Scalable IP-VPN Flow Control Mechanism Supporting Arbitrary Fairness Criteria — Part 1: Architecture Design —

Osamu Honda, Hiroyuki Ohsaki, Makoto Imase
Graduate School of Information Science and Technology
Osaka University, Japan
E-mail: {o-honda,oosaki,imase}@ist.osaka-u.ac.jp

Junichi Murayama, Kazuhiro Matsuda
NTT Information Sharing Platform Laboratories
NTT Corporation, Japan
Email: {murayama.junichi,matsuda.kazuhiro}@lab.ntt.co.jp

***Abstract*— In recent years, IP-based virtual private networks (IP-VPNs), which provide a virtual privately owned network over an IP network, have attracted attention. With existing IP-VPNs, however, there is a serious problem that fairness among IP-VPN customers is not satisfied. In this paper, we first discuss design objectives of a control mechanism for achieving fair IP-VPN services: achieving inter-VPN fairness, achieving intra-VPN fairness, easy deployment into existing IP networks, and achieving a high scalability. We then propose an IP-VPN fairness control called *I2FVC (Inter- and Intra-VPN Fairness Control)* for realizing a fair IP-VPN service in a scalable way. The core of I2VFC is an AIMD (Additive Increase and Multiplicative Decrease) window flow control operating among IP-VPN service provider's edge routers. I2VFC has the advantage that an IP-VPN service provider can arbitrarly specify inter-VPN fairness criteria by utilizing analytic results of AIMD window flow control. Moreover, I2VFC can be easily deployed into existing IP networks by simply modifying edge routers. Through several simulation experiments, we demonstrate that I2VFC realizes both inter-VPN fairness and intra-VPN fairness with extremely high accuracy.**

I. Introduction

IP-based virtual private networks (IP-VPNs) [1-3], which provide a virtual privately owned network over an IP network, have attracted attention. A virtual private network can be constructed on an IP network at a lower cost than with conventional dedicated lines.

However, there is a serious problem that existing IP-VPNs cannot guarantee fairness among IP-VPN customers. This is because the IP network is a best-effort network, so the IP-VPN constructed on it is also a best-effort network. In reality, however, IP-VPN service providers have ardently requested the provision of fair IP-VPN services [4]. In recent years, a variety of research has been conducted regarding traffic engineering techniques for IP networks. However, existing traffic engineering techniques are still inadequate for achieving fair IP-VPN services at a reasonable cost [2].

Our research focuses on a L3-PPVPN (Layer 3 Provider-Provisioned VPN) framework [5], which is a framework where the service provider provides layer-3 VPN service to customers. The main objective of our work is to achieve fair IP-VPN services within an L3-PPVPN framework by a method that can be easily deployed in the service provider's IP network.

This paper first discusses design objectives of a control for achieving fair IP-VPN services. Specifically, the following four design objectives and their necessities are discussed.

1) Achieving inter-VPN fairness
2) Achieving intra-VPN fairness
3) Easy deployment in a service provider's IP network
4) Achieving a high scalability for transfer rate/number of VPNs

This paper then proposes I2VFC (Inter- and Intra-VPN Fairness Control) to achieve fair IP-VPN services. I2VFC is an AIMD (Additive Increase and Multiplicative Decrease) window flow control [6] that operates between IP-VPN service provider's edge routers (i.e., PE routers) [2]. Specifically, multiple flows (i.e., streams of packets with identical protocol type, source/destination address, etc.) accommodated in a VPN are aggregated into a single flow at the ingress PE router, and AIMD window flow control is performed for each aggregated flow between ingress and egress PE routers.

The structure of this paper is as follows. First, Section II presents related work on IP-VPN fairness control mechanisms. Section III discusses design objectives for IP-VPN fairness control. Section IV describes an architectural overview of I2VFC, followed by explanation of the I2VFC operation algorithm. Section V presents several simulation results, demonstrating effectiveness of I2VFC. Finally, Section VI concludes this paper. Note that effectiveness of our I2VFC is extensively investigated in [7] through simulation experiments and prototype system experiments.

II. Related work

Several papers [8, 9] have proposed methods of achieving fair IP-VPN services though modifying a queue management mechanism of DiffServ routers. However, the methods proposed in [8, 9] require that all routers in the network be replaced with DiffServ routers having a specific queue management mechanism, so these methods cannot be easily deployed into a service provider's IP network.

On the contrary, methods of achieving IP-VPN fairness using an AIMD window flow control mechanism between routers have been proposed in [10, 11]. These approaches achieve fairness among IP-VPNs by aggregating multiple flows accommodated in a VPN into a single flow and by performing an AIMD window flow control for those aggregated flows. However, methods proposed in [10, 11] require that a specific active queue management mechanism be implemented at all core routers in the network.

III. Design Objectives

(1) Achieving Inter-VPN Fairness

The first design objective is to achieve inter-VPN fairness: i.e., achieving fairness among customers contracting for VPN services. In existing IP-VPNs, since the underlying IP network

is a best-effort network, inter-VPN fairness cannot be satisfied. However, IP-VPN services should use bottleneck link bandwidth fairly among VPNs even if a given VPN generates heavy traffic volume; i.e., it should not unduly restrict throughput of other VPNs. Namely, it is desired that each VPN flow (i.e., aggregated flows in a VPN) can utilize some ratio of the bottleneck link bandwidth, and the ratio can be specified by the IP-VPN service provider.

Given that providers are the ones providing for IP-VPN services, criteria for inter-VPN fairness should be freely specified by the IP-VPN's service provider. For example, instead of equally distributing bandwidth to multiple VPN flows sharing the same bottleneck link, distributing bandwidth according to some factors such as the VPN's geographic location (distance between sites, number of hops, etc.) and the contracted line rate might be appropriate. Appropriate fairness criteria are dependent on the policy of the IP-VPN service providers, so that IP-VPN fairness control is required to support arbitrary fairness criteria.

Specifically, if the throughput of VPN flow i ($1 \leq i \leq N$) is T_i and the weight of the throughput of VPN flow i as specified by the IP-VPN service provider is r_i, then achieving

$$\frac{T_i}{r_i} = \frac{T_j}{r_j} \quad (1)$$

with respect to all i, j ($i \neq j$) is required.

In addition, the timescale in which inter-VPN fairness is achieved is also important. Functions required to IP-VPN fairness control differ vastly depending on whether fairness with a fine granularity (e.g., at the order of milliseconds) or a rough granularity (e.g., at the order of seconds or minutes) is required.

In reality, inter-VPN fairness should be achieved on a timescale on the order of approximately 100 times the round-trip time because of the following two reasons: (1) currently, most of the traffic transmitted on an IP-VPN is data traffic, so achieving fairness on a timescale on the order of approximately 100 times the round-trip time would be adequate, and (2) achieving fairness on a timescale on the order of the round-trip time is not possible by just modifying the provider edge routers; i.e., core routers must be modified.

(2) Achieving Intra-VPN Fairness

The second design objective is to achieve fairness among users accommodated in the same VPN, i.e., intra-VPN fairness. IP-VPN services should not allow unfairness among users accommodated in the same VPN even if inter-VPN fairness is achieved so that fairness among VPN customers should be achieved. This paper defines intra-VPN fairness as being achieved when the ratios of throughput of flows accommodated in the same VPN are equal.

Given the characteristics of IP-VPN services, however, requirements for intra-VPN fairness are relatively lax compared to requirements for inter-VPN fairness. In general, how bandwidth should be distributed to users accommodated in the same VPN should be decided by the customer contracting the IP-VPN service, and not by the IP-VPN service provider. Thus, IP-VPN fairness control would be sufficient if it does not unduly restrict throughput of certain flows.

(3) Easy Deployment in a Service Provider's IP Network

The third design objective is to be able for an IP-VPN fairness control to be easily deployed into an existing IP network. One of reasons that IP-VPNs have been widely deployed these days is that an existing IP network infrastructure can be used as–is. Hence, fair IP-VPN services should be realized by modifying the existing IP-VPN framework as least as possible.

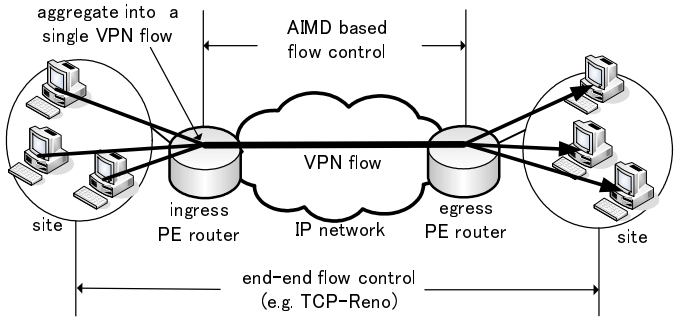

Fig. 1. Overview of I2VFC (Inter- and Intra-VPN Fairness Control)

Specifically, achieving IP-VPN fairness control that needs modifications only to PE routers, which are owned by the IP-VPN service provider, is strongly required [2, 5, 12]. Moreover, customer edge (CE) routers are managed by customers contracting the VPN service, so modification to CE routers by the IP-VPN service provider is virtually impossible. Thus, IP-VPN fairness control must be realized by simply changing PE routers of the IP-VPN service provider.

(4) Achieving a High Scalability for Transfer Rate/Number of VPNs

The fourth design objective is to achieve a high scalability for the transfer rate of VPN flows and the number of VPNs accommodated. In recent years, shift to faster networks has progressed rapidly, so throughput on the order of Gbit/s should be achieved for each VPN. Current customers of IP-VPN services are generally some organizational units like companies and groups, so the number of VPNs managed by an IP-VPN service provider has been rather low. In the future, however, VPN customers would be an individual user, so that the number of VPNs managed by the IP-VPN service provider will explode. Hence, it is important that IP-VPN fairness control has a high scalability also for the number of VPNs.

IV. Architecture and Algorithm

A. I2VFC Overview

An overview of the proposed I2VFC is shown in Fig. 1. The core of I2VFC is an AIMD window flow control that operates on IP-VPN service provider's PE routers.

Specifically, multiple flows accommodated in the same VPN are aggregated into a single VPN flow and stored in a logical queue for each VPN at ingress PE routers. PE routers distinguish VPN flows from source and destination IP address of a packet. Then, the round-trip time and packet loss rate of the network are periodically measured by exchanging management packets for each VPN between ingress and egress PE routers.

Based on this information, ingress PE routers perform AIMD window flow control for each VPN flow, and adjust the number of packets sent from every VPN flow. Only window flow control is performed between PE routers, and retransmission and error recovery are not performed. Note that VPN traffic is transferred bi-directionally, so window flow

control must be performed for VPN flows both upwards and downwards.

By performing AIMD window flow control for each VPN between PE routers, inter-VPN fairness is achieved. That is, parameters for AIMD window flow control are set appropriately based on the measured round-trip time and packet loss rate and fairness criteria specified by the IP-VPN service provider (see Section IV-C for details). Thus, the ratio of throughput for VPN flows can be arbitrary controlled as set by the service provider.

Intra-VPN fairness is achieved by simply relying on TCP congestion control operating between end hosts. That is, IP-VPN fairness control itself does not actively perform control to achieve intra-VPN fairness. All of the round-trip times and packet loss rates for flows accommodated in the same VPN are expected to be equal, so sufficient intra-VPN fairness can be achieved simply by TCP congestion control.

Any packet processing such as encapsulation is not performed on packets transferred between PE routers. That is, packets belonging to the same VPN are transferred as-is in the IP-VPN service provider's network. Thus, PE router processing is simplified so that a high scalability for transfer rate and the number of VPNs can be achieved.

B. Algorithm

In what follows, operation algorithm of I2VFC is explained. Refer to Appendix for the pseudo code of I2VFC.

For each VPN flow, I2VFC performs the following operations:

- Obtain feedback information by exchanging management packets
  Management packets are exchanged between the ingress and egress PE routers to obtain information (i.e., packet loss rate p, round-trip time R, and sequence number of successfully received packets) required for I2VFC's window flow control.
  An ingress PE router sends a management packet for each fixed number Δ of data packets sent to the corresponding egress PE router. A management packet carries the VPN identifier, the time when the management packet is generated, and the sequence number of the last data packet sent from the ingress PE router.
  The egress PE router calculates the packet loss rate p for each VPN flow from the received management packet as

$$p = 1 - \frac{\text{(\# of packets received by egress PE router after receiving previous management packet)}}{\text{(\# of packets sent by ingress PE router after sending previous management packet)}} \quad (2)$$

  A management packet sent back to the ingress router carries the VPN identifier, the calculated packet loss rate p, and the highest sequence number of all data packets received by the egress PE router.
  After receiving the management packet, the ingress PE router know the packet loss rate p for each VPN flow and the highest sequence number of all data packets received by the egress PE router. The ingress PE router calculates the round-trip time R for each VPN flows as

$$R = \text{(time of sending the management packet)} - \text{(time of receiving the management packet)}. \quad (3)$$

- Perform AIMD window flow control between ingress and egress PE routers
  An ingress PE router updates window size according to AIMD window flow control and controls the amount of packets sent from each VPN.
  When it receives a management packet, the ingress PE router updates window size w as follows based on the measured packet loss rate p for the VPN flow:

$$w \leftarrow \begin{cases} \min(W_{max}, w + \frac{a\,\Delta}{w}) & \text{if } p = 0 \\ \max(W_{min}, w - b\,w) & \text{otherwise} \end{cases} \quad (4)$$

  Here, W_{max} is the maximum window size and W_{min} is the minimum window size. Normally, the ingress PE router receives w/Δ management packets in a round-trip time. Thus, Eq. (4) means that window size is increased only by a per a round-trip time when the packet loss rate is zero.
  By appropriately configuring the maximum and minimum window sizes, the maximum throughput and the minimum throughput of each VPN flow can be freely specified. For instance, if the maximum window size W_{max} is configured as $T_{max} \times R$, the throughput of each VPN flow is upper-bounded by T_{max}. Moreover, if the minimum window size W_{min} is configured as $T_{min} \times R$, it is guaranteed that the throughput of each VPN never falls less than T_{min}.
  The ingress PE router can send w packets during a round-trip time. To prevent deadlock when management packets are repeatedly discarded in the network, re-transmission control is performed only for management packets using a timeout mechanism. Specifically, a ingress PE router receives no management packet during timeout period T_{out}, after receiving a management packet, it immediately sends a new management packet. Typical configuration of T_{out} would be $4R$ [13]. After transmission, the ingress PE router multiplicatively decreases window size using Eq. (4).
  It should be noted that overhead caused by sending management packets between PE routers is not significant; a fraction of the bandwidth consumed by management packets is given by

$$\frac{\text{(management packet size)}}{\text{(management packet size)} + \Delta \times \text{(data packet size)}} \quad (5)$$

  which is less than 0.5% for 1,500 [byte] data packet, 32 [byte] management packet, and $\Delta = 4$.

C. Achieving Inter-VPN Fairness

The basic idea for achieving arbitrary fairness at a large timescale sufficiently greater than the round-trip time is to adjust parameters of AIMD window flow control for each VPN operating between ingress and egress PE routers.

AIMD window flow control [6] additively increases window size w by a only when congestion does not occur in the network. Otherwise, it multiplicatively decrease window size by $b \times w$. AIMD window flow control is adopted in the TCP congestion avoidance phase, and a large amount of research has been conducted in the literature [6, 14-18].

For example, using a deterministic AIMD model, throughput T of AIMD window flow control in steady state is

approximately given as [14]

$$T = \frac{pa(b-2)+\sqrt{p(b-2)a(pab-8b-2pa)}}{4pbR} \quad (6)$$

$$\simeq \frac{\sqrt{2-b}\sqrt{a}}{\sqrt{2b}R\sqrt{p}} \quad (7)$$

where a and b are parameters of AIMD window flow control: i.e. the additive increase factor and multiplicative decrease factor of the window size. Also, R is the network's round-trip time and p is the packet loss rate.

One can view Eq. (6) from a different perspective. Namely, Eq. (6) indicates that the bandwidth allocation to all flows cannot only be distributed fairly, but also be distributed with an arbitrary ratio by using AIMD window flow control. Eq. (6) means that throughput of flows can be controlled at an arbitrary value if parameters a and b are configured appropriately according to the network's round-trip time R and packet loss rate p.

Let us consider fairness among multiple VPN flows. Parameters a and b of AIMD window flow control for the i-th VPN flow are respectively denoted by a_i and b_i. Similarly, throughput of the i-th VPN flow is denoted by T_i, and its round-trip time and packet loss rate are denoted by R_i and p_i.

First, we focus on the fairness between VPN flow i and VPN flow j. By letting $R_i/R_j = \gamma$ and $p_i/p_j = \delta$, according to Eq. (6), the ratio η for T_i and T_j is given by

$$\eta = \frac{T_i}{T_j} \quad (8)$$

$$\simeq \sqrt{\frac{a_i b_j(2-b_i)}{a_j b_i(2-b_j)\gamma^2\delta}} \quad (9)$$

Thus, arbitrary fairness can be achieved by setting a and b so that η takes the desired value. However, there are infinite combinations of a and b that satisfy Eq. (1). One of those combinations should be chosen by considering the transient performance of AIMD window flow control [13, 15].

For configuring parameters a and b to achieve arbitrary fairness, round-trip time R_i and packet loss rate p_i must be known for all VPN flows. In the proposed I2VFC, by exchanging feedback information between PE routers, round-trip time R_i and packet loss rate p_i for each VPN flow are measured. Note that the round-trip time and packet loss rate between PE routers are necessary, not between end hosts.

D. Achieving Intra-VPN Fairness

Intra-VPN fairness is achieved by simply relying on TCP congestion control operating between end hosts. That is, IP-VPN fairness control does not identify each flow accommodated in the VPN.

More than 90% of the Internet traffic is sent by TCP, so that relying on TCP congestion control operating on end hosts would be reasonable. Since the TCP congestion avoidance phase adopts AIMD window flow control, when the round-trip time and the packet loss rate are equal among different TCP flows, the bottleneck link bandwidth is expected to be shared fairly. The round-trip time and packet loss rate are expected to be equal for all flows accommodated in the same VPN, so intra-VPN fairness can be achieved simply by TCP congestion control.

However, a flaw of this method is that fairness between traffic of different protocol types accommodated in the same VPN cannot be achieved. For instance, when both TCP and UDP traffic share the bottleneck link, I2VFC cannot realize fairness between TCP and UDP traffic. Given the purpose of IP-VPN services, however, strict control of intra-VPN fairness should be performed at customer edge routers managed by customers contracting the IP-VPN service, but it is beyond the scope of this paper. For example, stricter intra-VPN fairness can be achieved by performing priority control for each protocol at the customer edge router.

In I2VFC, AIMD window flow control operates between PE routers, and TCP window flow control operates between end hosts. That is, two independent feedback-based control operate simultaneously between PE routers and between end hosts. As have been pointed out in studies on TCP over ABR [19], there might be a risk of performance degradation due to mutual interference of different feedback-based controls. I2VFC's window flow control avoid such interference of feedback controls by operating at much larger timescale than TCP's one.

The window flow control in the TCP congestion avoidance phase is an AIMD window flow control with additive increase factor $a = 1$ and multiplicative decrease $b = 1/2$. I2VFC sets a and b to much smaller values than those of TCP. This causes that I2VFC modestly changes window size, so two window flow control operate at different timescales. In Section V, through simulation experiments, we discuss how I2VFC achieves intra-VPN fairness by appropriately configuring a and b.

E. Ease of implementation

The proposed I2VFC achieves IP-VPN fairness control by modifying only PE routers. Hence, it can be easily deployed in a service provider's IP network. The proposed I2VFC performs AIMD window flow control between ingress and egress PE routers, so there is no need to modify existing core routers. In addition, intra-VPN fairness is achieved by simply using TCP congestion control operating at end hosts, so there is no need to modify end hosts.

F. Achieving scalability

The proposed I2VFC achieves a high scalability by not identifying each flow accommodated in the VPN at PE routers. Note that it is possible to realize fairness among different protocols by identifying each flow accommodated in a VPN and by applying some congestion control to each flow. However, for identifying each flow accommodated in a VPN, at least transport-layer packet header must be analyzed, leading significant processing burden on ingress PE routers. Moreover, when an IP packet is encrypted using IPsec or other security mechanisms, identifying each flow at ingress PE routers is impossible. On the contrary, with our I2VFC, there is no need to identify each flow accommodated in the VPN at PE routers, so PE routers can operate at a very high speed. Note that I2VFC operates without problem even when the payload of IP packets is encrypted using IPsec or other security mechanisms.

V. Simulation

By presenting simulation results, we demonstrate the effectiveness of the proposed I2VFC. Due to limited space of the current paper, only two simulation results for a network

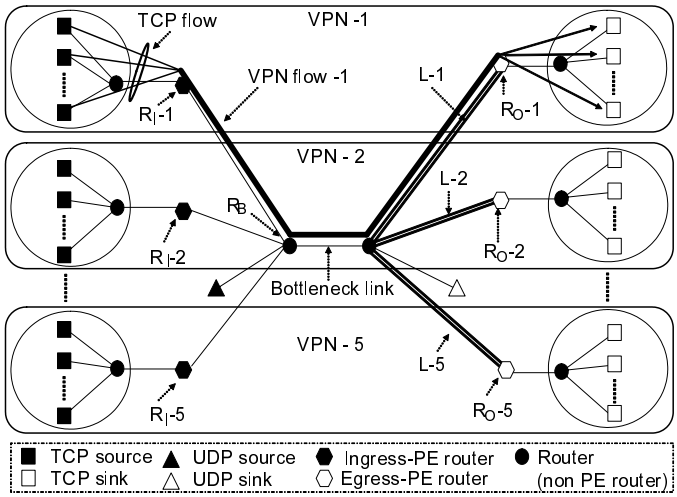

Fig. 2. Network topology used in simulation

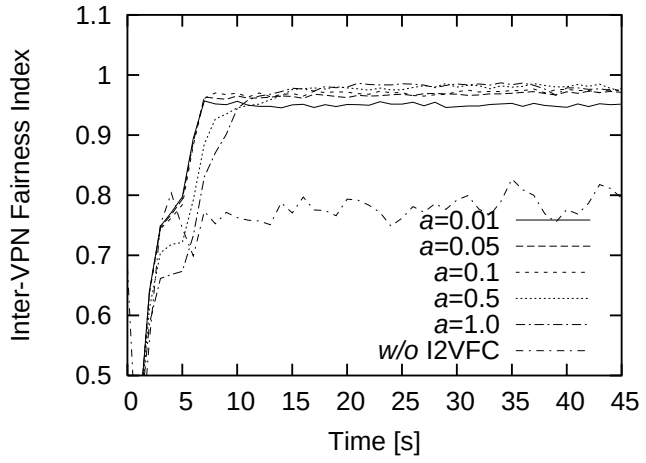

Fig. 3. Evolution of inter-VPN fairness index

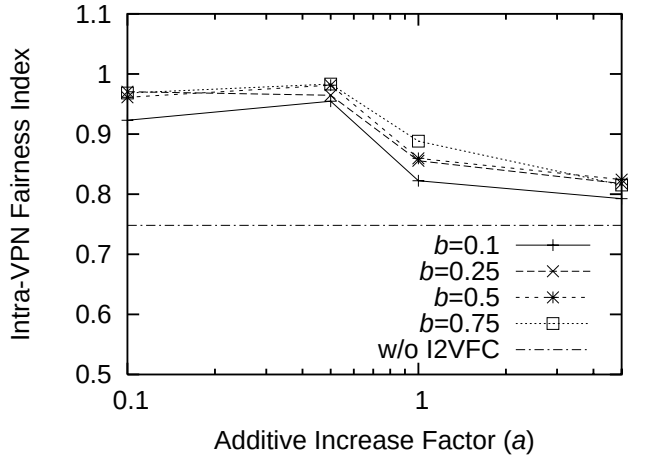

Fig. 4. Weighted fairness index for intra-VPN fairness (two TCP connections in each VPN flow)

topology (Fig. 2) are presented. For detailed simulation configurations and more extensive simulation results, refer to [7].

Data transfer is performed continuously using multiple TCP flows from the sending host to the receiving host starting at t = 0 [s]. There exist five VPN flows, and the weight r_i of VPN 1, 2, 3, 4, and 5 are respectively set to 1, 2, 2, 3 and 4. Propagation delay of links, L-1, L-2, L-3, L-4, and L-5, are respectively set to 0.05, 0.025, 0.075, 0.05 and 0.025 [s]. UDP traffic is generated on the bottleneck link as background traffic. The average arrival rate of background traffic is 30% of the bottleneck link bandwidth and the packet length is fixed at 1,500 bytes. The inter-packet arrival time is exponentially distributed. Unless otherwise noted, the following parameters are used in the simulation: the bottleneck link bandwidth is 50 [Mbit/s], the router buffer size is 50 [packet], there exist 30 TCP flows in each VPN flow, the management packet interval is $\Delta = 4$, and the propagation delay of links except L-1 through L-5 is a very small value (i.e., 5.06×10^{-6} [s]).

Figure 3 shows evolutions of the fairness index F [20] of all VPNs, as a performance metric for inter-VPN fairness. Note that F takes a value between 0 to 1, with $F = 1$ when fairness is completely satisfied and with F close to 0 when fairness is not satisfied. In this figure, the additive increase factors are fixed at $a = 0.01, 0.05, 0.1, 0.5$ or 1.0 and the multiplicative decrease factor b of VPN flow 1 is fixed at $b = 0.1$ whereas the other multiplicative decrease factors b are determined from Eq. (4). For comparison purpose, simulation results without the I2VFC control are also plotted. This figure indicates that I2VFC achieves inter-VPN fairness with extremely high accuracy (i.e., $F > 0.9$).

Moreover, Fig. 4 shows fairness index F for TCP connections in each VPN, as a performance metric for intra-VPN fairness. In this simulation, there exist two TCP flows in each VPN. In this figure, the additive increase factors are fixed at $a = 0.1, 0.5, 1$ or 5 and the multiplicative decrease factors are fixed at $b = 0.1, 0.25, 0.5$ or 0.75 . For comparison purpose, simulation results without the I2VFC control are also plotted.

One can find from this figure that intra-VPN fairness is good in particular when values of a and b are small. Such phenomenon can be explained by the interference between I2VFC's window flow control and TCP's window flow control; i.e., when values of a and b are large (e.g., $a \geq 1$ and $b \geq 0.5$) I2VFC's window flow control interferes with TCP's window congestion control. Note that intra-VPN fairness is improved by introducing I2VFC's window flow control regardless of settings of the additive increase factor a and multiplicative decrease factor b. Such fairness improvement can be explained by disparing congestion at the bottleneck link; i.e., by introducing I2VFC's window flow control between ingress and egress PE routers, the bottleneck link is less congested than the case without the I2VFC's control. Therefore, packets of TCP connections in each VPN flow are less likely to be dropped, leading more stable behavior (e.g., less timeouts) of TCP connections.

VI. Conclusion

In this paper, we have first discussed design objectives of a control for achieving fair IP-VPN services: achieving inter-VPN fairness, achieving intra-VPN fairness, easy deployment in a service provider's IP network, and achieving a high scalability for transfer rate/number of VPNs. We have then proposed I2VFC (Inter- and Intra-VPN Fairness Control) to achieve fair IP-VPN services over existing an IP network.

The core of I2VFC is an AIMD (Additive Increase and Multiplicative Decrease) window flow control that operates on ingress and egress PE routers. The most notable feature of I2VFC is that the IP-VPN service provider can freely specify inter-VPN fairness criteria by utilizing analysis results of AIMD window flow control. In addition, I2VFC can be easily deployed into existing IP networks by simply modifying provider's edge routers. Moreover, we have presented several simulation results, demonstrating effectiveness of I2VFC in realizing both inter-VPN fairness and intra-VPN fairness.

APPENDIX
PSEUDO CODE OF I2VFC

Ingress PE router variables (per VPN flow)

window	Window size of AIMD window flow control
seq	# of data packets sent
ack	# of acknowledged data packets
count	# of data packets sent since the last management packet sent
rtt	Measured round–trip time
a	Additive increase factor
b	Multiplicative decrease factor
Wmax	Maximum window size
Wmin	Minimum window size
Delta	# of data packets between management packets
Tout	Time–out period of management packet retransmission

Ingress PE router algorithm (per VPN flow)

```
initialization:
  seq = 0
  ack = 0
  count = 0

if data–packet–in–queue
  if seq − ack < window          ! window flow control
    send data packet             ! to egress router
    seq = seq + 1
    count = count + 1
  if count >= Delta
    send management packet (now, seq)
    count = 0

if receive management packet (time, ack, loss)
  if loss = 0                    ! additive increase
    window = window + a * Delta / window;
    window = min (window, Wmax)
  else                           ! multiplicative decrease
     window = window − b * window;
     window = max (window, Wmin);
  rtt = now − time               ! measure round–trip time

if now − last–management–packet–received > Tout
  send management packet (now, seq)
  count = 0
  window = window − b * window; ! multiplicative decrease
  window = max (window, Wmin);
```

Egress PE router variables (per VPN flow)

lastseq	Sequence number of data packets recorded in the last management packet
count	# of data packets received since the last management packet received

Engress PE router algorithm (per VPN flow)

```
initialization:
  lastseq = 0
  count = 0

if data–packet–in–queue
  send data packet               ! to ingress router
  count = count + 1

if receive management packet (time, seq)
  loss = (seq − lastseq − count) / (seq − lastseq)
  lastseq = seq
  send management packet (time, seq, loss)
  count = 0
```

REFERENCES

[1] B. Gleeson *et al.*, "A framework for IP based virtual private networks," *Request for Comments (RFC) 2764*, Feb. 2000.

[2] M. Carugi and D. McDysan, "Service requirements for layer 3 provider provisioned virtual private ne tworks (PPVPNs)," *Request for Comments (RFC) 4031*, Apr. 2005.

[3] A. Nagarajan, "Generic requirement for provider provisioned virtual private networks (PPVPN)," *Request for Comments (RFC) 3809*, June 2004.

[4] T. Braun, M. Guenter, and I. Khalil, "Management of quality of service enabled VPNs," *IEEE Communications Magazine*, vol. 39, no. 5, pp. 90–98, May 2001.

[5] R. Callon and M. Suzuki, "A framework for layer 3 provider provisioned virtual pricate networks PPVPNs," *Request for Comments (RFC) 4110*, July 2005.

[6] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, pp. 1–14, June 1989.

[7] O. Honda, H. Ohsaki, M. Imase, J. Murayama, and K. Matsuda, "Scalable ip-vpn flow control mechanism supporting arbitrary fairness criteria — part 2: Simulation and implementation —," in preparation.

[8] I. Khalil and T. Braun, "Edge provisioning and fairness in VPN-DiffServ networks," *Journal of Network and Systems Management*, vol. 10, no. 1, pp. 11–38, Mar. 2002.

[9] A. Sang, H. Zhu, and S. qi Li, "Weighted fairness guarantee for scalable diffserv assured forwarding," *Computer Communications Journal*, vol. 8, pp. 2365–2369, Mar. 2001.

[10] R. Pletka, A. Kind, M. Waldvogel, and S. Mannal, "Closed-loop congestion control for mixed responsive and non-responsive traffic," in *Proceedings of IEEE GLOBECOM 2003*, Dec. 2003, pp. 4180–4186.

[11] H. T. Kung and S. Y. Wang, "TCP trunking: Design, implementation, and performance," in *Proceedings of IEEE International Conference on Network Protocols '99*, Oct. 1999, pp. 222–231.

[12] N. Kavak, "Ericsson's network-based IP-VPN solutions," *Ericsson Review*, no. 3, pp. 178–191, Apr. 2000.

[13] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications: the extended version," International Computer Science Institute, Tech. Rep., Mar. 2000.

[14] S. Floyd, M. Handley, and J. Padhye, "A comparison of equation-based and AIMD congestion control," ACIRI, Tech. Rep., 2000, available at http://www.aciri.org/tfrc/aimd.pdf.

[15] D. Loguinov and H. Radha, "End-to-end rate-based cogestion control:convergence property and scalability analysis," *IEEE/ACM Transaction on networking*, vol. 11, pp. 564–577, Aug. 2003.

[16] F. Baccelli and D. Hong, "AIMD, fairness and fractal scaling of TCP traffic," in *Proceedings of IEEE INFOCOM 2002*, Apr. 2002.

[17] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proceedings of IEEE INFOCOM 2001*, Apr. 2001, pp. 631–640.

[18] M. Vojnovid, J.-Y. L. Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000, pp. 1303–1312.

[19] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, J. Jiang, and S.-C. Kim, "Performance of TCP over ABR on ATM backbone and with various VBR traffic patterns," in *Proceedings of IEEE ICC '97*, June 1997, pp. 11–14.

[20] R. Jain, *The Art of Computer Systems Performance Analysis*. New York: Wiley-Interscience, Apr. 1991.