## Thesis overview

# Netconf Element Management System – Design and Implementation

**Martín Bertolina**
**Universidad Nacional de La Plata, Fac. de Informática, July 2012**
**mbertolina@hotmail.com**

## Background

Communications networks grow in the same way new technologies are developed, followed by the demand from end users for a better quality of service and higher speeds, these represent as challenges for Telco carriers. As a result, operators are forced to increase the number of network devices to meet those needs and implement new solutions to manage and maintain their configuration.

As SNMP provides limited functionalities to those activities related to network configuration management, CLIs scripts automation and proprietary protocols have become easy solutions to exchange information between devices, but the main disadvantages are extended development times, high maintenance costs and the lack of interoperability between equipment from different vendors. Network configuration protocol (Netconf) was introduced as a result to overcome those challenges and provide a standard framework to communicate with network devices for these purposes.

The Network Configuration Protocol (Netconf) is an IETF network management protocol using an RPC-based communication model. It was developed in the Netconf working group and published in December 2006 as RFC 4741 [1] and later revised in June 2011 (RFC 6241) [2]. This protocol defines a simple mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. The protocol allows the device to expose a full and formal application programming interface (API). Applications can use this straightforward API to send and receive full and partial configuration data sets.

The NETCONF Working Group has produced a protocol suitable for network configuration, with the following characteristics [3]:

- Provides retrieval mechanisms which can differentiate between configuration data and non-configuration data.
- Is extensible enough so that vendors can provide access to all configuration data on the device using a single protocol.
- Has a programmatic interface (avoids screen scraping and formatting-related changes between releases).
- Uses an XML-based data representation, which can be easily manipulated using non-specialized XML manipulation tools.
- Supports integration with existing user authentication methods.

- Supports integration with existing configuration database systems.
- Supports multiple (e.g. candidate and running) data-stores to optimize configuration preparation and activation.
- Supports network wide configuration transactions (with features such as locking and rollback capability).
- Runs over a secure transport; SSH is mandatory to implement while TLS, BEEP, and SOAP are optional transports.
- Provides support for asynchronous notifications.

## Objective

This work proposed to study and present de new Network Configuration Protocol and its deep relationship with device management activities. The main objectives of the thesis were to design and implement a management server, called NEMS (Network Element Management System), to manipulate device's configuration on those with Netconf protocol support.

## Implementation

Netconf Element Management System (NEMS) is a management server entirely programmed in Python to easily edit, remove and add new configuration parameters on those devices with Netconf protocol support. It provides a multi-operator and friendly environment to perform basic operations over Netconf datastores. The Figure 1 shows the basic architecture implemented for the proposed solution, its components and relationship among them.

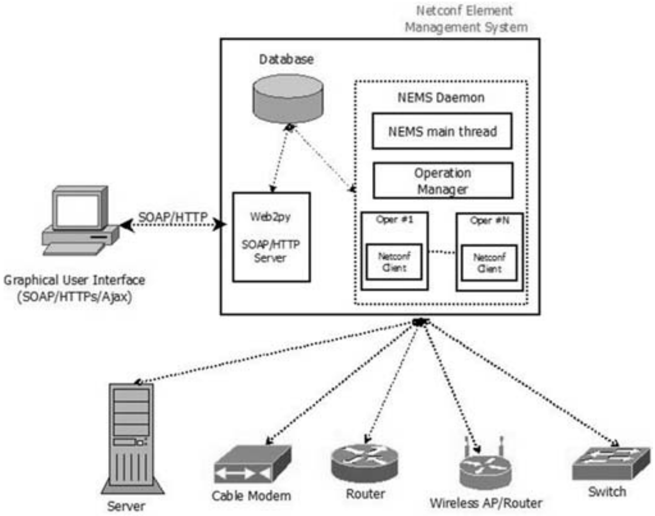

**Figure 1 NEMS Implementation**

The GUI was developed to be accessed through the web using an HTTP/SOAP server (based on Web2py [4], Javascript/JQuery and SOAP over AJAX) with secure HTTPS connection and provides two levels of users, Administrators for read/write permissions and Operators for just read access. Web2py is a web framework for rapid development of fast, scalable, secure and portable database-driven web-based applications. One of the advantages of Web2py is the fact that can be configured to be deployed with Apache, Cherokee or Lighttpd servers through Web Server Gateway Interface (WSGI) modules.

The main thread manages the whole system, starts, terminates and monitors (in case of failure) Web2py environment as well as the Operation Manager. It also loads and parses the configuration file and creates the loggers to be used by all components to log server's events for debug purposes.

The operation manager is in charge of the database connection establishment to be shared with operations threads. It receives the notifications from the database each time an operation changes its state, and handles those accordingly (creates, executes, cancels and terminates operation threads).

Operations thread is a module based on Ncclient API [5] (compliant with Netconf RFC 4741), exchanges messages encoded in Netconf RPC with the Netconf module running in the device. It establishes, maintains and terminates Netconf sessions. NEMS supports the following operations: OP_COPY (to copy one datastore to another), OP_DELETE (to delete all configuration loaded in the specified datastore), OP_REQ_CURRENT_CONF (to request current configuration running in the target datastore), OP_EDIT_CONFIG (to edit the configuration to be pushed to de device). More than one Netconf operations are used in conjunction with each other to run an NEMS operation.

The XML device configuration validation and processed by each operation thread. All configuration changes are first validated by the server before being sent to the device in order to verify the correct structure is in place and it is supported by the device. This validation uses configuration schemas (Relax NG, Schematron and DSDL) and yang2dsdl [6] library for processing. It also saves the device configuration in the database (Postgresql 9.x).

## Conclusion

This work was based on the design and implementation of Network Element Management System (NEMS) multi-user server to manipulate the configuration on those devices with Netconf protocol support, providing a friendly framework to perform changes datastores.

All NEMS server functionalities were successfully validated using Netconfd Netconf server, compliant with Netconf standard, distributed with YUMA tools libraries [7] (YANG-Based Unified Modular Automation). The code is available at Google Code [8].

**Martín Bertolina**
mbertolina@hotmail.com

## References

[1] Enns, R., "NETCONF Configuration Protocol",RFC 4741, December 2006. http://www.ietf.org/rfc/rfc4741.txt

[2] R. Enns and M. Bjorklund and J. Schoenwaelder and A. Bierman, “Network Configuration Protocol (NETCONF)”, RFC 6241, June 2011. http://www.ietf.org/rfc/rfc6241.txt

[3] IETF, Network Configuration (netconf) Working Group. http://datatracker.ietf.org/wg/netconf/

[4] Massimo Di Pierro. Web2py. 2007. URL http://www.web2py.com.

[5] Shikhar Bhushan. Ncclient Python library for netconf clients. 2009. URL http://oss.schmizz.net/ncclient/

[6] Pyang. URL http://code.google.com/p/pyang/.
[7] Yuma tools. URL http://www.netconfcentral.org/yuma.
[8] Network Element Management System. http://code.google.com/p/netconf-element-management-system/.