A Thesis for the Degree of Ph.D. in Engineering

Efficient data transport technologies for next generation backbone network

February 2010

Graduate School of Science and Technology
Keio University

Sho SHIMIZU

Contents

Lists of Figures

Lists of Tables

Summary

The backbone network traffic generated by the Internet continues to grow rapidly due to the emergence of many new broadband access services such as VoIP, P2P, video sharing like YouTube, and grid computing. WDM technologies can allow the backbone network to support the explosive growth in bandwidth demands. However, many of the new applications require different communication topologies and multi-layer operation, such as WDM, TDM, and packet layer. To fully support these and yet-to-be introduced services, new highly efficient communication network schemes are needed. This dissertation identifies efficient data transport technologies for the next generation backbone networks.

There are four requirements for next generation backbone network to handle rapidly increasing traffic volume and support emerging applications: 1) high speed and large capacity, 2) scalability, 3) traffic engineering capability, and 4) application frameworks for large data distribution. Topics to satisfy these requirements are investigated in Chapter 3 – 6. This dissertation is organized as follows.

Chapter 1 describes the background of this dissertation and clarifies its purpose and position.

Chapter 2 illustrates fundamental technologies for next generation backbone network and previous works related to the above requirements.

Chapter 3 focuses on high speed and large capacity transport. A novel wavelength assignment scheme for a wavelength-routed network with wavelength converters of limited range is proposed. It reduces the total number of wavelength conversions needed and the

number of wavelength converters. Consequently, it makes wavelength-routed networks cost effective.

Chapter 4 focuses on the scalability issue of the next-generation layer-2 network. The VLAN tag-swap-based wide area layer-2 networks architecture is proposed for the scalable next generation layer 2 network in this chapter.

The computational complexity of path calculation in traffic engineering is the focus of Chapter 5. The new approach of parallel shortest path search is proposed to realize sophisticated traffic engineering. The proposed approach uses dynamically reconfigurable processors (DRPs), and takes full advantage of their parallelism.

Chapter 6 focuses on an application framework for large data distribution, and introduces a new solution to the replica placement problem that is found in content delivery networks. The solution, which takes the form of application level technology, is specifically designed to achieve the efficient distribution of large volume contents. The proposed replica placement solution can generate all replica placement patterns at extremely high rates due to DRP parallelism. Optimal replica placement, which means the minimum number of replicas, can be obtained within reasonable time. The proposal is expected to trigger the emergence of exciting new cost-effective services based on large volume content distribution.

Chapter 7 draws this dissertation to its conclusion with a useful summary of the advances raised herein.

Chapter 1

Introduction

1.1 Background

The development of the Internet has changed our daily lives drastically. The Internet originates from the research project of the Advanced Research Project Agency (ARPA), which is an agency of the United States Department of Defense. The research project was called ARPANET at that time. In 1990s, the commercial use of the Internet was permitted, then many commercial services have been emerged. A typical commercial service in the early age of the Internet is connecting to the Internet. The service for home and business users began, and Internet Service Providers (ISPs) appeared.

The first breakthrough in terms of the development of the Internet was the invention of the World Wide Web (WWW). The WWW was invented in 1990 by Tim Bernes-Lee, who was working at the European Organization for Nuclear Research (CERN). The first web browser and the web server were developed on the NeXT STEP platform. After that, the WWW became a popular application on the Internet. One of the advantages of the WWW is hypertext structure, which links some information with related information. The WWW enabled us to retrieve a lot of information through the Internet easily, and also to distribute information by ourselves at low cost.

As the development of the WWW, the size of the contents on the web has been increasing. Originally, the contents were text-based information, which sometimes includes small (low resolution) still images. The access speed around 1995 was typically limited

to 64kbps since the normal way to access to the Internet was dial-up access. As a result, the total size of a web page was several or tens of kilo bytes. From the late 1990s to the early 2000s, several broadband access service like Asymmetric Digital Subscriber Line (ADSL) and Fiber To The Home (FTTH) has started. Currently, these broadband access network is in majority. Especially in Japan, the most popular broadband access is FTTH [1]. The total number of subscribers of FTTH is increasing while that of ADSL is decreasing. Therefore, a web page includes large volume contents such as large (high resolution) still images, sounds, and moving images.

Figure 1.1 shows the traffic volume of the backbone network observed at Internet eXchange (IX) of Japan Internet eXchange (JPIX) [2]. JPIX is the first commercial IX in Japan, which is founded by major network companies in 1997. The backbone Internet traffic in Japan has been increasing due to the development of broadband access technologies. Currently, the minimum traffic volume observed at JPIX reaches 50Gbps, and the maximum traffic volume is over 150Gbps. How to manage the increasing traffic is a big issue in backbone networks. From this perspective, next generation backbone network has to provide high speed and large capacity network.

The number of the Internet users has been increasing for about last 10 years since the price to connect to the Internet has been decreasing and the access speed has been growing year by year. In addition, in the era of ubiquitous computing, not only personal computers but also other devices such as mobile phones, personal digital assistants (PDAs), home appliances, sensors installing in home, will be connected to the Internet. IPv6 has been developed to expand the address space of IP. 128 bits are assigned for the address field in IPv6. It means the limitation of the number of addresses is practically removed since 2^{128} addresses which is 3.7×10^{28} times larger than the estimated total population of the earth in 2050 can be utilized. To handle a number of devices accessing to the Internet, scalability is one of key issues in next generation backbone network.

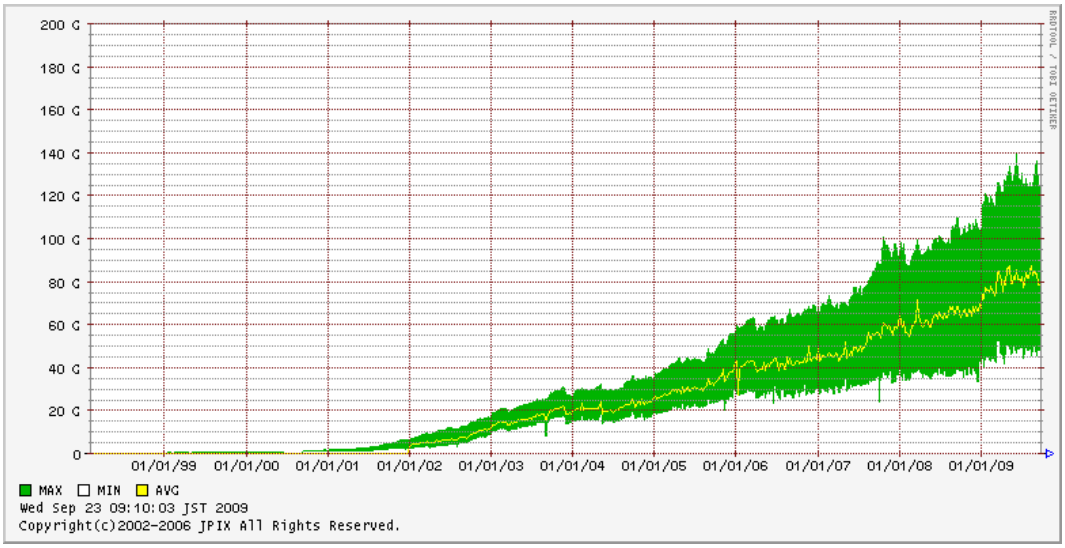


Figure 1.1: Traffic volume observed at JPIX

A lot of new applications have been emerged with the development of the Internet. The typical use cases of the Internet are sending and receiving e-mails, browsing across web pages, and transferring files. Transmitting multimedia data such as music and video contents through the Internet became possible due to increase of the bandwidth of access and backbone networks. Voice over IP (VoIP) and Video on Demand (VoD) are examples of multimedia service. Peer-to-Peer (P2P) applications such as WinMX, Gnutella, and Winny contributed to the increase of the traffic volume. Recently, the traffic volume of P2P applications have been dominant in the Internet traffic. Unlike the conventional application, P2P applications use large amount of the upstream bandwidth. The difference has a big impact to the traffic characteristics of the Internet.

Next generation backbone network should support new applications mentioned above. However, the characteristics of the traffic of these applications vary in terms of bandwidth, delay, reliability requirements and so on. In that case, there are many types of traffic mixed in backbone network. Supporting QoS is an important task in next generation backbone network. Traffic engineering is an essential technology to use network resources

efficiently and support QoS. Therefore, traffic engineering is a key issue in next generation backbone network.

One of main applications on the Internet is exchanging and sharing multimedia contents such as high resolution images of digital still cameras and high definition video. An application technology is built based on transport and control technologies. An application framework for large volume data distribution is an important function in next generation backbone network since the size of contents on the Internet increases from text to high definition video.

Spreading broadband access networks causes rapid growth of the traffic of backbone networks. New types of applications have been emerged as the bandwidth of access and backbone networks developed. Therefore, technologies for efficient data transport are required to deal with such applications. One of basic requirements is to provide high speed and high capacity backbone network to catch up the speed of traffic growth. Scalability is also a main requirement for next generation backbone network since so many devices are connected to the Internet in the era of ubiquitous computing. Traffic engineering is a key technology to support applications, which have different characteristics of their traffic. In addition, an application framework to distribute large volume data efficiently is an essential part of next generation backbone network.

1.2 Requirements for next generation backbone network

Next generation backbone network handles many different types of applications such as the conventional applications including E-mail and WWW, and emerging applications, for example video conference, P2P and multimedia content sharing. Each application demands different QoS characteristics. For example, E-mail and WWW are delay tolerant but loss sensitive applications. On the other hand, video conference is delay sensitive because delay makes a bad influence on the user experience. But, some data losses are

permitted in video conference because they causes only unnoticeable quality degradation.

As I mentioned in the previous section, the followings are requirements for next generation backbone network.

- High speed and large capacity
- Scalability
- Traffic engineering capability
- Application framework to distribute large volume data

Therefore, this dissertation focuses on the above four requirements to realize next generation backbone network.

1.3 Position of the dissertation

Figure 1.2 shows the relationship between the requirements and the research topics included in the dissertation. The first requirement for next generation backbone network is high speed and large capacity network. Wavelength Division Multiplexing (WDM) is a key technology to provide high speed and large capacity network economically. Then, Chapter 3 investigates a wavelength assignment in WDM network to realize high speed and large capacity network. The second requirement is scalability because more and more devices are accessing to the Internet in the near future. Chapter 4 focuses on the scalability issue in wide area Ethernet, which is recently attractive for carrier to provide high speed wide area network with reasonable price because of its cost effectiveness. The third requirement is traffic engineering capability to support different QoS demands from various applications. To realize sophisticated traffic engineering capability, Chapter 5 deals with high speed path calculation. Finally, an application framework for large data distribution in next generation backbone network is studied in Chapter 6.

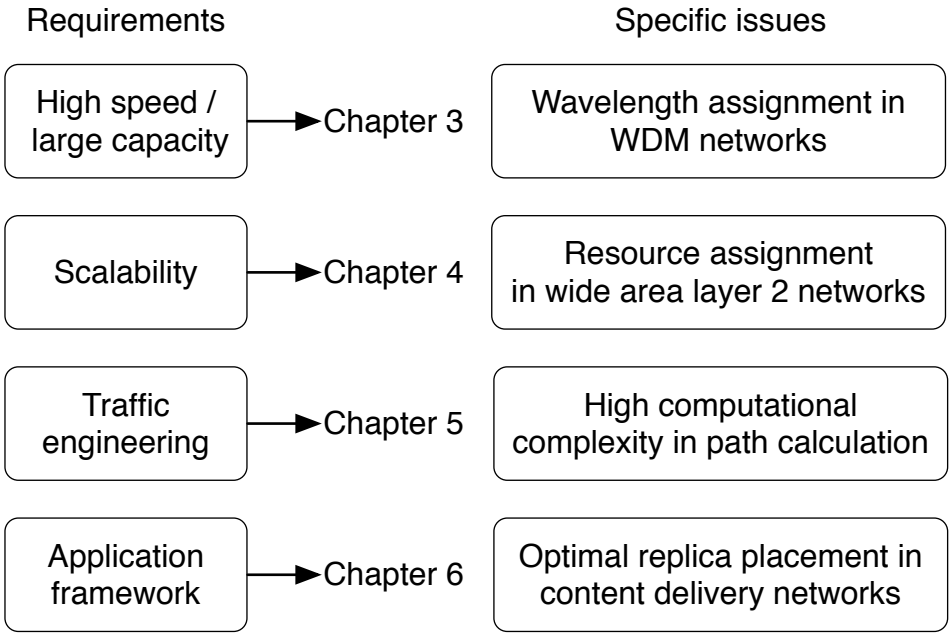


Figure 1.2: Relationship between the requirements and the research topics

Figure 1.3 shows technologies for next generation backbone network. They are classified into three categories: transport layer, control layer, and application layer. The Demand for high speed and large capacity network is related to WDM in transport layer technologies. Scalability issue is also in transport layer. Traffic engineering capability is included in control layer where Generalized Multi-Protocol Label Switching (GMPLS) is the framework to control and manage backbone networks. Application layer is built based on transport and control layer. There are many new types of applications, such as grid / cloud computing, P2P, CDN, and so on.

This work is for realizing highly efficient communication network technologies. This dissertation covers issues in transport layer to application layer. Chapter 3 and 4 focus on transport layer issues. A traffic engineering issue in control layer is investigated in Chapter 5. Application level approach, which deals with the optimal replica placement in CDN, is studied in Chapter 6.

The target of Chapter 3 is all optical wavelength-routed network to achieve high speed

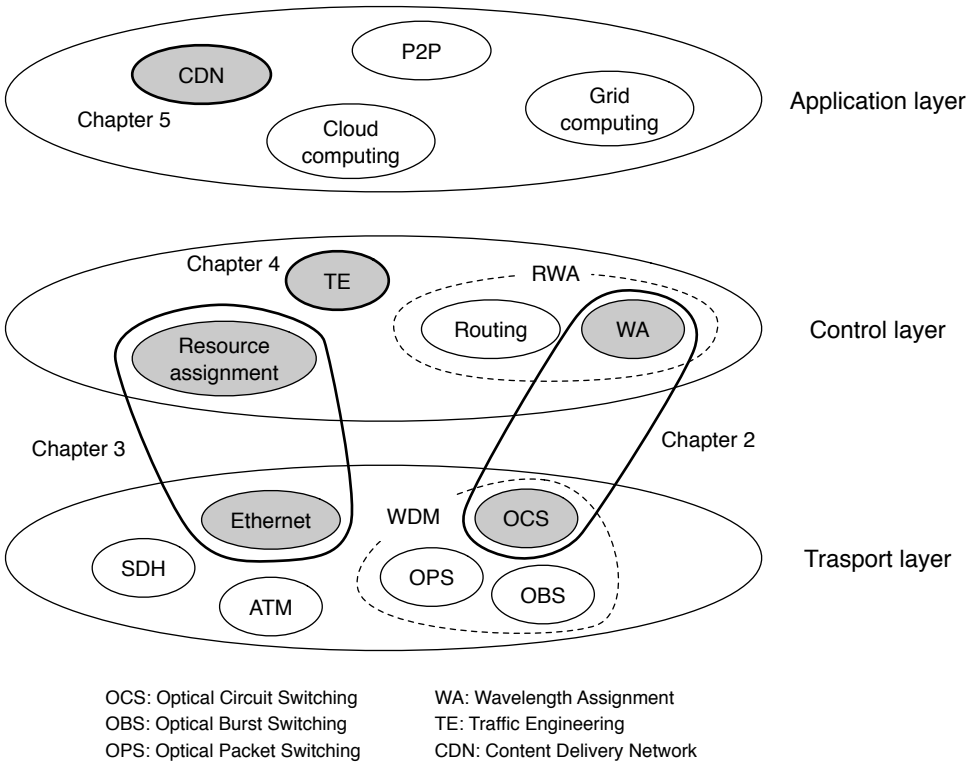


Figure 1.3: Position of this dissertation

and large capacity. The wavelength continuity constraint, that is the same wavelength have to be used on all of the links along a path, is the biggest limitation in wavelength-routed network [3]. Use of wavelength converters is a solution to remove this limitation. However, the cost of full range wavelength converters under current technology is extremely high, and use of limited range wavelength converters is a reasonable solution to relax the wavelength continuity constraint [4, 5]. The wavelength assignment scheme is important in wavelength-routed network with limited range wavelength converters since it decides the wavelength utilization efficiency and blocking probability. Review of wavelength assignment schemes is provided in [6]. A novel wavelength assignment scheme that considers the number of hops is proposed.

The target of Chapter 4 is next generation layer 2 network architecture to extend the scalability. Ethernet based wide area network [7–9] is promising layer-2 network due to its

cost effectivity. However, Ethernet originates from LAN technology, then the scalability of VLAN technology is an issue in wide area Ethernet. In the conventional VLAN tag-based Ethernet network (IEEE 802.1Q), a VLAN tag must be globally unique in a whole network. Only 12 bits are assigned to the field of VLAN tag. These imply that wide area Ethernet cannot support over 4096 Ethernet VLAN paths. That number of paths is not sufficient in WAN. VLAN tag swapped Ethernet architecture, which is an effective network architecture to increase network scalability, is proposed.

Chapter 5 focuses on traffic engineering to realize sophisticated traffic engineering. Sophisticated traffic engineering is an essential technology, and GMPLS is a framework for traffic engineering. Routing in GMPLS networks employing traffic engineering is based on multiple metrics such as the number of hops, link bandwidth, and transmission delay. In addition, in GMPLS networks, topology of IP layer is affected by a lightpath in optical layer. Lightpath establishment leads to change topology of IP layer, and re-calculation of the shortest paths is essential. Therefore, each router frequently re-calculates the shortest paths to create a routing table in GMPLS networks. In large-scale networks, the complexity of the shortest path search become more difficult. Ultra fast shortest path calculation can adopt to huge-size networks. A novel parallel shortest path algorithm called MPSA (Multi-route Parallel Search Algorithm) based on parallel data-flow type dynamically reconfigurable processors is proposed.

The target of Chapter 6 is an application framework for large volume data distribution. A new solution of the replica placement problem in CDN is proposed. [10] provides a comprehensive survey of replica placement algorithms. The proposed replica placement solution can generate all replica placement patterns at high rate by taking advantage of parallelism of DRP. The optimal replica placement, which means the number of replicas is minimum, can be obtained within reasonable time. As a result, it realizes efficient large volume content distribution.

Chapter 3 investigates wavelength assignment scheme in wavelength-routed network, which is a type of all optical WDM network, and it is the most fundamental research among the research topics in this dissertation. Chapter 4 deals with layer 2 network technology, which can be built on the topic in Chapter 3. Ethernet frames are sometimes transmitted on WDM networks. Both Chapter 3 and Chapter 4 propose resource assignment schemes, which are control layer technologies, respectively to solve the specific issue. In control layer, GMPLS is a basic framework to support traffic engineering and protection. The target networks in Chapter 3 and 4 are controlled by GMPLS in next generation backbone network. In GMPLS control plane, traffic engineering is a key function to control and manage the transport networks. Next, I focused on the issue in traffic engineering: high computational complexity of path calculation. Finally, Chapter 6 deals with the issue in application layer because application layer is built on transport and control layer technologies. In Chapter 6, a new approach to obtain the optimal replication is proposed to realize efficient large data distribution.

References

[1] M. of Internal Affairs and Communications, "Communications usage treand survey in 2008 compiled," http://www.johotsusintokei.soumu.go.jp/tsusin_riyou/data/eng_tsusin_riyou2008.pdf, 2008.

[2] "JPIX - technical information : Traffic," http://www.jpix.ad.jp/en/technical/traffic.html.

[3] M. Kovačević and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 868–880, June 1996.

[4] J. Yates, J. Lacey, D. Everitt, and M. Summerfield, "Limited-range wavelength translation in all-optical networks," *IEEE INFOCOMM'96*, vol. 3, pp. 954–961, Mar. 1996.

[5] L. Zhang and L. Li, "Effects of routing and wavelength assignment algorithms on limited-range wavelength conversion in WDM optical networks," *IEEE 2002 International Conference on Circuits and Systems and West Sino Expositions*, vol. 1, pp. 860–864, June 2002.

[6] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *SPIE Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, Jan. 2000.

[7] *IEEE Standards for Local and Metropolitan Area Networks Virtul Bridged Local Area Networks*, IEEE Standard 802.1Q, May 2006.

[8] *IEEE Standards for Provider Bridges*, IEEE Standard 802.1ad, Aug. 2005.

[9] *IEEE Standards for Provider Backbone Bridges*, IEEE Standard 802.1ah, Apr. 2008.

[10] M. Karlsson, C. Karamanolis, and M. Mahalingam, "A framework for evaluating replica placement algorithm," HP Laboratories Palo Alto, Tech. Rep., Aug. 2002.

Chapter 2

Backbone network technologies

This dissertation investigates transport layer to application layer to meet the requirements for next generation backbone network. In this chapter, fundamental technologies for next generation backbone network are described, and previous works related to the topics in Chapter 3 – 6 are explained.

2.1 Fundamental technologies

2.1.1 Wavelength Division Multiplexing (WDM)

Wavelength Division Multiplexing (WDM) contributed to increasing the capacity of backbone networks. WDM utilizes multiple wavelengths in single optical fiber concurrently. The capacity of the single optical fiber can be increased as the number of wavelengths in an optical fiber grows. Using WDM technology enables providing high speed and large capacity networks at low cost since the optical fibers, which was already installed, can be reused. It was recently reported that total capacity can reach up to 13.4 Tbps, which is sum of 134 wavelengths (the bandwidth of a wavelength is 111 Gbps) [1].

The domain of applicability of WDM has been growing. WDM first employed in long haul transmission, for example submarine cable systems or national level backbone networks. WDM technology can reduce the total cost of the network compared with using coaxial cables. The demand for high speed and large capacity transmission is limited only in long haul transmissions or national level backbone networks since the total traffic

of the networks was not formerly large. Japan-U.S. CN (Total capacity 400 Gbps: 10 Gbps $\times$ 40 wavelengths), which is between Japan and the U.S. and China, is an example of submarine cable systems, and JIH (Japan Information Highway) operated by KDDI is an example of national level backbone networks. Recently, WDM has been applied in Metropolitan Area Network (MAN) and access networks due to increasing of the traffic of the Internet.

Point-to-point based WDM network, which consists of WDM transponders and electric intermediate repeater, is the first generation. In the networks, O/E conversion, amplification of the input signal, and E/O conversion have to be done through the intermediate nodes. It is difficult to increase the total capacity of a node because all of the processing in a repeater is done in electrical domain. Then, optical fiber amplifiers such as Erbium Doped Fiber Amplifier (EDFA) have been employed in the intermediate nodes to remove the electrical processing. It led to increase the total capacity of WDM networks. WDM networks have been evolved and employed in ring networks such as MAN and mesh networks such as national backbone networks.

As mentioned before, the electrical processing is bottleneck since it is slower than the potential speed of optical fibers when it is used in nodes in a network. All optical networks, where no electrical processing is used, are desirable for increasing the total network capacity. However, it is technically difficult to realize optical RAMs, and they are not practically available now. We need an architecture of all optical networks which is fitted to the characteristics of optics because it is hard to store optical signals in optical domain. Switching techniques in all optical networks are classified in to the following three categories; Optical Circuit Switching (OCS), Optical Packet Switching (OPS), and Optical Burst Switching (OBS).

OCS

OCS is connection oriented optical switching [2]. A source node sets up a lightpath to the destination node before starting the communication. Signaling is used for lightpath establishment. OCS is also referred to as wavelength routed networks.

OPS

OPS [3–5], as the name implies, is packet based optical network. Data, which is represented in optical signals, is transferred through OPS networks. It is required to store optical packets in core node since OPS is based on store and forward. On the other hand, it is difficult that optical RAMs are available in current technologies. OPS is still in research phase while some researches on OPS for practical use have been done.

OBS

OBS [6] is an optical switching between OCS and OPS. OBS network consists of edge nodes and core nodes. An edge node assembles incoming IP packets destined for the same IP address into a burst, and send the burst. At a core node, only the header of a burst is electrically processed. On the other hand, the payload of the burst is forwarded in optical domain. The granularity of switching in OBS network is relaxed compared to that of OPS network. In addition, store and forward mechanism is not employed in core nodes. Consequently, the feasibility of OBS is higher than OPS.

2.1.2 Generalized Multi-protocol Label Switching (GMPLS)

GMPLS is a multipurpose control plane technology proposed by the IETF to support multiple types of switching paradigms, including not only packet switching but also time slot switching, wavelength (or waveband) switching, and fiber switching [7]. It general-

izes the MPLS traffic engineering control plane and supports not only devices that perform packet switching but also devices that perform switching in time, wavelength, and space domains. See [8,9] for more detail.

Switching Capability

In GMPLS, the concept of Label Switched Routers (LSRs) in MPLS [10] has been extended to include LSRs or, more precisely, LSR interfaces that forward data based on time slots, wavelengths, and physical ports or fibers. These new types of LSRs or LSR interfaces can be classified as follows:

- Packet-switch-capable (PSC) interfaces: This type of interface forwards data based on the header of the packets or cells that carry the data, such as an interface on an LSR that forwards data based on the shim header or interface on an ATM switch that forwards data based on the ATM cell header.

- Time-division-multiplex-capable (TDM) interfaces: This type of interface forwards data based on the time slots that carry the data, such as an interface on a SONET/SDH cross-connect.

- Lambda-switch-capable (LSC) interfaces: This type of interface forwards data based on the wavelength that carries the data, such as an interface on an OXC that operates at the level of single wavelength or at the level of a waveband (or a group of wavelengths).

- Fiber-switch-capable (FSC) interfaces: This type of interface forwards data based on the fibers or ports that carry the data, such as an interface on an OXC that operates at the level of single fiber or multiple fibers.

GMPLS supports the concept of a forwarding hierarchy or Label Switched Path (LSP) hierarchy, i.e., an LSP can be nested inside another LSP. In GMPLS, the concept of LSPs

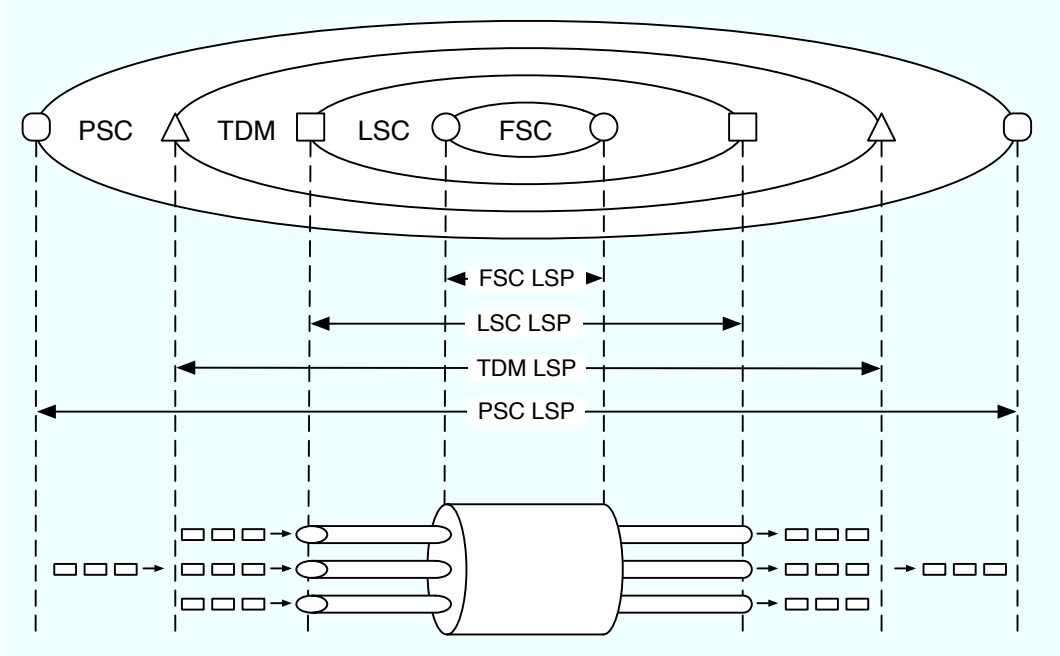


Figure 2.1: Illustration of the LSP hierarchy in GMPLS

has been extended to include LSPs established on different types of interfaces, such as a SONET connection and a lightpath. However, an LSP can be established only between interfaces of the same type. The LSP hierarchy exists between different types of interfaces. The top of this hierarchy is FSC interfaces, followed by LSC interfaces, followed by TDM interfaces, followed by PSC interfaces. As a result, an LSP that originates and terminates on a PSC interface can be nested (together with other LSPs) into an LSP that originates and terminates on a TDM interface. This TDM LSP, in turn, can be nested (together with other TDM LSPs) into an LSP that originates and terminates on an LSC interface, which in turn can be nested (together with other LSPs) into an LSP that originates and terminates on an FSC interface. The LSP hierarchy in GMPLS is illustrated in Fig. 2.1

Separation of Data Plane and Control Plane

It is a characteristic of GMPLS to separate data plane and control plane. In MPLS networks, all nodes which support MPLS protocol have interfaces that can deal with packets. Therefore, control packets about routing protocol and signaling protocol can be transmitted through the physical media which the data is transmitted. In GMPLS networks, however, not all interfaces have packet switching capability. TDM, LSC, and FSC interfaces do not have packet switching capability, so control packets are logically transmitted through the interfaces that do not transmit the data. Data plane and control plane are logically separated in GMPLS networks. The interfaces in control plane can distinguish packets and process them.

2.1.3 Carrier grade Ethernet

Recent innovations in Ethernet networking technology are attractive as backbone transport technology [11]. Ethernet is the most common networking technology in the world. A number of Ethernet equipped device and Ethernet switches is available in the market, and Ethernet is deployed not only in home but also in enterprises. The commoditization made Ethernet cost effective. In addition, The wide-spreading of IP networks made Ethernet more popular than other networking technology such as token Token Ring since Ethernet is familiar with IP networks.

However, the functionality of Ethernet is insufficient in the service provider domain because Ethernet originates from Local Area Network (LAN) technology. The legacy networking technology used in service providers such as synchronous digital hierarchy (SDH) and asynchronous transfer mode (ATM) provides scalability, protection, hard quality of service (QoS), and service management. These functionalities are required to use Ethernet in Wide Area Network (WAN). The largest difference between LAN and WAN is scalability. To extend the scalability of Ethernet, different Ethernet technologies have

been proposed and the standardization is driven by the IEEE 802.1 working group.

Virtual LAN

The basic technology standard used for delivering a multipoint-to-multipoint connection service is the IEEE 802.1Q standard [12] for virtual LANs (VLANs). This standard creates VLANs across a common LAN infrastructure to enable enterprises to support and separate traffic from different departments within a company. Each VLAN is identified by a Q-tag (also known as a VLAN tag or VLAN ID) that identifies a logical partitioning of the network to serve different communities of interest.

Virtual LAN works fine within a single organization, but it is found to be inappropriate when it is used in service providers. There are two problems. One is the administration of VLANs and the other is scalability. The administration problem occurs because enterprises need to keep control over their own VLAN administration, e.g. assigning Q-tags to VLANs, and the service provider must control this to ensure that one customer's Q-tags do not overlap with another's. The scalability problem occurs because only a 12-bit field is assigned to Q-tag. That means up to 4094 possible service instances can be created. (Note that 4096 Q-tags are available, but two of them are reserved for administration.) Although it is sufficient for enterprise's LANs, it does not provide the scalability required to use Ethernet in WANs. IEEE 802.1ad provider bridges (also know as Q-in-Q or VLAN stacking) [13] and IEEE 802.1ah provider backbone bridges (also known as MAC-in-MAC) [14] is standards to extend the scalability of IEEE 802.1Q. IEEE 802.1ad provider bridges standard was officially approved in December 2005, and IEEE 802.1ah provider backbone bridges standard was officially approved in June 2008.

Provider bridge

Provider bridges work by simply adding an additional service provider VLAN ID (S-Tag) to the customer's Ethernet frame. This new S-Tag is used to identify the service in the provider network, while the customer's VLAN ID (C-Tag) remains intact and is not altered by the service provider. It means the C-Tag is transparent within a Q-in-Q network. Each service instance requires a separate S-Tag, and a 12-bit field is allocated to the S-Tag. Therefore, provider bridges have the same limitation of the scalability as IEEE 802.1Q: Only 4094 service instances can be supported.

Provider bridges use the same MAC address for the provider's and customer's networks. The provider's switch treats both networks as one large network. In other words, the provider's and customers' MAC addresses are visible to all network elements of the service provider. This is a burden for core switches since they must maintain a forwarding table for each MAC address in the provider and customer networks. This implies any changes in the customer network will affect the provider network. In addition, there are potential security concern from the customers' perspective since their addressing information is visible to the other customers' networks.

Provider backbone bridge

Provider backbone bridges (PBBs) extend the Ethernet frame by adding a MAC header dedicated to the service provider. A backbone source and destination MAC address, a backbone VLAN ID (B-Tag), and a backbone service ID (I-Tag) are introduced to the provider bridge's frame. Figure 2.2 shows the PBB frame, the original Ethernet frame (IEEE 802.1), VLAN frame (IEEE 802.1Q), and provider bridge frame (IEEE 802.1ad). The I-Tag identifies the service in the PBB network, and 24 bits are assigned to the I-Tag. This means PBBs remove the scalability limitation of provider bridges since up to 16 million service instances can be supported in a PBB network. In addition, the MAC addresses

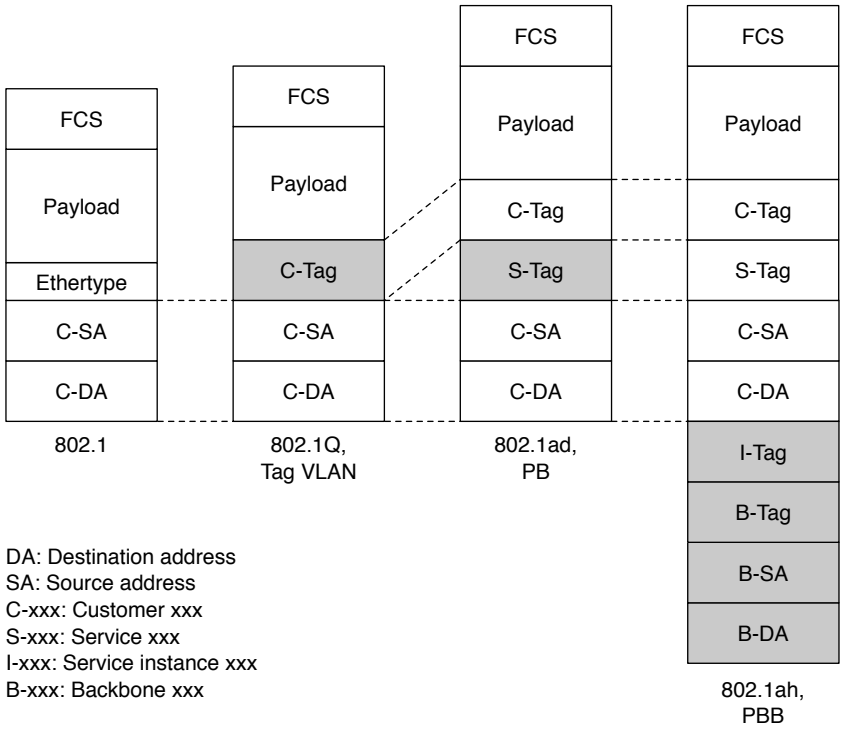


Figure 2.2: Frame formats of PBB, the original Ethernet, VLAN, and provider bridge

of Q-in-Q network and PBB network are completely separated in a PBB network because each has a dedicated set of MAC addresses. An Ethernet frame is encapsulated into the PBB frame at the edge switch of a PBB network when the Ethernet frame reaches the edge. The potential security vulnerability is removed because the customer's source and destination MAC address is invisible to the PBB network. Additionally, the burden on the forwarding tables in the PBB network is relaxed. Changes in the provider bridge network will not affect the PBB network and the stability of the PBB network is improved.

2.1.4 Content Delivery Network (CDN)

Generally the capacity of a single server does not scale well to serve increasing user requests. The bandwidth at the server is a possible limitation to serve contents to users in reasonable speed because the bandwidth assigned to a user decreases when many users

concurrently access to the server. Caching and replication have been useful techniques to reduce the latency of users and the load of the server. Content delivery network (CDN) has been proposed as a more systematic approach for caching and replication, and it is widely used in real Internet applications.

CDN consists of two types of servers as show in Fig. 2.3: origin server and replica server. The original data is stored in the origin server and then it is replicated to the replica servers, which are geographically distributed. Consequently, the data is geographically distributed. A user request to retrieve the data is redirected to the nearest replica server, then the user obtains the data from the nearest replica server. The distance between a user and the server is reduced and the download speed is improved. There are other two advantages in CDN system. One is decreasing in the load of the origin servers because the most of user requests is resolved at the replica servers. The other advantage is fault tolerance. In the single server model, the server is single point of failure. On the other hand, when a replica server has failure, a user request can be redirected to another replica server. The most famous CDN service is Akamai [15].

2.2 Previous works

2.2.1 Routing and Wavelength Assignment (RWA)

One of common issues in all optical wavelength routed networks is routing and wavelength assignment (RWA) problem. We have to determine what path and wavelength will be used. This is because lightpath establishment is required before starting communication in all optical wavelength routed networks. Wavelength continuity constraint, which means a lightpath must use the same wavelength on all links along the path, exists in all optical wavelength routed networks. This constraint affects the network performance such as the blocking probability. As a result, RWA is the most important issue there.

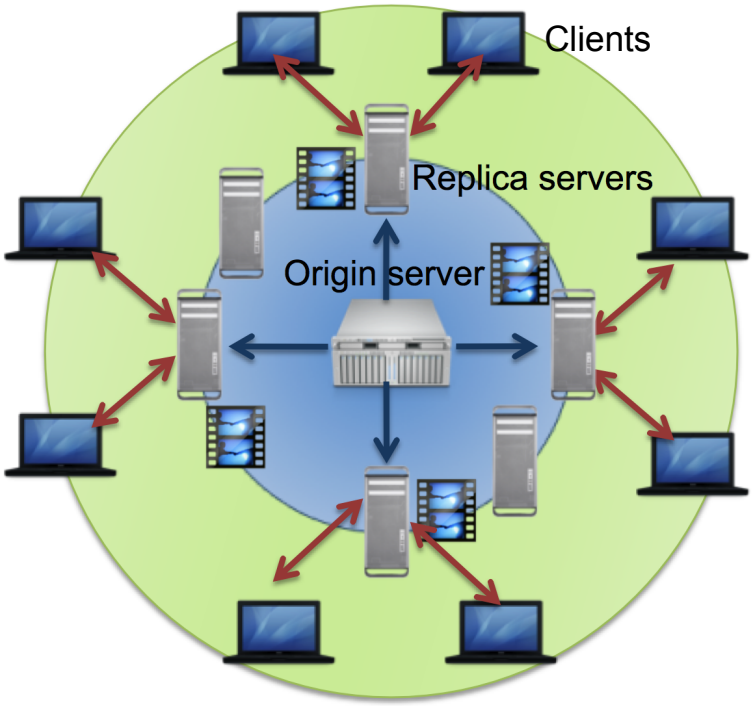


Figure 2.3: Origin server and replica servers in CDN

Figure 2.4 overviews the researches on RWA. First of all, RWA is NP-complete problem, consequently, heuristic approaches have been widely studied. The typical sequence of RWA is 1) Calculating a path, which a lightpath will go through, and then 2) Selecting a wavelength, which a lightpath will use on the calculated path. However, there are some approaches that a path and a wavelength are calculated at the same time. Some researches focused on routing, the other focused on wavelength assignment. In addition, researches on RWA can be classified in terms of targeted optical network architecture. The classification about the targeted networks is shown in Fig. 2.5. The networks can be classified into networks with and without wavelength converters. Additionally, the networks with wavelength converters can be classified by the limitation in wavelength conversion range; Network with full range wavelength converters, and network with limited range wavelength converters. Networks with full range wavelength converters are classified in terms

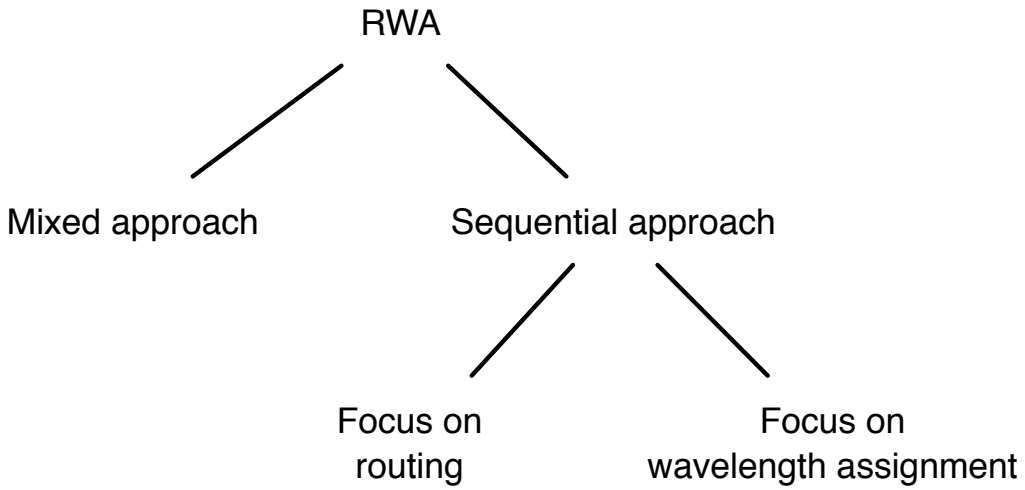


Figure 2.4: Classification of RWA algorithms

of placement type of wavelength converters. One is sparse wavelength conversion, which is a part of nodes has wavelength converters, and the other is non sparse wavelength conversion. An alternative to the use of wavelength converters with limited range is the use of wavelength converters at selected places in the network [16]. Wavelength converters are high cost device, so reducing them makes wavelength-routed networks more cost effective. In static problems, the difference between networks with no wavelength converter and networks with full range wavelength converters is very limited in the literature [17].

Routing

Routing algorithms are classified into three types; Fixed routing, fixed-alternate routing, and adaptive routing [18, 19]. The computation complexity to determine a path increases in this order. On the other hand, the blocking probability decreases in the order since more candidate of paths can be calculated as the complexity of an algorithm is higher.

Fixed routing is the simplest type of routing algorithms. Only one path is determined

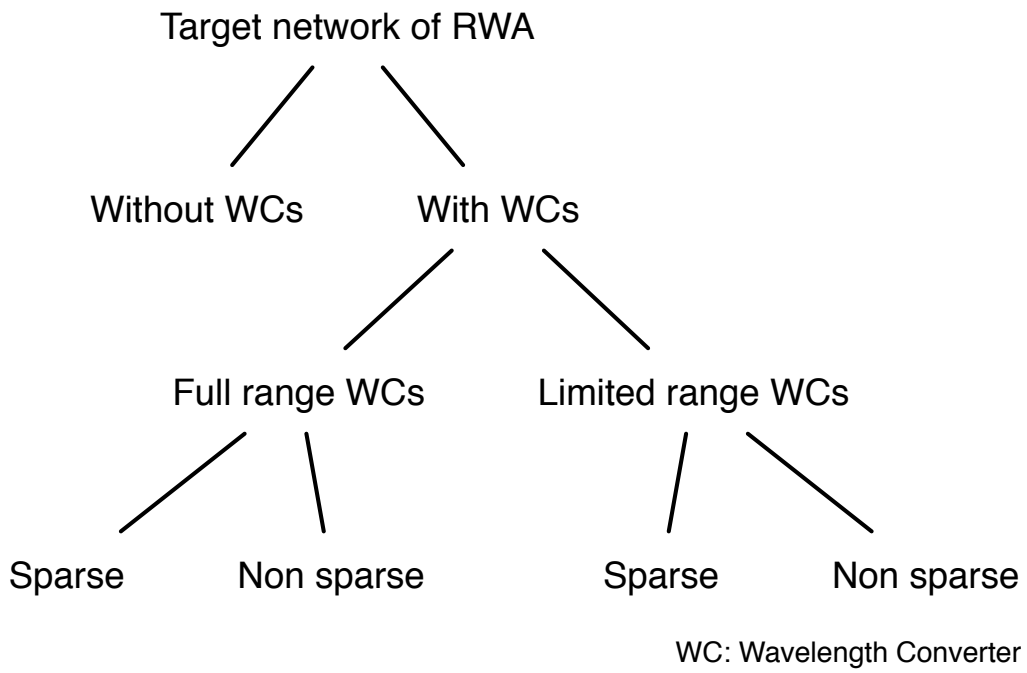


Figure 2.5: Classification of the target network of researches on RWA

between a pair of a source and a destination node. It is assumed that a link cost is set on each link. Then, the smallest cost path between every pair of two nodes is calculated with the shortest path algorithm, such as Dijkstra's algorithm, beforehand. When a connection request arrives, the corresponding path is used as a path. The number of hops or the distance of a link is usually used as link cost. The advantage of fixed routing is that the delay time of path set up is small since path selection is static and all of paths can be calculated before a connection request arrives. On the other hand, the disadvantage of fixed routing is inflexibility. The blocking probability is going to high when many paths go through a specific link due to the topology or non uniform traffic. In addition, a path cannot be changed when a failure occurs. Shortest path routing, where the number of hops is used as link cost, is a popular fixed routing.

Fixed-alternate routing is an approach to routing that considers multiple routes. In fixed-alternate routing, each node in the network is required to maintain routing table

that contains an ordered list of number of fixed routes to each destination node. For example, the shortest path, the second shortest path, and the third shortest path may be included. When a connection request arrives, the source node tries to find an available path from the routing table sequentially. The first available path is established. If no available path is found, the connection request is blocked. The blocking probability of fixed-alternate routing is lower than that of fixed routing since a path can be chosen from multiple candidate paths. Fixed-alternate routing provides some degree of fault tolerance upon link failures. However, in fixed-alternate routing, the current network usage is not considered to select a path. As a result, the blocking probability becomes high when the usage of the network greatly fluctuates. ”k-shortest path routing” is a popular fixed-alternate routing. In k-shortest path routing, k paths are stored in the routing table of a node to each destination in ascending order of the number of hops. The available path with minimum hops is established.

In adaptive routing, the path from a source node to a destination node is selected dynamically, depending on the network state. One form of adaptive routing selects a path from the pre-selected paths like fixed-alternate routing. The other form determined a path from the all of paths from the source node and the destination node dynamically when a connection request arrives. An advantage of adaptive routing is that it achieves lower blocking probability than fixed and fixed-alternate routing since it calculates the most adequate path dynamically. On the other hand, the computational complexity increases compared with fixed and fixed-alternate routing. In addition, extensive support from the control and management protocols to continuously update the current network state information at the node. One of popular adaptive routing is least-congested path (LCP) routing. In LCP routing, for each source-destination pair, multiple paths are pre-selected. When a connection request arrives, the least-congested path among the pre-determined paths is chosen. The congestion on a link is measured by the number of wavelength available on

the link. Links that have fewer available wavelengths are regarded to be more congested. The congestion of a path is measured by the congestion of the most congested link in the path. If there is some equal congested paths, the shortest path among the paths is selected.

Wavelength assignment

In this sub-section, popular wavelength assignment scheme, which includes random, first-fit, least-used, most-used, min-product, and least-loaded, is denoted. The following notation and definition are used in the description. The detail of various wavelength assignment scheme is mentioned in [19].

- L: Number of links
- M_l: Number of fibers to link l
- M: Number of fibers per link if all links contain the same number of fibers.
- W: Number of wavelength per fiber.
- $\pi(p)$: Set of links comprising path p.
- S_p: Set of available wavelengths along the selected path p.
- D: L-by-W matrix, where D_{lj} indicated the number of assigned fibers on linnk l and wavelength j. Note that the value of D_{lj} varies between 0 and M_l

Random wavelength assignment chooses one wavelength randomly among all the available wavelengths on a requested path.

In first-fit (FF), all wavelengths are numbered. When searching for available wavelengths, a lower-numbered wavelength has higher priority than higher-numbered wavelength. This scheme doesn't require global information of the network state. Compared to random wavelength assignment, the computational cost of this scheme is lower because

it is not needed to search the entire wavelength space for each route. The first available wavelength is selected. The idea behind this scheme is to pack all of the utilized wavelengths toward the lower end of the wavelength space. Then, continuous longer paths toward the higher end of the wavelength space will have a higher probability of being available. The blocking probability of this scheme is well, and preferred in practice because of its small computational complexity.

Least-used (LU) selects the wavelength that is the least used in the network. The purpose of this scheme is to balance the load among all the wavelengths. This scheme tends to break the long wavelength path. Hence, only connection requests whose number of hops is small will be established in the network. The performance of LU is worse than random wavelength assignment although LU requires global information. Therefore, LU is not preferred in practice.

Most-used (MU) is the opposite of LU. MU selects the most-used wavelength in the network. The performance of MU is much better than LU [20]. The communication overhead, and computational complexity are similar to those in LU. The performance of MU is also better than FF since it better packs connections into fewer wavelengths and preserve the space capacity of less-used wavelengths.

Min-Product (MP) is used in multi-fiber networks [21]. In a single-fiber network, MP is equal to FF. The goal of MP is to pack wavelength to fibers, thereby minimizing the number of fibers in the network. MP first calculates

$$\prod_{l \in \pi(p)} D_{lj} \tag{2.1}$$

for each wavelength j, which indicates the value between 1 and W. MP selects the minimum numbered wavelength among the set of wavelength j that minimizes the above value. MP does not outperform the multi-fiber version of FF [20]. However, it introduces additional computational costs.

Least-loaded (LL) is designed for multi-fiber networks as well as MP. This scheme selects the wavelength that has the largest residual capacity on the most-loaded link along path p. When it is used in single-fiber networks, the residual capacity is either 1 or 0; thus, LL selects the lowest-numbered wavelength with residual capacity 1. It is equal to FF in single-fiber networks. LL selects the minimum numbered wavelength j in S_p that achieves

$$\max_{j \in S_p} \min_{l \in \pi(p)} M_l - D_{lj} \tag{2.2}$$

The blocking probability of LL in a multi-fiber network is lower than that of MU and FF [22].

2.2.2 Path calculation

Path computation is a foundation of load balancing in traffic engineering. The demand for high speed path computation have been growing since the complexity of path computation in multilayer networks, such as GMPLS networks, is high. In addition, use of Path Computation Elements (PCEs) has been considered recently. It also demand the high speed path computation.

Dijkstra's algorithm

In general, a node exchanges informations about the network topology with the adjacent nodes, and calculates the shortest paths to create the routing table. In OSPF [23], a popular routing protocol, Dijkstra's algorithm [24] is employed as the shortest path algorithm. It searches for all shortest paths between a source node s and all other nodes in a directed graph with nonnegative edge weight $G = (V, E)$

We denote a set of vertices as $V = \{1, \cdots, n\}$, and the weighted graph made of a set of links E as $G = (V, E)$. $c(i, j)$ is the cost of edge (i,j). $d(v)$ is the distance between s and a

vertex v. $p(v)$ is a pointer indicating the upstream direction from vertex v on the shortest path. The following describes the algorithm for the source of vertex v_0.

Step 1 $U = \{v_0\}, W = \emptyset, d(v_0) = 0, d(u) = +\infty (u \in V \setminus \{v_0\})$.

Step 2 If $U = \emptyset$ then finish

Otherwise,

$w = \{v | v \in U, d(v)$ is the minimum for $U\}$.

For $\{a | a = (w, x) \in A, x \notin W\}$,

(*) if $p(x) > p(w) + c(w, x)$

$q(x) \leftarrow w, p(x) \leftarrow p(w) + c(w, x), U \leftarrow U \cup \{x\}$.

Step 3 : $W \leftarrow W \cup \{w\}, U \leftarrow U \setminus \{w\}$ then go to Step 3.

Improving Dijkstra's algorithm

Dijkstra's algorithm is the basic tool for shortest path search, and then many improvements have been developed since 1959. The computational complexity of Dijkstra's algorithm is $O(n^2)$, where n is the number of nodes. The bottleneck of Dijkstra's algorithm is sorting in order of increasing distance from the source [25]. The improved algorithms use data structures, such as priority queues and heaps, that allow faster sorts. Here, m is the number of edges in the graph. Applying Williams' heap, yields the time complexity of $O(m \log n)$ time [26]. Fibinacci heaps reduced the running time to $O(m + n \log n)$. Using Fredman and Willard's fusion tree, we get $O(m \sqrt{\log n})$ time [27]. Their later atomic heaps give the lower bounde $O(m + n \log n / \log \log n)$ [28]. The fastest algorithm we know of is Thorup's algorithm, which applies the hierarchical bucketing structure. This algorithm achieves $O(m)$.

2.2.3 Replica placement problem

In CDN, replica placement impacts the performance which includes the load on the origin server and the network since data placement decisions must be made on a per content basis and be made dynamically in response to user requests. Minimizing the number of mirroring resources (servers) under a Quality of Service (QoS) constraint is a key issue in CDN, so research in this area has been quite active. It is a tough problem to select which nodes should host which replicas.

Replica placement problem is derived from the set cover problem which is known to be NP-hard [29]. Therefore, calculation time increases rapidly with network scale. Greedy algorithms have been widely studied since they yield sub-optimal solutions reasonably quickly [30–37]. However, it has been proven mathematically that no greedy algorithm always can attain the optimal solution [29]. Johnson proposed a greedy algorithm against the minimum weight set cover problem [30]. This algorithm is a straightforward heuristic. The time complexity is proportional to n. In [31, 34], fan-out based replica placement algorithms were proposed. They put replicas on servers in descending order of server degree. Kangasharju et al. proved that their target replica placement optimization problem is NP-complete, and proposed some heuristic algorithms [35]. Tang et al. investigated QoS-aware replica placement problems to elucidate QoS requirements, and proposed the l-Greedy-Insert and l-Greedy-Delete algorithm [36]. They showed that the QoS-aware placement problem for replica-aware services was NP-complete. Wang et al. proposed a heuristic algorithm called Greedy-Cover [37]. Experiments indicated that the proposed algorithm found near-optimal solutions effectively and efficiently. Karlsson et al. provided a framework for evaluating replica placement algorithms [33], and compared several replica placement algorithms [32]. [32] also provides a comprehensive survey of replica placement algorithms.

References

[1] A. Sano, E. Yamada, H. Masuda, E. Yamazaki, T. Kobayashi, E. Eoshida, Y. Miyamoto, S. Matsuoka, R. Kudo, K. Ishihara, Y. Takatori, M. Mizoguchi, K. Okada, K. Hagimoto, H. Yamazaki, S. Kamei, and H. Ishii, "13.4-tb/s (134 x 111-gb/s/ch) no-guard-interval coherent OFDM transmission over 3,600 km of SMF with 19-ps average PMD," in *European Coference on Optical Communication 2008 (ECOC 2008)*, Brussels, Belgium, Sept. 2008, pp. 1–2.

[2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical wan's," *IEEE Transaction on Communications*, vol. 49, no. 7, pp. 1171–1182, July 1992.

[3] S. L. Danielsen, B. Mikkelsen, C. Joergensen, T. Durhuus, and K. E. Stubkjaer, "WDM packet switch architecture and analysis of the influence of tunable wavelength converters on the performance," *IEEE Journal of Lightwave Technology*, vol. 15, no. 2, pp. 219–227, Feb. 1997.

[4] S. L. Danielsen, C. Joergensen, B. Mikkelsen, and K. E. Stubkjaer, "Optical packet switched network layer without optical buffers," *IEEE Photonics Technology Letters*, vol. 10, no. 6, pp. 896–898, June 1998.

[5] P. Gambini, M. Renaud, C. Guillemot, F. Callegati, I. Andonovic, B. Bostica, D. Chiaroni, G. Corazza, S. L. Danielsen, P. Gravey, P. B. Hansen, M. Henry, C. Janz, A. Kloch, R. Krähenbühl, C. Raffaelli, M. Schilling, A. Talneau, and L. Zuc-

chelli, "Transport optical packet switching network architecture and demonstration in the KEOPS project," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1245–1259, Sept. 1998.

[6] C. Qiao and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical internet," *Journal of High Speed Switching*, vol. 8, no. 1, pp. 69–84, Mar. 1999.

[7] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," *Request for Comments (RFC)*, no. 3945, Oct. 2004.

[8] A. Banerjee, J. Drake, J. P. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label swtching: An overview of routing and management enhacements," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 144–150, Jan. 2001.

[9] A. Banerjee, L. Drake, L. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques," *IEEE Communications Magazine*, vol. 39, no. 7, pp. 144–151, July 2001.

[10] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," *Request for Comments (RFC)*, no. 3031, Jan. 2001.

[11] R. Sánchez, L. Raptis, and K. Vaxevanakis, "Ethernet as a carrier grade technology: Developments and innovations," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 88–94, Sept. 2008.

[12] *IEEE Standards for Local and Metropolitan Area Networks Virtul Bridged Local Area Networks*, IEEE Standard 802.1Q, May 2006.

[13] *IEEE Standards for Provider Bridges*, IEEE Standard 802.1ad, Aug. 2005.

[14] *IEEE Standards for Provider Backbone Bridges*, IEEE Standard 802.1ah, Apr. 2008.

[15] "Akamai," http://www.akamai.com/.

[16] N. Wauters, W. V. Parys, B. V. Caenegem, and P. Demeester, "Reduction of wavelength blocking through partitioning with wavelength convertors," in *Optical Fiber Communication Conference 1997 (OFC '97)*, Dallas, Texas, Feb. 1997, pp. 122–123.

[17] N. Nagatsu, Y. Hamazumi, and K. Sato, "Optical path accommodation designs applicable to large scale networks," *IEICE TRANSACTIONS ON COMMUNICATIONS E SERIES B*, vol. 78, pp. 597–597, 1995.

[18] N. Wauters and P. Demeester, "Design of the optical path layer in multiwavelength cross-connected networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 881–892, May 1996.

[19] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *SPIE Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, Jan. 2000.

[20] S. Subramaniam and R. A. Barry, "Wavelength assignment in fixed routing WDM netowrks," in *International Conference on Communications (ICC '97)*, vol. 1, Montreal, Canada, June 1997, pp. 406–410.

[21] G. Jeong and E. Ayanoglu, "Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks," in *IEEE INFOCOM '96*, vol. 1, San Francisco, CA, Mar. 1996, pp. 156–163.

[22] E. Karasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 186–196, Apr. 1998.

[23] J. Moy, "OSPF version 2," *Request For Comments (RFC)*, no. 2328, Apr. 1998.

[24] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, Oct. 1959.

[25] M. Thorup, "Undirected single source shortest paths in linear time," *IEEE Symposium on Foundations of Computer Science*, pp. 12–21, 1997.

[26] J. W. J. Williams, "Heapsort," *Communications of the ACM*, vol. 7, no. 6, pp. 347–348, May 1964.

[27] M. L. Fredman and D. E. Willard, "Surpassing the information theoretic bound with fusion trees," *Journal of Computer and System Science*, vol. 47, no. 3, pp. 424–436, Dec. 1993.

[28] ——, "Trans-dichotomous algorithm for minimum spanning trees and shortest paths," *Journal of Computer and System Science*, vol. 48, no. 3, pp. 533–551, June 1994.

[29] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[30] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Science*, vol. 9, pp. 256–278, 1974.

[31] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the internet," in *INFOCOM 2001*, 2001.

[32] M. Karlsson, C. Karamanolis, and M. Mahalingam, "A framework for evaluating replica placement algorithm," HP Laboratories Palo Alto, Tech. Rep., Aug. 2002.

[33] M. Karlsson and M. Mahalingam, "Do we need replica placement algorithms in content delivery netowrks?" in *The International Workshop on Web Content Caching and Distribution (WCW)*, Aug. 2002, pp. 117–128.

[34] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-informed internet replica placement," *Computer Communications*, vol. 25, no. 4, pp. 384–392, Mar. 2002.

[35] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, Mar. 2002.

[36] X. Tang and J. Xu, "Qos-aware replica placement for content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 921–932, Oct. 2005.

[37] H. Wang, P. Liu, and J.-J. Wu, "A qos-aware heuristic algorithm for replica placement," in *Grid Computing 7th IEEE/ACM International Conference*, Sept. 2006, pp. 96–103.

Chapter 3

Wavelength assignment scheme for high speed and large capacity WDM networks

3.1 Abstract

In this chapter, a new wavelength assignment scheme that improves the blocking probability of WDM networks that use limited-range wavelength converters is proposed to realize high speed and large capacity transport cost-effectively. The major issue in all optical WDM networks is the wavelength continuity constraint, which increases the blocking probability. Limited-range wavelength converters are attractive given current technology since they offer good utilization of the wavelength resource and improved blocking probability. However, their conversion ranges are limited. Thus, the existence of these limited-range wavelength converters have to be taken into account. In the proposed scheme, each connection request is assigned a different wavelength according to its hop number. Different wavelengths tend to be used for connection requests with different hop numbers. As a result, the blocking probability can be reduced by two decades compared to simply assigning the smallest indexed available wavelengths. In addition, it allows the number of wavelength converters used in each node to be reduced with almost no degradation in blocking probability. Simulation results show that the proposed scheme can reduce the wavelength converters by about 20 percent. The proposed wavelength assignment makes high speed and large capacity transport cheaper.

3.2 Introduction

Wavelength-routed networks are attractive for realizing the next generation wide-area networks since they offer a data transmission scheme for WDM all-optical networks [1]. In wavelength-routed networks, data is transferred on lightpaths. A lightpath is an optical path established between the source node and the destination node. When a connection request arrives, a lightpath is set up. This involves routing and signaling to reserve a wavelength on each link along the path selected. The benefit of wavelength-routed networks is the ability to more fully utilize the bandwidth of optical fibers since they do not require processing, buffering, and opto-electronic-optic (O/E/O) conversions at intermediate nodes.

The simplest wavelength-routed network assigns one wavelength to all links of the connection between the source node and the destination node. This requirement is known as the wavelength continuity constraint. The constraint can be avoided by the use of wavelength converters at intermediate nodes. A wavelength converter is a device which converts the input wavelength λ_i into a different wavelength which is then output λ_o. In wavelength-routed networks with wavelength converters, a lightpath can be established even though there is no common wavelength on all links along the path. This approach can improve the blocking probability and the efficient utilization of wavelengths [2]. The wavelength converters assumed in [2] provide full-range wavelength conversion capability. This means that any input wavelength can be converted to any output wavelength. While it is possible to realize full-range wavelength converters optoelectronically, such means that the networks lose the benefit of optically-transparent wavelength-routed networks. Given current technologies, one of the most popular all-optical wavelength conversion techniques is four-wave mixing (FWM) [3]. In FWM based wavelength converters, the output signal power is significantly degraded as the difference between the input wavelength and the output wavelength increases [4]. Therefore, realistic wavelength converters

have limited wavelength conversion capability, and the difference between the input and output wavelengths is limited. It follows that we need a wavelength assignment scheme that considers the existence of limited-range wavelength converters [5].

Routing and wavelength assignment (RWA) is a very important issue since it decides the wavelength utilization efficiency and blocking probability. Many papers on routing and wavelength assignment have been published [5–7]. First-fit assignment has been researched as a wavelength assignment scheme for networks with limited-range wavelength converters [4]. Starting with the assumption that all nodes can get global information on all links in the network, first-fit assignment achieves almost the same blocking probability as a network with full-range wavelength converters. Global information about current network resources is effective in reducing the blocking probability since it allows us to more efficiently use network resources. However, it is not practical to share the global information on all links in real time to realize adequate scalability [8]. We need to assign wavelengths in a distributed manner based on information on neighboring links. If the nodes have limited-range wavelength conversion capability, wavelength assignment on a link limits the wavelengths assigned to the other links along the path. For this reason, it is expected that the blocking probability in networks with limited-range wavelength converters will increase compared to networks with full-range wavelength converters.

In this chapter, a distributed wavelength assignment scheme that considers the number of hops under the existence of limited-range wavelength converters is proposed. The concepts of our proposed scheme were presented in [9, 10]. In order to allocate different wavelengths as the search area for connection requests with different numbers of hops, the proposed scheme proceeds as follows. The proposed scheme identifies available wavelengths starting from a different initial area according to the number of hops in the request. That is, nodes start the wavelength search from a wavelength closer to the center wavelength when the path has large number of hops. On the other hand, when it has a small

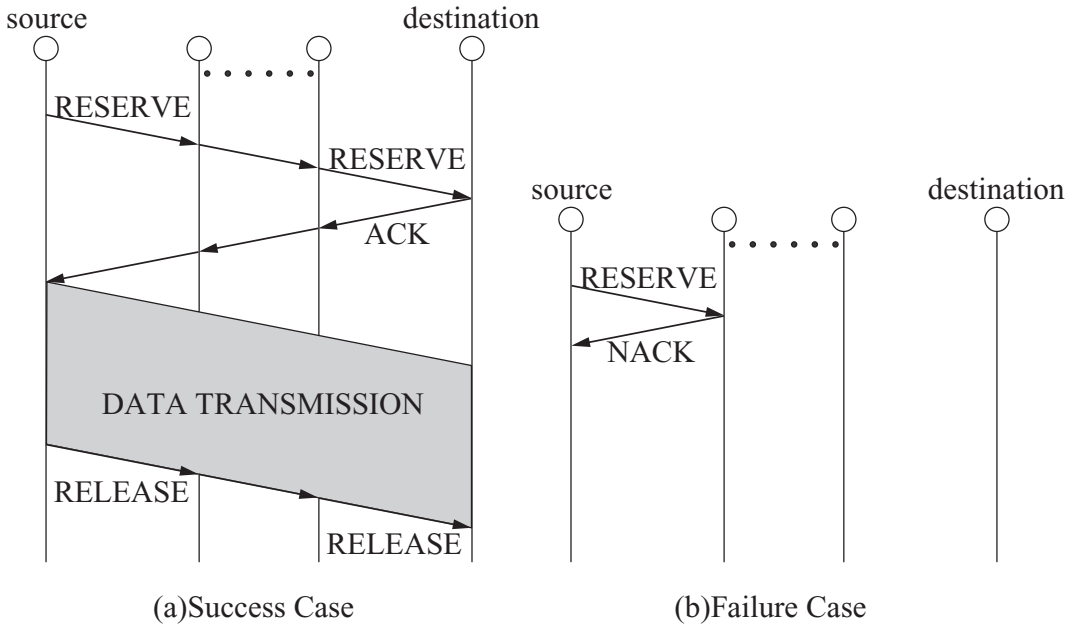


Figure 3.1: Forward reservation

number of hops, the search starts from a wavelength far from the center. This decreases the blocking probability and the number of wavelength conversions needed.

The rest of this chapter is organized as follows. Section 3.3 denotes the system model considered in this chapter. In Section 3.4, I propose a wavelength assignment scheme for wavelength-routed networks with limited-range wavelength converters. Section 3.5 presents the simulation results of the blocking probability and the average number of wavelength conversions. Finally, Section 3.6 concludes this chapter.

3.3 System model

3.3.1 Routing

In this chapter, I employ the simple shortest path routing, in order to focus on the effect of wavelength assignment schemes. It is assumed that OSPF (Open Shortest Path First) [11] is employed as the routing protocol. Each node exchanges packets describing

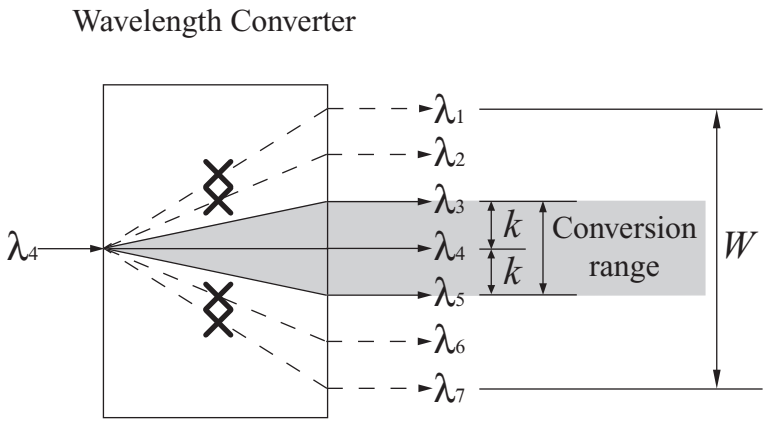


Figure 3.2: Limited-range wavelength Converter ($W = 7, k = 1, i = 4$)

its own adjacent link states using OSPF, and so obtains the topology of the network. Each node creates the shortest path tree, and also knows the number of hops of the paths to all other nodes in the network. Each source node selects the path based on its shortest path tree when a connection arrives.

3.3.2 Wavelength reservation protocol

Figure 3.1 illustrates an example of the wavelength reservation protocol. In this chapter, the forward reservation scheme is assumed. When a connection request arrives at a source node, the node determines the path used for the lightpath, and sends a RESERVE signal to the next node along the path to reserve a wavelength for the first link, as shown in Figure 3.1(a). When an intermediate node receives a RESERVE signal, it sends a RESERVE signal to the next node to reserve a wavelength based on the link state information. When a RESERVE signal reaches the destination node, it sends an ACK signal to the source node, which indicates the success of wavelength reservation on all links. The source node starts to send data after it receives the ACK signal. On completing data transmission, the source node sends a RELEASE signal toward the destination node along the path. A node receiving a RELEASE signal releases the wavelength on the lightpath used for the

data transmission. If there is a link with no wavelength available in a path, wavelength reservation fails. In this case, the node that failed to reserve a wavelength sends a NACK signal toward the source node as shown in Figure 3.1(b). Upon receiving the NACK signal, a node releases the wavelength and resends the NACK signal until the NACK signal reaches the source node.

3.3.3 Limited-range wavelength converter

A major problem of wavelength-routed networks is their inefficient utilization of the wavelength resource and the high blocking probability due to the constraint that the same wavelength must be used on all links along the path. This is known as the wavelength continuity constraint. The solution is the use of wavelength converters. A wavelength converter is a device that can convert the wavelength input to another wavelength that is then output. Full-range wavelength conversion can be achieved optoelectronically [2]. However, this approach loses one key advantage of wavelength-routed networks, optically-transparent processing. It is assumed that this weakness is not acceptable and focus our attention all-optical wavelength converters. Wavelength converters based on FWM (Four-Wave Mixing) are becoming extremely popular [4]. Unfortunately, as shown in Figure 3.3, the output signal strength of such a wavelength converter degrades as the difference between the input wavelength and the output wavelength increases. Current technologies restrict the range over which an optical wavelength conversion is possible. The relation between the input wavelength and the output wavelength is modeled as the following equation [4].

$$\lambda_{max(1,i-k)} \leq \lambda_o \leq \lambda_{min(W,i+k)} \tag{3.1}$$

where W wavelengths, λ_1 to λ_W, are multiplexed into an optical fiber, wavelength conversion range is k, the wavelength input to the wavelength converter is λ_i, and the wavelength

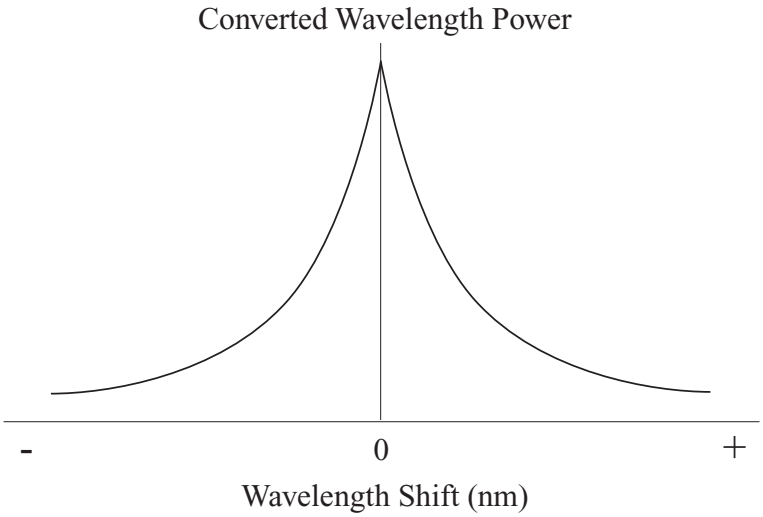


Figure 3.3: Converted wavelength signal power model

output by the wavelength converter is λ_o. Figure 3.2 illustrates an example where $W = 7$, $k = 1$, and $i = 4$. The output wavelength is limited to λ_2 to λ_4 for input wavelength λ_4.

3.3.4 First-Fit assignment

First-Fit assignment has been proposed as a wavelength assignment scheme for wavelength-routed networks with limited-range wavelength converters [4]. In this scheme, the smallest indexed available wavelength is assigned to a new connection request. The behavior of First-Fit assignment in our system model is expressed as follows. A node assigns the smallest indexed wavelength among all available wavelengths to a new connection request when it is a source node. When a node is intermediate node, it assigns the same wavelength assigned to the previous link if the wavelength is available. Otherwise, it assigns the smallest indexed wavelength among other wavelengths limited by Equation (3.1). Reference [4] shows that the blocking probability of First-Fit assignment is reasonable, but this approach assumes that each node can get the link states of the entire network. If this assumption is incorrect and the states of only the neighboring links is

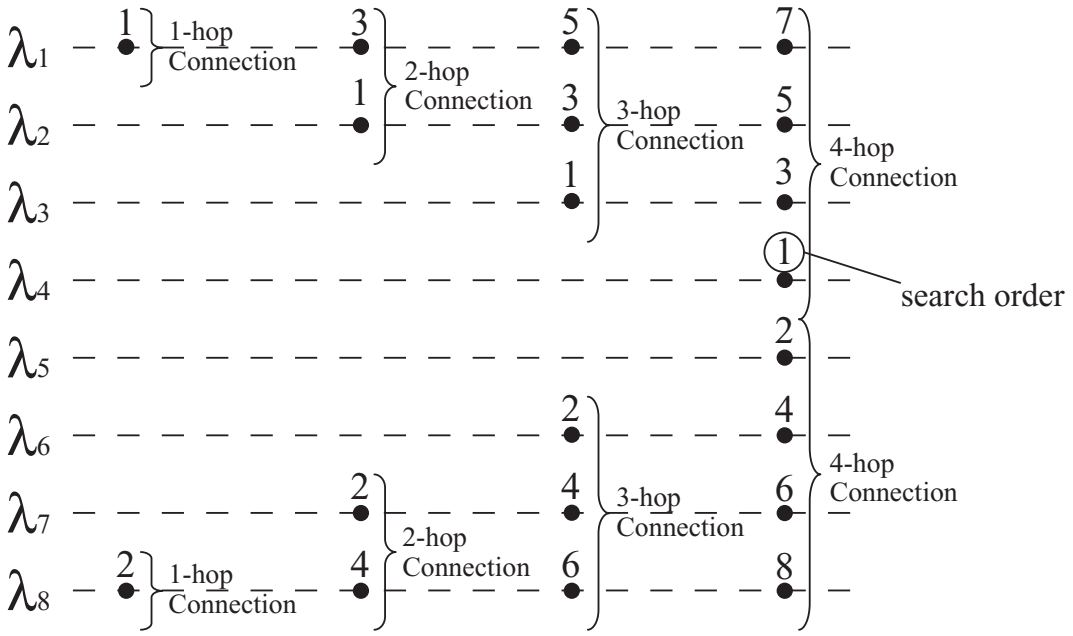


Figure 3.4: Search area and search order of the proposed scheme at the source node ($W = 8, H_{max} = 4$)

known, the blocking probability degrades dramatically.

3.4 Proposed scheme

In this chapter, I propose a wavelength assignment scheme that considers the number of hops in a connection. The blocking probability can be reduced by the proposed scheme since the wavelength search area depends on the number of hops in the connection. In the following, I denote the number of wavelengths as W, the maximum number of hops in a network as H_{max}, the wavelength conversion range is k, and the number of hops in a connection request as h. Furthermore, it is assumed each node knows the maximum number of hops in the network and that it can get adjacent link states. The source node can select any of the wavelengths in the search area, but intermediate nodes must select one the wavelengths that satisfy Equation (3.1) where the input wavelength is the wavelength

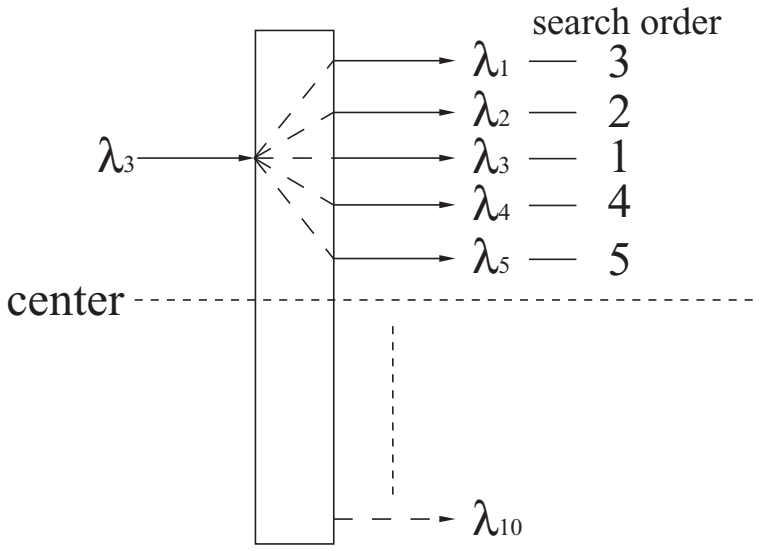


Figure 3.5: Search order of the proposed scheme at intermediate nodes ($W = 10, k = 2$)

assigned to the previous link. That is, wavelength assignment at the source node differs from that at an intermediate node.

3.4.1 Wavelength assignment at the source node

It is assumed that $W = 14$ and $k = 2$. When the input wavelength is λ_1, Equation (3.1) indicates that three wavelengths are possible, λ_1 to λ_3. When the input wavelength is λ_7, five wavelengths, λ_5 to λ_9, can be output. The number of output wavelengths possible varies with the input wavelength and the conversion range. When the input wavelength is near the center wavelength, the number of possible wavelengths is maximized. On the contrary, when it is distant from the center wavelength, the number is minimized. The necessity of wavelength conversion increases with the number of hops, as does the blocking probability. The goals of our proposed scheme are to reduce the blocking probability and the number of wavelength converters needed. Our approach is to vary the search area of available wavelengths with the number of hops; increasing the number of hops increases the search area. The initial search area is allocated from both ends of the wavelength

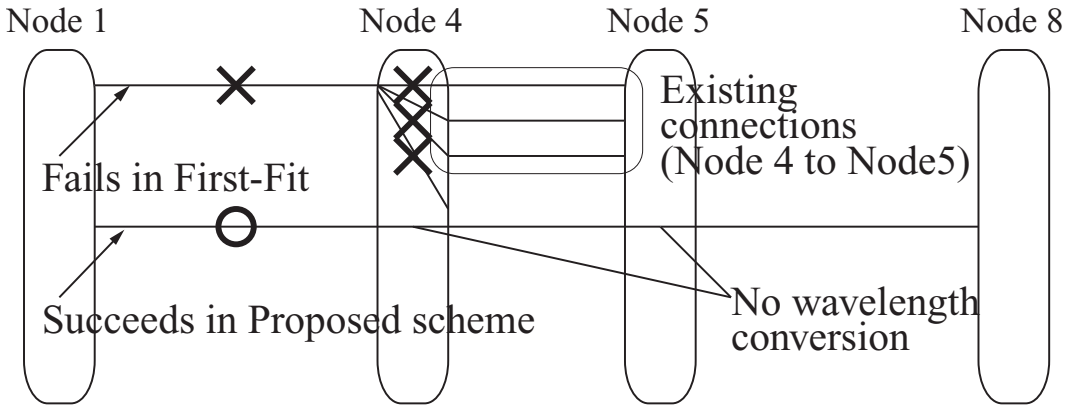
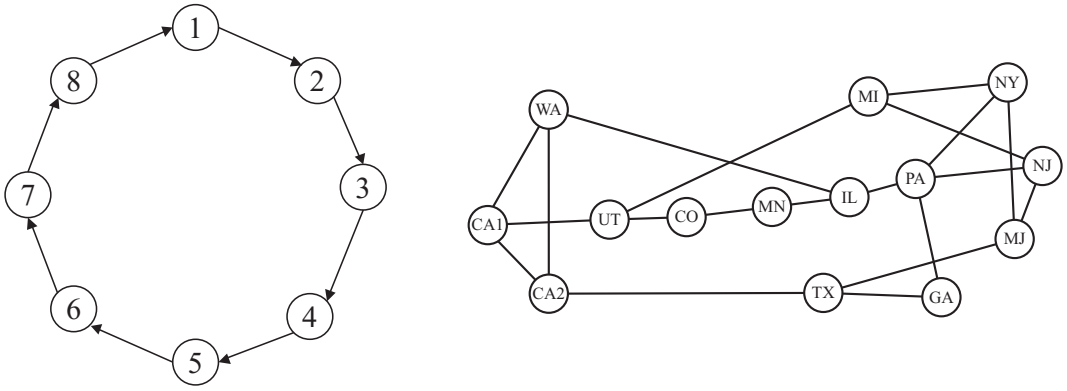


Figure 3.6: The effect of the proposed scheme

range. As a result, a connection request with a large number of hops tends to use wavelengths near the center wavelength. If there are vacant wavelengths in the search area, assignment starts from the wavelength nearest the center wavelength and the first available wavelength is assigned. In this way, connection requests that have many hops are more likely to be assigned wavelengths near the center wavelength. Requests with few hops tend to be assigned wavelengths far from the center. This procedure can improve the blocking probability of connections with many hops, which reduces the blocking probability of the entire network, as well as the number of wavelength converters needed.

Figure 3.4 shows the search area and the search order of our proposed scheme where $W = 8$, and $H_{max} = 4$. The numerals in this figure indicate the search order of wavelengths. As shown in Figure 3.4, λ_1 and λ_8 are the search area of a 1-hop connection. λ_1 to λ_3 and λ_6 to λ_8 are the search areas of a 3-hop connection. The search area contains as many wavelengths as there are hops from either end of the wavelength range. This symmetry makes the proposed scheme is effective regardless of the number of hops.



(a) 8-node unidirectional ring

(b) 14-node NSFNet

Figure 3.7: Network topology used in computer simulations

3.4.2 Wavelength assignment at intermediate nodes

The wavelength conversion range follows Equation (3.1) where the wavelength assigned to the previous link is λ_i. Figure 3.5 shows the search order of the proposed scheme at an intermediate node. If available, the input wavelength λ_i is assigned to the next link. Otherwise, we start the search from the wavelength most distant from the center wavelength. Next, the node checks the wavelength nearest the center. The first available wavelength is assigned to a connection request. In the example in Figure 3.5, the node searches for a wavelength in the order $\lambda_3, \lambda_2, \lambda_1, \lambda_4, \lambda_5$. As a result, fewer wavelengths near the center are selected which leaves them available for assignment to connections with many hops. Moreover, the number of wavelength conversions for a connection with large number of hops is also reduced. Consequently, the proposed scheme can reduce the impact of eliminating underutilized wavelength converters.

An example of the effect of the proposed scheme is shown in Figure 3.6. In this figure, it is assumed that the wavelength conversion range, k, is two, there are three existing connections between Node 4 and Node 5. When a new connection request from Node 1 to Node 8 arrives, in First-Fit assignment, we assign the smallest indexed wavelength.

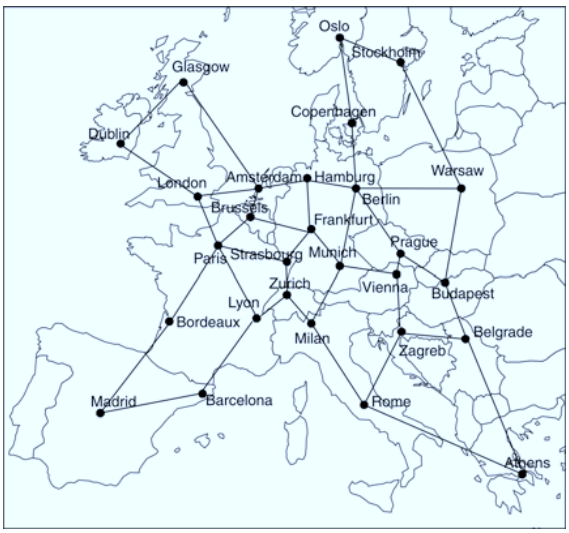


Figure 3.8: Pan European Network

Wavelength assignment fails because of existence of short-hop connections between Node 4 and Node 5. On the contrary, in the proposed scheme, we assign the center wavelength and the wavelength assignment succeeds. The center wavelength is available for the new connection, and no wavelength conversion is required at intermediate nodes in this example.

3.5 Performance evaluation

Computer simulations were conducted to evaluate the blocking probability and the average number of wavelength conversions on the 8-node unidirectional ring network and the 14-node NSFNet network in Fig. 3.7, and the 28-node Pan European Network [12] shown in Fig. 3.8. The number of wavelength $W = 14$ in the 8-node unidirectional ring network, $W = 8$ in the 14-node NSFNet network, and $W = 16$ in the 28-node Pan European Network. It is assumed that the connection requests arrive at each node independently following a Poisson process, and source-destination pairs were uniformly

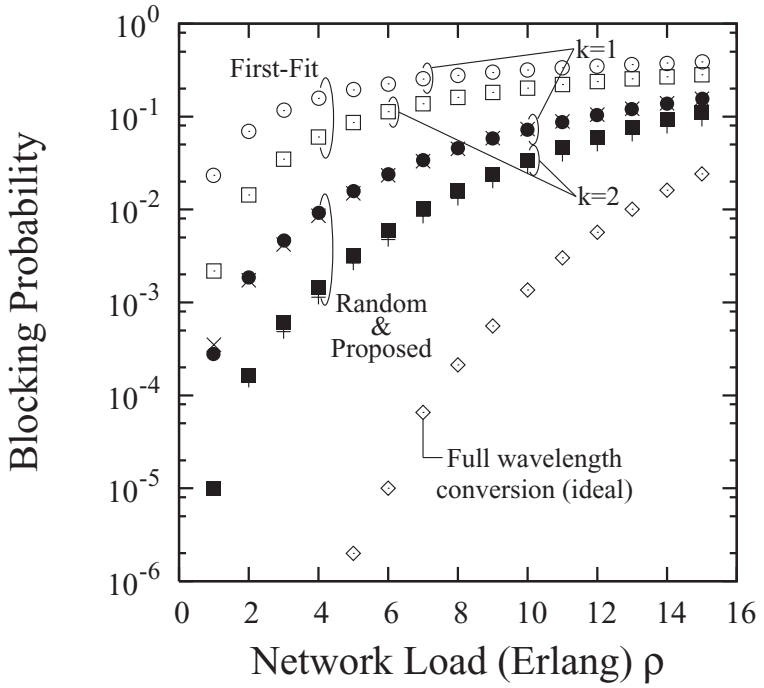


Figure 3.9: Blocking probability versus network load ρ on the 8-node unidirectional ring network

distributed. I compared three wavelength assignment schemes: First-Fit assignment, Random assignment, and the proposed scheme. Random assignment assigns a wavelength from available wavelengths randomly. It should have lower blocking probability than First-Fit assignment since it assigns wavelengths with uniform distribution, but the number of wavelength conversions is expected to be increased.

As the first step in examining the effectiveness of our wavelength assignment proposal, I evaluate the 8-node unidirectional ring network, which is a simple network topology as shown in Figure 3.7(a). Figure 3.9 shows the blocking probability versus the network load on the 8-node unidirectional ring network where each node has a limited-range wavelength converter. When wavelength converters have limited-range wavelength conversion capability, First-Fit assignment greatly increases the blocking probability compared to full-range conversion. The proposed scheme and Random assignment better suppress the

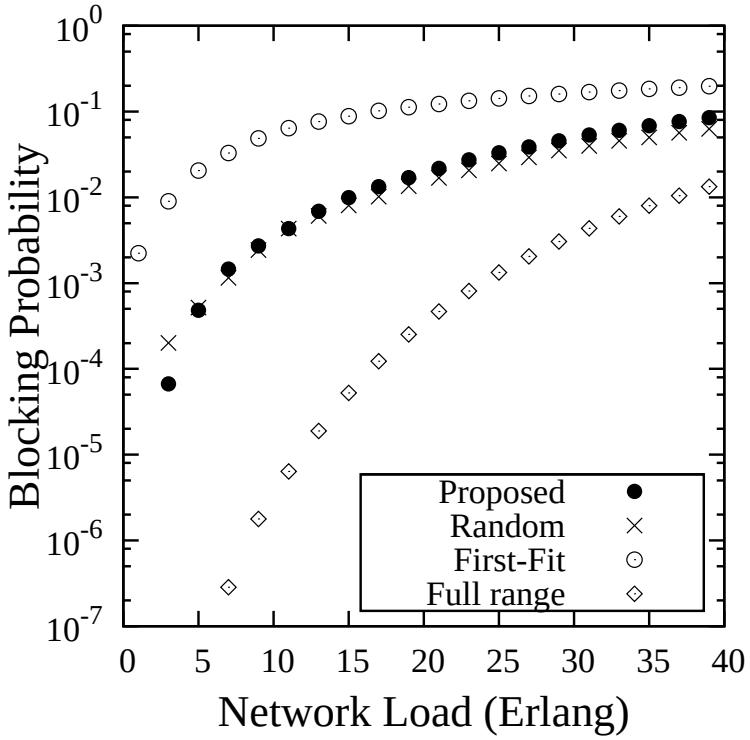


Figure 3.10: Blocking probability versus network load ρ on the 14-node NSFNet network

impact of limited-range wavelength conversion on the blocking probability than First-Fit assignment. If the wavelength conversion range, k, is large, the blocking probability is decreased. In the following evaluations $k = 1$ which is the most strict case for limited-range wavelength conversion. We see that the proposed scheme and Random assignment have almost same blocking probability. Given the above results, I compared the proposed scheme with Random assignment which has better blocking probability than First-Fit assignment.

Next, I evaluate the blocking probability on the 14-node NSFNet network shown in Figure 3.7(b) and the 28-node Pan European Network shown in Figure 3.8, which are more representative of real-world networks than a ring network. Figure 3.9 shows the blocking probability versus the network load on the 14-node NSFNet network. The result indicates the similar tendency observed for the 8-node unidirectional ring network. First-

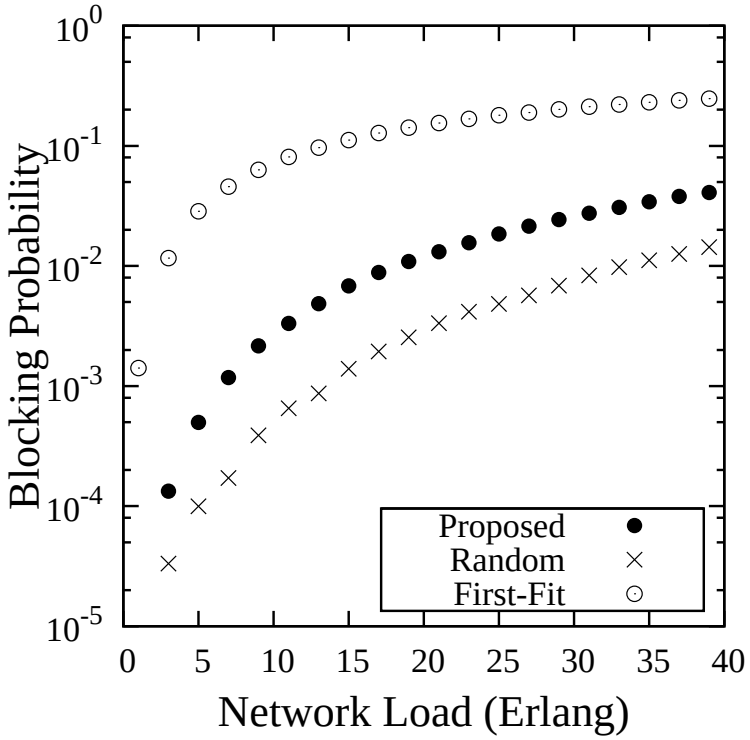


Figure 3.11: Blocking probability versus network load ρ on the 28-node Pan European Network

Fit assignment has higher blocking probability than either the proposed scheme or Random assignment, which have almost same blocking probability. The difference, however, shrinks at high loads. At high loads, the proposed scheme has slightly worse blocking probability than Random assignment. The reason for this is that the search area of the proposed scheme for a short hop path is less than that of Random assignment. This effect becomes significant only at high loads. We can reduce this effect by combining our proposed wavelength assignment with routing based on global information. In this chapter, however, I focus on wavelength assignment schemes. The combination is future work.

Figure 3.11 shows the blocking probability versus the network load on the 28-node Pan European Network. The blocking probability of First-Fit assignment is the worst among the three wavelength assignments. Random assignment and the proposal have

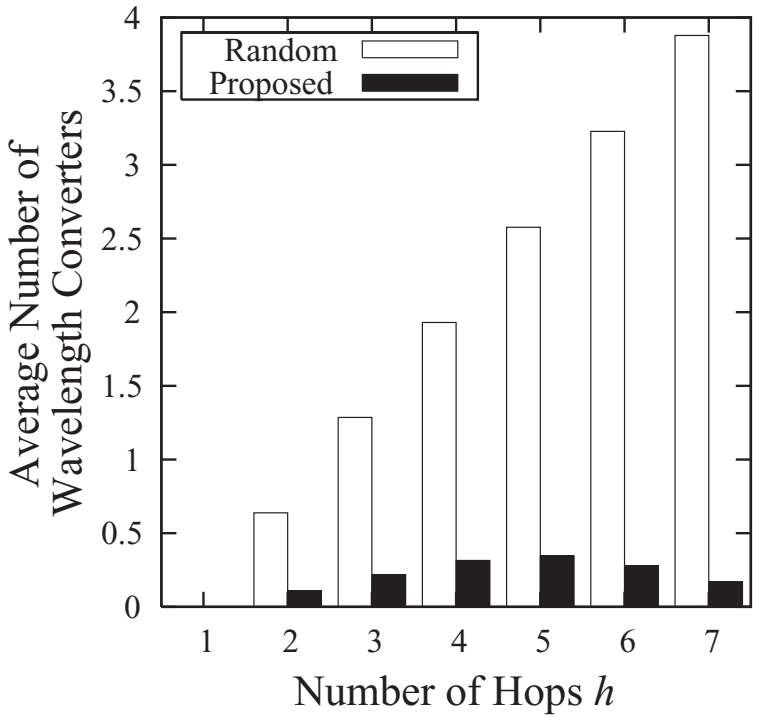


Figure 3.12: The average number of wavelength conversions needed versus the number of hops on the 8-node unidirectional ring network ($\rho = 7.0, k = 1$)

almost same blocking probability on the unidirectional ring and NSFNet. However, on the Pan European Network, there is difference between the blocking probability of Random assignment and the proposal. The difference is more apparent than in the case of the NSFNet. Some links tend to be more congested in the Pan European Network than the NSFNet. In the situation, the drawback of the proposal that the number of wavelength assigned for short hop connections is limited is significant. That's the reason why the proposal is worse than Random assignment.

Figure 3.12 and Figure 3.13 show the average number of wavelength conversions needed versus the number of hops. Figure 3.12 is the result of the 8-node unidirectional ring network, and Figure 3.13 is the result of the 14-node NSFNet network. This is done at the network load $\rho = 7.0$ in Figure 3.12, and $\rho = 11.0$ in Figure 3.12. It is assumed that each node has sufficient wavelength converters to handle all input wavelengths. The proposed

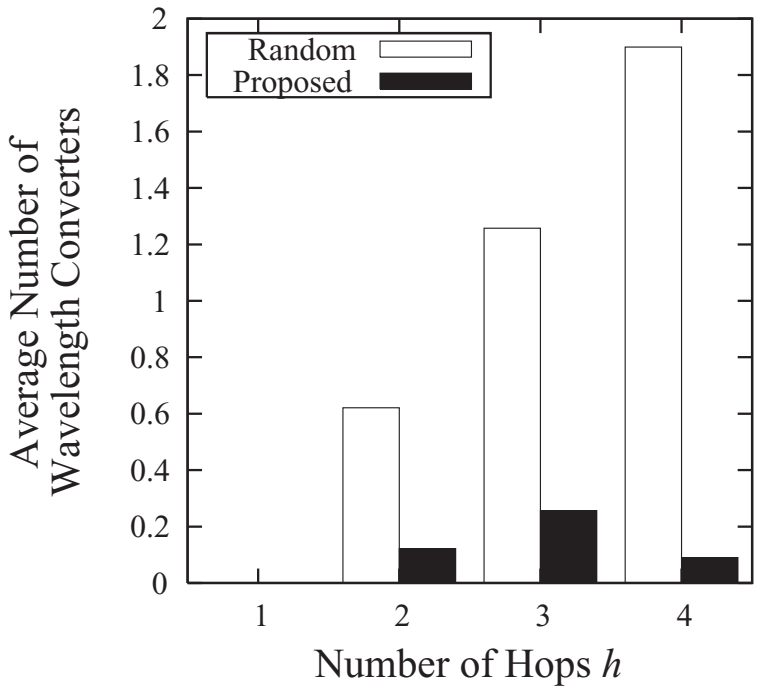


Figure 3.13: The average number of wavelength conversions needed versus the number of hops on the 14-node NSFNet network ($\rho = 11.0, k = 1$)

scheme requires fewer on average wavelength conversions than Random assignment in both networks. The key point is that proposed scheme makes it more likely that connections with many hops will undergo fewer wavelength conversions; this suggests that some wavelength converters will be underutilized.

Figure 3.14 and Figure 3.15 show the blocking probability versus the wavelength converter density on the 8-node unidirectional ring network and the 14-node NSFNet network, respectively. ”All” indicates all nodes have sufficient numbers of limited-range wavelength converters to handle all input wavelengths. ”Case 1”, ”Case 2”, and ”Case 3” represent the situations in which some wavelength converters are eliminated. In the proposed scheme, the utilization of wavelength converters whose input wavelength lies on the side or the center is lower than that of other wavelength converters. There is almost no difference in the blocking probability of the proposed scheme even though some

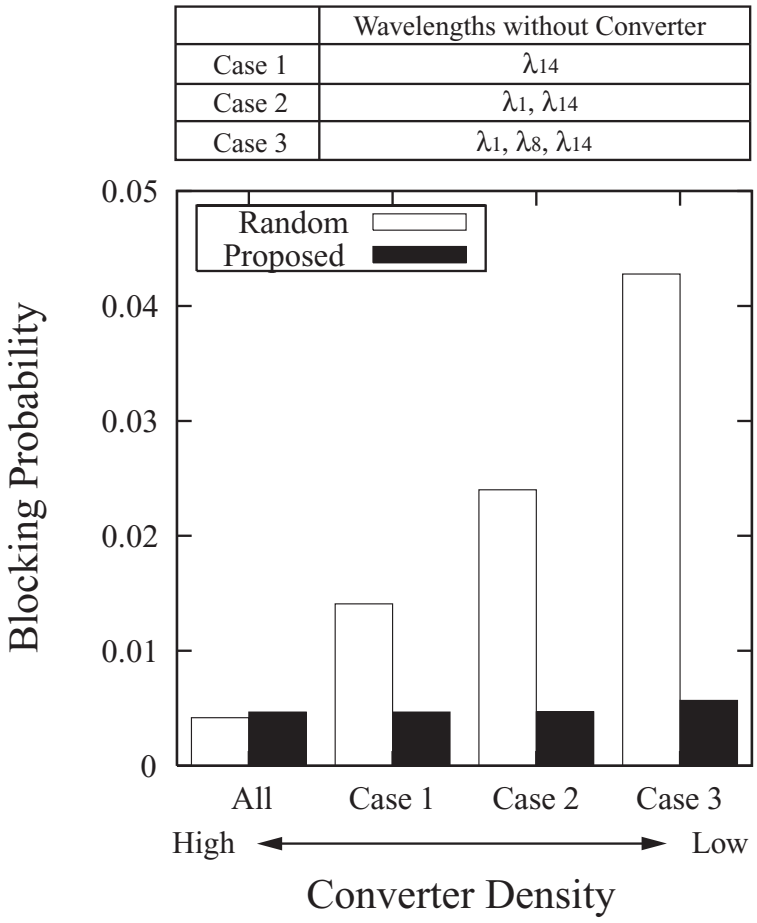


Figure 3.14: Blocking probability versus wavelength converter density ($\rho = 3.0, k = 1$) on the 8-node unidirectional ring network

wavelength converters were eliminated. On the other hand, Random assignment allowed the blocking probability to rapidly increase as the number of wavelength converters eliminated was increased. We can eliminate three of the fourteen wavelength converters in the 8-node unidirectional ring network in "Case 3" in Figure 3.14. This represents an improvement of about 20%. Moreover, we can eliminate three of the eight wavelength converters in the 14-node NSFNet network in "Case 3" in Figure 3.15. This yields an improvement ratio of 37.5%.

Figure 3.16 shows the increase ratio of the blocking probability versus the network load. I define the increasing ratio of the blocking probability as the ratio of the block-

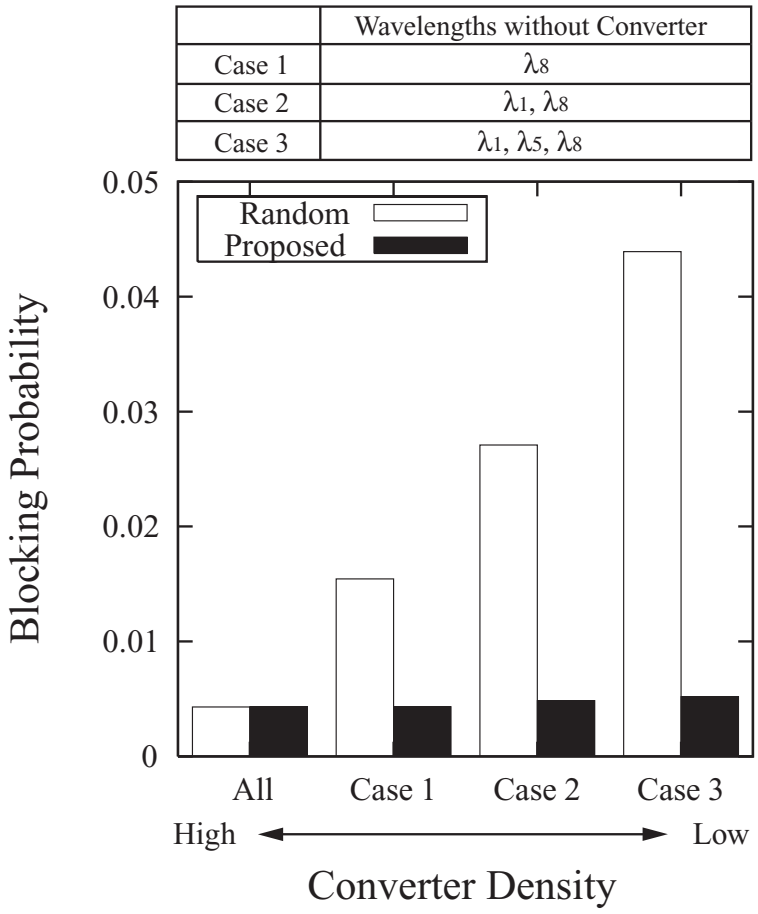


Figure 3.15: Blocking probability versus wavelength converter density ($\rho = 11.0, k = 1$) on the 14-node NSFNet network

ing probability with wavelength converters for all input wavelengths to that with fewer wavelength converters were eliminated. Eliminated wavelength converters correspond to $\lambda_1, \lambda_8, \lambda_{14}$ in the 8-node unidirectional ring network and $\lambda_1, \lambda_5, \lambda_8$ in the 14-node NSFNet network. It is found that the blocking probability is only slightly degraded in the proposed scheme when underutilized wavelength converters are eliminated. We also observe that the increase ratio decreases as the network load increases. At high network loads, the blocking probability is also high regardless of the existence of wavelength converters. In this case, the impact on the blocking probability of using wavelength converters is small. The difference in the increasing ratio in the proposed scheme is very small between

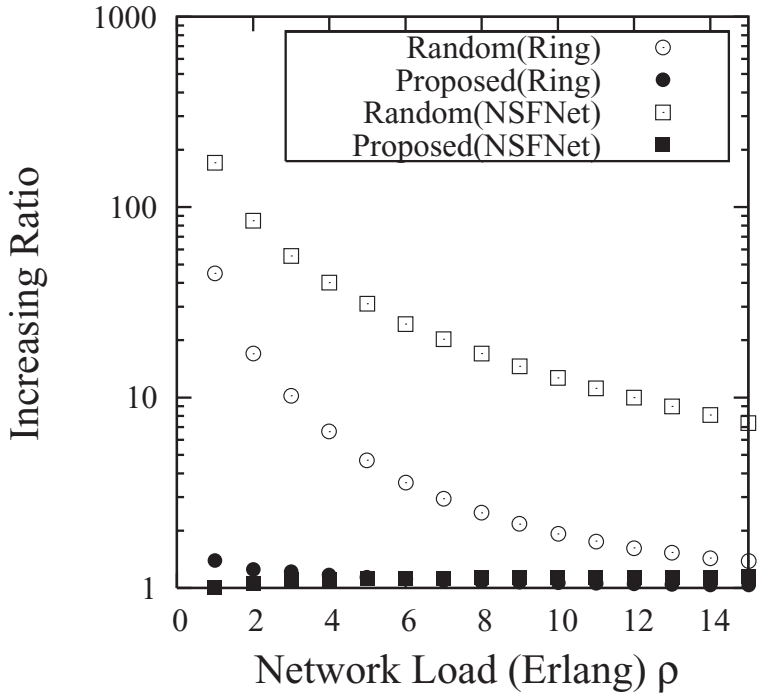


Figure 3.16: Increase ratio of blocking probability versus network load in case that three wavelength converters are reduced

these two networks while it is large in Random assignment. This shows that the proposed scheme has little dependency on the network topology when wavelength converters are reduced. All of the above results show that the proposed scheme is effective on the simple ring network, and also on practical networks like the NSFNet network.

3.6 Conclusion

This chapter introduced a wavelength assignment scheme for wavelength-routed networks with limited-range wavelength converters to realize high speed and large capacity transport. The proposal reduces the blocking probability and the number of wavelength converters with almost no performance degradation. It uses a center wavelength for a long hops connection and an edge wavelength for a short hops connection. First-Fit as-

signment does not consider wavelength conversion, and its blocking probability is high. The proposed scheme considers the number of hops in a connection request, and so offers lower blocking probability than First-Fit assignment. The computational simulations show the blocking probability of the proposal is almost equal to Random assignment on the unidirectional ring network and 14-node NSF network. Moreover, the proposal can reduce about 20 percent of the wavelength converters in the unidirectional ring network and 37.5 percent of the wavelength converters in 14-node NSFNet with almost no performance degradation. Therefore, the proposal realizes high speed and large capacity transport cost-effectively.

References

[1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical wan's," *IEEE Transaction on Communications*, vol. 49, no. 7, pp. 1171–1182, July 1992.

[2] M. Kovačević and A. Acampora, "Benefits of wavelength translation in all-optical clear-channel networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 868–880, June 1996.

[3] J. Zhou, N. Park, K. J. Vahala, M. A. Newkirk, and B. Miller, "Four-wave mixing wavelength conversion efficiency in semiconductor traveling-wave amplifiers measured to 65 nm of wavelength shift," *IEEE Photonics Technology Letters*, vol. 6, no. 8, pp. 984–987, Aug. 1994.

[4] J. Yates, J. Lacey, D. Everitt, and M. Summerfield, "Limited-range wavelength translation in all-optical networks," *IEEE INFOCOMM'96*, vol. 3, pp. 954–961, Mar. 1996.

[5] L. Zhang and L. Li, "Effects of routing and wavelength assignment algorithms on limited-range wavelength conversion in WDM optical networks," *IEEE 2002 International Conference on Circuits and Systems and West Sino Expositions*, vol. 1, pp. 860–864, June 2002.

[6] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," *SPIE Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, Jan. 2000.

[7] Y. Gong, P. Lee, and W. Gu, "A novel adaptive rwa algorithm in wavelength-routed network," in *Global Telecommunications Conference 2003 (GLOBECOM '03)*, vol. 5, Dec. 2003, pp. 2580–2584.

[8] C. Assi, Y. Ye, S. Dixit, and M. Ali, "Control and management protocols survivable optical mesh networks," *IEEE Journal of Lightwave Technology*, vol. 21, no. 11, pp. 2638–2651, Nov. 2003.

[9] S. Shimizu, Y. Arakawa, and N. Yamanaka, "A wavelength assignment considering the number of hops in limited-range wavelength-routed networks," in *Ninth International Symposium on Contemporary Photonics Technology (CPT 2006)*. Tokyo, Japan: NICT, Jan. 2006, pp. 104–105.

[10] ——, "Wavelength assignment scheme for WDM networks with limited-range wavelength converters," in *2006 IEEE International Conference on Communications (ICC 2006)*. Istanbul, Turkey: IEEE, June 2006.

[11] J. Moy, "OSPF version 2," *Request For Comments (RFC)*, no. 2328, Apr. 1998.

[12] S. D. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz, "Pan-european optical transport networks: An availability-based comparison," *Photonic Network Communications*, vol. 5, no. 3, pp. 203–225, May 2003.

Chapter 4

Scalable layer 2 network architecture using VLAN tag swapping

4.1 Abstract

Scalability is one of requirements for next generation backbone network. In this chapter, wide area Ethernet, which is an attractive transport technology for carrier recently, is focused on. VLAN tag swapping is proposed to extend the scalability of wide area Ethernet. In the conventional VLAN tagged Ethernet, a VLAN tag must be globally unique. On the other hand, in VLAN tag swapped Ethernet, a VLAN tag must be locally unique, and it can be reused in a different link. Therefore, the maximum number of VLAN paths is at least 4096 in the proposal. In addition, a prototype VLAN tag swapped Ethernet switch is implemented. The interoperability experiment is successfully conducted between two different implementations. The experiments confirms the feasibility of future scalable layer 2 network architecture with VLAN tag swapping.

4.2 Introduction

Wide area Ethernet is attractive for the next generation Internet backbone architecture, especially for carrier environments. This is because Ethernet is the most common networking technology and it's cost effective. In addition, the link bandwidth has been increasing: starting from 10 Mbps about 30 years ago, and reaching up to 100 Gbps nowa-

days. Ethernet became applicable to Wide Area Network (WAN) although it originated from Local Area Network (LAN) technology.

Providing an Ethernet Virtual Line (EVL) between customers is a basic service component. EVL is provisioned as an Ethernet Virtual LAN (VLAN) path. The Ethernet VLAN path can be established with VLAN technologies, especially with tag-based VLANs, which is standardized in IEEE 802.1Q [1]. The VLAN ID/tag is part of the Ethernet header. Wide area Ethernet using Ethernet Label Switching (ELS) [2] also forwards frames according to the value of its VLAN tag.

VLAN configuration of all switches along the Ethernet VLAN path is required when setting up or tearing down the path. Generalized Multi-Protocol Label Switching (GMPLS) [3] is a set of network control protocols to provide a next generation high performance transport network, and can be used for automatically configuring these switches in path provisioning. Therefore, we are proposing to employ GMPLS protocols for automatic Ethernet VLAN path provisioning. To increase the scalability of Ethernet, an automatic VLAN configuration technique is an important challenge especially for WAN.

In addition, the scalability of VLAN technology is an issue in wide area Ethernet. In the conventional VLAN tag-based Ethernet network (IEEE 802.1Q), a VLAN tag must be globally unique in a whole network. In other words, different Ethernet VLAN paths cannot reuse the same VLAN tag. In addition, only 12 bits are assigned to the field of VLAN tag. These imply that wide area Ethernet cannot support over 4096 Ethernet VLAN paths. That number of paths is not sufficient in WAN. We have proposed an effective network architecture to increase network scalability [4].

This paper presents experimental results of GMPLS controlled Ethernet VLAN path provisioning with VLAN tag swapping between Japan and Belgium. We successfully demonstrated the following things: 1) Interoperability between two different VLAN tag swapping based Ethernet implementations; one has been developed by Keio University,

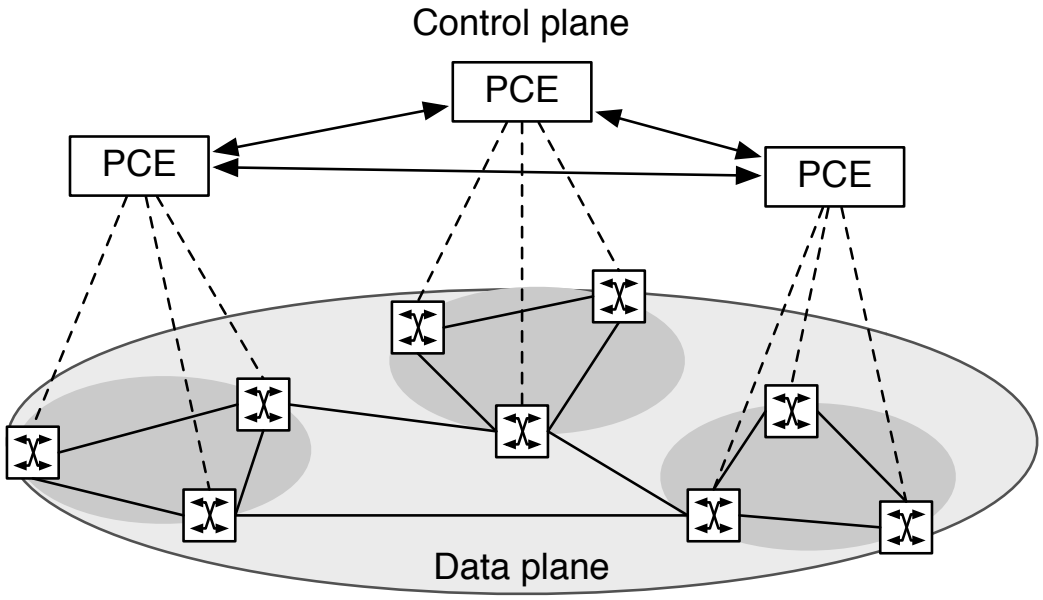


Figure 4.1: Centralized wide area Ethernet

and the other has been developed by Ghent University, 2) International Q-in-Q frame [5] transmission between Japan and Belgium, 3) High definition video streaming through the established Ethernet VLAN path.

4.3 Wide area Ethernet architecture

In this section, two types of wide area Ethernet architecture are discussed. One is a centralized model, and the other is a decentralized model.

Figure 4.1 shows the architecture of the centralized model. It has a Path Computation Element (PCE) based architecture. A PCE has responsibilities of resource management and path calculation in a domain. When an L2-LSP is requested, a layer-2 switch demands the PCE responsible for the corresponding domain to set up a path. The PCE calculates a path coordinating with PCEs in other domains. Finally, the PCE reserves an available VLAN tag for the new path.

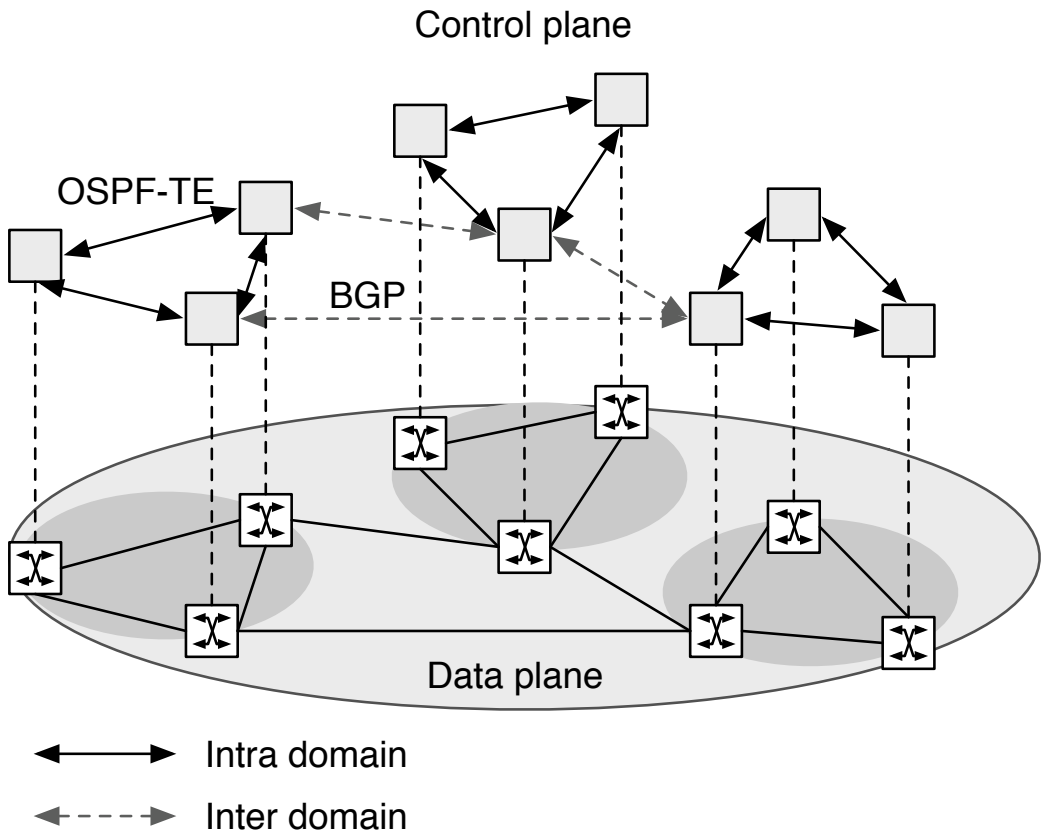


Figure 4.2: Decentralized wide area Ethernet

Figure 4.2 shows the architecture of the decentralized model. It employs GMPLS control protocols such as OSPF-TE, BGP, and RSVP-TE instead of PCEs. The resource information is distributed by OSPF-TE within a domain, and the information is shared among layer 2 switches in a domain. The resource information between other domains is advertised by BGP. When an L2-LSP is ready to set up, RSVP-TE signaling from the source switch is carried out towards the destination. A path is calculated in the source switch according to the current network information, and then an available VLAN tag is reserved as triggered by signaling.

The above architectures are compared. One of the advantages of the centralized model is complete management of network resources. In the centralized model, a PCE manages

all of network resources within the domain, and information about the current network resources can be received without delay. On the other hand, the lack of scalability is one of the largest drawbacks in the centralized model. Managing all network resources is a heavy task when the number of nodes in a domain increases. High performance PCEs are required. In addition, a PCE is single point of failure. The centralized model is weak against failures.

Therefore, the decentralized model is assumed in this paper. In this model, it is desirable that network resources are locally managed. However, a VLAN tag must be globally unique in the conventional network. To extend scalability of Ethernet, we have introduced VLAN tag swapping [4]. Figure 4.3 shows an example of VLAN tag swapping. The VLAN tag of an incoming frame is replaced with another VLAN tag for the corresponding outgoing frame. In this example, two configurations are stored in the forwarding table. The VLAN tag of the incoming frame is 100. It matches the first configuration of the forwarding table, therefore the VLAN tag of the outgoing frame becomes 200. In wide area Ethernet with VLAN tag swapping, a VLAN tag must be unique in a link, and the same VLAN tag can be reused in the other links (link local labeling). Therefore, the scalability increases, and the restriction of the number of connections is virtually eliminated.

Figure 4.4 shows the signaling sequence when the L2-LSP is established. There are 4 nodes, divided into 2 domains. Source routing is employed. First, Node A explicitly designates the path within Domain X and implicitly designates the path within Domain Y. A node checks availability of the VLAN tag of the incoming link and searches for an unused VLAN tag of the outgoing link when it receives a signaling message. Then, it manages a new mapping entry from label to VLAN tag. In the figure, Node B dynamically searches for an unused VLAN tag, then VLAN tag 200 is found as an unused VLAN tag. After all the entire procedure of establishing a new path is completed, data transmission

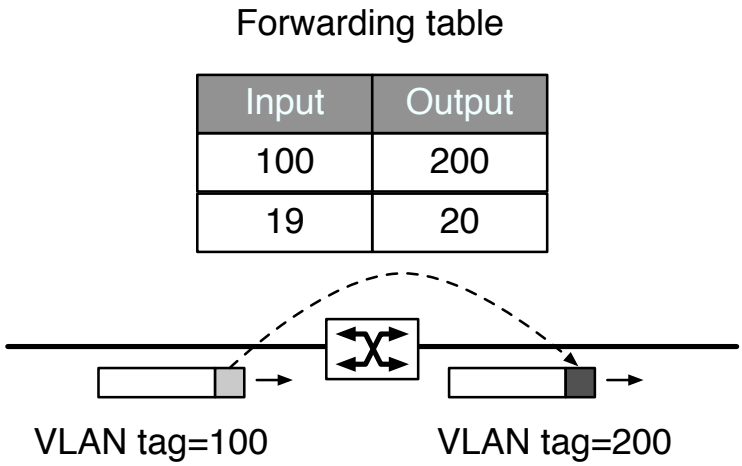


Figure 4.3: VLAN tag swapping

can happen through the VLAN tag swapped path.

4.4 Experiments

4.4.1 Experimental setup

Figure 4.5 shows the experimental setup of the international interoperability experiments between Japan and Belgium. There are 6 Ethernet switches (keio01, keio02, keio13, keio14, gent01, and gent02) and 2 end users (user01 and user02). The switches are controlled and configured by GMPLS protocols, and they contain VLAN tag swapping functionality. The data plane of all switches is based on the Click Modular Router framework [6]. The data plane of keio01, keio02, keio13, and keio14 is developed by Keio University, and that of gent01 and gent02 is developed by Ghent University. Figure 4.6(a) and Fig. 4.6(b) show the schematic diagram of the configurations of Click Modular Router in keio13 and keio01, respectively.

Two switches, keio13 and keio14, are placed in Keio University, Tokyo, Japan, and the

other 4 switches are placed in Ghent University, Ghent, Belgium. The switches in Japan work as edge switches and the switches in Belgium work as core switches. The ingress edge switch accepts untagged Ethernet frames and encapsulates them into Q-in-Q VLAN tagged frames for transmission to the core switches as shown in Fig. 4.6(a). Every core switch then swaps the outer VLAN (S-VID) to forward it towards its next hop. Finally the egress edge switch removes the Q-in-Q VLAN tag.

4.4.2 Path establishment

RSVP-TE establishes an L2-LSP between keio13 and keio14. Figure 4.7 shows the established L2-LSP in the experiment. The number below a link describes the value of the VLAN tag assigned to the link. For example, the VLAN tag of the link between keio13 and keio01 is 9, and that of the link between keio01 and keio02 is 10. In this case, the VLAN tag is swapped from 9 to 10 at keio01.

Each switch manages used tags and available tags. When reserving a path, each switch is looking for an available tag on the output port. In Fig. 4.7, it is assumed that the left port is port 0, and the right port is port 1 for all nodes. The first available tag of port 1 is used for a new L2-LSP at keio01. After the tag is assigned for a new L2-LSP, the configuration of the forwarding tables occurs automatically. In this case, the shaded entry (Input port, Input tag, Output port, Output tag) = (0, 9, 1, 10) is added. Similarly, all the other switches are configured, and finally the L2-LSP is established between keio13 and keio14.

4.4.3 High definition video transmission and numerical results

A high definition video stream was transmitted through an L2-LSP, which is established by RSVP-TE. Figure 4.10 shows the sender, the receiver of a high definition video

stream, and the edge switches, which are placed in Keio University, Japan. The sender corresponds to user01 in Fig. 4.5, and the receiver corresponds to user02 in Fig. 4.5. The core switches are located the ilab.t testbed in Ghent University, Belgium as shown in Fig. 4.11.

Figure 4.12 shows the experiment of transmitting high definition video. The video stream was captured by a video camera attached to the sender, and the TV monitor is attached to the receiver. The high definition video stream from the camera is transmitted through the established L2-LSP. The video is successfully displayed on the TV monitor at high definition quality as shown in Fig 4.12.

The round trip time and the UDP bandwidth between two users (user01 and user02) are measured. Figure 4.8 shows the round trip time measured by ping for 10 minutes. An ICMP packet is sent every 1 second from user01. The average round trip time is 575.5 msec and the standard deviation is 0.64 msec. This result indicates that tag swapping does not affect stability of the round trip time. Figure 4.9 shows the UDP throughput measured by iperf [7] for 10 minutes. user01 is the source node and the transmissions rate of the source node are 10Mbps, 15Mbps, and 20Mbps, respectively. The measured throughput is satisfactory for high definition video transmission.

The number of VLAN paths supported in the conventional system is at most 4096. On the other hand, our proposed system can support at least 4096 VLAN paths. The value varies with the number of hops of the paths through the network. 4096 VLAN tags can be used on each link in our proposed system, and there are 5 links in this experiment. The maximum number N of VLAN paths allowed in this experiment is expressed as follows:

$$N = \max(x), \text{ such that } \sum_{i=0}^{x} h_i \leq 4096 \times 5 \tag{4.1}$$

$$\approx \frac{4096 \times 5}{\bar{h}} \tag{4.2}$$

, where i is the index number of a VLAN path, h_i is the number of hops of the VLAN path i, and $\bar{h}$ is the average number of hops of the VLAN paths. The VLAN tag search time is

$O(1)$ with the number of nodes n since a hash table is used for the forwarding table in our implementation.

From this experiment, we verified the interoperability between two different VLAN tag swapping based Ethernet implementations. We successfully transmitted high definition video stream between Japan and Belgium with Q-in-Q frames. This result proves the feasibility of VLAN tag swapped based Ethernet, and proves that wide area Ethernet is realistic.

4.5 Conclusion

The scalability of Ethernet is a key issue in wide area layer 2 network because Ethernet originated from LAN technology. To cope with the issue, VLAN tag swapping is an effective solution. In this chapter, VLAN tag swapped Ethernet is proposed, and an interoperability experiment between two different implementations of VLAN tag swapped Ethernet in Japan and Belgium. In VLAN tag swapped Ethernet, the limitation of the number of virtual connections is practically removed because a VLAN tag can be reused in a different link. In the experiment, a GMPLS controlled L2-LSP was established between Japan and Belgium, and high definition video was transmitted through the L2-LSP. The interoperability of VLAN tag swapping and Q-in-Q frame transmission between the two countries are successfully demonstrated. The round trip time between two end users was 575.5 milli seconds and it was very stable. The UDP throughput is also satisfactory to transmit high definition video streaming. In addition, the maximum number of VLAN paths increases in the proposal. This result confirms that VLAN tag swapping is an effective solution to extend the scalability of wide area layer 2 network.

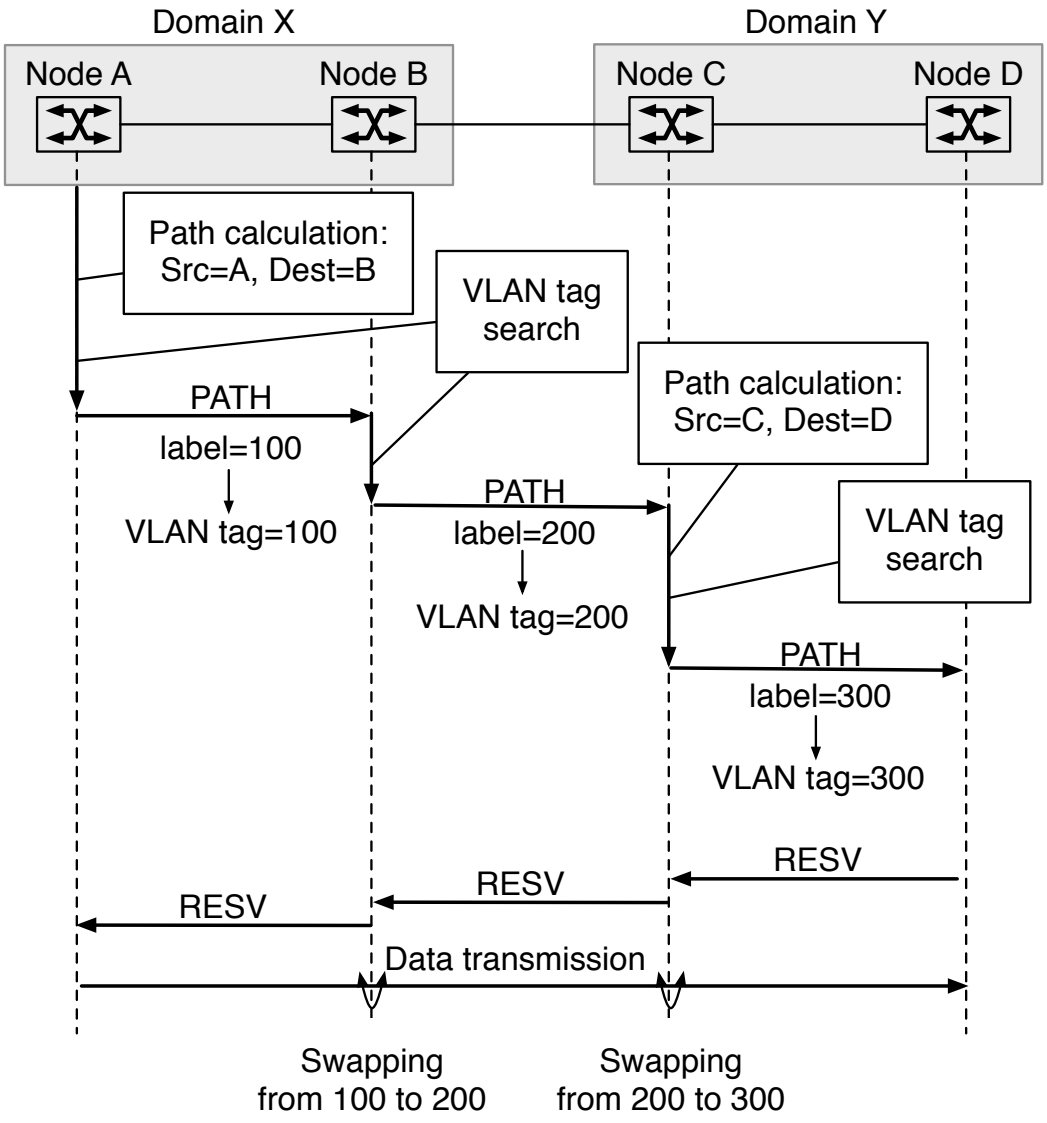


Figure 4.4: Signaling sequence of L2-LSP establishment

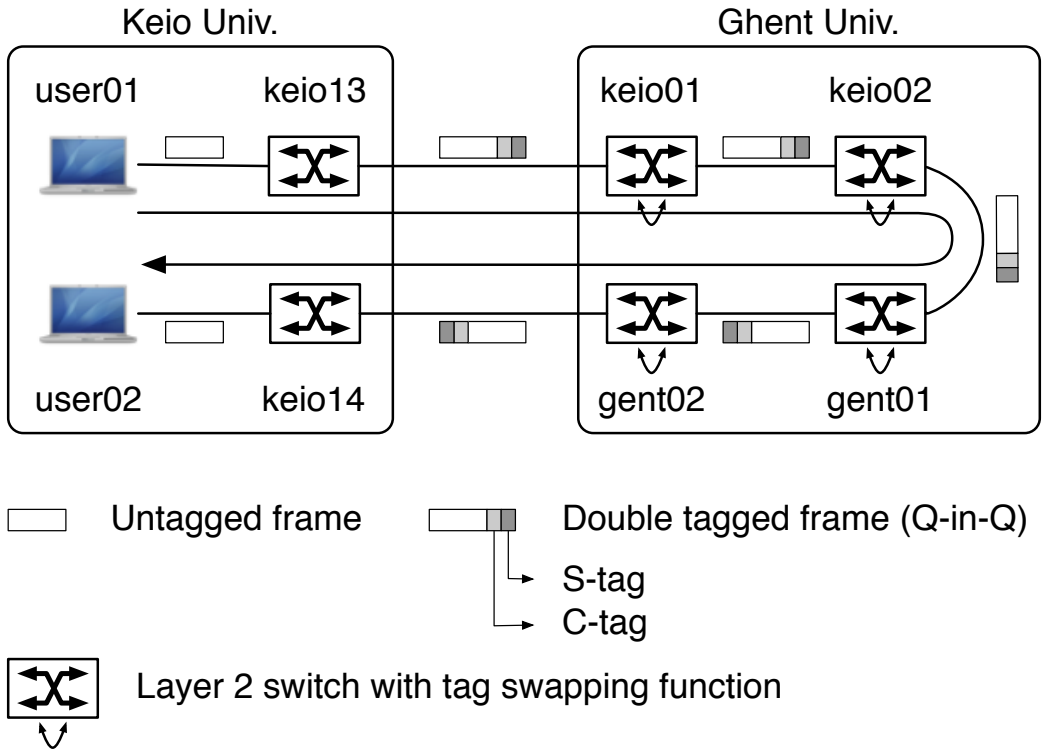
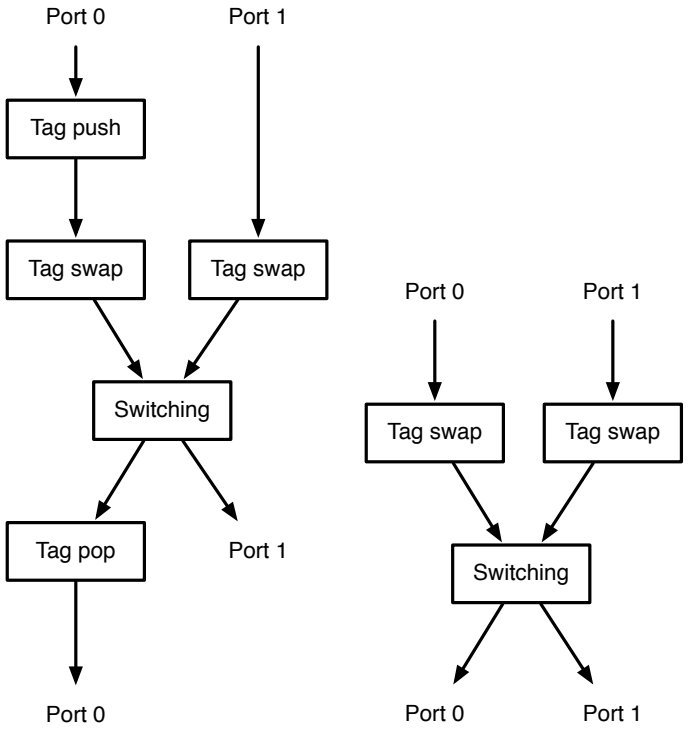


Figure 4.5: Experimental setup of demonstration



(a) keio13 (b) keio01

Figure 4.6: Click configuration

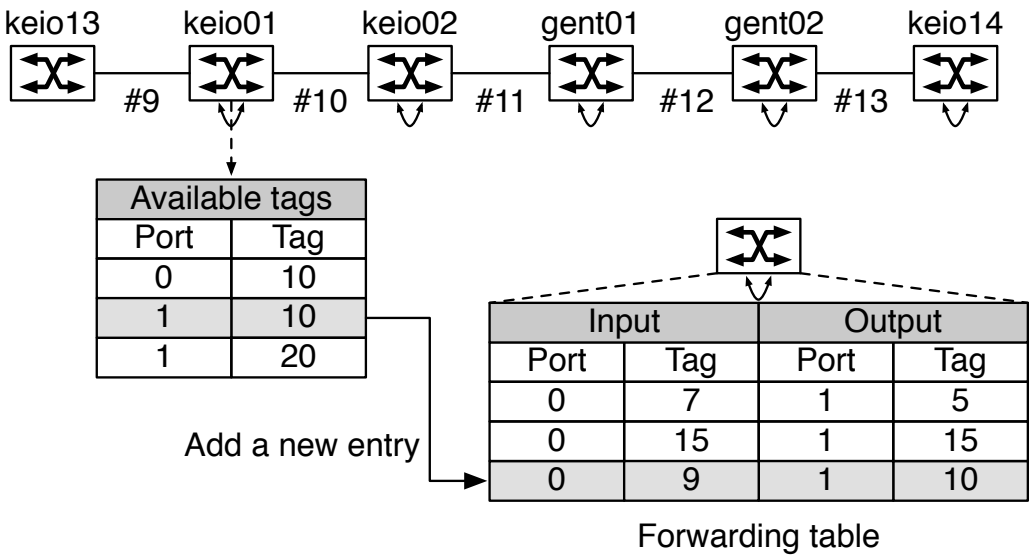


Figure 4.7: Changing configurations of a switch when signaling is occurred

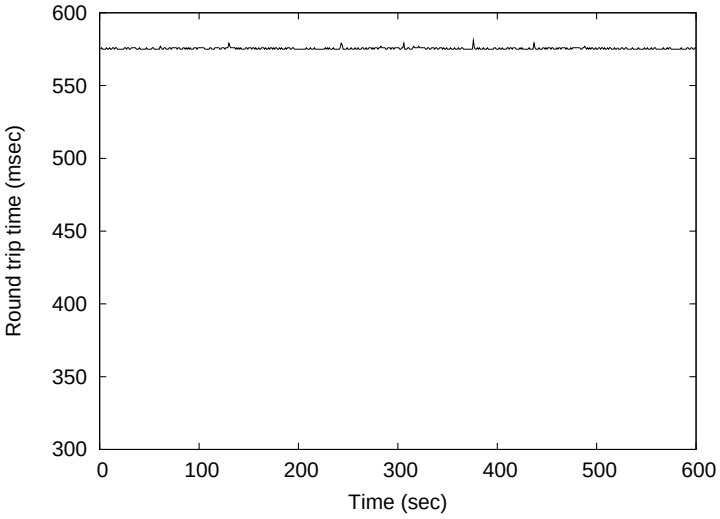


Figure 4.8: Round trip time between user01 and user02

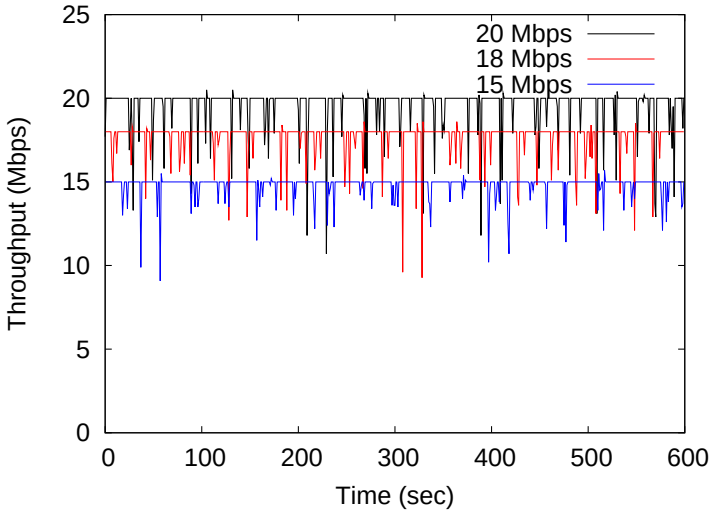


Figure 4.9: UDP throughput between user01 and user02

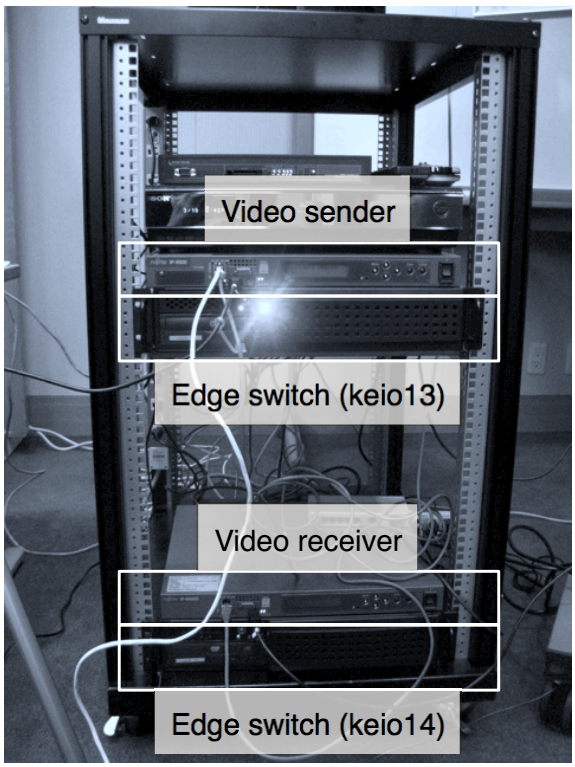


Figure 4.10: High definition video sender, receiver, and 2 edge switches placed in Keio University, Japan

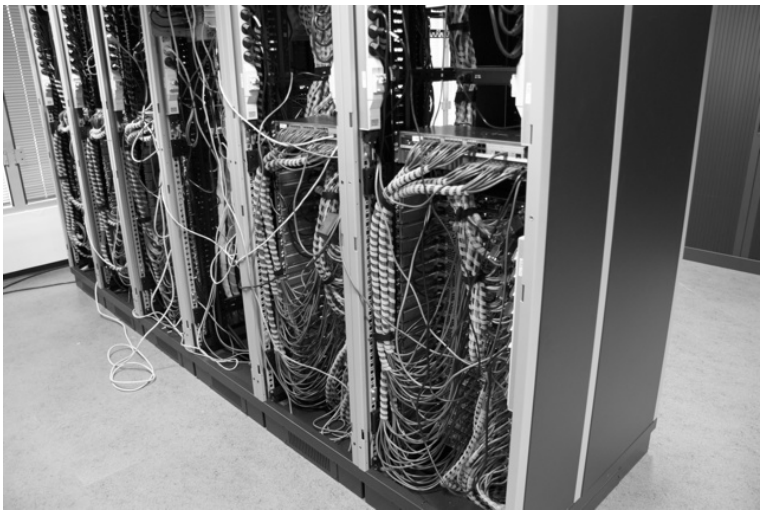


Figure 4.11: 4 Core switches placed in the ilab.t testbed in Ghent University, Belgium

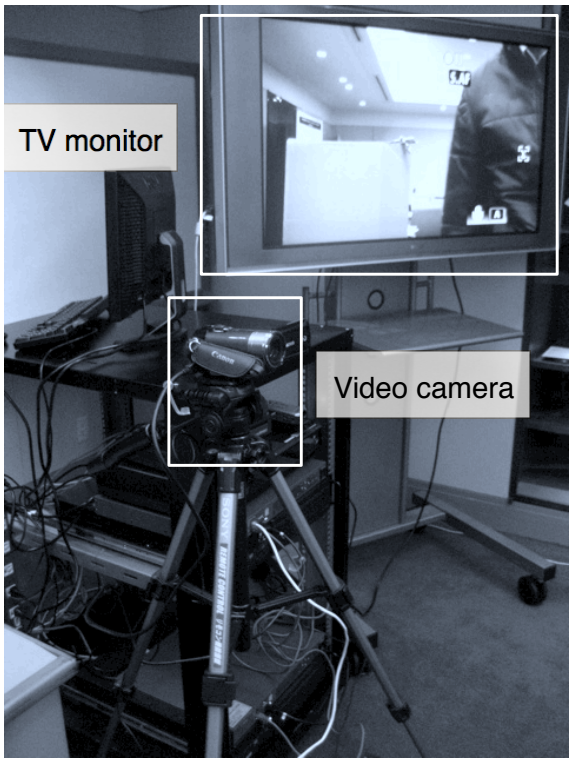


Figure 4.12: High definition video captured by video camera is displayed on TV monitor

References

[1] *IEEE Standards for Local and Metropolitan Area Networks Virtul Bridged Local Area Networks*, IEEE Standard 802.1Q, May 2006.

[2] W. Tavernier, D. Papadimitriou, D. Colle, M. Pickavet, and P. Demeester, "Emulation of GMPLS-controlled ethernet label switching," in *International Conference on Testbeds and Research Infrastructures for the Developement of Networks and Communities (TridentCom 2009)*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2009, pp. 1–9.

[3] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," *Request for Comments (RFC)*, no. 3945, Oct. 2004.

[4] K. Kikuta, M. Nishida, D. Ishii, S. Okamoto, and N. Yamanaka, "Establishment of VLAN tag swapped path on GMPLS controlling wide area layer-2 network," in *The Optical Fiber Communication Conference and The National Fiber Optic Engineers Conference 2009 (OFC/NFOEC 2009)*, Mar. 2009.

[5] *IEEE Standards for Provider Bridges*, IEEE Standard 802.1ad, Aug. 2005.

[6] "Click modular router," http://read.cs.ucla.edu/click/.

[7] "Iperf," http://iperf.sourceforge.net/.

Chapter 5

Parallel shortest path search algorithm for sophisticated traffic engineering

5.1 Abstract

Traffic engineering is an essential to utilize network resource efficiently and support QoS in next generation backbone network. The issue in sophisticated traffic engineering in next generation backbone network is high computational complexity of path calculation. A new parallel shortest path algorithm suitable for dynamically reconfigurable processor (DRP), called Multi-route Parallel Search Algorithm (MPSA), is proposed to speed up the shortest path calculation used in traffic engineering. In addition, a hardware off-loading engine is implemented on the actual DRP, DAPDNA-2. The proposed algorithm consists of simple processing, in which multiple paths are simultaneously searched by multiple Processor Elements (PEs) of DAPDNA-2. Therefore, it reduces the execution time of shortest path calculation to 2.8 percent compared with the popular shortest path algorithm, Dijkstra's algorithm. The proposed architecture and prototype system can be applied to future network sophisticated traffic engineering.

5.2 Introduction

Generalized Multi Protocol Label Switching (GMPLS) [1] is a key technology to control and manage next generation IP backbone networks. It enables not only high speed

and large capacity networks but also QoS controls and traffic engineering (TE). TE is an essential technique to utilize network resources efficiently and to avoid network congestion. Routing in GMPLS networks employing TE is based on multiple metrics such as the number of hops, link bandwidth, and transmission delay [2]. In current IP networks, shortest path routing which is based on interface cost only is often used [3]. Path calculation in GMPLS networks will be more complex due to the consideration of multiple metrics.

In addition, in GMPLS networks, topology of IP layer is affected by a lightpath in optical layer. Lightpath establishment leads to change topology of IP layer, and re-calculation of the shortest paths is essential. Therefore, each router frequently re-calculates the shortest paths to create a routing table in GMPLS networks. However, the conventional method for calculating the shortest path in OSPF [3] has $O(n^2)$ computational complexity where n is the number of nodes. Therefore, in large-scale networks, the complexity of the shortest path search become more difficult. Ultra fast shortest path calculation can adopt to huge-size networks.

Conventional approaches to calculate the shortest path is Dijkstra's algorithm [4] which is suitable for Neumann-type sequential processors and widely used, for example in creating routing tables in OSPF. The way to speed up Dijkstra's algorithm is to make clock cycle of a processor high since Dijkstra's algorithm is a sequential algorithm. However, it has limitations to speed up clock cycle because of the power consumption. To solve this problem, reconfigurable processors which is based on new architectures have been studied [5]. In this chapter, I employ a new approach that parallel shortest path algorithm is executed on reconfigurable processors to make the breakthrough for speeding up shortest path search. It is a software and hardware mixed approach.

I propose a parallel shortest path algorithm called MPSA (Multi-route Parallel Search Algorithm) based on parallel data-flow type dynamically reconfigurable processors. MPSA

is suitable for parallel data-flow machines since it can be expressed as matrix operations. MPSA searches for multiple paths in parallel. Current positions proceed by 1 cost unit with each step, and the first path to reach the target node is the shortest path to the node. After the current position reaches the target node, all links of the node are added to the discovered links. Results show the proposed algorithm is theoretically less execution time than Dijkstra's algorithm by about 97% since the proposed algorithm is $O(\sqrt{n})$ while Dijkstra's algorithm is $O(n^2)$.

I implement a hardware off-loading engine prototype to speed up the shortest path calculation in OSPF. MPSA is implemented as hard-wired logic on DAPDNA-2, which is a commercially available dynamically reconfigurable processor developed by IPFlex Inc., Japan [6]. Our prototype works together with GNU Zebra, a famous software-based router, running on commodity Linux PC. When Zebra creates a routing table in the process of OSPF, our off-loading engine calculates the shortest path.

5.3 Related works

5.3.1 Shortest path algorithm

Let $G = (V, E)$ be a directed graph where V is the set of nodes, and E is the set of edges. Let $|E| = m$, $|V| = n$, let s be the source node, and c be a function assigning a non-negative valued weight to each edge of G. The cost of a link can be thought of as the distance between the two nodes. For each $v \in V$, $d(v)$ represents the cost of the shortest path from the source node s to v. The theoretically most efficient sequential shortest path algorithm is Dijkstra's algorithm [7]. It calculates the shortest path between the source node and all other nodes. It is expressed as follows.

1. Set S to empty, where S is a set of nodes whose shortest paths from the source node s have already determined.

2. Add the source node s to S, and $d(s) = 0$. If there is a link from s to v, $d(v) = c(s, v)$, for all other nodes, $d(v) = \infty$.

3. Add a node u to S, where $d(u)$ is the smallest in $V - S$. If $S = V$, complete the algorithm.

4. If there is a link from u to $v \in V - S$, $d(v) = \min\{d(v), d(u) + c(u, v)\}$. Then go to 3).

Routing in IP networks is based on the shortest paths, a router calculates the shortest paths when it creates a routing table. Dijkstra's algorithm is suitable for Neumann-type processors, ant it is employed as the shortest path algorithm in the real routers, for example GNU Zebra [8] and XORP (eXtensible Open Router Platform) [9]. The computational complexity of Dijkstra's algorithm is $O(n^2)$ where n is the number of nodes in a network. Therefore, in large-scale networks with TE functionalities, Dijkstra's algorithm become heavy task for a router.

5.3.2 Reconfigurable processor

It has limitations to speed up clock cycle of processors because of the power consumption. To cope with the problems about power consumption, performance, and rapid development, dynamically reconfigurable processors have been developed. Recent dynamic reconfigurable devices have been developed to achieve high performance and flexibility [5].

DAPDNA-2 is a commercially available dynamically reconfigurable processor developed by IPFlex Inc., Japan [6]. DAPDNA-2 consists of two processors, DAP (Digital Application Processor) and DNA (Digital Network Architecture). These processors have different architectures. DAP is a 32-bit RISC CPU which is a Neumann-type processor, and DNA is a parallel data-flow machine. DNA consists of 376 small computing unit called PEs (Processor Elements), which is arranged in a array pattern. DAPDNA-2 is not

a complicated processor from the architectural point of view, and the size of the chip is smaller than Pentium 4. Therefore, in principle, the price of DAPDNA-2 is less than that of Pentium 4.

We can design the connections between PEs when implementing an algorithm on DAPDNA-2. Connection structure of PEs can yield a parallel data-flow machine. Each structure is called a configuration. DNA can keep three configurations in own cache. These configurations can be switched within one clock. Thus, this chip combines the advantages of the high-speed processing of hardware and the flexibility of software.

5.4 Multi-route parallel search algorithm

5.4.1 Summary

The basic idea of our proposed algorithm is that the shortest path is the first path reached to a node when we traverse all edges from the source node at the same pace. First, the current positions are set at the source node. After this, all the current positions simultaneously proceed by 1 cost unit per process. All the current positions are equally distant from the source node. When the current position reaches a node, we find that the path on which we traverse is the shortest path to the node.

Here, the following notations are used to explain the algorithm.

V_f The set of nodes whose shortest paths from the source node have been already determined.

V_r The set of nodes which is just reached.

E_t The set of edges which are presently traversed.

E_f The set of edges which are determined that they are on the shortest path tree.

i, j A node in the network

$r(i, j)$ The remaining cost to reach Node when (i, j) is traversed where $(i, j) \in E$.

The following is an accurate procedure of our proposed algorithm.

1. Set V_f, E_t, and E_f to empty. Set $V_r = \{s\}$, and $r(i, j) = c(i, j)$ for $(i, j) \in E$, $r(i, j) = 0$ for $(i, j) \notin E$.

2. Add node $i \in V_r$ to V_f. If V_f is equal to V, complete the algorithm.

3. Remove the edge (i, j) from E_t, and set $r(i, j)$ to zero for all i where $j \in V_r$. Add the edge (i, j) to E_t, where $(i, j) \in E$, $i \in V_r$, and $r(i, j) \neq 0$. Set V_r to empty.

4. $r(i, j) = r(i, j) - 1$ for all the edge $(i, j) \in E_t$. If $r(i, j)$ become 0, add node j to V_r. Also add the edge (i, j) to E_f. Then go to 2).

Figure 5.1 shows an example of our proposed algorithm. The denominator represents $c(i, j)$ and the numerator represents $c(i, j) - r(i, j)$ in edge (i, j). First subfigure is the initial state of the proposed algorithm. First, the algorithm adds edge $(1, 2)$ and $(1, 3)$ to E_t. Next, the current position proceeds by 1 cost unit. Then $r(1, 2) = 0$ and $r(1, 3) = 2$, so edge $(1, 2)$ reaches node 2. Node 2 is added to V_f, and edge $(1, 2)$ is added to E_f. In the next step, edge $(2, 3)$, and $(2, 4)$ are added to E_t. After the current positions are proceeded, $r(1, 3) = 1$, $r(2, 3) = 3$, and $r(2, 4) = 4$. In this step, no node are reached. The algorithm runs similarly until all shortest paths are determined, $V_f = \{1, 2, 3, 4\}$.

5.4.2 Matrix representation

Network topology can be expressed as a matrix $A = a_{ij}$ whose size is $n \times n$. It has row and column corresponding to every node, and its ijth entry a_{ij} equals the cost of the edge (i, j) if $(i, j) \in E$, otherwise 0. The matrix is called the node-node adjacency matrix,

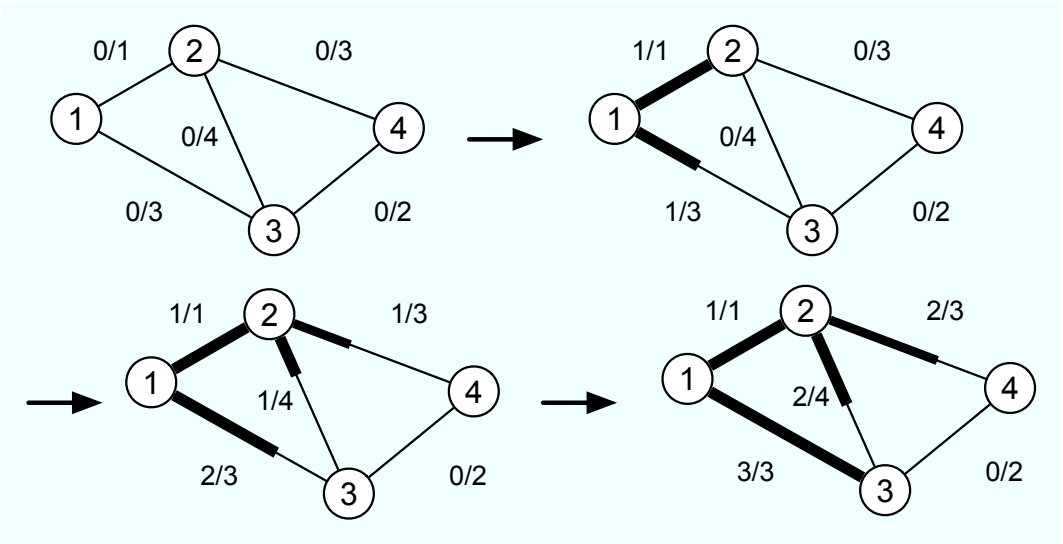


Figure 5.1: Our proposed algorithm finds the shortest paths by simultaneous multi-path search

or simply called the adjacency matrix. Equation 5.1 shows the adjacency matrix for the network shown in Fig. 5.1. In Fig. 5.1, the edge between node 1 and node 2 is 1, so a_{12} and a_{21} is 1.

$$A = \begin{pmatrix} 0 & 1 & 3 & 0 \\ 1 & 0 & 4 & 3 \\ 3 & 4 & 0 & 2 \\ 0 & 3 & 2 & 0 \end{pmatrix} \tag{5.1}$$

MPSA can be expressed as matrix representation. Matrix representation is space-inefficient, but suitable for hardware implementation. In addition, matrix operations offer high parallelism since the operations of the elements are independent. For these characteristics, MPSA is expressed as matrix-based operations to implement it on DAPDNA-2.

Matrix representation version of MPSA uses five matrices, called network matrix N, search matrix S, fix matrix F, reach matrix R, and path matrix P. Here, it is assumed that an element is an l-bit unsigned integer data.

The followings are definitions of these matrices. Network matrix $N = n_{ij}$ corresponds

to $r(i, j)$. For example, if edge (i, j) is now traversed and 5 cost units remain to reach node j, $n_{ij} = 5$. Search matrix $S = s_{ij}$ is a matrix representation of E_t. If edge (i, j) is not now traversed, $s_{ij} = 0$. Otherwise, $s_{ij} = 2^l - 1$. This number is the maximum unsigned integer value for l-bit data, and all of its bits are 1. In the later, I denote $2^l - 1$ as MAX_VALUE. Fix matrix $F = f_{ij}$ expresses V_f. In the case that node $j \in V_f$, f_{ij} = MAX_VALUE for all $i \in E$. On the contrary, $j \notin V_f$, $f_{ij} = 0$ for all $i \in E$. Reach matrix $R = r_{ij}$ is a matrix representation of V_r. Like fix matrix F, r_{ij} = MAX_VALUE for all $i \in E$ when $j \in V_r$, and $f_{ij} = 0$ for all $i \in E$ when $j \notin V_r$. Finally, path matrix $P = pij$ corresponds to E_f. If edge $(i, j) \in E_f$, p_{ij} = MAX_VALUE, otherwise $p_{ij} = 0$.

In addition, I denote the matrix operations as follows where $X = x_{ij}$ and $Y = y_{ij}$ is matrices. And $Z = z_{ij}$ denotes the result of each operation.

- TRANS(X)

  Transpose X, $z_{ij} = x_{ji}$ for all i, j.

- OPEA(X)

  Called Operation A later. If $x_{ij} \neq 0$, z_{ij} = MAX_VALUE. Otherwise, $z_{ij} = 0$.

- OPEB(X)

  Called Operation B later. If $x_{ij} \neq 0$, $z_{ij} = 1$. Otherwise, $z_{ij} = 0$.

- OPEC(X)

  Called Operation C later. z_{ij} is the result of bitwise OR operation for all element in column j. Figure 5.2 shows an example of Operation C.

Using above matrices and operations, MPSA can be expressed as follows. It is assumed that $T_x = tx_{ij}$ is a temporary matrix used in the following description.

1. Set fix matrix F, search matrix S, and path matrix P to zero matrix. Reach matrix R is set to be r_{is} = MAX_VALUE for all $i \in E$ where s is the source node. Set

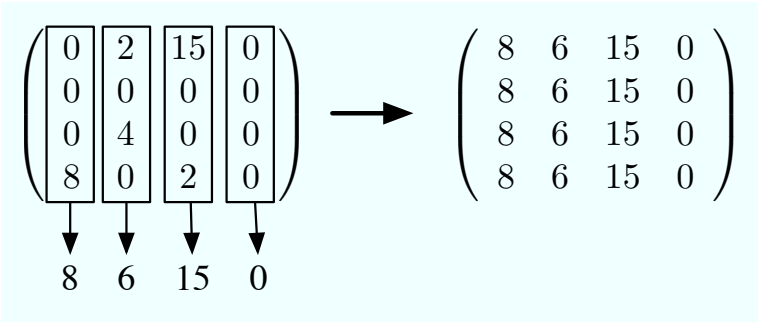


Figure 5.2: Example of Operation C where it is assumed data length of an element is 4 bits

network matrix N to the adjacency matrix representing the network topology and edge weight, i.e. $n_{ij} = c(i, j)$ if $(i, j) \in E$, otherwise $n_{ij} = 0$.

2. $F = \mathrm{OR}(F, R)$. If $f_{ij} = \mathrm{MAX\_VALUE}$ for all i, j, complete the algorithm.

3. $T_1 = \mathrm{AND}(\mathrm{NOT}(R), S)$, $T_2 = \mathrm{AND}(\mathrm{NOT}(R), N)$. These operations is to set all elements in column i to zero where $i \in V_r$. $T_3 = AND(\mathrm{TRANS}(R), T_2)$, and set $T_4 = \mathrm{OPEA}(T_3)$. $t3_{ij} = \mathrm{MAX\_VALUE}$ if node i is the reached nodes. Adding (i, j) to V_t in 3) of Section 5.4.1 corresponds to $S = \mathrm{OR}(T_1, T_4)$. Set $T_5 = \mathrm{NOT}(S)$.

4. $N = \mathrm{SUB}(T_2, \mathrm{OPEB}(S))$. If n_{ij} become 0, it indicates that node j is just reached. To pick up edge (i,j) where node j is just reached, set $T_6 = \mathrm{NOT}(\mathrm{OPEA}(\mathrm{OR}(T_5, N)))$. If edge (i, j) is on the shortest path tree, $t6_{ij}$ is MAX_VALUE. Finally, $P = \mathrm{OR}(P, T_6)$, and $R = \mathrm{OPEC}(T_6)$. Then go to 2).

Figure 5.3 shows the data-flow of above matrix representation version of MPSA. MPSA consists of the simple matrix operations, AND, OR, NOT, SUB, TRANS, OPEA, OPEB, and OPEC. They are suitable for hardware implementation because of their simplicity. In addition, their elements are independent except for TRANS and OPEC. By assigning a PE to a operation of an element, we can take advantage of parallel processing of dynamically reconfigurable processors.

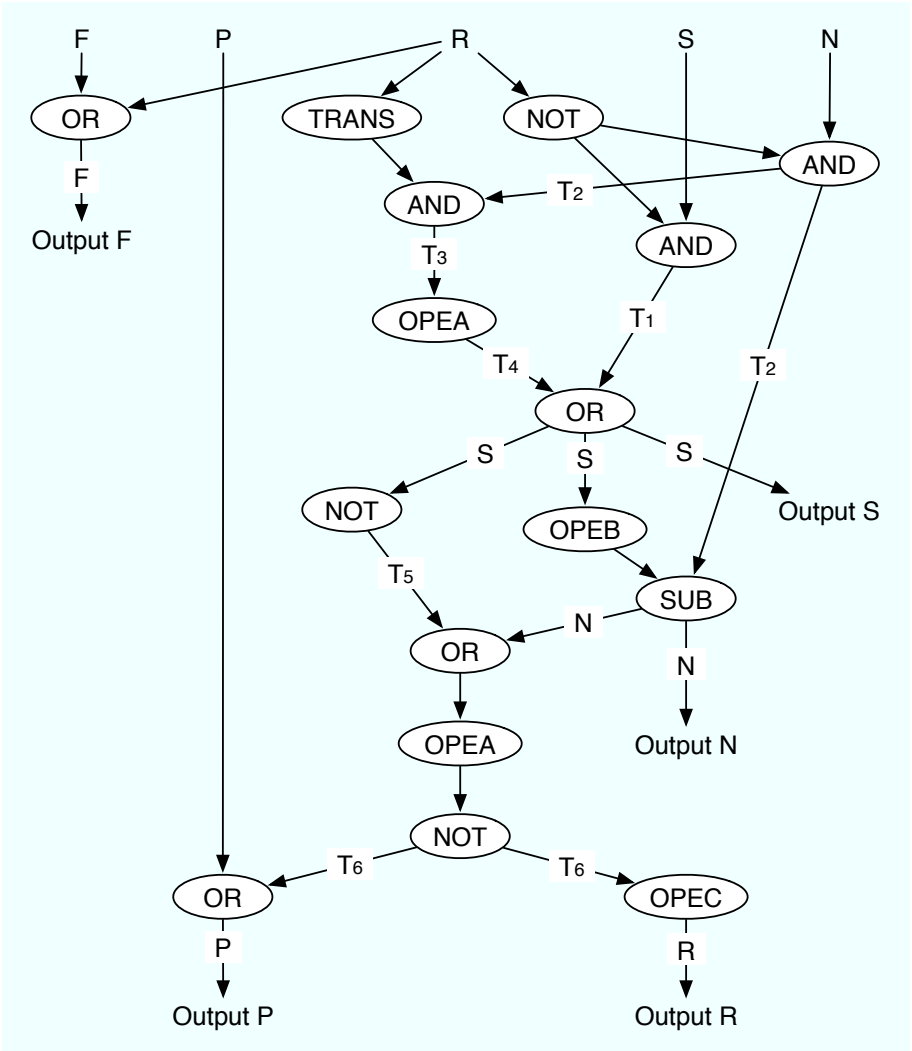


Figure 5.3: The data-flow of the matrix representation of MPSA

5.5 Evaluation

In this section, the execution time of MPSA on a dynamically reconfigurable processor is evaluated. It is assumed the maximum distance in the network is d_{max}, and the execution time of the sequence in Fig 5.3 is t_{seq}. The sequence shown in Fig.5.3 repeats until all shortest paths from node s is determined, in other words, all element in F become MAX_VALUE. Therefore the sequence must be repeated d_{max} + 1 times to obtain

all shortest paths. The execution time T_{exe} is expressed as follows.

$$T_{exe} = (d_{max} + 1) \times t_{seq} \tag{5.2}$$

The above result indicates the execution time is $O(d_{max})$. In square-mesh networks,

$$d_{max} = 2(\sqrt{n} - 1) \times c_{ave} \tag{5.3}$$

where n is the number of nodes, and c_{ave} is the average cost for all edges $(i, j) \in E$. From Equation (5.2) and (5.3), we obtain

$$T_{exe} = 2(\sqrt{n} - 1) \times c_{ave} \times t_{seq} + t_{seq}. \tag{5.4}$$

Equation (5.4) indicates the execution time is $O(\sqrt{n})$ in square-mesh networks. On other topologies, d_{max} is at most $O(n)$, so the computational complexity of MPSA is at most $O(n)$. This is lower than that of Dijkstra's algorithm.

To measure T_{seq}, I design the matrix operation units on DAPDNA-2. They corresponds to the operations used in the operation, respectively. In these design, an element in a matrix is 16-bit data, and the size of a matrix is 4×4. The latency for each matrix operation is shown in Table 5.1. The latencies are between 2 clocks and 4 clocks. We obtain $t_{seq} = 27$(clocks) from the result of the latencies and Figure 5.3.

I compare the execution time of Dijkstra's Algorithm and MPSA. Figure 5.4 shows the execution time versus the number of nodes where the network topology is square-mesh. It is assumed the average cost c_{ave} is 3, and the reconfigurable processor has enough PEs and memories. Dijkstra's algorithm is run on the PC whose processor is a Intel Pentium 4 3.0GHz, and which has 1024MB RAM. The plots of Dijkstra's algorithm in Fig. 5.4 is the actual measurement. On the other hand, that of MPSA is calculated theoretically from Equation 5.4 and the assumption that DAPDNA-2 runs at 166 MHz. In the proposed algorithm, the execution time increases slowly as n increases because MPSA on a reconfigurable processor runs at $O(\sqrt{n})$. When $n = 169$, 96.7% fewer time are used than with Dijkstra's algorithm.

Table 5.1: The latency of each matrix operation unit

Operation type	Time (clocks)
AND	2
OR	2
NOT	2
SUB	2
TRANS	3
OPEA	3
OPEB	3
OPEC	4

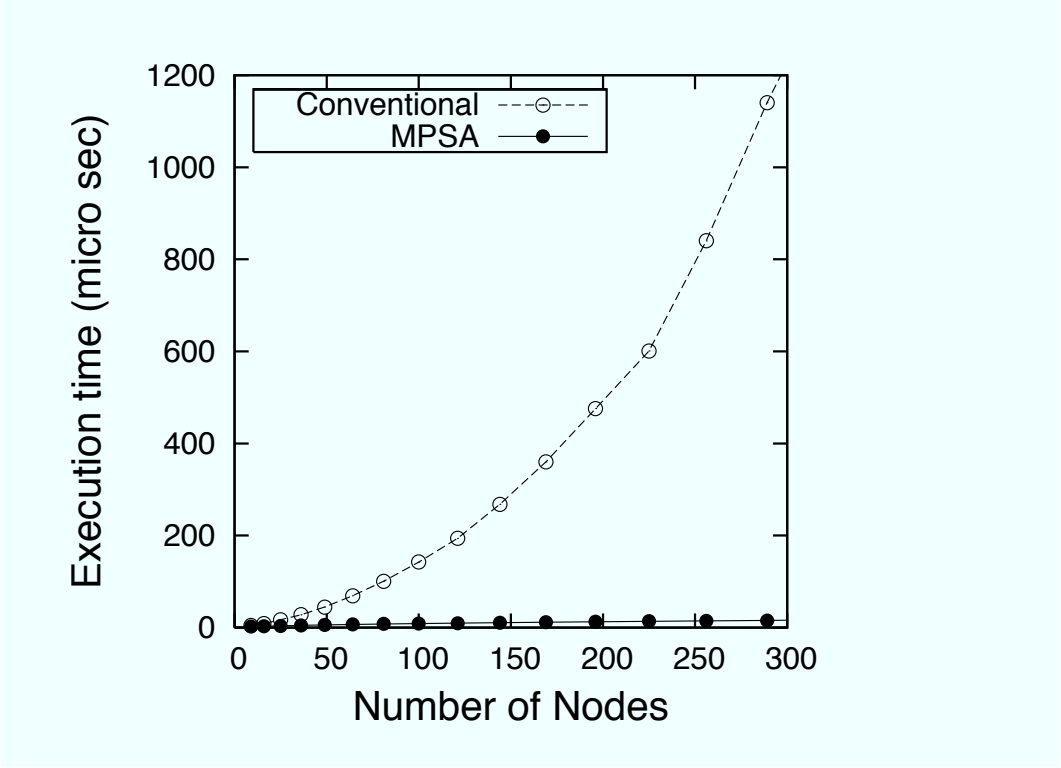


Figure 5.4: The execution time versus the number of nodes in Dijsktra's algorithm and MPSA

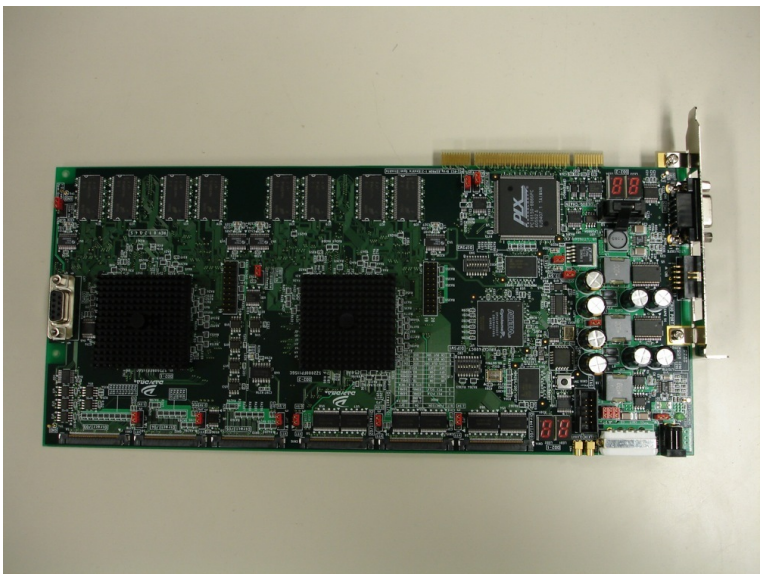


Figure 5.5: The Image of DAPDNA-EB4

5.6 Hardware off-loading engine prototype

Hard-wired logic is a good way to speed up an algorithm, but it has less flexibility than software-based approach. In this chapter, I employ an off-loading technique which is a hardware and software mixed approach. In this approach, only a heavy task is executed on the hardware specialized to a certain algorithm. It leads to reduction of the total execution time. On the other hand, a light task is executed by software. It leads to having flexibility of software. To speeding up the shortest path calculation of an actual router, I make a prototype of a hardware off-loading engine working together with GNU Zebra software router. The off-loading engine is made on IPFlex DAPDNA-2 using the matrix-based operations shown in Section 5.4.2. Our implementation is done on the evaluation board, DAPDNA-EB4 shown in Fig 5.5.

When the number of node is large, there are cases that not all calculations of elements of a matrix are parallelized because of insufficient number of PEs on an actual DRP. In the case, of course, the performance is degraded from the ideal performance shown in Fig.

5.4. There is another trade-off when hardware off-loading technique is used. The overhead of communicating between the host processor and off-loading engine is apparent when the size of the problem is small. In addition, off-loading technique introduces the additional cost because an off-loading engine is added. These are trade-off to introduce off-loading technique.

5.6.1 MPSA implementation on DAPDNA-2

In the implementation, I decide the size of an element in a matrix is 16 bits since the bit-length of metric in a header of OSPF is 16 bits [3]. One word of DAPDNA-2 is 32-bit length, so two element can be packed in a word. And I determine the size of the matrix is 32×32. It is lead to be able to calculate 32-node network at a time. The data size of the matrix is 2KB ($32 \times 32 \times 2$).

The data of the matrices is loaded from external memory, and input into the shortest path calculation unit. When the data pass through the unit, the operations shown in Fig. 5.3 are executed. Therefore, the unit runs multiple times until all the shortest paths from the source node are determined. After the output data is stored into external memory. In the implementation, the data of each matrix is input/output in serial. Serial I/O makes required memory bandwidth low, and reduces PE consumption compared to parallel I/O. Especially, reduction of PE consumption is a merit because of easy fitting.

Figure 5.6 shows the high level function constituting MPSA. The implementation of MPSA consists of three parts: pre-processing, main processing, and post-processing stage. The operation in Fig. 5.6 corresponds to several matrix operations.

Four operations exist in pre-processing stage as explained below.

- Update fixed nodes

  Add newly reached nodes which is determined in the previous processing to the set of fixed nodes.

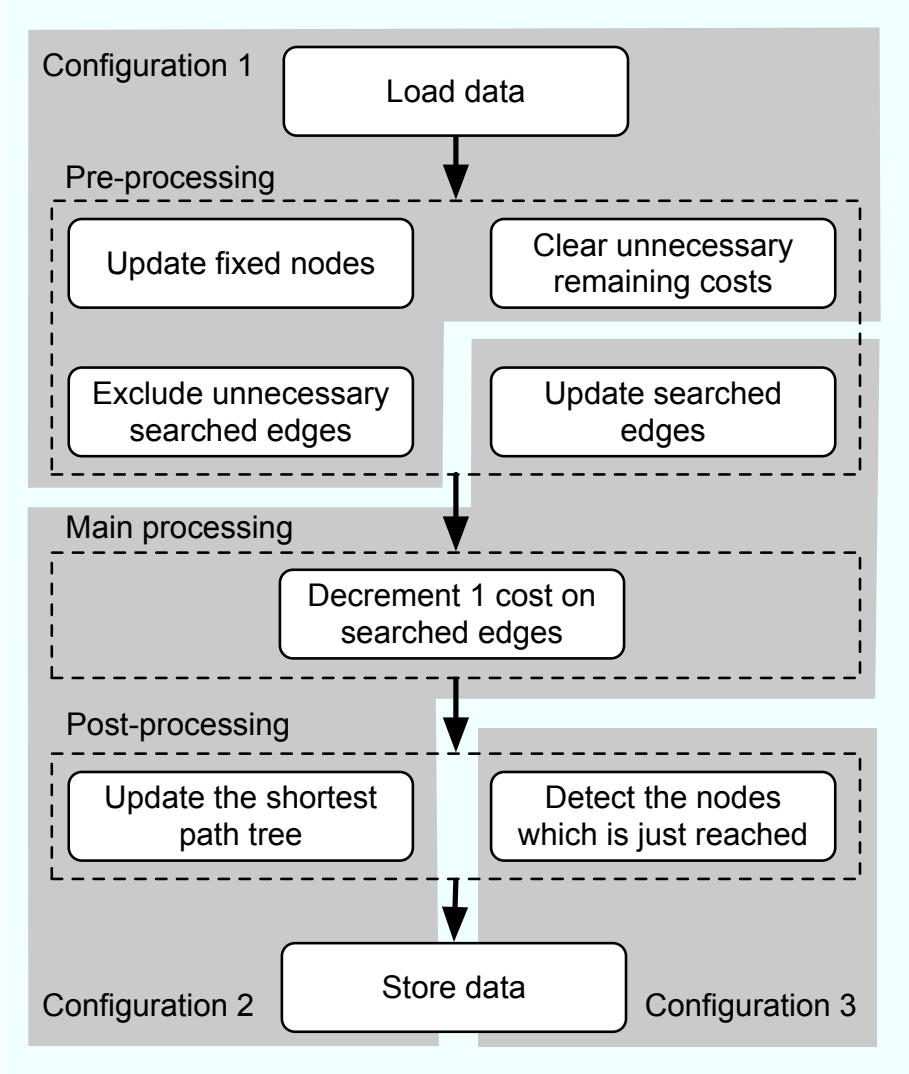


Figure 5.6: High level design of our implementation and splitting into three configurations

- Clear unnecessary remaining costs

  In order not to add the links towards the reached nodes to the searched nodes, we clear the corresponding values in the network matrix.

- Exclude unnecessary searched links

  We exclude the links towards the reached nodes from the set of searched links.

- Update searched links

  Add the links from newly reached nodes to the set of searched links.

In main processing stage, we executes the operations shown in Fig. 5.3. Finally, post-processing stage has two operations as explained below.

- Update the shortest path tree

  The links on the shortest path tree are determined in main processing stage. Add the links to the shortest path tree which is expressed as path matrix.

- Detect the nodes which is just reached

  The newly reached nodes in main processing stage is detected in this operation. The reached nodes are used in next pre-processing stage.

The implementation employes reconfiguration to reduce usage of PEs. As shown in Figure 5.6, all the functions constituting MPSA is splitting three configurations. Configuration 1 includes loading data and pre-processing stage except updating searched edges. Configuration 2 includes updating searched edges, main processing stage, updating the shortest path tree in post-processing stage, and storing the data. It is main configuration in our implementation. Configuration 3 includes detecting the nodes which is just reached and storing the part of the data. We split into three configurations at the point before the operation TRANS and OPEC. This is because there are dependencies in TRANS and OPEC.

Figure 5.7 shows the flowchart of reconfigurations. Configuration 1 to 3 run until all the shortest paths are determined, so they run d_{max} + 1 times.

5.6.2 Integration with GNU Zebra

I integrate our hardware off-loading engine explained in previous subsection with GNU Zebra. GNU Zebra is a famous software-based router which runs on UNIX-like operating system, for example Linux. Zebra works as a daemon, and it processes many routing protocol such as RIP, and OSPF, etc. To be able to off-load the calculation, I modify

Zebra version 0.94's source code. I add the off-loading architecture in ospfd which is a daemon processing OSPF included in Zebra.

Fig 5.8 shows the architecture of our DAPDNA-2 based hardware off-loading engine integrated with Zebra. As operating system, I employ RedHat Enterprise Linux 4 which is running on a commodity PC. The PC has Intel Pentium 4 3.0GHz and 1024MB RAM. MPSA is implemented on DAPDNA-2 and evaluation board DAPDNA-EB4 is used. DAPDNA-EB4 is Full-size PCI board and plugged into a PCI slot of the PC as shown in 5.9. The device driver makes DAPDNA-EB4 work on RedHat Enterprise Linux. The modified ospfd usually collects link-state information, and it triggers the shortest path calculation on DAPDNA-2 when re-calculation is required. Before executing MPSA, link-state informations are transformed to the matrix-based data format as show used in MPSA. Network matrix, Reach matrix, Fix matrix, Search matrix, and Path matrix are initialized and set to RAM on DAPDNA-EB4. Then, ospfd triggers executing MPSA on DAPDNA-2. After executing MPSA, ospfd gets the result of the shortest paths from RAM on DAPDNA-EB4. The result is represented as matrix-based format, so ospfd transforms the result to Zebra's internal data structure and makes a routing table.

5.7 Conclusion

In this chapter, to solve the high computational complexity of path calculation in traffic engineering, a new parallel shortest path algorithm called Multi-route Parallel Search Algorithm is proposed. The proposal takes advantage of parallelism of DRP and it is suitable for DRP. In next generation backbone network, traffic engineering with many functionalities is required to utilize network resources efficiently and support QoS. Under this situation, the shortest path calculation is frequently occurred and become a heavy task. The proposed algorithm consists of simple processing, in which multiple paths are simultaneously searched by multiple Processor Elements (PEs) of DAPDNA-2. There-

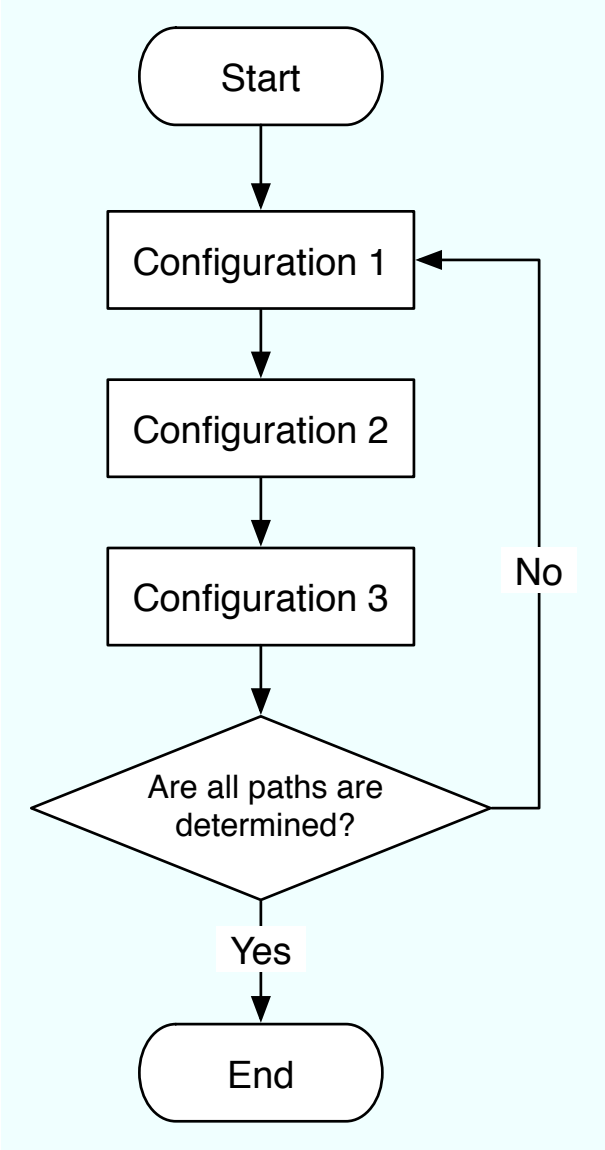


Figure 5.7: The flowchart of reconfigurations

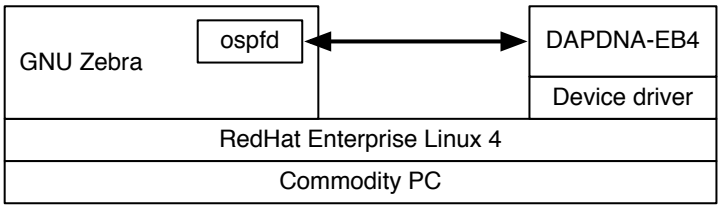


Figure 5.8: The architecture of our prototype

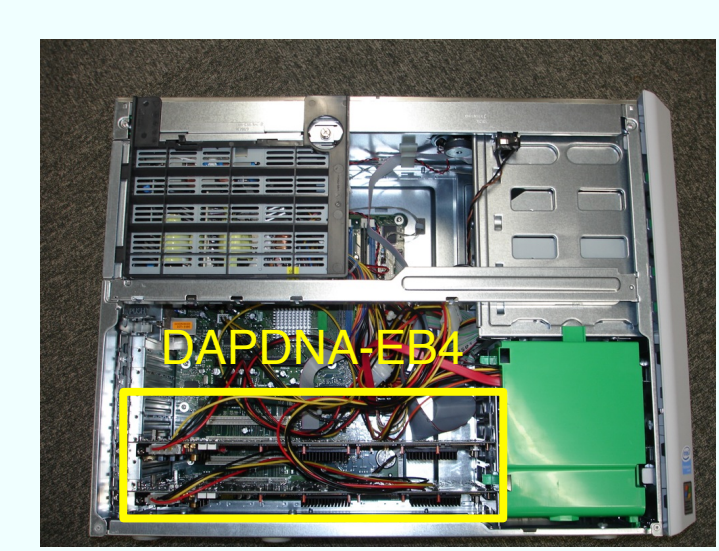


Figure 5.9: DAPDNA-EB4 is plugging into a PCI slot

fore, it reduces the execution time of shortest path calculation to 2.8 percent compared with the popular shortest path algorithm, Dijkstra's algorithm. The proposed architecture and prototype system can be applied to future network sophisticated traffic engineering.

References

[1] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," *Request for Comments (RFC)*, no. 3945, Oct. 2004.

[2] K. Kompella and Y. Rekhter, "OSPF extensions in support of generalized multi-protocol label switching (GMPLS)," *Request For Comments (RFC)*, no. 4203, Oct. 2005.

[3] J. Moy, "OSPF version 2," *Request For Comments (RFC)*, no. 2328, Apr. 1998.

[4] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, Oct. 1959.

[5] H. Amano, "A survey on dynamically reconfigurable processors," *IEICE Transactions on Coummnications*, vol. E89-B, no. 12, pp. 3179–3187, Dec. 2006.

[6] T. Sugawara, K. Ide, and T. Sato, "Dynamically reconfigurable processor implemented with IPFlex's DAPDNA technology," *IEICE Transactions on Information and Systems*, vol. E87-D, no. 8, pp. 1997–2003, Aug. 2004.

[7] A. Crauser, K. Mehlhorn, U. Meyer, and P. Sanders, "A parallelization of dijkstra's shortest path algorithm," in *3rd International Symposium on Mathematical Foundation of Computer Science (MFCS'98)*, Czech Republic, Aug. 1998, pp. 722–731.

[8] "GNU zebra – routing software," http://www.zebra.org/.

[9] "Welcome to xorp," http://www.xorp.org.

Chapter 6

Optimal application framework for distributing large volume data

6.1 Abstract

An application framework for distributing large volume data is one of requirements in next generation backbone network. Content delivery network (CDN) is effective framework for large data distribution. It is an important issue in CDN to obtain the optimal replica placement patter in practical time. This chapter proposes a novel approach that takes advantage of the parallelism of dynamically reconfigurable processors (DRPs) to solve the resource minimization problem, which is NP-hard. The proposal obtains the optimal solution by running an exhaustive search algorithm suitable for DRP. Greedy algorithms, which have been widely studied for tackling the replica placement problem, cannot always obtain the optimal solution. The proposed method is implemented on an actual DRP and in experiments reduces the execution time by a factor of 40 compared to the conventional exhaustive search algorithm on a Pentium 4 (2.8 GHz).

6.2 Introduction

Demand continues to grow for downloading rich contents, for example DVD-quality or high definition videos, through the Internet. Two factors are the keys to meeting this demand: local content sources and adequate transfer capacity. Optical networks can provide

the high-speed and high-capacity pipes needed; they are now commonly used in backbone networks and can handle bandwidth-consuming applications if the transfer distances are reasonable. this chapter focuses on the other factor, an shows how to determine where to site content sources.

Identifying the optimum number and location of content sources (servers) involves an understanding the trade-offs between performance and cost. Using just a few servers is very effective in reducing initial investment costs but the servers will experience extremely high loads since they must deal with simultaneous download requests from many clients. Moreover, the average transfer distance is high which degrades the QoS and indeed overall network performance.

The content delivery network (CDN) was proposed to improve network resource utilization efficiency for large contents distribution [1, 2]. CDN consists of two types of servers: origin server and replica server. The number of origin servers is usually one (for each contents provider), and the many replica servers are spread throughout the service area. Origin server holds the original contents and delivers them to the replica servers as needed to ensure user requests can be satisfied. The contents stored in a replica server are called replicas. CDN promises high-speed downloads since the client downloads the data from the server nearest to the client in terms of network connectivity.

In CDN, replica placement impacts the performance which includes the load on the origin server and the network since data placement decisions must be made on a per content basis and be made dynamically in response to user requests. Minimizing the number of mirroring resources (servers) under a Quality of Service (QoS) constraint is a key issue in CDN, so research in this area has been quite active. It is a tough problem to select which nodes should host which replicas.

The distance between two nodes is used as a metric for QoS in CDN. A request must be resolved by a server within the distance specified by the request because all clients

want to download contents within the allotted time period. Every node knows the nearest replica server that holds the requested data and the request is sent to the replica server that is closest to the client. The goal is to find a replica placement that satisfies all requests without violating any range constraint, and that minimizes the update and storage cost at the same time. This chapter emphasizes the optimization of the number of replicas under the delay constraint.

Replica placement problem is derived from the set cover problem which is known to be NP-hard [3]. Therefore, calculation time increases rapidly with network scale. Greedy algorithms have been widely studied since they yield sub-optimal solutions reasonably quickly [4–11]. However, it has been proven mathematically that no greedy algorithm always can attain the optimal solution [3]. Sub-optimum solutions have higher replicating cost, i.e. the number of replicas, than the optimal solution. The goal then is to secure the optimal replica placement within some practical time.

Our solution to obtaining the optimal solution to the replica placement problem is based on combining advanced processors with suitable algorithms. It is not realistic to obtain the optimal solution with a Neumann-type processor given the number of all solution candidates. To drastically reduce the calculation time, a novel approach that uses an exhaustive search algorithm that suits the parallelism offered by a dynamically reconfigurable parallel processor (DRP) is proposed. The proposal is the marriage of advances in software and hardware.

Our proposal is implemented on a commercially available DRP, DAPDNA-2 of IPFlex Inc [12]. DAPDNA-2 consists of a Digital Application Processor (DAP), a high-performance RISC core, and Digital Network Architecture (DNA), a dynamically reconfigurable two-dimensional matrix. DNA is embedded in an array of 376 Processing Elements (PEs), which are comprised of computation units, memory, synchronizers, and counters. The PE Matrix circuitry can be reconfigured freely into the structure that best suits the current

application.

It is not feasible to solve the large-scale replica placement problem on a program counter-based processor. Our proposed algorithm divides the problem in an optimal manner and subjects the pieces to pipeline operation. Whereas the time complexity of conventional exhaustive search using Beeler's algorithm [13] is $O({}_nC_k)$, the time complexity of the proposed algorithm is $O(\sqrt{{}_nC_k})$. Experiments show that the proposed method reduces the execution time by a factor of 40 times compared to conventional exhaustive search using Beeler's algorithm on a Intel Pentium 4 (2.8GHz).

The rest of this chapter is organized as follows. Section 6.3 details the related work on the replica placement problem and combination algorithm. In Section 6.4, a fast solution to the replica placement problem is proposed; it divides the candidates and pipelines them on a DAPDNA-2. Section 6.5 evaluates the performance of our implementation. Finally, the conclusion of this chapter is denoted in Section 6.6.

6.3 Replica placement problem

The network is represented by an undirected graph $G = (V, E)$, where V is the set of servers, and $E \subseteq V \times V$ denotes the set of network links among the servers. Each link $(u, v) \in E$ is associated with a cost $d(u, v)$ that denotes the communication cost of the link between servers u, v. It is assumed that the graph is connected, so that one server can connect to any other server via a path. I define the communication cost of a path as the sum of the communication cost of the links along the path. Because it is assumed that each server knows the nearest replica, I define $d(u, v)$ between two servers u, v to be the communication cost of the shortest path between them. Every server u has a storage cost $s(u)$, which denotes the cost to put a replica on server u. Different servers usually have different storage costs.

Fig. 6.1 illustrates replica placement. The numbers in the circles are server indices

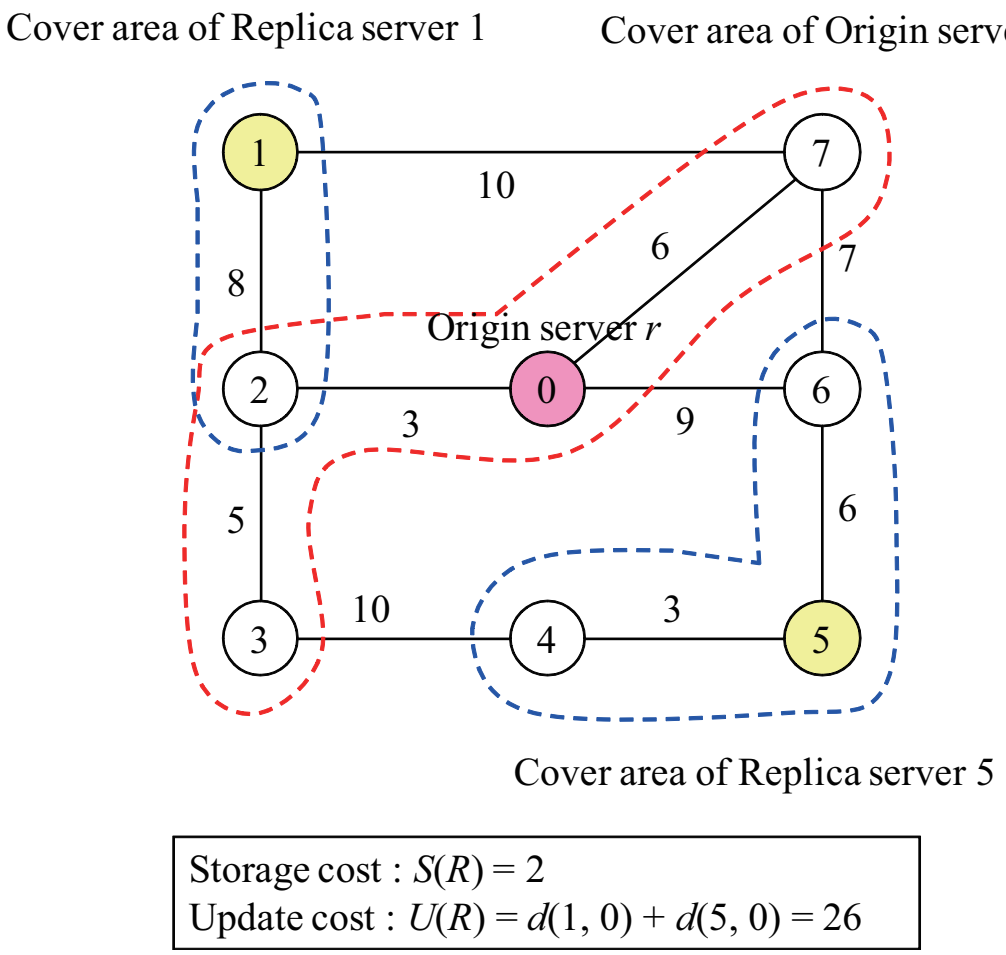


Figure 6.1: Origin server and replica servers {1, 5} can cover all nodes when the quality requirement is 8.

between 0 and $n - 1$, where n is the total number of servers. The number on a link is the communication cost of the link.

Each server in the network serves multiple clients, although the clients is not illustrated in Figure 6.1. A client sends its request to its associated server, which then processes the request. If the local server has the requested data, the request is processed locally. Otherwise, the request is directed to the nearest server that has the replica. In addition, the communication cost from clients to servers is ignored because it doesn't impact the replication decision.

Without loss of generality, it is assumed that server 0 is origin server r. Initially, only the origin server has the contents. A replica server is a server that has replicated contents. A replication strategy, $R \subseteq V - \{r\}$, is a set of replica servers.

The replication cost is used to evaluate replication strategies. The replication cost $T(R)$ of replication strategy R is defined as the sum of storage cost $S(R)$ and update cost $U(R)$.

$$T(R) = S(R) + U(R) \tag{6.1}$$

Storage cost: The storage cost of replication strategy R is the sum of all storage costs of the replica servers.

$$S(R) = \sum_{v \in R} s(v) \tag{6.2}$$

Update cost: In order to maintain data consistency, origin server r issues update requests to every replica server. It is assumed that there is an update distribution tree T, which connects all servers in the network. For example, a shortest path tree rooted at the origin server is used as the update distribution tree. Origin server r multicasts update requests through links on this tree until all replica servers in R receive the update request. Every node receives the update request from its parent and relays these requests to its children according to the update distribution tree.

Let $p(v)$ be the parent of node v in the update distribution tree, and T_v be the subtree rooted at node v. If $T_v \cap R \neq \phi$, link $(v, p(v))$ participates in the update multicast. As a result, the update cost is the sum of the communication costs from these links $(v, p(v))$. For example, if the replication strategy R is $\{1, 5\}$ in 6.1, then the update cost is $11 + 15 = 26$.

$$U(R) = \sum_{v \neq r, T_v \cap R \neq \phi} d(v, p(v)) \tag{6.3}$$

Every server u has a service quality requirement $q(u)$. The requirement mandates that all requests generated by u will be served by a server at less than $q(u)$ communication cost. It is assumed that every server in the network knows the replica server nearest to itself. If a request is served by the nearest replica server within $q(u)$, the request is satisfied,

otherwise, the request is violated. If all requests in the system are satisfied, the replication strategy is called feasible.

$$\min_{w \in R \cup r} d(v, w) \leq q(v) \tag{6.4}$$

The replica placement problem is to find the feasible replication strategy that minimizes the replication cost in Equation (6.1) [10]. As an example, it is assumed that the quality requirement is 8 for all servers and the replication strategy is {1, 5} in Figure 6.1. We can verify that the replication strategy together with the origin server can satisfy all requests within the network. The replication strategy {1, 5} covers all nodes in Figure 6.1. The replica placement problem is derived from the set cover problem which is known to be NP-hard [3]. The definition of the set cover problem is as follows.

Minimum Weight Set Cover Problem: Let U be the universal set and S be the family of U. The solution is sub-family S such that the weight is minimized and $\bigcup_{S \in S} S = U$ is satisfied.

The replica placement problem is NP-hard because the minimum weight set cover problem is known to be NP-hard. Several greedy algorithms have been proposed to decrease the calculation time [4–11]. Johnson proposed a greedy algorithm against the minimum weight set cover problem [4]. This algorithm is a straightforward heuristic. The time complexity is proportional to n. In [5,8], fan-out based replica placement algorithms were proposed. They put replicas on servers in descending order of server degree. Kangasharju et al. proved that their target replica placement optimization problem is NP-complete, and proposed some heuristic algorithms [9]. Tang et al. investigated QoS-aware replica placement problems to elucidate QoS requirements, and proposed the l-Greedy-Insert and l-Greedy-Delete algorithm [10]. They showed that the QoS-aware placement problem for replica-aware services was NP-complete. Wang et al. proposed a heuristic algorithm called Greedy-Cover [11]. Experiments indicated that the proposed algorithm found near-optimal solutions effectively and efficiently. Karlsson et al. provided a framework for

evaluating replica placement algorithms [7], and compared several replica placement algorithms [6]. [6] also provides a comprehensive survey of replica placement algorithms. However, note that it has been proven mathematically that no greedy algorithm can obtain the optimal solution [3]. Therefore, to get the optimal solution, a fast exhaustive search algorithm is required.

Exhaustive search algorithms generally consist of the following three procedures.

1. Generate all solution candidates, in other words all replication strategies
2. Check each solution candidate as to whether all nodes are covered
3. Calculate the replicating cost of each solution candidate

The above procedures are executed over all of replication strategies, and the optimal solution is selected. The parallelization of procedures 2 and 3 is easily achieved because the replication strategies are completely independent in these procedures. However, the parallelization of procedure 1 is not easy, so procedure 1 is likely to become a bottleneck. Therefore, I focus on a solution candidate generation scheme to speed up the exhaustive search algorithm in this chapter.

Exhaustive search algorithms to solve the Boolean Satisfiability Problem (SAT), which is an NP-hard problem as well as the set cover problem, have been implemented on FPGAs [14–17]. Instance-specific hardware is employed to reduce the execution time in these implementations. Thus, we have to re-generate instance-specific hardware for each problem instance, i.e. the hardware compilation, which is a significant overhead, is required. In addition, the problem instance is limited in the implementations of [15, 17] since it was assumed that the forms of the boolean expressions they contained were limited.

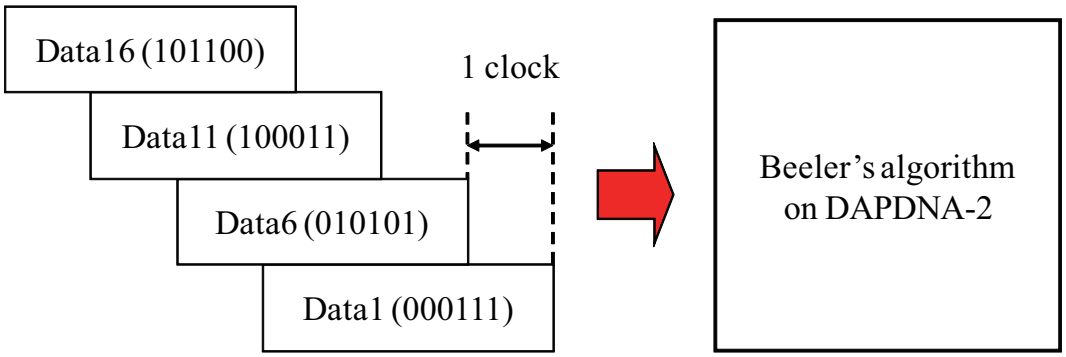


Figure 6.2: First data of each group are entered per clock cycle by pipeline operation. DNA matrix outputs Data2, Data7, Data12 and Data17, which are the next input data.

6.4 Proposed method

Combinatorial algorithms can be applied to problems derived from the set cover problem, such as the replica placement problem. The calculation time of the replica placement problem increases rapidly with network scale. I propose a new method that generates and tests all combinations rapidly to obtain the optimal solution in a feasible time. Our proposed method divides combinations into different groups which are executed in parallel. The first data of each group are entered per clock cycle following pipeline operation. I implemented Beeler's algorithm [13], which can generate all combinations in ascending order, on DAPDNA-2.

Figure 6.2 shows the pipeline operation when ${}_6C_3$ is divided into 4 groups. 1st, 6th, 11th and 16th data are input data because 20 combinations are divided into 4 groups. DNA matrix outputs Data2, Data7, Data12 and Data17, which are the next input data in Figure 6.2. The result of the last group is delayed by 3 clocks compared to that of the first group. The overall execution time is about 75 percent shorter than the original execution time.

There are two problems that need to be solved. First, how can we calculate the first data of each group when the combinations are divided into different groups? Beeler's algo-

rithm can generate all combinations in ascending order but there are data dependencies. It's difficult to calculate any order pattern because the difference between neighboring patterns is not constant. I solve this problem by proposing an algorithm that can generate patterns in any order.

Second, what is the optimal number of divisions in terms of minimizing the overall number of calculation clocks needed? Increasing the number of divisions decreases the overall calculation clock number but there is a lower limit beyond which the overall clock number starts to increase. The optimal number of divisions depends on the number of combinations and the calculation clocks of Beeler's algorithm. In order to solve this problem, I tackled the theory behind the optimal number of divisions.

6.4.1 Beeler's algorithm and any-order pattern algorithm

M. Beeler et al. proposed an algorithm that generates all combinations and picks k outcomes from n possibilities [13]. These combinations can be expressed in n-digit binary form. For example, 010110 represents $(2, 3, 5)$ when $n = 6$. Combinations can be ordered as follows; $(2, 3, 5) < (2, 4, 5)$ because $010110 < 011010$. Beeler's algorithm can generate all combinations from 000111 to 111000 in order. The details of the algorithm are as follows.

1. Let S_1 be the pattern in which all bits are unset except for the least significant bit of combination X.
2. $R = X + S_1$
3. Let S_2 be the pattern in which all bits are unset except for the least significant bit of combination R.
4. $S_3 = (S_2/S_1) >> 1 - 1$

5. $Y = R|S_3$ is next to X.

When $n = 6, k = 3, X = 001110$, for example, Y is calculated as follows.

1. $S_1 = 000010$
2. $R = X + S_1 = 010000$
3. $S_2 = 010000$
4. $S_3 = (S_2/S_1) >> 1 - 1 = 001000 >> 1 - 1 = 000011$
5. $Y = R|S_3 = 010011$

I propose a new algorithm that generates any order pattern in combinations sorted in ascending order. The following equation is generally true.

$$ {}_nC_k = \sum_{i=k-1}^{n-1} {}_iC_{k-1} \tag{6.5} $$

If you want to get m-th pattern, find x_1, which is the smallest value among the values of x satisfying Equation (6.6). ${}_iC_{k-1}$ corresponds to the number of the patterns whose i-th bit is the most significant bit, and where the number of 1's between 0 and the $(i-1)$-th bit is $k-1$. Therefore, x_1 means the patterns that include the m-th pattern, and the highest bit to be set at 1 of the patterns is the x_1-th bit.

$$ \sum_{i=k-1}^{x} {}_iC_{k-1} \geq m \quad (k-1 \leq x_1 \leq n-1) \tag{6.6} $$

${}_{x_1}C_{k-1}$ means the number of the patterns whose x_1-th bit is the most significant and the number of 1's between 0 and $(x_1 - 1)$-th bit is $k-1$ because the number of 1's is k in total. Hence, the x_1-th bit of the m-th pattern is 1. The m-th pattern corresponds to $m - \sum_{i=k-1}^{x_1-1} {}_iC_{k-1}$-th in ${}_{x_1}C_{k-1}$. Replace m as follows.

$$ m \rightarrow m - \sum_{i=k-1}^{x_1-1} {}_iC_{k-1} \tag{6.7} $$

Next, find x_2, which is the smallest value among the value of x satisfying the following inequality.

$$\sum_{i=k-2}^{x} {}_iC_{k-2} \geq m \quad (x \leq x_1 - 1) \tag{6.8}$$

${}_{x_2}C_{k-2}$ represents the patterns whose highest bit to be set at 1 is the x_2-th bit and there are $k-2$ 1's between 0 and $x_2 - 1$. Hence, the x_2-th bit of the pattern is 1. $x_1, x_2, \cdots, x_k$ can be obtained by repeating k times in a similar way. Setting the corresponding bits to 1 yields get the m-th pattern.

For example, the 6th pattern ($m = 6$) in ${}_6C_3$ can be obtained as follows.

$${}_6C_3 = {}_2C_2 + {}_3C_2 + {}_4C_2 + {}_5C_2 = 1 + 3 + 6 + 10 \tag{6.9}$$

Apply the equation (6.5) to ${}_4C_2$ because ${}_4C_2$ includes the 6th pattern. Hence, $x_1 = 4, m \rightarrow 2$.

$${}_4C_2 = {}_1C_1 + {}_2C_1 + {}_3C_1 = 1 + 2 + 3 \tag{6.10}$$

Apply the equation (6.5) to ${}_2C_1$ because ${}_2C_1$ includes the 2nd pattern. Hence, $x_2 = 2, m \rightarrow 1$.

$${}_2C_1 = {}_0C_0 + {}_1C_0 = 1 + 1 \tag{6.11}$$

The 1st pattern corresponds ${}_0C_0$. Hence, $x_3 = 0$. Setting the corresponding bits to 1 yields the 6th pattern, 010101.

6.4.2 Optimal number of divisions

Let a be the number of clocks taken to calculate any order pattern and b be the number of clocks to execute Beeler's algorithm. $b\,({}_nC_k - 1)$ clocks are required to generate all combinations and pick k outcomes from n possibilities. ${}_iC_j$ is the number of j-selections from i elements, where i, j are nonnegative integers. When we divide the combinations into 2 groups, $a + \frac{b({}_nC_k - 1)}{2} + 1$ clocks are required. When we divide the combinations into

3 groups, $2a + \frac{b({}_nC_k-1)}{3} + 2$ clocks are required. When we divide the combinations into x groups in a similar way, y clocks are required as follows.

$$\begin{aligned} y &= (x-1)a + \frac{b({}_nC_k - 1)}{x} + x - 1 \\ &= \frac{b({}_nC_k - 1)}{x} + (a+1)x - a - 1 \end{aligned} \tag{6.12}$$

According to a relationship between arithmetic mean and geometric mean,

$$\begin{aligned} y &= \frac{b({}_nC_k - 1)}{x} + (a+1)x - a - 1 \\ &\geq 2\sqrt{\frac{b({}_nC_k - 1)}{x}(a+1)x} - a - 1 \\ &= 2\sqrt{b({}_nC_k - 1)(a+1)} - a - 1 \end{aligned} \tag{6.13}$$

The equality is satisfied if and only if $\frac{b({}_nC_k-1)}{x} = (a+1)x$. Hence

$$x = \sqrt{\frac{b({}_nC_k - 1)}{a+1}} \tag{6.14}$$

This is the optimal number of divisions.

6.4.3 Implementation on DAPDNA-2

Let n be the number of nodes except for the origin server and $k(\leq n)$ be the number of replicas. In our implementation, $n \leq 32$ because the word size of PE is 32-bits long. For example, we generate all combinations from 0000011 to 1100000 when $n = 7, k = 2$. Each node is represented as 32-bit data. Let the i-th bit be 1 if this node covers node i. In Equation (6.4), the v-th and w-th bits of node w are 1 because node w covers v. This information is called the cover data of node w. If OR between the cover data of all replica servers and that of the origin server yields 1111111, the replication strategy covers all nodes. For example, the replication strategy is node {1, 5} when the combination is 0010001. The following equations are true in Figure 6.1.

$$d(2,0) \leq q(2), \quad d(3,0) \leq q(3), \quad d(7,0) \leq q(7)$$

$$d(2,1) \leq q(2), \quad d(4,5) \leq q(4), \quad d(6,5) \leq q(6)$$

One strategy is node 0 (1000110), node 1 (0000011), and node 5 (0111000). This replication strategy covers all nodes because the result of the OR operation between the cover data of the these nodes equals 1111111. If several replication strategies cover all nodes, we choose the minimum-cost replication strategy.

After calculating the optimal number of divisions, our proposed algorithm consists of following 3 processes.

1. Calculate the first replication strategy of each group by using the algorithm described in Section 3.1.

2. Execute Beeler's algorithm.

3. Using the corresponding cover data, check that all nodes can be covered.

The result of process (1), which is executed by DAP, is stored in main memory. DNA reads this result from main memory and executes processes (2) and (3) in pipeline manner. The hardware compilation for each problem instance is not required since our implementation is not instance-specific, but application-specific. In addition, it can be generally applied to combinatorial optimization problems including the set cover problem.

To support network with more than 32 nodes, we have to make a small modification to the implementation; the algorithm remains basically the same. Several words are required to express a replication strategy and cover data. Therefore, several words are treated as one data unit in the implementation for over 32 nodes.

6.5 Performance evaluation

In this section, I compare the execution time of a DAPDNA-2 (166MHz) with that of a Pentium 4 (2.8GHz). Let k be the number of replicas and n be the number of nodes, except for the origin server, and d be the number of partitions.

Figure 6.3 shows the execution time to generate all combinations when $k = 8$, in other words, 25 percent of all nodes hold replicas. This percentage is derived from the result shown in [11]. This reference shows that the average number of replicas is 25 percent. Black plots represent the conventional exhaustive search using Beeler's algorithm on the Pentium 4, and white plots represent the proposed method on the DAPDNA-2. Circle plots represent the theoretical execution time, and square plots represent the experimental execution time. Figure 6.3 has some margin of error between theoretical and experimental times, but both demonstrate the same overall tendency. In the proposed method, the execution time increases slowly with n because DAPDNA-2 calculates in parallel using a pipeline operation. When $n = 30$, DAPDNA-2 reduces the execution time by a factor of 40 compared to Pentium 4. It is noted that the clock frequency of DAPDNA-2 is only 1/17th that of the Pentium 4. Such large performance gain cannot not be attributed to only the difference in processor architecture. The performance gain is the result of combining the parallel processing of DRP with the proposed algorithm.

Figure 6.4 shows the theoretical execution time when k is 25 percent of the number of nodes, n. Let a be the calculation clock of any-order algorithm and b be the calculation clock of Beeler's algorithm. The measured values are $a = 330$, and $b = 33$. Let t_c be the theoretical execution time of the Pentium 4 and t_p be the theoretical execution time of the DAPDNA-2. t_c, t_p are as follows.

$$t_c = \frac{b({}_nC_k - 1)}{2.8 \times 10^9}(sec) \tag{6.15}$$

$$t_p = \frac{2\sqrt{b({}_nC_k - 1)(a+1)} - a - 1}{166 \times 10^6}(sec) \tag{6.16}$$

While the Pentium 4 requires about 7 days to generate all combinations when the number of nodes, n equals 60, the execution time of the proposed method is about 9 seconds. This is because the time complexity of proposed algorithm is $O(\sqrt{{}_nC_k})$. As a result, the proposed algorithm is scalable against the number of nodes, n. Figure 6.5, 6.6, 6.7 show the theoretical execution time when k is 12.5 percent, 50 percent, and 75 percent

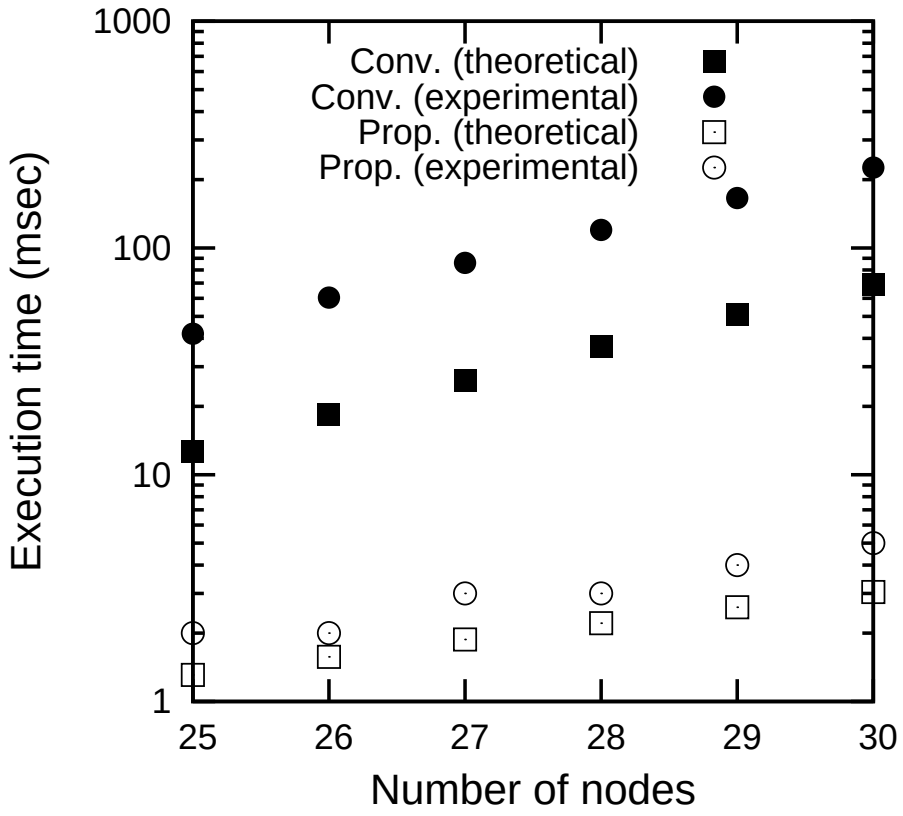


Figure 6.3: DAPDNA-2 can reduce the execution time by 40 times compared to Pentium 4 when the number of nodes is 30.

of the number of nodes, n, respectively. The dashed lines in the figures correspond to the value of 1 day. When k is equal to 12.5 percent of the number of nodes and $n = 88$, the conventional method requires over 4 days to generate all combinations. On the other hand, the execution time of proposed method is about 7 seconds. If k is 50 percent of the number of nodes and $n = 48$, the conventional method takes about 4 days, while our proposal takes only 7 seconds. The result when k is 75 percent of the number of nodes equals that when n is 25 percent of the number of nodes. This is because the execution time is a function of ${}_nC_k$ and the equation ${}_nC_k = {}_nC_{n-k}$ is always true. Until k reaches 50 percent of the number of nodes, the execution time increases. Therefore, when the value of k is small, we can extract the optimal replica placement for a large network within a

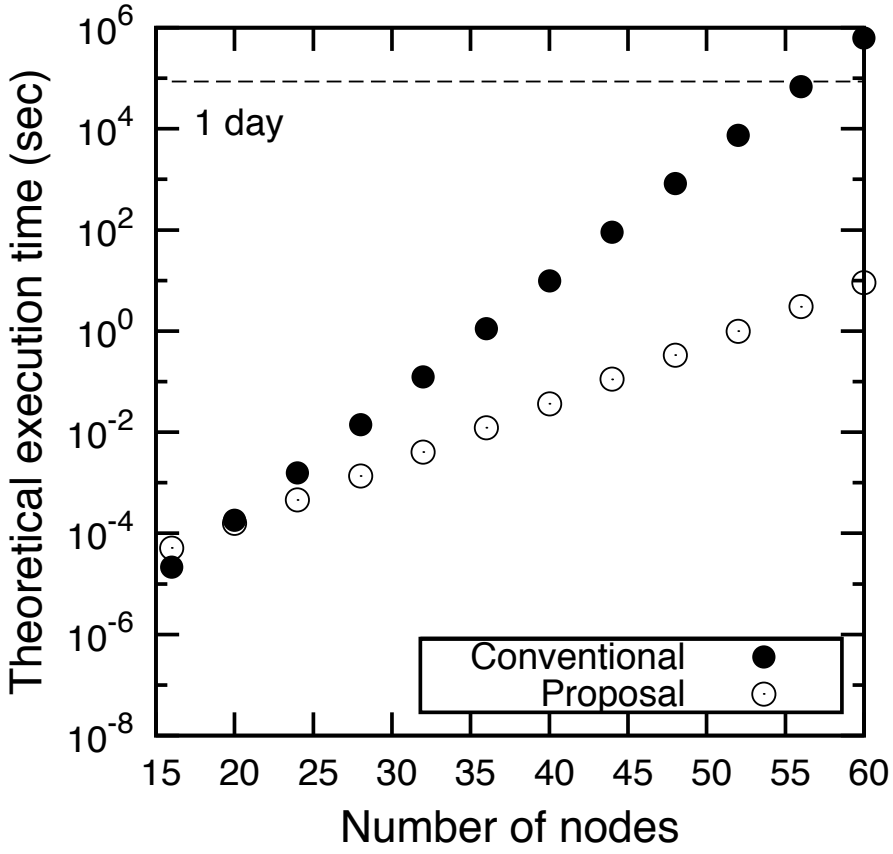


Figure 6.4: Theoretical execution time versus the number of nodes when the number of replicas k is 25 percent of the number of nodes n

certain value of the execution time. There are some cases that the execution time of the proposal is larger than that of the conventional approach in Fig. 6.4 to 6.7. This is because the overhead of calculating the seed patters directly is apparent when the number of nodes is small in the proposal.

Figure 6.8 shows the execution time of the DAPDNA-2 versus d when $k = 8$. Cross plots represent 25 nodes and triangular plots represent 27 nodes. Optimal number of divisions is calculated by Equation (6.14). $d = 328$ when $n = 25$ and $d = 472$ when $n = 27$. The execution time increases if d exceeds the optimal value.

Next, I compare the optimality of the replica placements yielded by a greedy algorithm

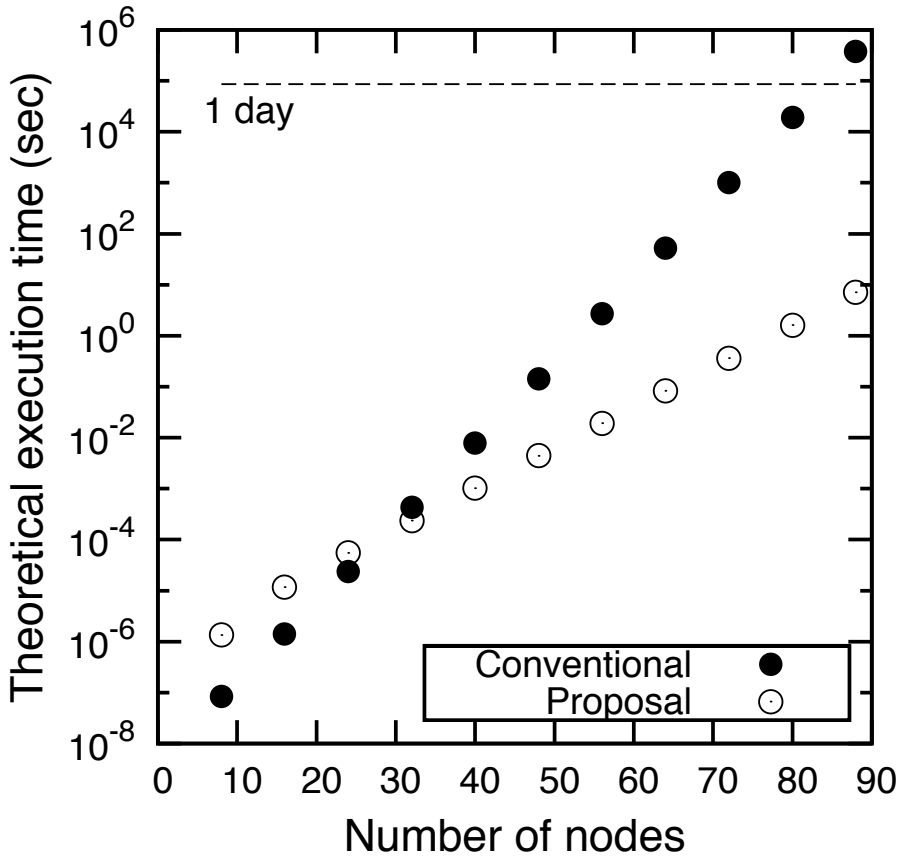


Figure 6.5: Theoretical execution time versus the number of nodes when the number of replicas k is 12.5 percent of the number of nodes n

and the proposed algorithm. The greedy algorithm employed in the evaluation is the algorithm proposed in [11], Greedy-Cover algorithm. I conducted simulations on 10000 different topologies. The topologies were generated by NetworkX library [18], and a random graph model (gnm_random_graph in NetworkX) is used. It is assumed average degrees of a node is 4, and service quality requirement $q(u) = 16, 20, 24$. The cost of a link is uniformly distributed between 1 and 15. Figure 6.9 compares the average optimality of Greedy-cover to that of the proposed algorithm. In the evaluation, optimality is defined as follows.

$$\text{optimality} = \frac{s}{o} \tag{6.17}$$

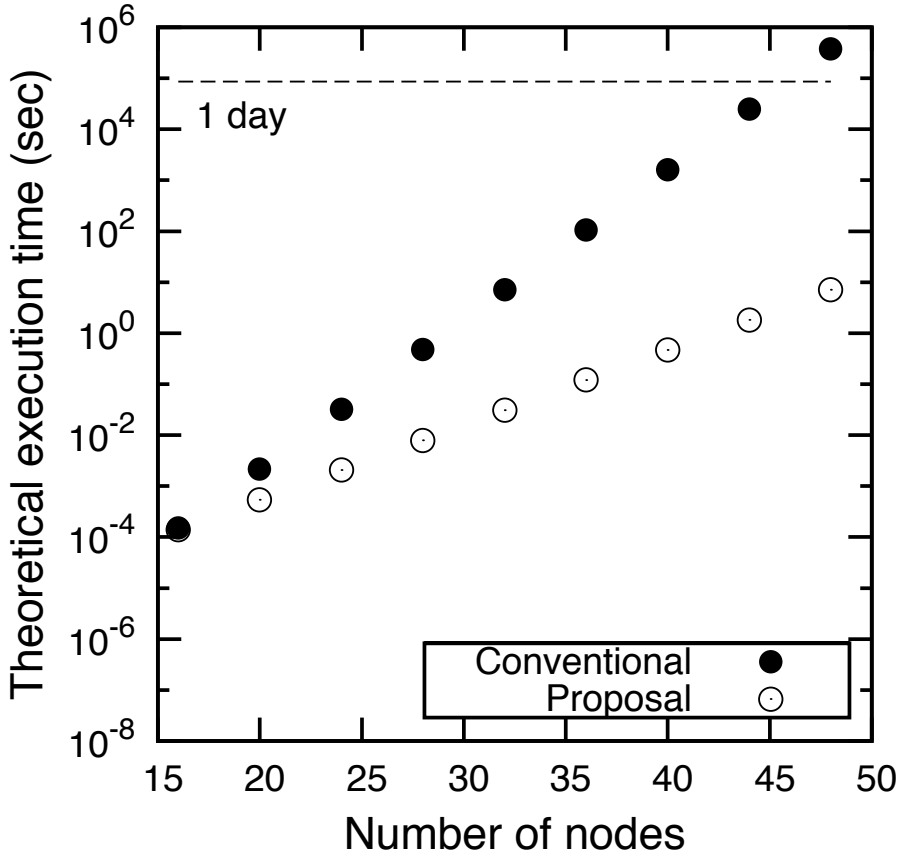


Figure 6.6: Theoretical execution time versus the number of nodes when the number of replicas k is 50 percent of the number of nodes n

where s is the number of replicas obtained by the replica placement algorithm, and o is the optimal number of replicas. From Fig. 6.9, the optimality of the Greedy-cover algorithm tends to increase with the number of nodes. On the other hand, the optimality of the proposed algorithm is always 1 since our proposal is based on exhaustive search, and can always obtain the optimal solution. When $q(u)$ is large, the optimality of Greedy-cover algorithm is getting near that of the proposal. It is because in case that the cover area is large the total number of replicas is decreasing in both of Greedy-cover and the proposed algorithm.

Figure 6.10 shows the execution time of Greedy-Cover and the proposed algorithm.

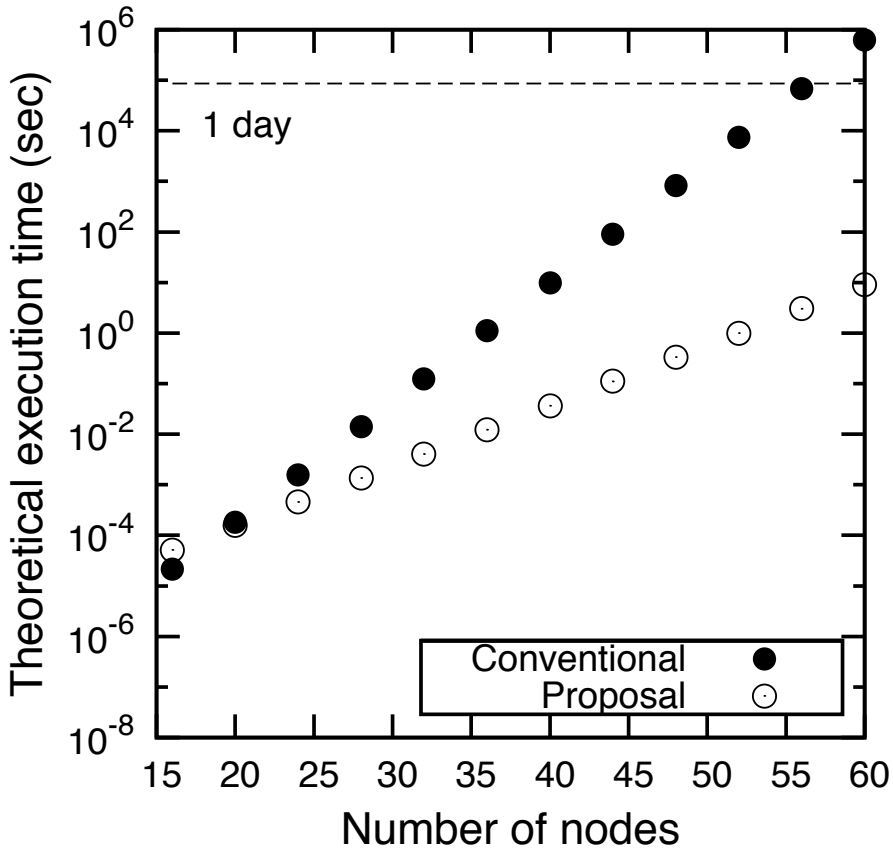


Figure 6.7: Theoretical execution time versus the number of nodes when the number of replicas k is 75 percent of the number of nodes n

The execution time is the average value over 10000 topologies and the parameters are as same as those used in the simulation of Fig. 6.9. The execution time of Greedy-Cover algorithm is always less than that of the proposed algorithm for the same $q(u)$. The execution time of the proposed algorithm is only 1.6 to 3.9 times as large as that of Greedy-Cover algorithm even though the proposed algorithm can always obtain the optimal solution. The factor of the execution time decreases as the number of nodes is increased. Thus, according to Fig. 6.9 and 6.10, the proposed algorithm is effective when the number of nodes is large or $q(u)$ is small.

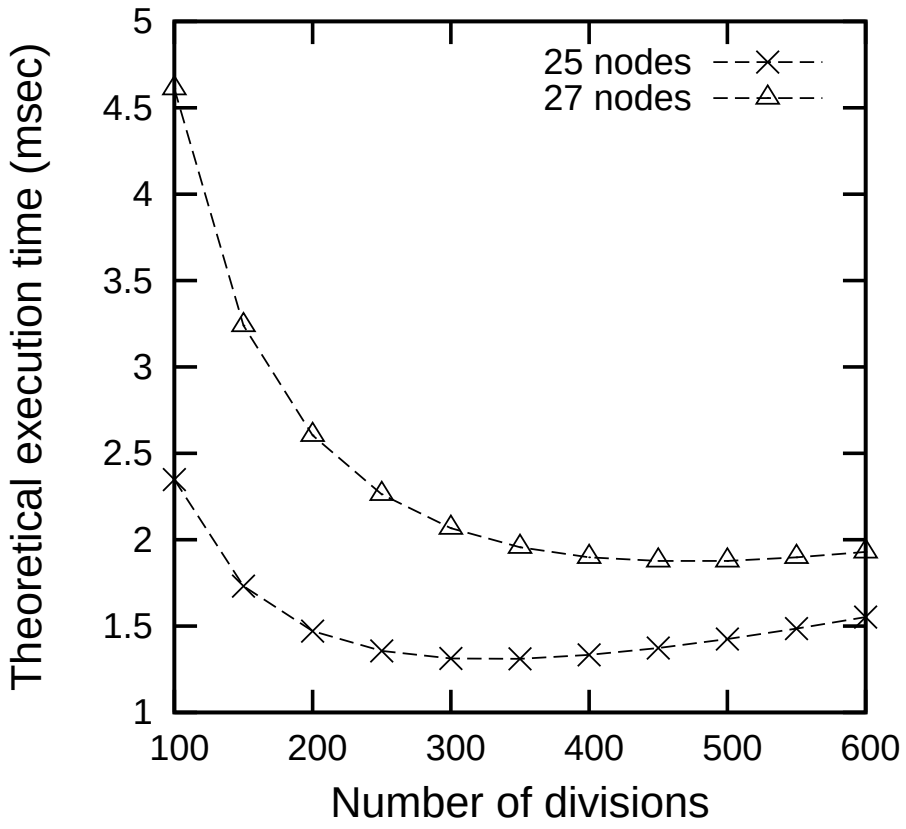


Figure 6.8: Theoretical execution time versus the number of partitions

6.6 Conclusion

The distribution of large contents is a promising application of the Internet, but care is needed to keep the costs feasible. CDN can achieve high resource efficiency in large content distribution if the placement of replica servers is optimal. In order to obtain the optimal solution, I have developed a novel approach that is based on the use of DRPs while the conventional approaches are based on sequential processors. I have also proposed a fast calculation method for exhaustive search that well suits the DRP by fully utilizing the parallelism offered by this type of processor. Our proposed method optimally divides the combinations and subjects the pieces to pipelined processing. I propose a new algorithm that generates any order pattern in combinations that are sorted in ascending order, and derived the optimal number of divisions theoretically. In addition, I implemented the

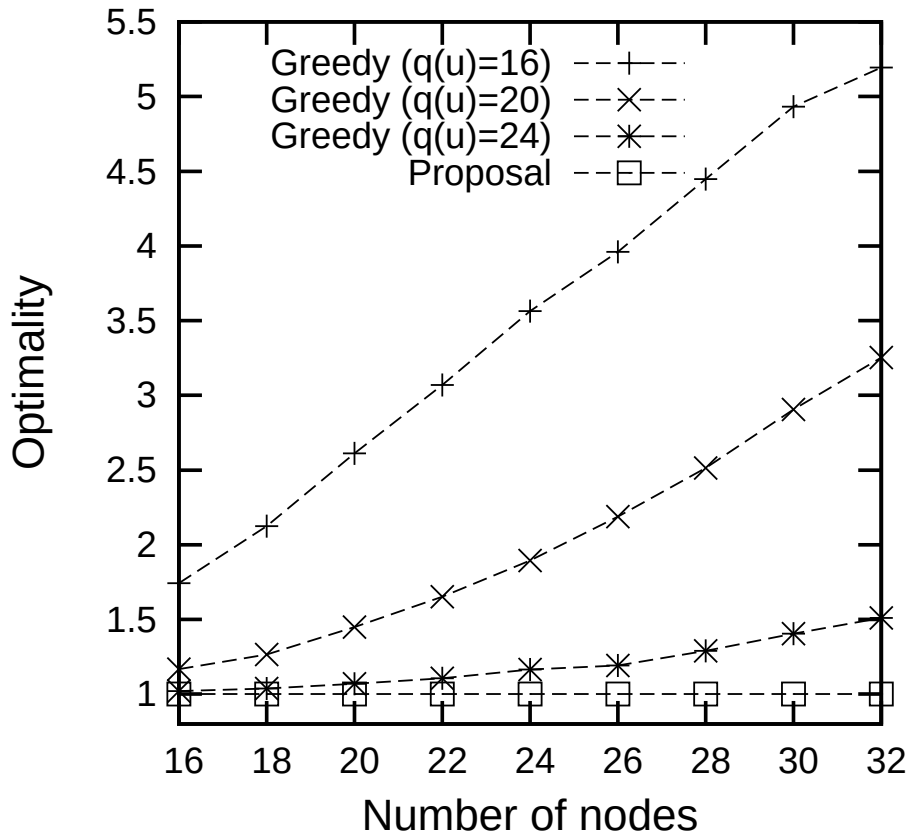


Figure 6.9: Comparison of the optimality of Greedy-Cover and the proposed algorithm

proposed algorithm on a commercially available DRP, DAPDNA-2, developed by IPFlex Inc. While the time complexity of conventional method is $O({}_nC_k)$, the time complexity of the proposed algorithm is $O(\sqrt{{}_nC_k})$.

Experiments have showed that the execution time of the proposed algorithm increases slowly as n increases because DAPDNA-2 calculates in parallel using pipeline operations. When $n = 30$, DAPDNA-2 reduces the execution time by a factor of 40 compared to that needed by a Pentium 4. These results confirm the feasibility of an optimal application framework to distribute large volume data.

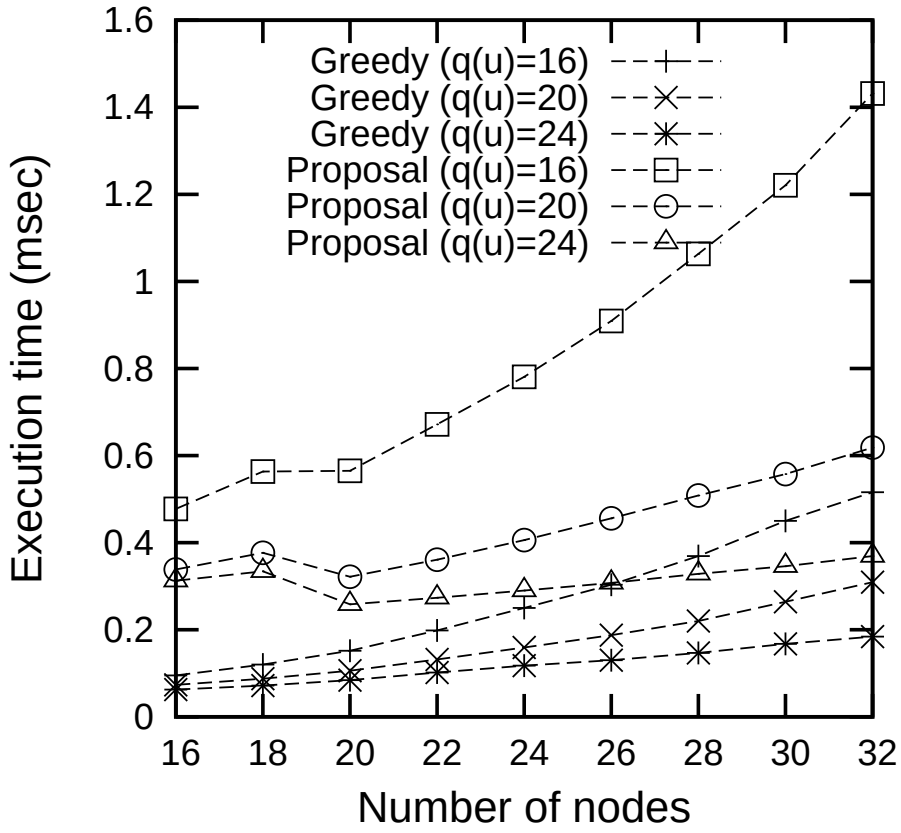


Figure 6.10: Comparison of the execution time of Greedy-Cover and the proposed algorithm

References

[1] "Akamai," http://www.akamai.com/.

[2] "Mirror imge," http://www.mirror-image.com/.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[4] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Science*, vol. 9, pp. 256–278, 1974.

[5] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the internet," in *INFOCOM 2001*, 2001.

[6] M. Karlsson, C. Karamanolis, and M. Mahalingam, "A framework for evaluating replica placement algorithm," HP Laboratories Palo Alto, Tech. Rep., Aug. 2002.

[7] M. Karlsson and M. Mahalingam, "Do we need replica placement algorithms in content delivery netowrks?" in *The International Workshop on Web Content Caching and Distribution (WCW)*, Aug. 2002, pp. 117–128.

[8] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-informed internet replica placement," *Computer Communications*, vol. 25, no. 4, pp. 384–392, Mar. 2002.

[9] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, Mar. 2002.

[10] X. Tang and J. Xu, "Qos-aware replica placement for content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 921–932, Oct. 2005.

[11] H. Wang, P. Liu, and J.-J. Wu, "A qos-aware heuristic algorithm for replica placement," in *Grid Computing 7th IEEE/ACM International Conference*, Sept. 2006, pp. 96–103.

[12] "IPFlex dynamically reconfigurable processor, DAPDNA-2," http://www.ipflex.com/, 2005.

[13] M. Beeler, R. W. Gosper, and R. Schroeppel, "Hakmem," http://www.inwap.com/pdp10/hbaker/hakmem/hakmem.html, Sept. 1972.

[14] M. Platzner and G. D. Micheli, "Acceleration of satisfiability algorithms by reconfigurable hardware," in *8th International Workshop on Field Programmable Logic and Applications (FPL98)*, 1998, pp. 69–78.

[15] P. Zhong, M. Martonosi, P. Ashar, and S. Malik, "Using configurable computing to accelerate boolean satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 6, pp. 861–868, June 1999.

[16] M. Abramovici and J. T. D. Sousa, "A SAT solver using reconfigurable hardware and virtual logic," *Journal of Automated Reasoning*, vol. 24, no. 1–2, pp. 5–36, Feb. 2000.

[17] T. Suyama, M. Yokoo, H. Sawada, and A. Nagoya, "Solving satisfiability problems using reconfigurable computing," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 9, no. 1, pp. 109–116, Feb. 2001.

[18] "NetworkX," http://networkx.lanl.gov/.

Chapter 7

Conclusion

There are four requirements for next generation backbone network. First, high speed and large capacity is a major requirement for next generation backbone network because the traffic volume of the Internet has been increasing for last 10 years rapidly. Second, scalability is required since the number of the Internet users has been increasing recently and so many devices are connected to the Internet in the era of ubiquitous computing. In addition, many new types of applications have been emerged for past few years, and these applications demand different QoS constraints. Therefore, as the third requirement, traffic engineering capability is essential to support QoS and to utilize network resources efficiently. Finally, an application framework to distribute large volume data is important in next generation backbone network. To satisfy these four requirements, efficient data transport technologies for next generation backbone network were proposed in this thesis.

Chapter 3 focused on high speed and large capacity transport. Wavelength division multiplexing is a key technology to realize high speed and large capacity transport since bandwidth can be increased easily just by introducing new wavelengths. Wavelength-routed networks is suitable network architecture for all optical network to remove the bottleneck of electrical processing. However, the wavelength continuity constraint, which leads to high blocking probability, is a major problem of wavelength-routed network. Wavelength converters are employed to relax the wavelength continuity constraint, but the range of wavelength conversion is limited under the current technologies. In this chapter, the wavelength assignment scheme for wavelength-routed network with limited

range wavelength converters was proposed in order to reduce the cost of all optical network. The proposed scheme assign a center wavelength for a long hop connection and an edge wavelength for a short hop connection. The proposed scheme considers the number of hops in a connection request, and offers low blocking probability than First-Fit assignment. The results of computer simulations show that the proposed wavelength assignment reduce the total number of wavelength conversions and it can reduce the number of wavelength converters with negligible performance degradation. As a result, the proposal can make all optical networks cheaper.

Chapter 4 dealt with the scalability issue in next generation backbone network. Wide area layer 2 network based on Ethernet technology has been attractive for carrier recently due to the cost effectiveness of Ethernet. In wide area Ethernet, a connection between users is established with VLAN technology, but the maximum number of connections in a whole network is very limited since only 12 bits are assigned to the field of a VLAN tag and the tag must be globally unique. To expand the scalability of Ethernet, VLAN tag swapping was proposed in this chapter. Distributed VLAN tag resource management can be applied in VLAN tag swapped Ethernet, and the tag can be reused in a different link. Consequently, the restriction of the number of connections is practically removed and the flexibility increases. In addition, the prototype Ethernet switch with VLAN tag swapping functionality was implemented, and the interoperability experiments between my implementation and Ghent University's implementation was successfully demonstrated. This achievement confirmed that VLAN tag swapping is an effective solution to extend the scalability of wide area layer 2 network.

An issue to introduce sophisticated traffic engineering is high computational complexity of path calculation, and it was investigated in Chapter 5. In this chapter, the parallel shortest path algorithm, called Multi-route Parallel Search Algorithm (MPSA), suitable for dynamically reconfigurable processor (DRP) was proposed to speed up the shortest

path calculation. The proposal takes advantage of parallelism of DRP, and searches multiple paths simultaneously. As a result, it reduces the execution time of shortest path calculation to 2.8 percent compared with the popular shortest path algorithm, Dijkstra's algorithm. To confirm the effectiveness of the proposal, the proposed algorithm was implemented on the actual DPR, DAPDNA-2. The proposed algorithm and the implemented off-loading engine can be applied to future network sophisticated traffic engineering.

Chapter 6 focused on an application level framework for distributing large volume data. Content Delivery Network (CDN) had been proposed to improve the users' download speed and to reduce the load of servers. In CDN, replica placement problem is an issue since it affects the performance and storage constraint. Greedy algorithms are widely studied due to its small computational complexity, but there is no greedy algorithm that can always obtains the optimal replica placement pattern. In this chapter, the replica placement scheme that can obtain the optimal solution within practical time was proposed for establishing the optimal application framework for large data distribution. A fast calculation method for exhaustive search that well suits the DRP by fully utilizing the parallelism offered by this type of processor was proposed. The proposed method optimally divides the combinations and subjects the pieces to pipelined processing. We propose a new algorithm that generates any order pattern in combinations that are sorted in ascending order, and derived the optimal number of divisions theoretically. Experiments have showed that the execution time of the proposed algorithm increases slowly as n increases because DAPDNA-2 calculates in parallel using pipeline operations. When $n = 30$, DAPDNA-2 reduces the execution time by a factor of 40 compared to that needed by a Pentium 4. These results confirm the feasibility of an optimal application framework to distribute large volume data.

As an overall conclusion, this dissertation contributes realizing next generation backbone network which have the following characteristics: high speed and large capacity,

scalability, traffic engineering capability, and an application framework for large data distribution.

List of the Related Papers

Journal papers

Papers Related to this Ph.D. Dissertation

(1) **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Wavelength Assignment Scheme for WDM Networks with Limited-Range Wavelength Converters, " Journal of Optical Networking, Optical Society of America, Vol. 5, No. 5, pp. 410–421, May, 2006.

(2) **Sho Shimizu**, Hiroyuki Ishikawa, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Resource Minimization Method Satisfying Delay Constraint for Replicating Large Contents," IEICE Transactions on Communications, Vol. E92-B, No. 10, pp. 3102–3110, October 2009.

(3) **Sho Shimizu**, Wouter Tavernier, Kou Kikuta, Masahiro Nishida, Daisuke Ishii, Satoru Okamoto, Didier Colle, Mario Pickavet, Piet Demeester, and Naoaki Yamanaka, "Interoperability Experiment of VLAN Tag Swapped Ethernet and Transmitting High Definition Video through the Layer-2 LSP between Japan and Belgium," IEICE Transactions on Communications (Letter), Vol. E93-B, No. 3, March 2010 (Accepted, will appear in March 2010).

Other Papers

(1) Satoru Okamoto, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Frame Loss Evaluation of Optical Layer 10 Gigabit Ethernet Protection Switching using PLZT Optical Switch System," IEICE Transactions on Communications, Vol. E92-B, No. 3, pp. 1017–1019, March 2009.

International conference papers

(1) **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "A Wavelength Assignment Considering the Number of Hops in Limited-Range Wavelength-Routed Networks," Ninth International Symposium on Contemporary Photonics Technology (CPT 2006), pp.104-105, Tokyo, Japan, January. 2006. (Presented by Sho Shimizu)

(2) **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Wavelength Assignment Scheme for WDM Networks with Limited-Range Wavelength Converters," 2006 IEEE International Conference on Communications (ICC 2006), OS13.4, Istanbul, Turkey, June 2006. (Presented by Sho Shimizu).

(3) Tomohiro Tsuji, Junichiro Honma, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "A New Accelerated Download Mechanism for Rich Contents using Prefetching Proxy with Automatic Optimal Mirror Selection and Protocol Conversion," Tenth International Symposium on Contemporary Photonics Technology (CPT 2007), Tokyo, Japan, January 2007 (Presented by Tomohiro Tsuji).

(4) Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Parallel Shortest Path Searching Algorithm on Dynamically Reconfigurable Processor," Tenth International Symposium on Contemporary Photonics Technology (CPT 2007), Tokyo, Japan, January 2007 (Presented by Hiroyuki Ishikawa).

(5) Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "New Parallel Shortest Path Searching Algorithm based on Dynamically Reconfigurable Processor DAPDNA-2," 2007 IEEE International Conference on Communications (ICC 2007), Glasgow, Scotland, June 2007 (Presented by Hiroyuki Ishikawa).

(6) Tomohiro Tsuji, Junichiro Honma, **Sho Shimizu**, Yutaka Arawaka, and Naoaki Yamanaka, "Prefetching Protocol Proxy with Optimal Mirror Selection and Burst Transmission," 12th OptoElectronics and Communications Conference, pp. 544–545, Yokohama, Japan, July 2007 (Presented by Tomohiro Tsuji).

(7) **Sho Shimizu**, Taku Kihara, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "A Prototype of a Dynamically Reconfigurable Processor Based Off-loading Engine for Accelerating the Shortest Path Calculation with GNU Zebra," International Conference on High Performance Switching and Routing 2008 (HPSR 2008), pp. 131–136, Shanghai, China, May 2008 (Presented by Sho Shimizu).

(8) Midori Terasawa, Masahiro Nishida, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "Fast Fault Recovery Method with Pre-established Recovery Table for Wide Area Ethernet," International Conference on Photonics in Switching (PS 2008), Session S-02-3, Hokkaido, Japan, August 2008 (Presented by Midori Terasawa).

(9) **Sho Shimizu**, Taku Kihara, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Hardware Based Scalable Path Computation Engine for Multilayer Traffic Engineering in GMPLS Networks," 34th European Conference on Optical Communication (ECOC 2008), Vol. 4, pp. 113–114, Brussels, Belgium, September 2008 (Presented by Sho Shimizu).

(10) Satoru Okamoto, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Experi-

ment of the In-band Message Communication Channel for GMPLS Controlled Ethernet," 34th European Conference on Optical Communication (ECOC 2008), No. P.5.2, vol. 5, pp. 177-178, Brussels, Belgium, September 2008 (Presented by Satoru Okamoto).

(11) Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Fast Replica Allocation Method by Parallel Calculation on DAPDNA-2," The 14th Asia-Pacific Conference on Communications (APCC 2008), No. 15-PM1-F-2, Tokyo, Japan, October 2008 (Presented by Yutaka Arakawa).

(12) Masahiro Nishida, Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "Adaptive Resource Reservation Protocol for High-speed Resource Information Advertisement," The 14th Asia-Pacific Conference on Communications (APCC 2008), No. 15-PM1-E-4, Tokyo, Japan, October 2008 (Presented by Masahiro Nishida).

(13) Taku Kihara, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Fast Link-Disjoint Path Algorithm on Parallel Reconfigurable Processor DAPDNA-2," The 14th Asia-Paciffic Conference on Communications (APCC2008), No. 15-PM1-C-4, Tokyo, Japan, October 2008 (Presented by Taku Kihara).

(14) Shan Gao, Taku Kihara, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Traffic Engineering based on Experimentation in On-chip Virtual Network on Dynamically Reconfigurable Processor," IEEE International Student Paper Contest, pp. 90–95, Seoul, Korea, November 2009 (Presented by Shan Gao).

(15) Midori Terasawa, Masahiro Nishida, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "Recover-Forwarding Method In Link Failure With Pre-established Recovery Table For Wide Area Ethernet," 009 International Conference

on Communications (ICC 2009), Session NGN-P1, Dresden, Germany, June 2009 (Presented by Midori Terasawa).

(16) Shan Gao, Taku Kihara, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Akifumi Watanabe, "A Novel Traffic Engineering Method using On-Chip Diorama Network on Dynamically Reconfigurable Processor DAPDNA-2," International Conference on High Performance Switching and Routing 2009 (HPSR 2009), Paris, France, June 2009 (Presented by Shan Gao).

(17) **Sho Shimizu**, Shan Gao, Daisuke Ishii, and Naoaki Yamanaka, "Newly Structured Router Network Architecture using Cloud Control Plane and Forwarding Plane," 2nd International Workshop on the Network of the Future, Session 10-10, Hawaii, USA, December 2009 (Presented by Sho Shimizu).

(18) Shota Yamada, Midori Terasawa, Yusuke Okazaki, **Sho Shimizu**, Daisuke Ishii, Satoru Okamoto, and Naoaki Yamanaka, "A Study of TCP over SCTP Parallel Networking and Parallel Route Selection Approach for Mass Data Transfer Applications," Optical Network Design and Modeling (ONDM 2010), Session 7-1, Kyoto, Japan, February 2010 (Accepted, will be presented by Shota Yamada).

Technical reports

(1) **Sho Shimizu**, Takanori Ito, Yutaka Arakawa, and Naoaki Yamanaka, "A Wavelength Assignment Scheme for WDM Networks with Limited Range Wavelength Converters," Technical Report of IEICE, Vol. 104, No. 690, pp. 329–332, March 2005 (Presented by Sho Shimizu).

(2) **Sho Shimizu**, Takanori Ito, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "A Study on Shortest Path Routing Algorithm on Data-flow Parallel Reconfigurable

Processor DAPDNA2," Technical Report of IEICE, Vol. 105, No. 451, pp. 1–6, December 2005 (Presented by Sho Shimizu).

(3) Hiroyuki Ishikawa, **Sho Shimizu**, Takanori Ito, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Shortest Path Algorithm on Parallel Reconfigurable Processor DAPDNA-2," Technical Report of IEICE, Vol. 105, No. 627, pp. 17–20, March 2006 (Presented by Hiroyuki Ishikawa).

(4) Masahiro Tateno, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Construction of Overlay Network Considering User Perference in Peer-to-Peer Systems," Technical Report of IEICE, Vol. 105, No. 628, pp. 143–146, March 2006 (Presented by Masahiro Tateno).

(5) Tomohiro Tsuji, Junichiro Honma, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Prefetching Protocol Proxy with Optimal Mirror Selection and Burst-Transmission," Technical Report of IEICE, Vol. 105, No. 667, pp. 115-119, March 2006 (Presented by Tomohiro Tsuji).

(6) Masahiro Nishida, Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "High-speed Resource Information Advertising Method in GMPLS," Technical Report of IEICE, Vol. 107, No. 188, pp. 33–38, August 2007 (Presented by Masahiro Nishida).

(7) Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "Fast Calculation Method of Set Cover Problem on Parallel Reconfigurable Processor DAPDNA-2," Technical Report of IEICE, Vol. 107, No. 418, pp. 67–72, January 2008 (Presented by Hiroyuki Ishikawa).

(8) Masahiro Nishida, **Sho Shimizu**, Daisuke Ishii, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "Approach for Flexible-Switch for Next-Generation Layer-

2 Networks," Technical Report of IEICE, Vol. 108, No 84, pp. 19–24, June 2008 (Presented by Masahiro Nishida).

(9) Taku Kihara, **Sho Shimizu**, Shan Gao, Yutaka Arakawa, and Naoaki Yamanaka, "A Study on High Speed Method of Link-Disjoint Path Calculation," Technical Report of IEICE, Vol. 108, No. 183, pp. 19–24, August 2008 (Presented by Taku Kihara).

(10) Satoru Okamoto, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "Experiment of the in-band GMPLS message Channel for GELS network," Technical Report of IEICE, Vol. 108, No. 183, pp. 43–48, August 2008 (Presented by Satoru Okamoto).

(11) Shan Gao, Taku Kihara, **Sho Shimizu**, Yutaka Arakawa, Naoaki Yamanaka, and Kosuke Shiba, "A Novel Network Optimization Method using On-Chip Virtual Network on Dynamically Reconfigurable Processor DAPDNA-2," Technical Report of IEICE, Vol. 108, No. 300, pp. 69–74, November 2008 (Presented by Shan Gao).

(12) Taku Kihara, **Sho Shimizu**, Shan Gao, Yutaka Arakawa, Naoaki Yamanaka, and Akifumi Watanabe, "Fast Solution of Link Disjoint Path Algorithm on Parallel Reconfigurable Processor DAPDNA-2," Technical Report of IEICE, Vol. 108, No. 414, pp. 201–206, January 2009 (Presented by Taku Kihara).

(13) Kazuko Yonezawa, Midori Terasawa, Masahiro Nishida, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto and Naoaki Yamanaka, "Queuing Method for Guaranteed Delay Jitter in Wide Area Ethernet," Technical Report of IEICE, Vol. 108, No. 455, pp. 7–12, March 2009 (Presented by Kazuko Yonezawa).

(14) Shota Yamada, Midori Terasawa, **Sho Shimizu**, Daisuke Ishii, Satoru Okamoto, and Naoaki Yamanaka, "A Study of TCP over SCTP Parallel Networking and Parallel Route Selection Approach for Mass Data Transfer Applications," Technical

Report of IEICE, Vol. 109, No. 172, pp. 19–24, August 2009 (Presented by Shota Yamada).

Oral Presentations

(1) Masahiro Nishida, Hiroyuki Ishikawa, **Sho Shimizu**, Yutaka Arakawa, Satoru Okamoto, and Naoaki Yamanaka, "Unreserved Resource Information Advertisement Method in GMPLS," 12th OptoElectronics and Communications Conference Student Workshop, July 2007 (Presented by Masahiro Nishida).

(2) Satoru Okamoto, **Sho Shimizu**, Yutaka Arakawa, and Naoaki Yamanaka, "In-band GMPLS Message Communication Channel for GELS Network," The Society Conference of IEICE, No. BS-9-9, August 2008 (Presented by Satoru Okamoto).

Acknowledgments

This dissertation has been written under the direction and guidance of Prof. Naoaki Yamanaka in Department of Information and Computer Science, Keio University, Japan.

My sincere gratitude and deepest appreciation should be first given to my supervisor Prof. Naoaki Yamanaka for their valuable suggestions, guidance and continuous encouragements throughout my works. With the guidance of Prof. Yamanaka, I did good studies and got splendid experiences in the Ph.D. course.

I am deeply grateful to Prof. Piet Demeester in INTEC Broadband Communication Networks research group (IBCN), Faculty of Engineering, Ghent University, Belgium. He gave insightful comments and suggestions for about 3 month in total (January 2009 to March 2009, and October 2009 to November 2009) in the short term overseas research programs of Global COE. Chapter 4 is a part of works during the programs in IBCN under his valuable directions and guidances.

I owe a great deal of thanks to the members of dissertation committee, Prof. Iwao Sasase in Deparment of Information and Computer Science, Keio University, Japan, Prof. Hideharu Amano in Department of Information and Computer Science, Keio University, Japan, and Assoc. Prof. Hiroshi Shigeno in Department of Information and Computer Science, Keio University, Japan for their comments, suggestions, and careful and critical reading of this dissertation. I want to thank Prof. Piet Demeester again for joining the dissertation committee.

Assoc. Prof. Satoru Okamoto of Yamanaka Lab., Department of Information Technol-

ogy, Keio University, Japan, gave insightful comments and suggestions, especially about GMPLS and wide area layer 2 network architecture. His support was invaluable for the achievements in Chapter 4.

I would like to thank to the colleagues who joined Yamanaka Lab. before me, Dr. Yutaka Arakawa, Assist. Prof., Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, gave me constructive comments and warm encouragements throughout the years in Yamanaka Lab. We had good discussions not only on research topics but also on wide range topics about information and communication technologies. Dr. Kohei Okazaki of NEC Corporation taught me what Ph.D. students are. He also gave me a lot of comments and suggestions about attitude towards research and lifestyle of a Ph.D. student in mealtime. Dr. Daisuke Ishii, Assist. Prof., Yamanaka Lab., Department of Information and Computer Science, Keio University provided a lot of technical advices about implementations of the GMPLS software. I spent much time with him especially when I was in the third year of Ph.D. course.

My deepest appreciation goes to the colleagues who joined Yamanaka Lab. with me, Mr. Masahiro Hayashitani of NEC Corporation, and Mr. Junichiro Honma of Sony Corporation are precious for me. We had been sharing good and bad time for about 3 years from the bachelor to master course in Yamanaka Lab.

I would like to express my gratitude to the colleagues in Sasase Lab., Mr. Motoki Shirasu of Ericsson Japan, Mr. Kazuhiko Hasegawa of Japan Broadcasting Corporation, Mr. Koki Oba of Denso Corporation, Mr. Takamasa Isohara of KDDI Corporation, and Mr. Tomoki Kimura. They encouraged me to enter, continue, and finish the Ph.D. course. Some of them came to a party before I went to Ghent in the beginning of 2009, and I was very happy.

I want to thank the colleagues who joined Yamanaka Lab. one year after me: Mr. Hiroyuki Miyagi of Nippon Telegraph and Telephone East Corporation, Mr. Hiroyuki

Ishikawa of Kansai Electric Power Co. Inc., Mr. Tomohiro Tsuji of TV Asahi Corporation, Mr. Teruo Kasahara of Nomura Research Institute, Ltd., Mr. Masahiro Tateno of u10 Networks, Inc. I would like to give special thanks to Mr. Hiroyuki Miyagi and Mr. Hiroyuki Ishikawa. I had long time with Mr. Hiroyuki Miyagi not only in the lab but also in private time, for instance, watching Japan Cup Cycle Road Race in Utsunomiya for many times. Mr. Hiroyuki Ishiwaka belonged to the same research group. Without the contributions of Mr. Hiroyuki Ishikawa, the achievements of Chapter 5 and Chapter 4 could not be done.

Mr. Wouter Tavernier, Ph.D. student in IBCN, Ghent University, Belgium is my very precious colleague in Ghent University. He contributed to the interoperability experiments of VLAN tag swapped Ethernet between Ghent University and Keio University described in Chapter 4. He kindly provided his source codes to me, and it helped to understand his system and to conduct experiments smoothly. Our collaboration was important experience for me. I am very thankful to Mr. Wouter Tavernier.

Special thanks also to Dr. Brecht Vermeulen, Post doctoral fellow in IBCN, Ghent University, Belgium. He is a system administrator of the test lab of IBCN, and he set up special network configurations in order that I could conduct the experiments described in Chapter 4.

I thank the colleagues who joined Yamanaka Lab. two years after me, especially Mr. Ko Kikuta, a Ph.D. student in Keio University, Mr. Masahiro Nishida of NTT Data Corporation, and Mr. Taku Kihara in Nippon Telegraph and Telephone Corporation. Mr. Ko Kikuta made a lot of contributions to prepare for the experiments in Chapter 4. He did work hard into the night to conduct the experiments as Japan side while I was staying in Ghent. I had long time with him especially in 2009 – 2010. Discussions Mr. Masahiro Nishida was a member of layer 2 network research group in Yamanaka Lab. He provided the source codes of VLAN tag swapped Ethernet switch. I modified the source codes

based on his codes. Without his contributions, the experiments in Chapter 4 could not be conducted successfully. Mr. Taku Kihara helped to implement the prototype router based on GNU Zebra with the shortest path off-loading engine in 5. I also thank the other colleagues, Mr. Ryota Usui of NTT Data Corporation, Mr. Mikio Kaneko of Citigroup Global Markets Japan, Mr. Kazuki Irie of Nomura Research Institute, Ltd., Mr. Shinpei Koda of Panasonic Corporation, and Ms. Fumiko Uehara.

I would like to thank the colleagues who joined Yamanaka Lab. three years after me, especially Ms. Midori Terasawa and Mr. Shan Gao. Ms. Midori Terasawa was a colleague of layer 2 network research group in Yamanaka Lab. She also helped to make preparations for the experiments in Chapter 4, and wrote tagging and untagging elements of Click modular router. The achievements of Chapter 4 are also based on her important efforts. Mr. Shan Gao is a colleague of the network design research group in Yamanaka Lab. We had many discussions about dynamically reconfigurable processors and implementations on them. I also thank the other colleagues, Mr. Kazumasa Tokuhashi for providing a lot of knowledges of clothes, shoes, wallet, and other fancy goods, Mr. Yusuke Okazaki for having and joining events and talking in vacant time, Mr. Hirofumi Yamashita, Ms. Motomi Akagi in Hewlett-Packard Japan, Ltd. I thank again Ms. Midori Terasawa, Mr. Shan Gao, Mr. Yusuke Okazaki, and Mr. Hirofumi Yamashita for allowing me to join their graduation trip in March 2010.

My gratitude is given to the colleagues who joined Yamanaka Lab. four years and five years after me: Mr. Shota Yamada, Mr. Junpei Marukawa, Mr. Kunitaka Ashizawa, Mr. Yuki Susa, Ms. Aya Tsurusaki of NTT DoCoMo Inc., Ms. Kazuko Yonezawa of JPMorgan Asset Management Japan Limited, Mr. Jun Matsumoto, Ms. Haruka Yonezu, Mr. Haruki Takahashi, Mr. Kenta Nakahara, Mr. Takehiro Sato.

I thank to other members of Yamanaka Lab., Mr. Takashi Kurimoto, Mr. Hidetoshi Takeshita, Mr. Alexandre Olivier, Mr. Alatengsongbuer, Ms. Jia Zhou. I appreciate the

works of secretaries of Yamanaka Lab., Ms. Yuki Uchiyama, Ms. Kaori Kozakai, Ms. Ayumi Sato, Ms. Haruko Iwama, Ms. Tomoko Kawasaki, and Ms. Aki Kishi.

My deepest appreciation also goes to the colleagues in IBCN, Ghent University, Belgium: Prof. Mario Pickavet, Dr. Didier Colle, Dr. Marc de Leenheer, Dr. Bart Puype, Mr. Dimitri Staessens for having valuable discussions and giving comments and suggestions about researches during my stay in Ghent. I thank also to staffs in IBCN, Mr. Bart De Knijf for technical supports of network systems of IBCN, Ms. Martine Buysse for non technical supports.

I am grateful to room mates in IBCN, Ghent University, Belgium: Mr. Olivier Van Laere, Mr. Wouter Haerick, Mr. Bart Jooris, and Mr. Diter Verslype. They were very kind to me throughout the stay in Ghent. They had Friday drink parties many times, and allowed me to join them. I could get friends with them because of Friday drink. Mr. Philip Leroux is also a important friends in IBCN. He told me a lot of things to live in Ghent.

I appreciate the support from IPlex Inc. especially Mr. Akifumi Watanabe. His contribution is included in the achievement of Chapter 6.

I am very thankful to professors and staffs in Global Center of Excellence "High-Level Global Cooperation for Leading-Edge Platform on Access Spaces" of Keio University for giving me an opportunity to do my researches in IBCN, Ghent University, Belgium and financial supports. The experiences in Belgium made me growing up.

Special thanks to friends in Azabu High School: Mr. Masayuki Suga of Ministry of Internal Affairs and Communications, Mr. Fumitaka Muramatsu of Faculty of Medicine, Osaka University, Japan, and Mr. Hiroyuki Yamanaka of Saiseikai Central Hospital. Special thanks also to friends in Keio University: Mr. Yosuke Hosokawa of Development Bank of Japan Inc., Dr. Hajime Hotta of Fujitsu Laboratories of America, Inc., Mr. Kazuma Tanaka of Honda Motor Co., Lts., Ms. Yoshiko Hashimoto of Future Design

Lab., Ms. Akiko Ueno of Dai Nippon Printing Co., Ltd. Special thanks also to all of colleagues in Keio Bicycle Racing Team and friends related to bicycle activities. I am really grateful to everyone around me for supporting and encouraging.

I gratefully appreciate the financial support of Yoshida Scholarship Foundation and that made it possible to complete the Ph.D. course. Special thanks are given to staffs of Yoshida Scholarship Foundation.

Finally, I would like to express my deepest gratitude to my family especially my father and mother for moral support and warm encouragements. This dissertation could not be completed without their supports and encouragements.

School of Science for Open and Environmental Systems

Keio University

Sho Shimizu

January 15, 2010