International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 3, ISSN 2249 –071X

Managing Heterogeneous Networks Using NETCONF

G. Harish Reddy
Assistant Professor, SreeNidhi Institute of Science & Technology, Ghatkesar
harsha.rex@gmail.com

M. Bala Sridhar
Software Technical Lead,
Free Scale Semi-coductors Pvt. Ltd., Hyderabad
mbs_sridhar@yahoo.com

***Abstract* - The Network management became more and more complex with the advancement of heterogeneous network devices. The efficient configuration management of heterogeneous network devices from different vendors is a great challenge.**

During the past twenty years, Simple Network Management Protocol (SNMP) as an industrial standard has played a very important role in managing the computer networks. However, as the networks are growing day by day, SNMP could not satisfy the need of configuration Management for huge networks. In 2006, the IETF released a new protocol called Network Configuration Protocol (NETCONF) [1] for configuration management to overcome the shortcomings of the SNMP and other management engines like Command Line interface (CLI) which is predominantly used for device configuration at present.

***Keywords* - NETCONF, SNMP, NMS, SOAP, WSDL, FCAPS.**

I. INTRODUCTION

Network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems [2].

- **Operation** deals with keeping the network (and the services that the network provides) up and running smoothly. It includes monitoring the network to spot problems as soon as possible, ideally before users are affected.
- **Administration** deals with keeping track of resources in the network and how they are assigned. It includes all the "housekeeping" that is necessary to keep the network under control.
- **Maintenance** is concerned with performing repairs and upgrades — for example, when equipment must be replaced, when a router needs a patch for an operating system image, when a new switch is added to a network. Maintenance also involves corrective and preventive measures to make the managed network run "better", such as adjusting device configuration parameters.
- **Provisioning** is concerned with configuring resources in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service.

A common way of characterizing network management functions is FCAPS [3]-Fault, Configuration, Accounting, Performance and Security.

General requirements of Network Management Systems:-

- Configuration state vs. Operational state
- Concurrency support
- Configuration Transactions
- Distributed Transactions
- Distinguish multiple configurations
- Persistence of configuration state
- Configuration change events
- Configurations backup and restore
- Support for standard tools
- Minimize impact of configuration changes

II. RELATED EXISTING SYSTEMS

A. Simple Network Management Protocol (SNMP):

Even though SNMP provides operations [12] to configure network devices, it is limited to collecting statistics and status information from network devices. It is hardly used for configuration purposes. There are several reasons for this deficiency. Here, we mention the important ones:

- The SNMP protocol is simple, leaving the burden of manipulating configuration data on the management application. For this reason, tool development based on SNMP is expensive.
- SET requests are sent independently. This may cause a serious network problem if a manager sends several SET requests to configure a particular device and one request fails.
- SNMP does not provide any mechanism to undo recent changes in the device configuration.
- SNMP does not provide synchronization among multiple network devices. If a manager sends a SET request to a group of devices (to have similar configuration), some of them can succeed, and others can fail.
- It does not employ the standard security mechanism.

  Instead, the security is self-contained within the protocol itself, which makes SNMP credentials and key management complex and difficult to integrate with other existing credentials and key management systems.
- The SNMP has been widely used as an industrial standard in network management for its simplicity and flexibility.
- However, SNMP could not satisfy the need of configuration management of multiple devices.
- Moreover, SNMP prefer to use UDP to transfer messages, which results in unreliability and limited size.
- While SNMP is easy to set up and access, it is also easy to hack due to its ever-present nature.

Companies with advanced security and monitoring needs will want to look to more advanced monitoring systems.
- SNMP protocol fails in achieving a good performance when the data to be transferred are bulky. This is because it cannot provide efficient operations and the method of encoding SNMP messages is not very efficient, hence the latency of SNMP transfers can be quite high sometimes.
- Does not provide Transaction based configuration Management.

B. Command Line Interface:

Another traditional configuration tool is CLI, which is insufficient nowadays for configuration because of the following reasons.
- First, CLI requires extra efforts by operators to learn different commands for different devices.
- Second, CLI lacks a formal description language to define all properties of the programmatic interface.
- Finally, CLI doesn't have any structured error responses.

III. Approach and Advantages

A. Approach

In this paper, we have applied Web Services to NETCONF network management [4] using Simple Object Access Protocol (SOAP)[7], described in RFC4743, WSDL [5] and UDDI [6].

B. Advantages

First advantage lies in its separation of configuration data and state data, which can avoid the problems that comparisons of configuration data sets would be dominated by irrelevant entries such as different statistics and incoming data could contain nonsensical requests such as attempting to write read-only data.

Secondly, its base configuration operations implemented care more complete and efficient. The <editconfig> operation has four attributes: merge, replace, create and delete. These attributes identifies the point in the configuration to perform the operation and may appear to multiple elements throughout the <config> tree. The prototype also provides <copy-config> and <delete-config> to create and delete the whole configuration datastore except the <running> configuration datastore. There are also formalized transaction mechanism such as <lock> and <unlock> operations to ensure that the manager can make changes without fear of interaction with others.

Finally, there are many additional capabilities. The writable-running capability indicates that the device supports directly the <running> configuration datastore. The candidate configuration capability provides operations of <commit> and <discard-changes> to ensure the integrity of the data when data are changed. The validation capability consists of checking a candidate configuration and semantic errors before applying the configuration to the device. Other advantages are:
- XML is human-readable, which facilitates debugging of erroneous implementations.
- Many standard libraries and tools for XML processing are available (GSOAP, AXIS).
- Configuration data can be structured in a flexible way.
- Message format and data models can be easily extended.
- One of the big advantages of NETCONF over SNMP is how the protocol works when manipulating a group of semantically related configuration data.
- It allows configuration to occur in a transactional manner. NETCONF allows a managed device to rollback to a known-state configuration.
- NETCONF helps in achieving a good performance when the data to be transferred are bulky.

IV. System Design

These are a few considerations a software designer must take into account while executing a project.
- Support widely available XML technology
- Support a wide variety of platforms via capability driven protocol features
- Separate configuration data from other data
- Provide standard and named configuration data stores
- Decouple the different conceptual protocol layers for robustness and extensibility
- Use a remote procedure call mechanism [9] to frame requests and responses
- Provide configuration locking to insure exclusive access to a configuration data store
- Support multiple transaction and content models
- Provide a simple set of protocol operations

A. System Overview

A NETCONF implementation [8] residing in a Manager works as a NETCONF client while network equipment acts as a NETCONF Agent/Server. In this document, we call NETCONF-client and NETCONF server/Agent implementations a NETCONF application and a NETCONF service provider, respectively. A SOAP implementation may be installed on both the Manager and the network equipment. Each instance of the SOAP implementations exchanges SOAP messages based on WSDL, as described in [RFC4743]. If Java libraries generated from the WSDL are provided in the Manager, NETCONF application can be developed, which configures network equipment via the NETCONF protocol, by utilizing the Java library [11]. There is no need to use XML or SOAP directly.

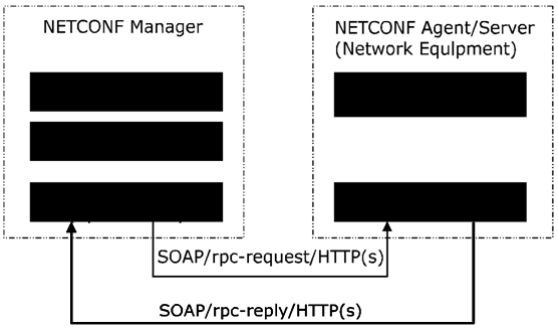

Fig.4.1. NETCONF System Overview

Axis [10] works as a SOAP implementation and a Network Management System-development tool. For instance, WSDL2Java, one of Axis' tools, which generates Java-class files from a WSDL file. Another tool called Java2WSDL does the opposite, which generates a WSDL file from Java-class files. Consequently, various benefits can be obtained if Axis is introduced as a SOAP implementation.

B. NETCONF Manager Design

NETCONF Manager Design consists of the following essential components.

GUI Presentation Layer:

This layer provides Graphical user interface to point click and select the configuration. This layer provides the graphical view of the configuration data. **Management Operations Layer:**

The operations layer defines a set of base operations invoked as RPC methods with XML - encoded parameters. This layer provides a small set of low- level operations to manage device configurations and retrieve device state information.

Transaction Layer:

This layer provides transaction management like starting the configuration session, end the configuration session and revoke configuration session on error.

RPC Layer or the SOAP Implementation Layer:

This layer provides the communication with the Agent/Server residing on the network equipment (device/router) through XML based remote procedural calls (RPC).

Java Virtual Machine:

Entire Manager Application runs on the Java Virtual Machine installed on the system as part of the Java Runtime Environment (JRE).

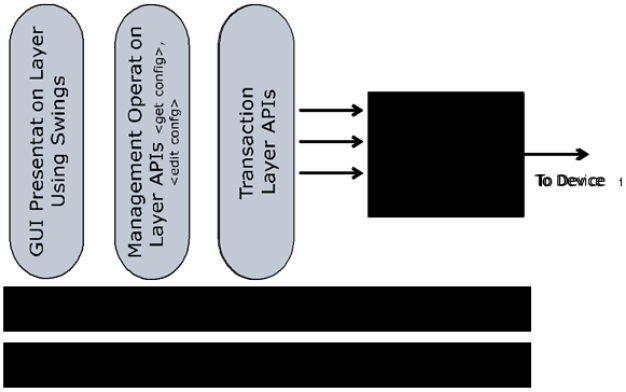

Fig.4.1.1. NETCONF Manager Design

C. NETCONF Agent Design

NETCONF Agent Design consists of the following essential components.

SOAP Service Agent APIs: These are the Web Service APIs exposed as services to the NETCONF Manager.

Capabilities APIs: These APIs provide capabilities services to be exchanged between the client and the server.

Management Operations APIs: These services provide operations to manage device configurations and retrieve device state information.

Transaction Layer: This layer provides transaction management services like starting the configuration session, end the configuration session and revoke configuration session on error.

Device Translation Layer: This layer provides translation of service calls into device specific format.

Java Virtual Machine: Entire Agent Application runs on the Java Virtual Machine installed on the system as part of the Java Runtime Environment (JRE)

.

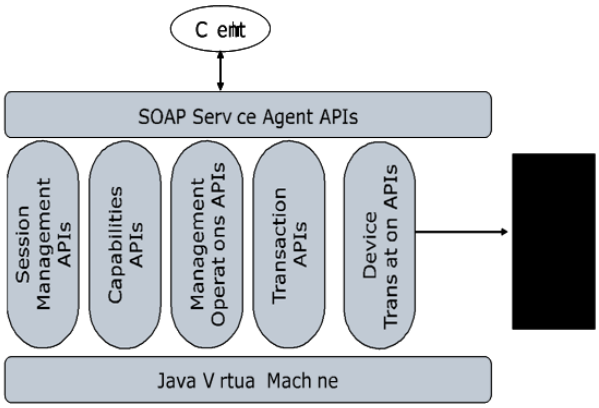

Fig.4.2. NETCONF Agent Design

D. Basic NETCONF Capabilities

This section lists a set of capabilities that a client or a server may implement. Each peer advertises its capabilities by sending them during an initial capabilities exchange. Each peer needs to understand only those capabilities that it might use and MUST ignore any capability received from the other peer that it does not require or does not understand.

Name	Description
writable-running	Agent allows to write to <running> config
candidate	Agent supports the <candidate> config
confirmed-commit	Agent supports “confirmed” mode of <commit> operation.
rollback-on-error	Agent supports rollback of <edit-config>
validate	Agent supports validation
startup	Agent supports <startup> config
url	Agent supports <url> parameter
xpath	Agent supports XPath filters

Table I: NETCONF Capabilities

E. Basic NETCONF Operations

The base protocol includes the following protocol operations:

Name	Description
get-config	Retrieve some all configuration
edit-config	Edit some or all configuration
copy-config	Copy contents of one config to
delete-config	Removes all contents of the config.
lock	Start exclusive write access of config.
unlock	Stop exclusive write access of config.
get	Retrieve config and/or state data
close-session	Cause the session to close
kill-session	Force another session to close

Table II: NETCONF Operations

F. Transaction Model

A configuration data store is the complete set of configuration information that is required to get a device from its initial default state into a desired operational state. Among several transaction models available candidate model is chosen as it is one of the common ways followed today.

The <candidate> configuration data store represents a configuration that may become a <running> configuration through an explicit commit.

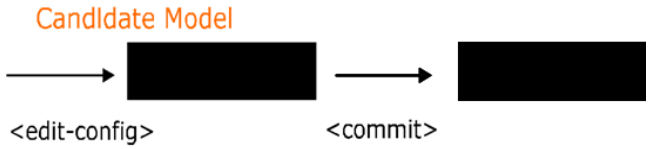

Fig.4.5. Transaction Model

V. IMPLEMENTATION

Implementation of the proposed design model which is based on Web Services consists of three parts, *a Manager, an Agent and a group of Modules* requiring configuring or monitoring.

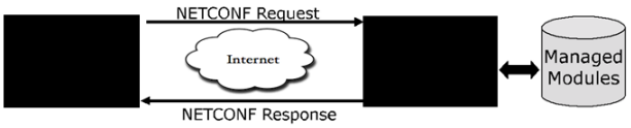

Fig.5.1. NETCONF Modules

- Manager is Java based User Interface with objects selection, parameter setting and display.
- Agent is the core part of the whole architecture which concerns the function and performance of system.
- Modules interested to be configured or monitored are registered with the agent

VI. RESULTS

6.1. Comparison: NETCONF vs SNMP

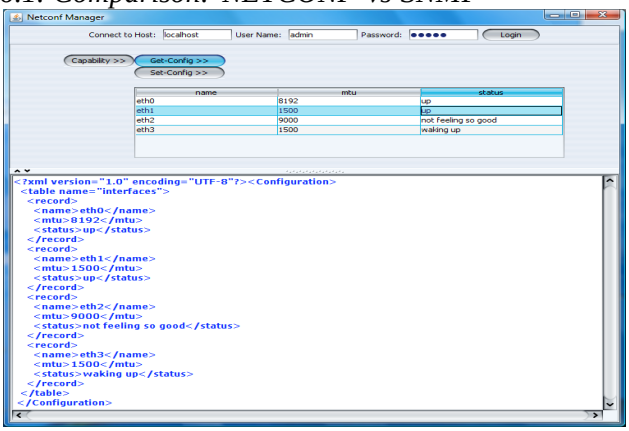

Fig.6.1. NETCONF Get Configuration

Above figure shows XML <rpc-reply> of ***bulk configuration*** received from the Agent. This response includes complete Interface table data received from the Agent in single shot.

Following is the GET request response flow from SNMP output one column at a time as described by "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)" [12] [13]:

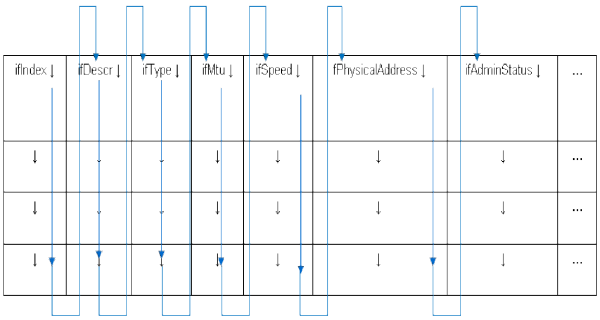

Fig.6.2. Multiple SNMP GET Configuration requests

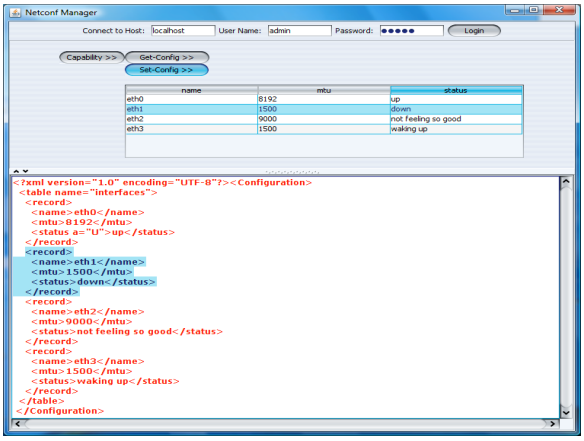

Fig.6.3. NETCONF Set Configuration

Above Figure shows the sample NETCONF Set Configuration of Interface table, to update the status of "eth1" interface to "down" state. Bottom window shows the XML Response from the server.

Like this multiple requests can be sent in a single transaction.

The Following are the operations involved:

NETCONF Client/Manager Operations:

1. Start Configuration session
2. Edit Configuration command (RPC call)
3. Send parameters
4. wait for response (RPC reply)
5. On receiving response (RPC reply)
6. End the Configuration session
7. Update Display with Configuration.

NETCONF Sample Request Format:

```
<rpc message-id="102“
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <my-own-method>
      <first-parameter>eth1</first-parameter>
       <another-parameter>down</another-parameter>
    </my-own-method>
  </rpc>
```

NETCONF Agent Operations:

1. Accept Configuration request
2. Validate request (check request type)
3. Start Configuration Session (Apply Lock)
4. Validate request (check parameters)
5. Execute configuration
6. Send rpc-reply
7. End Configuration Session (Release Lock)

6.2. Observations

The following are the observations we found:

NETCONF Operations:

1. Multiple SET operations can be achieved in a single request
2. Provides locking mechanism and protects data.
3. Provides Transaction support and can be revoked before applying.
4. Can support synchronization of configuration among multiple network devices using distributed transactions.
5. Transport layer is independent can support secure communications using SSL/HTTPS.
6. Message format and data models can be easily extended using standard XML data model.
7. Easy for debugging based on the simple human readable XML output.

SNMP Operations:

1. SET requests are sent independently.
2. One SET operation at a time
3. Does not involve any locking
4. Transactions cannot be revoked before applying
5. SNMP does not provide synchronization among multiple network devices. If a manager sends a SET request to a group of devices (to have similar configuration), some of them can succeed, and others can fail.
6. Does not employ the standard security mechanism.

VII. CONCLUSION & FUTURE WORK

In summary, it can be said that NETCONF is a promising alternative to SNMP with respect to the configuration of monitoring probes. Necessarily, the usage of NETCONF requires the standardization of XML-based configuration data models in order to guarantee interoperability between different NETCONF implementations.

This has been tested for a single network. Our future work includes the extension of it to *Clustered Environment* and *Distributed Cloud Management*.

REFERENCES

[1] R. Enns, “RFC4741: NETCONF Configuration Protocol”, Dec, 2006, ftp://ftp.rfc-editor.org/in-notes/ rfc4741.txt

[2] Alexander Clemm. (2006 Nov 21) Network Management Fundamentals (1st edition), Published by Cisco Press. Available: http://www.ciscopress.com

[3] FCAPS, International Telecommunication Union, "X.700: Management Framework For Open Systems Interconnection (OSI) For CCITT Applications", September 1992, Available: http://www.itu.int/rec/T-REC-X.700-199209-I/en.

[4] DANG Xiaochao, WANG Jinpeng, "NETCONF Network Management Model Based on Web Services", IEEE, Volume: 1, On page(s): 160-164, Aug. 2009.

[5] WSDL, ”Web Services Description Language (WSDL) 1.1”, http://www.w3.org/TR/wsdl

[6] UDDI ,”Universal Description, Discovery and Integration (UDDI)”, http://www.uddi.org.

[7] Goddard, “Using NETCONF over the Simple Object Access Protocol (SOAP)”, RFC 4743, Dec 2006.

[8] Yanan Chang, Debao Xiao, Member, IEEE, Hui Xu, Student Member, IEEE, "Design and Implementation of NETCONF-Based Network Management System", Volume: 1, on page(s): 256-259, Dec. 2008.

[9] "1994 – Andrew Birrell, Bruce Nelson: Remote Procedure Call". Software System Award citation. Association for Computing Machinery. Retrieved July 11, 2011.

[10] Axis ,”Web Services Project @ Apache”, http://axis.apache.org/axis.

[11] David A. Chappel & Tyler Jewell. (2002 March) “Java Web Services” (First edition), O’RELLY.

[12] Bob Stewart, chair Kaj Tesink, Glenn Water, Bert Wijnen, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[13] Douglas R. Mauro and Kevin J. Schmidt.(2001 July) "Essential SNMP" (First edition), O’RELLY

AUTHOR'S AFFILIATION

Mr. G.Harish Reddy

is from Hyderabad (Telangana). He had completed his M.Tech in Computer Science & Engineering from JNTU, Hyderabad. He is having 8 years of experience in teaching. He is working as an Assistant Professor in **Sree Nidhi Institute of Science and Technology,** Ghatkesar, Hyderabad. He got a chance as a visiting professor from APPA, and had designed a course on Network **Security** for them. He worked as a resource person for 2 workshops and convenor for a workshop organized by SNIST under TEQIP. He is having 5 International and 7 National publications. His areas of interest include Computer Networks & Protocols and Cryptography.

Mr. M.Bala Sridhar

is from Hyderabad. He had completed his M.Tech in Computer Science & Engineering from JNTU, Hyderabad. He is having 14 years of experience in Software Industry both in India and overseas (**USA, Sweden and France**), worked extensively in several domains like Datacom, Telecom, Smart Card Security and Client/Server Technologies. He worked with **Ericsson** from 1999-2003. He worked with **Active Identity** from 2003-04. He worked with **Intoto Software (I) Pvt. Ltd**. from 2004-2011. Currently he is working as Software Technical Lead in **FreeScale Semi-coductors(I) Pvt. Ltd**., Hyderabad. His areas of interest include Computer Networks & Protocols and Cryptography.