How asymmetric is the Internet?
A Study to Support the use of Traceroute

Wouter de Vries, José Jair Santanna, Anna Sperotto, and Aiko Pras

University of Twente
Design and Analysis of Communication Systems (DACS)
Enschede, The Netherlands
{w.b.devries-1@student.utwente.nl;j.j.santanna@utwente.nl
a.sperotto@utwente.nl;a.pras@utwente.nl}

Abstract. A network path is a path that a packet takes to reach its target. However, determining the network path that a host uses to reach it's target from the viewpoint of the latter is less trivial than it appears. Tools such as Traceroute allow the user to determine the path towards a target (i.e. the forward path), but not the path from the target to the source (i.e. the reverse path) due to routing asymmetry. Routing asymmetry means that the network path between two hosts may be different in opposite directions. Although previous studies have shown that this asymmetry is widespread, a more detailed characterization is lacking. In this paper routing asymmetry is investigated in depth using large scale measurements with 4.000 probes distributed world wide. The main goal of this paper is to provide characteristics about Internet asymmetry based on recent large scale measurements. Our findings contribute to a conclusive overview of Internet asymmetry, which assist researchers and engineers in making valid assumptions about routing asymmetry.

Keywords: Internet, Asymmetry, Large Scale Measurements

1 Introduction

The fact that Internet routing shows some degree of asymmetry has long been known [4,5,9,10]. Routing asymmetry means that, given two hosts A and B, the path from A to B (the forward path) is different from the the path from B to A (the reverse path). Asymmetry can, for example, be problematic when trying to troubleshoot problems at host A that occur on the reverse path. The reason for this is that standard tools, such as Traceroute, are only able to determine the forward path from the viewpoint of host A.

There have been various studies that quantify Internet routing asymmetry. This study aims to reinforce those studies and provide a more in depth analysis to determine where exactly this asymmetry occurs. A better understanding of the characteristics of Internet asymmetry can, for example, help when attempting to troubleshoot problems that occur on the reverse path when only the forward path is known.

In this paper we look into the asymmetry of network paths. We investigate to what extent the reverse path can still be determined using the forward path if the characteristics of Internet asymmetry are known. The goal of this study is to provide an in depth analysis of Internet routing asymmetry. To perform this analysis we measure network paths between 4.000 probes across the world. We analyze the resulting data for network path asymmetry from the Autonomous System (AS) level. We show that most routes are not completely symmetrical, although the routes do have properties that still make them useful for specific applications, such as troubleshooting and collaboration with upstream providers. The contribution of this paper is providing information that researchers and engineers can use for the practical applicability of forward/reverse paths.

This paper is organized as follows. In Section 2 we discuss the related work followed by Section 3, where we explain our hypothesis. In Section 4 we describe our data acquisition. Then, in Section 5, the analysis will be described. Finally, we will present our conclusions in Section 6.

2 Related work

Researchers have been studying Internet routing asymmetry for some time [4, 5, 7, 10]. In this section we will discuss a few studies that have investigated the level of routing asymmetry on the Internet and indicate what shortcomings they have that we aimed to solve.

First, the research in [5] on route asymmetry covers the AS level. They conclude that route asymmetry, on the AS level, is only present in approximately 14% of the routes. However, this research is based on results gathered using the Active Measurement Project (AMP) which runs mainly on academic networks and uses only 135 probes. In their follow up study [4], they use 350 probes selected from 1200 public traceroute servers. They note that the routing asymmetry percentage is much higher on commercial networks, namely 65%, which negatively impacts the usability of Traceroute to measure reverse network paths.

In addition, while they have conducted extensive research on route asymmetry on the AS level they have not looked at the relative position of asymmetry (e.g. close to the target of the traceroute, in the middle or close to the source of the traceroute). If we are interested in the remaining usability of reverse paths this is an interesting measurement, for example for applications that do not require the entire path to be symmetric. They proposed an interesting framework for quantifying the change in paths in which they use the the Levenshtein Edit Distance (ED) algorithm as a way to determine the distance between two paths.

Secondly, research in [10] concluded that the asymmetry on the AS level is substantially higher than in [4, 5]. According to them, asymmetry on the AS level is as high as 90%. The cause of this difference could, for example, be that this study was conducted 5 years later or that their dataset is obtained using only a total of 220 probes biased distributed.

Finally, the research in [7] proposes a way of determining the actual path that a packet has taken to reach a point in a network, with routing asymmetry in

mind, from the viewpoint of the receiver. They do this mainly for troubleshooting purposes (e.g. which network is dropping packets). Their method involves a system of widely deployed probes, IP spoofing and the use of an option in the IP header that is often not implemented. While the theory behind this method is sound, it can be difficult to deploy in practice for a few reasons. First, potential users need to have widely deployed probes in place. Secondly, their method uses the Record Route option in the IP header. However, this option is often ignored [6] and packets that use this option are usually dropped. Finally, the use of IP spoofing, the act of forging the source address, can be problematic due to issues with company policies, ethics and the fact that there are techniques to block IP spoofing such as proposed in Request for Comment (RFC) 2827 [2], which is currently known as Best Current Practice (BCP) 38.

3 Hypothesis

The goal of this section is to describe some terminology and concepts. Then, we will introduce the hypothesis.

In this paper we consider a *network path* an ordered list of networks that connect two end-systems on the Internet. Although there are studies that differentiate networks by IP address or even as IP address range [4], we chose to represent networks as Autonomous System (Autonomous System (AS)). By using ASes it is trivial to cluster IP addresses that belong to the same administrative network.

As shown in Fig. 1 there are two distinct paths between a pair of end-systems A and B: The forward path and the reverse path. When both paths are completely equal then the path is symmetric, otherwise it is asymmetric. Note that to reliably determine a complete network path from the viewpoint of the receiver, the Internet would have to be completely symmetric.

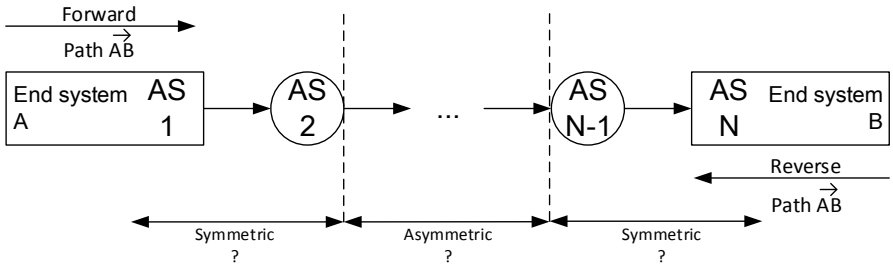


Fig. 1: Network path

Our goal is to show that network paths are symmetric near the end-systems, meaning that for that section of the network path standard tools can be used to determine the reverse path. We defined the following hypothesis: *The reverse*

network path can be reliably discovered via standard tools, such as Traceroute, near the end-systems..

4 Data acquisition

In this section we will describe the methodology that we used. We will first explain which measurement network we use and what it consists of. Then, we will explain how the measurements are configured. Lastly, we will present some preliminary considerations concerning the measurements.

The main requirement for investigating our hypothesis was having a large amount of Internet connected computer systems which we could control. In order to meet this requirement we use RIPE Atlas. This project manages probes around the world for the specific purpose of network measurements. A probe is a dedicated network measurement device that can be placed in a network to allow measurements to be performed remotely. The Atlas project consists of approximately 7.000 distributed probes[1] worldwide. Although we are aware of several other measurement infrastructures, such as PlanetLAB[2], EmanicsLAB[3] and the NLNOG Ring[4], these do not provide the scale and distribution that was required for measurements that are representative of the Internet.

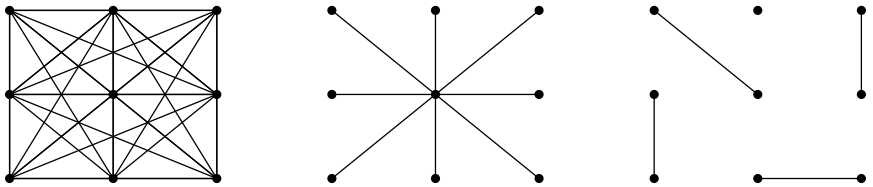
RIPE Atlas has imposed a credit system that limits measurements in three ways. The credits that are consumed per day, the number of measurements that can be run concurrently and the total number of credits that can be consumed. These limits have a consequence on the number of probes that can be used and in which combination. Credits can, for example, be earned by hosting a RIPE Atlas probe. It is due this credit limit that not all probes that are available can be used. This further depends on the measurement layout, which probe measures what and to what other probe.

4.1 Measurement configuration

We considered three layouts in which the probes can conduct the measurements. Note that to be able to determine route asymmetry between two probes, each probe has to traceroute the other. In the considered layouts each probe performs traceroutes to the probes to which it is connected.

Fully connected layout (Fig. 2a) - This layout has the advantage of utilizing the complete potential of the involved probes, every probe measures the path to every other probe. The disadvantage is that due to the credit limit only a very limited amount of probes from the total can be used. For example: considering the 1 million credit limit only 112 probes can be used due to the high amount of paths in this type of layout. A small amount of probes means that specific network issues that occur at individual probes have a large impact.

[1] RIPE Atlas System Statistics: https://atlas.ripe.net/
[2] https://www.planet-lab.org/
[3] http://www.emanicslab.org/
[4] https://ring.nlnog.net/



(a) Fully connected (b) Star (c) Random Tuples

Fig. 2: Probe layout

Star layout (Fig. 2b) - In comparison to the fully connected topology this has the advantage of allowing many more probes to be used. However, in this case the center probe will have a large impact on each measurement. Network issues at the center probe can cause the entire experiment to fail.

Random tuple layout (Fig. 2c) - In this layout random tuples of probes are selected. This has the advantage of minimizing the impact of a single misbehaving probe. Furthermore, it allows for a much larger selection of probes, considering the Atlas limits. Because of these advantages this is the layout that we used.

Using the random tuple layout we selected 4.000 probes, meaning 2.000 tuples, in a way that favoured longer geographic distances. The attempt to have longer geographic distances is to prevent a large concentration of probes in Europe, as most probes are located there. The algorithm used to select the probes works by randomly picking probes and comparing the distance between them to some threshold (in our case: 10.000 km), if the threshold is exceeded then the probe tuple is added to the final result set. If, after a number of attempts (in our case: 2.000), no probe tuples can be found that exceed the threshold then the threshold will be lowered.

The distribution over continents in terms of numbers is shown in Table 1. There is a large skew towards Europe which is caused by the relatively large number of probes located there. The average distance between two probes in a tuple is 6.945 kilometres (as the crow flies).

For every selected pair consisting of probe A and probe B two measurements were scheduled. One measurement, consisting of a traceroute, was configured from probe A to probe B (the forward path) and another from probe B to probe A (the reverse path).

Network variances over time were smoothed out by scheduling the measurements to run every three hours, for ten days. This was limited by the total amount of credits we were allowed to consume. The measurements were performed from 14:00 on the 28th of July 2014 to 14:00 on the 7th of August 2014, Coordinated Universal Time (UTC).

Table 1: Distribution over continents

Continent	Selected	Available	Fraction	Fraction of selected
Europe	2681	5200	51.56%	67.03%
North America	724	1003	72.18%	18.10%
Asia	267	420	63.57%	6.68%
Africa	157	223	70.40%	3.93%
Oceania	109	145	75.17%	2.73%
South America	59	87	67.82%	1.48%
Antarctica	1	1	100.00%	0.03%
Unknown	*2*	*4*	*50.00%*	*0.05%*
Total	***4000***	***7083***		***100%***

4.2 Preliminary considerations

RIPE Atlas probes conduct their traces on the IP level where each hop consists of a single IP address. Because we want to look at the network paths from the AS level it was necessary to convert the measured paths. In order to convert IP addresses to their corresponding AS numbers we used the BGP routing table dumps obtained from the Remote Route Collector (RRC)s managed by the Routing Information Service (RIS), which in turn is operated by RIPE. These routing tables contain a large amount of routes that are announced on the Internet by different ASes. Using these routes we are able to determine the AS number for a given IP range. The tool we used for this and its source is available online[5]. Alternatives to this method are, for example, provided by CAIDA[6] or MaxMind[7].

Each IP address in the paths on the router level was converted to their corresponding AS number. It is apparently common for multiple hops to occur within the same network. This is shown by the reduction in the number of hops in network paths on the router level in comparison to network paths on the AS level, which is, on average, 64.46%.

Our choice of probes was optimized to prevent a large cluster of probes in Europe by increasing the geographic distance between pairs, this may have introduced a bias in network path length. In order to show that this is not the case we plot the geographic distance, which is shortest distance between two points on a sphere (great circle distance), against the number of hops in the forward network path on the AS level. The result of this is shown in Fig. 3. As we expected there appears to be no clear relation between the geographic distance and the number of hops.

[5] IPASNExporter: https://bitbucket.org/woutifier/ipasnexporter
[6] http://www.caida.org/
[7] https://www.maxmind.com/

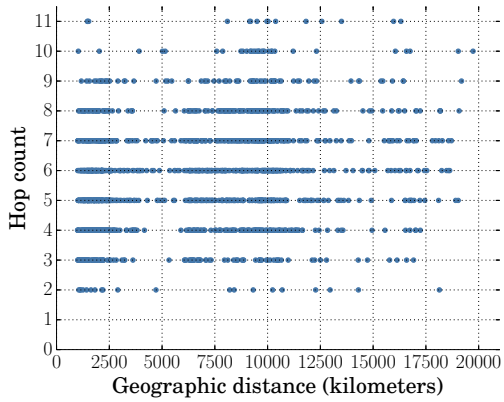


Fig. 3: Geographic distance vs path length

In Fig. 4 the distribution of the length of the measured paths is shown. Most paths contain five different AS-numbers. This means that in those cases three autonomous systems aside from the one the receiver and the sender are in (e.g. their ISPs) are involved in routing the packets.

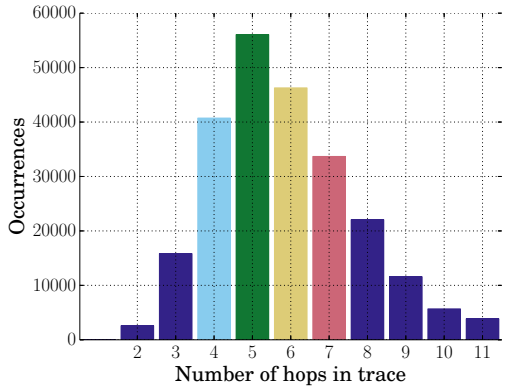


Fig. 4: Distribution of path length

The measurements that were performed by the probes were not completely perfect or complete. This may, for example, have been caused by probes that delayed their measurements for too long or did not perform them at all. Because of this we have applied some filters to the dataset. Prior to the filtering we had 153,638 potential forward/reverse path pairs, of a theoretical 160,000. The absence of some paths can be attributed to some probes that did not respond or did not complete any measurements.

Since forward and reverse path measurements are initiated from two different probes there can be a delay between the two measurements being executed. To

prevent this difference from influencing the results we only match paths if they were measured no more than 600 seconds apart. If they are outside that time limit the forward/reverse pair is discarded. This prevents path instability from being interpreted as path asymmetry. This filter reduced our potential forward/reverse path pairs by 16,103.

The second filter we implemented is based on the principle that the first hop of the forward path should be the last hop of the reverse path, because these are the origin and destination networks. The same principle applies in the opposite direction. Measurements where this is not the case can be caused by incomplete traces. We filtered all forward/reverse path pairs where this was the case. This removed 14,620 results from the set.

To prevent probes that measured completely empty paths to influence the results we filtered all pairs that contained a completely empty path. Completely empty paths do not exist in actual networks, as a network path always contains at least a single hop, even if the source and target IP addresses are in the same network. This can be caused by incomplete traces or probes that are not executing their measurements. This filter reduced our result set by 3,365.

The three filters that we implemented left a total of 119,550 or 74.72% of the theoretical 160,000 pairs.

For paths that contained unresolvable hops we considered a few options. The first option is to discard all path pairs that contained such a hop. However, this would impact a significant part of the result set as unresolvable hops are common. Another option, which was also implemented in [3] is to simply consider an unresolvable hop as a wild card, meaning that it will match any hop in the opposite path that is in the same position.

5 Analysis

In this section we analyze the dataset that was obtained using the methodology described in the previous section. Our dataset contains a total of 2275 unique AS numbers, of which 1717 contain one or more probes. Of all results in our dataset, 15053 (12.6%) forward/reverse path tuples are completely symmetric and 104497 (87.4%) show asymmetry. This is in line with the results found in [10], however, we use far more probes. The large percentage of asymmetric paths further justifies studying the characteristics of Internet asymmetry.

Before we start the analysis we introduce two variants for calculating the Edit Distance (ED) between two paths. One is the Levenshtein algorithm [8] which was first used for this purpose in [4] [5]. The Levenshtein algorithm counts the number of required insert, delete or change operations to make two paths equal to each other. The Levenshtein algorithm was originally intended to be used to measure the differences between strings, however, it can be used without modification for measuring the change in network paths. In addition to the Levenshtein algorithm we also use a variation called Damerau-Levenshtein [1]. Damerau-Levenshtein extends the original algorithm by also counting transpose operations as a single change. It is much less sensitive to swapped hops. The

extended algorithm is interesting in contexts where the presence of ASes on a path are of more importance than their specific location.

5.1 Stability over time

We begin our investigation by determining the change of paths over time. This is of interest because it is not always possible to measure the reverse path at the exact time that the forward path was established. We calculate the average ED over all paths over time. The ED is determined as follows: The first path to a destination is taken as a ground truth to which each consecutive path is compared. We then calculate the ED based on the Levenshtein algorithm. We had to modify the algorithm slightly because not all paths are of the same length, which would cause longer paths to have a much higher impact on the results than shorter paths. Therefore, we normalize the ED by dividing it by the path length as shown in formula 1.

$$\frac{ED(forward, reverse)}{MAX(len(forward), len(reverse))} \tag{1}$$

The normalized ED is between 0.0 (i.e. completely symmetric) and 1.0 (i.e. completely asymmetric). Fig. 5 shows the results of this analysis. Note that the graphs indicate that network paths are not subject to great change over time. The instability appears to stop increasing after 8 days, therefore measurements should be done over a longer period of time to show if this behavior persists. Furthermore, we compared the results using the Levenshtein algorithm to the Damerau-Levenshtein algorithm and this showed results which are almost completely identical. This indicates that the relative position of a network in a path is stable.

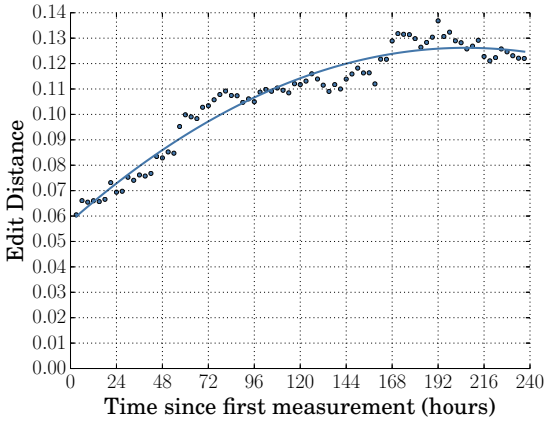


Fig. 5: ED over time using Levenshtein algorithm

5.2 Absolute difference

We look at the absolute difference between the forward and reverse path pairs to get an understanding of how big the impact of routing asymmetry is. We define the absolute difference as the ED between the forward and reverse path. The ED between all path pairs is shown in Fig. 6. Note that the difference between the results of the two algorithms indicates that it is a common occurrence for two hops to be swapped in either the forward or reverse path. Furthermore, most forward/reverse path pairs show a distance of either 1 or 2 from their counterpart.

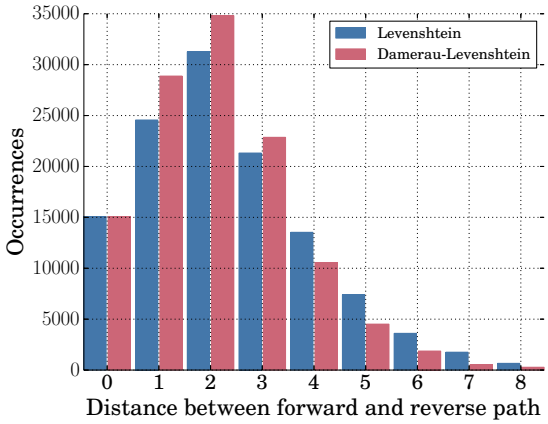
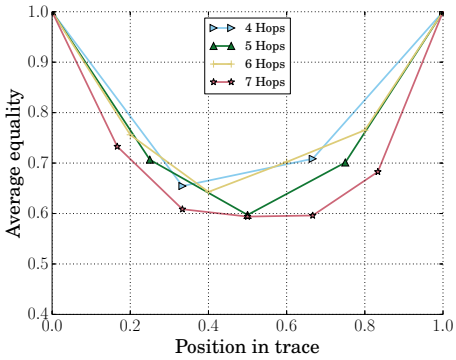


Fig. 6: Distance between forward and reverse path

5.3 Relative difference by position

In this section we show the similarity of hops based on their relative position in the path. This shows if a certain hop is usable for mitigation. If a forward and reverse trace have different lengths then they are not included in this figure, which results in 28139 result pairs being used in Fig. 7. This shows how the symmetry decreases as we move closer to the middle of the path, as expected. It also shows that for the longest path (7 hops) the middle hop is equal in both the forward and reverse path in approximately 60% of the cases.

Given this measure of asymmetry we try to find out if the majority of asymmetry is caused by a small number of networks (i.e. ASes). We look at which ASes are involved when asymmetry occurs. From the approximately 500 ASes that are involved we see that the top 10 is responsible for 48% of the total asymmetry. We manually categorized these ten ASes in three types: T1 for Tier 1 providers, T2 for Large ISPs and IXP for Internet Exchange Points. The results are shown in Table 2. It is obvious that the largest Internet Service Provides (i.e. Tier 1 providers), cause the largest part of the asymmetry. It is likely that



#	Average equality
1	1.00
2	1.00 1.00
3	1.00 0.81 1.00
4	1.00 0.65 0.71 1.00
5	1.00 0.71 0.60 0.70 1.00
6	1.00 0.76 0.64 0.70 0.77 1.00
7	1.00 0.73 0.61 0.59 0.60 0.78 1.00

Fig. 7: Average equality by position in trace

this is because those providers are also the ones which have the most peering connections.

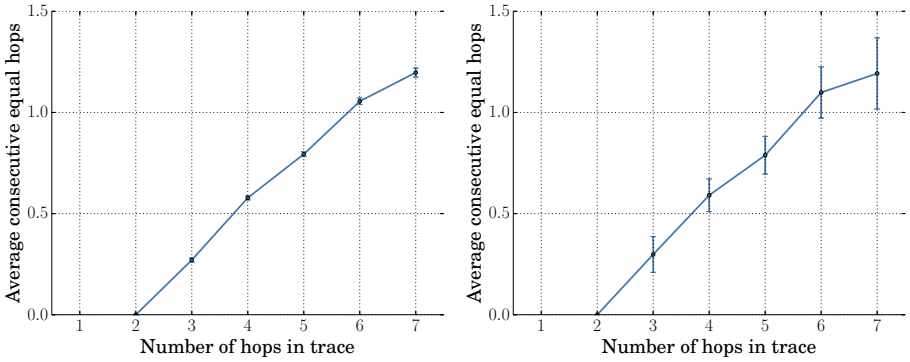
Table 2: Top 10 ASes involved in asymmetry

Position	ASN	Name	Type
1	3356	Level 3 Communications, Inc.	T1
2	174	Cogent Communications	T1
3	1299	TeliaSonera International Carrier	T1
4	3257	Tinet SpA	T1
5	3216	OJSC Vimpelcom	T2
6	34984	TELLCOM ILETISIM HIZMETLERI A.S.	T2
7	1200	Amsterdam Internet Exchange B.V.	IXP
8	2914	NTT America, Inc.	T1
9	6453	TATA Communications, Inc.	T1
10	6695	DE-CIX Management GmbH	IXP

5.4 Consecutive equal hops

We count the number Consecutive Equal Hops (CEH) from each side of the forward/reverse path that are equal, not counting the source and target networks. This approach can be used even if the lengths of the forward/reverse path are unequal. The average number of CEH, divided by two to get an average for each side, is plotted against the total number of hops in the forward path in Fig. 8a.

Included in Fig. 8a is the 95% confidence interval. This figure shows that for path lengths 6 and 7 there is on average at least one additional equal network aside from the source and target networks. For the most common path length, five, there is one network that is the same in both the forward and reverse path in approximately 75% of the cases.



(a) All results (b) First result per pair only

Fig. 8: Average number of CEH between forward and reverse paths

In Fig. 8b only the first complete result for each pair is considered. These graphs show that it is not necessary to do repeated measurements over a longer period of time to determine route asymmetry. Note that this suggests that route asymmetry does not vary significantly over time.

6 Conclusion

In this paper we have analyzed and characterized several aspects of Internet routing asymmetry. Our analysis has been conducted on a large scale using RIPE Atlas. The results from our study contribute to assist researchers and engineers in making valid assumptions while using forward/reverse paths data. In addition, we contribute to give a conclusive overview on the partial asymmetry of Internet routing.

The usability of Traceroute for measuring reverse paths is, depending on the application, questionable. We have confirmed the presence of asymmetry in the majority of Internet routes, and determined where this asymmetry occurs. Our hypothesis, that reverse network paths can be reliably discovered via standard tools near the end-systems has been confirmed. We have found, in the worst case, a hop, representing an AS, is the same in the forward and the reverse path in 59% of the cases, but often more.

As future work we plan to extend the analysis on the IP-level. Furthermore, we plan to apply machine learning to estimate network path accuracy given certain indicators, such as the type of networks that are involved and the length of the path.

Acknowledgments We would like to thank RIPE Atlas for facilitating the measurements. In addition, we would also like to thank Roland van Rijswijk for his insights. This work was funded by the Network of Excellence project FLAMINGO (ICT-318488), which is supported by the European Commission under its Seventh Framework Programme.

References

1. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM 7(3), 171–176 (1964)
2. Ferguson, P., Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice) (2000), http://www.ietf.org/rfc/rfc2827.txt
3. He, Y.H.Y., Chen, W.C.W., Xiao, B.X.B., Peng, W.P.W.: An Efficient and Practical Defense Method Against DDoS Attack at the Source-End. 11th International Conference on Parallel and Distributed Systems (ICPADS'05) 2 (2005)
4. He, Y., Faloutsos, M., Krishnamurthy, S., Huffaker, B.: On routing asymmetry in the Internet. In: Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE. vol. 2, pp. 6 pp.– (Nov 2005)
5. He, Y., Faloutsos, M., Krishnamurthy, S.V.: Quantifying routing asymmetry in the Internet at the AS level. In: GLOBECOM. pp. 1474–1479. IEEE (2004), http://dblp.uni-trier.de/db/conf/globecom/globecom2004.html#HeFK04
6. Iputils: ping(8) (2014), http://man7.org/linux/man-pages/man8/ping.8.html
7. Katz-Bassett, E., Madhyastha, H., Adhikari, V.: Reverse traceroute. In: Network Systems Design and Implementation (2010), https://www.usenix.org/legacy/event/nsdi10/tech/full_papers/katz-bassett.pdf
8. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
9. Paxson, V.: End-to-end Routing Behavior in the Internet. In: Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications. pp. 25–38. SIGCOMM '96, ACM, New York, NY, USA (1996), http://doi.acm.org/10.1145/248156.248160
10. Schwartz, Y., Shavitt, Y., Weinsberg, U.: On the Diversity, Stability and Symmetry of End-to-End Internet Routes. In: INFOCOM IEEE Conference on Computer Communications Workshops , 2010. pp. 1–6 (2010)