Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 30, 2013

Y. Teramoto
Kyoto University
R. Atarashi
IIJ Research Laboratory
Y. Atarashi
Alaxala Networks Corp.
Y. Okabe
Kyoto University

Feb 26, 2013

Experience of Designing Network Management System
draft-teramoto-experience-network-management-01

Abstract

This document describes our experiences from designing and implementing a large-scale local area network management system using mainly NETCONF. We designed the data models for device configurations and implemented NETCONF client to centrally control multiple devices of various vendors. The document provides insight on strong and weak points of current NETCONF approach. The document also makes some recommendations about NETCONF and future network management protocols.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

We designed a large-scale local area network management system that can manage the network independently of physical topology and the vendor of networking equipments composing the network system by modeling the functions of network equipments and using widely used network management protocols. To manage networks, we used mainly the NETCONF protocol and the SNMP [RFC3416] because these protocols are supported by various major vendors' networking equipments and does not depend on specific architectures such as an OpenFlow or a MPLS. This document describes the experiences from designing and implementing such management system.

An NETCONF protocol is defined in [RFC6241] and is intended to manage configurations of the networking equipment from computers. The NETCONF protocol is supported by various network equipments such as Cisco IOS and Junos and so on. We mainly used a NETCONF protocol to get and edit configurations of networking equipments. Although many networking equipments support the NETCONF protocol, there are some kinds of data model designed by vendors. It is because the model layer of NETCONF protocol is still developing. We designed the unified data model for device configurations to manage multiple equipments of various vendors and implemented NETCONF client to control multiple devices simultaneously.

This document evaluates the NETCONF protocol design and makes some recommendations about NETCONF and future network management protocols to support large-scale network management.

2. Management System Setup

Our Network management system contains some information in database to grasp network states and authorize management interface of equipments. The NETCONF protocol and SNMP requires login authorization on starting login. Moreover these protocols support only the information for the configurations and states, and hence the management system needs some information related to several networking equipments such as the network topology and the routes of IP packets. The remainder of this section describes the setup information that is required to start network managing.

2.1. Required Information

Network Management systems usually require some information in advance that the systems cannot retrieve from network devices automatically like followings:

Management Interface

One of the most fundamental information is login authorization for connecting to management interfaces of network devices: management IP address, user ID, password and so on. This is because management protocols such as NETCONF and SNMP usually requires authentication to get or edit device configurations at need to be secured.

Device Information

The system requires the information for devices, such as the vendor, the model and the version to determine the capabilities and providing resource of the device, because NETCONF provides no information and no unified models for them.

Network Information

Network Devices provides no information for network topology. There are some protocols to get neighbor information such as LLDP, however there is no standardized protocols to provide them into client. Therefore, to manage network with network topology, the topology information is to be prepared in advance.

3. Experiences of Implementing NETCONF Client

The NETCONF protocol is an XML-based protocol used to manage the configuration of networking equipment. We implemented NETCONF client to manage networking equipments centrally. This section provides insight on NETCONF protocol.

3.1. Transport Protocol

The NETCONF protocol defines an SSH protocol as mandatory transport protocol. The advantage of using the SSH protocol is that it provides strong authentication mechanism and full encrypted communication. However, there are some issues of using the SSH protocol. It is very large protocol for developers to implement full scratch client, therefore they need to use some existing libraries. Network management systems often have their own architecture for increasing performance, because it is often required high-performance operation and response. SSH libraries also often have their own architecture, and hence it is need to carefully bond two architecture. This may cause performance degradation on multi-threading software because these asynchronous events cannot be concentrated into one. Furthermore, if encrypted communication is forced like the SSH protocol, it is sometimes difficult to detect the

cause of transport problems on debugging.

3.2. Framing Mechanism on SSH transport

The current framing mechanism of SSH transport is defined in [RFC6242]. The previous version of NETCONF (version 1.0) defined framing mechanism to separate each messages by the character sequence "]]>]]>" according to the assumption that well-formed XML documents does not contain the sequence, however it was found later that the assumption is not collect. Therefore the current framing mechanism of SSH uses chunked framing mechanism.

However framing mechanism still have confusing specification. The framing mechanism defines no error notification mechanism when given chunk-size is invalid or an error occurs on transport layer. [RFC6242] requires the peer to terminate the NETCONF session immediately without notifying error information when receiving such an invalid message. This mechanism often causes confusing issues that the developer cannot determine the reason of unexpected disconnection, because the reason may exist on multiple layers from physical layer to application layer. This problem also occurs on the previous version.

3.3. Capability Exchange

Capabilities are advertised in messages sent by each peer during session establishment as <hello> message. Each peer determines the NETCONF version by comparing the sent capabilities with received one. Furthermore this list can contain other additional capabilities such as the capability of notification.

The mechanism of capability exchange have the same undesirable specification as framing mechanism of the SSH transport. [RFC6241] require peers to disconnect the session immediately when the supporting capability version is mismatched or the treatment of session-id is wrong.

The reaction of invalid <hello> varies by vendor implementations. For example, the Junos returns error message as XML comments like followings:

```
<!-- netconf error: unknown command -->
<!-- session end at 2012-08-30 19:01:10 UTC -->
```

Cisco IOS returns no error message and disconnected on receiving next message by the frame chunk. This causes unexpected disconnection on sending first <rpc> command.

This is notable that other management protocols that have capability

exchange such as OpenFlow often support error notification mechanism on receiving an invalid hello message.

3.4. Lock and Commit Mechanism

Lock mechanism of NETCONF is defined in [RFC6242] as mandatory function. Commit mechanism is defined as optional function. These commands are useful on controlling multiple devices.

3.5. Notification

The original NETCONF protocols does not provide networking equipments to push some information to client. To send some information from servers, notification mechanism is defined as optional capability in [RFC5277].

Notification mechanism supports replaying the previous logs by specifying startTime. After starting notification, the server can only operate close-session command to terminate current session and returns resource-denied error on receiving other commands. However, the server that supports interleave capability can operate any commands while notification phase. The client can specify stopTime to stop the notification on the time. After stop time, NETCONF session becomes ordinal mode and accept any NETCONF commands again.

This too complicated mechanism makes enormous number of session states and conditions like following diagram; this makes it difficult to create brief management system that support full NETCONF specifications.

```
        +-------+
        | Hello |
        +-------+
            | <hello> exchange
     S  +-------+ <------+
     t+>|  RPC  |  <rpc> | return result
     o| +-------+ -------+
     p|     | <create-subscription>
     T|     |<--------------------------+ Yes: return result
     i| +---------+ <rpc>  -----------   | No : return resource-denied
     m+-| Notify |------< interleave? >-+
     e  +---------+        -----------
            | <rpc><close-session/></rpc>
           End
```

Figure 1: Session States and Conditions

4. Experiences with Data Model Design

Current NETCONF protocol does not provide data models specification, and hence there is no unified data models. Therefore, to manage networking equipments of multiple vendors with same way, we designed some models of the function that networking equipments provides. This section describes the insight on current approach of standardizing data models.

4.1. YANG

YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF defined in [RFC6020]. The approach of YANG is to express various configurations using highly expressive schema, while the approach of SNMP is to predefine fundamental data model.

The same approaches of YANG often causes variability of configuration model and also causes too large and complex data schema. Furthermore YANG defines only the model schema, therefore the way to construct model data is owed to the developer. We think the fundamental model data, that is frequently used in configuration such as VLANs and Interfaces, should be defined by NETCONF core specification like SNMP in order to avoid such problems.

5. Experiences with Management Network

We designed the system to manage multiple network equipments of multiple vendors that support the NETCONF protocol. This section describes the experience with example of designing some function that uses multiple networking equipments via NETCONF.

5.1. VLAN Configuration

VLAN configuration is one of most fundamental functions; however the configuration needs various information. VLAN configuration requires following information:

- Current VLAN assignment

- VLAN ID and port name to be assigned

- Additional information such as bandwidth, port state (some ports may be disabled)

- Network topology

First two information can be retrieved from NETCONF configurations. The rest information, however, is more difficult than the former two. NETCONF provides no information for the capability of networking equipments, such as ports speed or the number of ports, that is not shown as configuration text. NETCONF also does not offer topology or neighbor information because of the same reason.

No current widely used protocols support the mechanism to retrieve such information, therefore we have no choice but prepare them in advance. However, an OpenFlow protocol now provide the way to get them by handling LLDP packets using packet-out operation. This is one of the reasons that an OpenFlow protocol is referred as a hot protocol.

6. Conclusions and Recommendations

We come to deeply know about the NETCONF protocol on the view of developers. This section conclude our experience and recommends some requirements that future management protocol would satisfy.

The current NETCONF protocol have some undesirable specifications like followings:

- Specify an SSH protocol as an mandatory transport protocol.

- Some undesirable error handling such as on capability exchange failure

- Too many state and conditions of each session by notification

Furthermore, a current approach of models is going toward complicated and large models by YANG. We propose future management protocols to use simple transport protocol, to define appropriate error handling that does not disconnect without any error notification, and to provide simple notification mechanism that is supported by default and supports interleave capability like function as default.

Current Network Management is required to use some similar protocols such as NETCONF and SNMP. We found that the two protocol is covering each other weakness that NETCONF does not support live information and SNMP does not support configuration management. However, there are few ways to get the capability of the networking equipments and the information for constructing topology graphs. There are often required to prepared in advance. There is a need to new management protocols for flexibly control whole network by providing appropriate models and information of the network.

7. Security Considerations

In our experience, the NETCONF protocol require high security considerations on the specification such as authorization and encrypted messaging; therefore the use of the NETCONF protocol improves the security of management.

To manage networking equipments centarally does not matter security issues if they are used in separated logical network and operated proper properly.

8. IANA Considerations

This document makes no request of IANA.

9. Normative References

[RFC3416] Presuhn, R., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

Authors' Addresses

Yasuhiro Teramoto
Graduate School of Informatics Kyoto University
Yoshida-Hommachi
Sakyo-ku, Kyoto 606-8501
Japan

Phone: +81-75-753-7417
Fax: +81-75-753-7440
Email: teramoto@net.ist.i.kyoto-u.ac.jp

Ray S. Atarashi
IIJ Research Laboratory
Jinbocho Mitsui Bldg., 1-105 Kanda Jinbo-cho
Chiyoda-ku, Tokyo 101-0051
Japan

Phone: +81-3-5205-6464
Fax: +81-3-5205-6466
Email: ray@iijlab.net

Yoshifumi Atarashi
Alaxala Networks Corp.
Shin-Kawasaki Mitsui Bldg.
890 Saiwai-ku Kashimada
Kawasaki, Kanagawa 212-0058
Japan

Phone: +81-44-549-1735
Fax: +81-44-549-1272
Email: atarashi@alaxala.net

Yasuo Okabe
Academic Center for Computing and Media Studies Kyoto-University
Yoshida-Hommachi
Sakyo-ku, Kyoto 606-8501
Japan

Phone: +81-44-549-1735
Fax: +81-44-549-1272
Email: okabe@i.kyoto-u.ac.jp