Lessons Learned from using the RIPE Atlas Platform for Measurement Research

Vaibhav Bajpai
Jacobs University Bremen
v.bajpai@jacobs-university.de

Steffie Jacob Eravuchira
Jacobs University Bremen
s.eravuchira@jacobs-university.de

Jürgen Schönwälder
Jacobs University Bremen
j.schoenwaelder@jacobs-university.de

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

We reflect upon our experience in using the RIPE Atlas platform for measurement-based research. We show how in addition to credits, control checks using rate limits are in place to ensure that the platform does not get overloaded with measurements. We show how the Autonomous System (AS)-based distribution of RIPE Atlas probes is heavily skewed which limits possibilities of measurements sourced from a specific origin-AS. We discuss the significance of probe calibration and how we leverage it to identify load issues in older hardware versions (38.6% overall as of Sep 2014) of probes. We show how performance measurement platforms (such as RIPE Atlas, SamKnows, BISmark and Dasu) can benefit from each other by demonstrating two example use-cases. We also open discussion on how RIPE Atlas deployment can be made more useful by relaying more probe metadata information back to the scientific community and by strategically deploying probes to reduce the inherent sampling bias embedded in probe-based measurement platforms.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*

General Terms

Measurement, Performance

Keywords

Access Networks, RIPE Atlas

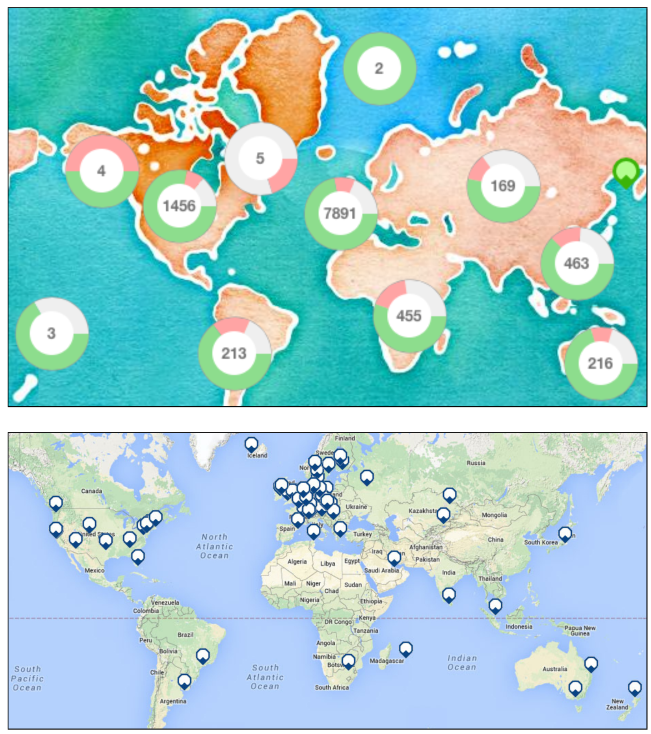


Figure 1: Coverage of the RIPE Atlas measurement platform as of Feb 2015. Around 12.8K probes (top) and 109 anchors (bottom) have been deployed in total: atlas.ripe.net/results/maps. The green, red and grey areas (above) represent connected, disconnected and abandoned probes respectively.

INTRODUCTION

RIPE Atlas [1] has deployed around 12.8K dedicated hardware probes and around 109 anchors (as of Feb 2015) all around the globe as shown in Fig. 1. Probes perform active measurements to ascertain network connectivity and reachability of the global Internet, while anchors are dedicated servers that can act as sources and sinks of measurement traffic. RIPE Atlas periodically schedules measurements using a batch of several hundred probes against anchors to measure region-based connectivity and reachability. A majority of these probes are running measurements either from the core or from within access networks. A discernible number of probes are also hosted by volunteers within their home networks. Table 1 provides a list of built-in measurements performed by probes by default. All hosted probes are made publicly available for measurement research. These probes in addition to built-in measurements can also run

MEASUREMENT	TARGET
ping, ping6	first hop, second hop, ns.ripe.net, *.root-servers.net, *.atlas.ripe.net
traceroute, traceroute6	*.root-servers.net, *.atlas.ripe.net, labs.ripe.net
dns, dns6	*.root-servers.net: TCP (SOA), UDP (SOA, version.bind, hostname.bind, id.server, version.server)
sslcert, sslcert6	www.ripe.net, atlas.ripe.net
http, http6	www.ripe.net/favicon.ico, ip-echo.ripe.net

Table 1: A list of built-in measurements performed by probes by default as of Feb 2015. (*) in the target fields indicate multiple servers within the domain.

User Defined Measurement (UDM)s. A UDM allows any user registered (around 19K as of Feb 2015) on RIPE Atlas to provision measurements supported by the platform (see Table 1) on probes with tailor-made measurement parameters. A registered user spends credits by provisioning a UDM on probes. Credits can be gathered by either hosting a probe (for no purchase cost) or an anchor (for a purchase cost). RIPE Atlas also released (on Feb 2013) a public API that allows one to programmatically provision UDMs. Using these public APIs and credits gathered by hosting probes for multiple years, we were able to provision UDMs on a large sample of probes. We share our experiences and lessons learned from using the RIPE Atlas platform for measurement research.

#1: RATE LIMITS

RIPE Atlas uses credits as a virtual currency to regulate UDM usage within the platform. Millions of credits can be accumulated by hosting probes for multiple years. Given the credit consumption of individual built-in measurement is fairly low (see Table 2), it provides an impression that given ample credits, large number of measurements can be provisioned on the platform. However, the platform also imposes four daily rate limit thresholds on each user account: a) No more than 100 simultaneous measurements, b) No more than 500 probes/measurement, c) No more than 1M credits may be used each day and d) No more than 10 ongoing and 10 one-off measurements of the same type against same target at any time. These rate limits, although documented[1], may not be well-known to the research community. These limits may coerse one to design experiments that span multiple days. As such a request to lift these limits can be made by proposing and gathering support for the measurement study on the `atlas` mailing list.

MEASUREMENT	CREDITS/RESULT ↓
traceroute, traceroute6	30
dns, dns6 (TCP)	20
dns, dns6 (UDP)	10
sslcert, sslcert6	10
ping, ping6	3

Table 2: Credit cost consumption of built-in measurements as of Feb 2015: atlas.ripe.net/docs/credits. These are credits consumed by measurements using default parameters. These costs can increase (or decrease) if default measurement parameters are tweaked.

#2: HEAVY-TAILED PROBE DISTRIBUTION

The geographical distribution of the probes (see Fig. 1) provides a decent high-level overview of the coverage of the platform. Although the network coverage map[2] provides a facility to filter probes by AS Number (ASN), the overall distribution of probes across ASes and density of probes

[1] *atlas.ripe.net/docs/udm*
[2] *atlas.ripe.net/results/maps/network-coverage*

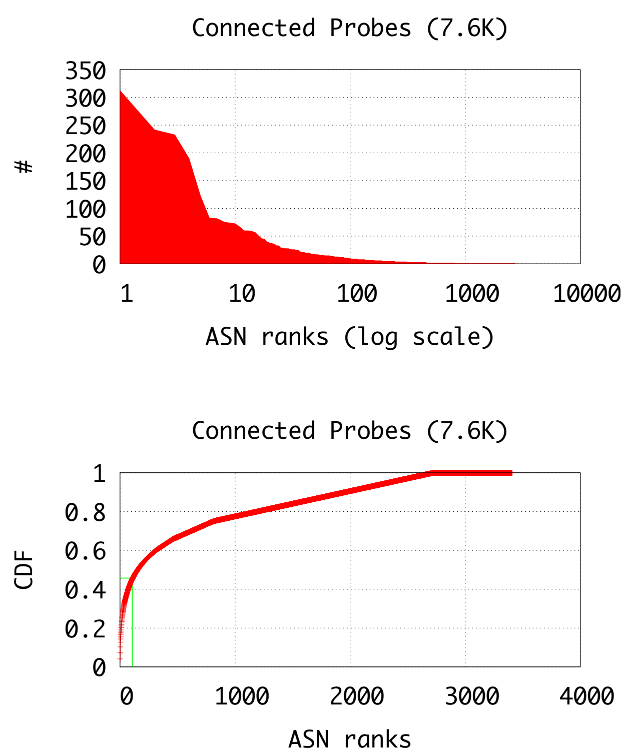


Figure 2: Distribution of a subset of connected and non-anchored probes (7672) sorted by AS rank as of Feb 2015. ASes are ranked by number of probes. 44.59% (3421) of connected probes fall within AS ranks <= 101. Rest of ASes contain < 10 probes. The dataset is available at: goo.gl/kmIydP

within each AS is not well known. Measurements sourced from a specific AS require high probe density to mantain a representative sample, while measurements destined towards a specific AS require diversity of network origins. As such, we performed an experiment to better understand the AS-based distribution of these probes.

Clustering probes by ASN

We use the RIPE Atlas probe API[3] to capture a list of connected probes in order to later cluster them by their origin AS. The API, however, does not reveal the ASN for all probes. For instance, some probes (2037, 15.9% of all registered probes as of Feb 2015) did not expose either their public IP or their origin-AS. We grabbed the probe IDs of these probes and provisioned a one-off (measurement that runs only once) `traceroute` measurement. The measurement was scheduled only on a few probes (43 out of 2037) while the rest were deemed disconnected by the scheduler. We identified the origin AS of these probes, and pruned the rest of the disconnected probes out of the list. We also used the mapping in Fig. 4 (described later in the paper) to rule out anchors (109 as of Feb 2015). Going forward, we use the term probe to refer to the connected and non-anchored subset (7672) of all RIPE Atlas probes (12790).

Ranking ASNs by number of probes

We ranked ASNs by sorting them by the number of deployed probes. Table 3 provides a list of top 10 ASes containing the highest number of probes. For instance, Comcast (AS7922) has 313 (out of 7672) probes which contributes to 4% of all probes. The cumulative probes within top 10 AS ranks contribute to 18% of all probes as of Feb 2015. Fig. 2 shows the long-tail probe distribution sorted by AS ranks. A corresponding CDF of this long-tail, shows how probes deployed within AS ranks > 101 have less than even 10

[3] *atlas.ripe.net/api/v1/probe*

AS RANK	AS (ASN)	#(PROBES) ↓
01	COMCAST (AS7922)	313
02	PROXAD (AS12322)	242
03	LGI-UPC (AS6830)	233
04	DTAG (AS3320)	190
05	ORANGE (AS3215)	124
06	ZIGGO (AS9143)	83
07	XS4ALL (AS3265)	82
08	BT (AS2856)	76
09	UUNET (AS701)	74
10	VIRGINMEDIA (AS5089)	73

Table 3: Distribution of a subset of connected and non-anchored probes (7672) sorted by AS rank as of Feb 2015. ASes are ranked by number of connected probes. The entire dataset is available at: goo.gl/kmIydP

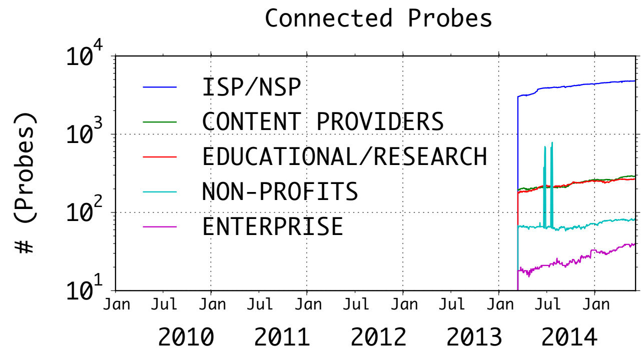


Figure 3: Evolution of probes by network type as mapped by PeeringDB. The plot is generated using the probe archive API: goo.gl/pMHs9Q which provides probe metadata since March 2014. Majority of probes are deployed behind service provider networks.

probes. To bring numbers into perspective, if we were to consider 10+ probes as a representative sample within each AS, the number of probes falling within AS ranks <= 101 would contribute 44.59% (3421 out of 7672) which is less than half of the entire population of probes.

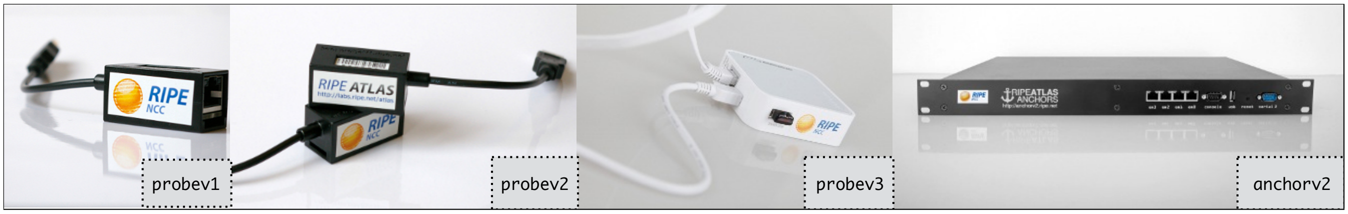
Clustering ASNs by network type

Using PeeringDB, we further mapped ASes hosting the connected probes (7672 as of Feb 2015) by their network type information. PeeringDB[4] is a database holding peering information of participating networks. Aemen Lodhi *et al.* in [4] show how the information maintained within this database is reasonably representative of network operator peering and is also up-to-date. Fig. 3 shows the evolution of probes by network type over a year. Few spikes occur in the non-profit network type due to a large fraction of probes (with a series of consecutive probe IDs) coming online for a day (or few days) from within the RIPE NCC network. Not all ASes hosting connected probes could be mapped to a network type due to missing AS information (encompassing 33.5% probes as of Feb 2015) in the PeeringDB database. Nevertheless, this mapping provides an indication on which type of networks hold major portion of connected probes. As such, RIPE Atlas is a potential platform for performing active measurements from within service provider networks.

Skewed distribution of probes

The RIPE Atlas platform ostensibly appears to have a large number of deployed (12.8K registered as of Feb 2015) probes. However, it turns out that the number of probes available for a measurement study sourced from a specific origin-AS is small. This is due to the skewed distribution of probes which considerably reduces the density of probes behind each AS. In all fairness, the platform was initially designed to measure connectivity and reachability. As such, there has been an inclination to deploy probes to increase coverage (than density) by biasing distribution in favor of under-served ASNs. As a result, the platform is more suitable for performing measurements targeted to a specific destination as it provides diversity of network origins.

[4] *peeringdb.com*



PROBE ID	HARDWARE VERSION	HARDWARE	RAM	WEBPAGE
[1, 1521]	probev1	Lantronix XPort Pro	8 MB	probev1.ripe.net
(2000, 5000)	probev2	Lantronix XPort Pro	16 MB	probev2.ripe.net
(10000, $+\infty$)	probev3	TP-Link TL-MR3020	32 MB	probev3.ripe.net
(6000, 6018)	anchorv1	Dell PowerEdge	-	-
[6018, 7000)	anchorv2	Soekris Net6501-70	-	anchorv2.ripe.net

Figure 4: Family of hardware probes deployed by the RIPE Atlas platform as of Sep 2014. v3 probes are more capable than v1/v2 probes in hardware specifications. Anchors are dedicated servers that act as sources and sinks of measurement traffic. The probeID can be used to identify the hardware version. Firmwares are kept in sync across hardware versions. The probe ID to hardware mappings were generated from: goo.gl/qABo1w.

#3: LOAD ISSUES IN OLDER PROBES

RIPE Atlas currently runs measurements from three (v1, v2, v3) different probe hardware versions as shown in Fig. 4. In order to have the same capabilities available, the platform tries to keep firmware versions in sync across hardware versions. In our pursuit to understand whether running the same firmware release on all hardware versions makes any impact on measurement results, we performed firmware and hardware calibration of the probes. We show how such a calibration allowed us to identify load issues in older (v1 and v2) hardware versions of the probes.

```
{
"prb_id": 10305,
"type": "traceroute"
"fw": 4660,
...
}
```

Listing 1: A snippet of a `traceroute` measurement result from a probe (as of Sep 2014).

Probe calibration

RFC 3432 [5] defines calibration as the process of determining the systematic (constant bias in measured values) and random error generated by the instruments themselves in as much detail as possible. In this work we focus on calibration to adjudicate the systematic error in probes.

Firmware variants: The firmware release running on the probes is one such parameter that can create a systematic error in measured values. Each firmware release brings with it, codebase changes either as bug fixes or as new feature updates that can have an impact on measurement results. Fig. 5 for instance shows that RIPE Atlas firmware release cycles have become more frequent since 2013. As a result, chances of a measurement campaign crossing these firmware release boundaries have also become more pertinent. Even if a measurement compaign does not cross a firmware boundary, it's generally useful to be able to track back to the firmware codebase in situations where an unexpected measurement result is observed. In order to allow firmware calibration, the platform inherently tags (see Listing 1) the firmware release for each measurement result to allow one to later trace back to the source code.

Hardware variants: While RIPE Atlas attaches each UDM with the firmware version of the probe, hardware versions are not tagged and therefore not reported. The platform

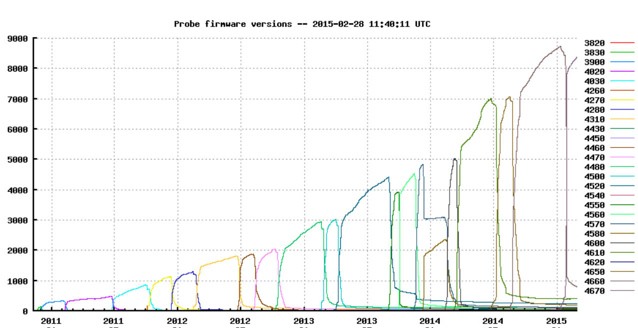


Figure 5: Firmware release cycles since 2011 (as of Feb 2015): atlas.ripe.net/results/graphs

runs measurements from three probe (v1, v2 and v3) hardware versions. v1 and v2 probes are made of a custom hardware built around a Lantronix XPort Pro module, while v3 probes are off-the-shelf TP-Link wireless routers flashed with OpenWrt[5]. As a result, v3 probes are more capable (in terms of hardware specifications) than older v1 and v2 probes. In addition, measurements can also be provisioned on anchors (dedicated servers), further adding to the hardware variability. Therefore, we asked on the `atlas` mailing list and identified how the probe ID itself can reveal hardware versions of the probes. Fig. 4 describes the mapping of a probe ID to its hardware version.

Segregating measurements by hardware

In our pursuit to study whether different hardware versions have effects on measurement results, we performed an experiment on probes deployed in a residential network. We specifically used probes that were directly wired behind the home gateway. This helps ensure that our measurements do not get skewed by probes that cross any wireless links (not wireless bridges) within the home network. The probe itself cannot associate to a wireless access point because RIPE Atlas has stripped all wireless capabilities out of the firmware. In order to filter for this sample, we searched for probes whose first-hop was in a private IPv4 address space [6], but their second hop was in a public IPv4 address space. Using this sample of residential probes, we provisioned IPv4 `traceroute` measurements once every 15 minutes for a day. In order to study effects of hardware (see Fig. 4), we further separated measurement results by each hardware version.

Fig. 6 shows the latency measured to the first hop (home gateway) observed over a day from all three (v1, v2, v3)

[5] *openwrt.org*

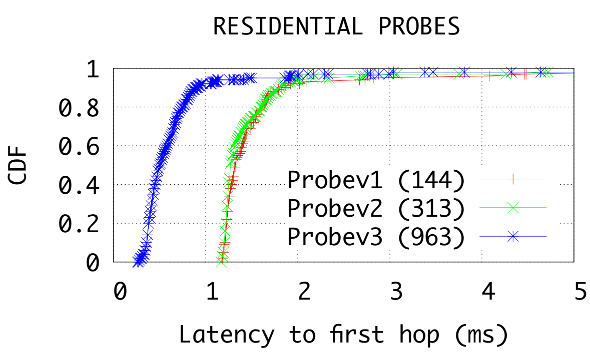


Figure 6: CDF of latencies to first hop observed over a day-long traceroute measurement for v1, v2 and v3 hardware probes wired behind residential gateways as of Sep 2014. v3 probes (in blue) show expected <1ms latencies, while v1 probes (red) and v2 probes (green) show higher latencies to the home gateway. Probes were running firmware version: 4650 and 4660. The x-axis of the plot is cut off at 5ms. The entire raw dataset is publicly released at: goo.gl/NRPxb7.

```
static struct trtbase *traceroute_base_new (
  struct event_base *event_base
) {
...
event_assign(&base->event4, base->event_base,
  base->v4icmp_rcv, EV_READ | EV_PERSIST,
  ready_callback4, base);
}

static void ready_callback4 (
  int __attribute((unused)) unused,
  const short __attribute((unused)) event,
  void *s
) {
...
struct timeval now;
gettimeofday(&now, NULL);
ms=(now.tv_sec-state->xmit_time.tv_sec)*1000 +
   (now.tv_usec-state->xmit_time.tv_usec)/1e3;
}
```

Listing 2: A traceroute code snippet from 4570 running on v1/v2 probes as of November 2013. The source code is available at: atlas.ripe.net/get-involved/source-code

probe hardware versions. A probe directly connected to the residential gateway should not show first-hop latencies of more than 1ms. We see how a significant number of v3 probes show such a behavior, however almost all v1/v2 probes show higher first-hop latencies.

Since the platform tags the firmware release in each measurement result, we were able to trace back to the source code of the firmware running these measurements to better understand the source of the issue. The source code reveals how the entire measurement framework is built around busybox [11]. Each measurement test has been adapted to run in an event-driven manner using `libevent`. As a consequence, whenever a UDM request is initiated, tests that run the measurement are not spawned as new processes, but are invoked as separate function calls. There is a single process that handles a single event loop for all incoming measurement requests. The source code has been designed in this way to help circumvent the unavailability of a Memory Management Unit (MMU) in v1 and v2 probes and to avoid allocating memory for multiple stacks (such as one would do in a multithreaded implementation). The latest family of v3 probes do have a MMU and significantly more memory (see Fig. 4), but in order to keep firmwares in synchronization across hardware versions, this implementation strategy has also been carried forward in v3 probes.

Listing 2 shows a sample snippet from the `traceroute` source code of the firmware release running these measurements. The function `traceroute_base_new(...)` is invoked when a `traceroute` measurement is requested, where it registers a callback. As can be seen, the Round-Trip Time (RTT) time stamping of a response to an Internet Control Message Protocol (ICMP) query is performed in the event callback function `ready_callback4(...)` in user space. This means that if a probe is *loaded* with multiple measurements, the user-space time stamping will be delayed. These delays will be more pronounced on constrained hardware such as v1/v2 probes (3961 of 10260 registered probes as of Sep 2014). As such v1/v2 probes (38.6% as of Sep

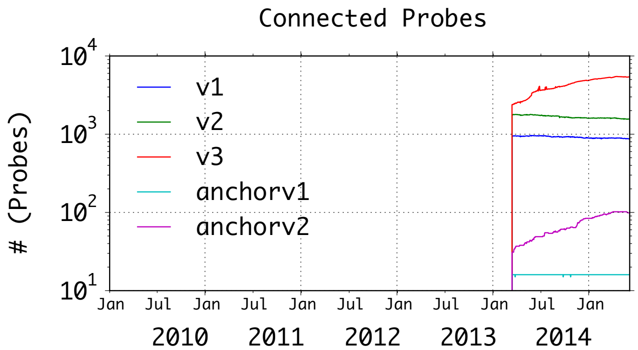


Figure 7: Evolution of probes by hardware family using probe ID to hardware mapping described in Fig. 4. The plot is generated using the probe archive API: goo.gl/pMHs9Q which provides probe metadata since March 2014. The contribution factor of older hardware version of the probes is fading away.

2014) experience load issues whenever a number of UDMs are provisioned on them.

RIPE Atlas has recently acknowledged our findings[6]. They confirm how adding more code has not had much effect on reducing load issues in v1/v2 probes. They add, in situations where measured first-hop latencies get upto 6ms (also witnessed by us) is when these slower probes are busy performing an Address Resolution Protocol (ARP) request to update their cache entries. In all fairness, the contribution factor of these older hardware versions will slowly fade away (31% as of Feb 2015) as shown in Fig. 7, since the RIPE Atlas platform now dispatches only v3 probes for new volunteers. RIPE Atlas also recently (starting October 2014) introduced the capability to filter probes by their hardware version using tags (such as `system-v1` *et al.*). Using this feature, older versions of the probes can be filtered out when running performance-based (such as latency) measurements. In hindsight, even though v3 probes reduce the impact of user-space timestamping, the platform would also benefit from using kernel-based timestamping using the `SO_TIMESTAMP` socket option on the packets's reception path.

#4: CROSS-TRAFFIC AGNOSTIC PROBES

The RIPE Atlas platform (unlike other performance measurement platforms) does not take cross-traffic detection into account when performing measurements. Broadband Internet Service Benchmark (BISmark)[7] [8] probes, for instance, read byte counters from `/proc/net/dev` to record passive traffic volume. SamKnows[8] [9, 10] probes use a threshold service to monitor both inbound/outbound traffic on the probe's Wide Area Network (WAN) interface to detect wired cross-traffic. They also record traffic volume exchanged on the user's wireless Service Set Identifier (SSID) to detect wireless cross-traffic. The test runs are delayed once cross-traffic is detected and re-tried with a back-off timer. The entire test cycle is abandoned if the threshold is crossed more than five times in a row. Dasu probes [7] follow a similar approach, but rely on Universal Plug and Play (UPnP) to query traffic counters on the WAN interface of the residential gateway. SamKnows probes also utilize this out-of-band technique in situations where hosts are not wired behind the probe, but are directly connected to the home gateway.

We performed an experiment to compare the behavior of RIPE Atlas and SamKnows probes in presence of cross-traffic. We requested `traceroute` measurements from both RIPE Atlas (96 samples) and SamKnows probes (84 samples). Fig. 8 shows the distribution of the number of measurements performed by probes within each platform. It can be seen how 20% of the SamKnows probes provided less than 10% samples due to cross-traffic detection during multiple measurement runs, while 90% of the RIPE Atlas probes being agnostic to cross-traffic contributed to more than 90% of all measurement samples.

In all fairness, the RIPE Atlas platform does not perform cross-traffic detection out of principle. The probes strictly perform active measurements only and no form of passive monitoring (even for cross-traffic detection) is performed in practise. Therefore, studies using RIPE Atlas for

[6] *ripe68.ripe.net/archives/video/240*
[7] *projectbismark.net*
[8] *samknows.com*

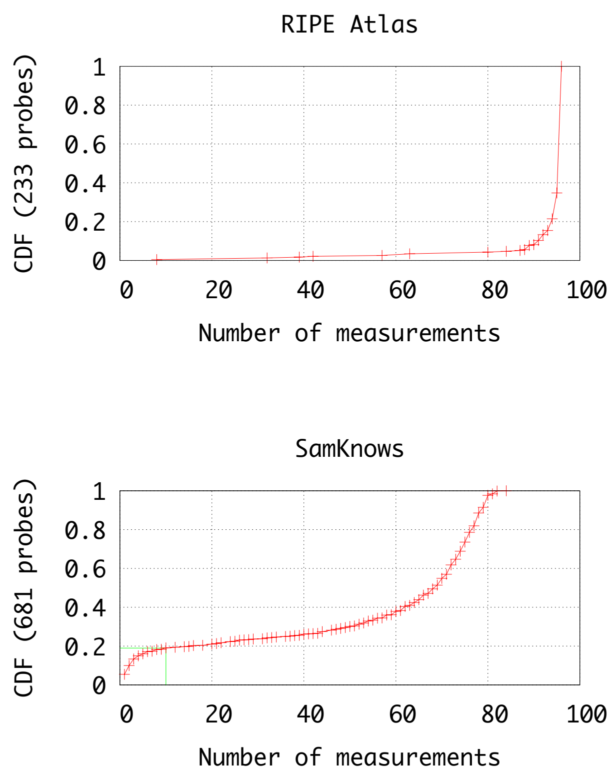


Figure 8: CDF of number of measurements performed by probes. Around 90% of RIPE Atlas probes (being agnostic to cross-traffic) performed most of the provisioned measurements (more than 90 out of 96) as of Nov 2013. 20% of SamKnows probes (due to cross-traffic detection) performed only few of the provisioned measurements (less than 10 out of 84) as of Feb 2014.

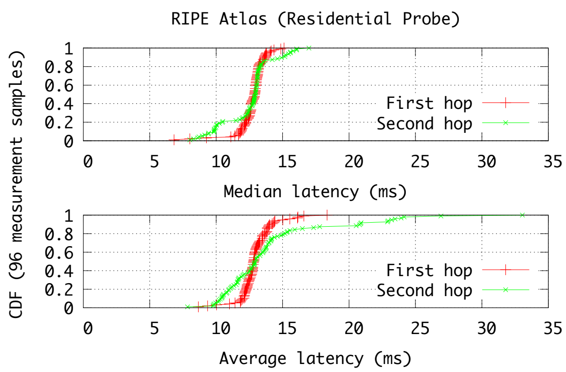


Figure 9: CDF of latency to the first and second hop from a RIPE Atlas probe as of November 2013. The effect of averaging (below) three queries becomes more pronounced over the second hop when compared to median (above) of three queries. A difference between the averaged latency to second (in green) and first (in red) hop will now lead to negative values.

performance-based measurements should be aware that their measurements can possibly run in presence of cross-traffic.

#5: PER-HOP LATENCY AGGREGATIONS

RIPE Atlas probes use `evtraceroute`, a modified version of `traceroute` available in `busybox`. SamKnows probes on the other hand use `mtr`[9]. Whenever a `traceroute` measurement request is initiated on these platforms; three ICMP queries are dispatched per hop by default. While RIPE Atlas probes separately report latencies measured by each ICMP query; SamKnows probes average latencies from multiple ICMP queries over each hop.

We investigated effects of averaging latencies from multiple ICMP queries over a single hop. Fig. 9 shows how averaging latencies over each hop can significantly vary observed results. It can be seen how effects of averaging latencies becomes more pronounced towards the second hop as the latency distribution starts to become more skewed. A mere difference between the averaged second and first hop latencies will now lead to negative results. The aggregation (if necessary) must be done by taking a median of latencies that can better tolerate outliers. We (in collaboration with SamKnows) have updated the `mtr` implementation used by SamKnows to expose each query result separately without any aggregation.

#6: METADATA IS (CHANGING) DATA

Proper interpretation of measurement results requires metadata to be treated as important as raw measurement data. RIPE Atlas does reveal the geographical location and origin AS of the probe deployment as a metadata entry. However, more metadata is needed to be able to perform specific measurement studies. For instance, the type of network where the probe is deployed, the connection speed and the WAN type of the upstream connection are details that facilitate data analysis. In fact, it requires tremendous manual effort to infer these connection properties through active measurements. Even though possible, these inferences are only heuristics and do not guarantee correct metadata, which only the probe host can accurately supply during the initial registration process. In fact, the current registration procedure[10] does allow a host to provide some details on its connection profile. However, this information is not currently relayed back through the public API. The platform should expose this metadata information alongwith the metadata history so that one can track changes. This would make it easier to isolate probes for a specific measurement study.

RIPE Atlas currently prefers not to report broadband subscription information because of two reasons: a) not all probe hosts record it correctly and b) subscription information tends to stale over time and it takes a major effort to track record changes in subscription switches.

#7: INHERENT SAMPLING BIAS

The deployment of RIPE Atlas probes is biased towards technically-inclined volunteers. A majority of volunteers are network enthusiasts or tend to have close degrees of connections with one. Volunteers hosting such probes tend to have a more complex home network than usual. Since the probe metadata available is currently bleak; the amount of this bias cannot be quantified. Nevertheless, it is important to state that measurements from such vantage points cannot be generalised, particularly in situations where the sample population is low. BISmark [3] and Dasu [7] measurement platforms acknowledge such a biasing limitation in their recent measurement research work.

CONCLUSION

The RIPE Atlas measurement platform was initially designed to measure connectivity and reachability of the Internet. With the deployment of 12.8K probes, the trend is shifting more towards using this platform for performance-based measurements. In this work, we identified how from among three hardware versions of probes, v3 probes are more suitable for performance (such as latency) measurements than older versions (38.6% of all probes as of Sep 2014) that suffer load issues. Studies using RIPE Atlas to measure latencies therefore need to take the hardware version into account because older versions can produce less accurate results. Given the platform dispatches only v3 probes for new hosts, the contribution impact of older versions (31% as of Feb 2015) is slowly fading away. Although older versions are still useful for measuring reachability and even latency if high precision accuracy is not the desired goal. We also demonstrated how measurement-based studies that require higher coverage of network origins would benefit more from the platform than those that require high probe density within each network. We also discussed two use-cases where measurement platforms can benefit from one another: a) SamKnows probes are cross-traffic aware (unlike RIPE Atlas probes) and b) RIPE Atlas probes do not aggregate latencies over each `traceroute` hop (unlike SamKnows probes) both of which when disabled can heavily impact measurement results.

[9] *bitwizard.nl/mtr*

[10] *atlas.ripe.net/register*

ACKNOWLEDGEMENTS

We would like to thank Daniel Karrenberg (RIPE NCC) for reviewing our manuscripts, Philip Homburg (RIPE NCC) for confirming the load issue, Robert Kisteleki (RIPE NCC) for reviewing our manuscripts and supporting us on the RIPE Atlas mailing list and Vesna Manojlovic (RIPE NCC) for lifting off daily rate limits on our accounts. We also thank Dagstuhl [2] for hosting a seminar where we discussed how to avoid pitfalls and worst practises with measurement research. This work was supported by the European Community's Seventh Framework Programme in (FP9/2007-2013) Grant No. 317647 (Leone). This work was also partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

REFERENCES

[1] V. Bajpai and J. Schönwälder. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *Communications Surveys Tutorials, IEEE*, PP(99):1–1, 2015.

[2] P. Eardley, M. Mellia, J. Ott, J. Schönwälder, and H. Schulzrinne. Global Measurement Framework (Dagstuhl Seminar 13472). *Dagstuhl Reports*, 3(11):144–153, 2014.

[3] S. Grover, M. S. Park, S. Sundaresan, S. Burnett, H. Kim, B. Ravi, and N. Feamster. Peeking Behind the NAT: An Empirical Study of Home Networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 377–390, New York, NY, USA, 2013. ACM.

[4] A. Lodhi, N. Larson, A. Dhamdhere, C. Dovrolis, and k. claffy. Using peeringDB to Understand the Peering Ecosystem. *SIGCOMM Comput. Commun. Rev.*, 44(2):20–27, Apr. 2014.

[5] V. Raisanen, G. Grotefeld, and A. Morton. Network performance measurement with periodic streams. RFC 3432 (Proposed Standard), Nov. 2002.

[6] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), Feb. 1996. Updated by RFC 6761.

[7] M. Sanchez, J. Otto, Z. Bischof, D. Choffnes, F. Bustamante, B. Krishnamurthy, and W. Willinger. A Measurement Experimentation Platform at the Internet's Edge. *Networking, IEEE/ACM Transactions on*, PP(99):1–1, 2014.

[8] S. Sundaresan, S. Burnett, N. Feamster, and W. de Donato. BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks. In *USENIX Annual Technical Conference (USENIX ATC 14)*, pages 383–394, Philadelphia, PA, June 2014. USENIX Association.

[9] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband Internet Performance: A View from the Gateway. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 134–145, New York, NY, USA, 2011. ACM.

[10] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Measuring home broadband performance. *Commun. ACM*, 55(11):100–109, Nov. 2012.

[11] N. Wells. BusyBox: A Swiss Army Knife for Linux. *Linux J.*, 2000(78es), Oct. 2000.