IETF104-Prague, BMWG

draft-vpolak-mkonstan-mlrsearch-00

Authors: Vratko Polák, Maciek Konstantynowicz

Presented by: Maciek Konstantynowicz

Motivation

- Using [RFC2544] for NFV, specifically NFV software data planes often yields not repetitive and not replicable end results.

- Multiple Loss Rate search (MLRsearch) aims to address some of the challenges with NFV data plane testing by defining a throughput search with optimized (shorter) run time and ability to find multiple throughput rates each associated with distinct Packet Loss Ratios (including zero packet loss).

- **MLRsearch is compatible with RFC2544.**

Overview

- MLRsearch is a packet throughput search algorithm suitable for deterministic (as opposed to probabilistic) systems.
- MLRsearch discovers multiple packet throughput rates in a single search
  - each rate associated with a distinct Packet Loss Ratio (PLR) criteria
  - e.g. discovering both
    - Non-Drop Rate (NDR, with PLR=0, zero packet loss) and
    - Partial Drop Rate (PDR, with PLR>0, non-zero packet loss)
  - instead of running separate binary searches for NDR and PDR.
- Execution time is reduced even further by using shorter trial durations in the intermediate steps, with only the final measurements conducted at the specified final trial duration.

Operation (abbreviated version, full version in draft)

- MLRsearch is a duration aware multi-phase multi-rate search algorithm.
- MLRsearch inputs
  maximum_transmit_rate
  minimum_transmit_rate (default value=20kpps)
  final_trial_duration (30sec)
  initial_trial_duration (1sec)
  final_relative_width (0.5%) – determines measurement accuracy
  PDR_packet_loss_ratio (0.5%)
  number_of_intermediate_phases (2)
- Listed default values are specific to FD.io CSIT implementation
- Draft can recommend different default values e.g. (min. 60sec) for final_trial_duration

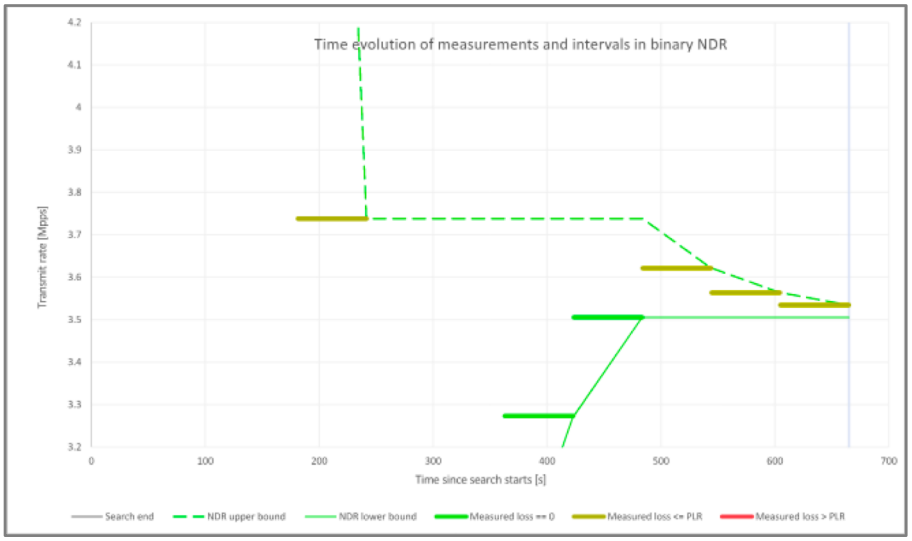
Operation (abbreviated version, full version in draft)

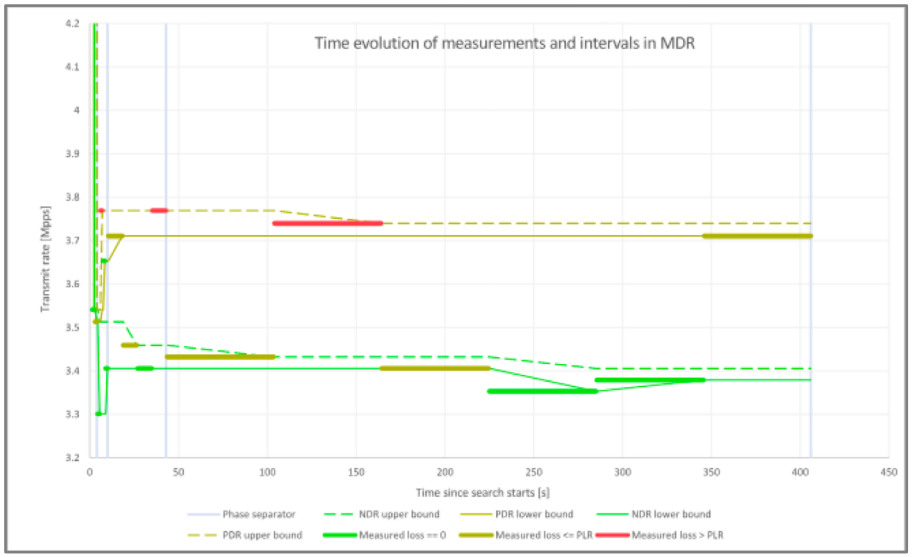
- Initial phase determines starting interval for the search
  - Uses maximum_transmit_rate to discover the Maximum Receive Rate (MRR).
  - Uses MRR as a transmit rate to discover MRR2, used in the first intermediate phase.
- Intermediate phases progress towards defined final search criteria
  - Start with initial_trial_duration and converge geometrically towards final_trial_duration.
  - Start with a large (lower_bound, upper_bound) interval width
    - Track two values, lower_bound and upper_bound, for each required PLR e.g. NDR and PDR.
    - Each value comes from a specific trial measurement.
    - Geometrically converge towards the interval width goal of the phase.
  - Each phase halves the previous width goal.
  - If (lower_bound, upper_bound) interval is
    - valid - use internal search, "binary search" within the valid interval, halving the interval width.
    - invalid - use external search, variant of "exponential search" outside the invalid interval.
- Final phase
  - Executed with final_trial_duration, and final_relative_width that determines resolution of the overall search.
- Outputs
  - (lower_bound, upper_bound) for NDR.
  - (lower_bound, upper_bound) for PDR.

MLRsearch Sample Implementation

- A working implementation of MLRsearch is in Linux Foundation FD.io CSIT project.
  - Used for continuous measurements of NDR and PDR rates of:
    - FD.io VPP
    - DPDK L3fwd
    - DPDK Testpmd
  - Sample throughput results:
    - https://docs.fd.io/csit/rls1901/report/vpp_performance_tests/packet_throughput_graphs/index.html
  - General project info:
    - https://wiki.fd.io/view/CSIT
    - https://git.fd.io/csit/
- MLRsearch Python package published on PyPI:
  - https://pypi.org/project/MLRsearch/

MLRsearch vs. Binary search comparisons





Source: https://docs.fd.io/csit/rls1804/report/vpp_performance_tests/mdr_search.html

MLRsearch vs. Binary search comparisons

DUT: VPP release 18.04
IP4: VPP test - IPv4 routing baseline
Vhost: VPP test - vhostuser VM L2 bridging baseline

Search part of test duration.

Duration+-avgdev [s]	IP4 10s	IP4 30s	IP4 60s	Vhost 10s	Vhost 30s	Vhost 60s
MDR (both intervals)	50.8+-1.2	109.0+-10.0	202.8+-11.7	80.5+-9.0	201.9+-20.6	474.9+-58.2
NDR binary	98.9+-0.1	278.6+-0.1	548.8+-0.1	119.8+-0.1	339.3+-0.1	669.6+-0.2
PDR binary	98.9+-0.1	278.6+-0.1	548.8+-0.1	119.7+-0.1	339.3+-0.1	669.5+-0.1
NDR+PDR sum	197.8+-0.1	557.2+-0.2	1097.6+-0.1	239.5+-0.1	678.7+-0.1	1339.2+-0.1

MDR duration as percentage of NDR duration.

Fraction+-stdev [%]	IP4 10s	IP4 30s	IP4 60s	Vhost 10s	Vhost 30s	Vhost 60s
MDR duration divided by NDR duration	51.4+-1.2	39.1+-3.6	37.0+-2.1	67.2+-7.5	59.5+-6.1	70.9+-8.7

Source: https://docs.fd.io/csit/rls1804/report/vpp_performance_tests/mdr_search.html

Next Steps

- Welcome all review comments
- Draft adoption by BMWG