NAIST-IS-DD0561018

Doctoral Dissertation

Studies on minimum throughput assurance service in provider provisioned virtual private networks

Masayoshi Shimamura

February 5, 2009

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Masayoshi Shimamura

Thesis Committee:

Professor Suguru Yamaguchi	(Supervisor)
Professor Hideki Sunahara	(Co-supervisor)
Professor Yuji Oie	(Kyushu Institute of Technology)
Associate Professor Youki Kadobayashi	(Co-supervisor)

Studies on minimum throughput assurance service in provider provisioned virtual private networks*

Masayoshi Shimamura

Abstract

This dissertation presents a minimum throughput assurance (MTA) service for providing high quality of service (QoS) in provider provisioned VPNs (PPVPNs). The importance of MTA services increasingly grows because VPNs have been utilized in a wide variety of areas. An essential issue on the MTA service is that two competing demands must be satisfied. While each VPN customer demands the minimum agreed throughput of the customer to a VPN provider, the provider has a demand on the network resource efficiency in order to avoid over-provisioning also known as wasteful solutions. To meet the both demands, we propose two essential mechanisms: the hose bandwidth allocation method and the provisioning algorithm for the feedback-driven traffic control mechanism. The former adopts a VPN hose model as a VPN contract model, and it achieves the both demands when the VPN provider allocates the bandwidth to the VPN customers. On the other hand, the latter is utilized when the provider determines the target bandwidth exceeding the minimum agreed throughput. This algorithm employs a mathematical programming, especially a nonlinear programming (NLP). Through computer simulation runs in the first study, we demonstrate effectiveness, scalability and stability of the hose bandwidth allocation method. In the second study, we show that the proposed provisioning algorithm can be applicable to determine appropriate bandwidth allocation parameters. Through these studies, we show

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561018, February 5, 2009.

that the studies lead to the realization of the MTA service in PPVPNs. Finally, contributions and future issues of this dissertation are discussed.

Keywords:

Minimum throughput assurance (MTA) service, provider provisioned virtual private networks (PPVPNs), quality of service (QoS), hose bandwidth allocation method, provisioning algorithm, nonlinear programming (NLP)

Acknowledgments

First, I wish to express my gratitude to Thesis Committees of my dissertation. I would like to thank Professor Suguru Yamaguchi, my supervisor in our laboratory and Committee Chair, for his support, advice and encouragement. He taught me a professional attitude to research. I would also like to thank Professor Hideki Sunahara for his kind support on my committee. His many valuable comments helped me to finish my dissertation. I would like to express my appreciation for Professor Yuji Oie at Kyushu Institute of Technology for his helpful support on my committee. I was able to improve my dissertation with his valuable comments. I would also like to thank Associate Professor Youki Kadobayashi, my co-supervisor in our laboratory and committee, for his continuous support and assistance. He gave me a great insight for research.

I would like to thank Assistant Professor Takeshi Okuda and Assistant Professor Shigeru Kashihara. Assistant Professor Okuda helped me in the Internet Engineering Laboratory (IPlab), and Assistant Professor Kashihara also took care of me. I with to express my gratitude to Associate Professor Katsuyoshi Iida at Tokyo Institute of Technology for his kind support, advice, encouragement on my research, dissertation and life. Without him, I could not have completed my dissertation. After he left IPlab, he continuously supervised me in spite of distant location. I would also like to thank Associate Professor Hiroyuki Koga at the University of Kitakyushu for his helpful support on my research. His valuable advice and comments always encouraged me. I would like to thank Professor Masato Tsuru at Kyushu Institute of Technology for his insightful comments on my research. He pointed out important issues for my research. I would also like to thank Associate Professor Takeshi Ikenaga at Kyushu Institute of Technology for his support on my researcher's life after I entered Network Design Research Center (NDRC) in Kyushu Institute of Technology.

Regarding my student life, I would like to thank Ms. Natsue Tanida and Ms. Rieko Aoki. Ms. Tanida in IPlab always cheered me up and helped the laboratory life. Ms. Aoki in Iida laboratory supported me when I moved to Tokyo Institute of Technology as Special Research Student.

I would also like to thank all my friends. Especially, I would like to thank Mr. Teruaki Yokoyama, Mr. Takuji Iimura, Mr. Kenji Masui, Mr. Mitsuaki Akiyama, Ms. Yuko Oota and Ms. Chiharu Kitamura. Mr. Yokoyama gave me philosophical perspective on my research. Mr. Iimura gave me encouragement. Mr. Masui helped me when I studied at Tokyo Institute of Technology as Special Research Student. Mr. Akiyama always gave me emotional support and encouragement. Ms. Oota always helped me when I was a doctoral student. My one and only Ms. Kitamura always supported me in my daily life.

Finally, I wish to express my appreciation for my family. My mother Hatsuko Shimamura perpetually helped me whenever I faced various problems. Her longstanding support has enabled me to complete my degree. I would also like to express my gratitude for my pet Genta and Edo. Their great presence cheered me up when I was depressed.

Contents

List of Figures

List of Tables

1. Introduction

The Internet has already been a social network infrastructure and enormous users broadly benefit from the use of the Internet. The benefit in the Internet has constantly increased not only the number of users but also the amount of traffic steadily [1]. Moreover, although few simple applications had been used in the past Internet, the wide variety of applications has been used in the recent Internet. In the past Internet, many users attached paramount importance to the connectivity and they recognized that the Internet provided the best-effort service. In contrast with these preconditions, many users are searching for the communication quality on the recent and future Internet because wide variety of applications need to share the common network infrastructure and the massive number of applications cause crucial congestion. If such a congestion scenario occurs in a business situation, a business company may suffer a serious loss and great damage. Therefore, the Internet representing the essential social network infrastructure should meet the needs of users even in a time of information explosion caused by the wide variety and the massive number of applications.

In a situation where many variety of applications compete against each other, many researchers recognize that network virtualization technologies should be utilized to slice a common network resource for these applications. This is because characteristics of these applications are different from each other, for example, one of them needs broad bandwidth and one of them needs small delay. If these differently characterized applications compete in a congestion point, selfish and greedy applications may deplete the entire network resources and other applications cannot communicate on such a network. Many researchers share the same sense of crisis about the congestion collapse [2, 3, 4, 5, 6] if we do not develop strategy and countermeasure against the network resource starvation. Therefore, in state-of-the-art research focusing on far future, many researchers reconsider the design of the Internet [7, 8, 9]. However, because the researchers cannot drastically deploy most of the novel approaches, they expect growth of the overlay network technologies [10, 11, 12, 9] and network virtualization technologies [13, 14, 8] which slice a single network resource to multiple virtual network resources and assure the use of the sliced virtual network resource. In a future, providers will construct overlay networks using such technologies and will provide quality of service (QoS) to users. In this case, QoS includes wide variety of services: performance assurance, providing availability with fault-tolerance, data aggregation, and service differentiation (classification).

These prospective network virtualization technologies have been derived from virtual private network (VPN) technologies. Since VPN technologies can slice a network flexibly, many business organizations can virtually construct their own communication networks connecting multiple distant sites transparently and inexpensively. From the viewpoint of the social stream, to obtain the performance assurance in terms of high QoS, these business organizations have a strong demand on a throughput assurance in their own private networks on the Internet. According to the information and communications report [15] in the ministry of internal affairs and communications in Japan, 89.5 percent of business organizations utilize their own communication networks and 39.8 percent of them exploit the benefit of VPNs. Especially, 57.4 percent of the VPN users utilize provider provisioned VPNs (PPVPNs) which assure the high QoS to the users by VPN providers. In this way, enormous VPN users expect high QoS networks in VPNs on the recent and future Internet.

1.1 Issue of efficient network resource distribution

Although many users and researchers have strong expectations for high QoS networks by VPN and/or network virtualization technologies, we have to raise an alert over the communication quality in such networks. Even if PPVPNs and network virtualization technologies are expected to improve the communication quality, a crucial issue remains because multiple virtualized networks eventually compete to obtain a single common network resource. In such a roughly sliced network resource, QoS providing mechanisms are eventually needed. It means that when similar type of applications in a common service compete to utilize the common resource, this competition causes crucial congestion and degrades the communication quality. Although many researchers have been tackling this problem and have been proposed network resource distribution mechanisms for transport flows and end hosts [16, 17, 18], these mechanisms cannot be adaptable to a time of information explosion. If a provider virtually slices network resource and distributes these virtualized resources to applications or end hosts individually (i.e., really fine-grained service), this service is equivalence to a private leased line for these applications and end hosts. In this case, the provider needs to manage the massive number of the fine-grained virtualized network resources, and then the provider cannot achieve such a service. Therefore, to achieve efficient network resource distribution for high QoS on the current best-effort and future Internet, efficient quality of service (QoS) traffic control mechanisms for roughly virtualized networks should be needed.

In PPVPNs with roughly virtualized network resource, a customer demands the required throughput to a provider in their contract. The provider must allocate the bandwidth exceeding the agreed throughput. A conventional solution which is the simplest approach to achieve a throughput assurance is over-provisioning, which the provider retains much network resources than all customers' demand. In this case, providers can absolutely assure and guarantee sufficient network resource distribution to the customers. However, since the utilization efficiency of network resources is really low, research improving the network efficiency to avoid over-provisioning has been studied [19, 20]. Even if the provider has redundant resources, efficient consumption of limited network resources makes two benefits. The first is that a provider can avoid much excessive cost for network resources; the second is that efficient consumption can improve QoS due to assure the available bandwidth and redistribute the excess resources to the customers.

Difficulties to achieve a throughput assurance improving QoS are that a provider must meet all customers' demand accommodating traffic condition on its own network and the provider should also effectively utilize the bandwidth representing limited network resources. To maximize the network resource efficiency under the constraints, the provider needs to make an adequate bandwidth allocation policy. Following two important steps are necessary to implement this policy: the first is a network provisioning step which determines the allocated bandwidth for each customer; the second is a bandwidth allocation step which needs to precisely allocate the target bandwidth determined by the provisioning step.

1.2 Contributions in this dissertation

In this dissertation, we consider PPVPNs as a kind of sophisticated overlay networks being the underlying basis of the future Internet. In prospective PPVPNs, we cope with the above-described issue from two perspectives. We present two mechanisms with online and offline approaches to achieve a minimum throughput assurance (MTA) service in which a provider meets the minimum agreed throughput for all customers and the provider maximizes the network resource efficiency in any traffic condition. In detail for the MTA service, the customer can obtain the agreed throughput immediately when traffic of the customer enters a network managed by the provider. Moreover, the allocated bandwidth of a customer is redistributed to other customers when traffic of the customer stops. This bandwidth allocation policy can achieve the MTA service which includes both bandwidth allocation satisfying the minimum agreed throughput and the network efficiency.

First key mechanism described in Chapter 2 is a bandwidth allocation method which enables to allocate the adequate bandwidth in online and accommodate instantaneous bursty Internet traffic. In this novel method, we integrate a weighted proportional fair rate allocation (WPFRA) method [21, 22], which uses feedback notifications and allocates the bandwidth accommodating a bottlenecked link inside a network, and a VPN hose model [23] which aggregates multiple site-to-site connections as a hose to improve the network efficiency. This mechanism can assure to divide the available bandwidth proportionally even if traffic fluctuates dynamically in the burst manner.

Second key mechanism described in Chapter 3 is a novel provisioning algorithm for achieving the MTA service. This algorithm can determine the adequate allocated bandwidth in offline even if traffic condition dynamically changes. In the proposed provisioning algorithm, we formulate a constraint problem and utilize a mathematical programming [24], especially a nonlinear programming (NLP). This algorithm can accommodate long-term bursty Internet traffic and can determine the adequate bandwidth exceeding the minimum agreed throughput for each customer.

These two mechanisms can solve the issue of efficient network distribution in a roughly sliced network resource. If we lack either mechanism, we cannot achieve the MTA service for high QoS which is essential basis on the future Internet. Moreover, these mechanisms can achieve the responsibility as the social network infrastructure. Therefore, the research in this dissertation positions a milestone for achieving the essential MTA service in the future.

1.3 Organization

In Chapter 1, we mentioned the possibility and issues about the future Internet. Through the rest of this dissertation, to achieve the essential MTA service, we describe the efforts of this dissertation.

In Chapter 2, we present a bandwidth allocation method for MTA service. Chapter 3 describes a provisioning algorithm for MTA service. Finally, after the discussion in Chapter 4, conclusion and future study in terms of studies in Chapter 2 are given in Chapter 5.

2. Bandwidth allocation method for MTA service

In Chapter 2, to achieve the MTA service, we consider PPVPNs as a kind of sophisticated overlay networks, and then we cope with a study for a bandwidth allocation mechanism for the MTA service in PPVPNs. The bandwidth allocation mechanism is essential QoS traffic control mechanisms in online.

2.1 Introduction

As described in Chapter 1 , provider provisioned virtual private networks (PPVPNs) are widely used by many enterprise organizations as private information networks connecting distant sites because a provider handles the VPN configuration and management for subscribers. Since VPN subscribers share the bandwidth representing limited network resources among all subscribers, mechanisms for assuring the quality of service (QoS), especially the minimum throughput, are strongly desired because TCP is necessary in the enterprise applications and it creates bursty and elastic traffic.

For assuring the minimum throughput to subscribers, a provider conventionally uses the *customer-pipe model* in terms of bandwidth contract. In this model, a subscriber contracts with the provider for a one-to-one connection, called the customer pipe, to other sites in the VPN. Therefore, the provider must assure the minimum throughput into numerous connections, such as $n(n-1)/2$, if the VPN has a full mesh topology, where n is the number of sites in the VPN.

To solve this problem, a *hose model* [23] has been proposed as a novel VPN service model in PPVPNs. The main target of the hose model is a customer having multiple sites, and the model has been proposed and proven to be effective for network resource efficiency and configuration complexity [23, 25]. Instead of using pipes between source and destination sites, this model uses hoses, which are bundles of pipes. By using hoses, the hose model can reduce the number of connections to n [1].

[1] The definition of the hose model is a bandwidth contract model in which a VPN provider offers the throughput assurance for subscribers having multiple destination sites. In the model, the hose is defined as the aggregation of customer pipes between source site and other sites

An illustration about the customer-pipe and hose models is shown in Fig. 1. In both models, there are two VPN subscribers A and B, where each represents a company with three sites which required to be interconnected by a VPN. In the customer-pipe model, the provider must manage two customer pipes to sites A2 and A3 for site A1 and another two pipes to sites B2 and B3 for site B1, whereas in the hose model only one hose needs to be allocated and managed for each site A1 and B1. Here, the hose represents the aggregation of all customer pipes connected to the site. This aggregation eases the configuration complexity of large-scale VPNs. Furthermore, the network utilization increases because network resources are shared among sites of the same subscriber. In this way, the hose model provides high efficiency in VPN services.

In order to clearly describe the difference in how to allocate the bandwidth between the customer-pipe and hose models, in Table 1, we show a simple example of the allocated bandwidth based on Fig. 1. In this example, sites A2, A3, B2, and B3 contract for the minimum agreed throughput of 30, 30, 20, and 20, respectively, in the customer-pipe model. On the other hand, subscribers A and B contract for the minimum agreed throughput of 60 and 40, respectively, in the hose model. The contracted bandwidth in the hose model can be shared among all the active sites belonging to the same subscriber. For example, if site A3 becomes idle, its excess bandwidth is redistributed to the other sites A2, B2, and B3 in the customer-pipe model, while the excess bandwidth can be reallocated only to site A2 in the hose model.

Although the hose model has been proven to be effective for network resource efficiency, no mechanism for assuring the minimum throughput in the hose model has been proposed because the method of allocating the bandwidth of hoses to achieve QoS assurance is not clear. For QoS assurance in the hose model, we consider the following minimum throughput assurance (MTA) service. A subscriber can obtain much more throughput than the minimum agreed throughput in its hose during a period with no congestion. During a period of congestion, at least the minimum agreed throughput is provided as an average in a certain period. Therefore, an MTA service can provide greater throughput predictability than

in the subscriber. The provider allocates the bandwidth to the hose, i.e., traffic toward all destination sites in the subscriber.

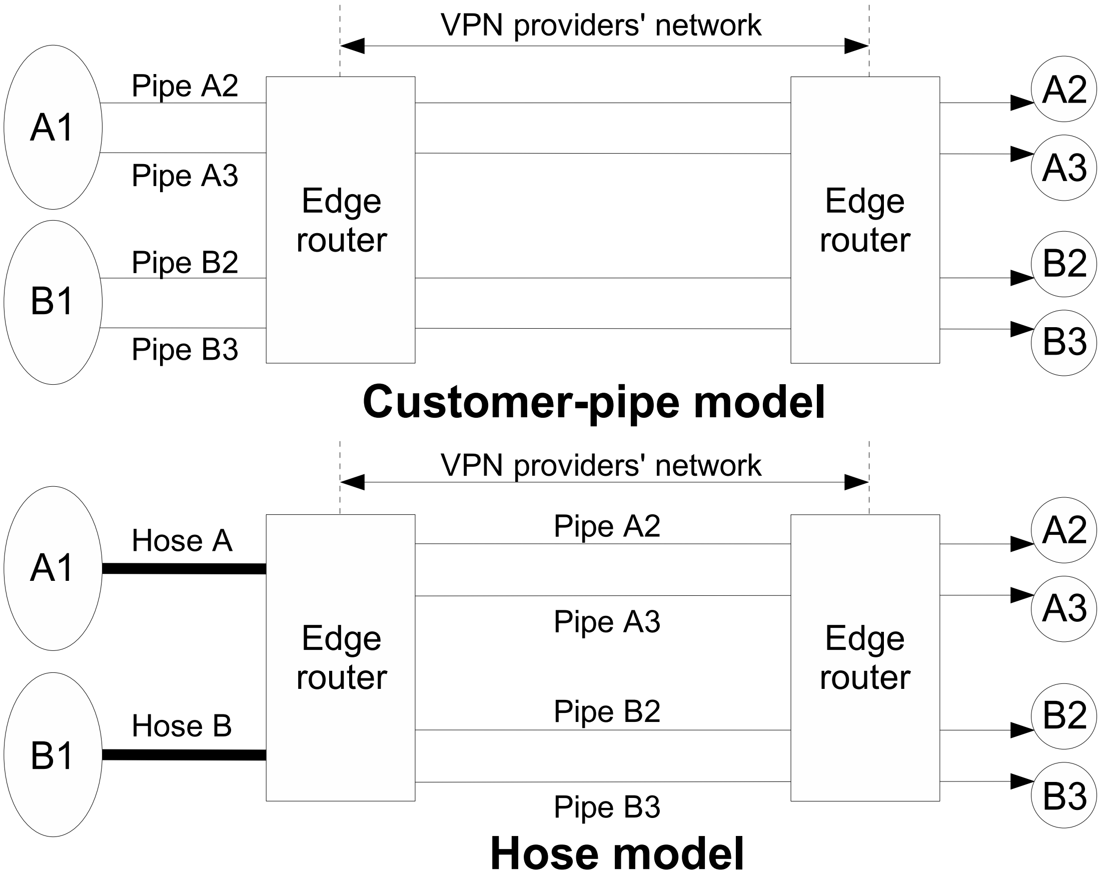


Figure 1. Illustration of customer-pipe and hose models

the best-effort VPN service.

To achieve an MTA service in the hose model, we need two mechanisms. The first is a provisioning algorithm that determines an adequate amount of allocated bandwidth that meets the minimum throughput requirements for each hose. A provisioning algorithm for the customer-pipe model has been proposed [26], and the basic idea of this algorithm can be applied to the hose model with slight modifications. In this dissertation, we assume that network provisioning for the hose model has already been completed. The second is an adaptive bandwidth allocation mechanism that allocates the bandwidth to hoses accommodating the available bandwidth of a bottleneck link in a network.

Table 1. Example of bandwidth allocation in customer-pipe and hose models

Customer-pipe model

Site	A2	A3	B2	B3
Bandwidth (Mb/s)	30	30	20	20

Hose model

Subscriber	A		B	
Site	A2	A3	B2	B3
Bandwidth (Mb/s)	60		40	

In this dissertation, to achieve an MTA service in the hose model, we propose a service model for the hose model and a hose bandwidth allocation method as an implementation of the service model. Our basic idea for the service model is that the available bandwidth in each bottleneck link is distributed to hoses based on a ratio determined by the provisioning algorithm. To implement the hose model with an MTA service, we propose a hose bandwidth allocation mechanism that allocates the bandwidth at the subscriber and customer levels. The difficulty of hose bandwidth allocation is that the ingress edge routers must obtain overall information about the available bandwidth inside the network and then allocate the bandwidth at two levels in a hierarchical manner. To obtain such information at ingress routers, the proposed method uses a feedback-driven traffic control mechanism. For the proposed method based on the service model, two requirements should be considered:

1. fairness and

2. high utilization.

The fairness requirement is divided into two levels, hose and pipe, to prevent bandwidth starvation in a pipe [27]. The proposed method meets the requirements by proportional fair bandwidth allocation in two levels based on collected feedback information and traffic monitoring at ingress routers.

The rest of Chapter 2 is organized as follows. Section 2.2 reviews related work and describes the shortcomings of an MTA service in the hose model. Section

2.3 describes the hose bandwidth allocation method and Section 2.4 presents computer simulations that demonstrate that the proposed method meets the three requirements. Section 2.5 concludes by briefly summarizing the main points and mentioning future work.

2.2 Related work on QoS traffic control mechanisms

As explained in the Introduction, no mechanism for achieving MTA service in the hose model has been proposed although the hose model is effective for network resource efficiency and configuration complexity. In this section, we review three related studies on the hose model in Section 2.2.1 and introduce three related studies and their shortcomings in terms of MTA services in the hose model in Section 2.2.2.

2.2.1 VPN hose model

First, we describe the research area of the hose model. The model was proposed by Duffield et al. [23] to improve the efficiency of network resource utilization. It was defined as a bandwidth contract model. The authors designed the hose model to reduce the required network resources by aggregating customer pipes because aggregated pipes, called a hose, can achieve a statistical multiplexing gain. They showed the effectiveness of the hose model by using trace-driven simulations.

Later, a provisioning algorithm for the hose model was proposed [25]. This algorithm constructs a tree-structure topology connected the VPN sites of subscribers and attempts to optimize the total network resources. This study proved the effectiveness of a tree structure in the hose model.

Other research focused on fault tolerance in the tree-structured hose model [28]. A proposed restoration algorithm can select backup paths when primary paths fail. It also minimizes the total network resources in the network topology by using backup paths.

Many provisioning algorithms for the hose model have been proposed. Although they can be effective in terms of network resource efficiency, they do not enable an MTA service in the hose model to be achieved because the studies did not provide bandwidth allocation mechanisms for the hose model.

2.2.2 Bandwidth allocation mechanisms

Next, we explain the research area of the bandwidth allocation mechanism and review three previous studies. The first two studies enhance the queuing method for fair bandwidth allocation in the hose model, while the third achieves high utilization and dynamic reallocation of spare bandwidth in the customer-pipe model.

The first queuing method is the two-dimensional deficit round robin (2-D DRR) [29], which is based on the deficit round robin (DRR) [30]. It was proposed to achieve fair allocation of bandwidth for pipes and flows. The authors assume that the set of all intermediate nodes and their links can be considered to be a *superswitch* model. All flows that pass through the superswitch are classified: flows toward the same destination are associated with the same group. 2-D DRR divides the bandwidth into group and flow levels.

Since the principle of the superswitch is conceptual, it is not clear how it can be implemented. Achieving 2-D DRR in an actual network requires a bandwidth allocation method in each router.

We then introduce a mechanism that integrates a reservation protocol and a queuing method. To enforce fair usage of reserved resources among sites of a subscriber, a resource management mechanism for the hose model with reservation protocol traffic engineering (RSVP-TE) [31] has been proposed [32, 33]. Two levels of weighted fair queuing (WFQ) are used in the incoming and outgoing queues, providing a WFQ service to different VPNs in the outer level and a logical WFQ service to different hoses of a VPN in the inner level. However, bandwidth allocation among different subscribers was not included in their evaluations.

Even if the above queuing methods achieve fair bandwidth allocation in the router, they cannot dynamically limit incoming traffic to a bottleneck link in the network. The significant shortcoming with respect to high utilization is that no routers have information about the degree of congestion inside networks. Without congestion-related information, dynamic reallocation of the spare network resources cannot be achieved.

Next, we describe an available bandwidth allocation method based on feedback-driven control. In contrast with the above two queuing methods, the inherent shortcoming of this method is that it cannot be applied to the hose model, al-

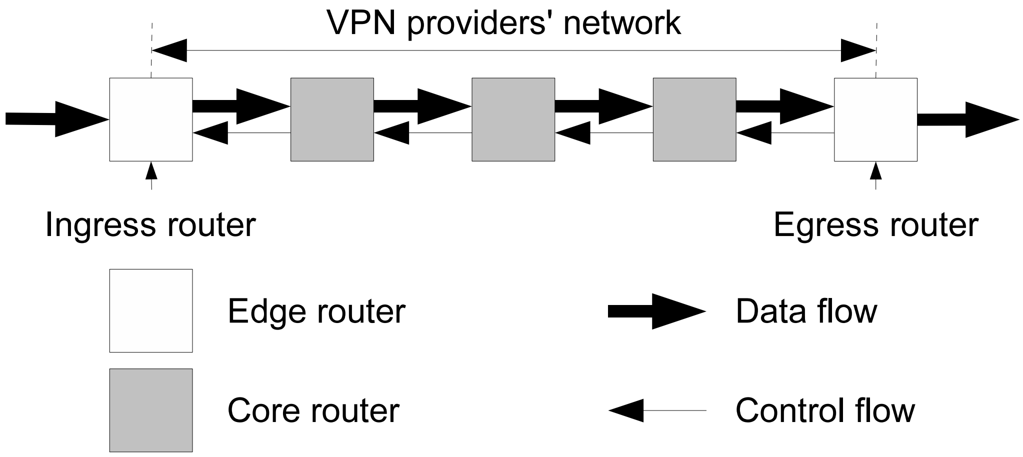


Figure 2. Weighted Proportional Fair Rate Allocation

though it achieves high utilization with the customer-pipe model.

To achieve high utilization and fair allocation, the weighted proportional fair rate allocation (WPFRA) method [21] has been proposed. Here, we explain a one-way version of the WPFRA method [22], which is an extension of the original. The original WPFRA method uses round-trip feedback control, whereas the one-way version has been introduced to improve the convergence speed to the target bandwidth in transition states.

A brief overview of the WPFRA method is given in Fig. 2. Two kinds of routers are used: edge and core routers. Outside traffic enters at ingress edge routers, goes through core routers, and finally exits at egress edge routers. In this mechanism, ingress edge routers obtain information about the available bandwidth from egress edge routers and then allocate the bandwidth fairly among sites based on a weight applied to each site. The main feature of the WPFRA method is the assignment of weights to provide weighted proportional fairness among all sites.

Here, we explain the calculation process for bandwidth allocation. Let B be the available bandwidth in a network, w_j be the value of a weight, and r_j be the optimum bandwidth allocation given by

$$r_j = w_j \times \frac{B}{\sum_{k=1}^{n} w_k},$$

where j indicates the path index from a source site toward a destination site.

The egress routers are responsible for periodically sending control packets toward ingress routers, whereas ingress routers shape traffic to reflect the information described in the control packets. Traffic shaping rates for path j are given by

$$r_j \Leftarrow \min\left(\mathrm{ER}_{(i,e)} \times w_j, r_j + \mathrm{IB} \times w_j\right),$$

where explicit rate ($\mathrm{ER}_{(i,e)}$) is an available rate for one weight on path (i,e) and incremental bound (IB) is a parameter for preventing sudden increases in sending rates. Note that path j between sites of subscribers contains path (i,e) between ingress and egress edge routers.

Core routers periodically measure the amount of arriving traffic, and they calculate the number of virtual flows based on

$$n_{vf} = \max\left(\frac{r_{arr}}{\hat{r}_f}, 1\right),$$

where r_{arr} indicates the amount of traffic arriving in a given measurement period and $\hat{r}_f$ is the exponential weighted moving average of the fair share rate of a virtual flow. Here, a virtual flow corresponds to the amount of traffic treated by one weight.

Note that n_{vf} represents the sum of each weight value corresponding to all traffic passing through a link of each core router, and the value of n_{vf} is always greater than 1.

Let U_t denote the target link utilization and C be the link capacity. We then calculate the fair share rate of a virtual flow (i.e., one weight) as

$$r_f = \frac{C \times U_t}{n_{vf}}.$$

To reduce the instantaneous changes in the value of r_f, we introduce its moving average:

$$\hat{r}_f \Leftarrow (1-\alpha) \times \hat{r}_f + \alpha \times r_f,$$

where α represents a parameter in the range of $(0:1]$ for determining the responsiveness.

Finally, when a core router receives a control packet, it updates the ER value in the packet according to

$$\text{ER}_{(i,e)} \Leftarrow \min\left(\text{ER}_{(i,e)}, r_f\right).$$

As a result of the above procedure, the smallest value of r_f along the path between the ingress and egress routers is selected as the value of ER, which is the explicit rate for a single virtual flow.

As described above, the WPFRA method achieves high link utilization by measuring the arriving traffic and applying dynamic bandwidth reallocation. We show an example to confirm the capability of the WPFRA method for the customer-pipe model. The network topology for this example is shown in Fig. 3. Site X0 transmits data toward sites X1 and X2. Similarly, site Y0 communicates with site Y1. The bandwidth of the bottleneck link is 100 Mb/s, and that of other links is infinite. After time 0, 1, and 2, traffic toward sites X1, Y1, and X2 is injected into the network. Routers I, C, E represent ingress edge, core, and egress edge routers, respectively.

The bandwidth allocation calculation process and the system variables of the WPFRA method are shown in Table 2. The weights for sites X1, X2, and Y1 are all 1. To confirm the convergence of system variables and the allocated bandwidth, we set $\hat{r}_f$ to a sufficiently high value. When new traffic appears at time 0.1, 1.1, and 2.1, the values of system variables are changed. Then, the system variables and the sending rates for all the sites gradually converge to certain values, and the available bandwidth of each site is determined after this convergence. At time 1, since other sites are not active, site X1 can utilize 100% of the available bandwidth (sending rate $r_{\text{X1}} = 100$ Mb/s). At time 2, the available bandwidth is divided into half for each site X1 and Y1 ($r_{\text{X1}} = r_{\text{Y1}} = 50$ Mb/s). At time 3, the available bandwidths of the sites are equally redistributed ($r_{\text{X1}} = r_{\text{X2}} = r_{\text{Y1}} = 33.3$ Mb/s) because traffic of other subscribers affects the bandwidth available to each site in the customer-pipe model. It is concluded that the WPFRA method achieves high utilization and fair bandwidth allocation in the customer-pipe model.

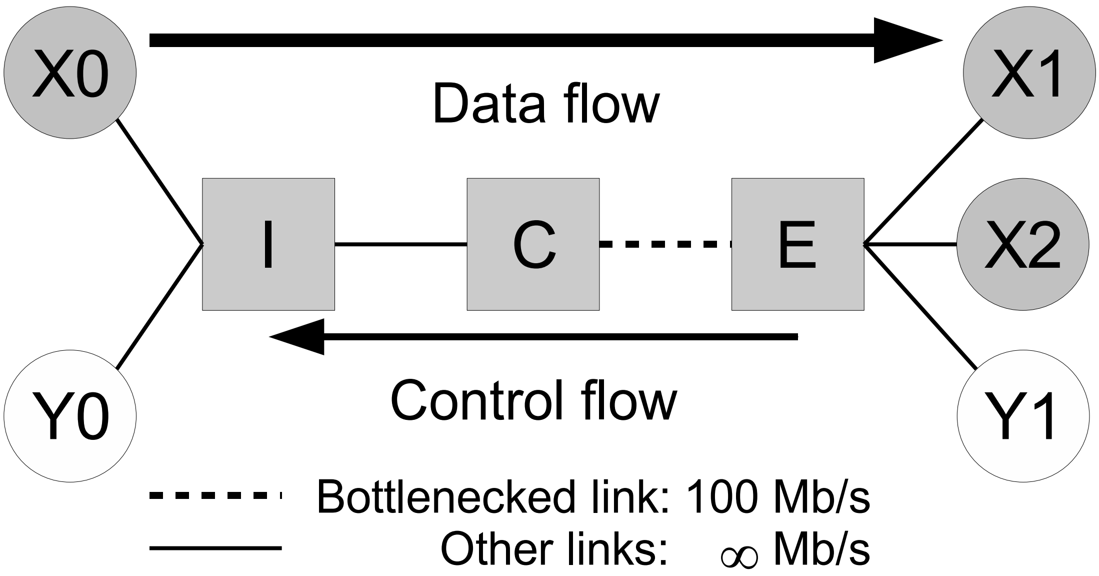


Figure 3. Example of bandwidth allocation and calculation process: Network topology

2.3 Hose bandwidth allocation method

An MTA service will provide better throughput predictability in PPVPNs, as described in Section 2.1. Since all the related works have substantial shortcomings, they cannot achieve an MTA service in the hose model. To overcome these inevitable shortcomings, we propose a novel bandwidth allocation method for the hose model. In this section, we first consider how to allocate the available bandwidth to hoses and then describe the requirements and details of a traffic control mechanism. Finally, we present the features of the proposed method through comparison with the previous related methods.

2.3.1 MTA service in hose model

In this section, we design a service model in the hose model. Our basic idea for the service model is that we proportionally distribute the available bandwidth to hoses in each bottleneck link. First, we determine the ratio of the allocated bandwidth

Table 2. Comparison of bandwidth allocation and calculation process for the WPFRA and the hose bandwidth allocation method

Time	WPFRA method								Hose bandwidth allocation method							
	Sending rate (Mb/s)			System variables					Sending rate (Mb/s)			System variables				
	r_{X1}	r_{X2}	r_{Y1}	r_{arr}	n_{vf}	r_f	$\hat{r}_f$	ER	r_{X1}	r_{X2}	r_{Y1}	r_{arr}	n_{vf}	r_f	$\hat{r}_f$	ER
0.0	0.0	0.0	0.0	0.0	1.0	100.0	999.0	100.0	0.0	0.0	0.0	0.0	1.0	100.0	999.0	100.0
0.1	100.0	0.0	0.0	**100.0**	**1.0**	**100.0**	**189.9**	**100.0**	100.0	0.0	0.0	**100.0**	**1.0**	**100.0**	**189.9**	**100.0**
0.5	100.0	0.0	0.0	100.0	1.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	1.0	100.0	100.0	100.0
0.9	100.0	0.0	0.0	100.0	1.0	100.0	100.0	100.0	100.0	0.0	0.0	100.0	1.0	100.0	100.0	100.0
1.0	**100.0**	**0.0**	**0.0**	100.0	1.0	100.0	100.0	100.0	**100.0**	**0.0**	**0.0**	100.0	1.0	100.0	100.0	100.0
1.1	100.0	0.0	100.0	**200.0**	**2.0**	**50.0**	**55.0**	**50.0**	100.0	0.0	100.0	**200.0**	**2.0**	**50.0**	**55.0**	**50.0**
1.5	50.5	0.0	50.5	101.0	2.0	50.0	50.1	50.0	50.5	0.0	50.5	101.0	2.0	50.0	50.1	50.0
1.9	50.0	0.0	50.0	100.0	2.0	50.0	50.0	50.0	50.0	0.0	50.0	100.0	2.0	50.0	50.0	50.0
2.0	**50.0**	**0.0**	**50.0**	100.0	2.0	50.0	50.0	50.0	**50.0**	**0.0**	**50.0**	100.0	2.0	50.0	50.0	50.0
2.1	50.0	50.0	50.0	**150.0**	**3.0**	**33.3**	**35.0**	**33.3**	25.0	25.0	50.0	**100.0**	**2.0**	**50.0**	**50.0**	**50.0**
2.5	33.5	33.5	33.5	100.5	3.0	33.3	33.4	33.3	25.0	25.0	50.0	100.0	2.0	50.0	50.0	50.0
2.9	33.3	33.3	33.3	100.0	3.0	33.3	33.3	33.3	25.0	25.0	50.0	100.0	2.0	50.0	50.0	50.0
3.0	**33.3**	**33.3**	**33.3**	100.0	3.0	33.3	33.3	33.3	**25.0**	**25.0**	**50.0**	100.0	2.0	50.0	50.0	50.0

Weights for sites in the WPFRA method: X1 : X2 : Y1 = 1 : 1 : 1

Weights for subscribers in the proposed method: X : Y = 1 : 1

Capacity of single bottleneck link = 100(Mb/s), parameter $\alpha = 0.9$

among hoses by using a provisioning algorithm, and we assume that the allocated bandwidth always exceeds the minimum agreed throughput in the MTA service even during a period of congestion. Then, we assign the bandwidth that exceeds the minimum agreed throughput to the hoses. Finally, we distribute the excess bandwidth to active hoses based on the ratio determined by the provisioning algorithm.

An example of the bandwidth allocation in the service model is shown in Fig. 4. Two customers A and B contract for hoses A1 and B1, which contain three and two pipes, respectively. All pipes except for the pipe from B1 to B3 (pipe B1-B3) are active. $R_{i(s_1)}$ represents the available bandwidth of the hose of source site s_1 in the bottleneck link i; $R_{i(s_1 \to s_2)}$ represents the available bandwidth divided equally into active pipes from source site s_1 to destination site s_2. Since pipe B1-B3 is idle, pipe B1-B2 can utilize 100% of $R_{i(\text{B1})}$. Therefore, active hoses always provide link utilization of nearly 100%.

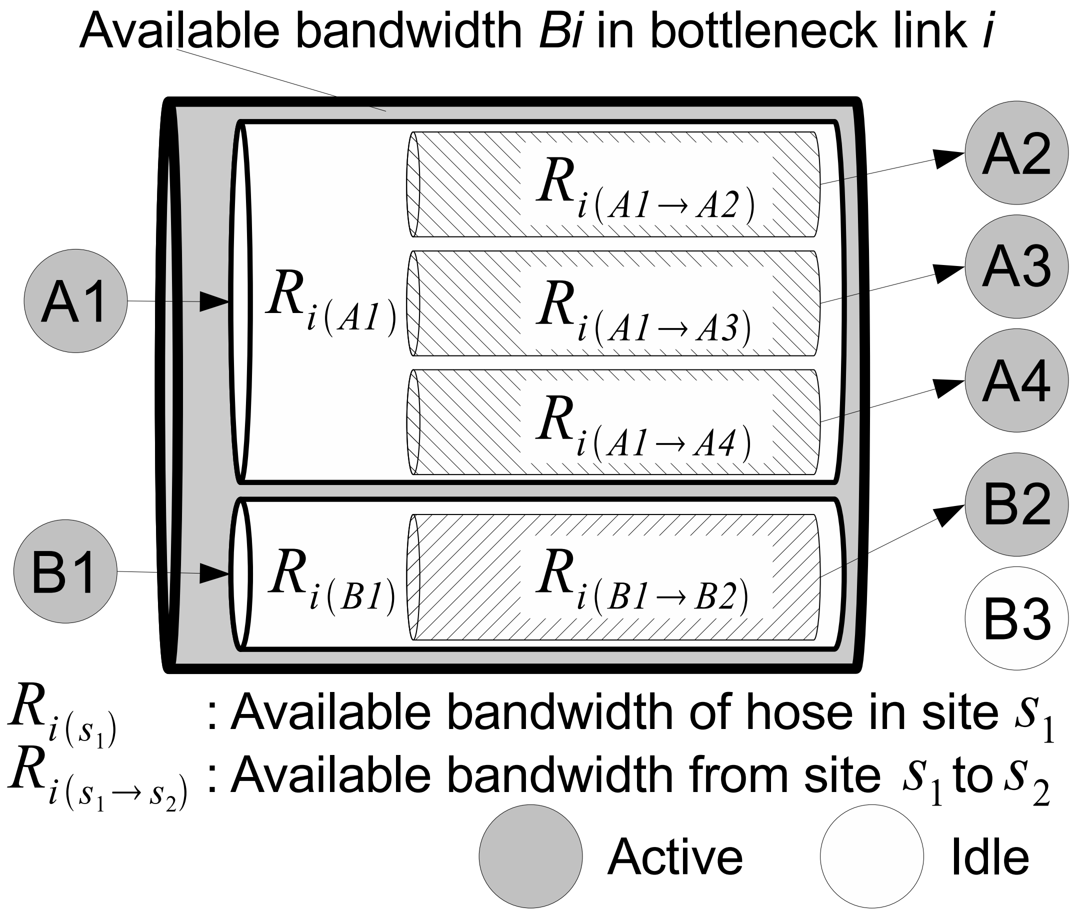


Figure 4. MTA service in the hose model

In this example, we divide the available bottleneck bandwidth B_i proportionally between hoses A1 and B1 in the ratio α to $(1-\alpha)$ during periods of congestion. $R_{i(s_1)}$ is given by

$$R_{i(\mathrm{A1})} = \alpha B_i, R_{i(\mathrm{B1})} = (1-\alpha)B_i, (0 \leq \alpha \leq 1).$$

The ratio α is determined by a provisioning algorithm, and $R_{i(s_1)}$ is calculated by a bandwidth allocation method. Then, $R_{i(s_1\rightarrow s_2)}$ is calculated by

$$R_{i(\mathrm{A1}\rightarrow \mathrm{A}x)} = \frac{R_{i(\mathrm{A1})}}{N_{i(\mathrm{A1})}}, \qquad R_{i(\mathrm{B1}\rightarrow \mathrm{B}x)} = \frac{R_{i(\mathrm{B1})}}{N_{i(\mathrm{B1})}},$$

where $N_{i(s_1)}$ represents the number of active pipes in a hose of source site s_1.

Our assumption is that MTA service can be achieved when the bandwidth allocation method precisely divides the bandwidth according to the ratio α, which is determined by the provisioning algorithm. Based on the service model in the hose model, we consider the mechanism of the hose bandwidth allocation method in the next section.

2.3.2 Mechanism

To meet the requirements for both bandwidth allocation and high utilization, we integrate a feedback-driven traffic control and QoS scheduler at an ingress router. Ingress routers can be aware of the congestion status of the network to obtain available information from egress routers. Based on the congestion-related information, the maximum amount of available bandwidth can be allocated to active subscribers and sites.

In the proposed method, we use a one-way version of the WPFRA method [22] and class-based queueing (CBQ) [34] as a feedback-driven traffic control and traffic scheduler. The WPFRA method can completely exploit the available bandwidth, and CBQ can hierarchically shape the input traffic rate at the subscriber and site levels. Moreover, we implement two mechanisms at ingress routers to simultaneously satisfy the above two requirements for fairness. The first is traffic measurement for each destination site to determine whether traffic to the destined site is active. Ingress routers periodically monitor the amount of incoming traffic and classify all sites. We utilize a measurement algorithm that is an exponentially weighted moving average to adjust the convergence speed. The other is dynamic control of CBQ based on active information. More specifically, the feedback control packets from the network are used to determine the amount of bandwidth to allocate, which is set in the CBQ parameters.

To set these parameters, the ingress router first looks up the identifier of a bottleneck link in a received control packet. The bottleneck link is specified by core routers when the control packet passes through them. Then, the ingress router calculates the bandwidth assignment to each subscriber corresponding to the bottleneck link, and the assigned bandwidth is divided evenly among the subscriber's active sites.

The pseudocode of these algorithms is shown in Fig. 5. First, the `RecvCtrlPkt` function is executed when ingress routers receive control packets. Then, the ingress routers obtain the value of ER and the location of a bottleneck link by using the `LookupER` and `LookupBottleneckLink` functions, respectively. Finally, the ingress routers calculate and allocate the bandwidth to subscribers and active sites based on the following equations (1) and (2).

Let w_h be the weight of subscriber h and $r_{l(h)}$ be the bandwidth allocated to bottleneck link l given by

$$r_{l(h)} \Leftarrow \min\Big(\mathrm{ER}_{(i,e)} \times w_h, r_{l(h)} + \mathrm{IB} \times w_h\Big), \tag{1}$$

where path (i, e) contains bottleneck link l. Note that, hereafter, we omit the adjustment parameter of IB; in other words, we set IB to a sufficiently high value.

Then, let $x_{l(hp)}$ be a binary flag that indicates whether site p of subscriber h is active in bottleneck link l and let $r_{l(hp)}$ be the bandwidth allocated to active sites given by

$$r_{l(hp)} = \begin{cases} \dfrac{r_{l(hp)}}{\sum\limits_{j=1}^{n} x_{l(hj)}} & (x_{l(hp)} = 1) \\ \\ 0 & (x_{l(hp)} = 0), \end{cases} \tag{2}$$

where n is an index identifying a subscriber's sites. Note that since these equations are conceptual, we must allocate a very small but nonzero amount of bandwidth to idle sites to prevent them discarding all of the incoming packets.

The difference between the WPFRA method and the proposed method is the impact of new traffic on the calculation process of system variables and bandwidth allocation. In Table 2, new traffic of the same subscriber (i.e., X2 at times 2.0 and 2.1) does not affect the values of system variables in the proposed method, whereas the start of any traffic does affect the system variables in the WPFRA method. Therefore, the proposed method can divide the available bandwidth among subscribers, while WFPRA divides it among sites.

Next, we explain the inherent difficulty of bandwidth allocation where the multiple ingress routers exist. If all subscribers are connected to the same ingress router, it can easily distribute the available bandwidth to them. In other words,

```
/*********************************************
 * variable_h:  Variable for a hose
 * variable_hp: Variable for a pipe in a hose
 * num_h:  Number of subscribers
 * num_hp: Number of sites of a subscriber
 * h: Index of hose
 * p: Index of pipe
 * s: Total number of active sites
 * l: Bottleneck link
 * w: Weight
 * x: Active flag (binary)
 * r: Available rate
*********************************************/
#define ZERO 0.01 // a very small value

RecvCtrlPkt ( CtrlPkt *p ){
  /*********************************************
   * Look up network information in control packet
  *********************************************/
  ER = LookupER(p);    l = LookupBottleneckLink(p);

  /*********************************************
   * Calculation for hoses
  *********************************************/
  // Calculate Eq. (1)
  for ( h=0; h<num_h[h]; h++ )
    r_h[l][h] = ER * w_h[h];

  /*********************************************
   * Calculation for pipes in hoses
  *********************************************/
  // Calculate the denominator of Eq. (2)
  for ( h=0; h<num_h[h]; h++ )
    for ( p=0; h<num_hp[h][p]; p++ )
      s_hp[l][h][p] += x_hp[l][h][p];

  // Calculate Eq. (2)
  for ( h=0; h<num_h[h]; h++ ){
    for ( p=0; h<num_hp[h][p]; p++ ){
      if ( x_hp[l][x][p] == 1 )
        r_hp[l][h][p] = r_h[l][h] / s_hp[l][h][p];

      else if ( x_hp[l][x][p] == 0 )
        r_hp[l][h][p] = ZERO;
      }
  }
}
```

Figure 5. Pseudocode of calculation of bandwidth allocation at ingress routers

the ingress router does not need to be aware of information about other subscribers. However, if other subscribers connect to other ingress routers, then all the ingress routers must shape the arriving traffic based on the information about all subscribers.

An example of bandwidth allocation with multiple ingress routers in the proposed method is shown in Fig. 6. There are two subscribers A and B. Sites A1 and B1 communicate with A2 and A3 and with B2 and B3, respectively. Ingress router I1 has CBQ class A and CBQ classes A2 and A3 as children of the CBQ class A. Similarly, another ingress router I2 has CBQ classes B, B2, and B3. Here, ER is the bandwidth available for a virtual flow. To explain the basic behavior of weighted proportional fair rate allocation among subscribers, we assume that the weights for subscribers A and B are assigned as 3.0 and 2.0, respectively. There are three active sites: A2, A3, and B2. In this example, subscribers A and B are allocated 3.0ER and 2.0ER based on their weights, respectively. At the site level, sites A2 and A3 can use 1.5ER because both sites are active. 3.0ER is evenly divided between two sites. On the other hand, site B2 can use all of the bandwidth allocated to subscriber B because site B3 is idle.

2.3.3 Qualitative comparisons

The benefit of the proposed method is that it can meet three substantial requirements: proportional fair bandwidth allocation among subscribers, fair bandwidth allocation among active sites in each subscriber, and high network utilization. Table 3 compares the proposed method and those in the related studies described in Section 2.2. The proposed method is the only one that satisfies all of the above three requirements.

As described in Reference [35], bandwidth allocation among flows has significant complexity because such an algorithm requires packet classification and per-flow state management. In the context of VPNs and the hose model, fair allocation among subscribers and sites is required, even though per-flow fairness is unnecessary. Since per-flow processing is complex, it should be performed at the gateway of each site. The main reason we use the WPFRA method is because this mechanism does not process at the per-flow level, so it inherently achieves high utilization.

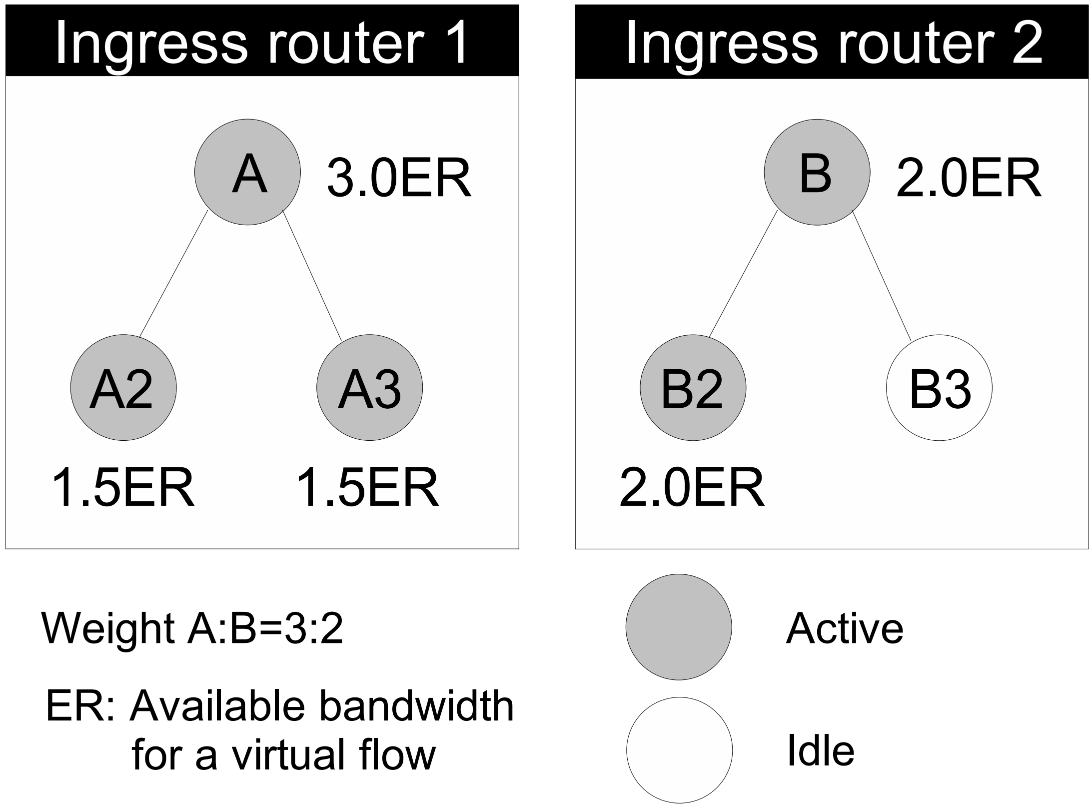


Figure 6. Bandwidth allocation at ingress routers

2.4 Evaluation

In the previous section, we described how to allocate the available bandwidth into hoses for an MTA service in the hose model and presented the hose bandwidth allocation method. In this section, we describe computer simulations performed to quantitatively evaluate the hose bandwidth allocation method. We first confirm the basic behavior of the proposed algorithm through comparisons with the WPFRA method. As a performance evaluation index, we use the "allocation error rate" as the ratio of the bandwidth in use to the previously allocated bandwidth. Since the minimum throughput may not be assured if there is a high allocation error rate, the allocation error rate must be reduced, so we investigate the impact of traffic parameters on the allocation error rate. Finally, we analyze the stability for the proposed method. On the basis of these computer simulation

Table 3. Comparison of features between the proposed method and ones in related work

Method	Model	Information gathering	Queuing method	Fairness			High utilization
				subscribers	sites	flows	
2-D DRR [29]	Hose	N	Modified DRR	N	Y	Y	N
Two-level WFQ [32, 33]	Hose	N	Modified WFQ	Y ‡	Y	N	N
WPFRA [21, 22]	Pipe	Y	Optional	N	Y	N	Y (bottleneck link)
Proposed method	Hose	Y	CBQ (optional)	Y	Y	N	Y (bottleneck link)

‡Though it has not been proved in their computer simulations.

runs, we clarify the characteristics of the proposed method and indicate how to decide the system parameters for the proposed method.

2.4.1 Simulation model

The computer simulator we used is the ns-2.29 simulator [36]. Since the original ns-2 does not contain the hose bandwidth allocation method, we add a module for it by extending the WPFRA module. The network topology used in the simulation is shown in Fig. 7. To check whether the proposed method has the designated hose behavior described in Table 3 in Section 2.3.3, we construct the simplest topology on the hose model where there are two subscribers each of which is connected to two destination sites. I1 and I2 represent ingress routers, C1 and C2 represent core routers, and E1 and E2 represent egress routers. These six routers form a VPN. A1, A2, and A3 are routers belonging to subscriber A. Similarly, B1, B2, and B3 belong to subscriber B. a1, ..., a6 and b1, ..., b6 are TCP senders of subscribers A and B, respectively. Traffic from a1, a2, and a3 and from a4, a5, and a6 goes to A2 and A3, respectively. Similarly, traffic from b1, b2, and b3 and from b4, b5, and b6 goes to B2 and B3, respectively. All links are 100 Mb/s, and the link propagation delays range from 1 to 10 ms, as shown in Fig. 7. As described in Section 2.3, we treat a topology with a single bottleneck link and multiple subscribers connected to different ingress routers.

To prevent TCP oscillation, we vary the propagation delay of access links between that of the TCP sender hosts and A1 or B1: 1, 2, and 3 ms. The TCP algorithm is TCP SACK, and the number of TCP flows is 100 for each TCP

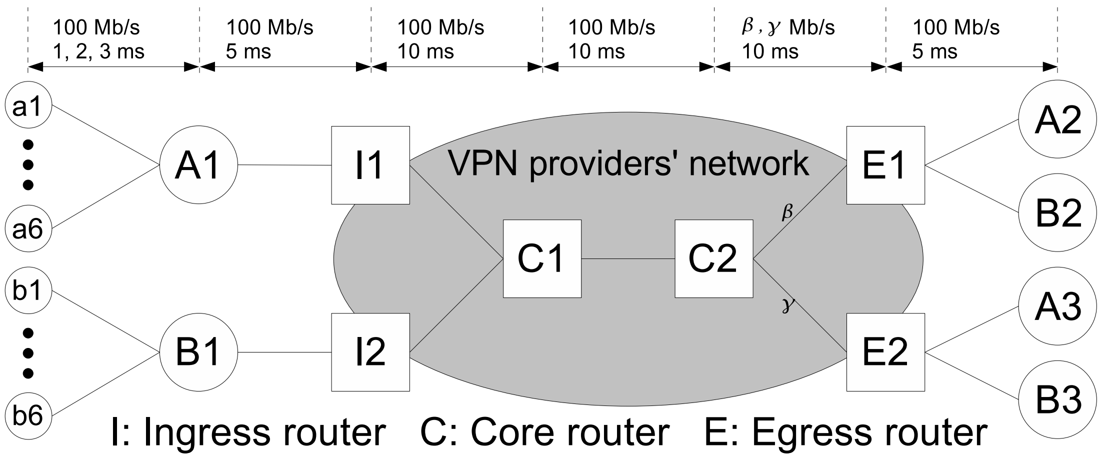


Figure 7. Simulation topology

sender, i.e., 300 flows for each destination site. TCP data and control packet size are 1500 and 100 bytes, respectively, including the headers. The buffer sizes in the routers are given in Table 4. Since ingress routers shape input traffic in the proposed method, the buffer size in a router inside the network should be small. Therefore, we set a lower value for the buffer size in core routers. Similarly, we set a higher value for the buffer size in routers outside the network to shape the enormous amount of input traffic.

The values of control parameters used in the proposed method are given in Table 5. To evaluate the basic behavior of the proposed method, we mitigate the effect of adjustment parameters; i.e., we set α and IB to sufficiently high values. The interval times of ingress, core, and egress routers is 500, 100, and 100 ms, respectively. In the proposed method, the responsibility of ingress routers is significant because we implement two additional mechanisms at ingress routers as described in Section 2.3.2. We explain why the interval time of ingress routers is longer than those of other routers in Section 2.4.5.

2.4.2 Comparison with WPFRA method

In this section, we compare the behavior of the WPFRA method and the proposed method. For this simulation scenario, we demonstrate that the proposed

Table 4. Output buffer length of each router

<table>
<tr><th colspan="2">Router</th><th>Buffer length</th></tr>
<tr><td rowspan="4">Right direction</td><td>A1, B1</td><td>1200</td></tr>
<tr><td>I1, I2</td><td>600</td></tr>
<tr><td>C1, C2</td><td>400</td></tr>
<tr><td>Others</td><td>∞</td></tr>
<tr><td>Left direction</td><td>Others</td><td>∞</td></tr>
</table>

Table 5. Parameter setting

Parameter	Value
α	0.9
IB	∞
Interval time of ingress routers	500 ms
Interval time of core routers	100 ms
Interval time of egress routers	100 ms

algorithm can allocate bandwidth for each active subscriber when traffic is inserted and removed. To simplify the network topology, we set β and γ to 100 Mb/s in Fig. 7. Traffic toward A2, B2, A3, and B3 starts at 1, 10, 30, and 50 s, respectively. The simulation ends at 100 s. We set weights of 3 for subscriber A and 2 for subscriber B in the proposed method. Since the WPFRA method is unable to allocate bandwidth for a subscriber, we set 3 for sites A2 and A3 and 2 for sites B2 and B3 in the WPFRA method.

The throughput dynamics of the two methods are shown in Fig. 8. The x-axes represent time in seconds, and the y-axes indicate the received throughput at A2, B2, A3, and B3 with a measurement interval of 1 s.

First, we focus on Fig. 8(b) which concerns the WPFRA method. Since bandwidth division is performed for each customer pipe, not for each subscriber aggregation, new starting traffic at 30 and 50 s and ending traffic at 70 and 90 s affect the throughput of all other traffic. When a site becomes active/idle, the throughput of all other sites decreases/increases. On the other hand, the proposed method illustrated in Fig. 8(a) shows a different behavior. The starting traffic toward A3 at 30 s and B3 at 50 s does not affect the throughput of other subscribers B and A, respectively. Similarly, the throughput of one subscriber is not affected by ending traffic of the other subscriber at 70 and 90 s. In the bandwidth allocation among sites belonging to the same subscriber, the bandwidth available to sites A2 and A3, and to B2 and B3 is divided equally at 30 and 50 s. This confirms that the proposed method can use the hose model because the proposed method divides the bandwidth at both the subscriber and site levels.

Regarding the value of ER which is one of the important parameters in the proposed and WPFRA methods, similar with the throughput dynamics, the ER dynamics between the proposed and WPFRA methods are different. In the WPFRA method in Fig. 9(b), the value of ER changes and converges to the other value when new traffic starts. In contrast with this result, in the proposed method in Fig. 9(a), the ER value converges to approximately 20 Mb/s. The convergence speed of ER is affected by the responsiveness parameter α. In Refs. [21, 22], the impact of α has been analyzed. In this dissertation, the remarkable evaluation points are the scalability and stability of the proposed method, so we set α to a high value 0.9.

Next, we investigate the bandwidth allocation error rate and utilization in certain scenarios. In all scenarios, traffic toward A2, B2, B3, and A3 starts simultaneously at 1 s, and the simulation ends at 100 s. We investigate three simulation scenarios: traffic toward A2 and A3, traffic toward A2, A3, and B2, and traffic toward A2, B2, and B3. The average throughput at the receiver side and the utilization in each scenario are given in Table 6. All the results in the table are averages of a 50-second measurement between 50 and 100 s. Regarding the utilization, the proposed method and the WPFRA method both achieved 100%. As for the allocation error rate, values inside the parentheses in the table indicate errors from the target throughput. These results show that the proposed method achieves fairly low error rates (i.e., below 1.5%) for each site. Regarding the subscriber level, in every scenario, subscribers A and B attain approximately 60 and 40 Mb/s, respectively. All allocation error rates for subscribers A and B are also below 1.5%.

Table 6. Bandwidth allocation error rate and utilization

Unit: Mb/s

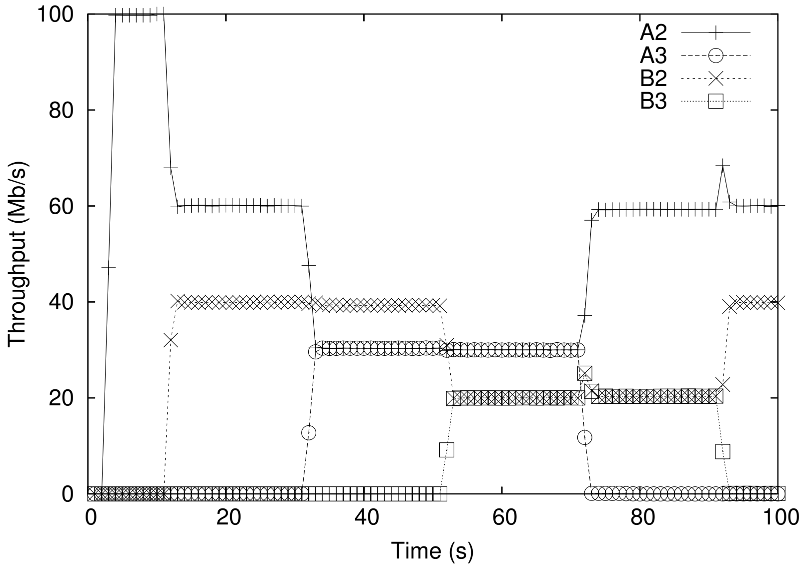
	Proposed method	WPFRA method
A2	60.00 (+0.00%)	59.96 (−0.07%)
B2	40.00 (+0.00%)	40.04 (+0.10%)
Utilization	100.00%	100.00%
A2	30.28 (+0.93%)	37.68 (+0.48%)
A3	30.28 (+0.93%)	37.68 (+0.48%)
B2	39.44 (−1.40%)	24.63 (−1.48%)
Utilization	100.00%	100.00%
A2	59.50 (−0.83%)	42.37 (−1.14%)
B2	20.25 (+1.25%)	28.82 (+0.87%)
B3	20.25 (+1.25%)	28.82 (+0.87%)
Utilization	100.00%	100.00%

The results demonstrate that the proposed method can allocate bandwidth for subscribers but not for sites and that it can provide high link utilization. Thus, from these computer simulations, we conclude that the hose bandwidth allocation method described in Table 3 in Section 2.3.3 can be achieved.

2.4.3 Impact of multiple bottleneck links in a hose

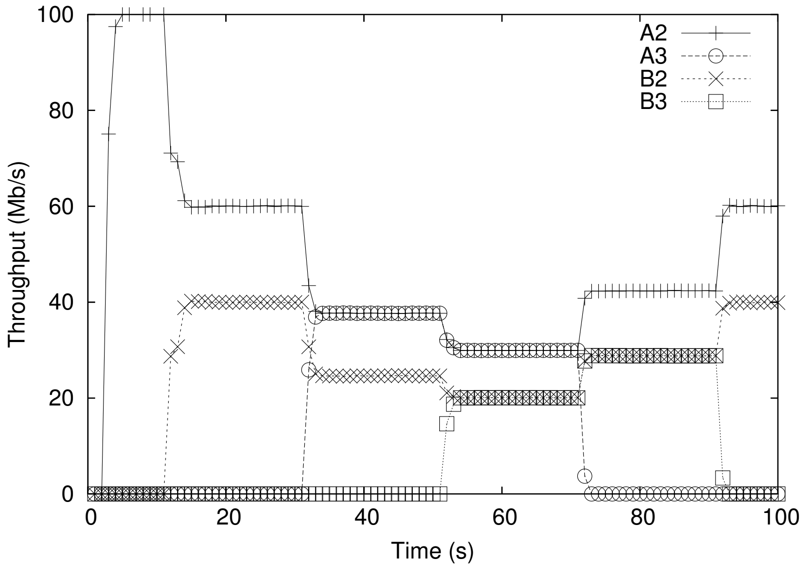
In the previous scenario, the bottleneck link was a single link between core routers C1 and C2, although traffic toward sites A2, A3, B2 and B3 passed through different paths I1 to E1, I1 to E2, I2 to E1, and I2 to E2, respectively. Next, we evaluate the proposed method in a network topology with multiple bottleneck links in a hose. In the hose model, each hose accommodates multiple pipes, so a hose can have multiple bottleneck links in different pipes. In contrast, in the customer-pipe model, a customer pipe cannot have multiple bottleneck links because it represents a one-to-one connection, not one-to-many connections like

Weight for subscribers: A : B = 3 : 2



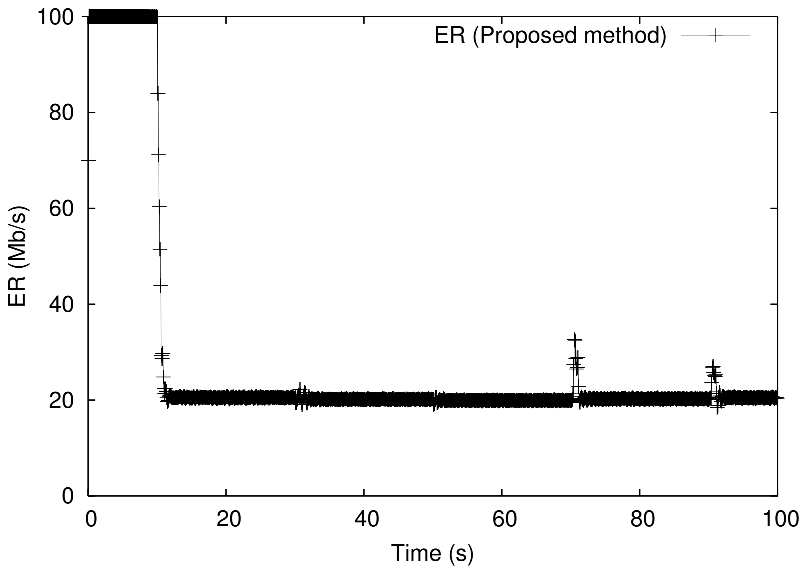
(a) Throughput dynamics in the proposed method

Weight for sites: A2 : A3 : B2 : B3 = 3 : 3 : 2 : 2

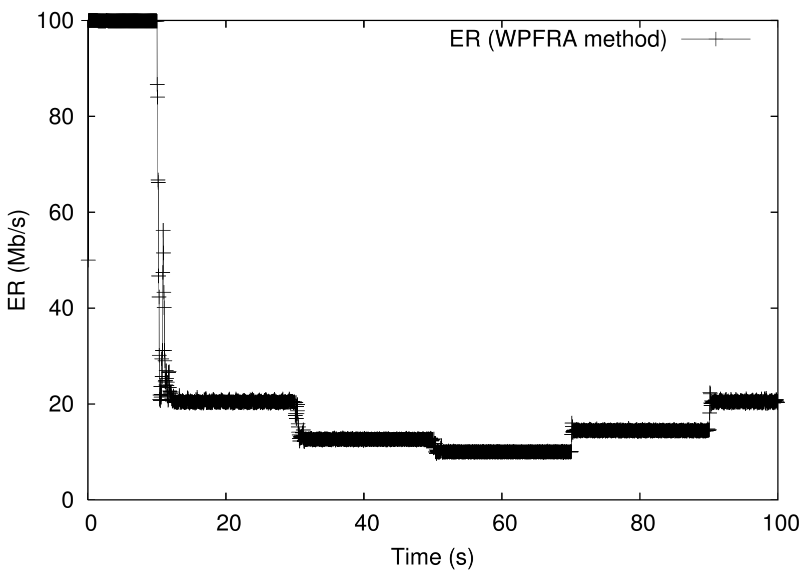


(b) Throughput dynamics in the WPFRA method

Figure 8. Comparison of throughput dynamics between the WPFRA method and the proposed method



(a) ER dynamics in the proposed method



(b) ER dynamics in the WPFRA method

Figure 9. Comparison of ER dynamics between the WPFRA method and the proposed method

Table 7. Results for multiple bottleneck links

Unit: Mb/s

	Subscriber A1	Subscriber B1
C2 → E1	17.98 (−0.11%)	12.02 (+0.17%)
C2 → E2	12.00 (+0.00%)	8.00 (+0.00%)
Total	29.98 (−0.07%)	20.02 (+0.10%)

Active: A2, A3, B2, B3

C2 → E1	30.00 (+0.00%)	0.00 (+0.00%)
C2 → E2	12.14 (+1.17%)	7.86 (−1.75%)
Total	42.14 (+0.33%)	7.86 (−1.75%)

Active: A2, A3, B3 Idle: B2

the hose model.

To construct a network topology with multiple bottleneck links in a hose, we set link capacities β and γ to 30 and 20 Mb/s in Fig. 7, respectively. The first bottleneck link is the link between C2 and E1; the other bottleneck link is the link between C2 and E2. This simulation scenario is the same as that in Section 2.4.2, except for the link capacity.

The simulation results are summarized in Table 7. The proposed method can divide the available bandwidth into the ratio of 3 to 2 in the congested link and achieve high link utilization. Even when site B2 is idle, the proposed method can provide at least approximately 8 Mb/s to subscriber B1. In both simulation scenarios, the maximum allocation error rate is 1.75%. We conclude that if no allocation error occurs, the minimum agreed throughput of at least 12 and 8 Mb/s can be assured to subscribers A1 and B1, respectively, in this scenario.

2.4.4 Scalability

The above results showed what the hose bandwidth allocation method is capable of doing. We now evaluate the proposed method in terms of the scalability and stability which are important performance metrics. First, we focus on the scalability where parameters of two subscribers are unbalanced because of the following reasons. In the proposed method, the ratio of two traffic parameters – the number of TCP flows and number of destination sites – among different subscribers may impact performance, i.e., if some subscribers have more TCP flows than others, then they may obtain a larger bandwidth. Similarly, the imbalance in the number of destination sites may also impact performance. In this simulation scenario, we simply set β and γ to 100 Mb/s in Fig. 7.

The impact of the imbalance ratio of the number of TCP flows is shown in Fig. 10. In this simulation, the number of TCP flows from subscriber A is fixed at 60, whereas the number of TCP flows from subscriber B varies in the range between 60 to 6000. This means that the imbalance ratio of the number flows of subscriber B to that of subscriber A is ranged from 1 to 100. The number of destination sites is 2 both for subscribers A and B, and the weight value is 1 for both subscribers A and B. The x-axis represents the imbalance ratio of the number of TCP flows, and the y-axis is the allocation error rate of the total throughput for each subscriber.

If the amount of incoming traffic is not regulated, the throughput ratio is normally proportional to the number of TCP flows. Nevertheless, the error in total throughput for each subscriber is nearly zero even when the number of flows increases. The maximum, average, and minimum error rates are 0.30, 0.23, and 0.00%, respectively. This result proves that the proposed method achieves the robustness of the number of flows.

In the next scenario, we examine the impact of increasing the number of destination sites, i.e., the scale of the hose (Fig. 11). The total number of destination sites connected to egress routers for subscriber A is fixed at 2, whereas the number of destination sites of subscriber B varies between 2 and 200. In other words, the range of the imbalance ratio for destination sites of subscriber B to that of subscriber A is 1 to 100. The number of TCP flows is 600 for both for subscribers A and B, and the weights are 1 for both subscribers. The x-axis in Fig. 11 represents the imbalance ratio of the number of destination sites, and the y-axis indicates the allocation error rate in the total throughput for each subscriber.

The error does not continue to increase even when the imbalance ratio of the destination sites increases. The maximum, average, and minimum error are

1.09, 0.75, and 0.01%, respectively. Most results are below 1.00%. As with the robustness against the number of flows, this result shows that the proposed method is robust with respect to the number of subscribers.

The results of the simulations clearly show that ingress routers can control the amount of incoming traffic even when the scale of a VPN between subscribers A and B is imbalanced in the ratio of 100. Thus, we conclude that the hose bandwidth allocation method can keep the low allocation error rate in a large-scale VPN.

2.4.5 Stability analysis

The above simulations showed the effectiveness of the hose bandwidth allocation method. However, if the allocation error rate is significant, the MTA service may not be achieved. Therefore, we must analyze the factors affecting the allocation error rate in the final simulation scenarios.

As described in Section 2.3.2, we implemented a traffic measurement mechanism in ingress routers. The ingress router is responsible for determining active sites by measuring the amount of incoming traffic for the destination sites, and allocating the available bandwidth to active sites. The most important aspect of the proposed method is that inaccurate measurement in ingress routers causes allocation error. For example, if $n-k$ sites are erroneously measured in ingress routers when n sites are active, then the allocation bandwidth for each site is shifted by a factor of $n/(n-k)$. However, n sites are actually active, hence the throughput for each site fluctuates. Moreover, the number of destination sites represents the number of queues in ingress routers. The total number of flows divided by the number of destination sites is the number of flows that passes through each queue. Therefore, the possibility of a queue instantaneously becoming empty increases as the number of destination sites increases. This characteristic may cause the error in measuring active sites.

An example of an inaccurate measurement of the number of active sites at an ingress router is shown in Fig. 12. The measurement interval time for ingress routers is 50 ms. The number of TCP flows of subscribers A and B is 600 for both, and the numbers of destination sites of subscribers A and B are 2 and 200, respectively. In other words, 300 and 3 flows pass through each queue,

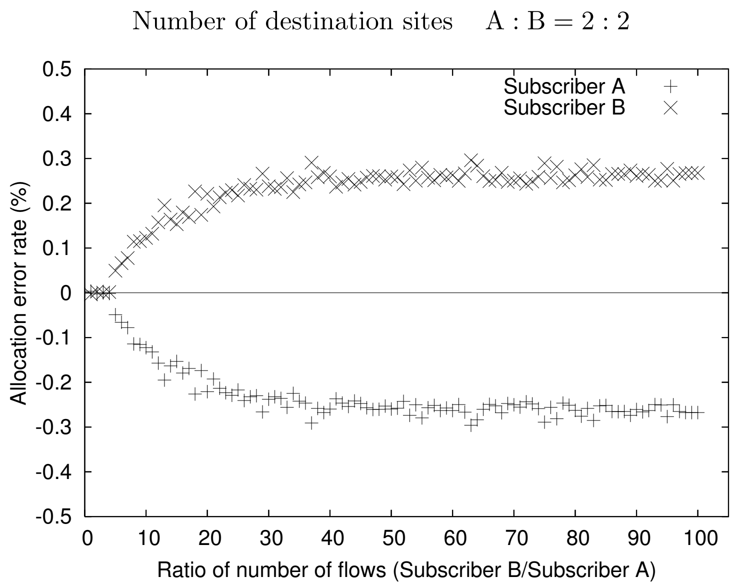


Figure 10. Impact of imbalance ratio of TCP flows on error rates

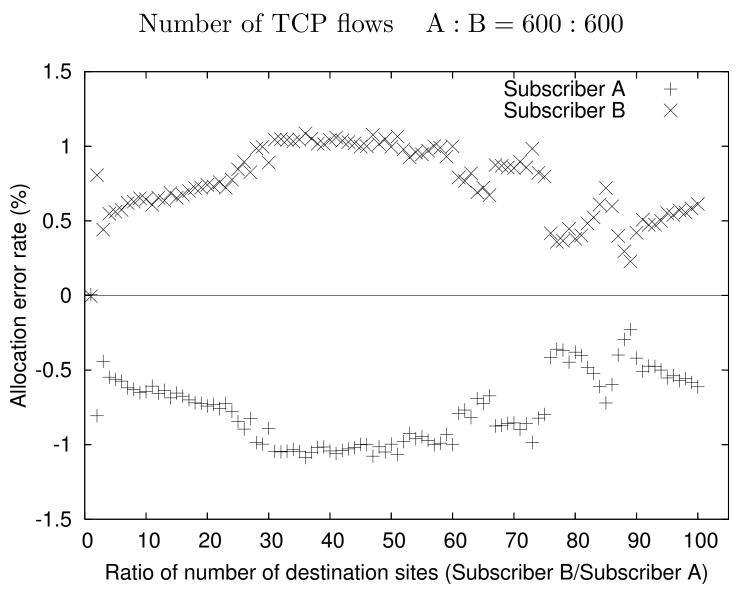


Figure 11. Impact of imbalance ratio of destination sites on error rates

respectively. Here, we focus on subscriber B. The correct measured value for subscriber B is 200 because all sites are in fact active, but the measurement results are approximately 40 and unstable. Consequently, each active site attempts to utilize a factor of 5 in the available bandwidth, which causes the fluctuations in throughput at each site.

To clarify the factors affecting the measurement error, we investigate the measurement interval time of ingress routers. The x-axis in Fig. 13 shows the measurement interval time of ingress routers in milliseconds, and the y-axis indicates the average number of active sites measured during a 50-s period. The measurement value converges on 200 as the measurement interval time increases. The reason for this is that fluctuating traffic passes through ingress routers at each measurement interval and ingress routers measure the bursty traffic, i.e., TCP traffic.

The impacts of the propagation delay between core routers and the measurement interval time of ingress routers on the allocation error rate are shown in Fig. 14. The link delays might be another significant factor in the erroneous measurement in addition to the measurement interval time because the proposed method is based on a feedback-driven control mechanism that is sensitive to delays. The values along the z-axis are the average allocation error rates for subscribers A and B. The simulation results show that the allocation error rate converges to nearly zero as the measurement time interval of ingress routers lengthens regardless of the propagation delays. In other words, the measurement time interval can limit the impact of the propagation delays. It follows from these results that the measurement time interval of ingress routers is definitely related to the allocation error rate. It is one of the adjustable parameters, unlike in the case of the link delays. Therefore, we conclude that the measurement time interval of ingress routers should be long enough to stabilize the proposed method.

The above three simulation results show the impact of inaccurate measurement on short measurement time intervals. To stabilize the proposed method and determine the measurement time interval, we analyze the required minimum measurement time interval in ingress routers. The impacts of link delays and subscriber imbalance ratio on the required minimum measurement time interval are shown in Fig. 15. The x-axis represents the propagation delay between core routers C1 and C2, and the y-axis indicates the imbalance ratio of the numbers of sites of subscribers B and A. The minimum values satisfying the precise measurement of the number of active sites (i.e., exactly 200) are plotted toward the z-axis. As for the propagation delays, the required value does not change or slightly increases depending on the imbalance ratio for subscribers. The imbalance ratio of the number of sites had more impact on the measurement time interval. The required value shows a linear increase. From this result, we clarify that the correlation between the stability of the proposed method and the link delays is low.

Finally, we show the reason why this inaccurate measurement occurs. Figure 16 shows the packet arrival timing at the core router C1. The x-axis represents the arrival time and the y-axis represents the source node identifier. The measurement time interval in Figs. 16(a), 16(b) and 16(c) is 10 ms, 100 ms and 500 ms, respectively. The propagation link delay between core routers C1 and C2 is 10 ms. In this case, no packet arrivals in certain measurement period in Fig. 16(a) while at least one packet arrivals in every measurement period in Figs. 16(b) and 16(c). We conclude these results that, in the short measurement time interval such as 10 ms in Fig. 16(a), the ingress routers cannot accurately determine whether the traffic is active or not.

In Fig. 17, the propagation link delay between core routers C1 and C2 is 500 ms. The measurement time interval in Figs. 17(a), 17(b) and 17(c) is 10 ms, 100 ms and 500 ms, respectively. By contrast with Fig. 16, even when the time interval is 10 ms and 100 ms, respectively, no packet arrivals in certain measurement period in Figs. 17(a) and 17(b). Regarding the time interval 500 ms in Fig. 17(c), at least one packet arrivals in every measurement period. Through these analysis, we can conclude that the measurement time interval should be increased as the propagation delay increases. In such ways, the hose bandwidth allocation method can be effective in high bandwidth delay product networks.

From the above simulation runs, we clarified that the measurement time interval of ingress routers is a significant factor for the stability of the hose bandwidth allocation method. Therefore, to reduce the allocation error rate, we should set the long time interval for the ingress routers. We note that it is necessary to determine the optimal measurement time interval for each router. We consider that a control theory may be applicable to the hose bandwidth allocation method such as the explicit control protocol (XCP) [37]. Although it is beyond the scope of this study, the stability of such feedback systems might be achieved by plotting their open-loop transfer function on a Nyquist plot.

2.5 Summary

In Chapter 2, we focused on an MTA service in the hose model and explained the need for a bandwidth allocation method. On the assumption that network provisioning has already been completed, we designed a hose bandwidth allocation method to achieve the MTA service in the hose model. The proposed method provides at least the minimum agreed throughput to active hoses even during a period of congestion. The basic idea is that the proposed method gathers available bandwidth information from inside a network and divides the available bandwidth into hoses on each bottleneck link using the information. The proposed method was designed to meet the following two requirements: (1) fairness in terms of hoses and pipes and (2) high utilization of network resources.

We ran computer simulations to examine the proposed method's advantages and to analyze its scalability and stability. First, we determined the basic behavior of the proposed method in throughput dynamics. The proposed method eliminated the impact of new traffic insertion and existing traffic removal on other traffic. We then investigated its accuracy and effectiveness. Simulation results showed that the proposed method achieved a fairly low allocation error rate. Regarding the utilization, the link utilization that it achieved was as high as that achieved with the WPFRA method. In the case of a topology with multiple bottleneck links, we found that the proposed method could satisfy the minimum agreed throughput by dividing the available bandwidth using the information inside the network. In the next simulation, to analyze the scalability of the proposed method, we examined the impact of the ratio of the scale of subscribers on the allocation error rates. We proved that the proposed method kept the allocation error rate low even when the scale of the VPN increased. Finally, to clarify the stability of the proposed method, we analyzed the factors affecting the allocation error in the last simulation runs and clarified that the measurement time interval of ingress routers was definitely related to the allocation error rate.

Number of destination sites	A : B = 2 : 200
Number of TCP flows	A : B = 600 : 600

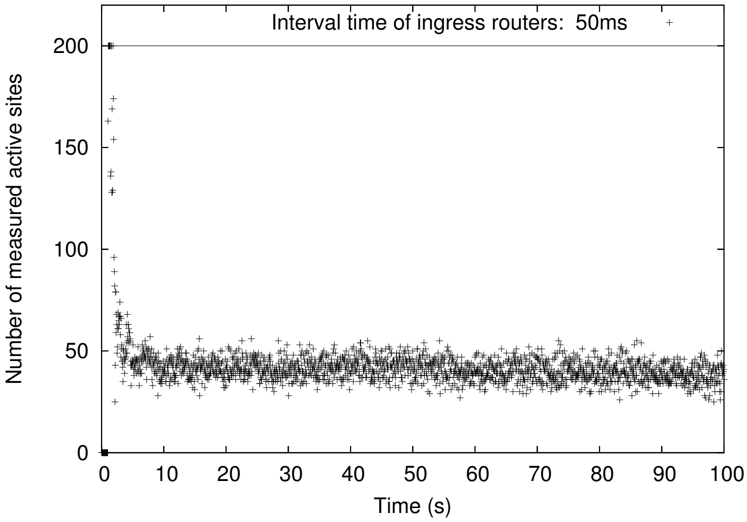


Figure 12. Error in number of active sites measured at ingress routers

Number of destination sites	A : B = 2 : 200
Number of TCP flows	A : B = 600 : 600

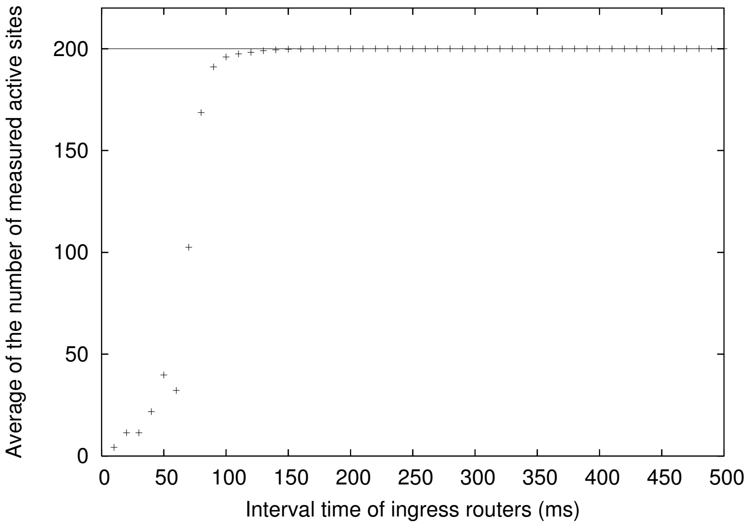


Figure 13. Impact of measurement time interval at ingress routers on measurement error

Number of destination sites	A : B = 2 : 200
Number of TCP flows	A : B = 600 : 600

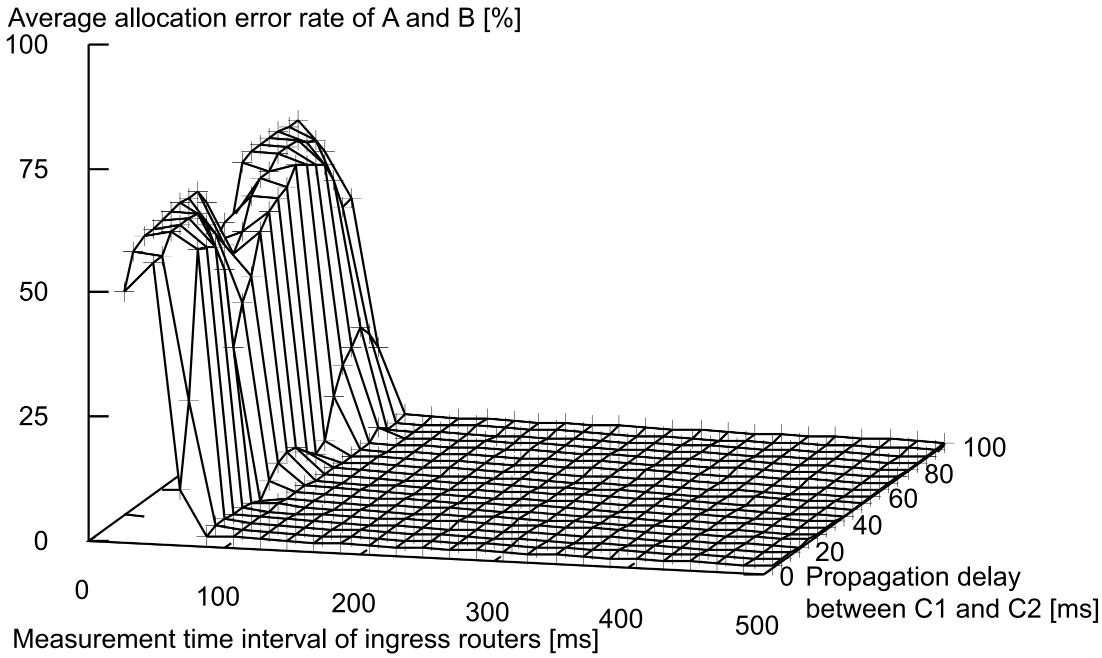


Figure 14. Impact of measurement time interval in ingress routers on allocation error rate

On the other hand, the link delay was not a significant factor for the stability of the proposed method.

In summary, we proposed the essential online QoS traffic control mechanism for achieving the MTA service. We showed that the proposed method satisfied both the requirements for achieving an MTA service in the hose model, clarified the scalability of the proposed method, and indicated how to decide its system parameters.

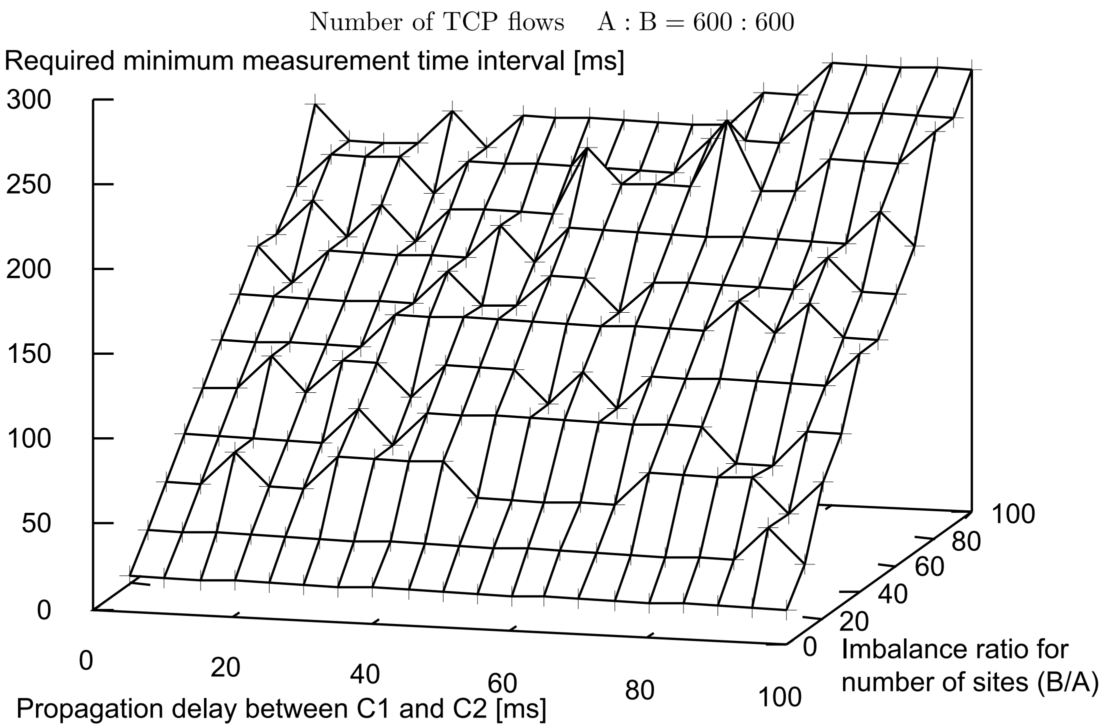
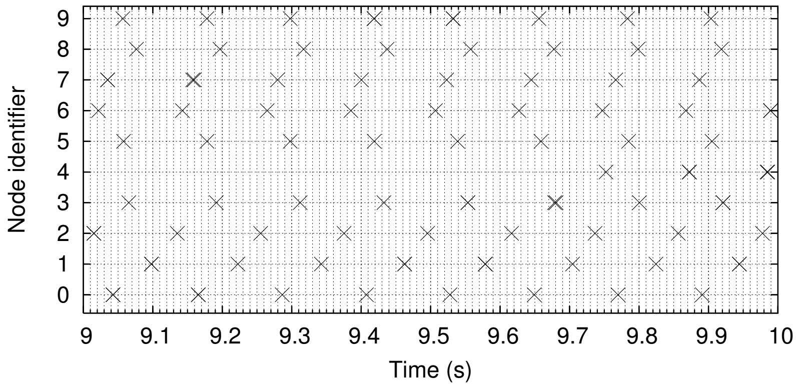
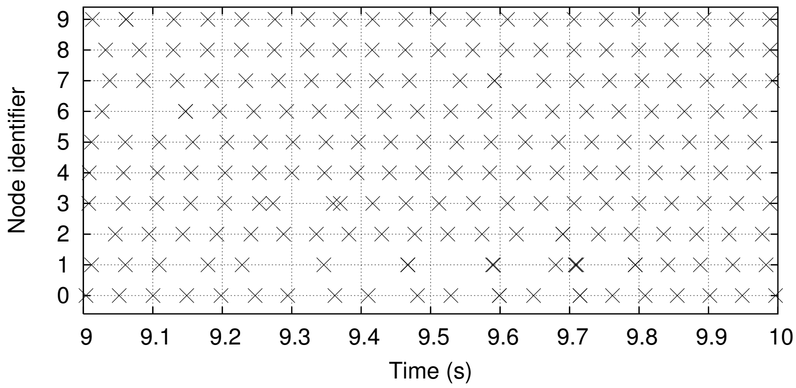


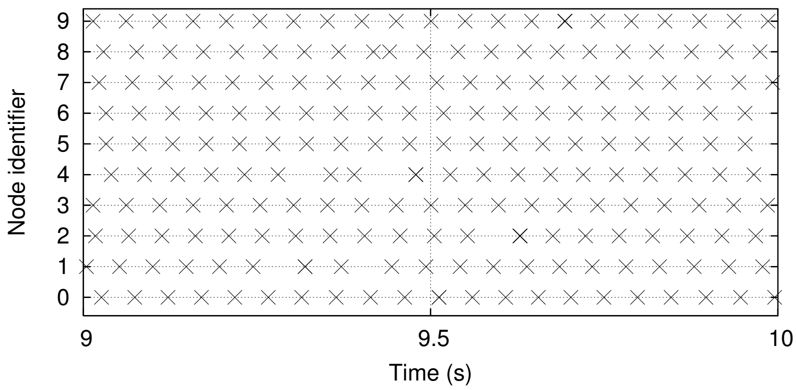
Figure 15. Required minimum measurement time interval in ingress routers for precise measurement



(a) Measurement time interval 10 ms / Link delay 10 ms

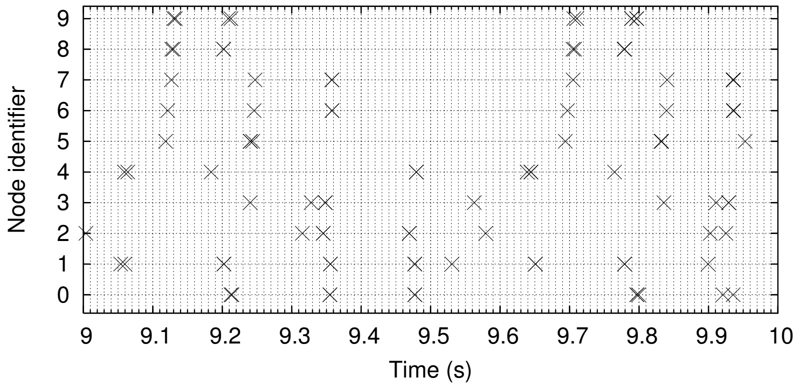


(b) Measurement time interval 100 ms / Link delay 10 ms

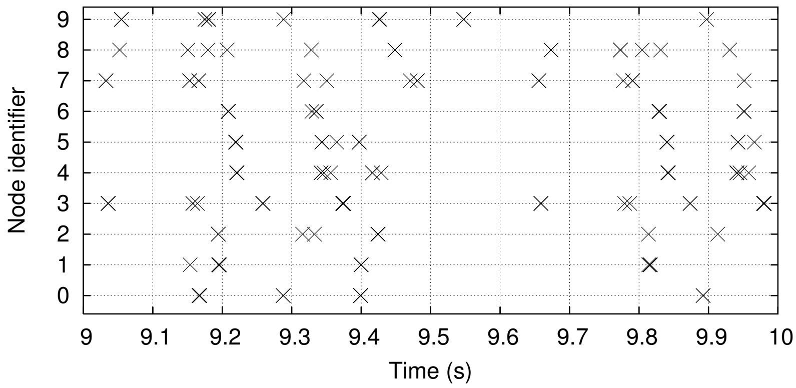


(c) Measurement time interval 500 ms / Link delay 10 ms

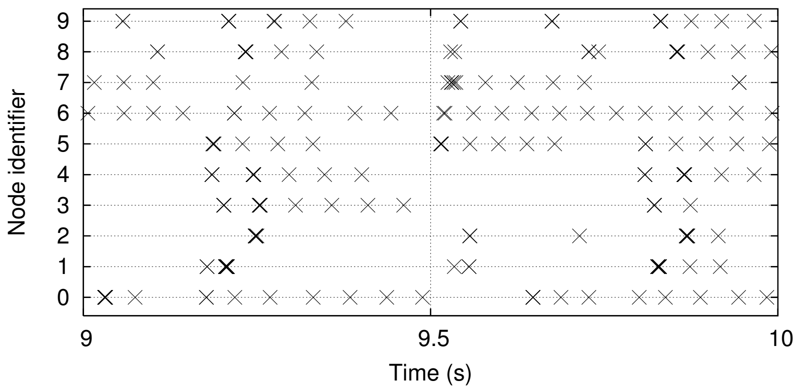
Figure 16. Packet arrival timing at core router C1 in link propagation delay 10 ms



(a) Measurement time interval 10 ms / Link delay 500 ms



(b) Measurement time interval 100 ms / Link delay 500 ms



(c) Measurement time interval 500 ms / Link delay 500 ms

Figure 17. Packet arrival timing at core router C1 in link propagation delay 500 ms

3. Provisioning algorithm for MTA service

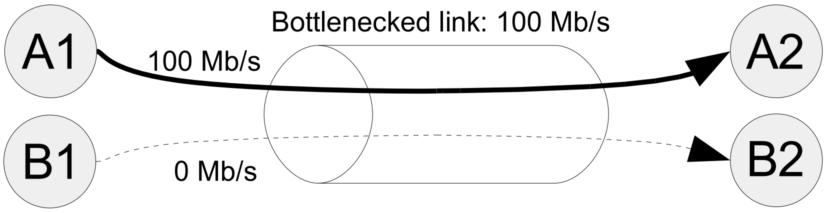
We described the importance of PPVPNs with QoS assurance in Chapter 1 and the bandwidth allocation method, which is one of the important components for PPVPNs with QoS assurance, in Chapter 2. However, merely with the bandwidth allocation method, we cannot provide QoS assurance in PPVPNs. To utilize the bandwidth allocation method for achieving PPVPNs with QoS assurance practically, the provisioning algorithm must be needed. The provisioning algorithm can determine the adequate bandwidth for all customers in offline.

3.1 Introduction

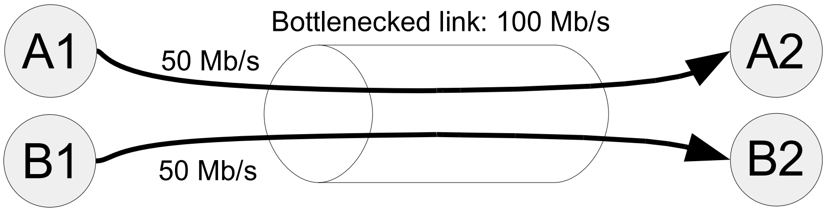
Virtual private networks (VPNs) are used by many organizations as private information networks connecting distant sites at costs lower than those possible with a network of leased lines; however, VPNs have difficulty meeting quality of service (QoS) requirements because they must share network resources among all users. To overcome this problem and satisfy some of the QoS requirements, a framework called provider-provisioned VPN (PPVPN) was developed by VPN providers. Because customers contract with the provider offering PPVPN, the provider must adequately allocate limited network resources and satisfy requirements for its customers.

The current PPVPN does not support bursty Internet traffic well because it allocates network resources to customers statically. If VPN providers guarantee customers a static amount of bandwidth, the result is low bandwidth utilization. One way of supporting bursty Internet traffic is to assure the minimum throughput to customers. With minimum throughput assurance (MTA) service, a customer can attain substantially higher throughput than an agreed minimum throughput during periods of non-congestion. During periods of congestion, the minimum agreed throughput is provided as an average over a certain span of time. Therefore, MTA service can provide higher throughput predictability than best-effort VPN service.

To further describe the MTA service, we illustrate throughput allocation scenarios in Fig. 18. In the figure, aggregated traffic from site A1 to A2 and from site B1 to B2 are each assigned 50 Mb/s as a minimum throughput. We denote



(a) No congestion scenario



(b) Congestion scenario

Figure 18. Throughput allocation in MTA service

macro flow as aggregated traffic. Fig. 18(a) depicts a scenario with no congestion, i.e., the macro flow from site B1 to B2 is idle. In this case, the macro flow from site A1 to A2 can obtain more than the minimum throughput. Conversely, the throughput for the macro flow from site A1 to A2 is only 50 Mb/s when the link is congested, as illustrated in Fig. 18(b); however, even in the congestion scenario, the MTA service can provide the minimum throughput for any macro flows.

Much research on MTA service has been performed. Clark et al. proposed random early drop routers with in/out bit (RIO) [38] to allocate capacity for best-effort VPN service. According to their performance evaluation, RIO can allocate capacity with slight effect on RTT, which is considered an important milestone in MTA service. Fàbrega et al. proposed a guaranteed minimum throughput service for TCP flows using measurement-based admission control [39]. Several MTA service methods have also been proposed based on ABR service of ATM. Li et al. proposed a core-stateless congestion avoidance scheme for IP networks [40]. Lee et al. proposed a weighted proportional fair rate allocation (WPFRA) method for differentiated service (DiffServ) on the Internet [21]. Yokoyama et al. proposed a one-way version of the WPFRA method to improve its performance [41]. Moreover, as described in Chapter 2, we proposed an extension of the WPFRA method [42] for the VPN hose model [23] which is a VPN model to utilize the network resource efficiently.

To apply the aforementioned methods, certain parameters should be decided offline. For example, RIO requires a target rate for each TCP micro/macro flow, and the WPFRA method requires the weight for each macro flow; however, no parameter determination algorithm has been proposed. The difficulty in achieving such a determination algorithm is to meet the minimum throughput requirements in any active state matrix[2], in which each element of the matrix represents an active/idle state of traffic from a source to a destination.

In this study, we determine adequate parameters for the MTA service, and we define this as a provisioning. Although determination algorithms for network topology and active queue management have been proposed [43, 44], they cannot satisfy the minimum required throughput of every customer in MTA service when active state matrices change dynamically, i.e., elements of the active state matrix may be idle or may frequently switch from active to idle states. Therefore, these algorithms cannot be applied to the provisioning process for the MTA service.

We propose a provisioning algorithm that uses mathematical programming, in particular, nonlinear programming (NLP), for providing MTA service in PPVPNs. Mathematical programming is widely and effectively used for network provisioning [24]. The proposed algorithm adequately distributes the limited network bandwidth to all customers within the given constraints.

Following this Introduction, Chapter 3 is organized as follows. We describe the WPFRA method in Sect. 3.2. Section 3.3 presents the proposed provisioning algorithm for the MTA service. We demonstrate the effectiveness of our approach with quantitative results in Sect. 3.4 and describe the difference between our studies and related studies in Sect. 3.5. Finally, we offer our conclusions in Sect. 3.6.

[2]A formal definition of an active state matrix is described in Fig. 20 within Sect. 3.3.2.

3.2 WPFRA method

In the Introduction, we argued that MTA service improves the throughput predictability of bursty traffic. To implement the MTA service, we set three requirements: 1) meet the minimum throughput requirement from customers in any active state matrix, 2) fairly distribute the spare bandwidth, and 3) utilize network resources efficiently.

The WPFRA method and its variants are approaches to the MTA service that has a potential to meet the three requirements above. By distributing spare bandwidth based on weight values, the WPFRA method provides weighted proportional fairness. Before using the WPFRA method, the VPN provider should decide the weight values for all combinations of source and destination sites. Deciding the weight values to satisfy the minimum throughput requirement (requirement (1) above) remains a challenge. Determining the weight values involves finding a common matrix of values that satisfy the minimum throughput requirement for all active state matrices.

Before we propose a common weight matrix determination algorithm to satisfy minimum throughput requirements for all elements of the active state matrix, we first describe the WPFRA method. The WPFRA method can distribute the available bandwidth in proportion to the weight value of each macro flow. Edge and core routers form a VPN provider's network and provide feedback-driven traffic control to utilize the network bandwidth efficiently.

Ingress and core routers periodically measure the amount of arriving traffic and calculate a fair share rate r_f that indicates the amount of allocated capacity for a macro flow with a weight of one. A fair share rate is calculated for each output link in a router. Egress edge routers periodically send a notification packet to each ingress edge router, and core routers update the value of r_f inside the notification packet if it is smaller than the current value stored in the packet. Ingress routers obtain the r_f from the received notification packet and set the value to the explicit rate ER, which indicates that the allocated throughput for a macro flow has a weight of one. Finally, they calculate the allocated throughput as ER multiplied by the customer's weight.

To illustrate how the WPFRA method determines the allocated throughput, we model the calculation process on Ref. [21] utilizing the following algorithm:

Initialization

Let C be the link capacity and S be the spare bandwidth in the current iteration, which is set to

$$S_{(s,t)} \Leftarrow C_{(s,t)},$$

where (s,t) is a link directly connecting source node s and target node t.

Step 1

Let $W_{(s,t)}$ be the sum of each weight value corresponding to all macro flows passing through a link (s,t) and $r_{f_{(s,t)}}$ be the allocated capacity for a macro flow with the weight of one in the link. Then, undetermined $r_{f_{(s,t)}}$ is given by

$$r_{f_{(s,t)}} \Leftarrow \frac{S_{(s,t)}}{W_{(s,t)}}.$$

Step 2

After $r_{f_{(s,t)}}$ is calculated, explicit rate $\mathrm{ER}_{(i,e)}$ is calculated by

$$\mathrm{ER}_{(i,e)} \Leftarrow \min(r_{f_{(s,t)}}),$$

where (i,e) is a path between ingress and egress routers that contains link (s,t).

Step 3

Let w be the previously assigned weight and B be the allocated capacity of each customer given by

$$B_{(m,n)} \Leftarrow \mathrm{ER}_{(i,e)} \cdot w_{(m,n)}, \tag{3}$$

where (m,n) is a path between source and destination sites that contains path (i,e).

Step 4

Then, $S_{(s,t)}$ and $W_{(s,t)}$ are updated by

$$\begin{aligned} S_{(s,t)} &\Leftarrow S_{(s,t)} - B_{(m,n)}, \\ W_{(s,t)} &\Leftarrow W_{(s,t)} - w_{(m,n)}, \end{aligned} \tag{4}$$

where path (m,n) also contains link (s,t).

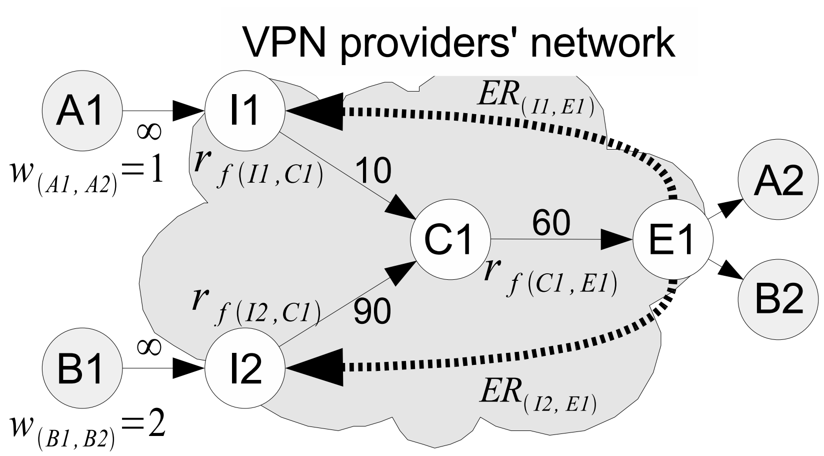


Figure 19. Example of throughput calculation: Network topology

Step 5

Finally, the calculation is terminated if all $W_{(s,t)}$ values become zero: i.e., all $r_{f_{(s,t)}}$, $\mathrm{ER}_{(i,e)}$, and $B_{(m,n)}$ are determined. If not, go back to Step 1.

To illustrate this algorithm, we provide the example depicted in Fig. 19. In this example, there are four sites: customer A has two sites, A1 and A2, and customer B has two sites, B1 and B2. The traffic of customer A (B) originates at A1 (B1) and is destined for A2 (B2). As described in the figure, the values $w_{(\mathrm{A1,A2})}$ and $w_{(\mathrm{B1,B2})}$ are assigned weights of 1 and 2, respectively. The numbers associated with links in the figure indicate the amount of bandwidth for each link. If $C_{(\mathrm{I1,C1})}$ is larger than 20, $B_{(\mathrm{A1,A2})}$ and $B_{(\mathrm{B1,B2})}$ become 20 and 40, respectively. Because $C_{(\mathrm{I1,C1})}$ is limited to 10, throughput calculation is difficult.

Suppose customers A and B are both active, meaning both are producing traffic. Then, $r_{f_{(\mathrm{I1,C1})}}$, $r_{f_{(\mathrm{I2,C1})}}$, and $r_{f_{(\mathrm{C1,E1})}}$ converge at 10, 45, and 25, respectively, based on the above algorithm. As a result, $\mathrm{ER}_{(\mathrm{I1,E1})}$ becomes 10 and $\mathrm{ER}_{(\mathrm{I2,E1})}$ becomes 25. Since $w_{(\mathrm{A1,A2})}$ and $w_{(\mathrm{B1,B2})}$ are 1 and 2, respectively, $B_{(\mathrm{A1,A2})}$ becomes 10 and $B_{(\mathrm{B1,B2})}$ becomes 50. These results are summarized in Table 8. The first two columns depict the active state field, a value of 1 indicating an active state, a value of 0 indicating an idle state.

As described above, the WPFRA method determines the explicit rate ER and the allocated bandwidth B of each customer, and these values are calculated from

Table 8. Throughput calculation results in Fig. 19

Active state		$B_{(\mathrm{A1,A2})}$	$B_{(\mathrm{B1,B2})}$
A1 → A2	B1 → B2	$(\mathrm{ER}_{(\mathrm{I1,E1})})$	$(\mathrm{ER}_{(\mathrm{I2,E1})})$
1	1	10 (10)	50 (25)
1	0	10 (10)	0 (50)
0	1	0 (10)	60 (30)
0	0	0 (10)	0 (60)

the topology, active state matrix, and weight matrix. The network topology is generally assumed to be constant during a long time, whereas the active state and weight matrices are dynamically changing. Therefore, the provider needs to determine a weight matrix that can accommodate any active state matrix.

3.3 Weight determination algorithm

In the previous section, we described the WPFRA method and the difficulty of determining adequate parameters for the WPFRA method. In this section, we propose the weight determination algorithm for allocating sufficient network resource to every customer using MTA service. The proposed algorithm is divided into two sub-algorithms: a weight matrix determination algorithm for each active state matrix; and a common weight matrix determination algorithm for every active state matrix.

3.3.1 Weight matrix determination algorithm

We first develop an algorithm to calculate adequate weight values based on network topology, a minimum throughput matrix, and an active state matrix. The provider needs to allocate sufficient bandwidth satisfying the minimum throughput of every customer using limited network resources. We assume these constraints as part of a mathematical programming problem in which the allocated throughput and spare bandwidth can be calculated using Eqs. (3) and (4). Let B_{ave} be the average throughput of all customers, B_{var} be the variance of the average throughput of all customers, M be the required throughput of customers, and U be the link utilization. The objective function and constraints are formulated as follows:

Objective function

* $B_{\mathrm{ave}} - B_{\mathrm{var}}$

Constraints

* $B_{(m,n)} \geq M_{(m,n)}$
* $U_{(s,t)} \leq 1$
* $w_{(m,n)} \geq 1$.

We adopt the objective function based on the evaluation index in Ref. [45], which specifies that each customer can obtain high throughput and the allocated throughput among customers is balanced. Regarding alternative objective functions, we can alternatively utilize the functions maximizing the link utilization of all links in the network or maximizing the total throughput of all customers.

The allocated throughput, spare bandwidth, and utilization are all represented by polynomials. Thus, this mathematical programming can be classified as nonlinear programming (NLP). Table 9 shows an example of calculated weights in a simple NLP. In this table, the first three columns represent an active state matrix, the second three columns represent a calculated weight matrix using NLP, the third three columns represent the allocated bandwidth calculated from the weight matrix, the fourth column represents the resulting value of the objective function, and the fifth column represents the link utilization. In the given example, three customers (X, Y, Z) share a single bottlenecked link. The required minimum throughput of (X, Y, Z) is $(10, 20, 60)$, and the bandwidth of the bottlenecked link is 100. The calculated weight values are listed for all active state matrices, i.e., $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) = (1, 1, 1)$, $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) = (1, 0, 1)$, and so on. All resulting allocated throughput values are equal to or greater than the respective minimum throughput requirements.

Table 9. Example of weights calculated using NLP

Active state			Calc. weights			Alloc. bw.			Obj. func.	Util.
X	Y	Z	X	Y	Z	X	Y	Z		
1	1	1	1	1	3	20	20	60	6.7	1.0
1	0	1	2	1	3	40	0	60	-13.3	1.0
0	1	1	1	2	3	0	40	60	-13.3	1.0
1	1	0	1	1	1	50	50	0	-8.3	1.0

Single bottlenecked link: 100

Required minimum throughput: $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) = (10, 20, 60)$

3.3.2 Common weight matrix determination algorithm

As we saw in Tables 8 and 9, the allocated throughput is affected by the active state matrix. An illustration of this effect is shown in Fig. 20. The active state matrix represents an active/idle state of traffic from a source to a destination site, i.e., the values of 0 and 1 represent idle and active states, respectively. The minimum throughput matrix describes each customer's minimum throughput requirements from source site i to destination site j. The zero element of the active state matrix results in a zero value in the minimum throughput matrix at the same element in the active state matrix.

The weight matrix consists of weight values used by the WPFRA method. When we fix the active state matrix, we can calculate a weight matrix W_k using NLP, however, if we change the active state matrix, the weight matrix $W_{k'}$ also changes. Since dynamic changes to the weight matrix is difficult implement, we aim to calculate a common weight matrix that meets the minimum throughput requirement for any active state matrix. Therefore, we need a strategy to find such a weight matrix.

To derive a common weight matrix, we iteratively use NLP. Since NLP requires an initial weight matrix, the previously derived weight matrix for an active state matrix can be used as the initial NLP weight matrix with the input of another active state matrix. We hypothesize that this iteration of NLP converges on the

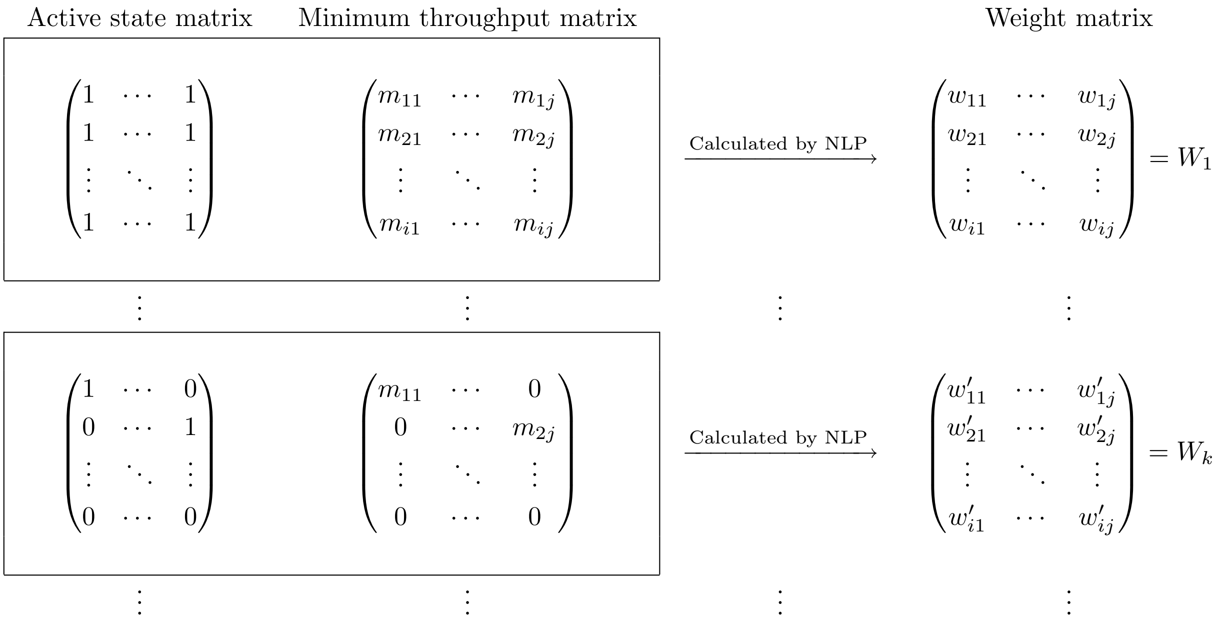


Figure 20. Common weight matrix

common weight matrix and meet the minimum throughput requirements. Based on this hypothesis, we define a basic strategy to obtain the common weight matrix detailed as follows:

Iteration

Previous output values (i.e., the weight matrix calculated by NLP) are utilized as an initial weight matrix in the next step.

One distance shift

First, we define the number of elements that differ between two active state matrices as a distance. Through the iteration of active state matrices, we need to shift an active state matrix to other active state matrices. To avoid drastic shifts in active state matrices, we shift the current active state matrix to the next active state matrix by a distance of 1. For example, suppose an active state matrix is $(0, 1, 1)$, where 0 and 1 represent idle and active states, respectively, we set the next active state matrix to $(0, 1, 0)$ because only the third element in these two active state matrices differ, thus the difference is 1.

Normalization

The weights represent a proportional ratio of the allocated bandwidth, i.e., the calculated weight matrix (a, b, c) is equivalent to (ka, kb, kc), where k is a constant number. The allocated bandwidth calculated by these two weight matrices are the same. Therefore, we divide the weight matrix into collision groups, then normalize the grouped elements of the weight matrix, such as $(ka, kb, kc)/k \Rightarrow (a, b, c)$, in each collision group.

Stabilization

An element of the weight matrix corresponding to the zero element in the active state matrix can be selected for any value. To avoid needless change in weight values in such cases, we define constraints for such elements, i.e., we select the value obtained in the previous iteration.

Search domain reduction

To avoid irrelevant searching, we set lower and upper limits on the variables of the objective function. The lower limit is 1, derived directly from the constraints; the upper limit is $(C_{\min} - M_{\min})/M_{\min}$, where $C_{\min}$ represents the minimum value of the link bandwidth in the topology and $M_{\min}$ represents the minimum value of the minimum required throughputs. To explain the reason of the upper limit, we suppose that the following: 1) multiple customers share the link with capacity $C_{\min}$ in the network, 2) one customer requires the minimum required throughput $M_{\min}$, 3) the minimum weight value 1 is assigned to this customer, and 4) the total weight value α is assigned to other customers. In this case, because the limited link capacity $C_{\min}$ should be shared among multiple customers, the following equation $1/(1 + \alpha) \cdot C_{\min} \geq M_{\min}$ (i.e., $\alpha \leq (C_{\min} - M_{\min})/M_{\min}$) should be satisfied.

We construct a common weight determination algorithm based on the above strategy and illustrate it in Fig. 21 using pseudocode. In the proposed algorithm, we determine a representative weight matrix with values selected from a large number of calculated weight matrices utilizing a convergence index, which is defined as the maximum ratio of the number of common weight matrices to the number of all weight matrices. We call this index the batched convergence index.

We show example behavior of the proposed algorithm in the context of Table

10. We suppose the required minimum throughput matrix of customers (X, Y, Z) is (10, 20, 60), respectively. There is a single bottleneck link with link capacity 100. In the first cycle of the proposed algorithm, after initializing the weight matrix to $(1, 1, 1)$, four weight matrices $(1, 2, 3)$, $(1, 1, 3)$, $(1, 1, 3)$, and $(2, 1, 3)$ are calculated via NLP. Note that we skip non-congestion active state matrices because the weight matrices in such active state matrices can be arbitrary values. We denote such skipped values by $(-, -, -)$.

The highest distribution of batched tuples of $(1, 1, 3)$ is 0.5. Therefore, we select the weight matrix $(1, 1, 3)$ as the representative weight matrix in the first cycle. This representative weight matrix is utilized as the initial weight matrix in the second cycle. Similar to the first cycle, four weight matrices are calculated via NLP, and the representative weight matrix $(1, 1, 3)$ is selected in the second cycle. In this example, the weight matrix $(1, 1, 3)$ is suitable to every active state matrix.

Users of the proposed provisioning algorithm should specify the required minimum throughput for each customer site so that the weight matrix satisfying minimum throughput requirements can be obtained. The fundamental calculation complexity is $l \prod_{k=1}^{u} 2^{v_k(v_k-1)}$, where u is the number of customers, v is the number of sites per customer, and l is the number of calculation cycles.

3.4 Numerical demonstration

In the previous section, we presented the common weight matrix determination algorithm, which is based on the hypothesis that weight matrices converge through iterations of NLP. In this section, we demonstrate that the proposed algorithm converges on a common weight matrix that meets the minimum throughput requirements. We also investigate the impact of imbalanced minimum throughput requirements and whether links have spare bandwidth available.

We use the network topology illustrated in Fig. 22 containing one core and three edge routers, which together form a star topology with a link capacity of 80 on all links. All edge routers behave as both ingress and egress routers, i.e., traffic is bidirectional between each of a particular customer's sites. There are two customers (A, B), each of whom has three sites $(\mathrm{A1}, \mathrm{A2}, \mathrm{A3})$ and $(\mathrm{B1}, \mathrm{B2}, \mathrm{B3})$. Customer sites are connected to respective edge routers.

```
LoopNLP(iteration){
  InitAllWeights(weights);
  SetInputValue(topology, reqbw);
  for ( i=0; i<iteration; i++ ) {
    foreach (pattern) {
      weights = RunNLP(topology,
        pattern, reqbw, weights);
    }
    normalization(pattern, weights);
  }
}

RunNLP(t, p, b, w){
  foreach (p) {
    if( active[i] == 0 ){
      # Stabilize value of w[i]
      SetConstraint(w[i]);
    }
  }
  SetSearchDomain(t, b);
  ans = SolveNLP(t, p, b, w);

  return ans;
}

InitAllWeights(weights){
  weights = (1, 1, ..., 1);
}

normalization(pattern, weights){
  group = divide_group(pattern);

  foreach (group) {
     normalized_w[i] =
         weights/min(weights);
  }
  return normalized_w[i];
}
```

Figure 21. Pseudocode of the weight determination algorithm

Table 10. Example behavior of the proposed algorithm

First cycle

Initial weight matrix in first cycle: $(X, Y, Z) = (1, 1, 1)$

Active state			Calc. weight			Alloc. bw.			Obj. func.	Util.
X	Y	Z	X	Y	Z	X	Y	Z		
0	0	1	–	–	–	–	–	–	–	–
0	1	1	1	2	3	0	40	60	-13.3	1.0
0	1	0	–	–	–	–	–	–	–	–
1	1	0	1	1	3	50	50	0	-8.3	1.0
1	1	1	1	1	3	20	20	60	6.7	1.0
1	0	1	2	1	3	40	0	60	-13.3	1.0
1	0	0	–	–	–	–	–	–	–	–

Rep. weight matrix in 1st cycle: $(A, B, C) = (1, 1, 3)$

Second cycle

Initial weight matrix in second cycle: $(X, Y, Z) = (1, 1, 3)$

Active state			Calc. weight			Alloc. bw.			Obj. func.	Util.
X	Y	Z	X	Y	Z	X	Y	Z		
0	0	1	–	–	–	–	–	–	–	–
0	1	1	2	2	3	0	40	60	-13.3	1.0
0	1	0	–	–	–	–	–	–	–	–
1	1	0	1	1	3	50	50	0	-8.3	1.0
1	1	1	1	1	3	20	20	60	6.7	1.0
1	0	1	2	1	3	40	0	60	-13.3	1.0
1	0	0	–	–	–	–	–	–	–	–

Rep. weight matrix in 2nd cycle: $(A, B, C) = (1, 1, 3)$

Single bottlenecked link: 100

Required minimum throughput: $(X, Y, Z) = (10, 20, 60)$

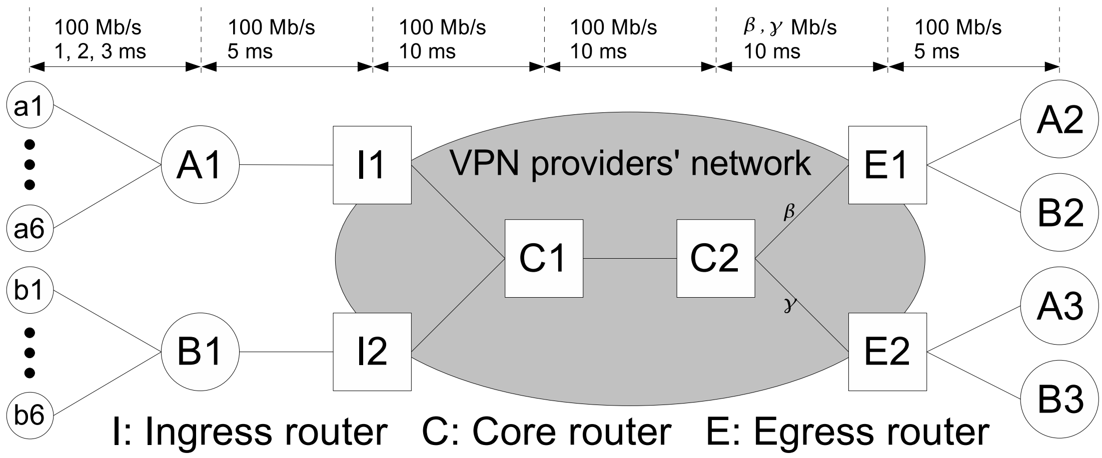


Figure 22. Experimental topology

In the numerical demonstration, we vary the required minimum throughput to investigate the impact of the imbalance ratio and the amount of the minimum throughput as follows:

(a) $\mathrm{A}_{(x,y)} : \mathrm{A}_{(x,z)} : \mathrm{B}_{(x,y)} : \mathrm{B}_{(x,z)} = 20 : 20 : 20 : 20$

(b) $\mathrm{A}_{(x,y)} : \mathrm{A}_{(x,z)} : \mathrm{B}_{(x,y)} : \mathrm{B}_{(x,z)} = 10 : 10 : 10 : 10$

(c) $\mathrm{A}_{(x,y)} : \mathrm{A}_{(x,z)} : \mathrm{B}_{(x,y)} : \mathrm{B}_{(x,z)} = 30 : 10 : 10 : 30$

(d) $\mathrm{A}_{(x,y)} : \mathrm{A}_{(x,z)} : \mathrm{B}_{(x,y)} : \mathrm{B}_{(x,z)} = 15 : 5 : 5 : 15.$

There is spare bandwidth in scenarios (b) and (d); further, the required minimum throughput values are imbalanced in scenarios (c) and (d).

To evaluate the proposed algorithm, we define the following two evaluation indices: (1) the success rate, which is the number of cases satisfying the minimum required throughput using the representative weight matrix in each active state matrix; and (2) the average link bandwidth defined as the average of the ratio of the total throughput to the total capacity of active links. We calculate these values using the representative weight matrix calculated using the proposed algorithm.

Fig. 23 shows results for the four required minimum throughput scenarios listed above. The x-axis represents the amount of processing from 1 to 10 iterations. The y-axis represents the resulting values of the batched convergence index, the success rate, and the average total link utilization.

As the number of iterations increases, the batched convergence index converges, indicating that the representative weight matrix is, in fact, determined through such iterations. The average total link utilization remains constant at approximately 0.85, although the number of iterations increases. This shows that the proposed algorithm retains the fundamental advantage of mathematical programming: i.e., the proposed algorithm can keep the overall average total link utilization under the given constrained, even as the number of iterations increases.

We found that the success rate progressively converges to 1.0 for all required throughput, showing that the proposed algorithm can determine the common weight matrix from the required minimum throughput matrix.

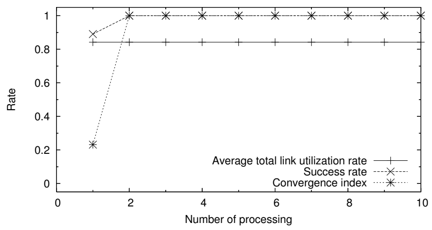
Next, we describe the impact of spare capacity on the success rate. Under spare capacity scenarios (b) and (d), the batched convergence index and the success rate is substantially lower than that of scenarios (a) and (c); this occurs because the calculated weights vary over a wide range due to the high latitude of values.

Finally, we discuss the impact on the success rate of imbalanced required throughput among sites. The proposed algorithm requires a relatively larger number of iterations for the success rate to converge in imbalanced request scenarios (c) and (d); this occurs because of the complicated network topology and requests, although it also demonstrates the effectiveness of the iterative algorithm.

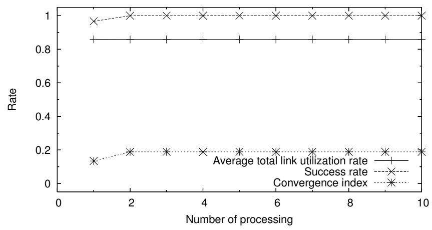
3.5 Related work on provisioning algorithms

In this section, we describe the difference between the proposed algorithm and two conventional provisioning algorithms.

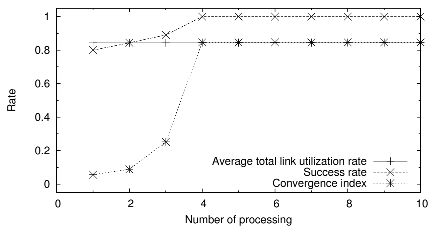
The first conventional algorithm [43] can determine adequate network topology for the VPN hose model [23]. It can reduce the required bandwidth and achieve high bandwidth efficiency. Although this algorithm is helpful in constructing the network topology, it cannot achieve the MTA service because the algorithm cannot allocate network resources to the customers adequately.



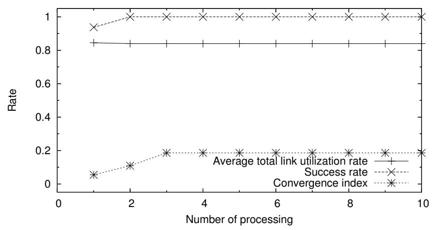
(a) Balanced requests/No spare capacity



(b) Balanced requests/Spare capacity



(c) Imbalanced requests/No spare capacity



(d) Imbalanced requests / Spare capacity

Figure 23. Success rate of the batched convergence index

The second conventional algorithm [44] adjusts parameters of a weight fair queuing (WFQ) scheduler. This algorithm achieves fairness for a single output link at the WFQ-supported router; however, it cannot support allocating available bandwidth across the entire network. Therefore, this algorithm cannot adapt to provide the MTA service.

3.6 Summary

In Chapter 3, we proposed the essential offline QoS traffic control mechanism for achieving the MTA service. To satisfy minimum throughput requirements of VPN customers, we proposed a provisioning algorithm that supports the MTA service. The proposed algorithm will be a necessary tool for QoS traffic control by providers. In the proposed algorithm, we first created an NLP to find a weight matrix for each active state matrix. Then we proposed a common weight matrix determination algorithm based on the hypothesis of convergence in elements of the iteration of the NLP. Through the numerical demonstration, we showed that the proposed algorithm converges on a common weight matrix using the convergence index. We also evaluated the impact on convergence of the imbalanced amount of required throughput and the presence of spare bandwidth.

Through this study, we established this research as a primary research of the MTA service provisioning although the computational complexity of the proposed algorithm was high. In future work, to concentrate the convergence index more precisely and rapidly, we will focus on a pruning algorithm and consider highly correlated convergence indices.

4. Discussion

Through this dissertation, we proposed two essential mechanisms for achieving the MTA service. We coped with the issue in the MTA service from two perspectives. The one is the hose bandwidth allocation method for online QoS traffic control in the MTA service; the other is the provisioning algorithm for offline QoS traffic control in the MTA service. In this chapter, we discuss the feasibility, efficiency, deployment strategy, and next milestone of the two proposed methods.

4.1 Feasibility about the hose bandwidth allocation method

Although this dissertation showed the effectiveness of the hose bandwidth allocation method described in Chapter 2, sophisticated routers will be needed for deploying the proposed method in the future network. Especially, these routers maintain the large amount of path information even though these routers do not need to maintain the massive amount of flow connection. Therefore, the relay speed and functionality of routers should be improved.

Regarding the development of highly-functional routers, Ref. [46] mentions the aggregation of commodity switches is effective. The interconnected commodity switches can deliver the large amount of traffic and maintain the information to cooperate with each other. From the viewpoint of flexible deployment, Ref. [13] proposes a flexible architecture for overlay network and network virtualization environments. A programmable router can flexibility change its functionality and provide sophisticated functionality with computational, storage, and relay functions.

In addition, VPNs can be supposed as a kind of advanced overlay network enabled by advanced relay nodes, which nodes possess not only the simple relay function but also advanced relay functions with computational function by CPU resources and information maintenance function by storage resources. These relay nodes can efficiently handle VPN traffic which this dissertation focus on. For example, these relay nodes can maintain aggregated flow information with storage resource; they can calculate the allocated bandwidth to the large number of VPN sites with high performance computational resource.

4.2 Network resource efficiency in the provider networks

To achieve the MTA service, the hose bandwidth allocation method utilizes the feedback notification mechanism. In this mechanism, the edge routers exchange the link information by transmitting the control packets. As the number of edge routers increases, the massive number of control packets may consume the limited network resources. Therefore, we should consider efficient information exchange among all edge routers.

We show an example of a network resource consumption scenario caused by the feedback notification mechanism. Let the packet size of the control packet be 32 byte, the interval time of sending the control packet be 100ms, and the number of the edge routers be 100. In this case, because the number of path connection among edge routers is $100(100-1)/2 = 4950$, the required bandwidth for the control information is approximately 12.7 Mb/s. This consumption of the limited network resource is not small. Therefore, traffic reduction mechanisms are needed to prevent competition between data and control traffic.

To solve this problem, we should deeply consider the notification mechanism among the edge routers. Here, we focus on the characteristics of the egress edge routers. These routers periodically transmit the control packet, and these routers send the control packet to certain ingress routers. In other words, the intermediate nodes can predict the arrival timing, the arrival amount, and route of the transmitted control packets. In this situation, we consider that the network coding mechanisms [47] are adaptable to the hose bandwidth allocation method. This reduction of the network resource consumption can expand the available bandwidth, hence the network coding mechanisms can achieve the network resource efficiency to reduce the number of packets passing through the networks.

4.3 Broad deployment of the hose bandwidth allocation method

In this dissertation, the hose bandwidth allocation method covered a single VPN provider environment. The proposed method was proven to be effective for achieving the MTA service. For the broad deployment of the hose bandwidth allocation method, the proposed method should cover multiple VPNs interconnected each other. In this case, edge routers in a provider should cooperate with edge routers in other provider. Therefore, in the near future, the cooperation among multiple VPN providers will be needed. In particular, we should define interfaces between individual providers to cooperate with different providers. Moreover, we should construct resource management systems which can efficiently manage wide variety of resources, e.g., computational, storage, and network resources.

By contrast with the near future, the network virtualization technologies [13, 14, 8] are expected to the broad deployment of the hose bandwidth allocation method across multiple VPN providers in the future network. In this case, a huge common network resource, i.e., network resource of multiple providers, can be sliced to multiple virtualized network resource by the network virtualization technologies. If we utilize network virtualization technologies, we should consider how to virtualize wide variety of resources. This study will be next milestone for efficient network management.

4.4 Next milestone of the provisioning algorithm for the MTA service

This dissertation presented the fundamental study about the provisioning algorithm for offline QoS traffic control in the MTA service. Through the numerical demonstration, we showed the effectiveness of the provisioning algorithm. This work positions one of the primary research for adequate parameter determination algorithm. Even though the proposed method cannot be adaptable to practical environment immediately, this study showed the knowledge to achieve the MTA service.

Next milestone of the provisioning algorithm is the development of more practical, efficient and sophisticated algorithm based on this fundamental study. We should consider a novel scheme for reducing the computational complexity. Specifically, heuristic and approximate techniques will be expected.

4.5 Research about QoS in the future

In this dissertation, we tackled performance assurance issues for achieving the high quality communication environment, and we defined the performance assurance issues as QoS issues. These issues are one of most important issues in QoS research area. However, to continue a serious commitment to QoS research, we should clarify what QoS is. In other words, we should categorize the big QoS issue to multiple sub issues.

Even if providers offer a perfect performance assurance service to customers, the performance assurance service may be occasionally unable to provide performance assurance when a network element fails or some failures occur. Therefore, the availability of high quality communication environment should be one of most important issues in QoS research. To efficiently improve the availability, we should consider solutions from wide perspectives. For example, several approaches are useful for improving the availability: fault-tolerant routing, redundant network elements and equipments, and sufficient resource provision.

Data aggregation is also one of most important issues to efficiently manage networks and improve the performance. As the ability and functionality of relay nodes are high, fine-grained relay (i.e., packet relay) is not efficient. Therefore, we should other data unit for providing QoS, for example transport flow, object transfer data, single site level flow, and multiple site level (s.t., multiple VPNs) level. However, since rough-grained service may degrade the quality of communication, hierarchical traffic control mechanisms including fine-grained and rough-grained relay are needed.

To efficiently provide the performance assurance, service differentiation (classification) can be a important issue. If wide variety of traffic streams cannot be classified well, this misclassification degrades the performance. In this dissertation, we classify the traffic streams to customer and site levels for providing the MTA service in PPVPN environment. High-speed and accurate classification mechanisms are needed.

When the following issues for QoS are solved, providers can reliably offer the performance assurance service. Some metrics of the performance assurance are provided to customers: throughput, data loss, delay and jitter. In particular, data communications require the throughput and real-time communications require delay and jitter. In a future scenario, we should consider mechanisms which can provide the performance assurance in high-scale and wide variety of networks.

5. Conclusion

This dissertation addressed the issue in terms of the MTA service in PPVPNs. As described in Chapter 1, the selfish and greedy applications cause the congestion collapse if we do not develop strategy for the network resource efficiency. However, if providers control the traffic to obtain the benefit for only providers, the essential solution cannot be achieved. In PPVPNs, the issue on the MTA service is how to achieve two competing demands. While a VPN customer demands the minimum agreed throughput of the customer to a VPN provider, the provider has a demand on the network resource efficiency. To meet the both demands, we proposed two essential mechanisms from two different perspectives: the hose bandwidth allocation method for online QoS traffic control and the provisioning algorithm for offline QoS traffic control. The first mechanism is essential when the VPN provider allocates the bandwidth to the VPN customers. On the other hand, the second mechanism is also needed when the provider determines the target bandwidth exceeding the minimum agreed throughput. The combination of the first and second mechanisms satisfies the both demands.

5.1 Contributions

Contributions on this dissertation was that we cope with the issue for achieving the MTA service in the virtualized network environment from two perspectives. In this dissertation, we considered PPVPNs as a kind of sophisticated overlay networks. The research in this dissertation positions a milestone for the research in terms of the MTA service for future Internet.

Contributions on the first study was that we demonstrate effectiveness, scalability and stability of the hose bandwidth allocation method. In the proposed method, ingress, core and egress routers measure and control traffic with each other. Especially, we clarify that the measurement interval time of the ingress routers significantly impacts on the stability and scalability of the proposed method. In the second study, we showed how to determine the weight values, utilized to calculate the allocated bandwidth in feedback-driven traffic control mechanisms. The main contribution of this dissertation was that the studies lead to the realization of the MTA service in PPVPNs. Especially, this dissertation clarified how providers control and manage the traffic for the both providers and customers.

5.2 Future work

Although the studies in this dissertation can achieve the MTA service in PPVPNs, this dissertation raises future issues improving the efficiency on practical aspect. In the future research, many researchers should be going to study the future issues because these studies lead to achievement of efficient and reliable future Internet.

The first issue is the realization of the MTA service on a inter-VPNs environment. If we will attempt to interconnect multiple networks managed by different VPN providers, we will need to reduce the number of notification packets passing through multiple interconnected networks. If the overlay network and network virtualization technologies will be deployed broadly in the future, the MTA service will be able to adaptable on these novel technologies.

The second issue is the reduction of the computational complexity in the provisioning algorithm. To efficiently utilize the provisioning algorithm in a situation of information explosion, we should consider that heuristic and/or approximate techniques will be appropriate. Through this study, even in a situation where wide variety and massive number of applications and services compete with each other, we will achieve a essential social network infrastructure in the future Internet.

References

[1] "Information and communications in Japan 2008", White paper (Japanese edition), Ministry of Internal Affairs and Communications in Japan (2008).

[2] V. Jacobson: "Congestion avoidance and control", Proc. Symposium proceedings on Communications architectures and protocols (SIGCOMM '88), pp. 314–329 (1988).

[3] R. Sherwood, B. Bhattacharjee and R. Braud: "Misbehaving TCP receivers can cause internet-wide congestion collapse", Proc. the 12th ACM conference on Computer and communications security (CCS '05), pp. 383–392 (2005).

[4] N. J. Gunther: "The dynamics of performance collapse in large-scale networks and computers", Int'l Journal of High Performance Computing Applications, **14**, 4, pp. 367–372 (2000).

[5] C. Albuquerque, B. J. Vickers and T. Suda: "Network border patrol: preventing congestion collapse and promoting fairness in the Internet", IEEE/ACM Trans. Networking, **12**, 1, pp. 173–186 (2004).

[6] L. Qiu, Y. R. Yang, Y. Zhang and S. Shenker: "On selfish routing in Internet-like environments", IEEE/ACM Trans. Networking, **14**, 4, pp. 725–738 (2006).

[7] M. S. Blumenthal and D. D. Clark: "Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world", ACM Transactions on Internet Technology, **1**, 1, pp. 70–109 (2001).

[8] A. Greenberg, G. Hjalmtysson, D. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan and H. Zhang: "A clean slate 4D approach to network control and management", ACM SIGCOMM Computer Communication Review, **35**, 5, pp. 41–54 (2005).

[9] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu and A. Silberschatz: "P4P: Provider portal for applications", Proc. ACM SIGCOMM08 (2008).

[10] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang: "IDMaps: A global internet host distance estimation service", IEEE/ACM Trans. Networking, **9**, 5, pp. 525–540 (2001).

[11] D. Choffnes and F. Bustamante: "Taming the torrent: A practical approach to reducing cross-ISP traffic in P2P systems", Proc. ACM SIGCOMM08 (2008).

[12] A. Nakao, L. Peterson and A. Bavier: "A routing underlay for overlay networks", Proc. ACM SIGCOMM03 (2003).

[13] Y. Mao, B. T. Loo, Z. Ives and J. Smith: "MOSAIC: Unified declarative platform for dynamic overlay composition", Proc. 4th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2008) (2008).

[14] J. Turner, P. Crowley, J. Dehart, A. Freestone, B. Heller, F. Kuhms, S. Kumar, J. Lockwood, J. Lu, M. Wilson, C. Wiseman and D. Zar: "Supercharging PlanetLab - high performance, multi-application, overlay network platform", Proc. ACM SIGCOMM07 (2007).

[15] "Information and communications in Japan 2005", White paper (Japanese edition), Ministry of Internal Affairs and Communications in Japan (2005).

[16] S. H. Alonso, A. H. González, M. F. Veiga, R. F. R. Rubio and C. L. Garía: "Simulation study of aggregate flow control to improve QoS in a differentiated services network", Proc. 2nd Int'l Workshop on Quality of Service in Multiservice IP Networks (QoS-IP 2003), Vol. 2601 of Lecture Notes in Computer Science, pp. 729–741 (2003).

[17] X. He and H. Che: "Achieving end-to-end throughput guarantee for individual TCP flows in a differentiated services network", Proc. 1st Int'l Conference on Networking (ICN 2001), Vol. 2093 of Lecture Notes in Computer Science, pp. 368–377 (2001).

[18] G. Zhang and H. T. Mouftah: "End-to-end QoS guarantees over diffserv networks", Proc. 6th IEEE Symposium on Computers and Communications (ISCC 2001), pp. 302–309 (2001).

[19] X. Zeng, C. H. Lung and C. Huang: "A bandwidth-efficient scheduler for MPLS diffserv networks", Proc. IEEE Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS 2004), pp. 251–258 (2004).

[20] P. Iovanna, M. Settembre, R. Sabella, G. Conte and L. Valentini: "Performance analysis of a traffic engineering solution for multilayer networks based on the GMPLS paradigm", IEEE Journal on Selected Areas in Communications, **22**, 9, pp. 1731–1740 (2004).

[21] C. L. Lee, C. W. Chen and Y. C. Chen: "Weighted proportional fair rate allocations in a differentiated services network", IEICE Trans. Communications, **E85-B**, 1, pp. 116–128 (2002).

[22] T. Yokoyama, K. Iida, H. Koga and S. Yamaguchi: "Proposal for adaptive bandwidth allocation using one-way feedback control for MPLS networks", IEICE Trans. Communications, **E90-B**, 12, pp. 3530–3540 (2007).

[23] N. G. Duffield, P. Goyal, A. G. Greenberg, P. P. Mishra, K. K. Ramakrishnan and J. E. van der Merwe: "Resource management with hoses: point-to-cloud services for virtual private networks", IEEE/ACM Trans. Networking, **10**, 5, pp. 679–692 (2002).

[24] J. L. Kenningtona, E. V. Olinicka and G. Spiridеb: "Basic mathematical programming models for capacity allocation in mesh-based survivable networks", Omega, **35**, 6, pp. 629–644 (2007).

[25] A. Kumar, R. Rastogi, A. Silberschatz and B. Yener: "Algorithms for provisioning virtual private networks in the hose model", IEEE/ACM Trans. Networking, **10**, 4, pp. 565–578 (2002).

[26] M. Shimamura, K. Iida, H. Koga, Y. Kadobayashi and S. Yamaguchi: "Provisioning algorithm for minimum throughput assurance service in VPNs using nonlinear programming", Proc. IEEE Australasian Telecommunication Networks and Applications Conference (ATNAC 2007), pp. 311–316 (2007).

[27] G. Rosenbaum, W. Lau and S. Jha: "Recent directions in virtual private network solutions", Proc. 11th IEEE Int'l Conference on Networks (ICON 2003), pp. 217–223 (2003).

[28] Y. L. Liu, Y. S. Sun and M. C. Chen: "Traffic engineering for provisioning restorable hose-model VPNs", Proc. 2nd Int'l Conference on Next Generation Internet Design and Engineering, pp. 139–146 (2006).

[29] D. Wei and N. Ansari: "Implementing fair bandwidth allocation schemes in hose-modelled VPN", Proc. IEE Communications, **151**, 6, pp. 521–528 (2004).

[30] M. Shreedhar and G. Varghese: "Efficient fair queueing using deficit round robin", IEEE/ACM Trans. Networking, **4**, 3, pp. 375–385 (1996).

[31] L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow: "RSVP-TE: Extensions to RSVP for LSP tunnels", IETF RFC 3209 (2001).

[32] H. Byun, H. Woo, K. Kim and M. Lee: "A resource management mechanism for hose model based VPN QoS provisioning", Proc. IEEE Int'l Conference on Information Networking 2006, pp. 562–571 (2006).

[33] H. Byun and M. Lee: "Fair bandwidth sharing in hose based VPN", Proc. 1st Int'l Conference on Next Generation Network, pp. 57–59 (2006).

[34] S. Floyd and V. Jacobson: "Link-sharing and resource management models for packet networks", IEEE/ACM Trans. Networking, **3**, 4, pp. 365–386 (1995).

[35] I. Stoica, S. Shenker and H. Zhang: "Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks", IEEE/ACM Trans. Networking, **11**, 1, pp. 33–46 (2003).

[36] The VINT Project: "The network simulator – ns-2". http://www.isi.edu/nsnam/ns/.

[37] D. Katabi, M. Handley and C. Rohrs: "Congestion control for high bandwidth-delay product networks", Proc. ACM SIGCOMM '02, pp. 89–102 (2002).

[38] D. D. Clark and W. Fang: "Explicit allocation of best-effort packet delivery service", IEEE/ACM Trans. Networking, **6**, 4, pp. 362–373 (1998).

[39] L. Fàbrega, T. Jové, P. Vilà and J. Marzo: "A guaranteed minimum throughput service for TCP flows using measurement-based admission control", International Journal of Communication Systems, **20**, 1, pp. 43–63 (2007).

[40] J. S. Li and J. S. Li: "A novel core-stateless ABR-like congestion avoidance scheme in IP networks", International Journal of Communication Systems, **18**, 5, pp. 427–447 (2005).

[41] T. Yokoyama, K. Iida, H. Koga and S. Yamaguchi: "Proposal for adaptive bandwidth allocation using one-way feedback control for MPLS networks", IEICE Trans. Communications, **E90-B**, 12, pp. 3530–3540 (2007).

[42] M. Shimamura, K. Iida, H. Koga, Y. Kadobayashi and S. Yamaguchi: "Hose bandwidth allocation method to achieve minimum throughput assurance service for provider provisioned VPNs", IPSJ Journal, **49**, 12, pp. 3967–3984 (2008).

[43] Y. L. Liu, Y. S. Sun and M. C. Chen: "MTRA: An on-line hose-model VPN provisioning algorithm", Telecommunication Systems, **31**, 4, pp. 379–398 (2006).

[44] R. R. F. Liao and A. T. Campbell: "Dynamic core provisioning for quantitative differentiated services", IEEE/ACM Transactions on Networking, **12**, 3, pp. 429–442 (2004).

[45] T. Ishida, K. Ueda and T. Yakoh: "Fairness and utilization in multipath network flow optimization", Proc. IEEE International Conference on Industrial Informatics 2006, pp. 1096–1101 (2006).

[46] M. A. Fares, A. Loukissas and A. Vahdat: "A scalable, commodity data center network architecture", Proc. ACM SIGCOMM08 (2008).

[47] R. Ahlswede, N. Cai, S. Y. R. Li and R. W. Yeung: "Network information flow", IEEE Trans. Information Theory, **46**, 4, pp. 1204–1216 (2000).

Appendix

A. List of Publications

A.1 Journal

1-1. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Hose bandwidth allocation method to achieve minimum throughput assurance service for provider provisioned VPNs," *Journal of Information Processing*, Vol. 16, pp. 201–218 (*IPSJ Journal*, Vol. 49, No. 12, pp. 3967–3984), December 2008.

1-2. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Hose bandwidth allocation method to achieve minimum throughput assurance service for provider provisioned VPNs," submitted to Special Section on New Technologies and their Applications of the Internet in *IEICE Transactions on Information and Systems*, October 2009.

A.2 International conference

2-1. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Performance evaluation of hose bandwidth allocation method using feedback control and class-based queueing for VPNs," In *Proceedings of the IEEE International Conference on Pacific Rim 2005 (PacRim 2005)*, pp. 241–244, August 2005.

2-2. Mitsuaki Akiyama, Takanori Kawamoto, Masayoshi Shimamura, Teruaki Yokoyama, Youki Kadobayashi, and Suguru Yamaguchi, "A proposal of metrics for botnet detection based on its cooperative behavior," In *Proceedings of the SAINT 2007 Internet Measurement Technology and its Applications to Building Next Generation Internet Workshop*, January 2007.

2-3. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Provisioning algorithm for minimum throughput assurance service in VPNs using nonlinear programming," In *Proceedings* *of the IEEE Conference on Australasian Telecommunication Networks and Applications Conference (ATNAC 2007)*, pp. 311–316, December 2007.

A.3 Domestic technical report

3-1. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Performance evaluation of hose bandwidth allocation method using dynamic weight distribution algorithm for VPN," In *IEICE Technical Report*, Vol. 104, No. 415, IN2004-130, pp. 1–6, December 2004.

3-2. Atsushi Konami, Masayoshi Shimamura, and Katsuyoshi Iida, "A provisioning algorithm for VPN hose bandwidth allocation method," In *IEICE Technical Report*, Vol. 105, No. 627, NS2005-170, pp. 49–52, March 2006.

3-3. Mitsuaki Akiyama, Masayoshi Shimamura, Teruaki Yokoyama, Youki Kadobayashi, and Suguru Yamaguchi, "Proposal and implementation of DoS/DDoS simulator for evaluation of DoS/DDoS Countermeasures," In *Proceedings of the 2007 Symposium on Cryptography and Information Security*, January, 2007.

3-4. Masayoshi Shimamura, Katsuyoshi Iida, Hiroyuki Koga, Youki Kadobayashi, and Suguru Yamaguchi, "Provisioning algorithm for minimum throughput assurance service in VPNs using nonlinear programming," In *IEICE Technical Report*, Vol. 107, No. 249, IN2007-83, pp. 35–40, October 2007.

3-5. Masayoshi Shimamura, Takeshi Ikenaga, Hiroshi Koide, Dirceu Cavendish, Masato Tsuru, "Adaptive network services enabled by advanced relay node," In *IEICE Technical Report*, Vol. 108, No. 203, NS2008-66, pp. 135–138, September 2008.

3-6. Masayoshi Shimamura, Takeshi Ikenaga, Masato Tsuru, "Packet compression scheme in advanced relay node," In *Proceedings of the 61st Joint Conference of Electrical and Electronics Engineers in Kyushu*, September 2008.

3-7. Masayoshi Shimamura, Takeshi Ikenaga, Masato Tsuru, "Performance analysis of adaptive packet compression scheme in advanced relay nodes," In

IEICE Technical Report, Vol. 108, No. 392, NS2008-142, pp. 95–100, January 2009.