

NATApp Plugin Proposal

- [Name](#)
- [Repo Name](#)
- [Description](#)
 - [Benefits of NAT implementation](#)
 - [Methods of NAT](#)
 - [UseCase: Network Port Address Translation with ODL in Enterprise Networks](#)
 - [NAT Plugin Interactions with other ODL modules:](#)
 - [NAT App Service Architecture](#)
- [Scope](#)
- [Resources Committed \(developers committed to working\)](#)
- [Initial Committers](#)
- [Vendor Neutral](#)
- [Meets Board Policy \(including IPR\)](#)

Name

NAT Application Plugin

Repo Name

natapp

Description

NAT functionality is one of the basic features of any enterprise network by which the local IP addresses are translated into global IP addresses and viceversa for Internet connectivity. This is traditionally done by the Gateway router in the enterprise network which is connected to the Service Provider network. In the SDN scenario, this functionality in the router has to be hosted as an application on the SDN controller (just like FW application).

Benefits of NAT implementation

- NAT conserves public Internet address space
- NAT facilitates network migration from one address space to another
- Enables connecting IoT devices with IPv6 addresses over IPv4 networks
- NAT simplifies routing
- Enhances security by masking the private IP addresses and restricting connections

Methods of NAT

- One-to-one NAT (Static NAT)
- One-to-many NAT (Dynamic NAT)
- Network Address Port Translation (NAPT) or Port Address Translation (PAT)
- Full cone NAT
- (Address)-restricted-cone NAT
- Port-restricted cone NAT
- NAT loopback (Hair pinning)
- Stateful and Stateless NAT64 for Address Family Translation (AFT) with IPv6 to IPv4 conversion

UseCase: Network Port Address Translation with ODL in Enterprise Networks

In the present project proposal we intend to bring in the NAT functionality as a plugin (NATApp plugin) at the ODL controller. Whenever an enterprise user connects to the Internet with his local IP address, the NATApp plugin on ODL will translate that local IP address with a global IP address and install OpenFlow rules accordingly on the switch. Nat App installs two flow rules; one for outgoing packet from and another for incoming to Enterprise network. The first rule will modify the source IP address from local to global IP address and the second rule will modify the destination IP address from global IP to local IP address.

[blocked URL](#)

NAT Plugin Interactions with other ODL modules:

- NAT Application shall be created as a modular plugin (OSGI bundle) using the MD-SAL approach
- NAT App registers itself in the Service Abstraction Layer (SAL) registry for the consumption of services.
- NAT App shall listen at SAL for OpenFlow based message notifications in network.
- Create the Yang Data Store and RPCs
- Expose REST API's for North Bound.
- NAT App offers a solution in the form of L3 plugin.
- NAT App shall use SQL database for global IP addresses and the mapping between global and local IP addresses.

NAT App Service Architecture

[blocked URL](#)

Scope

The scope of this project is mostly about implementing Static NAT, Dynamic NAT, Network Address Port Translation and other NAT mechanisms mentioned above.

Resources Committed (developers committed to working)

- Thomas Lee S (thomaslee.s@[tcs.com](mailto:thomaslee.s@tcs.com)) [gerrit id: thomaslees]
- Swetha Singadi (swetha.s8@[tcs.com](mailto:swetha.s8@tcs.com)) [gerrit id: Swetha]
- Anil Sonnad (anil.sonnad@[tcs.com](mailto:anil.sonnad@tcs.com)) [gerrit id: Anilsonnad]

Initial Committers

- Thomas Lee S (thomaslee.s@[tcs.com](mailto:thomaslee.s@tcs.com)) [gerrit id: thomaslees]
- Swetha Singadi (swetha.s8@[tcs.com](mailto:swetha.s8@tcs.com)) [gerrit id: Swetha]
- Anil Sonnad (anil.sonnad@[tcs.com](mailto:anil.sonnad@tcs.com)) [gerrit id: Anilsonnad]

Vendor Neutral

The project is made from scratch, no vendor code, logos nor is anything included.

Meets Board Policy (including IPR)

New Project. No Inbound Code Review required