Consolidated Review of

From Paris to Tokyo: On the Suitability of Ping to Measure Latency

1. Strengths

This paper makes the point that load balancers break ping just as they do traceroute. In fact, the authors discover that, due to link aggregation, even measurements along the same IP-level path may have very different delays. The paper proposes to apply the Paris-traceroute idea of controlling path identifier to ping, resulting in Tokyo-ping. It then examines this approach to show the problems that occur without it.

A good problem to work on: ping is widely used, and the paper helps us interpret its results and improve the tool. Very suitable for a short paper. Paper goes on to address many of the questions that I had while reading it. Interesting conclusion: "high variability ...is likely a measurement artifact of the [ping] tool itself." Well-written and interesting to read.

Interesting case-study analysis of RTT anomalies (though the study ends without resolution!)

Replacement tool is available for the community. The findings and the tool made available to the community have definitely impact.

2. Weaknesses

So few paths. Pretty small set of paths analyzed (though this is a six pager, and the authors go into substantial detail for all paths studied). Do the results showing that each flow has consistent RTT hold over bigger data sets?

Case study started in 3.2 ends without any resolution or summary discussion

The limitation on Linux machine as destination hosts is not negligible. Is it possible to devise a strategy to detect them only based on pinging?

Since Paris-traceroute identified the root problem and solution, applying that solution to ping is somewhat incremental.

3. Comments

I have mixed feelings about the paper. It's interesting, but ultimately I feel it needs a much bigger measurement study to warrant publication, not just the handful of paths in the paper. As is, I wouldn't feel comfortable trusting that ping variation I see on an arbitrary path is due to this problem, unless I had run a large study (or a study on that path) myself. Since the tool itself is a trivial tweak of Paris traceroute (and one of the Paris papers already showed latency varying with flow ID, although not with fixed IP level path like you show), I think you should go do a large study, then resubmit this paper. The key result to look at is whether the jitter on most paths can be explained by variation across flows, and, if so, whether individual flows are consistent. To clear room for that study, you can rewrite the paper in a more straightforward style, instead of the current mystery investigation. I want to see this work published, but I want it published with a complete Internet-scale study. That study seems very easy to conduct, so there is no reason not to.

A nice paper, with interesting detailed results, and important findings. I was somewhat shocked that this potential problem hadn´t occurred to anyone since the Paris-traceroute work was published. The level of detail in the paper was great, and made for fun reading. I just wish there was some summary or punch line to the study that ends 3.4. From the text, it seems like the answer left is "we still don´t know what´s behind the problem"

While that conclusion is a bit incremental, the importance of ping to the IMC community makes a carefully done study of the issues around ping worthwhile.

Intro: When you say you observed this in Atlas probes and want to get to the bottom of it, one possible explanation that came to my mind is that it could be an issue with Atlas specifically. Would be nice if you dealt with this. Intro: You mention how this could impact apps that have multiple flows. It would be nice to have a concrete example of a negative effect.

Page 1, "flow identifier is commonly used by network devices to perform load balancing": can you define flow identifier here (since IPv4 has none). Later you define the effective flow id on page 2 "Routers often perform hashing in hardware using bytes 12-19 of the IP header and bytes 1-4 of the IP payload.” but that's much later. Also, when you define that: is that a universal rule, or just one brand of router?

2.2: "Using UDP probes with Tokyo-ping, we were able to isolate the separate contributions to the RTT of the forward and return paths." I see how you are able to hold each direction fixed against load balancers, but I have no idea how you can separate forward delay from reverse.

Page 3: "For each run, we sent 100 sequential pings followed by 100 probes for each different flow-id to be tested." How many flow-ids were tested here?

Page 3, "We find that ping is, in general, a mediocre estimator for RTTs and an extremely pessimistic estimator for jitter.” "pessimistic" could be taken different ways, as low or high or something else. You may want a different term (perhaps "and overestimates jitter"?).

Figure 4: it's unclear why some of your results here are quantized and others not. If you can measure high precision flow-id values, why not use the same with ping? Having ping be quantized means your data mixes two root causes: operating-system effects and multi-path effects. These root causes should be separated, if possible.

Page 4: "Looking at the per flow-id curves..." and much of section 3.2: it's somewhat interesting to hear your process, but is it important? (This part of the work feels unnecessary.)

Why did you choose 157 millisecond spacing in the experiment repeat described at the end of 3.3?

Figure 7: please define "rtt delta"

Separating section 3.3 and 3.4 doesn't make much sense, since 3.4 looks incomplete (preliminary, manual investigation with not ground-shaking conclusions) and continues on the same investigation. What about the limitation on Linux destination hosts? Is it possible to devise a strategy to detect them only based on pinging?

Page 6: "If the application needs consistency across channel..." surely applications that need such consistency already due buffering and can tolerate variation across flow-id, no? "... such as Multi-Path TCP": doesn't multi-path TCP already accommodate diverse paths, with multiple RTT and cwnd estimates?

The LAG result is neat!

4. Summary from PC Discussion

PC discussion: the PC discussed this paper and was happy to see this result from Paris-traceroute applied to ping and backed up by detailed analysis of specific paths.

The PC had a one serious concern: the paper identifies one source of variance in latency (flow IDs) and shows that it can address that problem. However, it leaves unresolved to what extent other sources of variance may exist. (In fact, the paper doesn't explicitly state how it filters out variance due to cross-traffic queuing delay. We presume this problem is already addressed by keeping the lowest observation, the usual mechanism.)

Taken most negatively, lack of any study of a representative sample of internet paths leaves this paper as incomplete: no one can use it and be sure they've addressed variance sufficiently without doing additional measurements.

Some sort of larger-scale study would start to address this question: probing some large number of sites and seeing if Tokyo-ping results in low variance for individual paths. Ideally such a study would be "representative" of the Internet. If the analysis is automated, it should be easy to test 1000s of paths or more.

Even though such a study would lack ground truth, it would complement the paper's existing careful study of a few paths with ground truth. If Paris-traceroute is automated, such a study might require some traffic and some automation to analyze the results, but it should not be conceptually difficult. We strongly encourage the authors to consider some longer study such as this, and if possible, include the results in the final paper.

5. Authors' Response

The initial submission contained a detailed analysis of tightly controlled experiments where we were able to get ground truth about the data path(s). First, we avoided testing from or to virtualized or underpowered hosts, and did not use routers as end-points as sub-millisecond precision was desired. Second, we selected them in locations where we could understand the observations because we could work with the operators and see many of the router configurations to confirm details. This level of control of the experiments enabled us to delve into the per-flow delay phenomenon, avoiding measurement biases, for example, due to unknown cross-traffic. In controlled experiments, we were able to exclude potential causes of the observed per-flow behavior (e.g. congestion and MPLS settings), and focus on what appeared as the critical potential culprits (i.e., to ECMP and LAG).

In the final version of the paper, we have added a larger scale study involving 850 Internet-distributed destinations. The sources and destinations are in different ASs to get diversity in the set of measured paths. This study still shows a significant per-flow delay variability effect that ping is not able to capture. However, the problem with such a study is that we do not know what are the relative effects of various factors (e.g., network load, inter-domain routing paths, per-flow load-balancing, etc.) on the measured delay variability. We see no scalable way to isolate those causes. Worse, though it seems intuitive, we can not definitively show that the causes for flow-id revealed variance are limited to ECMP and LAG. We had confidence in this with the finely focused measurements.

To compound measurement complexity, there are two issues with Linux machines. When used as source, the precision of the measurements is limited to 3 digits, hence the staircase curves. The second limitation is that, when a Linux machine is the target of an UDP probe, it encapsulates the full offending payload in the return packet, making it impossible to select a return flow-id for UDP probes. To counter this, we sent ICMP probes to Linux machines. One could use UDP probes to determine if a destination is running Linux and adjust the probes accordingly, but this is not within the scope of our work.

We agree that our methodology is simple. We see this as a feature. We believe the findings are of interest to the community as ping is widely used for delay measurement by applications and by researchers.

High variance in measurements using Atlas probes led us to conduct the study in this paper. However, the paper does not present results from the Atlas probes. We used FreeBSD and Linux servers. Our results are thus not an artifact of the Atlas RIPE probes nor FreeBSD or Linux servers. We removed mention of the Atlas probes to reduce confusion.

Regarding the implication of our findings on applications, our goal was to underline that applications establishing multiple TCP connections should be aware that different delay and jitter can be observed on different connections. Hence, multi-channel applications are advised not to rely on a single control channel to accurately estimate delay and jitter of all opened TCP connections. Moreover, this work suggests that accurately monitoring per-channel performance from outside the application is harder than is commonly believed. Indeed, the performance that a monitoring tool (e.g., ping, IP-SLA, etc.) measures is not necessarily representative of the performance experienced by specific applications.

Our study relies on Round Trip Time (RTT) distributions, rather than observation of the minimum RTT. It demonstrates that the delay distribution measured by ping can in fact be composed of multiple per-flow distributions each of which may exhibit significantly less variability. We ensured that cross-traffic was not influencing our results by running our subset of controlled measurements for different durations (up to 8 hours), and at different times of the day/week. In addition, we verified with operators that the link utilization on the path with largest RTT dispersion was low (below 50%) during our experiments. For the larger scale measurements, we used a range of observations along the distribution, removing the lowest and highest data points.