Rate metrology

Context

To assess the throughput/rate in a deployed network, we can exploit different ways of computing the available link rate. Following RFC recommendations (e.g.: RFC 5136 “Defining Network Capacity, RFC 3148 “A Framework for Defining Empirical Bulk Transfer Capacity Metrics” or RFC “6349 “Framework for TCP Throughput Testing”), we have detailed in RFC ideas some interesting items regarding the ways to evaluate the network capacity.

We summarise below some of the specifications:

1. Repetitive tests are needed as well as different test durations.
2. The rate scenario shall include packets marked with different ToS.
3. The rate scenario must include tests with different packets sizes.
4. The rate scenario must include single- and multiple-TCP-connection throughput tests.
5. The measurements shall be taken on the TCP equilibrium state (as defined in RFC 6349).
6. Follow methodology of RFC 6349.
7. In addition to already available metrics of jobs iperf/nuttcp, it might be interesting to compute metrics like the maximum MTU size allowed by the network (computed by the PMTUd job, the RTT, send/received socket buffer, etc.
8. The rate scenario shall include tests with different rate measurement jobs.

Objective

We recommend to compare at least two of the following OpenBACH jobs (iperf3 and nuttcp), which are based on active rate measurements (i.e. they perform measurements based on their own generated traffic):

- `iperf3` (server or client) generate TCP/UDP traffic and performs different kind of measurements on this traffic. Regarding TCP traffic, it tries to charge the link (depending on the window size) and it is capable of measuring rate (b/s) and data sent (bits). Regarding the UDP traffic, it is possible to specify the bit rate, and it is capable of measuring rate (b/s), data sent (bits), packets sent, jitter (ms), loss and PLR.
- `iperf2` (server or client): uses the version 2 of iperf. The configuration parameters and the metrics are the same of iperf3 job.
- `nuttcp` (server or client): similar methodology and measurement of iperf3.

Regarding the rate metrology, it is also possible to perform passive test with jobs that measure the rate of the traffic generated by other components/jobs, such as the rate monitoring job (based on iptables packets/bits counting). It is recommended to do that for validation purposes, if you are not confident with the metrics shown by iperf3/nuttcp/iperf2.

We have prepared a OpenBACH reference scenario for the rate metrology:

- network_rate that launches iperf3/nuttcp in TCP mode and nuttcp in UDP mode. The scenario allows to modify different traffic parameters on TCP mode (such as MTU size, ToS, number of parallel flows), with a post-processing phase allowing to plot timeseries of the Throughput results per test and the CDF.