Information Technology Journal, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.

Research on ForCES Configuration Management Based on NETCONF

Lei Zhou, Ligang Dong and Rong Jin
College of Information and Electronic Engineering, Zhejiang Gongshang University, 310018,
Hangzhou, People's Republic of China

Abstract: Forwarding and Control Element Separation (ForCES) is a competitive technology to realize the Software Defined/driven Network (SDN). To meet the requirement of remote large-scale configuration with the extending from ForCES to SDN, a framework of ForCES configuration management system based on Network Configuration Protocol (NETCONF) is proposed. Configuration information is modeled by YANG and configuration protocol is based on NETCONF. Theoretical analysis and experimental results show that the new configuration method of ForCES based on NETCONF not only is more adaptable to the large-scale SDN network environment, but also has more excellent performance than the traditional method based on Simple Network Management Protocol (SNMP).

Key words: NETCONF, ForCES, configuration management

INTRODUCTION

Nowadays Internet has become an indivisible part of people's daily life and work. But with the development of society and network technology, drawbacks, such as security, reliability, have become increasingly prominent. In addition, the current architecture of the Internet based on IP is hardly to be changed which has limited the development of new functions. So, the current Internet has been getting more and more difficult to support security, reliability, content distribution, mobility and other emerging requirements. Therefore the international consensus is to design and establish a new architecture to adapt to the development of future networks.

SDN (Nadeau and Pan, 2011; Stiliadis *et al*., 2011) is a new generation network architecture which has quickly become a hot research area. SDN uses open interfaces and virtualization technology to separate network into three layers, application layer, control layer and data layer. So SDN can not only achieve centralized, unified and highly flexible control of multitudinous network devices, but also support efficient and low-cost deployment of new business.

ForCES (Yang *et al*., 2004) is a strong competitive technology to achieve SDN. Forces architecture separates FE (Forwarding Element) and CE (Control Element) and defines the ForCES protocol for the communication between FE and CE. A standard and efficient method of remotely configuring CEs and FEs needs to be proposed when extending ForCES technology to large-scale SDN network, because current ForCES technology is limited within a network node.

The easiest configuring method is Command Line Interface (CLI) which is simple and effective for device management. But the biggest problem is that managers must stay locally. Because of space and time limits, this method needs lots of manual labor.

Telnet method can configure managed devices remotely, but also has some shortages. First, Telnet has no security. Second, Telnet only provides a way of command transmission without unified abstract description of management information, so it can't be adapted to the multi-vendor network environment.

SNMP (Case *et al*., 1990) is a current industry standard of network management. SNMP uses SMI (Structure of Management Information) (McCloghrie *et al*., 1999) to define standard MIB (Management Information Base) (Case *et al*., 1996) in order to adapt a multi-vendor network environment. However, SNMP has some configuring congenital defects. Such as, lack of open and writable MIBs, disconnected and session-less features and so on. All these defects limit the SNMP's performance of configuration management. In addition, the most widely applied version is SNMPv2 which does not provide enough security.

NETCONF (Enns *et al*., 2011) is a new generation network configuration protocol based on Extensive Markup Language (XML) and its design purpose is to overcome the deficiencies of SNMP configuration management. A wealth of configuration management commands, rapid response time, outstanding scalability and security based on SSH (Secure Shell) (Ylonen and Lonvick, 2006) are the advantages of NETCONF.

Corresponding Author: Lei Zhou, College of Information and Electronic Engineering, Zhejiang Gongshang University, 310018, Hangzhou, People's Republic of China

Furthermore, the session-based method greatly improves the efficiency of configuration management. NETCONF has been supported by many software providers and is proposed as an international standard by W3C.

According to the above analysis, a method for ForCES configuration management based on NETCONF (Network Configuration Protocol) is proposed. Configuration information is modeled by YANG and configuration protocol is based on NETCONF. The method has the advantages of extensibility, security and efficient performance to adapt large-scale SDN network environment.

MATERIALS AND METHODS

Architecture: ForCES focuses on the separation of forwarding and control elements in a network element. SDN is a new network architecture for next generation network. We extend the ForCES framework to network region and propose an SDN architecture realized by ForCES technology and the configuration protocol in this architecture is based on NETCONF.

ForCES architecture: ForCES focus on the separation of forwarding and control elements in a network element. The basic structure of a standard network device meeting the ForCES protocol is shown in Fig. 1. RFC 3654 (ForCES Requirement Analysis) (Khosravi and Anderson, 2003) and RFC 3746 (the ForCES Framework) (Yang *et al.*, 2004) has made the basic definitions.

As shown in Fig. 1, a network device meeting the ForCES standard has at least one (or more for redundancy) CE and up to hundreds of FEs. The communication between the CE and FE is undertaken by a standard protocol called ForCES Protocol and this connecting point is called Fp reference point (ForCES control interface) which can be realized by single-hop network or multi-hop network. Fi/f is the outside network interface reference point of each FE, through which the network data is forwarding; Fi is the interface point between each two Fes in the same network device. Several FEs can constitute a distributed forwarding network to complete a complex forwarding task. Fr is the interface between each two CEs in the same ForCES network device.

All CEs are managed by a CEM (CE Manager) and all FEs are managed by a FEM, CEM and FEM can also exchange management information. But they just do some basic configuration management, for example assigning an ID number for each CE and FE. Current ForCES prototype systems (Wang *et al.*, 2013) are all use the way of local command line to manage CEs and FEs.

SDN architecture: SDN is new generation network architecture. SDN separates network into 3 layers, application layer, control layer and infrastructure layer, as shown in Fig. 2. Application layer can custom-make applications flexibly by APIs, Control layer controls thousands of network devices of the infrastructure layer through the southbound interfaces. Management plane is responsible for conventional network management to SDN and other function planes take charge of their corresponding network functions such as security.

ForCES network architecture based on NETCONF: Although, ForCES is proposed for network elements, its complete modeling technology and protocol design can absolutely apply to SDN. We call the SDN network SDN realized by the ForCES technology as ForCES network and its architecture is shown as Fig. 3. ForCES protocol acts as the southbound interface of SDN network, the control layer is composed of some distributed CEs and all kinds of network devices in the infrastructure layer are referred as FEs. The difference between ForCES network and ForCES net-element is that the number of FEs in the former is far more than that in the latter. In this case, the configuration of CEs and FEs should be implemented by a unified remote management way rather than by the

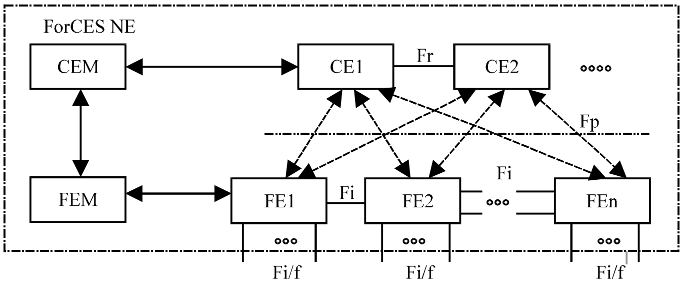


Fig. 1: Forwarding and control element separation (ForCES) architecture NE: Network element, CE: Control element, FE: Forwarding element, CEM/FEM: CE/FE manager, Fp: CE-FE interface, Fi: FE-FE interface, Fr: CE-CE interface and Fi/f: FE external interface

traditional CLI way. And the remote configuration management protocol should meet the high efficiency requirement to adapt to the needs of large-scale SDN networks. We put forwards that using NETCONF, a new generation of network configuration protocol, to configure and manage the CEs and FEs in ForCES network remotely as shown in Fig. 3. The management plane implements remote configuration management to each CE and FE with standard NETCONF protocol which is normative, secure and efficient. This configuring means based on NETCONF also can meet the high efficiency requirement of large-scale SDN network.

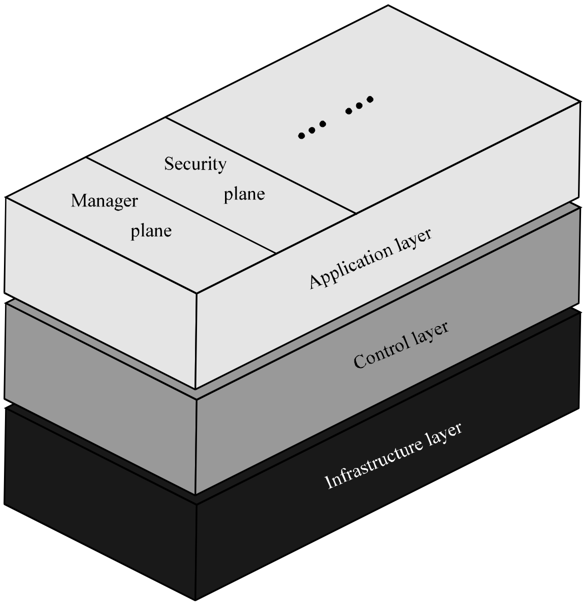


Fig. 2: Software driven/defined network architecture

Implementation of management of ForCES network based on NETCONF: There are two core contents of the ForCES configuration management system based on NETCONF. One is the modeling of the ForCES configuration information and the other is the implementation of NETCONF protocol.

YANG model for ForCES: The purpose of modeling ForCES configuration information is to describe management objects uniformly. On one hand it can facilitate human and machine's unified understanding, on the other hand make it easy to manage for specific management protocols such as NETCONF. Modeling is divided into two levels, one is IM (Information Model) and the other is DM (Data Model). Perkins (2002) IM is a conceptual abstraction of a managed object and is independent of specific realization and protocol. DM is a kind of lower-level abstraction which is realization-oriented and includes structures defined by specific protocols.

YANG (Bjorklund, 2010) is a data modeling language with powerful function, easy readability, intact

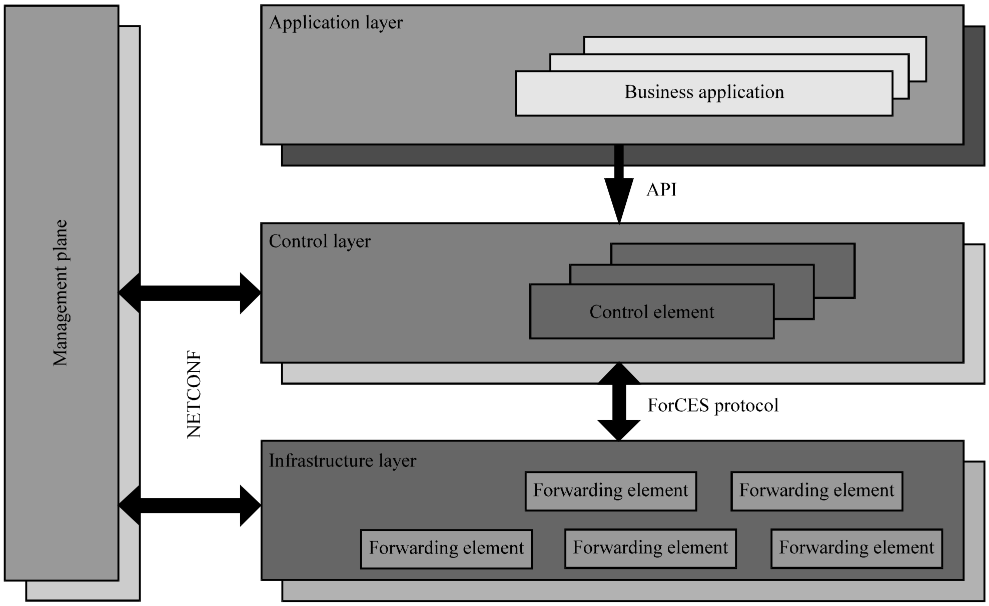


Fig. 3: Forwarding and control element separation (ForCES) network based on Network configuration protocol (NETCONF) architecture

information and strong scalability. It can not only model data as a tree structure and build module as the basic unit of data, but also define many built-in data types and methods of derived types. YANG is treated as the most suitable data modeling language for NETCONF in the industry. Therefore according to the ForCES configuration information, we use YANG tree to model an IM first and then use YANG language to model a DM.

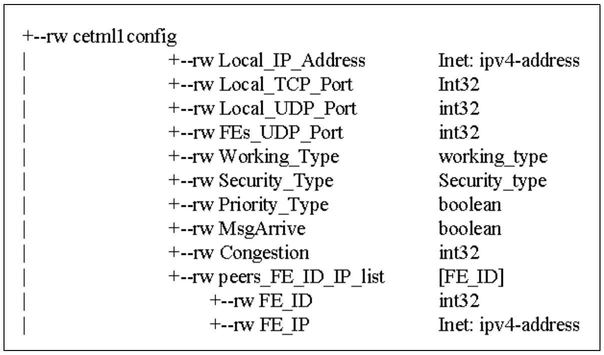
In order to make the design of the ForCES protocol independent to the transport layer used by ForCES itself, ForCES designers divide the control interface further into Transportation Mapping Layer (TM) and Protocol Layer (PL) (Wang *et al.*, 2010). So, the ForCES information also accordingly contains the configuration information of CE TML, CE PL, FE TML and FE PL. Here is an example of modeling of CE TML, the modeling steps with YANG is introduced as follows. First, we analyze the main parameters of CE TML which need to be configured, such as IP address, port number, priority level of the message queue, work type, congestion control mechanism and list of FEs which are allowed to connect with the CE. Second, we create an IM according to the configuration information of CE TML above in the form of YANG tree as shown in Fig. 4, a container with an identification of "cetml1config" is used to accommodate relatively independent configuration information of the CE TML, some readable and writable leaf nodes are used to abstract the configuration information except FEs list and a table node is used to abstract FEs list indexed by FE_ID. Finally, based on the IM, we build a DM in a form of YANG module with standard and detailed syntax of YANG language.

System implementation: NETCONF architecture is conceptually divided into 4 layers which are transport layer, RPC layer, operation layer and content layer from bottom to top. Figure 5 shows the implementation framework of NETCONF protocol. The framework is composed of a NETCONF manager and a NETCONF agent. The manager resides in the management plane of ForCES network, as shown in Fig. 3 and the agent resides in each CE or FE. Users submit NETCONF requests in the manager via the user interface and RPC request generator produce the corresponding <RPC> request and peer-to-peer communication processors is responsible for providing the services of transport layer. The RPC request parser in the managed CE or FE parses the <RPC> request and submits the result to the operation unit to be classified and processed. Information operation unit processes the corresponding managed objects and maps the managed information into corresponding YANG information. The YANG information is exactly the YANG DM. Finally agent sends a response message back to the manager.

Methods of performance evaluation: To discuss the performance of ForCES configuration management based on NETCONF, this study make a comparison between the method based on NETCONF and that based on SNMP. The main performance parameters include response time, network management traffic and numbers of transactions.

ASSUMPTIONS

Assumption 1: SNMP uses the version of SNMPv2c: A comparative analysis of performance between the network management model based on SNMPv2c and the one



```
+--rw cetml1config
|                 +--rw Local_IP_Address        Inet: ipv4-address
|                 +--rw Local_TCP_Port          Int32
|                 +--rw Local_UDP_Port          int32
|                 +--rw FEs_UDP_Port            int32
|                 +--rw Working_Type            working_type
|                 +--rw Security_Type           Security_type
|                 +--rw Priority_Type           boolean
|                 +--rw MsgArrive               boolean
|                 +--rw Congestion              int32
|                 +--rw peers_FE_ID_IP_list     [FE_ID]
|                      +--rw FE_ID              int32
|                      +--rw FE_IP              Inet: ipv4-address
```

Fig. 4: YANG tree style information model of the transport mapping layer in control element

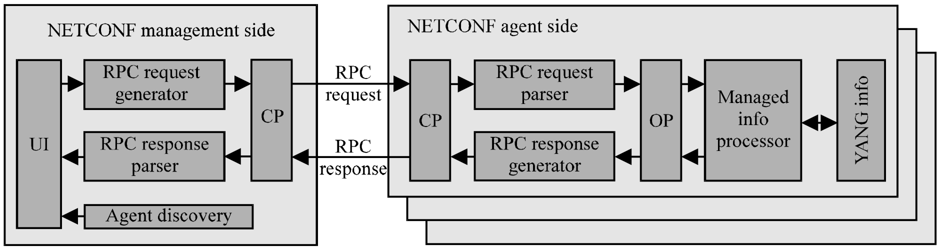


Fig. 5: Implementation block diagram of the system. UI: User interface, CP: Communication processor and OP: Operation processor

based on NETCONF is proposed. Currently the latest version of SNMP is SNMPv3. It increases the functions of encryption and authentication, but it is too complicated. So, what is commonly supported is still SNMPv2c by far.

Assumption 2: Just considering SNMP's and NETCONF's network management traffic packaged by the application layer in the comparison and taking no account of the traffic brought by the new features of NETCONF.

NETCONF owns some new features, such as connection-oriented, session-based, security, ability exchange and so on which are not equipped with SNMP. At the same time, these features are necessary to manage complex large-scale network. So, analysis of network management traffic should remove the part introduced by these new features as follows:

- SNMP is based on UDP, whose header is only 8 bytes; NETCONF is based on TCP, whose header is 20 bytes at least and 60 bytes at most. As a result, the header cost of transport layer in NETCONF is bigger than that in SNMP
- NETCONF bases on SSH to provide security, so it needs more cost of protocol header and data packaging. But SNMPv2c is only based on community to provide very simple security
- NETCONF is session-based while SNMP is session-less. So, NETCONF needs an extra cost of sessions creating and maintaining
- The client and severer of NETCONF exchange their abilities by <hello> message at the beginning. But SNMP has no ability exchange mechanism

Assumption 3: One SNMP configuring message only packed one MIB variable.

Although, an SNMP set message can carry multiple MIB variables in theory, those common SNMP tools such as MIB browsers only support setting a single MIB variable in one SNMP set message. In addition, the maximum size of SNMP response message is often configured by local devices. A response message carrying multiple MIB variables will often go beyond the size limit of the local device, then a 'TOOBIG' error will occur which leads to SNMP operation failure. Therefore, the assumption above is in general.

Network management traffic: L^{nq}_{req} denotes the length of the qth request message sent by a network management station to the nth managed device. L^{nq}_{req} denotes the length of the qth response message sent by the nth managed device to a network management station. N is the total number of managed devices. Q is the total number of variables needing to be queried of each managed device. L_{SNMP} denotes the SNMP traffic cost by inquiring NQ variables of N managed devices. L_{NETC} denotes the NETCONF traffic cost by inquiring NQ variables of N managed devices.

Definition 1: Based on the network management model of SNMP, the traffic cost by inquiring NQ parameters of N managed devices can be expressed as:

$$\begin{aligned} L_{SNMP} &= \sum\nolimits_{n=1}^{N} \sum\nolimits_{q=1}^{Q} \left(L^{nq}_{erq} + L^{nq}_{res}\right) = NQ\left(L_{Sreq} + L_{Sres}\right) \\ &= NQ\left[\left(H_{SNMP} + L_{OID}\right) + \left(H_{SNMP} + L_{OID} + L_{SOV}\right)\right] \\ &= 2NQH_{SNMP} + 2NQL_{OID} + NQL_{SOV} \end{aligned} \quad (1)$$

where, L_{Sreq} and L_{Sres} denote the average size of request messages and response messages of SNMP. H_{SNMP} denotes the average header length of SNMP messages; L_{OID} and L_{SOV} denote the average length of OID (Object ID) field and the average length of object value field in a SNMP message.

Definition 2: Based on the network management model of NETCONF, the traffic cost by inquiring NQ parameters of N managed devices can be expressed as:

$$\begin{aligned} L_{NETC} &= \sum\nolimits_{n=1}^{N} \sum\nolimits_{q=1}^{Q} \left(L^{nq}_{erq} + L^{nq}_{res}\right) = NQ\left(L_{Nreq} + L_{Nres}\right) \\ &= NQ\left[\left(H_{NETC} + QL_{Xpath}\right) + \left(H_{NETC} + QL_{Xpath} + QL_{NOV}\right)\right] \\ &= 2NH_{NETC} + 2NQL_{Xpath} + NQL_{NOV} \end{aligned} \quad (2)$$

where, L_{Nreq} and L_{Nres} denote the average size of request messages and response messages of NETCONF. H_{NETC} denotes the average header length of NETCONF messages, L_{Xpath} and L_{NOV} denote the average length of Xpath field and the average length of object value field in a NETCONF message.

Property 1: On the same scale of network (that is the managed device number N is equal), when management task (needing to request a managed device for Q variables) exceeds a certain value, there is $L_{SNMP}>L_{NETC}$.

Proof: Let $L_{SNMP}>L_{NETC}$, then $L_{SNMP}>L_{NETC}>0$. According to Definition 1 and 2:

$$2NQH_{SNMP}+2NGL_{OID}+NQL_{SOV}-2NH_{NETC}-2NQL_{Xpath}-NQL_{NOV}>0$$

Simplify it as:

$$Q(2H_{SNMP}+2NL_{OID}+L_{SOV}-2L_{Xpath}-L_{NOV})>2H_{NETC} \quad (3)$$

Because the Xpath addressing technique of NETCONF is more efficient than OID addressing technique of SNMP that is $L_{OID}>L_{Xpath}$. The managed variables are same that is $L_{SOV} = L_{NOV}$; In addition, $H_{SNMP}>0$. Then:

$$2H_{SNMP}+2L_{OID}+L_{SOV}-2L_{Xpath}-L_{NOV}>0$$

So, the Eq. 2 can be expressed as:

$$Q > \frac{2H_{NETC}}{2H_{SMNP} + 2L_{OID} + L_{SOV} - 2L_{Xpath} - L_{NOV}}$$

Therefore, when Q is greater than a certain value Q_0 (Q_0 is the smallest positive integer which is greater than $2H_{NETC}/(2H_{SNMP}+2L_{OID}+L_{SOV}-2L_{Xpath}-L_{NOV})$), there is $L_{SNMP}>L_{NETC}$. End.

Indicated by Property 1, the network management traffic of SNMP is less than that of NETCONF under the light management task (the Q value is small). But after the management task increases to a certain degree, the latter is less than the former. The essential reason is NETCONF is connection-oriented and needs only one interaction to configure a device while SNMP need Q interactions.

Response time: T^{nq}_{req} denotes the transmission time of the qth request message sent by a network management station to the nth managed device. T^{nq}_{req} denotes the transmission time of the qth response message sent by the nth managed device to a network management station. T^{nq}_{M} denotes the processing time of the request message for the qth variable of the nth managed device. T_{SNMP} denotes the response time for NQ variables of N managed devices in SNMP. T_{NETC} denotes the response time for NQ variables of N managed devices in NETCONF.

Definition 3: Based on the network management model of SNMP, the response time of NQ variables of N managed devices can be expressed as:

$$T_{SNMP} = \sum{}_{n=1}^{N}\sum{}_{q=1}^{Q}\left(T^{nq}_{req} + T^{nq}_{res} + T^{nq}_{M}\right) = NQ\left(T_{Sreq} + T_{Sres} + T_{M}\right) \quad (4)$$

where, T_{Sreq}, T_{Sres} and T_M, respectively denote the average transmission time of SNMP request messages, the average transmission time of SNMP response messages and the average processing time of managed devices.

Definition 4: Based on the network management model of NETCONF, the response time of NQ variables of N managed devices can be expressed as:

$$T_{NETC} = \sum{}_{n=1}^{N}\sum{}_{q=1}^{Q}\left(T^{nq}_{req} + T^{nq}_{res} + T^{nq}_{M}\right) = N\left(T_{Nreq} + T_{Nres} + T_{M}\right) \quad (5)$$

where, T_{Nreq}, T_{Nres} and T_M, respectively denote the average transmission time of NETCONF request messages, the average transmission time of NETCONF response messages and the average processing time of managed devices.

Property 2: On the same scale of network (that is the managed device number N is equal), when management task (needing to request a managed device for Q variables) exceeds a certain value, there is $T_{SNMP}>T_{NETC}$.

Proof: Let $T_{NSMP}>T_{NETC}$, then $T_{SNMP}-T_{NETC}>0$. According to Definition 3 and 4:

$$NQ\left(T_{Sreq} + T_{Sres} + T_{M}\right) - N\left(T_{Nreq} + T_{Nres} + T_{M}\right) > 0,$$

$$Q > \frac{T_{Nreq} + T_{Nres} + T_{M}}{T_{Sreq} + T_{Sres} + T_{M}}$$

Therefore, when Q is greater than a certain value Q_0 (Q_0 is the smallest positive integer which is greater than $T_{Nreq}+T_{Nres}+T_{M})/(T_{Sreq}+T_{Sres}+T_{M})$), there is $T_{SNMP}>T_{NETC}$. End.

Indicated by Property 2, the response time of SNMP is shorter than that of NETCONF under the light management task (the Q value is small). But after the management task increases to a certain degree, the latter is shorter than the former. The essential reason is NETCONF is connection-oriented and needs only one interaction to configure a device while SNMP need Q interactions.

RESULTS AND DISCUSSION

In the test system, NETCONF entity is developed based on an open source package called Yuma and SNMP entity is developed based on an open source package called net-snmp. The test management task is to configure the same managed object of the same device repeatedly. The performance of the configuration model based on NETCONF and the one based on SNMP are tested. SNMP uses the "set" message to do the configuration and NETCONF uses the "edit-config" message. The tested performances include network management traffic, response time and Number of Transactions.

Network management traffic: The test result of network management traffic is presented in Table 1, where the network traffic contains all of the data flow that is NETCONF network management traffic contains what is

Table 1: Network management traffic comparison between NETCONF and SNMP

	Network management traffic (byte)	
Managed object (N)	SNMP	NETCONF
1	194	10930
10	1940	11794
100	19400	21734
1000	194000	118790
10000	1940000	1057548

Table 2: Response time comparison between NETCONF and SNMP

	Response time (sec)	
Managed objects (N)	SNMP	NETCONF
1	0.001082	0.002425
10	0.522434	0.004990
100	5.734452	0.096478
1000	55.225364	1.613860
10000	553.546464	18.076838

Table 3: No. of transactions comparison between NETCONF and SNMP

	No. of transactions	
Managed objects (N)	SNMP	NETCONF
1	1	1
10	10	1
100	100	1
1000	1000	1
10000	10000	1

cost by establishing connections and configuring managed objects and that of SNMP contains what is cost by configuring managed objects only. Shown as Table 1, the test result accords with the theoretical analysis of Property 1, where $Q_0 \approx 100$. When configured objects are less than Q_0, SNMP's traffic cost is obviously less than NETCONF's. The reason is that NETCONF is added with some new features, such as connection-oriented, session-based, security, ability exchange and so on.

With the increase of managed objects, when the amount is over Q_0 (about 100), the NETCONF network management traffic is less than SNMP. The reason is that a NETCONF message can pack a large number of managed objects because of its session-based feature while an SNMP message of mainstream SNMP application can only pack one managed object, then SNMP needs more interactions to complete large network management tasks.

The test indicates that NETCONF has not only added with new features, such as security, handshake mechanism, session mechanism and so on, but also has better performance than SNMP on network management traffic in a large-scale network environment. Therefore, considering the performance of network management traffic, NETCONF is more adaptable to SDN than SNMP.

Response time: The test result of response time is presented in Table 2, where the response time refers to the time cost of completing the whole management task. Our test doesn't consider the time of building a session, but only considers the time of configuration management. Shown as Table 2, the test result accords with the theoretical analysis of Property 2 where $Q_0 \approx 2$. When managed objects are less than Q_0, the response time of SNMP is shorter, because the introduced new features bring NETCONF brings extra processing time. With managed objects increasing, when the number is more than Q_0 (about 2), the response time of NETCONF is shorter, because the advantage that a NETCONF message can pack a number of variables begins to play a role. According a same configuration management task, NETCONF can complete within only one interaction while SNMP needs a number of interactions.

The test indicates that the performance of NETCONF is absolutely better than that of SNMP on response time in large-scale network environment with heavy management tasks. Therefore, considering the performance of response time, NETCONF is more adaptable to SDN than SNMP.

No. of transactions: The test result of the Number of Transactions is presented in Table 3 which indicates NETCONF is obviously better than SNMP. NETCONF is session-based and only one transaction is needed to configure a device while SNMP is not session-based and the Number of Transactions needed is equal to the number of exchanges of messages.

The weakness of SNMP on transaction will limit many developments and applications of configuration management. First, to handle a big quantity (e.g., 10000) of operations alone greatly enhances the complexity of the development of management tools, devices and software. Second, SNMP has no capability of backup and recovery for device configuration which can be easily done with a single transaction operation by NETCONF. The last, SNMP can't verify the configuration, but NETCONF can support.

The test result indicates that the performance of NETCONF is absolutely better than that of SNMP on Number of Transactions in large-scale network environment with heavy management tasks. Therefore, considering the performance of Number of Transactions, NETCONF is more adaptable to SDN than SNMP.

CONCLUSION

Current Internet architecture has increasingly exposed its drawbacks of inflexible, so a clean-slate network architecture is needed. SDN is a kind of new generation network architecture. ForCES is a technology of control and forwarding element separation in net-elements. This study extends ForCES technology from net-element to network and proposes an SDN framework realized by ForCES technology. The original local CLI configuration management mode of ForCES no longer adapts to the SDN network environment, this study proposes a new configuration management method based on NETCONF for ForCES network. Configuration information is modeled by YANG and configuration protocol is NETCONF. Both theoretical analysis and test results indicate that the ForCES configuration management method based on NETCONF is not only adaptable to large-scale SDN network environment, but also superior to that based on traditional SNMP on performance. The large-scale SDN network will be a multi-domain network and the corresponding configuration management model should also be hierarchical and multi-domain, it is our further research object.

REFERENCES

Bjorklund, M., 2010. YANG-A data modeling language for the network configuration protocol (NETCONF). http://tools.ietf.org/pdf/rfc6020.pdf

Case, J., K. McCloghrie, M. Rose and S. Waldbusser, 1996. Management information base for version 2 of the simple network management protocol (SNMPv2). http://www.ietf.org/rfc/rfc1907.txt.

Case, J., M. Fedor, M. Schoffstall and J. Davin, 1990. A simple network management protocol (SNMP). Internet Engineering Task Force, RFC1157.

Enns, R., M. Bjorklund, J. Schoenwaelder and A. Bierman, 2011. Network configuration protocol (NETCONF). http://tools.ietf.org/pdf/rfc6241.pdf

Khosravi, H. and T. Anderson, 2003. Requirements for separation of IP control and forwarding. Network Working Group.

McCloghrie, K., D. Perkins and J. Schoenwaelder, 1999. Structure of management information version 2 (SMIv2). http://tools.ietf.org/pdf/rfc2578.pdf.

Nadeau, T. and P. Pan, 2011. Framework for software defined networks. http://tools.ietf.org/pdf/draft-nadeau-sdn-framework-01.pdf

Perkins, C., 2002. IP Mobility support for Ipv4. Internet Engineering Task force RFC 3344.

Stiliadis, D., F. Balus, W. Henderickx, N. Bitar and M. Pisica, 2011. Software driven networks: Use cases and framework. http://tools.ietf.org/pdf/draft-stiliadis-sdnp-framework-use-cases-01.pdf.

Wang, W., K. Ogawa, E. Haleplidis, M. Gao and J. Hadi Salim, 2013. Interoperability Report for Forwarding and Control Element Separation (ForCES). http://tools.ietf.org/pdf/draft-ietf-forces-interop-05.pdf

Wang, W., L. Dong, B. ZhuGe, C. Li, M. Gao, R. Jin and J. Zhou, 2010. Forwarding and Control Element Separation Technology and Application. Zhejiang University Press, Hangzhou, ISBN: 978-7-308-08296-9.

Yang, L., R. Dantu, T. Anderson and R. Gopal, 2004. Forwarding and control element separation (ForCES) framework. RFC 3746. Network Working Group, The Internet Society, April, 2004. http://www.elook.org/computing/rfc/rfc3746.html

Ylonen, T. and C. Lonvick, 2006. The secure shell (SSH) protocol architecture. http://www.ietf.org/rfc/rfc4251.txt