



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

draft-lencse-bmwg-benchmarking-stateful

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

Keiichi SHIMA shima@wide.ad.jp (WIDE Project)

IETF 114, BMWG, July 26, 2022.

Summary of the Proposal

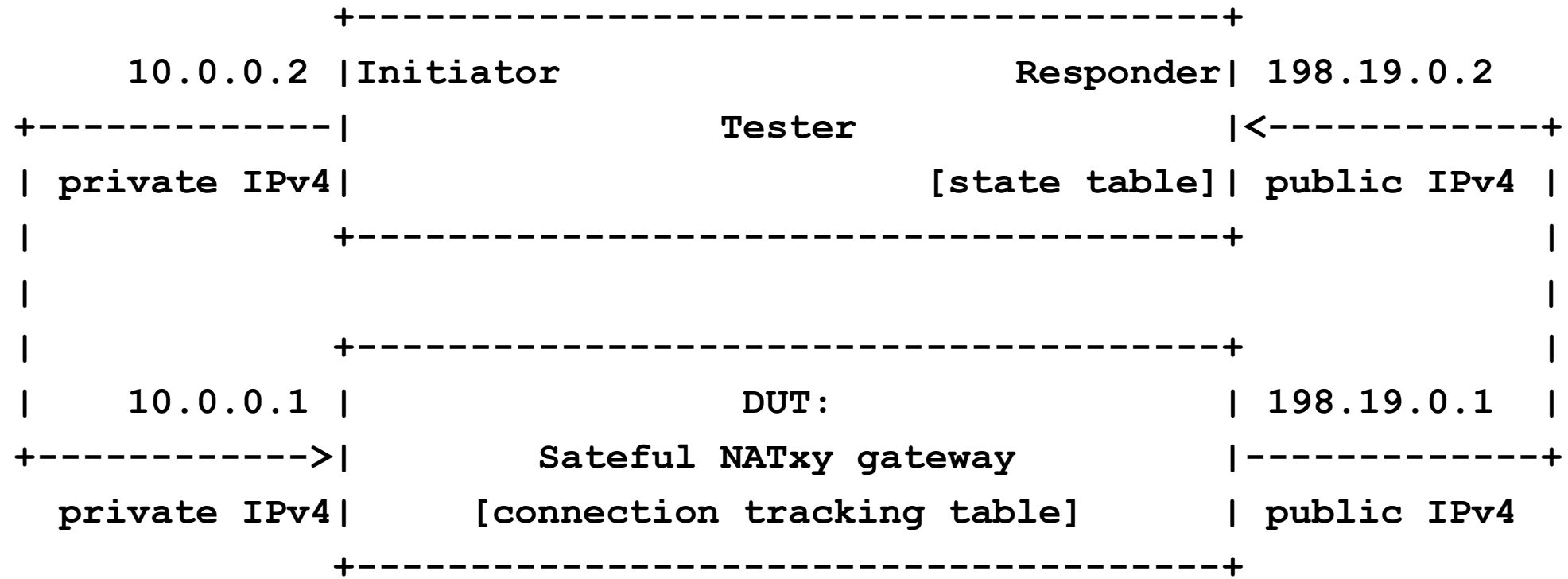
- Guides to achieve reproducible and meaningful stateful NATxy performance measurements
 - Facilitating to carry out all the measurement procedures of RFC 2544 / RFC 5180 / RFC 8219 like *throughput, latency, frame loss rate, etc.* to benchmark stateful NATxy (NAT44, NAT64, etc.) gateways
 - Adding new performance metrics specific to stateful testing:
 - Connection setup performance: *maximum connection establishment rate*
 - Connection tear down performance: *connection tear down rate*
 - Size of the connection tracking table: *connection tracking table capacity (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
- Version 03 (presented at IETF 113)
 - Connection tear down rate measurement method
- Version 04 (the current one)
 - Validation of connection establishment
 - Connection tracking table capacity measurement method

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 “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 the preliminary phase
 - Properly high timeout value in the DUT


Validation of connections establishment


- Maximum connection establishment rate measurement
 - Happens in the preliminary phase
 - The Initiator sends N frames with different four tuples at R rate to the Responder
 - The Responder counts the number of received frames
 - Binary search decision is based on: Did all N frames arrive to the Responder?
 - Problem: arrival of a frame does not prove connection establishment!
 - Solution: connection validation at the real test phase
 - The Responder sends validation frames using all N four tuples of the state table at a low enough rate $r=R*\alpha$ to the Initiator. (alpha is a “safety factor”, e.g. 0.5)
 - The Initiator counts the number of received frames
 - Binary search decision is based on: Did all N frames arrive to the Initiator?

Validated connection establishment rate of Jool NAT64

alpha	0.8	0.6	0.5	0.25	no validation
num. conn.	4,000,000	4,000,000	4,000,000	4,000,000	4,000,000
src ports	4,000	4,000	4,000	4,000	4,000
dst ports	1,000	1,000	1,000	1,000	1,000
num. exp.	10	10	10	10	10
error	100	100	100	100	100
cps median	323,534	429,491	479,296	479,199	478,417
cps min	322,948	426,464	473,339	474,120	474,902
cps max	325,097	431,542	483,690	483,299	484,667

 **degradation**

 **no difference**

 **reference**

Connection tracking table capacity measurement

- Heavily uses the validated conn. est. rate measurement
- It comprises two steps:
 - Exponential search for the order of magnitude of the size of the connection tracking table
 - Final binary search for the exact size of the connection tracking table
- It uses a few empirical parameters specific to the examined NATxy gateway implementation
 - alpha: “safety factor” for the R validated conn. est. rate measurement
 - beta: unacceptable drop of R during the exponential search (e.g. 0.1)
 - gamma: unacceptable drop of R during the final binary search (e.g. 0.5)

Exponential search algorithm for the range

```
// Variables:
//   C0 and R0 are beginning safe values for connection tracking table
//   size and connection establishment rate, respectively
//   CS and RS are their currently used safe values
//   CT and RT are their values for current examination
//   beta is a factor expressing unacceptable drop of R (e.g. 0.1)
R0=binary_search_for_maximum_connection_establishment_rate(C0,maxrate);
for ( CS=C0, RS=R0; 1; CS=CT, RS=RT )
{
    CT=2*CS;
    RT=binary_search_for_maximum_connection_establishment_rate(CT,RS);
    if ( DUT_collapsed || RT < RS*beta )
        break;
}
// here the size of the connection tracking table is between CS and CT
```

Final binary search algorithm for the exact value

```
// Variables:
//   CS and RS are the safe values for connection tracking table size
//   and connection establishment rate, respectively
//   C and R are the values for current examination
//   gamma is a factor expressing unacceptable drop of R (e.g. 0.5)
for ( D=CT-CS; D>E; D=CT-CS )
{
    C=(CS+CT)/2;
    R=binary_search_for_maximum_connection_establishment_rate(C,RS);
    if ( DUT_collapsed || R < RS*gamma)
        CT=C; // take the lower half of the interval
    else
        CS=C,RS=R; // take the upper half of the interval
}
// here the size of the connection tracking table is CS within E error
```

Connection tracking table capacity measurement

- The theoretical considerations and all the details can be found in Section 4.9 of the draft:

<https://datatracker.ietf.org/doc/html/draft-lencse-bmwg-benchmarking-stateful-04#section-4.9>

- A sample measurement script can be found a GitHub:

<https://github.com/lencsegabor/siitperf/blob/stateful/scripts/cttc.sh>

– It contains the optimization mentioned at the end of Section 4.9

Sample measurement results for `iptables`

- Parameters:

`hashsize=nf_conntrack_max=222=4,194,304;`

`R0=1,000,000; E=1; alpha=1.0; beta=0.2; gamma=0.4.`

Experiments were performed 10 times

[CS, CT] is the range found by the exponential search, C is the final result

	C0	R0	CS	RS	CT	C
median	1,000,000	2,562,500	4,000,000	2,250,945	8,000,000	4,194,301
min	1,000,000	2,437,500	4,000,000	2,139,953	8,000,000	4,194,300
max	1,000,000	2,687,500	4,000,000	2,327,269	8,000,000	4,194,302

Request for feedback

- What do you think of the validated connection establishment rate and connection tracking table capacity measurement methods?
 - Do they provide meaningful and reasonable results?
 - Could you recommend better measurement methods?
- Is there any performance metric or measurement missing?
- What do you think of WG adoption?