Benchmarking Methodology for Stateful NATxy Gateways using RFC 4814 Pseudorandom Port Numbers

draft-ietf-bmwg-benchmaring-stateful

Gábor LENCSE lencse@sze.hu (Széchenyi István University) – presenter
Keiichi SHIMA keiichi.shima@g.softbank.co.jp (SoftBank)

IETF 118, BMWG, November 6, 2023.

Summary of the Proposal

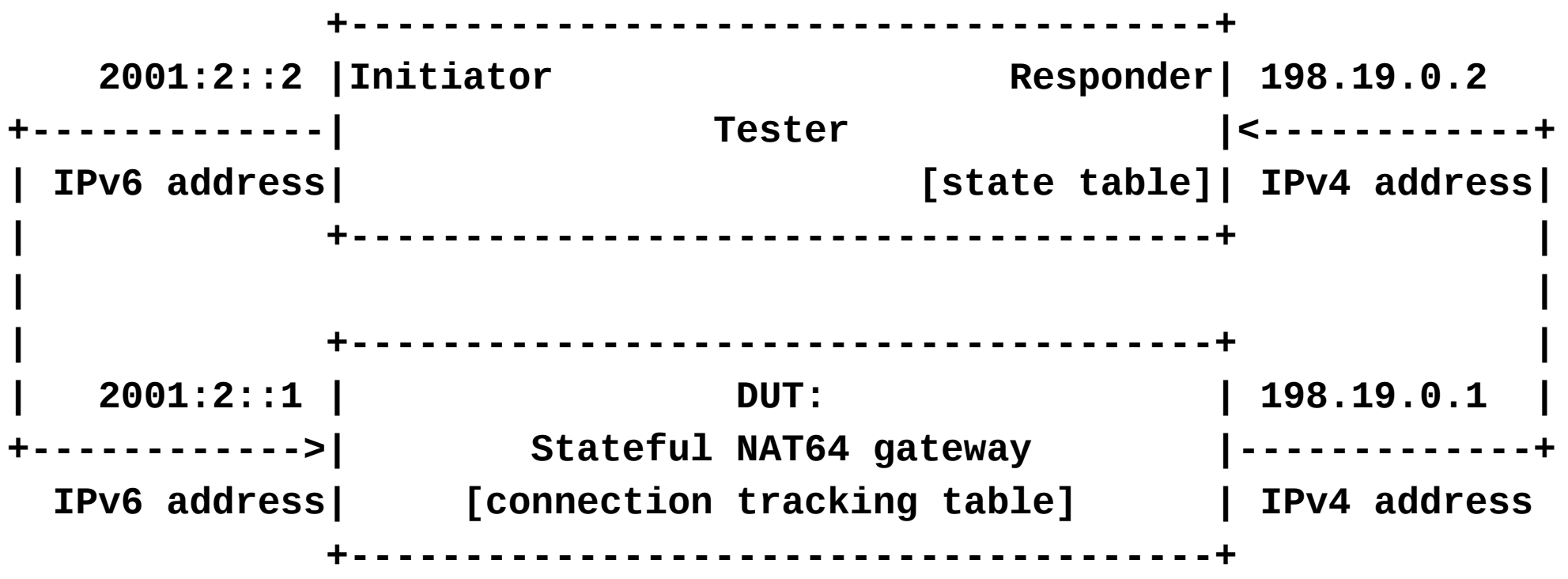
- Guides to achieve reproducible stateful NATxy performance measurements producing meaningful results
  - Facilitating to carry out all the measurement procedures of RFC 2544 / RFC 5180 / RFC 8219 like *throughput*, *latency*, *frame loss rate*, etc. to benchmark stateful NATxy (NAT44, NAT64, etc.) gateways
  - Adding new performance metrics specific to stateful testing:
    - Connection setup performance: *maximum connection establishment rate*
    - Connection tear down performance: *connection tear down rate*
    - Size of the connection tracking table: *connection tracking table capacity*
  - Providing guidelines how to use RFC 4814 pseudorandom port numbers with stateful NATxy gateways

Progress of the draft

- …
- WG draft “02” (Presented at IETF 116)
  - Added: the usage of multiple IP addresses
  - Section 5 (scalability measurements), Section 6 (reporting format)
- WG draft “03” (Presented at IETF 117)
  - Updated the usage of multiple IP addresses to have enough of them
  - Test phases were renamed
- WG draft “04” (version)
  - Added: validation by measurements with 3 different implementations

Reminder: Test Setup

- Methodology works with any IP versions
  - Now, we use the example of stateful NAT64



Reminder: Measurements in two Phases

- Test phase 1
  - It serves two purposes:
    - The connection tracking table of the DUT is filled.
    - The state table of the Responder is filled with valid four tuples.
  - It can be used without test phase 2 to measure the *maximum connection establishment rate*.
- Test phase 2
  - It MUST be preceded by test phase 1.
  - The “classic” measurement procedures (throughput, frame loss rate, latency, PDV, IPDV) are performed as defined in RFC 8219.

Reminder: To support repeatable measurements

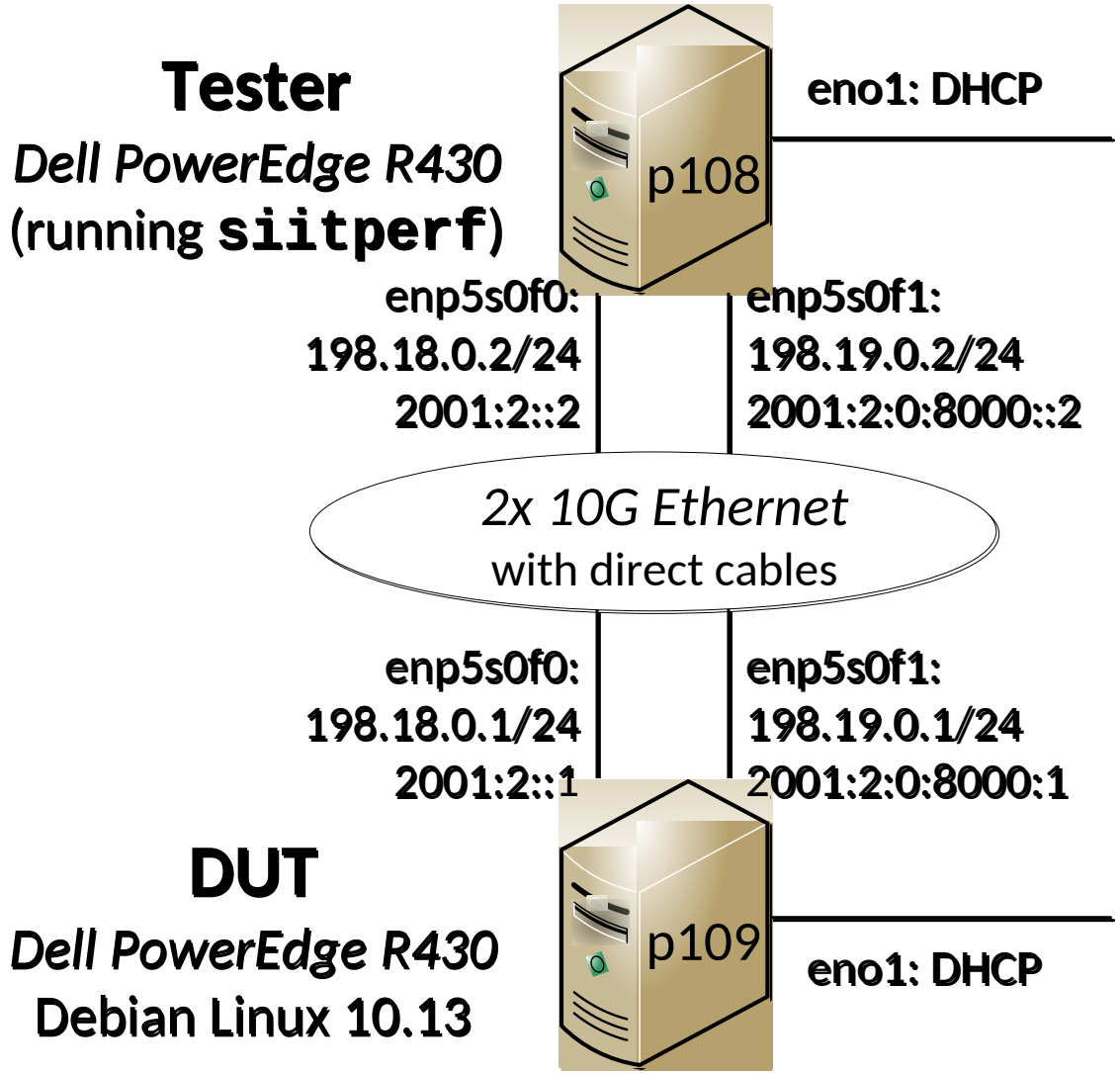
- There are two extreme situations that we can simply ensure
  1. When all test frames create a new connection
     - Ideal for measuring maximum connection establishment rate
  2. When test frames never create a new connection
     - Ideal for the “classic” tests: throughput, latency, frame loss rate, PDV, etc.
- Conditions to achieve them:
  - Large enough and empty connection tracking table for each test
  - Pseudorandom enumeration of all possible port number combinations in test phase 1
  - Properly high timeout value in the DUT

Proof of concept

- All measurements have been implemented in **`siitperf`**
- Source code is available from GitHub (under GPLv3 license): https://github.com/lencsegabor/siitperf
- To validate the methodology, benchmarking tests were performed using three radically different NAT64 implementations
  - Jool (Linux kernel based stateful NAT64)
  - Tayga (Linux user space stateless NAT64) + iptables (stateful NAT44)
  - OpenBSD PF (the firewall solution of OpenBSD)

Measurement Environment

- Dell PowerEdge R430 servers
  - Intel Xeon E5-2683v4 CPUs,
  - 384GB 2400MHz DDR4 RAM
  - Intel 10G dual-port X540 NIC
- Direct cable connections
- Tester: Debian 9.13
- DUT:
  - Debian 10.13
  - OpenBSD 7.1



Sample Results for Jool

- **Maximum Connection Establishment Rate*** of Jool as a Function of the *Number of the Active CPU Cores*, 4M Connections

Num. CPU cores	1	2	4	8	16
Error (cps)	100	200	400	400	400
Median (cps)	208,984	331,835	420,653	454,345	483,153
Minimum (cps)	208,299	324,804	416,747	452,392	472,411
Maximum (cps)	213,768	337,695	423,095	457,275	486,572
Median / prev. median	-	1.59	1.27	1.08	1.06

  - The results show a moderate scalability: significant increase up to 4 CPU cores and slight increase above 4 CPU cores.

**This is a new, stateful-specific performance metric*

Sample Results for Jool

- **Throughput*** of Jool as a Function of the *Number of the Active CPU Cores*, 4M Connections, Bidirectional Traffic

Num. CPU cores	1	2	4	8	16
Error (fps)	200	200	400	400	400
Median (fps)	236,717	371,286	475,780	491,794	497,654
Minimum (fps)	234,178	368,162	473,044	487,108	495,702
Maximum (fps)	237,694	375,194	480,858	495,702	501,952
Median / prev. median	-	1.57	1.28	1.03	1.01

  - The results show a moderate scalability: significant increase up to 4 CPU cores and negligible increase above 4 CPU cores.

**This is a classic RFC 2544 / RFC 8219 performance metric*

Sample Results for Jool

- **Maximum Connection Establishment Rate*** of Jool as a Function of the *Number of Connections*, 16 CPU Cores

Number of connections	400,000	4,000,000	40,000,000
Source port numbers	40,000	40,000	40,000
Destination port numbers	10	100	1,000
Error (cps)	500	400	200
Median (cps)	625,976	483,153	356,445
Minimum (cps)	612,792	472,411	349,804
Maximum (cps)	630,370	486,572	358,397
Median / previous median	-	0.77	0.74

  - The results show an acceptable performance degradation

**This is a new, stateful-specific performance metric*

Sample Results for Jool

- **Throughput*** of Jool as a Function of the *Number of Connections*, Bidirectional Traffic, 16 CPU Cores

Number of connections	400,000	4,000,000	40,000,000
Source port numbers	40,000	40,000	40,000
Destination port numbers	10	100	1,000
Error (fps)	400	400	200
Median (fps)	698,826	497,654	378,513
Minimum (fps)	684,764	495,702	373,240
Maximum (fps)	704,294	501,952	381,834
Median / previous median	-	0.71	0.76

  - The results show an acceptable performance degradation

**This is a classic RFC 2544 / RFC 8219 performance metric*

Sample Results for Jool

- **Latency*** of Jool at 497,654fps cumulated rate of Bidirectional Traffic, 4M Connections, 16 CPU Cores

	Upload latency		Download latency	
	typical	worst case	typical	worst case
Median (ms)	0.0316	0.0923	0.0302	0.0957
Minimum (ms)	0.0307	0.0795	0.0292	0.0847
Maximum (ms)	0.0364	0.1749	0.0351	0.1814

  - The results show low and consistent latency values

**This is a classic RFC 8219 performance metric*

Sample Results for Jool

- **PDV*** of Jool at 497,654fps cumulated rate of Bidirectional Traffic, 4M Connections, 16 CPU Cores

Packet Delay Variation	upload	download
Median (ms)	0.0742	0.0774
Minimum (ms)	0.0576	0.0611
Maximum (ms)	0.7788	0.7826

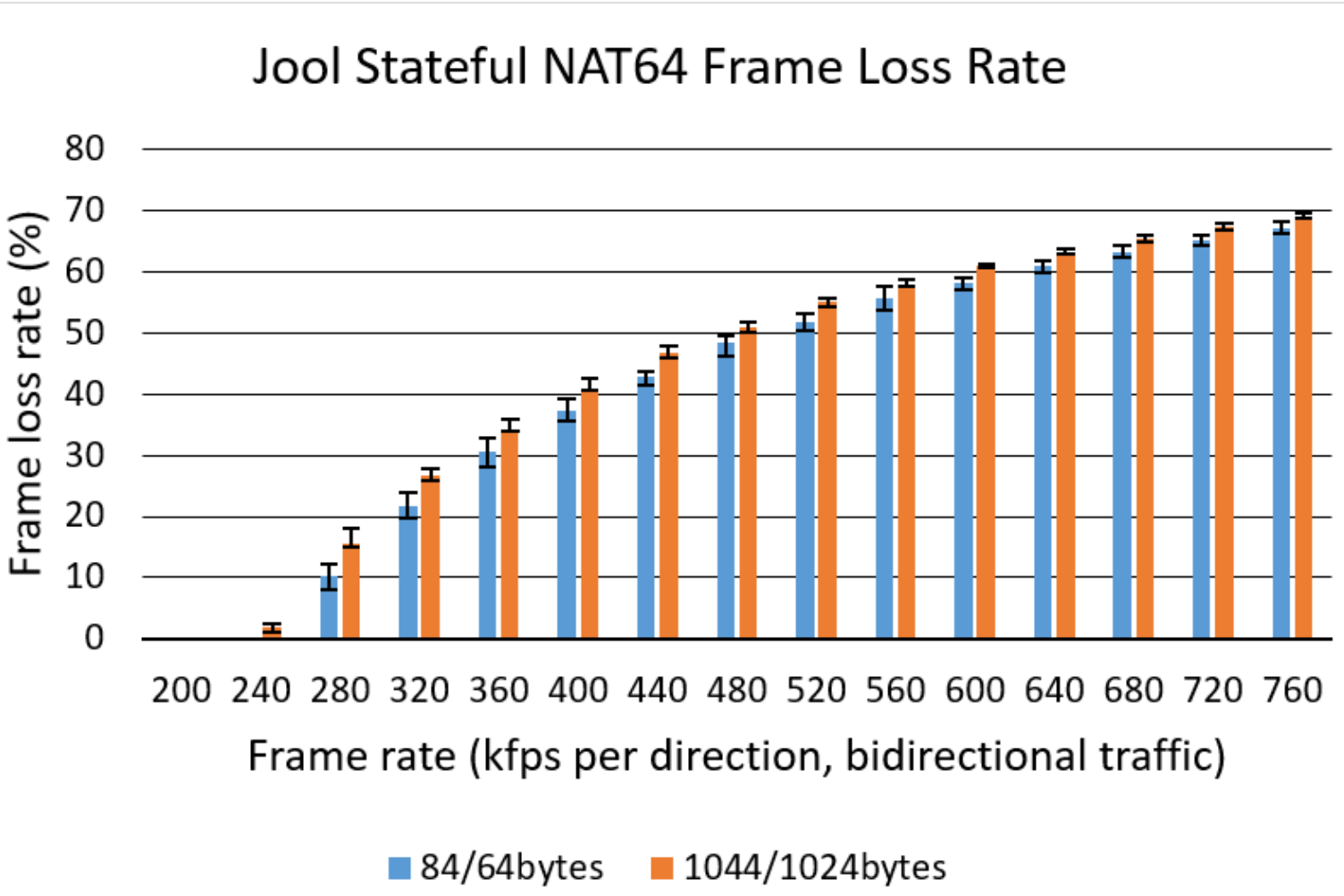
  - The results show low and consistent PDV values

**This is a classic RFC 8219 performance metric*

Sample Results for Jool

- **Frame loss rate*** of the Jool stateful NAT64 implementation as a function of *frame rate* and *frame size* using bidirectional traffic.

**This is a classic RFC 2544/ RFC 8219 performance metric*



Sample Results for Jool

- **Connection Tear Down Rate*** of Jool as a Function of the Number of Connections, 16 CPU Cores (**new, stateful-specific metric*)

Number of connections	400,000	4,000,000	40,000,000
Source port numbers	40,000	40,000	40,000
Destination port numbers	10	100	1,000
Filled table del. time med. (s)	0.46	1.37	11.79
Filled table del. time min. (s)	0.43	1.35	11.76
Filled table del. time max. (s)	0.48	1.39	11.95
Empty table del. time med (s)	0.30	0.30	0.30
Empty table del. time min (s)	0.29	0.29	0.29
Empty table del. time max (s)	0.31	0.31	0.31
Connections deletion time (s)	0.16	1.07	11.49
Connection tear down r. (cps)	(2,758,621)	3,755,869	3,481,288

Further Sample Results

- The results for Tayga+iptables and OpenBSD PF can be found in:

  G. Lencse, K. Shima, K. Cho, "Benchmarking methodology for stateful NAT64 gateways", *Computer Communications*, vol. 210, October 2023, pp. 256-272, DOI: http://doi.org/10.1016/j.comcom.2023.08.009

  (open access paper)

- Acknowledgements

  - This work was supported by the International Exchange Program of the National Institute of Information and Communications Technology (NICT), Japan.
  - The experiments were carried out remotely using the resources of NICT StarBED, 2-12 Asahidai, Nomi-City, Ishikawa 923-1211, Japan.

We would like to ask for WGLC

- Do you have any questions or concerns to be addressed?
- Do you support the progress of this draft to be published as an informational RFC?