**Developing a Software Defined CG-NAT**

Denis Plotnikov

<INTRO>

<WHAT IS NAT>

RFC

<WHAT IS CARRIER GRADE NAT>

RFC

<WHAT IS THE NOWADAYS APPROACH TO DO NAT>

Specific network devices

<WHAT ARE THEIR PRICES>

My doc about prices

<HOW CAN WE DO BETTER>

Algorithmic + technology

… let’s try to make one

<WHAT OUR NAT SHOULD DO>

<WHAT THE NAT KEY METRICS ARE>

<WHAT ITS PERFORMANCE SHOULD BE>

Our goal is to develop a working prototype of software defined carrier-grade network address translator (SD CG-NAT). To make sure that our SD CG-NAT is close to reality in terms of performance it is necessary to define the performance metrics and set their values. In order to get those metrics, a couple of sources are used. The first one is Rostelecom technical requirements to CG-NAT [ref\_TT\_ROS\_TEL]. The second one is the performance specification claimed by one of the on-market available NAT device producers which employ the same approach as this research does: **to use not task specific computer (a commodity server) to make a network specific solution using a mix of algorithmic and technological approaches. (our\_approach)**

<WHAT ARE THE NAT KEY METRICS>

The key characteristics are:

***Packets processing rate*** – (packets per second [PPS])

***Concurrent session support*** *–* (number)

***Connections setups rate*** – (connection setups per second [csps]) the number of new NAT records to be created in a second. This

***Throughput – (***bit per second[bps]***) – isn’t very important because mostly defined with NIC (if NIC is slow – than it’s a bottle neck if pps is low that router is bottle neck )***

<WHAT ARE THE METRICS VALUES>

<OUR NAT>

1. What the NAT should do (RFC tra-la-la)

2. What is the core functionality (CHECK SUM etc.)

3. What to achieve - target characteristics - rostelecom, rdp.ru

4. How to achieve - what is the main problem - the main problem is lookup data structure

5. Plan on what to test: linear, tree, tree-array, rb-tree(balanced) array, hash, parallel

6. Test program description (what is the bottle neck, what does it do, how much does it cost)

7. Performance comparison (packets per second, memory usage, limitation)

7.1 Base line

7.2 Linear

7.3 Tree

7.3.1 Tree-tree

7.3.2 Tree - plain array

7.3.3 RB-tree - plain array

7.4 Hash+array

7.5 Parallel hash+array

7.5.1 Size

7.5.2 Cores

8. Summary

**REFERENCES**

[ref\_TT\_ROS\_TEL] file:TT CGNAT 2014\_26\_06v1.doc

[ref\_RDP.RU] http://rdp.ru/