

Introduction

NOC is an Operation Support System (OSS) for the Network's Operation Centers (NOC). Primary users of NOC are the telecommunication companies, service providers and enterprise networks. NOC primary goal is to be a glue getting various operation processes together.

Areas covered by NOC include:

- Multi-VRF address space management: Full control over VRF, IPv4 block and address allocation and usage.
- Virtual circuits management: VLANs can be created and destroyed directly from NOC. All changes propagate to network equipment using defined policies.
- Service activation and provisioning: Interaction with network equipment to automate routine tasks like inventory, port activation, etc.
- Configuration management: Control over current configuration and changes of network equipment configs, DNS zones, prefix lists and RPSL objects
- DNS provisioning: DNS zone management, generation and provisioning to DNS servers according to defined policy
- Peering management: BGP peer inventory, various RPSL objects generation and provisioning to RIR's databases
- Fault management: Collect and process syslog messages and SNMP traps from equipment, maintain lifecycle of event, reduce amount of seen events to those which really need attention
- Performance management: Collect and process various performance data to analyze network trends.
- Knowledge base: wiki-like knowledge base to share the knowledge, store documentation, scans and charts.

NOC is Open Source Software distributed under the terms of BSD-like [LICENSE](#).

Key Features

Open Source

NOC is Open Source product distributed under the terms of BSD [LICENSE](#). You are free to use and modify NOC according to your needs without any restriction, though your feedback is always welcome.

Flexibility and Portability

- NOC mostly implemented in python language and available on all major telco platforms: [Solaris](#), [FreeBSD](#), Linux, [Mac OS X](#), [CentOS](#)
- [PostgreSQL](#) and [MongoDB](#) databases backend are available on all platforms supported by NOC

Reliability

- Integrated monitoring and self-healing capabilities allow to recover from error conditions

Scalability

- NOC major components are separated to the several daemons
- Number of daemons and number of CPU cores can be increased to serve growing load
- [PostgreSQL](#) database backend has good scalability and offers number of options to offer even more scalability
- NOC has been reported to process and serve over 1,000 units of equipment, over 10,000 allocated IPv4 blocks and addresses, over 500 autogenerated DNS zones, over 100,000 fault management events per day on single CPU server.

Security

- Privilege separation for all major components for collaborative use
- Record-level permissions
- SSL/TLS data encryption of critical communications

Web Interface

- Neat Django-based web interface
- All major operation and configuration can be performed via web interface
- Global search over all types of objects
- Custom tags on database objects

Address Space Management

- Multi-VRF address space management
- Nested address block allocations
- Used IP address database
- Address space usage reports
- Allocated and free blocks reports
- Direct import of IP addresses via DNS zone transfer or from CSV file
- IP Address Ranges with auto-generated FQDNs and classless reverse zone delegations

DNS Management

- Forward and Reverse zone generator
- Pluggable zone generator interface (BINDv9 supported)
- Web interface for zone creating and modification
- Address Space Management integration
- DNS server provisioning
- Web interface to distribute load between nameservers
- Transparent support for classless IN-ADDR.ARPA delegation (RFC2317)
- Domain expiration checking with registrar's databases synchronization via whois

Service Activation

- Interacts with [wide range of equipment](#)
- Flexible framework to quickly write additional equipment support (Sometime less then 2 minutes)
- Service activation application framework
- Collect SNMP traps and syslog messages from objects
- Multiple activator support
- Activators can reside in different overlapping address spaces
- Activators can work from behind NAT
- Experimental L2 topology discovery

Configuration Management

- Uses Service Activation subsystem
- Grabs configuration from wide range of equipment (See [Device Support](#))
- Configuration stored on version control system
- Any revision of configuration accessible via web interface
- Flexible notification on configuration changes
- Forced re-read of configuration on specific SNMP traps or syslog messages
- Pluggable VCS interface (Mercurial, CVS, etc)

Virtual Circuit Management

- Database of VC identifiers and tags
- 802.1Q VLANs, 802.1ad Q-in-Q VLAN stacks, FrameRelay DLCIs, MPLS label stacks, ATM VPI/VCI, X.25 logical groups/logical channels are supported
- Direct import of existing VLANs from equipment
- Flexible VLAN Provisioning with automatically VLAN roll-out from the database to network equipment

Fault Management

- Uses Service Activation subsystem
- Collects events from managed objects (SNMP Trap, Syslog)
- Performs event classification, correlation and root-cause analysis (Experimental)

Performance Management

- Extendable with plugins
- Distributed polling
- Scalable beyond thousands probes per second
- SVG/AJAX interface for data analysis
- Flexible thresholds verification
- Integrated with Fault Management
- Time series are stored in database

Peering Management

- Database of BGP peers
- Database of ASes and AS-SETS
- RPSL generator
- Prefix list generator
- Automatic RIPE database updates

Knowledge Base

- Built-in category management
- Pluggable markup syntax (Creole, CSV, Plain-Text)
- Unlimited number of attachments per article
- Attachments are stored inside database for integrity and easy backup
- Multi-Lingual support
- Global and user bookmarks
- IT-oriented set of macros

Features requested

See the `list of requested features <<http://bt.nocproject.org/issues/>>`

A Brief History

NOC starts its way in Effortel Russia in late 2007 as telephone number and IP address space database and was quickly recognized as a single information storage. DNS provisioning appears shortly after. After an year of internal usage NOC was released as an open source project. Then [Badoo Development](#) and [Innova Group](#) adopts NOC and the project became to gain momentum.