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

draft-lencse-bmwg-benchmaring-stateful

Gábor LENCSE lencse@sze.hu (Széchenyi István University)

Keiichi SHIMA keiichi@iijlab.net (IIJ Innovation Institute) – presenter

IETF 113, BMWG, March 23, 2021.

Summary of the Proposal

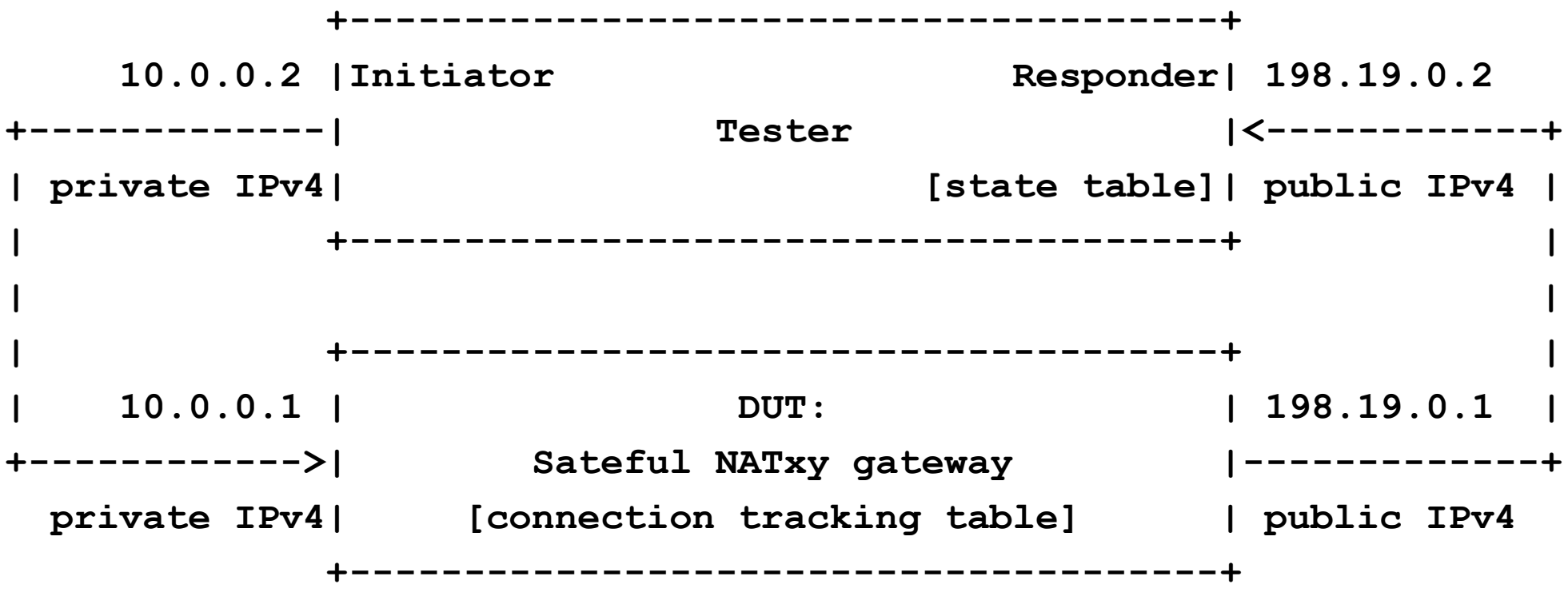
- Guides to achieve reproducible and meaningful stateful NATxy performance measurements
  - Facilitating to carry out all the measurement procedures of RFC 2544 / RFC 5180 / RFC 8219 like *throughput*, *latency*, *frame loss rate*, etc. to benchmark stateful NATxy (NAT44, NAT64, etc.) gateways
  - Adding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

Progress of the draft

- Version 00 (presented at IETF 111)
  - Framework for stateful testing, basics of the methodology
- Version 02 (presented at IETF 112)
  - Refinement of the management of the connection tracking table
    - Benefit: more straightforward measurements method, clear results
    - Also presented measurement results of **`iptables`** stateful NAT44
- Version 03 (the current one)
  - Introduction of connection tear down rate measurement method
  - Measurement results of Jool Stateful NAT64 are also available

Reminder: Test Setup

- Methodology works with any IP versions
  - To facilitate easy understanding, we use the example of stateful NAT44



Reminder: Measurements in two Phases

- Preliminary test phase
  - 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 the real test phase to measure the maximum connection establishment rate.
- Real test phase
  - It MUST be preceded by a preliminary test phase.
  - The actual measurement procedure (throughput, frame loss rate, latency, PDV, IPDV) is 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 all other 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 the preliminary phase
  - Properly high timeout value in the DUT

Connection tear down rate measurements

- Having no better opportunity due to black box testing, we recommend an aggregate measurement:
  - Load *N* number of connections into the connection tracking table of the DUT
    - Performed as a preliminary phase measurement step (without real test phase)
  - Delete the entire content of the connection tracking table of the DUT
    - Using some out of band method, e.g. removal of Linux kernel module

$$Connection\ tear\ down\ rate = \frac{N}{deletion\ time\ of\ the\ connection\ tracking\ table}$$

  - To be measured using different order of magnitude values for *N*

Connection tear down rate measurements

- Technical refinement (not yet in the draft)
  - Subtract the deletion time of an empty table from that of the full table
    - It counts when low number of connections are used
    - It also eliminates the remote command execution overhead
- Potential problem
  - The deletion of a connection due to timeout MAY require a different amount of work than its deletion due to the deletion of the entire content of the connection tracking table
    - And this my depend on the implementation ☹
- Actual measurements:
  - **`iptables`** stateful NAT44, Jool stateful NAT64 implementations

Connection tear down rate measurem. of `iptables`

- The *N* number of connections was set with the source port number destination port number ranges
  - Usually increased fourfold, except the last case (due to memory limit)
  - The hash table size was usually increased proportionally, except the last two cases (due to memory limit)
  - NUMA issue influenced the last measurement
    - The connection tracking table did not fit into the NUMA local memory
  - The connection tear down time of an empty connection tracking table was measured for all cases (and it was indeed significantly different)

Connection tear down rate of **`iptables`** stateful NAT44

num. conn.	1.56M	6.35M	25M	x4	100M	x4	400M	x2	**800M**
src ports	2,500	5,000	10,000		20,000		40,000		**40,000**
dst ports	625	1,250	2,500		5,000		10,000		20,000
conntrack t. s.	2^21	2^23	2^25		2^27		2^29		**2^30**
hash table size	2^21	2^23	2^25	x4	2^27	x2	**2^28**	x1	**2^28**
full cont. t. del t.	4.33	18.05	74.47		305.33		1178.3		2263.1
empty ct. t. del t.	**0.55**	**1.28**	**4.17**	x4	**15.74**	x2	**31.2**	x1	**31.2**
conn. del time	3.78	16.77	70.30	x4	289.59	x4	1147.2	x2	2232.0
conn. tear d. rate	**413.4**	**372.7**	**355.6**		**345.3**		**348.7**		**358.4**

Units: *seconds* for time; *1,000connections/s* for connection tear down rate

Connection tear down rate measurement of Jool

- The *N* number of connections was set with the source port number destination port number ranges
  - Increased fourfold (usually by doubling the size of both ranges)
  - Unlike previously with **`iptables`**, no tuning was done with Jool
  - The connection tear down time of an empty connection tracking table was measured only once (without tuning, there was no difference)

Connection tear down rate of Jool stateful NAT64

num. conn.	1.56M	6.35M	25M	100M	400M	**1600M**
src ports	2,500	5,000	10,000	20,000	40,000	**40,000**
dst ports	625	1,250	2,500	5,000	10,000	**40,000**
full cont. del med	0.87	2.05	7.84	36.38	126.09	474.68
full cont. del min	0.80	2.02	7.80	36.27	125.84	473.20
full cont. del max	0.91	2.09	7.94	36.80	127.54	481.38
emp. ct. del med	*0.46*	*0.46*	*0.46*	*0.46*	*0.46*	*0.46*
conn. del. time	0.41	1.59	7.38	35.92	125.63	474.22
conn. t. d. r. (M)	**3.811**	**3.931**	**3.388**	**2.784**	**3.184**	**3.374**

Units: *seconds* for time; *1,000,000connections/s* for connection tear down rate

Request for feedback

- What do you think of the connection tear down rate measurement method?
  - Does it provide meaningful and reasonable results?
  - Could you recommend a better measurement method?
- Not yet done: measuring the size of the connection tracking table
  - We have ideas that need to be tested how they work in practice
- Is there any other measurement missing?
- Potential WG adoption?