draft-vpolak-bmwg-plrsearch-02

IETF-105 Montreal BMWG Meeting

Authors: Vratko Polák, Maciek Konstantynowicz

Presented by: Vratko Polák

Draft changes -01 to -02

- General edits for better readability.
- Updated in “PLRsearch Building Blocks” section.
- Fixed fitting function formulas.
- Described newly implemented offered load selection.
- Added Security Considerations (standard BMWG boilerplate).

Motivation

- Using [RFC2544] for NFV, specifically NFV software data planes, often yields not repetitive and not replicable end results.
- Users are still interested in “throughput”, or some statistical analogue.
- Even if systems under test are modeled as probabilistic (as opposed to deterministic), simplifying assumptions have to be made to allow definition of such statistical analogue terms.

- Probabilistic Loss Ratio search (PLRsearch) is a class of algorithms which apply probabilistic theory to turn unreliable measurements into as reliable conclusions as practically possible.

Overview

- PLRsearch is a packet “throughput” search algorithm suitable for **probabilistic** (as opposed to deterministic) systems.
- It searches for probabilistically defined **critical load** satisfying given **target loss ratio**.
- It performs **sequential** trial measurements of offered load constant within a measurement.
- It still applies many assumptions on the system behavior, often unrealistic for some systems.
  - It assumes results of trial measurements are **independent** of each other.
  - It assumes possible loss counts follow **Poisson distribution**.
  - It assumes average **loss ratio** does not depend on trial duration.
  - It relies on heuristic **fitting functions** to relate results of trial measurements with different offered loads.
- It uses Bayesian inference computing both trial measurements’ offered load and final estimate.

Side comments

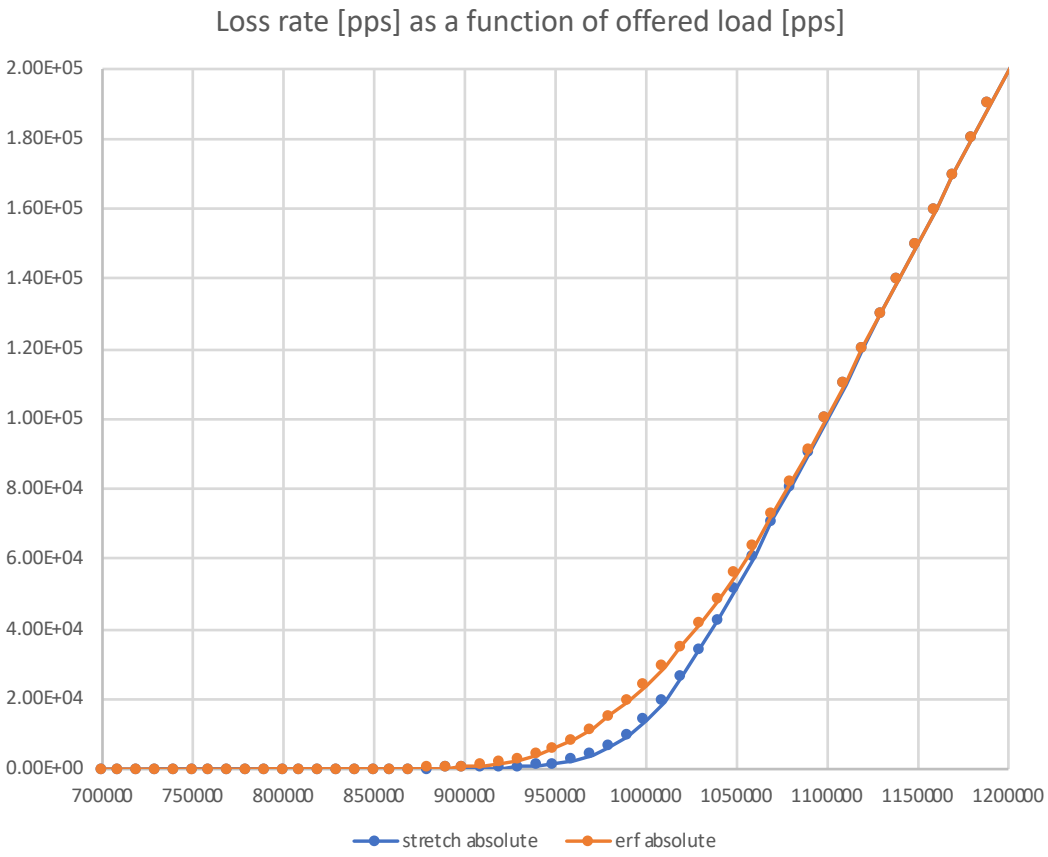
- PLRsearch is similar to RFC 2544 compatible throughput searches, but the values of offered load are given by complicated math, as opposed to simple rules.
- Exact zero probabilities cause technical problems for Bayesian inference.
  - Therefore target loss ratio of exact zero is not supported by PLRsearch.
  - Recommended value is 10^-7. Smaller values are possible, but the search would converge more slowly.
- Short trial durations are preferred, to keep offered loads at best values, but Bayesian inference takes some time to compute.
- Prototype implementation in FD.io CSIT project works, but still contains some deficiencies.

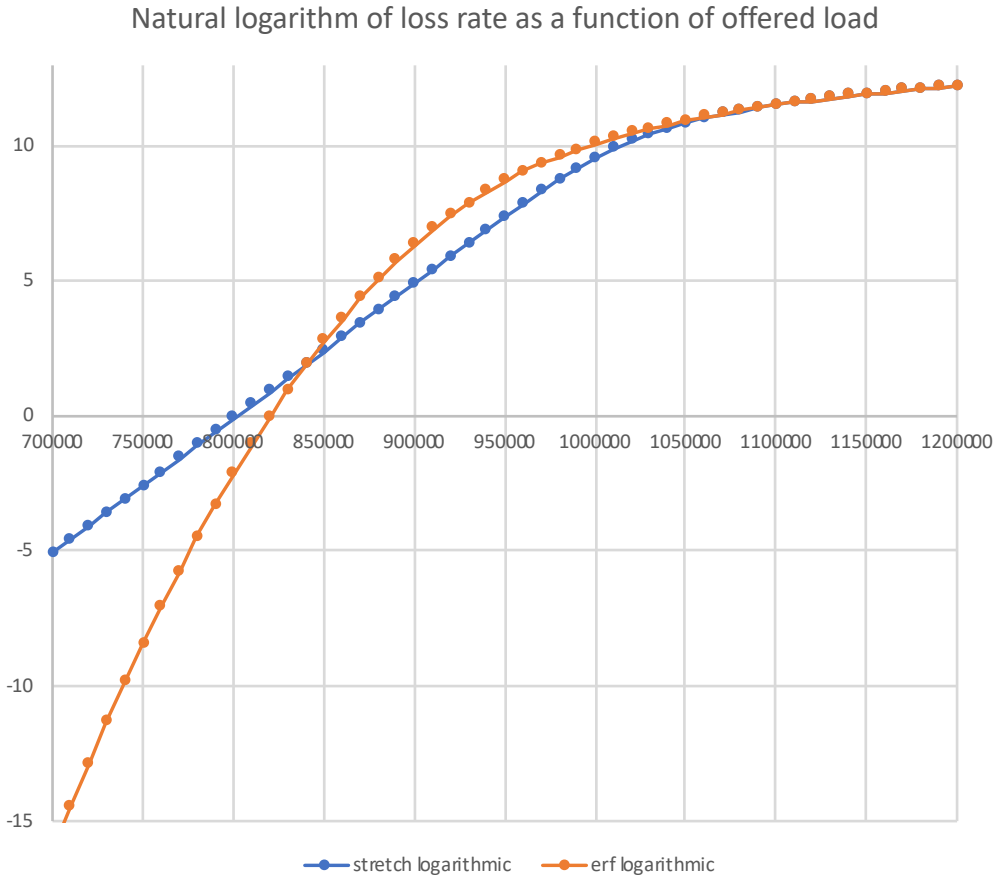
- Other advanced techniques (such as neural networks) could be used (also for deterministic searches) in hope to choose more relevant offered loads, but estimation of critical rate has to be done by sound statistical methods.

Comparison with Deterministic Search Algorithms

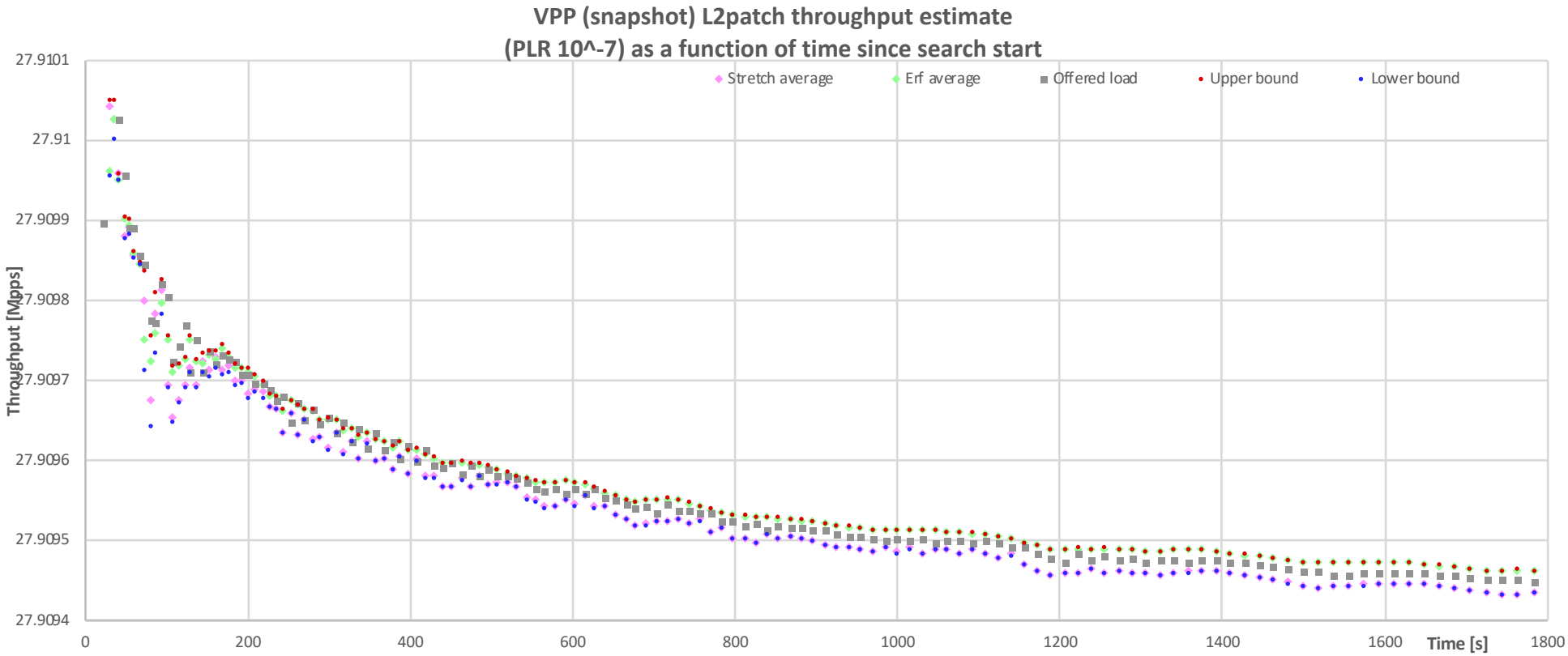
- Deterministic search result lands somewhere in non-deterministic region.
- Binary search is able to reach arbitrarily good accuracy...
- ... but repeated search can land in a different part of non-deterministic region.
- If performance changes with time, binary search tends to get stuck near "round" value.
- Accuracy of PLRsearch depends on "quality" of the system, infrequent but big losses leave large resulting interval.
- For almost-deterministic systems, accuracy of PLRsearch result is very good.
- If performance changes with time, PLRsearch returns "average" result, with medium accuracy."

Fitting Function Graphs

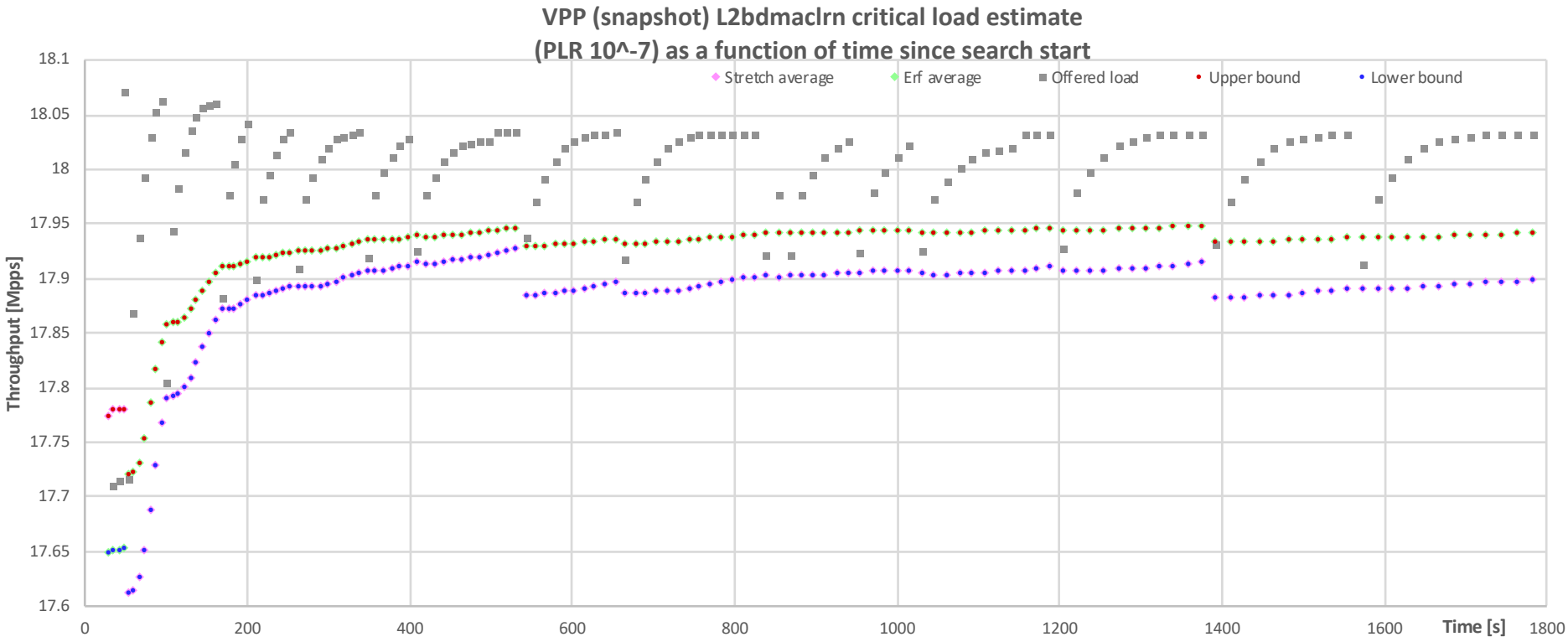




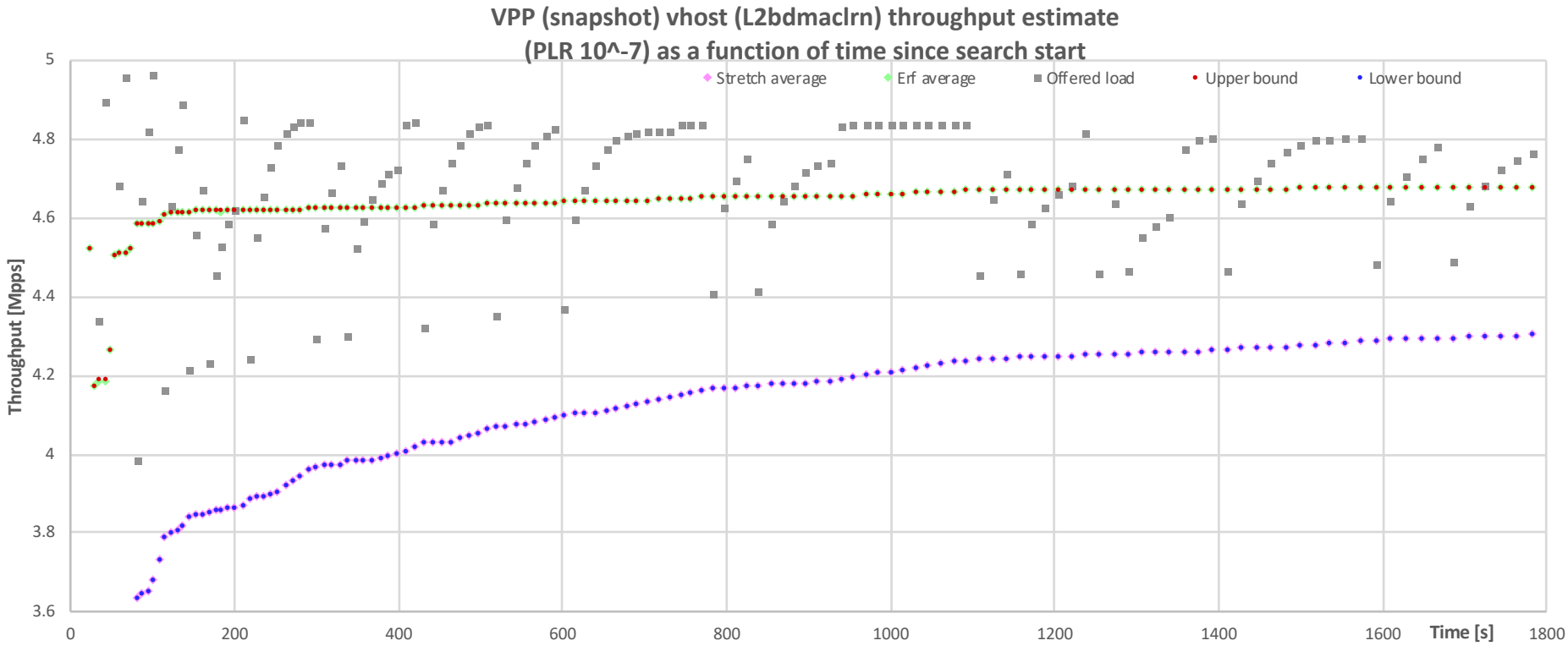
Estimate Evolution Graph



Estimate Evolution Graph



Estimate Evolution Graph



Sample Implementation

- Current work-in-progress implementation of PLRsearch is in Linux Foundation FD.io CSIT project.
- CSIT project general information:
  - https://wiki.fd.io/view/CSIT
  - https://git.fd.io/csit/
- The most recent code right now:
  - https://gerrit.fd.io/r/#/c/16667/29

Implementation specifics

- Monte Carlo numerical integration is used, even though parameter space is currently only two-dimensional.
- Importance sampling is needed, because posterior distributions are concentrated in very narrow areas.
- Some amount of samples is needed for locating new area after each additional trial measurement result.
- Two fitting functions (named “stretch” and “erf”) are used to introduce systematic error.
  - If estimates from the two functions do not agree, it is possible that neither of the estimates is good.
  - If the estimates agree, it might be just by luck, not by predictable system behavior.
  - Better ways to determine reliability of the estimates could be applied.
- Currently, zero-loss results move the critical load estimate too little. Some workarounds might be needed.
- Implemented in Python 2.7, using multiprocessing.
- Will be published in PyPI once CSIT starts using it in real tests.

Next Steps

- Welcome more reviews from BMWG
- Draft adoption by BMWG

draft-vpolak-bmwg-plrsearch-02

IETF-105 Montreal BMWG Meeting

Authors: Vratko Polák, Maciek Konstantynowicz

Presented by: Vratko Polák

THANK YOU!