**Developing a Software Defined CG-NAT**

Denis Plotnikov

<INTRO>

<WHAT IS NAT WHAT SHOULD IT DO (TYPES OF NAT and corresponding RFC overview)>

RFC

<WHAT IS CARRIER GRADE NAT, WHAT DOES IT DO, WHO CEARS>

RFC

<WHAT IS THE NOWADAYS APPROACH TO DO NAT>

Specific network devices

<WHY THIS APPROACH ISN’T GOOD: drawback of specialized NAT devices>

<WHAT ARE THEIR PRICES – the drawback #1>

My doc about prices with graph goes here

<HOW CAN WE DO BETTER>

Algorithmic(data structure) + technology(software/hardware architecture approaches (batching processing, NIC RSS queues, multithreading, locklessness, DDIO ) -> consequence DPDK already have it so let’s choose it)

<… 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[ref\_RDP.RU] 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]) – the router’s maximum rate of packet processing. This is the main metric describing the packet processing abilities of a NAT device.
* ***Concurrent session support*** *–* (number) – the maximum number of sessions produced by served network. It describes the maximum network size which can be served by the NAT device. As described later in this document than bigger the network than harder to maintain translations to its nodes.
* ***Connections setups rate*** – (connection setups per second [csps]) the number of new NAT records to be created in a second. This metric shows the NAT ability to create new NAT records and could be a drawback of the NAT device in a certain modes of network work like when the networks nodes start creating of new connections actively, for example in the beginning working hours
* ***Throughput – (***bit per second[bps]***) –*** it isn’t very clear metric of the NAT device because it is mostly defined with NIC (network interface card) performance used by NAT device. If the NAT device won’t have enough of packet processing rate its throughput can’t achieve the maximum throughput provided with NIC and vice versa. The main sense of having it in the metric list is to make sure that NAT device is able to transfer needed amount of information.

***Cycles per packet***

<WHAT ARE THE METRICS VALUES>

Based on the sources of information the performance requirements of the NAT device are set following:

1. Packet processing rate: 5.5 Mpps
2. Concurrent session support: 65.5M (a B-class network with up to 1000 ports dedicated to each node)
3. Connection setups rate: 3 Mcsps
4. Throughput: 10 Gbps

<MY TEST NAT DESCRIPTION>

Functionality (CHECK SUM etc.)

==What I’ve included in my NAT and why==

==Investigation the base line performance of my NAT==

==How much the NAT spends on what==

Add graph comparision of w/o check\_sum, w/o timestamps

==What is the NAT main problem -> look up table==

==Why is it so -> prove it by adding linear search to NAT==

Add chart of NAT performance with linear search for num\_of\_nodes [100, 250, 500, 1000, 1500, 2000, 2500, 5000 ]

<DATA STRUCTURES OVERVIEW>

1. tree
2. tree-array
3. rb-tree(balanced) array
4. hash
5. parallel

<PERFORMANCE OF EXPLORED APPROACHES>

(packets per second, memory usage, limitations)

1. Base line
2. Linear
3. Tree
   * 1. Tree-tree
     2. Tree - plain array
     3. RB-tree - plain array
4. Hash+array
5. Parallel hash+array
   * 1. Size
     2. Cores

Summary

**REFERENCES**

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

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