Solution Approaches for Wide-area Distributed Systems toward Integration of Enterprise Networks and Computing Resources

January 2012

Yukio OGAWA

Solution Approaches for Wide-area Distributed Systems toward Integration of Enterprise Networks and Computing Resources

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2012

Yukio OGAWA

List of Publications

Journal Papers

1. Y. Ogawa, T. Hirata, K. Takamura, K. Yamaha, S. Saitou, K. Iwanaga, and T. Koita, "Estimating the performance of a large enterprise network for updating routing information," *IEICE TRANSACTIONS on Communications*, vol. E88-B, pp. 2054–2061, May 2005.

2. Y. Ogawa, G. Hasegawa, and M. Murata, "A transport-layer approach for improving thin-client performance in a WAN environment," *International Journal of Internet Protocol Technology*, vol. 6, pp. 172–183, Nov. 2011.

3. Y. Ogawa, G. Hasegawa, and M. Murata, "Power consumption evaluation of distributed computing network considering traffic locality," *IEICE TRANSACTIONS on Communications*, 2012. [submitted for publication].

Refereed Conference Paper

1. Y. Ogawa, A. Nakaya, K. Takamura, K. Yamaha, S. Saitou, K. Iwanaga, and T. Koita, "Estimating the performance of a large enterprise network for the updating of routing information," in *Proceedings of IEEE Workshop on IP Operations and Management (IPOM 2002)*, pp. 161–165, Oct. 2002.

2. Y. Ogawa, G. Hasegawa, and M. Murata, “Transport-layer optimization for thin-client systems,” in *Proceedings of IEEE Workshop on Communications Quality and Reliability (CQR 2007)*, May 2007.

3. Y. Ogawa, G. Hasegawa, and M. Murata, “A transport layer approach for improving interactive user experience on thin clients,” in *Proceedings of Australasian Telecommunication Networks and Applications Conference (ATNAC 2009)*, Nov. 2009.

4. Y. Ogawa, G. Hasegawa, and M. Murata, “Effect of traffic locality on power consumption of distributed computing network,” in *Proceedings of 9th International Conference on Communications (COMM 2012)*, June 2012. [submitted for publication].

Non-Refereed Technical Papers

1. Y. Ogawa, G. Hasegawa, and M. Murata, “Delay analysis and transport-layer optimization for improving performance of thin-client traffic,” *Technical Report of IEICE*, vol. IN2008-56, pp. 75–80, Sept. 2008. (in Japanese).

2. Y. Ogawa, G. Hasegawa, M. Murata, and S. Nishimura, “Performance evaluation of distributed computing environment considering power consumption,” *Technical Report of IEICE*, vol. IN2009-172, pp. 169–174, Mar. 2010. (in Japanese).

Preface

Enterprise networks and applications performed over them have been changing according to the enterprises' strategies for reducing their total cost of ownership (TCO). As a wide area network (WAN) provides high-bandwidth connectivity (i.e., more than several megabits per second), this change has proceeded through roughly three phases. The first phase is the integration of multiple private WANs, as well as computing resources (i.e., servers and storage systems) used for applications in private data centers, e.g., when new enterprises are established through the merger of existing companies after 2000. The second phase is the consolidation of computing resources, such as desktop computers and shared storage equipments, from branches to a private data center, which has become widespread since around 2006. The third phase is the consolidation (i.e., outsourcing) of computing resources from private data centers to public data centers, utilizing *cloud computing services*, which is currently in progress.

The above integration process involves a lot of WAN issues. Through the first-phase integration, the scale of a private WAN, such as the number of network nodes and end systems, and amount of traffic, increases. We therefore need to consider scalability issues of the private WAN. The consolidations in the second and third phases changes the end-to-end communication path between a client and a server. After the second phase, the end-to-end path is no longer limited to the inside of a branch but traverses the private WAN of an enterprise. Moreover, the third phase results in the end-to-end path traversing a public WAN connecting the enterprise to external data centers. From the perspective of an application executed over the private or public WAN, either WAN could be the bottleneck in the end-to-end path because of its lower performance. This drawback of the WAN could be apparent when the application contains not only bulk data flows but also interactive data flows. On the other hand, from the perspective of the underlying network system, the consolidation of computing resources in data centers results in a greater traffic load on the WAN. In the third phase, when outsourced applications use multimedia data related to digital videos and cameras, this outsourcing will lead to a massive amount of data traversing the public WAN, which will increase the power consumed in the WAN. We have selected several important but not well-discussed issues from those mentioned above, and studied solution approaches for the private and public WANs and applications performed over them. In this thesis, we focus on three objectives, corresponding to the three phases of the integration process, as follows.

For the first phase, we focus on evaluating the scalability of the control plane in a large enterprise network. We develop an approach to estimate the network performance for updating routing information, which is applied to a private WAN constructed by integrating two large banking networks. We first draw up a formula to represent the effect of the increase in packet traffic on the decrease in central processing unit (CPU) utilization at a router, and hence on delays in the routing information updating. Then, this formula is applied to estimate the level of CPU utilization required for routing information convergence. The results of our experiments on the network show that routing information updating could be completed as long as the average CPU utilization during any five-minute period at the routers was less than 40 %.

For the second phase, we focus on evaluating the performance of a thin-client system based on transmission control protocol (TCP), which is a typical application traversing a private WAN (and/or public WAN) after computing resources have been consolidated from branches to a private data center. We first describe the download traffic of thin-client systems as a two-state model with interactive data flows in response to keystrokes and bulk data flows related to screen updates. Since users are more sensitive to the keystroke response time, our next objective is to minimize the latency of interactive data flows, especially when the network is congested. Through simulation experiments, we reveal that the main delays

are queuing delay in the bottleneck router connected to the WAN and buffering delay in the server. We then enhance two TCP mechanisms: retransmission timeout calculation and selective acknowledgment (SACK) control, which overcome the drawbacks of existing options and increase the interval between occurrences of large delays (more than about 1 second) by about four times (up to about 2,500 seconds).

For the third phase, we focus on evaluating the power consumed in a public WAN after computing resources have been consolidated in a few huge public data centers. Such consolidation is becoming widespread not only among enterprises but also homes and public offices. Therefore, in our consideration of this issue, we treated the whole society. A distributed computing network (DCN), such as a content delivery network, not only improves the response time to clients but also reduces the traffic to and from the data center over the public WAN, thereby decreasing the power consumed in the WAN. We concentrate on the energy-saving aspect of the DCN and evaluate its effectiveness, especially considering traffic locality, i.e., the amount of traffic related to the geographical vicinity. We first formulate the problem of optimizing the DCN power consumption. Numerical evaluations show that, when there is strong traffic locality and the router has ideal energy proportionality, the system's power consumption is reduced to about 50 % of the power consumed in the case where a DCN is not used. Moreover, this advantage becomes up to about 30 % when the data center is located farthest from the center of the network topology.

Finally, we discuss future work for the situation after the third phase of the integration process.

Acknowledgments

First of all, I would like to express my sincere appreciation to my supervisor, Professor Masayuki Murata of the Graduate School of Information Science and Technology, Osaka University, for his patient encouragement, meaningful and comprehensive advice, and valuable discussions. He directed me to the appropriate perspective in this domain and inspired me to aim at higher goals.

I am grateful to the members of my thesis committee, Professor Koso Murakami, Professor Makoto Imase, and Professor Teruo Higashino of the Graduate School of Information Science and Technology, Osaka University, and to Professor Hirotaka Nakano of the Cyber Media Center, Osaka University, for reviewing my dissertation and providing many valuable comments.

I would especially like to express my appreciation to Associate Professor Go Hasegawa of the Cyber Media Center, Osaka University, for his critical and beneficial comments and unerring guidance that greatly inspired me. My study would not have been possible without his continuous care and support.

I am most grateful to Mr. Satoshi Murabayashi, formerly of UFJ Bank Ltd. and Mr. Akihiro Nakaya and Mr. Teruhiro Hirata, formerly of UFIT. Co., Ltd. for giving me an opportunity in 2002 to participate in the project constructing one of the largest enterprise networks in the world. This experience served as a starting point for my career in the information networking area.

I am grateful to Dr. Takashi Hotta, General Manager of Hitachi Ltd., Yokohama Research Laboratory, Mr. Naoto Matsunami, General Manager of Hitachi Ltd., Yokohama

Research Laboratory, and Mr. Minoru Koizumi, Chief Staff Researcher of Hitachi Ltd., Yokohama Research Laboratory, for giving me an opportunity to study for a doctorate at Osaka University.

I am grateful to Mr. Takeshi Harakawa, Dr. Eiji Ohira, and Mr. Kouichi Iwanaga, formerly of Hitachi Ltd.; Mr. Shinichi Migiuma, Mr. Kouji Takamura, Mr. Shinji Kimura, Dr. Shinji Nishimura, Mr. Takashi Tashiro, Mr. Osamu Takada, Dr. Susumu Matsui, Mr. Satomu Saitou, Mr. Satoshi Hasegawa, Mr. Naoki Ishii, Mr. Yuji Onishi, and Mr. Kiyokazu Saigou of Hitachi Ltd.; Mr. Hiroshi Nishikawa of Hitachi Systems, Ltd.; and Mr. Keiichi Yamaha and Mr. Tsutomu Koita of Hitachi Information & Communication Engineering, Ltd. for their valuable discussion, advice, support, and encouragement during the course of this study.

I am thankful to all the members of the Social Infrastructure Information Research Dept., Hitachi Ltd., Yokohama Research Laboratory, for their continuous support and friendship.

Finally, I deeply thank my parents, my wife, Shizue, and my daughter, Manami, for their understanding and hearty support and encouragement in my daily life. This work would not have been achieved without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Overview of Enterprise Network

Enterprise networks are managed systems based on the Internet Protocol (IP) from the perspective of availability, performance (e.g., throughput and delay), security, and so on. An overview of an enterprise network system is depicted in Figure 1.1. This is a private system having a hierarchical structure consisting of a few data centers including outsourced public data centers, a private wide area network (WAN), and a lot of branches, which are interconnected by IP routers (simply called routers here) [1]. For example, a large banking network in Japan connects a few data centers and about 2,000 branches by about 5,000 routers [2]. There are server/storage devices in the data centers and client devices in the branches, where a pair of a server and a client establishes an end-to-end connection over the private/public WAN. The private/public WAN is composed of a core network, metro/edge networks, and access networks, where the core network is usually a nationwide network and the metro/edge network is mostly a prefecture-wide network. This WAN is usually constructed by using Ethernet over a DWDM optical network [3], where DWDM stand for dense wavelength division multiplexing.

There are three methods of private WAN management. One is for routers and the links

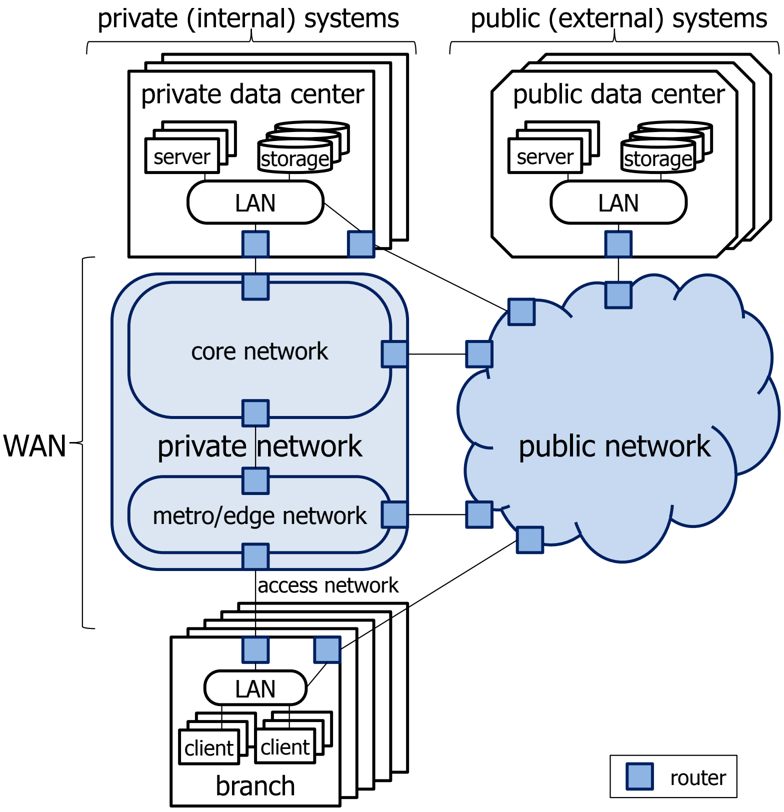


Figure 1.1: Overview of enterprise network

connecting them to be provided and operated by an enterprise, another is for IP routers (and switches for lower-level protocols like asynchronous transfer mode (ATM), in some cases) to be operated by the enterprise but the links connecting routers to be managed by a telecommunications carrier, and the third method is for the enterprise to outsource the entire WAN operations to the telecommunications carrier. The first method would be applied only to particular types of networks such as the network interconnecting two private data centers. Although large enterprises tend to select the second method while small and medium-sized enterprises commonly select the third method, the third method has become widely used as the level of service provided by telecommunication carriers has improved.

Clients in branches of an enterprise also access external servers in a public data center via a public WAN such as the telecommunications carrier's WAN, an Internet service provider's

(ISP's) WAN, and the Internet. In this case, private networks can access the external networks via the gateway router in a private data center or branch, or via that in the core/metro/edge network.

In enterprises, almost all applications performed over the network are IP-based [4]. These applications are classified in a number of ways. From the perspective of reliability (or availability) requirements, mission-critical applications, such as applications handling online transactions and money and ones related to customers' revenues, require guaranteed or predictable reliability, e.g., a required uptime of 99.999 % [1, 5]. Meanwhile, business-critical applications, such as e-mail, customer relationship management (CRM), and collaboration, are essential to business activities, although their reliability requirements are not as strict as those of mission-critical applications. From the capacity requirement perspective, bandwidth-critical applications including voice, video, and teleconferencing need a guaranteed minimum capacity. From the delay requirement perspective, real-time applications have a strict timing relationship between source and destination: one example of this is nonbuffered video playback [5]. In interactive applications, such as Telnet, remote desktop, and web applications, such timing is not so strict and could be defined by the human response time [5].

1.1.2 Integration of Enterprise Networks and Computing Resources

Enterprise networks and applications performed over them have been changing according to the enterprises' strategies for reducing their total cost of ownership (TCO). As a WAN provides high-bandwidth connectivity (i.e., more than several megabits per second), this change has proceeded through roughly three phases, as illustrated in Figure 1.2.

The first phase is the integration of multiple private WANs, as well as computing resources (i.e., servers and storage systems) used for applications in private data centers, e.g., when new enterprises are established through the merger of existing companies. Typical examples of this integration are the several mergers of large banking companies after 2000 in Japan [6]. When two enterprises are merged, their private WANs, which connects their

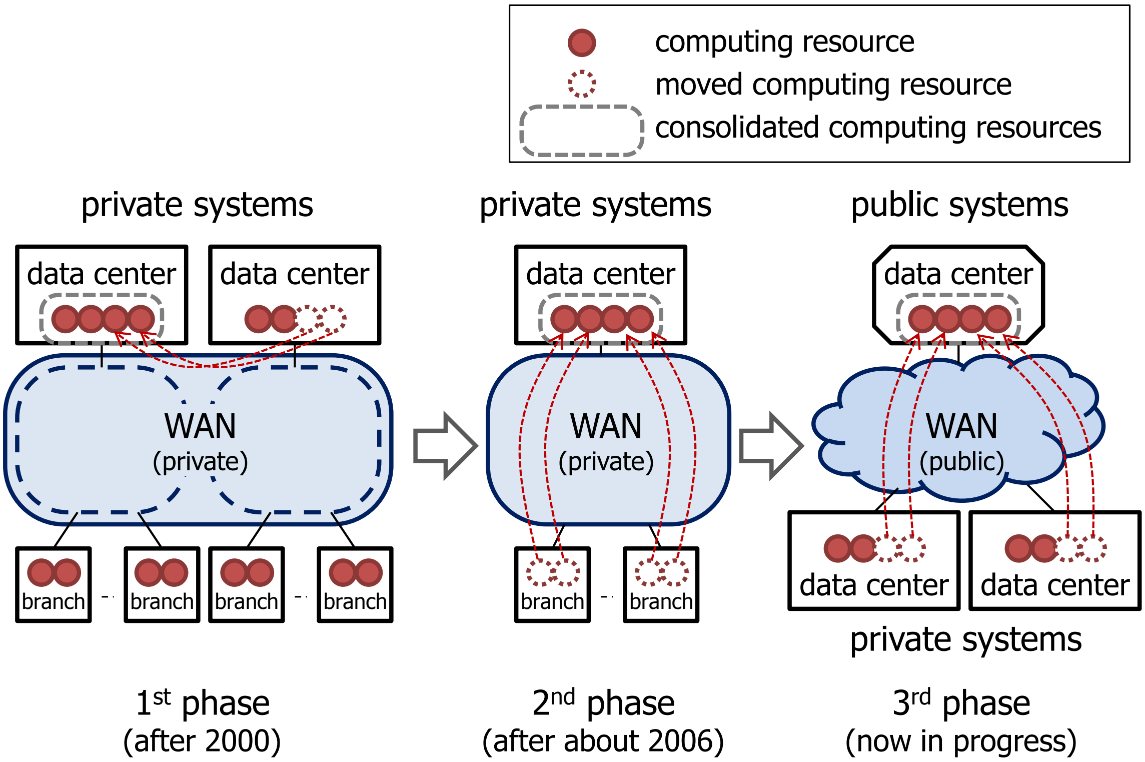


Figure 1.2: Integration of enterprise networks and computing resources

own existing private data centers and branches, are integrated in order to construct a new shared communication infrastructure. After the WANs have been integrated, computing resources for executing applications in each data center are then integrated through the interconnection of data centers or consolidated in a primary data center.

The second phase is the consolidation of computing resources from branches to a private data center to manage computing resources in an integrated fashion, e.g., to simplify backup and restoration procedures, as well as to reduce the risk of corporate information leaking from branches [7, 8]. In this phase, typical consolidated computing resources are desktop computers and shared storage equipment located in branch offices. This phase has become widespread since around 2006.

The third phase is the consolidation (i.e., outsourcing) of computing resources from enterprise private data centers to public data centers, utilizing *cloud computing services* [9]. In order to boost management efficiency and competitiveness, enterprises will outsource applications that deviate from the core business to external service providers. Examples of such applications are business-critical applications deployed in a private data center and monitoring services for checking a branch office's environment. This phase is currently in progress.

1.1.3 WAN Issues and Related Work

In this subsection, we discuss WAN issues that arise during the integration explained in the previous subsection. We also mention related studies and their main subjects.

Issues in the first phase of integration

The first-phase integration is achieved using IP technology. Through this integration, the scale of a private WAN, such as the number of networks nodes and end systems (e.g., servers and clients), and amount of IP traffic, increases. We therefore need to consider the scalability issues of the private WAN. In general, this network consists of three parts: the data plane carrying users' traffic between servers and clients, the control plane carrying route control information between network nodes, and the management plane carrying operations and administration traffic between a network management system and network nodes, as shown in Figure 1.3.

In the data plane, a network node forwards layer-2 or layer-3 packets by using a forwarding table. The number of end systems affects the node's required performance, such as switching capacity and input-output interface bandwidth. The requirements for the nodes also depends on the application traffic routed through them. The characteristics of enterprise traffic have been measured, for example, in respect to application protocol types [10], multicast traffic [11], and traffic redundancy [12].

In the control plane, each node runs control protocols such as routing protocols, exchanges network topology and reachability information with adjacent nodes, and build the

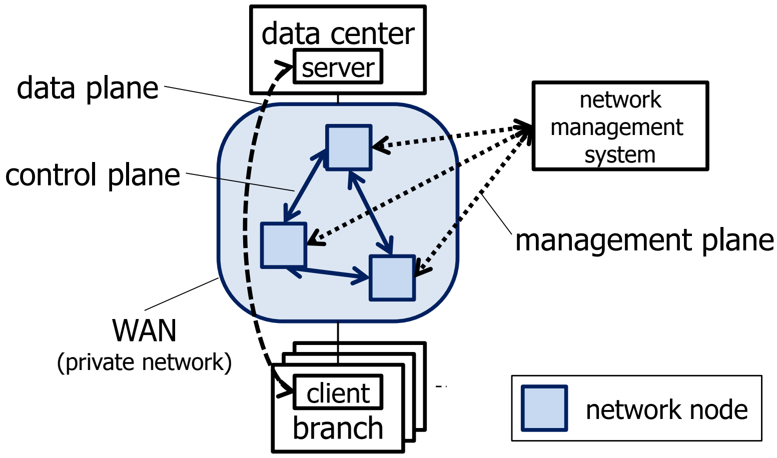


Figure 1.3: Data, control, and management planes in enterprise network

forwarding table by using the exchanged information. The numbers of network nodes and end systems have an large impact on the amount of the exchanged information and the sizes of the tables in the nodes. In comparison to IP routing, Ethernet bridging does not scale well because it relies on broadcast, which could overload the control plane. Therefore, there are many proposals for improving scalability of Ethernet in the case of a local area network (LAN) [13, 14, 15], especially in the case of a data center network [16, 17, 18, 19, 20], and in the case of a WAN [21, 22, 23, 24]. In addition, the optimum design of layer-2 networks has been considered, focusing on virtual LANs (VLANs) [25, 26]. Meanwhile, in general, IP routing can scale to support enterprise levels of usage, as long as it is well designed and configured [27, 28, 29, 30, 31].

In the management plane, the network operator configures the network nodes by using the network management system. The network management system also gathers management information from nodes, e.g., by using the Simple Network Management Protocol (SNMP). The scale of network nodes influences the amount of operation traffic between the management system and nodes. It also affects the time required to complete the operation. From the perspective of the management plane, the configuration management required for large and complex networks, such as modeling network elements and maintaining error-free configurations of the networks, is the main subject [32, 33, 34, 35, 36].

In our case, adjustment of the control plane scalability in an IP network was the most complicated among these three planes, because the network had to take over the IP address settings of the former networks, i.e., a newly merged enterprise's network had to retain the existing IP address settings of pre-merger networks. The enterprise owned and operated the routers in the WAN infrastructure of its private network. Then, network administrators of the enterprise had the responsibility for this adjustment.

Issues in the second and third phases of integration

The consolidation in the second and third phases changes the end-to-end communication path between a client and a server. After the second phase, the end-to-end path is no longer limited to the inside of a branch but traverses the private WAN of an enterprise. Moreover, the third phase results in the end-to-end path traversing a public WAN connecting the enterprise to external data centers. Consequently, application performance depends on the private and/or public WAN quality and application traffic has an effect on the network nodes in the WAN.

As shown in Figure 1.4, from the perspective of an application carried out over the private and/or public WAN, its performance is dependent on the WAN's bandwidth, latency, packet loss ratio, etc. The WAN could be the bottleneck in the end-to-end path because the performance of the WAN is usually lower than that of a local area network (LAN). In our case, the WAN's bandwidth is about one-tenth of that of the LAN in a data center. Moreover, the latency of the WAN, such as one between Tokyo and Osaka, is about 10 to 20 milliseconds, whereas that of the data center LAN is usually from several microseconds to less than a millisecond.

More than 95 % of the packets observed in enterprise networks is based on IP protocol; moreover, 66 % to 95 % of the total IP packets uses Transmission Control Protocol (TCP) as transport protocol [10]. To use TCP efficiently in high-speed long-latency networks, a number of enhancements to the TCP's congestion control mechanism have been developed [37]. These enhancements include loss-based approaches such as High-Speed

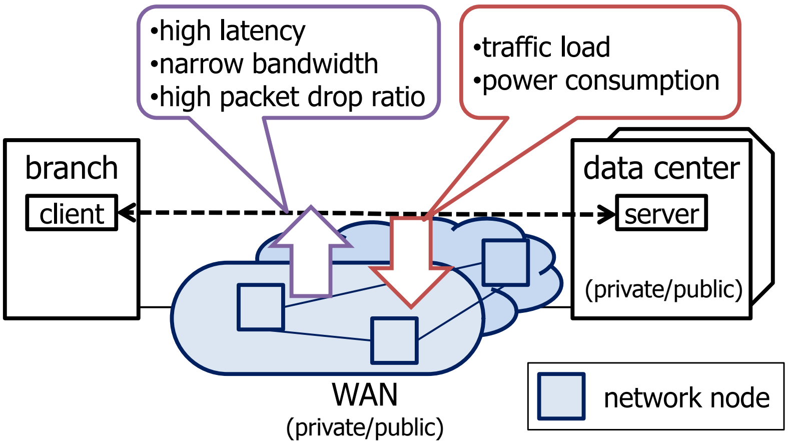


Figure 1.4: Interaction between application's end-to-end connection and WAN nodes

TCP [38] and its modification [39], Scalable TCP [40], delay-based approaches such as TCP Vegas [41] and FAST TCP [42], hybrid approaches such as Compound TCP [43] and TCP-Adaptive Reno [44], and other approaches including BIC-TCP [45], CUBIC-TCP [46], TCP Westwood [47]. Above enhancements are suitable for bulk data transfer (e.g., File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), HyperText Transfer Protocol (HTTP)). However, improvements of interactive data transfer (e.g., Telnet, rlogin, and Secure Shell (SSH) (, and HTTP in some cases)) in such networks are side effects of those enhancements. Although improvements of fairness between TCP long-lived and short-lived connections [48] have been discussed [49, 50], interactions between bulk data transfer and interactive data transfer, especially when they coexist in a TCP connection across the WAN, have not been addressed.

On the other hand, from the perspective of the underlying network system, the consolidation of computing resources in data centers results in a greater traffic load on the WAN. Especially in the third phase, applications like business-critical and environmental monitoring ones are outsourced to external service providers. These applications include multimedia data related to digital videos and cameras. Such outsourcing, i.e., using cloud computing services, is becoming widespread among enterprises, public offices, homes and so on. According to [51], IP traffic will grow at a compound annual growth rate (CAGR) of about 30 % from 2010 to 2015 and the biggest driver for the global traffic growth will be video. For example, multimedia traffic accounts for approximately half of the total IP traffic in a large ISP [52]. One of main challenges for such a WAN is reduction in the power consumption of the WAN. Power efficiency in telecommunications networks has been one of the major research issues [53, 54, 55, 56]. Proposed power reduction strategies includes putting idle components in sleep mode at an router/switch level [57, 58] or a port level [59, 60, 61], and modifying link rate [62, 63]. These strategies consider the optimal routing that minimize the total power consumption in the network. However, variety of traffic matrices in the network, e.g. the matrix describing the amount of traffic related to the geographic locality, have not been considered in their scope.

1.2 Outline of Thesis

We have selected several important but not well-discussed issues from those addressed in the previous section, and studied solution approaches for the private and public WANs and applications performed over them. In this thesis, we focus on the following three objectives, which correspond to the three phases of the integration of enterprise networks and computing resources.

- Performance of a large enterprise network for updating routing information
- Performance a thin-client system in a WAN environment
- Power consumption of a WAN with the aid of distributed computing.

The chapters of this thesis are summarized below.

Note that those three objectives deal with the same private/public WAN described in Section 1.1.1. The solution approaches for those objectives do not constrain each other.

1.2.1 Chapter 2: Performance of a Large Enterprise Network for Updating Routing Information [64, 2]

In Chapter 2, we focus on evaluating the scalability of the control plane in a large enterprise network, which is one of the important issues when the network have to take over the IP address settings of the former networks. We develop an approach that allows us to estimate the performance of the network for updating routing information. This approach is applied to the one of the world's largest private networks, which was constructed by integrating two large banking networks in Japan. Note that this enterprise managed the network nodes, i.e., routers and ATM switches, itself, rather than outsourcing them to a telecommunications carrier.

The main characteristic of this approach is based on a formula that represents the delays in updating routing information that accompany reductions in central processing unit (CPU) resources. This procedure consists of two steps: one estimates the reduction in the availability of CPU resources caused by forwarding data packets at a router, and the other estimates the levels of CPU resources required for replying to queries about new routes and subsequently updating the routing information. These steps are applied to estimate the level of CPU utilization required for routing information convergence. The results of our experiments on the network show that updating the routing information is possible as long as the average CPU utilization during any five-minute period at the routers is less than 40 %. We were able to apply this guideline and confirm the stability of the network.

1.2.2 Chapter 3: Performance of a Thin-Client System in a WAN Environment [65, 66, 67, 68]

In Chapter 3, we focus on evaluating the performance of a typical application traversing a private WAN (and/or public WAN) after computing resources have been consolidated from branches to a private data center. We thus concentrate on a thin-client system. The performance of thin-client systems based on TCP depends on network quality, so it becomes worse in a private WAN environment. However, the network traffic of the response from the server in thin-client systems is a mixture of interactive and bulk data flows in a TCP connection, and the effects of TCP mechanisms in such a situation have not been clarified.

In this chapter, we first describe the download traffic of thin-client systems as a two-state model with interactive data flows in response to keystrokes and bulk data flows related to screen updates. Since users are more sensitive to the keystroke response time, our next objective is to minimize the latency of interactive data flows, especially when the network is congested. Through detailed simulation experiments, we reveal that the main delays are queuing delay in the bottleneck router connected to the WAN and buffering delay in the server. We then enhance two TCP mechanisms: retransmission timeout calculation and selective acknowledgment (SACK) control, which overcome the drawbacks of existing options and increase the interval between occurrences of large delays (more than about 1 second) by about four times (up to about 2,500 seconds).

1.2.3 Chapter 4: Power Consumption of a WAN with the Aid of Distributed Computing [69, 70, 71]

In Chapter 4, we focus on evaluating the effect on a public WAN as a result of consolidating computing resources in a few huge public data centers. Such consolidation is becoming widespread not only among enterprises but also among homes and public offices. Therefore, we did not restricted this topic to enterprises but considered it for the whole society. When computing resources are consolidated in a few public data centers, this results in increased power consumption in a public WAN. A distributed computing network (DCN), such as a content delivery network, not only improves the response time to clients but also reduces the traffic to and from the data center over the WAN, thereby decreasing the power consumed in the WAN. We concentrate on the energy-saving aspect of the DCN and evaluate its effectiveness, especially considering traffic locality, i.e., the amount of traffic related to the geographical vicinity.

In this chapter, we first formulate the problem of optimizing the DCN power consumption and describe the DCN in detail. Numerical evaluations show that, when there is strong traffic locality and the router has ideal energy proportionality, the system's power consumption is reduced to about 50 % of the power consumed in the case where a DCN is not used. Moreover, this advantage becomes even larger (up to about 30 %) when the data center is located farthest from the center of the network topology.

1.2.4 Chapter 5: conclusion

Finally, in Chapter 5, we present the conclusions of this thesis and discuss future work. In particular, we discuss work for the situation after the third phase of the integration process.

Chapter 2

Performance of a Large Enterprise Network for Updating Routing Information

2.1 Introduction

The computer network of a certain large bank, which we call A Bank, was constructed by integrating the networks of two certain banks, which we call B Bank and C Bank, in preparation for the merging of the two enterprises to form the new bank on January 15, 2002. The A Bank network connects about 5,000 routers in about 2,100 buildings. The network is one of the largest enterprise networks in the world. It is also, of course, a mission-critical network because it is used for communicating banking data.

To build a new network by integrating the two existing networks, it was necessary to estimate network scalability to verify the reliability of the newly integrated network. However, a number of problems appeared during integration of the network in 2001. The most serious of these problems was that the routing information did not converge when the network topology was changed. This fault caused the routers to consume so much CPU resources that data communications were interrupted. Such faults arose because the scalability limit of the *B* Bank network was exceeded when it was merged with that of *C* Bank. Accordingly, we took appropriate steps to correct these faults and thus improved the quality of the network [72, 4].

We began by investigating the conditions under which the network was capable of updating the routing information with sufficient rapidity. We estimated the time taken for a router to reply to a request for a new route when the router was busy forwarding data packets, and we set up guidelines to determine whether the level of CPU utilization by a router was low enough for completing convergence [64]. The investigation's purpose was to confirm the scalability of the network and to obtain guidelines for extending the network.

This chapter describes our experimental approach to estimating the performance of large-scale enterprise networks in terms of updating routing information. First, we explain the features of the *A* Bank network and the faults that occurred during the integration process. We then explain the steps involved in estimating the time taken by a router to reply to a query received during the updating of routing information. Next, we give guidelines for determining whether a router's level of CPU utilization is low enough for it to update routing information with sufficient rapidity. Finally, we discuss how we verified these guidelines through measurements on the network and then applied them to confirm the reliability of the integrated network.

2.2 The *A* Bank Network System

2.2.1 The *A* Bank's network architecture

Network scale

The *A* Bank network connects about 2,100 buildings and has about 5,000 routers that were made by Cisco Systems Inc. The network was constructed by merging the *C* Bank network, which connected about 900 buildings, into the *B* Bank network, which connected about 1,200 buildings. The *B* Bank network was used as the prototype of the *A* Bank network.

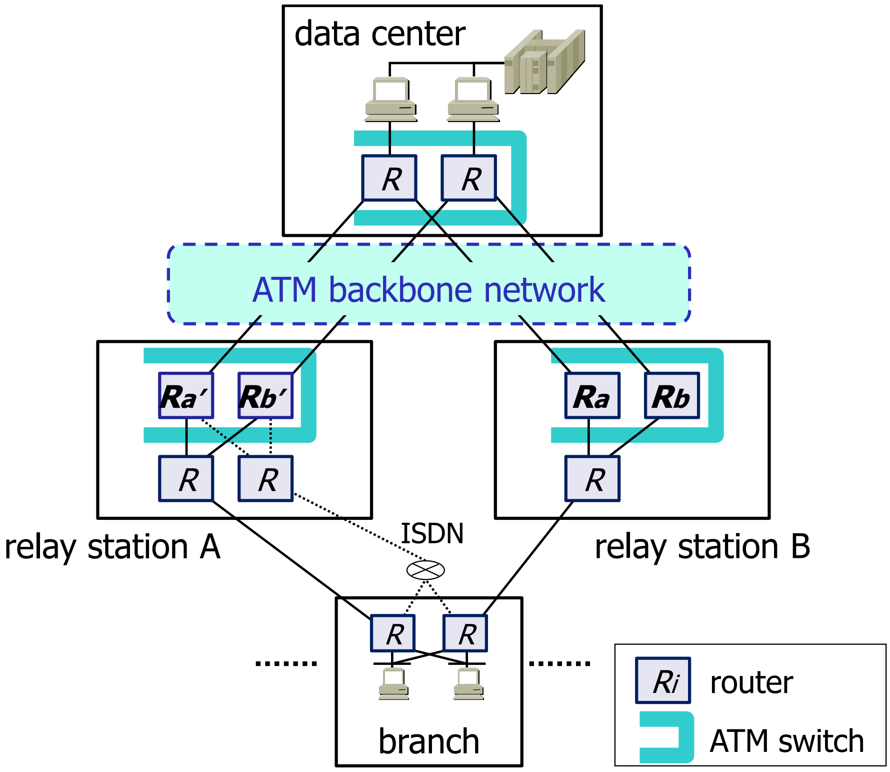


Figure 2.1: Logical framework of the *A* Bank network

Network topology

Figure 2.1 shows the logical framework of the network. The network consists of three parts: data centers, relay stations, and branches. Each relay station is a network hub and accommodates the lines from branches and from data centers. The relay-station layer is a critical part of the network because a fault in it causes the greatest topological changes in the network.

Traffic paths

Double paths are provided for each link between a branch and a data center to create a redundant fail-safe routing topology. Half of all paths bound for the main center are concentrated on the relay station A and the other half are concentrated on the relay station

B. The routers Ra and Ra', shown in Figure 2.1, have more neighboring routers than any others in the network (about 50 neighbors), and the routers Rb and Rb' have the second greatest numbers of neighbors. A fault in any single router triggers updating of the routing information held by most of the routers in the network.

Routing protocol

We use Enhanced IGRP (Interior Gateway Routing Protocol), EIGRP, which is a dynamic routing protocol from Cisco Systems Inc. Figure 2.2 explains the mechanism applied by EIGRP in updating routing information [73, 74]. When the link between the routers Ra' and Rc fails, the router Rc loses the routing information that was advertised by the router Ra'. If the router Rc does not have the information for an alternative route, it sends a query packet to all of its neighbors to inquire about the lost routing information. The router Ra, upon receiving a query from the router Rc, attempts to find a new path to the given network address. It finds one, so it sends a reply packet to the router Rc. The router Rc, upon receiving replies from its neighbors, updates its routing information and selects a certain router as the next hop for an alternative path. If the router Rc doesn't receive a reply from a neighbor within 3 minutes, it stops waiting for a reply. This timer value is based on the router's setting. Then, the router Rc clears its connection to the neighboring router that is not answering and restarts the session with the neighbor. This is called a stuck-in-active (SIA) route. This situation is called an SIA error.

2.2.2 Faults that appeared during integration

The faults that appeared during the integration of the network in 2001 were of two types. Both prevented the quick convergence of EIGRP in response to changes in the network topology. Convergence took several minutes, slowly appearing in successive parts of the network, after a fault had occurred. The faults brought about extremely high levels of CPU utilization at the routers, and this stopped some communication between the centers and the branches. As previously mentioned, the A Bank network was being constructed by

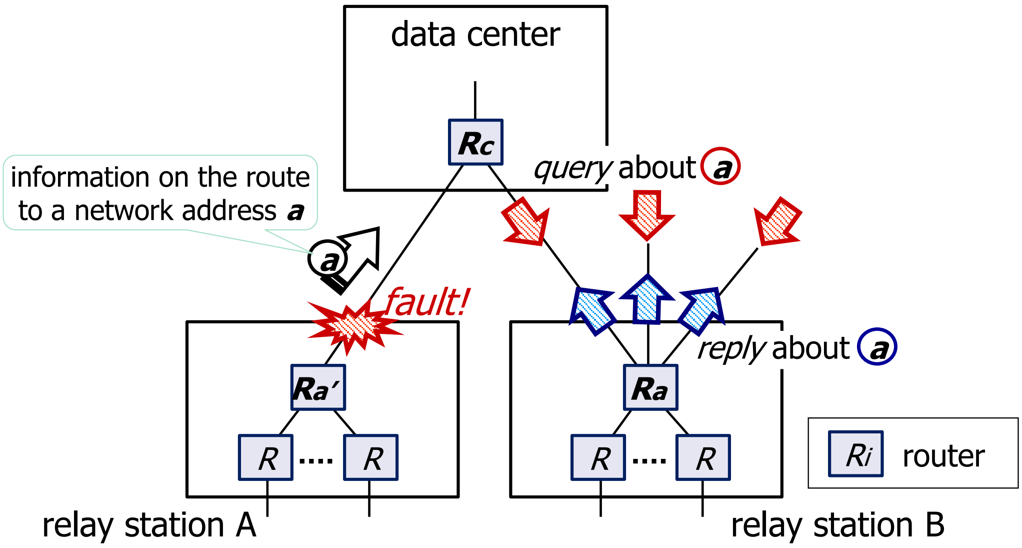


Figure 2.2: Updating of routing information by EIGRP

integrating the *C* Bank network into the *B* Bank network. The problems arose because the limits of scalability for the *B* Bank network were exceeded during this process. We settled this issue in the following ways.

Routers with overly complicated configuration

One problem arose because more than five paths with the same cost existed between two routers, and another problem arose because about twenty routers advertised the same default route to a single router. Both were attributed to the large scale of the network. We solved the problems by changing the routers' configurations rather than by installing a later version of the routers' operating system. We chose this path because the latter action would have had a strong impact on the network and we did not have enough time to test a later version of the software.

Routers with insufficient ability

If the router *Ra* in Figure 2.2, for example, is overloaded by receiving too many queries, and has insufficient capacity for processing them all, it becomes incapable of quickly responding to queries. If the delay in replying to a query exceeds 180 s, an SIA error occurs. This fault stops data communication. In order to prevent the triggering of an SIA error by a router's inability to reply, we have to reduce the load on the router so that the router can reply in time. We took the following actions to implement this solution.

- We reduced the numbers of advertised network addresses by filtering the routing packets (i.e., reduced the numbers of the queries sent).
- We reduced the numbers of neighboring routers by placing new routers in intermediate positions to act as neighbors (i.e., reduced the numbers of received queries).
- We improved router capacity by replacing some of them with higher-grade routers.

2.3 Scalability of Routing Protocol

We considered the scalability of the routing protocol and otherwise examined the EIGRP, (the protocol used in the network), in the following ways [75, 1, 76].

Protocol specification

Setting up the routers in a more complicated configuration than Cisco Systems Inc. had anticipated led to faults of the kinds explained in Section 2.2.2. Needless to say, we had to check the relevant technical reports and to use a test network in order to examine the configurations before we altered the existing networks. In particular, it was important that we expand the set of items examined to reflect the level of faults that we experienced.

Normal operation

To maintain the routing connection with its neighbors, each router sends hello packets to each of its neighbors every 5 s. The packet size for this is about 60 bytes, including the IP header, so this takes up very little bandwidth. If a hold timer expires because a hello packet has been lost, the neighbor is declared unreachable and its entry in the router's internal information is discarded. Therefore, the highest priority should be given to securing the bandwidth required by hello packets.

Operation in response to topological changes

EIGRP is said to be more suitable for large networks than other dynamic routing protocols such as RIP (Routing Information Protocol). This is because updates in EIPRP are non-periodic, may be partial, converge quickly, and take up less bandwidth. If a router has no appropriate alternate route when the cost of one route changes, the router, operating with EIGRP, queries its neighbors to discover an alternate route. This query is propagated until an appropriate route is found. When the timer expires before the router receives a reply from its neighbor, the router is presumed to be in a state of SIA error. As mentioned in Section 2.2.2, most of the faults that occurred during the integration of the network in 2001 were caused by SIA errors due to insufficient router capacity.

Address summarization may be applied to reduce the numbers of EIGRP-learned entries that are stored in the route tables. Address summarization is quite an effective way of reducing the total utilization of processors for updating routing information. We were, however, unable to configure the routers of the *A* Bank network to use summarized addresses because the network had to continue using the IP addressing scheme used previously in the *B* Bank network. As a result, the size of the routing table in each router increased in proportion to the increase in the number of network addresses. The greater the increase in the number of network addresses, the more the core routers were flooded with query packets from their neighbors so that they could not process them quickly enough. This is why we had to estimate the scalability of the network with a particular focus on the scalability of routing protocol operation. First, we reduced the load on the routers as explained in Section 2.2.2. After that, we estimated the scalability of the network. We will explain the estimation method in detail in the following section.

2.4 Performance Estimation in Advance of Full Integration

2.4.1 Overview

In order to verify the reliability of the integrated network, we took the following steps to estimate the scalability of EIGRP in the network.

First, we estimated the time the most critical router would take to respond to a query and to update its routing information in response to a change in topology during its reception of data packets. We thus carried out these experiments on the router *Ra* of the relay station B, to investigate its ability to respond to queries.

Second, we estimated the levels of CPU resources that were sufficient for a router to reply to queries while loaded with data packets. We derived a formula for the response time to a query for a router receiving a lot of packets and therefore using high levels of CPU resources for packet processing.

We intended to apply the approaches mentioned above to make measurements in the actual network environment. Of course, we conducted tests on a simulated network first. The *A* Bank network is, however, so large that it was not possible for us to quantitatively analyze the whole process of routing convergence in a practically useful way. For this reason, we decided to carry out experiments directly on the actual network.

Note that this chapter deal with a large banking network as a concrete example of a large enterprise network. Therefore, our proposed approaches are not limited to banking networks; these approaches are applied to common large enterprise networks.

2.4.2 Estimating response times for queries and the CPU resources while forwarding data packets

We observed that the replies of routers to queries were delayed when the routers were receiving many data packets for forwarding. We assumed that this delay was caused by the corresponding reduced availability of CPU resources. Based on this assumption, we followed the procedure described below to obtain a formula for the appropriate response time for a query of a router under a heavy packet load. From our observations, we derived a simple model that was appropriate for a large enterprise network operation.

Estimating CPU utilization

The more packets a router receives, the greater the level of CPU resources required for the forwarding of packets. When we measured the result for a router in our test system, we found that the level of CPU utilization was greatly increased by relatively small increases in a load with lower rates of packet reception at the router, but that the increases became smaller for a given load with higher rates of packet reception. Thus, the best-fitted function for the desired result should be selected. To find that equation, we approximated the relationship between the level of CPU utilization y (%) and the rate of packet reception x (pps (packets per second)) at a router by

$$y = \frac{100\,x}{x + c} \quad , \tag{2.1}$$

where c is a constant and equals the number of packets per second when the level of CPU utilization incurred in forwarding them is 50 %. Equation (2.1), which was derived from actual practice, not from theory, expressed the desired result for our test system well.

Estimating response times for queries

We used the following expression for the utilization of CPU resources in a router while the router is forwarding data packets:

$(CPU\ resources\ for\ replying\ to\ queries) = 1$ - $\{$ $(CPU\ resources\ for\ forwarding\ data\ packets)$ $+$ $(CPU\ resources\ expended\ as\ an\ overhead\ for\ processing\ queries$ $(a\ supplementary\ term))$ $\}$.

The second term in right-hand side of the formula was assumed to be expressed in the same way as Equation (2.1). We approximated the sum of the CPU resources taken up in the forwarding of data packets and the CPU resources expended as an overhead on the processing of queries by using the same formula as Equation (2.1). Therefore, the sum is represented by y' (%), which is defined by

$$y' = \frac{100\,x}{x + c'} \quad , \tag{2.2}$$

where x (pps) is the number of packets received per second and c' is a constant. The level of CPU resources y_1 (%) assigned to replying to queries can be expressed by subtracting the sum y' from the whole, which is defined by

$$\frac{y_1}{100} = 1 - \frac{y'}{100} = \frac{1}{1 + x/c'} \quad . \tag{2.3}$$

The response time for a query is inversely proportional to the amount of CPU resources given for processing queries. The relationship between the level of CPU resources y_1 (%) to reply to queries, the response time t_0 (seconds) for a query when the router is forwarding no data packets, and the response time t (seconds) for a query when the router is forwarding data packets is given by

$$t = \frac{100\,t_0}{y_1} \quad , \tag{2.4}$$

By substituting Equation (2.4) into Equation (2.3), the response time t (seconds) for a query when the router is forwarding data packets is defined by

$$t = t_0 \left(1 + \frac{x}{c'} \right) \quad , \tag{2.5}$$

where t_0 (seconds) is the response time for a query when the router is forwarding no data packets, x (pps) is the rate of packet reception, and c' is a constant. The response time could be approximated to a linear function of the packet-reception rate by using Equation (2.5) so that it became easier to handle.

2.4.3 Experiments on the *A* Bank network

We investigated the performance of the router *Ra*, the most critical router of the network. The results for this router could be taken as representative of the overall performance of the network.

Measuring CPU utilization by data-packet processing

We measured the CPU utilization for the router *Ra* at the following times:

- The *B* Bank network at the end of the year 2001
- The *A* Bank network during the rehearsal for integration at the beginning of the year 2002
- We also evaluated the same model as that of the router *Ra* on the test network under loading by a traffic generator

We measured the 5-minute average levels of CPU utilization by getting the data contained in the Management Information Base (MIB) from the router every 10 minutes [77].

Measuring response times for queries

We carried out experiments with the network before full integration of the *A* Bank. At that time, the network had been integrated and the routers had exchanged routing information, but business data traffic was not communicated, as the bank itself was not yet integrated. We explain our experimental method with the aid of Figures 2.1 and 2.2.

Step 1: A traffic generator placed a traffic load on the routers *Ra* and *Ra′*. We measured the packets received by the routers by getting the MIB every 10 minutes. We set a traffic-capturing tool on the router *Ra*.

Step 2: We initiated changes in the network topology by shutting off the power supply of the ATM switch in relay station A.

Step 3: Bringing this link down caused query and reply exchanges between the routers. The router *Ra* received queries for about 6,000 destination addresses from about 40 neighboring routers at the same time.

Step 4: The router *Ra* directly replied to queries because it had the requested routing information.

Step 5: We analyzed the EIGRP packets captured by the tool. We investigated the router *Ra*'s response time for queries by calculating the difference between the times at which the query and the corresponding reply passed the traffic-capturing tool.

To investigate the router *Ra*'s maximum response time for queries at various levels of traffic load, we repeated Steps 1 to 5 while varying the load of traffic on the routers *Ra* and Ra'.

2.4.4 Experimental results and drawing up the guidelines

Response times for queries

Figure 2.3 shows the experimental results. It shows the time the router *Ra* took to respond to a query, obtained in the way outlined in Section 2.4.3. We took the maximum time in each experiment as the response time. The response time of the router *Ra* was regarded as being equivalent to the time taken for the querying neighbor's routing information to be updated, because the neighbor was not overloaded and the delay on the link between the two routers was negligible. When we applied Equation (2.5) to approximate a straight line, t_0 was 70 s, and c' was 21,500 (line (a), minimum estimate) or 7,200 (line (b), maximum estimate).

We then determined the guidelines. The numbers of queries sent to router *Ra* varied according to the point where the link went down. The response time for a query when there was no other traffic, t_0, thus changed. We had to consider the worst case, which we defined as plural faults happening at the same time. Multiple equipment items might break down at the same time, so we supposed that the experimental result for t_0 was 80 %

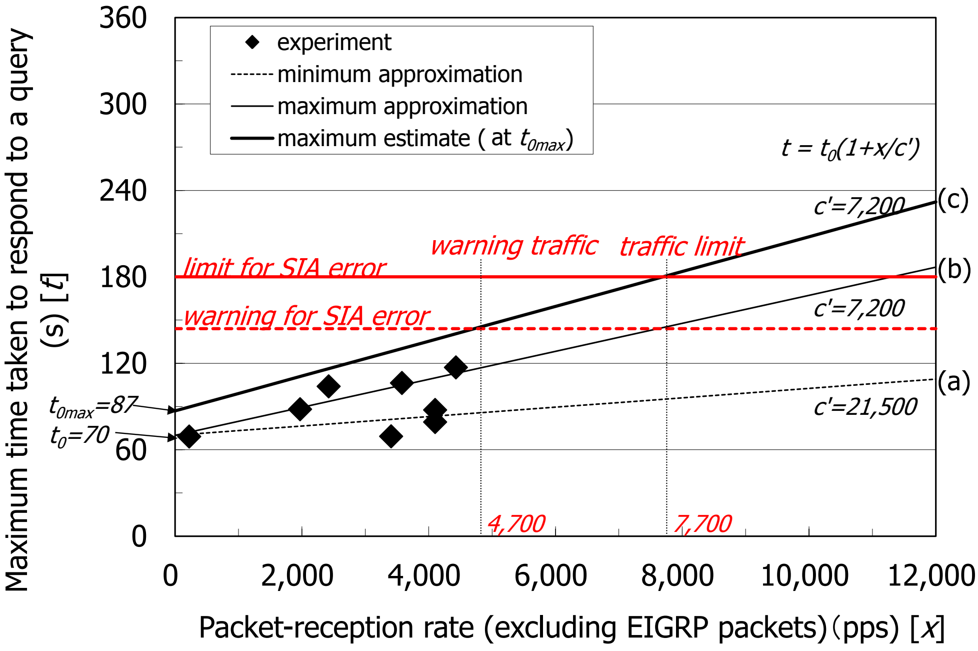


Figure 2.3: The time router Ra takes to respond to a query

of the maximum value. The maximum of t_0, i.e., t_{0max}, was then 87 s. We got the line (c) in Figure 2.3 by substituting t_{0max} and the upper estimate for c' in Equation (2.5). The line (c) in Figure 2.3 indicates that the maximum time taken to respond to a query could be assumed to be 87 seconds (t_{0max}) when the router was forwarding no data packets and the multiple faults occurred. This response time was delayed when the router received more data packets. We decided on the limit of 180 seconds as the time taken to update the routing information, which is the value for the SIA-error timer. We assumed a warning value of 144 s, that is, 80 % of the limit. The line (c) in Figure 2.3 shows that the respective rates of packet reception at the router were 4,700 pps for the warning value and 7,700 pps for the limit.

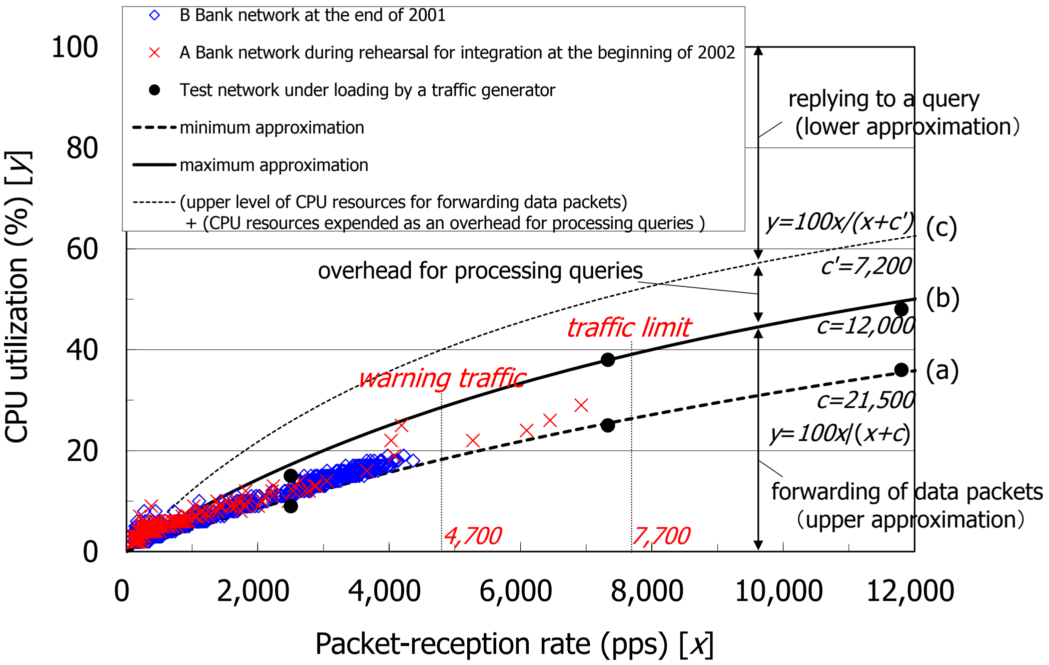


Figure 2.4: CPU utilization of the router *Ra*

Estimating sufficient levels of CPU resources for a reply

Figure 2.4 shows the experimental results for the router Ra, as obtained under the conditions outlined in Section 2.4.3. The levels of CPU utilization was relatively high, because we were unable to use Cisco Express Forwarding (CEF), which uses a route caching mechanism and makes the packet forwarding faster at the router [78]. The constant c in Equation (2.1) was 21,500 in the minimum approximation (a) and 12,000 in the maximum approximation (b). The value of c' as calculated from the response times (Figure 2.3) was from 21,500 to 7,200. The difference between c' and c at the upper limit was assumed to be because of the query processing overheads.

Figure 2.4 shows that CPU resources used for forwarding of data packets could increase to more than 40 % when the router became busy forwarding critical traffic (7,700 pps and over), and CPU resources available for replying to queries would be somewhat less than about 50 %, considerating that an overhead for processing queries was about 10 % or more.

If the plural faults occurred when the router was receiving this critical traffic, the router did not have sufficient CPU resources to process queries so that it could not reply to a query within the value of the SIA-error timer (180 seconds), as is shown in Figure 2.3. The use of CPU resources for forwarding data packets must be restrained until the completion of any process for updating routing information. We used line (b) of Figure 2.4 to determine the warning value for CPU utilization at the router *Ra* as 30 % when the router is receiving 4,700 pps and the critical value as 40 % for 7,700 pps.

Summary of the guidelines

We summarized the experimental results and determined summarized guidelines for network reliability:

- Even when a router has a large load of traffic and multiple faults occurr, it is able to reply to a query within the warning time for SIA errors (144 seconds) as long as the 5-minute average levels of CPU utilization is below 30 %, which corresponds to a packet-reception rate of 4,700 pps;
- The router is able to reply to a query within the limit for SIA errors (180 seconds) as long as the level of CPU utilization is below 40 %, which corresponds to a packet-reception rate of 7,700 pps.

These guidelines were designed to deal with the worst case faults scenario, which would be plural equipment breakdowns occurring at the same time. When an ordinary fault, such as the single fault observed in the experiments described in Section 2.4.3 occurred, the warning value for CPU utilization was 40 %, which corresponded to a packet-reception rate of 7,600 pps and critical value was 50 %, which corresponded a rate of 11,300 pps. These were calculated by using line (b) in Figure 2.3 and line (b) in Figure 2.4.

The guidelines were deduced from the router *Ra*'s behavior, i.e., the behavior of the most critical router of the network, so the guidelines were applicable to the network as a whole. If the guidelines were strictly considered, the critical level of CPU utilization would be different in individual routers because each router had different environment, such as the number of neighboring routers. However, we could not carry out experiments and make the guidelines for the individual routers. Moreover, we considered that these simple guidelines were also suitable for the complex network operation. The values of CPU utilization in these guidelines used as indexes to investigate details of the network's condition. In addition, if the level of CPU utilization for the router rose above these guideline values when a large load of data traffic occurred, the network system, as a matter of course, could keep handling the traffic unless a really large fault occurred.

2.5 Performance Verification after Full Integration

To verify the stability of the network after integration of the *A* Bank, we confirmed that the levels of CPU utilization for the routers *Ra* and *Rb* were below the guideline values. We investigated the levels of CPU utilization and the reception of packets via the physical interface by getting the MIB data from the routers. Figure 2.5 shows the change in the router *Ra*'s 5-minute average level of CPU utilization over January 17, 2002, and Figure 2.6 shows the equivalent figure for the router *Rb*. The thick solid lines show the levels of CPU utilization. The thin solid line shows the upper estimate, and the thin dotted line the lower estimate, as obtained by using Equation (2.1) and the rate of received packets.

Verifying the network's stability: comparison with the threshold

We concluded that the router *Ra* was stable because the peak level of CPU utilization was below the warning value (30 %). The peak level of CPU utilization for the router *Rb* rose above the warning threshold (30 %) but remained below the limit (40 %). We were therefore continuing to watch this router to determine whether this was a passing phase or a continuing phenomenon. If the peak frequently continued to exceed the warning level, we would need to take the measures listed below:

- persuade the team that is operating the applications to decrease the volume of the traffic or disperse it, and

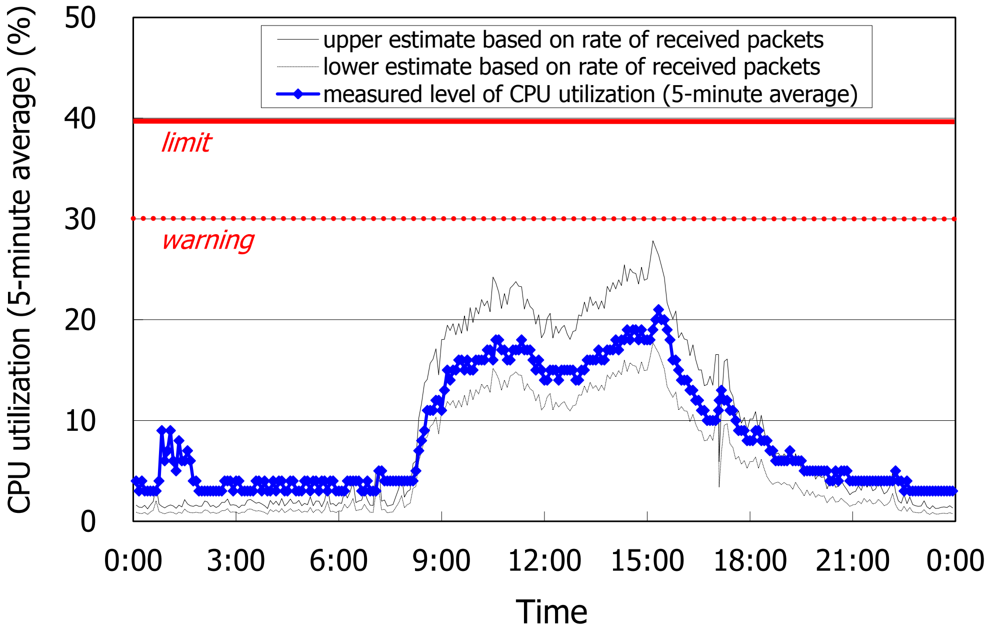


Figure 2.5: The router *Ra*'s levels of CPU utilization after integration (Jan. 17, 2002)

- exchange the CPU module or configuration of the router to improve the router's performance.

Discussion of conditions: a comparison of actual results with estimates calculated by using packet-reception rates

A level of CPU utilization that corresponds to the estimate produced by Equation (2.1) means that no CPU resources were used other than for forwarding packets. For the router *Ra*, the actual and estimated levels of CPU utilization corresponded except at levels below 10 %. For the router *Rb*, however, the peak value went beyond the upper estimate and the values were close to the upper estimate across most of the range. The routers *Ra* and *Rb* were of the same type and were set up with the same software configuration. When the router *Rb* was receiving the same number of packets as the router *Ra*, it should have shown the same level of CPU utilization as the router *Ra*. For given rates of packet reception,

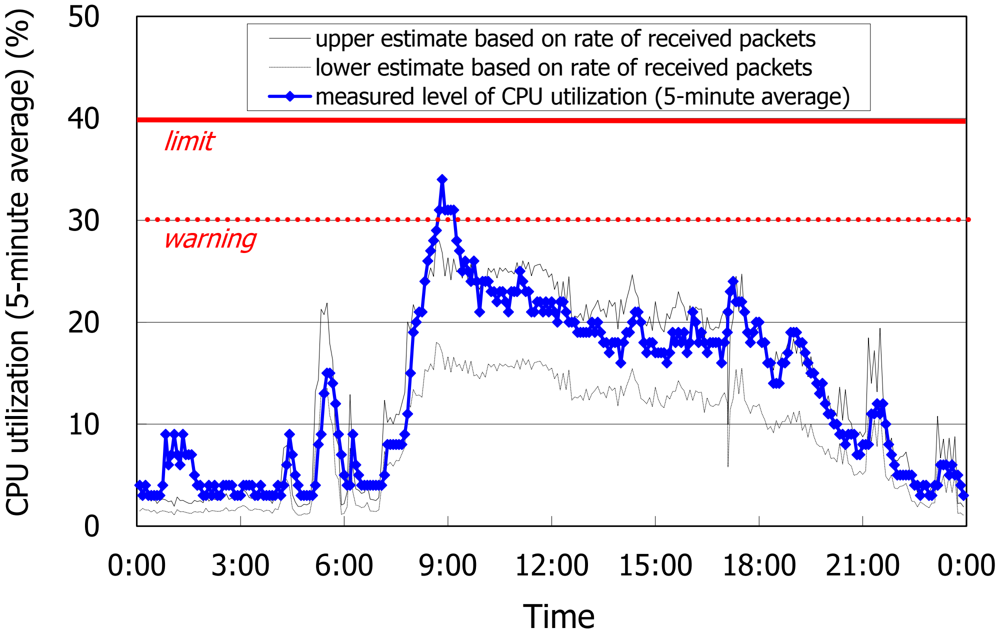


Figure 2.6: The router *Rb*'s levels of CPU utilization after integration (Jan. 17, 2002)

however, the router *Rb* actually showed higher levels of CPU utilization than did the router *Ra*. We investigated the difference between the two routers. Our conclusion was that the difference arose because of differences in the proportion of packets forwarded by process switching relative to packets switched by using a route cache.

Figure 2.7 is the result of performance analysis for the router *Rb*. The lower solid line shows the change in difference of CPU utilization between measured and median-estimated values. The upper dotted line shows the change in reception rate of the data packets that could not be switched by means of route caching. The highest rate for the sending of packets by process switching (about 180 pps) was at the peak seen at 9:00 AM in Figure 2.6. The difference between the estimated and measured values was proportional to the number of packets forwarded by process switching. The ratio of packets going from branch to branch was high at the router *Rb*, while almost all the packets being processed by the router *Ra* were those going from a center to a branch. For a given number of packets, more pairs of

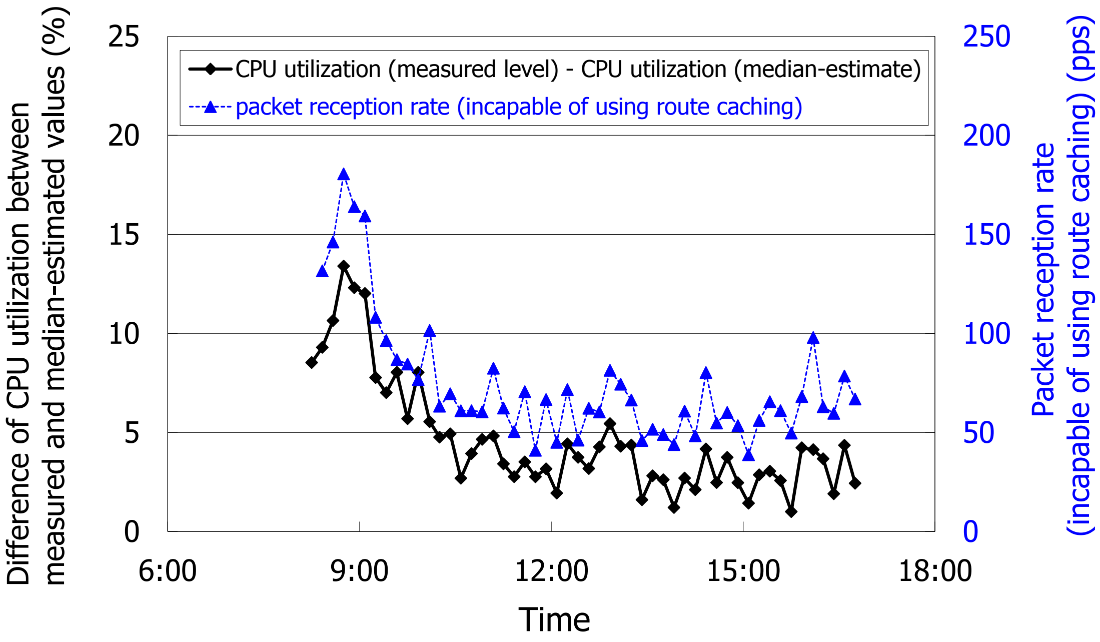


Figure 2.7: Performance analysis for the router *Rb* (Jan. 17, 2002)

source and destination addresses were switched by the router *Rb* than by the router *Ra*. We concluded that the router *Rb* had become incapable of using route caching to switch some of the packets and that this had raised its level of CPU utilization. We, thus, confirmed that no CPU processes other than IP forwarding was being executed, and that the router *Rb* was operating normally.

2.6 Conclusion

In order to verify the reliability and to improve the quality of the *A* Bank network, which was fully integrated in January 15, 2002, we investigated the scalability of this large enterprise network. We estimated the network's performance in updating routing information based on our experience of faults that were observed during the process of network integration.

Our approach to estimating network performance was to concentrate on the scalability of convergence. We thus drew up a formula to represent the effect of the levels of packet traffic on the level of CPU utilization, and hence on delays in the updating of routing information. The approach consisted of two steps. One was the evaluation of the CPU resources by forwarding data packets at a router. The other was estimation of the levels of available CPU resources needed to reply to queries regarding new routes.

We experimented on the actual network and applied the steps above to estimate the performance of the integrated network. We determined two guidelines for the worst case scenario from the experimental results:

- When the 5-minute average level of CPU utilization at a router is below 30 %, the routing information is updated within the warning time for SIA errors (144 seconds);
- When the 5-minute average level of CPU utilization at a router is below 40 %, the routing information is updated within the time limit for SIA errors (180 seconds).

After completing the process of integration, we confirmed that these guidelines were being maintained within the *A* Bank network. We thus ensured the stability of the network. An important aspect of this chapter's focus is the technical know-how of system engineers, so we will systematize and make available the know-how we have acquired [79].

Chapter 3

Performance of a Thin-Client System in a WAN Environment

3.1 Introduction

A thin-client system is a server-centric computing system in which a client terminal transmits user inputs (keystrokes, mouse clicks, etc.) to a remote server, and the server returns the corresponding screen updates to the desktop application interface on the terminal (Figure 3.1). This system enables an enterprise's IT department to manage its client computing resources in an integrated fashion and to promote flexibility in the workplace without leaking corporate information from client terminals [80].

The thin-client system is generally implemented by using a remote desktop protocol such as X11 [81], Virtual Network Computing (VNC®[1]) [82], or Microsoft®[2] Remote Desktop Protocol (RDP) [83]. From a transport-layer perspective, each remote desktop protocol is based on a persistent Transmission Control Protocol (TCP) connection. The system performance is thus affected by the TCP configuration. For example, the use of TCP buffering mechanisms—the Nagle algorithm [84] and delayed acknowledgment [85]—could lead to delayed delivery of small packets in an interactive application [86, 87]. Disabling

[1] VNC is a registered trademark of RealVNC Ltd.

[2] Microsoft is a registered trademark of Microsoft Corporation.

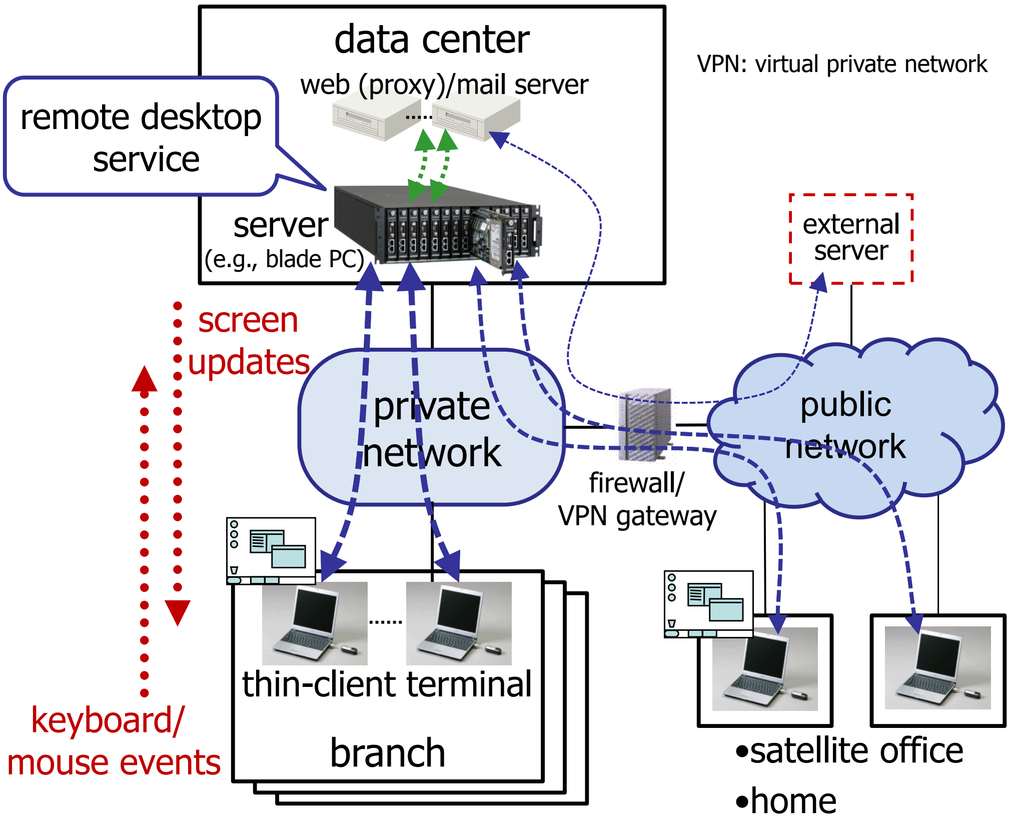


Figure 3.1: Overview of thin-client system

these buffering mechanisms is therefore appropriate for an application like X11 [88]. Furthermore, TCP's slow-start restart [89], i.e., reinitialization of the TCP congestion window (*cwnd*) after idle periods, even when there are no dropped packets, loses the benefit of a persistent connection and reduces the transmission rate [90, 91]. Moreover, unfairness occurs between a TCP connection transmitting a large amount of data (long-lived connection) and one transmitting a small amount of data (short-lived connection) because of the difference in the window sizes of the connections [48]. To achieve fairness among such connections, prioritization mechanisms based on Differentiated Services (DiffServ) and Random Early Detection (RED) have been developed [49, 50].

The performance of a thin-client system from the user's point of view under various network conditions has been measured for typical desktop applications like a word processor or presentation creator [92, 93]. Furthermore, network latency has been identified as a key factor determining the performance, especially in a wide area network (WAN) environment [94, 95]. According to [5], network delay is the predominant delay for an interactive application like Secure Shell (SSH), and this type of application is sensitive to response time. We have identified that such delay requirements should be met for thin-client systems because users are very sensitive to delays and jitter in the response packets when they type on a thin-client terminal in a remote office. Although it has been shown that the system response time becomes worse in a WAN environment, the effects of TCP mechanisms have not been clarified.

We have therefore been investigating this subject. We previously described the characteristics of traffic [65] and then outlined an approach for improving the interactive user experience [66, 67]. The present chapter provides three contributions. First, on the basis of actual data traffic analysis, we model download traffic of thin-client systems by using a two-state model with interactive data flows in response to keystrokes and bulk data flows related to screen updates. Since users are more sensitive to the keystroke response time, our objective is to minimize the latency of interactive data flows without increasing that of bulk data flows. Second, through simulation experiments using real field data, we investigate the primary delays during transfers of interactive data flows over a WAN. We then demonstrate that the primary delays are queuing delay in the router and TCP buffering delay in the server, which is caused by interactions between interactive and bulk data flows. Third, we describe comprehensive approaches for performance improvement. We first evaluate the effectiveness of existing options: priority queuing of interactive data flows and use of the TCP Selective Acknowledgment (SACK) option. Although these techniques reduce the probability of short delays occurring, a packet of an interactive data flow is sometimes held in the server for more than a second. We then propose two TCP mechanisms to reduce lengthy delays without modifying the remote desktop protocol itself: modifying the retransmission timeout calculation and enhancing the SACK control.

The rest of this chapter is organized as follows. We introduce our traffic model in Section 3.2. Then we evaluate the end-to-end delays without modifications in Section 3.3, with the existing options in Section 3.4, and with the addition of our improved TCP mechanisms in Section 3.5. We also evaluate the effect of WAN propagation delay in Section 3.6. We conclude and discuss remaining issues in Section 4.5.

3.2 Modeling of Thin-Client Traffic

To improve the performance related to keystrokes, we begin by identifying the data flows corresponding to them and by defining a metric of the flow performance. In this section, we therefore introduce a traffic model for the thin-client data flows and configure the input data for performing the simulation experiments described in following sections. We also define the usability metric for evaluating the experimental results.

3.2.1 Two-State Modeling

This section is based on a one-month observation of our test system (Figure 3.1). The system was composed of about 200 thin-client/server pairs and their communication traffic aggregated at the core switch of the data center was monitored using a network protocol analyzer [96]. Each server, implemented by a blade PC, ran Microsoft® Windows®[3] XP [97]. Each client had access to the corresponding server via Microsoft® RDP [98]. Most of the applications executed in the servers were typical desktop applications such as email clients, web browsers, and Microsoft® Office.

We constructed the model by targeting only response packets from the servers, not request packets to them, because the traffic due to request packets is much lower than the response packet traffic. Network congestion therefore arises when there are many response packets. Although the request packets may be delayed because of background traffic, this consideration is outside the scope of this chapter. Furthermore, to investigate the effect of network latency, we started with a model of response packets not affected by network

[3] Windows is a registered trademark of Microsoft Corporation.

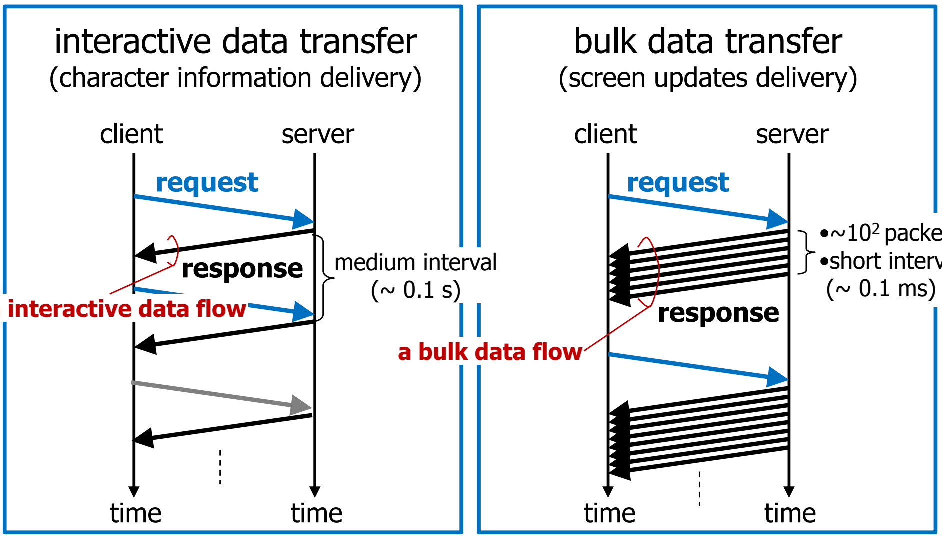


Figure 3.2: Two types of data transfer in thin-client system

latency, i.e., the delay of the TCP ACK response, but sent directly corresponding to the user's usual keyboard/mouse input rate (as if users were using traditional fat clients). We therefore analyzed only packets traversing the intranet, where almost all connections had a round-trip time (RTT) in milliseconds and an end-to-end bandwidth of nearly 100 Mbps.

Observation of data traffic revealed that response packet transfers could be classified roughly into two types, as shown in Figure 3.2. Interactive data transfer is used mainly for character information delivery. An interactive data flow is defined here to carry a single response packet of character information in response to a keystroke (like SSH traffic). Meanwhile, bulk data transfer is used mainly for screen update information delivery. A bulk data flow is defined to be composed of a block of less than about 100 response packets representing screen updates (like HTTP (web) traffic). In our test system, we found that whatever desktop application programs the server was executing, data flows of response packets could be expressed as a mixture of these two types of data flows, although the frequency of each flow depended on the application program used. We therefore considered that the data flows could be modeled using a two-state system, as shown in Figure 3.3,

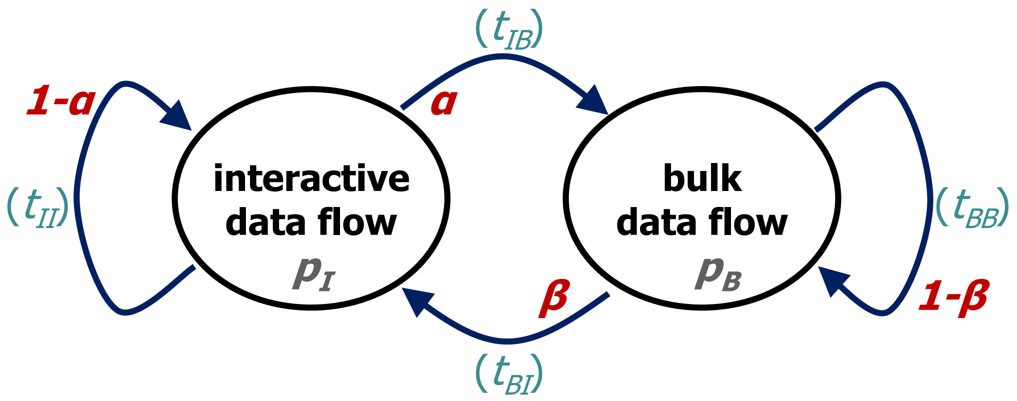


Figure 3.3: Two-state model of data flows

where α is the state transition probability of changing from sending an interactive data flow to sending a bulk data flow and β is that of the opposite change. In addition, p_I is the probability of sending an interactive data flow and p_B is that of sending a bulk data flow. Moreover, t_{II}, t_{IB}, t_{BI}, and t_{BB} are respectively the intervals between sending an interactive data flow and the next flow, between sending an interactive data flow and a successive bulk data flow, between sending a bulk data flow and a successive interactive data flow, and between sending a bulk data flow and the next flow.

To identify the two types of data flows, we analyzed the interarrival time distributions for response packets as well as request packets, as shown in Figure 3.4. The maximum time interval between two adjacent packets was set to 60 s. Furthermore, TCP acknowledgment (ACK) packets were excluded. The response packet distribution has roughly two peaks. The peak centered around 10^{-1} seconds overlaps with that of the request packet distribution, which should be related to the average frequency of up-and-down keystrokes. Meanwhile, the peak at 10^{-4} seconds is related to the interarrival time of two consecutive packets in a block of response packets, i.e., a bulk data flow. We could therefore identify interactive and bulk data flows by setting a threshold for the interarrival time of response packets. If the interarrival time was longer than the threshold, the packet was judged to be an interactive data flow or the head of a bulk data flow. We set the threshold to $10^{-2.2}$ seconds (6.3

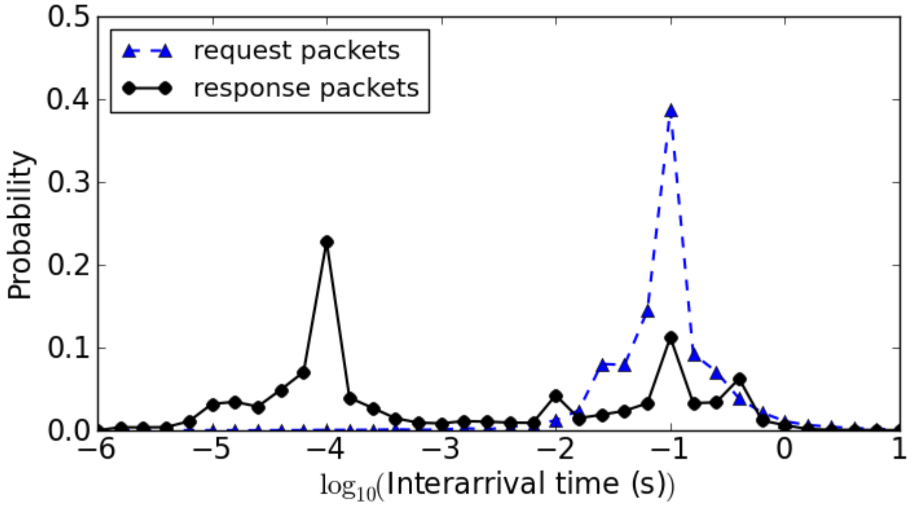


Figure 3.4: Interarrival time distributions of request and response packets

milliseconds) on the basis of the correspondence between the request and response packet distributions. By using the threshold, we classified one-month observed response packets into the two data flow types; this revealed that the average state transition probabilities were $\alpha = 0.09$ and $\beta = 0.42$.

To perform simulation experiments, we extracted the top hundred 300-seconds-long series of response packets whose state transition probabilities (α and β) were closest to the overall average probabilities. The average features of the extracted data are shown in Tables 3.1 and 3.2. An overview of the traffic feature is that interactive data flows predominated in terms of the number of flows while bulk data flows predominated in terms of bytes. The parameters in the tables, t_{II}, t_{IB}, t_{BI}, t_{BB}, p_I, p_B, and c_B (where c_B is the average number of packets included in a bulk data flow), which were calculated from the extracted data, were used to calculate Equation (3.4) in Section 3.2.2.

Table 3.1: Transition probability and interval of response traffic (average for 100-sample data, each 300 seconds long)

	next data flow	
current data flow	interactive	bulk
interactive	0.91	0.09 (α)
	0.27 seconds (t_{II})	0.22 seconds (t_{IB})
bulk	0.42 (β)	0.58
	0.24 seconds (t_{BI})	0.11 seconds (t_{BB}) *

* Average interval between two packets in a block was 0.36 milliseconds.

Table 3.2: Features of response traffic (average for 100-sample data, each 300 seconds long)

	interactive	bulk
number of flows	934.1 (82% (p_I))	198.4 (18% (p_B))
number of packets	934.1 (39%) ⟨1.0/flow⟩	1,467.0 (61%) ⟨7.4/flow (c_B)⟩
bytes	128,189 (8.5%) ⟨137.2/flow⟩ ⟨⟨137.2/packet⟩⟩	1,373,366 (91.5%) ⟨6922.2/flow⟩ ⟨⟨936.2/packet⟩⟩

3.2.2 Usability Metric

According to [95], thin-client users notice the response time when it exceeds 150 milliseconds, which corresponds to the *human response time* [5]; furthermore, they become frustrated and less productive when it exceeds 1 s. We therefore set two thresholds for packet delay in an interactive data flow:

$$\text{Unnoticeable}: \quad < \quad (150 - t_0) \text{ milliseconds} \tag{3.1}$$

$$\text{Productivity degrading}: \quad > \quad (1000 - t_0) \text{ milliseconds,} \tag{3.2}$$

where t_0 is the time for transferring a request packet from a client to a server. With these thresholds, two usability metrics for evaluating the performance of interactive data flows are defined as the ratio of the number of packets whose delays are longer than each threshold to the total number of packets in all the interactive data flows tested; these ratios are called $r_{I_{150}}$ for Threshold (3.1) and $r_{I_{1000}}$ for Threshold (3.2).

Furthermore, by using the two-state model, we can convert each metric to the average cycle of time during which the response time threshold is exceeded in order to understand the evaluation results intuitively. After a long enough period has elapsed, $p_I = \beta/(\alpha+\beta)$ and $p_B = \alpha/(\alpha+\beta)$, respectively. The relationship between the number of interactive data flows, f_I, which includes n_I packets, and the number of bulk data flows, f_B, which includes n_B packets, becomes $f_I : f_B \simeq p_I : p_B$, $f_I = n_I$, and $f_B = n_B/c_B$. The average cycle of time T for completely sending f_I interactive data flows and f_B bulk data flows in the steady state is

$$T = (1-\alpha)f_I t_{II} + \alpha f_I t_{IB} + \beta f_B t_{BI} + (1-\beta) f_B t_{BB} \; . \tag{3.3}$$

Consequently, the average cycle of time exceeding the threshold, $T_{I_{150}}$ or $T_{I_{1000}}$, can be expressed as a function of the $r_{I_{150}}$ or $r_{I_{1000}}$, respectively, as follows.

$$T_{I_k} = \frac{(1-\alpha)t_{II}}{r_{I_k}} + \frac{\alpha t_{IB}}{r_{I_k}} + \frac{\beta p_B t_{BI}}{c_B p_I r_{I_k}} + \frac{(1-\beta) p_B t_{BB}}{c_B p_I r_{I_k}} \quad (I_k = I_{150} \text{ or } I_{1000}) \tag{3.4}$$

3.3 End-to-End Delay Analysis

Using the traffic model in Section 3.2, we evaluated the delay of the current system through simulation experiments using ns-2 (release 2.33) [99].

3.3.1 Network and System Model in Experiments

Our target system has a typical architecture for a Japan-wide intra-firm system (Figure 3.1). It is reasonable to assume that network traffic over the wide-area intranet comprises only traffic between the servers and thin clients since the communication area for the traffic to and from other servers, such as web/mail servers, is mostly limited to the data center and other logical links. Given these considerations, we chose to use the simulation model shown in Figure 3.5. This model consists of 100 pairs of a server (sender) and a thin client (receiver) plus two routers and links between them. The networks in the data center and in the remote

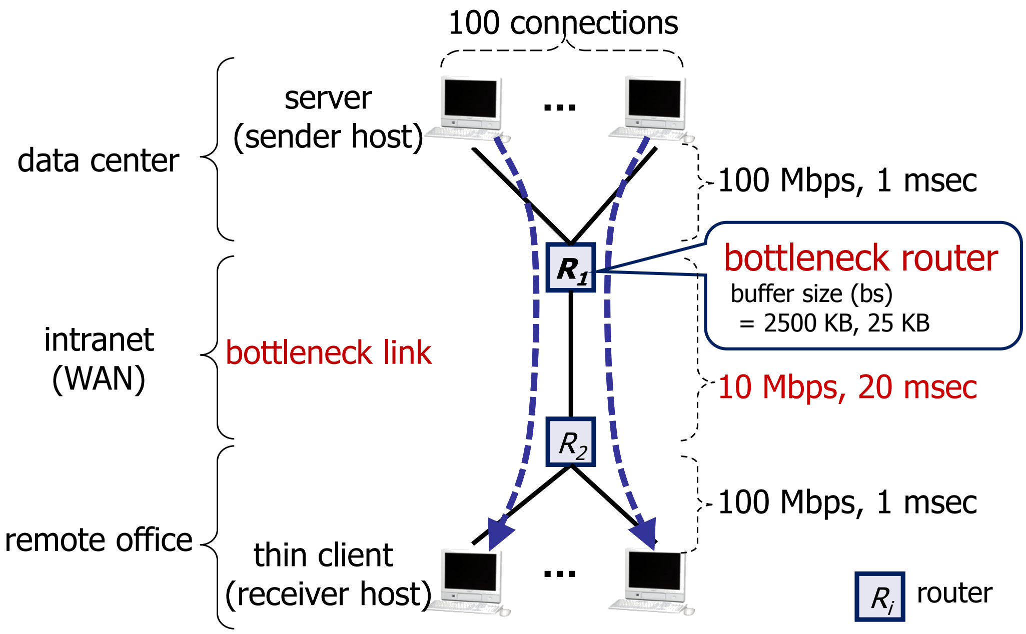


Figure 3.5: Network and system model in simulation experiments

office had a link bandwidth of 100 Mbps and a propagation delay of 1 milliseconds. The bandwidth of the link between the two routers, i.e., the bottleneck link corresponding to the wide-area intranet such as one between Tokyo and Osaka, was set at 10 Mbps, and its propagation delay was 20 milliseconds. An ns-2 *FullTCP* agent running on each server simultaneously transmitted a 300-seconds-long series of response packets extracted in Section 3.2.1 to a corresponding agent running on each client over a TCP Reno connection. A tail-drop (FIFO) discipline was used at the router's buffer. The buffer size was set to two different values: the bandwidth-delay product, i.e., 25 Kbytes, or a sufficiently large value of 2500 Kbytes. We repeated the 300-s simulation seven times with the transmission start time of each sender agent shifted at 0.25-milliseconds intervals because we expected the simulation result to change depending on the degree of synchronization between packets from different senders in the router buffer.

Note that the *FullTCP* agent was used because it counts a sequence in bytes rather than in packets. For convenience in our experiments, the agent on the server labeled packets as "interactive" or "bulk" on the basis of the sending interval. We shall abbreviate the buffer size as "bs" in the figures. We shall also call the router connected to the bottleneck link the "bottleneck router" or simply the "router" here. In addition, to make the model simple, we set the TCP settings so as to turn off the Nagle algorithm, delayed acknowledgments, and slow-start restart. Furthermore, additional experiments with several WAN propagation delays other than 20 milliseconds were executed, as mentioned in Section 3.6.

3.3.2 Factors of End-to-End Delay

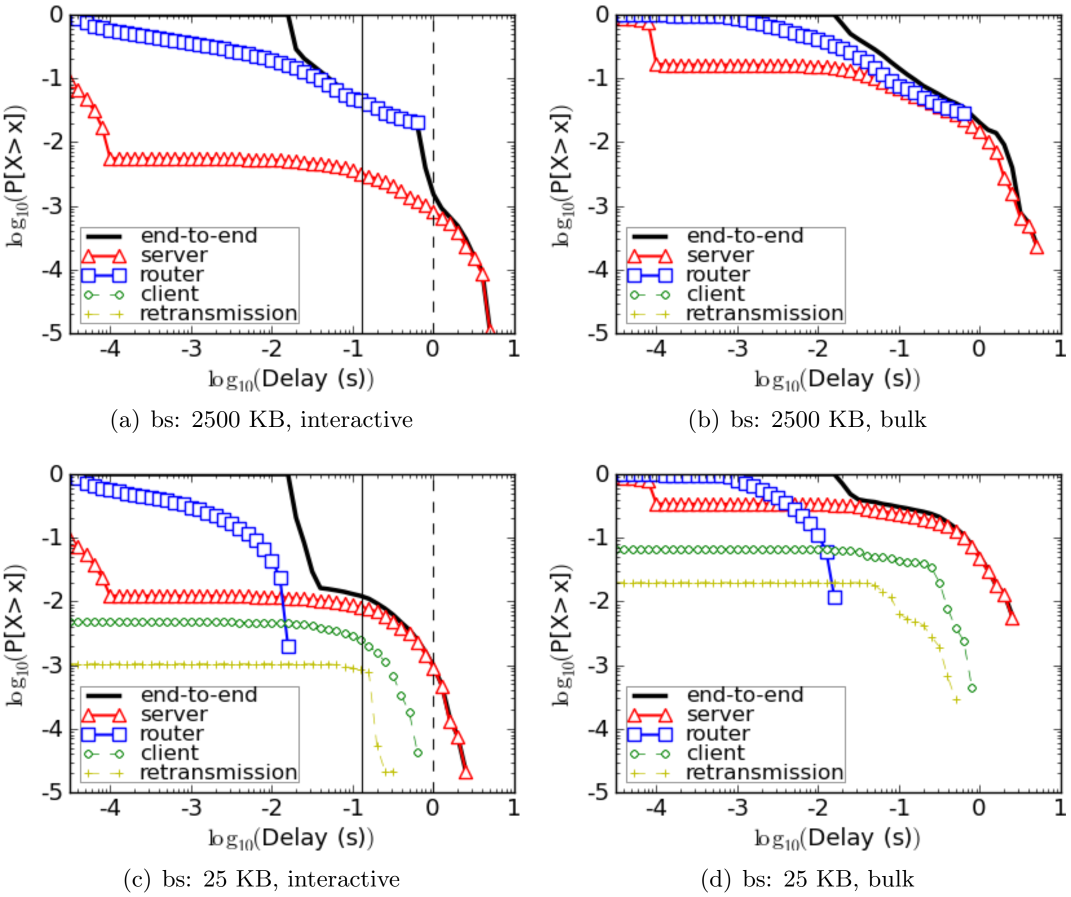
Let us enumerate the factors contributing to the end-to-end delay of a response packet. The end-to-end delay is defined here as the one-way trip time from when an application program on the server sends a response packet to when the application program on a client receives it. This delay consists of delays related to the server, router, client, retransmission, and propagation in each link. The delay in the server is considered to be the sum of the packet buffering and transmission delays in the server because the processing delay is relatively small in our model. The packet buffering in the server is due to small TCP congestion window. The delay in the router is similarly defined as the sum of the packet buffering (queuing) and transmission delays in the router. The delay in the client is buffering delay, which is the time during which the packet is buffered in the TCP layer before being sent to the application layer. We also analyzed the delay for packet retransmission initiated by TCP, which is the time from when a packet is dropped at the router to when a copy of the packet reaches the router. The delay in the client is called "head-of-line blocking" and does not occur unless the packet receiving order is switched owing to subsequent packets leaving earlier the router. This can be simply regarded as the movement of the delay source from the router to the client and does not increase the end-to-end delay. Therefore, we did not try to reduce it in this study.

3.3.3 Delay Distribution without Modifications

One set of the typical simulation results is shown in Figure 3.6. It is plotted in the form of complementary cumulative distribution functions, i.e., ratio of the count of packets whose delays are larger than the value indicated on the horizontal axis to the total packet count. Graphs (a) and (b) show the distributions for the end-to-end delay, the delay in the server, the delay in the (bottleneck) router, the delay in the client, and the retransmission delay for a buffer size of 2500 Kbytes (large buffer case). Graphs (c) and (d) are for a buffer size of 25 Kbytes (small buffer case). Furthermore, graphs (a) and (c) are for interactive data flows, while graphs (b) and (d) are for bulk data flows. In graphs (a) and (c), Thresholds (3.1) and (3.2) for interactive data flows, i.e., 130 and 980 milliseconds (which have had the time for transferring a request packet across WAN subtracted from them), are specified. Moreover, two ratios $r_{I_{150}}$ (the ratio of end-to-end delays exceeding 130 milliseconds) and $r_{I_{1000}}$ (the ratio of ones exceeding 980 milliseconds) are presented below in the from of average values for seven experiments.

Large Buffer Case. The changes in queue length at the router indicate that bursty traffic continuing for several hundred milliseconds (up to about 1 seconds) occurred periodically. This change rate was rapid for packets sent at average intervals of about 300 milliseconds (see Table 3.1). The maximum queue length during the 300-seconds simulation period was about 400 Kbytes; this means that the maximum queuing delay was around 300 milliseconds. The average utilization of the bottleneck link over the entire simulation period was 54%, while that for interactive data flows was 3.5%.

For interactive data flows, $r_{I_{150}}$ was 3.6×10^{-2} and $r_{I_{1000}}$ was 1.7×10^{-3}. As shown in Figure 3.6(a), at the threshold of 130 milliseconds, the end-to-end delay was mainly the delay in the router. Meanwhile, around 980 milliseconds, the delay in the server was predominant. As shown in Figure 3.6(b), the delay in the server was larger for bulk data flows than for interactive data flows. The main component of the end-to-end delays exceeding 100 milliseconds was the delay in the server. This delay was caused by the input of large blocks of packets into the TCP buffer; this made the buffering delay so large that



(a) bs: 2500 KB, interactive

(b) bs: 2500 KB, bulk

(c) bs: 25 KB, interactive

(d) bs: 25 KB, bulk

Figure 3.6: Delay distributions without modifications

the following bulk data flows as well as interactive data flows experienced possible delays. This type of delay seems reasonable because users sometimes experience non-smooth text display on thin-client terminals in a high-network-latency environment when they simultaneously start application programs for showing moving pictures and cause the screens to be updated frequently. Moreover, the delay in the server was even longer because TCP retransmission timeouts occurred without packet being dropped. Details are given in the next section.

Small Buffer Case. For interactive data flows, $r_{I_{150}}$ was 1.2×10^{-2} and $r_{I_{1000}}$ was 9.0×10^{-4}. Because a smaller buffer in the router caused less delay but more drops, the buffering delay in the router was smaller and that in the server was larger compared with the large buffer case, as shown in Figures 3.6(c) and 3.6(d). When a server transmitted large bulk data flows composed of more than a few dozen packets or more than a few hundred packets in some cases, many packets were dropped in less than a second. Furthermore, such bursty drops prevented packets sent by other servers from entering the router buffer. As a result, the server transmitting the large bulk data flows shrank its TCP *cwnd* repeatedly, and other servers shrank theirs as well. This increased the buffering delay in the servers of the bulk data flows and subsequent interactive data flows.

3.4 Existing Options and Problems

In this section, we present appropriate options developed from the existing techniques and evaluate them in the same manner as in the previous section.

3.4.1 Existing Techniques

Prioritizing Interactive Data Flows

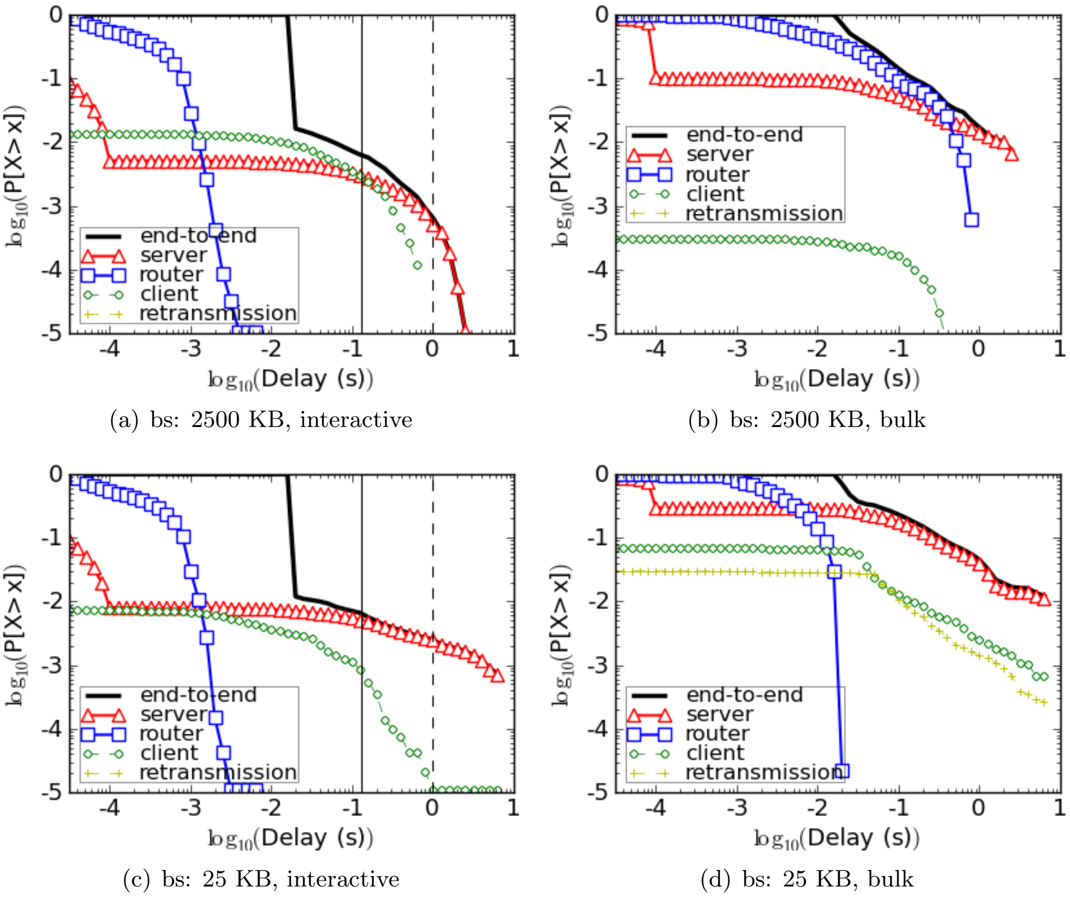
The delay in the router described in large buffer case in Section 3.3.3 can be reduced by implementing priority queuing in the router. This means using two buffers that share the available buffer size: a higher-priority buffer for processing interactive data flows and a lower-priority buffer for processing bulk data flows. Packets in the higher-priority buffer are always processed ahead of packets in the lower-priority buffer. Accordingly, packets of interactive data flows are forwarded with a minimal delay at the router. The holding time of packets in bulk data flows in the lower-priority buffer increases by only a few milliseconds because of the small volume of interactive data flows. Moreover, when the router does not have a large buffer, an interactive data flow can be sent without any possibility of its packets being dropped.

TCP SACK Option

As explained in the small buffer case in Section 3.3.3, if the router does not have a large buffer, multiple packets of a large bulk data flow may be dropped at the router, which in turn causes a delay in the server. This delay can be reduced by applying the TCP SACK option to the server so that it can recover dropped packets quickly and keep *cwnd* large [100].

3.4.2 Evaluation Results

Large Buffer Case. For interactive data flows, $r_{I_{150}}$ decreased to 6.1×10^{-3} because the buffering delay in the router was reduced by priority processing. In contrast, $r_{I_{1000}}$ was 5.5×10^{-4}, where the average value was slightly improved but some values became much worse. This sort of delay was mainly buffering delay in the server, as shown in Figure 3.7(a), and it was further increased by retransmission timeouts in spite of no packets being dropped. Figure 3.8 shows a server's behavior when it invoked such a retransmission timeout. The upper graph shows changes in the sequence number of packets sent by the server and changes in the server's *cwnd*. The lower graph shows changes in the parameters used for calculating the retransmission timeout value RTO, i.e., rtt, $srtt$, $rtttvar$ (see Equations (3.5)–(3.7) in Section 3.5.1), and the changes in the router's queue length. This sort of timeout occurred (A in Figure 3.8), because $srtt$ and $rttvar$, and hence RTO, did not quickly become sufficiently large when large bulk data flows entered the router's buffer and the router's queue length increased (B in Figure 3.8). Moreover, the RTT changed abruptly as a result of interactive data flow prioritization when bulk data flows were switched to interactive ones (C in Figure 3.8). These timeouts were concentrated on several servers sending bulk data flows at that time because the bulk data flows were always placed at the end of the router's queue. Such timeouts sometimes affected a server more than five times within a few seconds.



(a) bs: 2500 KB, interactive

(b) bs: 2500 KB, bulk

(c) bs: 25 KB, interactive

(d) bs: 25 KB, bulk

Figure 3.7: Delay distributions with prioritized interactive data flows and TCP SACK option

Small Buffer Case. For interactive data flows, $r_{I_{150}}$ decreased to 6.4×10^{-3} because the delay in the servers was reduced by using the TCP SACK option. In contrast, $r_{I_{1000}}$ increased to 2.2×10^{-3} as a result of the significant delays for buffering the preceding bulk data flows at the server, as shown in Figure 3.7(c). These buffering delays occurred because the servers sending large bulk data flows experienced bursty drops (hundreds of drops in some cases) at the router. These packet drops were concentrated at a few servers sending bulk data flows at that time.

The server using the TCP SACK option quickly retransmitted such dropped packets. When retransmitted packets were also dropped, the retransmission timeout triggered their

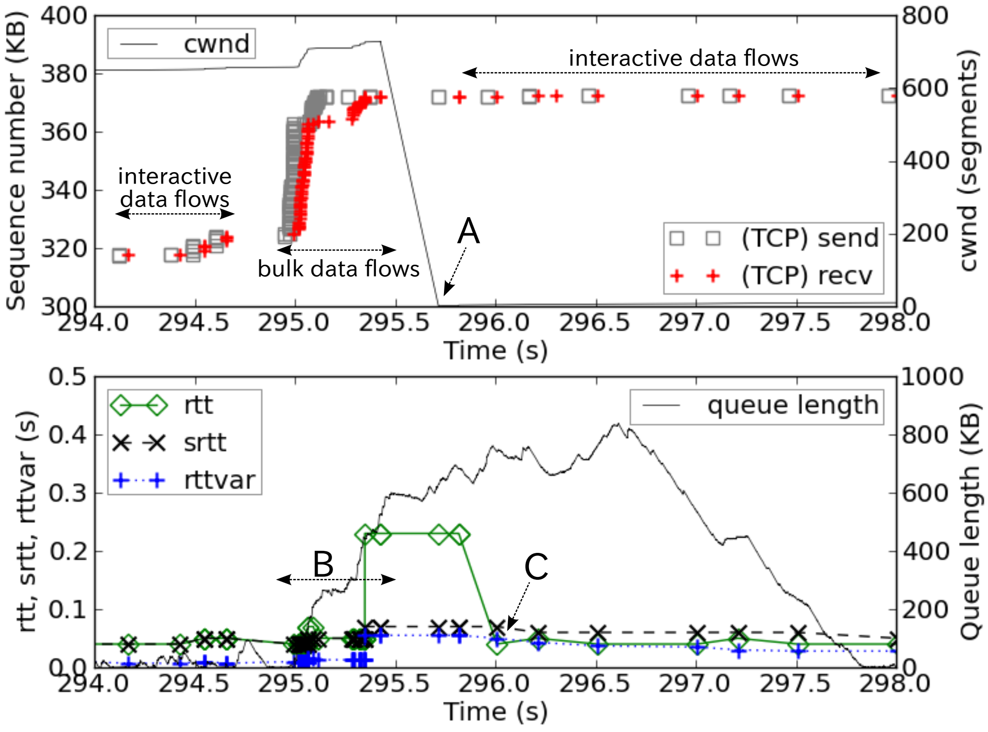


Figure 3.8: Retransmission timeout occurred without packet drops: changes in the sequence number of packets sent by a server and in the server's *cwnd* (upper) and changes in the server's *RTO* parameters and in router queue length (lower)

retransmissions because the TCP fast retransmission mechanism is not applied to lost retransmissions. Figure 3.9 illustrates two problems causing several-second delays after the retransmission timeout. The left graph shows changes in the sequence number of packets sent and received by the server and changes in the server's *cwnd* from 92.5 to 95.0 seconds, which indicates the retransmission timeout. The right graph shows those from 90 to 125 seconds; this is an overview of the several-second delays in the server. In the graphs, the packets are plotted once every ten packets for clarity. These problems were due to the TCP SACK implementation (ns-2's *FullTCP* is similar to 4.x BSD TCP [99]). The first problem occurred when the server received three duplicate ACKs and entered fast recovery mode (*C* in Figure 3.9). The server did not increase *cwnd* when it received an ACK whose value

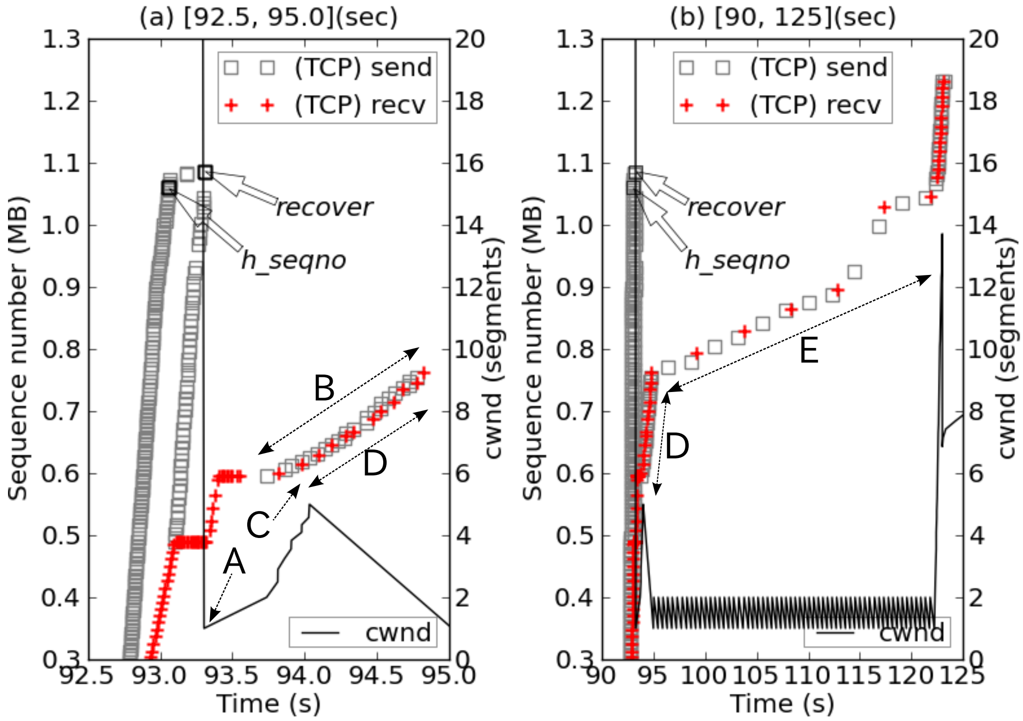


Figure 3.9: Server buffering delays of several seconds caused by bursty packet drops: magnified view of changes in the sequence number of packets sent and received by a server and changes in the server's *cwnd* (left) and their overall view (right)
(Packets sent and received are plotted once every 10 packets)

was lower than *recover* in fast recovery mode (D in Figure 3.9), where *recover* was the highest sequence number transmitted by the server when the first timeout occurred (A in Figure 3.9). The second problem happened after that. When the server received an ACK whose value was lower than *recover*, it determined which packet to retransmit by comparing *h_seqno* with the SACK blocks reported by the paired client, where *h_seqno* is the highest sequence number indicating a hole that has not yet been filled by the fast retransmissions when the first timeout occurred. The server sent the packet pointed to by the ACK when the *h_seqno* was higher than all SACK blocks (B in Figure 3.9). Furthermore, the server would have sent the packet corresponding to the *h_seqno* when the *h_seqno* was lower than a hole reported by the SACK blocks, but it could not do so because *cwnd* was not large enough to accommodate *h_seqno* at that time. As a result, the server waited for extra timeouts (E in Figure 3.9).

3.5 Proposed Mechanisms and Evaluation

The problems described in the previous section are due to the traffic balance between interactive and bulk data flows shown in Tables 3.1 and 3.2. We thus developed several mechanisms and evaluated them in the same way.

3.5.1 Mechanisms for Reducing Delay in Server

Modifying the Retransmission Timeout Value Calculation

The buffering delay in the server described in the large buffer case in Section 3.4.2 can be decreased by keeping the server's *cwnd* large enough. This can be done by preventing the retransmission timer from expiring. The timer expires when *RTO* does not follow a rapid increase in the delay of the router's buffer owing to bulk data flows. Here, *RTO* is calculated on the basis of the measured RTT for the given connection as follows [101].

$$\begin{aligned} srtt &\leftarrow (1-g)srtt + g \cdot rtt && (3.5)\\ rttvar &\leftarrow (1-h)rttvar + h|rtt - srtt| && (3.6)\\ RTO &\leftarrow srttt + 4 \cdot rttvar, && (3.7)\end{aligned}$$

where *rtt* is the latest measured RTT, *srtt* is the current smoothed RTT estimator, and *rttvar* is the smoothed mean deviation estimator. The gains g and h are usually set to 1/8 and 1/4, respectively. In the proposed mechanism, *srtt* is updated by using g:

$$g = \begin{cases} 7/8 & (rtt \geq srtt) \\ 1/8 & (rtt < srtt). \end{cases} \qquad (3.8)$$

As a result, *RTO* reflects the latest RTT, and it keeps pace with any rapid increase in the router queue when large bulk data flows are input. Moreover, *RTO* does not become too small when data flows switch from bulk to interactive under the interactive data flow prioritization setting.

Furthermore, the server initializes *RTO* after the idle period in order to match the change in queue length after a long interval between two data flows. We use the time when the server invokes TCP's slow-start restart to determine the idle period. At that time, the server initializes *srtt* and *rttvar*, but not *cwnd*.

Temporarily Turning Off TCP SACK Control

Significant buffering delays in the server described in small buffer case in Section 3.4.2 can be avoided by having the server recover from bursty drops as quickly as possible. This can be achieved by steadily increasing *cwnd* and not invoking any extra timeouts. Some studies have developed SACK mechanisms to detect and recover a lost retransmission [102, 103]. However, they are not appropriate for recovering bursty packet drops in the present case because they consider random multiple drops. We therefore temporarily turn off the TCP SACK control after a timeout occurs and do not turn it back on until the dropped packets have been recovered, as follows.

- To enable the server to increment *cwnd* when it receives an ACK whose value is lower than *recover* after the timeout, the server is configured to not enter fast recovery mode if it receives three duplicate ACKs after the timeout.
- To prevent the server from becoming incapable of sending a packet after the timeout, the server is configured to send the packet pointed to by an ACK, rather than selecting one by comparing the *h_seqno* and the SACK blocks, if the ACK value is lower than *recover* after the timeout.

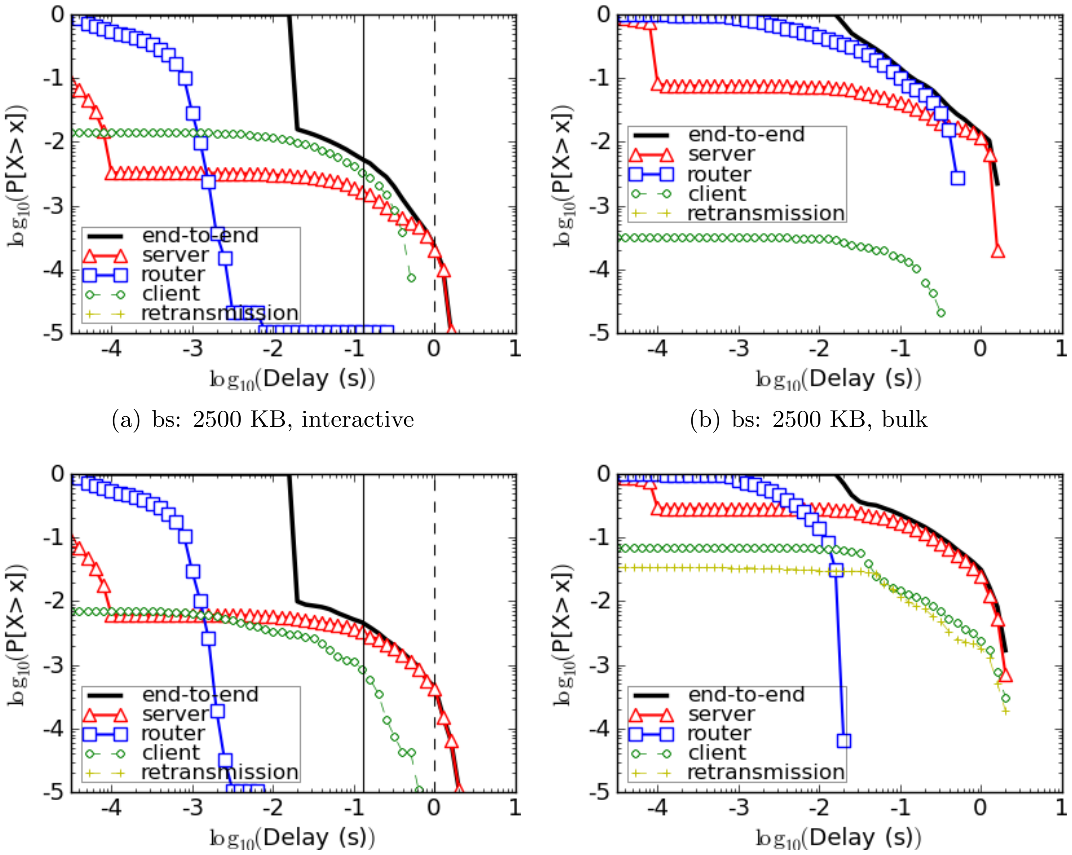
Since these modified processes are similar to those of TCP Reno, traffic burstiness during the modified period is considered to be equivalent to that of TCP Reno. Furthermore, an increase in burstiness is related to bulk data flows. The bulk data flows are given lower priority and thus do not affect the simultaneous interactive data flows at the router buffer.

3.5.2 Evaluation of Proposed Mechanisms

The underlying cause of the server delay described in Section 3.4.2 is that interactive and bulk data flows coexist in a TCP connection. We therefore additionally propose separating the TCP connection and evaluate the proposed mechanisms for two cases. Case 1: a single TCP connection is shared between interactive and bulk data flows; this is the current specification. Case 2: the TCP connection for interactive data flows is separated from that for bulk; this means the application protocols need to be modified. Note that Case 2 is impractical when the thin-client system is based on a proprietary application protocol (as in the case of this chapter), whereas Case 1 with the proposed mechanisms can be applied by using a transport-layer proxy mechanism [104]. The results for Case 2 are therefore used for clarifying the limitations of Case 1.

Case 1: Shared TCP Connection

Large Buffer Case. For interactive data flows, $r_{I_{150}}$ was 5.0×10^{-3}, which means that this ratio was slightly improved by applying only priority queuing. Moreover, $r_{I_{1000}}$ decreased to 2.1×10^{-4}, where the negative effect of the existing option disappeared. Figure 3.11 depicts an example of avoiding TCP timeouts without packet drops, which was obtained under the same experimental conditions as in Figure 3.8. As shown in the lower graph, *srtt* quickly coped with its rapid increase for bulk data flows in the router buffer (A in Figure 3.11). Subsequently, it decreased slowly even though *rtt* suddenly dropped owing to the switch to interactive data flows (B in Figure 3.11). These mechanisms prevented the server's retransmission timer from expiring and stopped its *cwnd* from shrinking. They thus reduced the number of timeouts experienced by all servers. The average number of timeouts over the 300-s simulation period was 3.1 per connection when we used priority queuing for the interactive data flows and the TCP SACK option, and this value decreased to 1.9 per



(a) bs: 2500 KB, interactive

(b) bs: 2500 KB, bulk

(c) bs: 25 KB, interactive

(d) bs: 25 KB, bulk

Figure 3.10: Delay distributions with prioritized interactive data flows, TCP SACK option, and proposed mechanisms for shared TCP connections

connection when the proposed mechanisms were used as well. As a result, the ratio of end-to-end delays exceeding 980 milliseconds for bulk data flows in the servers fell to the level before the priority queuing was configured in the router, as shown in Figure 3.10(b), which decreased $r_{I_{1000}}$ as shown in Figure 3.10(a).

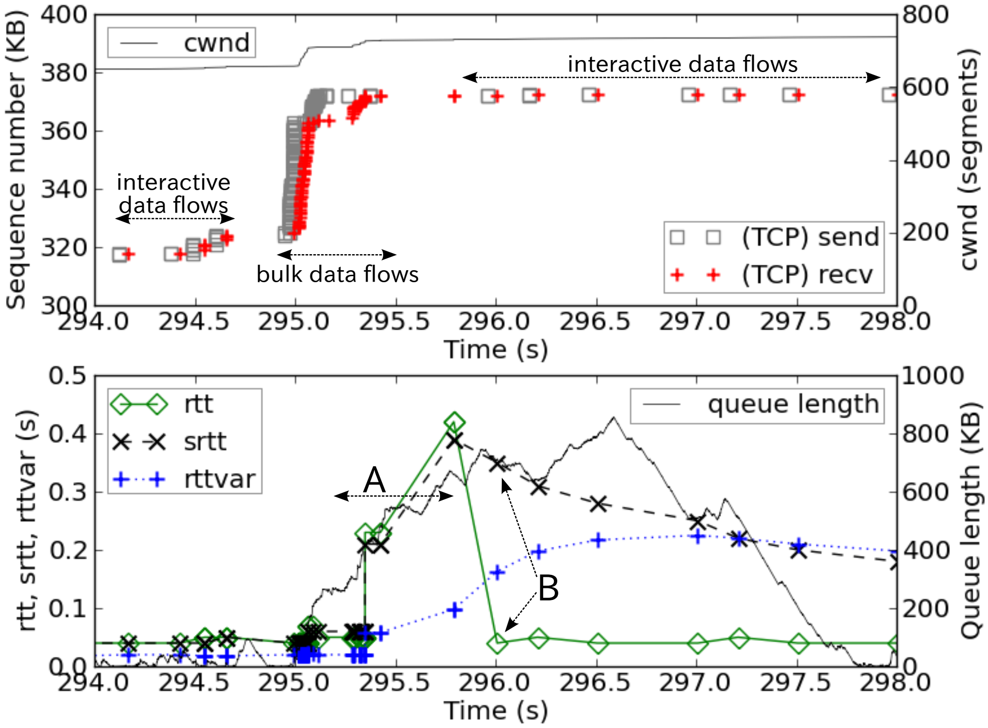


Figure 3.11: Avoidance of retransmission timeout shown in Figure 3.8: changes in sequence number of packets sent by a server and in the server's *cwnd* (upper) and changes in the server's *RTO* parameters and in router's queue length (lower)

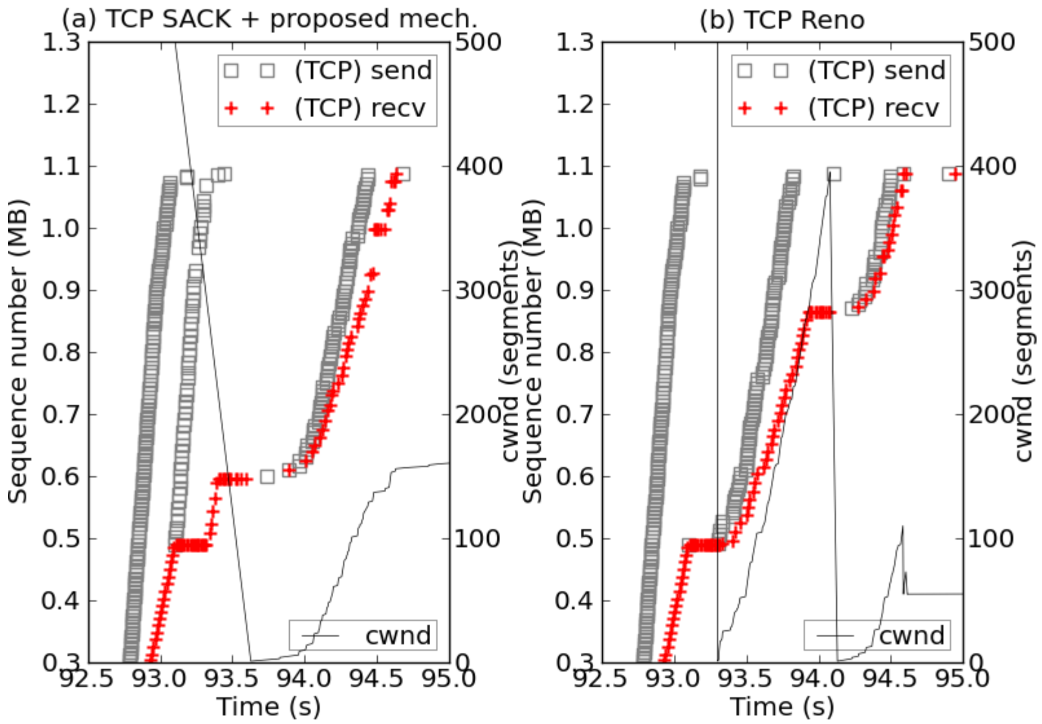
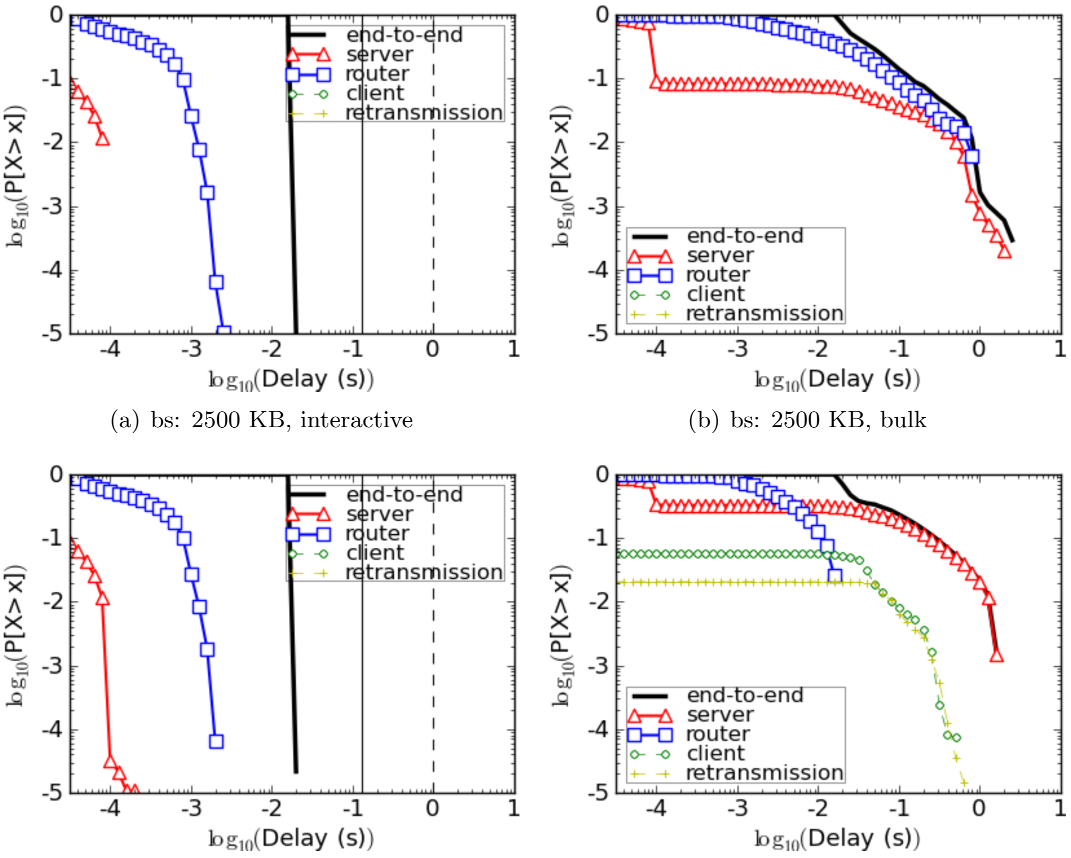


Figure 3.12: Faster recovery from bursty packet drops shown in Figure 3.9 by using the TCP SACK option and proposed mechanisms (left) and by using TCP Reno (right)
(Packets sent and received are plotted once every 10 packets)

Small Buffer Case. For interactive data flows, $r_{I_{150}}$ slightly decreased to 4.6×10^{-3} from the value for the case where only the TCP SACK option was applied. Furthermore, $r_{I_{1000}}$ greatly decreased to 4.0×10^{-4}. The left graph of Figure 3.12 shows a faster recovery from bursty packet drops, under the same setting as in Figure 3.9. The recovery speed, i.e., burstiness, with this sequence was nearly as high as when using TCP Reno in the server, as shown in the right graph of Figure 3.12. As a result, the several-second delays in the servers for transmitting bulk data flows were eliminated, as shown in Figure 3.10(d). This reduced the over-980-ms delays in the server and thus the end-to-end delays for interactive data flows, as shown in Figure 3.10(c).



(a) bs: 2500 KB, interactive

(b) bs: 2500 KB, bulk

(c) bs: 25 KB, interactive

(d) bs: 25 KB, bulk

Figure 3.13: Delay distributions with prioritized interactive data flows, TCP SACK option, and proposed mechanism for separate TCP connections

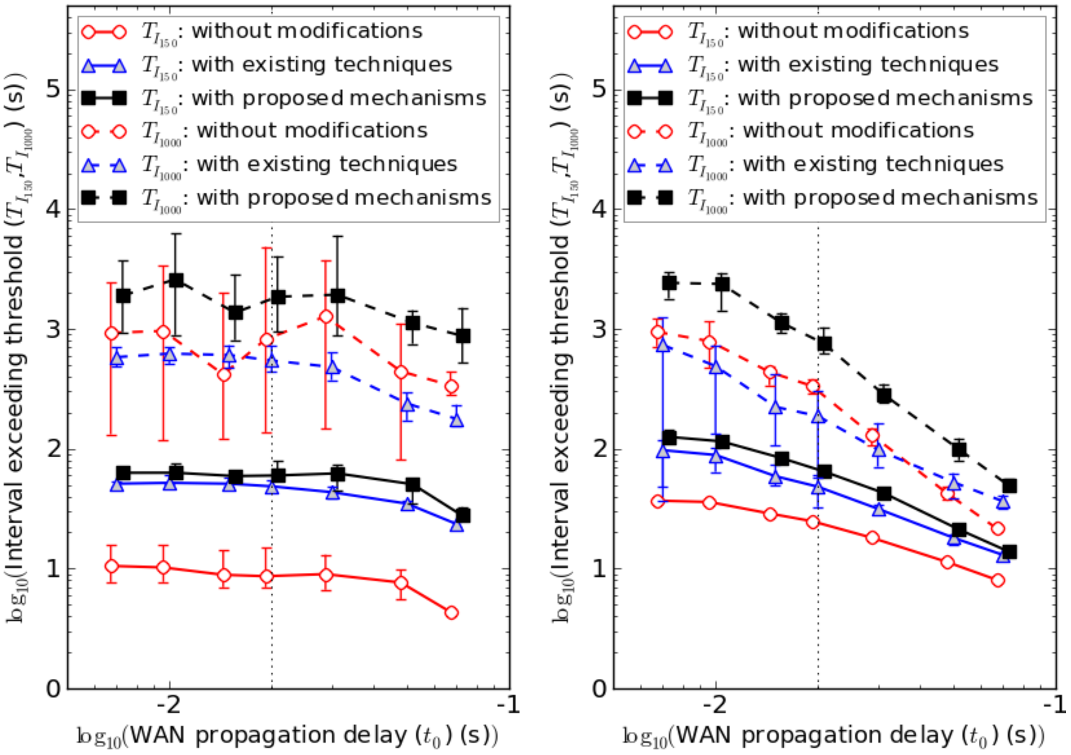
Case 2: Separate TCP Connections

We assumed that the client application did not have to maintain the packet order between the interactive and bulk data flows sent by the server application. In both buffer cases, the end-to-end delays for interactive data flows were almost completely due to propagation delay; these delays were much lower than 130 milliseconds, as shown in Figure 3.13. Since interactive data flows were transferred using a dedicated TCP connection, the delay for interactive data flows was independent of that for bulk data flows.

3.6 Effect of WAN Propagation Delay

Finally, we evaluate the effect of WAN propagation delay on the results for a single TCP connection shared by interactive and bulk data flows without modifications (Section 3.3.3) with the existing techniques (Section 3.4.2) and with the proposed mechanisms (Section 3.5.2). Here, we clarify the improvements among these three approaches. For each case, we performed additional experiments with several WAN propagation delays ranging from 7 to 70 milliseconds. The evaluation results were converted into the average intervals between delays exceeding either Threshold (3.1) or (3.2), i.e., $T_{I_{150}}$ and $T_{I_{1000}}$ by using Equation (3.4). This means that the longer the interval becomes the less often users notice the delay and get frustrated. Note that the probability of delays exceeding the threshold, and hence the intervals as well, is shared by all users in the system, rather than entirely affecting each user.

Figure 3.14 shows changes in the intervals for interactive data flows, where each marker specifies the mean and each error-bar specifies the maximum and minimum values for seven experimental runs for each setting. When the WAN propagation delay increased, its effect was not so obvious in the large buffer case; meanwhile, the intervals became shorter along with the delay in the small buffer case. In the large and small buffer cases, average $T_{I_{150}}$ values with the proposed mechanism were 55 and 67 seconds; these were 6.5 and 2.8 times longer, respectively, than those for without modification. Moreover, they both slightly increased the improvement obtained by applying only the existing techniques. Furthermore, average $T_{I_{1000}}$ values with the proposed mechanism were 1.7×10^3 and 1.0×10^3 seconds; these were 3.6 and 3.9 times longer, respectively, than those for with only the existing techniques. In addition, the variation in $T_{I_{1000}}$ without modification in the large buffer case and that with the existing techniques at less than 20 milliseconds in the small buffer case were both considerably large if large bulk data flows from some servers reached the router in a highly synchronized manner for a few milliseconds. Meanwhile, with the proposed mechanisms, the variation was relatively small. These results show that the proposed mechanisms negate the shortcomings of the existing approaches and improve upon them. In addition, for bulk



(a) bs: 2500 KB, interactive

(b) bs: 25 KB, interactive

Figure 3.14: Interval between delays exceeding threshold ($T_{I_{150}}$: solid lines and $T_{I_{1000}}$: dashed lines (Equation (3.4)))
(Horizontal positions are adjusted for error bars visibility.)

data flows, the intervals obtained with the proposed mechanisms were equivalent to other results.

Note that these improved $T_{I_{150}}$ and $T_{I_{1000}}$ values could be equivalent to (or at least, not much smaller than) the similar delay cycles in the case of common fat-clients, although that depends on the fat-client's specification such as memory size and CPU speed.

3.7 Conclusion

The thin-client system has a particular traffic pattern involving a mixture of interactive and bulk data flows. This study aimed to improve the performance of interactive data flows while not influencing bulk data flows. We found that the average end-to-end delay of the interactive data flows could be reduced by applying priority queuing and utilizing the TCP SACK option. The occurrences of lengthy delays exceeding about 1000 milliseconds resulting from the buffering delay in the server could be further reduced by modifying the retransmission timeout calculation and temporarily halting the TCP SACK control. Nevertheless, to completely eliminate the effect of bulk data flows on interactive data flows, it is necessary to establish separate TCP connections for these two types of data flows. The remote desktop protocol can change its setting and reduce the traffic volume of bulk data flows to cope with a low-bandwidth environment. Researchers have previously attempted to improve performance by caching screen updates [105, 106]. However, these measures do not reduce the bulk data flow volume to the level of the interactive data flow volume. Thus, delays caused by interactions between interactive and bulk data flows will still occur as long as the two flow types share a TCP connection. In the future, we would like to investigate network architectures for *cloud computing* using thin-client technology.

Chapter 4

Power Consumption of a WAN with the Aid of Distributed Computing

4.1 Introduction

With the development of software as a service (SaaS) and cloud computing, computing resources, i.e., servers and storage systems, are increasingly being concentrated in a few huge data centers [9]. Moreover, an enormous amount of data will be created as a result of promoting the information society such as data related to digital video, surveillance cameras, and sensor-based applications [107, 51]. These changes will lead to a massive amount of data traversing a wide area network (WAN), which will increase the power consumed in the WAN. A distributed computing network (DCN) such as a content delivery network, which deploys a caching mechanism and related processing functions at network nodes [108], is a one of the major approaches to complement the concentrated processing in a data center [109]. It not only improves the response time to clients but also reduces the traffic from/to a data center (e.g., in cases of peer-to-peer (P2P) systems [110, 111, 112, 113, 114]), resulting in that the power consumed in the WAN decreases. This chapter focuses on such an aspect of energy-saving through the use of the DCN [69].

When we investigate power efficiency in the WAN, it is important to consider traffic locality resulting from where traffic is supplied and demanded. For example, data collected from a local area for surveillance in a *smart city* [115] could be used mainly in that area and its surrounds. In such a case, to avoid the round-trip route to a distant data center, localizing the traffic in the area with the aid of a DCN is effective. Our goal is therefore to explore and describe the power efficiency profile of such a DCN, especially considering traffic locality.

In recent years, power efficiency in telecommunications networks has been one of the major research issues in this field [53, 54, 55]. A typical WAN uses Internet protocol (IP) over a wavelength division multiplexing (WDM) optical network. IP routers (often called routers here) are the dominant power consumer in the networks; they consume 90 % of the total power in such a network [60, 56]. There are several approaches for developing the network power consumption models and energy-saving techniques; this approaches are classified by their focus on the router whose power is mainly consumed by its chassis, switching fabric, line cards, and ports. The first approach treats a router as one power element. The number of active routers in the network is minimized by turning the router off or putting it to sleep, so that the total power is optimized while satisfying the traffic requirements [57, 58]. The second approach is based on router power consumption being the sum of its individual elements such as chassis and line card. Minimum number of elements are actually powered on to guarantee the traffic requirements [116]. The third approach focuses on the router's ports. Alternative routing for bypassing the active ports eliminates the power consumed by the ports themselves [59, 60, 61] or by their connecting links [117, 118]. The last approach assumes that the router exhibits energy proportionality [119, 120]. Such characteristics are that a router's power consumption is expressed as a linear function of the router's traffic load (i.e., energy per bit) [121], as a step-increasing function of the traffic load [122], and as the sum of the power consumption in active and idle modes [123].

Furthermore, with energy-proportional behavior assumed, the energy efficiency of distributed content delivery architectures have been evaluated in the case of intermediate routers capable of storing popular contents for retrieval throughout the network [124, 125], in the case of distributed servers within an Internet service provider (ISP) compared with those located at a data center [126], and in the case of ISP-controlled home gateways distributing contents in a P2P fashion [127]. These studies aimed to achieve efficient content distribution, so they took into account the popularity of the content, i.e., which content is more likely to be downloaded. However, the content locality, i.e., where the content is created, stored, and downloaded was beyond their scope.

In this chapter, we also assume that the router has energy proportionality; furthermore, we focus on evaluating the effect of traffic locality on the DCN's power efficiency as compared with the conventional computing scheme of processing only in the data center. For this purpose, we first formulate the problem of optimizing the DCN power consumption; this network can include a distributed server function providing data compression and caching to reduce the transit traffic through the WAN to a data center. Second, in order to evaluate the DCN power consumption, we describe the details of the network: topology, traffic supply and demand matrix, metric of data center location, and router power consumption model. The traffic matrix is introduced to quantify the traffic locality, i.e., the amount of traffic related to the geographical closeness between the demand and supply areas. Third, by applying a heuristic method, we examine how much the system power consumption is affected by traffic locality and the ratio of download traffic to upload traffic, as well as the data center location, and the router power consumption profile. Note that the system power consumption is assumed to depend on the traffic amount, so we will decrease the system power by reducing the traffic amount rather than by rerouting and grooming the traffic to increase the number of sleeping elements.

The rest of this chapter is organized as follows. In Section 4.2, we illustrate an application scenario and introduce a formulation of the power consumption problem. In Section 4.3, we describe the network model. Then, in Section 4.4, we evaluate the effect of traffic locality on the system power consumption. Finally, in Section 4.5, we give conclusions and discuss remaining issues.

4.2 Application and Power Consumption Models

We first give an application scenario suitable for the DCN to supplement the concentrated processing in a data center. We then construct a model of the system power consumption according to the scenario.

4.2.1 Application Scenario

A schematic diagram representing an application scenario is shown in Figure 4.1. One of the typical applications is a surveillance application system that continuously collects data from widespread sensors such as measuring devices and still & video cameras. The application system comprises a distributed infrastructure including the sensors for collecting data, clients for using data processed on servers, network nodes (called nodes here), a server (with bundled storage) in a data center, and links between them. Each network node is an IP router capable of binding a server function, which we call a *node-attached virtual server* (*NAVS*). The NAVS can be implemented by using a general server device or a service module card added to the router (e.g., [128]). The NAVS might have the same processing functions as a server in the data center; moreover, it provides application-level service in order to reduce the volume of traffic traversing the WAN, as follows.

When a sensor collects raw data and uploads it to the data center, the sensor sends the data to the nearest NAVS. The NAVS then compresses the received upload data by filtering, re-segmenting, or encoding it and transfers it to the data center server. Furthermore, the NAVS caches the received data in a local storage device, after processing it if necessary, in preparation for the future requests from clients. When a client downloads the data from the data center, it requests it from the nearest NAVS. If the requested data originated from the local area and the NAVS has cached it, the NAVS sends the cached data to the client. Otherwise, the NAVS downloads it from the data center server and forwards it to the client.

Here, a *local area* for each NAVS is defined as the area where the NAVS collects and delivers data. The total area of the sensors/clients is covered by all NAVSes. Thus, when another NAVS is added, each of the local areas becomes smaller than before.

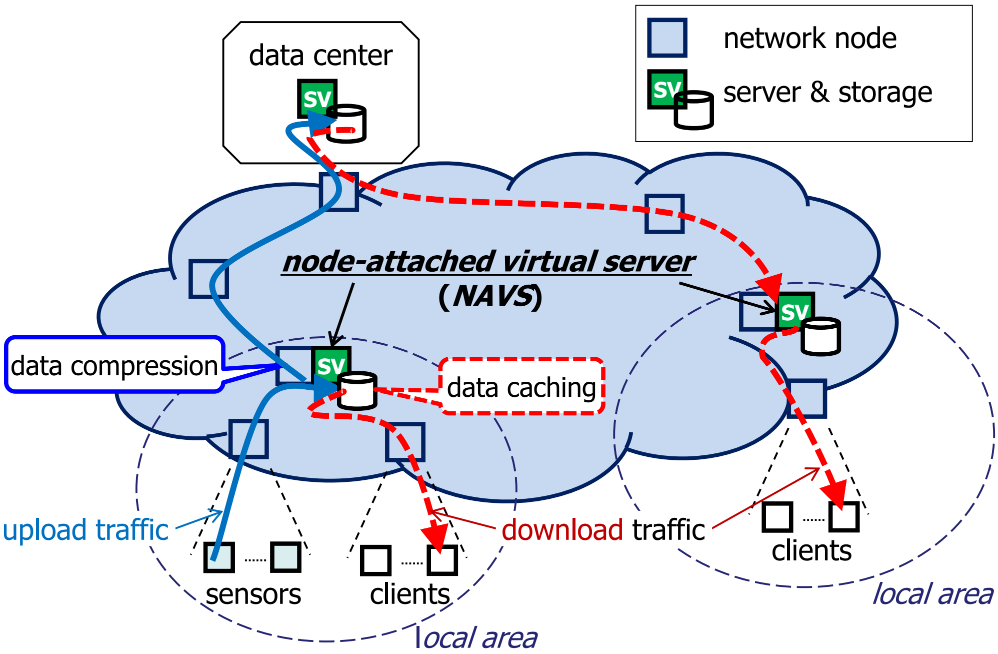


Figure 4.1: Traffic flows of application scenario

4.2.2 System Power Consumption Model

In what follows, we formulate the problem of optimizing the DCN power consumption. We set that the network topology $G = (N, E)$ consists of a set of network nodes N and that of links E. Moreover, We utilize the following notations and definitions, where the traffic volume is measured in bits per second:

Sets and parameters:

N set of nodes in the network topology.

N_i^+ set of neighboring nodes from which node $i{\in}N$ receives traffic.

N_i^- set of neighboring nodes to which node $i{\in}N$ sends traffic.

f_{ij} traffic volume flowing on the link from node $i{\in}N$ to node $j{\in}N$

g_i traffic volume routed through node $i{\in}N$; this is equal to the traffic volume processed by the NAVS attached to node i.

t_{sd} traffic volume flowing from source node $s{\in}N$ to destination node $d{\in}N$

f_{ij}^{sd} traffic volume flowing from source node $s{\in}N$ to destination nodes $d{\in}N$ that is routed through the link from node $i{\in}N$ to node $j{\in}N$. ($f_{ij}^{sd}{\in}[0, t_{sd}]$).

$a_{ij}(f_{ij})$ power consumption of the link from node $i{\in}N$ to node $j{\in}N$; this is expressed as a function of f_{ij}.

$b_i(g_i)$ power consumption of node $i{\in}N$; this is expressed as a function of g_i.

$c_i(g_i)$ power consumption of the NAVS attached to node $i{\in}N$. It is the power consumption of the data center server received traffic from node $i{\in}N$ as well. This is also expressed as a function of g_i.

Variables:

P number of NAVSes introduced in the network, each of which is attached to a corresponding node.

$x_{ij}{\in}\{0,1\}$ binary variables that take the value of 1 if there is a link from node $i{\in}N$ to node $j{\in}N$ and 0 otherwise.

$y_i{\in}\{0,1\}$ binary variables that take the value of 1 if a NAVS is attached to node $i{\in}N$ and 0 otherwise.

To make the model simple, we assume that there is one data center, which has one data center server connected to node $c{\in}N$. Given the above definition, our power-minimized design model is as follows.

Objective: minimize

$$E = \sum_{i\in N}\sum_{j\in N} a_{ij}(f_{ij})x_{ij} + \sum_{i\in N} b_i(g_i) + \left(\sum_{i\in N} c_i(g_i)y_i + c_c(g_c)\right) \quad (\forall c{\in}N) \tag{4.1}$$

Subject to:

$$\sum_{i\in N} y_i = P \tag{4.2}$$

$$\sum_{j\in N_i^+} f_{ji}^{sd} - \sum_{j\in N_i^-} f_{ij}^{sd} = \begin{cases} -t_{sd} & (\forall i\in N, \forall s\in N, \forall d\in N, i = s) \\ t_{sd} & (\forall i\in N, \forall s\in N, \forall d\in N, i = d) \\ 0 & (\text{otherwise}) \end{cases} \tag{4.3}$$

$$f_{ij} = \sum_{s\in N} \sum_{d\in N} f_{ij}^{sd} \quad (\forall i\in N, \forall j\in N) \tag{4.4}$$

$$g_i = \begin{cases} \sum_{j\in N_i^+} \sum_{s\in N} \sum_{d\in N} f_{ji}^{sd} & (\forall i\in N, i\neq s) \\ \sum_{j\in N_i^-} \sum_{s\in N} \sum_{d\in N} f_{ij}^{sd} & (\forall i\in N, i = s) \end{cases} \tag{4.5}$$

Objective (4.1) states that the system power consumption (E) is modeled as the sum of the power consumptions of all links, of all nodes, and of all NAVSes as well as of the data center server. Constraint (4.2) states that the number of NAVSes introduced in the network is P. Constraint (4.3) ensures that the traffic flow is conserved at any node on the path from any source node to any destination node. Constraints (4.4) and (4.5) evaluate the total flow routed on each link and through any node, respectively.

The above formulation is a combinatorial optimization problem for finding the locations of P nodes, each of which binds a NAVS, minimizing E in Objective (4.1). This problem is analogous to the uncapacitated facility location problem [129] and *NP*-hard to solve optimally.

4.2.3 Traffic Flow Formulation

In addition to the previous subsection, we also formulate t_{sd} when the traffic is flowing according to the application scenario described in Section 4.2.1. We define additional terms as follows.

N^e set of nodes accessed from sensors and clients. ($N^e \subseteq N$)

N^v set of nodes; each of which binds a NAVS. ($N^v \subseteq N$)

n_e upload traffic volume transmitted from node $e \in N^e$ to node with a NAVS ($v \in N^v$); this traffic is eventually transferred to the data center.

$m_{ee'}$ download traffic volume, which originated at node $e \in N^e$, received by node $e' \in N^e$.

γ compression ratio of upload traffic volume at the NAVS attached to node $v \in N^v$. ($0 < \gamma \leq 1$)

$z_{sd} \in \{0, 1\}$ binary variables that take the value of 1 if source node $s \in N$ sends data to destination node $d \in N$ and 0 otherwise.

Additional constraints on Objective (4.1) are stated below.

Subject to:

$$z_{ev} \leq y_v \quad (\forall e \in N^e, \forall v \in N^v) \tag{4.6}$$

$$\sum_{v \in N^v} z_{ev} = 1 \quad (\forall e \in N^e) \tag{4.7}$$

$$z_{v'e'} \leq y_{v'} \quad (\forall v' \in N^v, \forall e' \in N^e) \tag{4.8}$$

$$\sum_{v' \in N^v} z_{v'e'} = 1 \quad (\forall e' \in N^e) \tag{4.9}$$

For upload traffic, constraint (4.6) states that a NAVS is attached to destination node $v{\in}N^v$ receiving traffic from node $e{\in}N^e$. For download traffic, constraint (4.8) also states that a NAVS is attached to source node $v'{\in}N^v$ sending traffic to node $e'{\in}N^e$. Constraints (4.7) and (4.9) ensure that a node uploads and downloads traffic to/from a node with a NAVS, respectively.

Then, the end-to-end traffic volume is formulated as follows.

Upload traffic

The upload traffic volume originating from node $e{\in}N^e$ destined to node with a NAVS $v{\in}N^v$ is

$$t_{ev} = n_e z_{ev} \ (\forall e{\in}N^e, \forall v{\in}N^v). \tag{4.10}$$

When this node with a NAVS ($v{\in}N^v$) receives traffic, it uploads γ times the traffic to the node connected to the data center ($c{\in}N$). The traffic volume from node $v{\in}N^v$ to node $c{\in}N$ is

$$t_{vc} = \gamma \sum_{e{\in}N^e} n_e z_{ev} \ (\forall v{\in}N^v, \forall c{\in}N). \tag{4.11}$$

Download traffic

Node $e'{\in}N^e$ downloads all the traffic destined to itself from node with a NAVS ($v'{\in}N^v$). The traffic volume from node $v'{\in}N^v$ to node $e'{\in}N^e$ is therefore

$$t_{v'e'} = \sum_{e{\in}N^e} m_{ee'} z_{v'e'} \ (\forall v'{\in}N^v, \forall e'{\in}N^e). \tag{4.12}$$

The traffic $t_{v'e'}$ includes the traffic cached in node with a NAVS ($v{\in}N^v$) and that forwarded from the data center. The download traffic that node $e'{\in}N^e$ receives from the node connected to the data center ($c{\in}N$) is all the traffic destined to node $e'{\in}N^e$ itself except for that stored locally in node with a NAVS ($v'{\in}N^v$). The traffic cached in this node with a

NAVS ($v'{\in}N^v$) is the traffic from all the nodes $e{\in}N^e$ connected to node $v'{\in}N^v$ itself; this traffic is $\sum_{e\in N^e} m_{ee'} z_{ev'}$. Therefore, the total traffic volume from node $c{\in}N$ to node $v'{\in}N^v$ is

$$t_{cv'} = \sum_{e'\in N^e} \sum_{e\in N^e} m_{ee'}(1 - z_{ev'}) z_{v'e'} \quad (\forall c{\in}N, \forall v'{\in}N^v). \tag{4.13}$$

4.3 Network and System Model

In this section, in order to evaluate the power consumed by the DCN, we describe the network system in detail.

4.3.1 Network Topology

We consider the Japan-wide network system depicted in Figure 4.2. This network has a hierarchical architecture composed of a core network and edge networks. The core network is a Japan-wide network composed of 47 core nodes, i.e., IP routers, located in the 47 prefectural capitals. The core nodes are connected by 75 links according to the topology of road and railroad networks. Meanwhile, an edge network is a prefecture-wide network where each edge node, which are also IP routers, aggregates sensors/clients and connects to a core node at a distance of 1 hop. In all the edge networks, there are 1194 edge nodes; these are located in the areas covered by local governments, i.e., wards, cities, and districts as of 2008. Each of these core and edge nodes is capable of binding a NAVS. In addition, a server in a data center is connected directly to the core node where the data center is located. Note that access networks are not included in this topology, because their energy consumption is hardly affected when the traffic load changes [130].

4.3.2 Traffic Supply and Demand Matrix

In order to define the metric of traffic locality, we introduce a matrix giving the traffic volumes between source and destination nodes in the network.

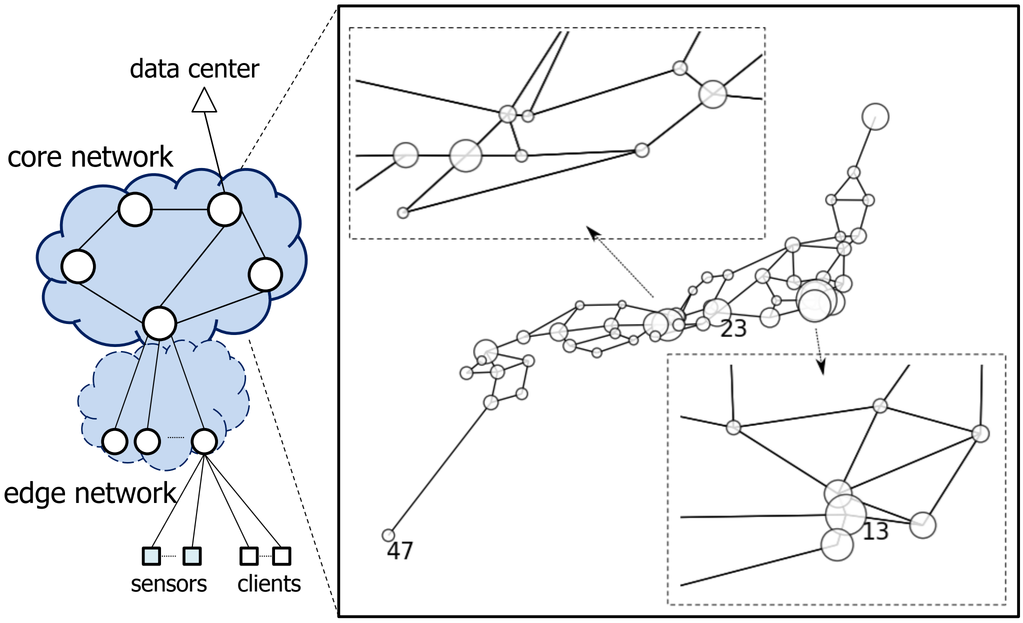


Figure 4.2: Network topology (node size proportional to the number of households in the prefecture where it is located (right figure))

Upload traffic

To upload traffic to a data center via a NAVS, each node in an edge network generates a traffic load proportional to the number of households (as of 2008) in the area such as ward, city, and district where the edge node is located; this traffic load is set to n_e $(e{\in}N^e)$ mentioned in Section 4.2.3.

Download traffic

To define the metric of traffic locality resulting from where traffic is supplied and demanded, we introduce a traffic supply and demand matrix based on a gravity model [131]. This matrix is expressed by $m_{ee'}$ $(e{\in}N^e, e'{\in}N^e)$ explained in Section 4.2.3, as follows:

$$m_{ee'} = k\frac{A_e A_{e'}}{d_{ee'}^{\beta}} \quad (\forall e{\in}N^e, \forall e'{\in}N^e), \tag{4.14}$$

where k is a proportionality constant, A_e and $A_{e'}$ are factors related to the traffic supply and demand amounts in the areas where the source node $e{\in}N^e$ and destination node $e'{\in}N^e$ are located, respectively, $d_{ee'}$ is the distance between these two nodes, β is a parameter for changing the relationship between the distance and traffic amounts exchanged between the two areas. Here, A_e and $A_{e'}$ are assigned to the number of households in the areas, respectively. In addition, distance $d_{ee'}$ is expressed as the hop count between the two nodes. If the source and destination nodes are the same node, the value of $d_{ee'}$ is set to 0.1 for convenience.

Furthermore, we denote the ratio of the total amounts of download traffic to upload traffic by α. The download traffic that originated at node $e{\in}N^e$ is as follows:

$$\sum_{e'\in N^e} m_{ee'} = \alpha n_e \quad (\forall e{\in}N^e). \tag{4.15}$$

The constant k in Equation (4.14) is therefore calculated as follows:

$$k = \frac{\alpha n_e}{\sum_{e'\in N^e} \frac{A_e A_{e'}}{d_{ee'}^{\beta}}} \quad (\forall e{\in}N^e). \tag{4.16}$$

β specifies the degree of traffic locality. Figure 4.3 shows an example of how β changes traffic locality. This figure shows the ratio of traffic flowing within the edge network to the total traffic as a function of β. If β is smaller than 1.0, the traffic flowing within the edge network is less than 30%. On the other hand, when the β is larger than 2.0, this traffic flow is more than 80%.

4.3.3 Metric of Data Center Location

The location of the data center has an large effect on the power consumed for forwarding the traffic to the data center. We therefore define a metric by using the closeness centrality

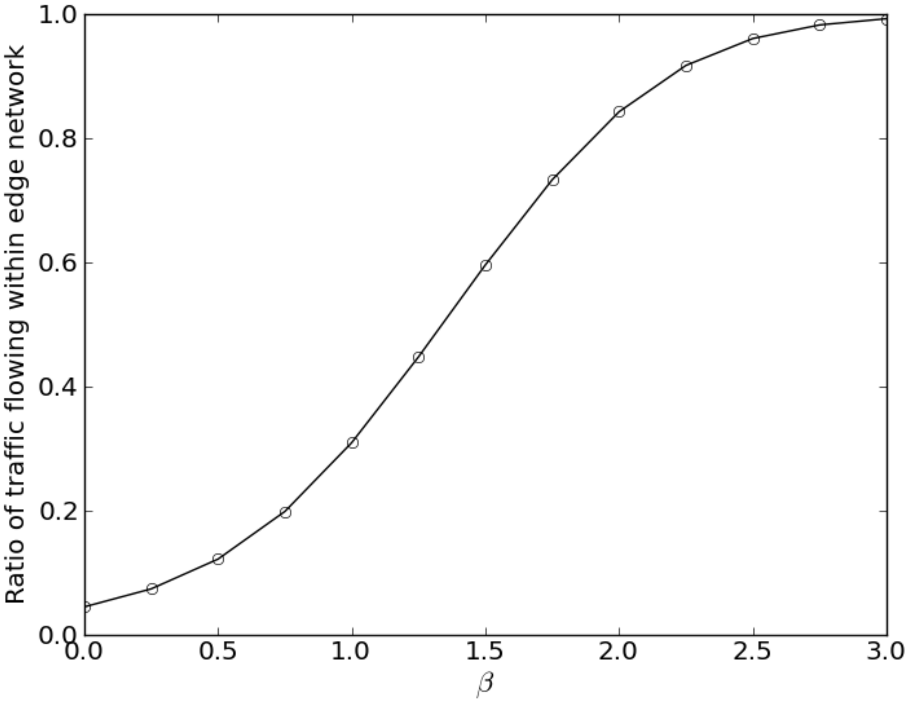


Figure 4.3: Ratio of traffic flowing within edge network as a function of β in Equation (4.14)

(C_c) [132] of the node connected to the data center ($c \in N$) as follows.

$$C_c = \left(\frac{\sum_{i \in N^e} (n_i + m_i) d_{ic}}{\sum_{i \in N^e} (n_i + m_i)} \right)^{-1} \quad (\forall c \in N), \tag{4.17}$$

where n_i and m_i are traffic volumes uploaded from node $i \in N^e$ and downloaded to it, respectively, and d_{ic} is the distance (i.e., hop count) between node $i \in N^e$ and node $c \in N$. A node that has higher C_c value is to receive data via shorter distance transmission than other nodes. In our network model, the 23th node (which corresponds to Aichi) has the maximum C_c of 0.175 and the 47th node (which corresponds to Okinawa) takes the minimum C_c of 0.084 (see Figure 4.2).

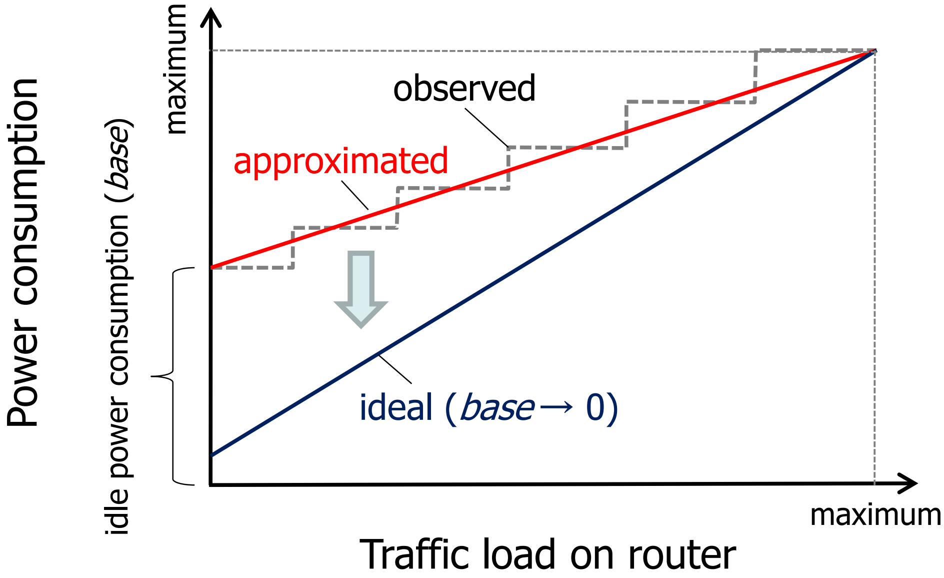


Figure 4.4: Router's energy proportionality [120, 134]

4.3.4 Router Power Consumption Model

To describe the power characteristics of a router, we consider steady-state traffic and energy-proportional behavior. As shown in Figure 4.4, the router's observed power consumption might be a step-increasing function of traffic load, in which the step size depends on the power consumption of the router's chassis, switching fabric, line cards, and active ports. This power profile (which is $b_i(g_i)$ $(i{\in}N)$ in Section 4.2.2) is approximated as $l_i g_i + base_i$, where l_i is a proportionality factor and $base_i$ is the baseline, i.e., power consumption while the router is idle. The baseline is as much as about 80 % of the maximum power consumption in the current router architecture [120]. However, we believe that the baseline will become smaller and that routers will come to have the ideal power proportional characteristics modeled in studies like [119, 120] in the future, as power saving techniques, e.g., [123, 133, 63, 134], are developed and applied.

Figure 4.5 plots the maximum switching capacity versus maximum power consumption for several series of core/edge router products; their specifications were gathered from the

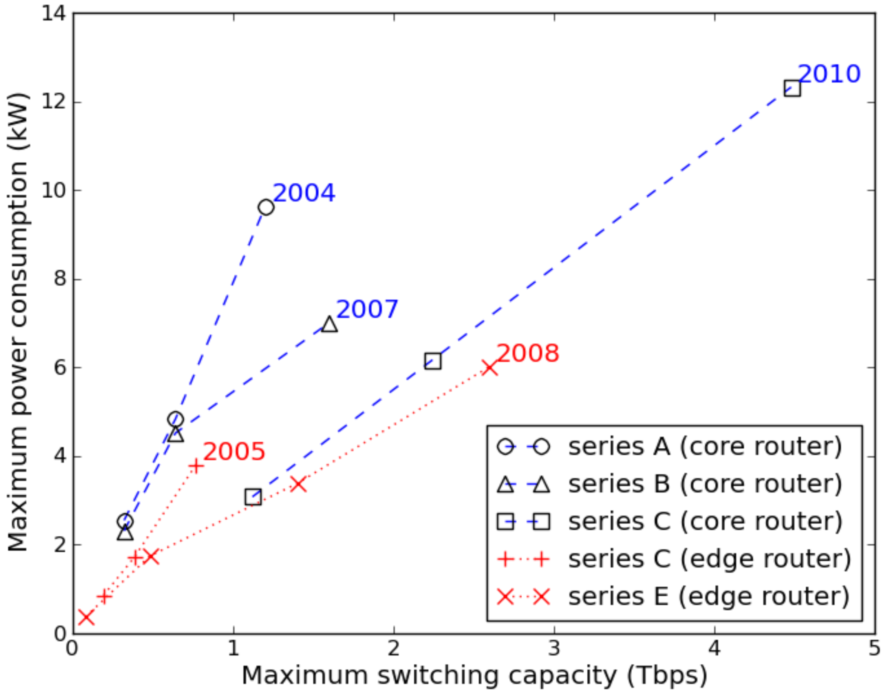


Figure 4.5: Maximum power consumption versus maximum switching capacity for several series of routers [135, 75, 136] (Four-digit numbers are series release years)

following web pages [135, 75, 136]. For routers within the same series, the ratio of router maximum power to maximum capacity is almost the same. We assume that the router's idle power is proportional to the maximum power consumption; this ratio is independent of $i \in N$ and denoted by R_{base}. As a result, the proportionality factor (l_i ($i \in N$)) is the same for all routers in a single series; moreover, the larger routers in the series consume more power at idle state. When a router has a large R_{base} value, the router consumes much power even if less traffic load is offered. Meanwhile, as R_{base} decreases, the router becomes to have perfect energy proportionality.

In addition, since we suppose that the NAVS is implemented by dedicated hardware [137], the power consumption of a NAVS could be close to that of a router. Hence, we assume that the NAVS's power profile ($c_i(g_i)$ ($i \in N$)) is same as the router's ($b_i(g_i)$ ($i \in N$)). Note that the power consumed by a node (i.e., a router) with a NAVS is handled separately as that consumed by a node and that consumed by a NAVS, according to Objective (4.1).

4.4 Evaluation

The DCN power consumption (E in Objective (4.1)) was evaluated by simulating the traffic load on each node of the network system described in Section 4.3. In this section, we first explain the evaluation method and then present detailed results.

4.4.1 Evaluation Method

To solve the combinatorial optimization problem described in Section 4.2.2 for such a large network, one needs to use an efficient heuristic. We chose to apply the greedy-interchange heuristic [129] to select the locations of P NAVSes minimizing E. Its algorithm is explained as follows.

step 1: The first node to be given a NAVS is found as follows. Starting from the state where all of the nodes do not yet have a NAVS attached, E is calculated when each node has it. For a given node, if NAVS attachment minimizes E, that node is selected as the first node to have the NAVS attached.

step 2: The next node to be given a NAVS is chosen as follows. E is calculated when a NAVS is added to each node that does not have it yet. For a given node, if NAVS attachment minimizes E, that node is selected next for NAVS attachment.

step 3: The NAVS tries to be moved to an adjacent node. E is calculated when the NAVS is moved to each adjacent node that does not yet have a NAVS. If E is smaller than before the NAVS was moved, the NAVS is moved to the adjacent node. This step is repeated until E becomes larger than NAVS movement, in which case the NAVS is not moved.

step 4: The number of NAVSes is counted. If the number is less than P, go to step 2.

Note that we neglect the first term of Objective (4.1) when calculating E because the WDM links of an IP-over-WDM network account for only around 10 % or less of the total power consumption [60, 56]. In addition, although the traffic route may be changed because of constraints on the available link bandwidth, this consideration is outside the scope of this chapter.

The following conditional settings were used for the evaluations conducted in the next subsection.

- Compression ratio of upload traffic volume at a NAVS (γ in Equation (4.11)) was set to a constant value of 0.1.

- The sum of total upload traffic ($\sum_{e \in N^e} n_e$) and total download traffic ($\sum_{e \in N^e} \sum_{e' \in N^e} m_{ee'}$) was set to the estimated volume of Internet traffic in Japan (2.6 Tbps (as of 2011)) [138].

- Routers of the latest series specified in Figure 4.5 were selected for nodes in the core and edge networks. Hence, "series C" specifications were used for core nodes and "series E" ones were used for edge nodes. Furthermore, the smallest router model (or the combination of smaller models) was chosen from the series; it satisfied the condition that the maximum possible traffic load on the router was less than 50 % of the router's maximum switching capacity regardless of the traffic route selected in the network.

- To avoid the impractical route from an edge node to the data center, the data center was conveniently supposed to provide the services of a NAVS. Hence, at step 1, the first NAVS was fixedly located in the data center and connected to the same node to which the data center server was also connected. Then, if the distance from an edge node to the data center was shorter than that from the edge node to the nearest NAVS outside the data center, the edge node could directly communicates the data center.

4.4.2 Evaluation Results

In Section 4.4.2 to Section 4.4.2, we present evaluation results for when NAVSes are distributed to only core nodes. In these sections, we repeated from step 1 to step 4 explained in the previous subsection for 46 times (i.e., in cases of $P = 2$ to $P = 47$) for one set of evaluations.

Overview of Power Optimization Results

The change in total power consumption (E) versus the number of NAVSes (P) is shown in Figure 4.6. Here, the traffic matrix parameters, α and β in Equations (4.14)–(4.16), were set to 1.0 and 2.0, respectively. The data center was connected to the 13th core node (see Figure 4.2); this node corresponds to Tokyo, which sent and received the largest volume in our network system. Each router's power consumption ratio at idle state (R_{base}) was set to 0.01. In addition, "$P = 1$" means that there was only the data center (containing the first NAVS) and no distributed NAVSes outside the data center. E with only the data center is denoted by E_1 here.

Overall, as more NAVSes were added, E decreased through a reduction in the amount of traffic; it was minimum (around 70 % of E_1) when P equals to 15. Then, it increased steadily with additional base power consumption of the NAVSes. We could say that the number and locations of NAVSes were optimized when E took the minimum. In what follows, the optimized E is denoted by E_{min} and the number of NAVSes providing E_{min} is denoted by P_{min}. In addition, E was composed of three portions: power consumed independently of traffic amount (denoted by the term "base") (Eb), that consumed for the upload traffic (Eu), and that consumed for the download traffic (Ed). Each portion was the sum of the power consumed by all nodes (not containing that consumed by NAVSes) and that consumed by all servers (i.e., P NAVSes and the data center server), which were related to the second and the third terms of Objective (4.1), respectively.

Most of the power consumed by all servers was that consumed by NAVSes. The power consumed by all NAVSes in Eu as well as that in Ed were constant and independent of P,

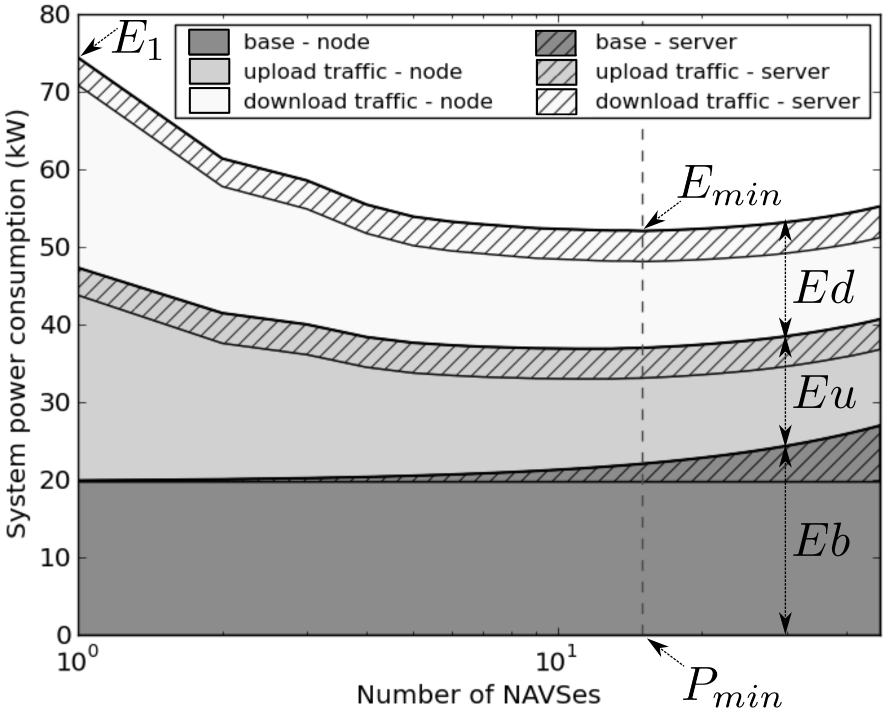


Figure 4.6: System power consumption (E) versus number of NAVSes (P)

because the total amount of upload traffic processed (and cached, as well) by all NAVSes was constant, and the total number of requests for downloading from all NAVSes remained unchanged regardless of P.

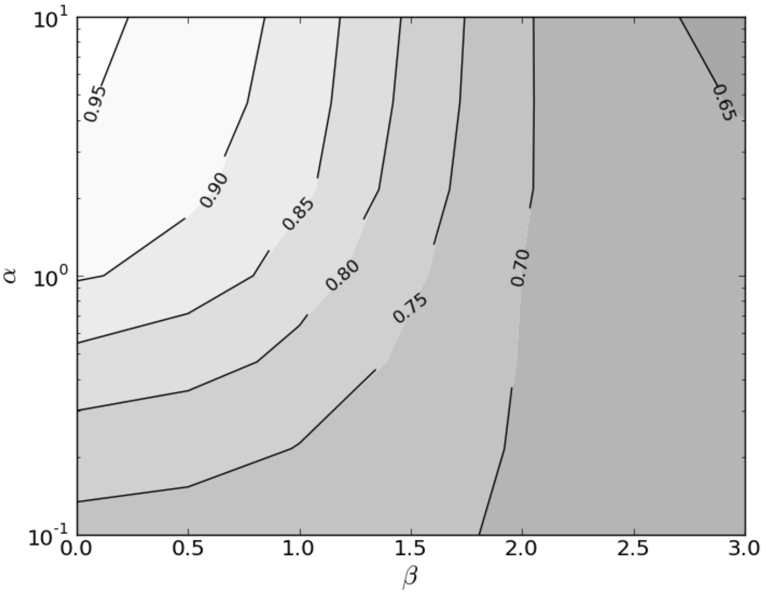
Meanwhile, as P was increased, the power consumed by all nodes in Eu decreased because the average hop count between an edge node and a corresponding NAVS became smaller. At this time, the power consumed by all nodes in Ed was affected by the trade-off between the hop count from an edge node to a NAVS and the cache amount at the NAVS. This was because, when a NAVS was added, each of all NAVSes covered fewer edge nodes, resulting in that less traffic was cached per NAVS and more traffic was downloaded from the data center server. The cache amount per NAVS, thus this trade-off as well, were dependent on the degree of traffic locality (β), which will be explained in the next subsection.

Effect of Traffic Locality

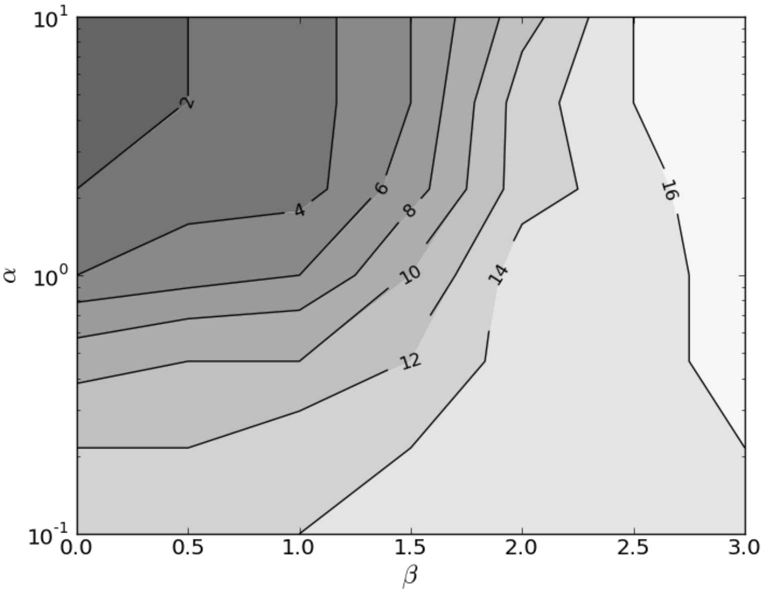
When both the ratio of download traffic to upload traffic and the ratio of traffic flowing within a local area to the total traffic become large, caching at a NAVS becomes more effective and the power consumed by the system thereby becomes smaller. We have therefore analyzed the optimized system power consumption (E_{min}) as a function of traffic matrix parameters (α and β), as shown in Figure 4.7. This figure indicates (a) the ratio of E_{min} to E_1 and (b) the number of NAVSes at E_{min} (P_{min}). Note that E_1 was hardly different for the values of α and β. As in the previous subsection, the data center was at the 13th core node and R_{base} was set to 0.01.

When α was set to nearly 0.1, most of the traffic was upload traffic from edge nodes to the data center server via NAVSes. Then, E was almost the sum of Eb and Eu. In this case, lower E_{min} (69 to 73 % of E_1) was achieved by deploying data compression in the NAVSes. Thus, E_{min} had little relation to β. In addition, when α was 0.1, P_{min} was approximately constant (14 to 16).

When α became large and approached 10, the contribution of Ed to E also became large and dominant. If β equaled to 0.0 in this case, the closeness between the node where the download traffic originated and the node downloading the traffic had no influence on the traffic amount exchanged between the two nodes. There was less downloaded traffic that originated at edge nodes in the same local area, so caching by the NAVSes made little contribution to the reduction in power consumption. As a result, E_{min} was 96 % of E_1 at the point (α,β)=(10,0.0). At this time, P_{min} was two. An extra NAVS resulted in increasing E. On the other hand, when β was 3.0, nearly 100 % of the download traffic originated from the local area. This led to the power reduction due to caching being very effective. Consequently, E_{min} was 64 % of E_1 at the point (α,β)=(10,3.0). In this case, P_{min} was maximum, for which there were 17 NAVSes.



(a) Ratio of optimized system power consumption (E_{min}) to system power consumption with only data center (E_1)



(b) Number of NAVSes at optimized system power consumption (P_{min})

Figure 4.7: Optimized system power consumption (E_{min}) and number of NAVes at E_{min} (P_{min}) as a function of traffic matrix parameters (α and β)

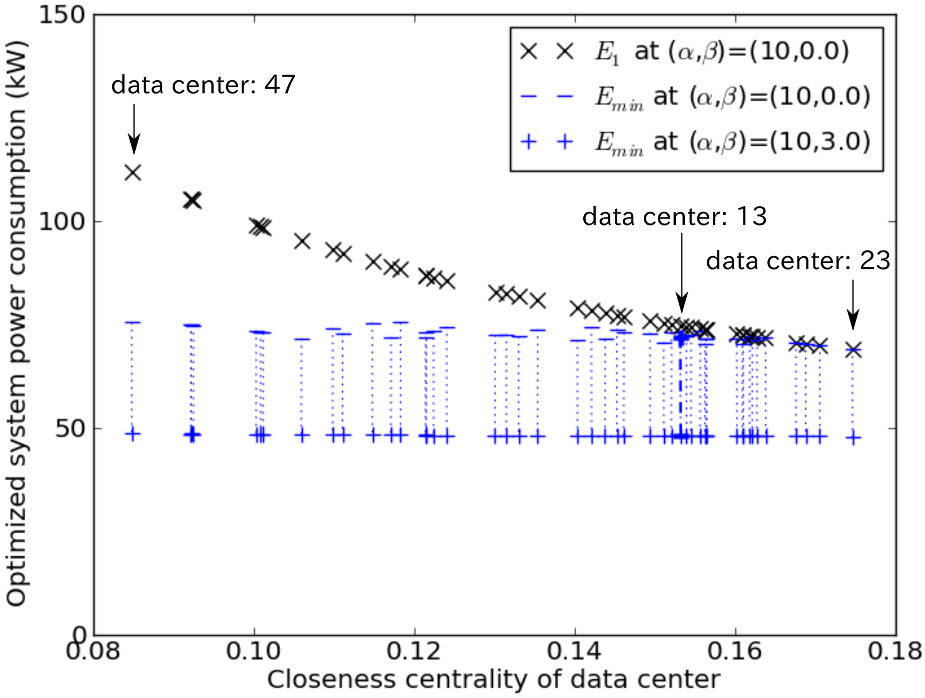


Figure 4.8: Change in optimized system power consumption (E_{min}) with data center's closeness centrality (C_c)

Effect of Data Center Location

In order to evaluate the effect of data center location, we analyzed the change in E_{min} with data center's closeness centrality (C_c in Equation (4.17)), as shown in Figure 4.8. Here, R_{base} was set to 0.01, as in the previous subsections. As shown in Figure 4.7(a), E_{min} was maximum at the point (α,β)=(10,0.0) and minimum at the point (α,β)=(10,3.0). We have therefore drawn a line connecting these two points for each result in Figure 4.8. Furthermore, we have added the point indicating E_1; this was indicated only for (α,β)=(10,0.0), because E_1 was hardly different for the values of α and β.

With only the data center, E_1 largely depended on C_c of the data center. In our network model, C_c was minimum when the data center was connected to the 47th node. Since this

47th node received traffic over a longer distance than others, E_1 in this case was maximum and 1.5 times larger than in the case where the data center was located at the 13th node.

On the other hand, when the NAVSes were distributed, the dependency on the data center location disappeared. When there was no traffic locality (i.e., (α,β)=(10,0.0)), E_{min} was almost the same as E_1 in the case where the data center's C_c was maximum, regardless of the data center location. In this case, P_{min} was two; this meant that the NAVS other than in the data center behaved as an alternative to the data center server located far from the center of the network topology. Furthermore, when there was strong traffic locality (i.e., (α,β)=(10,3.0)), E_{min} retained the minimum value regardless of where the data center was located. In this case, P_{min} was nearly constant at 16 or 17.

Effect of Energy Proportionality

Finally, we evaluated the effect of modeling a router's idle power consumption. Figure 4.9 shows how E_{min} as well as E_1 changed with the ratio of idle power consumption (R_{base}). In this figure, we show two cases for E_1: the data center's C_c was minimum (i.e., connected to the 47th core node) and approximately maximum (i.e., connected to the 13th core node). Moreover, we indicate two cases for E_{min}: E_{min} was maximum (i.e, (α,β)=(10,0.0)) and minimum (i.e., (α,β)=(10,3.0)). We omitted to show other results, because E_1 had little relevance to the values of α and β; furthermore, E_{min} was little dependent on the location of the data center, as mentioned in the previous subsections.

When each node had imperfect energy proportionality, i.e., a large R_{base} value, each node consumed a lot of base power even if there was less traffic load on it. This made the system power consumption very high. In this case, Eb accounted for a large portion of E. Since E was little dependent on both Eu and Ed, both E_1 and E_{min} were little influenced by the location of the data center or the values of α and β.

On the other hand, as each node had perfect energy proportionality, i.e., a small R_{base} value, the system power consumption became small. This was because each node consumed less base power and the traffic load was not so high for its capacity. In this case, the contribution of both Eu and Ed to E became relatively large. As a result, the effect of the

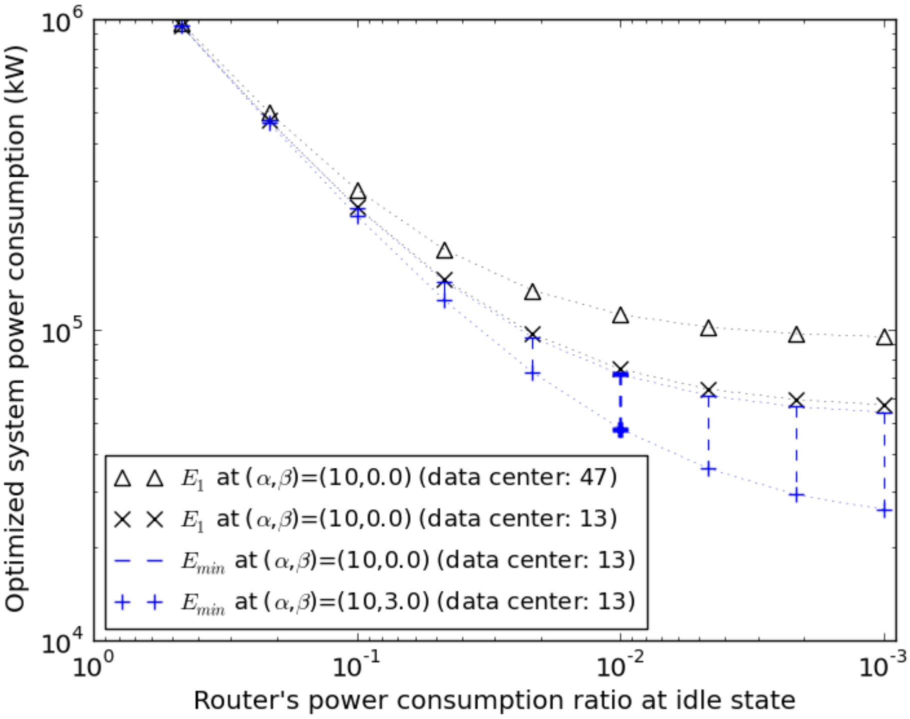


Figure 4.9: Optimized system power consumption (E_{min}) versus idle power consumption ratio (R_{base})

DCN (i.e., the difference between E_1 and E_{min}) and the influence of the data center location, α, and β also became large. When there was strong traffic locality (i.e., (α,β)=(10,3.0)), E_{min} approached 46 % of E_1 in the case where the data center was located at the 13th node; furthermore, E_{min} was 28 % of E_1 in the case where the data center was located at the 47th node.

Results for another application scenario

The DCN could have another application scenario in which a NAVS could communicates with other NAVSes as well. Consider the situation where a client requests data from a NAVS (called NAVS A) and the NAVS A does not have the data; this data is cached in another NAVS (called NAVS B) which lies closer to the NAVS A than the data center.

Then, the NAVS A does not download it from the data center server but from the NAVS B. We evaluated this case with the same settings in Section 4.4.2. The difference from Figure 4.7(a) was maximum at the point (α,β)=(1.0,0.5), although E_{min} at this point was reduced to 97 % of that in the case of Figure 4.7(a) for the given network topology.

We also conducted the situation where the NAVS could be attached to a node in the edge network. In this case, with the same settings in Section 4.4.2, the evaluation results showed that, when there was strong traffic locality (i.e., (α,β)=(10,3.0)), E_{min} in this case approached 93 % of that in the case of Figure 4.7(a). The power reduction was not so large because all edge nodes could reach the core network in a single hop for the given topology.

4.5 Conclusion

This chapter focused on the energy-saving aspect of the DCN, especially considering traffic locality. Through numerical evaluations of a Japan-wide network model, we revealed the following results. When most of the traffic is upload traffic to the data center, the power consumption of the DCN had little relation to traffic locality. On the other hand, as the download traffic becomes dominant, the dependence on traffic locality also becomes large. When there is strong traffic locality, the power consumed with the DCN is reduced up to about 30 % of that consumed with only the data center. We note that, although the NAVS's power model and specification affects the total power consumption (E), above results will not change as long as the power consumption of routers predominates in E. In this chapter, the DCN's topology was fixed. However, WAN power efficiency depends largely on the network topology. In the future, we would therefore like to investigate a power-efficient network topology. At that time, we will consider a network/system architecture across multiple data centers

Acknowledgment

This chapter was supported in part by “Research and Development on Secure Cloud Networking Technologies (Intelligent Distributed Processing Technologies)” of Ministry of Internal Affairs and Communications, Japan.

Chapter 5

Conclusion

As a WAN provides high-bandwidth connectivity, enterprises networks and applications executed over the WAN have been undergoing changes in their architectures through roughly three phases. In the first phase, after 2000, enterprises integrated their private WANs on the basis of IP technology, as well as computing resources used for applications in private data centers. In the second phase, after around 2006, enterprises consolidated their computing resources from branches to a private data center. In the third phase, now in progress, enterprises, as well as other social institutions, are outsourcing their computing resources to external public data centers. During these integration processes of enterprise networks and computing resources, several issues in the private and/or public WAN and its related applications arise. These issues are generally caused by WAN performance requirements increasing during the integration process. In this thesis, we therefore focused on evaluating such a situation and investigated solution approaches for improving it without enhancing the specifications of the WAN nodes.

For the first phase, we focused on evaluating the scalability of the control plane in a large enterprise network constructed by integrating two large banking networks. That is, we developed an approach to estimate the performance of the integrated network for updating routing information. Our approach is based on drawing up a formula to represent the effect of the increase in packet traffic on the decrease in CPU utilization at a router, and hence on delays in the updating of routing information. We experimented on the actual network and determined two guidelines. When the 5-minute average CPU utilization at a router is below 30 %, the routing information is updated within the warning time (144 seconds). Moreover, when the 5-minute average level of CPU utilization at a router is below 40 %, the routing information is updated within the time limit for errors (180 seconds). We confirmed that these guidelines were being maintained within the integrated network.

For the second phase, we focused on evaluating the performance of a thin-client system, which is a typical application traversing a private WAN after computing resources have been consolidated from branches to a private data center. The thin-client system has a particular traffic pattern involving a mixture of interactive data flows and bulk data flows. We aimed to improve the performance of interactive data flows while not influencing bulk data flows. Through simulation experiments using actual data traffic, we found that the average end-to-end delay of the interactive data flows could be reduced by applying priority queuing and utilizing the TCP SACK option. The occurrences of lengthy delays exceeding about 1 second resulting from the buffering delay in the server could be further reduced to about one-fourth by modifying the retransmission timeout calculation and temporarily halting the TCP SACK control. Nevertheless, to completely eliminate the effect of bulk data flows on interactive data flows, it is necessary to establish separate TCP connections for these two types of data flows.

For the third phase, we focused on evaluating the power consumed in a public WAN after computing resources have been consolidated in a few external huge data centers. Since this consolidation is becoming widespread among enterprises, public offices, homes, and so on, we considered this issue for the whole society. A DCN not only improves the response time to clients but also reduces the traffic over the WAN, thereby decreasing the power consumed in the WAN. We investigate such an energy-saving aspect of the DCN. The effectiveness of the DCN depends on the degree of traffic locality. Through numerical evaluations of a Japan-wide network model, we obtained the following results. When the data center is positioned near the center of the network topology, the DCN brings a small advantage in the case where the traffic has no locality. On the other hand, the power consumed with the DCN approaches about 50 % of that consumed with only the data center server when almost all of the traffic is created and consumed within the local area. Moreover, the advantage becomes up to about 30 % when the data center is located far from the center of the network topology.

After the third phase, computing resources will come to be extremely consolidated in huge public data centers. The challenges of such data centers include increasing reliability regardless of the huge scale, increasing server utilization, reducing power consumption and floor space, deploying new applications as soon as possible, and dynamically adapting to changing load conditions inside a data center and between data centers [139]. Moreover, related to these challenges, one of the most influential changes in end systems is server virtualization [17, 18], that is, deploying multiple virtual servers on the same physical server. As a result, networks in such data centers are being given the following new requirements. The first requirement is to scale in order to support a large number, e.g., incrementally up to a 100,000 [140], (virtual) servers. Each network node in the data center needs to reduce the number of entries in the forwarding table. The second one is to increase the upper limit on the number of sub-networks for isolating groups of servers, such as a VLAN allowing 4000 sub-networks. The third one is to enable virtual servers to migrate to any physical server while maintaining their IP addresses and reachability from clients. The last is to support multi-tenant environments, in which outsourced computing resources from many customers could have overlapping MAC/IP addresses.

To satisfy such new requirements, the networking industry has just proposed new standard technologies based on MAC-over-IP encapsulation [141, 142]. However, these proposals address only the basic idea such as frame format and have not being implemented in full-scale operation at actual data centers. Even if these proposals provide greater flexibility for locating virtual servers, we must still optimize the sub-network scheme and virtual servers locations as well as those of their associated data. At that time, it is important to consider not only the logical conditions of networks, such as end-to-end delay, throughput, bit error rate, and packet error rate, but also the physical or environmental conditions of data centers, such as geographical location, power supply and delivery channel, air conditioning design, fire suppression system, and local and national regulations. We believe that these considerations will contribute to the design and management of future cloud computing.

Bibliography

[1] P. Oppenheimer, *Top-Down Network Design.* Indianapolis:Cisco Press, 1998.

[2] Y. Ogawa, T. Hirata, K. Takamura, K. Yamaha, S. Saitou, K. Iwanaga, and T. Koita, "Estimating the performance of a large enterprise network for updating routing information," *IEICE TRANSACTIONS on Communications*, vol. E88-B, pp. 2054–2061, May 2005.

[3] I. Pepelnjak, "Market trends in service provider networks." `http://www.ipspace.net/Market_trends_in_Service_Provider_networks`, Oct. 2010.

[4] Y. Ogawa, A. Nakaya, E. Ohira, S. Hasegawa, and N. Ishii, "Development of network management database and network performance diagnosis system for a large-scale enterprise network," *IEICE TRANSACTIONS on Communications*, vol. J86-B, pp. 1278–1286, July 2003. (in Japanese).

[5] J. D. McCabe, *Network Analysis, Architecture and Design, Second Edition.* San Francisco:Morgan Kaufmann Publishers Inc., 2003.

[6] J. B. Association, "Changing banking industry." `http://www.zenginkyo.or.jp/en/banks/changing/`, Apr. 2009.

[7] L. DuBois, "The data protection imperative in the enterprise remote office," *White Paper of International Data Corporation*, June 2006.

[8] H. Biggar, B. Garrett, J. McKnight, and J. Gahm, "Branch office optimization: Data protection trends," *Research Report of the Enterprise Strategy Group, Inc.*, Jan. 2007.

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," *Technical Report of EECS Department, University of California, Berkeley*, Feb. 2009.

[10] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A first look at modern enterprise traffic," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC 2005)*, Oct. 2005.

[11] E. Karpilovsky, L. Breslau, A. Gerber, and S. Sen, "Multicast redux: a first look at enterprise multicast traffic," in *Proceedings of the 1st ACM workshop on Research on enterprise networking (WREN 2009)*, pp. 55–64, Aug. 2009.

[12] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: findings and implications," in *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems (SIGMETRICS 2009)*, pp. 37–48, June 2009.

[13] A. Myers, T. E. Ng, and H. Zhang, "Rethinking the service model: Scaling ethernet to a million nodes," in *Proceedings of the 3rd ACM SIGCOMM workshop on on Hot Topics in Networks (HotNets 2004)*, Nov. 2004.

[14] K. Elmeleegy and A. Cox, "Etherproxy: Scaling ethernet by suppressing broadcast traffic," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM 2009)*, pp. 1584–1592, Apr. 2009.

[15] K. Elmeleegy, A. L. Cox, and T. S. E. Ng, "Understanding and mitigating the effects of count to infinity in ethernet networks," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 186–199, Feb. 2009.

[16] M. Scott, A. Moore, and J. Crowcroft, "Addressing the scalability of ethernet with moose," in *Proceedings of 1st ITC 21 Workshop on Data Center - Converged and Virtual Ethernet Switching (DC CAVES)*, Sept. 2009.

[17] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable fault-tolerant layer 2 data center network fabric," in *Proceedings of the 2009 ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2009)*, pp. 39–50, Aug. 2009.

[18] C. Kim, M. Caesar, and J. Rexford, "SEATTLE: A scalable ethernet architecture for large enterprises," *ACM Transactions on Computer Systems*, vol. 29, pp. 1–35, Feb. 2011.

[19] J. Touch and R. Perlman, "Transparent interconnection of lots of links (TRILL): Problem and applicability statement." `http://tools.ietf.org/html/rfc5556`, May 2009.

[20] D. Allan, P. Ashwood-Smith, N. Bragg, J. Farkas, D. Fedyk, M. Ouellete, M. Seaman, and P. Unbehagen, "Shortest path bridging: Efficient control of larger ethernet networks," *IEEE Communications Magazine*, vol. 48, pp. 128–135, Oct. 2010.

[21] R. Sanchez, L. Raptis, and K. Vaxevanakis, "Ethernet as a carrier grade technology: developments and innovations," *IEEE Communications Magazine*, vol. 46, pp. 88–94, Sept. 2008.

[22] S. Salam and A. Sajassi, "Provider backbone bridging and mpls: complementary technologies for next-generation carrier ethernet transport," *IEEE Communications Magazine*, vol. 46, pp. 77–83, Mar. 2008.

[23] L. Caro, D. Papadimitriou, and J. Marzo, "Improving label space usage for ethernet label switched paths," in *Proceedings of IEEE International Conference on Communications (ICC 2008)*, pp. 5685–5691, May 2008.

[24] L. Caro, J. Marzo, and D. Papadimitriou, "Carrier ethernet "label" scalability," in *Proceedings of the 13th International Telecommunications Network Strategy and Planning Symposium (Networks 2008)*, pp. 1–15, Oct. 2008.

[25] Y.-W. E. Sung, S. G. Rao, G. G. Xie, and D. A. Maltz, “Towards systematic design of enterprise networks,” in *Proceedings of the 4th international conference on Emerging networking experiments and technologies (CoNEXT 2008)*, pp. 22:1–22:12, Dec. 2008.

[26] X. Sun and S. Rao, “A cost-benefit framework for judicious enterprise network redesign,” in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM 2011)*, pp. 221–225, Apr. 2011.

[27] C. Paquet and D. Teare, *Building scalable Cisco networks.* Indianapolis:Cisco Press, 2000.

[28] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, “Delayed internet routing convergence,” in *Proceedings of the 2000 ACM conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2000)*, pp. 175–187, Aug. 2000.

[29] R. Mahajan, D. Wetherall, and T. Anderson, “Understanding bgp misconfiguration,” in *Proceedings of the 2002 ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2002)*, pp. 3–16, Aug. 2002.

[30] D. A. Maltz, G. Xie, J. Zhan, H. Zhang, G. Hjálmtýsson, and A. Greenberg, “Routing design in operational networks: a look from the inside,” in *Proceedings of the 2004 ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2004)*, pp. 27–40, Aug. 2004.

[31] F. Le, G. G. Xie, D. Pei, J. Wang, and H. Zhang, “Shedding light on the glue logic of the internet routing architecture,” in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication (SIGCOMM 2008)*, pp. 39–50, Aug. 2008.

[32] M. Casado, T. Garfinkel, M. Freedman, A. Akella, D. Boneh, N. McKeowon, and S. Shenker, “SANE: A Protection Architecture for Enterprise Networks,” in *Proceedings of USENIX Security Symposium*, Aug. 2006.

[33] H. Ballani and P. Francis, "Conman: a step towards network manageability," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2007)*, pp. 205–216, Aug. 2007.

[34] R. Alimi, Y. Wang, and Y. R. Yang, "Shadow configuration as a network management primitive," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication (SIGCOMM 2008)*, pp. 111–122, Aug. 2008.

[35] T. Benson, A. Akella, and D. Maltz, "Unraveling the complexity of network management," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI 2009)*, pp. 335–348, Apr. 2009.

[36] W. Enck, T. Moyer, P. McDaniel, S. Sen, P. Sebos, S. Spoerel, A. Greenberg, Y.-W. E. Sung, S. Rao, and W. Aiello, "Configuration management at massive scale: system design and experience," *Selected Areas in Communications, IEEE Journal on*, vol. 27, pp. 323 –335, Apr. 2009.

[37] G. Hasegawa and M. Murata, "Research trends on TCP congestion control mechanisms," *IEICE TRANSACTIONS on Communications*, vol. J94-B, pp. 663–672, May 2011. (in Japanese).

[38] S. Floyd, "Highspeed TCP for large congestion windows." `http://tools.ietf.org/html/rfc3649`, Dec. 2003.

[39] Z. Zhang, G. Hasegawa, and M. Murata, "Performance analysis and improvement of HighSpeed TCP with TailDrop/RED routers," *IEICE TRANSACTIONS on Communications*, vol. E88-B, pp. 2495–2507, June 2005.

[40] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, pp. 83–91, 2002.

[41] L. S. Brakmo and L. L. Peterson, "TCP vegas: End to end congestion avoidance on a global internet," *IEEE Journal on selected Areas in communications*, vol. 13, pp. 1465–1480, Oct. 1995.

[42] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 1246–1259, Dec. 2006.

[43] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, pp. 1–12, Apr. 2006.

[44] H. Shimonishi, T. Hama, and T. Murase, "TCP-Adaptive reno for improving efficiency-friendliness tradeoffs of TCP congestion control algorithm," in *Proceedings of 4th International Workshop on Protocols for Future, Large-Scale & Diverse Network Transports (PFLDNeT 2006)*, Feb. 2006.

[45] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proceedings of 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, vol. 4, pp. 2514–2524, Mar. 2004.

[46] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, pp. 64–74, July 2008.

[47] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP westwood: end-to-end congestion control for wired/wireless networks," *Wireless Networks*, vol. 8, pp. 467–479, Sept. 2002.

[48] L. Guo and I. Matta, "The war between mice and elephants," *Technical Report of CS Department, Boston University*, May 2001.

[49] X. Chen and J. Heidemann, "Preferential treatment for short flows to reduce web latency," *Computer Networks*, vol. 41, pp. 779–794, Apr. 2003.

[50] K. Tokuda, G. Hasegawa, and M. Murata, "Analysis and improvement of the fairness between long-lived and short-lived TCP connections," in *Proceedings of 7th*

IFIP/IEEE Workshop on Protocols For High-Speed Networks (PfHSN 2002), pp. 33–40, Apr. 2002.

[51] Cisco Systems, Inc., "Visual networking index." `http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html`, June 2011.

[52] A. Gerber and R. Doverspike, "Traffic types and growth in backbone networks," in *Proceedings of the Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC 2011)*, pp. 1–3, Mar. 2011.

[53] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Communications Surveys Tutorials*, vol. 13, pp. 223–244, May 2011.

[54] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy efficiency in telecom optical networks," *IEEE Communications Surveys Tutorials*, vol. 12, pp. 441–458, Nov. 2010.

[55] H. Mellah and B. Sanso, "Review of facts, data and proposals for a greener internet," in *Proceedings of 6th International Conference on Broadband Communications, Networks, and Systems (BROADNETS 2009)*, pp. 1–5, Sept. 2009.

[56] W. Vereecken, W. Van Heddeghem, M. Deruyck, B. Puype, B. Lannoo, W. Joseph, D. Colle, L. Martens, and P. Demeester, "Power consumption in telecommunication networks: overview and reduction strategies," *IEEE Communications Magazine*, vol. 49, pp. 62–69, June 2011.

[57] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proceedings of IEEE International Conference on Communications (ICC 2009)*, pp. 1–6, June 2009.

[58] B. Sanso and H. Mellah, “On reliability, performance and internet power consumption,” in *Proceedings of 7th International Workshop on Design of Reliable Communication Networks (DRCN 2009)*, pp. 259–264, Oct. 2009.

[59] M. Gupta and S. Singh, “Greening of the internet,” in *Proceedings of the 2003 ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2003)*, pp. 19–26, Aug. 2003.

[60] G. Shen and R. Tucker, “Energy-minimized design for IP over WDM networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 1, pp. 176–186, June 2009.

[61] W. Van Heddeghem, M. De Groote, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester, “Energy-efficiency in telecommunications networks: Link-by-link versus end-to-end grooming,” in *Proceedings of 14th Conference on Optical Network Design and Modeling (ONDM 2010)*, pp. 1–6, Feb. 2010.

[62] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, “Reducing the energy consumption of ethernet with adaptive link rate (ALR),” *IEEE Transactions on Computers*, vol. 57, pp. 448–461, Apr. 2008.

[63] IEEE, “IEEE P802.3az energy efficient ethernet task force.” `http://www.ieee802.org/3/az/index.html`, June 2011.

[64] Y. Ogawa, A. Nakaya, K. Takamura, K. Yamaha, S. Saitou, K. Iwanaga, and T. Koita, “Estimating the performance of a large enterprise network for the updating of routing information,” in *Proceedings of IEEE Workshop on IP Operations and Management (IPOM 2002)*, pp. 161–165, Oct. 2002.

[65] Y. Ogawa, G. Hasegawa, and M. Murata, “Transport-layer optimization for thin-client systems,” in *Proceedings of IEEE Workshop on Communications Quality and Reliability (CQR 2007)*, May 2007.

[66] Y. Ogawa, G. Hasegawa, and M. Murata, “Delay analysis and transport-layer optimization for improving performance of thin-client traffic,” *Technical Report of IEICE*, vol. IN2008-56, pp. 75–80, Sept. 2008. (in Japanese).

[67] Y. Ogawa, G. Hasegawa, and M. Murata, “A transport layer approach for improving interactive user experience on thin clients,” in *Proceedings of Australasian Telecommunication Networks and Applications Conference (ATNAC 2009)*, Nov. 2009.

[68] Y. Ogawa, G. Hasegawa, and M. Murata, “A transport-layer approach for improving thin-client performance in a WAN environment,” *International Journal of Internet Protocol Technology*, vol. 6, pp. 172–183, Nov. 2011.

[69] Y. Ogawa, G. Hasegawa, M. Murata, and S. Nishimura, “Performance evaluation of distributed computing environment considering power consumption,” *Technical Report of IEICE*, vol. IN2009-172, pp. 169–174, Mar. 2010. (in Japanese).

[70] Y. Ogawa, G. Hasegawa, and M. Murata, “Effect of traffic locality on power consumption of distributed computing network,” in *Proceedings of 9th International Conference on Communications (COMM 2012)*, June 2012. [submitted for publication].

[71] Y. Ogawa, G. Hasegawa, and M. Murata, “Power consumption evaluation of distributed computing network considering traffic locality,” *IEICE TRANSACTIONS on Communications*, 2012. [submitted for publication].

[72] Y. Ogawa, A. Nakaya, E. Ohira, S. Hasegawa, and N. Ishii, “Development of a network management database and a network performance diagnosis system for a large-scale enterprise network,” *Technical Report of IEICE*, vol. TM2002-25, pp. 19–24, July 2002. (in Japanese).

[73] J. Doyle, *Routing TCP/IP Volume.* Indianapolis:Cisco Press, 1988.

[74] J. Garcia-Lunes-Aceves, “Loop-free routing using diffusing computations,” *IEEE/ACM Transactions on Networking*, vol. 1, pp. 130–141, Feb. 1993.

[75] "Cisco Systems, Inc.." `http://www.cisco.com/`, June 2011.

[76] I. Pepelnjak, *EIGRP Network Design Solutions.* Indianapolis:Cisco Press, 1999.

[77] K. McCloghrie and M. T. Rose, "Management information base for network management of TCP/IP-based internets:MIB-II." RFC 1213, Mar. 1991.

[78] P. L. D. Maggiora, C. E. Elliott, J. M. Thompson, R. L. P. Jr., and K. J. Phelps, *Performance and Fault Management.* Indianapolis:Cisco Press, 2000.

[79] A. Drago, A. Garcia, and M. Monteiro, "A methodology for performance management of networks," in *Proceedings of 25th Annual IEEE Conference on Local Computer Networks (LCN 2000)*, pp. 442–451, Nov. 2000.

[80] N. Sagawa and K. Koda, "Toward human-oriented office," *Hitachi Review*, vol. 58, pp. 169–173, Sept. 2009.

[81] R. W. Scheifler and J. Gettys, *X Window System: Core and Extension Protocols : X Version 11, Releases 6 and 6.1.* Oxford:Butterworth-Heinemann, 1997.

[82] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual Network Computing," *IEEE Internet Computing*, vol. 2, pp. 33–38, Jan. 1998.

[83] MSDN®, "Remote Desktop Protocol." `http://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx`, Mar. 2011.

[84] J. Nagle, "Congestion control in IP/TCP internetworks." RFC 896, Jan. 1984.

[85] R. Braden, "Requirements for Internet Hosts - Communication Layers." RFC 1122, Oct. 1989.

[86] G. Minshall, Y. Saito, J. C. Mogul, and B. Verghese, "Application performance pitfalls and TCP's Nagle algorithm," in *Proceedings of 2nd Workshop on Internet Server Performance (WISP '99)*, pp. 36–44, May 1999.

[87] J. C. Mogul and G. Minshall, "Rethinking the TCP Nagle algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 6–20, Jan. 2001.

[88] W. R. Stevens, *TCP/IP illustrated (vol. 1): the protocols*, ch. 19.4 Nagle Algorithm, pp. 267–273. Boston:Addison-Wesley, 1994.

[89] V. Jacobson and M. J. Karels, "Congestion avoidance and control," in *Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM '88)*, pp. 314–329, Aug. 1988. slightly-revised 1992 version of the 1988 paper.

[90] J. Heidemann, "Performance interactions between P-HTTP and TCP implementations," *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 65–73, Apr. 1997.

[91] V. Visweswaraiah and J. Heidemannk, "Improving restart of idle TCP connections," *Technical Report of University of Southern California*, Nov. 1997.

[92] J. Nieh, S. J. Yang, and N. Novik, "Measuring thin-client performance using slow-motion benchmarking," *ACM Transactions on Computer Systems*, vol. 21, pp. 87–115, Feb. 2003.

[93] D. Schlosser, A. Binzenhofer, and B. Staehle, "Performance comparison of windows-based thin-client architectures," in *Proceedings of Australasian Telecommunication Networks and Applications Conference (ATNAC 2007)*, pp. 197–202, Dec. 2007.

[94] A. M. Lai and J. Nieh, "On the performance of wide-area thin-client computing," *ACM Transactions on Computer Systems*, vol. 24, pp. 175–209, May 2006.

[95] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," *IEEE Computer*, vol. 39, pp. 46–52, Mar. 2006.

[96] Wireshark®, "wireshark®." `http://www.wireshark.org/`, Mar. 2011.

[97] Microsoft Corporation, "windows® XP home page." `http://www.microsoft.com/windows/windows-XP/`, June 2008.

[98] Hitachi, Ltd., "Secure client solution : Point-to-blade system." `http://www.hitachi.co.jp/products/harmonious/center/gl/main/demonstration/blade.html`, Sept. 2009.

[99] ns-2, "The Network Simulator - ns-2." `http://nsnam.isi.edu/nsnam/index.php/Main_Page`, Mar. 2011.

[100] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options." RFC 2018, Oct. 1996.

[101] W. R. Stevens and G. R. Wright, *TCP/IP illustrated (vol. 2): the implementation*, ch. 25. TCP Timers, pp. 817–850. Boston:Addison Wesley, 1995.

[102] D. Lin and H. Kung, "TCP fast recovery strategies: Analysis and improvements," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '98)*, vol. 1, pp. 263–271, Mar. 1998.

[103] B. Kim, Y. Kim, M. Oh, and J. Choi, "Microscopic behaviors of TCP loss recovery using lost retransmission detection," in *Proceedings of Consumer Communications and Networking Conference (CCNC 2005)*, pp. 296–301, Jan. 2005.

[104] K. Yamanegi, T. Hama, G. Hasegawa, M. Murata, H. Shimonishi, and T. Murase, "Implementation experiments of the TCP proxy mechanism," in *Proceedings of 6th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2005)*, pp. 17–22, Nov. 2005.

[105] S. Yang and T. T. Tiow, "Improving interactive experience of thin client computing by reducing data spikes," in *Proceedings of ACIS International Conference on Computer and Information Science (ICIS 2007)*, pp. 627–632, July 2007.

[106] B. Vankeirsbilck, P. Simoens, J. De Wachter, L. Deboosere, F. De Turck, B. Dhoedt, and P. Demeester, "Bandwidth optimization for mobile thin client computing through graphical update caching," in *Proceedings of Australasian Telecommunication Networks and Applications Conference (ATNAC 2008)*, pp. 385–390, Dec. 2008.

[107] J. Gantz, "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," *White Paper of International Data Corporation*, Mar. 2008.

[108] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks." Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia., Feb. 2007.

[109] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Computer Communication Review*, vol. 39, pp. 68–73, Dec. 2008.

[110] H. Wang, J. Liu, B. Chen, K. Xu, and Z. Ma, "On tracker selection for peer-to-peer traffic locality," in *Proceedings IEEE Tenth International Conference on Peer-to-Peer Computing (P2P 2010)*, pp. 1–10, Aug. 2010.

[111] Y. Liu, L. Guo, F. Li, and S. Chen, "A case study of traffic locality in internet P2P live streaming systems," in *Proceedings of 29th IEEE International Conference on Distributed Computing Systems (ICDCS 2009)*, pp. 423–432, June 2009.

[112] C. Tian, X. Liu, H. Jiang, W. Liu, and Y. Wang, "Improving bittorrent traffic performance by exploiting geographic locality," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2008)*, pp. 1–5, Dec. 2008.

[113] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving traffic locality in bittorrent via biased neighbor selection," in *Proceedings of 26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006)*, July 2006.

[114] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should internet service providers fear peer-assisted content distribution?," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC 2005)*, Oct. 2005.

[115] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson, *The future internet*, ch. Smart cities at the forefront of the future internet, pp. 447–462. Berlin:Springer-Verlag, 2011.

[116] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM 2008)*, pp. 457–465, Apr. 2008.

[117] H. Imaizumi, T. Nagata, G. Kunito, K. Yamazaki, and H. Morikawa, "Power saving technique based on simple moving average for multi-channel ethernet," in *Proceedings of 14th OptoElectronics and Communications Conference (OECC 2009)*, pp. 1–2, July 2009.

[118] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of the 1st ACM SIGCOMM workshop on Green networking (Green Networking 2010)*, pp. 29–34, Aug. 2010.

[119] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, pp. 33–37, Dec. 2007.

[120] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proceedings of the 8th International IFIP-TC 6 Networking Conference (NETWORKING 2009)*, pp. 795–808, May 2009.

[121] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, pp. 149–167, Jan. 2011.

[122] F. M. Ramos, R. J. Gibbens, F. Song, P. Rodriguez, J. Crowcroft, and I. H. White, "Reducing energy consumption in IPTV networks by selective pre-joining of channels," in *Proceedings of the 1st ACM SIGCOMM workshop on Green networking (Green Networking 2010)*, pp. 47–52, Aug. 2010.

[123] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2008)*, pp. 323–336, Apr. 2008.

[124] K. C. Guan, G. Atkinson, D. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *Proceedings of the 4th International Workshop on Green Communications (GreenComm 2004)*, GreenComm4, June 2011.

[125] U. Lee, I. Rimac, and V. Hilt, "Greening the internet with content-centric networking," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy 2010)*, pp. 179–182, Apr. 2010.

[126] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F.-J. Westphal, "Energy trade-offs among content delivery architectures," in *Proceedings of 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE 2010)*, pp. 1–6, June 2010.

[127] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT 2009)*, pp. 37–48, Dec. 2009.

[128] C. Francalanci, P. Giacomazzi, and A. Poli, "Cost-performance optimization of application- and context-aware distributed infrastructures," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 39, pp. 1200–1213, Nov. 2009.

[129] K. Hidaka and H. Okano, "Simulation-based approach to the warehouse location problem for a large-scale real instance," in *Proceedings of the 29th conference on Winter simulation (WSC '97)*, pp. 1214–1221, Dec. 1997.

[130] J. Baliga, R. Ayre, W. Sorin, K. Hinton, and R. Tucker, "Energy consumption in access networks," in *Proceedings of Conference on Optical Fiber communication/National Fiber Optic Engineers Conference (OFC/NFOEC 2008)*, pp. 1–3, Feb. 2008.

[131] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS 2003)*, pp. 206–217, June 2003.

[132] V. Latora and M. Marchiori, "A measure of centrality based on the network efficiency," *eprint arXiv:cond-mat/0402050*, Feb. 2004.

[133] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proceedings of the 2008 conference on Power aware computing and systems (HotPower 2008)*, pp. 7–7, Dec. 2008.

[134] M. Yamada, T. Yazaki, N. Matsuyama, and T. Hayashi, "Power efficient approach and performance control for routers," in *Proceedings of IEEE International Conference on Communications (ICC 2009)*, pp. 1–5, June 2009.

[135] "ALAXALA Networks Corporation." `http://www.alaxala.com/`, June 2011.

[136] "Juniper Networks, Inc.." `http://www.juniper.net/`, June 2011.

[137] M. Hiraiwa, H. Masukawa, and S. Nishimura, "Hitachi's involvement in networking for cloud computing," *Hitachi Review*, vol. 59, pp. 206–212, Dec. 2010.

[138] Ministry of Internal Affairs and Communications, "Estimate of Internet Traffic in Japan." `http://www.soumu.go.jp/main_content/000109282.pdf`, Mar. 2011. (in Japanese).

[139] I. Pepelnjak, "Data center 3.0 for networking engineers." `http://www.ipspace.net/Data_Center_3.0_for_Networking_Engineers`, June 2011.

[140] A. Vahdat, M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan, "Scale-out networking in the data center," *IEEE Micro*, vol. 30, pp. 29–41, July 2010.

[141] M. Mahalingam, D. G. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks." `http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-00`, Aug. 2011.

[142] M. Sridharan, K. Duda, I. Ganga, A. Greenberg, G. Lin, M. Pearson, P. Thaler, C. Tumuluri, N. Venkataramiah, and Y.-S. Wang, "NVGRE: Network virtualization using generic routing encapsulation." `http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-00`, Sept. 2011.