

attente de la proposition

Eric Lunaud Ngoupé and Sylvain Hallé and Omar Cherkaoui
Université du Québec à Chicoutimi, Canada
Email: eric.ngoupe-lunaud@uqac.ca, shalle@acm.org

Omar Cherkaoui
Université du Québec à Montréal, Canada
Email: Cherkaoui.Omar@uqam.ca

Abstract—abstract ici

Index Terms—router; netconf; yang; network management

I. INTRODUCTION

Computer networks are becoming more complex because of their growing capacity as well as the density of services they deploy. This complexity carries all the problems related to network administration. These problems must be solved to meet the needs of stability of existing networks, while integrating new services within an area where the growth of different networks does not follow with the supply of qualified network engineers REF:6244/2011. we know that configuration services on heterogeneous devices is a major factor in rising costs in provider networks [2]. More over having multiple manufacturers tend to worsen the case. Network device manufacturers most of the time develop solutions only for their own equipments. Router configurations assessment in a small sized networks is a simple task that can increase in complexity as the the size of the network and node increases, for network administrators. It will then be suitable for network administrator to have a solution that allows them to automatically recover part or the whole configuration file for validation or modification. Some available tools can recover the configuration of a specific device but there is still no available solution to browse the network while recovering some or all of the configuration files of each node.

The objective of this paper is to recover certain device configurations depending on defined criteria, in order to validate them using the validation implemented in ValidMaker tool appeared in the article [3]. We will base our work on the results obtained by (). Based on some defined criteria we will recover some or all equipment's configuration in a network, and for that reason we will use Netconf's operation get-copy . In the article [4] the authors have implements with the control 'Validate' of Netconf a model that consist to valid syntactically and semantically the configuration before pushing it on the Equipment. But what is there with the configuration that are already on the Equipment and that need to be verify also. This is to respond at this problematic that we offer you this article. Our solution To implant our solution, we consider the network like a tree (fig1) with branches and leaves. We will go through into this tree, branches after branches and leaves after leaves and recover the configuration of the equipment (router) regarded as nodes and this recovery will be based on the parameter given in entry. After this recovery these

configurations will be validated by following the model of the article [3]. We know that the nodes are identified by an IP address and the name and each nodes possesses an IP address of each nodes (leaves) associated to it. And to browse the network just reach a node so that it gives us a list of each nodes that are associated with it. To do this we will use Yang to define our tree then Netconf will browse and finally validMaker will validate the configurations.

NETCONF provides mechanisms to install, manipulate, and delete the configuration of network. By implementing some Netconf's operation like edit-config, get-config operations, we can manipulate a part or an entire devise's configuration of each network device [5]. This paper is structured as follows. In Section II we present related work and motivation the need for recuperation of router's configuration and validation methodology. In section III we present our solution by implementing the validation of a part of router's configuration into a network by using Edit-config Netconf operation. Section IV concludes and announces future work.

II. NETCONF & YANG

NETCONF

In recent years, several solutions have been developed in order to improve and correct the gaps in networks administration and configuration. These solutions were either partially adapted, specific to each manufacturer or simply directed to one particular aspect of the problem. Things started to change when the NETCONF was established. Not only NETCONF allows transactions, restoration, reducing implementation costs it also has the ability to save and restore configuration data equipment [6]. Further more NETCONF allows to distinguish between configuration data and other types of data [1]. It provides a set of diagnostic operations, handling and removal of network equipment configuration [6] [7]. According to [8]. it is an administration protocol that can provide a ways to perform network administration through equipment configuration and flexible hierarchical encoding mechanism that is XML (Milung et al., 2004) [4], [7] . NETCONF is based on XML to encode both protocol messages and configuration data on a model called YANG [9]. This allows, for example, a user can send NETCONF protocol a Juniper router would send the same commands to a Cisco router: we talk of independence seller (Hall et al, 2004). [5]

YANG

YANG is a modeling language used in the design of data used in the configuration and those used in the design of

the for NETCONF configuration protocol network. YANG classifies data as a tree in which each node has a name and a value or a set of subsequent nodes. YANG shows a clear description of the nodes and the interaction between these nodes [9]. Data in YANG Model are structured into modules and sub-modules. A set of Data can be imported from other external modules and included in sub-modules. YANG models are designed to be easily used by NETCONF. Once published, the YANG module acts as a contract between the client and the server and each party knows exactly what to expect. A customer knows how to create valid data for the server, and also knows what data will be sent to the server. The server in turn knows the constraints governing the data and how they should behave [1]. According to [9] the complexity of the hierarchy can be increased, allowing a module to add data nodes to the hierarchy defined in another module. This increase may be conditional, with new nodes appearing only if certain conditions are met, which makes YANG an extensible language. This flexibility allows both systems NETCONF and YANG to communicate and enables interoperability between both systems [9].

III. CONCLUSION

REFERENCES

- [1] P. Shafer, "An architecture for network management using netconf and yang," pp. 17–30, 2011.
- [2] *Configuration Management at Massive Scale: System Design and Experience*, 2007.
- [3] NOMS, Ed., *ValidMaker: A Tool for Managing Device Configurations Using Logical Constraints*. Network Operations and Management Symposium (NOMS), 2012.
- [4] S. Hallé, R. Deca, O. Cherkaoui, and R. Villemaire, "A formal validation model for the netconf protocol," in *of Lecture Notes in Computer Science*. Springer, 2004, pp. 147–158.
- [5] R. Enns, Ed, M. Bjorklund, J. Schoenwaelder, and A. B. Ed, "Network configuration protocol (netconf)." RFC 6241, 2011.
- [6] P. Aitken, B. Claise, and G. Muenz, "Configuration data model for the ip flow information," RFC 6728, 2012.
- [7] N. W. Group, Ed., *NETCONF Configuration Protocol*, 2006.
- [8] T. Lille, Ed., *Protocole d'administration Netconf*. Telecom Lille, 2010.
- [9] *A Data Modeling Language for the Network Configuration Protocol (NETCONF)*. RFC 6020 (Proposed Standard), 2010.