

Implementation and Evaluation of Network Management System to Reduce Management Cost Caused by Server Virtualization

Masahiro Yoshizawa, Toshiaki Tarui and Hideki Okita

Abstract—The cost of managing data center networks is increasing because data center operators have to use different Network Management Systems (NMSs) for existing network elements and virtual network elements created by server virtualization. This paper presents an integrated NMS to manage both network elements in an integrated fashion. The NMS is based on a four-layer information model from a physical layer to a logical ethernet layer. The NMS based on the information model can create new network configurations of both network elements from the operator's operations in the logical ethernet layer. We developed a prototype NMS and conducted experiments. The results indicated that the prototype reduced operator errors when they created new VLAN configurations related to both elements. This suggests that the information model is effective for reducing the cost of managing data center networks.

Index Terms—Network Management System (NMS), Virtual LAN (VLAN), Server Virtualization, Data Center

I. INTRODUCTION

With recent virtualization technologies, existing systems have been integrated into a small number of data centers. Hardware and software vendors have developed virtualization technologies that enable server virtualization [1-2], I/O virtualization [3-4], and configuration management between virtual servers and their I/O devices [5-6].

However, these virtualization technologies may actually increase the cost of managing data center networks due to an increased workload for data center operators. There are two main causes for this.

The first is that the virtualization technologies will increase the number of virtual network elements (e.g., virtual servers and virtual switches) that the data center operators have to manage. When the operators start, stop, and change configurations of the virtual network elements, they have to use different management interfaces from those of the existing physical network elements. For example, the operators use a Virtual Local Area Network (VLAN) to create a virtual

network that includes some servers without changing wiring in the data centers. In data centers with server virtualization technologies, the operators must consider both the physical and virtual network elements for managing VLANs at the same time.

The second cause is that network configurations need to be changed more frequently in a server virtualization environment. It is an advantage of the virtualization technologies that operators can easily create, delete, and move virtual network elements. However, the migration of virtual network elements requires frequent configuration changes.

Therefore, an integrated system to manage physical and virtual network elements is needed to decrease the costs of managing data center networks when virtualization technologies are introduced into the data centers. Many systems have been developed to enable integrated management of virtual and physical servers and of virtual and physical switches. However, the integrated management technologies for the entire virtual network have not been sufficiently standardized. So far, they have not sufficiently supported a mix of both physical and virtual servers as well as heterogeneous hypervisors. If some of the specifications for the integrated management of the entire virtual network can be standardized, the costs of network management in a server virtualization environment can be reduced.

In this paper, we propose a Network Management System (NMS) that allows data center operators to manage physical network devices and virtual network devices in an integrated fashion. In addition, we propose a four-layer information model for the NMS that represents the entire virtual network.

II. PROBLEMS OF SERVER VIRTUALIZATION ENVIRONMENT

A. Characteristics of Server Virtualization Environment

Server virtualization technologies such as VMware and Virtage¹ make virtual network elements (e.g., virtual switches and virtual network address translation (NAT) devices) in each physical server. Such virtual network elements stand between virtual servers and physical network elements and work in the

M. Yoshizawa, T. Tarui, and H. Okita are with Hitachi, Ltd., Central Research Laboratory, Tokyo, Japan (e-mails: {masahiro.yoshizawa.bt, toshiaki.tarui.my, hideki.okita.pf}@hitachi.com).

¹ "Virtage" is a registered trademark for Hitachi's hardware partitioning system. This trademark is only used in the United States and Japan.

same way as physical network elements. For example, virtual switches support common Ethernet switch functions like VLAN.

However, when the data center operators start, stop, and change configurations of the virtual network elements, they have to use management interfaces for virtual servers instead of the existing network management interfaces.

For example, Virtage [2] can partition physical server resources by constructing multiple logical partitions (LPARs) that are isolated, and each of these environments can run independently. With Virtage, virtual switches are created in each network interface card (NIC) of the physical servers. In addition, the connections between the LPARs and the virtual switches and the VLAN configurations of the virtual switches are included in the configurations of the LPARs. Therefore, when data center operators change a VLAN configuration of a virtual switch, they have to log in to a remote console of Virtage and change part of the LPAR configuration through a Virtage-specific management interface.

B. Network Management in Server Virtualization Environment

In the server virtualization environment, the costs of managing data center networks are increasing more than ever before. Two main reasons for this are outlined below.

Different Management Interfaces

The first reason is that physical network elements and virtual network elements that are created by server virtualization technologies have different management interfaces. We interviewed some data center operators about this difference. They were concerned that the different interfaces require additional operations in each phase of network management (e.g., designing, constructing, and monitoring the network).

In the design phase, the operators have to create the configurations of both the physical and virtual network elements that are included in the server configurations. For this operation, the operators need sufficient knowledge of both the servers and the network switches. In existing data centers, however, server operators and network operators are often distinct from each other. In this case, the server operators and the network operators have to work together closely to create the configurations, and this increases the costs of communication between them.

In the construction phase and the monitoring phase, the operators must create operations manuals based on the configurations created in the design phase. Due to the difference in management interfaces between the physical and virtual network elements, separate operations manuals are required.

Increase in Frequency of Changing Network Configurations

The second reason is that it is necessary to change network configurations much more frequently in a server virtualization environment. Virtual servers can be added, removed, and

moved around among some physical servers. This flexibility is regarded as an advantage of server virtualization. However, these operations require changes in the configurations of the physical and virtual network elements. Such configurations include those for the VLAN, quality of service (QoS), and access control list (ACL). Therefore, the costs of network management increase even further.

C. Problems of Server Virtualization Environment

As a result, there are two main network management problems in a server virtualization environment.

Problem 1: Integration of Management Interfaces between Physical Network Elements and Virtual Network Elements

The differences in management interfaces for these elements results in duplicate operations at each phase of network management. Thus, the management interfaces need to be integrated in order to avoid these duplicate operations.

Problem 2: Automation of Network Management Operations

In server virtualization environments, network configurations change more frequently. It is therefore necessary to automate the operations at each phase of network management in order to reduce the data center operators' workloads.

III. RELATED WORK

Much research has been done on the development of integrated management of physical and virtual switches. The research includes the improvement of the function of virtual switches, and the standardization of the information model of virtual switches.

Improving Function of Virtual Switches

The research improving the function of virtual switches is classified into two broad categories: improving management interfaces of virtual switches, and aggregating the function of the virtual switches into fewer physical switches.

Nexus 1000V [7] and Arista vEOS [8] are examples of the improved virtual switches that support the Simple Network Management Protocol (SNMP) and a familiar industry-standard command-line interface (CLI). With such improved virtual switches, the network operators can manage both the physical and virtual switches in the same way. However, it is still difficult to configure and monitor large number of virtual switches that are distributed into data center networks.

IEEE 802.1Qbg Edge Virtual Bridging (EVB) [9] and 802.1Qbh Bridge Port Extension [10] are standards that are intended to aggregate the functions of virtual switches on hypervisors into fewer physical switches. The standards are now under discussion. Cisco VN-Link technology [11] is a product technology to enable the aggregation that IEEE 802.1Qbg/h are intended to provide.

If the functions of virtual switches are aggregated into fewer

physical switches, the operators only have to manage the fewer physical switches, and the costs of network management can be reduced. For example, when the operator moves a virtual server from one physical server to another, the configurations of the physical and virtual switches related to the virtual server (e.g. VLAN configurations) can be migrated with the virtual server.

However, such technologies have not been standardized yet, and have not sufficiently supported a mix of both physical and virtual servers as well as heterogeneous hypervisors.

Standardizing Information Model of Virtual Switches

The Distributed Management Task Force (DMTF) has developed platform-independent standards to manage a virtualized environment. The standards are based on the Common Information Model (CIM) [12], which is a standard for defining how managed elements in an IT environment are represented.

Hypervisor vendors have defined their CIM extension models to provide a CIM interface to build management applications. For example, VMware CIM APIs [13] provide the model for VMware, and Xen-CIM project [14] provides the model for Xen. W. Fuertes, et al. [15] defined a generic CIM extension model that supports heterogeneous virtual switches, and implemented a CIM client based on the model. In addition, the DMTF is defining standards for managing virtual network elements [16-17].

As described above, the models for virtual switches are being standardized. When such a generic model will be defined and standardized, it will be easy to build NMSs to support virtual switches on heterogeneous hypervisors. However, the simple model for representing an entire virtual network that supports a mix of both physical and virtual servers as well as heterogeneous hypervisors has not been standardized yet.

In this paper, we propose an information model that can be used to represent the entire virtual network.

IV. PROPOSAL OF VIRTUAL NETWORK MANAGEMENT INFORMATION MODEL

We propose a Network Management System (NMS) that supports network management in a server virtualization environment. To integrate management interfaces between physical network elements and virtual network elements, the NMS treats the configurations of each element based on an information model for the virtual-network management information. This section describes the information model.

A. Requirements

The requirements for the information model are listed below. These requirements arise from the two problems stated previously.

Requirement 1: The NMS based on the information model can create information about logical networks that consist of physical network elements and virtual network elements from their configurations.

Requirement 2: The NMS based on the information model can create new configurations of the elements and operating procedures in order to set the configurations based on the operator's request to the logical networks.

Requirement 3: The information model that the NMS is based on can be used to represent a mix of both the physical and virtual servers and heterogeneous hypervisors.

If the NMS meets Requirement 1, the operator can figure out connections between physical network elements and virtual network elements through the NMS. If the NMS meets Requirement 2, the operator can change the configurations of the physical network elements and virtual network elements in an integrated fashion through the NMS. If the NMS meets Requirement 3, the operator can manage the entire data center network with the integrated NMS.

The details of the information model are as follows.

B. Multilayered Structure

The virtual network management information model has a multilayered structure. The information model has a basic layer called the "Physical Layer" that represents the configurations and connections of the physical elements, and three logical layers that represent the configurations and connections of the logical elements. The highest logical layer, called the "Logical Ethernet Network Layer," corresponds to a VLAN. More than one logical layer is defined here, because it allows data center operators to see the logical network from different perspectives.

The information model represents the connection of elements of each layer as combinations of "Nodes," "Interfaces," and "Links." A node has a number of interfaces. A link is a connection between the interfaces. The nodes interconnected through the interfaces and the links are called a "Network" in the information model. This basic structure is common to all layers because the same functions can be used in all layers.

Fig. 1 is an example of a network represented using the information model. The assumptions of the example are as follows:

- LPAR 1 of Blade 0 is connected to a VLAN that has VLAN ID 10 through a virtual NIC (vNIC).
- LPAR 2 of Blade 0 is connected to a VLAN that has VLAN ID 20 through a vNIC.
- LPAR 3 of Blade 1 is not connected to any VLAN. The operators have not created a vNIC of the LPAR yet.

Each layer has the following characteristics:

(1) Physical Layer

This layer represents the configurations and connections of the physical elements, i.e. physical servers and physical network switches. A physical node has a field that can contain the configurations of a physical element in a device-specific data format. This layer also has "Procedure" elements that each

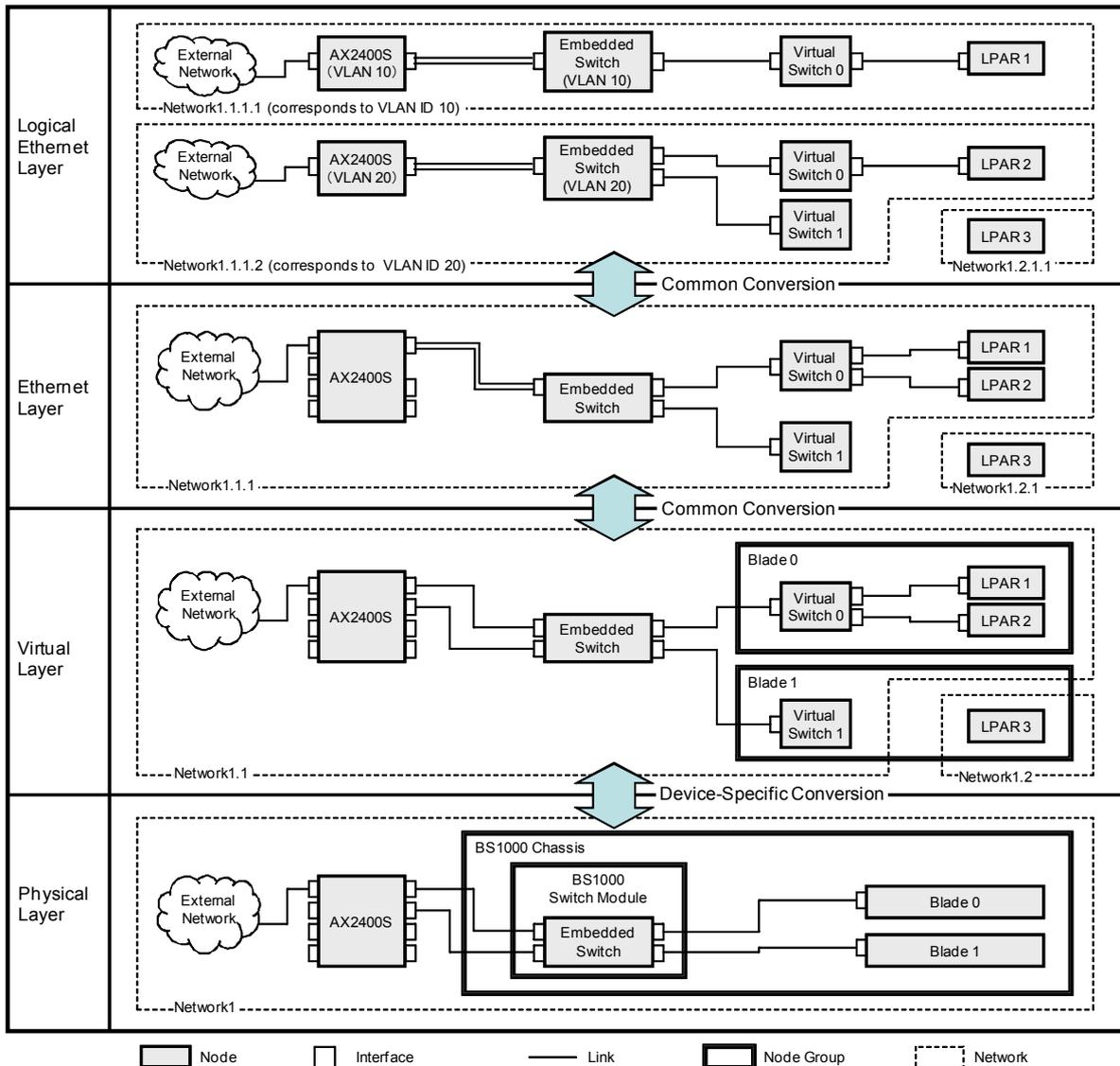


Figure 1. Multilayered Structure

correspond to an operating procedure to change the configurations.

(2) Virtual Layer

This layer represents the configurations and connections of the logical elements, i.e. physical servers, physical switches, virtual servers, and virtual switches. In the virtual layer and the higher layers, the configurations are formatted into common data formats.

In the future, any layer other than the Ethernet Layer can be defined above the virtual layer. For example, a Fiber Channel Layer for a storage area network (SAN) can be defined above the virtual layer.

(3) Ethernet Layer

This layer represents the configurations and the connections of the logical elements that have Ethernet support. In this layer, special nodes that are used as Ethernet switches are called

"Hubs." This layer represents the area where the nodes can communicate with each other through the hubs if the node configurations (e.g. VLAN configurations) are properly set as an ethernet network. And in this layer, virtual links in the same link aggregation group are bundled into one link.

(4) Logical Ethernet Layer

This layer is similar to an Ethernet Layer. This layer, however, represents the area where the nodes can communicate with each other through the hubs with current node configurations as a logical ethernet network. In other words, one logical ethernet network corresponds to one VLAN.

If one ethernet node belongs to a number of VLANs, the ethernet node is divided into a number of logical ethernet nodes that belong to each logical ethernet network that corresponds to each VLAN. AX2400S in Fig. 1 is an example of such a node.

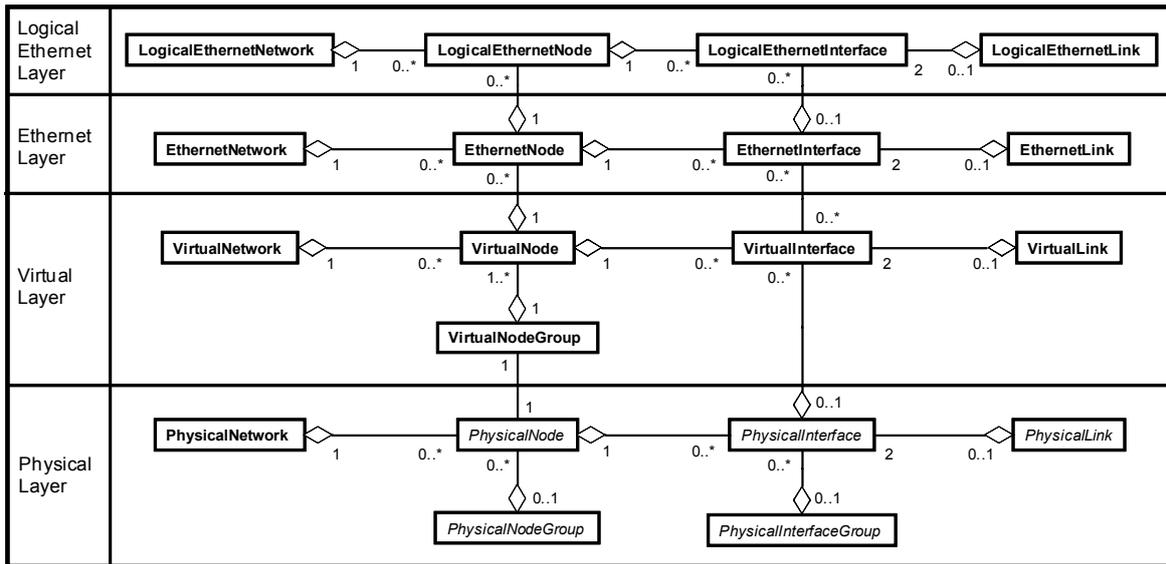


Figure 2. Class Diagram of Virtual Network Management Information Model

C. Creation of Logical Networks

Data of each layer can be created from the data of its lower layer. For example, the data of the virtual layer can be created from the data of the physical layer. Thus, a number of NMSs can exchange data on the configurations and connections of the logical elements with each other if the NMSs exchange only the data of the physical layer.

Conversion processes in the logical layers (i.e., conversion processes between the virtual layer and the ethernet layer and between the ethernet layer and the logical layer) can be common because the configurations are formatted into common data formats in the virtual layer and the higher layers. However, conversion processes between a physical layer and a virtual layer must be device-specific, because the configurations are formatted into a device-specific data format.

Here, we address the conversion processes between a physical layer and a virtual layer as plugins for a software library of the information model. If the operators want to manage a new physical element which may include some virtual elements, the operator adds some plugins that correspond to the elements of the new physical element (i.e., physical nodes, physical interfaces, and physical links) into the software library.

Each plugin must include the following data format and processes.

- A data format of the device-specific configurations
- A process to create the configurations and connections of the virtual layer from the device-specific configurations
- A process to create operating procedures to change the device-specific configurations and change the device-specific configurations from operations in the virtual layer

This information model and the software library of the information model meet Requirement 1--3.

D. Implementation of Virtual Network Management Information Model

The following is an example of the implementation of the virtual network management information model in Java.

Class Design

Fig. 2 is a class diagram of the virtual network management information model. For the multilayered structure, the classes corresponding to the elements of each layer are defined. For example, PhysicalNode class, which corresponds to a physical node, and LogicalEthernetNode class, which corresponds to a logical ethernet node, are defined.

For the plugin structure of the physical layer, all of the elements of the physical layer are defined as abstract classes. For example, PhysicalNode class is defined as an abstract class. The device-specific data format and processes are defined as child classes that inherit the abstract classes. For example, the Bs1000Blade class, which corresponds to a Hitachi blade server that supports Virtage, inherits the PhysicalNode class.

Fig. 3 is an example of the implementation of a PhysicalNode class and its child classes.

The virtualize() method is used to create the configurations and connections of a virtual layer from the device-specific configurations. This method creates a set of virtual nodes (i.e., VirtualNodeGroup class) that correspond to the physical node.

The applyProcedures() method is a method to create operating procedures to change the device-specific configurations. This method creates a set of the operating procedures (i.e., PhysicalProcedure class) from a set of operations in the virtual layer (i.e., VirtualProcedure class). At the same time, this method changes the data of the configurations in the PhysicalNode object.

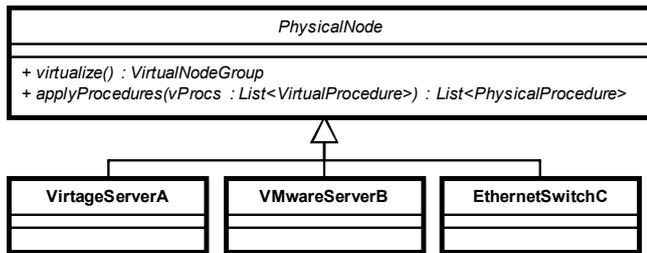


Figure 3. Class Diagram of PhysicalNode class and its Child Classes

Example of Creating VLAN Configuration

When the operator creates a VLAN configuration for connecting a virtual node (i.e., a virtual machine or an LPAR) to a logical ethernet network (i.e., VLAN), the software in the NMS checks the connections of each layer starting from the top (i.e., logical ethernet layer) and moving down (i.e., physical layer). If there is no connection between the virtual node and the other nodes that belong to the targeted logical ethernet network in a layer, the software creates a set of operating procedures to connect them in the layer. In a stepwise fashion, the software can finally create a complete set of operating procedures.

V. EXPERIMENTS AND EVALUATIONS

A. Experiments

To evaluate the advantages of the NMS based on the information model, we developed a prototype NMS and conducted experiments. In these experiments, examinees created VLAN configurations with and without the prototype, and an examiner measured the time required to create them.

Prototype

The prototype consists of a software library of the information model and a CLI to create the VLAN configurations with the software library. Table I is a list of the CLI commands.

Three types of data of the physical layer (Table II) were embedded in the prototype for the experiments. The load command loads each kind of data and creates data of the virtual layer and its higher layer from the loaded data. At this time, the prototype supports two types of physical nodes: Hitachi's blade server BladeSymphony² 1000 (BS1000) and Alaxala's gigabit layer 2 switch AX2400S. In these experiments, one chassis of the BS1000 consists of from one to four blades and one embedded layer 2 switches.

Details of the Experiments

In these experiments, the examinees were instructed to create operating procedures for connecting a specified LPAR to a specified VLAN in two different networks. Question one involved a small network. Question two was more difficult, as

² "BladeSymphony" is a registered trademark for Hitachi's blade server system. This trademark is used in the United States, the EU, China, Korea, and Japan.

TABLE I
LIST OF CLI COMMANDS

Type	No. of Cmds	Commands
Basic	5	Commands to load data of the physical layer (load), save data that are changed by the design commands (commit), and so on.
Show	6	Commands to show LPARs and their IDs (lparlist), show switches and their IDs (switchlist), show logical ethernet networks and their IDs (Inshow), show operating procedures (show-p-proc), and so on.
Design	1	Commands to change the configurations of the physical elements and create the operating procedures to connect a virtual node (i.e., a virtual machine or a LPAR) to a logical ethernet network (i.e., a VLAN) (Inconnect). Arguments of the design command are the elements' IDs (e.g., logical ethernet network ID), which are assigned by the prototype.

TABLE II
OPERATION TESTS GIVEN TO EXAMINEES

Test	BS1000 Chassis	BS1000 Blade	LPAR	AX2400S
Example	1	2 (2 blades per chassis)	2 (1 LPAR per blade)	1
Question 1	2	4 (2 blades per chassis)	8 (2 LPARs per blade)	1
Question 2	4	14 (3-4 blades per chassis)	26 (1-2 LPARs per blade)	4

it involved a large network.

We provided a network diagram and configuration sheets of each question to the examinees. The examinees created the operating procedures with the documents and wrote them on an answer sheet. The prototype automatically creates the operating procedures and displays them on the screen, but the examinees had to write them on the answer sheet in order to compare the results with the prototype and without the prototype. The examinees handed in their answer sheets to the examiner, who checked whether the answers were correct. During this check, the clock measuring the time to complete the assignment was stopped. If the answer was wrong, the examiner gave the answer sheet back to the examinee and asked him to revise the answer, and the clock was restarted. They repeated this exchange until the examinee handed in the correct answer sheet to the examiner.

In these experiments, we used the same questions for the cases with and without the prototype. This affected the time required to create the operating procedures, because the examinees made adjustments as they worked on the operations and created the VLAN configurations in degrees. Thus, we divided the examinees into group A and group B. Group A used the prototype in the first experiment. Group B used the prototype in the second experiment.

Procedures

We used the following procedure in the experiments:

- 1) Explanation on VLAN configuration (about thirty minutes)

The examiner explains to the examinee the VLAN configurations of BS1000 and AX2400S and how to change them. This information is needed in order to ask the following questions.

- 2) Explanation of the Prototype NMS (about thirty minutes)
The examiner explains to the examinee the prototype and how to create the operating procedures with the prototype. At this time, the examinee uses the prototype and asks an example question with the prototype.
- 3) The First Experiment: Question One and Question Two
The examinees in group A use the prototype. The examinees in group B do not use the prototype.
- 4) The Second Experiment: Question One and Question Two
The examinees in group A do not use the prototype. The examinees in group B use the prototype.

Examinees

Six examinees took part in these experiments. Table III gives some brief background information on each person.

B. Results

Tables IV and V list the time required to answer and the results of the tests. Table VI indicates the reduced time with the prototype and the percentage of the reduced time to the time required in the experiments without the prototype.

C. Evaluations

The experiments showed the following two results:

- 1) The prototype helped the examinees reduce mistakes when they created a VLAN configuration. The result implies that the prototype is effective to reduce the cost of managing data center networks.
- 2) The prototype reduced the time required to create the operating procedures, although the amount of time was small.

Consideration on the Examinees' Mistakes

Table VI shows that the examinees made small and large mistakes in the experiments without the prototype. This could have been due to the complexity of the networks in the test, in which physical elements and virtual elements were mixed. On the other hand, Table V shows that the examinees made few mistakes in the experiments with the prototype. The only mistake was one that an examinee made when transferring the displayed procedures on the screen to the answer sheet. This could have been due to the automatic design by the prototype.

The reduction in the mistakes made to create the configurations would reduce the cost of network management in each phase. In data centers, the operators carefully examine the new configurations during the design phase (e.g., through peer review). Having fewer mistakes would reduce the time required to examine the new configurations. In addition, operators have to recreate operations manuals in the

TABLE III
BRIEF BACKGROUNDS OF EXAMINEES

Examinee	Type of Job	Experience in BS1000 setting	Experience in VLAN setting
1	Operator (Server)	NO	YES
2	Researcher	YES	NO
3	Researcher	YES	NO
4	Operator (Server)	YES	YES
5	Operator (Server)	NO	NO
6	Operator (Server)	NO	NO

TABLE IV
TIME REQUIRED TO ANSWER AND THE RESULT OF THE TESTS
WITHOUT THE NMS

Examinee	Question 1	Question 2
1	13 min 3 sec 1st answer is correct but includes redundant procedure.	21 min 18 sec 1st answer is correct but includes redundant procedure.
2	17 min 36 sec 1st answer is wrong and 2nd answer is correct.	14 min 34 sec 1st answer is wrong and 2nd answer is correct.
3	7 min 56 sec 1st answer is correct but includes redundant procedure.	10 min 0 sec 1st answer is correct.
4	15 min 24 sec 1st answer is correct but includes redundant procedure.	11 min 9 sec 1st answer is correct.
5	20 min 44 sec 1st answer is correct but includes redundant procedure.	18 min 44 sec 1st answer is correct.
6	14 min 1 sec 1st answer is wrong and 2nd answer is correct.	12 min 1 sec 1st answer is correct.
Average	14 min 47 sec	14 min 37 sec
Standard deviation	3 min 58 sec	4 min 7 sec

TABLE V
TIME REQUIRED TO ANSWER AND THE RESULT OF THE TESTS
WITH THE NMS

Examinee	Question 1	Question 2
1	12 min 25 sec 1st answer is correct.	21 min 10 sec 1st answer is correct.
2	11 min 16 sec 1st answer is correct.	21 min 19 sec 1st answer is correct.
3	5 min 17 sec 1st answer is correct.	6 min 51 sec 1st answer is correct.
4	13 min 49 sec 1st answer is correct.	13 min 28 sec 1st answer is correct.
5	18 min 11 sec 1st answer is correct.	15 min 7 sec 1st answer is correct.
6	9 min 57 sec 1st answer is correct.	11 min 30 sec 1st answer is wrong and 2nd answer is correct.
Average	11 min 49 sec	14 min 57 sec
Standard deviation	3 min 54 sec	5 min 9 sec

construction and monitoring phases if the configurations are wrong. The smaller number of mistakes means that less time would be required to recreate such operations manuals.

TABLE VI
REDUCED TIME WITH THE NMS

Examinee	Question 1		Question 2	
1	38 sec	(4.9 %)	8 sec	(0.6 %)
2	6 min 20 sec	(36.0 %)	- 6 min 45 sec	(-46.3 %)
3	2 min 39 sec	(33.4 %)	3 min 9 sec	(31.5 %)
4	1 min 35 sec	(10.3 %)	- 2 min 19 sec	(-20.8%)
5	2 min 33 sec	(12.3 %)	3 min 37 sec	(19.3 %)
6	4 min 4 sec	(29.0 %)	31 sec	(4.3 %)

Consideration on the Time Required to Create the Operating Procedures

Table VI shows that with the prototype, examinees took less time to answer the questions in many cases; a time reduction of 36.0 percent was achieved in Question 1 for Examinee 1 compared to the experiment without the prototype. On the other hand, with the prototype, Examinees 2 and 4 took longer to complete Question 2.

One reason for this is that it was difficult for the examinees to use the prototype's CLI, so it took time for them to understand how to use the prototype. If we implement some useful functions that ordinary network management software has (e.g., command history and commands and element IDs that are suggested automatically with partial operator input), the usability of the prototype can be improved.

VI. CONCLUSION

This paper presents a network management system (NMS) to manage both existing network elements and virtual network elements in an integrated fashion in order to reduce the cost of managing data center networks. The NMS is based on a virtual network management information model to manage both network elements.

We developed a prototype NMS based on the information model and conducted experiments. The experiments showed that the prototype reduced operator errors when they created a VLAN configuration. This reduction in mistakes reduces the cost of network management at each phase. This result suggests that the NMS would be effective in reducing the cost of managing data center networks that include a server virtualization environment.

Our next step is to apply the NMS to other phases of network management (construction phase and monitoring phase) and evaluate whether the NMS is advantageous in those phases.

We will also implement the plugins that support other hypervisors to verify the feasibility of the information model. We intend to standardize the information model, and conform part of the information model to existing standards as much as possible. We are proposing the information model in the Internet Engineering Task Force (IETF) Operations & Management (OPS) area working group [18].

ACKNOWLEDGMENT

We would like to thank Y. Yamamoto, S. Orikawa, K. Nagahiro, and T. Kobayashi at Hitachi Systems & Services, Ltd.

for their insightful comments. We also express our gratitude to K. Naono, K. Uehara, and K. Mizuno at Hitachi, Ltd., Central Research Laboratory for encouraging us to conduct this research project.

REFERENCES

- [1] VMware vSphere 4, <http://www.vmware.com/products/vsphere/>
- [2] Virtage, <http://www.hitachi.co.jp/virtage/>
- [3] Cisco Nexus Series, http://www.cisco.com/web/JP/product/hs/switches/dc_switches.html
- [4] Xsigo VP780, <http://www.xsigo.co.jp/products/index.html>
- [5] IBM Dynamic Infrastructure, <http://www-03.ibm.com/systems/dynamicinfrastructure/>
- [6] HP Virtual Connect technology for the HP BladeSystem c-Class, 4th edition, HP whitepaper, HP, 2010. Available: <http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00814156/c00814156.pdf>
- [7] Cisco Nexus 1000V Series Switches, <http://www.cisco.com/en/US/products/ps9902/>
- [8] vEOS - Virtualized Extensible Operating System, <http://www.aristanetworks.com/en/products/veos>
- [9] IEEE 802.1Qbg Edge Virtual Bridging, <http://www.ieee802.org/1/pages/802.1bg.html>
- [10] IEEE 802.1Qbh Bridge Port Extension, <http://www.ieee802.org/1/pages/802.1bh.html>
- [11] Cisco Systems, Inc., Cisco VN-Link: Virtualization-Aware Networking, White Paper, 2009. Available: http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894/white_paper_c11-525307.pdf
- [12] DMTF, DSP0004, Common Information Model (CIM) Infrastructure, version 2.6.0, 2010.
- [13] VMware CIM APIs, <http://www.vmware.com/support/developer/cim-sdk/>
- [14] The Xen-CIM Project, <http://wiki.xensource.com/xenwiki/XenCim>
- [15] W. Fuertes, F. Meneses, J. E. López de Vergara, et al., "A Generic Model for the Management of Virtual Network Environments", In Proc. of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010), pp. 813-816, 2010.
- [16] DMTF, DSP1097, Virtual Ethernet Switch Profile, version 1.0.0, 2010.
- [17] DMTF, DSP1050, Ethernet Port Resource Virtualization Profile, version 1.0.0, 2010.
- [18] H. Okita and M. Yoshizawa, "Virtual Network Management Information Model," Internet Draft, 2010. Available: <http://tools.ietf.org/html/draft-okita-ops-vnetmodel-02>