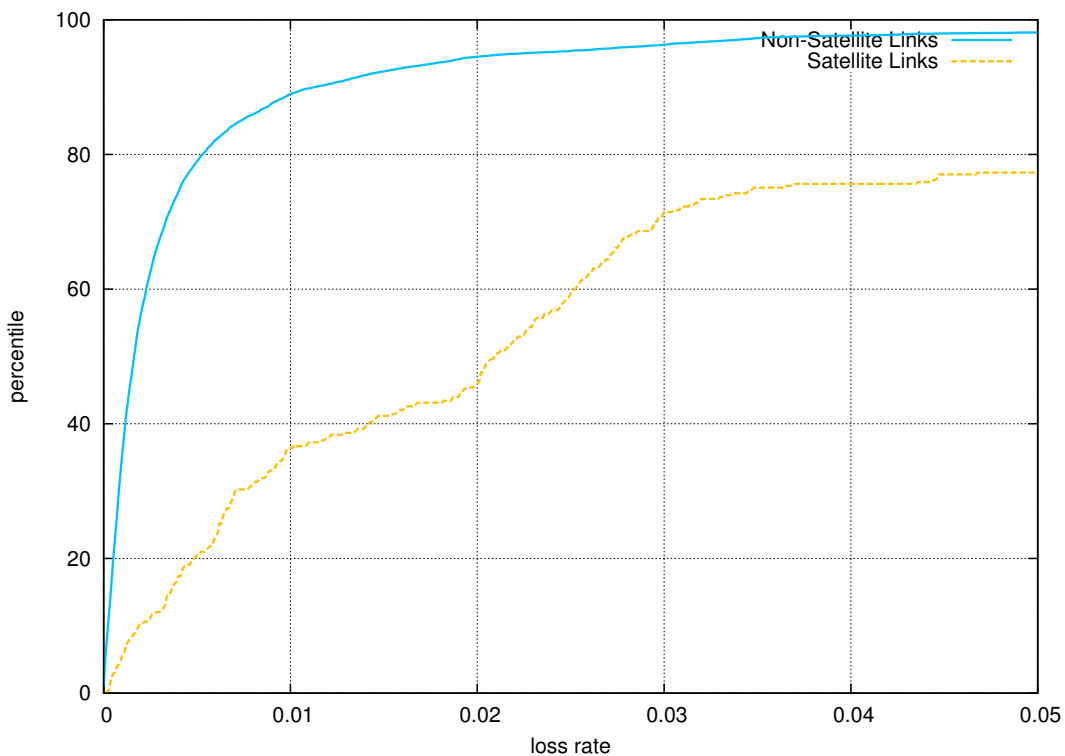


Figure 3.9.: The average packet loss rate distribution for different technologies

⁶Correlation Coefficient: <http://mathworld.wolfram.com/CorrelationCoefficient.html> (last visit: March 2012)

⁷Spearman Rank Correlation Coefficient <http://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html> (last visit: March 2012)

⁸Calculated with *R*: <http://www.r-project.org> (last visit: March 2012)

Packet loss directly affects the quality of a link. Lost packets either have to be retransmitted which causes additional traffic and uses bandwidth (e.g. TCP) or they are just missing and the application has to deal with it (e.g. UDP).

Tunnel Usage

The usage statistics that are available show the inbound and outbound load of each tunnel of an endpoint. It would be interesting to analyze whether packet loss occurs in connection with high tunnel usage. The problem is that the load on a single tunnel is only part of the overall traffic of a host such that the bottleneck is not given by the tunnel usage itself. The limiting factor is the bandwidth of the network interface which can be used by multiple tunnels and which also handles non-encapsulated traffic.

We will use the available tunnel load information for a different purpose. We can estimate how severe it is if a certain tunnel is performing badly. In general the impact is bigger on a link with more traffic.

Since we are constantly sending ICMP probes, there is always a certain amount of traffic independent of the tunnel usage. Figure 3.10 shows the usage of the links for one company. We see that there is a big difference among the tunnels and that almost two third of the links have less than 0.5kbps of traffic. These links are merely used as backup links and no traffic is actively routed to use these tunnels. The basic traffic on every link is caused by the probe traffic (ICMP) and the OSPF routing information exchange.

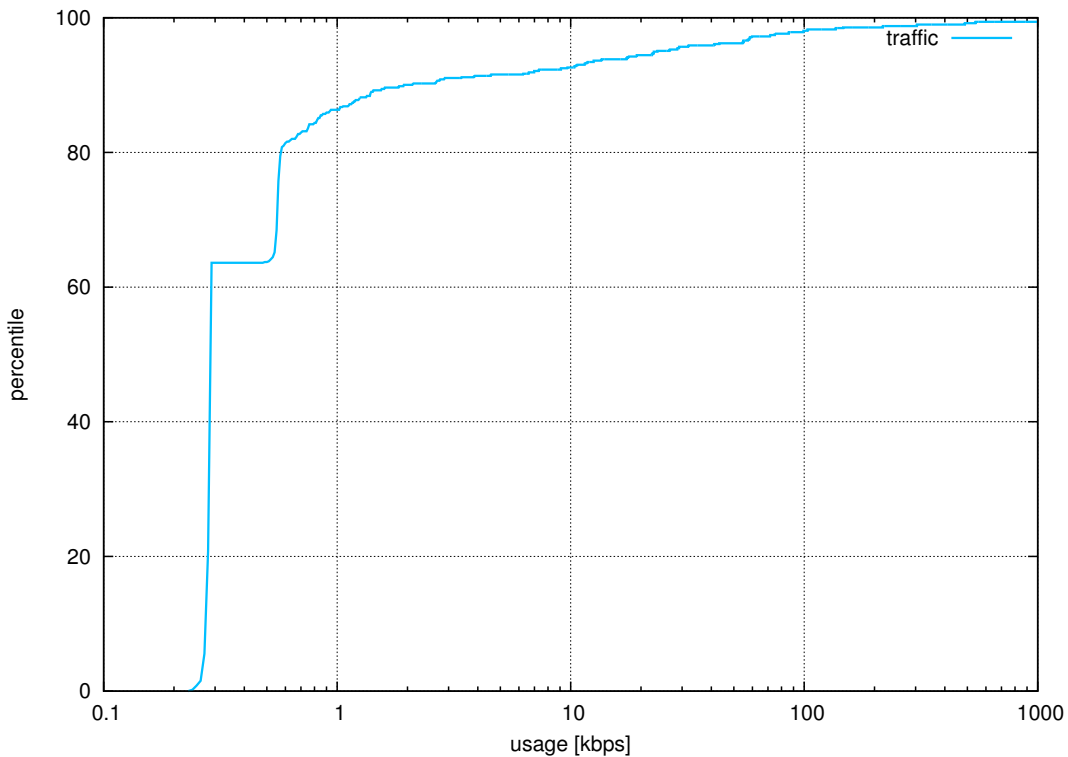


Figure 3.10.: The distribution of the tunnel usage for one company (log-scale)

Coordinates

The location of the place where a host is situated is already stored in a database. Since we are using positioning information for our analysis we have to rely on the coordinates assigned to each host. We check if the locations stored in the database are correct by using the Google Geocoding API⁹. For every host we use its coordinates and send a query with the following HTTP GET request:

```
http://maps.googleapis.com/maps/api/geocode/xml?latlng=[lat],[lon]&sensor=false
```

For [lat] and [lon] we enter the coordinates in decimal format. The returned answer in XML format is then parsed and the retrieved country and city are compared to the one in the database. Since there are many different ways how to write city names all the endpoints which have been reported to differ from the retrieved answer have been checked manually.

To verify the database locations regularly, the method could be improved using different look-ups. For example we could combine the method above with a query for the location (city, country, ...) which is stored in the database. We could use the following HTTP GET request and provide the city of an endpoint to receive its coordinates.

⁹Google Geocoding API: <http://code.google.com/apis/maps/documentation/geocoding> (last visited: Feb. 2012)

```
http://maps.googleapis.com/maps/api/geocode/xml?address=[city]&sensor=false
```

A comparison between the coordinates that are retrieved and the ones in the database would reveal errors too. If they differ too much this would indicate that the coordinates have been stored incorrectly. For our check of the coordinates in the database the first approach was sufficient.

3.2. Global View

In this section we show that a link's performance depends on the region the two endpoints are located in. We will analyze the performance in different countries and rate them.

3.2.1. Basics

It is a big topic of different research institutions to make world-wide performance measurements. As mentioned in the chapter about the related work the project *PingER* contains the reporting of minimum RTT measurements to locations all around the world. The main site which originally measured the RTT to the different locations is located in Stanford (US). Recently they also started to use different locations too. They also provide an annual *Network Monitoring Report* about the general RTT performance and some specific events¹⁰. The focus is to show how the Internet performance has evolved over time.

While the performance reports of *PingER* go back to the year 2002 the *RIPE Atlas* is a relatively new project. At the time that this thesis is being written RIPE NCC is building their own network of probes. They are distributed all around the world and operated by volunteers. The goal is to get detailed performance measurements in a full mesh network. It would be interesting to compare the data being collected to the results of this thesis. Or even to use information from the *RIPE Atlas* in addition. Unfortunately we do not have access to the data during this thesis.

¹⁰PingER: <http://www-iepm.slab.stanford.edu/pinger> (last visited: Feb. 2012)

3.2.2. RTT Locality

The experiments done with PingER already showed that the performances of all the tunnels from one continent to another are not as homogeneous as they should be to use one single threshold. Figure 3.11 shows the median RTT from Switzerland to all other countries.

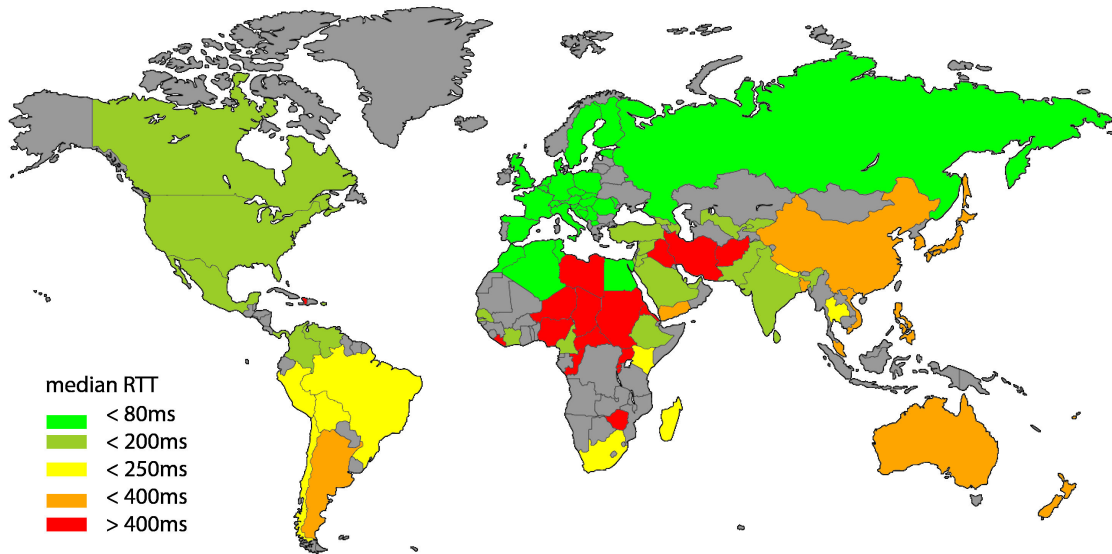


Figure 3.11.: The median RTT from Switzerland to all other countries

We can clearly see that the distance is not the only factor influencing the performance but that there are regions in the world which are still not well-connected to the Internet. At Open Systems so far only RTT thresholds were used to detect bad performing links. These thresholds are based on the continents the endpoints are located. Indeed it allows to get an impression about really bad performing links. However, we found that continental clustering is not detailed enough. Links from Switzerland to South-Africa perform a lot better in general than links from Switzerland to Ghana. We claim that besides technology also the path of the cable which connects two regions is important. An easy improvement would be to use thresholds per country connection. We go even further with our approach (see Chapter 4).

The evaluation of the RTT from Switzerland to the rest of the world indicates which performance differences exist for connections originating from Switzerland. It does not show how good a country is connected to the Internet in general because the links to Switzerland could be the only ones that are performing badly. To get a global view of regions that are suffering from high RTT values in general, we analyze every single country. If a country has only high RTT values to all its connected countries, this country is more likely to be the reason itself for the bad performance. If only a few connections are suffering from a high RTT and others are not, the reason can be on the path to the opposite country or even at the destination endpoint. In our approach we want to look at a more global perspective and rate every country based on

its connections to all other countries it is connected to. This should give us a general insight on how good a single country is connected to the Internet. To rate the countries we take the RTT approximation from section 3.1.3 and take the difference between the approximated RTT and the measured median RTT of a link.

$$\Delta_{\text{RTT}} = \text{RTT}_{\text{real}} - \text{RTT}_{\text{approx}}$$

For every country we take the median of all Δ_{RTT} values and color the country according to this value.

$$\Delta = \text{median}(\Delta_{\text{RTT}})$$

Figure 3.12 shows especially that connections from and to Africa and South Asia have a high RTT in general. For China i.e. we know that there exist different ISP offers. Links that are well performing for international connections are relatively expensive whereas a cheap line will not perform well for international traffic. In Africa, most countries do not have a cable landing station with high bandwidth cables or they might be only connected by using satellite connections.

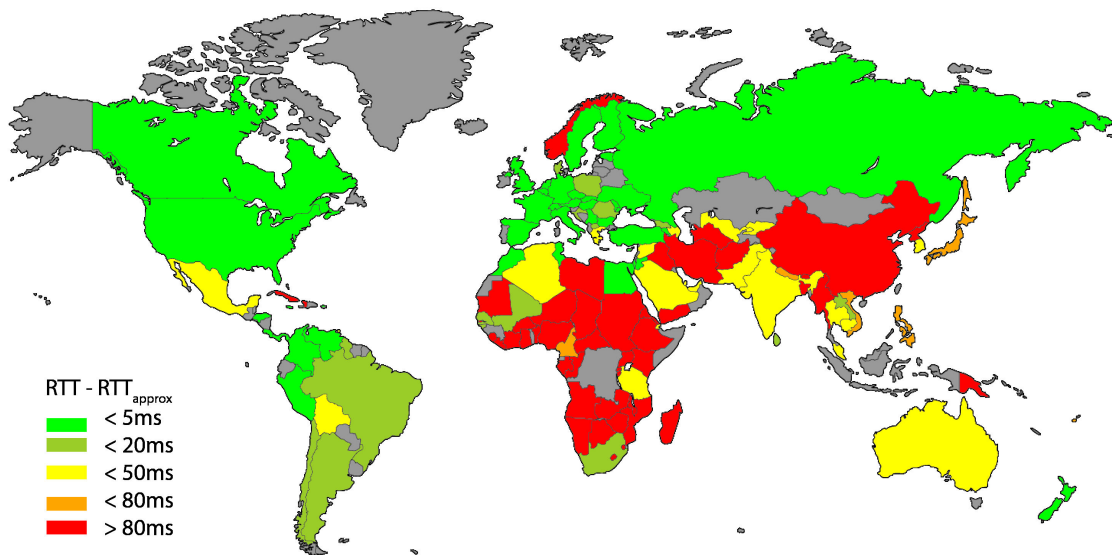


Figure 3.12.: Countries rated by the difference between the RTT of all their connections and $\text{RTT}_{\text{approx}}$

3.2.3. Loss Locality

We are also interested in whether the packet loss rate is dependent on the location of the VPN endpoints in the same way as the RTT is. To allow a quick analysis we proceed the same way as with the RTT. Figure 3.13 shows the average loss from Switzerland to the rest of the world. To represent the a typical average loss rate for tunnels between an arbitrary country and Switzerland, we calculate the median.

$$\text{Loss}_{\text{typical}}(\text{country}_c) = \text{median}(\{\text{avgLoss}_i | \forall \text{tunnel } i: \text{one endpoint in CH and one in country}_c\})$$

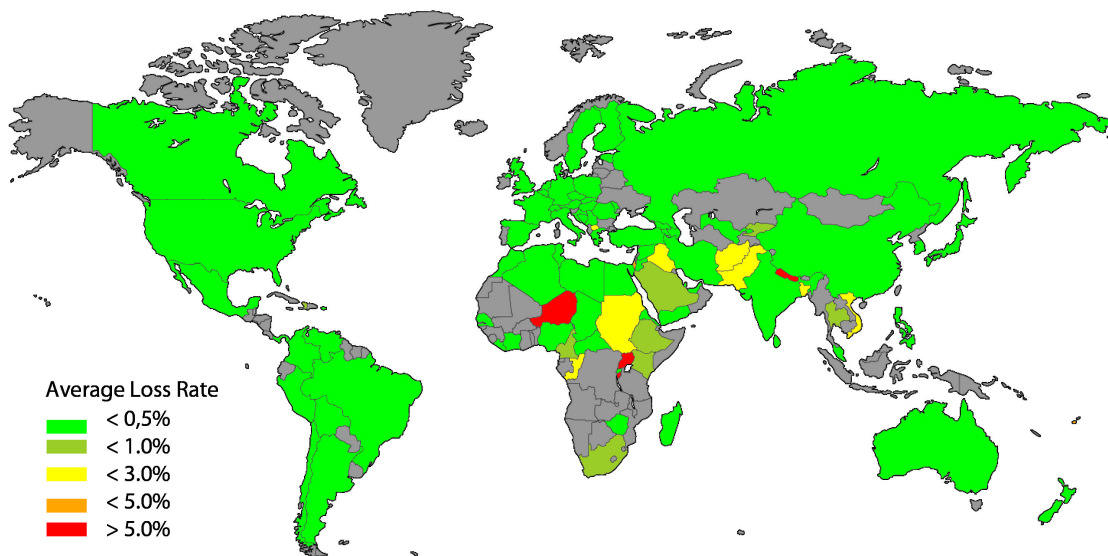


Figure 3.13.: Typical average loss rate for connections to Switzerland

We already see that in general in most of the countries there are endpoints of tunnels to Switzerland which do not suffer from extensive packet loss. To get an impression if the packet loss is depending on a country itself we also analyze for each country the loss of all its outgoing connections. Figure 3.14 shows the typical loss for a country's tunnels represented by the median of all the average loss rates.

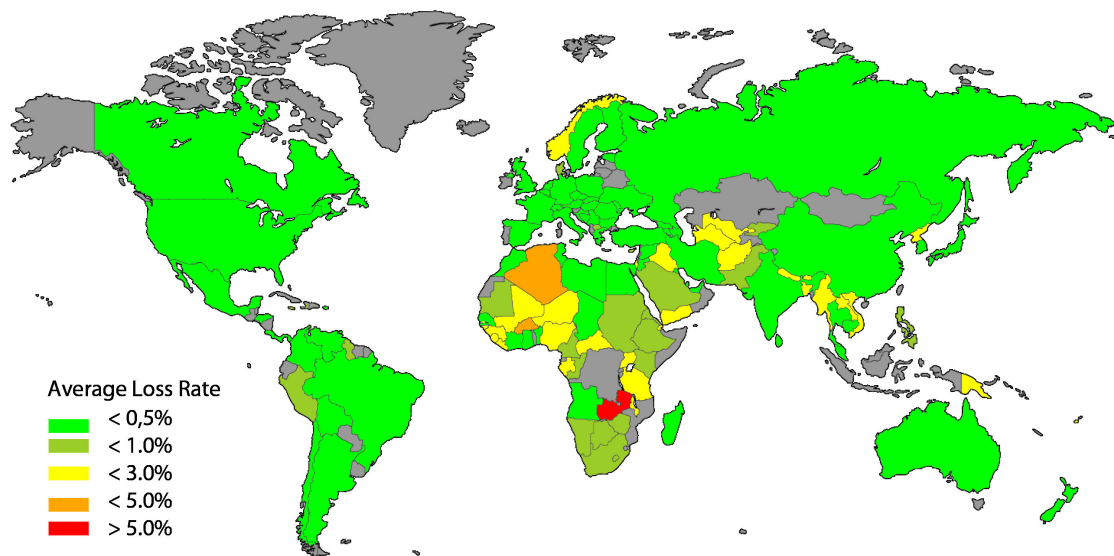


Figure 3.14.: Typical average loss rate $Loss_{typical}$ for a countries connections

Again we see that in most of the countries there exist to a major extent connections that do not suffer from big loss rates. We still analyze locality of loss in a finer-grained way in section 3.4.

3.3. Clustering

Since the RTT and packet loss rate are not simply based on the length of a tunnel, the analysis of a tunnel's performance gets complicated if we try to calculate an approximation for the performance and compare the actual performance to this approximation. If we also try to take the local performance differences into account to calculate an approximation it would get very complicated.

We decide to use a different method which is only possible due to the big network which we analyze. The idea is that two tunnels which connect the same two locations should perform the identically in theory. In the real world this is not exactly true because the performance is influenced by different paths and queuing characteristics. This results in the fact that we have faster and slower links and links with more or less packet loss, depending on the intermediate conditions between the endpoints.

Another approach typically classifies a connection as 'performing well in general' and detects anomalies. Since we are interested in the performance of the links in general this would be an addition to our rating.

If we look into more than one tunnel that connect the same two regions, the distributions of the RTT and packet loss rate give us an impression about the typical performance between these locations. If we are talking about same locations we need to define up to which distance an endpoint is considered to be in the same location as the other one. This leads to groups

of endpoints that represent a location. To build these groups we will use clustering algorithms which are described in this section.

We will discuss different clustering possibilities and analyze what additional information we get by using the cluster approach.

3.3.1. Algorithms

In general it is NP-hard to solve a clustering problem, even if there are only two clusters[1]. Nevertheless there are plenty of clustering algorithms that try to find good approximations for a clustering problem. We look at two well-known clustering algorithms: k-means and k-means++. Both algorithms allow us to automatically build clusters of VPN endpoints based on their geographical location. The endpoints are shown in Figure 3.15

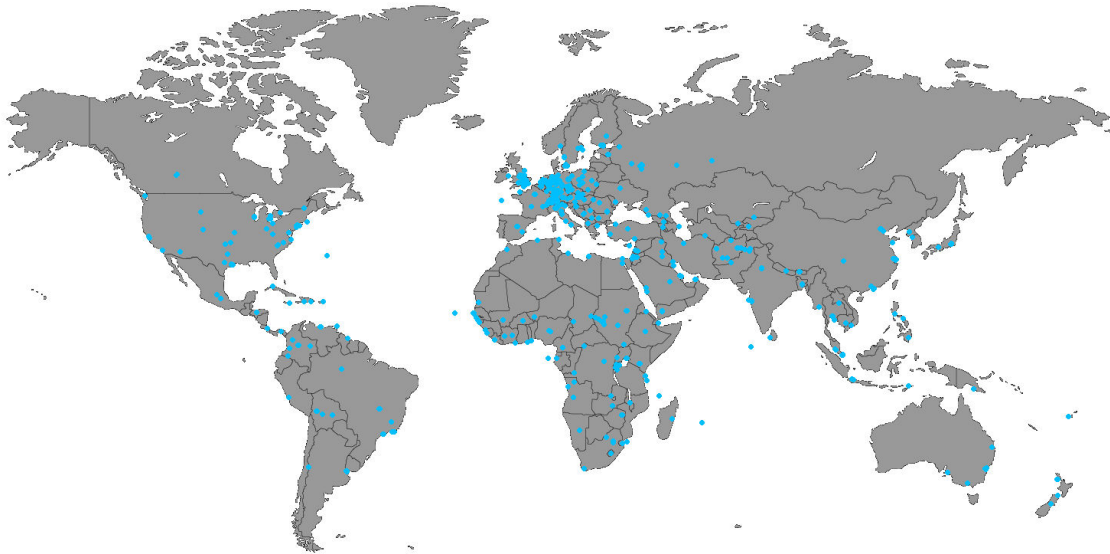


Figure 3.15.: Endpoints which need to be clustered

For the clustering of our network we have different criteria:

- geographical position
- cluster nodes should represent the 'center of communication'
- upper limit for the distance between a node and its cluster center

The clustering algorithm will assign every endpoint to a cluster node. In the resulting mapping we can define inter-cluster connections, if we see the tunnels as a connection between clusters (see Figure 3.16).

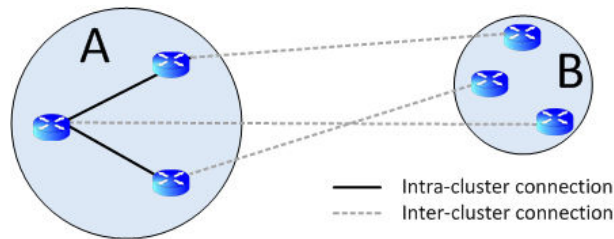


Figure 3.16.: Inter- and intra-cluster connections for cluster A and B

k-means

The k-means clustering algorithm is a simple and fast algorithm. It is guaranteed to find a solution but there are no approximation guarantees [1]. The algorithm selects initially k arbitrary cluster centers. There are two repeating steps: First the k-means assigns every host $i \in X$ its nearest cluster center. Second the center of the cluster is recalculated by finding the mass center of each cluster. These two steps are repeated until the mass centers do not change anymore.

Algorithm 1 k-means (Lloyd's Algorithm) [1]

- 1: Arbitrarily choose k initial centers $C = \{c_1, \dots, c_k\}$
 - 2: For each $i \in \{1, \dots, k\}$, set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all $j \neq i$.
 - 3: For each $i \in \{1, \dots, k\}$, set c_i to be the center of mass of all points in C_i : $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
 - 4: Repeat Steps 2 and 3 until C no longer changes.
-

k-means++

The k-means++ clustering is an adapted form of the k-means clustering algorithm where we initialize the cluster centers uniformly at random. We first select one cluster center at random. Then for every node, the minimum distance to the cluster center node is calculated and is used as a weight for a weighted uniform selection of the next cluster center. These steps are repeated until k cluster centers are found. The iteration after the initialization phase stays the same like in the k-means algorithm.

3.3.2. Cluster Number

Both algorithms presented need the number of clusters as an input but in our case the amount of clusters is not known or is dependant on a changing network structure and needs to be found. In addition the quality of a clustering is also dependent on the amount of clusters. There are different approaches which give an indication about what is a good clustering and selection of k .

Algorithm 2 k-means++ [1]

- 1: Choose an initial center c_1 at random from X .
- 2: Choose the next center c_i , selecting $c_i = x' \in X$ with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$.
- 3: Repeat 2 until we have chosen a total of k centers.
- 4: For each $i \in \{1, \dots, k\}$, set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all $j \neq i$.
- 5: For each $i \in \{1, \dots, k\}$, set c_i to be the center of mass of all points in C_i : $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
- 6: Repeat Steps 4 and 5 until C no longer changes.

Distance Criteria

As long as the maximum distance from any node to its cluster center is longer than a certain threshold, we increase the number of clusters. We analyzed this for the k-means and k-means++ algorithm which can be seen in Figure 3.17. Since the two algorithms are not deterministic we show the averaged result of 6 independent clustering rounds with the same input data.

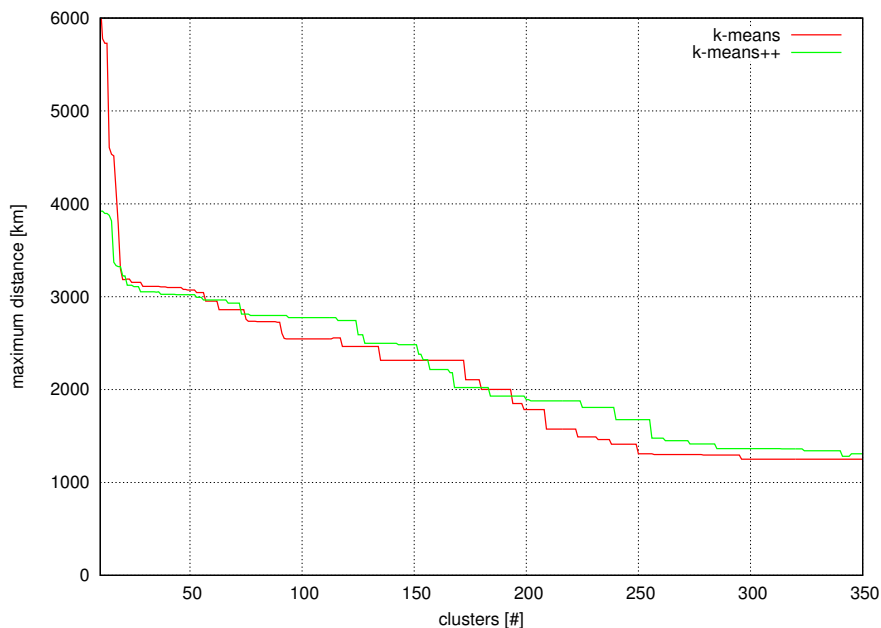


Figure 3.17.: Averaged distances for the k-means/k-means++ clustering algorithm

The distance of a node to its cluster center should not be too big since the RTT is directly depending on the distance. A tunnel with endpoints that are far away from the cluster centers will deviate in its RTT from a tunnel with endpoints that are close to the cluster centers. As we can see for both clustering algorithms, we need more than 250 cluster centers to fulfill e.g. a requirement of a maximum cluster radius of about 1500km.

Silhouette

The paper by J. and Rousseeuw [14] suggests a concept called silhouettes. If we cluster data with an arbitrary clustering algorithm, the silhouette concept allows us to assign a rating to the clustering that indicates how good it is.

Therefore we calculate for every data point i the average dissimilarity to its own cluster. In our case the dissimilarity is the distance between the endpoints.

$$a(i) = \text{average dissimilarity of } i \text{ to all other endpoints of } A$$

Where A denotes the cluster to which the datapoint i belongs to. Secondly we calculate the dissimilarity of i to all other data points in the other clusters C where $C \neq A$.

$$d(i, C) = \text{average dissimilarity of } i \text{ to all other objects of } C$$

$$b(i) = \min_{C \neq A} d(i, C)$$

Now we calculate the silhouette $s(i)$ which is defined to be:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

This results in a value $-1 \leq s(i) \leq 1$ whereas a value near to 1 indicates a perfect matched clustering of the node i . The node is assigned to the best cluster and its distance to any other cluster is a lot bigger than the distance to all the nodes in its own cluster. A value -1 would indicate that it is a poor matching. In general this would mean that a node i is rather similar to nodes of other clusters.

By taking the average of all $s(i)$ we get an indication on how good our clustering is. Now it is possible to find an amount of clusters where the clustering reaches an acceptable silhouette value. We use this approach for the two algorithms k-means and k-means++. Figure 3.18 shows the different averaged silhouettes for the k-means and k-means++ clustering algorithm. We see that the difference between the two algorithms is small.

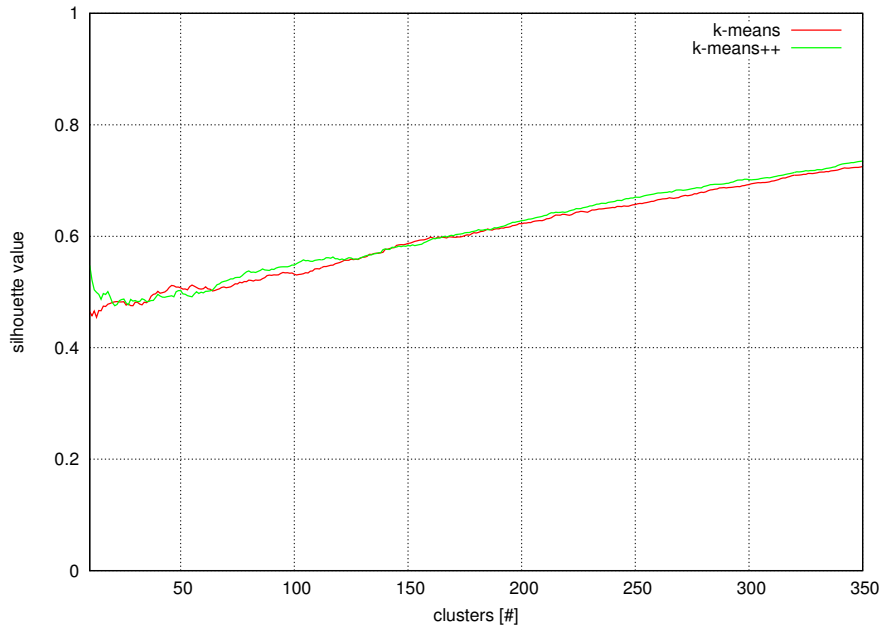


Figure 3.18.: Averaged silhouettes of the k-mean/k-mean++ clustering algorithm

The con of this method is the difficulty to define a value at which we declare a clustering to be sufficient. But it is an indicator for comparing different clustering algorithms.

3.3.3. Adapted Clustering

The two clustering algorithms presented, mainly focus on topological clustering. But we defined two additional criteria like that hosts with a lot of connections should tend to be cluster centers to find the 'centers of communication'. Also cluster nodes should not be too far away from their center nodes, which is not the case in either of the algorithms presented. Over 300 cluster centers are needed to have a maximum distance from a node to its center, which is still longer than 1000km. We suggest to adapt the k-means algorithm using the main idea of k-means++ to influence the selection of cluster centers. In addition we add a different initialization phase where we pre-select center candidates and some upper and lower boundaries to the cluster's radius. We also change the algorithm to be deterministic for the same set of data. We see an advantage in using a deterministic algorithm because it introduces stability to the evaluation method.

Initialization

We define a radius of $r = 5$ km and calculate for every host the amount of links it has in its radius. The 5km should represent the radius of a city or a district if the city is of bigger size. Therefore every endpoint x_i counts also the links that connect its geographical neighbors inside this radius to the world. The sum of all these tunnels gives a weight w_i for every tunnel i .

$$w_i = \text{tunnels of } x_i + \text{tunnels of nearby endpoints of } x_i$$

We build a sorted list of these centers of communication and define the node which has the highest weight as the first cluster node. We proceed accordingly for the endpoints in the list until we add a defined amount of initial cluster centers.

A node from the sorted list is only classified to be a cluster center if the distance to any other cluster center is bigger than 100 km. This limits clusters from being too close to each other. In addition, we define an upper boundary for the distance to a cluster center. If a node is more than 500 km away from the next center, it decides to build a cluster on its own. If clusters get too big the endpoints do not represent the same region anymore. We selected 500km because for bigger countries (China, USA, Brazil, ...) this enforces multiple clusters per country.

The advantage of the fix list of nodes that we add is that the more important nodes become, the more likely they are a cluster center or are close to one. The maximum distance boundary guarantees that no node is too far away from its cluster center.

Processing

After the initialization is done the remaining nodes are assigned to be part of their closest cluster center c_i . If the closest cluster center is more far away than the allowed maximum cluster radius it builds its own cluster. The cluster centers are then shifted towards the mass point of a cluster in the same way as in k-means or k-means++ with the exception that we define a node to be a cluster center and not a virtual point in the mass center. This process is repeated until there is no change in cluster allocation anymore.

To add the nodes in a sorted way implies a bigger density of clusters in areas of a dense network. This implies as well that clusters in a dense area are in general smaller. This represents the characteristic of the network we are looking at. The maximum distance guarantees that no node is too far away from a cluster center and therefore represents the stop condition. The result is shown in figure 3.19.

Algorithm 3 adapted k-means++

```

1: variables:
2: WEIGHTRADIUS=5km,  MAXINITCLUSTER=140,  MINRADIUS=100km,  MAXRADI-
   DIUS=500km
3: input:
4: Assign every endpoint  $x_i \in X$  a weighting  $w_i$  based on the number of outgoing links and the
   links of its geographical neighbors in a radius smaller than WEIGHTRADIUS.
5: Sort the weighted set with decreasing weight
6: initialization:
7: while amount of clusters  $|C| \leq \text{MAXINITCLUSTER}$  do
8:   if minimum distance to any  $c_j \in C$  is bigger MINRADIUS then
9:     add this node  $x_i$  to the cluster centers  $C$ 
10:  end if
11: end while
12: processing:
13: while  $C$  changes do
14:   For each  $j \in \{1, \dots, |C|\}$ , set the cluster  $C_j$  to be the set of points in  $X$  that are closer to
      $c_j$  than they are to  $c_k$  for all  $j \neq k$ .
15:   For each  $i \in \{1, \dots, |X|\}$ , if the distance to closest cluster is bigger than MAXRADIUS
     the node builds its own cluster and is added to  $C$ .
16:   For each  $j \in \{1, \dots, |C|\}$ , set  $c_j$  to be the center of mass of all points in  $C_j$ .  $c_j =$ 
      $\frac{1}{|C_j|} \sum_{x \in C_j} x$ .
17:   For each  $i \in \{1, \dots, |X|\}$ , cluster center  $c_i$  takes the position of the closest node to this
     position.
18:   Every node  $x_i \in X$  which has a distance greater than MAXRADIUS to its cluster center
     builds its own cluster
19: end while

```

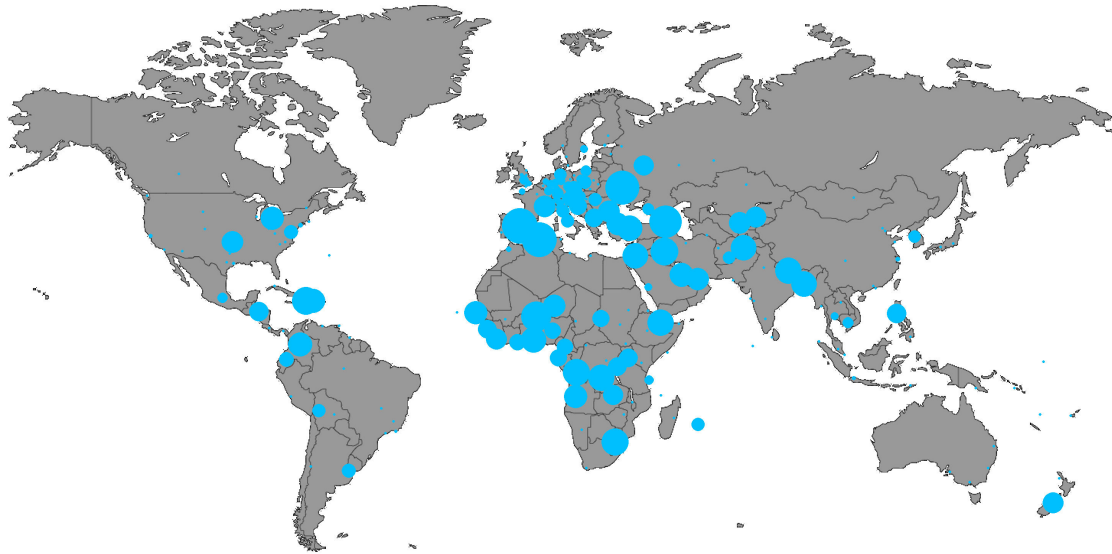


Figure 3.19.: Clustering of the VPN endpoints

Also for the adapted algorithm we calculated the silhouette and the maximum distance for a different amount of initial clusters. For the silhouette we can see that it is higher than for k-means and k-means++ (see Figure 3.20). We selected the amount of initial cluster centers to 140, which results in 203 clusters in our setup. The plot about the maximum distances only shows that the distances are smaller or equal to the allowed maximum and therefore fulfill the condition.

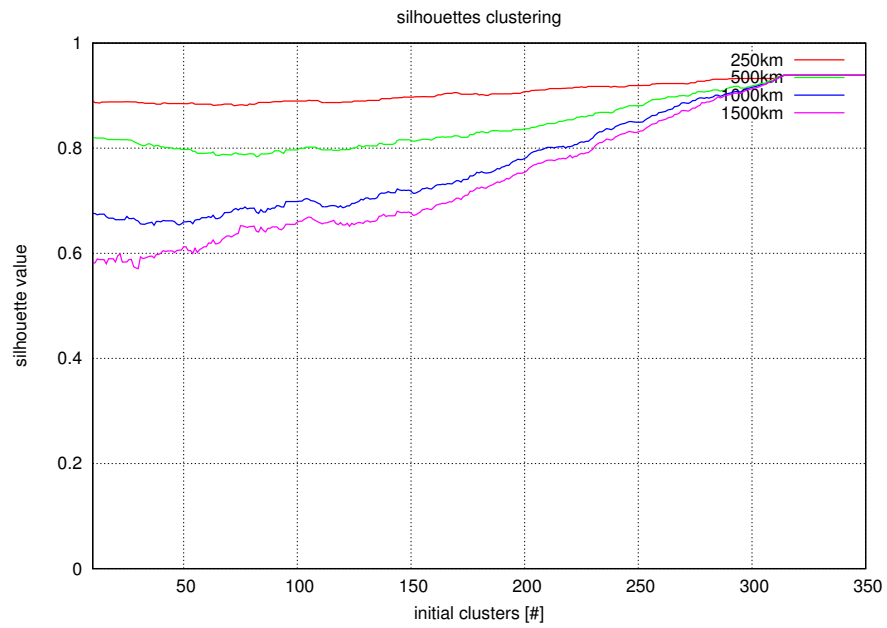


Figure 3.20.: Silhouette of the adapted clustering for different allowed maximum distances

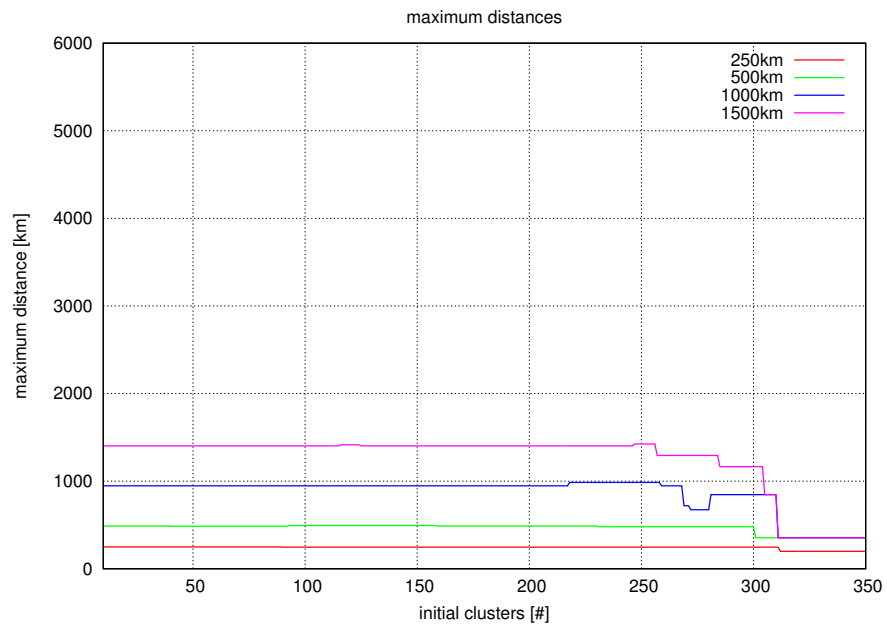


Figure 3.21.: Maximum distances between any node and its cluster-center

3.4. Clustered Endpoints

Due to the geographical clustering of the endpoints, we are able to analyze the characteristics of tunnels between different locations. We can do a similar analysis like for the different countries but in a finer-grained way which would allow almost the granularity of cities. In addition we will show that among different tunnels between two regions there is a certain similarity in the RTT behaviour. This is crucial for our rating idea presented in chapter 4.

3.4.1. Tunnel Analysis

Connections between clusters merely will use the backbone of the Internet to send packets over long distances. As an example we take all tunnels that connect the two clusters in Zurich and London. In total there are 50 links. Figure 3.22 shows the distribution of the median RTT of all the different tunnels. We can see that about 70% of the tunnels have a median RTT below 40ms.

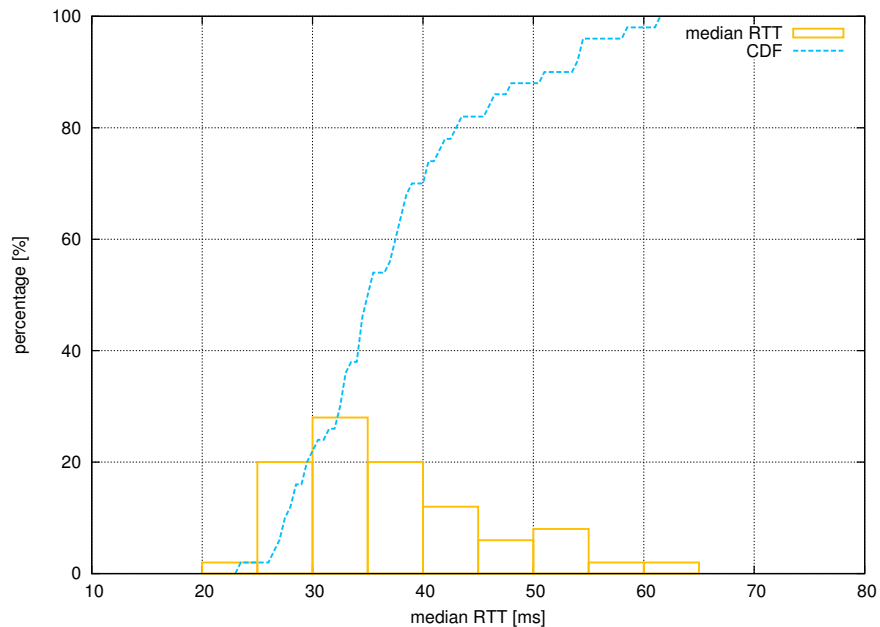


Figure 3.22.: Median RTT distribution for VPN tunnels between Zurich and London

The distribution allows defining a baseline for the RTT. Just looking at this example we could say a median RTT of 30 ± 10 ms is a reasonable value. Everything above 50ms seems unreasonably high. The analysis of the root-cause for a higher RTT is quite complex since there can be several reasons. A high RTT can be caused for example by inefficient routing or peering whereas a packet from an endpoint needs unusually many hops to reach the backbone.

For the loss rate we can do the same. But since the tunnels between Zurich and London do not suffer from any loss, we do not see a characteristic loss rate distribution, except that there are

no problems between these clusters (Figure 3.23).

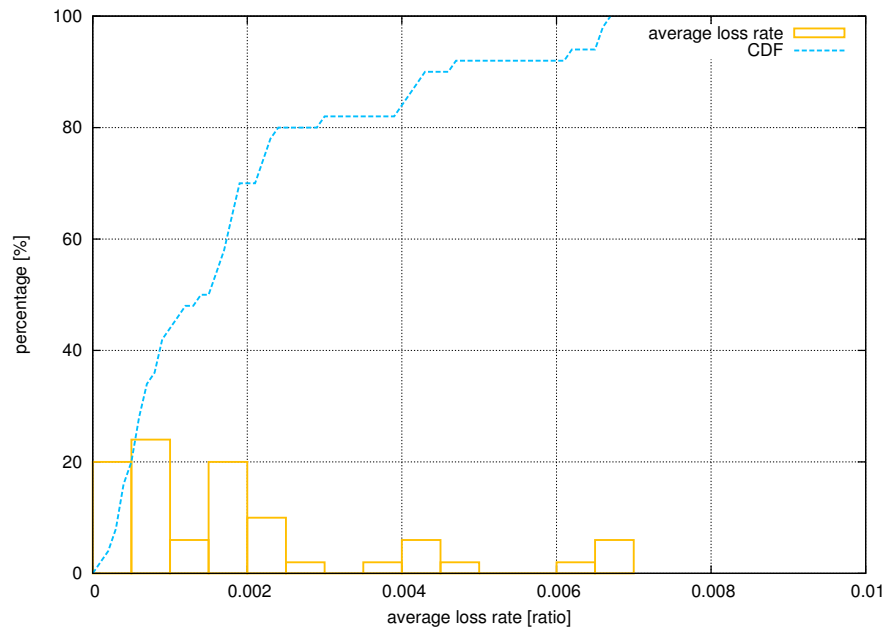


Figure 3.23.: Average loss rate distribution for VPN tunnels between Zurich and London

Although we also discovered differences in the loss behaviour, the loss rate is typically not related to the tunnels between two regions. A packet loss problem is rather related to the endpoint itself or to the Internet connection of an endpoint.

3.4.2. Cluster Rating

Similar to the country rating we can rate the clusters' RTT and loss rate.

RTT Figure 3.24 shows the rating of the clusters similar to the rating of the countries in section 3.2.2. In, e.g., China we see that there is a difference within the same country (see Figure 3.25). The coast-line in south-east China (Hong Kong, Macao, Shanghai) is performing better than the rest in China. Similar for Australia or Brazil. We see that Sydney has in general lower RTT values than the rest of Australia. In Brazil, Rio de Janeiro and Sao Paulo are also showing lower RTTs.

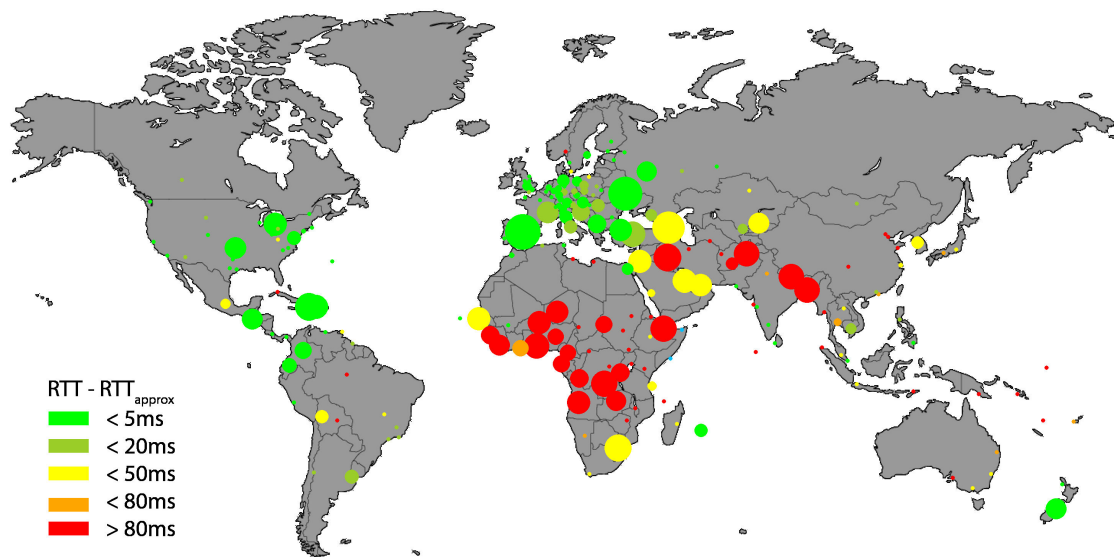


Figure 3.24.: Clusters rated by the difference between the RTT of all their connections and RTT_{approx}

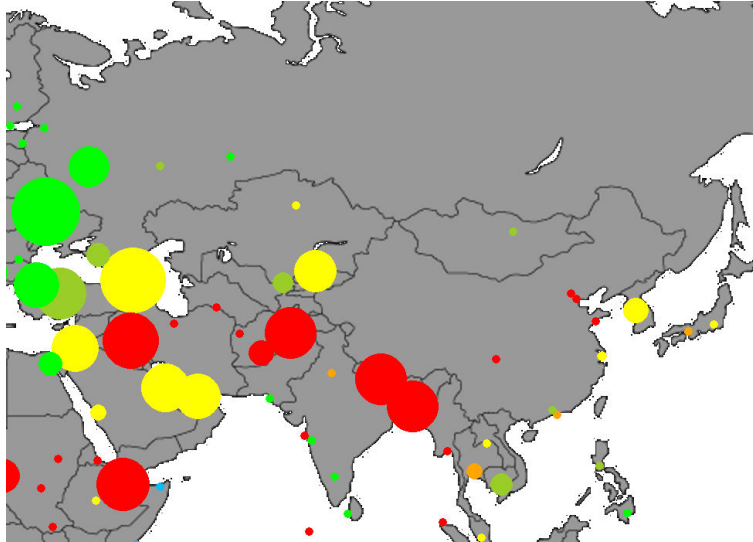


Figure 3.25.: Clusters rated by the difference between the RTT of all their connections and RTT_{approx} (China)

Loss Rate Figure 3.26 shows the situation for the average loss rate. Again a cluster is colored according to the median of all tunnels' average loss rate.

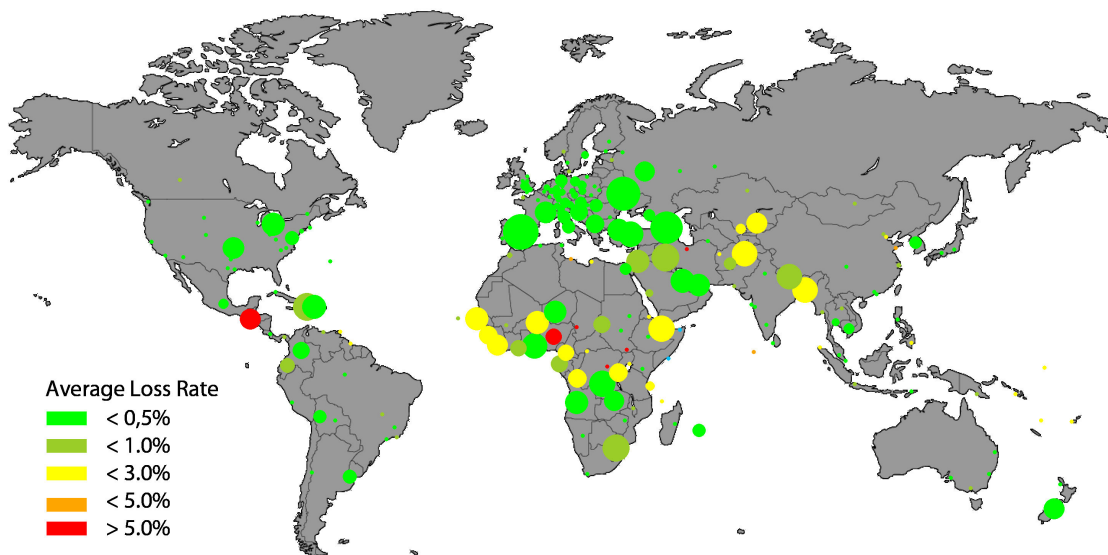


Figure 3.26.: Clusters rated by their average loss rate

3.5. Stability and Availability

Although the focus of this thesis is the RTT and loss rate, we would like to mention also the stability of a link. Evaluating the stability of a link is an important addition to the typical RTT and loss rate. When we analyze the stability we are interested in the way the RTT and loss rate change over time.

Especially in this field, there has been intensive research. All anomaly detection systems that rise alerts if the network performance suddenly changes, analyze stability. There was as well a thesis done by Wagner [28] previously at Open Systems which used anomaly detection to detect bad performance of VPN tunnels.

In this thesis we are interested in how to use stability information as a complement to median or average values. Especially in the case where the RTT is oscillating a lot, an average or median value can still be relatively normal, whereas a tunnel can be almost unusable. Zimmerli [30] already found out in his thesis that RTT values are heavy-tailed and therefore the inter-quartile range (IQR) can be used to measure the RTT stability.

The availability of VPN tunnels can be evaluated by analyzing how often a connection was down or a tunnel was broken.

CHAPTER 4

Rating Methods

In this chapter, we will discuss different possibilities how a tunnel can be rated and we describe our rating approach.

4.1. Rating Idea

To be able to rate a tunnel we need criteria for characterizing and comparing the differences in performance. Based on these criteria we have to find a method which allows to rate a tunnel's performance to be above or below standard. There are various metrics that could be used as criteria to define a tunnel's performance. The most important ones are:

- round-trip time
- loss rate
- performance stability
- availability
- jitter
- bandwidth

In this thesis we focus on the RTT and the loss rate of a tunnel. Stability and availability are also taken into account as long as the information can be retrieved using RTT and loss rate information. Due to our measurement data resolution, jitter analysis is not possible and is disregarded. Also we do not measure the bandwidth because this would lead to excessive additional traffic which would affect the throughput of the customer's network.

The goal of the rating is to be able to evaluate if a tunnel is performing as it should, in a long-term perspective. Therefore the aim is to figure out how good a link between two locations should perform and then evaluate whether the tunnel under test performs accordingly.

4.2. Performance Reference

In chapter 3 we described that especially the RTT is depending on the tunnel's distance as well as on the locations of the endpoints. To get a reference value we could 'calculate' a hypothetical RTT based on the distance between the endpoints or the length of the intermediate paths and compare this RTT to the measurement. The difference can then be evaluated and used for the rating. We suggest a different approach though.

Thanks to the large global VPNs operated by Open Systems, we have access to the performance measurements of hosts all over the world. Instead of only looking at one link, we want to use the information of multiple links to calculate a performance reference. If we rate one tunnel, we want to use the information of all links that connect similar locations. This allows us to retrieve a reference which indicates how good a link between two regions normally performs. It represents a generic connection between two regions. Links that connect the same two regions should perform similar (see section 3.4.1). On the other hand if a tunnel is performing differently it is an indication that it suffers from a performance issue.

4.2.1. Data of Interest

We already specified how to rate the RTT, packet loss, stability and availability of a tunnel. Now we need to decide which statistic values represent the performance best.

RTT

Basically there are two things that can be rated. The first one is related to a specific service that has its requirements for RTT. For Internet applications, like VoIP (Voice over IP) or video conferencing, there are different thresholds than for non-interactive applications such as bulk data transfer. Many researchers tried to evaluate what RTT should not be exceeded. For example, the ITU-T recommends for general networks a one-way delay below 400ms but indicates the user satisfaction for voice to be good for one-way delays below 150ms [13]. For the RTT we could get an estimation if we double the one-way delay thresholds[13]. Also Calyam et al. [2] analyzed how delay affects real-time multimedia. The quality levels he defined are shown in table 4.1.

	Thresholds
Rating	one-way delay [ms]
good	≤ 150
acceptable	≤ 300
poor	> 300

Table 4.1.: Quality levels for one-way delays according to Calyam et al. [2]

But if we have a tunnel from Europe to Australia, a one-way delay of 150ms is hardly possible. We can not rate this tunnel to be bad. Of course there will be restrictions to real-time media quality but it might be the best possible performance. The RTT rating should indicate if a tunnel is performing below standard with respect to the distance between the endpoints.

We are interested in a typical value for the RTT of a connection. Since the RTT distribution is heavy-tailed [30] it makes sense to use the median as a typical RTT value for a tunnel. In general there exist different patterns for the RTT. Some tunnels are performing with an almost constant RTT where the typical value is already sufficient. Other tunnels have a daily pattern or a very unstable behaviour. Hence it is also interesting to know the RTT in times where a tunnel has a higher RTT because of daily traffic patterns. Especially during office hours there is more traffic flowing through an ISP's network and it could cause congestion, which leads to an increase in the RTT. To get a measure in case of a daily pattern we use a sliding window approach. Figure 4.1 shows the sliding windows (blue) which are shifted through the days. The windows have a size of 4 hours and we calculate the median RTT for all measurements which are in the sliding windows. The timespan with the worst median RTT is called the '4hr median RTT'.

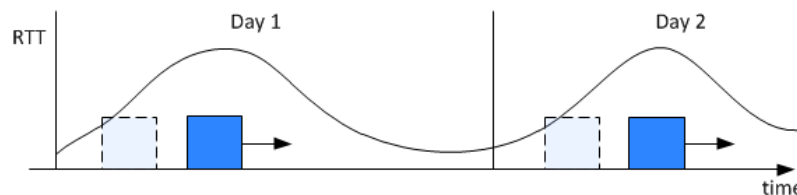


Figure 4.1.: Sliding window approach to calculate the median RTT of the worst 4 hours

The minimum RTT values do not represent the typical performance for a link but allow an assumption of the propagation delay for a connection.

Loss

The impact of packet loss is depending on the service used. In the early Internet the main focus was not on interactive communication. E.g. E-mail or FTP applications can deal with a relatively high packet loss. Today's interactive applications are a lot less tolerant concerning packet loss. Various studies have been made about packet loss and its effects on connection quality [2, 5, 15]. It was analyzed to what extent packet loss can appear without affecting the quality of real-time media. It seems hard to define quality levels by only looking at the packet loss rate. The quality levels for packet loss for H.323 traffic defined by Calyam et al. [2] are shown in Table 4.2.

	Thresholds
Rating	packet loss rate [%]
good	$\leq 0.5\%$
acceptable	$\leq 1.5\%$
poor	$> 1.5\%$

Table 4.2.: Quality levels for packet loss according to Calyam et al. [2]

In our setup we are interested in a long-term rating of packet loss where many tunnels do not suffer from packet loss and therefore they have an average loss rate close to zero. An often-used metric for loss is the average loss rate. The problem is that we are not able to distinguish a link that has a constant small amount of loss from a link that has only a high packet loss for a short time. Also the total average packet loss can hardly be compared to quality levels found by other researchers since most tunnels seldomly suffer from packet loss. To overcome this problem we suggest two possibilities:

- **Worst Time** Similar to the RTT, we look for the worst 4 hours of the day and calculate the average loss rate for this time. We found that the packet loss is following a daily pattern to a lesser extent than the RTT. This is why we will not use this approach in first place.
- **Split the Average** For a more detailed analysis we split the average loss rate measurements in a *time with loss* and a *time without loss*. Most links do not suffer from packet loss for most of the time. For this reason we analyze how often a link has loss. We calculate the average packet loss for the times a link is suffering from packet loss. The average packet loss can still be retrieved from these two values.

$$\text{loss}_{avg} = (\text{time with loss [rate]}) \cdot (\text{loss}_{avg} \text{ while loss appears})$$

Stability

Getting an insight into the stability of a tunnel is the most difficult part because there is no single stability metric. Stability shows how RTT and the loss rate characteristics change over time. The more stable a link's performance is, the more predictable and reliable it is.

- RTT Stability** Known stability measures are the coefficient of variation (COV) or the inter-quartile range (IQR). Zimmerli [30] already stated in his thesis that it is preferable to use the IQR rather than the COV because of the heavy-tailed distribution of the RTT.

The IQR is a measurement on how much the RTT is oscillating. The higher the IQR, the less typical the median RTT value for a link is. We found out that the IQR is not dependent on the distance. Figure 4.2 shows the distribution of the IQR for different distances. We do not see a general increase in the IQR for increasing distance. Therefore we will use the absolute value for the IQR as criteria for stability.

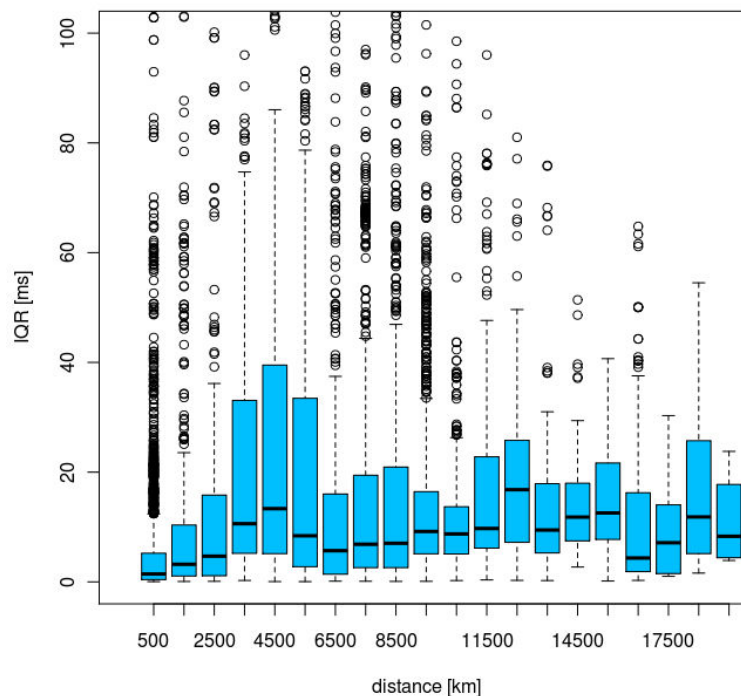


Figure 4.2.: RTT-IQR for different distances between the endpoints of the tunnels

The smaller the IQR, the smaller the changes are in the RTT over time. The absolute RTT distribution over all tunnels is shown in Figure 4.3. We see that more than 50% of all the tunnels have a RTT-IQR that is within a range of less than 5ms.

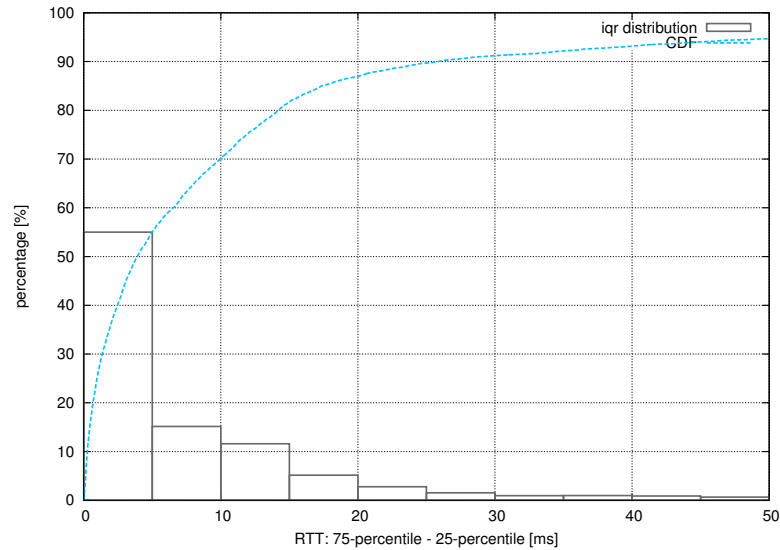


Figure 4.3.: RTT-IQR distribution

The difference between the overall median RTT and the one of the 4 hours, in which the RTT is the highest, is interesting too. The higher the difference, the more a link's performance shape is characterized by some daily pattern.

It would also be interesting to detect periods where the RTT suddenly increases because of routing changes. If a tunnel suffers from many routing changes the IQR will be high although a link still performs stable but on multiple RTT levels.

- **Loss Stability** For the loss rate the IQR is not sufficient. Links that are not explicitly suffering from packet loss have during more than 75% of the time a packet loss of 0. This would result in an IQR equal to 0 in all these cases.

We decide to take the inter-decile-range (IDR) which shows the difference between the 10th and 90th percentile.

$$IDR_{loss} = loss(90^{th} \text{ percentile}) - loss(10^{th} \text{ percentile})$$

It would be interesting to analyze the patterns of loss too. The amount of periods of loss or the average length of them are indicators showing even more detailed how loss appears. It would show if loss appears in a constant way or in bursts.

Availability

The worst thing that can happen regarding connectivity, is a tunnel that is down. Therefore we have to look at this performance indicator and rate the availability of a tunnel. If 100% packet loss appears and we are not able to get a RTT measurement, we call a tunnel *down*. The figures that we use are the time a tunnel is down (down time) and the number of independent periods of down time (down events).

4.3. Comparison of Tunnel Performance

To rate a tunnel, we use the performance information of other tunnels that connect geographically similar hosts. A comparison of the measurements among different tunnels is possible because of the fact that we evaluate long term performance. The first approach would be to use the clustering we used in section 3.3.3. That is, for every tunnel that we rate, we take the information of all tunnels that connect the same two clusters (see Figure 4.4).

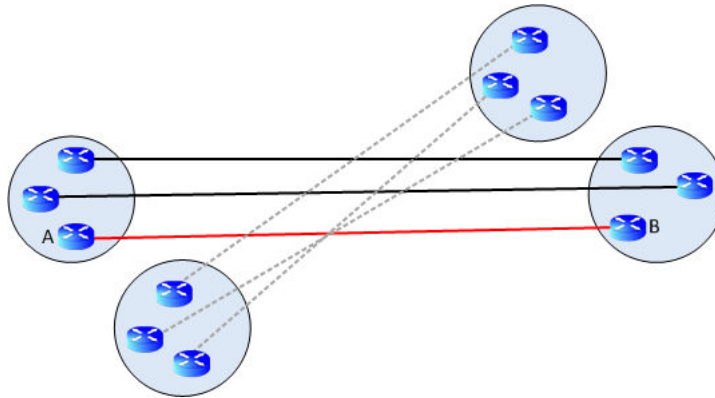


Figure 4.4.: If a tunnel A to B should be rated, we take the information of the tunnels which connect the same two clusters (black line)

To have a good reference we need a certain amount of similar tunnels such that this method is limited to cases where we have enough tunnels between two clusters. If we stick to the mentioned approach we could only rate about 60% of the tunnels if the clusters would have to be inter-connected with 5 tunnels at least. Figure 4.5 shows for how many tunnels a rating can be made if we ask for a certain minimum amount of tunnels between the two clusters.

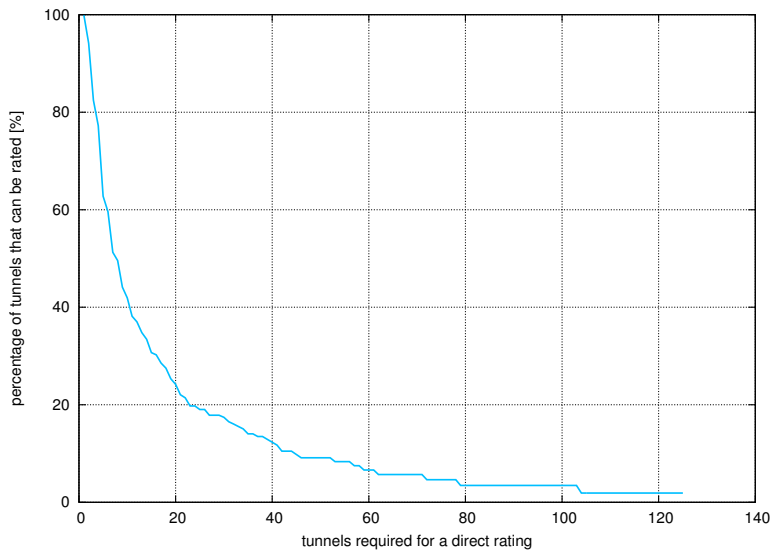


Figure 4.5.: Percentage of rateable tunnels for different numbers of minimum inter-cluster tunnels which are required as reference

The amount of tunnels that can be rated has to be increased. But it will be a trade-off between being able to make a good statistical statement about the reference performance and being able to give a rating for a high amount of tunnels. In order to reach this we need to get additional tunnels which we can use to calculate the performance reference.

4.3.1. Proximity Metrics for Tunnels

We discuss different proximity metrics for tunnels to find similar tunnels in general and to analyze the dependency of the rating quality on the similarity of the location of the VPN endpoints.

Similar Length

Just to look at connections which have the same linear length would solve the problem of not having enough links for a good statistical comparison. Since we would entirely neglect the locality of the endpoints, it is not a suitable solution. It would also only rate a link's performance in comparison to a global average. This approach was used to rate the regions in section 3.2.

Similar Clusters

To rate a tunnel which connects two clusters, we use all tunnels that are between the same two clusters. In addition we also could use the information of tunnels which connect clusters that are close to the original ones.

We ask for a certain amount of connections that are needed to rate a connection between two endpoints A and B. Figure 4.6 shows the situation where we do not have enough connections between the clusters of endpoint A and B and therefore it is not possible to give a reliable rating. We now start to look for connections that connect clusters close to the ones of endpoint A and B until we have the desired amount of tunnels. The bigger the search radius, the less related the additional tunnels are to the location of the original tunnel.

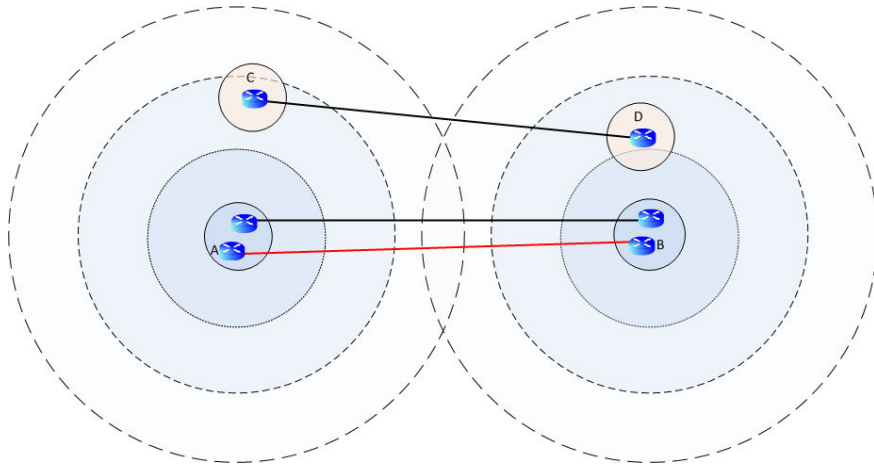


Figure 4.6.: If we want to rate the tunnel from endpoint A to B we take all tunnels which connect the same two clusters and in addition we look for tunnels which connect clusters which are nearby (tunnel from C to D)

Geographically Close Tunnels

Instead of strictly using the cluster criteria, we can define a distance based proximity metric. By defining a distance between tunnels, we are able to look for every tunnel individually for its closest and therefore most similar tunnels. This basically leads to a dynamic clustering for each individual tunnel's endpoints. Figure 4.7 shows the setup to calculate the proximity between two tunnels.

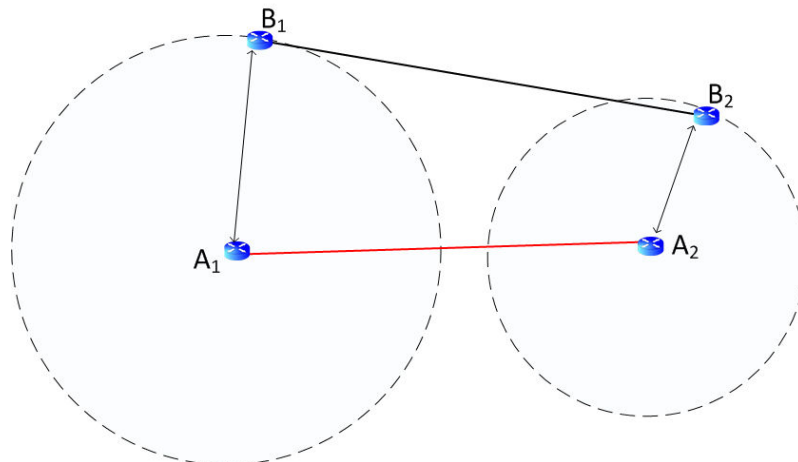


Figure 4.7.: Model for distance between close tunnels

- Total Distance** We only take the geographical distance of the endpoints into account to define the proximity of two tunnels. If we have a tunnel A connecting an endpoint A_1 with an endpoint A_2 and we want to know the distance to the tunnel B, the distance is defined to be the sum of the distances between the endpoints. First we select the endpoints which are closest to each other. In our case it is A_2 and B_2 . Then we look at the distance between the other two endpoints. The sum of the two distances is defined to be the distance between the two tunnels. In our case this is:

$$\text{distance}(A, B) = \overline{A_1 B_1} + \overline{A_2 B_2}$$

Figure 4.8 shows that the tunnels' median RTTs are most similar if the distance between the tunnels is small. The plot shows the absolute difference in RTT between a tunnel and the tunnels that are within a certain distance to it. To generate the graph we used 500 random tunnels and compared them to all tunnels which are in a distance smaller than the tunnel's length itself. The general conclusion is that with an increasing distance between two tunnels the similarity of the typical RTT is decreasing. The dispersion is increasing with increasing distance as the inter-quartile range (blue boxes) indicates, such that in general the similarity is decreasing.

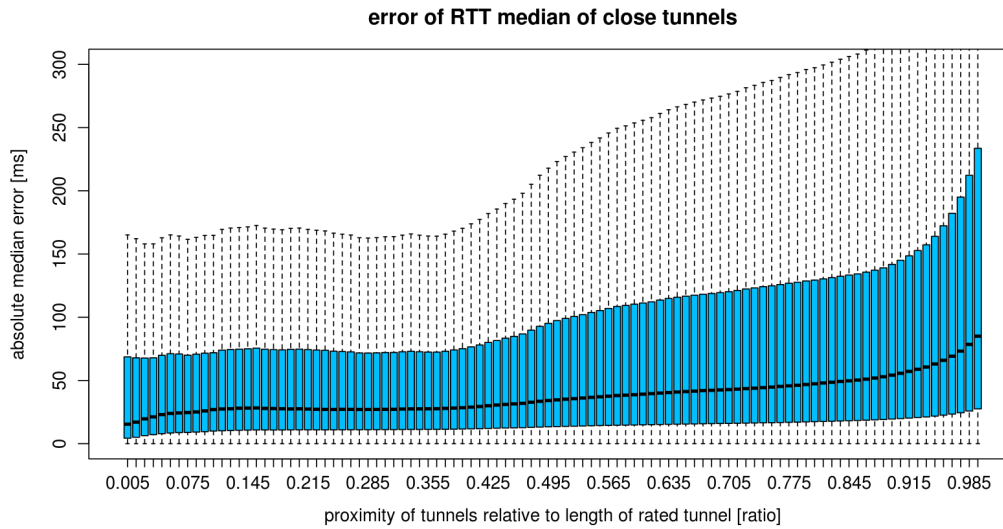


Figure 4.8.: RTT difference of tunnels depending on their distance to each other

- Maximum Distance** The approach above has the disadvantage that two endpoints can be relatively far apart from each other as long as the other two endpoints are close. Figure 4.9 shows the fact that the distance between tunnel A and B would be identical to the distance of A and C.

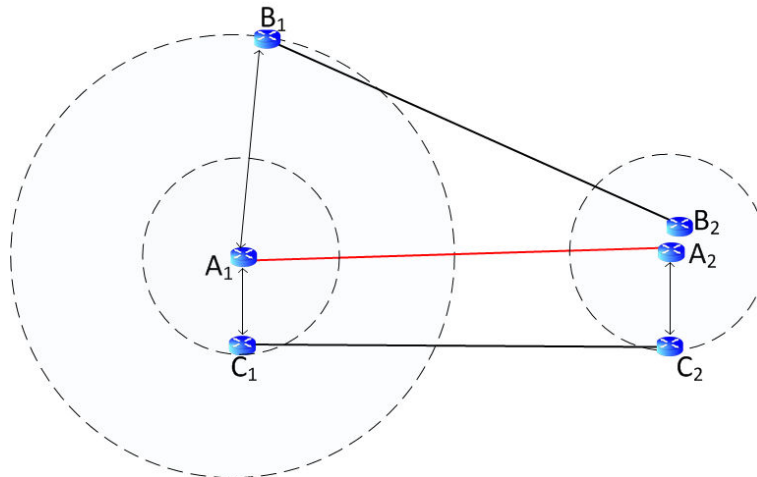


Figure 4.9.: The problem when using *total distance* as proximity: $\text{distance}(A, B)$ is equal to the $\text{distance}(A, C)$

To overcome this problem, we change the definition of the distance to be the maximum distance between the endpoint pairs. For the calculation of the distance between tunnel A and B, we first have to select the two nodes which are closest to each other, in our case

A_2 and B_2 . Then we calculate the distance between the other two endpoints (A_1 and A_2). The distance is then defined to be:

$$\text{distance}(A, B) = \max(\overline{A_1B_1}, \overline{A_2B_2})$$

Figure 4.10 shows the difference between the tunnel's RTT versus the distance relative to the length of the rated tunnel. Again we see that with increasing distance the similarity is decreasing. Here we are even able to spot a limit where tunnels seem to lose their similarity. It is around a third of the length of the tunnel that is compared.

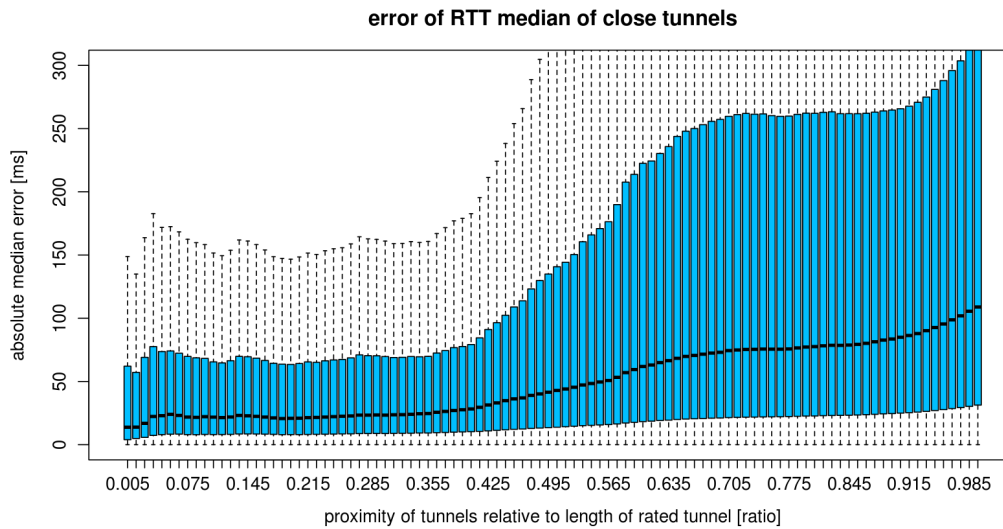


Figure 4.10.: RTT difference of tunnels depending on their distance to each other

We decide to use the maximum distance approach as our distance criteria because it is more intuitive in telling in which area a similar tunnel is.

4.3.2. Reference Selection

So far we only defined how to measure the distance between two tunnels. In this section we will explain how to look for similar tunnels in an efficient way. A first approach was to use the clustering for the search of the closest tunnels to lower the complexity of finding close tunnels. But since the endpoints' positions are static in general, we can pre-compute the distance between all tunnels. Of course it makes no sense to store the distance for all tunnel combinations. A certain amount of the closest tunnels for every tunnel is sufficient.

We need to find the trade-off between many tunnels as reference and a small distance to the tunnel that should be rated. To get an impression how the amount of tunnels that are taken as reference influences the similarity, we plot the difference between the reference value for the median RTT and the tunnel itself. Figure 4.11 shows how the similarity changes for 500

random tunnels if we use a different number of tunnels to calculate the reference median RTT. The reference value is defined to be the median of all the *median RTT* values of the reference tunnels.

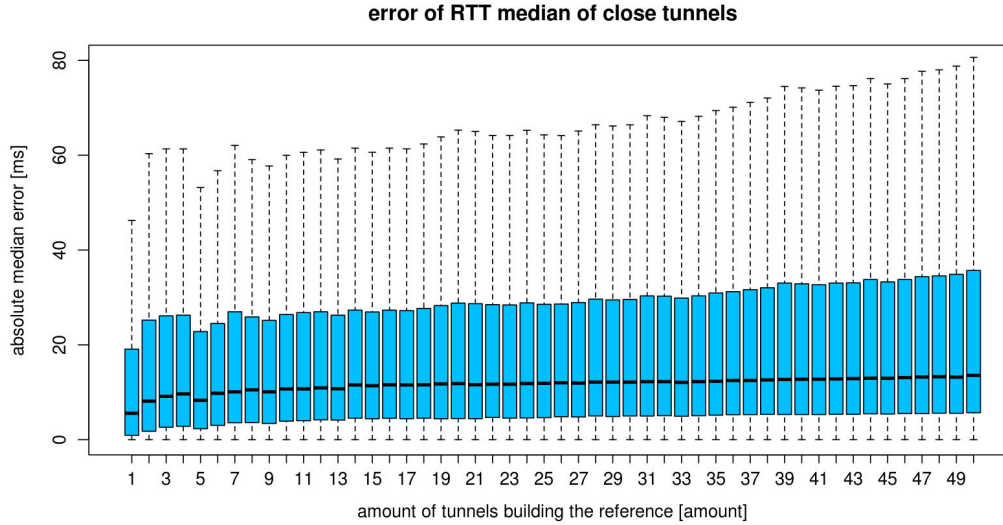


Figure 4.11.: Difference between median RTT of the tunnel and its reference depending on the number of reference tunnels. The maximum search radius is limited to 0.3 of the tunnels length.

We see that the amount of tunnels that we take to build the reference does have a limited influence in general. This leads to the policy that we pay more attention to the distance and do not ask for a high number of similar tunnels. The trade-off that we use is to request a minimum amount of 10 tunnels to be able to generate a reference that is based on a few values. In addition we look for similar tunnels with a step-wise increasing distance. We increase the search radius until the amount of tunnels is above 10. The *step* is defined to be 5km to increase the search radius slowly. If we did not find at least 10 similar tunnels within a search radius of 20km we lower the requirement to 5 tunnels. If the search radius exceeds 20km with a number of reference tunnels below 10, the tunnel density between the regions is low and asking for too many tunnels will result in a bad reference.

Algorithm 4 Reference Tunnel Selection

- 1: Choose an initial tunnel t_j to be rated from the set of all tunnels T .
 - 2: Let $R = \{\}$ be the set of reference tunnels
 - 3: Let $d = 0$ be the search radius
 - 4: **while** ($|R| < 10$ and $d \leq 20$) or ($|R| < 5$ and $d > 20$) **do**
 - 5: R is the set of all tunnels with $d(t_i, t_j) < d$ for $i \in [1, \dots, |T|]$ and $j \neq i$
 - 6: $d = d + \text{step}$
 - 7: **end while**
-

4.3.3. Short Tunnels

Links that have to be evaluated differently are links that connect two endpoints in the same building or in a radius smaller than one minute (coordinate resolution). In Switzerland the minimum measurable distance is about 1.85km for the latitude and 1.32km for the longitude. Tunnels with a smaller distance are mapped on the same coordinates and therefore have a length of 0 meters when calculating the distance. Those links are simply compared to all other links that have the distance of 0 and are around this place. This affects approximately 200 tunnels in all the VPNs¹.

4.3.4. Coverage

The described method allows to rate every tunnel. It is possible to rate a tunnel independent of the tunnel itself or the location of its endpoints. The density of endpoints is increasing with every newly-added tunnel. Hence the distances between tunnels should decrease with a growing network.

The only case we are not able to rate a tunnel is if there is no measurement data about the tunnel itself.

4.4. Accuracy

In general, we try to get as many similar tunnels as possible to be sure to have a rating based on a good statistical reference. On the other hand, we are only able to know how the connection between two sites should perform if the reference links are representing this region too. We can not guarantee that the tunnels that we take as reference are always of sufficient relevance to get a good rating, especially if we have to increase the distance by a lot to find enough similar tunnels. Hence the search radius should be taken into consideration when using the rating.

An additional factor is the different performance of the reference tunnels themselves. If they are all very similar in their RTT and loss rate behaviour, we can assume that the characteristics are very typical for the inter-connection of the two regions. If they differ a lot, this indicates that there is no clear reference value. We compare, e.g., a tunnel's RTT to the median of the references because we do not want to compare it to a reference value which is influenced by a tunnel that is performing far below standard. Still the similarity of the reference tunnels is important to be taken into account.

We would like to point out that the rating is always based on statistical values. It is a good indication but a rating can not be taken as a fact. It still needs human interpretation but it supports us to figure out the links that perform differently than others in the same area. It is up to us to analyze why a link is performing different than the reference links. There are reasons like different technology (MPLS, DSL, Satellite) or different link qualities (cost) that can not be taken into account for this thesis. Unless we differentiate between different technologies and lines offered by the ISP, we have to be aware that a bad rating can also mean that all reference tunnels just

¹Status: January 2012

use a different technology or a more expensive Internet connection than the tunnel which is being rated.

4.5. ISP Selection

A system that is able to rate an ISP according to its general performance would be of extreme value in selecting good Internet service providers which would provide the best performing links.

ISP Rating

We analyze if the average loss rate is related to the ISP by using the Kruskal-Wallis rank sum test [16]. We have a vector x containing all the different loss measurements and a vector y containing all the ISPs. We want to test if the loss distribution mean is similar for the different providers. The H_0 Hypothesis states that for different providers we have the same loss rate mean. The analysis is done by using the statistic software R^2 .

The 10 most used ISPs are analyzed:

- *ISP - Average Loss Rate* The Kruskal-Wallis test results in a p-value of 0 which means that we have to reject the hypothesis H_0 . This means that we can not assume that different ISPs have similar loss behaviour.
- *ISP - Loss Periods* The Kruskal-Wallis test results in a p-value of 0 which means as well for the periods of loss that there are different loss period characteristics for different ISPs.

In general this is an indication that distinct providers perform differently. If we plot for different providers their average loss rate distribution of all tunnels they are part of, then we are able to see different patterns for different providers. See Figure 4.12 to see the different distributions for Swiss providers. Figure 4.13 shows the same distribution for China. In China we actually know that 'China Unicom' is often the provider of Internet lines which are known to be related to badly performing tunnels. In some cases a change to a different provider indeed solved the problem³.

²Kruskal-Wallis rank sum test: <http://stat.ethz.ch/R-manual/R-patched/library/stats/html/kruskal.test.html> (last visited: March 2012)

³Information from Dominique Chappuis: Head of WAN Management at Open Systems

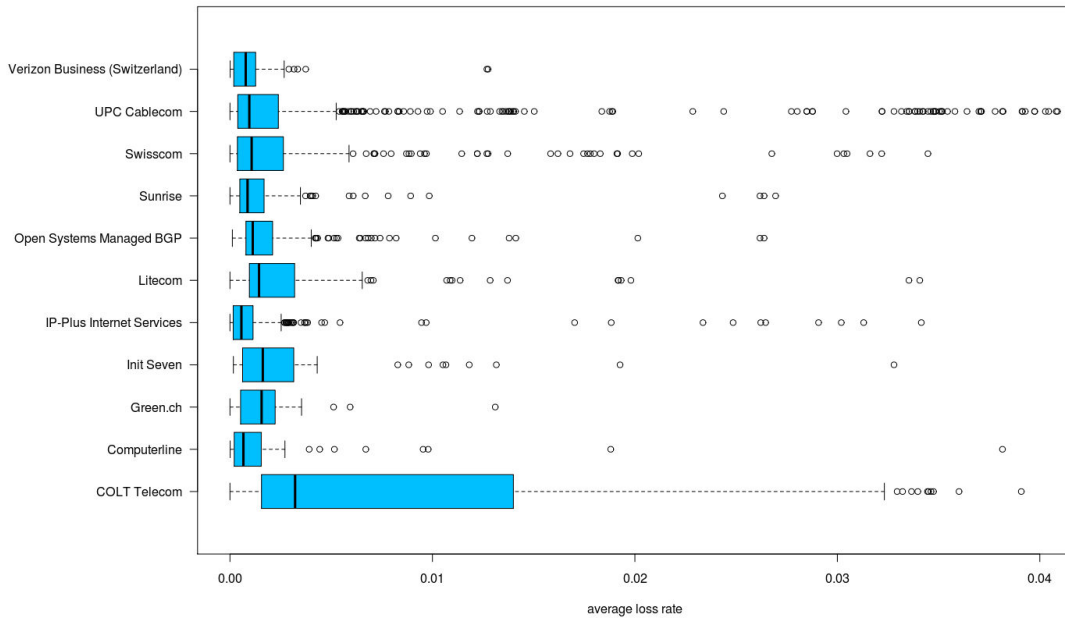


Figure 4.12.: Average loss rate of tunnels using Swiss providers at one endpoint

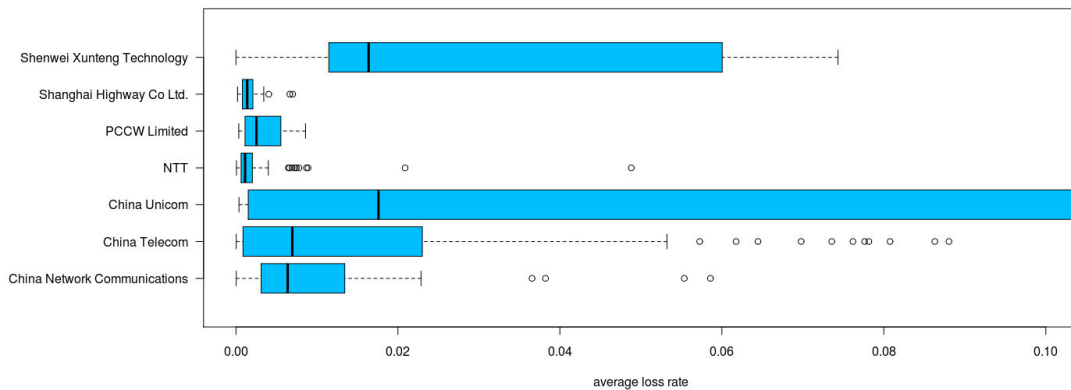


Figure 4.13.: Average loss rate of tunnels using Chinese providers at one endpoint

Although some general provider-related problems could be visible in these plots it leads to the wrong assumption that a provider can be seen as a homogeneous object that can be rated. In the example of the Swiss providers we would assume that 'COLT Telecom' performs badly and that they suffer from packet loss. A detailed analysis shows that 'COLT Telecom' is the main ISP of a Non-Governmental Organization (NGO). The VPN endpoints of this NGO are distributed all over the world and very often established using satellite links. It will not be possible to rate an

ISP in general without respect to the type of connections representing an ISP in the setup we use. It even happens that the link with the smallest and the one with the highest packet loss rate in a region have the same ISP. Also Zimmerli [30] came to the conclusion that it is not possible to rate the ISPs in a reliable way.

Internet Line Rating

We will not be able to suggest a specific ISP for a certain endpoint or region. But in specific cases we should be able to differentiate between general bad tunnel performance and bad tunnel performance caused by one of the two endpoints.

Let us assume we have two endpoints A and B connected with a tunnel and the tunnel suffers from a high loss rate (see Figure 4.14). We analyze all connections outgoing from A and the ones outgoing from B. If now all connections from A suffer from loss and for B only the connection to A suffers from loss this would be a strong indication that the problem is related to the Internet connection of A. According to SLAC [4], packet loss often happens at the last mile. The back-bone network is usually over-provisioned and therefore is not the bottleneck in a communication channel [17].

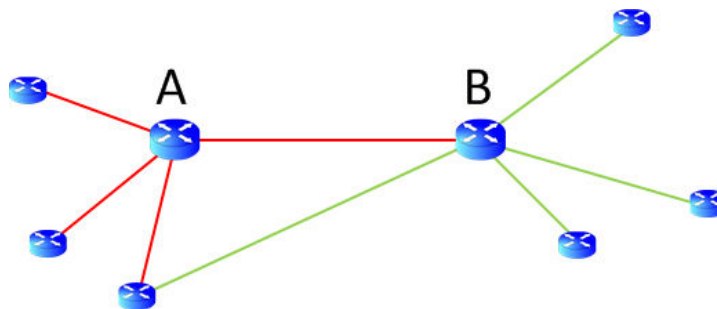


Figure 4.14.: Situation where we would be able to detect the endpoint which is the reason for the bad performance between A and B. Red indicates high and green low packet loss.

A case like above might be solved by informing or changing the local ISP. We could look for typical cases where packet loss merely happens at one specific endpoint. With a more complex analysis it should be possible to evaluate cases where both endpoints suffer from bad performance by evaluating their neighbours' connections.

4.6. Performance Estimation

Besides the rating of VPN tunnels, reference tunnels can be used to estimate the RTT and loss rate for a newly-planned connection. For highly connected regions there could even be made a suggestion of the providers and technology to be used. For regions where no links exist so far at least an estimate could be made by taking links which are close to the planned one into account. Thanks to the global coverage of the Open Systems VPNs, there is almost everywhere in the world a reference point which allows the estimation of a new tunnel's performance.

CHAPTER 5

Implementation

This chapter gives an insight into the implementation of the prototype which automatically generates a VPN performance report.

5.1. Setup

Figure 5.1 gives an overview of the process on creating the performance report. We differentiate between a distributed part, where the necessary performance information is extracted, and a centralized part, where the rating for each individual VPN tunnel is calculated. We also have two data sources:

- information about the network's topology
- performance measurements

The network topology is stored in a database and the performance measurements are originally stored on the VPN endpoints. The rating is calculated by combining this information. In the end a report is generated which shows the performance rating of the individual VPN tunnels.

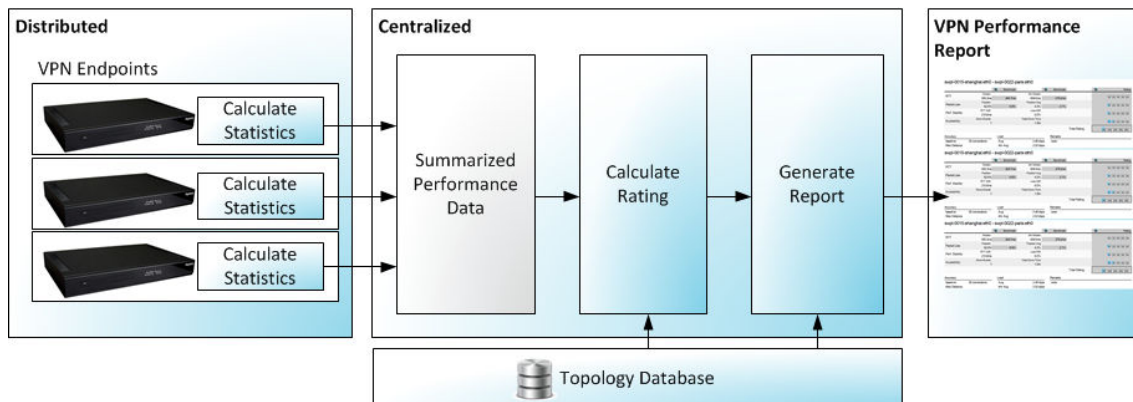


Figure 5.1.: The process to create a performance report

5.2. Preparation

We need to know the topology of a network to calculate a tunnel rating. This information is stored in a database. In this section we explain how the structure of the database looks like and the data is imported.

5.2.1. Topology Database

We use *SQLite3*¹ as our database. The advantage in comparison to a MySQL database is that no server is needed. The data is stored in a single file, resulting in good portability.

5.2.2. Database Structure

Figure 5.2 shows the database structure and the relations between the different tables.

¹SQLite: <http://www.sqlite.org> (last visited: March 2012)

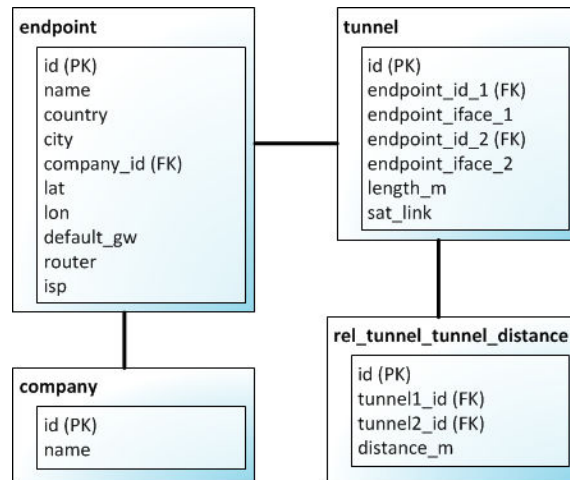


Figure 5.2.: The database scheme for the tunnel information. (PK) and (FK) indicate private and foreign keys.

- **endpoint** It contains information about the endpoints such as their location, the company they belong to and identifiers. Additional information like the default gateway is also stored but not needed for the rating.
- **company** Contains all the companies which have VPN endpoints in their network.
- **tunnel** This table contains the information about all the VPN tunnels and therefore the network topology. It shows the two endpoints and their interfaces of every tunnel. The length of the tunnel is stored. The information if a tunnel is a satellite link or not could be stored in the field *sat.link* but since a general classification is not always possible, we do not use this field so far.
- **rel_tunnel_tunnel_distance** It contains the information about the distance between two tunnels. The distance information is pre-calculated when importing the network topology. This allows a fast and easy search for similar tunnels.

5.2.3. Precalculation

Since the distance information between tunnels is merely static we already pre-calculate the distances between a tunnel and its closest tunnels using the distance metric presented in 4.3.1. Of course we do not store all tunnel combinations. Only the 50 closest tunnels are stored with the actual implementation.

5.2.4. Import

A single script can be used to import all the information to the database.

```
./import.pl <t=t_path> <p=p_path> <s=s_path> <database.db>
```

Use the path to the csv-file which contains the tunnel information as *t.path*. The tunnel information is the only information that is mandatory. It contains the information about the network topology. The information about satellite links can be given by using *s.path*. It allows us to indicate in the report if a satellite connection is used to establish a certain VPN tunnel. Since we are not able to determine for every tunnel whether there is a satellite link involved, this information is optional.

For the data analysis in Chapter 3 we also needed access to some simple performance information. We can directly import them using the *-t* option.

With *database.db* we indicate where to store the information. The database can already be existing or will be created if it is not.

5.3. VPN Performance Rating

5.3.1. Data Retrieval

Every host continuously sends probes to measure the RTT and packet loss rate. The endpoints keep track of their own statistics and store the probe measurements by using RRDtool². Every time we want to rate a tunnel the calculation of a performance reference is necessary. Hence we need to access the information of various similar tunnels.

It is not an option to collect the files which contain the measurement data and process them on a central entity. It would use too much bandwidth. Especially if we want to make the rating efficient, we have to come up with a scalable solution.

To characterize long-term tunnel performance and to rate a tunnel, only a few key figures are needed. The values which are needed for the report are:

- median RTT
- 4hr median RTT
- inter-quantile range of RTT
- how often a tunnel was suffering from packet loss
- how severe the packet loss was in these periods (average)
- average packet loss
- inter-decile range of the average loss rate

²RRDtool: <http://oss.oetiker.ch/rrdtool> (last visited: March 2012)

- usage information of the tunnel
- availability information (down events / down time)

These values can be computed locally on the endpoint itself. This distributed approach allows that only the necessary information is transferred to the entity which calculates the rating.

We implemented a script which can be run on every single endpoint to extract the necessary key figures. The script can be called using

```
./t_rating_keys_extract.pl <host_id> <host_name> <start> <period>
```

The *host.id* is the unique identifier of the endpoint on which we run the script. The *host.name* is the name of the endpoint. With *start* we define at which time in the past we want to start the analysis³. The *period* is the amount of seconds we like to analyze. The longer ago the specified start time, the less resolution for the measurements we get. We use 2 weeks for the *period*.

The script returns the statistical key figures in an XML format to the console. Appendix A.3 shows a typical output. It contains all performance information needed to calculate the rating for the VPN tunnels.

Data Resolution

As mentioned in section 3.1.3 the resolution of the data depends on the time span that we want to analyze. For the past 2 days we have the maximum available resolution of 5 min, whereas for the past month we only have 2-hour averages. We aim for the highest possible resolution for the measurement data but also for a time span that allows long-term analysis of a link's performance. With every aggregation we lose the granularity of the oscillations in the measurement data. For this thesis we decided to use the last two weeks of monitoring which leads to a resolution of 30-minute averages for the measurement data (see section 3.1.3).

5.3.2. Collect Data

So far the calculation of the statistical information is separated from the generation of the report itself. The information is exported to files which contain the performance information in an XML format. These files are imported when a report needs to be generated. If the rating should be calculated in a productive way, this part could be replaced by using a database. The database would only have to store a few values per tunnel and therefore an efficient data-access would be possible.

³times are indicated in seconds since January 1st, 1970 (UTC)

5.3.3. Report Generation

The report generation can be initialized after we have collected the performance information of all endpoints and we have built the database. Run:

```
./performance_report.pl -c=<comp1>,<comp2>,... -p=<p_folder> <database>
```

The script allows generating a report for multiple companies at once. Just use a ';'-separated list to specify for which companies a report should be generated. If no company is specified, a rating for all companies will be generated. The *p_folder* defines where the collected performance measurement files are stored. The database can be defined by using *<database>*.

The report is generated in two steps. First the most similar tunnels for every tunnel are used to calculate a performance reference. Every tunnel is then rated by using this reference. In the second part the ratings are written to a PDF-File. To generate the PDF we first write a performance report in HTML. The Template Toolkit⁴ is used to generate the HTML-Report. For the conversion of the HTML-Report to a PDF we use PDFReactor⁵.

5.4. Performance Report

There are different ways to present the rating of the tunnels to a network administrator. One would be a tool for a website showing the tunnel's performance in an interactive way, which would allow to get a quick overview and also some detailed information on request. Although the described solution would be preferable for somebody who would like to analyze the tunnel's performance in detail, we decide to make a monthly report showing the long-term performance of the VPN tunnels of a specific company. In this thesis we want to focus on the quality of the performance rating and not on the usability of a website. A monthly report is already standard for different services at Open Systems and could be a complement to an interactive tool.

5.4.1. Information

The report should give some general information about the network, like how many tunnels it contains and how many endpoints there exist. The rating of the tunnel should allow an unexperienced user to see immediately if a tunnel is performing good or bad. We decide to map the calculated rating to a star rating. One star indicates poor performance and five stars indicate an excellent link. For a deeper analysis it also contains selected figures which already allow a quick analysis.

We give a star rating for every individual metric:

- Round trip time
- Packet Loss

⁴Template Toolkit: <http://template-toolkit.org> (last visit: March 2012)

⁵PDFReactor: <http://www.realobjects.com/products/pdfreactor/overview> (last visited: March 2012)

- Performance Stability
- Availability

In addition we calculate an overall performance rating for the tunnel which contains the metrics from above and indicates how a tunnel performs with a single figure.

5.4.2. Rating Algorithms

We use the metrics defined in section 4.2.1 and make a rating of the links based on the values that we have for a specific link and its reference links. The goal is to transform several different metrics to a single rating such that we immediately see how a tunnel is performing. In addition we will show a summary rating of every category (RTT, loss rate, ...). The following sections will explain how the different ratings are being calculated.

RTT

The metrics that we will use for the rating are:

- median RTT
- median RTT of the worst 4hrs of the day

We will rate a tunnel's RTT in comparison to other tunnels which allows us to rate whether a tunnel could be better or if there not a possibility for a better connection. Hence the *median RTT* value is compared to the *median RTT* of the reference tunnels and rated accordingly with 1 to 5 points.

Often the RTT differs with a daily pattern from the general median, e.g., caused by higher load in the network during business hours. Hence we calculate the worst 4 hours of the day and show the median RTT of these 4 hours on the report. This is again compared to the reference values and rated with 1 to 5 points.

The thresholds are based on the distribution of the differences between tunnel and reference. Since also the RTT difference among the reference tunnels is a heavy-tailed distribution we define the *median RTT* rating for a tunnel to be based on:

$$\Delta_{median(RTT)} = RTT_{median} - median(median(RTT_{reference}))$$

The thresholds are based on the percentiles of the $\Delta_{median(RTT)}$ distribution which can be seen in Figure 5.3. The goal is not to have an equally distributed rating which means that we do not aim for a rating where the same amount of tunnels get a 1 star-rating as the ones that get a 5 star-rating. We still assume that the majority of the tunnels to a region build the baseline and therefore should be rated to perform good. Hence we use the range between the 50th and 100th percentile to define the thresholds for the rating. The different thresholds approximately represent the 50th, 70th, 80th and 90th percentile. This rule is also used for the loss rate, stability and availability.

Rating	Thresholds	
	$\Delta_{median(RTT)}$ [ms]	$\Delta_{median(4hrRTT)}$ [ms]
5	≤ 0	≤ 0
4	≤ 8	≤ 8
3	≤ 15	≤ 15
2	≤ 25	≤ 30
1	> 25	> 30

Table 5.1.: Rating thresholds for the RTT

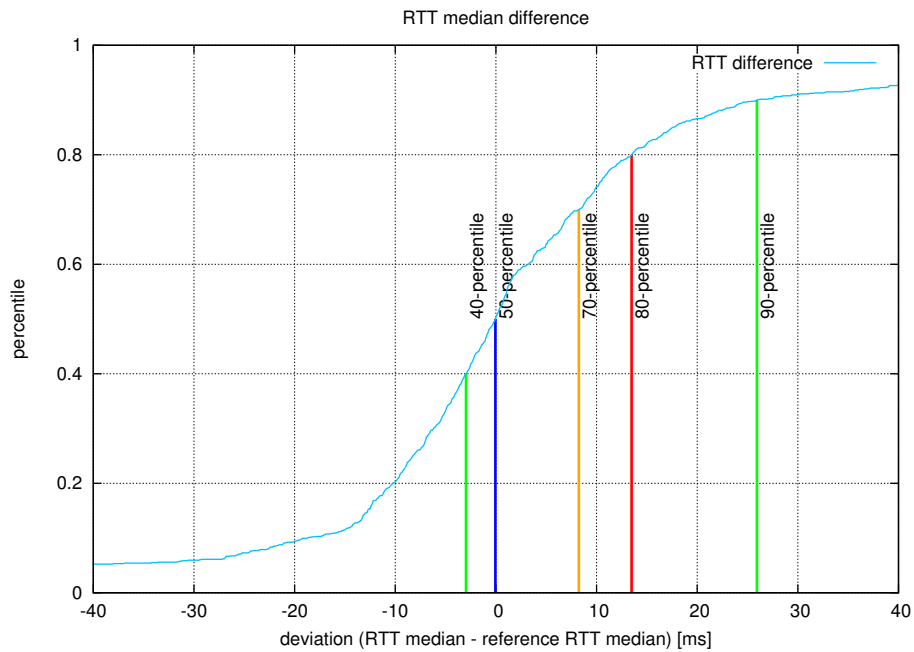


Figure 5.3.: Distribution of the differences between a tunnel's median RTT and its reference median RTT

For the rating of the *4hr median* we proceed identically. The thresholds are almost the same because the distribution is very similar. The final RTT rating is the average of the rating R_{median} for the *median RTT* and the rating R_{4hr} for the *4hr median RTT*.

$$R_{RTT} = \frac{1}{2} \cdot (R_{median} + R_{4hr})$$

Loss Rate

We decided to take the ratio of *time during which loss appeared* and the *average loss during this time*. Good links have most of the time no loss and if they do, it is only a small rate.

Figure 5.4 shows the distribution of the difference between the ratio with loss of the tunnel and the one of the reference value. The rating R_{ratio} is depending on the difference Δ_{ratio} between the amount of *time with loss* and the median amount of *time with loss* of all the reference tunnels.

$$\Delta_{ratio} = (\text{time with loss}) - (\text{reference time with loss})$$

The difference between the average loss rate in times of loss and the one of the reference values is used for the rating $R_{avgRatio}$. Figure 5.5 shows the distribution for $\Delta_{avgRatio}$.

$$\Delta_{avgRatio} = (\text{loss}_{avg} \text{ if loss}) - (\text{reference loss}_{avg} \text{ if loss})$$

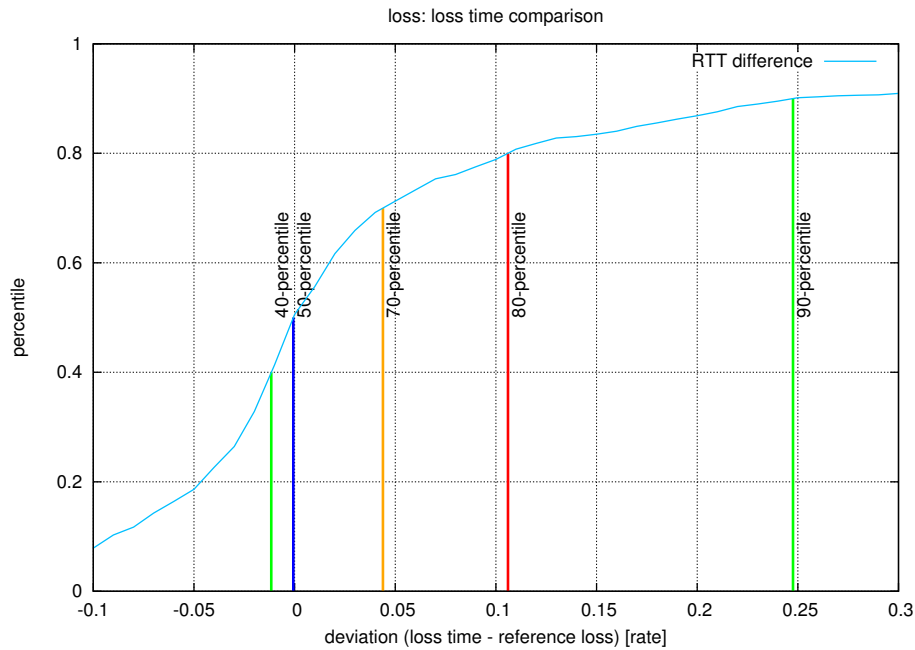


Figure 5.4.: The distribution of the difference for the *loss time* of a rated tunnel and its reference

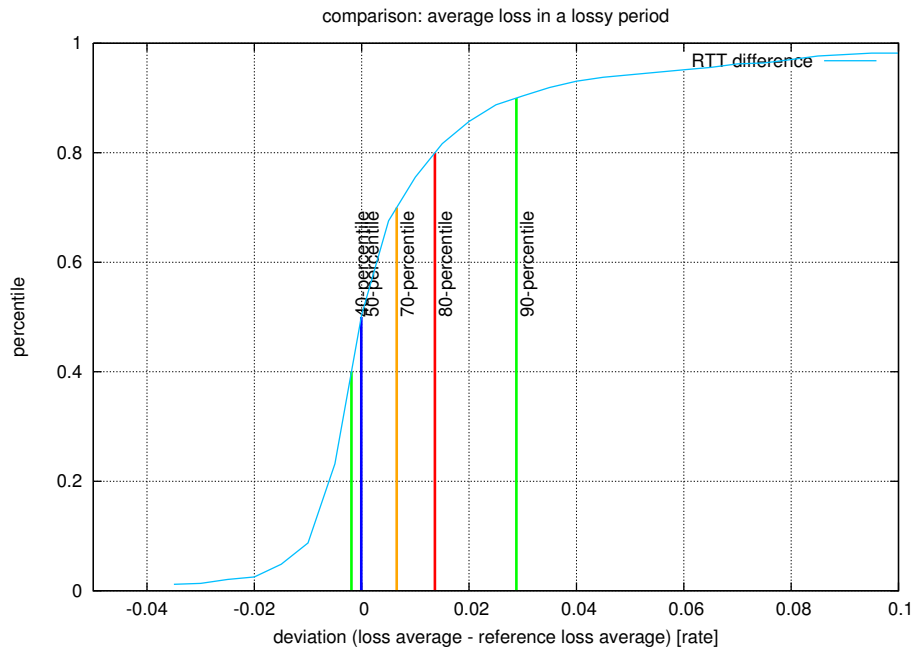


Figure 5.5.: The distribution of the difference for the *loss rate in times of loss* of a rated tunnel and its reference

Since loss is not depending merely on the location of the two endpoints we compare the ratio of *time with loss* also to the global distribution. In addition we use the quality levels defined by SLAC⁶ for the time where loss appears. All thresholds are summarized in Table 5.2.

Rating	Thresholds			
	Δ_{ratio} [%]	$\Delta_{avgRatio}$ [%]	time with loss [%]	$Loss_{average}$ in times of loss [%]
5	≤ 0	≤ 0	≤ 10	≤ 0.5
4	≤ 4	≤ 0.5	≤ 15	≤ 1
3	≤ 10	≤ 1	≤ 20	≤ 5.5
2	≤ 25	≤ 3	≤ 30	≤ 12
1	> 25	> 3	> 30	> 12

Table 5.2.: Rating thresholds for the packet loss

The overall packet loss rating can be calculated with

$$R_{loss} = \frac{1}{4} \cdot (R_{ratio} + R_{avgRatio} + R_{ratioGlobal} + R_{avgRatioGlobal})$$

⁶Quality Loss: <http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html#loss> (last visited: March 2012)

Stability

With the ratings for the RTT and the loss rate we only benchmark the median or average performance. It does not show if a tunnel performs constantly or if there are big variations over time. To measure the stability of a tunnel's performance we decided to use the IQR for the RTT and the IDR for the loss rate. In addition we use the difference between the median RTT and the *median 4hr RTT*.

The thresholds are defined to be

Rating	Thresholds		
	$IQR(RTT_{median})[ms]$	$\Delta_{RTT}[ms]$	$IDR(Loss_Rate)\%$
5	≤ 5	≤ 3	≤ 0
4	≤ 10	≤ 5	≤ 1
3	≤ 15	≤ 10	≤ 2
2	≤ 20	≤ 15	≤ 4
1	> 20	> 15	> 4

Table 5.3.: Rating thresholds for the stability

The total stability rating is calculated to be

$$R_{Stability} = \min \left(\frac{1}{2} (Rating(IQR(RTT_{median})) + Rating(\Delta_{RTT})), Rating(IDR(Loss_Rate)) \right)$$

Availability

The availability of a tunnel is also a very important criteria. Here we rate the number of down events and the total time the tunnel was down during these events.

The thresholds are defined to be

Rating	Thresholds	
	Down Events [#]	Total Downtime [hrs]
5	0	0
4		≤ 0.5
3	≤ 1	≤ 1
2	≤ 2	≤ 2
1	> 2	> 2

Table 5.4.: Rating thresholds for the stability

The rating is

$$R_{Availability} = \frac{1}{2} (Rating(Down\ Events) + Rating(Average\ Downtime))$$

Accuracy

To indicate the accuracy or how representative the reference values are, we inform about the number of tunnels taken as reference and the needed search radius. We also show how similar the reference tunnels are among each other. We calculate the IQR of the *median RTT* and the IQR for the *average loss rate* among the reference tunnels.

	Thresholds	
Rating	IQR(RTT_{median}) [ms]	IQR($Loss_{avg}$) [%]
5	≤ 15	≤ 0.25
4	≤ 25	≤ 0.5
3	≤ 40	≤ 0.75
2	≤ 60	≤ 1
1	> 60	> 1

Table 5.5.: Rating thresholds for the accuracy

The average rating indicates how similar the reference tunnels are among each other. The thresholds are given in Table 5.6.

$$R_{Similarity} = \frac{1}{2} (Rating(IQR(RTT_{median})) + Rating(IQR(Loss_{avg})))$$

	Thresholds
Rating	$R_{Similarity}$
very good	≥ 4.5
good	≥ 3
poor	≥ 2
very poor	≥ 1

Table 5.6.: Rating thresholds for the accuracy rating

Additional Information

In addition to the rating itself we indicate how high the load on a specific tunnel is. This is valuable information on estimating whether bad performance is affecting an important link or if it is not critical.

Overall Rating

In the end every tunnel will be rated with one single overall rating. It is defined to be

$$R = \min(R_{RTT}, R_{Stability}, R_{Loss}, R_{Availability})$$

We take the minimum out of all the ratings because every criteria affects the quality of the link.

5.4.3. Result

The implemented prototype generates a monthly performance report per customer. It is divided into different sections. To see a sample report please refer to appendix B.

- An overview gives general information about the network. A map shows the customer's network where the tunnels are colored according to their rating. The overview also informs about the amount of tunnels belonging to the network and which of the connections are most used.
- The next sections inform about the performance of the most used tunnels. Also the worst and the best performing links are listed with their rating (idle tunnels excluded).

Figure 5.6 shows a sample rating of a tunnel. The total rating allows a good overview of the tunnel's overall performance whereas each subsection gives a more detailed view on the measurements. The accuracy information allows deliberating whether the reference tunnels are a typical representation for a connection between the two locations of the tunnel that should be rated. It shows how many tunnels are used to calculate the reference values, how far apart they are from the rated tunnel and how similar the reference tunnels are among each other. Information about the usage shows how important the tunnel is. The rating of idle tunnels is not listed in the report.

	Benchmark		Benchmark		Rating
RTT	Median		4hr Median		★ ★ ★ ★ ★
	103.9ms	33.2ms	694.6ms	34.2ms	
Packet Loss	Fraction		Fraction Avg		★ ★ ★ ★ ★
	44.0%	11.9%	2.7%	1.3%	
Perf. Stability	RTT IQR		Loss IDR		★ ★ ★ ★ ★
	625.0ms		0%		
Availability	Down-Events		Total Down-Time		★ ★ ★ ★ ★
	0		0hr		
Total Rating					★ ★ ★ ★ ★

Figure 5.6.: A sample performance rating of a tunnel

CHAPTER 6

Evaluation

In this chapter we will describe how we evaluated our rating method.

6.1. Rating Evaluation

In order to get a good rating, we could make a survey where we would ask people whether the performance rating reflects their own opinion about a certain tunnel's performance. A good possibility would be to survey a mean opinion score (MOS) for tunnel performance and compare it to the rating. The MOS was originally developed to evaluate speech quality with respect to the quality of experience (QoE). Such survey would have to be done in a long-term evaluation by involving customers. Especially the interaction with the customers would represent the main difficulty.

Since a survey is not possible, the evaluation of our performance rating is not straightforward. The rating is based on statistical information and it will show how good a tunnel is performing in comparison to similar tunnels. If the report tells us that a tunnel is performing badly and we analyze the tunnels it took as reference, we will retrieve the same rating by definition. To evaluate the quality of the rating, we have to evaluate whether known problems are visible in the report or whether the reported problems can be verified manually.

In the following sections we will present a few cases where the performance of a link was known to be unsatisfying.

6.2. Known Provider Issues

We have two cases where we know that the loss rate was unsatisfying. Open Systems tried to improve the performance of the links by either changing or intervening with the ISP. Engineers noticed an improvement and we evaluated whether this was visible in our report.

6.2.1. Case 1 - Zurich to Shanghai

We have a tunnel connecting a site in Zurich with one in Shanghai. The tunnel is suffering from intensive packet loss. In mid February the provider in Shanghai was changed to overcome this problem. Figure 6.1 shows the loss statistic for the time in January and February. A pillar in the plot represents the loss average of one day.

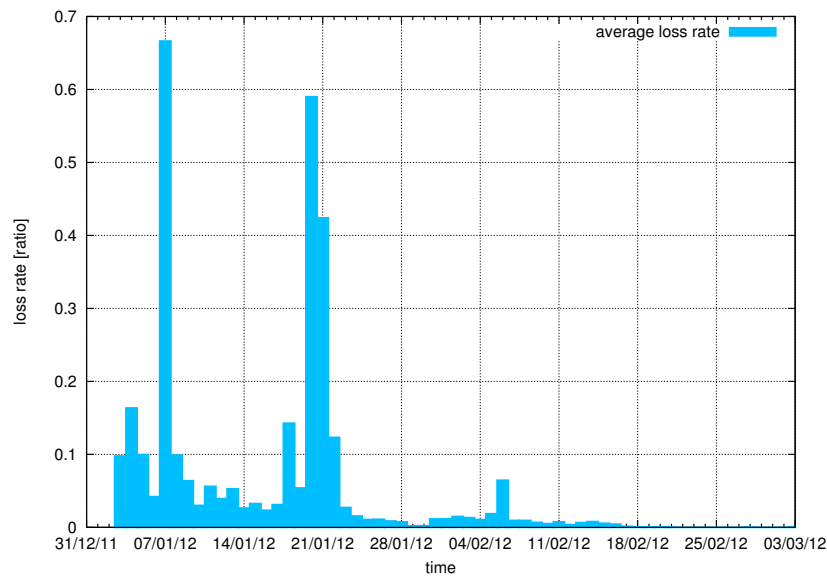


Figure 6.1.: Link suffering from packet loss till mid of February

Figure 6.2 shows the rating in January and Figure 6.3 the one after the ISP-change in February. First of all, we see that the ISP-change goes along with a better rating of the tunnel. If we look at the report in a more detailed way, we see that the packet loss rating and availability indeed get a better rating. We can call the tunnel to perform normal again. Together with a more stable RTT this leads to a higher overall tunnel rating.

	Benchmark		Benchmark		Rating
RTT	Median	371.8ms	4hr Median	410.9ms	★★★★★
		373.9ms		411.8ms	
Packet Loss	Fraction	40.5%	Fraction Avg	3.6%	★★★★☆
		26.6%		3.2%	
Perf. Stability	RTT IQR	99.9ms	Loss IDR	0%	★★★★☆
Availability	Down-Events	2	Total Down-Time	25.6hr	★★★★☆
Total Rating					★★★★☆

Figure 6.2.: Tunnel rating before the ISP-change (Zurich - Shanghai)

	Benchmark		Benchmark		Rating
RTT	Median	308.0ms	4hr Median	330.7ms	★★★★★
		373.8ms		374.6ms	
Packet Loss	Fraction	5.2%	Fraction Avg	2.2%	★★★★★
		33.5%		2.7%	
Perf. Stability	RTT IQR	9.5ms	Loss IDR	0%	★★★★★
Availability	Down-Events	0	Total Down-Time	0hr	★★★★★
Total Rating					★★★★★

Figure 6.3.: Tunnel rating after ISP-change (Zurich - Shanghai)

6.2.2. Case 2 - Zurich to Guernsey

A tunnel between Zurich and Guernsey is suffering from packet loss and outages. Figure 6.4 shows the situation. After intervening with the local ISP in Guernsey the situation changed drastically and the tunnel does not suffer from packet loss and outages any longer.

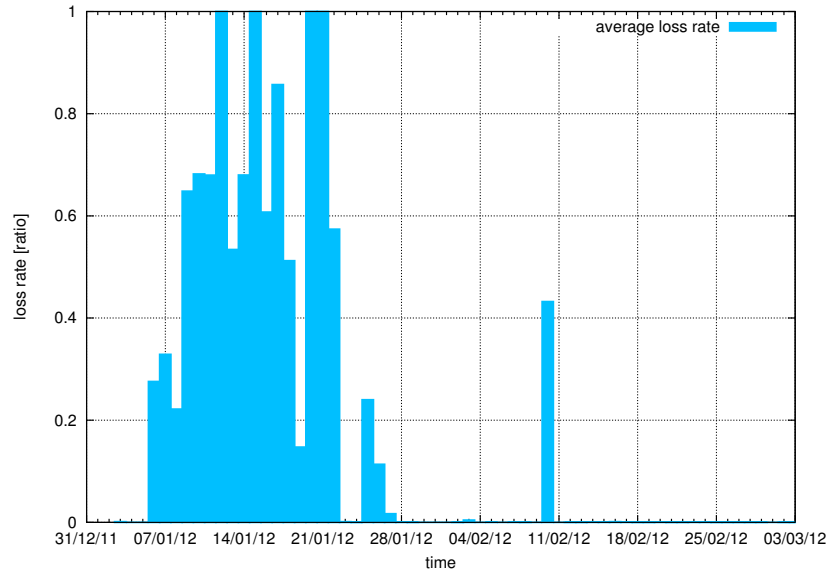


Figure 6.4.: Link suffering from packet loss and outages till mid of February

Again, the increase in performance can be seen in Figure 6.5 and Figure 6.6. In January the tunnel was suffering from 3 outages and loss periods which had an average packet loss of more than 19%. In February the tunnel did not suffer from outages or intensive packet loss anymore.

	Benchmark		Benchmark		Rating
RTT	Median	47.9ms	4hr Median	48.0ms	★★★★☆
		36.7ms		36.8ms	
Packet Loss	Fraction	14.7%	Fraction Avg	19.9%	★★★★☆
		15.0%		3.9%	
Perf. Stability	RTT IQR	3.4ms	Loss IDR	20%	★★★★☆
Availability	Down-Events	3	Total Down-Time	66.6hr	★★★★☆
Total Rating					★★★★☆

Figure 6.5.: Tunnel rating before the ISP was informed (Zurich - Guernsey)

	Benchmark		Benchmark		Rating
RTT	Median	47.3ms	4hr Median	47.4ms	★★★★☆
		35.9ms		36.0ms	
Packet Loss	Fraction	4.8%	Fraction Avg	0.8%	★★★★☆
		4.6%		0.9%	
Perf. Stability	RTT IQR	0.3ms	Loss IDR	0%	★★★★☆
Availability	Down-Events	0	Total Down-Time	0hr	★★★★☆
Total Rating					★★★★☆

Figure 6.6.: Tunnel rating after the ISP was informed (Zurich - Guernsey)

In this example we see an additional characteristic of our rating method. Since almost all tunnels which have been taken as reference are from the same company and they all use the same ISP, the improvement in the packet loss behaviour also directly affects the benchmark.

6.3. Known RTT Issue

We know of a tunnel which is suffering from a very high and instable RTT. The endpoints of the tunnel are in London and Zurich. The reason for the bad RTT performance of this tunnel is not yet known. The ISP claims that it is caused by an overloaded MPLS line but the analysis at Open Systems did not confirm this. We will analyze all tunnels which connect Zurich with London in the section below.

6.3.1. Zurich to London

The data to rate the tunnels are from February 2012. We analyze all tunnels which show a RTT rating below 3 and which are not idle. A RTT rating below 3 means that their RTT is significantly higher than the typical one.

Known Performance Issue

First we analyze the tunnels that are known to be performing below standard. In a second step we will look at the other tunnels. Figure 6.7 shows the RTT measurements of the known tunnel.

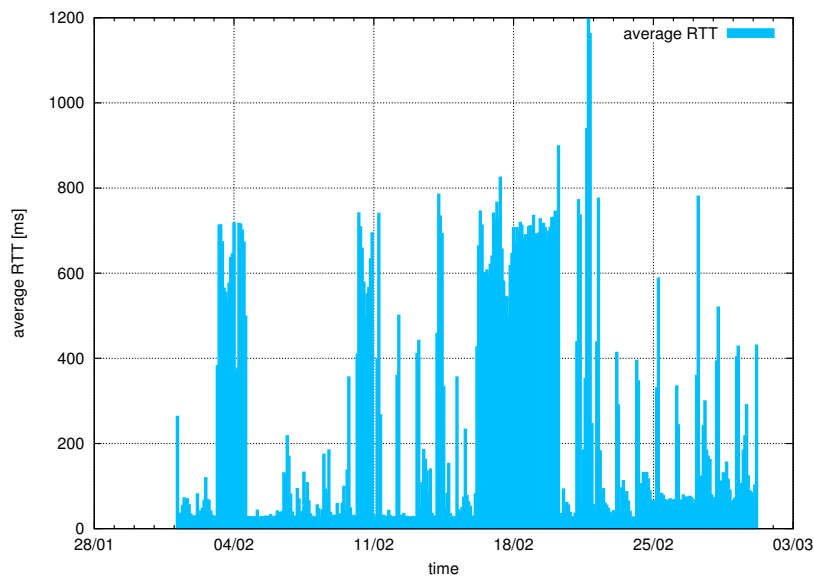


Figure 6.7.: Link suffering from bad RTT performance (Zurich - London)

We did a ranking of the RTT rating of every tunnel between Zurich and London and the mentioned tunnel was ranked to perform worst. Figure 6.8 shows the February rating of the tunnel.

6.3. Known RTT Issue

It shows that the RTT is generally much too high. The typical RTT for tunnels between Zurich and London is in the range of 30ms. A tunnel with a RTT above 103ms is definitely performing badly. The rating also shows that there quite often appears packet loss in the tunnel and that the performance is not stable at all.

	Benchmark		Benchmark		Rating
RTT	Median	103.9ms	4hr Median	694.6ms	★☆☆☆☆
		33.2ms		34.2ms	
Packet Loss	Fraction	44.0%	Fraction Avg	2.7%	★★★★☆
		11.9%		1.3%	
Perf. Stability	RTT IQR	625.0ms	Loss IDR	0%	★★★★☆
Availability	Down-Events	0	Total Down-Time	0hr	★★★★★
Total Rating					★★★★☆

Figure 6.8.: Rating of the known tunnel which suffers from bad RTT performance

Additional Performance Issue

The rating of an additional tunnel which shows a bad RTT rating is shown in Figure 6.10. We see that it is only suffering from a high RTT. If we look at the RTT distribution in Figure 6.10 then we see that the RTT is unreasonably high during the time when we analyzed the measurement data. Jumps in the RTT as seen in this example, mostly happen because of routing changes.

	Benchmark		Benchmark		Rating
RTT	Median	98.2ms	4hr Median	98.8ms	★☆☆☆☆
		38.4ms		38.5ms	
Packet Loss	Fraction	5.2%	Fraction Avg	2.5%	★★★★★
		7.1%		2.0%	
Perf. Stability	RTT IQR	67.2ms	Loss IDR	0%	★★★★☆
Availability	Down-Events	0	Total Down-Time	0hr	★★★★★
Total Rating					★★★★☆

Figure 6.9.: Rating of a tunnel which suffers from bad RTT performance.

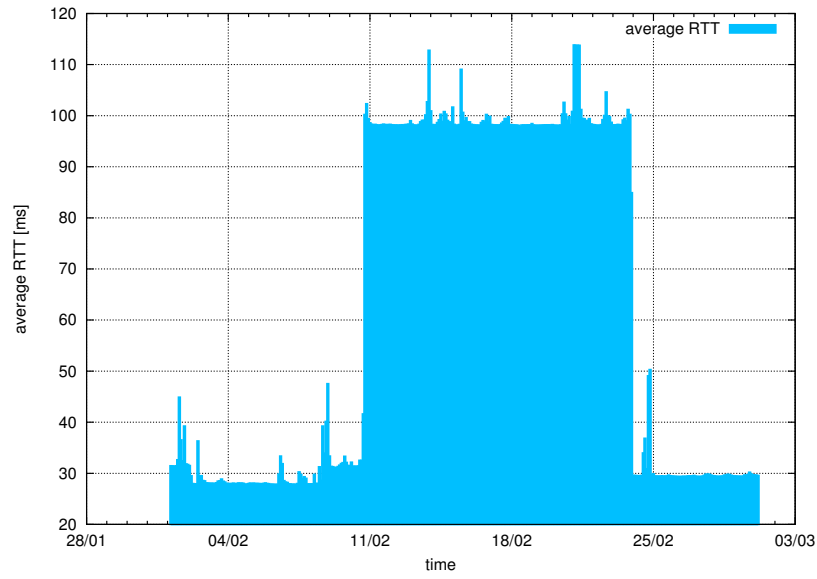


Figure 6.10.: Link suffering from bad RTT performance during the time of the evaluation. Only the RTT in February is shown.

The evaluation showed that the proposed rating methodology allows detecting VPN tunnels which perform below standard.

CHAPTER 7

Conclusion

7.1. Summary

We developed a method to rate the long-term performance of VPN tunnels based on its RTT and packet loss behaviour. The method takes advantage of large scale networks. For every tunnel, geographically similar connections are analyzed. By aggregating the information of similar tunnels, it is possible to retrieve a performance baseline for connections between two locations. A tunnel can then be compared against this performance baseline. We analyzed different approaches on how to search for similar tunnels, including clustering methods.

To rate the long-term performance of VPN tunnels, we specified appropriate metrics which characterize a tunnel's performance. Our rating approach only needs key figures which characterize the performance behaviour of every individual tunnel. Since these key figures can be computed on the VPN endpoints, this approach minimizes additional traffic which is introduced by the rating method.

A ready-to-use prototype was developed, which shows the performance of a company's VPN network in a report. It allows to spot badly performing tunnels in a network. The report was tested against known performance issues and it was verified that the proposed approach allowed detecting tunnels which perform below standard.

7.2. Conclusions

Network performance is highly dependent on geographical factors. We showed that the network latency baseline between two locations is characteristic for tunnels which connect them. Hence the performance of VPN tunnels should be rated by taking the location of the endpoints into account. We developed a rating method for VPN tunnels based on the geographical similarity between tunnels, which allows the long-term rating of tunnel performance. Since our analysis only focuses on end-to-end network performance measurements, the developed method is also applicable for non-VPN networks.

Our rating approach is complementary to existing network monitoring systems and allows detecting links that constantly perform below standard. Spotting the problems in a network allows analyzing the root-cause and finding appropriate solutions.

We showed that it is possible to rate VPN tunnels based on a few characteristic performance metrics which can be locally computed on the monitoring device itself. This distributed calculation of performance metrics ensures good scalability of the performance rating mechanism even in large networks.

7.3. Future Work

7.3.1. Evaluation

It would be interesting to make a long-term evaluation by using the mentioned MOS approach. This would give important insights into the quality and the limits of this rating approach for different purposes.

7.3.2. Technology

Although we showed a way on how to estimate whether a tunnel uses satellite technology or not, we do not take the different technologies into account. It would be of interest to know if a tunnel is established by using xDSL-, MPLS- or satellite technology and also to take the cost-benefit analysis into account. It is not sure that expensive links perform better than low-cost links.

7.3.3. Integration to Network Monitoring

Our prototype was implemented as a stand-alone rating system, whereas it is complementary to existing monitoring systems. The integration of our approach to an existing network monitoring system would introduce the possibility to detect tunnels which are constantly performing badly. A combination of short- and long-term performance analysis would allow a network engineer to get maximum information about a network.

7.3.4. Additional Metrics

We mainly focused on RTT and packet loss measurement in this thesis, while we did not measure the bandwidth of a VPN tunnel. Analyzing the bandwidth of VPN tunnels would give additional information about the tunnel performance.

7.3.5. Internet Line Rating

The analysis showed that, to a certain extent, hosts suffering from packet loss and bad performance because of their local Internet line. Automatically spotting these hosts and estimating whether an ISP change could solve the problem would be very valuable.

APPENDIX A

Appendix

A.1. Terms

AS	Autonomous System
BGP	Border Gateway Protocol
DNS	Domain Name System
ESP	Encapsulating Security Payload
FTP	File Transfer Protocol
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technology
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPPM	IP Performance Metrics
ISOC	Internet Society
ISP	Internet Service Provider
MPLS	Multi-Protocol Label Switching
NGO	Non-Governmental Organization
OSPF	Open Shortest Path First
QoE	Quality of Experience
QoS	Quality of Service
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RRD	Round Robin Database
RTT	Round Trip Time
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over IP
VPN	Virtual Private Network

A.2. Spline Interpolation

```

1 %% Import the data
2 % the file has the following structure
3 % distance_km, median
4 [~, ~, raw] = xlsread('<path to file>', 'rtt_median_aggregation');
5 raw = raw(2:end:);
6 %% Create output variable
7 data = cell2mat(raw);
8
9 %% Allocate imported array to column variable names
10 distancekm = data(:,1);
11 medianrtt = data(:,2);
12
13 %% Calculate the approximation
14 figure
15 maxPlots = 6; % has to be even
16 for i=1:1:maxPlots
17     % calculate the approximation
18     pp = spap2(i,2,distancekm,medianrtt);
19     pp = fn2fm(pp,'pp'); % change to pp form
20     % plot the result
21     subplot(maxPlots/2,2,i); % select the subplot
22     axis = linspace(0,max(distancekm));
23     hold on
24     grid on
25     if (i>1) title_text = sprintf('%d segments', i);
26     else title_text = sprintf('%d segment', i);
27     end
28     title(title_text)
29     xlabel 'distance [km]'
30     ylabel 'RTT [ms]'
31     dataPlot = plot(distancekm, medianRTT, '.');
32     set(dataPlot, 'Color', [.2, .2, .2], 'LineWidth', 1);
33     apprPlot = plot(axis, fnval(pp,axis));
34     set(apprPlot, 'Color', [.0, .749, 1], 'LineWidth', 3);
35     ylim([0 600]);
36
37     % calculate RMSE
38     approxValues = [];
39     for idx=1:size(distancekm)
40         approxValues(idx) = fnval(pp,distancekm(idx));
41     end
42     approxValues = approxValues';
43     MSE = sum((approxValues—medianrtt).^2) ./ size(distancekm,2);
44     RMSE = sqrt(MSE);
45     fprintf('The RMSE with %d anchors is: %d\n', i, RMSE);
46 end

```

Listing A.1: Matlab script to calculate the RTT approximation

A.3. XML Data Export

```
1 <host host_id="0000" host_name="xxxx-xxx-xx-x-x">
2   <tunnel endpoint1_id="0000" endpoint2_id="0001" interface="eth0">
3     <start >1326789000</start>
4     <end>1327998600</end>
5     <step >1800</step>
6     <down_time>0</down_time>
7     <down_events>0</down_events>
8     <load>
9       <data_size >450</data_size>
10      <in_mean>14.8</in_mean>
11      <out_mean>19.1</out_mean>
12    </load>
13    <rtt>
14      <data_size >441</data_size>
15      <median>15.1</median>
16      <min>13.4</min>
17      <max>20.7</max>
18      <median_interval >16.6</median_interval>
19      <percentiles end="100" start="5" step="5">
20        <val >13.708</val>
21        <val >13.860</val>
22        ...
23      </percentiles>
24    </rtt>
25    <loss>
26      <data_size >441</data_size>
27      <loss_steps >6</loss_steps>
28      <loss_steps_mean >0.0066</loss_steps_mean>
29      <mean >5.949953e-05</mean>
30      <min >0</min>
31      <max >0.00668</max>
32      <percentiles end="100" start="5" step="5">
33        <val >0</val>
34        <val >0</val>
35        ...
36      </percentiles>
37    </loss>
38  </tunnel>
39 </host>
```

Listing A.2: Simplified version of the exported statistical data

A.4. File Structure

All report related code is stored in the *script* folder. We describe the content of this folder and where the files are located which are needed to generate a report.

- **data_analysis** Contains all scripts which are needed to create the plots which are shown in this report. It also contains all script which are related to the analysis and understanding of network performance. The clustering evaluation is located also in this folder, since it is not necessary for the creation of a performance report. The content of this folder is not needed for the report.
- **data_export** Contains the scripts which are needed for the export of the performance information from the VPN endpoints.
- **import** Contains all scripts which are needed to import the necessary data about the network topology to the database.
- **lib** Contains all perl-modules which are written during this thesis. The modules are necessary to create a performance report.
- **report** Contains the script which creates a performance report.
- **res** Contains resource files such as world-maps, database scheme and the report templates.
- **statistics** Contains matlab and R scripts to get some more statistic evaluations. The content of this folder is not needed for the report.

APPENDIX B

Sample Performance Report

The following pages show a shorted sample of a VPN performance report as it is created by using our prototype.



mission control™
security services



Open Systems AG
VPN Tunnel Performance

Report - February 2012

Created on March 22, 2012, 10:12

open systems ag
www.open.ch

räffelstrasse 29
8045 zürich (schweiz)

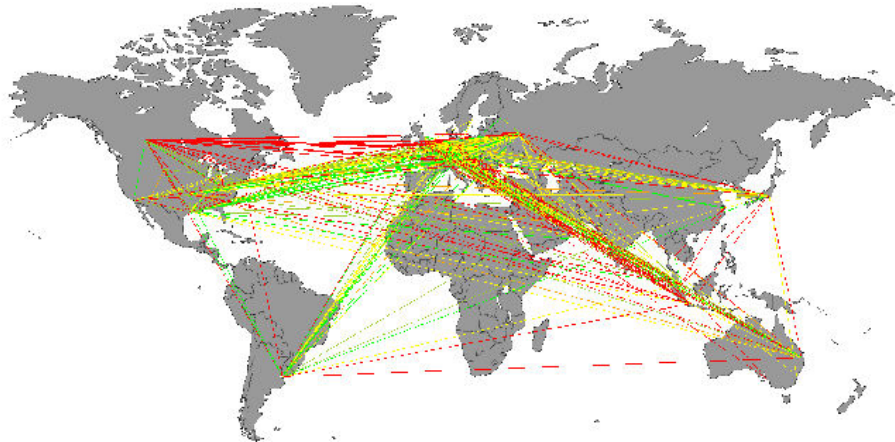
t +41 44 455 74 00
f +41 44 455 74 01



VPN Tunnel Summary

VPN-Tunnels	Change	Type	Endpoints
1542	↔ 0	Change	97

VPN Worldmap



5 Most Used Tunnels

Tunnel	Usage [kbps]
n-0034-shanghai:eth0 - n-0097-singapore:eth0	6144.76
n-0001-winterthur-1:eth0 - n-0113-haag-1:eth0	5121.76
n-0034-shanghai:eth0 - n-0116-shanghai:eth0	3396.51
n-0001-winterthur-1:eth0 - n-0020-tulsa-2:eth0	1449.02
n-0001-winterthur-1:eth0 - n-0121-allschwil:eth0	1419.50

5 Top Outages

Tunnel	Outages	Average Duration [hrs]
n-0003-westbury-2:eth0 - n-0110-barboursville:eth0	2	37.8
n-0003-westbury:eth0 - n-0110-barboursville:eth0	2	37.8
n-0110-barboursville:eth0 - n-0133-durham:eth0	1	74.5
n-0102-troy:eth0 - n-0110-barboursville:eth0	1	74.5
n-0096-dayton:eth0 - n-0110-barboursville:eth0	1	74.5



Executive Management Report February 2012
VPN Tunnel Performance: 5 Most Used Tunnels

n-0034-shanghai:eth0 - n-0097-singapore:eth0

	Benchmark		Benchmark		Rating
RTT	Median	82.3ms	4hr Median	83.2ms	★★★★☆
	91.5ms		164.7ms		
Packet Loss	Fraction	26.5%	Fraction Avg	2.1%	★★★★☆
	39.4%		2.7%		
Perf. Stability	RTT IQR		Loss IDR		★★★★☆
	27.8ms		0%		
Availability	Down-Events		Total Down-Time		★★★★☆
	0		0hr		
Total Rating					★★★★☆

Accuracy	Usage	Remarks
Based on	15 connections	6144.76 kbps
Search Radius	85 km	Average
Similarity	good	4hr Average
		7881.31 kbps

n-0001-winterthur-1:eth0 - n-0113-haag-1:eth0

	Benchmark		Benchmark		Rating
RTT	Median	12.2ms	4hr Median	12.2ms	★★★★★
	8.8ms		9.5ms		
Packet Loss	Fraction	4.6%	Fraction Avg	0.7%	★★★★★
	1.6%		0.7%		
Perf. Stability	RTT IQR		Loss IDR		★★★★★
	1.0ms		0%		
Availability	Down-Events		Total Down-Time		★★★★★
	0		0hr		
Total Rating					★★★★★

Accuracy	Usage	Remarks
Based on	11 connections	5121.76 kbps
Search Radius	25 km	Average
Similarity	very good	4hr Average
		6502.14 kbps

n-0034-shanghai:eth0 - n-0116-shanghai:eth0

	Benchmark		Benchmark		Rating
RTT	Median	1.0ms	4hr Median	1.1ms	★★★★☆
	4.3ms		4.8ms		
Packet Loss	Fraction	6.4%	Fraction Avg	2.2%	★★★★☆
	1.5%		0.9%		
Perf. Stability	RTT IQR		Loss IDR		★★★★☆
	0.6ms		0%		
Availability	Down-Events		Total Down-Time		★★★★☆
	0		0hr		
Total Rating					★★★★☆

Accuracy	Usage	Remarks
Based on	13 connections	3396.51 kbps
Search Radius	1215 km	Average
Similarity	very good	4hr Average
		5775.80 kbps

n-0001-winterthur-1:eth0 - n-0020-tulsa-2:eth0

	Benchmark		Benchmark		Rating
RTT	Median	146.9ms	4hr Median	149.1ms	★★★★★
	149.5ms		149.5ms		
Packet Loss	Fraction	10.9%	Fraction Avg	0.8%	★★★★★
	13.5%				
Perf. Stability	RTT IQR		Loss IDR		★★★★★
	0.9ms		0%		
Availability	Down-Events		Total Down-Time		★★★★★
	0		0hr		
Total Rating					★★★★★

Accuracy	Usage	Remarks
Based on	14 connections	Average
Search Radius	40 km	1449.02 kbps
Similarity	very good	4hr Average
		1771.75 kbps

n-0001-winterthur-1:eth0 - n-0121-allschwil:eth0

	Benchmark		Benchmark		Rating
RTT	Median	6.8ms	4hr Median	7.0ms	★★★★★
	10.9ms		11.4ms		
Packet Loss	Fraction	0.4%	Fraction Avg	1.0%	★★★★★
	1.5%		1.3%		
Perf. Stability	RTT IQR		Loss IDR		★★★★★
	0.7ms		0%		
Availability	Down-Events		Total Down-Time		★★★★★
	0		0hr		
Total Rating					★★★★★

Accuracy	Usage	Remarks
Based on	13 connections	Average
Search Radius	20 km	1419.50 kbps
Similarity	very good	4hr Average
		6803.42 kbps



Executive Management Report February 2012
VPN Tunnel Performance: 3 Worst Tunnels

n-0034-shanghai:eth0 - n-0097-singapore:eth0

		Benchmark		Benchmark	Rating		
RTT	Median	91.5ms	82.3ms	4hr Median	164.7ms	83.2ms	
Packet Loss	Fraction	39.4%	26.5%	Fraction Avg	2.7%	2.1%	
Perf. Stability	RTT IQR	27.8ms		Loss IDR	0%		
Availability	Down-Events	0		Total Down-Time	0hr		
Total Rating							

Accuracy	Usage	Remarks	
Based on	15 connections	Average	6144.76 kbps
Search Radius	85 km	4hr Average	7881.31 kbps
Similarity	good		

n-0085-laporte:eth0 - n-0137-phoenix:eth0

		Benchmark		Benchmark	Rating		
RTT	Median	74.7ms	44.5ms	4hr Median	77.5ms	45.5ms	
Packet Loss	Fraction	10.0%	6.7%	Fraction Avg	2.7%	1.3%	
Perf. Stability	RTT IQR	3.8ms		Loss IDR	0%		
Availability	Down-Events	1		Total Down-Time	0.5hr		
Total Rating							

Accuracy	Usage	Remarks	
Based on	14 connections	Average	581.96 kbps
Search Radius	1496 km	4hr Average	944.50 kbps
Similarity	very good		

n-0001-winterthur-1:eth0 - n-0071-purwakarta:eth0

		Benchmark		Benchmark	Rating		
RTT	Median	324.6ms	335.5ms	4hr Median	330.1ms	339.1ms	
Packet Loss	Fraction	53.0%	56.2%	Fraction Avg	3.1%	3.1%	
Perf. Stability	RTT IQR	25.6ms		Loss IDR	10%		
Availability	Down-Events	0		Total Down-Time	0hr		
Total Rating							

Accuracy	Usage	Remarks	
Based on	10 connections	Average	345.92 kbps
Search Radius	235 km	4hr Average	556.20 kbps
Similarity	good		No variation in the ISPs among the reference tunnels






Executive Management Report February 2012
VPN Tunnel Performance: 3 Best Tunnels

n-0001-winterthur-1:eth0 - n-0113-haag-1:eth0

		 Benchmark		 Benchmark		 Rating	
RTT	Median	8.8ms	12.2ms	4hr Median	9.5ms	12.2ms	★★★★★★
	Fraction	1.6%	4.6%	Fraction Avg	0.7%	0.7%	
Packet Loss	RTT IQR	1.0ms		Loss IDR	0%		★★★★★★
Perf. Stability	Down-Events	0		Total Down-Time	0hr		★★★★★★
Availability	Total Rating					★★★★★★	

Accuracy	Usage	Remarks
Based on	11 connections	Average
Search Radius	25 km	5121.76 kbps
Similarity	very good	4hr Average
		6502.14 kbps




n-0001-winterthur-1:eth0 - n-0059-wohlen:eth0

		 Benchmark		 Benchmark		 Rating	
RTT	Median	5.6ms	17.1ms	4hr Median	6.1ms	18.6ms	★★★★★★
	Fraction	4.0%	6.4%	Fraction Avg	0.9%	1.0%	
Packet Loss	RTT IQR	0.7ms		Loss IDR	0%		★★★★★★
Perf. Stability	Down-Events	0		Total Down-Time	0hr		★★★★★★
Availability	Total Rating					★★★★★★	

Accuracy	Usage	Remarks
Based on	62 connections	Average
Search Radius	20 km	798.66 kbps
Similarity	good	4hr Average
		1428.70 kbps

No variation in the ISPs among the reference tunnels

n-0113-haag-1:eth0 - n-0123-gia-1:eth0

		 Benchmark		 Benchmark		 Rating	
RTT	Median	5.8ms	15.4ms	4hr Median	6.0ms	15.4ms	★★★★★★
	Fraction	2.5%	3.9%	Fraction Avg	0.8%	1.0%	
Packet Loss	RTT IQR	0.3ms		Loss IDR	0%		★★★★★★
Perf. Stability	Down-Events	0		Total Down-Time	0hr		★★★★★★
Availability	Total Rating					★★★★★★	

Accuracy	Usage	Remarks
Based on	10 connections	Average
Search Radius	30 km	20.11 kbps
Similarity	very good	4hr Average
		35.68 kbps

Bibliography

- [1] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. URL <http://dl.acm.org/citation.cfm?id=1283383.1283494>.
- [2] Prasad Calyam, Mukundan Sridharan, Weiping Mandrawa, and Paul Schopis. Performance Measurement and Analysis of H.323 Traffic. In Chadi Barakat and Ian Pratt, editors, *Passive and Active Network Measurement*, volume 3015 of *Lecture Notes in Computer Science*, pages 137–146. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-21492-2. URL http://dx.doi.org/10.1007/978-3-540-24668-8_14. 10.1007/978-3-540-24668-8_14.
- [3] M. Costa, M. Castro, R. Rowstron, and P. Key. Pic: Practical Internet Coordinates for Distance Estimation. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, pages 178 – 187, 2004. doi: 10.1109/ICDCS.2004.1281582.
- [4] Les Cottrell. Talk at University of Helwan about: How is the Internet Performing? 2010. URL <http://www.slac.stanford.edu/grp/scs/net/talkk10/perform.ppt> (visitedMarch2012).
- [5] L. Ding and R.A. Goubran. Assessment of Effects of Packet Loss on Speech Quality in voip. In *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE International Workshop on*, pages 49 – 54, sept. 2003. doi: 10.1109/HAVE.2003.1244724.
- [6] Paul Ferguson and Geoff Huston. What is a VPN. April 1998.

- [7] P. Francis, S. Jamin, V. Paxson, Lixia Zhang, D.F. Grynewicz, and Yixin Jin. An Architecture for a Global Internet Host Distance Estimation Service. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 210–217 vol.1, mar 1999. doi: 10.1109/INFCOM.1999.749285.
- [8] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, IMW '02*, pages 5–18, New York, NY, USA, 2002. ACM. ISBN 1-58113-603-X. doi: <http://doi.acm.org/10.1145/637201.637203>. URL <http://doi.acm.org/10.1145/637201.637203>.
- [9] Qi He, Constantine Dovrolis, and Mostafa Ammar. On the Predictability of Large Transfer TCP Throughput. *SIGCOMM Comput. Commun. Rev.*, 35:145–156, August 2005. ISSN 0146-4833. doi: <http://doi.acm.org/10.1145/1090191.1080110>. URL <http://doi.acm.org/10.1145/1090191.1080110>.
- [10] ITU-T. ITU-T G.107E: The E-model: A Computational Model for Use in Transmission Planning Subjective Determination of Transmission Quality, .
- [11] ITU-T. ITU-T Recommendation G.826: End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections, .
- [12] ITU-T. ITU-T P.800: Methods for Subjective Determination of Transmission Quality, .
- [13] ITU-T. ITU-T G.114: One-way transmission time, May 2003.
- [14] Peter J. and Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL <http://www.sciencedirect.com/science/article/pii/0377042787901257>.
- [15] Wenyu Jiang and Henning Schulzrinne. Modeling of packet loss and delay and their effect on real-time multimedia service quality. In *PROCEEDINGS OF NOSSDAV '2000*, 2000.
- [16] William H. Kruskal and W. Allen Wallis. Use of Ranks in One-criterion Variance Analysis. *Journal of the American Statistical Association*, 47:584 – 618, 1952.
- [17] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP Quality over Internet Backbones. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 150 – 159 vol.1, 2002. doi: 10.1109/INFCOM.2002.1019256.
- [18] T.S.E. Ng and Hui Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 170 – 179 vol.1, 2002. doi: 10.1109/INFCOM.2002.1019258.

-
- [19] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling tcp Throughput: A Simple Model and its Empirical Validation. *SIGCOMM Comput. Commun. Rev.*, 28:303–314, October 1998. ISSN 0146-4833. doi: <http://doi.acm.org/10.1145/285243.285291>. URL <http://doi.acm.org/10.1145/285243.285291>.
- [20] F. Palmieri. VPN Scalability over High Performance Backbones Evaluating MPLS VPN against Traditional Approaches. In *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pages 975 – 981 vol.2, june-3 july 2003. doi: 10.1109/ISCC.2003.1214243.
- [21] J.T. Parrott, J.R. Moyne, and D.M. Tilbury. Experimental Determination of Network Quality of Service in Ethernet: UDP, OPC, and VPN. In *American Control Conference, 2006*, page 6 pp., june 2006. doi: 10.1109/ACC.2006.1657491.
- [22] V. Paxson, J. Mahdavi, M. Mathis, and G. Almes. Framework for ip performance metrics, October 2011. URL <http://tools.ietf.org/html/rfc2330>.
- [23] Vern Paxson. End-to-End Internet Packet Dynamics. *SIGCOMM Comput. Commun. Rev.*, 27:139–152, October 1997. ISSN 0146-4833. doi: <http://doi.acm.org/10.1145/263109.263155>. URL <http://doi.acm.org/10.1145/263109.263155>.
- [24] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41:53–56, April 2011. ISSN 0146-4833. doi: <http://doi.acm.org/10.1145/1971162.1971171>. URL <http://doi.acm.org/10.1145/1971162.1971171>.
- [25] Joel Sommers, Paul Barford, Nick Duffield, and Amos Ron. Improving Accuracy in End-to-End Packet Loss Measurement. *SIGCOMM Comput. Commun. Rev.*, 35:157–168, August 2005. ISSN 0146-4833. doi: <http://doi.acm.org/10.1145/1090191.1080111>. URL <http://doi.acm.org/10.1145/1090191.1080111>.
- [26] Manuel Stich. Application Level Network Performance Monitoring. Master’s thesis, EPFL Lausanne, 2011.
- [27] Li Tang, Hui Zhang, Jun Li, and Yanda Li. End-to-end Delay Behavior in the Internet. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, pages 373 – 382, sept. 2006. doi: 10.1109/MASCOTS.2006.22.
- [28] Jonas Wagner. Detection of Bad Performance in VPN Tunnels. Master’s thesis, EPFL Lausanne, 2011.
- [29] Li Zheng, Liren Zhang, and Dong Xu. Characteristics of Network Delay and Delay Jitter and its Effect on Voice over IP (VoIP). In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 1, pages 122 –126 vol.1, jun 2001. doi: 10.1109/ICC.2001.936286.

- [30] Laurent Zimmerli. Rating Autonomous Systems. Master's thesis, ETH Zurich, 2008.