Benchmarking Working Group M. Konstantynowicz, Ed.
Internet-Draft V. Polak, Ed.
Intended status: Informational Cisco Systems
Expires: May 17, 2019 November 13, 2018

Probabilistic Loss Ratio Search for Packet Throughput (PLRsearch)
draft-vpolak-bmwg-plrsearch-00

Abstract

This document addresses challenges while applying methodologies described in [RFC2544] to benchmarking NFV (Network Function Virtualization) over an extended period of time, sometimes referred to as "soak testing". More specifically to benchmarking software based implementations of NFV data planes. Packet throughput search approach proposed by this document assumes that system under test is probabilistic in nature, and not deterministic.

Table of Contents

1. Motivation

Network providers are interested in throughput a device can sustain.

RFC 2544 assumes loss ratio is given by a deterministic function of offered load. But NFV software devices are not deterministic (enough). This leads for deterministic algorithms (such as MLRsearch with single trial) to return results, which when repeated show relatively high standard deviation, thus making it harder to tell what "the throughput" actually is.

We need another algorithm, which takes this indeterminism into account.

2. Model

Each algorithm searches for an answer to a precisely formulated question. When the question involves indeterministic systems, it has to specify probabilities (or prior distributions) which are tied to a specific probabilistic model. Different models will have different number (and meaning) of parameters. Complicated (but more realistic) models have many parameters, and the math involved can be very complicated. It is better to start with simpler probabilistic model, and only change it when the output of the simpler algorithm is not stable or useful enough.

TODO: Refer to packet forwarding terminology, such as "offered load" and "loss ratio".

TODO: Mention that no packet duplication is expected (or is filtered out).

TODO: Define critical load and critical region earlier.

This document is focused on algorithms related to packet loss count only. No latency (or other information) is taken into account. For simplicity, only one type of measurement is considered: dynamically computed offered load, constant within trial measurement of predetermined trial duration.

Also, running longer trials (in some situations) could be more efficient, but in order to perform trial at multiple offered loads withing critical region, trial durations should be kept as short as possible.

3. Poisson Distribution

TODO: Give link to more officially published literature about Poisson distribution.

Note-1: that the algorithm makes an assumption that packet traffic generator detects duplicate packets on receive detection, and reports this as an error.

Note-2: Binomial distribution is a better fit compared to Poisson distribution (acknowledging that the number of packets lost cannot be higher than the number of packets offered), but the difference tends to be relevant in loads far above the critical region, so using Poisson distribution helps the algorithm focus on critical region better.

When comparing different offered loads, the average loss per second is assumed to increase, but the (deterministic) function from offered load into average loss rate is otherwise unknown.

Given a loss target (configurable, by default one packet lost per second), there is an unknown offered load when the average is exactly that. We call that the "critical load". If critical load seems higher than maximum offerable load, we should use the maximum offerable load to make search output more stable.

Of course, there are great many increasing functions. The offered load has to be chosen for each trial, and the computed posterior distribution of critical load can change with each trial result.

To make the space of possible functions more tractable, some other simplifying assumption is needed. As the algorithm will be examining

(also) loads close to the critical load, linear approximation to the function (TODO: name the function) in the critical region is important. But as the search algorithm needs to evaluate the function also far away from the critical region, the approximate function has to be well- behaved for every positive offered load, specifically it cannot predict non-positive packet loss rate.

Within this document, "fitting function" is the name for such well-behaved function which approximates the unknown function in the critical region.

Results from trials far from the critical region are likely to affect the critical rate estimate negatively, as the fitting function does not need to be a good approximation there. Instead of discarding some results, or "suppressing" their impact with ad-hoc methods (other than using Poisson distribution instead of binomial) is not used, as such methods tend to make the overall search unstable. We rely on most of measurements being done (eventually) within the critical region, and overweighting far-off measurements (eventually) for well-behaved fitting functions.

4. Fitting Function Coefficients Distribution

To accomodate systems with different behaviours, the fitting function is expected to have few numeric parameters affecting its shape (mainly affecting the linear approximation in the critical region).

The general search algorithm can use whatever increasing fitting function, some specific functions can be described later.

TODO: Describe sigmoid-based and erf-based functions.

It is up to implementer to chose a fitting function and prior distribution of its parameters. The rest of this document assumes each parameter is independently and uniformly distributed over common interval. Implementers are to add non-linear transformations into their fitting functions if their prior is different.

TODO: Move the following sentence into more appropriate place.

Speaking about new trials, each next trial will be done at offered load equal to the current average of the critical load.

Exit condition is either critical load stdev becoming small enough, or overal search time becoming long enough.

The algorithm should report both avg and stdev for critical load. If the reported averages follow a trend (without reaching equilibrium),

avg and stdev should refer to the equilibrium estibated based on the trend, not to immediate posterior values.

TODO: Explicitly mention the iterative character of the search.

5. Algorithm Formulas

5.1. Integration

The posterior distributions for fitting function parameters will not be integrable in general.

The search algorithm utilises the fact that trial measurement takes some time, so this time can be used for numeric integration (using suitable method, such as Monte Carlo) to achieve sufficient precision.

5.2. Optimizations

After enough trials, the posterior distribution will be concentrated in a narrow area of parameter space. The integration method could take advantage of that.

Even in the concentrated area, the likelihood can be quite small, so the integration algorithm should track the logarithm of the likelihood, and also avoid underflow errors bu ther means.

6. Known Implementations

The only known working implementatin of Probabilistic Loss Ratio Search for Packet Throughput is in Linux Foundation FD.io CSIT project. https://wiki.fd.io/view/CSIT. https://git.fd.io/csit/.

6.1. FD.io CSIT Implementation Specifics

In a sample implemenation in FD.io CSIT project, there is around 0.5 second delay between trials due to restrictons imposed by packet traffic generator in use (T-Rex), avoiding that delay is out of scope of this document.

TODO: Describe how the current integration algorithm finds the concentrated area.

7. IANA Considerations

..

8. Security Considerations

..

9. Acknowledgements

..

10. Normative References

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

Maciek Konstantynowicz (editor)
Cisco Systems

Email: mkonstan@cisco.com

Vratko Polak (editor)
Cisco Systems

Email: vrpolak@cisco.com