A Single Common Metric to Characterize Varying Packet Delay

Bob Briscoe (Ed.) (ietf@bobbriscoe.net), Greg White (G.White@CableLabs.com),
Vidhi Goel (Vidhi_Goel@Apple.com) and Koen De Schepper (koen.de_schepper@nokia-bell-labs.com)

September 2021

Introduction

The most commonly reported delay metric used to characterize network performance is average delay.

A real-time application generally discards any packet that arrives after the play-out deadline, whether for streamed video, interactive media or online gaming. For this broad class of applications an average delay metric or median delay metric is a distraction---waiting for the median delay before play-out would discard half the packets. To characterize the delay experienced by an application it would be more useful to quote the delay of a high percentile of packets. But which percentile? The 99th? The 99.99th? The 98th? The answer is application- and implementation-dependent, because it depends on how much discard can effectively be concealed in the tradeoff with delay (1%, 0.01% or 2% in the above examples, assuming no other losses). Nonetheless, it would be useful to settle on a single industry-wide percentile to characterize delay, even if it isn't perfect in every case.

This brief discussion paper aims to start a debate on whether a percentile is the best single delay metric and, if so, which percentile the industry should converge on.

Note that the question addressed here is how to characterize varying delay. That is orthogonal to the questions of what delay to measure and where to measure it. For instance, whether delay is one-way or two-way, and whether delay is measured in the application, at the transport layer or just at the bottleneck queue depends on the topic of interest and is an orthogonal question to the focus of this paper. Similarly, for delay under load, the question of which background traffic loads and patterns to use for the scenario of interest is extremely important, but it is an orthogonal question to the narrow focus of this paper; which is solely about how to characterize delay variability most succinctly and usefully in *any* of these scenarios.

Target Usage

The potential uses for this delay metric are similar to those for packet delay variation listed in IETF RFC 5481 [RFC5481]. That is: service level comparison, adapting de-jitter buffer size; adapting forward error correction parameters; determining queue occupancy; etc. Note though that the goal here is to characterize the delay that is effectively experienced not just to quantify the delay variation.

Don't we need two metrics?

In systems that aim for a certain delay, it has been common to quote mean delay and jitter. The distribution of delay is usually asymmetric, mostly clustered around the lower end, with the median close to the minimum, but with a long tail of higher delays [Sundar20]. Most industry jitter metrics [Sundar20] are insensitive to the shape of this tail, because they are dominated by the *average* variability in the bulk of the traffic around the mean. However, for real-time, it doesn't matter how little or how much variability there is in all the traffic that arrives before the play-out time. It only matters how much traffic arrives too late. The size of all the lower-than-average delay should not be allowed to counterbalance a long tail of above-average delay.

The argument for a single percentile delay metric is strongest for real-time applications, including real-time media [Bouch00], [Yim11] and online games. But a delay metric is also important for non-real-time applications, e.g. web and transactional traffic more generally (e.g. RPC). Here, average delay is indeed important. But still, the user's perception is dominated by the small proportion of longer delays [Wilson11].

Average packet delay would be the best metric if delay at the application level only depended on each packet in isolation. But it it rarely does. Typically interactions are dependent on each other [Wischik08], so higher delay packets hold back progress of the whole flow of logic. This is clearly the case for isochronous media (real-time) and for ordered delivery. But, even where a connection consists of multiple streams or multiple objects that have no explicit protocol sequencing, the interdependencies will typically still be present in the application logic. For instance, a waterfall diagram shows how the loading of objects within a web page depends on other objects in other connections. Thus, even with unordered delivery at the transport layer, a high percentile delay metric will invariably be more useful, or at least as useful, as mean or median delay.

Arguments can be made for more than one delay metric to better characterize the delay distribution. But, if we can settle on a single metric, we should, for the sake of simplicity---simplicity for users and the regulators that represent them, and simplicity in measurement techniques and equipment. A single metric would not preclude anyone quoting other delay metrics, as long as they also quoted the one metric that everyone else quoted.

Which percentile?

The factors that influence the choice of percentile are:

- For real-time, the degree of late packet discard that can be efficiently concealed by coding or motion interpolation (both that which is typical today and that which could be typical in future).
- The lag before results can be produced. To measure a high delay percentile accurately requires a large enough number of packet measurements so that at least ten or so fall above the percentile. For instance, when measuring the 99th percentile on average 1 in 100 packets will lie above the percentile, so at least an order of magnitude more than 100 packets (1,000 and preferably more) have to be measured. In contrast the 99.999th percentile would require at least 1,000,000 packets. At a packet rate of say 10k packet/s, they would take respectively 100 ms and 100 s.
    - The P99.999 would be impractical to use to adapt internal parameters accurately, while the P99 makes it possible to start controlling parameters nearly immediately after a flow has started
    - Similarly, it is possible to display the 99th percentile on a dashboard soon after the flow has started (see for instance the open source GUI we provide to display the P99 as well as a delay distribution plot that updates every second, for comparing different congestion controllers and Active Queue Management (AQM) algorithms [Bond16], [l4sdemo]).
    - Similarly, for research or product testing where a large matrix of tests has to be conducted, it would be far more practical if each test run took 100 ms, rather than 100 s.
    - At the slowest packet rate of a typical game control stream (perhaps 30 pkt/s), even the P99 is only just practical, taking at least 33 s before an accurate result is available (the P99.999 would be completely impractical, taking over 9 hours).
- To calculate a high percentile requires a significant number of bins to hold the data (unless the likely result is known in advance). This can make a high percentile prohibitively expensive to maintain, e.g. on cost-reduced consumer-grade network equipment.

As a strawman, we propose **the 99th percentile (P99)** as a lowest common denominator delay metric for the communications industry. We believe this is a workable compromise between the above somewhat conflicting requirements, but the purpose of proposing a specific number is to provoke debate, not close it off.

We emphasize that a single common metric such as the P99 would only serve as a lowest common denominator. Ideally, the whole range of percentiles would be plotted as a cumulative distribution function (CDF)---preferably a log-scaled complementary CDF to reveal the tail latencies clearly (see Figure 1). However, where nothing more than a single number is needed, we advocate at least the P99.

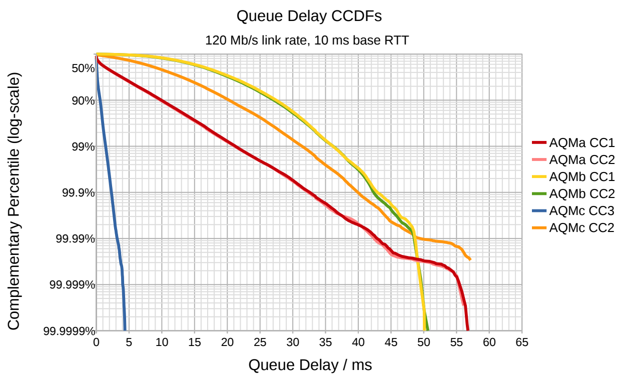

Figure 1: *Example log-scale Complementary Cumulative Distribution Function (CCDF) of Queue Delay (legend obfuscated, to focus on the visualization, not the details of the particular example)*

The 'Benchmark Effect'

As explained in the introduction, defining a delay metric is not just about choosing a percentile. The layer to measure and the background traffic pattern to use also have to be defined. As soon as these have been settled on, researchers, product engineers, etc. tend to optimize around this set of conditions---the so-called 'benchmark effect'. In fact, it could be argued that the historical use of average latency as the single metric led to issues like bufferbloat being ignored for so long. It is possible that harmonizing around one choice of percentile will similarly lead to a benchmark effect.

However, a percentile metric seems robust against perverse incentives, because it seems hard to contrive performance results that fall off a cliff just beyond a certain percentile. Nonetheless, the best way to mitigate any benchmark effect is to ensure that the metric chosen for the benchmark realistically reflects the needs of most applications.

How to articulate a percentile to the public?

Delay is not an easy metric for public consumption, because it exhibits the following undesirable features:

- It is measured in time units (ms) that seem too small to matter, and which are not common currency for a lay person
  - The only foreseeable remedy here would seem to be tutorial material explaining the impact of even small delays on natural human interactions, and how small delays accumulate into large ones over serial flows of logic.
- Larger is not better.
  - It might be possible to invert the metric [RPM21], but rounds per minute carries an implication that it is only for repetitive tasks, which would limit the scope of the metric

A delay percentile is expressed as a delay, so it shares the same failings. But a percentile carries additional baggage as well:

- It's not immediately obvious why the particular percentile has been chosen.
  - It would need some indication that it was an industry-standard metric, perhaps IETF-P99.
- A percentile is not the easiest of metrics to explain.
  - One has to say something like "the delay of 99% of packets was below x ms".
  - Nonetheless, people are used to looking up the meaning of metrics on the web;
  - And anyway people can still use a numeric metric for comparison without understanding its inherent meaning.

How to reach consensus?

The IETF seems like the appropriate body to reach consensus on the percentile to use to express delay, across countries, across standards bodies, and across industry sectors. And IPPM would seem to be the appropriate WG.

Are there arguments against this idea? Or interest in taking it further?

References

[Bouch00] Bouch, Anna & Sasse, M. Angela, "The case for predictable network service," In Proc. Multimedia Computing and Networking Conference (MMCN'2000), 188--195 (2000)

[Bond16] Bondarenko, Olga; De Schepper, Koen; Tsang, Ing-Jyh; Briscoe, Bob; Petlund, Andreas & Griwodz, Carsten, "Ultra-Low Delay for All: Live Experience, Live Analysis," In Proc. ACM Multimedia Systems; Demo Session, ACM, 33:1--4 (2016)

[l4sdemo] Code repository; Video of GUI

[RFC5481] Morton, Al & Claise, Benoit, "Packet Delay Variation Applicability Statement," RFC Editor, RFC5481 (2009)

[RPM21] Stuart Cheshire & Vidhi Goel, "Reduce network delays for your app" Apple Worldwide Developer Conference'21 (Jun 2021)

[Sundar20] Sundaresan, Karthik; White, Greg & Glennon, Steve, "Latency Measurement: What is Latency and How Do We Measure It?" In Proc. Fall Technical Forum and NCTA Technical Papers (2020)

[Wilson11] Wilson, Christo; Ballani, Hitesh; Karagiannis, Thomas & Rowstron, Ant, "Better Never than Late: Meeting Deadlines in Datacenter Networks," Proc. ACM SIGCOMM'11, Computer Communication Review 41(4):50–-61 (Aug 2011)

[Wischik08] Wischik, Damian "Short Messages," Philosophical Transactions of the Royal Society, 366(1872):1941--1953 (2008)

[Yim11] Yim, Changhoon & Bovik, Alan C., "Evaluation of Temporal Variation of Video Quality in Packet Loss Networks," Image Commun., Elsevier Science, 26(1):24-38, (Jan 2011).