Virtual Network Management Information Model
draft-okita-ops-vnetmodel-02

Abstract

Virtual switches on server virtualization platforms cause a problem in managing data center networks containing several hundred switches. Accordingly, a management information model for the network structure of data center networks containing virtual switches is proposed. The proposed model consists of a physical layer (which represents connections between physical switches) and a virtual layer (which represents connections between virtual switches). These layers also represent the association of the virtual switch with the corresponding physical switch. The model shortens the virtual LAN (VLAN) configuration time taken by operators of data center networks by a maximum of 35%. This result shows that the proposed model is effective in reducing the management time of data center networks containing virtual switches.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 9, 2010.
1. Introduction .................................................. 3
2. Requirements ................................................. 4
   2.1. Problem Statement ........................................ 4
   2.2. System Architecture Requirements ....................... 6
   2.3. Requirements for Virtual Network Management
       Information Model ........................................ 8
3. Relationships to Existing MIBs ............................... 9
   3.1. Relationships to LLDP-MIB ............................... 9
   3.2. Relationships to ENTITY-MIB ............................ 9
4. Proposals of Virtual Network Management Information Model .. 11
   4.1. TargetedNetwork Object ............................... 11
   4.2. PhysicalNetwork Object ............................... 12
   4.3. VirtualNetwork Object ............................... 14
   4.4. Id Object ............................................. 16
5. Experiment .................................................. 18
   5.1. Environment ............................................. 18
   5.2. Results ................................................ 20
6. Summary .................................................... 22
7. Security Considerations ..................................... 23
8. IANA Considerations ......................................... 24
9. References .................................................. 25
   9.1. Normative References .................................. 25
   9.2. Informative References ................................ 25
Authors’ Addresses ............................................. 26
1. Introduction

In data center networks, a virtual switch on a server virtualization platform works as a virtual network element [VEB] [EVB-PAR] [PE-PAR]. The virtual switch connects multiple virtual machines on the same server virtualization platform and connects these virtual machines to external physical switches.

Virtual switches, however, cause a problem in managing data center networks because, mainly, a virtual switch and a physical switch require different management systems. Operators of data center networks therefore have to use multiple management systems for managing the whole data center network.

To avoid this management difficulty, an integrated network management system (NMS) is effective. The integrated NMS collects and stores virtual-network management information that describes network structure of a managed target network. It then displays or transmits this management information as a response to a request from operators or other NMSs.

The purpose of this document is to provide a management information model that represents the network structure of a data center containing virtual switches. Section 2 describes the model requirements, Section 3 describes the relationships to the existing MIBs, Section 4 defines the model, and Section 5 evaluates the model.
2. Requirements

2.1. Problem Statement

Virtual switches cause a difficulty in managing data center networks. They expand the data center network into the server virtualization platforms. Therefore, to manage the whole network structure of data center networks, network operators have to manage virtual switches in addition to physical switches.

To manage these virtual and physical switches, operators have to use multiple management interfaces. Specifically, to manage virtual switches, they have to use a specific management system for the server virtualization platform that the target virtual switches are created on. Moreover, to manage physical switches, they use a network management system. Figure 1 shows an architectural overview of a conventional data center network management system.
This conventional management architecture causes the following two problems which increase the operation time taken by operators of the data center networks and thus increase operational costs.

1. When operators want to examine the network structure of a virtual network containing virtual switches, they have to access multiple management systems.

2. When operators want to examine the mapping of a virtual network to corresponding physical components, they have to access multiple management systems.

To solve these problems and save the operation time for data center networks, the following two requirements must be met.
1. The data center network should provide an integrated management system that enables operators to get network structure information about virtual network.

2. The data center network should provide an integrated management system that enables operators to get mapping information about virtual switches and their underlying physical platforms.

2.2. System Architecture Requirements

A system architecture that effectively satisfies the above-described requirements is proposed in the following.

An integrated network management system (NMS) effectively reduces the network operation time needed for managing virtual switches and physical switches. It is referred to as a VNMS (Virtual Network Management System.) It integrates multiple existing management interfaces into a single interface. Operators can thus reduce their operation time.

The VNMS manages device connectivity in the managed target network. To perform this task, it stores network management information about configured virtual networks in the target network.

Figure 2 shows an overview of the system architecture of the target system. The virtual-network management information about the VNMS is based on the proposed model.
Figure 2: Overview of system architecture

The following three types of elements exist around this VNMS.

- User clients or traditional NMS
- Network switches
- Server virtualization platforms

The user client or network application uses management information about device connections in the managed network. The network switches are virtualized as multiple virtual switches. Moreover, the server virtualization platforms are virtualized as multiple virtual machines and internal virtual switches. A set of virtual switches
and virtual machines forms a virtual system for a user.

Among the elements described above, we define the following two management interfaces.

- Network Management Interface (NMI)
- Device Management Interface (DMI)

The network management interface (NMI) is set between the network application and the VNMS. This interface is used by the VNMS to transport virtual-network management information to network applications in response to their request.

Datamodels provide the definition and format of the virtual-network management information transported on the NMI. The definition describes an encoding scheme and an underlying transport protocol. The VNMS may use, for example, SNMP (Simple Network Management Protocol) and MIB (Management Information Base) specified in the Internet-standard management framework [RFC3410] or an XML-based management framework [RFC3535] as the datamodel.

The device-management interface (DMI) is set between the VNMS and network devices, which include the server virtualization platforms and network switches. The DMI is used by the VNMS to query management information about a target device. This interface is device specific and not standardized by this document.

### 2.3. Requirements for Virtual Network Management Information Model

This document focuses on an information model for the virtual-network management information described in the previous section. The requirements for the information model are listed below. These requirements arise from the two problems stated above.

1. **Connection Information:** The proposed model should represent a connection between virtual switches, a connection between physical switches, and a connection between a virtual switch and a physical switch in the target network.

2. **Virtual-Physical Mapping Information:** The proposed model should represent mapping of a virtual switch to the physical server that the virtual switch is created on.
3. Relationships to Existing MIBs

A lot of RFCs about MIBs have been published from the IETF. These existing MIBs provide each information models implicitly. For avoiding inventing the wheel, we researched relationships between the requirements for the virtual network management information model and existing MIBs.

3.1. Relationships to LLDP-MIB

Protocols for network topology discovery like Link Layer Discovery Protocol (LLDP) use some of MIB modules. These MIB modules are used to describe link state information in the managed network. For example, the LLDP-MIB [IEEE.802-1AB.2005] standardized as IEEE Standard 802.1AB supports this function.

The LLDP-MIB can be used to describe a connection between neighboring layer-2 MAC bridges. In the LLDP-MIB, there is an lldpRemTable which contains one or more rows per physical network connection. The row contains a chassis ID, a port ID, a port description, and system information for each neighboring layer-2 MAC bridge.

As described above, the LLDP-MIB can be used to describe the connection information between physical entities like physical switches. However, the LLDP-MIB cannot be used to describe the connection information between logical entities. Thus, it cannot be used to describe the connection information between a virtual switch and a virtual machine on the same physical server. Moreover, it cannot be used to describe the connection information between a virtual switch and an external physical switch.

As the result, the LLDP-MIB does not satisfy the first requirement in section 2.3 for the virtual network management information model.

3.2. Relationships to ENTITY-MIB

The ENTITY-MIB [RFC2737] was published by the IETF entmib WG. It can be used to represent a single SNMP agent which supports multiple instances of one MIB. For example, a single physical switch having a single SNMP agent can support multiple instances of a bridge with the ENTITY-MIB.

The ENTITY-MIB can be used to describe following two types of information.

One is mapping information between logical entities and physical entities on one network element. The information can be represented by the entLPMappingTable and the entAliasMappingTable in the
entityMapping group. For example, these tables support logical entities which contain OSPF instances and 802.1d bridges. Moreover, these tables support physical entities which contain bridge ports, backplanes and chassis.

Another is information about hierarchy relationship among physical entities. The information can be represented by the entPhysicalContainsTable in the entityMapping group. The entPhysicalContainsTable contains simple mapping information between ‘container’ entity and ‘containee’ entity. For example, a chassis is a ‘container’ entity. Its bridge ports and its backplane are ‘containee’ entities.

As described above, the ENTITY-MIB can be used to describe the mapping information between logical entities and physical entities. Therefore, the ENTITY-MIB satisfies the second requirement in section 2.3 for the virtual network management information model.

However, the ENTITY-MIB cannot be used to describe the connection information between logical entities. For example, it is impossible to describe connection information between virtual switches with the ENTITY-MIB.

As the result, the ENTITY-MIB does not satisfy the first requirement in section 2.3 for the virtual network management information model.
4. Proposals of Virtual Network Management Information Model

This section defines the proposed virtual-network management information model, which is an object-oriented information model. The model can satisfy both of the requirements included in section 2.3. The model is an abstract-information model independent from encoding schemes and management protocols. The model is written in Unified Modeling Language (UML) [UML] .

4.1. TargetedNetwork Object

The proposed model starts with a TargetedNetwork object. This object represents the overall network. In the network, two types of network exist: a physical network and a virtual network. In the proposed model, a PhysicalNetwork object represents a physical network, and a VirtualNetwork object represents a virtual network. To represent this structure, the TargetedNetwork object has one or multiple references to PhysicalNetwork objects and VirtualNetwork objects.

Furthermore, the PhysicalNetwork object and the VirtualNetwork have a reference between them. Since a physical network can create multiple virtual networks, the PhysicalNetwork object can have multiple references to corresponding VirtualNetwork objects. On the contrary, the VirtualNetwork object has only one reference to the PhysicalNetwork object, since the virtual network is created on the specific physical network.

Figure 3 shows a class diagram of the proposed virtual-network management information model containing the TargetedNetwork object, PhysicalNetwork objects, and VirtualNetwork objects.

```
+-------------------+
|TargetedNetwork|   +-------------------+
+-------------------+
<>                  <>
1                   1
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1-------------------</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
</tbody>
</table>
1                   0..n
<>                  <>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
</tbody>
</table>
0..*                0..n
+-------------------+ (Figure.5)
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
</tbody>
</table>
1                   1
<>                  <>
+-------------------+ (Figure.4)
```

Okita & Yoshizawa       Expires September 9, 2010              [Page 11]
4.2. PhysicalNetwork Object

To represent the structure of a physical network, the proposed model defines the following six types of managed objects under the TargetedNetwork object.

- PhysicalNetwork
- PhysicalNode
- PhysicalNodeGroup
- PhysicalInterface
- PhysicalInterfaceGroup
- PhysicalLink

PhysicalNetwork:
This object represents an actual network composed of actual devices. This object aggregates zero or more PhysicalNode objects.

PhysicalNode:
This object represents an actual device in a physical network. The actual device is a server, a server virtualization platform, or a network switch. The object has an association with a PhysicalNetwork object. It also has an association with a PhysicalNodeGroup object when the actual device is a member of a group of devices. It also aggregates zero or more PhysicalInterface objects. The PhysicalNode object can contain one "Configurations" object, which stores configuration data of the device represented by the PhysicalNode object. The Configurations object contains, for example, virtual LAN (VLAN) configuration, link aggregation (LAG) configuration or server virtualization configuration. Although this memo defines the Configurations object as a child object of the PhysicalNode object, defining the model for the configuration information is out of scope of this memo. The main reason is that the model of the Configurations object differs from one device to another.
PhysicalNodeGroup:
This object represents a set of multiple actual devices. For example, this object represents the chassis of a blade server, which includes multiple server blades and multiple network switches. This object aggregates one or more PhysicalNode objects.

PhysicalInterface:
This object represents an actual network interface of an actual device. The network interface is a port of a network interface card equipped in a server or a port of a network switch. The object also represents an internal network interface used to connect a server blade and an internal switch in a blade server. This object has an association with a PhysicalNode object. This object also has an association with a PhysicalInterfaceGroup object when the network interface is a port of the line card represented by the PhysicalInterfaceGroup object. This object also has an association with a PhysicalLink object when the network interface is connected to another network interface by an actual network cable.

PhysicalInterfaceGroup:
This object represents a set of actual network interfaces. For example, it represents a network interface card or a network switch’s line card (which is equipped with multiple ports). It aggregates one or more PhysicalInterface objects.

PhysicalLink:
This object represents an actual network cable used to connect two actual network interfaces. For example, it represents a generic Ethernet cable. It also represents an internal connection between a server blade and an internal switch in a blade server. This object aggregates two PhysicalInterface objects.

Figure 4 shows an abstract class diagram of the objects related to the physical network.
To represent the structure of a virtual network, the proposed model defines the following five types of managed objects under the TargetedNetwork object.

- VirtualNetwork
- VirtualNode
- VirtualNodeGroup
- VirtualInterface

Figure 4: Class diagram of physical-network-related objects

4.3. VirtualNetwork Object

To represent the structure of a virtual network, the proposed model defines the following five types of managed objects under the TargetedNetwork object.

- VirtualNetwork
- VirtualNode
- VirtualNodeGroup
- VirtualInterface
o VirtualLink

VirtualNetwork:
This object represents a virtual network composed of multiple virtual network devices, including not only actual devices but also virtual devices. It aggregates zero or more VirtualNode objects.

VirtualNode:
This object represents a virtual network device in a virtual network. Examples of the virtual devices are virtual switches and virtual machines on a server virtualization platform. Other examples are virtual-router functions configured on a router. The object has an association with a VirtualNetwork object and a VirtualNodeGroup object.

VirtualNodeGroup:
This object represents a set of virtual devices that are created from the same actual device. It aggregates one or more VirtualNode objects. It also has an association with a PhysicalNode object, which represents an actual device.

VirtualInterface:
This object represents a virtual network interface of a virtual device. An example of such an interface is a virtual network-interface card (VNIC) of a virtual machine on a server virtualization platform. This object has an association with a VirtualNode object. This object also has an association with a VirtualLink object when the virtual network interface is connected to another virtual network interface by a virtual network link.

VirtualLink:
This object represents a virtual network link used to connect two virtual network interfaces. For example, it represents a connection between a virtual machine and a virtual switch created on a server virtualization platform. This object aggregates two VirtualInterface objects.

The relationship between the VirtualNetwork, the VirtualNode, the VirtualInterface, and this VirtualLink object is almost the same as the relationship between the PhysicalNetwork, the PhysicalNode, the PhysicalInterface, and the PhysicalLink object.

Figure 5 shows an abstract class diagram of the objects related to the virtual network.
Figure 5: Class diagram of virtual-network-related objects

4.4. Id Object

All objects except the TargetedNetwork object must contain each "id" object which stores an identifier (ID). The ID must be unique within the group formed by the same type of objects associated with the same parent object as following.

- PhysicalNetwork object ID is unique within a TargetedNetwork object.
- PhysicalNodeGroup object ID is unique within a PhysicalNetwork object.
- PhysicalNode object ID is unique within a PhysicalNetwork object.
o PhysicalInterface object ID is unique within a PhysicalNode object.

o PhysicalInterfaceGroup object ID is unique within a PhysicalNode object.

o PhysicalLink object ID is unique within a PhysicalNetwork object.

o VirtualNetwork object ID is unique within a TargetedNetwork object.

o VirtualNode object ID is unique within a VirtualNetwork object.

o VirtualInterface object ID is unique within a VirtualNode object.

o VirtualLink object ID is unique within a VirtualNetwork object.
5. Experiment

5.1. Environment

To confirm the applicability of the proposed model, a virtual network management system, deployed with virtual network information based on the proposed information model, was developed. This system stores configuration information of the managed network and displays the contents of this information via a command line interface. Operators can confirm the network structure with this system.

In the experiment using the virtual network management system, operators made a virtual LAN (VLAN) configuration of switches in sample networks with and without the integrated NMS. The operation time for each VLAN configuration operation was evaluated.

Two sample networks containing multiple virtual machines were used. One of the networks corresponds to a small private network. The other corresponds to a medium-sized enterprise network. Table 1 lists the parameters of these sample networks.

<table>
<thead>
<tr>
<th></th>
<th>Example1</th>
<th>Example2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server blades</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>Virtual machines</td>
<td>8</td>
<td>26</td>
</tr>
<tr>
<td>Switches</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Parameters used in the experiment

The sample network in example 1 consisted of an external physical switch, two internal physical switches, and four server blades in two blade-server chassis. Each server blade ran a server virtualization platform. The internal switches connected the two blades in the chassis, and the external switch connected these two internal switches. Each virtualization platform operated two virtual machines. On these server virtualization platforms, virtual switches connected an external physical switch and multiple virtual machines on the same platform.

The sample network in example 2 consisted of four external switches, four internal switches, fourteen server blades in a four-blade server chassis. Each server blade ran a server virtualization platform, each of which operated one or two virtual machines. And four physical switches worked as an external gateway switch, a core
switch, and two edge switches connected to a server virtualization platform.

Figure 6 shows the virtual-network management information of the network used in example 1. Here, "VM" and "VSW" stand for a virtual machine and a virtual switch created on a server virtualization platform. PhysicalLink objects and VirtualLink objects are omitted to help readability.

```
TargetedNetwork0
+PhysicalNetwork0
 | +ServerBlade0
 |   | +NIC0
 |   | +NIC1
 | +ServerBlade1
 |   | +NIC0
 |   | +NIC1
 | +ServerBlade2
 |   | +NIC0
 |   | +NIC1
 | +ServerBlade3
 |   | +NIC0
 |   | +NIC1
 | +InternalSwitch0
 |   | +IF0
 |   | +IF1
 |   | +IF2
 | +InternalSwitch1
 |   | +IF0
 |   | +IF1
 |   | +IF2
 | +Switch2
 |   | +IF0
 |   | +IF1
 |   | +IF2
 | +IF3
+VirtualNetwork0
 | +VM0.0
 |   | +VNIC0
 | +VM0.1
 |   | +VNIC0
 | +VM1.0
 |   | +VNIC0
 | +VM1.1
 |   | +VNIC0
 | +VM2.0
```
5.2. Results

The following tables show the results of the experiment. Two users partook in the experiments. User 1 had 20 years of experience as a system engineer. User 2 had 10 years of experience as a researcher. Both are familiar with server technologies but not familiar with network technologies.
<table>
<thead>
<tr>
<th></th>
<th>User1</th>
<th>User2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without an integrated NMS</td>
<td>783sec</td>
<td>1056sec</td>
</tr>
<tr>
<td>With a proposed integrated NMS</td>
<td>451sec</td>
<td>676sec</td>
</tr>
</tbody>
</table>

Table 2: VLAN configuration operation time of example 1

<table>
<thead>
<tr>
<th></th>
<th>User1</th>
<th>User2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without an integrated NMS</td>
<td>1278sec</td>
<td>874sec</td>
</tr>
<tr>
<td>With a proposed integrated NMS</td>
<td>1270sec</td>
<td>1279sec</td>
</tr>
</tbody>
</table>

Table 3: VLAN configuration operation time of example 2

Without the virtual-network management system, user 2 took 1,056 seconds to make the VLAN configuration by adding a virtual machines VM3.0 and VM3.1 to a specified VLAN in example 1.

In contrast, using the virtual-network management system, user 2 reduced VLAN configuration time by 35%. It took 676 seconds for the operator to make the VLAN configuration by adding a virtual machine to the specified VLAN in example 1.

While performing the VLAN-configuration operation without the virtual-network management system, the operators had to access each management interface of the physical switch and server virtualization platforms or check the configuration sheet of the network.

In contrast, while performing the operation with the virtual-network management system, the operators could read all the required management information from the management interface of the system. That is the main reason that the operation time was reduced.
6. Summary

This document proposes a management information model for a virtual network in a data center network. This information model can represent the network structure of a virtual network composed of virtual switches and physical switches. It can also represent the mapping between the virtual switch and the physical switch.

The network management system, which manages virtual-network management information according to the proposed information model, reduced VLAN configuration time by 35%. This result demonstrates that the virtual-network management information model is effective in reducing the management time of a data center network containing virtual switches.

The proposed management information model does not contain implementation specifications. Therefore, to implement the information model, developers have to select an encoding scheme and a management protocol for transporting management information data. For example, developers can use SNMP and MIB specified in the Internet-standard management framework [RFC3410] or an XML [W3C.REC-xml] -based management framework [RFC3535]
7. Security Considerations

The virtual-network management information as defined in this document provides administrative information about a data center network. This information could be used to aid an attack on the network.

It is assumed that accesses to the data defined in this document are subject to appropriate access control in the network management system.
8. IANA Considerations

The document does not request any IANA action, since the proposed model is an abstract information model. However, a concrete data model based on this information model should request IANA actions if necessary.
9. References

9.1. Normative References

[IEEE.802-1AB.2005]  
"Local Area Networks and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery", IEEE Standard 802.1AB, May 2005.


9.2. Informative References


Authors’ Addresses

Hideki Okita
Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo
Kokubunji, Tokyo  185-8601
Japan

Phone: +81-42-323-1111
Fax:   +81-42-327-7756
Email: hideki.okita.pf@hitachi.com

Masahiro Yoshizawa
Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo
Kokubunji, Tokyo  185-8601
Japan

Phone: +81-42-323-1111
Fax:   +81-42-327-7756
Email: masahiro.yoshizawa.bt@hitachi.com