Testing the reachability of (new) address space

Randy Bush
IIJ
randy@psg.com

James Hiebert
University of Oregon
james@hiebert.org

Olaf Maennel, Matthew Roughan
University of Adelaide
{olaf.maennel,matthew.roughan}@adelaide.edu.au

Steve Uhlig
TU Delft
S.P.W.G.Uhlig@ewi.tudelft.nl

ABSTRACT

The Internet was originally designed to provide connectivity from every node to every other node. However, policies can impede this connectivity [1]. This is especially true for newly allocated address space. Some Internet Service Providers (ISPs) simply do not realize that the status of previously unallocated address space has changed, and they continue blocking that space. Therefore, it would be desirable to test whether filters block newly apportioned address space before it is allocated to ISPs and/or end users.

In this paper we present a methodology that aims to detect incorrectly configured filters, so that ISPs can be contacted and asked to update their router configurations. Our methodology is capable of detecting paths on which reachability is actually present but which are routed around an outdated filter configuration, as well as cases where a destination is inaccessible. To help narrowing down the most likely ASs that actually filter, we introduce a filtering likelihood index.

We apply our methodology on newly allocated address space and perform initial experiments on a large fraction of ISPs, covering over 80% of all Autonomous Systems (ASs).

1. INTRODUCTION

1.1 Problem statement

The fundamental goal of the Internet was originally to provide connectivity between computers. However, its very success has created a problem. Now that the Internet is a crucial part of our modern infrastructure, the focus has moved from providing a free, open connectivity service to providing guarantees to paying customers. Part of this migration has been the increasing focus on securing the Internet against malicious, and careless damage to its infrastructure. However, there is an inherent tension between providing open connectivity and adequate security. Firewalls, middleboxes, QoS appliance and so on all enhance certain aspects of the Internet while breaking the simple end-to-end semantics that made the Internet so successful in the first place.

Malicious address hijacking [2,3], various attacks (incl. spoofed source addresses) [4], and even inadvertent misconfigurations [5] are among the problems that ISPs face daily. It is common for these attacks/problems to appear to originate from unallocated address space – the so called bogon address space [6,7] – and hence they can be easily eliminated by simply filtering out traffic or routing announcements from this address space. Bogon filters therefore provide a useful management tool to remove certain problems that plague the Internet.

However, new address space becomes allocated continuously [8], so it is important that these filters are up to date. If they are not, they themselves may result in partitioning the Internet – some parts may not be able to reach others. For example, a previously unallocated prefix might have been used in the past by spammers. Thus it was filtered to help protect the ISP, but after the address is allocated, the filter may now cut off paying customers. The problem mainly arises for ISPs who receive newly allocated address space from the Regional Internet Registries (RIRs), when portions of the Internet still filter this address space.

The operator community is well aware of this issue. Still, today's common practice is to announce the newly allocated prefixes and then manually debug where it is visible and where it is not. This takes a significant amount of human labour and communication with other ISPs. Often those problems remain undetected until a customer calls and complains. Ideally, before address space is allocated to ISPs and end users, the RIR's should be able to test whether the address space is blocked.

In this paper we present a methodology to detect the location of such filters. We show that those incorrectly configured filters are common for newly allocated prefixes, and observe that it takes some ISPs a substantial period of time to update their filters.

We observe further that because some ISPs update their filters, while others still block, traffic flows around those ASs. Even though reachability exists for most locations in the Internet, those intermediate filters cause traffic to take different IP-level paths than intended.

Our goal is to obtain a better understanding of the reachability/unreachability of address space, how ISPs manage their policies, and how they update their filters. We hope that this work thereby helps to reduce the workload of operators as well as improve the service that the RIRs provide, ultimately improving upon the quality of the Internet.

1.2 Approach

In this section we introduce our approach for a system that identifies incorrectly configured filters.

First, the portion of address space that is intended to be allocated in the near future is temporarily assigned to the testing service. A set of *test-boxes* that are strategically scattered throughout the Internet announce *test-prefixes* (a separate test-prefix for each test-

INM'07, August 31, 2007, Kyoto, Japan.

ISP	Location	AS	test-block	test-IP	anchor-IP
PSGNet	USA	AS 3130	96.0.0.0/24	96.0.0.1	147.28.0.35
SpaceNet	Germany	AS 5539	96.0.16.0/20	96.0.16.1	194.97.144.209
Citylink	New Zealand	AS 23754	97.64.46.0/24	97.64.46.2	202.8.44.44
IIJ	Japan	AS 2497	96.0.32.0/20	96.0.32.2	210.130.133.54

Figure 1: Configuration details of our four test-sites.

box). In addition each test-box announces an *anchor-prefix* [9]. The *anchor-prefix* is a well established prefix, part of an address block that has been used for some time and is known to be reachable. Each of the test- and anchor-prefixes have a pingable IP address on a computer belonging to the testing service, called the *test-IP* and *anchor-IP* respectively.

The next step is to run traceroutes from various locations towards the test-IP as well as the anchor-IP. We call these *in-probes*. As the test-prefix and the anchor-prefix are announced from the same router, the paths through the Internet should typically be the same for both prefixes. If the two paths differ, we can compare them and derive candidates of routers that handle the test-prefix differently from anchor-prefix. While not a trivial problem (paths may differ for legitimate reasons), we will see in section 2.2 how to derive a set of candidates that are most likely to be filtering the test-prefix. Unfortunately, in-probes can only detect filters that are located between the site running the traceroute and the test-box. We would thus need traceroutes originating from a large fraction of ASs to achieve an adequate filter detection quality. Even though desirable, we currently do not hope to find that many traceroute servers.

In order to detect filters at the edge of the Internet, we have to use a different technique: we search pingable addresses across the Internet and then send probe packets *from* the test-box towards those pingable addresses. We run one traceroute whose *source-address* is the test-IP and another where the *source-address* is the anchor-IP. We call this method an *out-probe* and discuss it in detail in section 2.3. The catch with out-probes is that the path the packet takes towards the pingable address is almost meaningless for our analysis. The path we are interested in is the path of the return-packet. Since Internet paths are often asymmetric, we do not know the path followed by the return-packet. Therefore, with out-probes we only learn about *usable connectivity* for the IP addresses we are pinging. By usable connectivity we mean bidirectional reachability between two given end-hosts. Thus, if we see the return-packet coming back, we know that this address can establish communication with the test-address space.

The challenge is that if the test-IP packet does not come back, we cannot conclude that the otherwise pingable IP is filtering the test-prefix. We can only conclude that we did not successfully establish usable connectivity. Even if bogon filtering is involved it might have been applied by any AS on the return path. Not seeing the test-IP packet come back might also be due to some probe packet drop on the return back, not bogon filtering. Sending several probes can be used to minimize the effect of such probe packet drops. Using our methodology we can derive a list of ASs which have usable connectivity and a list of ASs for which we did not succeed in establishing usable connectivity. Further analysis is needed to reveal where potential filters might be located.

This paper outline is as follows. In section 2 we give a more detailed description of our methodology together with a description of our initial experiments. In section 3 we present a preliminary validation to test the effectiveness of our methodology. We review related work in section 4 and finally conclude and discuss future work in section 5. Note that our methodology can be extended to investigate the more fundamental problem of the reachability of arbitrary prefixes.

2. METHODOLOGY AND RESULTS

In this section we combine a detailed presentation of our approach with some results of our experiments. We begin with our test address space and then discuss the two different kinds of probes we use to detect bogon filters: in-probes and out-probes.

2.1 Test-address space

We obtained four $/16$-prefixes from ARIN [10] of previously unallocated address space. We announced a subset as test-prefixes of varying prefix-length. See Figure 1 for a detailed listing. The prefixes were announced from mid-November 2006 until April 2007, from four different locations that volunteered to participate in our experiment: PSGNet in Seattle (USA), SpaceNet in Munich (Germany), CityLink in Wellington (New Zealand), and IIJ in Tokyo (Japan). Each test site announced one of the test-prefixes. The anchor-IP was the address of one machine inside the ISP that ran our experiments. The test-IP was configured as a secondary IP on the same box.

2.2 In-probes

In-probes are a straightforward kind of probing, as we run traceroutes to each test-prefix and to each anchor-prefix. By combining a set of in-probes from various locations in the Internet we can obtain a picture of which ASs are filtering the newly allocated prefix.

The primary components of the in-probes are 480 traceroute-servers and PlanetLab nodes [11]. Most of those traceroute servers are located in the US or in Europe, but overall we cover 56 countries. Note that we run traceroutes towards the test-box (both the test-prefix and the anchor-prefix).

In-probes serve two purposes: first, we want to understand which ASs cannot reach the test-address space. Second, we want to study why different paths are taken by traceroutes towards the test- and anchor-IP. We assume that since the test-prefix and the anchor-prefix are announced from the same router, the paths used should be similar. If the paths differ, then it might be caused by the test-prefix being “treated differently” along the path that is used to the anchor-prefix. Thus, we can derive a set of candidate ASs that potentially filter. A major limitation of in-probes is that we are only able to detect filters between a traceroute server and a test-prefix.

Cleaning in-probes

Before we can try to locate possible candidates, we need to determine the appropriate topological granularity for deriving our candidates. If we want to contact ISPs and ask them to update their filters, then an AS-granularity may seem appropriate. Unfortunately, we cannot assume that filtering is homogeneous across all routers within an AS, as the network administrator might have forgotten to update any number of links. In fact, our analysis confirmed that more than a quarter of all filtering ASs might have inconsistent fil-

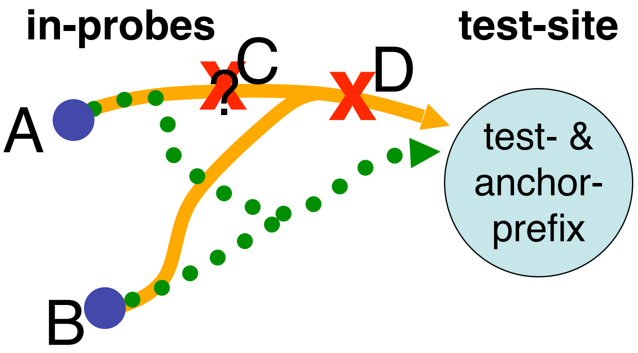

Figure 2: Example of the in-probes methodology. The test-box, which announces the test- and anchor-prefix, is depicted on the right. The test- and anchor-traceroutes are run from within AS *A* and AS *B* and are represented by the dotted and solid lines, respectively. Traceroutes go from the traceroute server towards the test-box. The crosses indicate ASs only observed by the anchor trace, and not occurring on the test path. Thus, they are potential places for filters.

ters. Therefore we work on a per "link" basis[1] to derive a set of candidates.

If we "observe" a test-probe packet going over a certain link, we assume that no filtering towards the test-address space occurred on that link. Thus, if filtering is happening, it has to be on the portion that was only observed on the anchor trace.

On the other hand, if the test- and anchor-traceroutes paths diverge, it does not necessarily mean that filtering has to be the cause. One obvious example is intra-domain load-balancing. We observe different IP addresses on the test- and anchor-traceroutes, with all of them belonging to the same AS. This phenomenon is quite common (we observe it in 68% of all our traceroutes), indicating that intra-domain load-balancing is widely used in today's Internet [12]. If we see the test- and anchor-traceroutes diverging for a few hops and then rejoining before "exiting the AS", then we assume load-balancing and do not include them in the set of candidates.

Deriving candidates

We then derive a set of candidates for each traceroute-pair. Figure 2 illustrates this process. Note that a single filter might be responsible for a set of candidates. Our list of candidates is likely to be larger than the actual list of filters. It is thus necessary to apply further techniques/heuristics to find out the location of the potential filters. The first and most straightforward heuristic is to remove a link from the set of candidates as soon as we have seen that link on any test-traceroute. The links observed on the path of a test-traceroute cannot be filtering the test-prefix. Then, we consider candidates that may be explained by several observations as more likely than those that can only be explained by a few observations. Hence, we try to find a minimal set of filters that can explain all our observations. This is based on the assumption that incorrectly configured filters in the Internet are isolated and the majority of ASs propagate correctly the test-prefix. Note further that we initially generate a lot of candidates, as we consider any link on the anchor path that is not on the test path. Therefore, we expect to start-off with a high number of false-positives. Consider the example shown in Figure 2. Assume we observe that the test path is not passing though AS *C* and AS *D*, while the anchor-trace does. In this case predicting filtering at AS *D* could explain the observations for AS *A* and AS *B*, while predicting a filter at AS *C* would only explain the observation at AS *A*. To explain AS *B*'s observation we need to assume a filter at AS *D* anyway. Thus, we conclude that filtering at AS *C* is less likely. Using this simple method, we can sort our potential filtering list by this kind of likelihood. ASs at the beginning of the list explain a lot of observations, while ASs at the end of the list are more likely to be false-positives. To achieve a better inference quality, we compute those numbers on a per link basis first and then look at the AS-wide statistics. The assumption behind that is that if a certain AS is always avoided in any test-prefix traceroutes, then it is again more likely to filter compared to an AS where we observe only partial filtering. Note that partial filtering does occur. We have talked to administrators and they have confirmed that they forgot to update certain routers. However, to be able to remove false-positives from our list, we alter the likelihood of an AS if we observe only partial filtering.

By taking into account highly likely filters we reduce the risk of false-positives, but also potentially incorrectly exclude filters that we consider as not being likely. As our methodology correlates traceroutes from different locations, our quality improves with the amount of intersecting information. The closer we get to the test-prefix, the more information is available and the better our inference is.

Experiment

We ran one in-probes experiment on January 22nd-23rd and another experiment one month later on February 21st-22nd. We derived 28 ASs as candidates the first time, while the second time we identified 34 ASs as candidates. We find 17 common ASs in both sets. 11 ASs are detected in the January experiment only. Those ASs could have changed their filtering between the two experiments. 17 ASs are detected in the February experiment only. Currently, we believe the difference in detected ASs is due mainly to the number of traceroute servers. In January we had only 311 servers, whereas in February we had 480 servers.

To further illustrate how important many traceroute servers are, we posted a request to the NANOG [13] mailing list to run traceroutes towards our test- and anchor-address space. We received 413 replies to our posting at the beginning of January. Adding those results to the traceroutes obtained from the public traceroute servers (January data-set) increases the number of potential filtering ASs (from 28) to 73. Now 18 ASs are in both sets, 50 in the January set only and 14 in the February set only. Comparing those ASs that have been in the February set (one time with NANOG - the other time without NANOG), we observe that 3 ASs have been moved to the "both" set category and two dropped from the "both" set category.

2.3 Out-probes

Out-probes are important to sample a large fraction of ASs. By sending out-probes from the test-boxes to pingable addresses, we obtain answers from many ASs, thus helping us to find out where usable connectivity exists and where it does not. While the detection quality with the in-probes is limited by the number of traceroute servers, the quality of the out-probes suffers because the asymmetry of Internet paths means that the return-path cannot be measured, and predicting the paths taken by packets is difficult [14, 15]. However, the return path is the one we are most interested in. Therefore, with out-probes we only learn about *usable connectivity*. We cannot detect whether the shortest path is being used or if the Border Gateway Protocol (BGP) is routing around an intermediate filter. On the other hand, usable connectivity is the primary

[1] Actually, a "link" is an IP to IP connection as observed in the traceroutes. This does not necessarily have to correspond to a physical link, since tunnels (e.g., MPLS) may hide intermediate hops.

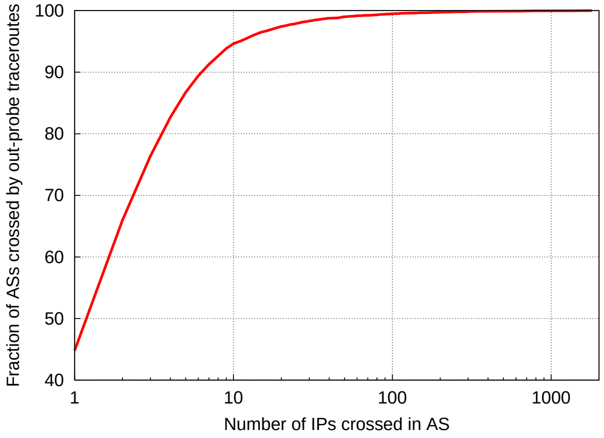

Figure 3: Cumulative fraction of ASs for which we see at least a given number of IP addresses in our traceroutes for out-probes.

concern of Internet operators.

Finding pingable-address-space

Before we can execute out-probes, we first have to find a suitable set of pingable IP addresses. The goal is to keep the required probing to a minimum while still being able to achieve a good coverage of the Internet, since ideally we want our newly allocated address to have a path to every AS. The bulk of those addresses are derived from CAIDA's skitter project [16], an Internet measurement project which actively probes the Internet to analyze its topology and performance. After obtaining the list from skitter we removed addresses which were not pingable.

We then augmented this skitter list by attempting pings on random address space in every prefix for which skitter did not have pingable addresses. We ended up with a list of 567,422 pingable IPv4 addresses which cover close to 20,000 ASs. The coverage of this list is still lacking pingable addresses in almost 5,000 ASs. This can be due to firewalls which block pings to large fractions of the network. If a ping is administratively prohibited by a network, we excluded this address space from any further study. Another reason why we might not be able to find a pingable address within some ASs is that not all ASs are "real ASs". They can be duplicates (bought by other companies or decommissioned), can be unused, special purpose ASs, etc. We are, therefore, confident that this list of pingable-address-space provides sufficient AS coverage for our experiment.

Experiment

To limit the burden on the Internet, we selected a subset of 46,569 IPs from the full list. This subset covered 18,574 ASs. Results in the remainder of this paper refer to this data-set. As our list of IPs contains dial-up IPs, we do not further probe if the anchor-IP is unreachable. About 5% of our IPs were not pingable anymore.

More than 80% of the pinged addresses returned success for pings originated by the anchor- and test-IPs. However, for approximately 10% of the pinged addresses, the test failed while the anchor succeeded. That means that at some point the traceroute towards the test-prefix stops before reaching the destination. This is an interesting result in itself, as we did not expected that the filtering-problem is so evident in today's Internet.

More than 5% of the pinged addresses had failures for both the test- and the anchor-prefix. On an AS-level granularity, 11,662 out of the 18,569 ASs seem to have working reachability, i.e. all test-traceroutes cross them. Only 910 ASs do not have properly working reachability, i.e. all test-traceroutes are stopped before reaching those ASs. Finally, the remaining 5,997 ASs show unclearly working reachability, with some test-traceroutes being stopped in the AS and others not.

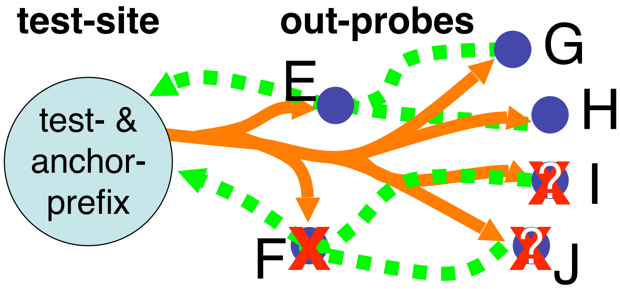

Figure 4: Example of the out-probes methodology. The test-box is depicted in the left. Out-probe traceroutes go from the test-box towards pingable IPs, located in AS E to AS J. The dashed lines indicate the return-path of the probes (which cannot be seen in our measurements). ASs E, G, and H, successfully return the probe packets. ASs F, I, and J do not answer to test-prefix probes. They are marked by crosses, as they are potential candidates for filters.

The large number of ASs with unclearly working reachabily are partly caused by the topological sampling of our traceroutes. The location of the sites from which we perform the out-probes implies uneven sampling of the IP-level topology of the Internet. Figure 3 illustrates this uneven number of IP addresses that are sampled in each AS. In 45% of the ASs we see in the out-probe traceroutes (among 8,032), only a single IP address is sampled. For less than 5% of those ASs do we sample more than 10 IP addresses. Most of the ASs crossed by our traceroutes sample again and again those large tier-1 ASs.

Recall, that ASs that does not have usable connectivity, do not necessarily have a filter configured on their routers. They can also be victims of filtering by an upstream provider. As a matter of fact, whether an AS has usable connectivity or not depends on the location where the prefix is announced. Depending on which path the return packet takes, it may or may not be filtered. Correlating information from different places is the only way to determine which AS is actually filtering.

Deriving candidates

As a first step, we build a list of candidates in a similar fashion to the in-probes: if we received a reply from a router we assume that this router interface does not filter and has usable connectivity. If the test-IP out-probe is dropped, while the anchor out-probe continues then we include the routers from the anchor path in our set of candidates. In addition, we annotate all candidates with a distance index from the point of failure on the test-traceroute (in observed router hops from the closest failure point). This reduces the number of false-positives at a later stage.

We can use such a "proximity" index, because the return path is changing at the AS closest to the failure point. Therefore, if the AS is not filtering itself, then at least this is the closest point in the topology one should continue debugging. If an AS further downstream has no connectivity at all, chances are high that those ASs only follow the same return path. Note that looking at "proximity" works because we are testing usable connectivity only, and again assuming that most of the ASs do not filter and also because our

AS coverage is high.

To illustrate our approach refer to Figure 4. In the example we assume that AS *F*, *I*, and *J* are not responding to our probes. Thus, all three ASs do not have any usable connectivity and may appreciate being informed that they have a reachability problem. But as illustrated, AS *I* and *J* may use *F* on the way towards the test-prefix and if *F* filters then all other ASs may be disconnected. Note that in this case, *F* is the only possible alternate path for *I* and *J*, because if several alternative paths exits, then BGP would route around *F* and AS *I* and *J* would have usable connectivity. However, if AS *F* is the only way to reach the considered prefix it will score higher on our “proximity” index and is therefore more likely to be filtering than ASs *I* and *J* which are further downstream. Using this methodology, we can reduce the number of false-positives or at least introduce an ordering of ASs that are most likely filtering.

Note this approach highly relies on our good AS coverage. If we never probe AS *F* during the measurements, it will be impossible to detect *F* as filtering candidate. To have very high confidence that AS *F* is indeed filtering we need to observe that all AS *F* neighbors either do not have working reachability because of AS *F* or that their paths go around AS *F*. Unless we would have in-probes from a very large fraction of the ASs in the Internet, it is unlikely to be able to have such high confidence in the inferred bogon filters. Another way to improve our confidence in the location of bogon filters would be to rely on a topological model such as [15] and predict the paths from each AS towards each other AS.

The above concepts are simple and straightforward, but there are a few details that need to be considered carefully. One of the issues is that it is not trivial to map router interface IPs to AS numbers [17]. For example consider the connection between two ASs: one AS is a provider, while the other a customer. The interconnecting link between those two ASs may now be established from the address space of the provider. Therefore, a probe that reaches the first router within the customer’s AS still uses a return address from the providers address space. As a filtering policy might now be configured on the customer router, but the return IP is still from the providers address space, it may appear as if the traceroute stops in the provider AS, while the customer AS is the one to blame. Our weighting of the “proximity” index takes particular care of this.

Most likely candidates

The next step is to evaluate the likelihood that some AS is filtering. Even though we are aware of the fact that ASs may partially filter, we believe that if we probe one AS several times at different places and it never replies, it is more likely that it does not have usable connectivity compared to an AS that partially responds (or responds to a specific probe-site only). Thus, it should score higher. This “filtering likelihood” is a mixture of the “proximity” and the fraction of links that are likely filters (likely links against total observed links). To compute such an index, we aggregate all observations from various traceroutes where this link has been seen, then normalize and weight this with our “proximity index”. Based on this score, we decide whether to include that AS in our report or not. An AS scores high if it is often quite close to the divergence point and if it has not replied to many probes. The key idea is similar to that of the in-probes: we try to find a minimal set of potential filters which can explain most of the observations.

After removing the obvious false-positives, there are 443 ASs remaining in our list of potential candidates. While a link is typically traversed only a few times (especially because we used our restricted IP set which only has a few IPs within each AS), there are some ASs where we have up to 220 different links inside a single AS traversed and marked as “likely”. Let us now look in more detail into this “likelihood”. Figure 5 shows a CDF of the “likelihood” for both links (triangles) and ASs (circles). The values of the likelihood itself are not important, as they depend on the number of measurements. We use this index mainly to provide an ordering of ASs that are likely to be filtering. All of the ASs on this list have at least one router that lacks connectivity to the test-prefix, but note that some ASs on this list may not be filtering. They may be victims of another AS’s filters. However, it is still useful to find these “victim” ASs, and alert them, so they may assist in debugging the problem before it becomes a problem for their customers.

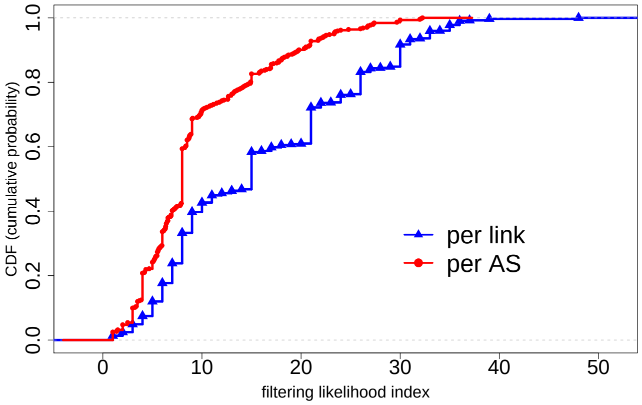

Figure 5: Cumulative Distribution Function (CDF) of our likelihood. The curve with triangles (lower curve, blue) shows the index we are computing on a per link basis, while the curve with the circles (upper curve, red) shows on a per AS basis (the mean is computed over all links within the particular AS).

3. INITIAL VALIDATION

In this section we present some initial results from our validation. In-probe validation is particularly hard, as we have to ask operators about the actual status of their filters. So far, we have only received six answers from operators – all confirming the filters that we have predicted. One case was of particular interest, as the network operator initially indicated not to filter. Later it turned out, that on most routers the filters actually had been updated, but a few routers had been forgotten. Thus operators were unaware of the fact that parts of the AS could not reach the newly allocated address space.

Regarding out-probe validation, recall that we derived a set of 443 candidate ASs that might have issues with their connectivity to the test-address space. Of those 443 out-probe candidate-ASs, we found 15 ASs which have public traceroute servers. Of those 15 ASs, 7 ASs filter the test-prefix themselves and are thus correctly identified. 5 ASs exhibit a different behavior: while not filtering themselves, they do not have usable connectivity (e.g., due to packet filters upstream). This means 12 out of 15 ASs have been correctly detected with non-working reachability. The remaining three ASs actually had usable connectivity, but unfortunately they showed up as filtering candidates in our characterization. However, as the validation data-set was taken at a different time, those ASs might have updated their filters in the meantime.

These positive early results are very promising, however due do the time-consuming nature of validation we will perform a more comprehensive validation in future work.

4. RELATED WORK

This work addresses one of the most fundamental services of the Internet: reachability. It is therefore not surprising to see numerous papers and presentations at conferences and workshops. Essentially, most those studies are interested in how "happy" [18] the packets are [19], this comprises also work such as [1–5, 7].

However, most research studies have so far concentrated on BGP [20]. Slow BGP convergence [21, 22], issues with policy routing [23], oscillations in BGP [24], and routing instabilities [25] are among the many problems encountered.

Researchers and practitioners either have to gather their own data or rely upon data collected by various sources, such as CAIDA's skitter [16], and large BGP data collection projects like RIPE Routing Information Service [26] or Oregon RouteViews [27].

5. CONCLUSION AND FUTURE WORK

In this paper we proposed a methodology to detect bogon filters that block the reachability of (newly) advertised prefixes. We proposed two types of probes, called in-probes and out-probes, that complement each other to detect bogon filters. We discussed the advantages and drawbacks of each type of probing, and described our system that leverages both to detect bogon filters.

Out-probes scan large fractions of the Internet, but can only detect usable connectivity. In-probes can detect intermediate filters, but are limited by the number of looking glasses that are available. However, with in-probes we can find those intermediate filters that are closer to the core of the network – which impact more ISPs. Note that the denser the connectivity the greater the need to find intermediate filters, as it is more likely that BGP can route around a filter – while at the edge, usable connectivity is typically more severely impacted, and even if alternate paths exists, they only affect a smaller fraction of users. Thus, while we strive to increase the number of traceroute servers, we can already capture the most significant contributors, with the existing number of traceroute servers.

As further work, we will include BGP data to improve our knowledge of ASs having reachability and those that may not have working reachability. We also plan to deploy our methodology in RIR's, to test not only new but also existing address space. This would also facilitate the dialog with network operators and thus our validation.

Acknowledgments

We are grateful to ARIN for giving us four previously unallocated prefixes. In addition we thank Gert Doering and SpaceNet, Andy Linton and Citylink, Matsuzaki Yoshinobu and IIJ for their efforts in announcing the test address space and doing all the test runs for us. We like to thank Belinda Chiera, Ashley Flavel, Bingjie Fu, Tim Griffin, Bamba Gueye and Wolfgang Mühlbauer for their valuable comments on earlier versions of this paper.

We are also grateful to ISPs who maintain a publicly available looking-glass and traceroute-server. This work was supported by the National Science Foundation (NSF) award ANI-0221435 and the Australian Research Council (ARC) grant DP0557066.

6. REFERENCES

[1] C. Labovitz and A. Ahuja, "Shining Light on Dark Internet Address Space," *NANOG 23*, 2001.

[2] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," in *15th USENIX Security Symp.*, 2006.

[3] P. Boothe, J. Hiebert, and R. Bush, "How Prevalent is Prefix Hijacking on the Internet?," *NANOG 36*, February 2006.

[4] Y. Chen, A. Bargteil, D. Bindel, R. Katz, and J. Kubiatowicz, "Quantifying network denial of service: A location service case study," in *ICICS '01: Proceedings of the Third International Conference on Information and Communications Security*, 2001.

[5] D. Wetherall, R. Mahajan, and T. Anderson, "Understanding BGP misconfigurations," in *Proc. ACM SIGCOMM*, 2002.

[6] "The Team Cymru Bogon Reference Page." `http://www.cymru.com/Bogons/`.

[7] N. Feamster, J. Jung, and H. Balakrishnan, "An empirical study of bogon route advertisements," *ACM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 63–70, 2005.

[8] G. Huston, "IPv4 Address Report," 2007. `http://www.potaroo.net/tools/ipv4/index.html`.

[9] Z. M. Mao, R. Bush, T. G. Griffin, and M. Roughan, "BGP Beacons," in *Proc. ACM IMC*, 2003.

[10] American Registry for Internet Numbers. `http://www.arin.net/`.

[11] "Planetlab." `http://www.planet-lab.org/`.

[12] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *Proceedings of the 6th ACM SIGCOMM on Internet measurement*, 2006.

[13] North American Network Operators Group. `http://nanog.org/`.

[14] Z. Mao, J. Rexford, J. Wang, and R. Katz, "Towards an accurate AS-level traceroute tool," in *Proc. ACM SIGCOMM*, 2003.

[15] W. Muehlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an AS-topology model that captures route diversity," in *Proc. ACM SIGCOMM*, 2006.

[16] Cooperative Association for Data Analysis, "Skitter." `http://www.caida.org/tools/measurement/skitter/`.

[17] Z. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz, "Scalable and accurate identification of AS-level forwarding paths," in *Proc. IEEE INFOCOM*, 2004.

[18] R. Bush, T. Griffin, Z. M. Mao, E. Purpus, and D. Stutzbach, "Happy Packets - Initial Results," 2004. NANOG 31.

[19] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Networking*, 1997.

[20] T. G. Griffin Interdomain routing links. `http://www.cl.cam.ac.uk/users/tgg22/interdomain/`.

[21] C. Labovitz, R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Trans. Networking*, 1998.

[22] Z. M. Mao, R. Govindan, G. Varghese, and R. Katz, "Route flap damping exacerbates Internet routing convergence," in *Proc. ACM SIGCOMM*, 2002.

[23] T. G. Griffin and G. Huston, "BGP Wedgies," 2005. RFC 4264.

[24] D. McPerson, V. Gill, D. Walton, and A. Retana, "Border Gateway Protocol (BGP) Persistent Route Oscillation Condition." RFC 3345.

[25] A. Feldmann, O. Maennel, M. Mao, A. Berger, and B. Maggs, "Locating Internet Routing Instabilities," in *Proc. SIGCOMM*, 2004.

[26] RIPE's Routing Information Service. `http://www.ripe.net/ris/`.

[27] University of Oregon RouteViews project. `http://www.routeviews.org/`.