

Hose Bandwidth Allocation Method to Achieve a Minimum Throughput Assurance Service for Provider Provisioned VPNs

MASAYOSHI SHIMAMURA,^{†1,*1} KATSUYOSHI IIDA,^{†2}
HIROYUKI KOGA,^{†3} YOUKI KADOBAYASHI^{†1}
and SUGURU YAMAGUCHI^{†1}

We propose a hose bandwidth allocation method to achieve a minimum throughput assurance (MTA) service for the hose model. Although the hose model, which has been proposed as a novel VPN service model for provider provisioned virtual private networks (PPVPNs), has been proven to be effective for network resource efficiency and configuration complexity, there has been no consideration of a mechanism to assure quality of service (QoS) in the hose model. The basic idea of our method is to gather available bandwidth information from inside a network and use it to divide the available bandwidth into hoses on each bottleneck link. We evaluate and clarify our method through computer simulation runs. The simulation results show that our method can achieve an MTA service in the hose model.

1. Introduction

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*¹⁾ 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^{1),2)}. 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} .

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

^{†1} Graduate School of Information Science, Nara Institute of Science and Technology
^{†2} Global Scientific Information and Computing Center, Tokyo Institute of Technology
^{†3} Department of Information and Media Sciences, University of Kitakyushu
^{*1} Presently with Network Design Research Center, Kyushu Institute of Technology

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

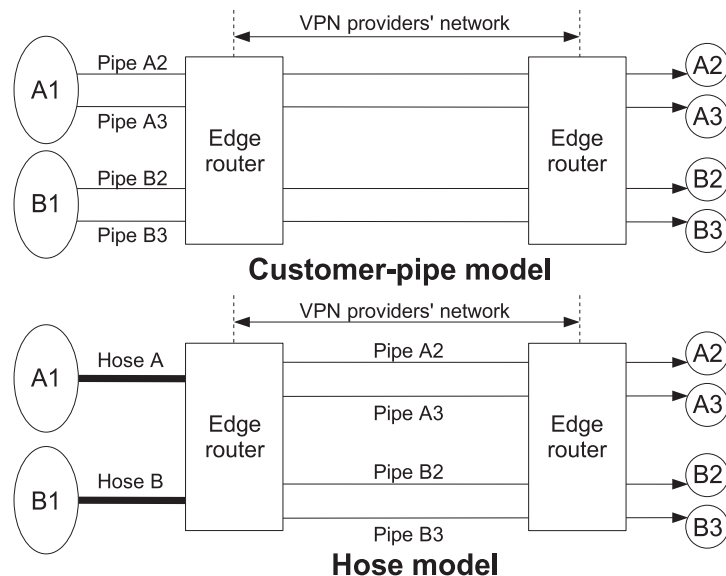


Fig. 1 Illustration of customer-pipe and hose models.

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	

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 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 proposed³⁾, and the basic idea of this algorithm can be applied to the hose model with slight modifications. In this paper, 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.

In this paper, 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, our method uses a feedback-driven traffic control mechanism. For our 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 pipe⁴⁾. Our 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 the paper is organized as follows. Section 2 reviews related work and describes the shortcomings of an MTA service in the hose model. Section 3 describes our hose bandwidth allocation method and Section 4 presents computer simulations that demonstrate that our method meets the three requirements. Section 5 concludes by briefly summarizing the main points and mentioning future work.

2. Related Work

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 2.1 and introduce three related studies and their shortcomings in terms of MTA services in the hose model in Section 2.2.

2.1 Related Work on Hose Model

First, we describe the research area of the hose model. The model was proposed by Duffield, et al.¹⁾ 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 proposed²⁾. 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 model⁵⁾. 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 Related Work on 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)⁶⁾, which is based on the deficit round robin (DRR)⁷⁾. 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)⁸⁾ has been proposed^{9),10)}. 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

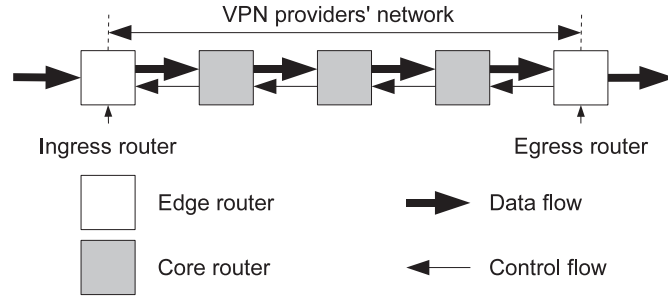


Fig. 2 Weighted proportional fair rate allocation.

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, although it achieves high utilization with the customer-pipe model.

To achieve high utilization and fair allocation, the weighted proportional fair rate allocation (WPFRA) method¹¹⁾ has been proposed. Here, we explain a one-way version of the WPFRA method¹²⁾, 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

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

where explicit rate ($\text{ER}_{(i,e)}$) 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 re-

sponsiveness.

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

$$ER_{(i,e)} \leftarrow \min(ER_{(i,e)}, r_f).$$

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,

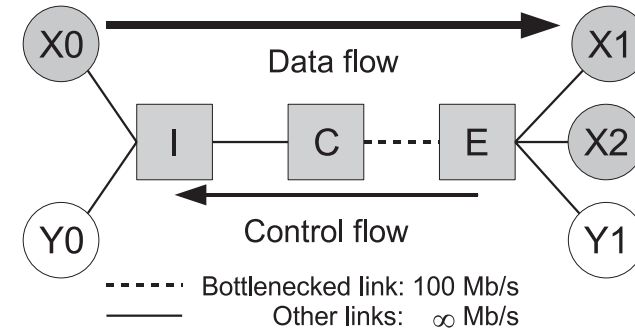


Fig. 3 Example of bandwidth allocation and calculation process: Network topology.

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

Time	WPFRA method								Our 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 our method: X : Y = 1 : 1

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

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 $r_{X1} = 100$ Mb/s). At time 2, the available bandwidth is divided into half for each site X1 and Y1 ($r_{X1} = r_{Y1} = 50$ Mb/s). At time 3, the available bandwidths of the sites are equally redistributed ($r_{X1} = r_{X2} = r_{Y1} = 33.3$ 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.

3. Hose Bandwidth Allocation Method

An MTA service will provide better throughput predictability in PPVPNs, as described in Section 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 our traffic control mechanism. Finally, we present the features of our method through comparison with the previous related methods.

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 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 our 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)

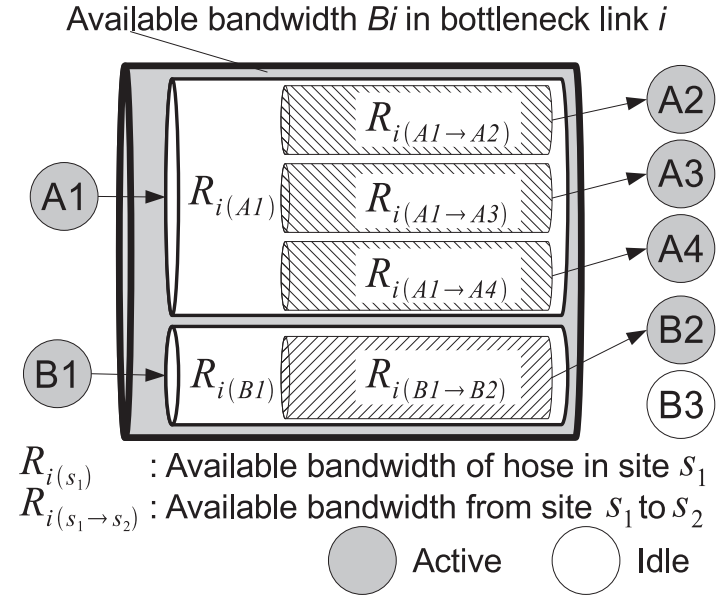


Fig. 4 MTA service in the hose model.

are active. $R_{i(s_1)}$ represents the available bandwidth of the hose of source site s_1 in the bottleneck link i ; $R_{i(s_1 \rightarrow s_2)}$ 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 $R_{i(B1)}$. Therefore, active hoses always provide link utilization of nearly 100%.

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

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

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

$$R_{i(A1 \rightarrow Ax)} = \frac{R_{i(A1)}}{N_{i(A1)}}, \quad R_{i(B1 \rightarrow Bx)} = \frac{R_{i(B1)}}{N_{i(B1)}},$$

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 our service model in the hose model, we consider the mechanism of our hose bandwidth allocation method in the next section.

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 our method, we use a one-way version of the WPFRA method¹²⁾ and class-based queueing (CBQ)¹³⁾ 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`

```

/*****
 * 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][h][p] == 1 )
        r_hp[l][h][p] = r_h[l][h] / s_hp[l][h][p];

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

```

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

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 Eqs. (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

$$r_{l(h)} \Leftarrow \min (ER_{(i,e)} \times w_h, r_{l(h)} + IB \times w_h), \quad (1)$$

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

$$r_{l(hp)} = \begin{cases} \frac{r_{l(h)}}{n} & (x_{l(hp)} = 1) \\ \sum_{j=1}^n x_{l(hj)} & \\ 0 & (x_{l(hp)} = 0), \end{cases} \quad (2)$$

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 our 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 our method, whereas the start of any traffic does affect the system variables in the WPFRA method. Therefore, our method can divide the available bandwidth among subscribers, while WPFRA 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,

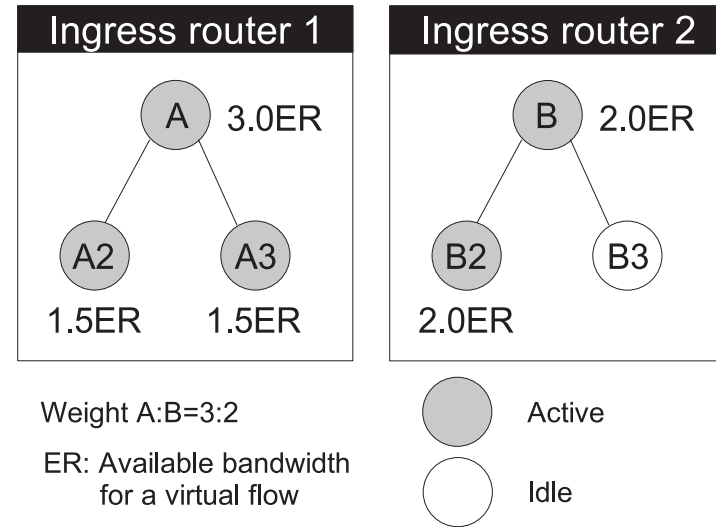


Fig. 6 Bandwidth allocation at ingress routers.

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 our 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

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

Method	Model	Information gathering	Queuing method	Fairness			High utilization
				subscribers	sites	flows	
2-D DRR ⁽⁶⁾	Hose	N	Modified DRR	N	Y	Y	N
Two-level WFQ ^(9),10)	Hose	N	Modified WFQ	Y ‡	Y	N	N
WPFRA ^(11),12)	Pipe	Y	Optional	N	Y	N	Y (bottleneck link)
Our method	Hose	Y	CBQ (optional)	Y	Y	N	Y (bottleneck link)

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

bandwidth allocated to subscriber B because site B3 is idle.

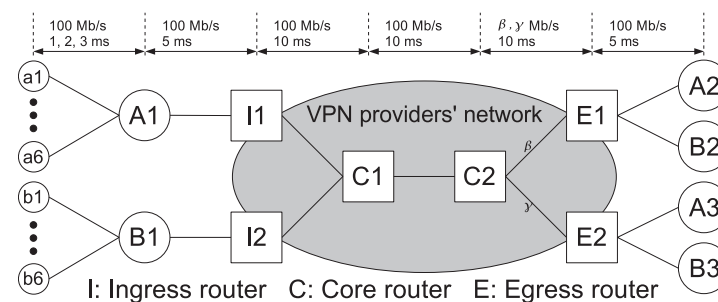
3.3 Qualitative Comparisons

The benefit of our 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 our method and those in the related studies described in Section 2. Our method is the only one that satisfies all of the above three requirements.

As described in Reference⁽¹⁴⁾, 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.

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 our hose bandwidth allocation method. We first confirm the basic behavior of our 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

**Fig. 7** Simulation topology.

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 our method. On the basis of these computer simulation runs, we clarify the characteristics of our method and indicate how to decide the system parameters for our method.

4.1 Simulation Model

The computer simulator we used is the ns-2.29 simulator⁽¹⁵⁾. Since the original ns-2 does not contain our 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 our method has the designated hose behavior described in Table 3 in Section 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.

Table 4 Output buffer length of each router.

Router		Buffer length
Right direction	A1, B1	1200
	I1, I2	600
	C1, C2	400
	Others	∞
Left direction	Others	∞

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 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 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 our 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 our method are given in **Table 5**. To evaluate the basic behavior of our 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 our method, the responsibility of ingress routers is significant because we implement two additional mechanisms at ingress routers as described in Section 3.2. We explain why the interval time of ingress routers is longer than those of other routers in Section 4.5.

Table 5 Parameter setting.

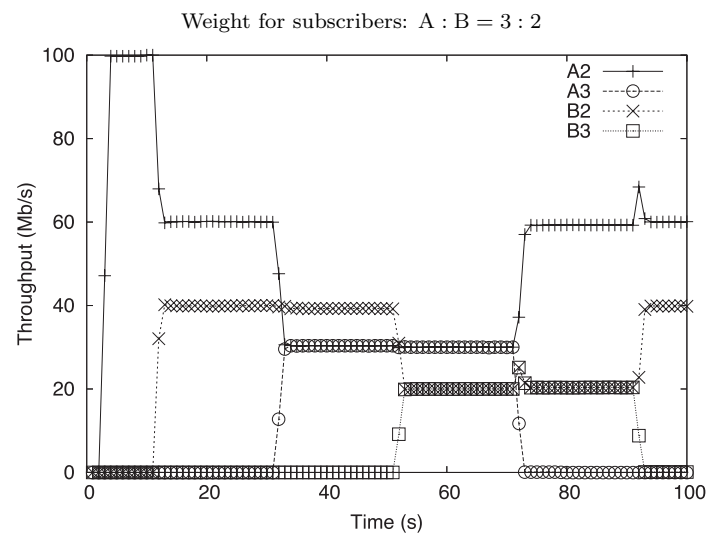
Parameter	Value
α	0.9
IB	∞
Interval time of ingress routers	500 ms
Interval time of core routers	100 ms
Interval time of egress routers	100 ms

4.2 Comparison with WPFRA Method

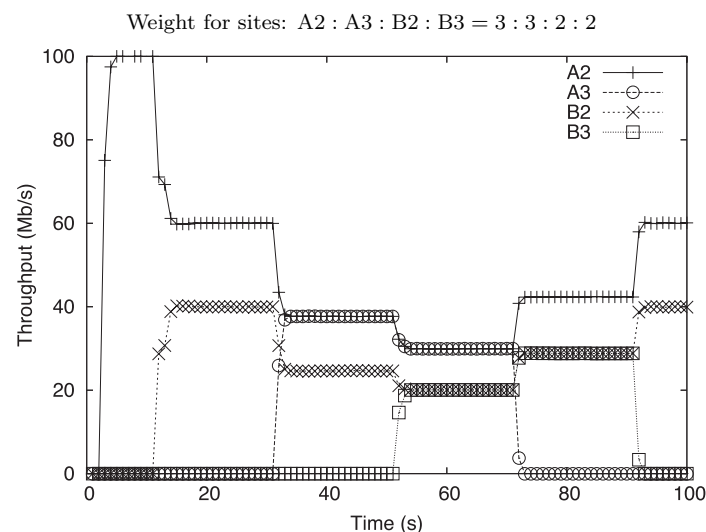
In this section, we compare the behavior of the WPFRA method and our method. For this simulation scenario, we demonstrate that our algorithm can allocate bandwidth for each active subscriber when traffic is inserted and removed. To simplify the network topology, we set β and γ to 100 Mb/s in Fig. 7. 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 our 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, our 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 our method can use the hose model because our method divides the bandwidth at both the subscriber and site levels.



(a) Our method



(b) WPFRA method

Fig. 8 Comparison of throughput dynamics between the WPFRA method and our method.**Table 6** Bandwidth allocation error rate and utilization.

Unit: Mb/s		
	Our 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%

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 1s, and the simulation ends at 100s. 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 100s. Regarding the utilization, our 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 our 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%.

The results demonstrate that our 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 3.3 can be achieved.

4.3 Impact of Multiple Bottleneck Links in a Hose

In the previous scenario, the bottleneck link was a single link between core

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		

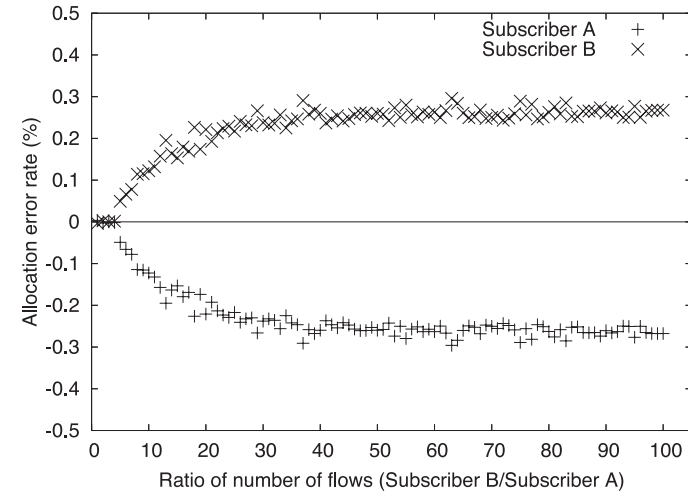
routers C1 and C2, although traffic toward sites A2, A3, B2 and B3 passed through different paths I1 to E1, I1 to E2, I2 to E1, and I2 to E2, respectively. Next, we evaluate our method in a network topology with multiple bottleneck links in a hose. In the hose model, each hose accommodates multiple pipes, 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 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 4.2, except for the link capacity.

The simulation results are summarized in **Table 7**. Our 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, our 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.

4.4 Scalability

The above results showed what our hose bandwidth allocation method is capable of doing. We now evaluate our method in terms of the scalability and stability

**Fig. 9** Impact of imbalance ratio of TCP flows on error rates.

which are important performance metrics. First, we focus on the scalability where parameters of two subscribers are unbalanced because of the following reasons. In our 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. 9**. 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.

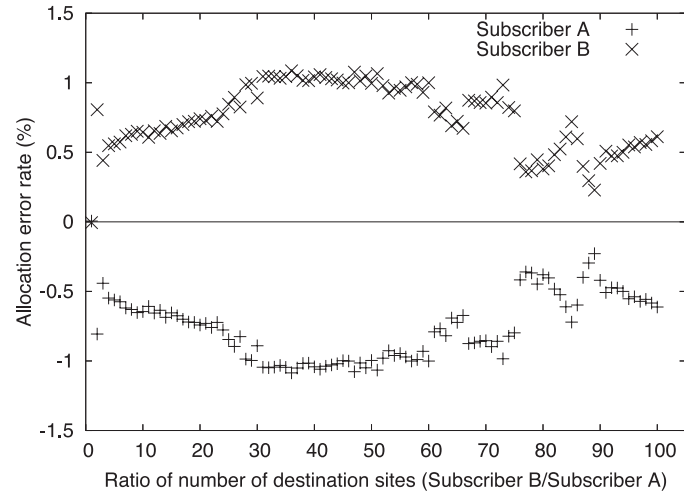


Fig. 10 Impact of imbalance ratio of destination sites on error rates.

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 our 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. 10**). 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. 10 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 our method is robust with respect to the number of subscribers.

The results of our 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 our hose bandwidth allocation method can keep the low allocation error rate in a large-scale VPN.

4.5 Stability Analysis

The above simulations showed the effectiveness of our 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 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 our 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. 11**. 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, respectively. Here, we focus on subscriber B. The correct measured value for

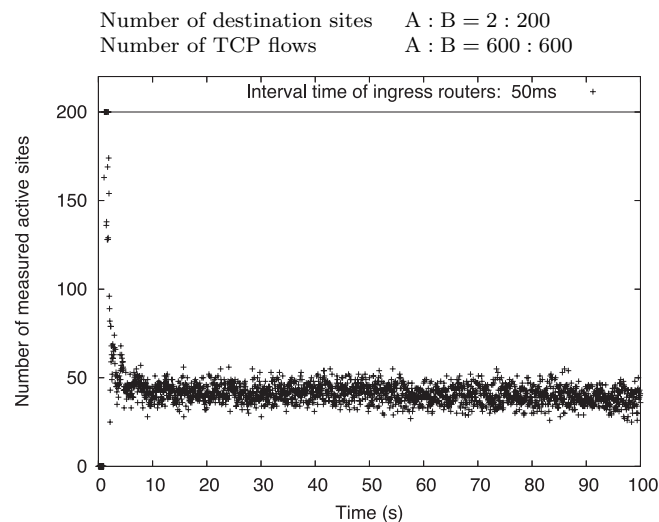


Fig. 11 Error in number of active sites measured at ingress routers.

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. 12** 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. 13**. The link delays might be another significant factor in the erroneous measurement in addition to the measurement interval time because our method

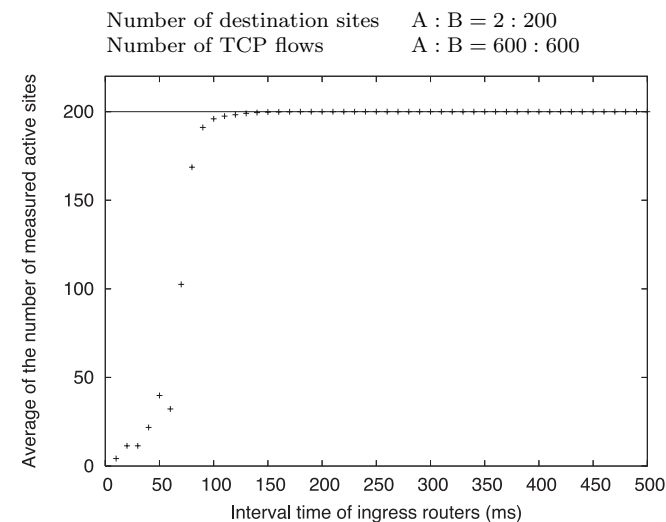


Fig. 12 Impact of measurement time interval at ingress routers on measurement error.

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 our method.

The above three simulation results show the impact of inaccurate measurement on short measurement time intervals. To stabilize our 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. 14**. The x-axis represents the propagation delay between core routers C1 and C2, and

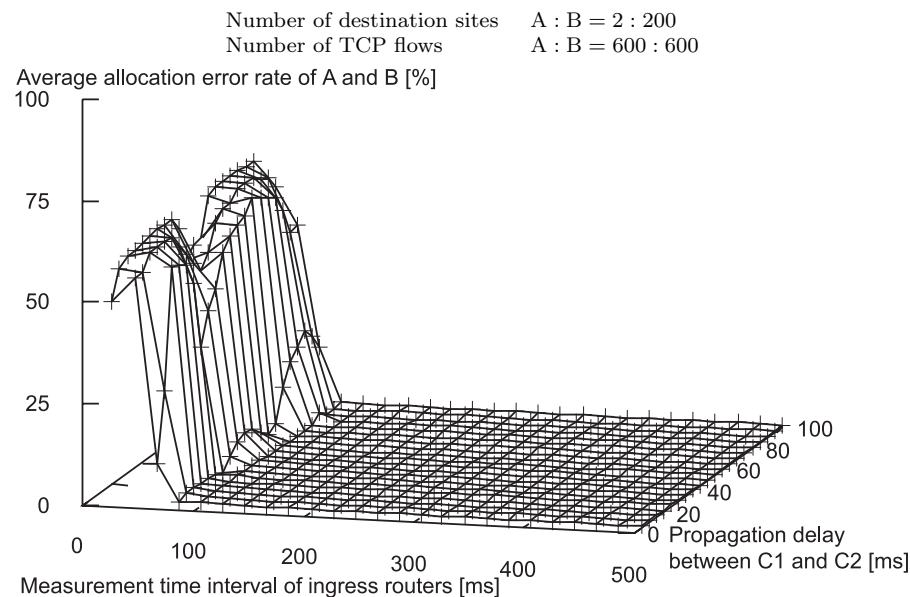


Fig. 13 Impact of measurement time interval in ingress routers on allocation error rate.

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 our method and the link delays is low.

From the above simulation runs, we clarified that the measurement time interval of ingress routers is a significant factor for the stability of our 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 our hose bandwidth allocation method

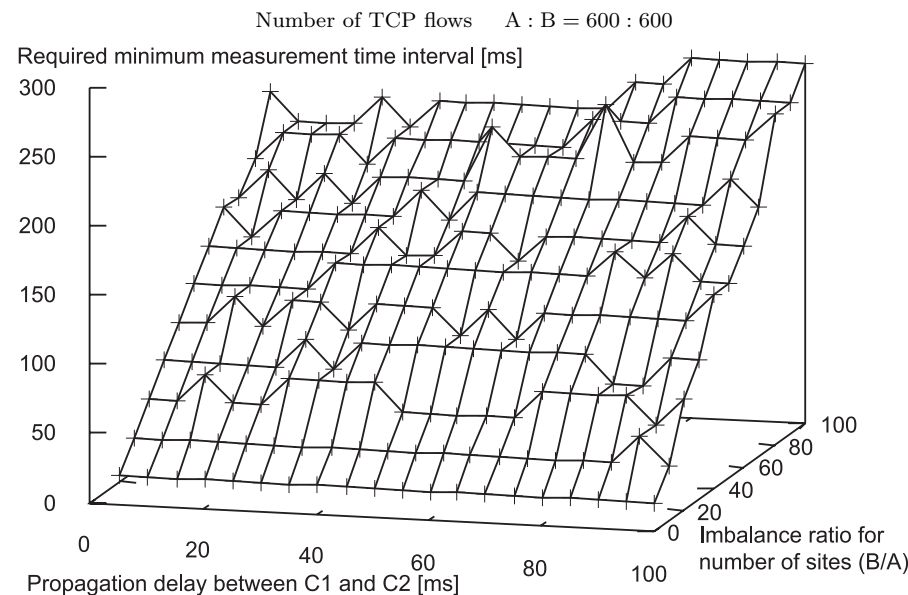


Fig. 14 Required minimum measurement time interval in ingress routers for precise measurement.

such as the explicit control protocol (XCP)¹⁶⁾. Although it is beyond the scope of our research, the stability of such feedback systems might be achieved by plotting their open-loop transfer function on a Nyquist plot.

5. Conclusion

In this paper, 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. Our method provides at least the minimum agreed throughput to active hoses even during a period of congestion. Our basic idea is that our method gathers available bandwidth information from inside a network and divides the available bandwidth into hoses on each bottleneck link using the information. Our 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 our method's advantages and to analyze its scalability and stability. First, we determined the basic behavior of our method in throughput dynamics. Our 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 our 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 our 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 our method, we examined the impact of the ratio of the scale of subscribers on the allocation error rates. We proved that our method kept the allocation error rate low even when the scale of the VPN increased. Finally, to clarify the stability of our 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. On the other hand, the link delay was not a significant factor for the stability of our method.

In summary, we showed that our method satisfied both the requirements for achieving an MTA service in the hose model, clarified the scalability of our method, and indicated how to decide its system parameters. In future, we will apply our method to an inter-VPN environment.

Acknowledgments We thank Assistant Professor Takeshi Okuda and Assistant Professor Shigeru Kashiwara for their insightful advice. This work is supported in part by the 21st Century COE Program "Ubiquitous Networked Media Computing," Graduate School of Information Science, Nara Institute of Science and Technology, Japan, and the Ministry of Education, Science, Sports and Culture, Japan, Grant-in-Aid for Young Scientists, 18700054, 2006.

References

- 1) Duffield, N.G., Goyal, P., Greenberg, A.G., Mishra, P.P., Ramakrishnan, K.K. and vander Merwe, J.E.: Resource management with hoses: Point-to-cloud services for virtual private networks, *IEEE/ACM Trans. Networking*, Vol.10, No.5, pp.679–692 (2002).
- 2) Kumar, A., Rastogi, R., Silberschatz, A. and Yener, B.: Algorithms for provisioning virtual private networks in the hose model, *IEEE/ACM Trans. Networking*, Vol.10, No.4, pp.565–578 (2002).
- 3) Shimamura, M., Iida, K., Koga, H., Kadobayashi, Y. and Yamaguchi, S.: 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).
- 4) Rosenbaum, G., Lau, W. and Jha, S.: Recent Directions in Virtual Private Network Solutions, *Proc. 11th IEEE Int'l Conference on Networks (ICON 2003)*, pp.217–223 (2003).
- 5) Liu, Y.L., Sun, Y.S. and Chen, M.C.: Traffic engineering for provisioning restorable hose-model VPNs, *Proc. 2nd Int'l Conference on Next Generation Internet Design and Engineering*, pp.139–146 (2006).
- 6) Wei, D. and Ansari, N.: Implementing fair bandwidth allocation schemes in hose-modelled VPN, *Proc. IEE Communications*, Vol.151, No.6, pp.521–528 (2004).
- 7) Shreedhar, M. and Varghese, G.: Efficient Fair Queueing Using Deficit Round Robin, *IEEE/ACM Trans. Networking*, Vol.4, No.3, pp.375–385 (1996).
- 8) Berger, L., Gan, D., Li, T., Srinivasan, V. and Swallow, G.: RSVP-TE: Extensions to RSVP for LSP Tunnels, IETF RFC 3209 (2001).
- 9) Byun, H., Woo, H., Kim, K. and Lee, M.: A Resource Management Mechanism for Hose Model based VPN QoS Provisioning, *Proc. IEEE Int'l Conference on Information Networking 2006*, pp.562–571 (2006).
- 10) Byun, H. and Lee, M.: Fair Bandwidth Sharing in Hose Based VPN, *Proc. 1st Int'l Conference on Next Generation Network*, pp.57–59 (2006).
- 11) Lee, C.L., Chen, C.W. and Chen, Y.C.: Weighted Proportional Fair Rate Allocations in a Differentiated Services Network, *IEICE Trans. Commun.*, Vol.E85-B, No.1, pp.116–128 (2002).
- 12) Yokoyama, T., Iida, K., Koga, H. and Yamaguchi, S.: Proposal for Adaptive Bandwidth Allocation Using One-Way Feedback Control for MPLS Networks, *IEICE Trans. Commun.*, Vol.E90-B, No.12, pp.3530–3540 (2007).
- 13) Floyd, S. and Jacobson, V.: Link-sharing and Resource Management Models for Packet Networks, *IEEE/ACM Trans. Networking*, Vol.3, No.4, pp.365–386 (1995).
- 14) Stoica, I., Shenker, S. and Zhang, H.: Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high-speed networks, *IEEE/ACM Trans. Networking*, Vol.11, No.1, pp.33–46 (2003).
- 15) The VINT Project: The Network Simulator – ns-2. <http://www.isi.edu/nsnam/ns/>
- 16) Katabi, D., Handley, M. and Rohrs, C.: Congestion control for high bandwidth-delay product networks, *Proc. ACM SIGCOMM '02*, pp.89–102 (2002).

(Received December 28, 2007)

(Accepted September 10, 2008)

(Released December 10, 2008)



Masayoshi Shimamura received the B.E. and M.E. degrees from the Faculty of Engineering of Chiba Institute of Technology, Chiba, Japan, in 2003 and from the Graduate School of Information Science of Nara Institute of Science and Technology (NAIST), Nara, Japan, in 2005, respectively. From April 2005 to March 2007, he was an encouragement researcher in the 21st Century COE Program “Ubiquitous Networked Media Computing” in the Graduate School of Information Science, NAIST. Currently, he is a researcher at the Network Design Research Center of Kyushu Institute of Technology (KIT). His research interests include performance evaluation of networking systems. He is a member of IEICE.



Katsuyoshi Iida received the B.E. degree in computer science and systems engineering from Kyushu Institute of Technology (KIT), Iizuka, Japan, in 1996, the M.E. degree in information science from the Nara Institute of Science and Technology (NAIST), Ikoma, Japan, in 1998, and the Ph.D. degree in computer science and systems engineering from KIT in 2001. From October 2000, he was an assistant professor in the Graduate School of Information Science, NAIST. Since December 2004, he has been an associate professor in the Global Scientific Information and Computing Center, Tokyo Institute of Technology, Tokyo, Japan. His research interests include performance evaluation of networking systems, mobile networks, and Internet telephony. He is a member of the WIDE project, IEICE, and IEEE. In 2003, he received the 18th TELECOM System Technology Award from the Telecommunications Advancement Foundation, Japan.



Hiroyuki Koga received the B.E., M.E. and Ph.D. degrees in computer science and electronics from Kyushu Institute of Technology, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2004 he was a postdoctoral researcher in the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. From August 2004 to 2006 he was a researcher in the Kitakyushu JGN2 Research Center, National Institute of Information and Communications Technology, Japan. Since April 2006, he has been an assistant professor in the Department of Information and Media Engineering, Faculty of Environmental Engineering, University of Kitakyushu, Japan. His research interests include performance evaluation of computer networks, mobile networks, and communication protocols. He is a member of the IEEE and IEICE.



Youki Kadobayashi received the B.E., M.S., and Ph.D. degrees in computer science from Osaka University in 1992, 1994, and 1997, respectively. He is currently an Associate Professor in the Graduate School of Information Science, NAIST. His research interests include overlay networks and network security architecture.



Suguru Yamaguchi received the M.E. and D.E. degrees in computer science from Osaka University in 1988 and 1991, respectively. From 1990 to 1992, he was an assistant professor in the Education Center for Information Processing, Osaka University. From 1992 to 1993, he was with the Information Technology Center, Nara Institute of Science and Technology (NAIST), Nara, Japan, as an associate professor. From 1993 to 2000, he was with the Graduate School of Information Science, NAIST, Nara, Japan, as an associate professor. Currently, he is a professor at the Graduate School of Information Science, Nara Institute of Science and Technology. He has also been a member of the WIDE Project since its inception in 1988, where he has been conducting research on network security systems for wide area distributed computing environments. His research interests include technologies for information sharing, multimedia communication over high-speed communication channels, network security, and network management for the Internet.
