A Study on Construction Method of Fault Tolerant Virtual Networks

A Study on Construction Method of Fault Tolerant Virtual Networks

耐障害仮想ネットワークの構築法に関する研究

A dissertation presented
by

Katsuya Suto

submitted to
Tohoku University
in partial fulfillment of the requirements
for the degree of Doctor Philosophy

Supervisor: Professor Nei Kato

Department of Applied Information Sciences
Graduate School of Information Sciences
Tohoku University

January 2016

To My Family

Acknowledgments

I wish to thank my dear father who was always encouraging me to study and work hard and my dear mother who was always giving me kind words to heal my mind. I cannot imagine how much sacrifices they have made over the years to bring me up. I will not be able to repay any of their favors by any worldly means. I dedicate this thesis to you.

I am deeply indebted, grateful to my supervisor, Professor Nei Kato, for providing me this noble research environment and opportunity as a PhD student. His continuous guidance and warm support have helped me to do my research. I am also grateful to him for being such a patient and consistent supporter.

I would like to express my sincere gratitude to Professor Xiao Zhou and Professor Kazuyuki Tanaka for their great guidance which helped me to write this dissertation. Their meticulous comments have been an enormous help to me.

I owe my deepest gratitude to Associate Professor Hiroki Nishiyama for his continuous support and valuable advices. Without his guidance and persistent help, this dissertation would not have been possible.

This dissertation would not have materialized without the help of my colleague, Yuichi Kawamoto. I will never forget his valuable suggestions, discussions, and encouragement for the rest of my life. Also, I would like to offer my special thanks to Ms. Motoko Shiraishi and Mr. Kaoru Chiba for their kind support in my laboratory life. Also, I would like to express my gratitude towards past and present members of our laboratory for their continuous and friendly support over the years. Irreplaceable discussions with lab mates have helped me many times.

Acknowledgments are also given to Japan Society for the Promotion of Science (JSPS) for their strong support that enabled me to pursue the challenging route during my doctoral years.

Finally, I would like to express my sincere appreciation to my beloved wife, Yurie, for her complete support and patience with me to continue this research. Without her encouragement and support, my life would have been very hard.

Contents

Acknowledgments **i**

1 Introduction **1**

1.1 Background 1

1.2 Purpose of Research 3

1.3 Summary and Organization of the Thesis 4

2 Overview of Fault Tolerant Virtual Networks **7**

2.1 Introduction 7

2.2 Search Architecture using Virtual Networks 7

2.2.1 Overview of Envisioned Search Architecture 8

2.2.2 Basic Technologies in Virtual Networks 10

2.2.2.1 Virtual Network Topology 10

2.2.2.2 VM Mapping Strategy 10

2.2.2.3 Dynamic Alteration System 11

2.2.3 Issue of Virtual Networks 12

2.3 Fault Tolerant Virtual Network Technologies 12

2.3.1 Fault Tolerant Virtual Network Topology 13

2.3.2 Fault Tolerant VM Mapping 14

2.3.3 Fault Tolerant Dynamic Alteration 15

2.3.4 Contribution of Proposed Technologies 16

2.4 Summary 16

3 Virtual Network Topology based on Bimodal Degree Distribution **17**

3.1 Introduction 17
3.2 Models of Faults in Virtual Network 18
3.3 Optimized Degree Distribution 20
3.3.1 Degree Distribution Taxonomy 20
3.3.1.1 Random Network 20
3.3.1.2 Regular Network 22
3.3.1.3 Scale-Free Network 23
3.3.1.4 Bimodal Network 24
3.3.2 Optimal Setting of Bimodal Degree Distribution 26
3.4 Optimized Virtual Network Topology 29
3.4.1 Topology Assumption 30
3.4.2 Topology Optimization 31
3.5 Performance Evaluation 35
3.5.1 Performance Comparison in Global Network Connectivity 35
3.5.2 Performance Comparison in Local Network Connectivity . 37
3.5.3 Performance Comparison in Communication Efficiency . . 40
3.6 Summary 42

4 A Method to Construct Virtual Network based on Physical Network Information 45

4.1 Introduction 45
4.2 Failures in Physical Network 46
4.2.1 Impact of FPN on the Physical Network 47
4.2.2 Impact of FPN on the Virtual Network 50
4.3 Utilization of Physical Topology Information 51
4.3.1 HDVM Mapping Strategy based on Depth Information . . 52
4.3.2 LDVM Mapping Strategy based on Neighbor Information . 53
4.4 A Centralized Method to Construct Virtual Network 54
4.4.1 Initial Setup Procedure 55
4.4.2 VM Joining Procedure 57
4.5 Analysis on Virtual Network Connectivity 60
4.5.1 VM Removal Probability 61
4.5.2 Connectivity of FAT-VN 64

4.5.3 Connectivity of FAT-VN+ 67

4.5.4 Number of Available VMs 69

4.6 Performance Evaluation 70

4.6.1 Performance Comparison in Maximum Number of Removable VMs 70

4.6.2 Performance Comparison in Number of Available VMs . . 73

4.7 Summary 75

5 Fault Tolerant Big Data Mining Architecture 77

5.1 Introduction 77

5.2 Conventional Big Data Mining Architecture 78

5.3 Envisioned Big Data Mining Architecture 82

5.4 A Decentralized Method to Construct Virtual Network 84

5.4.1 ID Assignment 85

5.4.2 VM Joining Procedure 86

5.5 A Decentralized Data Management Method 89

5.5.1 Data Placement Procedure 90

5.5.2 Data Lookup Procedure 91

5.6 Performance Evaluation in Service Availability 92

5.6.1 Definition of Service Availability 92

5.6.2 The Impact of Virtual Network Construction Scheme on Service Availability 93

5.6.3 The Impact of Data Placement Scheme on Service Availability 94

5.7 Summary 95

6 Conclusion 97

Bibliography 101

Publications 109

Copyright Permissions 113

List of Figures

2.1 Our envisioned search architecture using virtual networks. 9
2.2 The impact of failures on the performance of virtual networks. . . 13

3.1 Degree distribution of random network. 21
3.2 Degree distribution of regular network. 22
3.3 Degree distribution of scale-free network. 23
3.4 Degree distribution of bimodal network. 25
3.5 Impact of network size on the parameters of bimodal degree distribution. 29
3.6 An example of assumed virtual network topology. 30
3.7 An example of virtual network after all HDVMs are removed by AVM. 32
3.8 An example of virtual network having 1 extra link. 33
3.9 An example of virtual network having multiple extra links. 34
3.10 Performance comparison in terms of global network connectivity in different network sizes. 36
3.11 Performance comparison in terms of local network connectivity in different scales of FVN. 39
3.12 Performance comparison in terms of communication efficiency in different scales of FVN. 41

4.1 An example of assumed physical network. 47
4.2 The network failures taxonomy based on scale and probability. . . 49
4.3 The impact of FPN on the virtual network that is constructed based on random manner. 51

4.4 An example of our considered HDVM mapping strategy based on the depth information. 53
4.5 An example of our considered LDVM mapping strategy based on the neighbor information. 54
4.6 An analytical model of the impact of FPN on the virtual network that is constructed by the proposed scheme, and notations defined in the model. 62
4.7 Types of links between surviving VMs and removed VMs. 64
4.8 Connectivity improvement against FPN in FAT-VN+. 71
4.9 Number of removable VMs by FVM in post-FPN virtual networks. 72
4.10 Number of removable VMs by AVM in post-FPN virtual networks. 73
4.11 Assumed physical network for numerical calculation. 74
4.12 Number of available VMs in different FPN scenarios. 75

5.1 Architecture of conventional big data mining architecture. 78
5.2 Procedure of big data mining in the conventional architecture. . . 79
5.3 An example of our envisioned big data mining architecture and the procedure of data mining. 83
5.4 An example of virtual network and ID assignment in FA-Hadoop. 86
5.5 An example of data placement in our envisioned FA-Hadoop. . . . 91
5.6 An example of data lookup in our envisioned FA-Hadoop. 92
5.7 Service availability in different network construction schemes. . . . 94
5.8 Service availability in different data placement schemes. 95

List of Tables

3.1 Performance comparison of networks. 26

4.1 The relationship between the depth of devices and the probability that a device is unable to communicate in the tree-based physical network. 52

4.2 Database of management server. 55

5.1 Network information list. 86

Chapter 1

Introduction

1.1 Background

At the same time as Internet became an important infrastructure of our society, Transportation Control Protocol (TCP)/ Internet Protocol (IP) network architecture was developed as a standard architecture in the world. Recently, due to its good utility, TCP/IP network is also used for providing other Information and Communication Technology (ICT) services. For instance, Next Generation Network (NGN) utilizes TCP/IP network instead of the telephone networks. Additionally, in the smart society plan, it is expected that all services in human society will be provided via TCP/IP network. However, TCP/IP network cannot support a variety of requirements since it is specialized in the stability of end-to-end communications. This gap between the requirement of services and the capability of network results in the decrease of user satisfaction. Therefore, the TCP/IP network is at an important turning point on the path for reaching the goal of realizing the smart society based on ICT services.

A virtual network technology is used for a new-generation network architecture, such as Content-Centric Networking (CCN) and Information-Centric Networking (ICN). In the new architecture, a virtual network is used instead of the IP network in order to establish the data communication between any devices. Additionally, by adequately utilizing the resources in physical networks, the virtual network technology can construct optimal virtual network for satisfying various kinds of requirements. Moreover, since the virtual network technology constructs multiple virtual networks in a physical network, it can provide multiple ICT services in a physical network. This phenomenon is dubbed as horizontal integration of ICT services and is important for the future big data based society. Consequently, the research on virtual network has attracted much attention.

For a new-generation network architecture, we must construct the virtual network by considering the resources of the whole network. However, the considered network failures become complex by the virtualization of the whole network. For instance, we should consider the breakdown of communication devices, communication link disruption, attacks, and disaster. Since these kinds of failures significantly affect the performance of virtual networks, this work investigates a fault-tolerant virtual network technologies, which will enable the achievement of high communication performance against every conceivable failures. In Japan, the National Institute of Information and Communications Technology (NICT) has been developing a disaster-resilient virtual network in order to provide ICT services in the environment where large-scale disasters such as The Great East Japan Earthquake and Tsunami occur. Additionally, some researchers tackle the research issue on fault-tolerant virtual networks. However, the research and development on the fault-tolerant virtual networks have only started just recently in countries all over the world. Since there are a lot of research and development issues, the world is in urgent need of a novel virtual network that can achieve high performance in the failure-prone environment.

1.2 Purpose of Research

In response to the social background that the fault-tolerance issue has emerged in virtual networks, this work aims to develop a method to construct a fault-tolerant virtual network, which can be used as a promising new-generation network architecture. In this work, we focus on the following failures.

- Failures in Virtual Network (FVN): In the case of FVN, Virtual Machines (VMs) cease to function due to the software bugs or Denial of Service (DoS) attacks. Due to such failures, the failed VMs are removed from the virtual network and the virtual network may be disrupted, which will cause the isolation of working VMs. Therefore, both computation and communication performance of the virtual network will decrease in FVN-prone environments. To cope with this issue, this work aims at optimizing virtual network topology by improving the tolerance to both software bugs and DoS attacks, considering the different characteristics of software bugs and DoS attacks.

- Failures in Physical Network (FPN): FPN includes mechanical troubles of communication and processing devices (e.g., router, switch, server, and user terminals) and physical link disruptions. Due to these failures, the VMs accommodated by the router or the VMs launched on a server cease to function, which results in the decrease of the number of working VMs. To deal with this issue, this work also aims at developing a method to construct the virtual network by considering the topological information in the physical network. Our developed method drastically improves the virtual network performance in FPN-prone environments.

Our objective also includes a principle to introduce the aforementioned virtual network to ICT services. Although the big data mining services have much attention and will be increasingly important in future ICT, the recent popular architecture still suffers from faults. To tackle this challenge, we envision the fully distributed big data mining architecture and, by introducing our construction methodology, it achieves the performance that provides enough service availability in future big data mining environment.

As above, this work proposes a failure-tolerant virtual network that substitutes the existing network architecture and contributes to the tremendous improvement of failure-tolerance in future-generation network architecture.

1.3 Summary and Organization of the Thesis

This chapter demonstrates the research background and motivations. The remainder of this thesis is organized as follows.

Chapter 2 presents recent researches on virtual network techniques and the issues. The chapter at first demonstrates an overview of our envisioned architecture that is based on virtual networks. Since this architecture manages the communication and computation devices in the whole network and it suffers from various kinds of failures, the related works on the fault-tolerant virtual network is also illustrated. Based on the objective and approach, we classify these related works and clarify the contribution of our proposed fault-tolerant virtual networks.

Chapter 3 presents the virtual network topology based on the bimodal degree distribution. We focus on the two types of failures in virtual network, i.e., attacks on VM (AVM) and failures of VM (FVM). Then, we define these failures as a mathematical model based on the degree of VMs. Since the promising technique is not capable of providing enough performance in the environment where these two failures occur, a unique virtual network topology based on the bimodal degree distribution, dubbed as FAT-VN, is envisioned for combating against both AVM and FVM. This chapter also illustrates that the topology based on bimodal degree distribution improves the tolerance to considered failures by considering the location of VMs.

Chapter 4 presents a method to construct the virtual network topology that is presented in chapter 3. First, we demonstrate that the topology in chapter 3 does not make it possible to ensure the tolerance to FPN. To cope with this issue, a novel method to construct the virtual network, dubbed FAT-VN+, is proposed. This method achieves higher tolerance to both FPN and FVN by utilizing the topological information in the physical network. In addition, a framework to evaluate the impact of the FVN on the service availability of the virtual networks that are constructed by the proposal and random manner is presented.

Chapter 5 presents decentralized big data mining architecture using the virtual network. At the beginning of this chapter, the single-failure point problem of conventional big data mining architectures is demonstrated. To tackle this problem, we envision the big data mining architecture based on virtual networks. Since all participating VMs execute both processing and management functions, this architecture makes it possible to resolve the single-failure point problem. The autonomous and distributed method to construct a virtual network is also provided to further improve of fault tolerance.

Finally, **Chapter 6** concludes this thesis.

Chapter 2

Overview of Fault Tolerant Virtual Networks

2.1 Introduction

In this chapter, we intend to provide a comprehensive overview of search architecture based on virtual networks and also related works to address the issue of fault-tolerance in virtual networks. This chapter firstly addresses the mechanism of search architecture based on virtual networks, and its applicability in the data-oriented communication. The remainder of this chapter describes the prior researches, which have investigated various kinds of failures in the considered architecture. We clearly demonstrate the contribution of our proposed method in this thesis, by comparing it with the related works.

2.2 Search Architecture using Virtual Networks

Here, we introduce our envisioned search architecture, which can provide efficient lookup by using virtual networks. Additionally, we demonstrate the basic technology for improving the performance of our envisioned architecture. Also, a research issue is provided at the end of this section.

2.2.1 Overview of Envisioned Search Architecture

The device-to-device and server-to-devices communications are the main traffic in conventional services such as voice call and Web. In other words, users need to look up a certain device, i.e., users need to know the location of devices, to receive the conventional services, i.e., device-oriented services. Therefore, we have always believed that the IP-based network, in which each device looks up its desired device by using IP addresses, is the absolute for the Internet. On the other hand, users provide/acquire a certain content to/from the Internet in the current ICT services, i.e., content-oriented services, such as content delivery and big data mining. Therefore, the conventional IP-based network is not suitable for the content-oriented services and a novel architecture, which provides a lookup mechanism based on content/data information, is essential for providing high-performance services. Indeed, many researchers attempt to develop such architecture, e.g, Content-Centric Networking (CCN) and Information-Centric Networking (ICN) [1, 2, 3, 4].

It is expected that the content/data will explosively increase in the future due to the following reasons: all services will be provided via Internet, the realization of big data mining service will generate numerous data in the network, and a lot of rich contents are already emerging currently with the high functionality of user terminals and the development of Internet of Things (IoT) devices. Therefore, it is required to develop a technology that makes it possible to provide/acquire numerous content/data with high efficiency and low cost. For this reason, we envision a search architecture using virtual networks.

Fig. 2.1 shows our envisioned search architecture using virtual networks. In

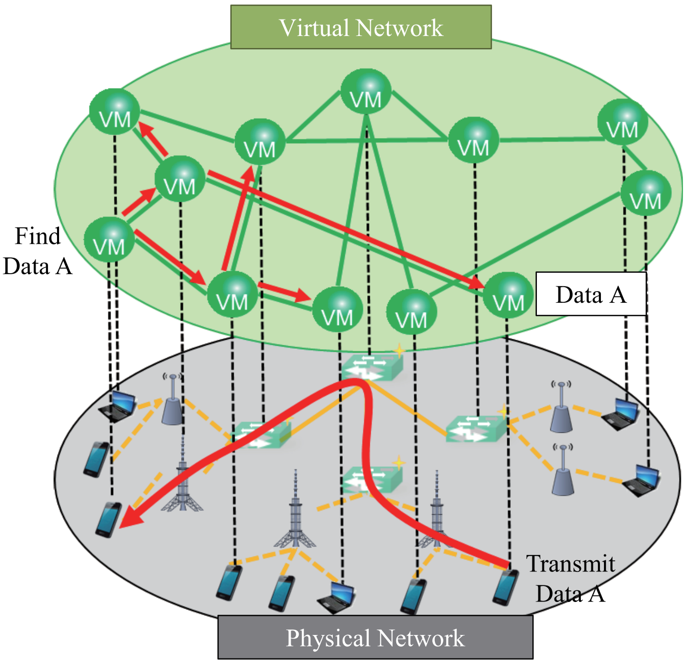


Figure 2.1: Our envisioned search architecture using virtual networks.

this architecture, users (or VMs) look up their desired data by using the virtual network. Each VM knows the IP address of neighbor VMs to find the data. When finding data, each VM, i.e., client VM, transmits a search query, which contains the ID of the desired data and its IP address, to its neighbor VMs, and then they forward the received query to their neighbors if they do not have the required data. This procedure continues until the desired data is found. When the VM that has the desired data receives the search query, it transmits data to the client VM via the physical network.

In the conventional IP-based architecture, a central VM stores the location of data, i.e., the IP address of the VM that has the data, and each client VM asks the location of data to the central VM. Since the central VM needs to get the location of data periodically, this architecture drastically decreases the service availability when there are a lot of data in the network. In contrast to this, in our envisioned architecture, each VM stores only the IP address of its neighbor VMs and does not need to know the location of data. Therefore, this architecture maintains high service availability even when the amount of data/content increases.

2.2.2 Basic Technologies in Virtual Networks

The technology for improving the performance of search architecture using virtual networks can be classified into the following technical components: virtual network topology, VM mapping strategy, and dynamic alteration system.

2.2.2.1 Virtual Network Topology

Virtual network topology is a factor that affects the performance of our envisioned search architecture. Virtual network topology is decided regardless of the physical network information. This means that the service providers construct an adequate virtual network topology that is able to satisfy their requirements or objectives. The existing research on virtual network topology aims to achieve high search efficiency by reducing the average hop count (or the diameter of virtual network). The basic idea for improving search efficiency is to increase the number of links, i.e., degree, for each VM. However, this approach has a shortage because a longer time is needed to construct the virtual network and burst traffic occurs by transmitting search message to all VMs. Therefore, a lower degree setting is suitable for providing scalable service in the actual environment.

2.2.2.2 VM Mapping Strategy

VM mapping is the selection of a VM in the physical network as a VM in the virtual network. In other words, since this technology decides the relationship between the physical and virtual networks, this technology has a big impact on search performance. Although the service providers attempt to satisfy their requirements by considering the virtual network topology, the search performance, i.e., search speed and success ratio, depends on the capabilities of the communication links and devices in the physical network, i.e., link speed, link capacity, and processing speed. Therefore, we need to design an adequate VM mapping strategy in order to satisfy the requirements.

There are two types of VM mapping strategies, which are described as follows: (i) VM mapping strategy based on the given resources and (ii) VM mapping strategy based on dynamic resources. In case of (i), the capability of VMs and that of links between any VMs are pre-defined by the network operators. Thus, the service providers consider the optimization of VM mapping based on the given information. On the other hand, the strategy (ii) is used in the scenario where service provider and network operator co-operate to provide services or the network operator provides services. In this scenario, the service provider requests to change the utilization of physical resources and dynamically control the capability of VMs and links. Therefore, the service provider can integrate the VM mapping strategy and resource allocation for improving the performance of services.

2.2.2.3 Dynamic Alteration System

The performance requirement should change in order to satisfy user requests, which periodically change according to the situation. The capability of physical VMs and links also change in different network environments, e.g., failure of VMs. Therefore, we must alter both the virtual network topology and VM mapping in order to satisfy the service requirements when the situation or environment changes.

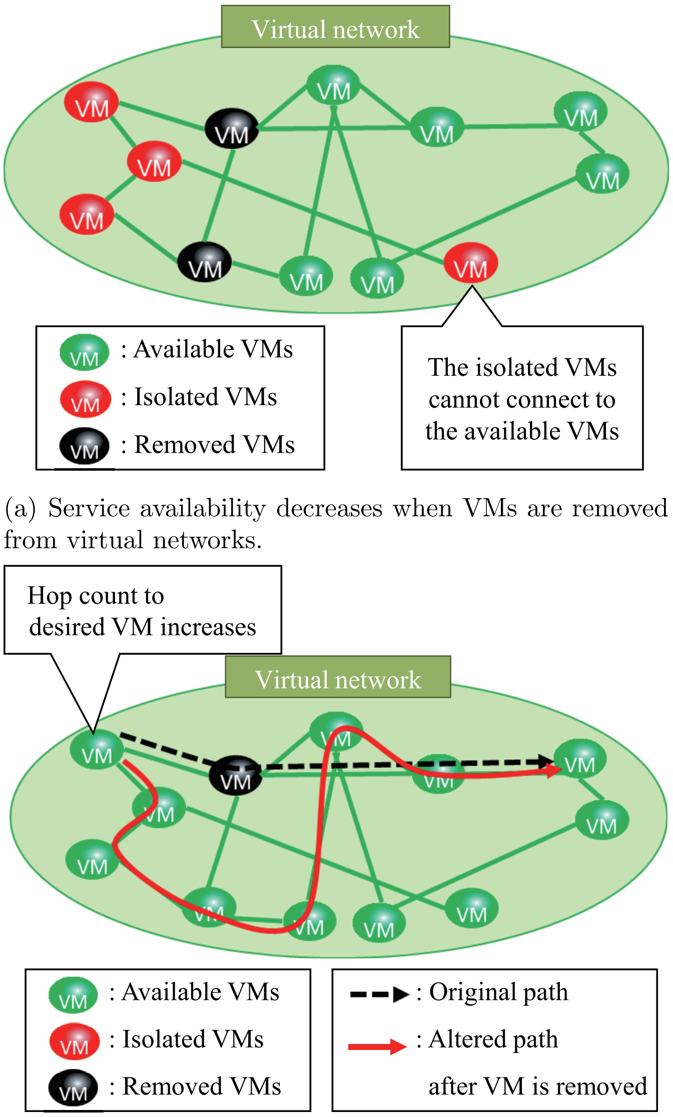
Additionally, “perception” and “intelligence” functions are required for realizing the dynamic alteration. The “perception” function is used to collect information used for observing the situation and environment in both the virtual and physical networks. On the other hand, the "intelligence" function is used to obtain the optimal states of virtual network topology and VM mapping in real-time by mining the collected information, i.e., big data. When the optimal state is different from the current state, the service provider attempts to alter the virtual network.

2.2.3 Issue of Virtual Networks

Although the search architecture using virtual network with the aforementioned technologies achieves high performance and satisfies the requirement, VMs cease to function due to FVN and FPN. This indicates that such VMs are removed from the virtual network and some surviving VMs are isolated form the virtual network, i.e., the VMs that do not cease to function lose the virtual links to the neighbor VMs. In this case, since the architecture cannot find such isolated VMs, the service availability drastically decreases as shown in Fig. 2.2(a). The removal of VMs also decreases the search efficiency because the hop count between VMs increases when some VMs are removed as shown in Fig. 2.2(b). Consequently, we need to address the issue of failure-tolerance, which is essential for a high-performance search architecture using virtual network.

2.3 Fault Tolerant Virtual Network Technologies

This section presents existing technologies for improving the fault tolerance of virtual networks. Additionally, by comparing the characteristics of existing technologies, we clarify the contribution of our proposed method.


Virtual network
VM
: Available VMs
: Isolated VMs
: Removed VMs
The isolated VMs cannot connect to the available VMs
Hop count to desired VM increases
Virtual network
: Original path
: Altered path after VM is removed

(a) Service availability decreases when VMs are removed from virtual networks.

(b) Hop count to desired VM increases when VMs are removed from virtual networks.

Figure 2.2: The impact of failures on the performance of virtual networks.

2.3.1 Fault Tolerant Virtual Network Topology

Many fault tolerant virtual network topologies have been investigated so far. These technologies are able to guarantee the connectivity of virtual network based on mathematical theories, which can be classified into the graph theory and complex network theory. The fault tolerant virtual network topologies based on graph theory have been proposed in [5, 6, 7, 8]. Ulysses [5] is a protocol that constructs a robust, low-diameter, low-latency virtual network topology based on the butterfly topology. CayleyCCC [6] is a virtual network topology based on the Cayley graph. The authors mathematically show that CayleyCCC can improve the connectivity against FVMs in comparison with Ulysses. The de Bruijn graph is also introduced to the virtual network topologies in [7, 8]. It is shown that the virtual network based on the de bruijin graph can achieve high tolerance to FVMs while keeping lower average degree.

On the other hand, virtual network topology based on a complex network theory is also explored in many researches [9, 10, 11, 12]. In contrast to the graph theory, this approach, which does not need exact network topology, decides the network topology by using network characteristics, e.g., degree distribution and degree correlation. Cyclon [9] employs the normal distribution to construct a virtual network topology, which achieves a higher tolerance to both FVM and AVM. However, since the diameter of the virtual network topology is large, it is not adequate from the point of search efficiency. To cope with the issue, Phenix [10], LLR [11], and SRA [12] construct a virtual network topology which follows a power-law degree distribution. While it can drastically improve the search efficiency, it is intolerant to AVM. As described above, there does not exist a topology that achieves high tolerance to both AVM and FVM with high search efficiency.

2.3.2 Fault Tolerant VM Mapping

As mentioned above, research on virtual network topologies aims to improve the tolerance to FVN. In addition to FVN, we need to consider the improvement of tolerance to FPN. A single FPN can result in the failure of one or more VMs and virtual links as it effects all VMs with a mapping that spans over the whole network. To address this issue, numerous researchers have investigated fault tolerant VM mapping strategies [13, 14, 15, 16]. For improving the service availability of virtual network, the work conducted in [13] proposed a VM mapping algorithm referred to as RMap, by considering the failure of physical devices. Additionally, the authors in [14] investigated a VM mapping strategy that is tolerant to both device and link failures in the physical network. Also, one type of FPN is a natural disaster, which has a big impact on both the physical and virtual networks. Therefore, the works [15, 16] tackled the issue of disaster tolerance and they proposed a VM selection mechanism to improve the performance of post-disaster virtual networks. However, all strategies attempt to solve the issue based on the backup approach, which indicates that redundant VMs are allocated to virtual networks. Therefore, these strategies require higher cost to construct the virtual network.

2.3.3 Fault Tolerant Dynamic Alteration

The greatest advantage of virtual networks is its flexibility where it can dynamically change the topology and resources. Therefore, the fault-tolerance issue can be addressed by using this advantage, i.e., the dynamic alteration technology is imperative for improving fault-tolerance. As mentioned before, the dynamic alteration technology can be classified into “perception” and “intelligence”.

The works conducted in [17, 18] proposed “perception” methods, which predict faults in the virtual networks by using the collected physical information. With these methods, the service provider can know the exact information of both physical and virtual networks in real time. Therefore, based on such information, real-time control of the virtual topology and VM mapping is possible, to improve fault tolerance. Indeed, the work[19] proposed a fault tolerant dynamic alteration strategy, referred to as SVNE, which dynamically selects VMs according to the change of states and environments.

2.3.4 Contribution of Proposed Technologies

Through the aforementioned discussion, we can see that the following issues should be addressed: (i) A virtual network that is tolerant to both FVM and AVM while achieving high search efficiency, (ii) An optimal relationship between physical and virtual network for improving FPN tolerance without redundancy, (iii) A technology for improving the tolerance to both FVN and FPN. To cope with the issue (i), chapter 3 presents a novel virtual network topology, referred to as FAT-VN. Additionally, to address the concerns (ii) and (iii), we present a novel VM mapping strategy based on physical topology information, referred to as FAT-VN+. Furthermore, based on the developed technologies, we envision a novel big data mining architecture that achieves high service availability in failure-prone environments.

2.4 Summary

In this chapter, we first explained that the communication type of the current or future ICT services, i.e., data-oriented communication, cannot achieve high performance in the current IP-based architecture. Therefore, we have presented a search architecture based on virtual networks, which is suitable for lookup in the environments where large number of data/contents exist in the Internet. Then, we presented the main issue of virtual network and the related works, which have attempted to address methods to combat against failures in the envisioned architecture. Finally, we have clarified the contribution of our proposed technologies in this thesis, through the discussion on the issues found in the related works.

Chapter 3

Virtual Network Topology based on Bimodal Degree Distribution

3.1 Introduction

Chapter 2 revealed that more attention needs to be paid in the virtual network that is tolerant to FVN, as contemporary techniques fail to adequately combat against this threat. In this chapter, we present a virtual network topology which maintains its stability under the effect of FVN. The topology is referred to as FAT-VN.

First, this chapter mathematically defines FVN, which can be modeled by the removal probability of each VM based on the number of links. Based on the understanding of the effect of FVN on the virtual network, we compare the virtual networks, which can be classified by the degree distribution and we show that a virtual network that is based on bimodal degree distribution is the best solution to achieve higher tolerance to FVN while maintaining communication efficiency. Also, this chapter presents an appropriate topology for improving the FVN tolerance. By exploring both the degree distribution and topology, the tolerance of a virtual network can be optimized. Performance evaluation conducted through computer simulations demonstrates that our proposed virtual network topology, FAT-VN, substantially improves the performance under the effect of FVN in comparison with other contemporary virtual networks.

Some parts of the content in this chapter are presented in the following journal and transactions papers, which were written by the author of this dissertation.

- K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "THUP: A P2P Network Robust to Churn and DoS Attack based on Bimodal Degree Distribution," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 247-256, Sep. 2013.

- K. Suto, H. Nishiyama, S. Shen, and N. Kato, "Designing P2P Networks Tolerant to Attacks and Faults based on Bimodal Degree Distribution," *Journal of Communications*, vol. 7, no. 8, pp. 587-595, Aug. 2012.

3.2 Models of Faults in Virtual Network

Since some VMs are removed from virtual networks due to FVN, the surviving VMs are isolated and cannot connect to other VMs. Therefore, in this section, we define the FVN models to evaluate the effect of FVN on the connectivity (or stability) of virtual networks. We consider Failures of VMs (FVM) and Attacks on VMs (AVM) as FVN models [20]. These FVN can be modeled by removing VMs from the considered virtual network by taking into account that the probability of removing each VM is determined by the number of links connected to the VM (i.e., by the degree of the VM).

There are many reasons for FVM. For instance, VMs cease to function because of lost or deleted data and application, corrupted operating system, erroneous user operation, and so on. Since aforementioned failures occur randomly, a VM is randomly removed form the considered network regardless of the degree of the VM. Here, $K = \{k_1, k_2, ..., k_m\}$ is the set that the degree of VMs is arranged in an ascending order, where m denotes the maximum number of the set, i.e., maximum degree in the considered network. There exist VMs having any degree $k \in K$ in the virtual network and the ratio of VMs having degree k, i.e., the degree distribution of the virtual network is denoted as p_k. Let s_k be the probability that a VM having degree k survives after some VMs are removed from the network due to FVM. $\sum_k s_k p_k$ indicates the fraction of VMs that have survived after some VMs are removed by FVM. Hence, the VMs removing probability, f_k^{FVM}, is defined by the following equation.

$$f_k^{\mathrm{FVM}} = 1 - \sum_{k \in K} s_k p_k, \tag{3.1}$$

where s_k is the same value for any k.

In case of the AVM such as DoS attacks and distributed DoS attacks, the probability of removal of each VM should be proportional to the degree of the VMs. Since the removal of the VMs having high degree substantially degrades the network connectivity, malicious attackers may disrupt the function of the VM that has the highest degree in order to degrade the performance of the virtual networks. Consequently, the VM departure model, f_k^{AVM}, can be formulated as follows.

$$f_k^{\mathrm{AVM}} = \begin{cases} 1, & \text{for} \quad k > k', \\ f_{k'}^{\mathrm{AVM}}, & \text{for} \quad k = k', \\ 0, & \text{for} \quad k < k', \end{cases} \tag{3.2}$$

where, k' denotes the maximal degree of the surviving VMs. Note that VMs are removed randomly regardless of the degree of the VM if the degree distribution of the virtual network is the same value for any k. In such virtual network, the model of AVM can be expressed as follows.

$$f_k^{\mathrm{AVM}} = 1 - \sum_{k \in K} s_k p_k. \tag{3.3}$$

3.3 Optimized Degree Distribution

Since the FVN can be modeled based on the degree of VMs as mentioned before, an appropriate degree distribution of a virtual network should be investigated. Therefore, this section initially constructs taxonomy of virtual networks based on the degree distribution. Then, we present a bimodal degree distribution, which can ensure higher connectivity of virtual networks in the environments where both AVM and FVM occur, in comparison with other degree distributions.

3.3.1 Degree Distribution Taxonomy

We explore four specific networks, i.e., random, regular, scale-free, and bimodal networks, and compare their performance by analysis based on their degree distributions. Degree distribution is one characteristic of virtual networks, which indicates the probability distribution of degrees over the network. This characteristic is frequently used for quantitative evaluation of the failure tolerance.

3.3.1.1 Random Network

The network which follows the normal distribution is classified as a random network[21, 9]. The normal distribution is shown in Fig 3.1 and is defined by

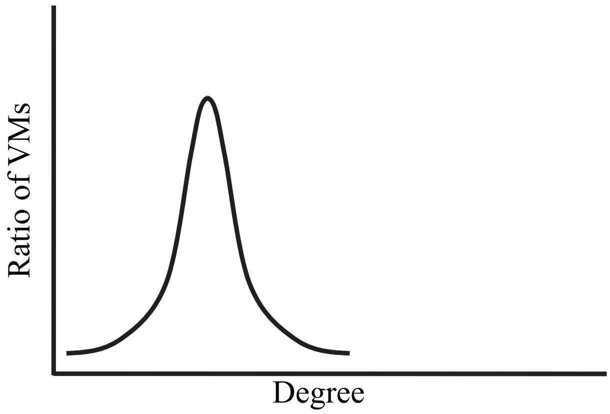


Figure 3.1: Degree distribution of random network.

the following equation.

$$p_k = \binom{N-1}{k} b^k (1-b)^{N-1-k}, \tag{3.4}$$

where b, N, and k denote the probability of connection to a VM, the total number of VMs, and the degree of VMs in the network, respectively. Random networks are basically constructed in such a way that a newly joining VM randomly selects a certain number of VMs.

Because the number of VMs having high degree (i.e., far above the average degree of the network) is small, the communication efficiency is low. Additionally, the tolerance to AVM is also low because VMs having higher degree are targeted by the malicious attackers and the removal of these VMs drastically degrades the connectivity. These shortcomings can be solved by increasing the average degree of the network. However, it is not a practical solution due to non-negligible overhead to establish connections with a large number of VMs.

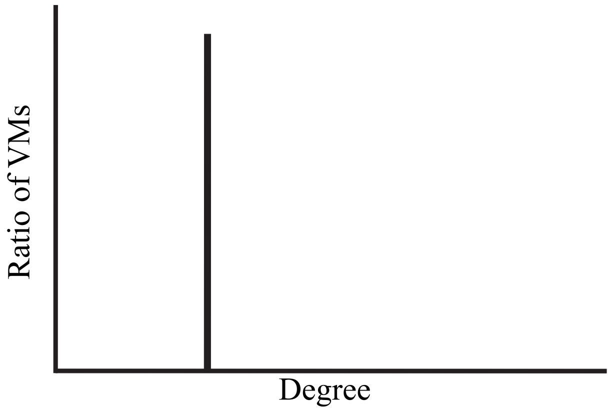


Figure 3.2: Degree distribution of regular network.

3.3.1.2 Regular Network

The network where all VMs have the same degree is referred to as a regular network. The regular network is often seen in overlay and virtual networking technologies [22, 23]. Fig. 3.2 shows the degree distribution of the regular networks and it can be defined by the following equation.

$$p_k = \begin{cases} N & \text{if} \quad k = \langle k \rangle \\ 0 & \text{otherwise} \end{cases}, \tag{3.5}$$

where $\langle k \rangle$ denotes the average degree in the network. The regular network is constructed in such a way that a newly joining VM selects a certain number of VMs as its neighbors in order to equalize the degrees of each VM in the network. By connecting newly joining VMs to the VMs with a lower degree, the network can be approximated to a regular network.

The regular networks have the lowest communication efficiency because there is no VM having higher degree. On the other hand, the connectivity of the regular networks is high even when AVM and FVM occur since the number of lost links is smaller than other degree distribution.

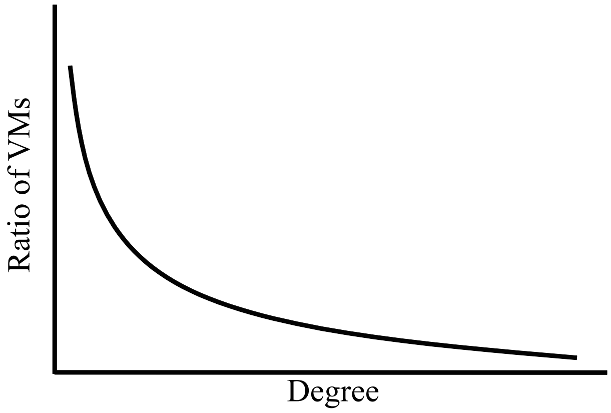


Figure 3.3: Degree distribution of scale-free network.

3.3.1.3 Scale-Free Network

Almost all real-world networks such as the Internet, World Wide Web (WWW) [24] and some social networks [25] are considered as instances of the scale-free network whose degree distribution follows power-law as shown in Fig. 3.3. The power-law degree distribution can be defined by the following equation.

$$p_k \propto k^{-\gamma}, \tag{3.6}$$

where γ is a parameter typically in the range ($2 \leq \gamma \leq 3$), which decides the range of degree. A Barabási-Albert (BA) model [26] is considered as the most famous method to construct a scale-free network. In this model, a newly joining VM stochastically selects a certain number of VMs to construct links.

The work conducted in [10] proposed a virtual network based on the power-law degree distribution, referred to as Phenix. This virtual network achieves high tolerance to FVM, as well as communication efficiency since the existence of several VMs having a quite high degree promotes the network connectivity even under the effect of FVM. However, since malicious attackers target the VMs having high degree, the performance of the networks drastically decreases when

AVM occurs [27, 28, 29].

The aforementioned taxonomy indicates that there is no degree distribution that achieves high tolerance to both AVM and FVM while keeping high communication efficiency. Therefore, we must derive an appropriate degree distribution.

3.3.1.4 Bimodal Network

Generally speaking, VMs having higher degree can promote the communication efficiency of virtual networks because they can reduce the diameter of the networks [30]. However, VMs having higher degree than other VMs become potential targets of malicious attackers because their breakdown affects the whole network and drastically degrades the network connectivity. Thus, a broader degree distribution such as scale-free distribution is intolerant to AVM. Indeed, the breakdown of a few high degree VMs will disrupt the network, which results in the isolation of VMs [30]. In order to construct a virtual network that is tolerant to AVM, a regular network with all VMs having a constant degree is the best choice [31]. While a regular network cannot achieve high communication efficiency, it is possible to improve the communication efficiency by increasing the average degree. However, to achieve a higher average degree is difficult in large-scale virtual networks since a significant amount of overhead for installation and operation is required.

From the aforementioned discussion, we can obtain the following knowledge for improving the tolerance to both AVM and FVM while keeping high communication efficiency: (i) the limitation of the type of VMs (i.e., the limitation of degree) and (ii) the existence of VMs having higher degree, and limitations of their degree and number. In order to satisfy these conditions, we use a bimodal network, which has a bimodal degree distribution. This network has mixed features from both regular and scale-free networks to exploit their benefits in the maximum way possible. As shown in Fig. 3.4, in bimodal degree distribution,

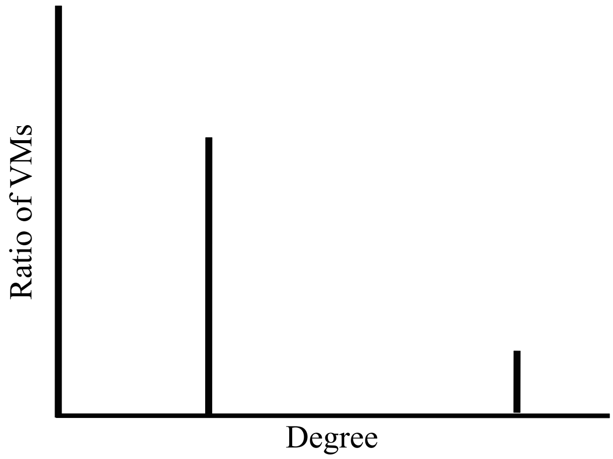


Figure 3.4: Degree distribution of bimodal network.

there exist two types of VMs, i.e., many VMs having lower degree, referred to as LDVM, and a few VMs having higher degree, referred to as HDVM. A bimodal network can achieve high tolerance to AVM since the maximum degree of the bimodal network is lower than that of the scale-free network, which results in smaller network disruption attributed to AVM. In addition, communication efficiency of bimodal networks is higher than regular networks since HDVMs reduce the diameter of networks. Consequently, we can conclude that bimodal networks inherit tolerance to the AVM from regular networks and communication efficiency from scale-free networks.

Table 3.1 summarizes the characteristics of each network. Bimodal networks have no weak point while other networks have at least one weak point. This characteristic is very important for real virtual networks. Therefore, we study a method to construct a virtual network based on the bimodal degree distribution. The study of the bimodal degree distribution is carried out in complex network theory and virtual network field.

In complex network theory, [32] is the first work on the bimodal degree distribution and it mathematically analyzed the loss of connectivity under random

Table 3.1: Performance comparison of networks.

	FVM Tolerance	AVM Tolerance	Communication Efficiency
Random Network	Good	Not Good	Bad
Regular Network	Best	Best	Bad
Power-law Network	Good	Bad	Best
Bimodal Network	Better	Better	Better

VM removal such as FVM. Tanizawa et al. [33] extended the mathematical model in the work [32] to analyze the tolerance to higher degree VM removal such as AVM. The work conducted in [34] revealed the reason why the bimodal degree distribution achieves higher tolerance to both FVM and AVM. Sonawane et al. proposed an algorithm, which is used to change any degree distribution to the desired bimodal degree distribution [35].

In virtual network field, the work conducted in [36, 37] proposed a mathematical model to evaluate the connectivity of virtual networks by assuming that networks have a bimodal degree distribution. In [38], the author presented an emergence model of virtual network based on bimodal degree distribution. The condition that virtual networks are tolerant to AVM and FVM is presented in [39]. Although a bimodal degree distribution is well researched by many researchers so far, there is no study on an appropriate virtual network topology that is based on bimodal degree distribution. Therefore, an optimal parameter for bimodal degree distribution and an appropriate topology based on the parameter settings are required.

3.3.2 Optimal Setting of Bimodal Degree Distribution

A virtual network based on the bimodal degree distribution makes it possible to improve the performance in the environment where FVN occurs. However, its performance depends on the parameters such as the number of HDVMs (or

LDVMs) and the degree of HDVMs (or LDVMs). Therefore, we present optimal parameter settings for the bimodal degree distribution.

According to the work [40], optimal parameters for maximizing the sum of tolerance to AVM and FVM in bimodal degree distributions are given as follows. In bimodal degree distribution, there are only two different types of VMs, i.e., HDVMs and LDVMs. HDVMs have a constant high degree k_{HD} and LDVMs have a constant low degree k_{LD}. The relation between k_{HD} and k_{LD}, which optimizes tolerance to both AVM and FVM, is expressed with the total number of VMs, N, as

$$k_{\mathrm{HD}} = \sqrt{N k_{\mathrm{LD}}}. \tag{3.7}$$

By using the above equation with the default value of k_{LD}, we can decide an ideal degree of HDVM. HDVMs and LDVMs should satisfy this degree constraint in order to improve the tolerance to AVM.

On the other hand, since the total number of VMs, N, is the sum of the number of HDVMs, N_{HD}, and the number of LDVMs, N_{LD}, the value of N can be represented as

$$N = N_{\mathrm{HD}} + N_{\mathrm{LD}} = rN + (1-r)N, \tag{3.8}$$

where r denotes the ratio of the number of HDVMs to the total number of VMs. While a higher value of r improves the tolerance to FVM, the virtual network is tolerant to AVM when we set r to a lower value. Since the value of r affects the tolerance of virtual networks, an optimal value of r, which maximizes the sum of tolerance to AVM and FVM, is derived from statistical analysis as follows.

$$\begin{aligned} r &= \left(\frac{A^2}{\langle k \rangle N} \right)^{\frac{3}{4}}, & (3.9) \\ A &= \left\{ \frac{2\langle k \rangle^2 (\langle k \rangle - 1)^2}{2\langle k \rangle - 1} \right\}^{\frac{1}{3}}, & (3.10) \end{aligned}$$

where $\langle k \rangle$ denotes the average degree of the virtual network, which can be calculated as

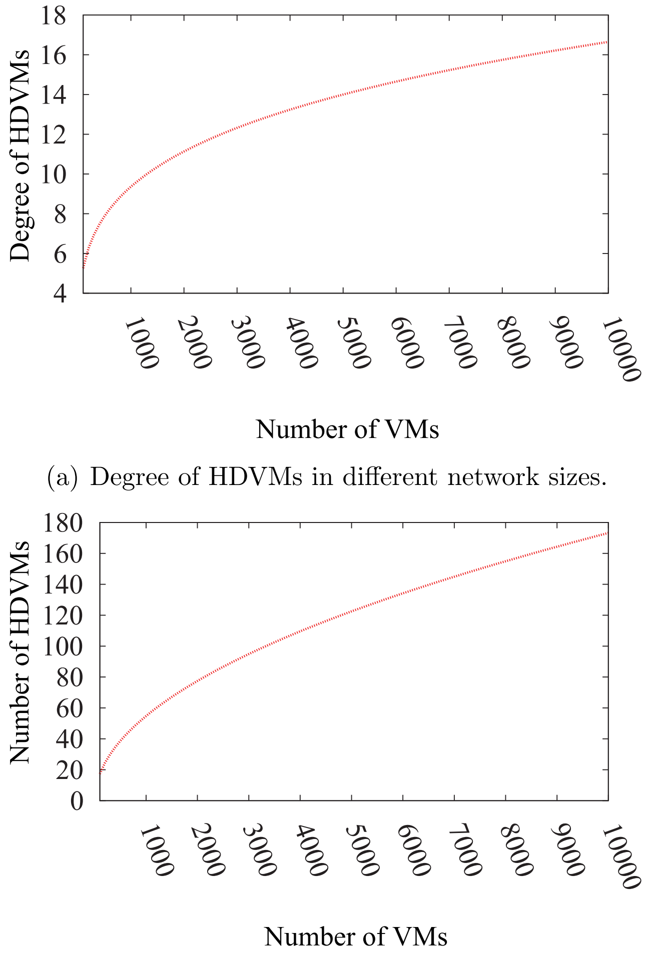
$$\langle k \rangle = \sum_{k \in K} k p_k. \tag{3.11}$$

Based on the aforementioned values, an optimal bimodal degree distribution, p_k, can be expressed with N and k_{LD}, as

$$p_k = \begin{cases} (1-r)N, & \text{if} \quad k = k_{\mathrm{LD}}, \\ rN, & \text{if} \quad k = \sqrt{N k_{\mathrm{LD}}}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

From the above equation, it is clear that the degree of LDVMs affects the network performance. If we set a higher value as k_{LD}, the virtual network achieves higher tolerance and communication efficiency. On the other hand, a lower value of k_{LD} can reduce the overheads of installation and maintenance. In this thesis, we assume that the value of k_{LD} is 3.

Fig. 3.5 shows the impact of network size on the parameters of the bimodal degree distribution, where the degree of LDVMs is set to 3. Fig. 3.5(a) and Fig. 3.5(b) demonstrate the degree of HDVMs and the number of HDVMs, respectively. From these graphs, we can confirm that both the degree of HDVMs and number of HDVMs rise with the increase of network size. Also, we can show that the ratio of growth decreases with the increase of network size. This



(a) Degree of HDVMs in different network sizes.

(b) Number of HDVMs in different network sizes.

Figure 3.5: Impact of network size on the parameters of bimodal degree distribution.

logarithmic-shaped function makes it possible to achieve higher tolerance to the AVM even if virtual network size is large.

3.4 Optimized Virtual Network Topology

Although the previous section presents an optimal degree distribution, i.e., bimodal degree distribution, it cannot decide a unique virtual network topology.

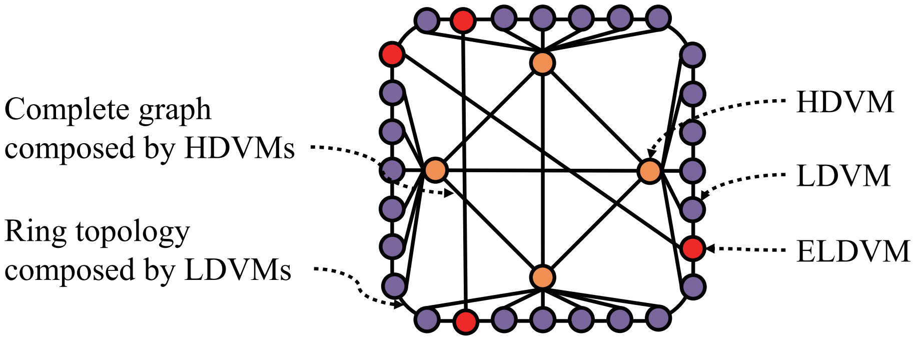


Figure 3.6: An example of assumed virtual network topology.

Since virtual network topologies may affect the tolerance to FVN, this section envisions an appropriate virtual network topology for improving the tolerance to FVN, which is referred to as FAT-VN.

3.4.1 Topology Assumption

Here, we demonstrate the assumed virtual network topology. Fig. 3.6 shows the assumed topology that is constructed based on the bimodal degree distribution. In the assumed topology, the links can be classified into three categories, i.e., the links between HDVMs, links between HDVMs and LDVMs, and links between LDVMs. Each HDVM connects to the other HDVMs to construct a complete graph since each HDVM acts as a hub to efficiently forward the data. Therefore, the total number of links between HDVMs, l_{HD}, is expressed as

$$l_{\mathrm{HD}} = \frac{N_{\mathrm{HD}}(N_{\mathrm{HD}}-1)}{2}. \tag{3.13}$$

Since the degree of HDVMs is limited, the total number of links between HDVMs and LDVMs, l_{HDLD}, is represented as

$$l_{\mathrm{HDLD}} = N_{\mathrm{HD}}\{k_{\mathrm{HD}} - (N_{\mathrm{HD}}-1)\}. \tag{3.14}$$

Additionally, the number of links between LDVMs, l_{LD}, is decided as

$$l_{\mathrm{LD}} = \frac{N_{\mathrm{LD}} k_{\mathrm{LD}} - l_{\mathrm{HDLD}}}{2}. \tag{3.15}$$

Here, l_{HDLD} is smaller than N_{LD} when $N \gg k_{\mathrm{LD}}$. This means that the number of links between HDVMs and LDVMs is limited. Therefore, we assume that each LDVM has one link to a HDVM, which can diminish the diameter of the virtual network. Additionally, we assume that each LDVM connects to other LDVMs to construct a ring topology, which gives an opportunity to get at least 2 links for each LDVM. Since the number of links that are required to construct a ring topology, l_{ring}, is equal to the number of LDVMs, N_{LD}, and $l_{\mathrm{LD}} > l_{\mathrm{ring}}$ when $k_{\mathrm{LD}} \geq 2$ and $N \gg k_{\mathrm{LD}}$, there exist some LDVMs having links to LDVMs, which are not involved in constructing the ring topology. In this thesis, such LDVMs and their links are called extra LDVMs (ELDVMs) and extra links, respectively. Assuming that the degree of LDVMs is 3, the number of extra links, l_{ELD}, and the number of ELDVMs, N_{ELD}, are decided as

$$\begin{aligned} l_{\mathrm{ELD}} &= l_{\mathrm{LD}} - l_{\mathrm{ring}} = \frac{N_{\mathrm{LD}}(k_{\mathrm{LD}} - 2) - l_{\mathrm{HDLD}}}{2}. && (3.16) \\ N_{\mathrm{ELD}} &= 2 l_{\mathrm{ELD}} = N_{\mathrm{LD}}(k_{\mathrm{LD}} - 2) - l_{\mathrm{HDLD}}. && (3.17) \end{aligned}$$

3.4.2 Topology Optimization

We assume the scenario where FVM occurs after all HDVMs are removed by AVM as a FVN model. Fig. 3.7 shows the network after all HDVMs are removed by AVM. Since the LDVMs (including ELDVMs) are randomly removed from the network by the FVM, the location of ELDVMs may affect the tolerance (i.e., network disruption probability). Consequently, we derive an appropriate location of ELDVMs for maximizing the tolerance to the considered FVN model,

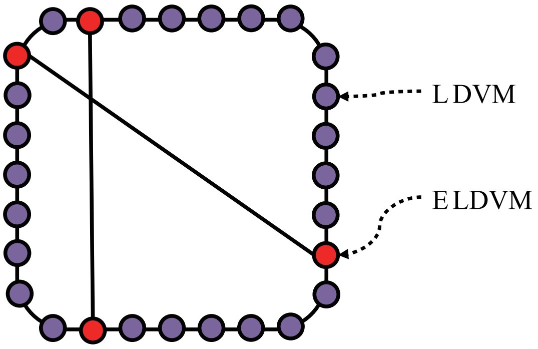


Figure 3.7: An example of virtual network after all HDVMs are removed by AVM.

i.e., minimizing the network disruption probability under the considered FVN model.

First, we are interested in the appropriate location of the set of ELDVMs in the virtual network having 1 extra link, i.e., the considered network consists of a ring topology and 1 extra link. Fig. 3.8 depicts an example of a virtual network having 1 extra link, i.e., there exist two ELDVMs. Let h_1 be the hop count between two ELDVMs via a path of ring topology, where $2 \le h_1 \le (N_{\mathrm{LD}}/2)$. The ring topology can be considered to be divided by ELDVMs into two segments, w_1 and w_2, which are defined as $h_1 - 1$ and $N_{\mathrm{LD}} - h_1 - 1$, respectively. Since the network can be disrupted by only removing two LDVMs, we consider the scenario where two LDVMs are removed in a random manner. The network disruption events can be classified into two cases, i.e., the network disruption by the removal of any of the LDVMs after the removal of one of ELDVMs, and the network disruption by the removal of any of the LDVMs after the removal of a non-ELDVMs. In the considered scenario, the network disruption probability can

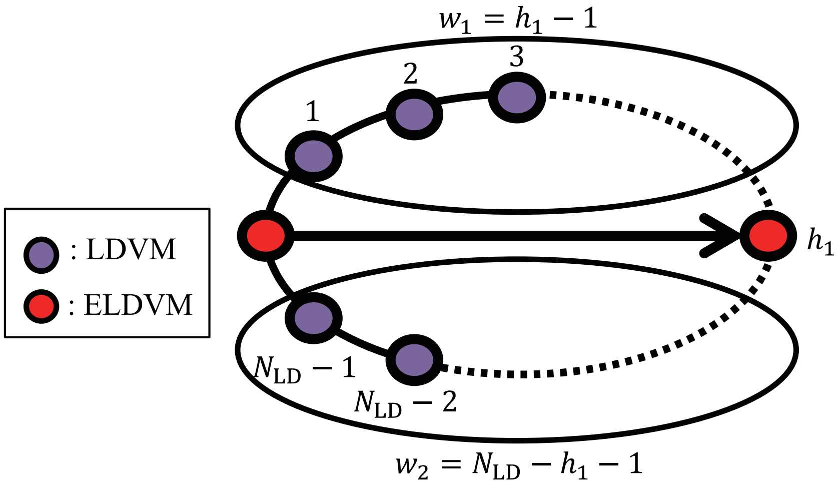


Figure 3.8: An example of virtual network having 1 extra link.

be expressed with h_1 as

$$\begin{aligned} P(h_1) &= \frac{2(N_{\mathrm{LD}}-3)}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)} \\ &\quad + \frac{(h_1-1)(h_1-1)-1)+(N_{\mathrm{LD}}-h_1-1)(N_{\mathrm{LD}}-h_1-1)-1)}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)} \\ &= \frac{2h_1(h_1-N_{\mathrm{LD}})+(N_{\mathrm{LD}}-2)(N_{\mathrm{LD}}+1)}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)}. \end{aligned} \tag{3.18}$$

Since it is clear from the above equation that the network disruption probability is a quadratic function of h_1, the optimal value of h_1, which minimizes the network disruption probability, can be derived as.

$$h_1^{\mathrm{opt}} = \operatorname*{argmin}_{h_1} P(h_1) = \frac{N_{\mathrm{LD}}}{2}. \tag{3.19}$$

Consequently, it can be concluded that LDVMs that are $(N_{\mathrm{LD}}/2)$-hops away should be selected as ELDVMs.

The aforementioned analysis can be easily introduced to the networks having multiple extra links as depicted in Fig. 3.9. In this network, non-ELDVMs are divided into $g = 2l_{\mathrm{ELD}}$ segments by l_{ELD} extra links. Assuming that two LDVMs

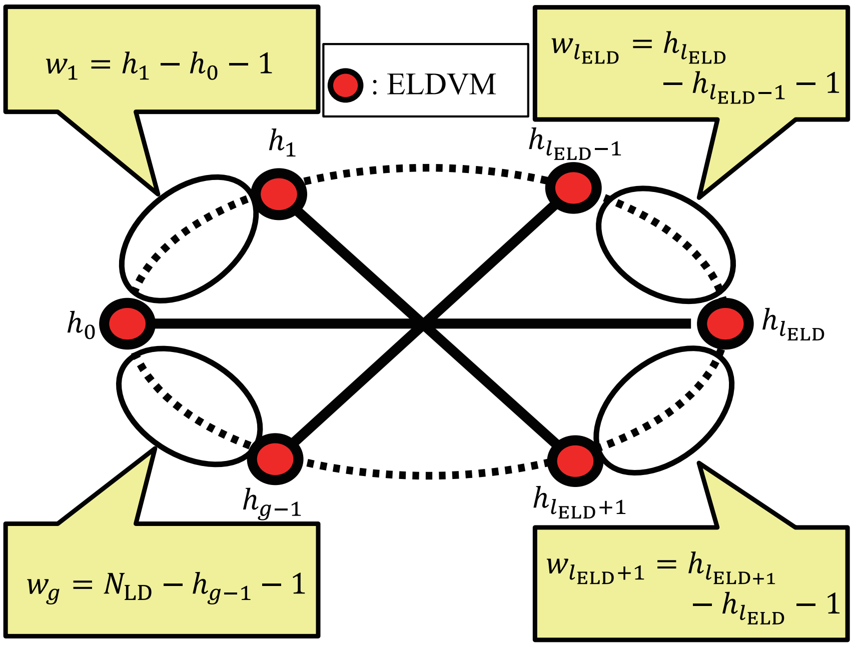

$w_1 = h_1 - h_0 - 1$
: ELDVM
$w_{l_{\mathrm{ELD}}} = h_{l_{\mathrm{ELD}}} - h_{l_{\mathrm{ELD}}-1} - 1$
h_1
$h_{l_{\mathrm{ELD}}-1}$
h_0
$h_{l_{\mathrm{ELD}}}$
h_{g-1}
$h_{l_{\mathrm{ELD}}+1}$
$w_g = N_{\mathrm{LD}} - h_{g-1} - 1$
$w_{l_{\mathrm{ELD}}+1} = h_{l_{\mathrm{ELD}}+1} - h_{l_{\mathrm{ELD}}} - 1$

Figure 3.9: An example of virtual network having multiple extra links.

are randomly removed, the probability of network disruption can be formulated in a similar way as before by using the size of each segment as the following equation.

$$\begin{aligned} P(h_l) &= \frac{\sum_{i=1}^{g-1}(w_i + w_{i+1}) + w_1 + w_g}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)} + \frac{\sum_{i=1}^{g} w_i(w_i-1)}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)} \\ &= \frac{(N_{\mathrm{LD}}-g) + \sum_{i=1}^{g} w_i^2}{N_{\mathrm{LD}}(N_{\mathrm{LD}}-1)}. \end{aligned} \tag{3.20}$$

Since there is a condition that the summation of w_i is a constant value equal to $N_{\mathrm{LD}} - g$, the network disruption probability will be the minimum value when the segment size is equal in all segments, i.e., $w_i = N_{\mathrm{LD}}/g - 1$.

3.5 Performance Evaluation

In this section, we evaluate the performance of the proposed FAT-VN through extensive computer simulations programmed in Ruby [41]. We compare the performance of FAT-VN with Phenix, which has the power-law degree distribution [10], and a Bimodal-random, which is constructed based on the bimodal degree distribution in a random manner. By comparing with the topology constructed in a random manner, the advantage of the proposed FAT-VN that considers the optimal network topology can be obviously understood. Three different metrics defined in [42, 43], i.e., (*i*) global network connectivity, (*ii*) local network connectivity, and (*iii*) communication efficiency, are used to evaluate the performance of the considered virtual networks.

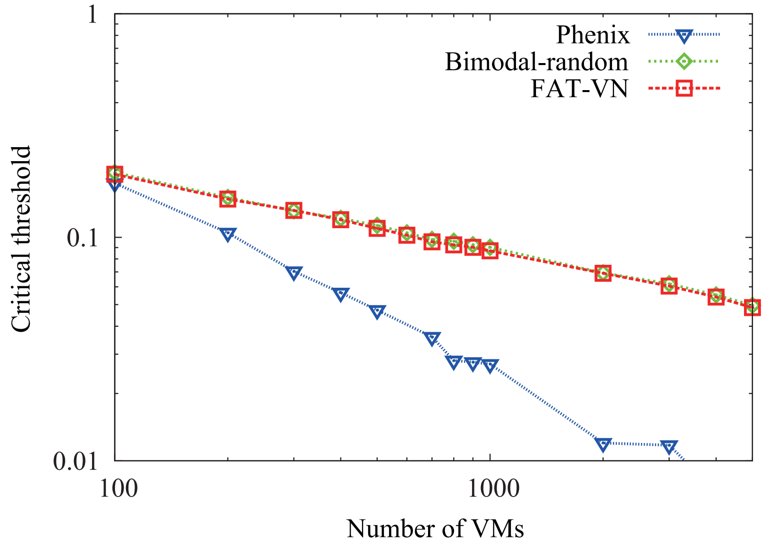
3.5.1 Performance Comparison in Global Network Connectivity

We evaluate the global connectivity of the virtual networks by using critical threshold, Q, which quantifies how many VMs can be removed from a network without disrupting the network. The value of Q is given by the following equation.

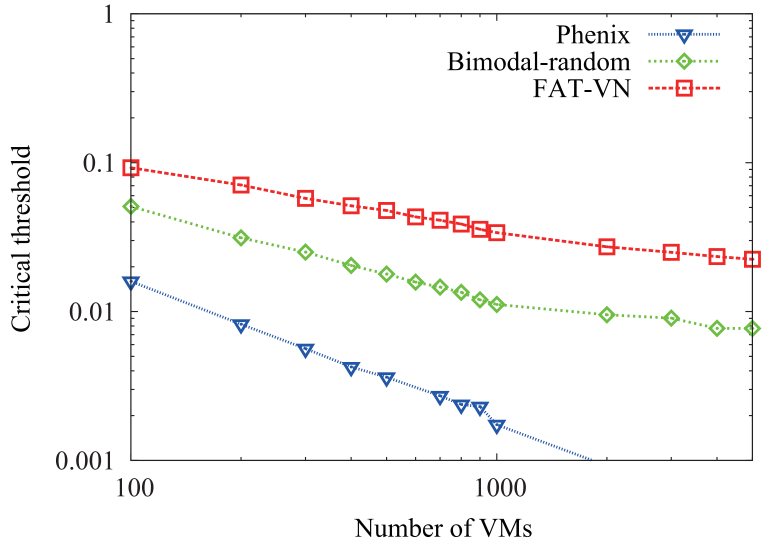
$$Q = \frac{N_{\mathrm{th}}}{N}, \quad (0 \leq f \leq 1), \tag{3.21}$$

where N denotes the number of VMs in the whole network, and N_{th} denotes the number of VMs which can be removed from the network without the network disruption. The value of Q closer to 1 indicates higher global connectivity.

Fig. 3.10 demonstrates the impact of network size on the critical threshold. We consider two FVN models, i.e., a scenario where the VMs are randomly removed by FVM and a scenario where FVM occurs after all HDVMs are removed by AVM. The average degree of each network is set to 3 and the number of VMs is varied from 100 to 5000.



(a) The impact of network size on critical threshold in case of FVM.



(b) The impact of network size on critical threshold in case of AVM and FVM.

Figure 3.10: Performance comparison in terms of global network connectivity in different network sizes.

Fig. 3.10(a) shows the critical threshold in the case of FVM. The networks which have bimodal degree distribution, i.e., the FAT-VN and bimodal-random, achieves higher performance, while the critical threshold of the Phenix, which follows the power-law degree distribution, drastically decreases with increase in the number of VMs. From this result, it is clear that the tolerance to FVM can be improved by utilizing bimodal degree distribution.

Fig. 3.10(b) demonstrates the critical threshold in the case of AVM and FVM. The performance of all networks is lower compared with in case of FVM since the HDVMs are preferentially removed in this case. Although the critical threshold of the Phenix drastically decreases, both the FAT-VN and bimodal-random maintain their performance even when network size is larger. This is because the number of HDVMs and degree of HDVMs are limited and set to adequate values based on the network size in the FAT-VN and bimodal-random, while the degree of HDVMs in the Phenix is too high when network size is large and the number of HDVMs is proportional to the network size. In addition, the FAT-VN achieves higher tolerance compared to the bimodal-random. This result clearly demonstrates that the proposed network topology is effective in the scenarios where both the AVM and FVM occur. Considering both scenarios, it is evident that FAT-VN is, indeed, the most suitable virtual network.

3.5.2 Performance Comparison in Local Network Connectivity

Local network connectivity is a suitable measure to evaluate the performance of the disrupted network consisting of the surviving VMs after some VMs are removed by FVN. If the network splits into a lot of smaller size networks, each VM cannot connect to other VMs in the small network and is unable to find the desired target resources, i.e., VMs or data. On the other hand, almost all VMs can connect to other VMs and find the resources if the network splits into a small number of larger size networks. Therefore, we evaluate the local network connectivity by using the maximum cluster ratio, S, which denotes the ratio of the number of surviving VMs in the maximum cluster to the total number of surviving VMs. This explains the reason that the maximum cluster ratio signifies the impact of the network disruption on the number of available VMs. The maximum cluster ratio, S, is defined by the number of VMs in the maximum cluster, N_{mc}, and the number of surviving VMs, N_{surv}, as

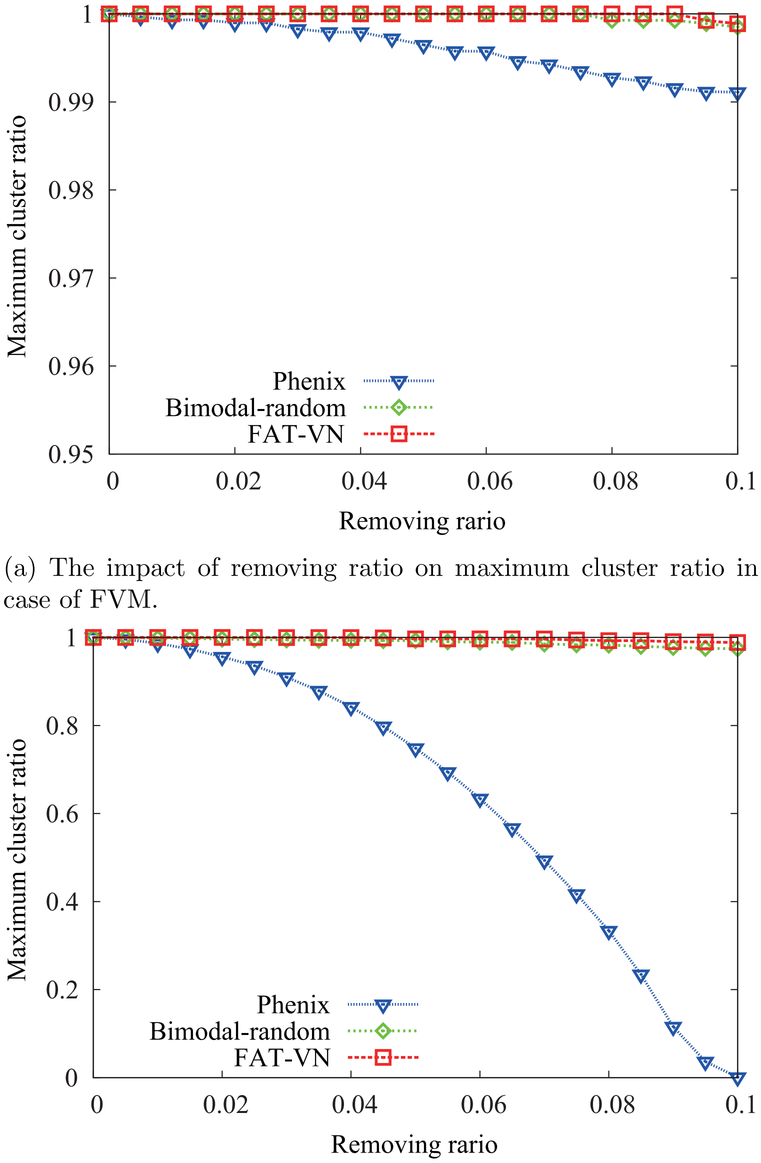
$$S = \frac{N_{\mathrm{mc}}}{N_{\mathrm{surv}}}, \quad (0 \leq S \leq 1). \tag{3.22}$$

The value of S closer to 1 implies a higher local network connectivity, which also means that many VMs remain free from the influence of network disruption.

Fig. 3.11 depicts the impact of VM removing ratio on the maximum cluster ratio, which is obtained by the following simulation environments. In this simulation, we consider two FVN models, i.e., a scenario where the VMs are randomly removed by FVM and a scenario where HDVMs are preferentially removed by AVM. The total number of VMs and the average degree are set to 10^3 and 3, respectively. Additionally, the VM removing ratio varies from 0 to 0.1 in 0.005 increments.

Fig. 3.11(a) demonstrates the maximum cluster ratio in case of FVM. The maximum cluster ratio in the Phenix is lower while the other networks achieve almost maximum performance. This is because the Phenix has multiple VMs having the lowest degree, i.e., 1, and these VMs are easily isolated from the network. On the other hand, because the FAT-VN and Bimodal-random have the bimodal degree distribution, a VM having the lowest degree connects to at least three VMs. Consequently, these networks achieve higher local network connectivity.

Fig. 3.11(b) shows the maximum cluster ratio in case of AVM. The Phenix falls to an extremely low maximum cluster ratio with a progressive increase of



(a) The impact of removing ratio on maximum cluster ratio in case of FVM.

(b) The impact of removing ratio on maximum cluster ratio in case of AVM.

Figure 3.11: Performance comparison in terms of local network connectivity in different scales of FVN.

the removing ratio. This is because almost all VMs having a lower degree connect to the VMs having a higher degree, which are preferentially removed in the AVM-prone environments. On the other hand, the FAT-VN and Bimodal-random achieve almost maximum performance, the same as in the case of FVM. This is because these networks have enough connectivity to not be disrupted even if there exist only VMs having a lower degree. From these results, it is clear that the bimodal degree distribution is also effective for local network connectivity.

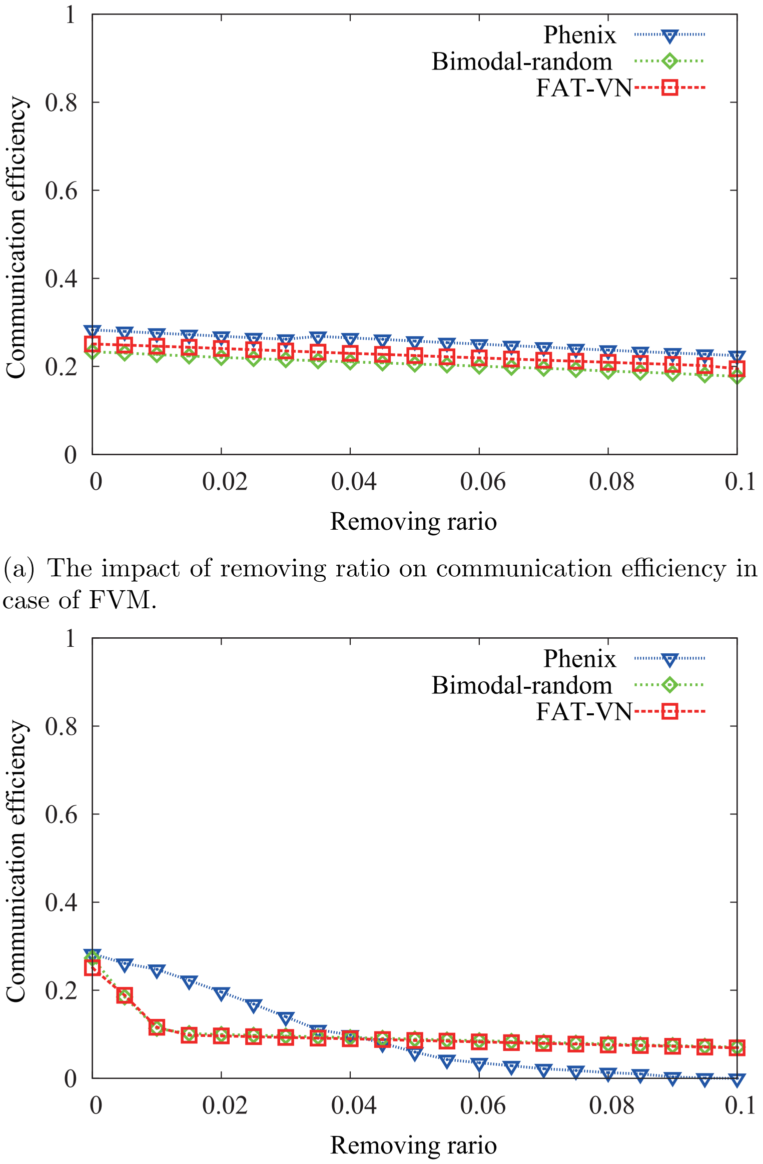
3.5.3 Performance Comparison in Communication Efficiency

From the aforementioned results, we can conclude that FAT-VN is the best choice on the issue of FVN because it achieves the highest tolerance in comparison to the other networks. From here on, we study virtual networks from the point of view of communication efficiency, and demonstrate that FAT-VN is as competent as the Phenix. The communication efficiency is defined by the following equation.

$$E = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}, \quad (0 \leq E \leq 1), \tag{3.23}$$

where d_{ij} is the hop count between the i^{th} and j^{th} VMs. Here, if there is no available virtual links between two VMs, the hop count between them is infinitely large, i.e., the inverse of the hop count is equal to zero. The maximum value of E is 1, which implies a complete graph, and a larger value indicates higher communication efficiency.

Since the VM removals due to the FVN increase the hop count and also degrade the communication efficiency, here we investigate the impact of the removing ratio on the communication efficiency, as shown in Fig. 3.12. This result is obtained by the simulation with the same setting as in the evaluation of the local network connectivity. A scenario where FVM happens and a scenario where AVM occurs are considered to quantify the communication efficiency in the FVN-prone environments. Each network consists of 10^3 VMs and the average degree is set to 3. The removing ratio varies from 0 to 0.1 in 0.005 increments.



(a) The impact of removing ratio on communication efficiency in case of FVM.

(b) The impact of removing ratio on communication efficiency in case of AVM.

Figure 3.12: Performance comparison in terms of communication efficiency in different scales of FVN.

Fig. 3.12(a) demonstrates the communication efficiency for different values of the removing ratio in case of FVM. The Phenix achieves higher communication efficiency regardless of removing ratio in comparison with the other networks, because there exist much more VMs having higher degree even when some VMs are removed by FVM. On the other hand, the communication efficiency of FAT-VN is higher than that of Bimodal-random because HDVMs construct a complete graph in FAT-VN.

The communication efficiency in case of AVM is shown in Fig. 3.12(b). The communication efficiency of FAT-VN drastically decreases up to 0.015 while the decreasing ratio becomes lower from 0.015. This is because a large number of links are removed up to 0.015 since the HDVMs are removed, and the number of removed links is small from 0.015 because the LDVMs are removed. In contrast to the FAT-VN, the communication efficiency of the Phenix gradually degrades because there are many VMs having a high degree. From 0.04, the communication efficiency of Phenix is lower than that of FAT-VN. This is because VMs having high degree are still removed from Phenix while VMs having low degree are removed from FAT-VN. From these results, it can be concluded that FAT-VN is suitable as a virtual network in FVN-prone environments.

3.6 Summary

In this chapter, we have investigated the optimal virtual network topology for improving both FVM and AVM tolerance. First, we have established stochastic formulas for FVM and AVM, which can be modeled in terms of removing VMs from the considered network by taking into account that the probability of removing each VM is determined by the degree. By creating a taxonomy of virtual networks based on the degree distribution, we have demonstrated that the bimodal degree distribution is effective to a construct virtual network that is tolerant to FVM and AVM. Additionally, an optimal network topology for improving tolerance under the effect of both FVM and AVM, i.e., FAT-VN, was also proposed. Through extensive computer simulations, we have verified the effectiveness of FAT-VN. In particular, we demonstrated that FAT-VN can offer high network connectivity, which increases the tolerance to FVM and AVM, thereby ensuring significantly higher communication efficiency in contrast with the existing virtual networks.

Chapter 4

A Method to Construct Virtual Network based on Physical Network Information

4.1 Introduction

In chapter 3, we presented an optimal virtual network topology, FAT-VN, that is able to maximize the FVN tolerance while keeping high communication efficiency. This chapter aims to propose a method to construct a virtual network for improving the connectivity against FPN. This method and the constructed virtual network are referred to as FAT-VN+.

This chapter initially discusses the impact of FPN on both physical and virtual networks. Also, we reveal that FPN tolerance drastically changes with the methods to a virtual network, and the reason why FAT-VN is intolerant to FPN. Then, we demonstrate that the key point for improving FPN tolerance of virtual networks is to utilize a physical network information, and also present two basic ideas to construct FPN tolerant virtual networks. Furthermore, a novel method to construct a virtual network using the basis ideas, FAT-VN+, is proposed. We construct mathematical models based on degree distribution in order to evaluate the performance of FAT-VN+ and FAT-VN. Based on the constructed mathematical models, the connectivity of FAT-VN+ against FPN is evaluated and the effectiveness of FAT-VN+ is also confirmed by comparing it with FAT-VN.

Some parts of the content in this chapter are presented in the following papers, which were written by the author of this dissertation.

- K. Suto, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "An Overlay-based Data Mining Architecture Tolerant to Physical Network Disruptions," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 292-301, Oct. 2014.
- K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "Toward Integrating Overlay and Physical Networks for Robust Parallel Processing Architecture," *IEEE Network*, vol. 28, no. 4, pp. 40-45, Jul.-Aug. 2014.
- K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "An Overlay Network Construction Technique for Minimizing the Impact of Physical Network Disruption in Cloud Storage Systems," in *Proc. of ICNC*, Feb. 2014, pp. 68-72.

4.2 Failures in Physical Network

In this section, we first describe the impact of FPN on devices in the considered physical network. Then, we introduce a taxonomy of FPNs, which classifies FPNs based on their probability and scale. Additionally, we mathematically define the impact of FPN on the virtual networks.

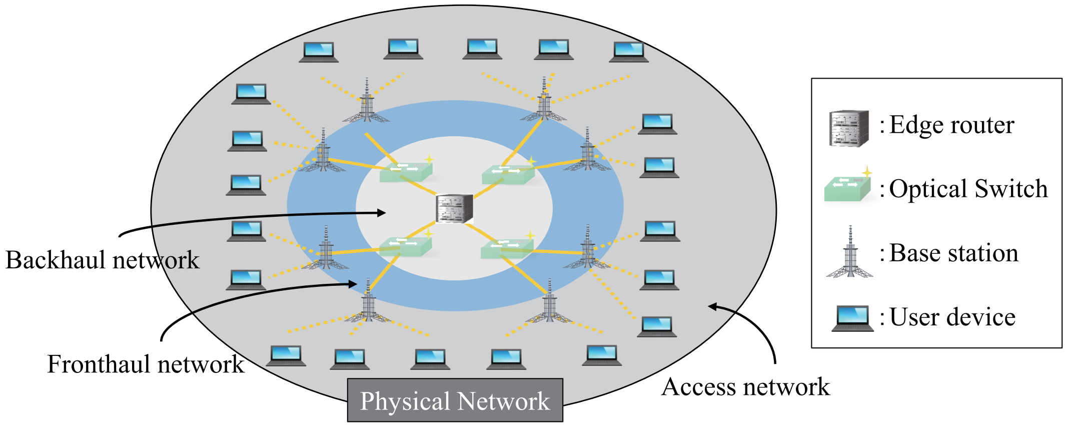


Figure 4.1: An example of assumed physical network.

4.2.1 Impact of FPN on the Physical Network

In this thesis, we assume that the physical network is tree topology. Fig. 4.1 shows the example of the assumed network, which is composed of access, fronthaul, and backhaul networks. In access networks, Base Station (BS) or Access Point (AP) provides communication services to multiple user devices by using mobile radio systems such as Long Term Evolution (LTE) [44] and Wireless Fidelity (WiFi) [45]. In these radio systems, users cannot connect to multiple BSs (or APs) simultaneously. Thus, it can be said that access networks are tree topology. Similarly, the common communication systems used in fronthaul and backhaul networks such as Passive Optical Network (PoN) [46] also construct tree topology. Therefore, we can assume that the considered physical network is tree topology.

In the tree topology, each child node connects to a single parent node while each parent node has links to multiple child nodes. Therefore, in the considered physical network, devices that connect to the failed device due to the FPN cannot communicate to the other devices. For instance, in Fig. 4.1, if the base station ceases to function due to the FPN, the user devices that are accommodated by the failed base station cannot communicate to the other devices in the entire physical network.

The number of devices that will be less likely to be able to communicate due to FPN, i.e., the scale of FPN, is different according to the failed devices and the probability that FPN happens also depends on the type of devices. Therefore, we explore the probability and scale of FPN in the entire physical network including the access, fronthaul, backhaul, core, and data center networks. Here, FPN taxonomy in data center networks is already demonstrated in the literature [47]. According to the work [47], device in physical networks may cease to function due to various reasons, i.e., DoS attacks, hardware troubles, software bugs, and so forth. For simplicity, this thesis mainly points out the hardware troubles and the software bugs of devices in physical networks.

Fig. 4.2 shows the FPN taxonomy on the entire network based on their scales and probability. Indeed, hardware troubles and software bugs are the main reasons that many user devices such as smartphones and laptops are prone to failures. The failure of the smartphones per annum is indicated to be around 10% in the report by Square Trade [48], and this probability is stated to be much higher than those associated with other network devices. Additionally, the user devices tend to be temporarily unable to communicate due to their mobility. In the future networking perspective, since Device-to-Device (D2D) communications such as Wi-Fi Direct and Long Term Evolution D2D (LTE D2D) are likely to become promising architectures for forming the user access networks [49, 50, 51], the failure of the user devices with D2D continuations has an impact on the other terminals/devices.

In access networks, user devices are rendered unable to communicate with others due to the failure of the connected BS. Since the current BSs accommodate few hundreds of user, the failure of BSs has a larger impact than that of user devices. According to our survey, the Mean Time Between Failures (MTBF) of the real BSs that is developed by the NTT DOCOMO is around $30,000$ hours [52].

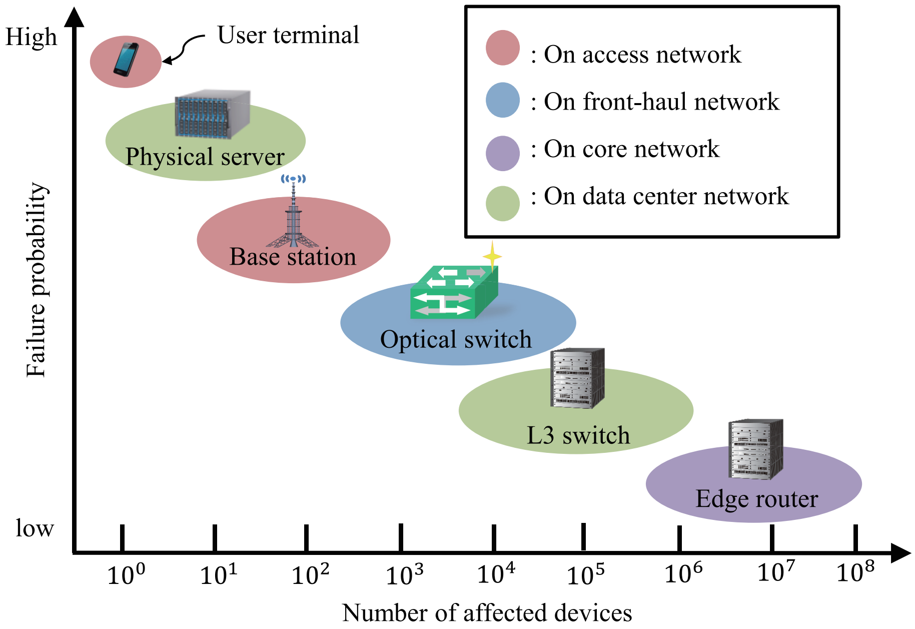


Figure 4.2: The network failures taxonomy based on scale and probability.

Consequently, the probability that the BSs cease to function due to the FPN is considered much less than that of user devices.

While fronthaul and backhaul networks concentrate data traffic from access networks, the core network associates the devices in these networks to provide communication services between these networks. Thus, the failure of L3 switches in the core network has the biggest impact on other devices. Additionally, the MTBF of the optical L3 switch that is used for both the backhaul and core networks ranges from $300,000$ to $800,000$ hours [53]. In general, because network operator deploys a higher reliable switch in the core network, the probability of the switch failing in the core network is lower than that in the fronthaul and backhaul networks.

On the other hand, the data center network is composed of hundreds of thousands of physical servers, where the number of servers will be increased in the future. In addition, since each physical server runs hundreds of virtual servers by using virtualization technologies, it is expected that the impact of the failure of L3 switches in the data center network is larger than that in the fronthaul and backhaul networks. Furthermore, the scale of the failure of the physical servers is situated between the user devices and BSs. Since the MTBF of the physical servers is around $6,000$ hours [54], the probability of physical servers failing is larger than that of BSs.

4.2.2 Impact of FPN on the Virtual Network

Since the devices that connect to the failed device are less likely to be able to communicate in the physical network, the FPN causes numerous VMs to be removed from the virtual network. The probability of removing each VM is determined by the virtual network construction schemes. Therefore, we focus on a virtual network that is constructed based on the random manner, i.e., FAT-VN, and investigate VM removal probability, f_k, which denotes the probability that a VM having degree k will be removed from the virtual network.

As described in chapter 3, FAT-VN is an optimal virtual network topology based on bimodal degree distribution, which is constructed based on the random manner. In other words, the degree and location of VMs in the virtual network is independent of the physical network information. Since VM is randomly selected regardless of the degree and location of devices in the physical network, the FPN causes VMs to be randomly removed from the virtual network regardless of their degree and physical location as shown in Fig. 4.3. Therefore, the VM removal probability in the FAT-VN, f_k^{FPN}, can be expressed with the number of removed nodes, $N^{\boldsymbol{R}}$, as follows.

$$f_k^{\mathrm{FPN}} = \frac{N^{\boldsymbol{R}}}{N}. \tag{4.1}$$

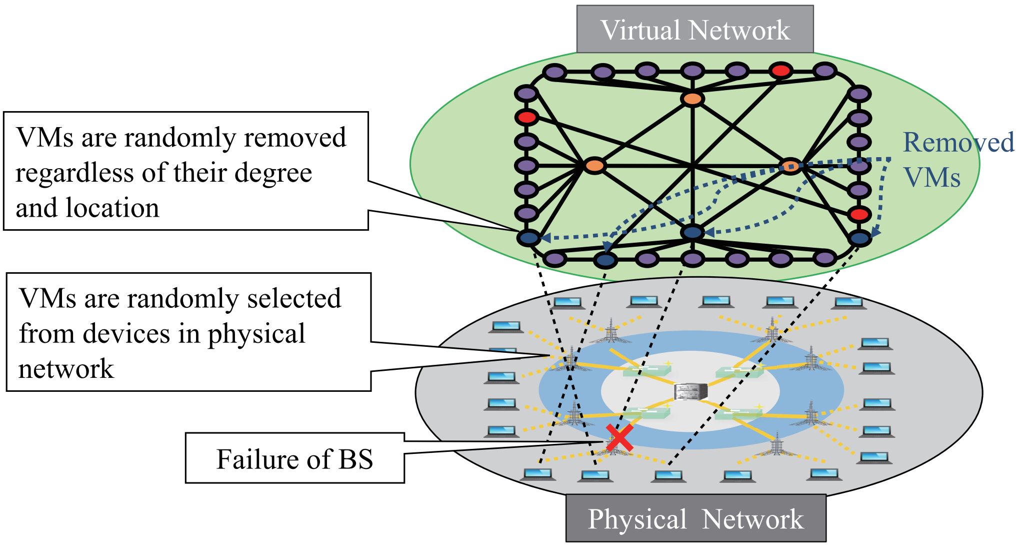

Virtual Network
VMs are randomly removed regardless of their degree and location
Removed VMs
VMs are randomly selected from devices in physical network
Failure of BS
Physical Network

Figure 4.3: The impact of FPN on the virtual network that is constructed based on random manner.

Since the numerous VMs are randomly removed from the virtual network in case of the large-scale FPN, the virtual network is prone to be disrupted. Therefore, we need to explore a method to construct a virtual network that achieves high connectivity against FPN.

4.3 Utilization of Physical Topology Information

In this section, we demonstrate the usage of physical topology information to construct a virtual network that is tolerant to the FPN. First, we show the usage of depth information in the physical network for selecting VMs having higher degree, i.e., HDVMs, in the virtual network. Then, a strategy to select VMs having lower degree, i.e., LDVM, is presented. This strategy can improve FPN tolerance of virtual networks by considering neighbor information of devices in the physical network.

Table 4.1: The relationship between the depth of devices and the probability that a device is unable to communicate in the tree-based physical network.

Type of device	Depth of device	Probability of FPN	Probability that device is unable to communicate
Edge router	$\delta = 0$	ϕ_0	$\Phi_0 = \phi_0$
Optical switch	$\delta = 1$	ϕ_1	$\Phi_1 = \phi_0 + \phi_1$
Base station	$\delta = 2$	ϕ_2	$\Phi_2 = \phi_0 + \phi_1 + \phi_2$
User device	$\delta = 3$	ϕ_3	$\Phi_3 = \phi_0 + \phi_1 + \phi_2 + \phi_3$

4.3.1 HDVM Mapping Strategy based on Depth Information

Since we assume that the physical network is tree topology, all child devices of the failed device are less likely to be able to communicate to the others in the physical network. Therefore, the probability that a device will be unable to communicate due to the FPN is proportional to the depth of the device in the physical network as shown in Table 4.1. Let ϕ_δ be the probability that a δ-depth device ceases to function, i.e., the probability of FPN of δ-depth device. The probability that a δ-depth device will not be able to communicate due to any FPN, Φ_δ, can be expressed with sum of ϕ_i for $i \leq \delta$, as follows.

$$\Phi_\delta = \sum_{i=0}^{\delta} \phi_i. \tag{4.2}$$

From the above equation, it is clear that the probability Φ_δ becomes higher with the increase in the depth of devices. Since the HDVMs are much important than LDVMs due to their high connectivity, the devices that are close to the root device, i.e., low depth, should be selected as HDVMs. This results in low probability that HDVMs will be removed by the FPN. Fig. 4.4 shows an example of the HDVM mapping strategy based on the depth information in physical network

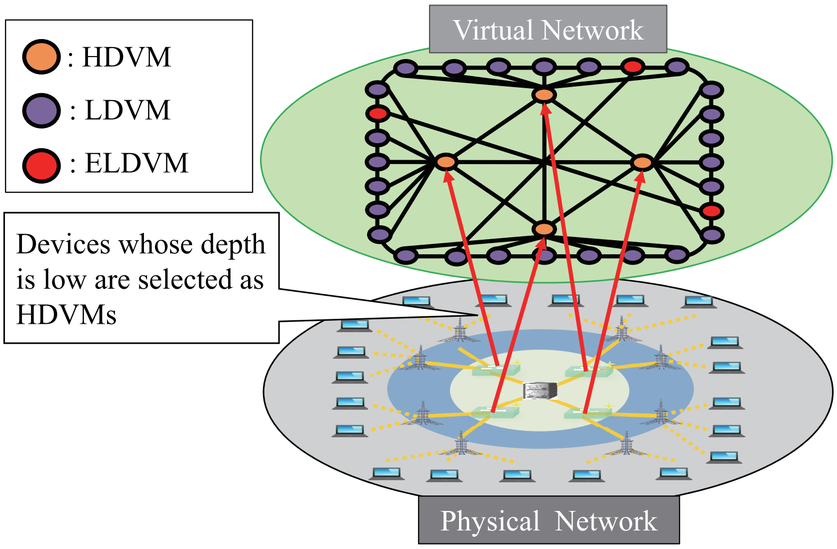


Figure 4.4: An example of our considered HDVM mapping strategy based on the depth information.

when the number of HDVMs is 4.

4.3.2 LDVM Mapping Strategy based on Neighbor Information

This subsection presents a LDVM mapping strategy. This mapping strategy utilizes neighbor information to achieve higher connectivity even when FPN occurs.

In order to keep the connectivity of the post-FPN virtual networks, it is required to reduce the link loss of the surviving VMs. Therefore, we need to obtain a condition of the post-FPN virtual networks, where each surviving VM connects to the other surviving VMs as far as possible. In other words, each removed VM has a lot of links to the other removed VMs as far as possible.

In order to approach this ideal condition, our strategy preferentially selects the devices whose hop count is low in the physical network as the neighbor LDVMs in the virtual network. With this selection strategy, the virtual network can be divided into sub networks that are composed of VMs running on devices that

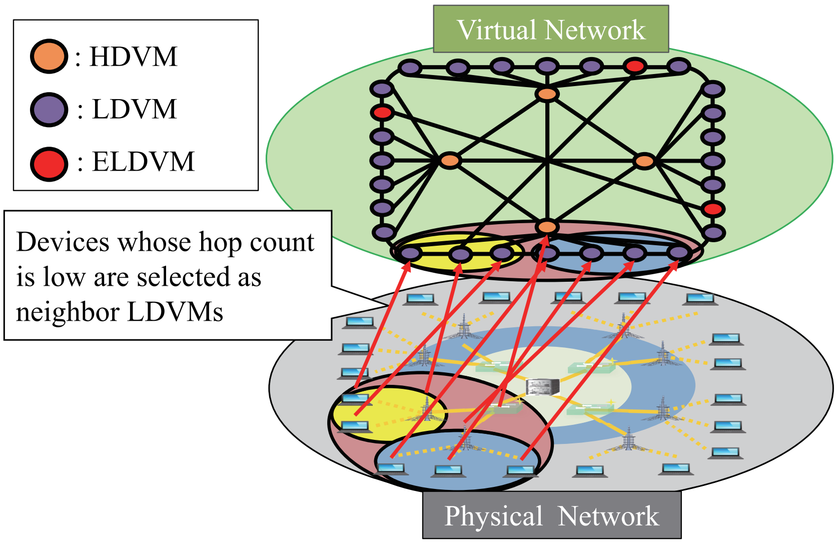

: HDVM
: LDVM
: ELDVM
Virtual Network
Devices whose hop count is low are selected as neighbor LDVMs
Physical Network

Figure 4.5: An example of our considered LDVM mapping strategy based on the neighbor information.

are located in the same segment. For instance, the red virtual subnetwork is composed of VMs of devices in the red segment in the physical network, as shown in Fig. 4.5. When the FPN occurs, the virtual subnetwork is removed and each VM in the subnetwork has much more intra-links, i.e., links in the subnetwork, in comparison with the inter-links, i.e., links between the virtual subnetworks. For instance, in the Fig. 4.5, surviving VMs lost only six links even when the red virtual subnetwork is removed by the FPN. Consequently, our strategy can keep the connectivity of the post-FPN virtual network.

4.4 A Centralized Method to Construct Virtual Network

Here, we propose a method to construct a virtual network based on the aforementioned usage of physical network information. Our proposed method is referred to as FAT-VN+, which consists of two procedures: (*i*) initial setup procedure

Table 4.2: Database of management server.

	Parameter	**Variable**
Virtual network	Predefined degree of LDVM	k_{LD}
	Ideal degree of HDVM	k_{HD}
	Ideal number of LDVMs	N_{LD}
	Ideal number of HDVMs	N_{HD}
	Total number of VMs	N
	ID of each VM	$v_i \in V$
	Type of each VM	t_{v_i}
	Current degree of each VM	$\lvert k_{v_i} \rvert$
	Current number of LDVMs	$\lvert N_{\mathrm{LD}} \rvert$
	Current number of HDVMs	$\lvert N_{\mathrm{HD}} \rvert$
	Topology information	$\boldsymbol{C}$
Physical network	IP address of each VM	d_i
	Hop count from root to each VM	$h_{d_i}^{\mathrm{root}}$ or $h_{v_i}^{\mathrm{root}}$
	Hop count between any VMs	$h_{d_i}^{d_j}$ or $h_{v_i}^{v_j}$

and (*ii*) VM joining procedure. These procedures are executed by a management server. While a service provider executes an initial setup procedure when it starts the ICT service using the virtual network, VM joining procedure is periodically executed when a VM joins in the network in order to maintain the virtual network topology in an optimal state. In order to execute these procedures, a management server stores the information summarized in table 4.2 in its database, where the degree of LDVM, k_{LD} is set by the service provider in advance.

4.4.1 Initial Setup Procedure

Procedure 1 shows the initial setup procedure, which is executed by a management server when starting the service. Additionally, we assume that the IP addresses of joining VMs are given in advance.

First, the management server obtains the total number of VMs, N, by counting IP address up. Then, the management server calculates the ideal parameters in the virtual network, i.e., the degree of HDVMs, k_{HD}, the number of LDVMs,

Procedure 1 Initial Setup Procedure

```
Given: degree of LDVM, $k_{\mathrm{LD}}$, and IP address of each VM, $d_i$
Calculate the total number of joining VMs, $N$
Derive ideal parameters, $k_{\mathrm{HD}}$, $N_{\mathrm{LD}}$, and $N_{\mathrm{HD}}$, by calculating (3.12) with $k_{\mathrm{LD}}$ and $N$
Decide a virtual network topology, $\boldsymbol{C}$, based on the ideal parameters
HDVM mapping phase based on depth information /* Select VMs having lower value of $h_{d_i}^{\mathrm{root}}$ as HDVMs */
LDVM mapping phase based on neighbor information /* Select VMs having lower value of $h_{d_i}^{d_j}$ as neighbor LDVMs */
Update information in its database
```

N_{LD}, and that of HDVMs, N_{HD}, by calculating (3.12) with k_{HD} and N. Based on the calculated parameters, it constructs the virtual network topology. Hence, the constructed virtual network follows the optimal parameters. Here, the virtual network topology is represented with the adjacency matrix, $\boldsymbol{C}$, where each matrix element $c_{v_i v_j}$ denotes the link information between the VMs, v_i and v_j, i.e., there exists a link between v_i and v_j if $c_{v_i v_j} = 1$. Additionally, the management server updates the current degree of each VM, $|k_{v_i}|$, the current number of LDVMs, $|N_{\mathrm{LD}}|$, and that of HDVMs, $|N_{\mathrm{HD}}|$, based on the constructed virtual network topology.

After setting up the virtual network topology, the management server attempts to decide the location (or ID, v_i) and type of each VM, t_{v_i}, in the virtual network based on the IP address of each VM, d_i, in the physical network. In other words, it decides the relation between the virtual and physical networks. In order to optimally decide ID and type of each VM, the management server executes the procedure consisting of two phases, i.e., (i) HDVM mapping phase and (ii) LDVM mapping phase, which are described as follows:

(i) *HDVM mapping phase* – The HDVM mapping phase aims to select appropriate HDVMs by considering the depth of VMs in the physical network. The management server initially calculates hop counts from a root device to all VMs in the physical network by using the IP address of all VMs. In other words, the management server derives $h_{d_i}^{\mathrm{root}}$ for all d_i. Then, it selects $|N_{\mathrm{HD}}|$ VMs, which have a lower value of $h_{d_i}^{\mathrm{root}}$ compared with the other VMs, as HDVMs. Additionally, it allocates the ID of HDVMs to the selected VMs.

(*ii*) *LDVM mapping phase* – Following the HDVM mapping phase, the management server starts the LDVM mapping phase, in which neighbor LDVMs are selected based on the hop count between any VMs in the physical network. First, the management server calculates hop counts between any VMs in the physical network for all cases. Therefore, it obtains the $h_{d_i}^{d_j}$ for any d_i and d_j. After that, it randomly selects a VM, d_i, as a LDVM, v_i, and then selects VM, d_j, which has the lowest value of $h_{d_i}^{d_j}$, as a neighbor LDVM of v_i. This procedure continues until all IDs are assigned to VMs. Finally, the management server updates the information in its database.

4.4.2 VM Joining Procedure

If the manager server receives the participation request from a new joining VM, it executes the VM joining procedure as shown in procedure 2. This procedure aims to establish virtual links of new joining VM while keeping the optimal virtual network topology.

First, the management server stores the IP address of a newly joining VM, d_i, which is included in the request message of new joining VM. Additionally, it increments the total number of VMs, N, and the current number of LDVMs, $|N_{\mathrm{LD}}|$ since the newly joining VM becomes LDVM. Next, it attempts to allocate an ID of a newly joining VM, where an unused ID, v_i, in list of IDs, V, is assigned to a newly joining VM. Additionally, t_{v_i} is set to 0, which indicates that the type of newly joining VM is LDVM. In order to set up the virtual links to optimally select neighbor VMs, the management server executes the procedure consisting of

Procedure 2 VM joining procedure

```
Store IP address of a newly joining VM, $d_i$, in its database and Update $N$ and $|N_{\mathrm{LD}}|$
Decide ID of a newly joining VM, $v_i \in V$
Insertion phase /* Construct two virtual links to LDVMs that are closed in physical network */
Expansion phase /* Construct additional virtual links to closed HDVM or the LDVMs while $k_{\mathrm{LD}} > |k_{v_i}|$ */
Calculate $N_{\mathrm{HD}}$ based on (3.12) with the updated $N$ and $k_{\mathrm{LD}}$
if $N_{\mathrm{HD}} > |N_{\mathrm{HD}}|$ then
    HDVM selection phase /* Select a LDVM that is closed to root device in the physical network as a new HDVM and reconstruct the virtual network topology */
end if
Update information in its database
```

two phases, i.e., (*i*) insertion phase and (*ii*) expansion phase, which are described as follows:

(*i*) *Insertion phase* – The objective of the insertion phase is to establish two virtual links from the newly joining VM to VMs that are closed to the newly joining VM in the physical network. First, in order to select a neighbor VM of the newly joining VM, v_i, the management server attempts to find a VM, v_{n1}, whose hop count to v_i is the lowest in the physical network, i.e., v_{n1} is selected based on the following equation.

$$v_{\mathrm{n1}} = \operatorname*{argmin}_{v_j \in V, i \neq j} h_{v_j}^{v_i} \tag{4.3}$$

Next, the management server selects another VM, v_{n2}, as another neighbor VM of v_i. The VM, v_{n2}, is selected from neighbor VMs of v_{n1} and also needs to be close to v_i in the physical network. Assume that $v_{\mathrm{n1}}^j \in V_{\mathrm{n1}}$ is the neighbor VMs of v_{n1}, v_{n2} can be selected based on the following equation.

$$v_{\mathrm{n2}} = \operatorname*{argmin}_{v_{\mathrm{n1}}^{j} \in V_{\mathrm{n1}}} h_{v_{\mathrm{n1}}}^{v_{\mathrm{n1}}^{j}} \tag{4.4}$$

The management server inserts v_i into the link between v_{n1} and v_{n2}. In this vein, it breaks the existing link and creates new two links from v_i to v_{n1} and v_{n2}, respectively. After creating the links, the management server updates the topology information, i.e., $c_{v_{\mathrm{n1}}v_{\mathrm{n2}}} = 0$, $c_{v_i v_{\mathrm{n1}}} = 1$, and $c_{v_i v_{\mathrm{n2}}} = 1$. Additionally, it increments the degree of v_i, i.e., $|k_{v_i}| = 2$.

(*ii*) *Expansion phase* – Following the insertion phase, the management server moves to the expansion phase, in which it establishes virtual links as long as $|k_{v_i}|$ is lower than k_{LD} or the candidates for the neighbor VMs exist. First, the management server attempts to establish the link between v_i and a HDVM, v_i^{HD}, which is closest to v_i in the physical network if $|k_{v_i^{\mathrm{HD}}}|$ is lower than k_{HD}. Here, v_i^{HD} is given by the following equation.

$$v_i^{\mathrm{HD}} = \operatorname*{argmin}_{v_j \in V, i \neq j, t_{v_j} = 1} h_{v_i}^{v_j} \tag{4.5}$$

Then, the management server creates a candidate list, $\overline{V} = \{\overline{v_1}, \overline{v_2}, \dots, \overline{v_{|\overline{V}|}}\}$, where LDVMs having a lower degree than k_{LD} are selected as candidates. From the candidate list, $\overline{V}$, the management server selects a VM, v_{ne}, which is far from v_i, and creates links between v_i and v_{ne}, where degree of v_i and that of v_{ne} are incremented, and v_{ne} is removed from $\overline{V}$. This procedure continues while $|k_{v_i}| < k_{\mathrm{LD}}$ or $\overline{V} \neq \emptyset$.

After creating the links of the newly joining VM, the management server attempts to maintain the degree distribution of virtual network in an optimal state because the ideal number of HDVMs, N_{HD}, changes according to the increase of joining VMs. Therefore, it calculates the value of N_{HD} by using (3.12) with the updated N and k_{LD}. If the ideal number of HDVMs is larger than the current number of HDVMs, i.e., $N_{\text{HD}} > |N_{\text{HD}}|$, it executes HDVM selection phase, which is described as follows:

(*iii*) *HDVM selection phase* – In this phase, the management server initially finds a LDVM, v_{NHD}, which is close to the root device in the physical network. The LDVM, v_{NHD}, will be new HDVM and can be selected by calculating the following equation.

$$v_{\text{NHD}} = \operatorname*{argmin}_{v_i \in V, t_{v_i}=0} h_{v_i}^{\text{root}} \tag{4.6}$$

Then, the management server breaks the existing links of v_{NHD}. It creates the links between v_{NHD} and other HDVMs in order to construct complete graph. Additionally, it finds LDVMs that are closed to v_{NHD} in the physical network and creates new links between v_{NHD} and these LDVMs. In this situation, some VMs do not satisfy their degree temporarily. However, the virtual network gradually satisfies the degree of such VMs with the participation of other VMs. Finally, it updates the changed information in its database.

4.5 Analysis on Virtual Network Connectivity

In this section, we mathematically analyze the connectivity of the virtual network in the environment where FPN occurs. The objective of this analysis is to derive the following performance metrics: (*i*) connectivity after FPN occurs, (*ii*) connectivity after FPN and FVM occur, (*iii*) connectivity after FPN and AVM occur, and (*iv*) number of available VMs after FPN occurs. The performance metrics can be derived through the following steps: (*i*) modeling of a probability that VMs are removed from the virtual networks by failures, (*ii*) formulating the degree distribution of post-failure virtual networks.

4.5.1 VM Removal Probability

The FPN causes numerous VMs to be removed from the virtual network and the probability that a VM is removed differs depending on the virtual network construction schemes. Therefore, we model the VM removal probability in virtual networks that are constructed based on the existing and proposed construction schemes, i.e., FAT-VN and FAT-VN+, respectively. Here, we define the VM removal probability, f_k, which denotes the probability that a VM with degree k will be removed from the virtual network.

In the existing scheme, since neighboring VMs are randomly selected regardless of their location in physical network, a single FPN causes VMs to be randomly removed from the virtual network regardless of their type and location. Consequently, the VM removal probability in the virtual network that is constructed in the existing scheme, f_k^{FPN}, can be expressed with the number of removed VMs, $N^{\boldsymbol{R}}$, as follows.

$$f_k^{\mathrm{FPN}} = \frac{N^{\boldsymbol{R}}}{N}. \tag{4.7}$$

On the other hand, in the proposed construction scheme, VMs that are close to each other in the physical network become neighbor VMs in the virtual network. Therefore, FPN removes a cluster, which is composed of the neighbor VMs, from the virtual network. In order to simplify the analysis, the original virtual network can be classified into sets of surviving and removed VMs, $\boldsymbol{S}$ and $\boldsymbol{R}$, respectively, as shown in Fig. 4.6. With this classification, we can regard the set of removed VMs, $\boldsymbol{R}$, as a virtual node having degree k_{vn}. Therefore, the VM removal probability in the virtual network that is constructed based on the proposed construction

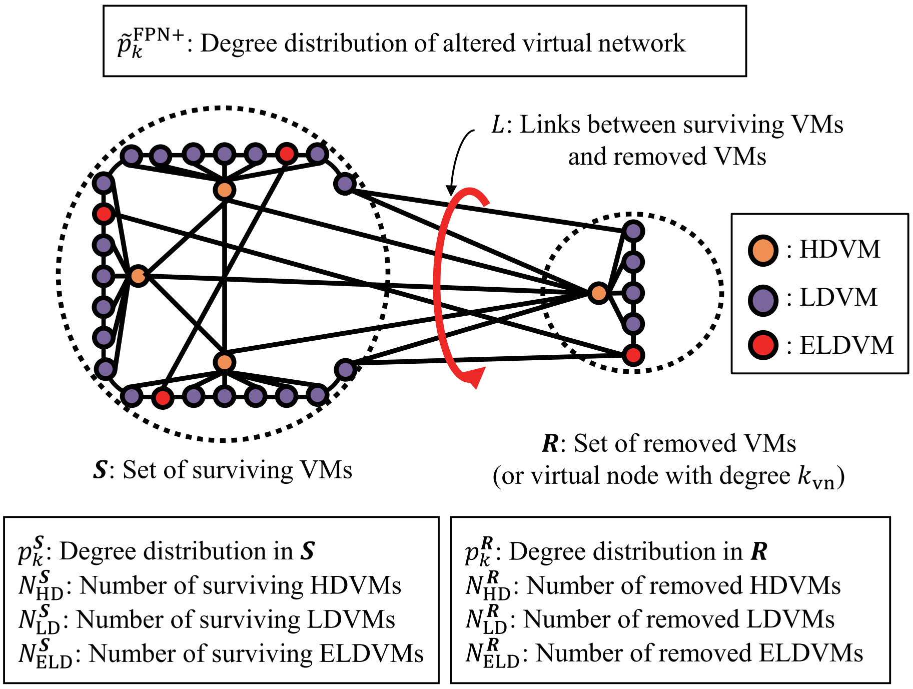

$\tilde{p}_k^{\mathrm{FPN+}}$: Degree distribution of altered virtual network
L: Links between surviving VMs and removed VMs
●: HDVM
●: LDVM
●: ELDVM
$\boldsymbol{S}$: Set of surviving VMs
$\boldsymbol{R}$: Set of removed VMs (or virtual node with degree k_{vn})
$p_k^{\boldsymbol{S}}$: Degree distribution in $\boldsymbol{S}$
$N_{\mathrm{HD}}^{\boldsymbol{S}}$: Number of surviving HDVMs
$N_{\mathrm{LD}}^{\boldsymbol{S}}$: Number of surviving LDVMs
$N_{\mathrm{ELD}}^{\boldsymbol{S}}$: Number of surviving ELDVMs
$p_k^{\boldsymbol{R}}$: Degree distribution in $\boldsymbol{R}$
$N_{\mathrm{HD}}^{\boldsymbol{R}}$: Number of removed HDVMs
$N_{\mathrm{LD}}^{\boldsymbol{R}}$: Number of removed LDVMs
$N_{\mathrm{ELD}}^{\boldsymbol{R}}$: Number of removed ELDVMs

Figure 4.6: An analytical model of the impact of FPN on the virtual network that is constructed by the proposed scheme, and notations defined in the model.

scheme, $f_k^{\mathrm{FPN+}}$, can be defined as follows.

$$f_k^{\mathrm{FPN+}} = \begin{cases} 1, & \text{if } \ k = k_{\mathrm{vn}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

Due to this virtualization, the degree distribution of the original virtual network, p_k, which is described as (3.12) in chapter 3, gets altered. Let $N_{\mathrm{HD}}^{\boldsymbol{S}}$ and $N_{\mathrm{LD}}^{\boldsymbol{S}}$ denote the number of surviving HDVMs and that of surviving LDVMs (including ELDVMs), where the total number of VMs is expressed as the sum of $N_{\mathrm{HD}}^{\boldsymbol{S}}$, $N_{\mathrm{LD}}^{\boldsymbol{S}}$, and the number of virtual node, i.e., 1. Therefore, the degree distribution of the altered virtual network, $\tilde{p}_k^{\mathrm{FPN+}}$, can be expressed as follows.

$$\tilde{p}_k^{\mathrm{FPN+}} = \begin{cases} 1/(N_{\mathrm{HD}}^{\boldsymbol{S}} + N_{\mathrm{LD}}^{\boldsymbol{S}} + 1), & \text{if } k = k_{\mathrm{vn}}, \\ N_{\mathrm{HD}}^{\boldsymbol{S}}/(N_{\mathrm{HD}}^{\boldsymbol{S}} + N_{\mathrm{LD}}^{\boldsymbol{S}} + 1), & \text{if } k = k_{\mathrm{HD}}, \\ N_{\mathrm{LD}}^{\boldsymbol{S}}/(N_{\mathrm{HD}}^{\boldsymbol{S}} + N_{\mathrm{LD}}^{\boldsymbol{S}} + 1), & \text{if } k = k_{\mathrm{LD}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

Moreover, supposed that the degree of LDVMs is 3 and the number of removed VMs is less than half of the total number of VMs in the original virtual network, degree of virtual node, k_{vn}, is expressed as the sum of links between the sets of $\boldsymbol{S}$ and $\boldsymbol{R}$, L. Additionally, as shown in Fig. 4.7, links L can be classified into four kinds of links as follows: (*i*) the links between the removed HDVMs and surviving HDVMs, (*ii*) the links between the surviving LDVMs and removed HDVMs, (*iii*) the links between the surviving ELDVMs and removed ELDVMs, and (*iv*) the links between the surviving LDVMs and removed LDVMs. Therefore, k_{v} is formulated as follows.

$$k_{\mathrm{vn}} = N_{\mathrm{HD}}^{\boldsymbol{S}} N_{\mathrm{HD}}^{\boldsymbol{R}} + \left[N_{\mathrm{HD}}^{\boldsymbol{R}}\{k_{\mathrm{HD}} - (N_{\mathrm{HD}} - 1)\} - N_{\mathrm{LD}}^{\boldsymbol{R}}\right] + N_{\mathrm{ELD}}^{\boldsymbol{R}} + 2. \tag{4.10}$$

In order to derive the degree distribution after FPN occurs, we need to quantify the link loss probability of each surviving VM, $q_k^{\mathrm{FPN+}}$, which indicates the probability that a VM having degree k loses a link. Since the surviving HDVMs lose the links to removed HDVMs, the number of lost links in the case of the surviving HDVMs is given as $N_{\mathrm{HD}}^{\boldsymbol{S}} N_{\mathrm{HD}}^{\boldsymbol{R}}$. On the other hand, surviving LDVMs (or ELDVMs) lose the links to removed HDVMs, the links to removed ELDVMs, and the links to removed edge LDVMs. Therefore, the number of lost links in the case of the surviving LDVMs is decided as $\left[N_{\mathrm{HD}}^{\boldsymbol{R}}\{k_{\mathrm{HD}} - (N_{\mathrm{HD}} - 1)\} - N_{\mathrm{LD}}^{\boldsymbol{R}}\right] + N_{\mathrm{ELD}}^{\boldsymbol{R}} + 2$.

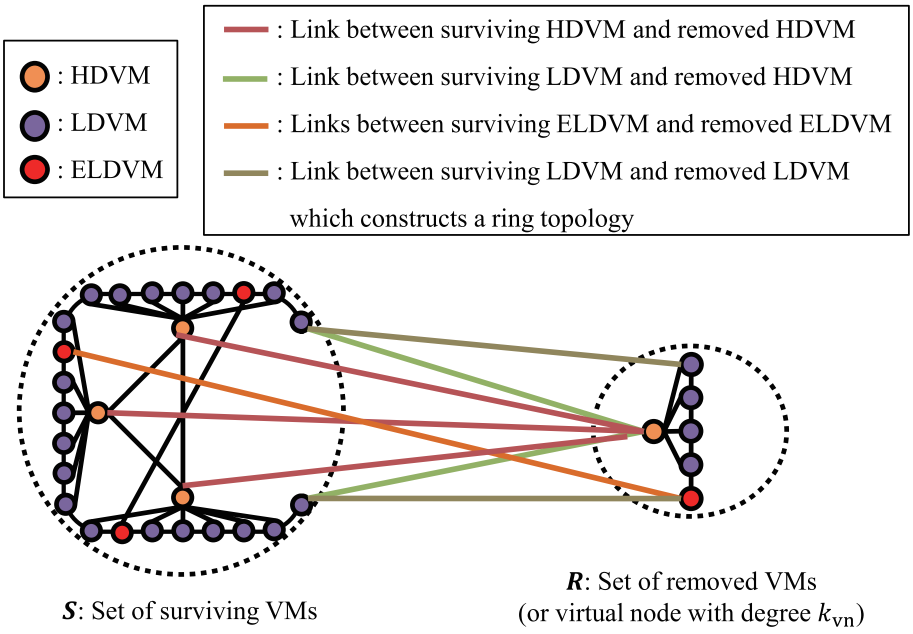

● : HDVM
● : LDVM
● : ELDVM
: Link between surviving HDVM and removed HDVM
: Link between surviving LDVM and removed HDVM
: Links between surviving ELDVM and removed ELDVM
: Link between surviving LDVM and removed LDVM which constructs a ring topology
S: Set of surviving VMs
R: Set of removed VMs
(or virtual node with degree k_{vn})

Figure 4.7: Types of links between surviving VMs and removed VMs.

Consequently, the link loss probability due to the FPN in the FAT-VN+, $q_k^{\mathrm{FPN+}}$, can be formulated as follows.

$$q_k^{\mathrm{FPN+}} = \begin{cases} \frac{N_{\mathrm{HD}}^{\boldsymbol{S}} N_{\mathrm{HD}}^{\boldsymbol{R}}}{k_{\mathrm{HD}} N_{\mathrm{HD}}^{\boldsymbol{S}}}, & \text{if } \ k = k_{\mathrm{HD}}, \\ \frac{\left[N_{\mathrm{HD}}^{\boldsymbol{R}} \{k_{\mathrm{HD}} - (N_{\mathrm{HD}} - 1)\} - N_{\mathrm{LD}}^{\boldsymbol{R}}\right] + N_{\mathrm{ELD}}^{\boldsymbol{R}} + 2}{k_{\mathrm{LD}} N_{\mathrm{LD}}^{\boldsymbol{S}}}, & \text{if } \ k = k_{\mathrm{LD}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

4.5.2 Connectivity of FAT-VN

Here, we derive the connectivity of FAT-VN. In order to quantify the connectivity, we use critical threshold, which is defined in (3.21).

At first, we derive the critical threshold of FAT-VN when a single FPN occurs, Q_{FPN}. In other words, we quantify how many VMs can be removed by a single FPN without network disruption. As described in (4.7), the VM removal probability in FAT-VN has the same value regardless of the degree of VMs, which indicates that the FPN randomly removes VMs from the virtual network. Therefore, the value of Q_{FPN} can be formulated by using the percolation threshold in the case of random removal in [33].

$$Q_{\mathrm{FPN}} = 1 - \frac{1}{\langle k \rangle^2 / \langle k \rangle - 1}, \tag{4.12}$$

where $\langle k \rangle$ denotes the average degree of the virtual network.

Then, we construct a mathematical model to evaluate the critical threshold of post-FPN FAT-VN when two kinds of FVNs, i.e., FVM and AVM, occur. To derive the critical threshold in this case, we first derive the degree distribution of FAT-VN after FPN, p_k^{FPN}, which can be expressed as the sum of the probability that VMs with degree i become VMs with degree k after VMs are removed by the VM removal probability, f_i^{FPN}, where $k \leq i$. Thus, we have

$$p_k^{\mathrm{FPN}} = \begin{cases} \sum_{i=0} (f_i^{\mathrm{FPN}})^i p_{i,\boldsymbol{S}}^{\mathrm{FPN}}, & \text{if } \ k = 0, \\ \sum_{i=k} \begin{pmatrix} i \\ k \end{pmatrix} (f_i^{\mathrm{FPN}})^{i-k} (1 - f_i^{\mathrm{FPN}})^k p_{i,\boldsymbol{S}}^{\mathrm{FPN}}, & \text{otherwise,} \end{cases} \tag{4.13}$$

where $p_{i,\boldsymbol{S}}^{\mathrm{FPN}}$ is the degree distribution of FAT-VN consisting of surviving VMs before the links between surviving and removed VMs are removed, and is given as follows.

$$p_{i,\boldsymbol{S}}^{\mathrm{FPN}} = \frac{(1 - f_i^{\mathrm{FPN}}) p_i}{1 - \sum_j f_j^{\mathrm{FPN}} p_j}. \tag{4.14}$$

When there exist the VMs whose degree is 0, the virtual network is disrupted. Therefore, if $p_0^{\mathrm{FPN}} \geq 1$, the critical threshold of post-FPN virtual network in case of FVM, Q_{FPNFVM}, can be formulated as follows.

$$Q_{\mathrm{FPNFVM}} = 0. \tag{4.15}$$

When $p_0^{\text{FPN}} < 1$, VMs are randomly removed from the post-FPN virtual network due to the FVM. Therefore, we have

$$Q_{\text{FPNFVM}} = 1 - \frac{1}{(\langle k \rangle_{\text{FPN}})^2 / \langle k \rangle_{\text{FPN}} - 1}, \tag{4.16}$$

where $\langle k \rangle_{\text{FPN}}$ denotes the average degree of the post-FPN virtual network and the value of $\langle k \rangle_{\text{FPN}}$ can be calculated based on the degree distribution of post-FPN virtual network, p_k^{FPN}.

On the other hand, in case of AVM, the VMs having higher degree are preferentially removed from the post-FPN virtual network. Therefore, we use the percolation threshold in the case of degree-dependent removal in [33] to evaluate the critical threshold of post-FPN virtual network when AVM occurs, Q_{FPNAVM}. When $p_0^{\text{FPN}} < 1$, the value of Q_{FPNAVM} can be expressed as follows.

$$Q_{\text{FPNAVM}} = \alpha \langle k \rangle_{\text{FPN}} - \sum_{k=|K|_{\text{FPNAVM}}}^{|K|_{\text{FPN}}} (k-1) p_k^{\text{FPN}}, \tag{4.17}$$

where $|K|_{\text{FPN}}$ and $|K|_{\text{FPNAVM}}$ denote the maximum degree of degree distribution p_k^{FPN} before AVM occurs and the maximum degree of degree distribution p_k^{FPN} after AVM occurs, respectively. Additionally, the value of α is defined as follows.

$$\alpha = 1 - \frac{1}{(\langle \tilde{k} \rangle_{\text{FPN}})^2 / \langle \tilde{k} \rangle_{\text{FPN}} - 1}, \tag{4.18}$$

where $\langle \tilde{k} \rangle_{\text{FPN}}$ is the average degree of degree distribution p_k^{FPN} from 0 to $|K|_{\text{FPNAVM}}$. Note that $Q_{\text{FPNAVM}} = 0$ when $p_0^{\text{FPN}} < 1$.

4.5.3 Connectivity of FAT-VN+

Here, we derive the connectivity of FAT-VN+ in three scenarios, i.e., when FPN occurs, when FVM occurs after FPN, and when AVM occurs after FPN.

First, we construct a mathematical model to evaluate the critical threshold of FAT-VN+ when a single FPN occurs, $Q_{\text{FPN+}}$. Since the FPN removes the cluster from the virtual network, the VM removal probability in FAT-VN+, $f_k^{\text{FPN+}}$, is defined as shown in (4.8). Based on $f_k^{\text{FPN+}}$, the degree distribution of virtual network after FPN occurs, $p_k^{\text{FPN+}}$, can be expressed as the sum of the probability that VMs with degree i become VMs with degree k after VMs are removed. Since the links between the sets of $\boldsymbol{S}$ and $\boldsymbol{R}$, L, are removed by the link loss probability, $q_i^{\text{FPN+}}$, $p_k^{\text{FPN+}}$ can be formulated as follows.

$$p_k^{\text{FPN+}} = \begin{cases} \sum_{i=0} (q_i^{\text{FPN+}})^i p_{i,\boldsymbol{S}}^{\text{FPN+}}, & \text{if } \ k = 0, \\ \sum_{i=k} \begin{pmatrix} i \\ k \end{pmatrix} (q_i^{\text{FPN+}})^{i-k} (1 - q_i^{\text{FPN+}})^k p_{i,\boldsymbol{S}}^{\text{FPN+}}, & \text{otherwise,} \end{cases} \tag{4.19}$$

where $p_{i,\boldsymbol{S}}^{\text{FPN+}}$ is the degree distribution of the virtual network consisting of surviving VMs before the links L are removed. Since the virtual node with degree k_{vn} is removed from the virtual network, $p_{i,\boldsymbol{S}}^{\text{FPN+}}$ can be defined as follows.

$$p_{i,\boldsymbol{S}}^{\text{FPN+}} = \begin{cases} N_{\text{HD}}^{\boldsymbol{S}} / (N^{\boldsymbol{S}} + 1), & \text{if } \ k = k_{\text{HD}}, \\ N_{\text{LD}}^{\boldsymbol{S}} / (N^{\boldsymbol{S}} + 1), & \text{if } \ k = k_{\text{LD}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.20}$$

When there exist the VMs whose degree is 0, the virtual network is disrupted. In other words, the virtual network is disrupted due to the FPN when $p_0^{\text{FPN+}} \geq 1$. Since the critical threshold indicates how many VMs can be removed from a network without disrupting the virtual network, it can be derived by calculating the number of removed VMs, $N_{\boldsymbol{R}}^{\mathrm{FPN+}}$, when $p_0^{\mathrm{FPN+}} \geq 1$. Therefore, the value of $Q_{\mathrm{FPN+}}$ can be formulated as follows.

$$Q_{\mathrm{FPN+}} = \frac{N_{\boldsymbol{R}}^{\mathrm{FPN+}}}{N}. \tag{4.21}$$

Then, we derive the critical threshold of post-FPN virtual network when FVM occurs, $Q_{\mathrm{FPNFVM+}}$, in the same way as FAT-VN. If the number of removed VMs due to the FPN, i.e., the scale of FPN, is larger than $N_{\boldsymbol{R}}^{\mathrm{FPN+}}$, the virtual network is disrupted due to FPN. Therefore, if the scale of FPN is larger than $N_{\boldsymbol{R}}^{\mathrm{FPN+}}$, we can obtain $Q_{\mathrm{FPNFVM+}} = 0$. Otherwise, VMs are randomly removed from the post-FPN virtual network due to the FVM. Therefore, the critical threshold of post-FPN FAT-VN+ when FVM occurs, $Q_{\mathrm{FPNFVM+}}$, is given with the average degree of the post-FPN virtual network, $\langle k \rangle_{\mathrm{FPN+}}$, as follows.

$$Q_{\mathrm{FPNFVM+}} = 1 - \frac{1}{(\langle k \rangle_{\mathrm{FPN+}})^2 / \langle k \rangle_{\mathrm{FPN+}} - 1}. \tag{4.22}$$

On the other hand, in case of AVM, if the scale of FPN is smaller than $N_{\boldsymbol{R}}^{\mathrm{FPN+}}$, the critical threshold of post-FPN FAT-VN+ when AVM occurs, $Q_{\mathrm{FPNAVM+}}$ can be expressed with the maximum degree in degree distribution $p_k^{\mathrm{FPN+}}$ before AVM occurs, $|K|_{\mathrm{FPN+}}$, and the maximum degree in degree distribution $p_k^{\mathrm{FPN+}}$ after AVM occurs, $|K|_{\mathrm{FPNAVM+}}$, as follows.

$$Q_{\mathrm{FPNAVM+}} = \alpha \langle k \rangle_{\mathrm{FPN+}} - \sum_{k=|K|_{\mathrm{FPNAVM+}}}^{|K|_{\mathrm{FPN+}}} (k-1) p_k^{\mathrm{FPN+}}, \tag{4.23}$$

where the value of α can be given as follows.

$$\alpha = 1 - \frac{1}{(\langle \tilde{k} \rangle_{\mathrm{FPN+}})^2 / \langle \tilde{k} \rangle_{\mathrm{FPN+}} - 1}, \tag{4.24}$$

where $\langle \tilde{k} \rangle_{\mathrm{FPN+}}$ is the average degree in degree distribution $p_k^{\mathrm{FPN+}}$ from 0 to $|K|_{\mathrm{FPNAVM+}}$.

4.5.4 Number of Available VMs

Here, we present a mathematical model to evaluate the number of available VMs in the virtual networks after FPN occurs, N_{AVL}.

The FPN removes VMs from the virtual network, which results in network disruption. If the network is disrupted, there are multiple clusters in the virtual network. The divided clusters can be classified into giant cluster, which has maximum number of VMs after FPN, and other smaller clusters [55]. The smaller clusters can be also classified into a cluster composed by a single VM (shortly referred to as the “single-VM cluster”) and a cluster composed by more than one VMs (shortly referred to as the “multiple-VMs clusters”). The ratio of the VMs that belong to the single-VM clusters is expressed as the probability that there exists VMs with degree 0 after FPN occurs, p'_0. Additionally, the ratio of the VMs that belong to multiple-VMs clusters is expressed as $\sum_{k=1}^{\infty} p'_k (u_k)^k$, where u_k denotes the average probability that a link connected to a VM with degree k leads to another VM that does not belong to the giant cluster. Consequently, giant cluster ratio, G_{c}, can be formulated as follows.

$$G_{\mathrm{c}} = 1 - p'_0 - \sum_{k=1}^{K} p'_k (u_k)^k. \tag{4.25}$$

Additionally, we assume that only VMs in a giant cluster continue the service because VMs in smaller clusters cannot find the desired VMs or contents. Therefore, the number of available VMs in the virtual networks after FPN occurs,

N_{AVL}, can be expressed with giant cluster ratio, G_{c}, as follows.

$$N_{\mathrm{AVL}} = N^{\boldsymbol{S}}(1 - p'_0 - \sum_{k=1}^{K} p'_k (u_k)^k). \tag{4.26}$$

For instance, we can obtain the number of available VMs in FAT-VN+ after FPN occurs by calculating the above equation with degree distribution of FAT-VN+ after FPN, $p_k^{\mathrm{FPN+}}$.

4.6 Performance Evaluation

In this section, we evaluate the performance of our proposed FAT-VN+ through numerical calculation using the aforementioned mathematical formulas. We compare the performance of FAT-VN+ with FAT-VN, which is described in chapter 3. By comparing with the existing construction scheme based on random manner, the advantage of the proposed FAT-VN+ that considers an optimal method to construct virtual network based on the physical network information can be obviously understood. Two different metrics defined in [42, 55], i.e., (*i*) maximum number of removable VMs without network disruption and (*ii*) number of available VMs are used to evaluate the performance of the considered virtual networks.

4.6.1 Performance Comparison in Maximum Number of Removable VMs

Here, we evaluate the maximum number of removable VMs without network disruption, N_{MR}. Since N_{MR} indicates how many VMs can be removed from a network without disrupting the network, the value of N_{MR} can be calculated by using critical threshold, Q, which can be derived based on the aforementioned

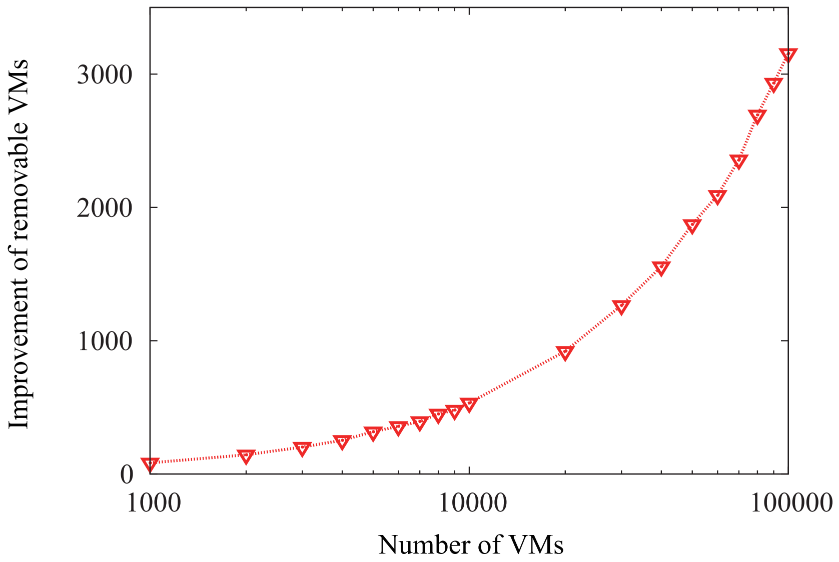


Figure 4.8: Connectivity improvement against FPN in FAT-VN+.

mathematical expressions. N_{MR} is represented with the number of VMs before VM removal, N, and critical threshold, Q, as follows.

$$N_{\mathrm{MR}} = QN, \tag{4.27}$$

The value of N_{MR} closer to $N*$ indicates much higher connectivity.

Fig. 4.8 demonstrates the improvement of the maximum number of removable VMs when a single FPN occurs. The physical network is tree topology and the average degree of each virtual network is set to 3 and the number of VMs is varied from 1000 to 100000. It can be shown that FAT-VN+ achieves higher connectivity against FPN regardless of the virtual network size. This is because, in FAT-VN, the probability that the surviving VMs have links to the removed VMs is higher because the VMs are randomly removed. In contrast to this, in FAT-VN+, the surviving VMs do not have links to the removed VMs by choosing closed VMs in the physical network as neighbor VMs in the virtual network.

Fig. 4.9 shows the number of removable VMs by FVM when the number of

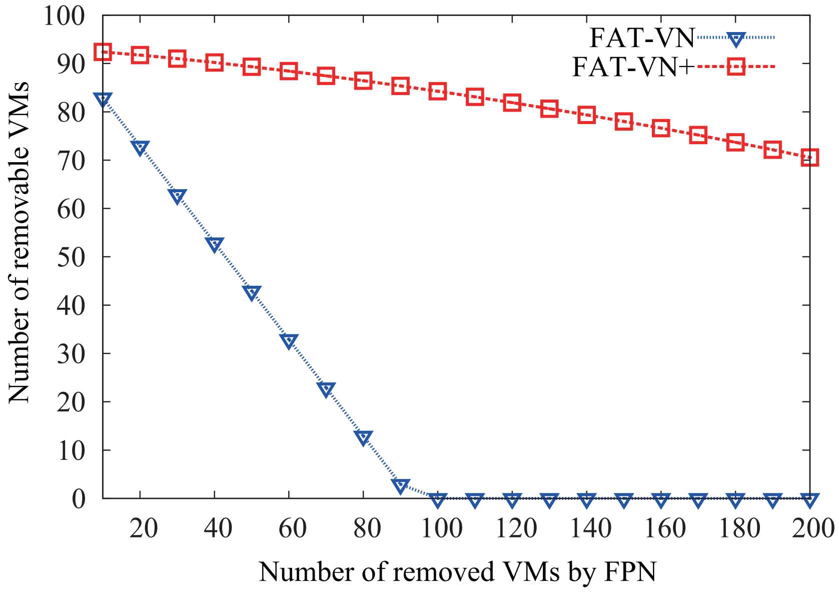


Figure 4.9: Number of removable VMs by FVM in post-FPN virtual networks.

removed VMs by FPN changes. In this result, the average degree of each virtual network is set to 3 and the number of VMs is set to 1000. Additionally, we change the number of removed VMs by FPN from 10 to 200 in 10 increments, in order to evaluate the impact of scale of FPN on the connectivity of virtual networks. Both virtual networks decrease the number of removable VMs with the increase in the number of removed VMs by FPN. However, the performance of FAT-VN drastically decreases with the increase of the number of removed VMs and it becomes 0 from 100 because FAT-VN is disrupted with the removal of 100 VMs. On the other hand, our proposed FAT-VN+ gradually decreases the performance because connectivity of FAT-VN+ after FPN is still high, i.e., the surviving VMs have a lot of links in FAT-VN+ even when FPN occurs. Consequently, form this result, we obviously understand the effectiveness of our proposed FAT-VN+ when both FPN and FVM occur.

Fig. 4.10 depicts the number of removable VMs by AVM when the number of removed VMs by FPN changes. This result is obtained by the numerical

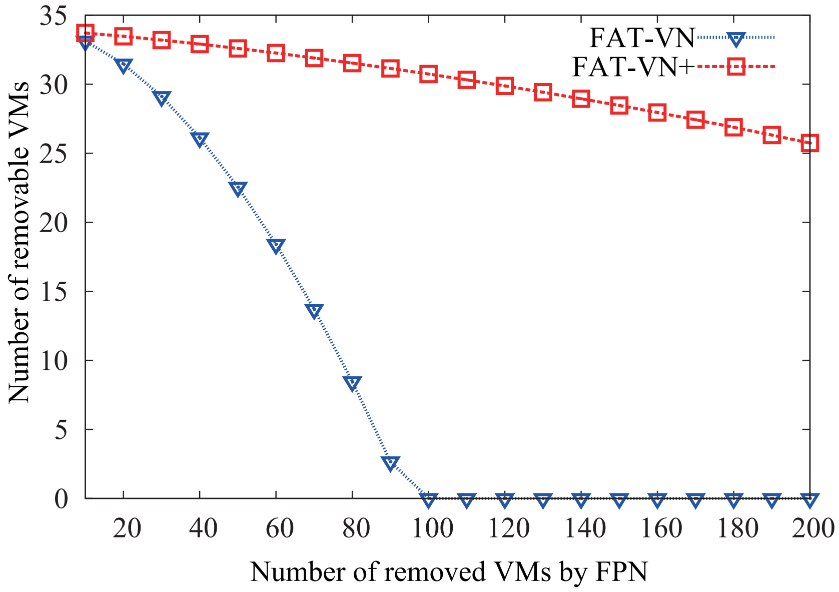


Figure 4.10: Number of removable VMs by AVM in post-FPN virtual networks.

calculation with the same setting as in the Fig. 4.9. Since AVM has a bigger impact on the connectivity of virtual networks, the number of removable VMs is lower in comparison with the case of FVM. Additionally, for the same reason in case of FVM, the decrease ratio of FAT-VN+ is lower than that of FAT-VN. Therefore, form this result, we obviously understand the effectiveness of our proposed FAT-VN+ when both FPN and FVM occur. All cases considered, we can conclude that FAT-VN+ is effective to improve the connectivity against FPN.

4.6.2 Performance Comparison in Number of Available VMs

If the virtual network splits into a lot of smaller size networks, referred to as clusters, each VM cannot connect to VMs in other clusters and is unable to find the desired target resources, i.e. VMs or data. Therefore, we evaluate the number of available VMs, N_{AVL}, in the virtual networks after FPN occurs by using the mathematical formulas (4.25) and (4.26). The maximum value of N_{AVL} is $N^{\boldsymbol{S}}$,

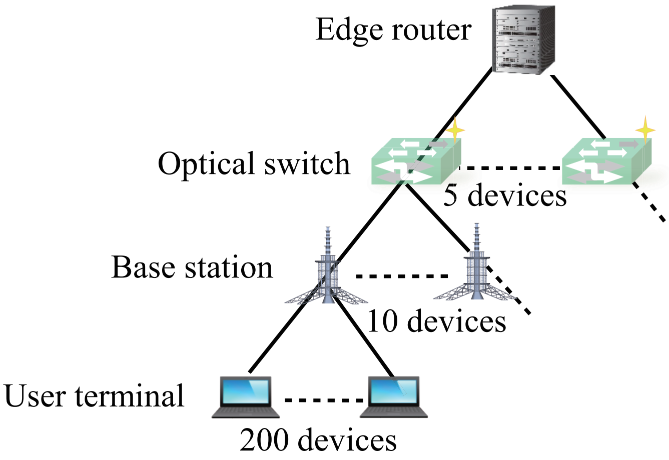


Figure 4.11: Assumed physical network for numerical calculation.

which means that all surviving VMs can connect to each other.

In this numerical calculation, we assume the following situation. The physical network is tree topology consisting of edge router, optical switches, base stations, and user terminals, as shown in Fig. 4.11. In the physical network, An edge router connects to 5 optical switches and each optical switch accommodates 10 base stations. Additionally, each base station accommodates 200 user terminals. There exist 10000 devices except an edge router and VMs of these devices join to the virtual networks. In other words, the number of VMs in virtual networks is 10000. Here, average degree of virtual networks is set to 3. In this setting, we evaluate the number of available VMs after FPN occurs, where we consider different scale of FPN, i.e., small scale FPN and Large scale FPN. In the small scale FPN, a base station ceases to function and thus 2% VMs are removed from virtual networks. On the other hand, an optical switch ceases to function to model the large scale FPN and 20% VMs are removed from virtual networks in this model. Additionally, we consider two virtual networks, i.e., FAT-VN+ and FAT-VN in order to confirm the effectiveness of the proposed FAT-VN+ by comparing the performance of FAT-VN+ with FAT-VN.

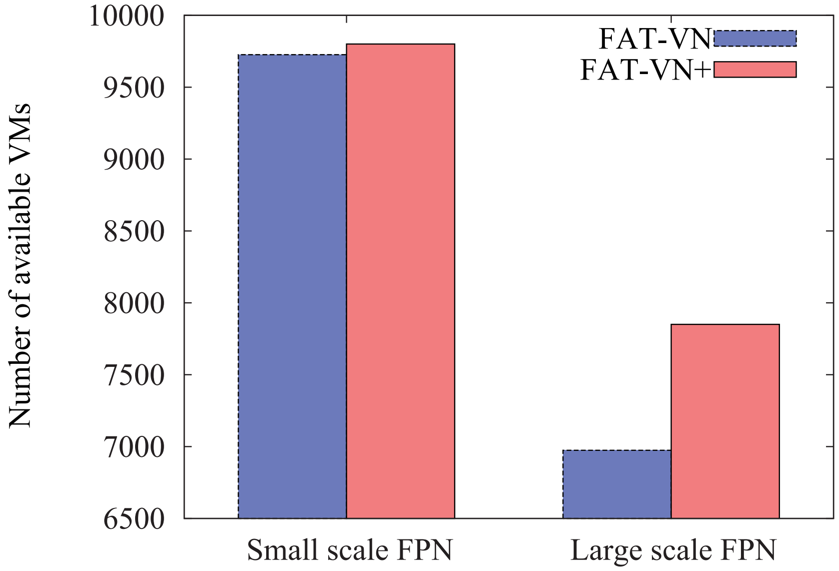


Figure 4.12: Number of available VMs in different FPN scenarios.

Fig. 4.12 demonstrates the number of available VMs in different FPN scenarios. While the number of available VMs in the existing virtual network represents lower value, the proposed virtual network achieves maximum number of available VMs regardless of the scale of FPN. This is because the proposed virtual network is not disrupted since the removed VMs are located in the same area in the virtual network. Moreover, the proposed network attains much better performance in the larger scale scenario. Indeed, the improvement ratio in the small scale FPN is 1% and the proposed virtual network improves the performance by 13% in the large scale FPN. Consequently, we can confirm the effectiveness of the proposed method to construct virtual networks.

4.7 Summary

In this chapter, we investigated a method to construct a virtual network based on physical network information for improving FPN tolerance. First, we demonstrated that the relationship between the VM locations in virtual and physical networks affects the FPN tolerance. By clarifying the impact of this relationship on the FPN tolerance, we presented that the key strategy for improving the FPN tolerance is to utilize the physical network information. Additionally, a method to construct a virtual network using physical network information, FAT-VN+, was proposed. Through numerical calculations, we confirm the effectiveness of FAT-VN+. In particular, the numerical results demonstrated that FAT-VN+ achieves maximum connectivity against FPN, which means that the virtual network is not disrupted by FPN. We also showed that the FAT-VN+ achieves high connectivity even when both FPN and FVN occur.

Chapter 5

Fault Tolerant Big Data Mining Architecture

5.1 Introduction

Highly scalable big data mining architecture has not been well studied in spite of the fact that big data mining provides many valuable and important information for us. Since the conventional architecture, where a master VM manages the data mining functions, is intolerant to failures, we envision a novel data mining architecture, where the data mining functions are fully distributed and managed by utilizing the virtual networks. Additionally, we propose a decentralized method to construct a virtual network that is tolerant to FVN and FPN. Also, we improve the service availability of big data mining by considering the data placement scheme. Through numerical calculation, we confirm the effectiveness of our envisioned big data mining architecture in terms of service availability after FPN.

Some parts of the content in this chapter are presented in the following papers, which were written by the author of this dissertation.

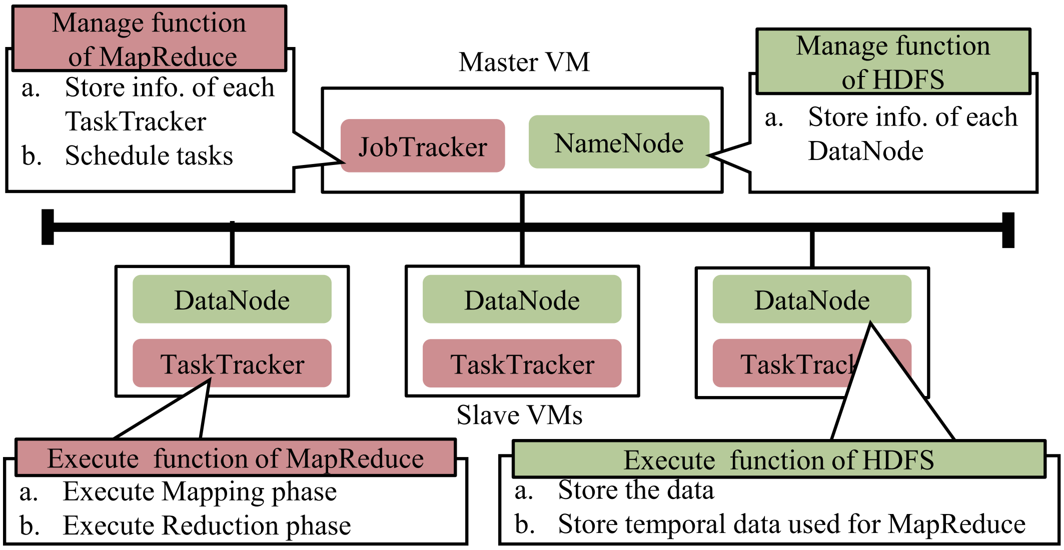


Figure 5.1: Architecture of conventional big data mining architecture.

- K. Suto, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "An Overlay-based Data Mining Architecture Tolerant to Physical Network Disruptions," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 292-301, Oct. 2014.

- K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "Toward Integrating Overlay and Physical Networks for Robust Parallel Processing Architecture," *IEEE Network*, vol. 28, no. 4, pp. 40-45, Jul.-Aug. 2014.

5.2 Conventional Big Data Mining Architecture

This section presents the overview of conventional big data mining architecture such as Hadoop, and also demonstrates that Hadoop is unable to provide high-performance services in the future network environments.

Hadoop is the most popular architecture to provide big data mining services [56, 57] since this architecture can accomplish data mining at a speed pro-

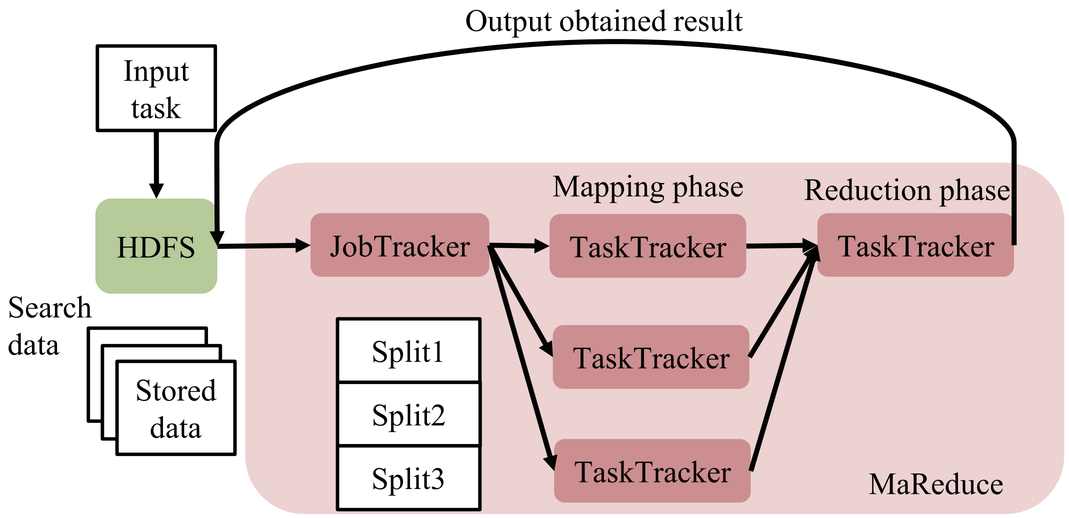


Figure 5.2: Procedure of big data mining in the conventional architecture.

portional to the number of processing VMs. Fig. 5.1 shows the architecture of Hadoop. As shown in this figure, Hadoop consists of MapReduce [58, 59] and Hadoop Distributed File System (HDFS) [60]. While MapReduce is a function for processing large data sets with a parallel and distributed algorithm by using multiple VMs, HDFS manages the big data that are used by MapReduce. Additionally, processing VMs are classified into two types, i.e., a single master VM and multiple slave VMs. The master VM has functions of JobTracker and NameNode to manage big data mining. While JobTracker manages the information of each TaskTracker and decides the task schedule based on the information, NameNode has a database of data to look up the desired data. In contrast, each slave VM has two functions, i.e., TaskTracker for executing MapReduce and DataNode for storing the data in HDFS.

Fig. 5.2 demonstrates the procedure of the considered Hadoop. When a processing request is injected, the master VM finds the data appertaining to the injected processing request by using HDFS, i.e., the NameNode attempts to look up the slave VMs that have the required data by searching its database. After the master VM receives the data from slave VMs, the JobTracker of the master VM decides mappers, which are slave VMs that execute the mapping process based on the processing load of each TaskTracker. Followed by the selection of mappers, the JobTracker divides the received data into some “splits” and allocates them to the selected mappers. Here, the number of splits is decided based on the number of mappers, e.g., the number of splits and mappers is 3 in Fig. 5.2. After the mappers receive the splits from JobTracker, the mappers perform the mapping process that classifies a large amount of information and pick out the information required for the next reduction phase. After all mappers finish the mapping process, the JobTracker selects a reducer which is a slave VM that performs the reduction process. The reducer is selected from the mappers. The reducer collects the information extracted from the mappers in the mapping process. Then, the reducer performs the reduction process, which summarizes the collected information to obtain the result. Finally, the reducer transmits the obtained result to the NameNode and then it decides the slave VMs to store the data.

While Hadoop can execute the data mining at a speed proportional to the number of VMs, the performance depends on task allocation and scheduling schemes. Therefore, high-performance parallel data mining architectures that aim to improve processing speed, network resource efficiency, computational resource efficiency, and energy efficiency, have been developed in valuable literatures [61, 62]. The works [63, 64, 65] have developed computational load-aware task allocation and scheduling schemes. In [63], the authors have tackled the issue of various types of VMs and have developed a parallel big data mining architecture for improving the overall resource utilization and reducing the processing cost in the environments where there exist multiple types of VMs. The work [64] has investigated a dynamic task scheduling for heterogeneous workloads. The proposed scheduling algorithm can optimize the workloads by using the complex queue models based on I/O and CPU utilization. In the work conducted by A. Verma *et al.* [65], a task scheduling scheme for optimizing the computation resource utilization while reducing the completion time under realistic workloads, has been proposed. Another direction to develop network-aware task allocation schemes have been considered in the works [66, 67, 68]. M. Asahara *et al.* takes the network characteristics into account and the proposed algorithm can void network congestion by considering the network topology information [66]. In [67], the authors have designed a novel MapReduce framework for wireless networks. Through simulation, they verify the effectiveness of MapReduce in wireless environment. The work [68] has considered radio and computing resources sharing problem and proposed a cooperative resource management to provide an efficient cloud computing in wireless networks.

Despite the significant advantages of Hadoop, this architecture still suffers from network failures of VMs. The success probability of data mining and mining speed drastically decrease when VMs cease to function due to hardware troubles or software bugs [69, 70]. To cope with this issue, the common Hadoop architecture [60] utilizes the replication approach, which increases the service availability against VM breakdowns by allocating redundant tasks to distinct VMs. In addition to this, the current Hadoop utilizes a multiple master VMs mechanism in order to increase service availability against the breakdown of master VM. However, it is difficult to ensure the service availability under real environments since an optimal number of replications or master VMs depends on the probability and scale of breakdowns. The work [71] has proposed processing scheduling technique that can shorten execution time of the data mining under failure-prone environment. The network failure issue has also been addressed in [72, 73, 74]. Bressoud *et al.* proposed a check pointing scheme to ensure the processing service even if data processing is disrupted due to network failures [72]. While this approach effectively uses computational resource in comparison with the replication approach, it increases the data traffic for check pointing. Behga *et al.* have demonstrated that the execution time of data mining under network failure-prone environments can be improved with failure prediction [73]. In the work in [74], a task allocation scheme was proposed by taking into consideration the predicted physical network failures.

Although the aforementioned works have addressed the network failure issue, they assume the scale of VM breakdowns is small. In other words, since the existing works do not take into account the large scale failures, their proposed techniques drastically decrease the performance of data mining when a large scale failure occurs, i.e., FPN. Therefore, data mining architecture that is tolerant to FPN is absolutely imperative to provide future "ubiquitous big data mining service" [75, 76]

5.3 Envisioned Big Data Mining Architecture

Since the conventional big data mining utilizes the central management architecture, the architecture has the following shortages: (*i*) it cannot operate if the master VM ceases to function i.e., single point of failure, and (*ii*) the performance drastically decreases with the increase of VMs, which results in service disruption [77]. In order to cope with these issues, we envision a parallel big data mining architecture utilizing virtual networks. In this architecture, the management functions, i.e., JobTracker and NameNode, are executed by all VMs in a distributed manner. The virtual network is constructed by all VMs and it is used to find data and processing VMs, instead of the master VM. Since all VMs execute both the management and processing functions, this architecture achieves high performance even when the failure of some VMs occurs or the number of VMs becomes large [78, 79, 80]. In other words, the envisioned architecture can keep

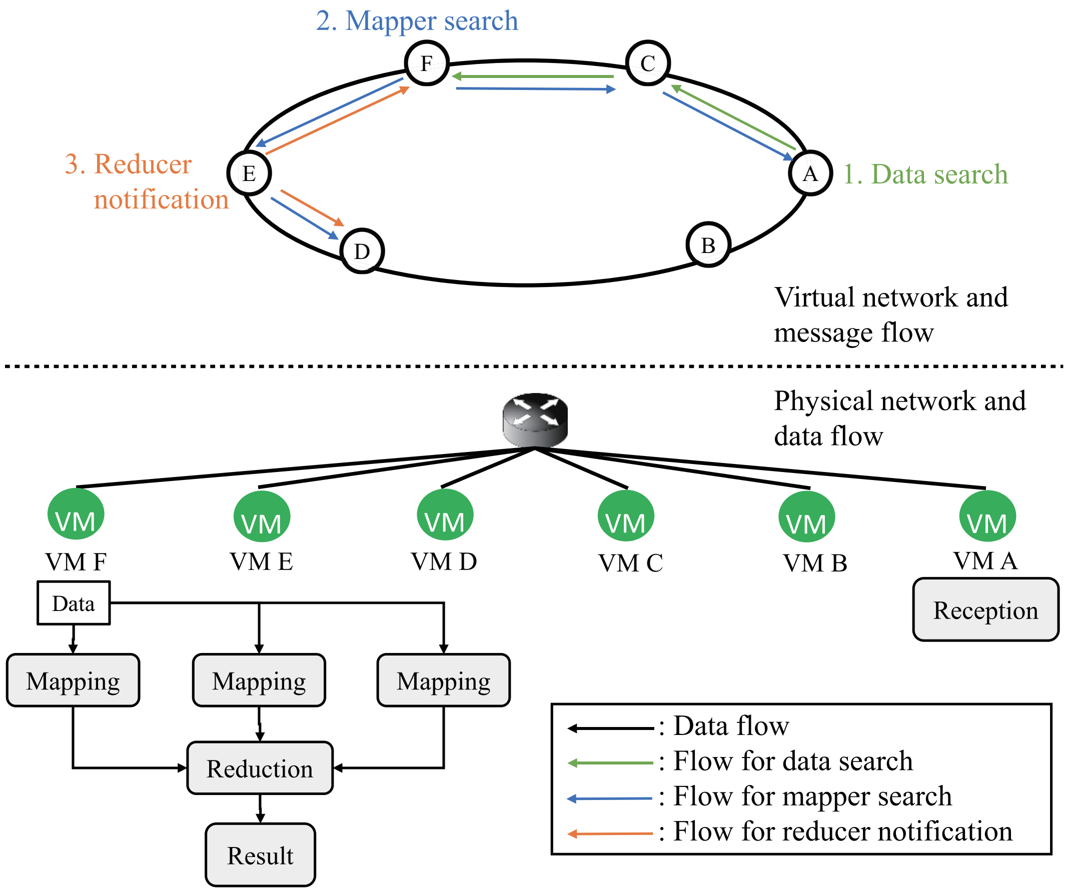


Figure 5.3: An example of our envisioned big data mining architecture and the procedure of data mining.

high-performance big data mining service unless the virtual network is disrupted.

Fig. 5.3 shows an example of our envisioned big data mining architecture and also demonstrates how our envisioned architecture executes the MapReduce and HDFS functions by using the virtual network that consists of all VMs in the network. When a data mining request is injected, a reception VM, which is the VM that received the request from a client, e.g., VM A in Fig. 5.3, executes the reception process. In the reception process, the reception VM finds the VM (or VMs), which possesses the data appertaining to the injected mining request, by using the virtual network. The VM (or VMs) is referred to as the possessor.

Here, there are many mechanisms for searching the possessor, such as flooding, hash table, and so forth. Next, the possessor (e.g., VM F in Fig. 5.3) decides the number of splits based on its own CPU utilization and partitions the data used for mining into multiple splits. Additionally, the possessor attempts to select mappers in close order in the virtual network. Here, VMs with higher CPU utilization than a certain threshold are not selected. For instance, VM D, E, F are selected as mappers in Fig. 5.3. The mappers perform the mapping process by using the split data that is transmitted from the possessor. Then, the mapper that is earliest to finish the mapping process becomes the reducer (e.g., VM E in Fig. 5.3). The reducer transmits a notification message to other mappers that requests to transmit the processed data to the reducer, and it executes the reduction process after receiving the processed data from all mappers. Finally, the reducer stores the obtained result.

As described above, all management and processing functions can be executed in a distributed manner by utilizing the virtual network. Although our envisioned architecture can solve the issue of single point of failure in the conventional architectures, the service availability of our envisioned architecture is affected by the connectivity of the virtual network [81]. Therefore, it is required to introduce a virtual network that is tolerant to failures for the big data mining architecture.

5.4 A Decentralized Method to Construct Virtual Network

In chapter 4, we proposed a method to construct a virtual network tolerant to both FPN and FVN, i.e., FAT-VN+. However, FAT-VN+ is not suitable for our envision big data mining architecture because the centralized server is a threat to failure tolerance. Therefore, we propose a distributed method to construct a virtual network, which is based on the key ideas of FAT-VN+, i.e., bimodal degree distribution and usage of physical network information. Our proposed big data mining architecture utilizing the virtual network is referred to as FA-Hadoop.

5.4.1 ID Assignment

First of all, we demonstrate our considered ID assignment in FA-Hadoop. In the FAT-VN+, IDs in the virtual network are assigned to VMs in a random manner. However, this mechanism results in low lookup efficiency in the big data mining architecture based on the virtual network because the lookup is limited to only the flooding-based method. Therefore, we first explain a mechanism to assign IDs to VMs in the virtual network.

Fig. 5.4 shows an example of the considered virtual network topology and ID assignment in the proposed FA-Hadoop. As shown in this figure, all VMs are lined up into the ring topology and IDs are assigned to each VM in a clockwise direction [82]. With this regular assignment, VMs can decide the direction of lookup while the flooding-based lookup attempts to send messages to all neighbor VMs. Therefore, this mechanism can achieve high lookup efficiency.

Because all VMs are lined up into ring topology, the virtual network topology in FA-Hadoop is different from FAT-VN+. Although the optimal condition is that the ELDVMs are evenly located in the ring topology, FA-Hadoop cannot satisfy this condition since LDVMs that are next to each HDVM in the ring should become ELDVMs to follow the optimal bimodal degree distribution. However, since the other ELDVMs are evenly located in the ring topology, it is possible to achieve higher tolerance to FVN in comparison with other virtual networks. Especially, in large scale environments, our considered virtual network in FA-Hadoop achieves almost the same performance as FAT-VN+.

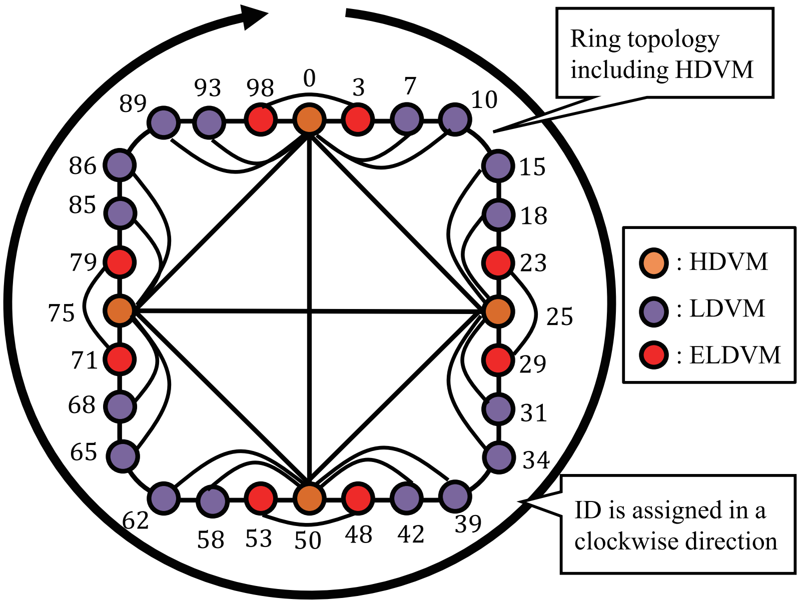


Figure 5.4: An example of virtual network and ID assignment in FA-Hadoop.

Table 5.1: Network information list.

Parameter	**Variable**
Predefined degree of LDVMs	k_{LD}
Current number of VMs	N
Current number of HDVMs	$\lvert N_{\text{HD}} \rvert$
ID of VM closest to root device	$v_{\text{root}}^{\text{cl1}}$
Hop count between $v_{\text{root}}^{\text{cl1}}$ and root device	$h_{\text{root}}^{v_{\text{root}}^{\text{cl1}}}$

5.4.2 VM Joining Procedure

Here, we propose a VM joining procedure to construct a virtual network tolerant to both FVN and FPN. A newly joining VM selects the appropriate ID and neighbor VMs in the virtual network in this procedure. While the centralized management VM constructs a virtual network in FAT-VN+, each VM autonomously executes the procedure in FA-Hadoop. Therefore, the information required for the joining procedure is shared by using the network information list, which contains the data summarized in table 5.1.

Procedure 3 demonstrates how a newly joining VM establishes its virtual

Procedure 3 VM joining procedure

```
Get network information list
Execute ID decision procedure
    /* Decide its ID, $v_i$, based on its IP address, $d_i$ */
Execute insertion procedure
    /* Establish virtual links to VMs that have close ID */
Execute expansion procedure
    /* Establish links to a HDVM and LDVMs while $k_{\mathrm{LD}} > |k_{v_i}|$ */
Calculate $N_{\mathrm{HD}}$ based on (3.12) with $N$ and $k_{\mathrm{LD}}$
if $N_{\mathrm{HD}} > |N_{\mathrm{HD}}|$ then
    if $h_{\mathrm{root}}^{v_{\mathrm{root}}^{\mathrm{cl1}}} \leq h_{\mathrm{root}}^{v_i}$ then
        Send network reconstruction request and the updated list to $v_{\mathrm{root}}^{\mathrm{cl1}}$
        /* $v_{\mathrm{root}}^{\mathrm{cl1}}$ executes network reconstruction procedure */
    else
        Execute network reconstruction procedure
    end if
end if
```

links. First, a newly joining VM contacts a reception VM and receives a network information list from the reception VM to know the information that is required for establishing links. Then, the newly joining VM executes the ID decision procedure, which is described as follows.

ID decision procedure – This procedure aims to decide the ID of the newly joining VM, v_i, based on its IP address, d_i. As described in chapter 4, the VMs with low hop count in the physical network should become the neighbor VMs in the virtual network in order to improve the FPN tolerance. Since close IP addresses of VMs imply that the hop count between the VMs is low, v_i can be decided by calculating the following hash function.

$$v_i = \mathrm{mod}(d_i, V). \tag{5.1}$$

Here, the maximum value of the ID, V, is decided with the maximum value of the IP address, D, as μD, where μ is a natural number larger than 0.

After deciding its ID, the newly joining VM starts to establish k_{LD} virtual links since we assume that all newly joining VMs become LDVMs. In order to optimally select neighbor VMs, it executes the following procedures: (*i*) the insertion procedure and (*ii*) the expansion procedure, which are described as follows:

Insertion procedure – This procedure attempts to establish the virtual links to appropriate neighbor VMs in the ring topology. First, the newly joining VM attempts to find a neighbor VM, v_i^{cl1}, which has the closest ID to its own ID. Note that, since the newly joining VM can connect only to the reception VM at this time, the reception VM finds the neighbor VM on behalf of the newly joining VM by using the virtual network. In the same way, the newly joining VM finds another neighbor VM, v_i^{cl2}, which has the second closest ID to its own ID. After finding the neighbor VMs, v_i^{cl1} and v_i^{cl2}, it creates two new links from v_i to v_i^{cl1} and v_i^{cl2}, respectively, and breaks the existing link between v_i^{cl1} and v_i^{cl2}.

Expansion procedure – Following the insertion procedure, the newly joining VM executes the expansion procedure, in which it establishes the virtual links as long as $|k_{v_i}|$ is lower than k_{LD} or candidates for neighbor VMs exist. If the newly joining VM did not establish a link to a HDVM in the insertion procedure, it initially attempts to establish a link to a HDVM, v_i^{CLHD}, which has the closest ID to its own. Then, it finds LDVMs that have lower degree than k_{LD} by using the virtual network, and establishes links to these LDVMs as long as $|k_{v_i}|$ is lower than k_{LD}.

Due to the participation of the newly joining VM, the ideal parameter of degree distribution might change. Therefore, the newly joining VM calculates the value of N_{HD} by using (3.12) with the N, which includes itself, and k_{LD}. If the ideal number of HDVMs, N_{HD}, is higher than the current number of HDVMs, $|N_{\text{HD}}|$, it is required to execute the network reconstruction procedure in order

Procedure 4 Network reconstruction procedure

```
Construct virtual links to other HDVMs
Find LDVMs that have close ID compared with other HDVMs
Establish virtual links to the LDVMs
Update network information list
```

to maintain the virtual network at an optimal state. Here, since the LDVM that is close to a root device in the physical network should be selected as a HDVM, the newly joining VM compares the hop count between itself and the root device, $h^{v_i}_{\text{root}}$, with the minimum hop count, $h^{v^{\text{cl1}}_{\text{root}}}_{\text{root}}$, which is stored in the list. If $h^{v^{\text{cl1}}_{\text{root}}}_{\text{root}} \leq h^{v_i}_{\text{root}}$, i.e., if VM $v^{\text{cl1}}_{\text{root}}$ is closer to the root device, the newly joining VM selects $v^{\text{cl1}}_{\text{root}}$ as a new HDVM, and requests $v^{\text{cl1}}_{\text{root}}$ to reconstruct the virtual network. On the other hand, if the newly joining VM is closer to the root device, it becomes a new HDVM and performs the network reconstruction procedure, which is described as follows:

Network reconstruction procedure – The objective of this procedure is to maintain the virtual network topology at an optimal state. The network reconstruction procedure, which is shown in Procedure 4, is executed by a new HDVM. First, the HDVM creates the links to the other HDVMs to construct a complete graph. Then, it finds LDVMs whose ID are closer to itself compared with the other HDVMs and creates new links to the LDVMs while breaking the existing link between the other HDVMs and the LDVMs. Finally, it increments N and $|N_{\text{HD}}|$, and updates the information of $v^{\text{cl1}}_{\text{root}}$ and $h^{v^{\text{cl1}}_{\text{root}}}_{\text{root}}$. The updated list is shared between all VMs.

5.5 A Decentralized Data Management Method

This section presents a decentralized method to manage the data in our envisioned big data mining architecture. First, a data placement procedure that is tolerant to FPN is presented. Also, a data lookup procedure based on hash table is presented.

5.5.1 Data Placement Procedure

Our proposed data placement procedure ensures the existence of any data, even when FPN occurs, by distributing data to appropriate VMs. This procedure is executed in a decentralized manner. In other words, each reception VM autonomously decides the appropriate location of data that can guarantee the existence of data even when FPN occurs.

When the data storing request is injected from the client, the reception VM starts the data placement procedure. First, it decides the hash value of the data, w_j, which is given by calculating the hash function such as SHA-1. Then, it finds a VM, v_{w_j}, which has a higher and close ID to w_j and sends the data to v_{w_j}.

In addition to this, the reception VM creates redundant data, i.e., replica, and allocates it to a distinct VM. Since the neighbor VMs will probably be removed from the virtual network by FPN, it is clearly understood that choosing farthest VM as data owner ensures the existence of the data. Therefore, the reception VM calculates the diagonal hash value, w_j^{diagonal}, in order to allocate data to the VM that is diagonally opposite to v_{w_j}. The value of w_j^{diagonal} can be expressed with w_j as follows.

$$w_j^{\text{diagonal}} = \begin{cases} w_j + V/2, & \text{if } w_j < V/2, \\ w_j - V/2, & \text{otherwise.} \end{cases} \tag{5.2}$$

Fig. 5.5 demonstrates an example of data placement, where $V = 100$. By calculating the hash function, the hash value and diagonal hash value are decided as 33 and 83, respectively. According to these values, the reception VM selects the corresponding VMs. In this example, the original data is stored by the VM

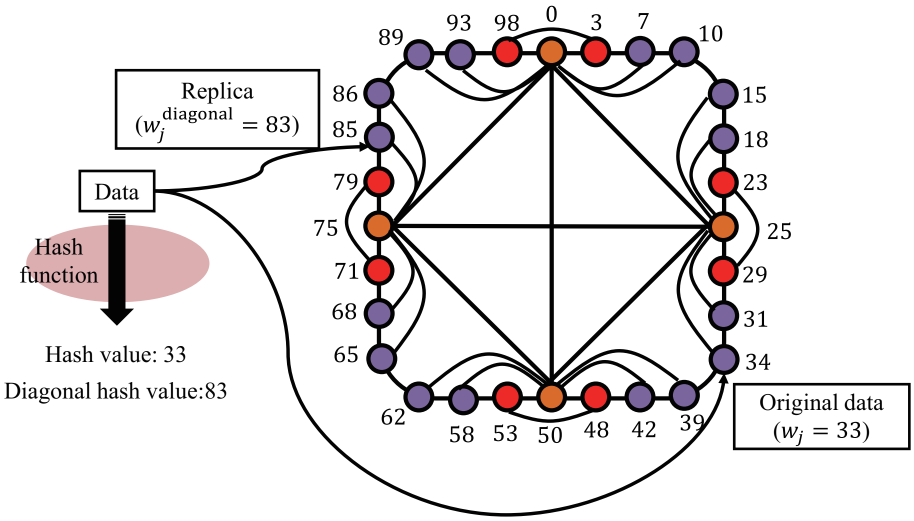


Figure 5.5: An example of data placement in our envisioned FA-Hadoop.

whose ID is 34 and the replica is sent to the VM whose ID is 85.

5.5.2 Data Lookup Procedure

Next, we explain the data lookup procedure in our envisioned architecture. Similar to the data placement procedure, the reception VM autonomously executes this procedure. In order to reduce the lookup overhead, this procedure attempts to find a data that is stored in a VM close to the reception VM.

When the big data mining request is injected, the reception VM executes the data lookup procedure to find the data required for data mining. First, it calculates the hash value and diagonal hash value of data by using the aforementioned hash functions. After that, in order to know a close VM that stores data, the reception VM calculates the differences between its ID, $v_{\text{reception}}$, and hash values, w_j and w_j^{diagonal}. Then, the reception VM transmits lookup message in the direction that reaches the VM that has the lower difference.

Fig. 5.6 demonstrates an example of data lookup, where $V = 100$ and the ID of the reception VM is 65. By calculating the hash function, the reception VM

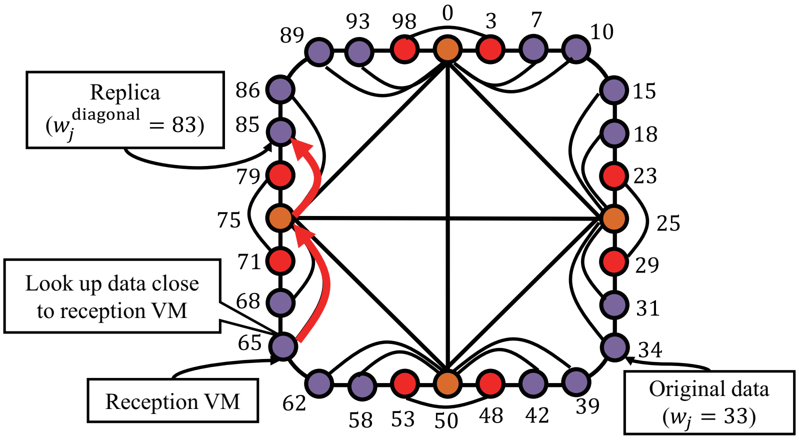


Figure 5.6: An example of data lookup in our envisioned FA-Hadoop.

obtains the hash value (33) and the diagonal hash value (83). Since the difference between its ID and diagonal hash values (18) is lower than the other value (32), the reception VM looks up the replica that is stored in the VM whose ID is 85.

5.6 Performance Evaluation in Service Availability

In this section, we use numerical calculation to evaluate the service availability of the proposed big data mining architecture, referred to as FA-Hadoop, which utilizes the proposed methods to construct a virtual network and manage data in this chapter.

5.6.1 Definition of Service Availability

Since each task uses multiple-data and each datum is replicated and distributed to distinct VMs, the considered architecture succeeds in a data mining task if there are all data appertaining to the task in the virtual network. Therefore, the success probability of each task, P_{success}, decreases with the removal of VMs.

With the number of data appertaining to the task, B, redundancy of each datum, R, and the probability that there exists a VM that has replication y of datum x in giant cluster, $a_{x,y}$, the value of P_{success} can be expressed as follows.

$$P_{\text{success}} = \prod_{x}^{B} \left\{ 1 - \prod_{y}^{R} (1 - a_{x,y}) \right\}. \tag{5.3}$$

5.6.2 The Impact of Virtual Network Construction Scheme on Service Availability

Here, we evaluate the service availability of data mining architectures which utilize the proposed virtual network construction scheme based on physical network information and the random network construction scheme, respectively.

We suppose that the physical network topology is tree-structured, the average degree of virtual networks is set to 3 and the number of VMs is $10,000$. Additionally, the supposed big data mining executes 1000 tasks, where the number of data appertaining to each task and the number of replicas are set to 5 and 2, respectively. In this setting, we evaluate the service availability by calculating the success probability of each task after FPN occurs, where we consider different scales of FPNs, i.e., small-scale, medium-scale, and large-scale FPNs. These FPNs result in approximately, 100, 1000, and 3000 VMs being unable to connect to other VMs, respectively.

Fig. 5.7 demonstrates the effect of our proposed construction scheme on the service availability of data mining for three scales of FPNs. From the result, it is clearly evident that our proposed construction scheme achieves higher service availability, which is approximately 100 percent, regardless of the scales of FPNs. The reason behind the result is that, with our proposed scheme, all data appertaining to all tasks exist in the giant cluster because the virtual network does not disrupt. On the other hand, in case of the random network construction scheme,

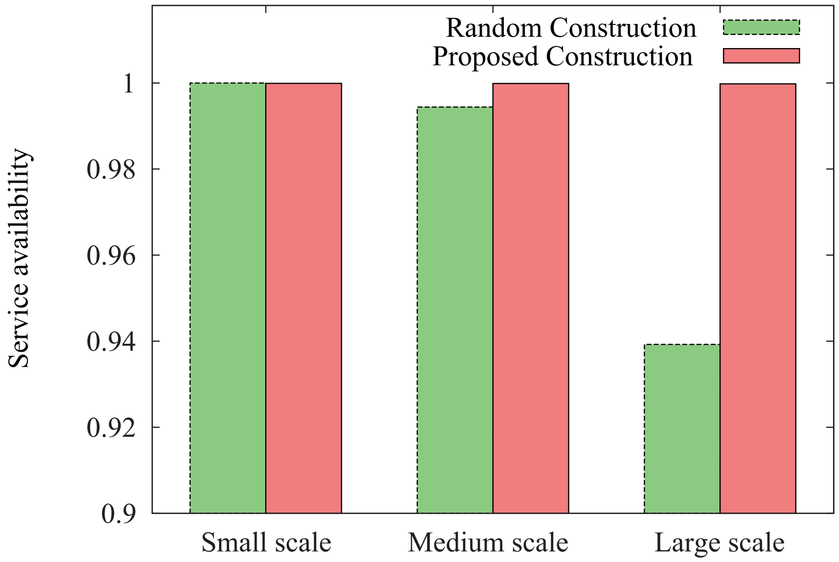


Figure 5.7: Service availability in different network construction schemes.

the service availability decreases with the increase of removed VMs. This is because the number of isolated VMs increases with the increase of removed VMs.

5.6.3 The Impact of Data Placement Scheme on Service Availability

In the remainder of this section, we verify the effectiveness of the proposed data placement scheme by comparing it to the random data placement scheme in terms of the service availability after FPN occurs. Here, the virtual network is constructed based on the proposed scheme.

We assume the same setting as the aforementioned evaluation. The physical network topology is tree-structured, the average degree of virtual networks is set to 3 and the number of VMs is $10,000$. We assume that 1000 tasks are injected and each task utilizes 5 data. While the proposed scheme replicates 2 times, the number of replicas is set to either 2, 3, or 4 in the random placement scheme. We evaluate the service availability when the number of removed VMs due to the

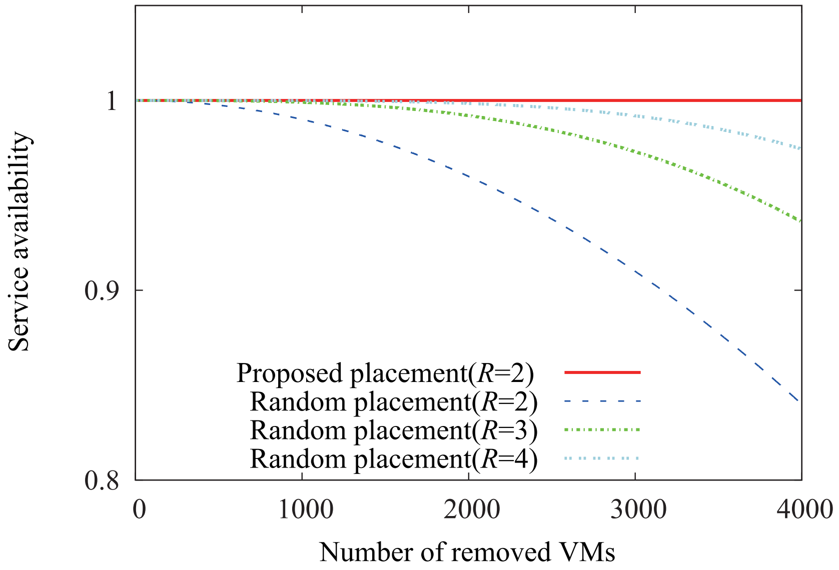


Figure 5.8: Service availability in different data placement schemes.

FPN is varied from 0 to 4000.

Fig. 5.8 demonstrates the effect of our proposed data placement scheme on the service availability of data mining. The existing data placement scheme falls to an extremely low availability with a progressive increase of the number of removed VMs even the higher redundancy. On the other hand, the proposed placement scheme achieves 100% success probability of data mining with minimum redundancy regardless of the number of removed VMs because it ensures the existence of the VMs that have the data appertaining to the task in the giant cluster. It can be concluded that our envisioned big data mining architecture with the proposed construction and placement schemes can provide big data mining service with a higher success rate even when FPN occurs.

5.7 Summary

In this chapter, we investigated the issue of big data mining architecture in failure-prone environments. First, we demonstrated that the conventional data mining architecture cannot provide service in failure-prone environments. To cope with this issue, we presented a novel big data mining architecture, which manages the big data functions in a decentralized manner and achieves high service availability in the failure-prone environment. Additionally, in order to improve the tolerance to FVN and FPN, we proposed decentralized methods to construct a virtual network and manage data. Through numerical calculations, we showed that our proposed architecture can provide big data mining service with high success probability even when FPN occurs.

Chapter 6

Conclusion

Current network architecture is at an important turning point on the path for reaching the goal of realizing the smart society based on ICT services. There are many kinds of networks and ICT services that evolved in their own way so far. These networks should be integrated to efficiently provide many kinds of ICT services. This integration approach results in high quality of experience for all users. In order to integrate the networks and services, we envision a novel network architecture that is based on virtual network technology. The envisioned architecture is suitable for providing the many kinds of services in the integrated networks. On the other hand, the performance of the envisioned architecture differs according to the virtual network construction schemes. Additionally, since the performance of virtual networks drastically decreases due to the network failures, we must address the fault-tolerance issue of the virtual network. Because of this research background, this work aims to develop the fault-tolerant virtual network technology, which will be used for future promising network architecture. Our contributions are summarized as follows:

1. In Chapter 1, we provided the research background and purpose of this thesis.

2. In Chapter 2, we provided a detailed survey of existing virtual network technologies for improving the tolerance to FVN and FPN. We also highlighted the fact that these existing technologies are not appropriate to provide the communication services in the environment where both FVN and FPN occur. Additionally, we classified the existing virtual network technologies based on the method and objective to clarify the novelty of the proposed virtual network technologies.

3. In Chapter 3, we proposed a novel virtual network topology that is based on bimodal degree distribution, which can achieve a higher tolerance to AVM and FVM. Additionally, by considering the location of each VM, we optimize the virtual network topology for maximizing the tolerance to AVM. As a consequence, the proposed topology, referred to as FAT-VN, can achieve high connectivity in comparison with the existing virtual networks.

4. In chapter 4, we presented a method to construct FAT-VN. First, we demonstrated that FAT-VN drastically decreases the connectivity when FPN occurs. To mitigate the effect of the FPN, we proposed a method to construct a virtual network by utilizing the topology information of the physical network. Additionally, we constructed a framework to evaluate the service availability of the virtual networks. Also, we confirmed the effectiveness of the proposal by numerical evaluation.

5. In chapter 5, we provided a novel big data mining architecture, which is possible to improve the tolerance to failures. In order to solve the single point of failure problem in the conventional architecture, all VMs execute the management and processing functions in our envisioned architecture. We also proposed an autonomous and distributed algorithm to construct an appropriate virtual network. Through numerical calculations, we confirm

that the proposed architecture can achieve the desired service availability with the lowest data replications.

This dissertation proposed the fault-tolerant virtual network technologies, which can satisfy the requirements of future ICT services. In other words, we succeeded in the establishment of the basic technologies for a new-generation network architecture. This work will contribute to create new communication technologies and promote the practical ICT services that are based on the virtual network technologies.

Bibliography

[1] B. Ahlgren, P. Aranda, P. Chemouil, S. Oueslati, L. Correia, H. Karl, M. Sollner, and A. Welin, “Content, connectivity, and cloud: ingredients for the network of the future,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 62–70, Jul. 2011.

[2] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, “A survey of naming and routing in information-centric networks,” *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–54, Dec. 2012.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[4] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, “A survey of information-centric networking research,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, Second Quarter 2014.

[5] A. Kumar, S. Merugu, X. Jun, and Y. Xingxing, “Ulysses: a robust, low-diameter, low-latency peer-to-peer network,” in *Proc. of IEEE International Conference on Network Protocols*, Atlanta, Georgia, USA, Nov. 2003, pp. 258–267.

[6] W. Xiao, M. He, and H. Lian, “Cayleyccc: A robust p2p overlay network with simple routing and small-world,” *Academy Publisher Journal of Networks*, vol. 6, no. 9, pp. 1247–1253, Sep. 2011.

[7] P. Flocchini, A. Nayak, and M. Xie, “Enhancing peer-to-peer systems through redundancy,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 15–24, Jan. 2007.

[8] A. Datta, S. Girdzijauskas, and K. Aberer, “On de bruijn routing in distributed hash tables: There and back again,” in *Proc. of International Conference on Peer-to-Peer Computing*, Zurich, Switzerland, Aug. 2004, pp. 159–166.

[9] S. Voulgaris, D. Gavidia, and M. V. Steen, “Cyclon: Inexpensive membership management for unstructured p2p overlays,” *Journal of Network and Systems Management*, vol. 13, no. 2, pp. 197–217, Jun. 2005.

[10] R. H. Wouhaybi and A. T. Campbell, “Phenix: Supporting resilient lowdiameter peer-to-peer topologies,” in *Proc. of the Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004, pp. 108–119.

[11] M. Sasabe, N. Wakamiya, and M. Murata, “Llr: A construction scheme of a low-diameter, location-aware, and resilient p2p network,” in *Proc. of IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Atlanta, GA, Nov. 2006, pp. 1–8.

[12] E. Bulut and B. Szymanski, “Constructing limited scale-free topologies over peer-to-peer networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 919–928, Apr. 2014.

[13] Y. Yu, C. Shan-zhi, L. Xin, and W. Yan, “Rmap: An algorithm of virtual network resilience mapping,” in *Proc. of 7th International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, Sept. 2011, pp. 1–4.

[14] A. Jarray and A. Karmouch, “Cost-efficient mapping for fault-tolerant virtual networks,” *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 668–681, Mar. 2015.

[15] C. Meixner, F. Dikbiyik, M. Tornatore, C. Chuah, and B. Mukherjee, “Disaster-resilient virtual-network mapping and adaptation in optical networks,” in *Proc. of 17th International Conference on Optical Network Design and Modeling*, Brest, France, Apr. 2013, pp. 107–112.

[16] X. Liu, Y. Wang, A. Xiao, X. Qiu, and W. Li, “Disaster-prediction based virtual network mapping against multiple regional failures,” in *Proc. of 2015 IFIP/IEEE International Symposium on Integrated Network Management*, Ottawa, Canada, May 2015, pp. 371–378.

[17] Ya-lian, X. song Qiu, and S. li Zhang, “Fault diagnosis in network virtualization environment,” in *Proc. of 18th International Conference on Telecommunications*, May 2011, pp. 517–522.

[18] M. Le and Y. Tamir, “Fault injection in virtualized systems–challenges and applications,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 3, pp. 284–297, May-Jun. 2015.

[19] M. Rahman and R. Boutaba, "Svne: Survivable virtual network embedding algorithms for network virtualization," *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 105–118, June 2013.

[20] G. Shields, "5 ways your vmware and hyper-v backups may fail you," Microsoft, Tech. Rep., 2012.

[21] A. J. Ganesh, A. Kermarrec, and L. Massouli, "Peer-to-peer membership management for gossip-based protocols," *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 139–149, February 2003.

[22] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, June 2001, pp. 329–350.

[23] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, February 2003.

[24] R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 401, pp. 130–131, September 1999.

[25] A.-L. Barabasi, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life.* Plume, 2003.

[26] R. Albert, H. Jeong, and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, January 2002.

[27] Y. Wang, X. Yun, and Y. Li, "Analyzing the characteristics of gnutella overlays," in *Proc. of International Conference on Information Technology-New Generations*, April 2007, pp. 1095–1100.

[28] D. Stutzbach, S. Zhao, and R. Rejaie, "Characterizing files in the modern gnutella network," *ACM/Springer Multimedia Systems Journal*, vol. 13, no. 1, pp. 25–50, March 2007.

[29] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing unstructured overlay topologies in modern p2p file-sharing systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 267–280, April 2008.

[30] S. V. Buldrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, pp. 1025–1028, April 2010.

[31] T. Tanizawa, "Percolation on correlated complex networks," *JSS Computer Software*, vol. 28, no. 1, pp. 135–144, February 2011.

[32] G. Paul, S. Sreenivasan, and H. E. Stanley, "Resilience of complex networks to random breakdown," *Physical Review E*, vol. 72, no. 5, pp. 1–9, July 2005.

[33] T. Tanizawa, G. Paul, R. Cohen, S. Havlin, and H. E. Stanley, "Optimization of network robustness to waves of targeted and random attacks," *Physical Review E*, vol. 71, no. 4, pp. 1–4, April 2005.

[34] Y. Shiraki and Y. Kabashima, "Cavity analysis on the robustness of random networks against targeted attacks: Influences of degree-degree correlations," *Physical Review E*, vol. 82, no. 2, pp. 1–12, September 2010.

[35] A. R. Sonawane, A. Bhattacharyay, M. S. Santhanam, and G. Ambika, "Evolving networks with bimodal degree distribution," *The European Physical Journal B*, vol. 85, no. 4, pp. 1–6, April 2012.

[36] B. Mitra, S. Ghose, and N. Ganguly, "How stable are large superpeer networks against attack?" in *Proc. of IEEE International Conference on Peer to Peer Computing*, September 2007, pp. 239–242.

[37] B. Mitra, F. Peruani, S. Ghose, and N. Ganguly, "Analyzing the vulnerability of superpeer networks against attack," in *Proc. of ACM Conference on Computer and Communications Security*, Oct.-Nov. 2007, pp. 225–234.

[38] B. Mitra, A. K. Dubey, S. Ghose, and N. Ganguly, "Formal understanding of the emergence of superpeer networks: A complex network approach," in *Proc. of International Conference on Distributed Computing and Networking*, January 2010, pp. 219–230.

[39] A. Srivastava, B. Mitra, F. Peruani, and N. Ganguly, "Attacks on correlated peer-to-peer networks: An analytical study," in *Proc. of International Workshop on Security in Computers, Networking and Communications*, April 2011, pp. 1093–1098.

[40] T. Paul, G. an Tanizawa, S. Havlin, and H. E. Stanley, "Resilience of complex networks to random breakdown," *European Physics Journal B*, vol. 38, pp. 187–191, April 2004.

[41] Y. Matsumoto, "Ruby [Online]." Available: http://www.ruby-lang.org/en/.

[42] S. Sun, Z. Liu, Z. Chen, and Z. Yuan, “Error and attack tolerance of evolving networks with local preferential attachment,” *Physica A: Statistical and Theoretical Physics*, vol. 373, no. 1, pp. 851–860, Jan. 2007.

[43] M. Molloy and B. Reed, “The size of the giant component of a random graph with a given degree sequence,” *Combinatorics, Probability, and Computing*, vol. 7, no. 3, pp. 295–305, Sep. 2000.

[44] O. Galinina, A. Pyattaev, S. Andreev, M. Dohler, and Y. Koucheryavy, “5g multi-rat lte-wifi ultra-dense small cells: Performance dynamics, architecture, and trends,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1224–1240, Mar. 2015.

[45] B. H. Jung, N.-O. Song, and D. K. Sung, “A network-assisted user-centric wifi-offloading model for maximizing per-user throughput in a heterogeneous network,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1940–1945, Mar. 2014.

[46] K. Saito, H. Nishiyama, N. Kato, H. Ujikawa, and K.-I. Suzuki, “A mpcp-based centralized rate control method for mobile stations in fiwi access networks,” *IEEE Wireless Communications Letters*, vol. 4, no. 2, pp. 205–208, Apr. 2015.

[47] P. Gill, N. Jain, and N. Nagappan, “Understanding network failures in data centers: measurement, analysis, and implications,” in *Proc. of the ACM SGCOMM 2011 conference*, Toronto, Aug. 2011, pp. 350–361.

[48] S. Trade, “Square trade research: iPhone more reliable than blackBerry, one year in [Online],” Available: http://www.squaretrade.com/htm/pdf/SquareTrade_iPhone_Study_1108.pdf.

[49] A. Asadi and V. Mancuso, “Wifi direct and lte d2d in action,” in *Proc. of 2013 IFIP Wireless Days (WD)*, Valencia, Nov. 2013, pp. 1–8.

[50] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with wi-fi direct: overview and experimentation,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, Jun. 2013.

[51] H. Nishiyama, M. Ito, and N. Kato, “Relay-by-smartphone: Realizing multihop device-to-device communications,” *IEEE Communications Magazine*, vol. 52, no. 4, pp. 56–65, Apr. 2014.

[52] A. Hikuma, Y. Fuke, N. Nakaminami, H. Ohyane, and H. Kobayashi, "Radio base stations equipments toward economical expansion of foma coverage areas," *NTT DoCoMo Technical Journal*, vol. 6, no. 1, pp. 52–59, Jun. 2004.

[53] Cisco, "Cisco Catalyst 6500-E Series Chassis [Online]," Available: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-6500-series-switches/data_sheet_c78-708665.pdf.

[54] Intel, "Calculated MTBF Estimates [Online]," Available: http://download.intel.com/support/motherboards/server/sb/s3420gpmtbfcalculationrev10.pdf.

[55] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 208701, pp. 1–5, Oct. 2002.

[56] Apache, "Hadoop [Online]," Available: http://hadoop.apache.org/.

[57] F. Gebara, H. Hofstee, J. Hayes, and A. Hylick, "Big data text-oriented benchmark creation for hadoop," *IBM Journal of Research and Development*, vol. 57, no. 3/4, pp. 10:1–10:6, May.-Jul. 2013.

[58] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Proc. of 6th Symposium on Operating Systems Design and Implementation*, San Francisco, USA, Dec. 2004, pp. 137–150.

[59] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *Proc. of 19th Symposium on Operating Systems Principles*, New York, USA, Oct. 2003, pp. 29–43.

[60] Apache, "HDFS Fereration [Online]," Available: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/Federation.html.

[61] M. Cardosa, A. Singh, H. Pucha, and A. Chandra, "Exploiting spatio-temporal tradeoffs for energy-aware mapreduce in the cloud," *IEEE Transactions on Computers*, vol. 61, no. 12, pp. 1737–1751, Dec. 2012.

[62] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "Priter: A distributed framework for prioritizing iterative computations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1884–1803, Sept. 2013.

[63] D. Warneke and O. Kao, "Exploiting dynamic resource allocation for efficient parallel data processing in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 985–997, Jun. 2011.

[64] C. Tian, H. Zhou, Y. He, and L. Zha, “A dynamic mapreduce scheduler for heterogeneous workloads,” in *Proc. of 9th International Conference on Grid and Cooperative Computing*, China, Aug. 2009, pp. 218–224.

[65] A. Verma, L. Cherkasova, and R. H. Campbell, “Orchestrating an ensemble of mapreduce jobs for minimizing their makespan,” *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 314–327, Spt.-Oct. 2013.

[66] M. Asahara, S. Nakadai, and T. Araki, “Loadatomizer: a locality and i/o load aware task scheduler for mapreduce,” in *Proc. of IEEE 4th International Conference on Cloud Computing Technology and Science*, Taipei, Dec. 2012, pp. 317–324.

[67] H. Kim, J. Jung, M. Bae, and H. Kim, “A simulation study on map/reduce framework in wireless data center environment,” in *Proc. of International Conference on ICT Convergence*, Jeju, Korea, Oct. 2013, pp. 440–445.

[68] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, “A framework for co-operative resource management in mobile cloud computing,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[69] A. Rabkin and R. H. Katz, “How hadoop clusters break,” *IEEE Software Magazine*, vol. 30, no. 4, pp. 88–94, Jul.-Aug. 2013.

[70] H. Jin, X. Yang, X.-H. Sun, and I. Raicu, “Large-scale distributed systems at google: Current systems and future directions,” in *in Keynote Speech at the 3rd ACM SIGOPS International Workshop Large Scale Distributed Systems and Middleware*, Montana, USA, Oct. 2009.

[71] Q. Zheng, “Improving mapreduce fault tolerance in the cloud,” in *Proc. of IEEE International Symposium on Parallel and Distributed Processing Workshop*, Atlanta, USA, Apr. 2010, pp. 1–6.

[72] T. C. Bressoud and M. A. Kozuch, “Cluster fault-tolerance: An experimental evaluation of checkpointing and mapreduce through simulation,” in *Proc. of IEEE International Conference on Cluster Computing and Workshops*, New Orleans, USA, Aug.-Sept 2009, pp. 1–10.

[73] A. Bahga and V. K. Madisett, “Analyzing massive machine maintenance data in a computing cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1832–1843, Oct. 2012.

[74] J. Dean, "Adapt: Availability-aware mapreduce data placement for non-dedicated distributed computing," in *Proc. of 32nd IEEE International Conference on Distributed Computing Systems*, Macau, Jun. 2012, pp. 516–525.

[75] C. He, D. Weitzel, D. Swanson, and Y. Lu, "Hog: Distributed hadoop mapreduce on the grid," in *Proc. of SC Companion: High Performance Computing, Networking, Storage and Analysis*, Salt Lake City, USA, Nov. 2012, pp. 1276–1283.

[76] N. Elmqvist and P. Irani, "Ubiquitous analytics: Interacting with big data anywhere, anytime," *IEEE Computer Magazine*, vol. 46, no. 4, pp. 86–89, Apr. 2013.

[77] K. McKusick and S. Quinlan, "Gfs: Evolution on fast-forward," *ACM Queue Magazine*, vol. 7, no. 7, pp. 1–11, Aug. 2009.

[78] F. Azzedin, "Towards a scalable hdfs architecture," in *Proc. of International Conference on Collaboration Technologies and Systems*, San Diego, USA, May 2013, pp. 155–161.

[79] J. Zhang, G. Wu, X. Hu, and X. Wu, "A distributed cache for hadoop distributed file system in real-time cloud services," in *Proc. of ACM/IEEE 13th International Conference on Grid Computing*, Beijing, China, Sept. 2012, pp. 12–21.

[80] S. Loughran, J. M. A. Calero, A. Farrell, J. Kirschnick, and J. Guijarro, "Dynamic cloud deployment of a mapreduce architecture," *IEEE Internet Computing*, vol. 16, no. 6, pp. 40–50, Nov.-Dec. 2012.

[81] H.-C. Hsiao, H.-Y. Chung, H. Shen, and Y.-C. Chao, "Load rebalancing for distributed file systems in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 951–962, May 2013.

[82] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, USA, Aug. 2001, pp. 149–160.

Publications

Journals

[1] K. Suto, K. Miyanabe, H. Nishiyama, N. Kato, H. Ujikawa, and K. Suzuki, "QoE-Guaranteed and Power-Efficient Network Operation for Cloud Radio Access Network with Power over Fiber," *IEEE Transactions on Computational Social Systems*, vol. 2 no. 4, pp. 127-136, Dec. 2015.

[2] K. Miyanabe, K. Suto, Z. Md. Fadlullah, H. Nishiyama, N. Kato, H. Ujikawa, and K. Suzuki, "A Cloud Radio Access Network with Power over Fiber toward 5G Network: QoE-Guaranteed Design and Operation," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 58-64, Aug. 2015.

[3] K. Suto, H. Nishiyama, N. Kato, and C. Huang, "An Energy-Efficient and Delay-Aware Wireless Computing System for Industrial Wireless Sensor Networks," *IEEE Access*, vol. 3, pp.1026-1035, Jul. 2015.

[4] K. Suto, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "An Overlay-based Data Mining Architecture Tolerant to Physical Network Disruptions," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 292-301, Oct. 2014.

[5] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "Toward Integrating Overlay and Physical Networks for Robust Parallel Processing Architecture," *IEEE Network*, vol. 28, no. 4, pp. 40-45, Jul.-Aug. 2014.

[6] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "THUP: A P2P Network Robust to Churn and DoS Attack based on Bimodal Degree Distribution," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 247-256, Sep. 2013.

[7] K. Suto, H. Nishiyama, S. Shen, and N. Kato, "Designing P2P Networks Tolerant to Attacks and Faults based on Bimodal Degree Distribution," *Journal of Communications*, vol. 7, no. 8, pp. 587-595, Aug. 2012.

Refereed Conference Papers

[8] Y. Lee, K. Suto, H. Nishiyama, N. Kato, H. Ujikawa, and K. Suzuki, "A Novel Network Design and Operation for Reducing Transmission Power in Cloud Radio Access Network with Power over Fiber," in *Proc. of ICCC*, Nov. 2015, pp. 1-5.

[9] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Sakano, and A. Takahara, "A Failure-Tolerant and Spectrum-Efficient Wireless Data Center Network Design for Improving Performance of Big Date Mining," in *Proc of VTC*, May 2015, pp. 1-5.

[10] K. Suto, H. Nishiyama, and N. Kato, "Context-aware Task Allocation for Fast Parallel Big Data Processing in Optical-Wireless Networks (Invited Paper)," in *Proc. of IWCMC*, Aug. 2014, pp. 423-428.

[11] S. Arai, K. Suto, and H. Nishiyama, "An Energy Efficient Upload Transmission Method in Storage-Embedded Wireless Mesh Networks," in *Proc. of ICC*, Jun. 2014, pp. 2785-2790.

[12] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, "An Overlay Network Construction Technique for Minimizing the Impact of Physical Network Disruption in Cloud Storage Systems," in *Proc. of ICNC*, Feb. 2014, pp. 68-72.

[13] K. Suto, P. Avakul, H. Nishiyama, and N. Kato, "An Efficient Data Transfer Method for Distributed Storage System over Satellite Networks," in *Proc. of VTC*, Jun. 2013, pp. 1-5.

[14] K. Suto, H. Nishiyama, H. Yoshino, K. Ishibashi, and N. Kato, "A Method to Construct an Attack and Fault Tolerant Scalable Distributed Network," in *Proc. of CMC*, May 2012, pp. 1-5.

Non-refereed Conference Papers

[15] 西山大樹, 須藤克弥, 山田直人, "災害時の通信インフラ復旧の在り方を変えるリジリエントなネットワーク技術の実現に向けて," 電子情報通信学会ソサイエティ大会講演論文集, 2015年_通信, no. 1, pp. 27-28, 2015年9月.

[16] 齋藤悠, 須藤克弥, 西山大樹, 加藤寧, "無線データセンタにおける規模と耐障害性の関係の評価," 電気関係学会東北支部連合大会, 2015年8月.

[17] K. Suto, H. Nishiyama, and N. Kato, “Resource-Efficient Data Placement for MapReduce in Wireless Data Center Networks,” 電子情報通信学会総合大会講演論文集, vol. 2015年_通信，no. 2，p. S-119, Mar. 2015.

[18] 須藤克弥, 西山大樹, 加藤寧, “広域分散ストレージシステムにおける効率的データ転送方式,” 電子情報通信学会ソサイエティ大会講演論文集, vol. 2013年_通信, no. 1，p. 297, 2013年9月.

[19] 須藤克弥, 西山大樹, 加藤寧, 仲地孝之, 藤井竜也, 高原厚, “仮想ストレージにおけるiSCSI通信の基礎実験,” 電子情報通信学会総合大会講演論文集, vol. 2013年_通信，no. 2，p. 44, 2013年3月.

[20] K. Suto, H. Nishiyama, and N. Kato, “A Study on Distributed Storage System with Erasure Coding over Satellite Networks,” *Technical Report of IEICE*, vol. 112, no. 255, SAT2012-46, pp. 131-135, Oct. 2012.

[21] 須藤克弥, 西山大樹, 加藤寧, 仲地孝之, 藤井竜也, 高原厚, “攻撃・離脱に強い階層型P2Pネットワークトポロジに関する一考察,” 電子情報通信学会技術研究報告, vol. 112, no. 231, NS2012-83, pp. 19-24, 2012年10月.

Workshop

[22] K. Suto, “A Study on Task Allocation Scheme for Fast Big Data Mining in Optical-Wireless Data Center Networks,” *Annual Workshop on A3 Foresight Program*, Jul. 2014.

[23] K. Suto, “A Study on Overlay Networks Tolerant to Attacks on Routers,” *Annual Workshop on A3 Foresight Program*, Jul. 2013.

[24] K. Suto, “A Study on Overlay Networks Tolerant to Attacks on Routers,” *Annual Workshop on A3 Foresight Program*, Jul. 2013.

[25] K. Suto, H. Nishiyama, and N. Kato, “A Study on Disaster Resilient Distributed Storage Systems,” *Annual Workshop on A3 Foresight Program*, Feb. 2013.

[26] K. Suto, “A Study on Designing P2P Networks Tolerant to Attacks and Faults Based on Bimodal Degree Distribution,” *Workshop on A3 Foresight Program*, Feb. 2012.

Awards

1. Certificate of appreciation from the municipality of San Remigio, Republic of the Philippines, Nov. 2015
2. IEEE/CIC ICCC 2015 Best Paper Award, Nov. 2015
3. IEEE VTS Japan Encourage Award, May. 2015
4. IEICE Encourage Award, Mar. 2014
5. IEEE VTC 2013-Spring Best Paper Award, Jun. 2013
6. Dean's Award of Graduate School of Information Sciences, Tohoku University, Mar. 2013

Copyright Permissions

We enclose the permissions that were used to write this dissertation. Please see the attached documents for a detailed description of permissions.

1. Certificate of Permission to reuse the paper entitled "THUP: A P2P Network Robust to Churn and DoS Attack based on Bimodal Degree Distribution"

2. Certificate of Permission to reuse the paper entitled "Designing P2P Networks Tolerant to Attacks and Faults based on Bimodal Degree Distribution"

3. Certificate of Permission to reuse the paper entitled "Toward Integrating Overlay and Physical Networks for Robust Parallel Processing Architecture"

4. Certificate of Permission to reuse the paper entitled "An Overlay-based Data Mining Architecture Tolerant to Physical Network Disruptions"

5. Certificate of Permission to reuse the paper entitled "An Overlay Network Construction Technique for Minimizing the Impact of Physical Network Disruption in Cloud Storage Systems"

Copyright Clearance Center RightsLink®

Home | Create Account | Help | Live Chat

IEEE Requesting permission to reuse content from an IEEE publication

Title:	THUP: A P2P Network Robust to Churn and DoS Attack Based on Bimodal Degree Distribution
Author:	Suto, K.; Nishiyama, H.; Kato, N.; Nakachi, T.; Fujii, T.; Takahara, A.
Publication:	Selected Areas in Communications, IEEE Journal on
Publisher:	IEEE
Date:	September 2013

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

ACADEMY PUBLISHER
http://www.academypublisher.com/

Copyright Transfer Agreement

Title of Contribution Designing P2P Networks Tolerant to Attacks and Faults Based on Bimodal Degree Distribution

Author(s) Katsuya Suto, Hiroki Nishiyama, Xuemin (Sherman) Shen, and Nei Kato Paper ID 1569495643

Publication Title (Journal, Conference, etc.) Journal of Communications, Special Issue on Security and Privacy in Communication systems and Networks

1. The copyright to the Contribution identified above, if published in the Publication, is hereby irrevocably transferred to Academy Publisher. The copyright transfer covers the exclusive right to electronically store, reproduce and distribute the Contribution wholly or in part, throughout the World, including publish, reprints, translations, photographic reproductions, microform, electronic form (offline, online), or any other reproductions of similar nature.
2. The undersigned warrants and represents that the Contribution is original and has not been previously published elsewhere in a similar form, and that he/she is the author of the Contribution, except for material that is clearly identified as to its original source, with permission notices from the copyright owners where required, and that he/she has full power and authority to make and execute this paper submission and grant for Academy Publisher, and that he/she has not granted or assigned any rights in the Contribution to any other person, publisher or entity, and that the Contribution is a copyrightable, and that the Contribution does not infringe upon any existing right of privacy, copyright, trademark, patent, statutory right, or propriety right of others, and that the Contribution does not contain infringing, libellous, obscene or other unlawful matter, and that he/she has obtained any necessary permissions from the copyright owner or his legal representative whenever a text passage from copyrighted material is quoted or a table, data, figure or other material from such material is used.
3. Should the Contribution contain any material protected by the copyright of others, it is the Author's responsibility to obtain permission from the copyright owner to reproduce any such material in his/her work prior to its submission and subsequent publication. The undersigned agree to supply such permission(s) to Academy Publisher upon request. The undersigned will defend and indemnify Academy Publisher for, and hold Academy Publisher harmless from any loss, expense or damage occasioned by, any claim or suit by a third party for copyright infringement or arising out of any breach of the foregoing warranties as a result of publication of the Contribution.
4. The undersigned declares that any person named as co-author of the Contribution is aware of the submission and has agreed to being so named. The undersigned signs for and accepts responsibility for releasing this material on behalf of any and all co-authors. The undersigned declares that statements and opinions given in the Contribution are the expression of the Authors. Responsibility for the content of the Contribution rests upon the Authors, not Academy Publisher.
5. The Authors retain all proprietary rights, other than copyright, such as patent rights. The Authors retain the right to reuse any portion of the work, without charge, in future works of the author's own, including books, lectures and presentations in all media, provided that the publication by Academy Publisher is properly credited and the relevant copyright notice is prominently included. The Authors retain the right to post the Contribution on their personal Web Page and on a publicly accessible server of their employer. Such posting is limited to noncommercial access and personal use by others, and it must be clearly pointed out, by prominently adding "© Academy Publisher", that the copyright for this Contribution is held by Academy Publisher. For all uses not covered above, the Authors must request permission from Academy Publisher to reproduce or authorize the reproduction of the Work or material extracted verbatim from the Work, including figures and tables. Although the Authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. Academy Publisher must handle all such third-party requests.
6. This agreement is to be signed by at least one of the Authors who has obtained the assent of the co-author(s) where applicable. If the Authors are employed and prepared their paper as a part of their job, the rights to the Contribution initially rest with their employer. In that case, when the undersigned signs the copyright transfer agreement, Academy Publisher assumes the undersigned is authorized to do so by his/her employer and his/her employer has consented to all the terms and conditions of this agreement.

☑ **Crown Copyright**. The Contribution is subject to Crown Copyright and is not assigned to the Academy Publisher as set forth in item 1 above. The undersigned acknowledges, however, that the Academy Publisher has the right to publish, distribute and reprint the Contribution in all forms and media, and indicates the acceptance of all the other items above except item 1. (where applicable)

Printed Name Katsuya Suto Date (Day/Month/Year) 20/12/2011

Signature Katsuya Suto

PLEASE SIGN THE FORM AND SEND BY MAIL, FAX OR AS AN EMAIL ATTACHMENT TO:

Academy Publisher, P.O.Box 40, FIN-90571, Oulu, Finland
Fax: +358-207-81-8199.
E-mail: editorial@academypublisher.com

Copyright Clearance Center RightsLink®

Home | Create Account | Help | Live Chat

IEEE Requesting permission to reuse content from an IEEE publication

Title:	Toward integrating overlay and physical networks for robust parallel processing architecture
Author:	Suto, K.; Nishiyama, H.; Kato, N.; Nakachi, T.; Fujii, T.; Takahara, A.
Publication:	IEEE Network: The Magazine of Global Internetworking
Publisher:	IEEE
Date:	July-August 2014

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

Copyright Clearance Center RightsLink®

Home Create Account Help Live Chat

IEEE Requesting permission to reuse content from an IEEE publication

Title: An Overlay-Based Data Mining Architecture Tolerant to Physical Network Disruptions

Author: Suto, K.; Nishiyama, H.; Kato, N.; Mizutani, K.; Akashi, O.; Takahara, A.

Publication: IEEE Transactions on Emerging Topics in Computing

Publisher: IEEE

Date: Sept. 2014

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Copyright Clearance Center RightsLink®

Home | Create Account | Help | Live Chat

IEEE Requesting permission to reuse content from an IEEE publication

Title:	An overlay network construction technique for minimizing the impact of physical network disruption in cloud storage systems
Conference Proceedings:	Computing, Networking and Communications (ICNC), 2014 International Conference on
Author:	Suto, K.; Nishiyama, H.; Kato, N.; Nakachi, T.; Fujii, T.; Takahara, A.
Publisher:	IEEE
Date:	3-6 Feb. 2014

Copyright © 2014, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW