International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Network Configuration Management Using NETCONF and YANG

Hemalatha K[1], Karthik R[1], Dr. R Chinnaiyan[2]

[1]Master of computer Application/New Horizon College of Engineering/Bangalore, Karnataka, India
[2]Professor,Department of Master of Computer Applications, New Horizon College of Engineering, Bangalore, India

ABSTRACT

The standardization of network configuration management in internet engineering task force using the protocol is NETCONF that provides an mechanisms that is used to install and manipulate delete the configuration of all network deivces.The NETCONF article which describes the protocol which was recently introduced by the NETCONF data modelling language known as YANG. This language supports all the data modellers who defines all the syntax semantics in device configuration support.

Keywords : NETCONF, YANG, IETF, CLI, SNMP, Internet Architecture Board

I. INTRODUCTION

The management of the configuration of a large number of networked devices remains a highly important practical problem. Device configurations and the mechanisms to retrieve and modify them are largely vendor-specific, and the most widely used configuration interfaces today are proprietary command line interfaces (CLIs), making it costly to achieve a high level of efficiency and reliability through automation. In 2003 the Internet Engineering Task Force (IETF) started an effort to develop and standardize a network configuration management protocol, which led to the publication of the Network Configuration (NETCONF) protocol at the end of 2006.

The NETCONF protocol supports several features required for configuration management that were lacking in other network management protocols such as Simple Network Management Protocol (SNMP). NETCONF operates on so-called datastores and represents the configuration of a device as a structured document, serialized using the Extended Markup Language (XML). The protocol distinguishes between running configurations, startup configurations, and protocol design and its specification. It was, however, clear that a common data modeling language is needed in addition to the protocol to express the structure and semantics of configuration information in a vendor-neutral format. A proposal for a NETCONF data modeling language called YANG was developed in 2007 and is being standardized in the IETF since 2008.

The aim of this article is threefold. First, we provide an overview of the NETCONF protocol. Second, we describe the recently defined YANG data modeling language. Finally, we discuss some of the available implementations and how they have been used to provide a programmatic configuration interface that integrates well with other management interfaces of devices.

The rest of the article is structured as follows. The next section provides additional background information before the NETCONF protocol is described. The YANG data modeling language is then introduced. Implementation experience is reported and related work is discussed before the article concludes in the final section.

BACKGROUND AND MOTIVATION

In 2002 the Internet Architecture Board (IAB) organized a workshop in order to guide future network management standardization activities in the IETF. The workshop was attended by network operators and protocol developers, and resulted in several concrete recommendations. One of the recommendations was to focus IETF resources on the development of standards for network device configuration management.

Another recommendation was to use XML for data encoding purposes. In 2003 a working group was formed in the Operations and Management area of the IETF to produce a protocol supporting network configuration. The working group charter mandated that XML be used for data encoding purposes.

The protocol produced by this working group is called NETCONF. The design of NETCONF has influenced by proprietary protocols such as Juniper Networks' JUNOScript application programming interface (API). Network operators today use differing approaches to managing device configurations **166** 0163-

In the Network is the Record approach, operators directly modify the configuration of devices. To make the configuration change process (and any errors incurred due to configuration changes) trackable, all configurations are copied from the devices into a configuration backup repository. In the *Generate Everything* approach, a network-wide configuration database is used to generate device configurations that are pushed to the devices. In both approaches it is necessary to be able to push configuration changes to devices and dump/restore complete configurations. In addition, it is vital that devices distinguish between configuration data and data describing operational state that has been obtained via other means (e.g., via routing protocols or signaling protocols). The driving force behind NETCONF is the need for a programmatic cross-vendor interoperable interface to manipulate configuration state. The approach of automating CLIs using programs and scripts has proven problematic, especially it comes to maintenance and versioning issues. Several operators reported during the IAB workshop that they find it time consuming to maintain programs or scripts that interface with different versions of a CLI.

II. NETWORK CONFIGURATION AND PROTOCOL

The NETCONF protocol has a simple layered architecture . The core of NETCONF is a simple remote procedure call (RPC) layer running over secure transports such as SSH, TLS, SOAP, or BEEP. Secure Shell (SSH) transport is mandatory to implement as a means of promoting interoperability.

The operations layer residing on top of the RPC layer provides specific operations to manipulate configuration state. An additional document Defines operations to subscribe to notification Streams and receive notifications. Further operations are expected to be added in the future in order to support data-model-specific Management operations.

NETCONF assumes that the configuration State of a device can be represented as a structured document that can be retrieved and manipulated (document-oriented approach). In order to deal with large configurations, the protocol supports filtering mechanisms that allow clients to retrieve only a subset of the configuration.

NETCONF supports multiple configurations data stores. A configuration data store contains all information needed to get a device from its initial default state into the desired configuration state. The running datastore is always present and describes the currently active configuration.

In addition, NETCONF supports the notion of a startup configuration datastore, which is loaded by the device as part of its initialization when it reboots or reloads, and a candidate datastore, which is a scratch buffer that can be manipulated and later committed to the running datastore. NETCONF features a rich set of protocol operations. It is generally expected that new protocol operations will be added in the future by vendors and standardization bodies. This essentially means that NETCONF can easily support a command-oriented approach in addition to the already defined document-oriented approach to manipulating configuration state. Table 1 shows the protocol operations that have been defined so far by the NETCONF working group of the IETF. The first six operations all operate on configurations stored in configuration datastores selected by the source or target arguments. The lock and unlock operations do coarse-grained locking, and locks are intended to be short-lived. More fine-grained locking mechanisms are currently being defined in the IETF.

The get operation is provided to retrieve a device's configuration state together with its operational state, while the get-config operation only returns configuration state. The distinction between operational state and configuration state is a very important feature of NETCONF missing from other network management protocols, such as SNMP, where it is assumed that management applications have the necessary knowledge to identify which data item belongs to the operational or configuration state.

The get and get-config operations both support an optional filter parameter to select the subset of the configuration and state data that should be retrieved. Implementations can support different filter mechanisms. The mandatory subtree filter mechanism selects the branches of an XML tree matching a provided template. As an optional feature, implementations can choose to support XPATH expressions as filters. The most powerful and also most complex operation is the edit-config operation. The edit-config operation can be used to modify the content of a configuration datastore by creating, deleting, replacing, or merging configuration elements. In a nutshell, edit-config works by applying a patch to a datastore in order to generate a new configuration. Since a treebased representation is used to represent configurationtate, it is necessary to describe which branches in the tree should be created, deleted, replaced, or merged. NETCONF solves this by adding an operation attribute to the XML payload of the edit-config operation. With this approach, relatively complex configuration changes can be achieved in a single edit-config invocation. Notification support in NETCONF is based on an event stream abstraction. The event stream abstraction enables NETCONF to support several different event sources. Clients interested in receiving notifications subscribe to event streams. On systems that maintain event logs, it is possible to subscribe to an event stream at some time in the past and the device will playback all recorded notifications at the beginning of the notification stream. A subscription to an event stream establishes a filter that is applied before event notifications are sent to the client. This allows one to select only the relevant notifications and improves scalability.

III. DATA MODELING LANGUAGE YANG

Since NETCONF uses XML to encode network management data, it may seem obvious to use one of the existing XML schema languages to formally specify the format of these XML documents. While some parts of the industry favour the XML Schema

Definition Language (XSD) there is significant uptake of RelaxNG in recent years. But putting aside the differences between XSD and RelaxNG, it is clear that additional NETCONF-specific information needs to be specified that goes well beyond the capabilities of these XML schema languages. Both XSD and RelaxNG only address part of the problem to be solved.

During the development of the NETCONF protocol specifications, which are formally defined using XSD, it has been observed that XSD notation is difficult to read and verify by humans. Other schema notations such as RelaxNG (and especially its compact notation) seem to be easier to read and write. Still, both schema languages tend to be relatively far away from an implementer's view of a configuration datastore and tend to become cumbersome to use when all the necessary NETCONF-specific extensions are added. Furthermore, the validation capabilities of standard tools have limited value; what is most urgently needed is the validation of configuration datastores and not so much the validation of individual protocol messages that contain the serialization of (parts of) a configuration datastore. Given the nature of the edit-config operation, individual messages might not satisfy all data model constraints but can still lead to a valid configuration datastore at commit time.
The YANG data modeling language [8] therefore takes a different approach. YANG aims to be a highly readable and compact domain-specific language for defining NETCONF data models.

YANG comes with a twin called YIN, which is an XML representation of YANG so that standard XML tools can be used to process YANG data model definitions. A lossless twoway conversion between YANG and YIN is defined. In addition, one-way conversions to XSD and RelaxNG are available so that corresponding tools can be used.

IV. IMPLEMENTATIONS

Several NETCONF implementations have been created and are being deployed. Major network device manufactures such as Juniper Networks, Ericsson, Cisco Systems, and Nortel ship NETCONF as part of their device software. Other companies such as Tail-f Systems, Netconf Central, SNMP Research, and Silicon and Software Systems license complete NETCONF development kits to device manufacturers. Several open source NETCONF implementations are under development. Efforts to develop interoperability test suites have started recently [9], and discovered implementation bugs as well as ambiguities in the protocol specification are being fleshed out. Several commercial and open source implementations of the YANG data modeling language have been developed. Some YANG compilers support translation of YANG models to the YIN format, XSD, and RelaxNG. To facilitate access to SNMP data models through NETCONF, an SMIv2-to-YANG translation algorithm has been implemented as part of an open source SMIv2 toolset. Finally, some high-level programming frameworks are being developed to make it easy to glue NETCONF into network-wide configuration and policy systems. A good example is the ncclient API for Python [10]. Shows a Python script modifying the nameservers used by the set of hosts identified on the command line. The script uses the candidate datastore and protects itself against concurrent scripts by locking the candidate datastore.

V. RELATED WORK

The YANG language has been influenced by the design of the SMIng data modeling language Different from SMIng, YANG does not aim at being a protocol-independent language.

Based on the experience with the SMIng approach [12], a design decision was taken very early to design YANG as a domain-specific NETCONF data modeling language.

An initial study of performance aspects of the NETCONF protocol can be found in . While it has been acknowledged by the IETF working group that an access control subsystem for NETCONF is needed, standardization work in this area has not yet been started. Some early research in this area can be founded.

VI. CONCLUSION

The design of NETCONF and YANG incorporates decades of experience from network device IEEE Communications Magazine • September 2010 173 vendors, standardization bodies, implementers, and operators, and therefore has great potential to make network configuration management simpler, more effective, and more robust.

The NETCONF protocol addresses the requirements for configuration management protocols defined in well and is therefore a good choice for this task. The pragmatic approach to layer NETCONF on top of a secure and reliable transport greatly simplifies the protocol. The development of the YANG data modeling language and its standardization in the IETF is progressing well.

The availability of open source YANG tools has already motivated IETF working groups to use YANG for writing configuration data models, even though some aspects of the YANG language are still being finalized, and some toolkit vendors have already replaced their proprietary data modeling language with YANG. While NETCONF and YANG provide a strong base technology for simpler, more effective, and more robust configuration management, additional cost savings will be achieved by the definition of standard configuration data models. This will be a longer-term effort because it requires identifying and agreeing on common subsets of configuration information that can be easily augmented with vendor-specific extensions for the vendor-specific features that differentiate their products. But even with just a small set of common configuration data models, NETCONF and YANG offer technology to deal with proprietary configuration data models in a much more cost-effective and robust way.

VII. REFERENCES

[1]. R. Enns, "NETCONF Configuration Protocol," Juniper Networks, RFC 4741, Dec. 2006.

[2]. J. Case et al., "Introduction and Applicability Statements for Internet Standard Management Framework," SNMP Research, Network Associates Laboratories, Ericsson, RFC 3410, Dec. 2002.