Quantifying interference between measurements on the RIPE Atlas platform

Thomas Holterbach  
ETH Zürich

Cristel Pelsser  
Internet Initiative Japan

Randy Bush  
Internet Initiative Japan

Laurent Vanbever  
ETH Zürich

ABSTRACT

Public measurement platforms composed of low-end hardware devices such as RIPE Atlas have gained significant traction in the research community. In addition to be widely used, these platforms also provide their users with an ever growing set of measurement tools. To be scalable though, they allow for concurrent measurements. This paper answers a fundamental question for any platform user: Do measurements launched by others impact my results? If so, what can I do about it?

We measured the impact of multiple users running experiments in parallel on the RIPE Atlas platform. We found that overlapping measurements do interfere with each other in at least two ways. First, we show that measurements performed from and towards the platform can significantly increase timings reported by the probe. We found that increasing hardware CPU greatly helped in limiting interference on the measured timings. Second, we show that measurement campaigns can end up completely out-of-synch (by up to one hour), due to concurrent loads. Unlike for precision, better hardware does not help.

1. INTRODUCTION

Public measurement platforms composed of many low-end devices or probes, such as RIPE Atlas [1], are increasingly used by researchers and network operators. In addition to measure network performance [2, 3, 4], these platforms are now used to map the Internet [5], detect routing attacks [6], routing anomalies [7], censorship [8, 9].

To scale and be practical, measurement platforms schedule measurements in parallel, without providing feedback to the user. When put together with the limited hardware and software capabilities, this raises the question of measurement interferences. What is the impact of an increased load on the precision of measurements performed? Do the measurements performed by one participant impact the results obtained by others? If so, by how much? Can this have an impact on previous research results? This paper answers these questions empirically for the RIPE Atlas platform.

By measuring the interference between our own measurements, we show that measurements do indeed interfere with each other, sometimes significantly.

We show that user-induced interferences can impact two aspects of measurements: precision and synchrony. First, the precision of delay measurements (e.g., using ping) performed either from or towards probes can be significantly impacted when other measurements are launched from or toward them (§4). Second, we show that user-induced interferences can heavily desynchronize experiments performed on multiple probes, even when launched at the same time (§5).

Our key findings are as follows:

- The precision of measurements performed from and towards the probe are impacted when other measurements use the probe at the same time. On older hardware, delays increase by more than 1 ms in the median case and by more than 7 ms for the 95th percentile (Table 2).
- Measurements are very quickly desynchronized when other measurements are run in parallel. Under heavy load, completion time may be delayed by close to 1 hour (Figure 7).
- Upgrading the probe hardware seriously improves precision levels, but does not help ensuring good synchronization levels (§5).
- Previous research results, as well as the RIPE Atlas historic dataset, may have been affected by interfering measurements. We also highlight two techniques to mitigate interferences in the future (§6).

Overall, our results show that measurement interferences should be systematically taken into account when analyzing results from public platforms. To ensure reproducibility, all our measurement and analysis tools are available at [9].

2. THE RIPE ATLAS PLATFORM

We now describe how Atlas works and highlight its increasing popularity among the academic community.

As of April 2015, RIPE Atlas is composed of over 6,700 public probes scattered in 197 countries. Three versions of the probes exist, differing only
by their hardware. Version 1 and version 2 are identical except for the amount of RAM they have. Both are Lantronix XPort Pro with a 167MHz CPU, 8MB or 16MB of RAM, respectively and a 16MB flash. The version 3 probe is a revamped TP-Link TL-MR3020 router with a 400MHz CPU, 32MB of RAM and a 4MB NAND. The v3 probes are therefore more powerful.

Figure 1 depicts the evolution of the number of public probes per version since the platform inception. The number of v3 probes increased rapidly after they started to be distributed in 2013. While the number of v1 and v2 probes decreases, they remain non-negligible, accounting for 28.2% of the probes in April 2015.

**RIPE Atlas uses credits to regulate the platform usage and schedules users’ measurements concurrently.** As of 2015, RIPE Atlas offers four types of measures to its users: ping, traceroute, DNS and SSL \[10\]. In RIPE Atlas, a measurement is defined by a type, a frequency and set of probes. It can therefore refer to an arbitrary number of requests performed from multiple probes. Users can also provide a start date and an end date. If none is provided, the measurement will start as soon as possible and has to be stopped manually. Measurements can be repeated or run only once (one-off). One-off measurements are near real-time if no start time is defined: users should expect results within 10 seconds \[10\].

RIPE Atlas regulates users load via a credit system. Users earn credits by hosting a probe and use them to perform measurements. RIPE’s cost model is based on the resources each measurement needs. traceroute is the most expensive measurement, while ping is the cheapest. One-off measurements are also more expensive (twice more) than their scheduled counterparts as their arrival is not predictable.

**RIPE Atlas uses basic scheduling strategies on each probe to handle concurrent load.** Probes control the measurements frequency with `eperd`, a cron-like utility that can run measurements at regular inter-

\[\text{HTTP measurements are possible but are restricted to researchers and other interested users on a case-by-case basis.}\]

![Figure 1: RIPE Atlas is composed of more than 6000 low-end probes which differ by their hardware: v1 and v2 probes are not powerful with respect to v3.](image)

**Table 1: Overall, RIPE Atlas has hosted 29.8 million requests. When we collected those results, 592,000 concurrent requests were running on the platform.**

<table>
<thead>
<tr>
<th></th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>In progress</td>
<td>58K</td>
<td>120K</td>
<td>414K</td>
<td>592K</td>
</tr>
<tr>
<td>Total</td>
<td>3.1M</td>
<td>7M</td>
<td>19.7M</td>
<td>29.8M</td>
</tr>
</tbody>
</table>

\[\text{We denote by request the individual measures performed by each probe. As a measurement may involve a large number of probes, the number of requests is more representative of the load of the platform.}\]
RIPE—aims at selecting Atlas probes based on spatial distribution rather than traditional uniform selection, thus increasing the load on isolated probes.

3. QUANTIFYING INTERFERENCE

We describe how we quantify interference between measurements performed on a RIPE Atlas probe. We take the perspective of one user \( \lambda \) and one probe \( \rho \) and measure the effects on the results reported by \( \rho \) to \( \lambda \) when: (i) \( \rho \) originates; or (ii) it is the target of concurrent measurements. In particular, we look at changes in the delay reported by \( \rho \) when concurrent one-off traceroutes are originated or when \( \rho \) is being used as ping destination. We use NL Ring nodes [22] as destinations (resp. as sources) of the pings sourced on (resp. destined to) \( \rho \). We also look at changes in the completion time of one-off traceroute experiments performed on \( \rho \).

We measure the delay reported by a probe using ping. Delay-based measurements are indeed the most sensitive to concurrent load. In contrast, traceroute, SSL, and DNS output is less impacted by extra delay.

We also study the decrease in asynchrony by measuring the completion time of one-off traceroutes performed on the probe.

...when increasing the number of concurrent measurements sourced from a probe. To generate load on a probe, we launch an increasing number of one-off traceroutes from it using the REST API [23]. We use traceroute because it uses the most resources, as indicated by the higher cost. It is also one of the tools mostly used by researchers.

...when increasing the number of concurrent measurements targeting a probe. The second technique we use to load a probe consists in gradually increasing the number of ICMP echo requests (800 bytes) targeted to it. We use a set of NL Ring nodes as sources. Each source sends 16 echo requests per second. We start with a single source. Every 2 minutes, we add a new source. We stop when there are 115 sources (115 * 16 = 1840 ping/s). While such frequencies are not common, experiments that use Atlas probes in a mesh-like fashion [15, 16, 18, 19] or that ping them from machines not limited in ping frequency may generate such a load. We use several Ring nodes as sources to limit their individual frequency and to mimic real experiments. To perform remote pings on multiple Ring nodes and collect the results, we built a tool [9] atop Scamper [24].

...and while preventing the effects from external factors. We want to focus on the behavior of the probe and avoid network interferences. For each experiment, we measure the delay between an Atlas probe and a colocated Ring node in the same LAN. We obtained these pairs by a traceroute campaign between each Ring node and Atlas probes in the same AS.

4. DECREASED PRECISION

We now use our methodology (§3) to measure: (i) the decrease in precision of delay-based measurements (this section); and (ii) the decrease in synchrony produced by concurrent measurements (§5). We performed all our measurements on multiple probes (at least two per version) to ensure conformity. As their number is not negligible and their decrease in precision and synchrony is serious, the next figures only focus on v2 probes.

Delays measured from the probe increase when concurrent measurements are launched on it. We launched ping measurements from the Atlas probe and towards eight random Ring nodes plus the colocated Ring node. The ping rate towards each destination is 9 ping/min, averaging 1.4 ping/s over all destinations. We increase the load on the probe by launching successively 10, 25, 50, 100, 250, and 500 one-off traceroutes.

Figure 2 shows the impact of the concurrent one-off traceroutes on the delay measured from a v2 probe. The blue points are the RTTs between the Atlas probe and its colocated Ring node, while the red points are the RTTs between the Atlas probe and another Ring node. The gray areas are the periods when one-off traceroutes are running. The number above each gray area is the number of one-off traceroutes executed. To quantify the impact, we compare the median, 95th percentile, and standard deviation of the ping measurements before the one-off traceroutes (the white area preceding the gray area) and during the one-off traceroutes execution time (gray area). The difference is reported in Table 2.

Delays measured from the probe systematically increase when one-off traceroutes are performed. Starting 100 one-off traceroutes increases the median delay of the concurrent pings by more than 1 ms. For v1 and v2 Atlas probes, the standard deviation is seriously impacted: +16.3 ms (v1) and +7.4 ms (v2). Atlas probes v3 show less effect, the median is only increased by 0.06 ms while the standard deviation is not impacted; this is due to v3 probes having more power. Surprisingly, the num-
impact on ping delay . . . sourced on probe destined to probe when increasing load . . .

<table>
<thead>
<tr>
<th>on probe</th>
<th>50th</th>
<th>95th</th>
<th>stdev</th>
<th>50th</th>
<th>95th</th>
<th>stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>(on : 100 traceroutes +</td>
<td>1.10 ms</td>
<td>7.30 ms</td>
<td>16.3 ms</td>
<td>v1</td>
<td>0.61 ms</td>
<td>0.72 ms</td>
</tr>
<tr>
<td>1.4 ping/s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1</td>
<td>1.20 ms</td>
<td>7.70 ms</td>
<td>7.40 ms</td>
<td>v2</td>
<td>0.50 ms</td>
<td>0.62 ms</td>
</tr>
<tr>
<td>v3</td>
<td>0.06 ms</td>
<td>0.10 ms</td>
<td>0.00 ms</td>
<td>v3</td>
<td>0.06 ms</td>
<td>0.05 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>towards probe</th>
<th>50th</th>
<th>95th</th>
<th>stdev</th>
<th>50th</th>
<th>95th</th>
<th>stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>(on : 9 ping/min, to : 400 ping/s)</td>
<td></td>
<td></td>
<td></td>
<td>(on*: 9 ping/min, to : 1000 ping/s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1</td>
<td>0.11 ms</td>
<td>1.90 ms</td>
<td>15.2 ms</td>
<td>v1</td>
<td>0.20 ms</td>
<td>5.40 ms</td>
</tr>
<tr>
<td>v2</td>
<td>0.22 ms</td>
<td>2.90 ms</td>
<td>3.90 ms</td>
<td>v2</td>
<td>0.45 ms</td>
<td>2.60 ms</td>
</tr>
<tr>
<td>v3</td>
<td>0.00 ms</td>
<td>0.04 ms</td>
<td>0.00 ms</td>
<td>v3</td>
<td>0.00 ms</td>
<td>0.00 ms</td>
</tr>
</tbody>
</table>

Table 2: Quantification of interferences for v1, v2 and v3 probes. At the top, the probe is loaded by sourcing 100 one-off traceroutes. At the bottom, the load comes from incoming pings. Columns represent benchmarking measurements. On the left, we look at the impact of a load on the ping delay reported by the probe. On the right, pings are destined to the probe. With more powerful hardware, v3 probes are less sensitive to load than v1 and v2. *We used these pings to quantify the impact a load towards the probe produces on ping delay sourced on the probe (bottom-left).

Delays measured towards a v2 probe systematically increase when concurrent one-off traceroutes are launched on the probe.

Delays measured towards the probe increase when concurrent measurements are launched on it. We chose eight random Ring nodes plus the colocated Ring node and ping from them towards the Atlas probe with a frequency of 1 ping/s, summing up to a load of 9 ping/s. We then perform successively 10, 25, 50, 100, 250, and 500 one-off traceroutes from the Atlas probe.

Figure 3 shows the impact of the one-off traceroutes on the delay measured towards the Atlas probe. The blue points are the delays reported between the colocated Ring node and the Atlas probe while the red points are the delays reported between another Ring node and the Atlas probe. Again, gray indicates periods when one-off traceroutes are running.

The impact on pings targeting the probe is relatively lower (Table 2). When 100 one-off traceroutes are executed, the median of the RTTs targeting a v2 Atlas probe increases by 0.5 ms. Despite the lower impact, we can easily see the RTT shifts.

Delays measured from and towards a probe increase when it is used as a destination by concurrent measurements. We first launch pings from the Atlas probe towards its colocated Ring node with a frequency of 9 ping/min. We then use an increasing set of Ring nodes to target the probe with 800 bytes pings, each of them sending 16 ping/s (§3).

Figure 4 shows the impact on delay measured from the probe. Unlike with one-off traceroute measurements, the impact now increases with the number of pings directed towards the probe. When the frequency reaches 400 ping/s, the median delay reported by the probe increases by 0.22 ms, while the 95th percentile increases by 2.90 ms and the standard deviation by 3.90 ms. The probe becomes completely overloaded when the frequency reaches 1000 ping/s. This leads to very high delays (~1000 ms). Also, 10% of the pings are lost when the frequency becomes higher than 1280 ping/s. Here, the probe is the target of the load. Traffic is just sent to the probe, without involving the RIPE Atlas controller. We believe the inaccuracy increases progressively because the load
per unit of time also increases. The controller cannot smooth the load by spreading it in time.

Figure 5 illustrates similar effects on the delays measured towards the probe. At the bottom of the figure, each box shows the inter-quartile range of the RTTs between the colocated Ring node and the Atlas probe. The line in the box depicts the median value; the whiskers show the 1st and the 99th percentile, respectively. The top figure indicates the packet loss percentage. When reaching 1000 ping/s, the median RTT increases by 0.45 ms and the 95th percentile increases by 2.60 ms. As in Figure 4 when the frequency becomes even higher, the probe becomes completely overloaded. Reported delays skyrocket (∼1000 ms) and some requests are lost.

Interference effects are compounded when combining source and destination load. So far, we have quantified separately the impact of using a probe as source or as destination. In reality, a probe may be used both as source and as destination at the same time. We could expect these interference effects to be additive, but our experiments show that these effects are compounded.

To quantify, we first start pings between an Atlas probe and its colocated Ring node (9 ping/min). We then start to flood the probe using the set of Ring nodes as described before. Finally, we start series of 25 one-off traceroutes. Figure 6 shows the results. The blue points are the measured delays between the probe and the colocated Ring node. The red vertical line indicates when we start to flood the probe with pings. The gray areas are the periods when one-off traceroutes are running. Before starting to flood the probe, we performed 25 one-off traceroutes in order to be able to compare the interference effects produced by these traceroutes with and without the ping flood. Each green point on the top indicates a traceroute success. The success rate of each one-off traceroute series is also mentioned.

Figure 7 shows that the completion time of a v2 probe increases as the pings frequency targeting this probe increases. Packet losses may appear if the ping frequency towards the probe becomes too high.

Completion time significantly increases with the number of traceroutes. Figure 7 shows that the completion time may be 6.7 minutes (resp. 4.5 minutes) when requesting 50 one-off traceroutes on a v2 (resp. v3) probe. It takes up to 41 minutes with 500 one-off traceroutes on a v3 probe. All probe versions, including v3, are subject to a significant increase in completion time. Further experiments have shown completion times greater than one hour, even for v3 probes.

Completion time increases with the load towards the probe. In Figure 6, while the completion time for the first 25 one-off traceroutes takes up to 6.2 minutes, it takes up to 11.3 minutes for the second series of one-off traceroutes and up to 20.2 minutes for the third series. Sending 500 ping/s to a probe may then multiply...
Figure 7: One-off traceroutes completion time is also impacted by concurrent measurements, independent of the hardware used. Results can be delayed by more than half an hour—making it impossible to perform synchronized experiments.

the one-off traceroutes completion time by more than 3. When the ping frequency becomes too high, most of the traceroutes fail.

Key points Under load, requested measurements may be delayed, rendering the platform unsuitable to synchronized measurements. One could not ensure that pings or traceroutes start simultaneously on multiple probes. This is especially a problem when one wants to measure the effect of a single event from multiple vantage points, or an exogenous event. This problem applies to all probe hardware—including the most powerful v3.

6. DISCUSSIONS

We now describe the impact for researchers working with the platform (§6.1) as well as two solutions on how to mitigate interference in practice (§6.2).

6.1 Impact for researchers

On previous works. As described earlier, many research papers have used RIPE Atlas. Some of them relied on delay-based measurements [25, 6, 3, 4, 15] which can be impacted by interferences. For instance, both [6] and [3] relied on shifts in the measured delay from probes to detect hijack or routing changes. Increased delay due to load could therefore be misclassified as hijack or a routing change, polluting the results. Based on this paper, any delay-based measurement obtained from v1 and v2 probes should be avoided if a precision below 15 ms is required.

On publicly available data. RIPE Atlas makes publicly available all the results collected with the platform since its inception in 2010 [26]. Researchers using these data should consider the impact of interferences. Especially for data collected before 2013—prior to v3 probes. We suggest researchers to be very careful when using publicly available delay measurements.

6.2 Solutions

Provide feedback to users with a measurement confidence index. A fundamental problem with Atlas is that the user has no visibility on the concurrent load of the platform. For that, we argue that RIPE can return a “confidence index” along with each result. The index would be function of the platform concurrent load. High (resp. low) load would lead to low (resp. high) confidence. Obviously, computing this metric should be done based on passive measurements to not stress the platform even more. We are currently working on calibrating such a metric using our measurements.

Enforce synchronization. While real-time is not a reasonable objective on shared platforms, more precise scheduling is achievable by maintaining a lower load on the probes and delaying upcoming measurements in favor of already scheduled events. Upon a measurement request, the user could then be informed of the exact timing of her experiment. Such an approach is however not possible if users do not all have the same privileges and some experiments can be preempted.

7. RELATED WORK

Other researchers have observed measurement interference and its impact on RTT. In [27], Whiteaker et al. highlighted the effect virtualization can produce on measured delays. As a number of large-scale platforms use VMs [28, 29, 22], tools such as [30] for PlanetLab, provide users with information about the state of these platforms and their nodes. In contrast, RIPE Atlas does not use virtualization and relies on a scheduler to share resources among users.

In [31], Bajpai et al. showed that RTTs from v2 probes to the first hop router are consistently higher than for v3 probes. They do not however study the relation between the measured delays and the load of the probes.

Mok et al. [32] proposed a technique to reduce packet sending time on low-end devices such as Atlas probes. This technique may be useful to counteract some of the interference effects we expose in this paper.

8. CONCLUSION

We presented the first measurement study of user-induced interferences on the RIPE Atlas platform. We found that measurements do interfere with each other. Delays reported from the probe increase and vary more when they compete with concurrent measurements. Measurement campaigns can further be arbitrary delayed, making it hard to perform simultaneous experiments from multiple probes.

Our findings also bring up new, non-trivial research questions: how can we design measurement platforms that provide more isolation between users, while still be efficient (i.e., not requiring a global lock). We plan to explore this direction in the future.

9. REFERENCES


