Integrated Network Operations and Management across Virtualized Servers and Networks of Large-Scale Data Center

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2022

Hideki OKITA

List of Publications

1. Journal Papers

1. Masahiro Yoshizawa, Hideki Okita, Keitaro Uehara, and Toshiaki Tarui, "Implementation and Evaluation of Network Management System Supporting Documentation for Virtual Networks," IPSJ Journal, vol.52, no.3, pp.1334-1347, March 2011 (in Japanese).

2. Hideki Okita, Masahiro Yoshizawa, Keitaro Uehara, Kazuhiko Mizuno, Toshiaki Tarui, and Ken Naono, "Virtual Network Configuration Management System for Data Center Operations and Management," IEICE Transactions on Communications, vol.E95-B, no.6, pp.1924-1933, July 2012.

3. Hideki Okita, Hayato Hoshihara, Norihisa Komoda, and Toru Fujiwara, "Dynamically Prioritized Failure Management According to Reliability Model in Large-Scale Data Center," IADIS International Journal on WWW/Internet, vol.18, no.2, pp.101-115, Dec. 2020.

2. International Conference Papers

1. Masahiro Yoshizawa, Toshiaki Tarui, and Hideki Okita, "Implementation and Evaluation of Network Management System to Reduce Management Cost Caused by Server Virtualization," in Proceedings of 2nd Workshop on Data Center – Converged and Virtual Ethernet Switching (DC CAVES) (in CD-ROM), Sept. 2010.

2. Hideki Okita, Masahiro Yoshizawa, Keitaro Uehara, Kazuhiko Mizuno, Toshiaki Tarui, and Ken Naono, "Proposal of Virtual Network Configuration Acquisition Function for Data Center Operations and Management System," in Proceedings of Euro-Par 2010 Parallel Workshops, LNCS, vol.6586, pp.625-632, Aug. 2010.

3. Yuncheng Zhu, Hideki Okita, and Seishi Hanaoka, "A Practical Approach for Network Fault Detection," in Proceedings of 2016 International Conference on Computing, Networking and Communications (ICNC 2016), pp.1-5, Feb. 2016.

4. Hideki Okita, Keitaro Uehara, and Yoshiko Yasuda, "Proposal and Evaluation of Data-Model Translation for Server-Virtualization Environment to Improve Virtual-

Network Configuration Management," in Proceedings of 5th IIAI International Congress on Advanced Applied Informatics (IIAI 2016), pp.878-883, July 2016.

5. Yuncheng Zhu, Hideki Okita, and Seishi Hanaoka, "Quality degradation localization in complex message processing networks," in Proceedings of 2017 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR 2017), pp.1-6, May 2017.

6. Hideki Okita, Hayato Hoshihara, Norihisa Komoda, and Toru Fujiwara, "Dynamically Prioritized Virtual-Network Monitoring according to Lifecycle of Virtual Machines in Large Scale Data Center," in Proceedings of 19th International Conference on WWW/Internet 2020 (ICWI 2020), pp.123-131, Nov. 2020.

3. Domestic Conference Papers

1. Masahiro Yoshizawa, Toshiaki Tarui, and Hideki Okita, "Implementation and Evaluation of Network Management System to Reduce Management Cost Caused by Server Virtualization," IEICE Technical Report, vol.109, no.273, NS-2009-116, pp.71-76, Nov. 2009 (in Japanese).

2. Hideki Okita, "PBB-VPN Operations and Management System," IEICE Technical Report, vol.109, no.463, ICM-2009-51, pp.35-40, March 2010 (in Japanese).

3. Hideki Okita, Masahiro Yoshizawa, Keitaro Uehara, Kazuhiko Mizuno, Toshiaki Tarui, and Ken Naono, "Proposal of Virtual Network Configuration Acquisition Function for Data Center Operations and Management System," IEICE Technical Report, vol.110, no.119, ICM-2010-20, pp.67-72, July 2010 (in Japanese).

4. Hideki Okita, Masahiro Yoshizawa, Keitaro Uehara, Yoshiko Yasuda, and Mariko Yamada, "Proposal and Evaluation of Data-Model Translation for Server-Virtualization Environment to Improve Virtual-Network Configuration Management," IEICE Technical Report, vol.110, no.466, ICM-2010-61, pp.41-46, March 2011 (in Japanese).

5. Hideki Okita, Chie Hasegawa, Hayato Hoshihara, Masashi Yano, and Seishi Hanaoka, "Large-Scale EPC Control-Plane Simulator -(1) Modeling and Evaluation-," in Proceedings of 2014 IEICE Society Conference, B-14-12, p.347, Sept. 2014 (in Japanese).

6. Chie Hasegawa, Hideki Okita, Hayato Hoshihara, Seishi Hanaoka, and Seiya Kudoh, "Development of Large-scale EPC Control-Plane Simulator -(2) A Study on Signaling Load Reduction-," in Proceedings of 2014 IEICE Society Conference, B-14-13, p.348, Sept. 2014 (in Japanese).

7. Hayato Hoshihara, Hideki Okita, Seishi Hanaoka, and Ryoichi Tanaka, "Predictive Congestion Detection of Attach Signaling in LTE/EPC Networks," IEICE Technical Report, vol.114, no.389, ICM-2014-32, pp.1-6, Jan. 2015 (in Japanese).

8. Yuncheng Zhu, Hideki Okita, and Seishi Hanaoka, "Locating the Cause of QoS Performance Decline in EPC," in Proceedings of 2015 IEICE General Conference, BS-3-33, pp.S-70-S-71, March 2015.

9. Daisuke Ito and Hideki Okita, "A Study on Utilization of Informational Universal Middleware for Network Analytics," in Proceedings of 2015 IEICE Society Conference, B-7-11, p.82, Sept. 2015 (in Japanese).

10. Hideki Okita, Hayato Hoshihara, Norihisa Komoda, and Toru Fujiwara, "Dynamically Prioritized Virtual-Network Monitoring according to Uptime of Virtual Machines," IEICE Technical Report, vol.120, no.259, ICM2020-31, pp.61-66, Nov. 2020 (in Japanese).

4. Others

1. Hideki Okita, Masahiro Yoshizawa, Junji Yamamoto and Daisuke Matsubara, "Research and Development of Network Solution Technology for Next Generation Networks," Hitachi Review, vol.88, no.6, pp.494-497, June 2006 (in Japanese).

2. Daisuke Matsubara, Hideki Okita, Yasushi Kanada and Hitoshi Yoshida, "Research and Development of Traffic Control Technology for Next Generation Networks," Hitachi Review, vol.89, no.6, pp.472-475, June 2007 (in Japanese).

Abstract

This dissertation describes the results of research on network operations and management for large-scale data centers. The research was conducted at the Research and Development Group, Hitachi, Ltd. from April 2008 to October 2021 and at the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University from April 2016 to March 2019.

IT systems that were previously located in the server room on premise of the company are now installed in data centers of service providers. The operations and management of computing and networking resources are outsourced to data center operators. Network and server administrators cooperate to operate the data centers: server administrators manage the computing resources of the servers, while network administrators manage the connectivity between the servers and external networks.

In data centers to which server virtualization and network virtualization have been applied to make multi-tenant environments on a physical environment, the location of the boundary between the server management and the network management has changed from the case of traditional data center management. Specifically, when multiple virtual machines are running on a single physical server with server virtualization, the connectivity among virtual machines and external networks is configured in accordance with the user requirements. It is thus necessary for server administrators and network administrators to design and operate virtual networks across both physical servers and network devices.

However, it is often difficult for network administrators to operate and manage the server virtualization mechanism of physical servers, as this was not something that needed to be done in the conventional operations and management. Therefore, a mechanism to support the operations and management of virtual networks across physical servers and network devices in an integrated manner is required.

To achieve the operations and management of virtual networks, a system that supports both configuration management and failure management is proposed in this disser-

tation. These two types of management play a critical role in maintaining the service level of data centers. In this dissertation, the following issues are raised as related to configuration management and failure management, respectively.

(1) Integrated configuration management of virtual networks

(2) Dynamic monitoring of data-center networks

To achieve (1) integrated configuration management of virtual networks in data centers, the focus is on the point that existing configuration management methods for virtual networks in data centers are independently studied on the server side and the network side. A centralized management system for virtual networks across servers and network devices is proposed. In addition, a data model of the configurations of virtual networks in data centers for the method is proposed. The time needed for network administrators to verify the configuration of virtual networks in a data center using the proposed method was evaluated with a prototype system for the network operations and management of virtual networks. The results showed that the proposed method shortens the time to verify the configurations of virtual networks across servers and networks compared to the conventional method of having server administrators and network administrators independently manage the configurations of virtual networks.

To achieve (2) dynamic monitoring of data-center networks, the focus is on the frequent replacement of the physical server or virtual machine (VM) in data centers where server virtualization has been introduced. A dynamic monitoring method for data center networks is proposed using Ethernet Operations, Administration, and Maintenance (OAM), which is a standard network monitoring protocol. The key feature of the method is the dynamic allocation of network monitoring resources to dynamically selected ports of network devices, rather than static allocation of the resources. In detail, dynamic monitoring based on (2-1) the probabilistic reliability model of VMs and (2-2) the system configuration information is considered.

For the dynamic monitoring in (2-1), the focus is on the time variation of the failure rate of a physical server or VM in its lifecycle. A dynamic monitoring method for data-center networks using Ethernet OAM based on a probabilistic reliability model of VMs is proposed. Using the reliability model as a basis, the network management system selects VMs with a high failure rate and sets the ports to which the selected VMs are connected as the target of continuous monitoring with Ethernet OAM. This increases the probability to immediately determine if the root cause exists on the network side. The improvement in the time to localize the root causes of failures was evaluated by simulation. The results

showed that the proposed method shortens the time to localize the root cause of the service failure compared to the conventional method in which network monitoring resources are randomly allocated to ports. This demonstrates that coordinating the server management and network management enables efficient failure management in large-scale data centers.

Also, for the dynamic monitoring in (2-2), the focus is on the initial failure and later failure due to wear-out of components that can occur on all physical servers even if those operational states and failure characteristics are different. A dynamic monitoring method for data-center networks using Ethernet OAM based on the server configuration information, specifically, the uptime of servers, is proposed. The method does not require probabilistic VM reliability models, which require statistical management information to create. Rather, the network management system selects VMs with a short uptime as monitoring targets for initial failure in accordance with the configuration information. Similarly, it selects VMs with a long uptime as monitoring targets for the failure due to wear-out of components. The improvement in the time to localize the root causes of failures was evaluated by simulation. The results showed that the proposed method reduces the time to localize root causes of failures even without the probabilistic VM reliability models compared to the conventional method in which network monitoring resources are randomly allocated to ports. However, the effect of shortening the time to localize the root causes of failures is 5.5% lower than that of the conventional method. This method can be widely applied to data centers that do not usually collect the failure statistics of VMs. It thus expands the applicability of the dynamic monitoring of data-center networks.

The research results in this dissertation contribute to making the operations and management of virtual networks in large-scale data centers where virtualization technologies are deployed more efficient and to maintaining the service levels of the services provided from the data centers.

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Traditional Operations and Management of Data Center Network

The market for data center services, which operate and monitor the information systems of client companies within the data centers of information service providers, is rapidly expanding and is expected to maintain a high growth rate in the future. The IT infrastructure of companies is increasingly installed in data centers and operated remotely. In addition, in order to provide services via the Internet, companies need to install IT systems in the data centers. Therefore, data center service providers are expanding not only colocation services for IT systems but also server hosting services that they procure, operate, and manage on behalf of their customers. This means that they now need to operate and manage a wide variety of IT systems co-existing in a data center.

Networks play a significant role in providing services via data centers[1, 2]. These networks connect many devices both inside and outside the data center to enable the handling of a large number of requests for services running on the IT systems[3]. This leads to a complicated structure of data-center networks. Therefore, efficient operation and management of data-center networks is crucial for maintaining the service levels and routine operations.

Prior studies have analyzed the characteristics and causes of IT system failures in data centers including a high-performance computing site[4], the research center of an IT company[5], data centers of commercial Internet service providers[4], and data centers of cloud computing service providers[6, 7, 8, 9, 10, 11, 12, 13]. Networking problems related

to hardware or software failures are a significant cause of service failures and accounted for 76% of the failure factors according to a failure analysis in a large-scale Internet service provider[14]. Also, the investigation results of a research institution [15, 16] and the survey results in a commercial data center[17] have shown that failures due to network problems are not insignificant.

To maintain the service levels of open systems in the telecommunication field, ITU-T specified standard management functions for open systems that interwork using communication media in the 1990s, which are often called "FCAPS"[18, 19]. The management functions are categorized into the following five management function areas that were originally specified in the standard management framework for open systems interconnection[20] in the 1980s.

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management

Although the FCAPS framework was developed before cloud computing was widely implemented, it is still valid for the operation and management of networks in large-scale data centers because a data-center network is still a large open system composed of many network devices[21, 22, 23]. Among the five categories, configuration management and fault management (failure management) are essential to maintain the service levels of data centers. Configuration management includes functions to a) set the parameters that control the routine operation of the open system, b) associate a name with managed objects and sets of managed objects, c) initialize and close down managed objects, d) collect information on demand about the current condition of the open system, e) obtain announcements of significant changes in the condition of the open system, and f) change the configuration of the open system[20]. In other words, the configuration management function provides underlying management information for all the other management function areas.

As for fault management, it includes multiple functions to a) maintain and examine error logs, b) accept and act upon error detection notifications, c) trace and identify faults, d) carry out sequences of diagnostic tests, and e) correct faults[20]. In other

words, the fault management function directly affects the service levels. Both passive and active monitoring methods are used to monitor failures in networks. The passive monitoring methods analyze statistical information[24, 25, 26, 27, 28, 29, 30] or operational information[31, 32, 33, 34, 35, 36] to detect anomalies and analyze the root causes of failures. They tend to take up a lot of the resources of the network monitoring systems. In contrast, active monitoring methods are used by network monitoring systems or devices to actively transmit packets or frames dedicated to monitoring the connectivity in networks. Communication protocols are standardized and widely used to remotely monitor server availability and network connectivity in data centers. The Internet Control Message Protocol (ICMP)[37] is the standard protocol for network management systems to monitor servers and network devices in TCP/IP[38] networks by exchanging monitoring packets, well known as "ping"[37]. Ethernet Operations, Administration, and Maintenance (Ethernet OAM) is another active monitoring method that monitors connectivity in Ethernet-based wide-area networks (WANs)[39]. IEEE802.1ag[40] and ITU-T Y.1731[41] are defined as the standard of Ethernet OAM and are widely implemented in network devices. Ethernet OAM is suitable for monitoring failures of data-center networks, as network devices directly monitor the failures with frequently transmitted monitoring frames.

In traditional data centers, the networks are generally operated and managed separately from servers. Network administrators are responsible for the operation and management of networks, while server administrators are responsible for the servers. The knowledge required for the operation and management differs between the network and the server. For example, the operation and management processes of networks are typically designed on the basis of the above-mentioned FCAPS framework.

1.1.2 Virtualization of Data Center Networks

Recent trends in data centers have seen the wide deployment of server virtualization for commercial use. Data centers have essentially been virtualized to provide utility computing offerings[42]. Server virtualization technology enables data centers to split computer resources such as CPU, memory, storage, and I/O and allocate them separately to multiple operating systems. As a result, multiple logical servers, called virtual machines (VMs), can run on one physical server. This has led to an increase in the number of monitored devices, especially in the large-scale data centers that serve many large companies.

In addition, network devices in data centers now become equipped with a virtualization function that can create multiple virtual networks that are isolated from each other. A typical virtualization technology for data-center networks is Virtual LAN (VLAN),

which is standardized as IEEE802.1Q[43]. Each virtual network is identified with an identifier called a VLAN ID. Virtual networks with different VLAN IDs are isolated from each other.

Figure 1.1 shows the virtualization of servers and network devices in a data center. Networking in such virtual environments is typically implemented as a software-based layer-2 switch function running in a hypervisor or the management domain of a physical host[44]. Multiple VMs and the virtual switch connecting the VMs on a physical server form an in-server virtual network. The in-server virtual network and the above VLAN-based virtual network form a data-center-wide virtual network. Therefore, in contrast to traditional non-virtualized data centers, the new virtualized data centers have a boundary between the computing function and the networking function that is different between the physical environment and the virtualized environment. In the virtual environment, the boundaries are on servers

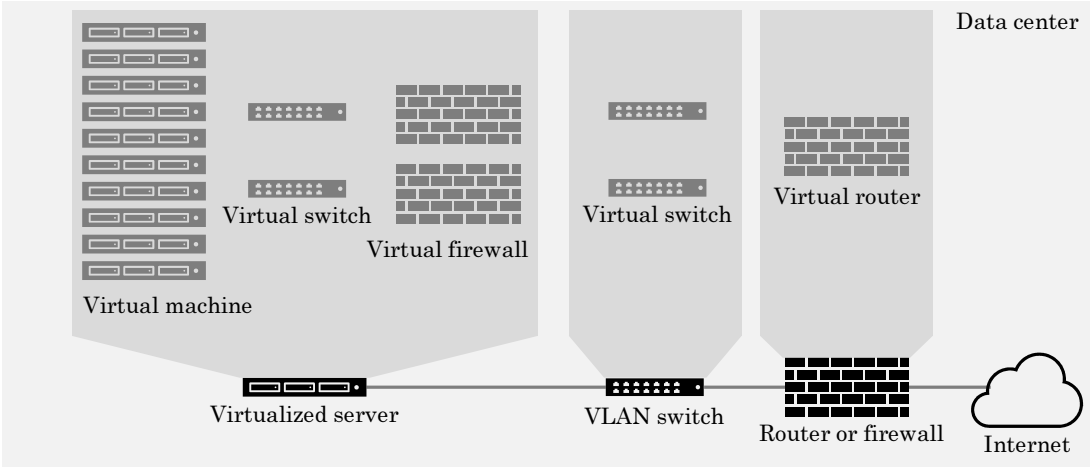


Figure 1.1: Virtualization in a Data Center

The configuration management process has changed due to the adoption of server virtualization. In addition to the standard operations in the data center, there are now administrative requirements unique to the management of virtualized environments[45, 46]. In accordance with the user's request for computing and networking resources, system engineers act as contact points for the users and extract the requirements for virtual servers and virtual networks. Server administrators configure the server virtualization environment of physical servers and run the necessary numbers of VMs on them. Further, network

administrators configure VLAN-supported switches and create the necessary number of virtual networks to connect the VMs.

Since the boundaries between computing functions and networking functions have shifted into physical servers, server administrators in addition to network administrators must now participate in the operation and management of virtual networks in data centers. Server administrators are responsible for managing not only the computing resources but also the networking resources of the physical servers. Such network operation and management includes network-specific configuration tasks, such as specifying the virtual network to which the device connects by specifying a VLAN ID. Server administrators configure virtual switches for connecting VMs on physical servers with the VLAN IDs that are provided by the network administrators.

There are several problems in the configuration management of virtual networks in the data centers that deploy server virtualization and network virtualization. First, most server administrators do not have much experience in network operations and management. If the server administrator is asked to set up such a network-specific configuration, it is likely for some omission or error to occur. Also, when the server administrator and the network administrator exchange configuration information about virtual networks to verify that the configuration meets the user requirements, a human error may result in missing or erroneous configuration information. Because the functions of the server and the network equipment are different, the data models of configuration information are different as well. Moreover, because the standardization body that creates the standard of each data model is unique, the description form of each data model is unique as well.

There is also a problem from the viewpoint of fault management. When service failure occurs in a large data center with server virtualization enabled, it takes a long time to locate the cause of the failure even if Ethernet OAM is activated as a failure detection method. This is because it is a lengthy process to determine whether the cause is on the server side or the network side. Network switches send and receive frequently monitored frames if Ethernet OAM is activated. The load on the network devices becomes high when they are configured to use Ethernet OAM, so only a part of the network end points in the data center can be monitored by Ethernet OAM. When the rest of the network end points connect to the VM related to the service, a network administrator must verify that each network end point is working properly and notify the server administrator of the verification result.

1.1.3 Issues in Network Operations and Management for Large-Scale Data Center

On the basis of the above two problems facing network operation management in data centers to which virtualization has been introduced, the following two requirements are important for the centralized operation and management of servers and networks.

1. It should be possible to centrally manage the configuration of the physical resources and virtual resources across the servers and networks in the data center, and to always synchronize the configuration information of the VMs and networks that the network administrator is managing with the actual configurations of the servers and network devices.

2. When there is a failure of a service provided in a data center, the network port that connects the server of the service should be included in the set of ports monitored by the active monitoring method (e.g., Ethernet OAM) and checked promptly by the network administrator to determine whether it is still available or not.

To meet the requirement for configuration management, it is sufficient to provide a configuration information database that data center administrators can exchange, share and utilize to manage the management information of the servers and networks of the data center in an integrated manner. The configuration information in the database needs to be synchronized with the configurations of the actual server virtualization platforms and network devices running in the data centers. To synchronize the database with the details of the actual situation, it is necessary to collect the configuration information from the server virtualization platforms and network devices at a high frequency and to update the database.

To meet the requirement for failure management, it is necessary to efficiently utilize the limited resources of Ethernet OAM for network monitoring. Even without monitoring all the network end points in a data center, the network end point connecting to the VM that provides the failed service needs to be included in the monitoring targets. It can be a challenge to dynamically select such a network end point in accordance with the service operation situation in the data center.

As described above, misconfiguration and delays in failure correspondence due to virtualization are problems in data centers at which server virtualization and network virtualization are introduced. It is thus crucial to integrate the operation and management of the servers and the networks while considering virtualization. In this dissertation,

we propose methods to solve the following issues for the integrated operations and management as shown in Figure 1.2.

1. Providing configuration information that administrators can exchange, share and utilize to manage the configurations of servers and networks in an integrated manner

2. Selecting the network end points related to services that would fail as the monitoring targets of Ethernet OAM

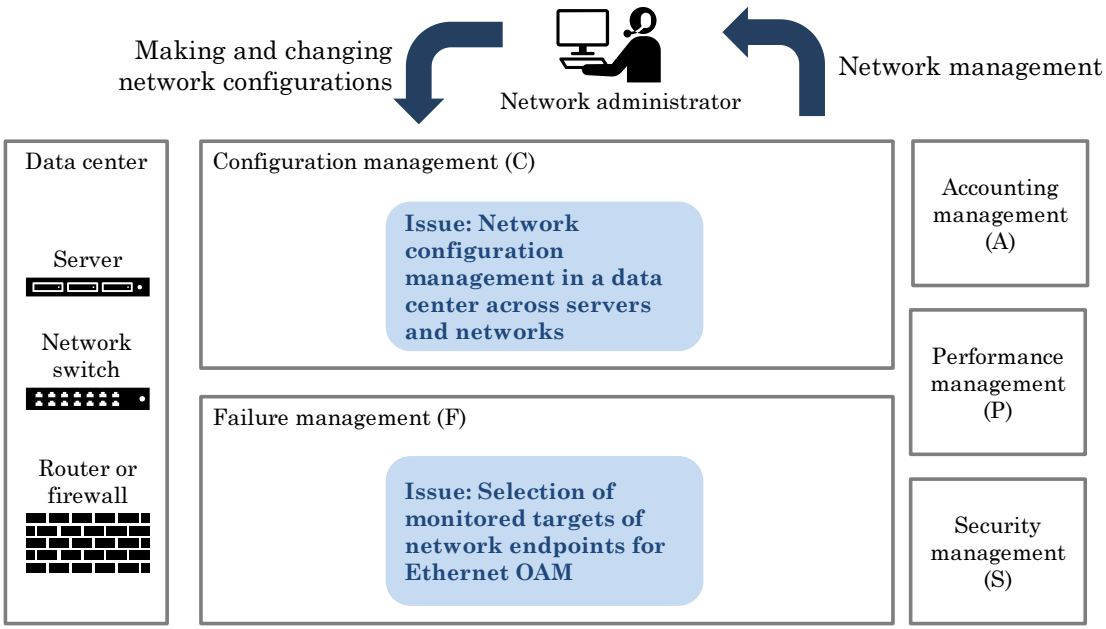


Figure 1.2: Issues in Operations and Management of Data Center

1.2 Related Work

1.2.1 Configuration Management of Virtualized Data Centers

As far as we know, there has been no research examining the issue of providing configuration information that server and network administrators can exchange, share, and utilize in a data center where server virtualization and network virtualization are introduced and where virtual networks are formed across multiple operation management domains. This

section describes prior studies on the network configuration management for virtualized data centers.

There are currently several commercial products available for improving the functionality and efficiency of network operations and management in data centers. On the network side, for example, Cisco Data Center Network Manager (DCNM)[1] supports the operations and management of their own network switch products. On the server side, VMware vCenter Server[2] enables integrated operations of their server virtualization platform. However, since most data centers are made up of not only cloud service environments but also various network devices and servers for server hosting business, along with a combination of server virtualization infrastructures, the network configuration of the entire data center spans several different servers and networks. In addition, as these products store the configurations of virtual networks inside the system as proprietary data, it is difficult for a data center operator to provide the data to an external tool or flexibly utilize the data on the basis of their own needs. For example, a data center operator cannot use an external version control system to store a snapshot of the configuration data of virtual networks to manage historical versions of virtual-network configuration documents. In addition, it is difficult for the operator to utilize an external visualization tool to visualize the configuration of virtual networks from the individual perspectives of operations and management. It is also hard for the operator to utilize external data analysis tools to detect a configuration error or to check the compliance of the network configuration with the network design policy of a data center.

The approach most similar to ours was proposed as a generic-model approach[47, 48, 49]. This approach defines a generic model that makes it possible to deploy and manage a virtual network environment (VNE) regardless of the virtualization platform. This model is an extension of the Distributed Management Task Force (DMTF)'s Common Information Model (CIM) and describes the topology of an IP-based network including VMs.

One study on network virtualization proposed a schema to describe the virtual network specifications[50]. These specifications include a list of network components (routers, switches, etc.), information on the topology of the components, and information on service-level specifications. A study on network monitoring also proposed a schema that describes end-to-end links that cross the networks of different providers (domains)[51].

In standardization activities, especially in the DMTF, standard management models

[1] `https://www.cisco.com/c/ja_jp/products/cloud-systems-management/`
[2] `https://www.vmware.com/jp/products/vcenter-server.html`

have been developed[52, 53]. These models are used to describe the configurations of servers and switches on server-virtualization platforms.

From another viewpoint, a proposal for extensions to data-center bridging[54, 55] is somewhat related. These extensions change the architecture of the virtual networks. Therefore, a future task is to investigate how to manage these extended virtual networks.

1.2.2 Failure Management of Data Center Networks

A framework for achieving proactive network management was developed to predict exceptions in IP/MPLS networks by using OAM functions[56]. In addition, a framework to integrate service-level monitoring with fault management using Ethernet OAM was developed for the interconnected networks of Ethernet service providers so that they can rapidly identify the root cause of a failure in the networks[57]. However, these studies focused mainly on networks with fixed monitoring targets for network service providers.

From the viewpoint of network architecture, Software-Defined Networking (SDN) is a new approach for the enterprise with a centralized network controller that manages the routing of network flows[58, 59, 60]. Various fault management methods for virtualized environments that utilize a key feature of SDN to extract state and analytics information from network switches (e.g., OpenFlow switches) have been studied[61, 62, 63, 64, 65, 66]. For example, a mechanism to promptly detect channel failure between an OpenFlow controller and an OpenFlow switch while reducing the number of keep alive messages[67] was proposed. Further, to overcome the scalability limitations of the centralized fault management of software-defined networks, the OAM function has been distributed and deployed close to the data plane to provide a scalable means of data-plane connectivity monitoring and protection switching[68].

Ways of reducing the load on monitoring systems have also been investigated. Most of the monitoring systems that monitor a server's availability process the TCP/IP protocol with the software. In a large-scale data center with thousands or even several tens of thousands of servers running, server administrators must keep the transmission intervals of a ping long to reduce the load on monitoring systems when processing TCP/IP protocols. This, unfortunately, leads to a longer average failure detection time. To solve this problem, a high-performance monitoring server that utilizes a high-performance TCP software library or distributed system architecture has been developed[69]. There is also a monitoring method in which adjacent database servers monitor each other[70].

Hard-drive failure prediction methods based on machine learning techniques such as support vector machines and artificial neural networks have been studied for real-world

data centers[12, 13]. These methods utilize multiple attributes of the Self-Monitoring Analysis and Reporting (SMART) dataset extracted from hard drives. However, these are not suitable for large-scale data centers running many servers.

The reliability of monitoring systems in large-scale data centers has also been studied. For example, a monitoring method in which multiple monitoring systems are allocated to a server cluster has been proposed[71]. When one monitoring system fails, the other monitoring system checks the connectivity to the server cluster to avoid a missed failure detection and delay of service recovery. However, this method is not suitable for Ethernet OAM, which utilizes network devices acting as monitoring systems.

Selection of monitored targets is also studied to reduce the load on monitoring systems and networks. For example, a monitoring method to select monitored servers that affect surrounding servers is studied[72]. However, this method is not suitable for data centers in which multiple services of multiple users are independently provided. Ways of selecting which targets to monitor have also been investigated to reduce the load on monitoring systems and networks. For example, a monitoring method to select monitored servers that affect the surrounding servers has been developed[72]. However, this method is not suitable for data centers in which multiple services to multiple users are independently provided.

1.3 Research Strategy

On the basis of the above issues and prior studies, we take the approaches shown in Figure 1.3 to improve the operation and management of the networks in large-scale data centers.

(1) Integrated configuration management of virtual networks

We focus on the point that existing configuration management methods for virtual networks in data centers are independently implemented on the server side and the network side. We propose a centralized management method for virtual networks across servers and network devices. In addition, we propose a data model of the configurations of the virtual networks in data centers for the method. We evaluated the time that network administrators took to verify the configuration of virtual networks in a data center using the proposed method with a prototype of the network operation and management system for virtual networks.

(2) Dynamic monitoring of data-center networks

We focus on the frequent replacement of the physical server or VM in data centers

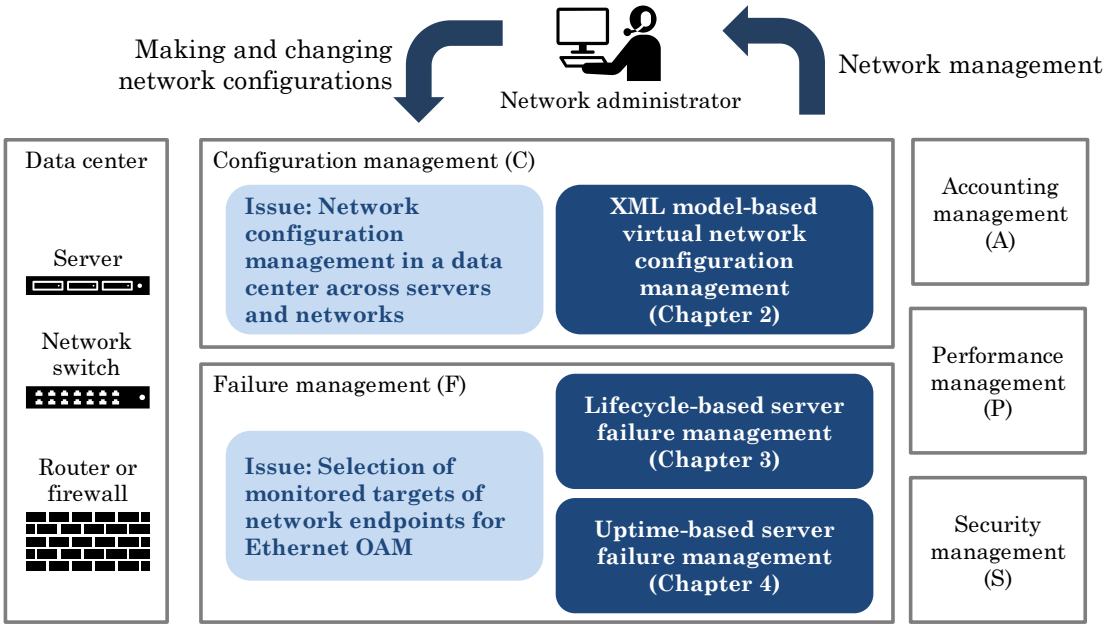


Figure 1.3: Issues and Research Strategies

where server virtualization has been introduced. We propose a dynamic monitoring method for data-center networks. The key feature of the method is the dynamic allocation of the monitoring resources of Ethernet OAM to dynamically select network end points, rather than static allocation of the resources. Specifically, we propose a dynamic monitoring method based on the server failure model and another one based on the system configuration information.

(2-1) Dynamic monitoring based on reliability model of VMs

We focus on the change in the failure rate in the lifecycle of a physical server or VM. We propose a dynamic monitoring method for data-center networks using Ethernet OAM based on a probabilistic reliability model of the VMs. We evaluated the improvement in failure localization operations using the proposed method by simulation.

(2-2) Dynamic monitoring based on server configuration information

We focus on the initial failure and the later failure due to wear-out of components, hereinafter referred to as wear-out failure, that can occur on all physical servers even if their operational states and failure characteristics are different. We propose a dynamic monitoring method for data-center networks using Ethernet OAM based

on the server configuration information, i.e., the uptime of servers. We evaluated the improvement in failure localization operations using the proposed method by simulation.

The reason we consider approaches (2-1) and (2-2) is that the amount and type of the management information are different for each data center. The advantage of approach (2-1) is its higher failure-detection rate, while with approach (2-2), the advantage is that the types of management information required are reduced. Therefore, data center network administrators and server administrators may choose to adopt either approach (2-1) or (2-2) depending on whether they are able to collect sufficient failure statistics to build the server's failure model.

1.4 Outline of the Dissertation

The organization of Chapter 2 to 5 of this dissertation is as follows:

Chapter 2 proposes a network configuration management method with a new data model that can model virtual networks across servers and network devices in an integrated manner and evaluates the operational load using a prototype system to achieve a data-center network with a low operational cost according to the system in publications[73, 74, 75, 76, 77, 78, 79, 80].

Chapter 3 proposes a network failure management method with dynamic monitoring of data-center networks based on a probabilistic reliability model of VMs and evaluates the time it takes to localize the root cause of failure using a simulation environment to achieve rapid localization of the root cause of a service failure in a large-scale data center running many VMs according to the method in publications[81, 82].

Chapter 4 proposes a network failure management method with dynamic monitoring of data-center networks based on the uptime of VMs and evaluates the time it takes to locate the root cause of failure using a simulation environment to achieve rapid localization of the root cause of a service failure in a large-scale data center that lacks failure logs of servers according to the method in a publication[83].

Chapter 5 summarizes the results obtained in this research and describes the topics remaining for future work.

Chapter 2

Configuration Management across Servers and Network Devices

2.1 Introduction

This chapter proposes a virtual-network configuration management system for large-scale data centers running many VMs with virtual networks. The effectiveness of the proposed system for improving the operational efficiency of the configuration verification process is evaluated.

In data centers, commercial or open source software (OSS) hypervisor-type server virtualization platforms such as VMware vSphere®[1] are utilized to make effective use of limited space. By utilizing these platforms, the virtual servers of multiple customers can be deployed on a single physical server, thus increasing the implementation density of the system. This can present challenges because the data center operators need to manage the configurations of many virtual machines. To help with this, the operators leverage the vendor-specific proprietary VM management APIs of the server virtualization platforms.

To operate data centers using server-virtualization technology, the operators need to manage virtual networks as well as VMs. Running multiple VMs enables the formation of an internal virtual network on a physical server, so the virtual networks exist across servers and network devices. On the server side, the above-mentioned proprietary management I/F of the server virtualization platforms is used to manage the internal virtual networks in physical servers. On the network side, among the various network-virtualization technologies such as VLAN, Virtual eXtensible LAN (VXLAN) and Network Virtualization

[1] `https://www.vmware.com/products/vsphere/`

using Generic Routing Encapsulation (NVGRE) that could be used, VLAN is typically deployed because it is widely supported by many network devices. Most network switches support Internet Standard Management Framework[84] to manage the configurations of VLAN. Prior studies have provided several methods for clarifying the structure of virtual networks in accordance with the configurations of VLAN switches and connection information[85, 86].

However, to reduce the cost of implementing management functions, servers with server-virtualization functions (virtualized servers) typically have configurations with forms that differ from those of VLAN switches. As a result, operators need to collect data on multiple forms of configurations, which increases the configuration management time.

The objective of this chapter is to provide a virtual-network management system that helps operators manage the configurations of virtual networks created in a data center that is a heterogeneous environment made of VLAN-supported switches and virtualized servers. Section 2.2 describes the current issues facing virtual-network management in data centers. Section 2.3 presents the proposed virtual-network configuration management system. Sections 2.4 and 2.5 report and discuss the evaluation results. The conclusion is presented in Section 2.6.

2.2 Issues in Configuration Management of Virtual Network in Data Center

2.2.1 Current Data-Center Management

As described above, the data-center service market has been increasing. To follow the market trend, data-center service providers invest continuously to upgrade and expand equipment of data centers. On the other hand, while expanding the size of the data centers, they try to control the increase in operations and management cost of data centers because the operations and management task does not directly generate revenues. Operators are thus required to manage large scale systems by fewer people. To reduce the operational cost, they have been automated their operational process by developing operational tools and improved their operational process.

However, as virtualization technologies for networks and servers has been introduced with the development of cloud computing, the operational scope was expanded to entire data-center network including servers and layer-2/3 switches to manage virtual networks provided on the data-center network. It causes complicated operational process and the

operators' tools became unable to manage the data centers. These are the operational issues in data-center services.

2.2.2 Prior Virtual-Network Configuration Management

As IT systems have become larger and more widely distributed, various network virtualization technologies have been developed. For layer-2 networks, VLAN was developed to virtually divide networks by attaching network identification tags to frames and standardized as IEEE 802.1Q. In addition, there are various overlay networking technologies. Provider Backbone Bridge (PBB), which is a L2-over-L2 overlay networking protocol, was developed mainly for wide area L2 network services and standardized as IEEE 802.1ah[87]. Virtual eXtensible LAN (VXLAN) and Network Virtualization using Generic Routing Encapsulation (NVGRE) were developed as L2-over-Layer 3 (L3) overlay networking protocols and standardized as IETF RFC 7348[88] and IETF RFC 7637[89], respectively. In addition, IP-in-IP was developed as an IP tunneling protocol for L3-over-L3 overlay networking and standardized as IETF RFC 2003[90]. Generic Routing Encapsulation (GRE) was standardized as IETF RFC 2784[91] and IETF RFC 2890[92] as well.

Among these network virtualization technologies, VLAN is the most basic one and is supported by many server-virtualization platform products and network-switch products. It is thus widely used in many data centers. Therefore, we focus on VLAN as a typical network-virtualization technology for data centers. An example of network virtualization using a VLAN is shown in Figure 2.1. Without VLAN activated, there is only a single physical network in which all servers can communicate with each other, as shown on the left. In contrast, with VLAN activated on a VLAN-supported network switch, there are multiple virtual networks on a physical network, as shown in the center. This is equivalent to running multiple physical networks with multiple network switches, as shown on the right.

The logical structure of a VLAN-enabled network varies according to the VLAN configurations. Therefore, to clarify the structure of the networks, operators have to manage the VLAN configurations shown in the center of Figure 2.1. A VLAN configuration is described as a table, in which each row has a set composed of a VLAN ID and the numbers of the ports assigned to the VLAN.

Management Information Base (MIB), part of the Internet Standard Management Framework[84], is a standard approach for defining management information. Figure 2.2 shows the management procedure for the previously described VLAN configurations through a MIB. The MIB is a database of management information about managed

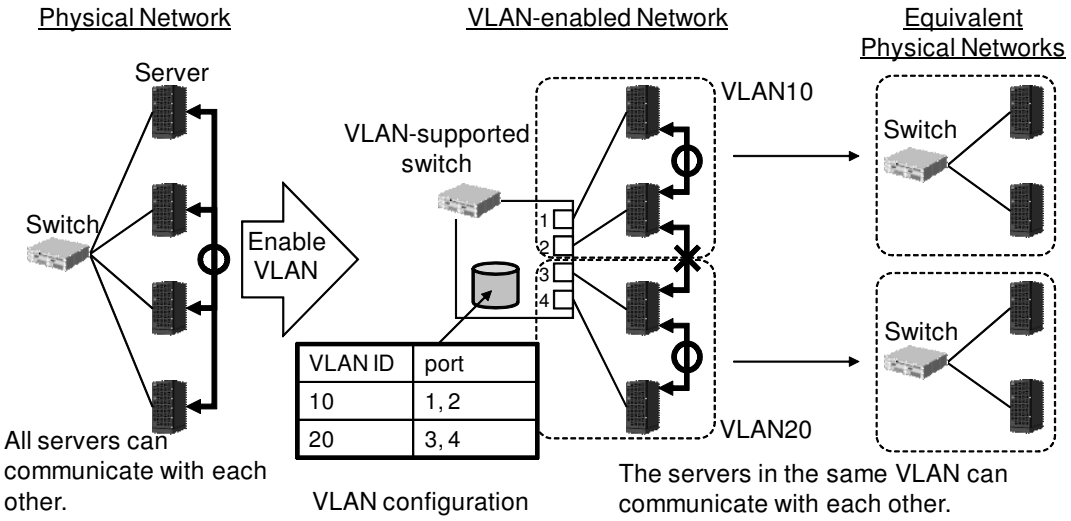


Figure 2.1: Network virtualization with VLAN

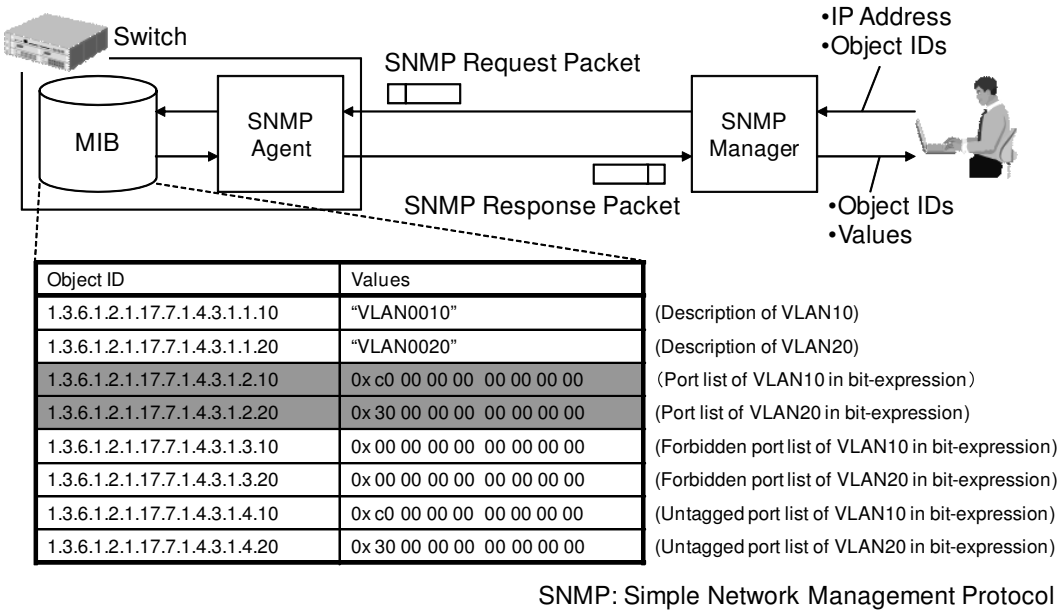
switches. The management information is determined by the object. Each row in the table in the bottom of Figure 2.2 represents an identifier and a value of a MIB object for the VLAN. Identifier, type and restriction of its value, and semantics of the value of each MIB object are defined by the external document, which is called MIB definition file. A MIB definition file also defines the data structure of a MIB.

It can be said that a characteristic of the MIB is separation of configuration data into multiple data store of the multiple devices. Although operators need to manage the VLAN configurations of multiple switches correlatively, the VLAN configuration described by a MIB contains only information about the switch that the VLAN configuration is acquired from.

2.2.3 Virtual-Network Configuration Management in Server Virtualization Environments

In data centers, commercial or open source software (OSS) hypervisor-type server virtualization platforms, e.g. VMware vSphere®[2], Microsoft Hyper-V™[3], and Kernel-based Virtual Machine (KVM)[4], are utilized to make effective use of limited space. Virtualized

[2] `https://www.vmware.com/products/vsphere/`
[3] `https://docs.microsoft.com/virtualization/hyper-v-on-windows/`
[4] `https://www.linux-kvm.org/page/Main_Page`



Object ID	Values	
1.3.6.1.2.1.17.7.1.4.3.1.1.10	"VLAN0010"	(Description of VLAN10)
1.3.6.1.2.1.17.7.1.4.3.1.1.20	"VLAN0020"	(Description of VLAN20)
1.3.6.1.2.1.17.7.1.4.3.1.2.10	0x c0 00 00 00 00 00 00 00	(Port list of VLAN10 in bit-expression)
1.3.6.1.2.1.17.7.1.4.3.1.2.20	0x 30 00 00 00 00 00 00 00	(Port list of VLAN20 in bit-expression)
1.3.6.1.2.1.17.7.1.4.3.1.3.10	0x 00 00 00 00 00 00 00 00	(Forbidden port list of VLAN10 in bit-expression)
1.3.6.1.2.1.17.7.1.4.3.1.3.20	0x 00 00 00 00 00 00 00 00	(Forbidden port list of VLAN20 in bit-expression)
1.3.6.1.2.1.17.7.1.4.3.1.4.10	0x c0 00 00 00 00 00 00 00	(Untagged port list of VLAN10 in bit-expression)
1.3.6.1.2.1.17.7.1.4.3.1.4.20	0x 30 00 00 00 00 00 00 00	(Untagged port list of VLAN20 in bit-expression)

SNMP: Simple Network Management Protocol

Figure 2.2: Collecting VLAN configurations with MIB

servers, as shown in Figure 2.3, run virtual switches (VSWs) to connect VMs on the same server in data centers in which server virtualization is used. The VSWs have MAC bridge[93] functionality to provide L2 connectivity among the VMs and physical network I/F cards (NICs). Since most of these virtual switches support VLAN technology, the structure of a virtual network will vary in accordance with its VLAN configuration.

Data center service providers offer VM resources as an extended menu of server hosting services. When an operator of the services receives a request from a customer who wants to use a particular VM resource, the operator creates a VLAN on VLAN-supported switches and creates a VM on a server virtualization platform. The operator then configures a VSW to connect the VM to the VLAN. The operator needs to isolate the traffic of multiple customers and connect each VLAN to the appropriate destination point inside and outside the data center.

To clarify the structure of virtual networks in the managed data center, the data-center operators must therefore manage the configurations of the virtualized servers in addition to those of the virtual switches. However, the configurations of the virtualized servers are vendor-specific, unlike those of the Internet-standard MIBs. Each configuration has its own semantics and syntax. Additionally, the management interface (I/F) of the

server virtualization platforms, e.g., VMware vSphere®, Automation API[5], and KVM API[6], are vendor-specific interfaces as well.

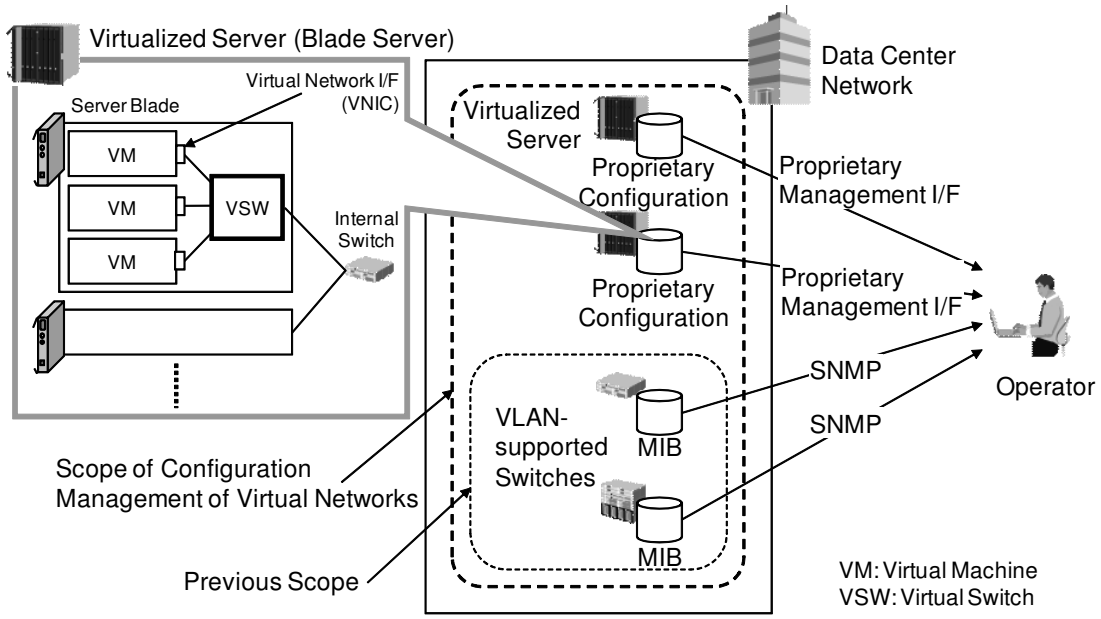


Figure 2.3: Virtual network management in a server-virtualized environment

Table 2.1 shows an example of VLAN configurations of a virtualized server. These configurations are modeled from the VM viewpoint. Each row of the table means the configuration status of each virtual-network I/F (VNIC) of VMs. It consists of the ID of the virtual switch that the VNIC connects, the MAC address of the VNIC, the ID of the VLAN that the VNIC accepts, and the mode of the assigned VLAN. Example configurations in Table 2.1 define that VMs 1$\tilde{4}$ run on the virtualized server. Also, VMs 1 and 2 are connected to both of VLAN 10 and VLAN 20. Further, VMs 3 and 4 are connected to VLAN 20. Every virtualized server in a data center stores configuration information like this.

In brief, models of VLAN configurations are different between the virtualized server and the VLAN-supported switch. As shown in Figure 2.2, VLAN-enabled switches define configurations of virtual networks by using the configuration information that uses a VLAN ID as an index. On the other hand, as shown in Table 2.1, the configuration information of virtualized servers is modeled from the server viewpoint. It uses the com-

[5] `https://developer.vmware.com/apis/vsphere-automation/`
[6] `https://www.kernel.org/doc/html/latest/virt/kvm/api.html`

Table 2.1: Example of VLAN configurations of a virtualized server

VM	VNIC	VSW	MAC Address	VLAN	Mode
1	0	1a	00:00:87:62:17:00	10	UnTag
1	1	1a	00:00:87:62:17:01	20	UnTag
2	0	1a	00:00:87:62:17:08	10	UnTag
2	1	1a	00:00:87:62:17:09	20	UnTag
3	0	1a	00:00:87:62:17:10	20	UnTag
4	0	1a	00:00:87:62:17:18	20	UnTag

bination of a VM ID and a VNIC ID as an index. It uses a VLAN ID as an attribute of a VNIC setting to define the VLAN-based virtual network in a virtualized server.

2.2.4 Issues in Virtual-Network Configuration Management in Server Virtualization Environments

The previously described characteristics of MIBs and configurations of virtualized server lead to the following issues in operations and management of virtual networks in a data center.

First, there is an issue of configuration verification time. In a large-scale data center, due to frequent requests for VM resources from customers, operators are required to change VLAN configurations of the managed devices including VLAN-supported switches and virtualized servers as rapidly as possible. For example, an operator changes a VLAN configuration every hour to connect/disconnect a VM. In this situation, due to the separated configurations and differences of structure and management I/F, the operators face a problem that they do not have sufficient time to verify the operational results of the VLAN configuration changes with configuration database or configuration orders. It leads to operational mistakes, for example, a VM is not connected to a supposed VLAN or a VM is isolated from all VLANs.

Second, there is an issue of configuration snapshot. In a large-scale data center, operators are required to manage large number of devices in the data center and their virtual entities. For example, an operator manages more than a thousand of VMs or several hundreds of VLANs. In this situation, due to the previously stated two characteristics of virtual-network management, the operator has difficulty in keeping higher

frequency of configuration snapshots about virtual-networks, because the time to make the snapshot increases as the number of managed devices increases. Thus, interval of the snapshots becomes long. Contents of such infrequent snapshots may differ from actual configurations of managed devices, and therefore cause operators lose actual state of the virtual networks in the data center.

Also, in prior researches for the virtualized servers, virtual-network configuration management methods that make it possible to deploy and manage the virtual network environment (VNE) regardless of the virtualization platform were studied[47, 48, 49]. The studies define a generic model extended based on Common Information Model (CIM) which is a standard information model by the Distributed Management Task Force (DMTF). For example, it is evaluated for a virtual network created with User Mode Linux[94] on a physical server. However, it assumes a single server-virtualization mechanism as a managed target and thus has a difficulty in managing virtual networks formed across servers and switches.

2.3 Proposal of Virtual-Network Configuration Management System

2.3.1 Overview of Proposed System and Issues

In this section, we aim to solve the above-mentioned problems of configuration verification time and the frequency of the configuration snapshot in the existing configuration management method, which manages independently virtual-network configurations of server virtualization platforms and those of VLAN-supported switches. We thus propose a new virtual-network configuration management system for large-scale data centers as shown in Figure 2.4.

The system stores configuration information of a data center including virtual networks in an integrated manner. We refer to the data model of the configuration information as a system configuration model. The data model must be able to describe the whole structure of the data center network and the configuration parameters of the virtual networks of virtualized servers and VLAN-supported switches. Also, it must be exchangeable and flexible enough that operators can utilize it for various purposes. The details of this system configuration model are described in Section 2.3.2.

To create a configuration file based on the system configuration model, the system has a configuration acquisition function, which is composed of a server-information acquisition

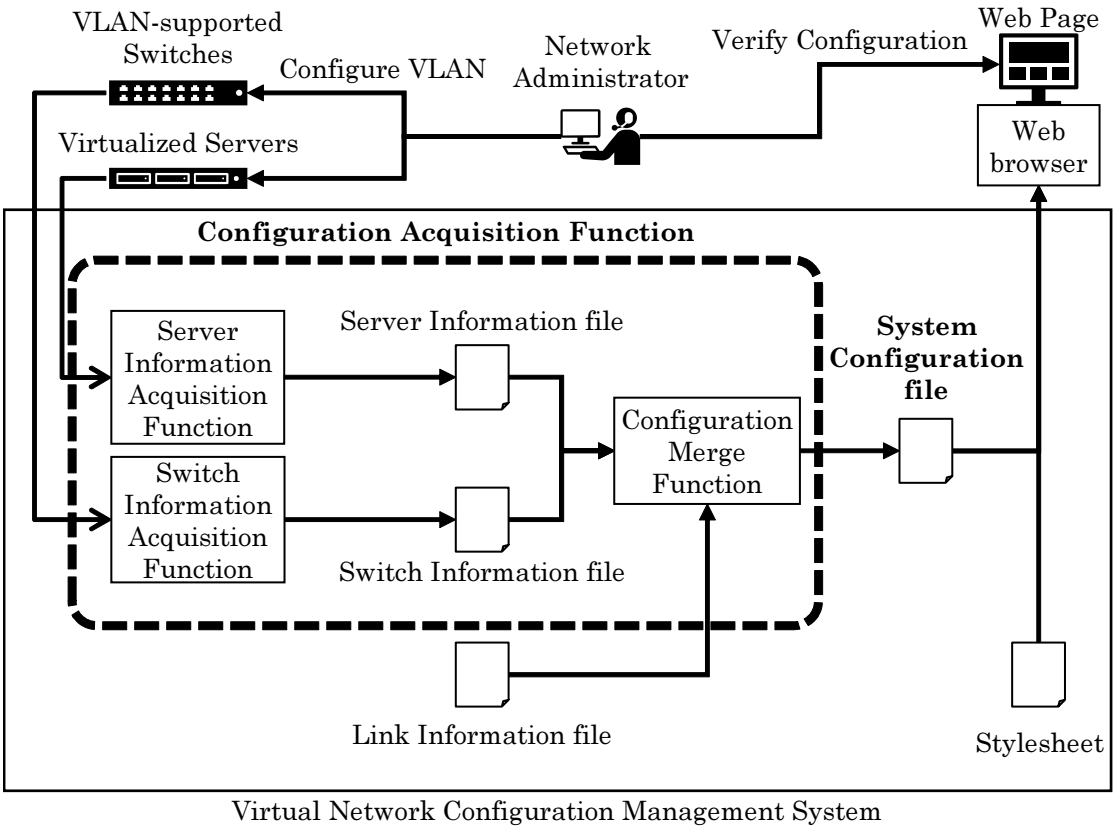


Figure 2.4: Architecture of the virtual-network configuration management system

function, a switch-information acquisition function and a configuration merge function. These functions collect VLAN configurations from managed devices via each management I/F and merge them into the system configuration file. The configuration acquisition function must be able to collect configuration information from monitored devices in a data center and make system configuration file in a short time. The detail of this configuration acquisition function is described in section 2.3.3.

The proposed virtual-network configuration management system generates automatically the management information of network configuration of virtual networks in a data center from multiple types of configuration information distributed across multiple devices in the data center. This enables integration management of virtual network configurations spanning the server virtualization platforms and VLAN-enabled switches of a large data center, which is difficult in prior studies. In addition, the network operator can provide an accurate history of the configurations of the virtual networks.

To adopt the proposed system to actual operations for multi-tenant data centers, it requires more management information such as tenant information which lists up served tenants and virtual resources allocated to them. However, this chapter does not touch on the point as it focuses on the configuration management of virtual networks across servers and network switches.

2.3.2 System Configuration model

The structure of the developed system configuration model is shown in Figure 2.5. It consists of the elements listed in Table 2.2. The numbers in the figure represent the multiplicity of the elements. The elements form a tree structure. Therefore, we use eXtensible Markup Language (XML)[95] to create a configuration file of virtual networks in a data center based on the system configuration model.

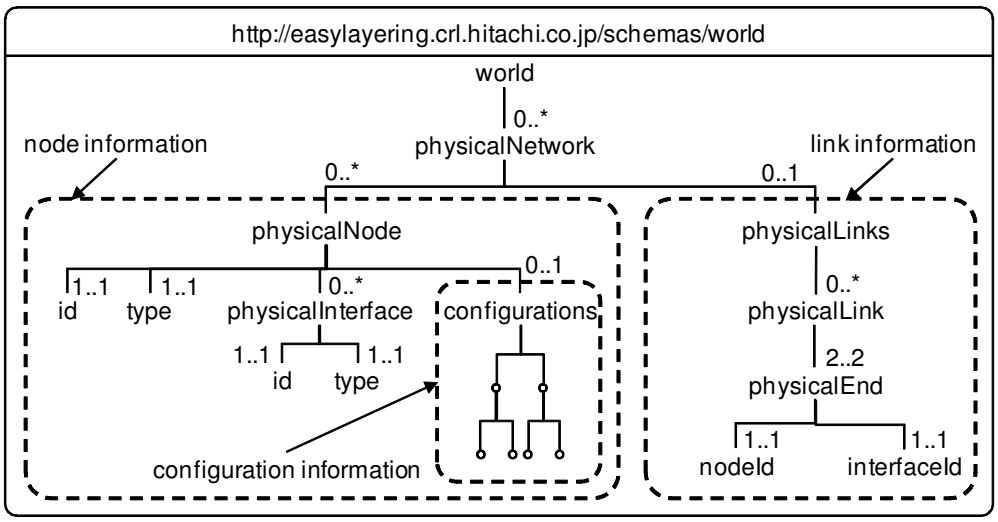


Figure 2.5: XML tree of a system configuration model

This proposed system configuration model has the following characteristics.

- Capability to model both of multiple VLAN-supported switches and virtualized servers
- Capability to store topology information
- Capability to store VLAN configurations

Table 2.2: Elements in a system configuration model

Element	Meaning
world	The whole network including multiple physical networks
physicalNetwork	A physical network
physicalNode	A physical node in a physical network
physicalInterface	A network I/F of a physical node
configurations	Configurations of a physical node
physicalLinks	A set of physical links
physicalLink	A physical link
physicalEnd	An end point of a physical link
nodeId	An identifier of a physical node
interfaceId	An identifier of a physical interface

First, the proposed model has a capability to model an entire data-center network including virtualized servers. Under a *world* element which means the whole managed environment, this model has a *physicalNetwork* element for each separated physical network. And, as its child element, the model has a *physicalNode* element for each managed device. A *physicalNode* represents an abstract device, which is actually a VLAN-supported switch or a virtualized server. A *type* attribute of a *physicalNode* element represents the type of the target device. As the child element of a *physicalNode* element, a *physicalInterface* element means a physical network interface; a NIC of a virtualized server or a network port of a VLAN-supported switch.

By these abstract XML elements, the proposed system configuration model can describe multiple types of managed devices including virtualized servers and VLAN-supported switches in the same manner. It then realizes to gather their management information into a single data store. Therefore, it can help operators to reduce their operational frequency to access management information, which is formerly distributed in multiple data stores.

Second, the proposed model has a capability to store management information about network topology. To describe network topology, the proposed XML model stores the link information. The link information represents direct and physical links between two devices. A *physicalLink* represents a link and has two sets of a device identifier and a port identifier of an end of a link. This link information enables the system configuration model to describe all types of network topologies that can be represented as non-directed

graphs.

Third, the proposed model has a capability to store the VLAN configuration of VLAN-supported switches and virtualized servers. The *configurations* element, which represents the configuration of a specific node in the managed network, has child nodes that represent the VM configurations of virtualized servers or the VLAN configurations of switches. Figures 2.6 and 2.7 show the XML trees of these child nodes.

The *HVMinfo* element shown in Figure 2.6, which represents the VM configuration, has *LparInfo* elements as child nodes. The child nodes correspond to a name, a CPU core number, and the amount of memory of a VM. The *LparInfo* element also has *VNICInfoArray* elements that represent VNICs and their VLAN IDs.

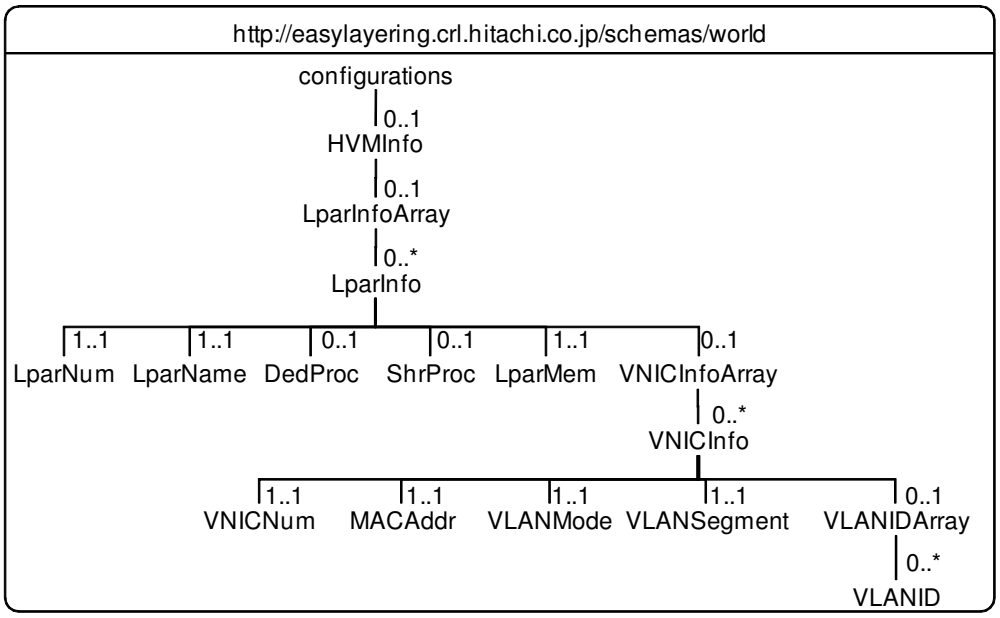


Figure 2.6: XML elements for VM configurations

The *configurations* element for the switch shown in Figure 2.7 has the XML tree defined in a standard VLAN data model[96]. It has an *AbstractionPorts* element that represents LAG (link aggregation)[97] configurations and a *Vlans* element that represents VLAN configurations as the child nodes. These two elements have a set composed of *AggregationPort* elements and a set of *Vlan* elements as child nodes, respectively.

By these definitions of data models, VLAN configurations of virtualized servers and VLAN-supported switches, formerly had each own syntax, are integrated in a single XML tree. It helps operators to reduce the time to verify VLAN configurations because they

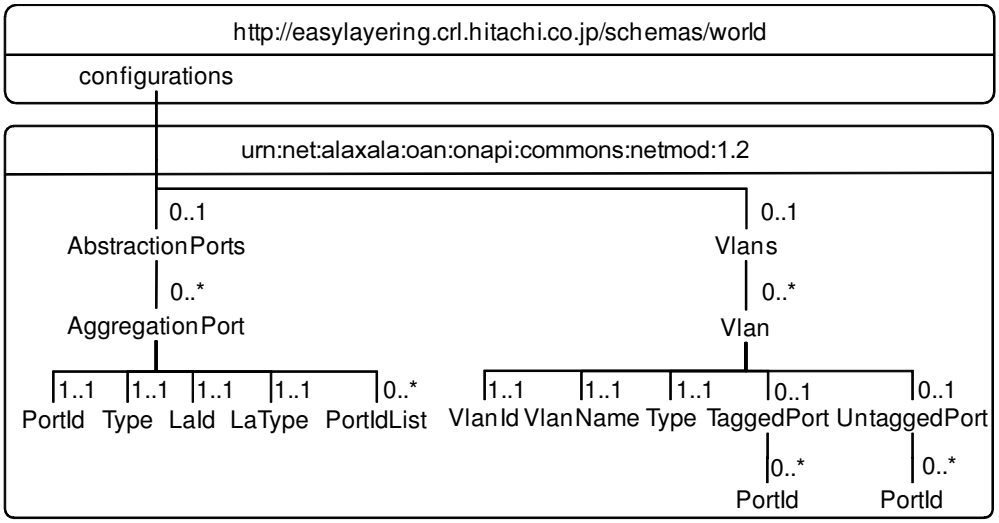


Figure 2.7: XML elements for VLAN configurations

have only to know a structure of the XML file without learning multiple types of syntax. Further, this proposed structure, in which a *physicalNode* element has a *configurations* element as a child, enables the proposed management system to merge VLAN configurations of multiple devices into an XML tree in a short time.

2.3.3 Configuration Acquisition Function

As previously described, the proposed configuration acquisition function is composed of three functions: a server-information acquisition function, a switch-information acquisition function, and a configuration merge function.

First, the server-information acquisition function, as shown in the upper half of Figure 2.8, collects the system information including the list of server blades, VM configurations, and configurations of network I/Fs, LAGs, and VLANs from virtualized servers. The VM configurations are collected via the proprietary management interfaces of the server-virtualization platforms of the virtualized servers. For each entry of the list of server blades, the function creates a *physicalNode* element under a *physicalNetwork* element. It also creates *physicalInterface* elements for each physical network I/F of the server blade according to the configurations of the network I/Fs. Further, it also creates *configurations* element under the created *physicalNode* element with the elements for each configurations of virtualized servers. It uses the proprietary management I/F of the management module of the server chassis, server-virtualization platform, and internal switch,

respectively.

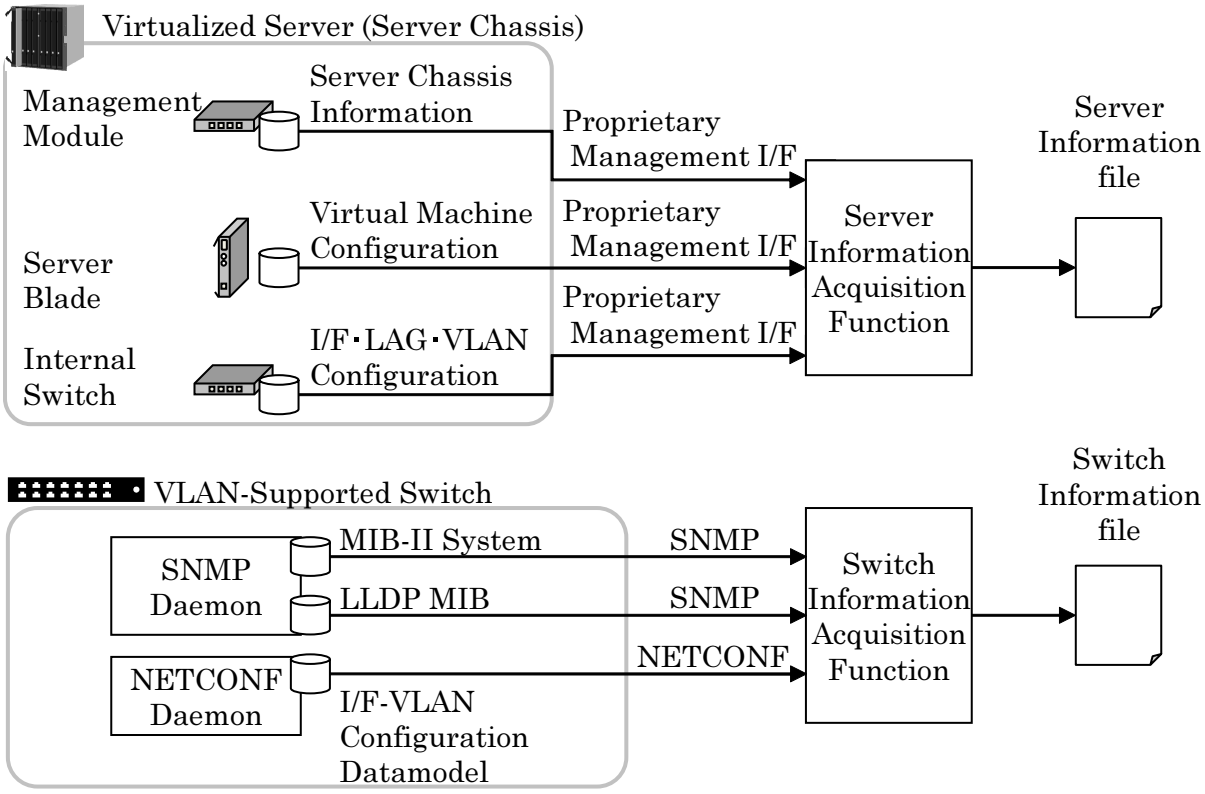


Figure 2.8: Generating a server information file and a switch information file

Second, the switch-information acquisition function, as shown in the lower half of Figure 2.8, collects configurations from the daemons on the switches, which manage MIBs and NETCONF[98] configuration data models. This function collects the contents of the MIB-II system group[99] for the *physicalNode* elements. This function gets the value of the *sysName* object of the MIB-II system group and sets the value to the *id* attribute of the *physicalNode* element. In addition to identifier of each device, this function gets the information about device type from *sysObjectID* object of the MIB-II system group and sets corresponding string to the *type* attribute of the *physicalNode* element. It also collects the contents of the LLDP (Link Layer Discovery Protocol) MIB[100] for the *physicalLinks* elements. To create these elements, it analyzes each entry in *lldpRemTable* table in the *lldpRemoteSystemData* group of the LLDP-MIB. According to the results, it sets the value of *lldpRemSysName* object and *lldpRemPortId* object to the *nodeId* and *interfaceId* of a *physicalEnd* element, respectively. A set of the two *physicalEnd* elements corresponding

to two ports facing each other are attached to a *physicalLink* element. Further, it uses the NETCONF protocol[98] to collect NETCONF configuration data models of network I/F and VLAN for *physicalInterface*, *AggregationPort*, and *Vlans* elements.

Third, the configuration merge function obtains multiple XML files compliant with the system configuration model and then merges them into a system configuration file. It rearranges the XML elements extracted from the input XML files in a XML tree. In this process, it merges multiple *physicalNetwork* elements that have the same value of *id* attributes. And, in turn, it merges *physicalNode* and *physicalInterface* similarly. However, in case of these two types of elements, even if value of *id* attribute matches between multiple XML elements, they are not merged if their parent XML elements are not identical.

These functions enable operators to acquire configurations continually from several tens of or several hundreds of target devices in a data center with fewer operations. As a result, the operators can keep high frequency of VLAN-configuration snapshots.

2.4 Evaluation

In this section, we evaluate the effectiveness of the proposed virtual-network configuration management system in improving the operational efficiency of the configuration verification process for virtual networks configured across physical servers and network switches. To confirm the effectiveness of the proposed method, we measure and evaluate the operational efficiency of actual verification processes for virtual networks.

2.4.1 Evaluation Environment

For the evaluation, we implemented a prototype of the virtual-network configuration management system with the proposed function implemented in software codes with Perl and Java. This prototype was run on a Windows Server® 2003 PC, incorporating a Xeon® 3 GHz CPU and 2 GB RAM. Also, we prepared a test network with the topology shown in Figure 2.9. Table 2.3 shows the number of devices in the network for each device type. This network consists of a blade-server chassis mounted with four server blades that support server virtualization. It also includes two VLAN-supported switches embedded in the blade-server chassis, hereinafter referred to as internal switches, and four VLAN-supported layer-2/3 switches.

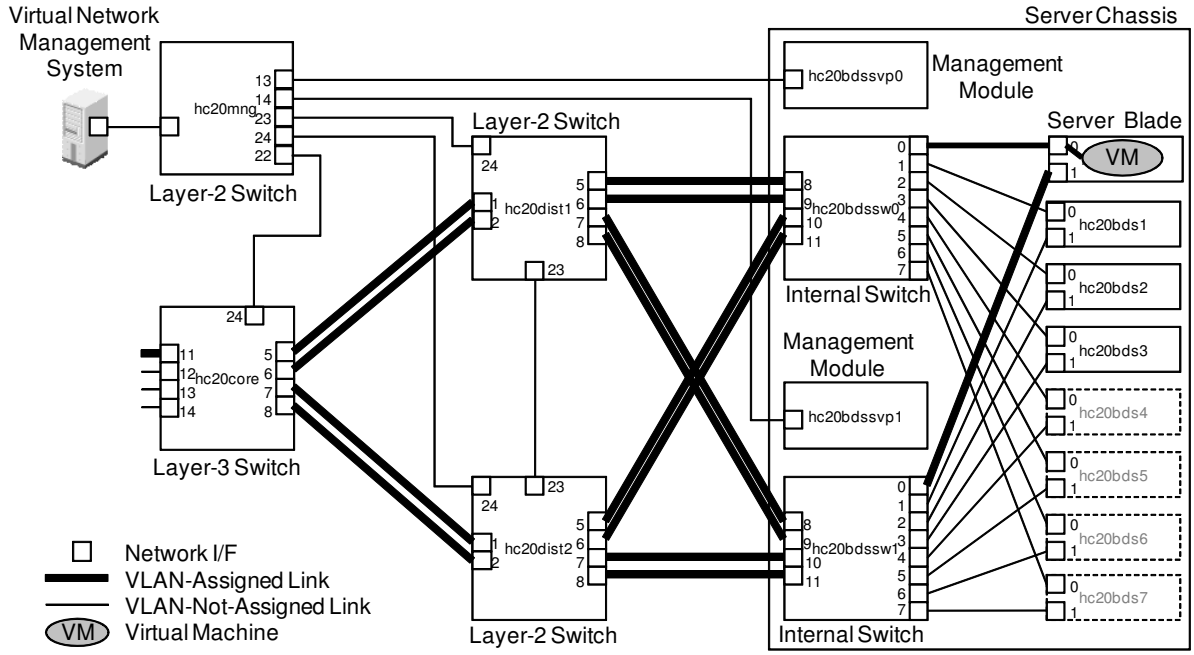


Figure 2.9: Network topology of evaluation environment

Table 2.3: Numbers of devices in the evaluation network

Device Type	Number
Server Chassis	1
Physical Servers (Blade, Server Virtualization and VLAN Supported)	4
Internal Switches (VLAN Supported)	2
Layer-2/3 Switches (VLAN Supported)	4

2.4.2 Evaluation Method

We evaluated the operational improvement on virtual-network configuration management by the following two experiments.

1. Measurement of the operational time to correct configuration management database
2. Measurement of the time to create system configuration XML file

First, to evaluate the improvement of a configuration verification operation by the

Table 2.4: Number of managed physical devices and configured virtual components in evaluation network

Type	Measurement 1	Measurement 2
Servers	4	$1 \rightarrow 3$
VMs/Server	1	$0 \rightarrow 8$
VMs	4	$0 \rightarrow 24$
Cores/VM	1	1
VNICs/VM	2	2
VLANs	4	4

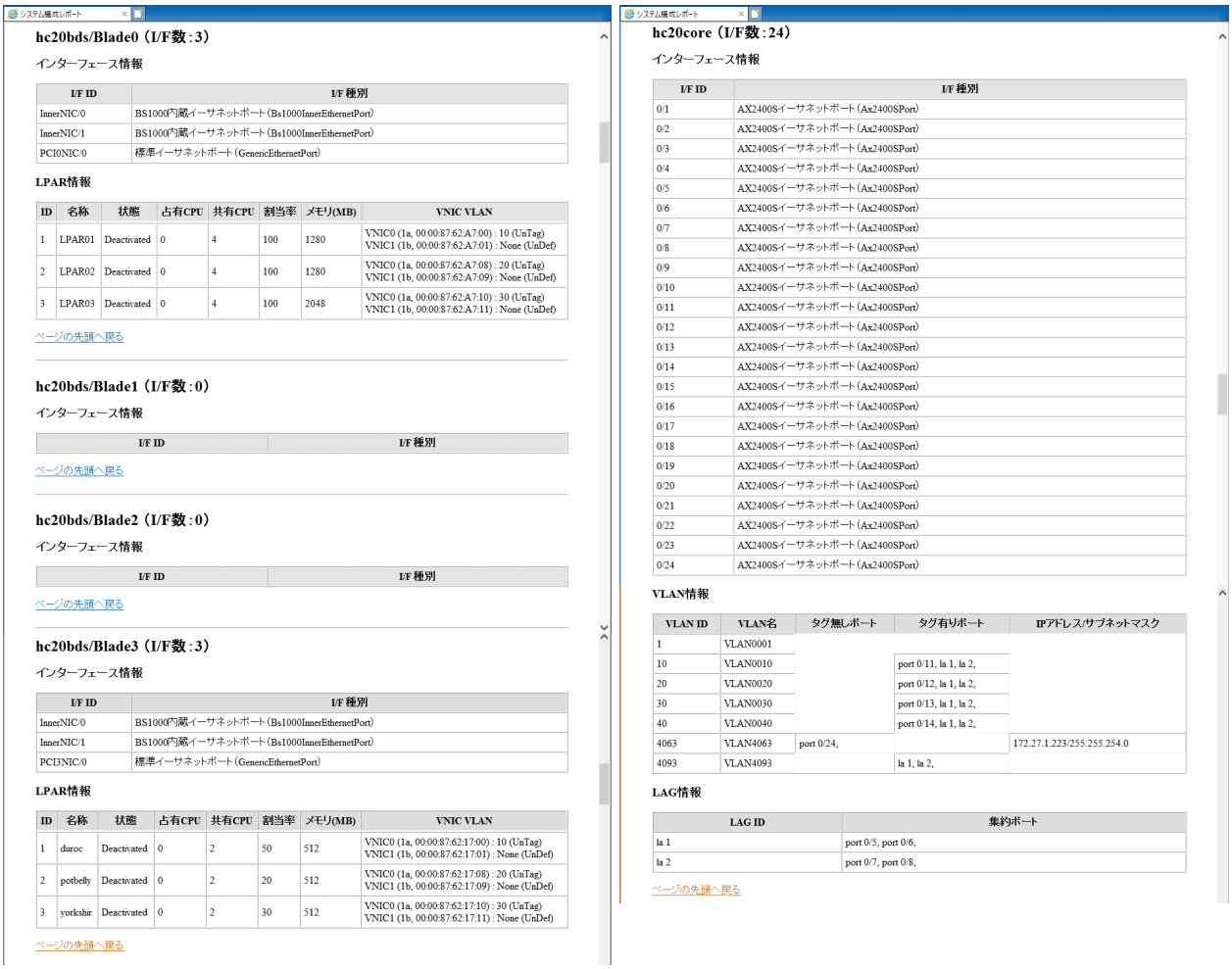
proposed system configuration model, we take a comparative experiment for an example operation by using our prototype of the proposed system.

As a pre-assumption of this experiment, we supposed a situation that an operator refers the configuration data on their managed devices and corrects the data on his configuration management database to correct the inconsistency between the data and actual configurations on the managed devices. This is a common scene in the large-scale systems because of difference of configuration formats and management I/Fs among managed devices. So, it can be said that operational efficiency improves if we can improve the efficiency of the correction operation.

The center column of Table 2.4 shows the number of managed physical devices and virtual components in the evaluation network. To create an inconsistency between the operator's database and the managed devices, we prepared the database with entries for only three VMs and three VLANs, and excluded entries for one VM and one VLAN that are configured and active on the devices. In the experiment, the database was provided as a spreadsheet file that lists up the VMs and the VLANs.

Five examinees tried the above-mentioned correction operation with 1) conventional operations featuring multiple types of configurations and management I/Fs and 2) the proposed system. When using the proposed system, the examinees utilized a system configuration file via a Web page generated through an XSLT stylesheet, as shown in Figure 2.10. During these operations, we measured the reduction of time for each operation, and the reduction of their operational processes. Hereinafter, the five examinees are referred to as A, B, C, D, and E.

The examinees are familiar with the VLAN and the server virtualization because



システム構成レポート

hc20bds/Blade0（I/F数：3）

インターフェース情報

I/F ID	I/F 種別
InnerNIC/0	BS1000内蔵イーサネットポート（Bs1000InnerEthernetPort）
InnerNIC/1	BS1000内蔵イーサネットポート（Bs1000InnerEthernetPort）
PCI0NIC/0	標準イーサネットポート（GenericEthernetPort）

LPAR情報

ID	名称	状態	占有CPU	共有CPU	割当率	メモリ(MB)	VNIC VLAN
1	LPAR01	Deactivated	0	4	100	1280	VNIC0 (1a, 00:00:87:62:A7:00) : 10 (UnTag) VNIC1 (1b, 00:00:87:62:A7:01) : None (UnDef)
2	LPAR02	Deactivated	0	4	100	1280	VNIC0 (1a, 00:00:87:62:A7:08) : 20 (UnTag) VNIC1 (1b, 00:00:87:62:A7:09) : None (UnDef)
3	LPAR03	Deactivated	0	4	100	2048	VNIC0 (1a, 00:00:87:62:A7:10) : 30 (UnTag) VNIC1 (1b, 00:00:87:62:A7:11) : None (UnDef)

ページの先頭へ戻る

hc20bds/Blade1（I/F数：0）

インターフェース情報

I/F ID	I/F 種別

ページの先頭へ戻る

hc20bds/Blade2（I/F数：0）

インターフェース情報

I/F ID	I/F 種別

ページの先頭へ戻る

hc20bds/Blade3（I/F数：3）

インターフェース情報

I/F ID	I/F 種別
InnerNIC/0	BS1000内蔵イーサネットポート（Bs1000InnerEthernetPort）
InnerNIC/1	BS1000内蔵イーサネットポート（Bs1000InnerEthernetPort）
PCI3NIC/0	標準イーサネットポート（GenericEthernetPort）

LPAR情報

ID	名称	状態	占有CPU	共有CPU	割当率	メモリ(MB)	VNIC VLAN
1	duroc	Deactivated	0	2	50	512	VNIC0 (1a, 00:00:87:62:17:00) : 10 (UnTag) VNIC1 (1b, 00:00:87:62:17:01) : None (UnDef)
2	potbelly	Deactivated	0	2	20	512	VNIC0 (1a, 00:00:87:62:17:08) : 20 (UnTag) VNIC1 (1b, 00:00:87:62:17:09) : None (UnDef)
3	yorkshir	Deactivated	0	2	30	512	VNIC0 (1a, 00:00:87:62:17:10) : 30 (UnTag) VNIC1 (1b, 00:00:87:62:17:11) : None (UnDef)

ページの先頭へ戻る

システム構成レポート

hc20core（I/F数：24）

インターフェース情報

I/F ID	I/F 種別
0/1	AX2400Sイーサネットポート（Ax2400SPort）
0/2	AX2400Sイーサネットポート（Ax2400SPort）
0/3	AX2400Sイーサネットポート（Ax2400SPort）
0/4	AX2400Sイーサネットポート（Ax2400SPort）
0/5	AX2400Sイーサネットポート（Ax2400SPort）
0/6	AX2400Sイーサネットポート（Ax2400SPort）
0/7	AX2400Sイーサネットポート（Ax2400SPort）
0/8	AX2400Sイーサネットポート（Ax2400SPort）
0/9	AX2400Sイーサネットポート（Ax2400SPort）
0/10	AX2400Sイーサネットポート（Ax2400SPort）
0/11	AX2400Sイーサネットポート（Ax2400SPort）
0/12	AX2400Sイーサネットポート（Ax2400SPort）
0/13	AX2400Sイーサネットポート（Ax2400SPort）
0/14	AX2400Sイーサネットポート（Ax2400SPort）
0/15	AX2400Sイーサネットポート（Ax2400SPort）
0/16	AX2400Sイーサネットポート（Ax2400SPort）
0/17	AX2400Sイーサネットポート（Ax2400SPort）
0/18	AX2400Sイーサネットポート（Ax2400SPort）
0/19	AX2400Sイーサネットポート（Ax2400SPort）
0/20	AX2400Sイーサネットポート（Ax2400SPort）
0/21	AX2400Sイーサネットポート（Ax2400SPort）
0/22	AX2400Sイーサネットポート（Ax2400SPort）
0/23	AX2400Sイーサネットポート（Ax2400SPort）
0/24	AX2400Sイーサネットポート（Ax2400SPort）

VLAN情報

VLAN ID	VLAN名	タグ無しポート	タグ有りポート	IPアドレス/サブネットマスク
1	VLAN0001			
10	VLAN0010		port 0/11, la 1, la 2,	
20	VLAN0020		port 0/12, la 1, la 2,	
30	VLAN0030		port 0/13, la 1, la 2,	
40	VLAN0040		port 0/14, la 1, la 2,	
4063	VLAN4063	port 0/24,		172.27.1.223/255.255.254.0
4093	VLAN4093		la 1, la 2,	

LAG情報

LAG ID	集約ポート
la 1	port 0/5, port 0/6,
la 2	port 0/7, port 0/8,

ページの先頭へ戻る

Figure 2.10: Screenshot of Web page system configuration file in proposed method

they are researchers or engineers about server virtualization (A, C, D), a network researcher (B), and a network administrator (E). As a result, the experiment can keep off the examinees' time to learn and understand VLAN and server virtualization, and therefore, can evaluate adequately the effectiveness of the proposed method to reduce an operational time.

Second, to evaluate the improvement of configuration snapshot operation, we measured the time the examinees took to correct their configuration management database. In detail, we investigated increase of the time according to the increase of the number of VMs. If the time do not increase so much, it can be said that the proposed system has high scalability because operators can keep update frequency of their database. The

right column of the Table 2.4 shows the number of managed elements in the evaluation network of this measurement.

We used three server blades as the managed virtualized servers and allocated one core for each virtual machine. Since each server blade is equipped with eight cores, we can create up to a maximum of 24 VMs on the evaluation network. Therefore, we increased the number of VMs from 0 to 24 in sequence and measured the operational time and file size of the exported system configuration file in each case. For every eight VMs, we added a new server blade.

2.4.3 Evaluation Result

Operational Time to Verify Configuration

Figure 2.11 and Figure 2.12 show evaluation results of operational time to verify VLAN configurations, to find out and to correct the inconsistency of VLAN configurations between the configuration management database and actual devices.

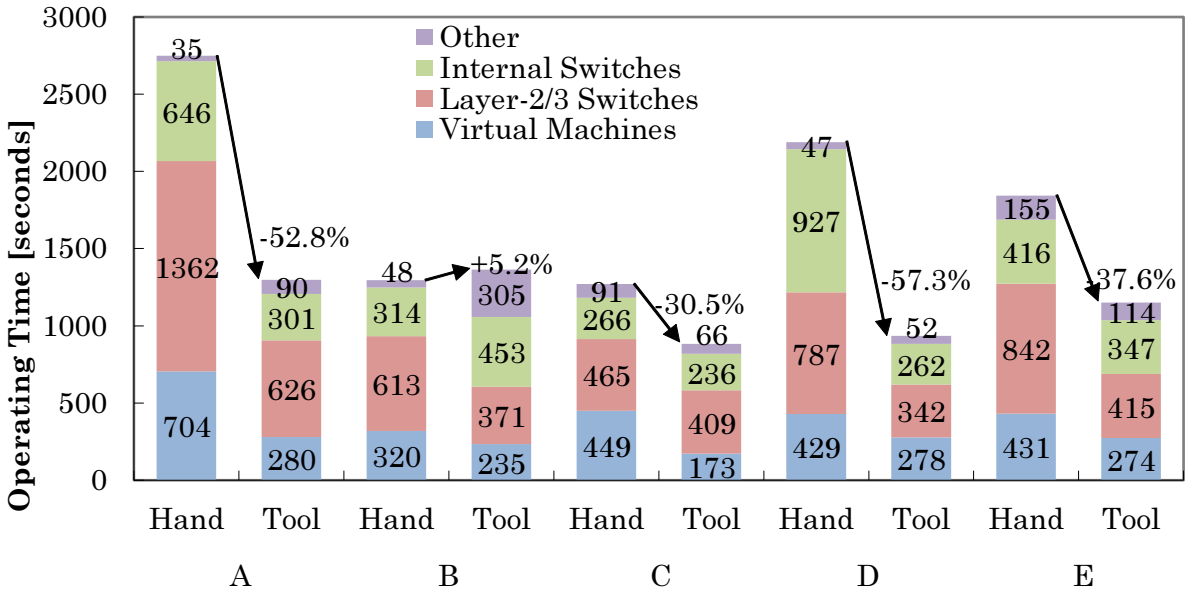


Figure 2.11: Time required to correct configuration management database

Figure 2.11 shows the comparative measurement results for the time taken by the operator to update the configuration sheets. With the conventional method ("Hand"), an average 1870 seconds was needed, and with the proposed method ("Tool"), only 1126 sec-

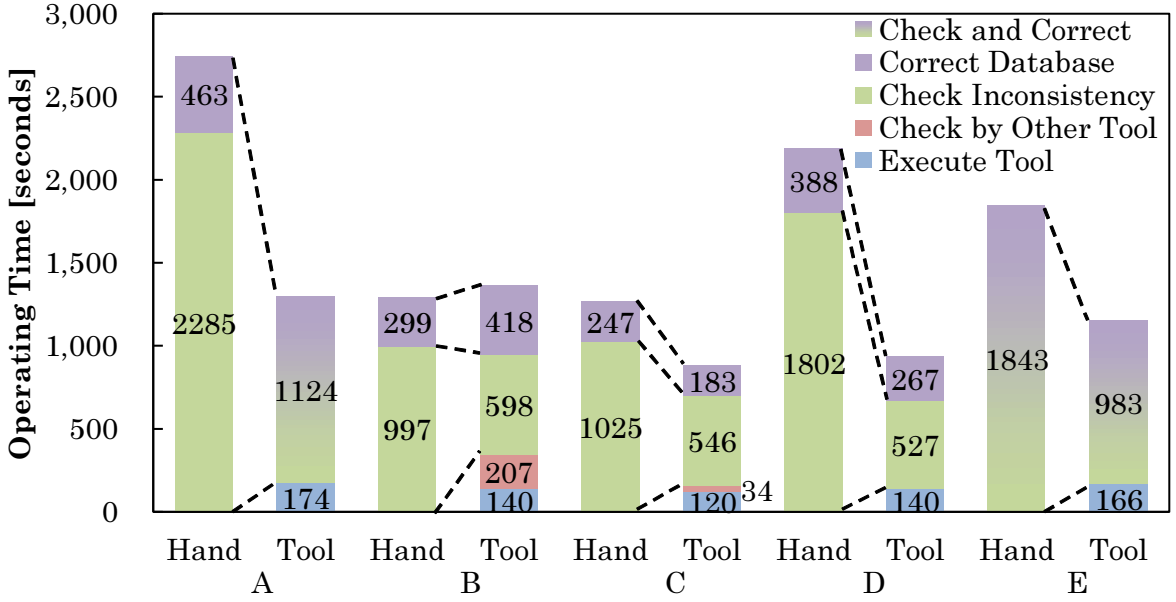


Figure 2.12: Time reduction in each operational process

onds was required; in other words, the proposed method reduced the time by maximally 57.3% and 39.8% on average. Moreover, the proportions of the times for each task are not significantly different in the two cases.

In addition, we investigated the breakdown of operational time by operational processes. Figure 2.12 shows the results of the comparative experiments of five examinees. For the conventional operations ("Hand"), the measured time is divided into check configurations operation and correct configurations operation. On the other hand, for the proposed method ("Tool"), the measured time is divided into four types of operations: tool execution, check by other tool, and previous two operations. The first operation means that the examinee creates a system configuration file by the proposed system. Second one means that the examinee checks configurations by an external management tool for the system configuration file. In these measurements, we could not separate the measured time of examinees A and E into check operation and correction operation. These operations are shown as a "check and correct" operation.

These results showed that operators reduced the time for check operations by 70.7% maximally, and 52.5% in average. This time reduction is more than the case of correction operations. To get these values of time reduction, results of examinee A and E are not included because their operations could not be distinguished.

However, as shown in Figure 2.11, only examinee B's time increased (be 5.2%). The first reason is that the examinee failed to transcribe the information about internal switches to his database. And the second reason is that, in addition to the proposed system, the examinee used other network management tool to double-check actual configurations of managed devices. If this additional operation is excluded, the result is shortened to 1156 seconds.

Since the examinee B used the additional tool for double check after the examinee finished the check-inconsistency operation and correct-configuration operation, the effect of the additional tool is independent from that of the proposed system. Therefore, as far as we evaluate the effect of the proposed system, we can remove the time corresponding to the additional tool from the result. Therefore, it can be said that all examinee can reduce their operational time if they use only the proposed system.

Figure 2.13 shows the difference of process of examinee C between the conventional method and the proposed method. At the conventional operations, for each virtualized server, VLAN-supported switch, and internal switch, he took three operations: login to the device's CLI, check inconsistency, and correction of configuration management database. That is to say, his operations are separated into nine blocks of operations. On the other hand, at the measurement with proposed system, his operations are distinguished into three parts of operations: creation of system configuration file, check inconsistency, and correction of the system configuration file.

As shown in Figure 2.11, there is no difference in the ratio of the time for each device type between the conventional method and the proposed system. From this result, it can be said that device type is not improvement factor of operational time in the experiment. On the other hand, as shown in Figure 2.12, time reduction in check-inconsistency operation is larger than that in correct-configuration operation. This result implies that time reduction in check-inconsistency operation contributes to the total time reduction.

We would like to consider how the time reduction in check-inconsistency operation is realized. As shown in Figure 2.13, the examinee C's operational process changed when he used the proposed system. In case of conventional method, for each device type, he did a series of operations from login to correction at a time. In the case of the proposed system, on the other hand, he did each check-inconsistency operation for all device types at once. This iteration of the operations is an important improvement factor of the time reduction.

Such continuous operations are realized by the proposed system configuration model. When the examinee used the proposed system configuration file, he was able to get con-

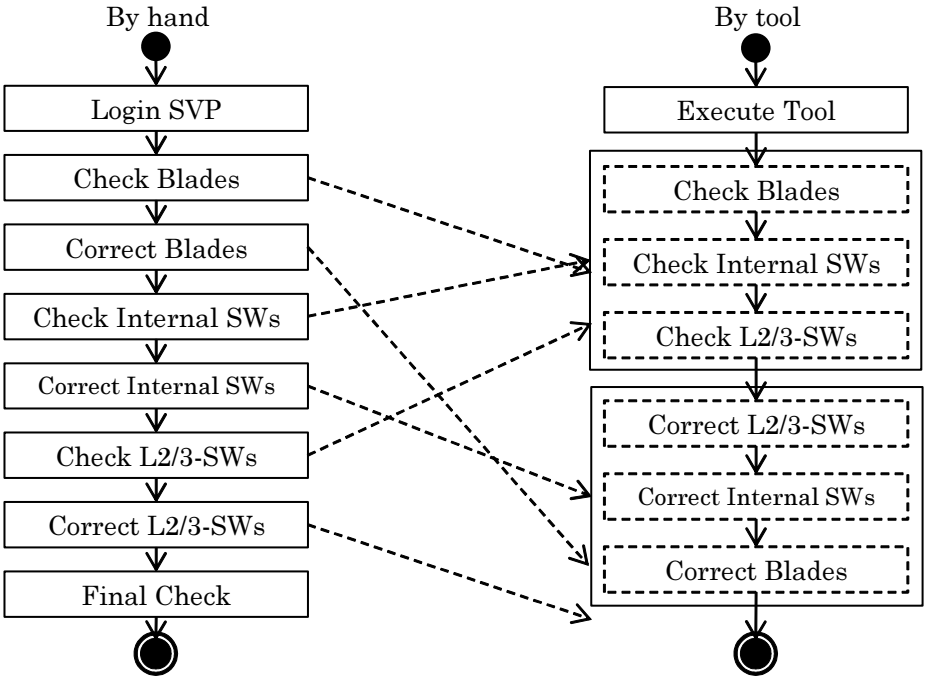


Figure 2.13: Comparison of operational process of an operator

figuration data of all managed devices from this file. Therefore, when he got an exported system configuration file, he checked configurations of all target devices at once, and later corrected the configuration management database according to those check results.

Operational Time to Make Configuration Snapshot

Figure 2.14 plots the measurement results of the time taken to create a system configuration file with the virtual-network management system. When we break down the total time in terms of the target devices, only the segment for VM increases. On the contrary, during the entire period, the time for other segments does not change. Figure 2.15 shows the extracted result of the time to acquire VM configurations. It increases almost linearly with the number of VMs. Furthermore, when the number of virtualized servers increases, the time increases further.

Furthermore, Table 2.5 shows the breakdown of a result in Figure 2.15. In this case, we created three VMs and measured the time to acquire the configurations of each component in the VM configurations. When the number of VMs increases, the time taken to acquire information about the status and VNIC, and the time to export a system

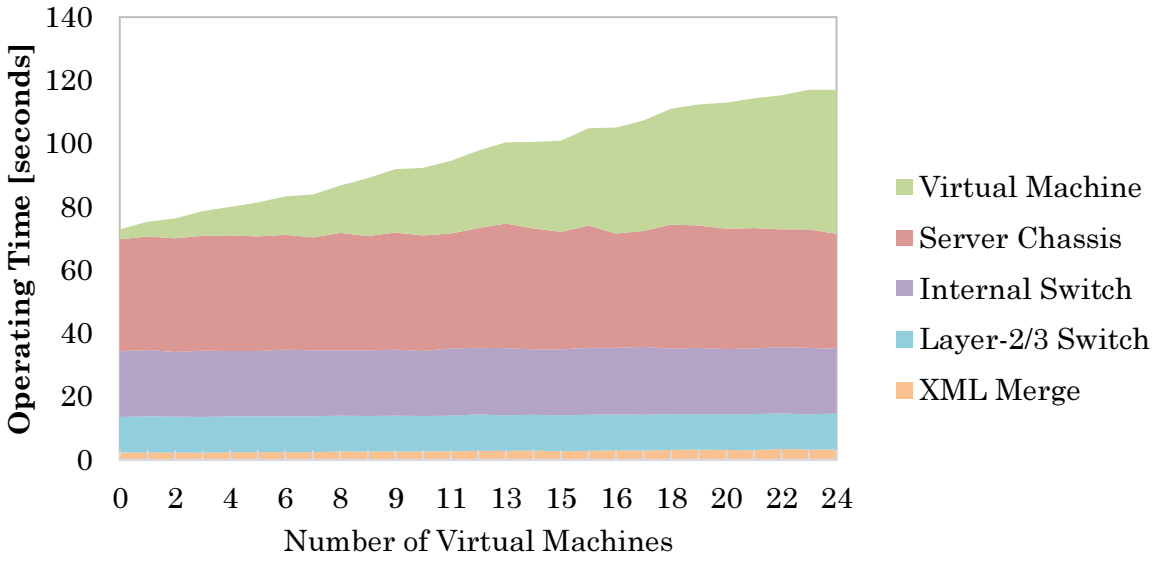


Figure 2.14: Time required to create a system configuration file

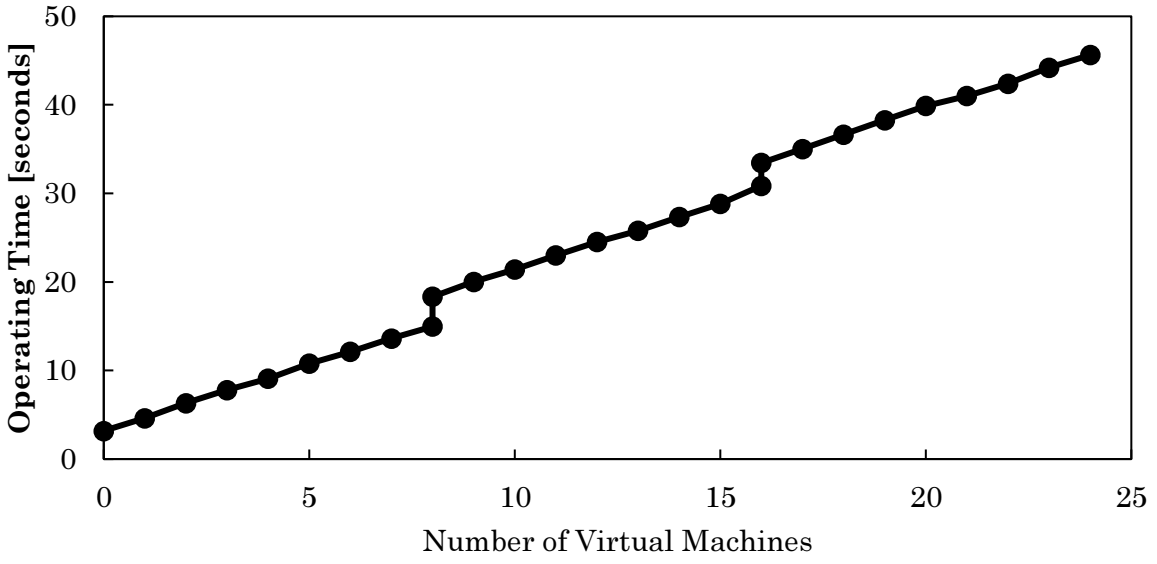


Figure 2.15: Time required to obtain the VM configurations

Table 2.5: Time required to obtain each component in the VM configurations

Components/Processes	Time
Initialization	0.188 seconds (2.42%)
Name	1.047 seconds (13.48%)
Status	1.284 seconds (16.53%)
Processor	0.135 seconds (1.74%)
PCI device	0.287 seconds (3.70%)
FC-HBA	0.044 seconds (0.57%)
VNIC	4.734 seconds (60.96%)
XML export	0.047 seconds (0.61%)
Total	7.766 seconds (100%)

configuration file, increase. In particular, the time taken to acquire VNIC information increases from 1.32 seconds to 31.6 seconds when the number of VMs increases from 0 to 24.

2.5 Discussion

2.5.1 Effect of System Configuration model

According to the evaluation results in Section 2.4.3, it can be said that the unit of operational processes changes from a device to a type of operation by the proposed system configuration model. If the operations and management processes are divided by equipment, the number of the processes increases in large data centers with many devices. The verification process for configuration information of one device, when proceeding from a process for a device to a process for other device, a network administrator is necessary to change the device to which a network management system connects. Therefore, as the number of processes increases, the time to change the device affects the verification process. On the other hand, when we adopt the proposed system configuration model, the verification process for virtual network configuration is consolidated to the process of checking and correcting process without overhead to switch the connected destination of the network management system. It can be said that this change is effective to improve operational efficiency of virtual-network configuration management.

An unresolved issue in this chapter is automation of the operators' correct-configuration operations. Since we use XML as the data representation format of configuration data for managed devices, it is expected to automate to correct the operators' configuration management database according to the configuration data. However, there remain two problems to be solved to automate the operations. One problem is that the time to find the difference between the configuration data and the database increases as the size of the configuration data increases. The other problem is that a management system cannot apply deterministically the detected change to the database because there are several cases that the changes should be ignored or confirmed by the operators.

2.5.2 Effect of Configuration Acquisition Function

At first, we would like to consider further reduction of the time to create a system configuration file. As shown in Table 2.5, the time to acquire VNIC configurations occupies more than 60% of the total time to acquire VM configurations. The main reason of this high ratio is that our prototype is implemented to acquire configurations about all of 8 VNICs a VM could have, even if few VNIC is activated. So, if we implement the prototype as it gets only the configurations about activated VNICs, the time to create a system configuration file would be reduced. Then, scalability would be improved.

Further, we wanted to estimate the time taken to create a system configuration file in a large-scale data center based on the previously described result. Since the time increases almost linearly with the number of VMs, we made a linear projection. We assumed the two cases in Table 2.6 as an extension of the previously described evaluation network in Figure 2.9. Estimation (1) assumes a fully mounted server chassis, and estimation (2) assumes five server chassis to completely fill the port of layer-2/3 switches. In both cases, there were four layer-2/3 switches, which is the same as that of the evaluation network in Figure 2.9.

Table 2.6: Numbers of devices in the networks for estimation

	Evaluation	Estimation (1)	Estimation (2)
Switches	4	4	4
Server chassis	1	1	5
Server blades	3	8	40
VMs	24	64	320

We calculated the estimated time for the above two cases by linear regression method. In detail, we estimated the time of each of VMs, server chassis, internal switches, layer-2/3 switches, and configuration merge process, by extrapolating on respective regression lines calculated based on the experimental results shown in Figure 2.14.

Table 2.7 lists the previously described evaluation result and the estimated time taken to create a system configuration file in the two cases of networks. The total time was 196.1 seconds and 927.2 seconds in estimation (1) and estimation (2), respectively.

These estimated results can vary due to the margin of error included in the above experimentally measured results. The error is summed up to 2.14 seconds, which is defined as the sum of the standard deviations of data sets corresponding to the rows of the Table 2.7.

From these estimations, it can be said that the proposed function takes less than 20 minutes to acquire configuration information about a virtual network that includes 300 VMs. Consequently, the operators in data centers can update and maintain current network configurations every hour. Furthermore, they can automate the entire process by launching the proposed function from Cron or task scheduler services.

Table 2.7: Estimated time to create a system configuration file

	Evaluation	Estimation (1)	Estimation (2)
VMs	45.61 seconds	121.8 seconds	609.7 seconds
Server chassis	36.33 seconds	37.08 seconds	185.4 seconds
Internal switches	20.45 seconds	20.85 seconds	104.3 seconds
Layer-2/3 switches	11.32 seconds	11.32 seconds	11.32 seconds
Configuration merge	3.234 seconds	5.071 seconds	16.49 seconds
Total	117.0 seconds	196.1 seconds	927.2 seconds

Additionally, we would like to point out the possibility of increased operational time. In this experiment, the proposed system acquired the name and type of the managed layer-2/3 switches via management objects in the MIB-II. However, several layer-2/3 switches could provide Web-I/F or CLI rather than the MIB and SNMP. In such cases, increased operational time would be required because Web-I/F and CLI often require much time to return the responses than MIB.

2.6 Conclusion

This chapter presented a virtual-network configuration management system for data centers. The proposed system provides operators with configuration information about virtual networks by integrating the VLAN configurations of virtualized servers and VLAN-supported switches into a single XML tree based on a new data model called system configuration model. Further, it automates the entire process of constructing the configuration information by acquiring the VLAN configurations directly from the managed devices via their configuration I/F.

We evaluated its improvement in operational efficiency by conducting a configuration verification operation and a configuration snapshot operation using a prototype on a test network with a maximum of 24 VMs. The results showed that the proposed system reduces the operational time taken to verify VLAN configurations and reduces the inconsistency between the actual configurations on managed devices and the configuration management database by about 40%. The results also showed that it can automate the configuration management for virtual networks including virtualized servers, and that it has sufficient scalability to update, on an hourly basis, the configuration data for several hundred VMs.

These results demonstrate that the proposed system is effective for improving the configuration-management process of virtual networks in data centers.

As the number of IT services offered by data centers increases, the demand for variety in both the performance and cost of the services will only intensify. Therefore, data center service operators or cloud service operators will be using different types of server virtualization platforms depending on the performance and cost requirements of the users. As the data models of the configuration information of virtual network functions differ among the server virtualization platforms of different vendors, a future issue will be determining how to manage the configurations of a virtual network across multiple server virtualization platforms with different types of data models[101, 102].

Chapter 3

Failure Management with Dynamically Prioritized Monitoring according to Lifecycle of Virtual Machines

3.1 Introduction

This chapter proposes a network failure management method with dynamic monitoring of data-center networks. The goal is to achieve rapid localization of the root causes of service failures in a large-scale data center running many virtual machines. The monitoring is based on the reliability models of virtual machines. The average time to localize the root causes of failures is evaluated using a simulation environment.

It is difficult to localize the root cause rapidly, since failures are managed independently on the server side and the network side. On the server side, ICMP is used as a standard protocol to monitor the availability of servers, and on the network side, Ethernet OAM is used as a standard protocol to monitor failures in the network. The network failure management method has limited monitoring resources, as it must utilize a large number of resources of the network devices to achieve rapid detection of failures. Therefore, not all routes in the data-center network can be monitored, especially in large-scale data centers.

To keep the number of monitored routes in the data-center network low, we focus on the change in the failure rate of a VM through its lifecycle. A network monitoring system dynamically selects a limited number of ports to be monitored in a network using

the network failure management method, depending on the change in the failure rate of the VM.

The contribution of this chapter is to demonstrate that root causes can be localized more rapidly than before. Our method utilizes fewer monitoring resources thanks to monitoring only some of the routes in the networks of a data center.

Section 3.2 describes issues in network failure management in large-scale data centers with Ethernet OAM. Section 3.3 proposes an efficient virtual network monitoring system based on the lifecycle of the physical servers and VMs in the data center. Section 3.4 evaluates the effectiveness of the proposed method by simulation. In addition, we examine the characteristics of the assumed model parameters when they are different from the model parameters being monitored. Section 3.5 provides the conclusion.

3.2 Issues in Network Failure Management with Ethernet OAM in Data Center

Communication protocols are standardized and widely used to remotely monitor server availability and network connectivity in data centers. ICMP is the standard protocol for network management systems to monitor servers and network devices in TCP/IP networks by exchanging monitoring messages, well known as "ping". Specifically, network management systems use the combination of "echo request" messages and "echo reply" messages defined in ICMP to check the availability of IP hosts.

Also, Ethernet OAM is being developed to monitor connectivity in Ethernet-based WANs. IEEE 802.1ag and ITU-T Y.1731 are defined as standards of Ethernet OAM. Ethernet OAM functions are generally implemented in network devices. The maintenance end point (MEP) functions of Ethernet OAM-supported devices transmit and receive continuously monitoring frames to check connectivity, as shown in Figure 3.1. This function is called Ethernet OAM Continuity Check (CC). Since wide-area networks are shared by multiple users, failures within them need to be detected rapidly, often within milliseconds. The monitoring frames are frequently sent for rapid failure detection.

However, the combination of ICMP Echo (ping) and Ethernet OAM CC is not sufficient for the failure management of large-scale data centers. As mentioned above, a huge number of VMs, VNFs, and physical servers are connected to the networks in large-scale data centers. Due to their increasing complexity and size, failures in data-center networks inevitably cause service failures in the data centers. Thus, the need to monitor data-center networks closely has increased. Ethernet OAM was originally designed for

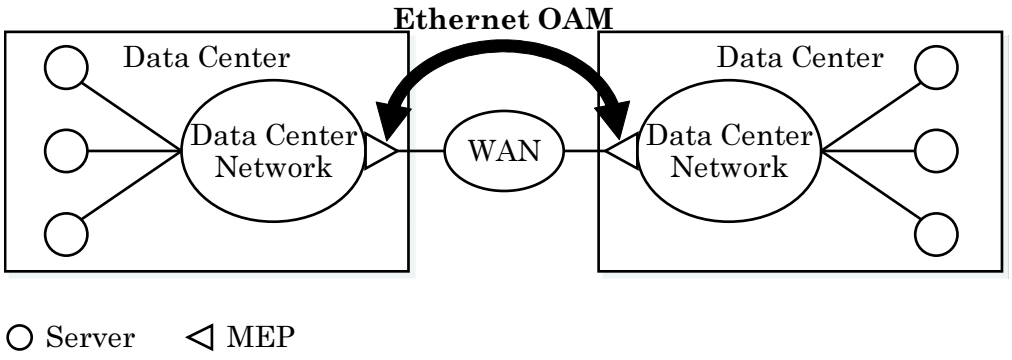


Figure 3.1: Connectivity Check of Ethernet-Based Wide-Area Network with Ethernet OAM

monitoring aggregated links in wide-area networks among multiple data centers. Ethernet OAM CC leads to a network scalability problem due to heavy periodical traffic[103]. In most cases, the number of devices that can be monitored with Ethernet OAM CC is less than the actual number of VMs and physical servers. This means that not all failures in a data-center network can be detected with Ethernet OAM CC and that the network connectivity has to be checked after the failure is detected in the service layer, as shown in Figure 3.2. Loop Back (LB), which is a standard function of Ethernet OAM, is typically used for temporal connectivity checks. As an additional setup is required for Ethernet OAM LB, it takes longer to obtain a connectivity check result than with Ethernet OAM CC. One solution to this issue is to scan the network by periodically changing the port on the switch that activates the MEP of Ethernet OAM. However, since the monitoring interval becomes longer to patrol a large number of targets, the time to detect the failure becomes longer.

3.3 Proposal of Dynamically Prioritized Monitoring based on Lifecycle of Virtual Machines

3.3.1 Method Overview

We propose a new method of monitoring to reduce the average failure localization time. Figure 3.3 shows the failure management method in a data center with ICMP Echo (ping) and Ethernet OAM CC.

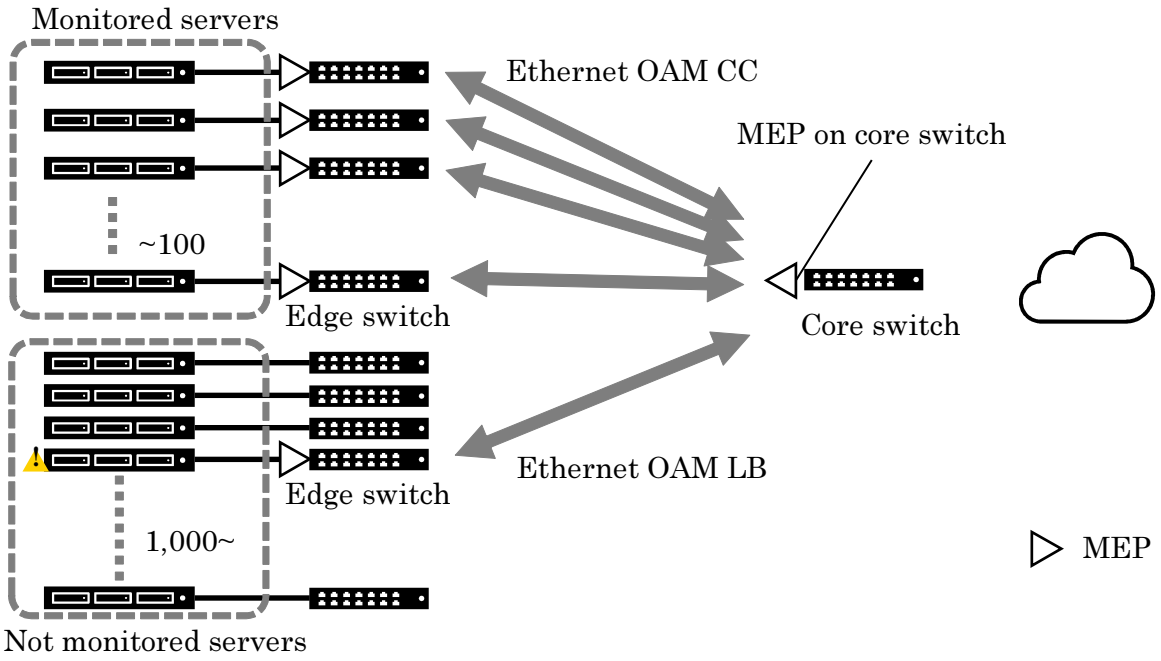


Figure 3.2: Configuration and Execution of Ethernet OAM LB for a Not-Monitored Server after Detection of Service Failure

Since the number of ports that can be monitored with Ethernet OAM in a data center is limited, network administrators must carefully select the most effective set of ports to manage failures. In the event of a service failure detected by ICMP Echo (ping), network administrators want to immediately check the connectivity between the core network and the port that is connected to the VMs providing the failed service. At that time, if the port is being continuously monitored with Ethernet OAM CC, the administrator can immediately identify whether the server side or the network side is the cause of the failure. However, it is difficult to predict VM failures based on network management information (e.g., statistics of network traffic), especially in large-scale data centers with many VMs.

With a focus on the reliability of VMs, which will vary through their lifecycles, we propose a method to model the varying failure rate of a VM and to select the monitored target based on the failure rate. We call this the reliability model-based failure management (RFM) method.

With RFM, a network monitoring system continuously selects a set of ports to be monitored with Ethernet OAM CC, giving high priority to ports that connect to VMs with high estimated failure rates. The failure rates are estimated based on a reliability

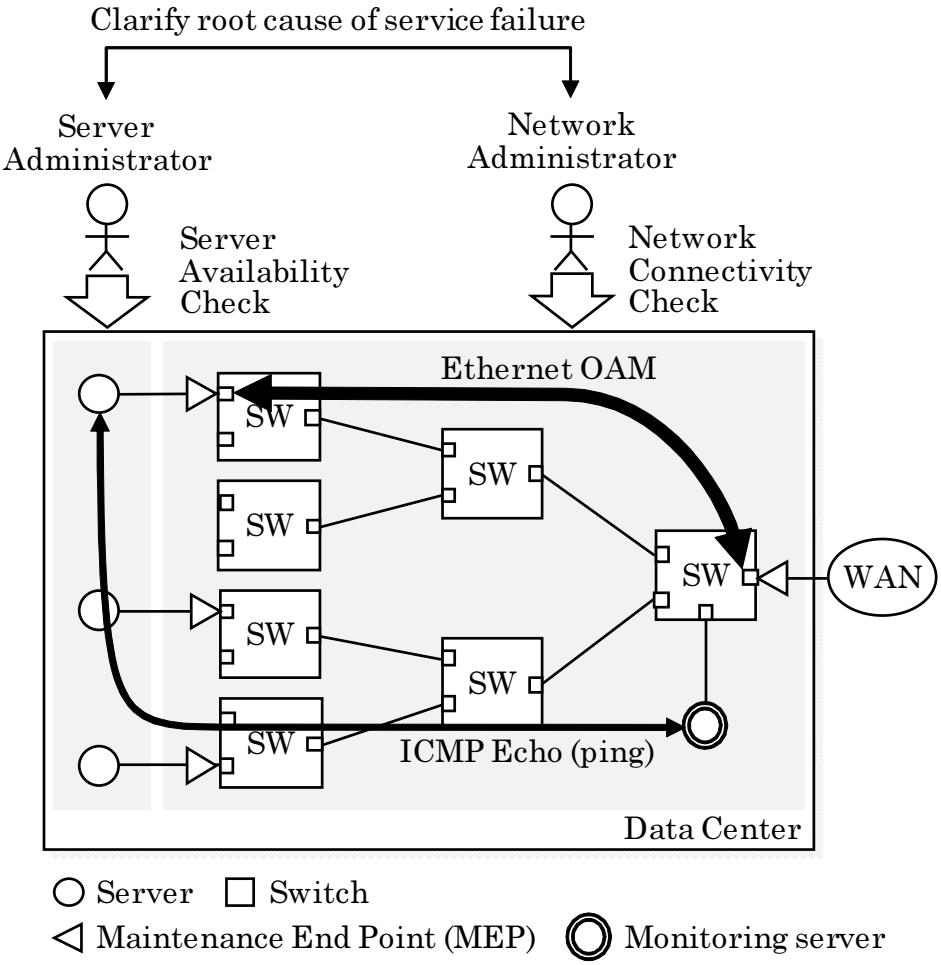


Figure 3.3: Overview of Proposed Monitoring for Data Center

model of the VMs. Therefore, it is likely that the port is being monitored with Ethernet OAM CC in the event of a VM failure. In such cases, the root cause, i.e., the VM, is localized rapidly since the network side is immediately proved available by Ethernet OAM CC. This leads to a reduction in the average time taken to localize the root causes of failures.

3.3.2 Failure Management Operations with RFM Method

Usually, there are two types of administrators in a data center—server administrators who use ICMP Echo (ping) to monitor servers and network administrators who use Ethernet

OAM CC to monitor networks—and they need to cooperate to clarify the root cause of service failures. When a server administrator detects a service failure by noticing there are missing responses to ICMP Echo (ping) messages from a monitoring server, the root cause of the failure is not yet clarified. The network administrator thus explores the Ethernet OAM monitoring point (called MEP) of the port that the server is connected to. If the network administrator finds a failure, it means the root cause of the service failure exists on the network side. If no failure is detected, the root cause is considered to exist on the server side.

We introduce a new concept of dynamically selecting the VMs to be monitored based on the stage of the lifecycle of VMs and servers. In the past, the monitoring priorities of VMs have been fixed throughout the lifecycle. However, when an IT system deploys a new physical server to activate a VM and changes its settings, if the elapsed time since it was activated is long, the hardware running the VM is likely to fail. Therefore, by increasing the allocation monitoring resources for VMs in such a stage, and by decreasing the allocation monitoring resources for VMs in other stages, it should be possible to shorten the average failure detection time without increasing the number of monitoring resources.

To select which port to monitor, it is first necessary to define and calculate the priority, which changes in accordance with the lifecycle stage of each VM. Therefore, we change the monitoring priority of a VM in accordance with the time-series change of the failure rate per VM. An example of the time-series change is shown in Figure 3.4. The pattern of failure rate for the operating time of a typical hardware (including HDDs) on the server in a data center follows a bathtub curve: the failure rate shows a high value early on, decreases to a lower value, and eventually maintains the value over time, with the high value appearing again (due to wear) at the end of the lifecycle[104, 4]. It is the same with software, i.e., undiscovered defects will cause high failure rates early in the life of a software program and the curve will flatten after the defects are corrected. However, any configuration changes that are made cause the failure rate curve to spike. Before the curve can return to the original steady-state failure rate, another change is requested, causing the curve to spike again. Slowly, the minimum failure rate level begins to rise[105]. Therefore, since the initial failure rate is high at the time of introduction to the server, the monitoring priority of a VM running on the server is high, and the priority due to the initial failure is lowered over this period. In contrast, the monitoring priority increases due to wear-out failure over time and configuration changes to VMs.

Figure 3.5 shows which VMs in the data center are selected for monitoring based on

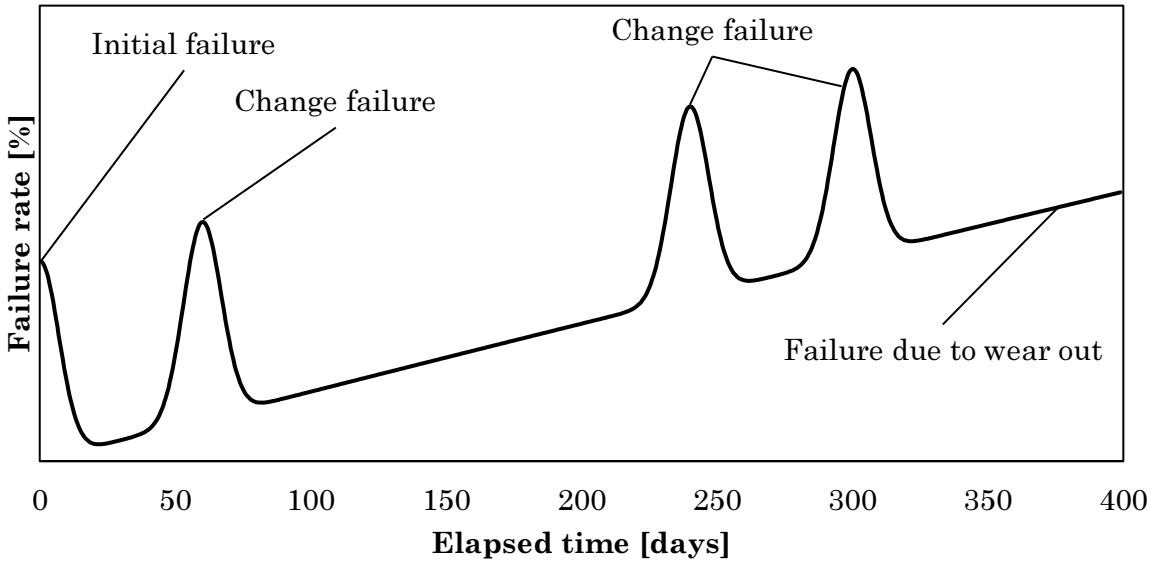


Figure 3.4: An Example of Time-Series Change of Failure Rate

the monitoring priority calculated for each VM. The time variation of the selected VM combinations is also shown. The four graphs on the left show the time variation of the monitoring priority for each of the four VMs. The thick lines represent the state in which the VM is selected for monitoring.

In this example, the number of MEPs that can be allocated is limited to two. Two MEPs are first allocated to network switch SW21 to monitor the connectivity to the ports connected to two servers when the servers activated two VMs in Feb. 2016. When the third server with a VM was connected to network switch SW22 in Mar. 2016, the port that connects to the first server was excluded from the monitored ports and the port that connects to the third server was added to the monitored ports. In addition, when the VM on the second server moved to the third server in Apr. 2016, the priority of the port of network switch SW22 that the third server is connected to was increased to reflect the increase in the failure rate due to the configuration change, and was also added to the monitored ports. At the same time, the port of network switch SW21 connecting to the first server was re-added to the monitored ports because the failure rate of the first server increased due to it being larger than the third server installed in Mar. 2016.

The network monitoring system using the proposed RFM method chooses which port of the network switch of the data center should be monitored for Ethernet OAM based

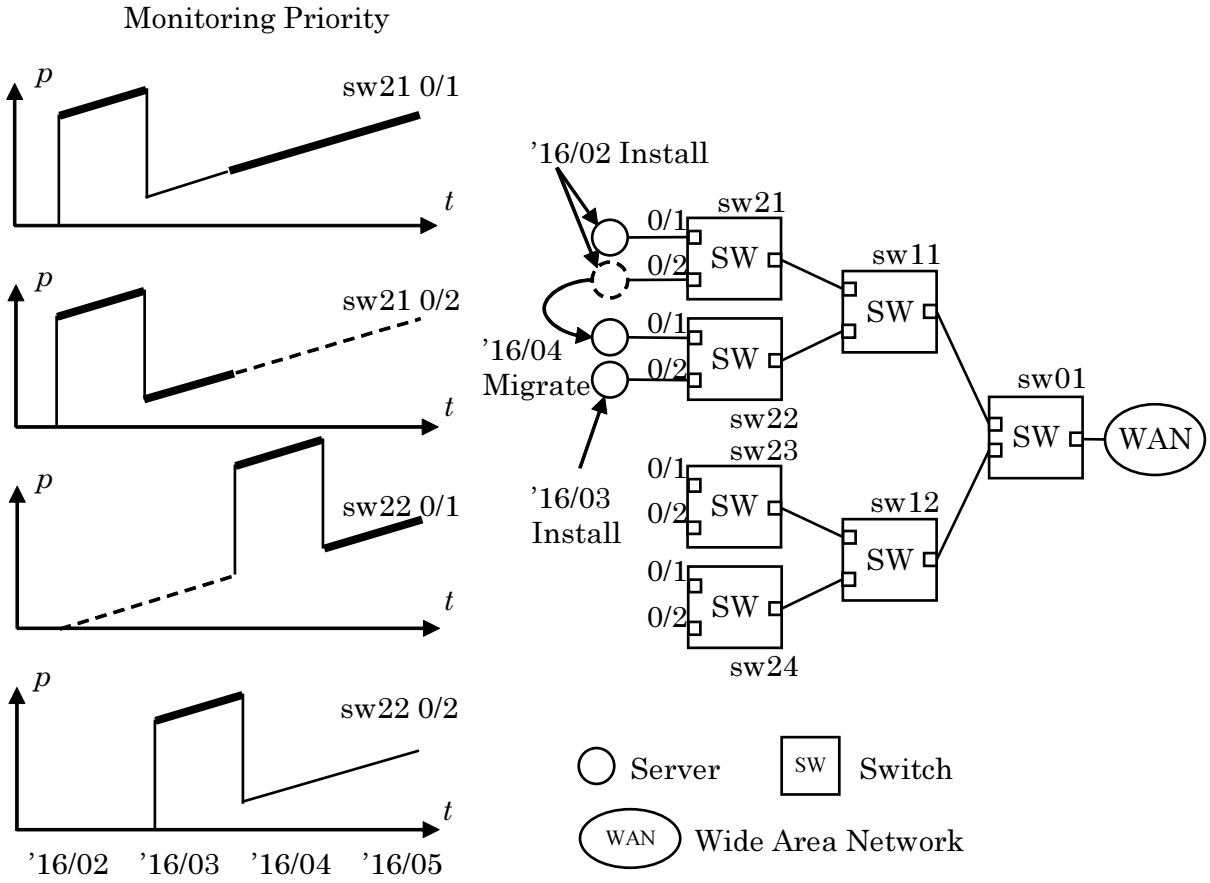


Figure 3.5: Server Monitoring Priority

on the above-mentioned port monitoring priority. The network monitoring system, based on the port monitoring priority calculated in the manner described above, selects which port among all of the ports connected to the server should be monitored. In this case, the system selects the monitored port so that it does not exceed the maximum programmable MEP number of each switch.

3.4 Evaluation

As discussed in Section 3.3, our objective here is to select which VMs to monitor while minimizing the number of failures that are not detected. We therefore measured and evaluated the average server failure detection rates to determine the effectiveness of the

RFM method. Also, based on the average server failure detection rate, we calculated and evaluated the average failure localization time as an operational metric. The parameters used to calculate it are described at the end of Section 3.4.1.

3.4.1 Evaluation Method

We compared (1) the proposed RFM method with (2) a conventional method using Ethernet OAM CC and (3) a ping-only method, as shown in Table 3.1. All three methods use ICMP Echo (ping) to detect failures at the service level. Further, in method (1), a set of ports is selected periodically as the ports to be monitored with Ethernet OAM CC in accordance with the estimated failure rates of the corresponding VMs. In method (2), a set of ports is randomly selected as the ports to be monitored with Ethernet OAM CC. This method is equivalent to the preferential link tomography methods of preferentially selecting the target to be monitored preferentially independently of the failure rate, such as the importance of the site[106, 107]. In method (3), Ethernet OAM CC is not used. The average failure localization time was evaluated for all three methods. The average server failure detection rate was evaluated only for methods (1) and (2), since Ethernet OAM CC is not used in method (3).

Table 3.1: Evaluated Methods

Method	ICMP Echo (ping)	Ethernet OAM CC	Port Selection Method
Proposal	Used	Used	RFM Method
Conventional	Used	Used	Random Allocation
Ping only	Used	Not Used	N/A

To evaluate the applicability for large-scale data centers, the evaluation was conducted as a simulation so that we could prepare a large-scale evaluation environment. A discrete event network simulator was developed in C++. When the simulation starts, the object of the network node with the start-up time information is generated for each server. For each time step, the simulator checks if server failure happened or not for each object based on its uptime, configuration changes, and the probability distribution of server failure. Also, it dynamically selects the objects that are monitored by using Ethernet OAM with MEP from all objects in accordance with the operating principle of the proposed RFM method. At the same time, the simulator checks if the server failures have been detected with Ethernet OAM. If MEP is configured for a network node in

failure, the Ethernet OAM quickly detects the failure. On the other hand, if MEP is not configured, the failure is detected by ICMP Echo (ping) as usual. On the basis of the results, the simulator calculates the average failure localization time for the entire data center.

The simulation parameters are listed in Table 3.2. The simulator simultaneously runs 1,000 network nodes as simulated servers with the step interval of seven days for five years. A new network node is generated as a new running server when a network node is evaluated as failed. Each network node increases its failure rate periodically, since the simulated servers change their connections to the data-center network every 100 days.

Table 3.2: Simulation Parameters

Parameter	Value
Number of nodes	1,000
Simulation duration	5 years
Simulation step interval	7 days
Server connection change frequency	Once in 100 days

We set evaluation parameters to calculate the failure localization time based on the server failure detection rate measured in the simulation attempts. The failure localization time is composed of the time that the monitoring system takes for detecting a server failure by ICMP Echo (ping) and the time that a network administrator takes for identifying the network side or server side as the cause of failure by using the CC or LB functions of Ethernet OAM. The server failure detection rate is defined as the percentage of servers that were flagged to be monitored by using CC of Ethernet OAM with MEP among the servers that actually failed during the simulation. The parameters used to calculate the failure localization time according to the server failure detection rate are shown in Table 3.3.

3.4.2 Port Monitoring Priority

To prioritize the above-mentioned monitored ports, the proposed RFM method calculates the port monitoring priority for each port of the switch in the network based on the total failure rate. The port monitoring priority p is expressed as

$$p = \lambda(t) = 1 - \{1 - \lambda_{\text{wear}}(t)\} \{1 - \lambda_{\text{init}}(t)\} \{1 - \lambda_{\text{chg}}(t)\}, \tag{3.1}$$

Table 3.3: Evaluation Parameters for Failure Localization Time

Parameter	Value
Ping interval	15 minutes
Ethernet OAM CC interval	1 minute
Ethernet OAM LB execution	5 minutes

where $\lambda(t)$, $\lambda_{\text{wear}}(t)$, $\lambda_{\text{init}}(t)$, and $\lambda_{\text{chg}}(t)$ represent the total failure rate of a server, the failure rate of wear-out failure of a server, the failure rate of initial failure of a server, and the failure rate of a server due to changes of configuration, respectively.

The method assumes the server wear-out failure rate $\lambda_{\text{wear}}(t)$ as

$$\lambda_{\text{wear}}(t) = \begin{cases} 1 - \exp\left(-a(t - t_1)\right) & \text{for } t \geq t_1, \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where a and t_1 represent the speed of wear-out and the time when the server has been deployed, respectively. Since the components of a server wear out as time goes on, the failure rate increases as well.

The method then calculates the server initial failure rate $\lambda_{\text{init}}(t)$ from the uptime of the server. This $\lambda_{\text{init}}(t)$ is assumed as

$$\lambda_{\text{init}}(t) = \frac{W_{\text{init}}}{\sqrt{2\pi}} \exp\left(-\frac{(t - t_1)^2}{2\sigma_{\text{init}}^2}\right), \tag{3.3}$$

where W_{init} indicates the amplitude of the initial failure used to estimate the server failure rate and σ_{init} indicates the speed of stability of the server at the initial failure. With such an expression, the method can express whether the service failure occurred due to a configuration error for a certain time since the server had been deployed or changed or since a virtual machine had been deployed on the server. Since the initial failure disappears as time goes on, the failure rate decreases as well.

Also, regarding the server change-derived failure rate $\lambda_{\text{chg}}(t)$, in order to select the port for activating the Ethernet OAM MEP, this method considers failures due to the changes in the servers and VMs. For example, as shown in Figure 3.5, operational events such as connecting a server to a different port, migrating a VM onto a different server, or changing the configuration parameters of a server-virtualization platform or a VM, etc. are considered. Server change-derived failure rate $\lambda_{\text{chg}}(t)$ spikes when these events occur.

The failure rate $\lambda_{\text{chg}}(t)$ is assumed as

$$\lambda_{\text{chg}}(t) = W_{\text{chg}} \left[1 - \prod_{m} \left\{ 1 - \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(t - t_m)^2}{2\sigma_{\text{chg}}^2} \right) \right\} \right], \tag{3.4}$$

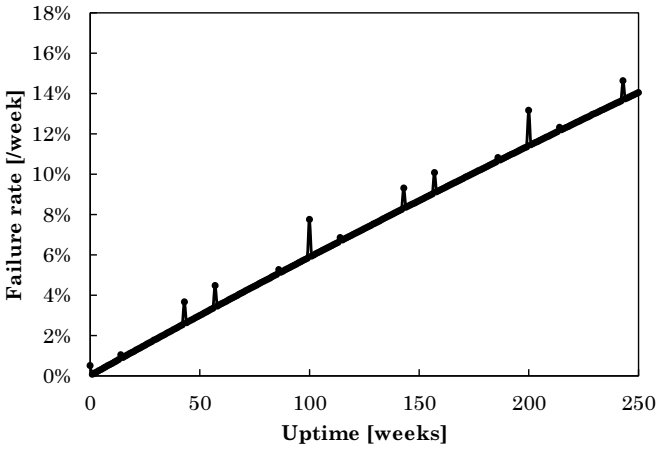
where W_{chg} indicates the amplitude of the change failure used to estimate the server failure rate, σ_{chg} indicates the speed of stability of the server at the change failure, and t_m with $m = 0, 1, 2, \ldots$ is the moment the connection of the server is changed. It combines the failure rates caused by individual connection changes. It is assumed that a failure rate caused by each connection change is represented by a normal distribution centered on $t = t_m$.

In this evaluation, we simulated two types of server reliability models—a wear-out intensive model and an initial-failure intensive model—based on the above definition of failure rate. The simulation parameters of the models are shown in the second and third columns of Table 3.4. In the initial-failure intensive model, initial failures and change failures significantly affect the health of servers, while in the wear-out intensive model, wear-out of servers has the strongest effect.

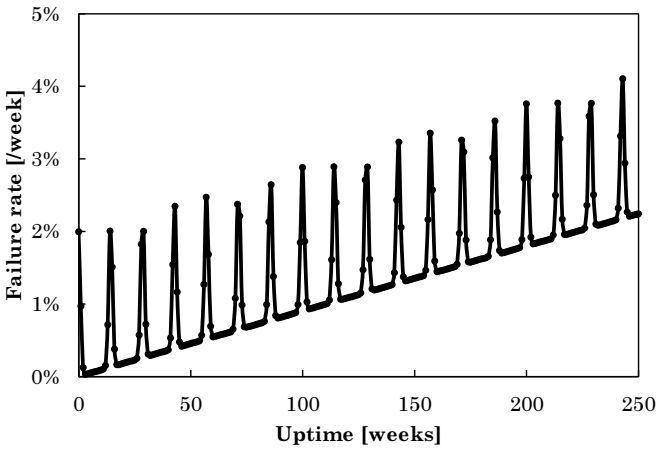
Table 3.4: Simulation Parameters of Server Failures

Parameter	Meaning	Wear-out intensive	Initial-failure intensive
a	Wear-out failure gradient	1.0×10^{-9}	6.67×10^{-11}
W_{init}	Initial failure rate amplitude	1.25×10^{-2}	5.0×10^{-2}
σ_{init}	Initial failure rate standard deviation	8.0×10^{4}	5.0×10^{5}
W_{chg}	Change failure rate amplitude	5.0×10^{-2}	5.0×10^{-2}
σ_{chg}	Change failure rate standard deviation	8.0×10^{4}	5.0×10^{5}

In the simulation, the status of a VM is treated as a random variable that can be running or failed. The probability of the failed status is the time-varying failure rate $\lambda(t)$. The status of each VM is thus evaluated at each simulation step according to the failure rate $\lambda(t)$. Figure 3.6 shows graphs of the calculated failure rates for five years for a VM in the initial-failure intensive model and a VM in the wear-out intensive model. The failure rates are calculated in accordance with Eqs. 3.1–3.4 and the simulation parameters in Table 3.4.



(a) Wear-Out Intensive Case



(b) Initial-Failure Intensive Case

Figure 3.6: Server Failure Rate of Simulated VMs

3.4.3 Evaluation Results

The simulation results of the number of server failures are shown in Table 3.5. We can see that as the server change rate increased from 10% to 50%, the number of total failures was reduced from 1357.65 to 213.47 in the wear-out intensive case. The average running time of the server was shortened as the server change rate increased. The server failures decreased since the servers were replaced before server failure occurred due to the components of the servers wearing out. In contrast, in the initial-failure intensive case,

the server change rate did not have a large effect on the number of failures.

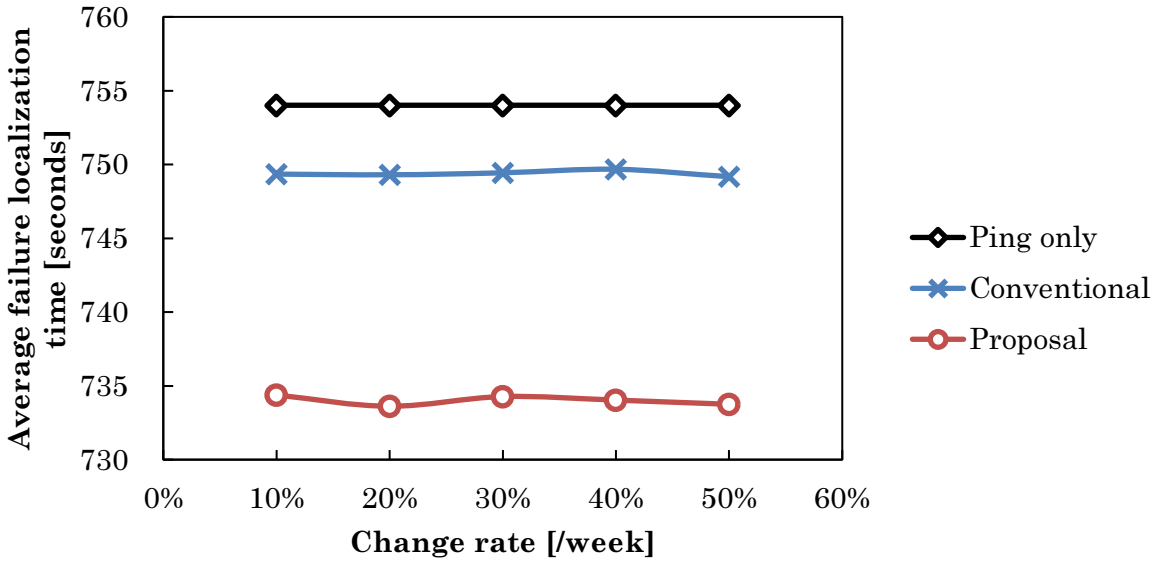
Table 3.5: The Number of Simulated Server Failures

Server change rate [/week]	Wear-Out Intensive	Initial-Failure Intensive
10%	1357.65	1149.38
20%	700.13	1152.33
30%	444.9	1168.65
40%	307.11	1139.02
50%	213.47	1067.61

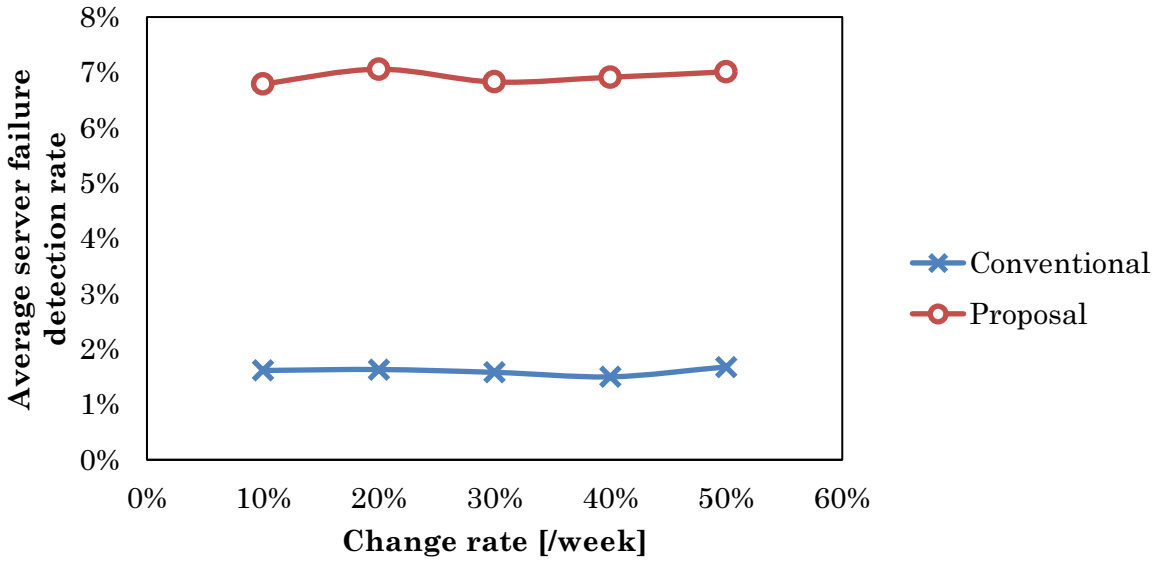
Figures 3.7 and 3.8 show the simulation results of the failure localization time and the server failure detection rate by the monitoring system. The per server change rate, the number of failures, the server failure detection rate in the conventional method, and the server failure detection rate in the proposed RFM method are shown. The MEP number was set to 16, and the server change rate was 10% per week to 50% per week every 10%. The average value of the failure localization time was calculated by executing the simulation 100 times for each of the five server change rates.

In the wear-out intensive case, the results of the average failure localization time for server change rates from 10% to 50% in the wear-out intensive case were 754 seconds for the ICMP Echo (ping)-only method, 749 seconds for the conventional method, and 734 seconds for the proposed RFM method, as shown in Figure 3.7(a). The change in server change rate did not affect the average failure localization time or the average server failure detection rate. The proposed RFM method reduced the average failure localization time by 2.1% and 2.7% compared to the conventional method and ICMP Echo (ping)-only method, respectively. Also, the proposed RFM method improved the server failure detection rate by 5.32 points on average compared to the conventional method, as shown in Figure 3.7(b). These results demonstrate that the proposed RFM method can increase the average server failure detection rate in data centers and reduce the average failure localization time regardless of the server change rate.

As for the initial-failure intensive case, the failure localization times and the server failure detection rates are shown in Figures 3.8(a) and 3.8(b), respectively. When the server change rate increased from 10% to 50%, the average failure localization time of the proposed RFM method increased and got closer to the result of the conventional method, since the server failure detection rate decreased from 4.1% to 2.3%.

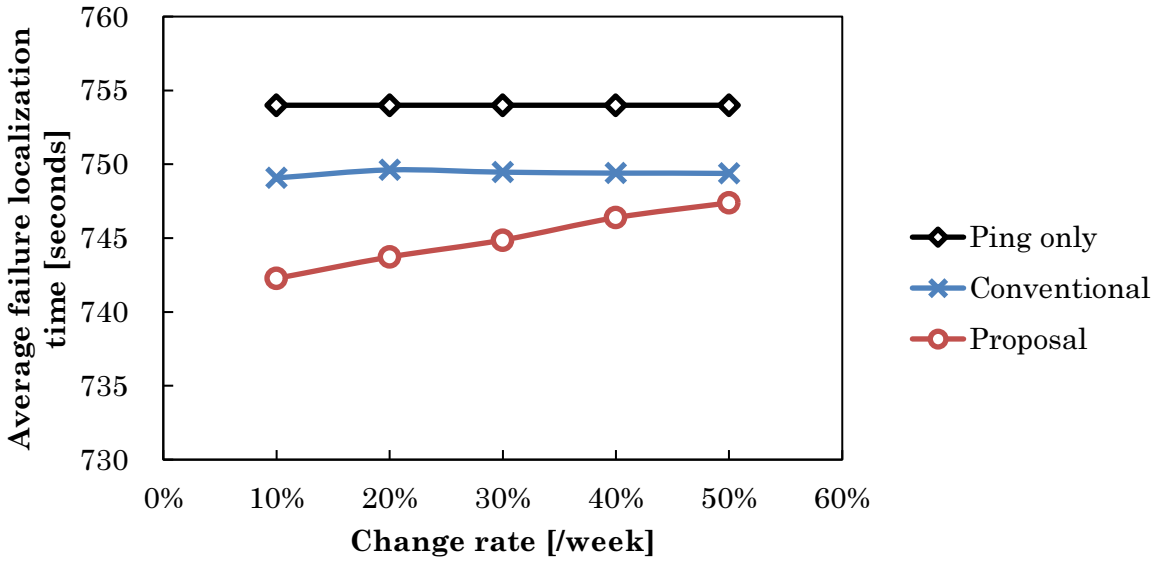


(a) Average Failure Localization Time

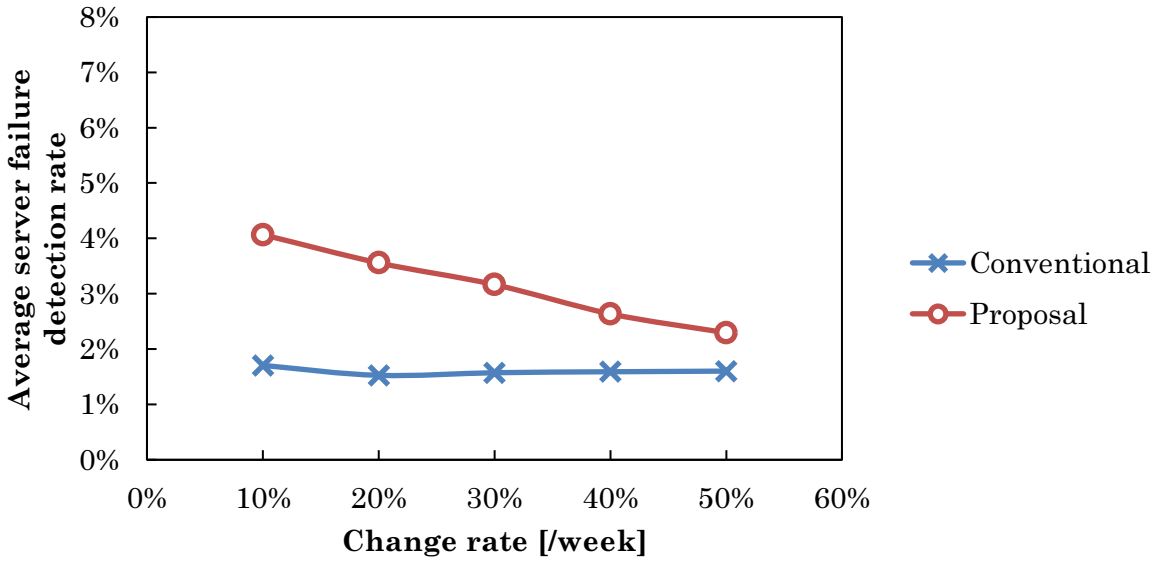


(b) Average Server Failure Detection Rate

Figure 3.7: Evaluation Results in 16 MEPs and Wear-Out Intensive Case



(a) Average Failure Localization Time



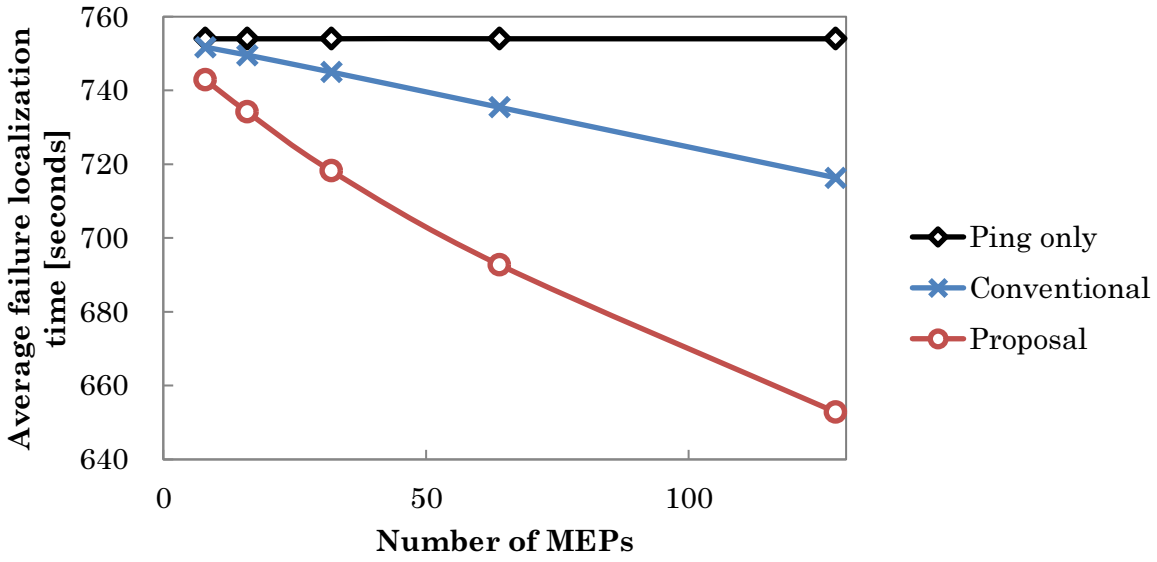
(b) Average Server Failure Detection Rate

Figure 3.8: Evaluation Results in 16 MEPs and Initial-Failure Intensive Case

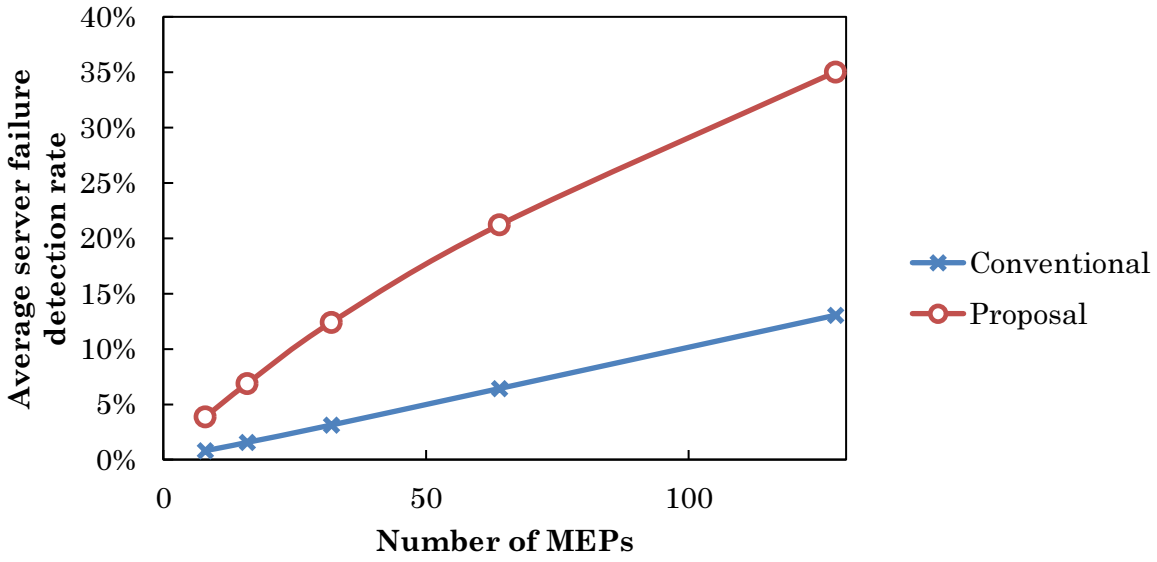
Next, we evaluated the effect on the average failure localization time by the number of MEPs. The results are shown in Figures 3.9 and 3.10. In this evaluation, the server change rate was fixed at 20% per week, as this is a typical rate in data centers. We performed the simulation 100 times for 8, 16, 32, 64, and 128 MEPs. In each case, the average failure localization time was measured.

The results of this measurement in the wear-out intensive case are shown in Figure 3.9. An average of about 702 server failures occurred. When 128 MEPs were used to monitor servers, the average failure localization time was shortened from 754 seconds to 716 seconds when the server was monitored by Ethernet OAM in the conventional method. When servers were monitored with Ethernet OAM in the proposed RFM method, the average failure localization time was shortened from 716 seconds to 653 seconds. The proposed RFM method reduced the average failure localization time by 8.8% and 13% compared to the conventional method and ICMP Echo (ping)-only method, respectively. Also, the average server failure detection rate was improved from 13% with the conventional method to 35% with the proposed RFM method. More than a third of all failures were detected with MEPs numbering only a tenth of that used in typical data centers.

The evaluation results in the initial-failure intensive case are shown in Figure 3.10. Here we can see almost the same trends as in the wear-out intensive case. The average failure localization time was shortened from 717 seconds with the conventional method to 670 seconds with the proposed method. Also, the average failure detection rate was improved from 13% with the conventional method to 29%.

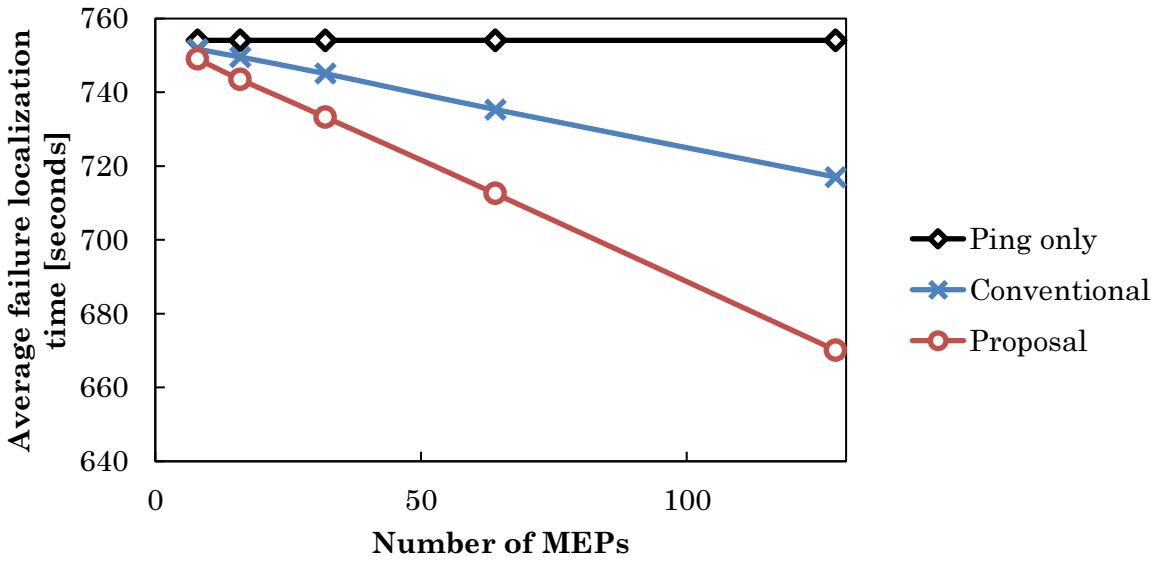


(a) Average Failure Localization Time

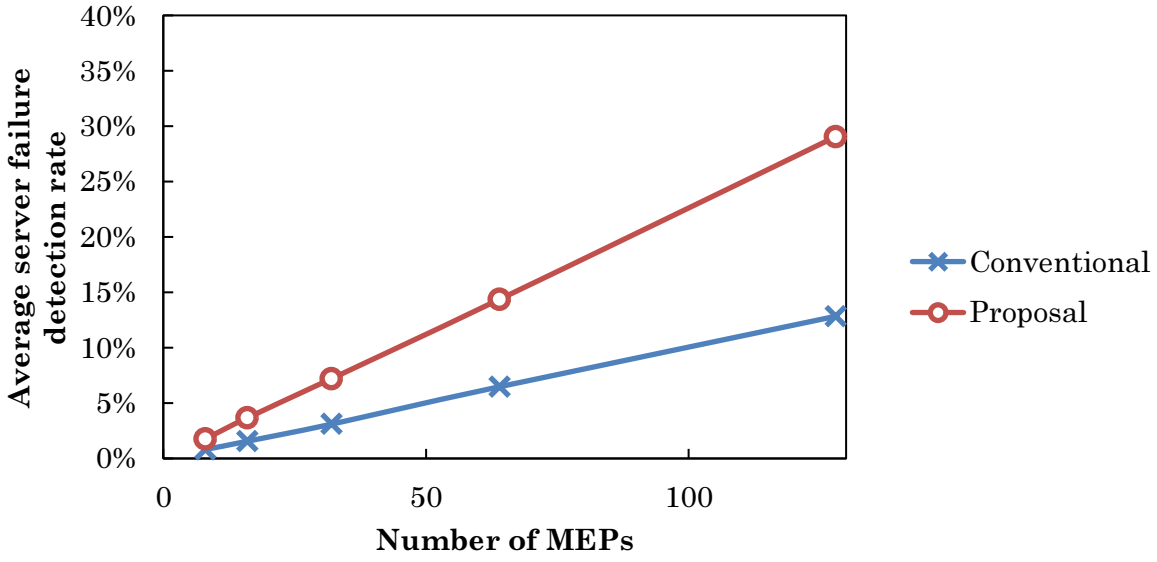


(b) Average Server Failure Detection Rate

Figure 3.9: Evaluation Results in 20% server-change rate and Wear-Out Intensive Case



(a) Average Failure Localization Time



(b) Average Server Failure Detection Rate

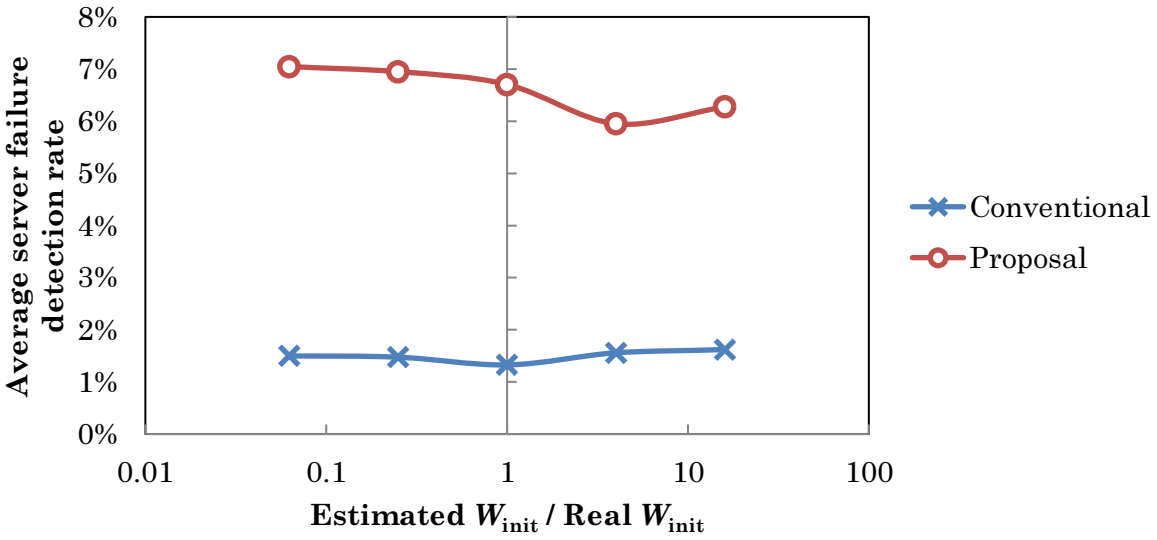
Figure 3.10: Evaluation Results in 20% server-change rate and Initial-Failure Intensive Case

3.4.4 Sensitivity Analysis

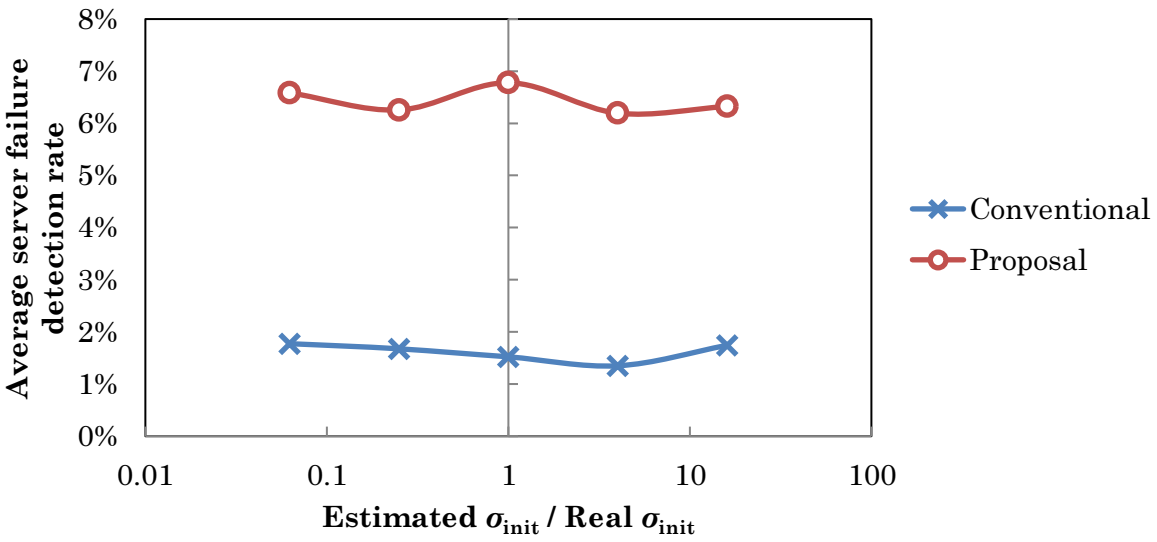
For the evaluation with the wear-out intensive model and 16 MEPs, the effect on the server failure detection rate is shown in Figure 3.11(a), where the parameter W_{init} is different from the actual value. In addition, Figure 3.11(b) shows the effect of the parameter σ_{init}, which represents the deviation in the time axis of the initial failure when the change failure used to estimate the server failure rate is different from the actual value. We found that even if the estimated value of the variance σ_{init} or amplitude W_{init} was shifted from the actual value, there was no significant change in the evaluation result of the server failure detection rate for either the conventional method or the proposed RFM method.

For the evaluation with the initial-failure intensive model and 16 MEPs, the effect of the parameter W_{init} on the magnitude of the initial failure when the amplitude of the change failure is different from the actual value used to estimate the server failure rate is shown in Figure 3.12(a). In addition, the effect of the parameter σ_{init} on the size of the variance of the time axis of the initial failure when the change failure used to estimate the server failure rate is different from the actual value is shown in Figure 3.12(b). In the case of the proposed RFM method, the server failure detection rate changed significantly when the deviation σ_{init} was shifted. Specifically, when the ratio of the estimated value of σ_{init} to the actual deviation σ_{init} was 0.5 or less, the server failure detection rate was almost unchanged from the conventional method. On the other hand, when the ratio of deviation σ_{init} was 1 or more, the proposed RFM method exhibited a higher server failure detection rate of about 2.5 points compared to the conventional method.

The results of the evaluation with the wear-out intensive model and 128 MEPs are shown in Figure 3.13, and the results of the evaluation with the initial-failure intensive model and 128 MEPs are shown in Figure 3.14. As with the results of the evaluation with 16 MEPs, even if the estimated value of the variance σ_{init} or amplitude W_{init} was shifted from the actual value, there was no significant change in the server failure detection rate Also, the proposed RFM method exhibited a higher server failure detection rate when the ratio of deviation σ_{init} was 1 or more in the initial-failure intensive case. These results demonstrate that the number of MEPs does not have an effect on the sensitivity of the server failure detection rate.

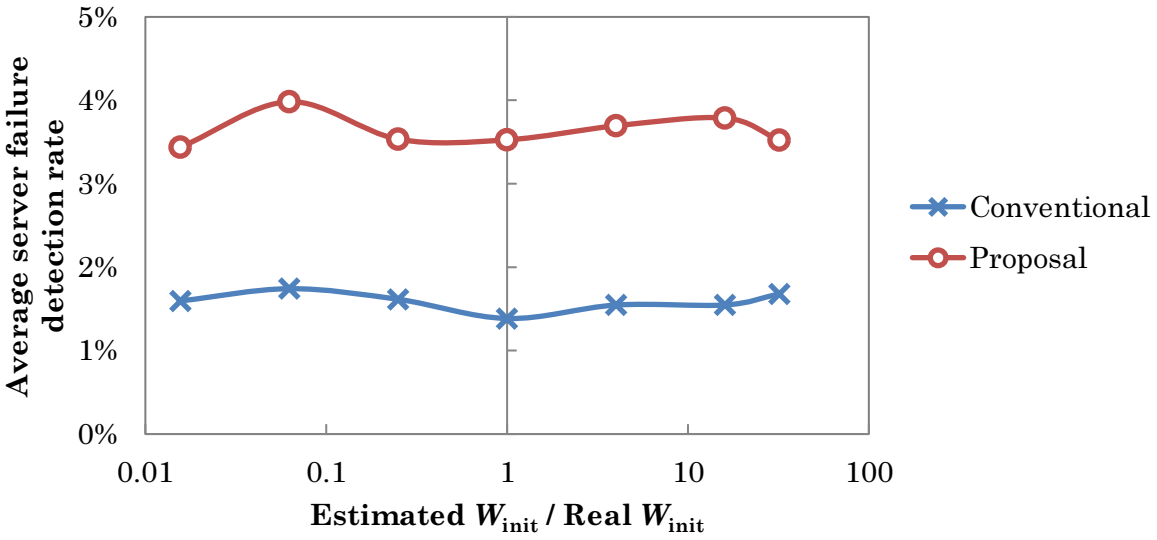


(a) Analysis Results for Amplitude W_{init} of Failure Rate

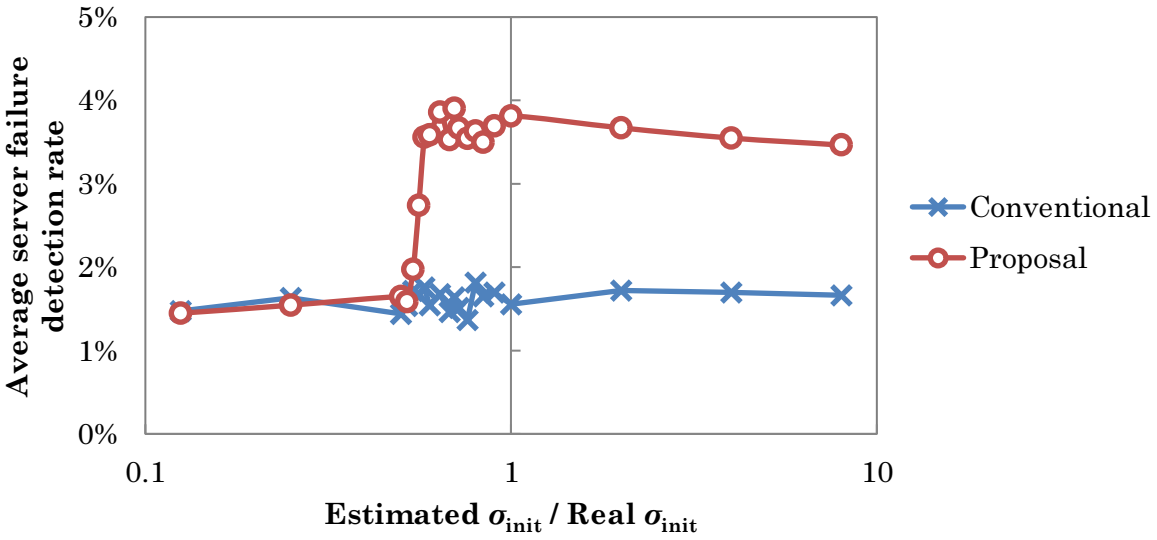


(b) Analysis Results for Standard Deviation σ_{init} of Failure Rate

Figure 3.11: Sensitivity Analysis in 16 MEPs, 20% server-change rate and Wear-Out Intensive Case

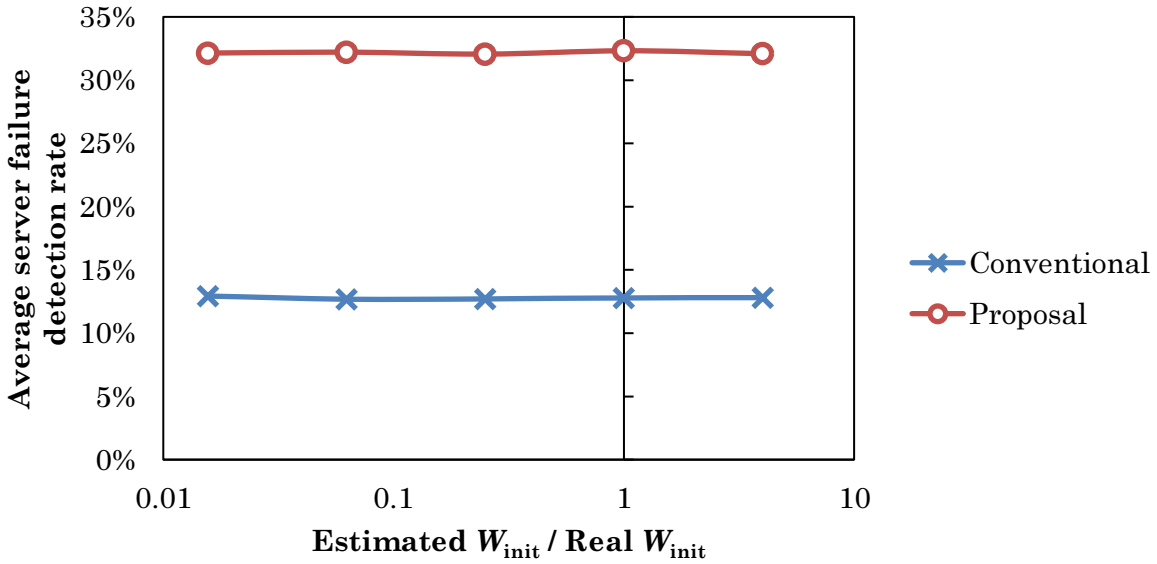


(a) Analysis Results for Amplitude W_{init} of Failure Rate

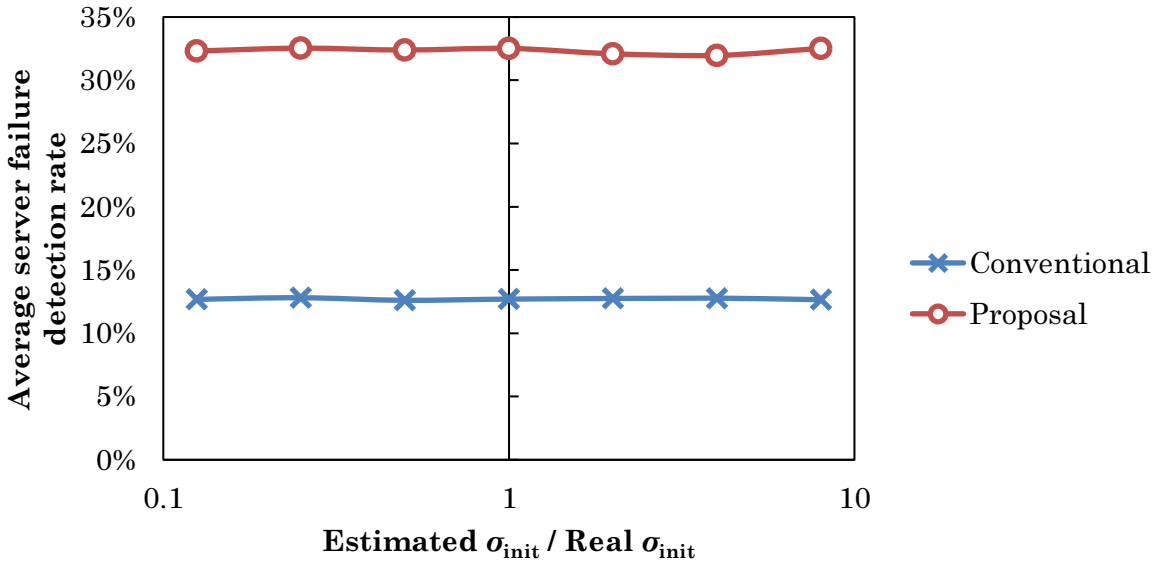


(b) Analysis Results for Standard Deviation σ_{init} of Failure Rate

Figure 3.12: Sensitivity Analysis in 16 MEPs, 20% server-change rate and Initial-Failure Intensive Case

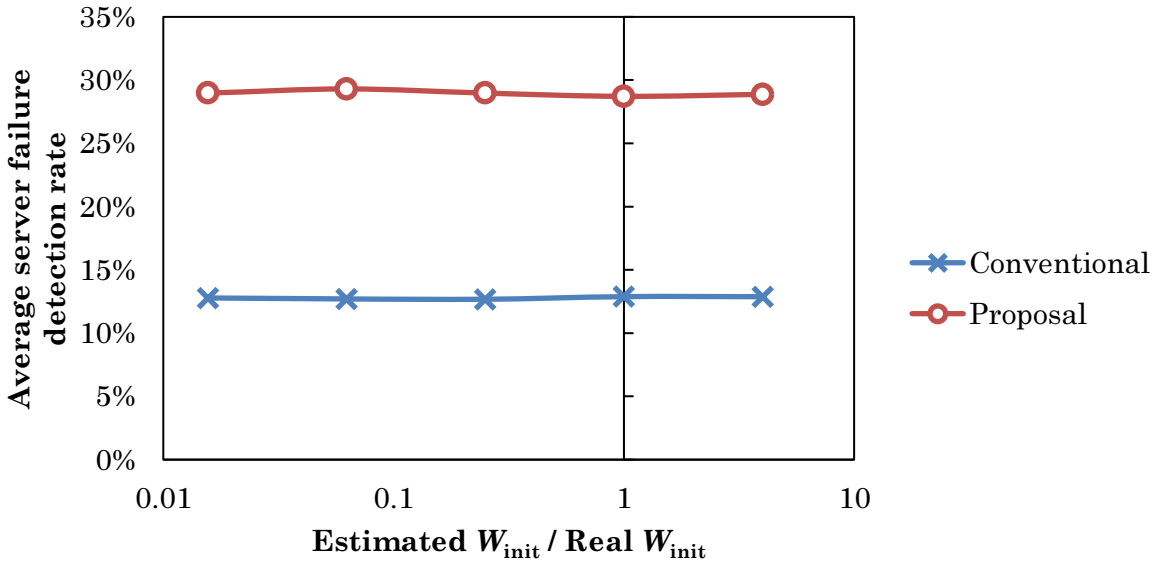


(a) Analysis Results for Amplitude W_{init} of Failure Rate

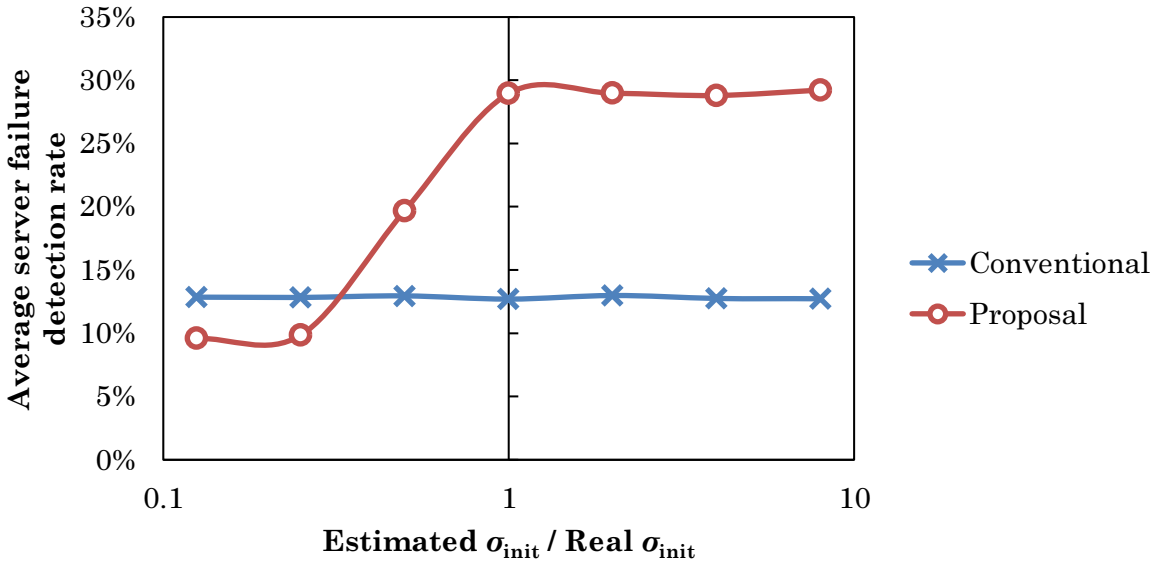


(b) Analysis Results for Standard Deviation σ_{init} of Failure Rate

Figure 3.13: Sensitivity Analysis in 128 MEPs, 20% server-change rate and Wear-Out Intensive Case



(a) Analysis Results for Amplitude W_{init} of Failure Rate



(b) Analysis Results for Standard Deviation σ_{init} of Failure Rate

Figure 3.14: Sensitivity Analysis in 128 MEPs, 20% server-change rate and Initial-Failure Intensive Case

3.4.5 Discussion

Since the server failure detection rate varies according to the number of MEPs, the number of MEPs should be determined according to the target of the server failure detection rate or the failure localization time of the data center. Let us additionally consider the results of the above sensitivity analysis for the case when the initial failure and change failure are dominant. If the ratio of the deviation σ_{init} of the estimated server failure rate to that of the actual server failure rate is less than 0.6, the server failure detection rate of the proposed RFM method decreases to the same level as that of the conventional method. This is likely explained by the fact that if the actual value of the variance is greater than the estimate, it is more likely that a server that is not set as a monitored target by an MEP will fail. Presumably, it is the same when the targets monitored by MEPs are selected randomly. To avoid this problem, it is necessary to accurately calculate the deviation σ_{init} of the actual server failure rate in advance based on the failure history of servers in that data center in the past.

3.5 Related Work

To the best of our knowledge, no prior research has combined network-based monitoring methods (such as Ethernet OAM) with service-level monitoring methods (such as ping) to localize the cause of failure between servers and networks in a data center within a short time. However, from a broader perspective, network failure management has long been studied for the large-scale data centers of commercial server-hosting services and cloud computing services. Among these, there have been studies on how to determine indirectly whether a failure cause exists on the server side or network side when a service-level failure occurs.

One major approach is to use an active monitoring method for examining the state of the network from the outside. The Sherlock system[108] lists candidate factors that cause performance problems according to the dependencies between the components of the data center at the service level. However, the Sherlock system does not collect information from the monitored Web servers or network devices. It collects the results of Web server responses to performance monitoring requests from agents deployed on Web clients. It infers the root cause of the failure from these monitoring results and from the network topology. The time taken by the Sherlock system to localize a cause of failure increases in proportion to the number of nodes in the inference graph used by the system to infer the failure cause. It takes about 24 minutes for the Sherlock system to localize the failure cause

when 500,000 nodes exist in the graph for monitoring 2,300 clients and 70 servers[108]. Although a precise comparison is difficult, it appears that the Sherlock system takes longer than ours in terms of the average failure localization time when using the RFM method.

Another active monitoring method for data-center networks[109] combines the results of active-monitoring methods (such as ping) with the results of passive monitoring methods (such as alerts from network devices and network links) to localize failure causes. A server running a monitoring agent sends a ping every minute to randomly selected IP end hosts in the cluster, among clusters and among data centers. When the above active monitoring and passive monitoring combination method was applied to 73 network incidents in commercial data centers, 50% of the network incidents were localized within 12 minutes and within 23 minutes at maximum. This is longer than the average failure localization time when using our RFM method with more than 64 MEPs.

In addition, a ping-based active measurement system called Pingmesh was developed for large-scale and commercial data centers[69]. It measures the packet drop rate and latency in the data center network by a ping agent on servers that periodically send ping messages to each other. It then analyzes the failure causes in the network according to the measurement results. Network team developers, engineers, and customers use its visualization portal to make sure they are not experiencing any network problems. However, Pingmesh monitors the data center network hierarchically. Specifically, it monitors the complete graph of all servers under a Top-of-Rack (ToR) switch, a complete graph of all ToR switches at the intra-data center level, and a complete graph of all data centers at inter-data center level, and then determines which network tier has suffered a failure. As operators have to use traceroute to check the availability of the corresponding route for the failed service after determining the tier, it takes a longer time than the RFM method proposed in our research.

Compared to the above studies that use ping, which is an active monitoring method of servers, the constraints of the RFM method are less tolerant to network device failures because it relies on Ethernet OAM, which is a function of network switches. When a network device fails and its Ethernet OAM CC function does not work, it can take a long time for the server and network administrators to recognize that the root cause of the failure is on the network side. This means we should monitor the status of the Ethernet OAM CC functions of the selected MEP-enabled network switches from the network management system to deal with the constraint.

3.6 Conclusion

We proposed a new network monitoring method called RFM that coordinates server management and network management. This dynamically prioritized failure management method dynamically changes the server to be monitored with Ethernet OAM according to the server failure rate estimated based on the uptime and configuration changes of the server. Simulation results showed that more than a third of all failures were detected with MEPs numbering only a tenth of that normally seen in data centers. We conclude that coordinating the server management and network management enables more efficient failure management in large-scale data centers.

As a future challenge for data-center networks, we will need to further improve the failure management in order to keep up with the increasing number of services provided by data centers. In addition to simple layer-2 or layer-3 networking services, data centers are now providing mobile networking services. For example, in a mobile network such as an Evolved Packet Core (EPC) network composed of a variety of devices, the root cause of a failure will need to be rapidly localized on the device level when it occurs[110, 111]. By identifying the problematic device, network administrators can fix or replace it promptly when a failure occurs.

Chapter 4

Failure Management with Dynamically Prioritized Monitoring according to Uptime of Virtual Machines

4.1 Introduction

This chapter extends the RFM method, the network failure management method proposed in Chapter 3, so that it can be more easily applied to data centers. This extended method uses the cumulative uptime of VMs in the data center as an approximate reliability model.

To widely apply the dynamic monitoring method for data-center networks, it is necessary to find out how to effectively use the RFM method even in data centers where only a low-precision reliability model of VMs can be obtained. The key point is that it should be easy to create the reliability models.

The difficulty is that it takes a certain amount of time to create a reliability model; if one is created too quickly, it will be inaccurate. In addition, new models of IT products such as servers are frequently being released, which means that reliability models based on failure data of similar products in the past will be less accurate.

Therefore, to widely apply the dynamic monitoring method, it is necessary to be able to create the reliability models of VMs in a simple way. The easiest way to do this is to create an approximate reliability model based only on information that is readily available at the data center. We also need to consider how much the failure detection rate decreases when we use the approximate reliability model instead of the reliability model

utilized in the RFM method.

As the contribution of this section, we clarify that failures of data-center networks can be detected without significantly degrading the accuracy of failure detection even if we use an approximate reliability model. We propose an approximate reliability model based on the VM uptime, which is easily collected from configuration management information without having to rely on detailed statistical information about daily failures in the data center.

Section 4.2 proposes an efficient virtual network monitoring system based on the uptime of physical servers and VMs in the data center. Section 4.3 describes an example procedure to select the monitored port in a data-center network using Ethernet OAM CC with the proposed method. Section 4.4 evaluates the effectiveness of the proposed method by simulation. We provide the conclusion in Section 4.5.

4.2 Selection of Monitored Ports based on Uptime of Virtual Machines

As described above, the RFM method has a problem in that its applicable range is limited to data centers that have easy access to a reliability model of VMs. To solve this problem, we propose uptime-based failure management (UFM), a new method to select which VMs to monitor with Ethernet OAM based on the cumulative uptime of VMs in the data center.

We aim to be able to select the network endpoints monitored with Ethernet OAM without having to rely on the data center's failure statistics. To achieve this, we take a new approach that allows the monitoring target VM to be selected without creating a reliability model of VMs by determining the monitoring priority per VM from only the configuration information of the data center. Specifically, the cumulative uptime of each VM is used to determine the monitoring priority.

As shown in Figure 3.3, the operations of the server administrator and the network administrator are the same in both the reliability model-based method and the newly proposed method. Namely, the server administrator monitors the availability of all servers in the data center at a low frequency with ICMP Echo (ping), and the network administrator monitors the connectivity of a limited number of selected routes in the data-center network at a high frequency with Ethernet OAM CC. When a service failure occurs, the server administrator retrieves the VM corresponding to the failed service, and the network administrator checks the connectivity between the port that connects the retrieved VM and the core switch of the data center network with Ethernet OAM CC. If Ethernet OAM

CC provides the notification of the connectivity error, network switches are the root cause of the service failure; otherwise, the root cause is assumed to stem from the server side.

The left side of Figure 4.1 shows the procedure of the RFM method proposed in Chapter 3. In this method, the network management system monitors the multiple types of failure in an integrated manner. Specifically, it calculates the monitoring priority for each VM based on the reliability model of the VMs and then selects which ports to be monitored with Ethernet OAM CC from those with higher monitoring priorities. The monitoring priorities and monitored ports are updated periodically.

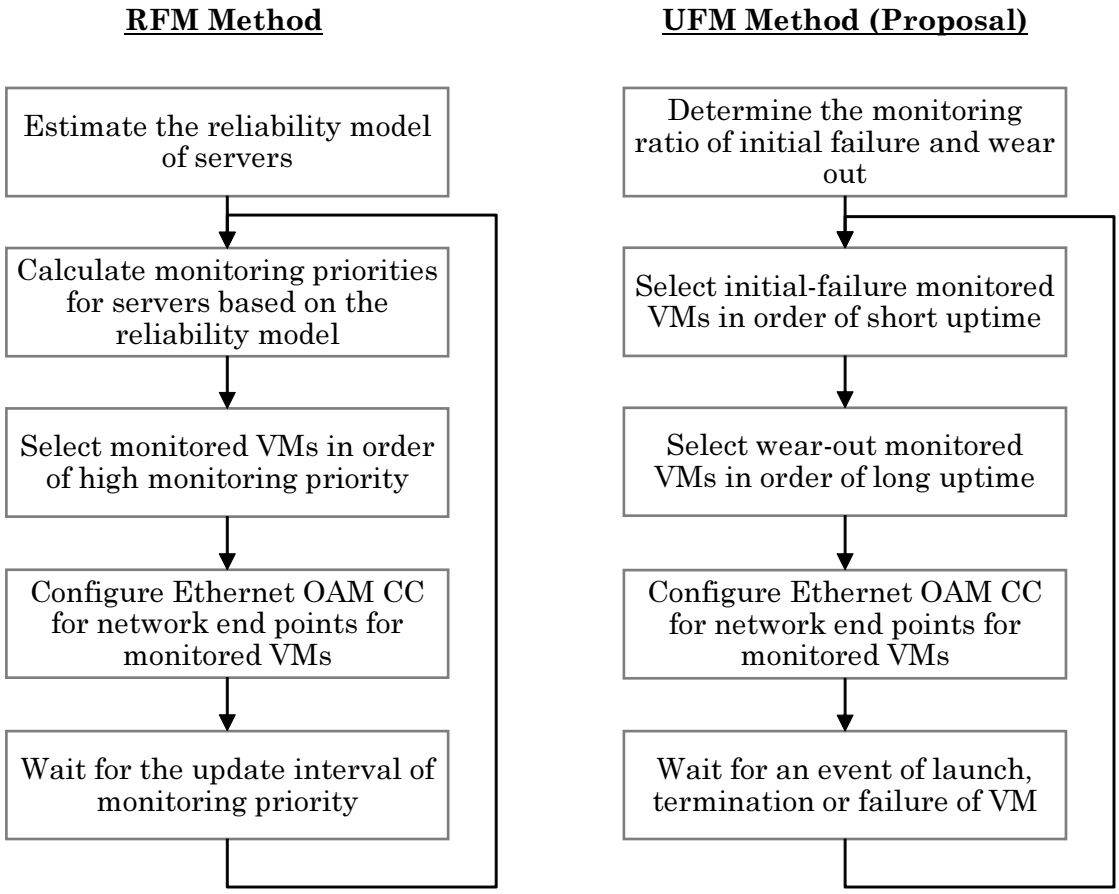


Figure 4.1: Procedure to Select Network End Points Monitored with Ethernet OAM

The procedures of our new method to select monitored ports with Ethernet OAM CC are shown on the right side of Figure 4.1. The network management system can select the monitored VMs even if a network operator does not manage the failure statistics in the data center. Also, initial failure and wear-out are monitored independently, and change

failure is not monitored. As the initial step of the proposed method, monitoring resources of Ethernet OAM in the data-center network are allocated to monitor $N_{\text{wear}} = N - N_{\text{init}}$ ports for the initial failure of VMs and the rest are allocated to the N_{init} network and the points to monitor wear-out of VMs, as shown in Figure 4.2. For example, in a network where 16 ports can be monitored at the same time with Ethernet OAM, eight ports are monitored for initial failure and eight ports are monitored for wear-out. The network management system selects the N_{init} VMs from all VMs as targets to monitor for initial failure with Ethernet OAM in order of short uptime, and it selects the N_{wear} VMs from all VMs as targets to monitor for wear-out in order of long uptime.

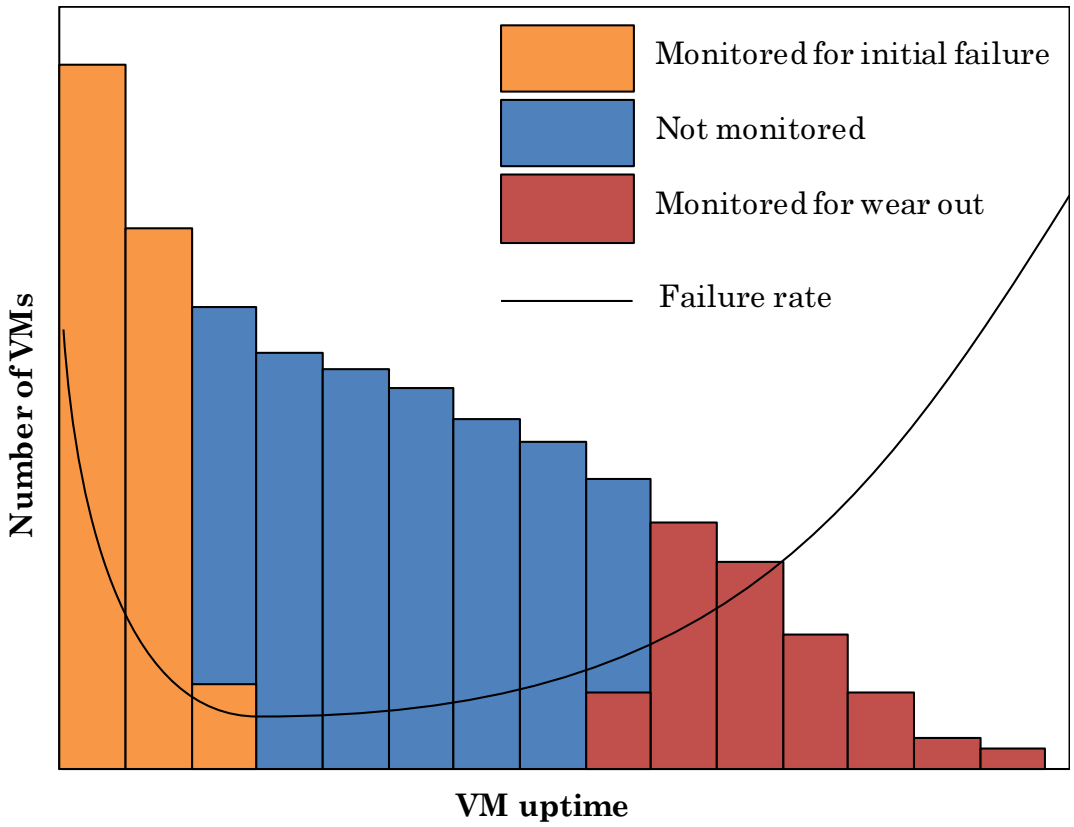


Figure 4.2: Selection of Monitored Targets based on VM Uptime

A simple way to determine the ratio of N_{wear} and N_{init} at the initial deployment of the UFM method is to allocate the same number of MEPs for initial failure monitoring and for wear-out monitoring. As the failures in a data center increase, the network operator can adjust the ratio of N_{wear} and N_{init} in accordance with the ratio of the initial failures

and wear-out that occur.

The system independently selects the ports to be monitored for initial failure and for wear-out based only on the uptime of the VM. It then changes the monitoring target only when an event occurs, rather than changing it periodically. Specifically, the system updates the monitoring targets at the time of administrative or operational events such as launch, termination, or failure of a VM or a physical server. This is because the order of ports for monitoring does not change unless such events occur.

4.3 Example Procedure of Uptime-based Failure Management Method

An example of the time-series change of the monitored ports when ports monitored by Ethernet OAM are selected using the proposed method is shown in Figure 4.3. In this simple example, the number of ports monitored for initial failure and those monitored for wear-out are set to $N_{\text{init}} = 2$ and $N_{\text{wear}} = 2$, respectively.

When a VM fails while being monitored for the initial failure with Ethernet OAM, the network management system launches a new VM to replace it and sets the new one as the target of the initial failure monitoring. In the example in Figure 4.3, when VM1, which is the target of the initial failure monitoring, fails at t_1, VM2 and VM3 are subsequently set as the new targets. In addition, if a new VM is launched, the network management system adds it to the targets of initial failure monitoring and removes the VM with the longest uptime so as to keep the number of monitored VMs at N_{init}. In the example in Figure 4.3, when VM4 is started with t_2, the network management system removes VM2 from the target of initial failure monitoring. It also sets VM2 as a target of wear-out monitoring because it now has the longest uptime. In addition, when VM6 is started at t_4, the system removes VM4 from the targets of initial failure monitoring. As five VMs are running from t_4, one of them is not included as a target of either initial failure monitoring or wear-out monitoring.

If a VM monitored for wear-out is terminated by a server administrator, the network management system removes it from the targets of wear-out monitoring. It then adds the VM that has the longest uptime next to the VMs monitored for wear-out. In the example in Figure 4.3, when VM2 is terminated at t_5, the network management system adds VM3, which has the longest uptime next to VM2. Further, if the VM that is monitored for wear-out fails, the system removes it from the targets of wear-out monitoring. It then removes the VM that has the longest uptime among the target VMs of initial failure monitoring.

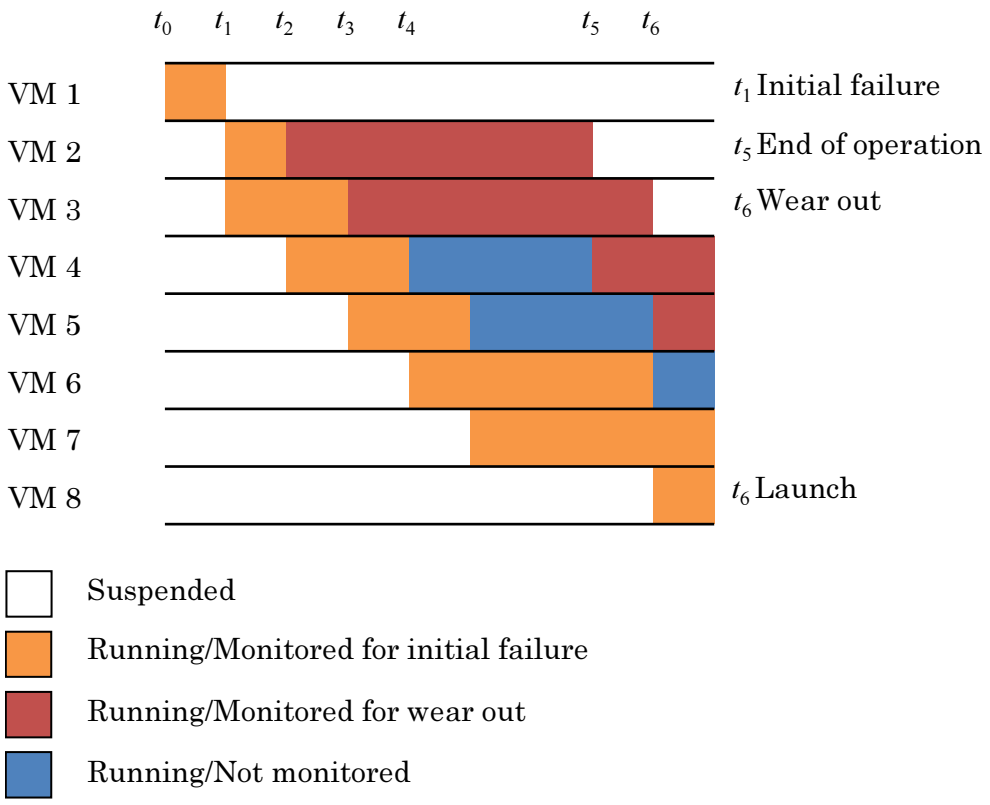


Figure 4.3: Example of Changes of Set of Monitored Virtual Machines due to Addition, Termination and Failure of Virtual machines

It also adds the newly launched VM to the target of initial failure monitoring. In the example in Figure 4.3, when wear-out failure occurs in VM3 at t_6, the system removes it from the targets of wear-out monitoring and adds VM5 in its place. In addition, the newly launched VM8 to replace VM3 is added to the target of initial failure monitoring.

4.4 Evaluation

To determine the effectiveness of the proposed UFM method, we need to know if the priority calculated based on the uptime of VMs can be used for an approximation reliability model. Therefore, we simulated VM failures in a data center and evaluated the average failure localization time and the average server failure detection rate when we used the UFM method to select the monitored networks. We compare the results with those of the conventional random and static monitoring method and the RFM method. Further,

to clarify the applicability range of the UFM method, we evaluated the average failure localization time and average server failure detection rate when changing the monitoring ratio of the initial failure and wear-out, which is a configuration parameter of the UFM method.

4.4.1 Evaluation Method

We conducted a Monte Carlo simulation of the operational states and failures of multiple VMs in a large data center. The simulator generates the object of the network node with the startup time information, generates a random number for each simulation interval to change the state of the network node to the failure state based on the reliability model of the servers, and checks whether the network node failure is detected by Ethernet OAM or not. If MEP is configured on the port of the switch connected to the failed network node, it is considered that the Ethernet OAM CC quickly identified the failure point. If MEP is not configured, it is considered that it takes a longer time to check connectivity on the network side.

The simulation parameters of the evaluation environment (e.g., number of nodes, Ethernet OAM CC interval) are the same as those shown in Tables 3.2 and 3.3 in Chapter 3. The failures of VMs are simulated according to the reliability models used in Chapter 3. The simulation parameters and characteristics of the failures of VMs are listed in Table 3.4 and Figure 3.6 in Section 3.4.2 of Chapter 3. The server change rate is fixed to 20 % per week, the same as in Section 3.4.3. The simulation was executed 100 times to calculate the mean values of the server failure detection rate and server failure localization time.

First, we evaluated the effect of increased MEPs by conducting two simulations under two conditions: one with 16 MEPs and the other with 128 MEPs. In both cases, the same numbers of MEPs are allocated for initial failure and wear-out, namely, $N_{\mathrm{init}} = 8$ and $N_{\mathrm{wear}} = 8$ in the 16-MEP case and $N_{\mathrm{init}} = 64$ and $N_{\mathrm{wear}} = 64$ in the 128-MEP case. We compare the average failure localization time and the server failure detection rate of the following four methods.

- Ping only
- Conventional method: Random allocation
- UFM method (Proposal)
- RFM method proposed in Chapter 3

Here, "Ping only" indicates the case where only ICMP Echo (ping) is used without Ethernet OAM CC. "Conventional method: Random allocation" represents the reliability model-based method that uses Ethernet OAM CC to monitor the network connectivity of some VMs. The randomly selected target is fixed. "UFM method (Proposal)" is the proposed method that dynamically changes the monitored target based on the uptime of VMs. "RFM method proposed in Chapter 3" represents the case of dynamically changing the monitoring target in the reliability model-based method to calculate the monitoring priority per VM based on the reliability model.

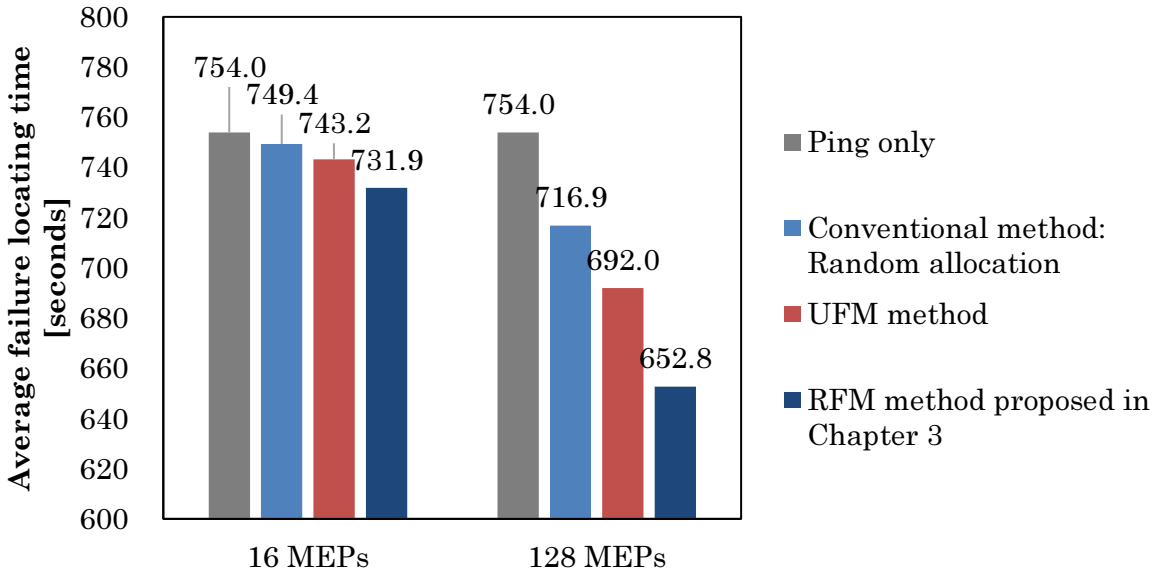
We also analyzed the sensitivity of the proposed UFM method by changing the ratio of N_{init}, the number of MEPs for initial failure, and N_{wear}, the number of MEPs for wear-out. The ratio was changed from 0:10 to 10:0 in increments of 10%. When the monitoring ratio is 0:10, all of the VMs monitored with Ethernet OAM CC are selected in order of long uptime. When the monitoring ratio is 10:0, they are selected in order of short uptime.

4.4.2 Evaluation Results of Failure Localization Time and Server Failure Detection Rate

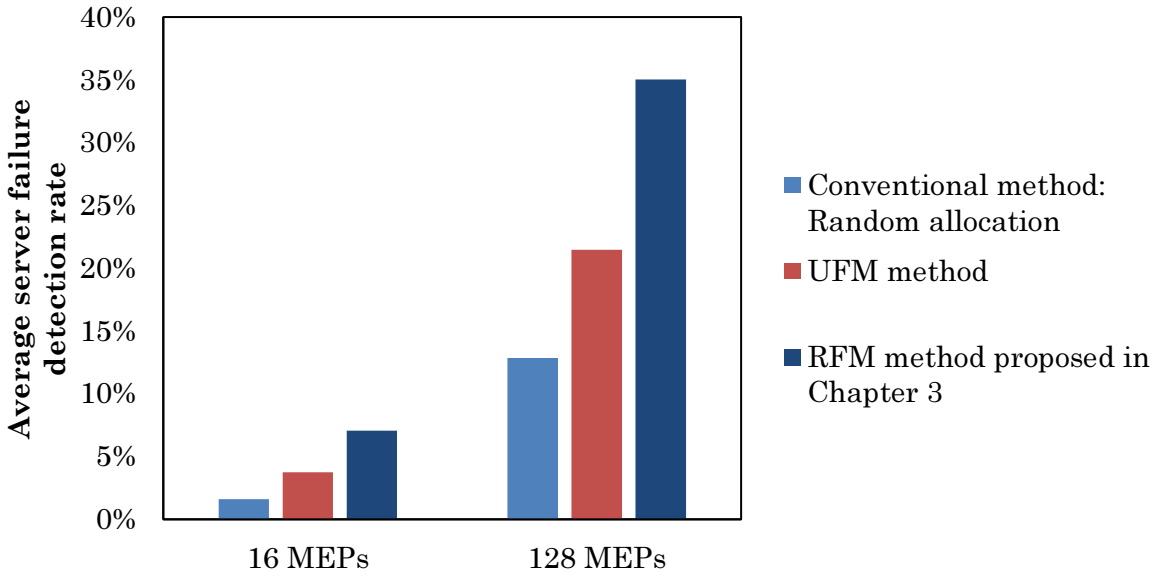
We evaluated the improvement in the failure localization time and the server failure detection rate of the proposed method. In this case, the failure localization time means the elapsed time from the moment when a service failure occurs to the moment when a server administrator and network administrator determine the root cause by checking the connectivity in the data center network with Ethernet OAM CC or Ethernet OAM LB. As for server failure detection rate, it means the ratio of the number of servers monitored with Ethernet OAM CC to the number of all servers. Therefore, the higher the failure detection rate, the shorter the failure localization time.

The simulation results of the failure localization time and server failure detection rate in the wear-out intensive case are shown in Figure 4.4.

In the 16-MEP case, the average failure localization time was 754.0 seconds for "Ping only," 749.0 seconds for "Conventional method: Random allocation," and 743.2 seconds for the proposed method. The fastest time was for "RFM method proposed in Chapter 3," which was 731.9 seconds on average. These results show that the newly proposed method reduced the average failure localization time by 1.6 % compared to "Ping only" and by 0.9 % compared to "Conventional method: Random allocation." In addition, the average failure detection rate was improved by an average of 2.1 points compared to "Conventional method: Random allocation." However, the average failure localization time increased by 1.5 % compared to "RFM method proposed in Chapter 3," and also had a 3.3 % worse



(a) Average Failure Localization Time



(b) Average Server Failure Detection Rate

Figure 4.4: Evaluation Results in Cases of 16 MEPs and 128 MEPs in Wear-Out Intensive Case

average server failure detection rate.

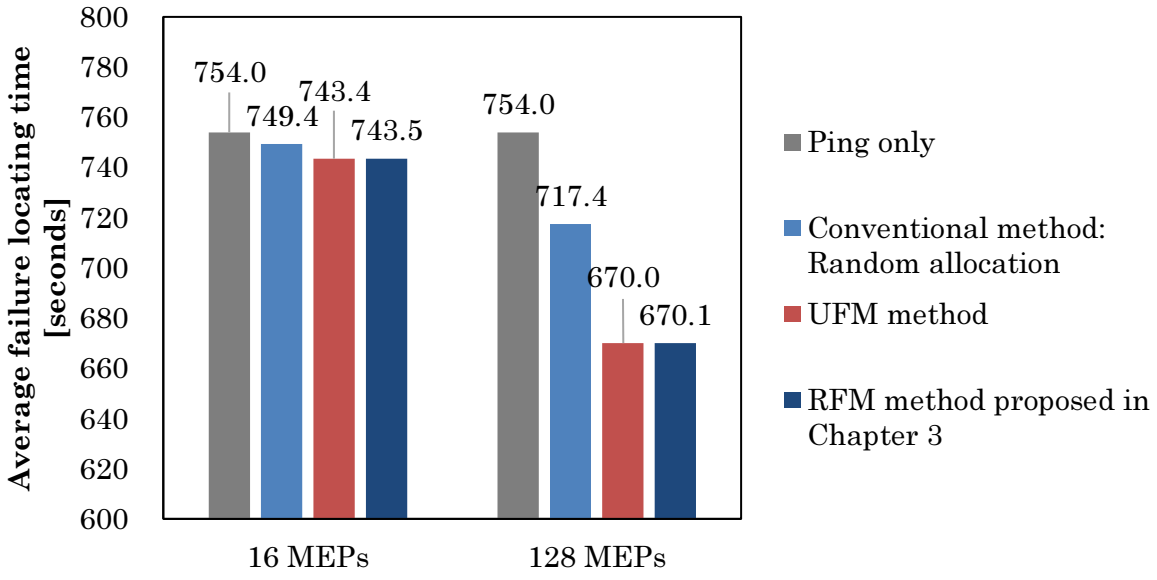
In the 128-MEP case, the average failure localization time was 754.0 seconds for "Ping only," 716.9 seconds for "Conventional method: Random allocation," and 692.0 seconds for the proposed method. "RFM method proposed in Chapter 3" was again the fastest, with an average of 652.8 seconds. These results show that the newly proposed method reduced the average failure localization time by 8.2 % compared to "Ping only" and by 3.5 % compared to "Conventional method: Random allocation." In addition, the average failure detection rate was improved by an average of 8.6 points compared to "Conventional method: Random allocation." However, the average failure localization time increased by 6.0 % compared to "RFM method proposed in Chapter 3," and also had a 13.6 % worse average server failure detection rate.

Next, the simulation results of the failure localization time and server failure detection rate of the proposed method in the initial-failure intensive case are shown in Figure 4.5.

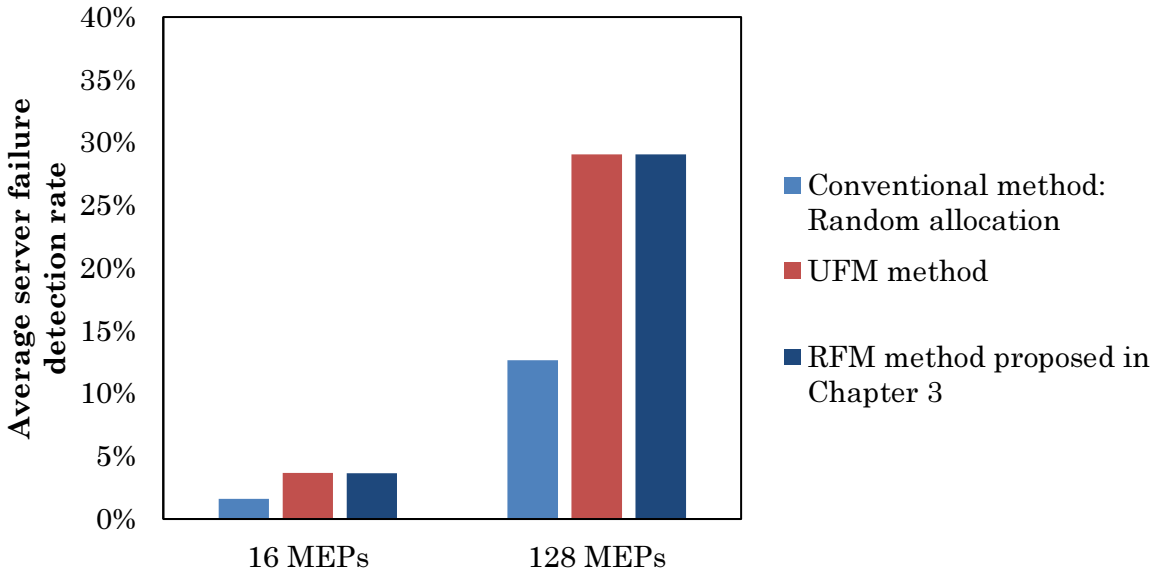
In the 16-MEP case, the average failure localization time was 754.0 seconds for "Ping only," 749.4 seconds for "Conventional method: Random allocation," and 743.4 seconds for the proposed method. The fastest time was for "RFM method proposed in Chapter 3," which was 743.5 seconds on average. These results show that the newly proposed method reduced the average failure localization time by 1.4 % compared to "Ping only" and by 0.8 % compared to "Conventional method: Random allocation." In addition, the average failure detection rate was improved by an average of 2.1 points compared to "Conventional method: Random allocation." Moreover, the average failure localization time and the average server failure detection rate were almost the same as those of "RFM method proposed in Chapter 3."

In the 128-MEP case, the average failure localization time was 754.0 seconds for "Ping only," 717.4 seconds for "Conventional method: Random allocation," and 670.0 seconds for the proposed method. "RFM method proposed in Chapter 3" was slightly faster, at an average of 670.1 seconds. These results show that the newly proposed method reduced the average failure localization time by 11.1 % compared to "Ping only" and by 6.6 % compared to "Conventional method: Random allocation." In addition, the average failure detection rate was improved by an average of 16.4 points compared to "Conventional method: Random allocation." Moreover, the average failure localization time and the average server failure detection rate were almost the same as those of "RFM method proposed in Chapter 3."

These results demonstrate that the proposed UFM method performs better than



(a) Average Failure Localization Time



(b) Average Server Failure Detection Rate

Figure 4.5: Evaluation Results in Cases of 16 MEPs and 128 MEPs in Initial-Failure Intensive Case

the conventional random allocation method even without detailed statistical information about server failures in a data center. The degradation of the average server failure detection rate of the UFM method compared to the RFM method was similar to the degradation of the conventional random allocation method compared to the UFM method in the wear-out intensive case. Further, in the initial-failure intensive case, the UFM method had a similar performance to the RFM method.

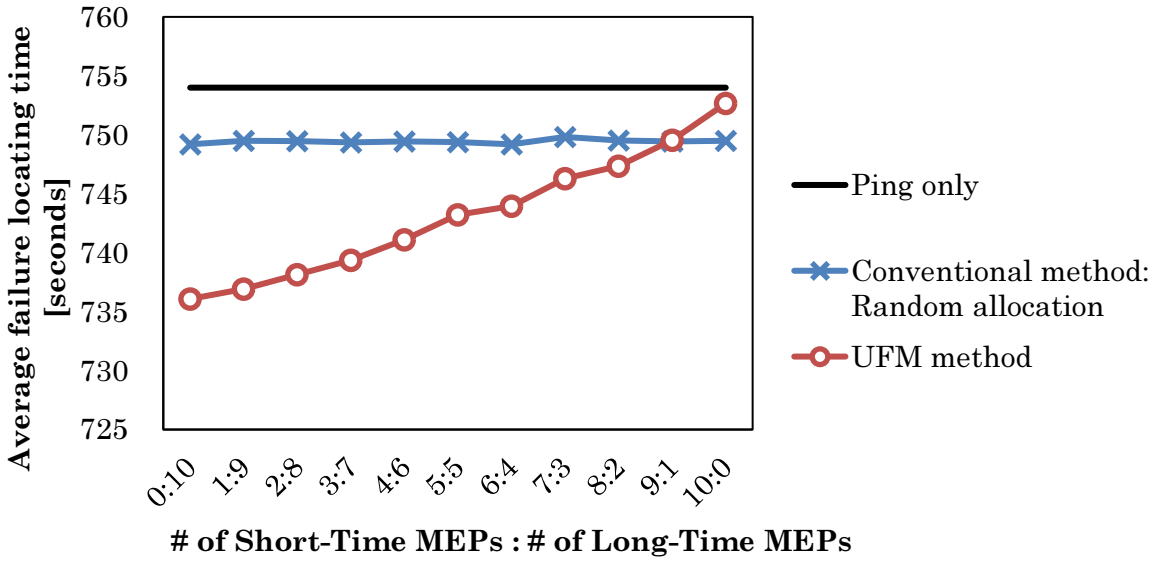
4.4.3 Evaluation Results for Monitoring Ratios of Initial Failure Monitoring and Wear Out Monitoring

Table 4.1 lists the details of four simulation scenarios with different combinations of failure models and numbers of MEPs we ran to analyze the sensitivity of the UFM method. The results of the sensitivity analysis are shown in Figures 4.6–4.9.

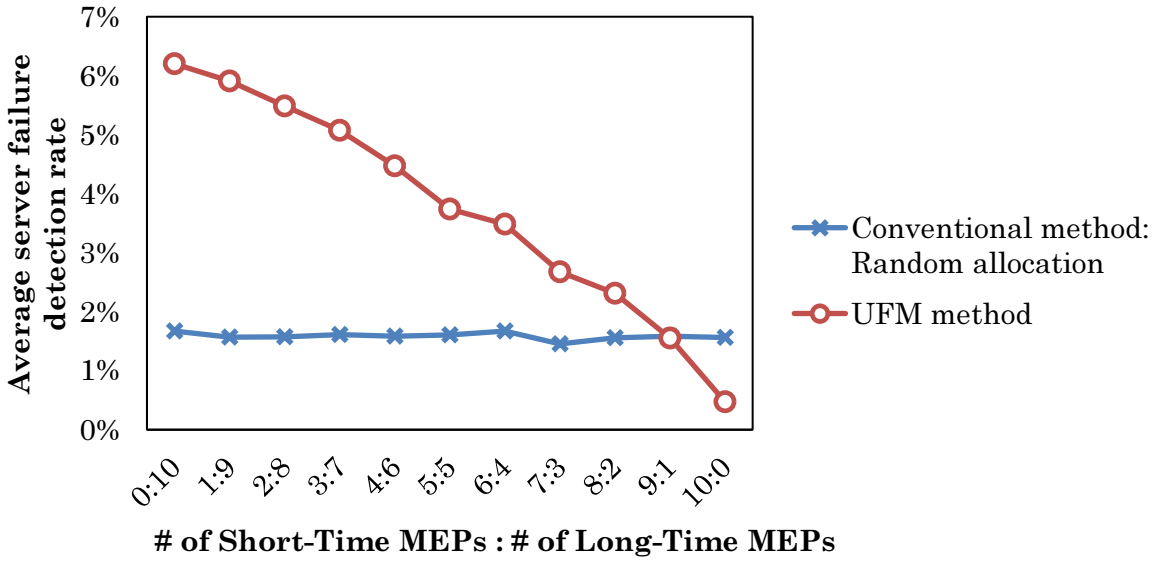
Table 4.1: Simulation Scenarios for Sensitivity Analysis

Scenario number	Failure model	Number of MEPs	Figure
1	Wear-Out Intensive	16	Figure 4.6
2	Wear-Out Intensive	128	Figure 4.7
3	Initial-Failure Intensive	16	Figure 4.8
4	Initial-Failure Intensive	128	Figure 4.9

In the wear-out intensive cases shown in Figures 4.6 and 4.7, when the monitoring ratio was changed from 0:10 to 10:0, the average failure localization time of the UFM method increased from 736.1 seconds to 752.7 seconds in the 16-MEP case and from 660.8 seconds to 742.3 seconds in the 128-MEP case. Also, the average server failure detection rate decreased from 6.2% to 0.5% in the 16-MEP case and from 32.3% to 4.0% in the 128-MEP case. In both cases, when the ratio exceeded 9:1, the average failure localization time was longer than that of the conventional method that randomly assigns MEPs: approximately 750 seconds in the 16-MEP case and 720 seconds in the 128-MEP case. Also, the average server failure detection rate was lower than that of the conventional method: approximately 2% in the 16-MEP case and 13% in the 128-MEP case. These results indicate that the performance of the failure management degrades. The reason is presumably that the MEPs are configured for VMs that have been running for almost no time, and failures do not occur on these VMs since wear-out is the cause of most failures.

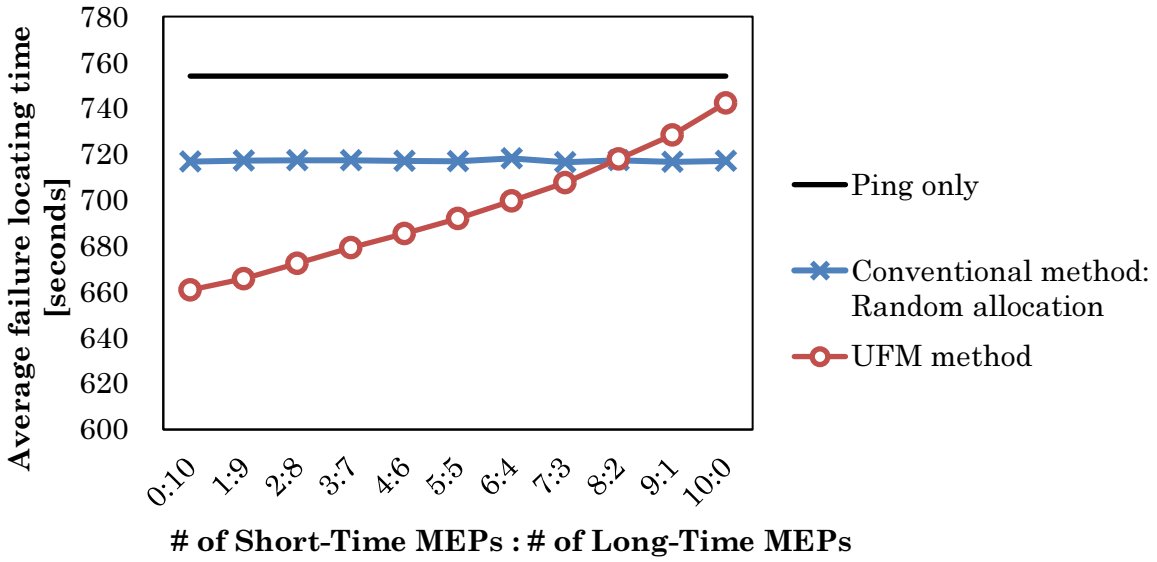


(a) Analysis Results of Average Failure Localization Time

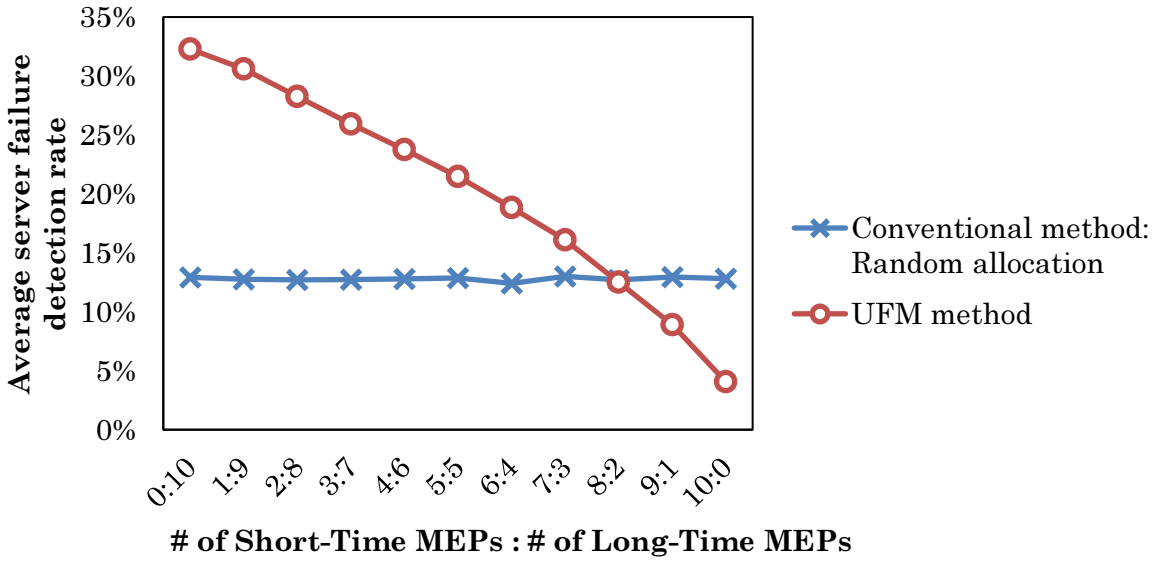


(b) Analysis Results of Average Server Failure Detection Rate

Figure 4.6: Sensitivity Analysis for Monitoring Ratios in 16 MEPs and Wear Out Intensive Case

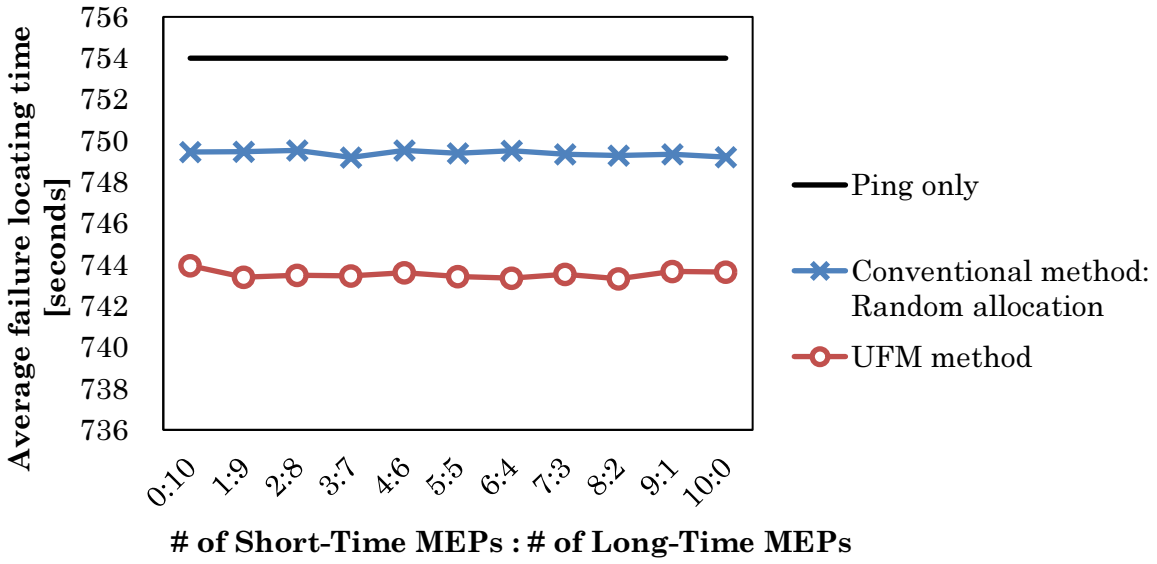


(a) Analysis Results of Average Failure Localization Time

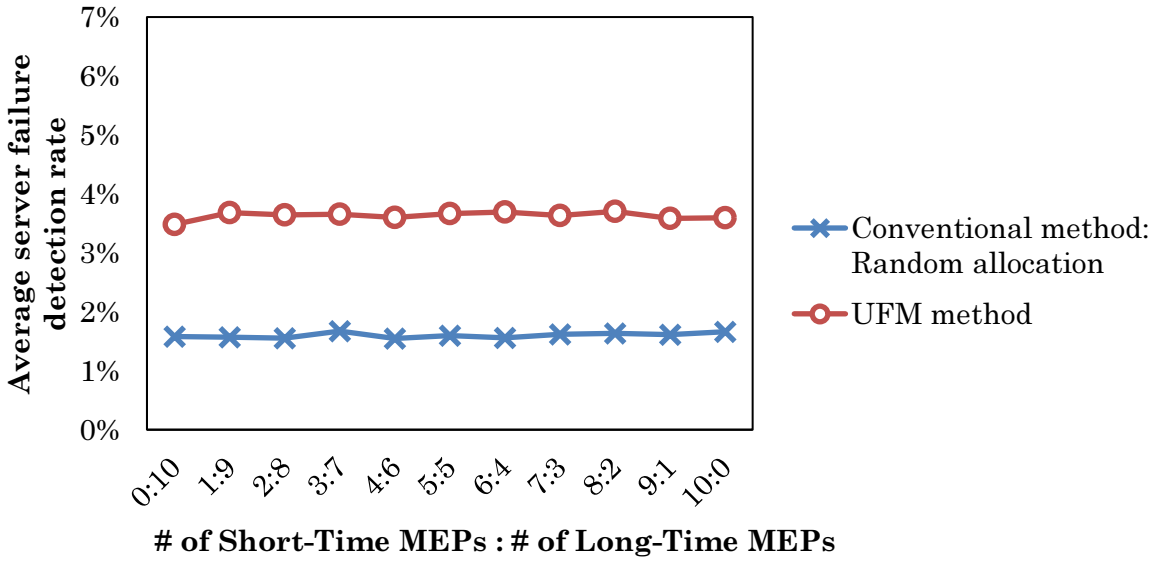


(b) Analysis Results of Average Server Failure Detection Rate

Figure 4.7: Sensitivity Analysis for Monitoring Ratios in 128 MEPs and Wear Out Intensive Case

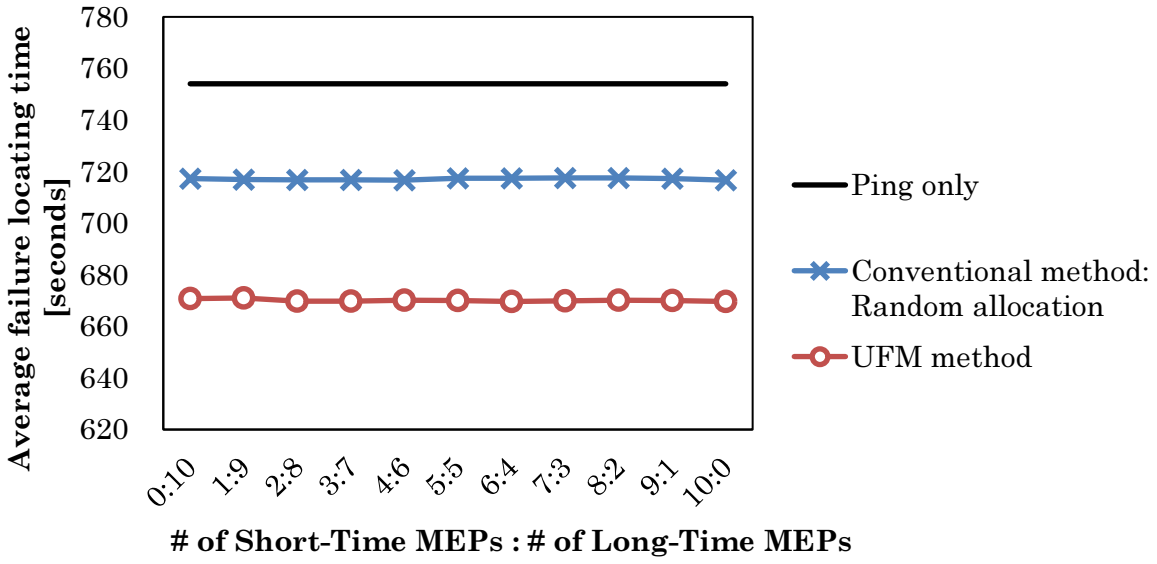


(a) Analysis Results of Average Failure Localization Time

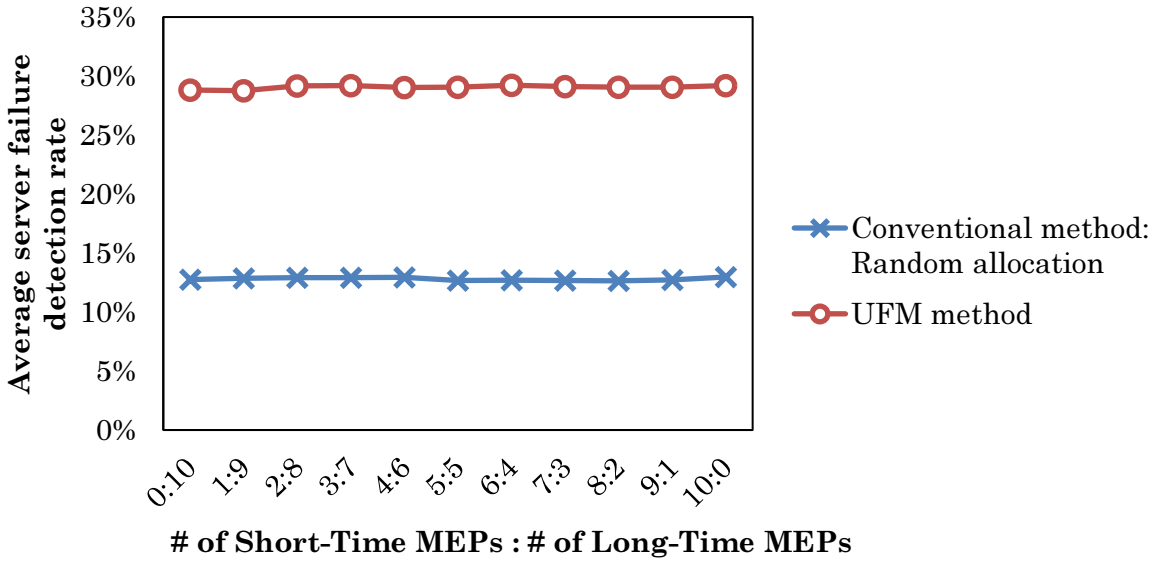


(b) Analysis Results of Average Server Failure Detection Rate

Figure 4.8: Sensitivity Analysis for Monitoring Ratios in 16 MEPs and Initial Failure Intensive Case



(a) Analysis Results of Average Failure Localization Time



(b) Analysis Results of Average Server Failure Detection Rate

Figure 4.9: Sensitivity Analysis for Monitoring Ratios in 128 MEPs and Initial Failure Intensive Case

In the initial-failure intensive cases shown in Figures 4.8 and 4.9, even if the monitoring ratio was changed from 0:10 to 10:0, the average failure localization time and the average server failure detection rate kept almost the same values. The average failure localization time was maintained at approximately 744 seconds in the 16-MEP case and at 670 seconds in the 128-MEP case. Also, the average server failure detection rate was maintained at approximately 3.6 % in the 16-MEP case and at 29 % in the 128-MEP case. The UFM method always exhibited a better performance than the conventional method that randomly assigns MEPs.

4.4.4 Guidelines for choosing UFM method or RFM method

As mentioned in Chapter 1, the UFM method was developed so that the dynamic network monitoring method could be widely applied to other data centers. As its performance is basically inferior to that of the RFM method proposed in Chapter 3, it is preferable for data center operators to adopt the RFM method when possible. The guidelines for clarifying which method to use are as follows. First, if a network operator cannot obtain the failure history of the servers or VMs in a data center due to difficulties in the introduction or operations of a failure management system, the RFM method cannot be used and the UFM method should be selected instead. Also, even when it is possible to obtain the failure history, the operator should consider applying the UFM method if the server or server virtualization infrastructure has been frequently updated before constructing the reliability models. This is because, as the results of the above sensitivity analysis show, the server failure detection rate drops to the same level as the conventional method with random assignment of MEPs when initial failures are dominant. In contrast, if the operator can obtain a sufficient failure history and if the servers and server virtualization platforms to be used have not been significantly updated, the RFM method should be used.

4.5 Related Work

To the best of our knowledge, as discussed in Chapter 3, no prior works have combined the network-level and high-frequency monitoring methods (such as Ethernet OAM) and service-level monitoring methods (such as ping) to improve the failure localization operations in data centers. On the other hand, various methods have been proposed to identify failures for wide-area network services that operate and manage many network devices, although these are not aimed at data center networks composed of servers and network

devices. For example, there is an active probing method that reduces the number of times a probe is needed[112], where a probe is selected from multiple available candidates in the network on the basis of the result of the previous probing. There is also a method of shortening the time required to identify the cause of failure while avoiding the inconvenience of estimating the cause of active failure[27]. This method selects the appropriate action to collect the information needed to narrow down the failure factors on the basis of the symptoms found by passive monitoring. When we compare the UFM method in our study with the conventional methods, it is clear that the operational efficiency of data center networks can be improved since the intrusiveness of active monitoring can be suppressed by the UFM method, which select the targets of active monitoring in accordance with the configuration information of servers or VMs only.

4.6 Conclusion

We proposed a new network monitoring method in which a network management system dynamically selects network endpoints as the targets of Ethernet OAM without needing detailed statistical information about the VMs running in data centers. In this method, the network management system determines which network endpoint should be monitored according to the uptime of each VM only. The system focuses on monitoring the initial failure and wear-out. Simulation results showed that the proposed method could reduce the average failure localization time by 0.8 % in the 16-MEP case and by 6.6 % in the 128-MEP case. Also, the average server failure detection rate was improved by 2.1 points in the 16-MEP case and by 8.6 points in the 128-MEP case compared to the conventional method of randomly selecting the monitoring targets of Ethernet OAM CC. Using this method, data center administrators can dynamically allocate monitoring resources for Ethernet OAM without having to create a reliability model of the servers in advance. As such, this method can be widely applied to data centers that do not usually collect the failure statistics of VMs.

A future challenge will be extending the data-center network failure management method to support network appliances deployed in the form of VMs when Network Function Virtualization (NFV)[113] becomes more widely applied in data centers. Efficient deployment methods for virtual network functions have been studied before[114, 115]. In a data center where NFV is applied, the server virtualization platform runs network appliances such as firewalls and load balancers which are installed in the core side of the virtual network. Therefore, when a network failure is detected by Ethernet OAM CC,

it cannot be immediately determined whether it is a failure on the server side or the network side. It is thus necessary to develop a method that can localize the root cause of the failure in a virtual network featuring NFV-based network appliances.

Chapter 5

Conclusion

5.1 Concluding Remarks

This dissertation described the research on network operations and management for the service availability of large-scale data centers in which servers and networks are virtualized to provide various services for many users.

Chapter 1 described the conventional methods and trends in the operations and management of data-center networks through which various services are provided. It clarified the following operational issues that need to be resolved: (1) providing configuration information that enables administrators to manage the configurations of servers and networks in an integrated manner, and (2) selecting the network end points related to services that would fail as the monitoring targets of Ethernet OAM.

It defined the following strategies to resolve these issues: (1) Integrated configuration management of virtual networks (2) Dynamic monitoring of data-center networks The latter issue was further divided into (2-1) Dynamic monitoring of data-center networks based on a reliability model of VMs (2-2) Dynamic monitoring of data-center networks based on server configuration information

Chapter 2 proposed a data-center network configuration management method using a data model that consolidates the configuration information of servers and networks to enable centralized management for the configurations of both. It evaluated the operational efficiency and extensibility in a prototype system. The evaluation results showed that, compared to the conventional method of independently managing the configuration information of servers and networks, it is possible to shorten the time needed to verify the configurations of virtual networks across servers and networks and also to reduce the number of operational errors.

Chapter 3 proposed a data-center network failure management method called RFM to select the network ports monitored with Ethernet OAM based on a probabilistic reliability model of VMs in a data center. It evaluated the operational time to locate the root cause with a simulation. The results showed that, compared to the conventional method of network monitoring, it is possible to shorten the time to locate the root cause of the service failure.

Chapter 4 proposed a new failure management method called UFM by extending the RFM method proposed in Chapter 3 to enable wide deployment to data centers that do not have enough management information about failures. In this method, the network ports monitored with Ethernet OAM are selected based on the uptime of VMs in the data center. It evaluated the operational time to locate the root cause with a simulation. The results showed that, compared to the conventional method of network monitoring, it can reduce the amount of information required to select the monitored network ports.

The proposed configuration management method enables the configuration information of virtual networks across multiple servers and networks to be provided in an integrated manner. In addition, the proposed failure management method enables the network end points monitored by Ethernet OAM to be appropriately selected for service failures. Combined, these methods enable the efficient management of virtual networks across servers and networks in a data center from the viewpoints of both configuration management and failure management among network operations and management. As such, the research presented in this dissertation will contribute to making the operations and management of virtual networks in large-scale data centers where virtualization technologies are deployed more efficient and help to maintain the service levels of services provided from the data centers.

5.2 Future Directions

In the future, network configuration management for large-scale data centers will be required to support more transport protocols in addition to VLAN. In response to the growing demand for large-scale data centers, more scalable communication protocols are being used for network virtualization as described in Section 2. Provider Backbone Bridge (PBB), which is standardized as IEEE 802.1ah, is one such communication protocol. In addition to PBB, overlay transport protocols such as Virtual eXtensible Local Area Network (VXLAN) and Network Virtualization using Generic Routing Encapsulation (NVGRE) are being standardized and deployed in large-scale data centers. It is thus an

additional issue to manage the configurations of the many virtual networks that are created using scalable transport protocols other than VLAN[116]. Also, since these protocols can create more virtual networks than VLAN, they are also used to serve many users in multiple areas by creating a layer-2 wide area network that spans geographically. Therefore, in the future configuration management of data-center networks, it will be necessary to manage the geographical configuration information as well.

As data-center networks continue to provide more functions, failure management will require operations more complex than a simple connectivity check. It is already necessary for network operators to detect signs of failure before the data center network loses its ability to satisfy the guaranteed service level so that the network device can be fixed promptly. Therefore, it is necessary to analyze the behavior of the network based on the network configuration and protocol processing[117, 118, 119, 120]. It is also necessary to take advantage of passive monitoring methods utilizing the log information and statistical information of network devices in addition to active monitoring methods such as Ethernet OAM. This requires higher performance in terms of processing the log information and statistical information. In response to this increasing demand for real-time performance, it may be conceivable to utilize event stream processing platforms or messaging platforms[121].

In data centers, lightweight container technologies such as Docker[1] have become widely used in place of VMs for the virtualization of server resources. Since containers can be started, stopped, and moved more easily than VMs, the load on operation management is quite high. In response, Kubernetes[2] was developed and released as an orchestration system for container-oriented cluster systems[122] and is now being researched from various viewpoints such as flexibility and performance[123, 124]. It automates the operational flows of the configuration management and failure management of containers. Unfortunately, this orchestration system further abstracts the network configuration of the system. This makes it more difficult to associate the physical environment with the logical environment, and subsequently the configuration management and failure management become more difficult from the viewpoint of the physical environment. In the future, it will be necessary to integrate the operations and management across multiple layers of network hardware and software by utilizing the methods for collecting information on the physical configuration and virtual configuration of the network proposed in this research.

[1] `https://www.docker.com/`
[2] `https://kubernetes.io/`

Acknowledgment

I express my sincere gratitude to Professor Toru Fujiwara and Professor Emeritus Norihisa Komoda of the Graduate School of Information Science and Technology at Osaka University for their visionary directions and constructive guidance on this research and on writing up the thesis and academic papers. Without their support and encouragement, this study could have been neither carried out nor concluded at all.

I would like to thank Professors Yasuyuki Matsushita and Shinji Shimojo of the Graduate School of Information Science and Technology at Osaka University for their numerous suggestions for revising this dissertation.

I also so much appreciate to Professors Takahiro Hara and Makoto Onizuka of Graduate School of Information Science and Technology at Osaka University for their valuable comments and supports on this research.

I also so much appreciate to Dr. Masakatsu Mori, General Manager of Global Center for Social Innovation, Research and Development Group, Hitachi, Ltd. for his suggestion of my admission to the doctoral program at Osaka University.

I also so much appreciate to Mr. Masahiro Yoshizawa, Dr. Keitaro Uehara, Mr. Kazuhiko Mizuno, Mr. Toshiaki Tarui, Dr. Ken Naono and Dr. Hayato Hoshihara whom I worked with together to establish the research domain of the operations and management of virtual networks across the server and the network at Central Research Laboratory, Hitachi, Ltd. This study would not have been possible without their deep knowledge in computing and networking and their cooperation across different research fields.

I also so much appreciate to Mr. Tetsuhiko Hirata, Mr. Itaru Mimura, Dr. Kenichi Sakamoto, Dr. Seishi Hanaoka, Dr. Yuichiro Nakaya, Mr. Nobuo Nukaga, Mr. Hiroki Kitagawa and Mr. Masaaki Tanizaki for their managerial support and leadership while my assignment at Central Research Laboratory, Center for Technology Innovation and Global Center for Social Innovation, Research and Development Group, Hitachi, Ltd.

I must express my gratitude to the current colleagues at Global Center for Social Innovation, Research and Development Group, Hitachi, Ltd. for a lot of their supports

so that I could proceed this study.

Last but not least, I would also like to express my gratitude to my family for their moral support and warm encouragement.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in Proceedings of the ACM SIGCOMM 2008, pp.63–74, Aug. 2008. https://doi.org/10.1145/1402958.1402967

[2] D. Li, H. Zhao, M. Xu, and X. Fu, "Revisiting the design of mega data centers: Considering heterogeneity among containers," IEEE/ACM Transactions on Networking, vol.22, no.5, pp.1503–1515, Oct. 2014. https://doi.org/10.1109/TNET.2013.2280764

[3] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (dcn): Infrastructure and operations," IEEE Communications Surveys & Tutorials, vol.19, no.1, pp.640–656, 2017. https://doi.org/10.1109/COMST.2016.2626784

[4] B. Schroeder and G.A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?," in Proceedings of 5th USENIX Conference on File and Storage Technologies (FAST 2007), pp.1–16, Feb. 2007.

[5] R.K. Sahoo, A. Sivasubramaniam, M.S. Squillante, and Y. Zhang, "Failure data analysis of a large-scale heterogeneous server environment," in Proceedings of 2004 International Conference on Dependable Systems and Networks (DSN 2004), pp.772–781, June 2004. https://doi.org/10.1109/DSN.2004.1311948

[6] E. Pinheiro, W.D. Weber, and L.A. Barroso, "Failure trends in a large disk drive population," in Proceedings of 5th USENIX Conference on File and Storage Technologies (FAST 2007), pp.17–29, Feb. 2007.

[7] B. Schroeder, E. Pinheiro, and W.D. Weber, "DRAM errors in the wild: A large-scale field study," in Proceedings of ACM SIGMETRICS 2009, pp.193–204, June 2009. https://doi.org/10.1145/1555349.1555372

[8] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements & analysis," in Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC 2009), pp.202–208, Association for Computing Machinery, New York, NY, USA, 2009. https://doi.org/10.1145/1644893.1644918

[9] K.V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in Proceedings of ACM SoCC 2010, pp.193–204, June 2010. https://doi.org/10.1145/1807128.1807161

[10] D. Ford, F. Labelle, F.I. Popovici, M. Stokely, V.A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in Proceedings of 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010), pp.61–74, Oct. 2010.

[11] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in Proceedings of the ACM SIGCOMM 2011 Conference, pp.350–361, Aug. 2011. https://doi.org/10.1145/2018436.2018477

[12] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," in Proceedings of 2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST 2013), pp.1–5, 2013. https://doi.org/10.1109/MSST.2013.6558427

[13] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu, "Hard drive failure prediction using classification and regression trees," in Proceedings of 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014), pp.383–394, June 2014. https://doi.org/10.1109/DSN.2014.44

[14] D. Oppenheimer, A. Ganapathi, and D.A. Patterson, "Why do internet services fail, and what can be done about it?," in Proceedings of 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003), pp.1–16, March 2003.

[15] B. Schroeder and G.A. Gibson, "A large-scale study of failures in high-performance computing systems," in Proceedings of 2006 International Conference on Dependable Systems and Networks (DSN 2006), pp.249–258, June 2006. https://doi.org/10.1109/DSN.2006.5

[16] B. Schroeder and G.A. Gibson, "A large-scale study of failures in high-performance computing systems," IEEE Transactions on Dependable and Secure Computing, vol.7, no.4, pp.337–350, 2010. https://doi.org/10.1109/TDSC.2009.4

[17] R. Birke, I. Giurgiu, L.Y. Chen, D. Wiessman, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," in Proceedings of 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014), pp.1–12, June 2014. https://doi.org/10.1109/DSN.2014.18

[18] "Principles for a telecommunications management framework," ITU-T Recommendation M.3010, Feb. 2000.

[19] "TMN management functions," ITU-T Recommendation M.3400, Feb. 2000.

[20] "Information processing system - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework," ISO/IEC 7498-4, Nov. 1989.

[21] P. Goyal, R. Mikkilineni, and M. Ganti, "Fcaps in the business services fabric model," in Proceedings of 2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, pp.45–51, June 2009. https://doi.org/10.1109/WETICE.2009.21

[22] F. Tusa, A. Celesti, and R. Mikkilineni, "AAA in a cloud-based virtual DIME Network Architecture (DNA)," in Proceedings of 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp.110–115, June 2011. https://doi.org/10.1109/WETICE.2011.20

[23] M. Mohamed, D. Belaïd, and S. Tata, "Self-managed micro-containers for service-based applications in the cloud," in Proceedings of 2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp.140–145, June 2013. https://doi.org/10.1109/WETICE.2013.59

[24] I. Katzela and M. Schwartz, "Schemes for fault identification in communication networks," IEEE/ACM Transactions on Networking, vol.3, no.6, pp.753–764, Dec. 1995. https://doi.org/10.1109/90.477721

[25] M. Thottan and C. Ji, "Anomaly detection in IP networks," IEEE Transactions on Signal Processing, vol.51, no.8, pp.2191–2204, Aug. 2003. https://doi.org/10.1109/TSP.2003.814797

[26] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in Proceedings of ACM SIGCOMM 2005, pp.217–228, 2005. https://doi.org/10.1145/1080091.1080118

[27] Y. Tang, E. Al-Shaer, and R. Boutaba, "Active integrated fault localization in communication networks," in Proceedings of 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), pp.543–556, May 2005. https://doi.org/10.1109/INM.2005.1440826

[28] D. Liu, C.-H. Jung, I. Lambadaris, and N. Seddigh, "Network traffic anomaly detection using clustering techniques and performance comparison," in Proceedings of 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2013), pp.345–348, 2013. https://doi.org/10.1145/1080091.1080118

[29] H. Mi, H. Wang, Y. Zhou, M.R.T. Lyu, and H. Cai, "Toward fine-grained, unsupervised, scalable performance diagnosis for production cloud computing systems," IEEE Transactions on Parallel and Distributed Systems, vol.24, no.6, pp.1245–1255, June 2013. https://doi.org/10.1109/TPDS.2013.21

[30] R.L. Gomes, L.F. Bittencourt, E.R.M. Madeira, E.C. Cerqueira, and M. Gerla, "Software-defined management of edge as a service networks," IEEE Transactions on Network and Service Management, vol.13, no.2, pp.226–239, May 2016. https://doi.org/10.1109/TNSM.2016.2538821

[31] A. Watanabe, K. Ishibashi, T. Toyono, T. Kimura, K. Watanabe, Y. Matsuo, and K. Shiomoto, "Workflow extraction for service operation using multiple unstructured trouble tickets," in Proceedings of 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), pp.652–658, April 2016. https://doi.org/10.1109/NOMS.2016.7502872

[32] Y. Matsuo, Y. Nakano, A. Watanabe, K. Watanabe, K. Ishibashi, and R. Kawahara, "Root-cause diagnosis for rare failures using bayesian network with dynamic modification," in Proceedings of 2018 IEEE International Conference on Communications (ICC 2018), pp.1–6, May 2018. https://doi.org/10.1109/ICC.2018.8422955

[33] H. Ikeuchi, A. Watanabe, T. Kawata, and R. Kawahara, "Root-cause diagnosis using logs generated by user actions," in Proceedings of 2018 IEEE Global Communications Conference (GLOBECOM 2018), pp.1–7, Dec. 2018. https://doi.org/10.1109/GLOCOM.2018.8647957

[34] A. Shiozu, M. Suzuki, S. Shibata, and Y. Suwa, "A silent anomaly detection approach using machine learning in mobile access networks," IEICE Technical Report ICM, vol.119, no.299, pp.1–6, Nov. 2019 (in Japanese).

[35] T. Kimura, R. Saito, Y. Suwa, S. Shibata, A. Shiozu, Y. Inagaki, M. Suzuki, and K. Yamamoto, "Efforts to apply AI to network operation analysis," IEICE Technical Report ICM, vol.119, no.438, pp.95–99, March 2020 (in Japanese).

[36] H. Ikeuchi, A. Watanabe, T. Hirao, M. Morishita, M. Nishino, Y. Matsuo, and K. Watanabe, "Recovery command generation towards automatic recovery in ICT systems by Seq2Seq learning," in Proceedings of 2020 IEEE/IFIP Network Operations and Management Symposium (NOMS 2020), pp.1–6, April 2020. https://doi.org/10.1109/NOMS47738.2020.9110370

[37] J. Postel, "Internet Control Message Protocol (ICMP)," IETF RFC 792, Sept. 1981. https://datatracker.ietf.org/doc/html/rfc792

[38] J. Postel, "Transmission Control Protocol," IETF RFC 793, Sept. 1981. https://datatracker.ietf.org/doc/html/rfc793

[39] M. McFarland, S. Salam, and R. Checker, "Ethernet OAM: Key enabler for carrier class metro ethernet services," IEEE Communications Magazine, vol.43, no.11, pp.152–157, Nov. 2005. https://doi.org/10.1109/MCOM.2005.1541707

[40] IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks Amendment 5: Connectivity Fault Management," IEEE Std 802.1ag - 2007 (Amendment to IEEE Std 802.1Q - 2005 as amended by IEEE Std 802.1ad - 2005 and IEEE Std 802.1ak - 2007), pp.1–260, Dec. 2007. https://doi.org/10.1109/IEEESTD.2007.4431836

[41] "Operation, administration and maintenance (OAM) functions and mechanisms for Ethernet-based networks," ITU-T Recommendation G.8013/Y.1731, Aug. 2015.

[42] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, vol.53, no.4, pp.50–58, April 2010. https://doi.org/10.1145/1721654.1721672

[43] IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks," IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014), pp.1–1993, July 2018. https://doi.org/10.1109/IEEESTD.2018.8403927

[44] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker, "Extending networking into the virtualization layer," in Proceedings of 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII), pp.1–6, Oct. 2009.

[45] V. Soundararajan and J.M. Anderson, "The impact of management operations on the virtualized datacenter," in Proceedings of the 37th Annual International Symposium on Computer Architecture (ISCA 2010), pp.326–337, 2010. https://doi.org/10.1145/1815961.1816003

[46] V. Soundararajan and K. Govil, "Challenges in building scalable virtualized datacenter management," ACM SIGOPS Operating Systems Review, vol.44, no.4, pp.95–102, Dec. 2010. https://doi.org/10.1145/1899928.1899941

[47] W. Fuertes, J.E.L. deVergara, F. Meneses, and F. Galán, "A generic model for the management of virtual network environments," in Proceedings of 2010 IEEE Network Operations and Management Symposium (NOMS 2010), pp.813–816, April 2010. https://doi.org/10.1109/NOMS.2010.5488367

[48] F. Galán, J.E.L. deVergara, D. Fernández, and R.M. noz, "A model-driven configuration management methodology for testbed infrastructures," in Proceedings of 2008 IEEE Network Operations and Management Symposium (NOMS 2008), pp.747–750, April 2008. https://doi.org/10.1109/NOMS.2008.4575204

[49] F. Galán, J.E.L. deVergara, D. Fernández, and R.M. noz, "Scenario-based configuration management for flexible experimentation infrastructures," in Proceedings of 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops (TRIDENTCOM 2009), pp.1–10, April 2009. https://doi.org/10.1109/TRIDENTCOM.2009.4976228

[50] I. Fajjari, M. Ayari, and G. Pujolle, "VN-SLA: A virtual network specification schema for virtual network provisioning," in Proceedings of 9th International Conference on Networks (ICN 2010), pp.337–342, April 2010. https://doi.org/10.1109/ICN.2010.60

[51] M. Yampolskiy and M.K. Hamm, "Management of multidomain end-to-end links - a federated approach for the pan-european research network geant 2," in Proceedings of 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007), pp.189–198, May 2007. https://doi.org/10.1109/INM.2007.374783

[52] "CIM system virtualization model white paper," DMTF DSP 2013, Nov. 2007.

[53] "Virtual ethernet switch profile version 1.0.0," DMTF Profile DSP1097, Oct. 2010.

[54] P. Congdon, A. Fischer, and P. Mohapatra, "A case for VEPA: Virtual ethernet port aggregator," in Proceedings of 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC CAVES 2010) (in CD-ROM), Sept. 2010.

[55] J. Pelissier and R. Raeber, "Introduction to port extension (IEEE P802.1Qbh)," in Proceedings of 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC CAVES 2010) (in CD-ROM), Sept. 2010.

[56] P. Dini, M.Z. Hasan, M. Morrow, G. Parr, and P. Rolin, "IP/MPLS OAM: Challenges and directions," in Proceedings of IEEE International Workshop on IP Operations and Management (IPOM 2004), pp.1–8, 2004. https://doi.org/10.1109/IPOM.2004.1547584

[57] P. Varga and I. Moldovan, "Integration of service-level monitoring with fault management for end-to-end multi-provider ethernet services," IEEE Transactions on Network and Service Management, vol.4, no.1, pp.28–38, June 2007. https://doi.org/10.1109/TNSM.2007.030103

[58] M. Casado, M.J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," ACM SIGCOMM Computer Communication Review, vol.37, no.4, pp.1–12, Aug. 2007. https://doi.org/10.1145/1282427.1282382

[59] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.69–74, March 2008. https://doi.org/10.1145/1355734.1355746

[60] D. Marschke, J. Doyle, and P. Moyer, Software Defined Networking (SDN): Anatomy of OpenFlow Volume I, Lulu Publishing Services, March 2015.

[61] P.C. Fonseca and E.S. Mota, "A survey on fault management in Software-Defined Networks," IEEE Communications Surveys & Tutorials, vol.19, no.4, pp.2284–2321, 2017. https://doi.org/10.1109/COMST.2017.2719862

[62] U.C. Kozat, G. Liang, and K. Kökten, "On diagnosis of forwarding plane via static forwarding rules in Software Defined Networks," in Proceedings of 33rd IEEE Conference on Computer Communications (INFOCOM 2014), pp.1716–1724, April 2014. https://doi.org/10.1109/INFOCOM.2014.6848109

[63] S.S.W. Lee, K.Y. Li, K.Y. Chan, G.H. Lai, and Y.C. Chung, "Path layout planning and software based fast failure detection in survivable OpenFlow networks," in Proceedings of 2014 10th International Conference on the Design of Reliable Communication Networks (DRCN 2014), pp.1–8, 2014. https://doi.org/10.1109/DRCN.2014.6816141

[64] S.S.W. Lee, K.Y. Li, K.Y. Chan, G.H. Lai, and Y.C. Chung, "Software-based fast failure recovery for resilient OpenFlow networks," in Proceedings of 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM 2015), pp.194–200, 2015. https://doi.org/10.1109/RNDM.2015.7325229

[65] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "OpenFlow: Meeting carrier-grade recovery requirements," Computer Communications, vol.36, no.6, pp.656–665, March 2013. https://doi.org/10.1016/j.comcom.2012.09.011

[66] D. Gyllstrom, N. Braga, and J. Kurose, "Recovery from link failures in a Smart Grid communication network using OpenFlow," in Proceedings of 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm 2014), pp.254–259, Nov. 2014. https://doi.org/10.1109/SmartGridComm.2014.7007655

[67] D. Kotani and Y. Okabe, "Fast failure detection of OpenFlow channels," in Proceedings of the 11th Asian Internet Engineering Conference (AINTEC 2015), pp.32–39, Nov. 2015. https://doi.org/10.1145/2837030.2837035

[68] J. Kempf, E. Bellagamba, A. Kern, D. Jocha, A. Takacs, and P. Sköldström, "Scalable fault management for OpenFlow," in Proceedings of 2012 IEEE International Conference on Communications (ICC 2012), pp.6606–6610, 2012. https://doi.org/10.1109/ICC.2012.6364688

[69] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z.-W. Lin, and V. Kurien, "Pingmesh: A large-scale system for data center network latency measurement and analysis," in Proceedings of ACM SIGCOMM 2015, pp.139–152, Aug. 2015. http://dx.doi.org/10.1145/2785956.2787496

[70] H. Singh, "Fault-tolerance in Windows Azure SQL database, Azure Blog," July 2012. https://azure.microsoft.com/en-us/blog/fault-tolerance-in-windows-azure-sql-database/

[71] M. Szpuszta and S. Vaitinadin, "Microsoft Azure - fault tolerance pitfalls and resolutions in the cloud," MSDN Magazine, vol.30, no.9, pp.38–42, Sept. 2015.

[72] Z. Zheng, T.C. Zhou, M.R. Lyu, and I. King, "Component ranking for fault-tolerant cloud applications," IEEE Transactions on Services Computing, vol.5, no.4, pp.540–550, 2012. https://doi.org/10.1109/TSC.2011.42

[73] M. Yoshizawa, T. Tarui, and H. Okita, "Implementation and evaluation of network management system to reduce management cost caused by server virtualization," IEICE Technical Report, vol.109, pp.71–76, NS2009-116, Ishikawa, Nov. 2009 (in Japanese).

[74] M. Yoshizawa, H. Okita, K. Uehara, and T. Tarui, "Implementation and evaluation of network management system supporting documentation for virtual networks," IPSJ Journal, vol.52, no.3, pp.1334–1347, March 2011 (in Japanese).

[75] H. Okita, M. Yoshizawa, K. Uehara, K. Mizuno, T. Tarui, and K. Naono, "Proposal of virtual network configuration acquisition function for data center operations and management system," IEICE Technical Report, vol.110, pp.67–72, ICM2010-20, Hokkaido, July 2010 (in Japanese).

[76] H. Okita, M. Yoshizawa, K. Uehara, K. Mizuno, T. Tarui, and K. Naono, "Virtual network configuration management system for data center operations and management," IEICE Transactions on Communications, vol.E95-B, no.6, pp.1924–1933, July 2012. https://doi.org/10.1587/transcom.E95.B.1924

[77] M. Yoshizawa, T. Tarui, and H. Okita, "Implementation and evaluation of network management system to reduce management cost caused by server virtualization," in Proceedings of 2nd Workshop on Data Center – Converged and Virtual Ethernet Switching (DC CAVES 2010) (in CD-ROM), Sept. 2010.

[78] H. Okita, M. Yoshizawa, K. Uehara, K. Mizuno, T. Tarui, and K. Naono, "Proposal of virtual network configuration acquisition function for data center operations and management system," in Proceedings of 17th International European Conference on Parallel and Distributed Computing (Euro-Par 2010) Parallel Processing Workshops, LNCS, vol.6586, pp.625–632, Aug. 2010. https://doi.org/10.1007/978-3-642-21878-1

[79] H. Okita, M. Yoshizawa, D. Matsubara, and J. Yamamoto, "Research and development of network solution technology for next generation networks," Hitachi Review, vol.88, no.6, pp.494–497, June 2006 (in Japanese).

[80] D. Matsubara, H. Okita, Y. Kanada, and H. Yoshida, "Research and development of traffic control technology for next generation networks," Hitachi Review, vol.89, no.6, pp.472–475, June 2007 (in Japanese).

[81] H. Okita, H. Hoshihara, N. Komoda, and T. Fujiwara, "Dynamically prioritized virtual-network monitoring according to lifecycle of virtual machines in large scale data center," in Proceedings of 19th International Conference on WWW/Internet 2020 (ICWI 2020), pp.123–131, Nov. 2020.

[82] H. Okita, H. Hoshihara, N. Komoda, and T. Fujiwara, "Dynamically prioritized failure management according to reliability model in large-scale data center," IADIS International Journal on WWW/Internet, vol.18, no.2, pp.101–115, Dec. 2020.

[83] H. Okita, H. Hoshihara, N. Komoda, and T. Fujiwara, "Dynamically prioritized virtual-network monitoring according to uptime of virtual machines," IEICE Technical Report, vol.120, pp.61–66, ICM2020-31, Online, Nov. 2020 (in Japanese).

[84] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction and applicability statements for internet standard management framework," IETF RFC 3410, Dec. 2002. https://datatracker.ietf.org/doc/html/rfc3410

[85] R. Israel, Y. Fang, P. Cohen, and E. Eichen, "Configuration management of large IP telephony networks," in Proceedings of 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS 2000), pp.435–446, April 2000. https://doi.org/10.1109/NOMS.2000.830401

[86] M.S. Kim and A. Leon-Garcia, "Autonomic network resource management using virtual network concept," in Proceedings of 10th Asia-Pacific Network Operations and Management Symposium (APNOMS 2007), pp.254–264, Oct. 2007.

[87] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks Amendment 7: Provider Backbone Bridges," IEEE Std 802.1ah-2008 (Amendment to IEEE Std 802.1Q-2005), pp.1–110, Aug. 2008. https://doi.org/10.1109/IEEESTD.2008.4602826

[88] M. Mahalingam, D.G. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," IETF RFC 7348, Aug. 2014. https://datatracker.ietf.org/doc/html/rfc7348

[89] P. Garg and Y.S. Wang, "NVGRE: Network Virtualization Using Generic Routing Encapsulation," IETF RFC 7637, Sept. 2015. https://datatracker.ietf.org/doc/html/rfc7637

[90] C. Perkins and J. Solomon, "IP encapsulation within IP," IETF RFC 2003, Oct. 1996. https://datatracker.ietf.org/doc/html/rfc2003

[91] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic Routing Encapsulation (GRE)," IETF RFC 2784, March 2000. https://datatracker.ietf.org/doc/html/rfc2784

[92] G. Dommety, "Key and sequence number extensions to GRE," IETF RFC 2890, Sept. 2000. https://datatracker.ietf.org/doc/html/rfc2890

[93] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) bridges," IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998), pp.1–281, June 2004. https://doi.org/10.1109/IEEESTD.2004.94569

[94] J. Dike, "A user-mode port of the Linux kernel," in Proceedings of the 4th Annual Linux Showcase & Conference (ALS 2000), pp.1–10, Oct. 2000. https://www.usenix.org/conference/als-2000/user-mode-port-linux-kernel

[95] T. Bray, J. Paoli, C. Sperberg-McQueen, and E. Maler, W3C REC-xml Oct. 2000. http://www.w3.org/TR/REC-xml/

[96] "Management information eXchange over Internet - Network Management Ver.1.0 (MAXI-NM-1.0)," INTAP/OSMIC, Feb. 2007.

[97] IEEE Computer Society, "IEEE Standard for Local and Metropolitan Area Networks–Link Aggregation," IEEE Std 802.1AX-2020 (Revision of IEEE Std 802.1AX-2014), pp.1–333, May 2020. https://doi.org/10.1109/IEEESTD.2020.9105034

[98] R. Enns, "NETCONF Configuration Protocol," IETF RFC 4741, Dec. 2006. https://datatracker.ietf.org/doc/html/rfc4741

[99] K. McCloghrie and M.T. Rose, "Management Information Base for network management of TCP/IP-based internets: MIB-II," IETF RFC 1213, March 1991. https://datatracker.ietf.org/doc/html/rfc1213

[100] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery," IEEE Std 802.1AB-2016 (Revision of IEEE Std 802.1AB-2009), pp.1–146, March 2016. https://doi.org/10.1109/IEEESTD.2016.7433915

[101] H. Okita, K. Uehara, and Y. Yasuda, "Proposal and evaluation of data-model translation for server-virtualization environment to improve virtual-network configuration management," in Proceedings of 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI 2016), vol.1, pp.878–883, July 2016. https://doi.ieeecomputersociety.org/10.1109/IIAI-AAI.2016.168

[102] H. Okita, M. Yoshizawa, K. Uehara, Y. Yasuda, and M. Yamada, "Proposal and evaluation of data-model translation for server-virtualization environment to improve virtual-network configuration management," IEICE Technical Report, vol.110, pp.41–46, ICM2010-61, Okinawa, March 2011 (in Japanese).

[103] A. Iwata, "Carrier-grade ethernet technologies for next generation wide area ethernet," IEICE Transactions on Communications, vol.E89-B, no.3, pp.651–660, March 2006. https://doi.org/10.1093/ietcom/e89-b.3.651

[104] J. Yang and F.B. Sun, "A comprehensive review of hard-disk drive reliability," in Proceedings of 1999 Annual Reliability and Maintainability Symposium (RAMS 1999), pp.403–409, Jan. 1999. https://doi.org/10.1109/RAMS.1999.744151

[105] R.S. Pressman and B.R. Maxim, Software Engineering: A Practitioner's Approach, eighth edition, McGraw-Hill Education, 2014.

[106] Y. Gao, W. Dong, W. Wu, C. Chen, X.-Y. Li, and J. Bu, "Scalpel: Scalable preferential link tomography based on graph trimming," IEEE/ACM Transactions on Networking, vol.24, no.3, pp.1392–1403, June 2016. https://doi.org/10.1109/TNET.2015.2411691

[107] W. Dong, Y. Gao, W. Wu, J. Bu, C. Chen, and X.-Y. Li, "Optimal monitor assignment for preferential link tomography in communication networks," IEEE/ACM Transactions on Networking, vol.25, no.1, pp.210–223, Feb. 2017. https://doi.org/10.1109/TNET.2016.2581176

[108] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D.A. Maltz, and M. Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies," in Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.13–24, Aug. 2007. https://doi.org/10.1145/1282380.1282383

[109] H. Herodotou, B. Ding, S. Balakrishnan, G. Outhred, and P. Fitter, "Scalable near real-time failure localization of data center networks," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014), pp.1689–1698, Aug. 2014. https://doi.org/10.1145/2623330.2623365

[110] Y. Zhu, H. Okita, and S. Hanaoka, "Locating the cause of QoS performance decline in EPC," in Proceedings of 2015 IEICE General Conference, no.BS-3-33, pp.S-70–S-71, March 2015 (in Japanese).

[111] Y. Zhu, H. Okita, and S. Hanaoka, "Quality degradation localization in complex message processing networks," in Proceedings of 2017 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2017), pp.1–6, May 2017. https://doi.org/10.1109/CQR.2017.8289442

[112] I. Rish, M. Brodie, N. Odintsova, S. Ma, and G. Grabarnik, "Real-time problem determination in distributed systems using active probing," in Proceedings of 2004 IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), vol.1, pp.133–146, April 2004. https://doi.org/10.1109/NOMS.2004.1317650

[113] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," IEEE Communications Magazine, vol.53, no.2, pp.90–97, Feb. 2015. https://doi.org/10.1109/MCOM.2015.7045396

[114] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, and L.P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in Proceedings of 14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), pp.98–106, May 2015.

[115] X. Shang, Z. Li, and Y. Yang, "Placement of highly available virtual network functions through local rerouting," in Proceedings of IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2018), pp.80–88, Oct. 2018. https://doi.org/10.1109/MASS.2018.00022

[116] H. Okita, "PBB-VPN operations and management system," IEICE Technical Report, vol.109, pp.35–40, ICM2009-51, Kagoshima, March 2010 (in Japanese).

[117] H. Okita, C. Hasegawa, H. Hoshihara, M. Yano, and S. Hanaoka, "Large-scale EPC control-plane simulator -(1) modeling and evaluation-," in Proceedings of 2014 IEICE Society Conference, no.B-14-12, p.347, Sept. 2014 (in Japanese).

[118] C. Hasegawa, H. Okita, H. Hoshihara, S. Hanaoka, and S. Kudoh, "Large-scale EPC control-plane simulator -(2) a study on signaling load reduction-," in Proceedings of 2014 IEICE Society Conference, no.B-14-13, p.348, Sept. 2014 (in Japanese).

[119] H. Hoshihara, H. Okita, S. Hanaoka, and R. Tanaka, "Predictive congestion detection of attach signaling in LTE/EPC networks," IEICE Technical Report, vol.114, pp.1–6, ICM2014-32, LOIS2014-39, Fukuoka, Jan. 2015 (in Japanese).

[120] Y. Zhu, H. Okita, and S. Hanaoka, "A practical approach for network fault detection," in Proceedings of 2016 International Conference on Computing, Networking and Communications (ICNC 2016), pp.1–5, Feb. 2016. https://doi.org/10.1109/ICCNC.2016.7440710

[121] D. Ito and H. Okita, "A study on utilization of informational universal middleware for network analytics," in Proceedings of 2015 IEICE Society Conference, no.B-7-11, p.82, Sept. 2015 (in Japanese).

[122] D. Bernstein, "Containers and cloud: From LXC to Docker to Kubernetes," IEEE Cloud Computing, vol.1, no.3, pp.81–84, Sept. 2014. https://doi.org/10.1109/MCC.2014.51

[123] V. Medel, O. Rana, J.c. Bañares, and U. Arronategui, "Modelling performance & resource management in Kubernetes," 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), pp.257–262, Dec. 2016.

[124] Z. Zhong and R. Buyya, "A cost-efficient container orchestration strategy in Kubernetes-based cloud computing infrastructures with heterogeneous resources," ACM Transactions on Internet Technology, vol.20, no.2, pp.15:1–15:24, April 2020. https://doi.org/10.1145/3378447